Summer 2008

Biology-Inspired Approach for Communal Behavior in Massively Deployed Sensor Networks

Kennie H. Jones
Old Dominion University

Recommended Citation
Jones, Kennie H.. "Biology-Inspired Approach for Communal Behavior in Massively Deployed Sensor Networks" (2008). Doctor of Philosophy (PhD), dissertation, Computer Science, Old Dominion University, DOI: 10.25777/gqgc-3061
https://digitalcommons.odu.edu/computerscience_etds/58

BIOLOGY-INSPIRED APPROACH FOR COMMUNAL BEHAVIOR IN MASSIVELY DEPLOYED SENSOR NETWORKS

by

Kennie H. Jones
B.S. December 1977, Christopher Newport College
M.S. May 1981, Old Dominion University

A Dissertation Submitted to the Faculty of
Old Dominion University in Partial Fulfillment of the
Requirement for the Degree of

DOCTOR OF PHILOSOPHY

COMPUTER SCIENCE

OLD DOMINION UNIVERSITY
August 2008

Approved by:

Stephan Olariu (Co-Director)

Larry Wilson (Co-Director)

Irwin Levinstein (Member)

Chunsheng Xin (Member)

Douglas Nark (Member)

ProQuest LLC
789 E. Eisenhower Parkway
PO Box 1346
Ann Arbor, MI 48106-1346

ABSTRACT

BIOLOGY-INSPIRED APPROACH FOR COMMUNAL BEHAVIOR IN MASSIVELY DEPLOYED SENSOR NETWORKS

Kennie H. Jones
Old Dominion University, 2008
Co-Directors: Dr. Stephan Olariu
Dr. Larry Wilson

Research in wireless sensor networks has accelerated rapidly in recent years. The promise of ubiquitous control of the physical environment opens the way for new applications that will redefine the way we live and work. Due to the small size and low cost of sensor devices, visionaries promise smart systems enabled by deployment of massive numbers of sensors working in concert. To date, most of the research effort has concentrated on forming ad hoc networks under centralized control, which is not scalable to massive deployments. This thesis proposes an alternative approach based on models inspired by biological systems and reports significant results based on this new approach. This perspective views sensor devices as autonomous *organisms* in a *community* interacting as part of an *ecosystem* rather than as nodes in a computing network. The networks that result from this design make local decisions based on local information in order for the network to achieve global goals, thus we must engineer for emergent behavior in wireless sensor networks. First we implemented a simulator based on cellular automata to be used in algorithm development and assessment. Then we developed efficient algorithms to exploit emergent behavior for finding the average of distributed values, synchronizing distributed clocks, and conducting distributed binary voting. These algorithms are shown to be convergent and efficient by analysis and simulation. Finally, an extension of this perspective is used and demonstrated to provide significant progress on the noise abatement problem for jet aircraft. Using local information and actions, optimal impedance values for an acoustic liner are determined in situ providing the basis for an adaptive noise abatement system that provides superior noise reduction compared with current technology and previous research efforts.

This dissertation is dedicated to my wife, Linda Jones, for all of her support in this endeavor and to my children for whom I hope for a lifetime of learning.

ACKNOWLEDGMENTS

I would like to thank my co-advisors, Dr. Stephan Olariu and Dr. Larry Wilson, for all of their help and guidance in this research. I would also like to thank the other members of my committee, Dr. Irwin Levinstein, Dr. Chunsheng Xin, and Dr. Douglas Nark, for the time they took to review my work and the contributions they made to this publication.

I would also like to thank Dr. Douglas Nark, Michael Jones, and Tony Parrott of NASA Langley Research Center for their help and guidance in the acoustics research sections of this publication and Kenneth Lodding at NASA Langley Research Center for all of the help he gave me on many parts of this research.

Finally, I would like to thank Dr. Frank Allario of NASA Langley Research Center for his encouragement in my pursuit of an advanced degree and Dr. Jules Lambiotte and Ronnie Gillian of NASA Langley Research Center for their support of this research.

TABLE OF CONTENTS

Page

LIST OF TABLES viii

LIST OF FIGURES ix

Section

1. INTRODUCTION 1

2. RELATED WORK 5
   WIRELESS SENSOR NETWORKS 5
   EMBODIED COGNITIVE SCIENCE AND
      EMERGENT BEHAVIOR 11
   EMERGENT BEHAVIOR IN BIOLOGY 14
   SIMULATION USING CELLULAR AUTOMATA 16
   CURRENT APPROACH FOR AIRCRAFT NOISE REDUCTION 18

3. PRELIMINARY WORK 24
   RANGE OF WIRELESS SENSOR NETWORK APPLICATIONS 26
   BIOLOGICAL APPROACH TO
      SENSOR NETWORK ARCHITECTURE 27
   SIMULATING A SENSOR NETWORK
      USING CELLULAR AUTOMATA 29

4. RESEARCH RESULTS 31
   COMPUTING AVERAGE 31
   DETERMINING MAJORITY RULE 38
   SYNCHRONIZATION 42
   STOCHASTIC WAKEUP 49
   ANALYZING A COMMUNAL SENSOR NETWORK
      ECOSYSTEM 53
   COMMUNAL NETWORK FOR ADAPTIVE NOISE REDUCTION 76

5. CONCULSION 119
   ALTERNATIVE ARCHITECTURE FOR SENSOR NETWORKS
      MODELED AFTER A NATURAL ECOSYSTEM 120
   REAL TIME DETERMINATION OF OPTIMAL ACOUSTIC
      LINER 123
   A RANGE OF COOPERATION IN SENSOR COMMUNITIES 124
   FUTURE RESEARCH 124

BIBLIOGRAPHY 126

Page

APPENDIXES

A. PUBLICATIONS RELATED TO SENSOR NETWORKS 131
B. PREVIOUS PUBLICATIONS 133

VITA 134

LIST OF TABLES

Table Page

1. Vision versus reality 12

2. Tested transceiving ratios 52

3. Symbol Definitions for Theorem 1 55

4. A neighborhood in the impedance domain I 84

LIST OF FIGURES

Figure Page

1. Six adjacent Helmholtz resonators..........20
2. 8-segment liner. ζ = impedance..........21
3. Six adjacent adaptive resonators..........22
4. Illustrating the neighborhood of a cell..........30
5. Initial distribution of colors..........34
6. Distribution of colors after the first iteration..........34
7. Distribution of colors after 10 iterations..........34
8. Total CVC operations required for convergence..........35
9. CVC operations per cell required for convergence..........35
10. Iterations required for convergence..........36
11. Cells within tolerance in time..........36
12. CVC operations for cells within tolerance..........36
13. Initial even distribution of colors..........37
14. Color distribution after one iteration..........37
15. Initial distribution of colors for majority rule..........38
16. Distribution of colors after the first iteration..........38
17. Distribution of colors after 4 iterations..........38
18. Time required for majority rule..........39
19. Transactions required for majority rule..........39
20. Minimum, average, and maximum number of CVCOs..........40
21. Stabilizing time requirement for percentage of cells with majority value..........40

Figure Page

22. Stabilizing transaction requirement for cells with majority value 40

23. Minimum, average, and maximum number of CVCOs 42

24. Percentage of cells with majority value. 42

25. Initial distribution of clock values for TPSN 44

26. 25% of cells are synchronized 44

27. 50% of cells are synchronized 44

28. 75% of cells are synchronized 44

29. Energy expenditure per sensor node in TPSN 44

30. Time required in TPSN 46

31. Initial distribution of clock values for distributed synchronization 47

32. Distribution of clock values after the first iteration 47

33. Distribution of clock values after 10 iterations 47

34. Iterations required for a solution in decentralized synchronization 47

35. Illustrating the minimum, average, and maximum number of CVCOs for distributed synchronization 48

36. Percentage of cells having clock values within tolerance 48

37. Time required to reach consensus 52

38. Energy consumed to reach consensus 53

39. Percent deviation of converged average from average of initial distribution 53

40. Reduction in variance with each iteration 62

41. Reduction in Z with each iteration 63

42. Convergence of algorithm with reducing Z 64

43. Parallel CVCOs 66

Figure Page

44. One iteration requires 25 CVCOs.....67

45. One iteration consists of 34 parallel CVCOs.....68

46. Two iterations optimizing cell participation.....70

47. DRAND TDMA frame for a CVCO.....71

48. TDMA slots required for convergence.....71

49. Transmissions per cell required for convergence.....72

50. Percentage of CVCOs for various neighborhood sizes.....73

51. Percentage of transmissions for various neighborhood sizes.....73

52. TDMA slots required for convergence of various neighborhood sizes.....74

53. Transmissions required for convergence of various neighborhood sizes.....75

54. Transmissions per cell required for convergence of neighborhood sizes.....75

55. Rectangular duct for analysis.....77

56. Attenuation values for uniform liners of impedance values in domain **I**.....83

57. Adapting attenuation from 0.6-1.0*i* to 0.9-0.8*i*.....86

58. Segmented liner.....91

59. Maximum attenuation for uniform liner at selected frequencies.....94

60. Power levels for initial neighborhood surrounding impedance, 1.1-1.1i.....95

61. Adapting to best uniform liner for 1500 Hz.....97

62. Adapting to Best Uniform Liner for 1000 Hz.....98

63. Adapting to best uniform liner for 2000 Hz.....98

64. Adapting to best uniform liner for 2500 Hz.....99

65. Adapting uniform liner.....99

Figure Page

66. Delta power level calculations for segments, S_i and S_{15} 102

67. Best uniform liner for 1500 Hz. S_0 is frozen, S_1 is leader 104

68. Global attenuation and delta power for neighborhood centered at 1.3-1.0*i*............ 105

69. Liner configuration for 1st neighborhood of segment, S_1, as leader 106

70. Final segmented liner for 1500 Hz. Global attenuation = 33.5 dB 107

71. Adaptation progression for segmented liner at 1500 Hz 107

72. Final segmented liner for 1000 Hz. Global attenuation = 46.57 dB 109

73. Adaptation progression for segmented liner at 1000 Hz 110

74. Final segmented liner for 2000 Hz. Global attenuation = 23.33 dB 111

75. Adaptation progression for segmented liner at 2000 Hz 112

76. Final segmented liner for 2500 Hz. Global attenuation = 17.33 dB 113

77. Adaptation progression for segmented liner at 2500 Hz 114

78. Adaptation to best segmented liner for 4 frequencies.. 115

SECTION 1

INTRODUCTION

Networks of sensors, actuators, and computers have been employed in numerous applications for several decades[1]. Automobiles and aircraft contain sensor networks to monitor system health and manage performance, mass transit systems augment human control with sensor networks that constantly monitor the location and speed of all traffic to ensure safety and efficiency, sensor networks are used to control environments and provide intrusion detection for buildings and complexes, and sensor networks control complex industrial processes to enhance efficiency of the manufacturing process as well as to prevent disasters. Because of cost, size and other constraints, these networks have been restricted to relatively small numbers of components. The conventional architecture of these networks positions a computer at the apex of a network, where its role is to collect sensor data, process that data, and issue commands to control the process.

The advent of Micro-Electrical-Mechanical Systems (MEMS) technology in the last decade has made it possible to build inexpensive, small, self-powered devices, containing a sensor, a computer, and wireless communication capability [1,2]. These devices are sometimes referred to as *motes* [3] to emphasize their small size and at other times are called *nodes* to emphasize their role in a sensor network. It is expected that massive numbers of these small devices will be deployed in Wireless Sensor Networks (WSN) that change the way be live and work by monitoring and modifying the environment.

[1] This dissertation follows the style of the *Proceedings of the IEEE.*

However, current implementations as well as the research literature have primarily concentrated on establishing networks for collecting information and routing that information to a centralized processor. These implementations rely on a central computer and do not allow massive networks of sensors to achieve their full potential. We believe that biological ecosystems provide valuable models that may be emulated to create scalable designs of sensor networks. These biological systems have led us to view sensor nodes as autonomous entities that cooperate with neighboring nodes, relying on local information and actions to achieve desired global behavior.

This dissertation develops a methodology in which sensor nodes interacting with their environment collectively behave as a *community* in an *ecosystem*. To support this development a simulator has been constructed and used to establish the value of this approach. Scalable algorithms that use only local information and take only local actions were developed and then verified by simulation and analysis. Further, the boundaries and limitations of this approach have been assessed as a part of this research. The result is that, rather than acting as passive leaves on a hierarchical tree, autonomous devices cooperate by sharing local knowledge, adapting to changes in their environment, and achieving global goals by taking local actions based on local information. Finally, we used this methodology to provide meaningful progress on an acoustics abatement problem of interest to NASA.

The major contributions of this dissertation are as follows:

1. Cellular automata are implemented and used to simulate wireless sensor networks, providing a tool for algorithm development and assessment.
2. An alternative to the conventional architecture for wireless sensor networks is explored based on models inspired by biological systems. This perspective views sensor devices as autonomous *organisms* in a *community* interacting as part of an *ecosystem* rather than as nodes in a computing network. Following this perspective we engineer for emergent behavior in wireless sensor networks.
3. Algorithms for average, synchronization and voting are designed, simulated, and analyzed using the methodology described above to demonstrate that global goals can be achieved using local information under local control.
4. Design options are described for wireless sensor systems that range from centralized control to entirely local control. We then exploit an intermediate design to make significant progress on the real world problem of providing optimal noise reduction for jet aircraft engines.

The remaining sections are organized as follows. Section 2 describes related work that inspired this research including a history and current status of the development of sensor networks, concepts and research from Biology that are related to this research, prior work on the use of cellular automata for simulation, and an introduction to the jet engine noise abatement problem. Section 3 describes terminology and concepts to define our biologically-inspired approach to sensor network design and the use of cellular automata for simulation of sensor networks. Section 4 details the results of our work in modeling sensor networks after biological ecosystems. Section 5 presents conclusions and describes future work.

SECTION 2

RELATED WORK

For completeness we present the background material that is pertinent to this thesis. The following subsections describe the history, birth and current status of work in wireless sensor networks including a subsection on the limitations of this prior work, the phenomenon of emergent behavior and its affect in biological communities and ecosystems, simulation using cellular automata, and the current approach for aircraft noise reduction.

Wireless Sensor Networks

The dream of ubiquitous machinery was born of the Industrial Revolution: that machinery could eventually be developed to service man's needs, particularly replacing man's direct participation in physical work for provision of food, clothing, housing, and other necessities. The realization of this dream progressed for several centuries before taking revolutionary leap with the aid of electronic computers that facilitated further automation of machines. Futurists have long predicted that machines would eventually communicate and cooperate with each other to accomplish extraordinarily complex tasks without human intervention [4]. As wireless sensor network hardware technology develops even smaller devices with improved sensing, computing and communicating capability, futurists predict that societies of machines will evolve to be autonomous, cooperative, fault-tolerant, self-regulating, and self-healing [5]. In 2001, The National

Research Council's (NRC) Committee on Networked Systems of Embedded Computers [6] called for increased investment in sensor network research. They viewed the full potential of sensor networks as much more than a sensory collection apparatus but, rather, a network of heterogeneous computing devices pervasively embedded in the environment of interest that *both sense and actuate*. Currently, sensor nodes are energy constrained, greatly restricting their transmission range capabilities as well as their longevity. As a consequence, most of the recent research has concentrated on how to form and execute global, ad hoc, multi-hop networks in which sensor nodes transmit only to their immediate neighborhood.

The low cost of mass producing sensor nodes will open opportunities not available to traditional sensor networks: massive (10,000+) deployment of "smart", communicating sensors nodes will allow higher density in sensor networks as well as networks that cover larger areas. As an example, geographical areas, such as forests and grasslands, could be monitored at the square meter, or even sub-meter level to provide meteorological and environmental information in a cost efficient manner. In the next subsections, we describe early research in the field, progress that has been made, and limitations of the dominant architectures that have been used or proposed to date.

History and Current Status of Wireless Sensor Networks

In the late 1990's DARPA sponsored the *SmartDust* program [1,7] with the goal of developing smart self-powered sensing devices with communication capability. These devices were to be small and inexpensive enough to be used in massive numbers to gather information for a variety of applications. To demonstrate the potential of this program [8], six sensor nodes were dropped along a road from an Unmanned Aerial Vehicle (UAV). These nodes self-organized by synchronizing clocks and forming a multi-hop network. They magnetically detected vehicles and reported the time of the passing. To illustrate larger ad-hoc networks, researchers from UCB and the Intel Berkeley Research Lab demonstrated a self-organizing sensor network formed from autonomous nodes carried by students. Later in the same auditorium, quarter-sized sensor nodes were hidden under each of 800 chairs and, after simultaneous initiation, they formed what was described as, "… the biggest ad hoc network ever to be demonstrated" [9].

Successful application of sensor networks have been reported in areas ranging from smart kindergartens [10,11,12], to smart learning environments[13], to habitat monitoring [14,15,16], to environment monitoring [17,18,19,20], to greenhouse and vineyard experiments [21,22], to helping the elderly and the disabled [13], among many others. These prototypes provide solid evidence of the usefulness of sensor networks and suggest that rapidly expanding applications will lead to pervasive sensor networks that will fulfill the promises of visionaries [5,6].

However, it would be expected that even more progress would have been made in the decade since the birth of sensor networks especially since the hardware devices have made great strides in computing power, memory, and wireless communication range and bandwidth [23,24,25,26]. A major concern is that the applications all employ only a few devices: massive deployments have not materialized. Crossbow Technologies is a major producer of the nodes used in sensor network, but a snapshot of their web site [27] describing "Customer References" in March 2008 revealed the following:

- Prior to 2005, the U.S. Army implemented a MICA2 mote-based counter-sniper system: acoustic sensors detect a weapon shot and send acoustic timing via a meshed network to a central processor which triangulates data from several motes to locate a sniper [28].
- In 2005, Sandia National Labs implemented a sensor network to apply Controlled Environmental Agriculture (CEA). Efficiency in growing livestock forage in arid regions is increased by motes monitoring moisture. Motes collect data in Mexico and route it to a server in California for analysis [29].
- In 2006, BP used motes to perform Condition-Based Maintenance (CBM). Motes monitor critical rotating machinery, such as the pumps and motors, and measurements are passed through a gateway to a central controller.
- In 2006, Desert Mountain Golf Courses implemented a sensor network to monitor moisture, temperature, and salinity locally and route data through a gateway to a central controller for analysis [30].
- Life Fitness, Inc. attached motes to fitness equipment to provide information about usage. Motes are also deployed to implement a virtual trainer. Data collected is sent to a central controller for processing [31].
- In 2007, Cornucopia Tool & Plastics deployed motes on machines for

Enterprise Resource Planning (ERP). Data is sent to a gateway that is meshed to central controller [32].

These applications all use a small number of nodes to *sense and send* data through a multi-hop network to a central processor for analysis. In 2002, Wokoma et al. [33,34,35] realized that a centralized approach will not be sufficient for many applications and introduce the possibility of emulating the behavior of ant colonies instead. In 2005, Britton et al. discussed the advantages of designing some classes of sensor networks modeled after biological-like systems, in both structural design characteristics and operational processes. They argued that such designs lead to sensor network systems that are self-organizing, scalable, and robust. While they present sound arguments for an analogy between living systems and sensor networks, it is not clear how their system architectures or applications use the biological model or significantly differ from classical sensor networks.

Some researchers have begun to investigate the wireless sensor network capability for interacting with the environment as well as sensing. New terminologies were developed in [36], Wireless Sensor Actor Networks (WASN), and in [37], Sensor Actor Network (SANET). They employ *actors* or *actuators* that take immediate, simple, action in response to sensory input. However, the architectures they developed were sense and send with the central control.

Limitations of Sense and Send

The successes described above demonstrate the enormous value of sensor networks. They share a number of similarities in that most use nodes with capabilities similar to the UCB-Intel large-scale demonstration [9]: networks with homogeneous nodes and PCs serving as sinks to collect data from the sensors and to control necessary actions. Note that, other than WASNs or SANETs, all are passive, sensing systems (i.e., no actuators) observing the environment and reporting these observations to a central authority where decisions are often made by human observers.

However, the use of sensors to measure parameters and deliver data to a computer is not new. What was novel is the wireless networking as well as the small size and cost of the sensors. These features allowed easy and inexpensive placement of nodes unobtrusively into the environment.

Note that the research and experiments described above employ conventional computing networks under centralized control and involving a small number of sensor nodes (between 6 and 800). They do not approach the massive deployment envisioned for sensor networks [1,5,7,22].

In 2004 [38], we wrote of concerns that sensor networks could not reach their full potential using techniques common within the research community. We noted that researchers are adapting methods from conventional computer networks to the new energy constrained, wireless environment of sensor networks but, that these methods will not scale to massive networks. Sensor nodes have the potential to both sense and act autonomously. There is no inherent need to subject them to the inefficiencies and limitations that come with centralized control. In the same vein, Ian Akyildiz, Director of the Broadband and Wireless Networking at Georgia Institute of Technology, dedicated his keynote speech at the 8th International Symposium on Modeling, Analysis and Simulation of Wireless and Mobile Systems in October 2005 [39] to concern for what he called a "paper writing race." He is concerned that, in haste to get recognition in this promising area of research, too many people are "tweaking known classical communication protocols ... [and] ... topology and power management schemes." He concurred with the view of our paper that these centralized control systems cannot be sufficiently adapted to realize the full potential of sensor networks; new techniques must be developed.

The significant discrepancies between the potential of wireless sensor networks and the current research and development are summarized in Table 1.

Embodied Cognitive Science and Emergent Behavior

Cognitive science, the interdisciplinary study of mind, intelligence and behavior, has lately recognized the importance of interaction with the environment and behavior emerging from the combined interactions of component subsystems. Influenced greatly by the development of electronic computers, behaviorists proposed that all behavior can

Table 1
Vision versus reality.

Visionaries Promise:	***Reality:***
Massive deployment	Most implementations under 50 sensor nodes. Most simulations under 300 sensor nodes
Unlimited capabilities	Sensor nodes only observe and report. Cooperation only under centralized control for routing, data aggregation, etc.
Distributed control	Most implementations use centralized control
Energy efficiency: sensor nodes must sleep whenever possible.	Implementations either have all sensor nodes listening when not transmitting, or depend on TDMA which requires strict, global synchronization
New paradigm	Most techniques are adaptations of existing technologies

be understood by the language of information processing. Lachman, Lachman, and Butterfield [40] stated that behavior can be modeled by "how people take in information, how they recode and remember it, how they make decisions, how they transform their internal knowledge states, and how they translate these states into behavioral outputs." This conveniently aligns with computer simulation where thinking is modeled as a computer program having input, data structures, computation, and output. Pfeifer and Scheier [41] described this as the *cognitivistic paradigm* or *functionalism* that formed the basis of *informational processing psychology*.

While this model served well for simulated problem solving, culminating with the chess championship of IBM's "Deep Blue", it proved less than adequate for modeling simple behavior even a child can perform, particularly those requiring interaction with the environment (e.g., identification of colors and shapes, depth perception, navigation, etc.). Pfeifer and Scheier [41] promote a new model of learning they call, *embodied cognitive science* (alias *new artificial intelligence* or *behavior-based artificial intelligence*). They see the cognitivistic paradigm as fundamentally flawed by our own introspection and propose that we should abandon that approach in order to focus on interaction with the real world. This leads to the concept that behavior *emerges* from the interaction of an organism with its environment. There is general agreement among biologists and researchers of complex systems in the concept of *emergent behavior*: that in complex systems, global behavior of a system results, or *emerges*, from evolution and interaction of their constituent parts. The behavior of an individual, driven by feedback from the local environment and interactions with numbers of other individuals, produce a system-wide behavior not predictable from the individual behaviors.

Social homeostasis [42], hive-minds [43], swarming and similar behavior associated with large communities of social insects share a common thread: the intelligence that these behaviors seem to exhibit is not the result of a cognitive process within individual members of the community. The intelligence of the swarm is emergent and results from the interactions of many thousands of autonomous individuals. Each individual follows its own set of rules and reacts to local state information. Individuals are connected within their immediate neighborhood and share information, but they do not have a central controller with which to communicate.

Although there is agreement on what *is* emergent behavior, there are significant differences in opinions on how to *engineer* complex systems for *desirable* emergent behavior. Behavior will emerge when massive numbers of agents are released to follow simple rules, regardless of how the rules are formulated. But the debate is how to develop interactions among components to produce desirable behavior from complex systems: how can a designer anticipate, predict, and control emergent behavior? That is, how does one *engineer* for emergent behavior.

Most research in embodied cognitive science and emergent behavior is related to intelligence in robotics and automated systems. Sensor network research is, for the most part, ignoring the application of this research, choosing rather to treat sensor networks as conventional computer networks with special constraints. However, sensor networks, with massive numbers of autonomous components interacting directly with their environment and confined to direct communication only with neighbors, is a powerful application that can benefit from the concepts of embodied cognitive science and emergent behavior.

Emergent Behavior in Biology

In Biology, an *organism* is an autonomous living entity. Organisms are autonomous in that they make their own decisions about interactions with their environment. However, they are not independent as they cooperate with other organisms. An *ecosystem* is defined as a *community*, a collection of *organisms*, interacting with one another and their environment and interconnected by an ongoing flow of material and information exchange. Organisms react to and operate on the environment autonomously and asynchronously, yet they can cooperate with local neighboring organisms. The amount of, sophistication of, and motivation for cooperation varies immensely from simple unicellular organisms up through multicellular organisms to populations that form great societies.

Organisms may simplistically be modeled as finite state machines whose genetic material defines an initial state and rules for interaction with other organisms and the environment. State conditions and/or the rules may or may not change as the organism interacts with the environment and neighboring organisms. An organism may "learn" by remembering its interaction with the environment in a local memory. This learning implies some level of cognitive capability.

Biological ecosystems provide excellent examples of how global behavior of a large, complex community can emerge from the combined behavior of individual community members. It is important to note that the behavior of individuals often occurs without those individuals knowing all information available within the community and the individuals may not be aware of any goal of the community or the resulting global behavior. That is, there is no central control of individual behavior. That is, there is no central control of individual behavior and there may not be any global awareness on the part of the individual. That is, there is no central control of individual behavior and there may not be any global awareness on the part of the individual.

We will illustrate with one example of an ecosystem chosen from many possible candidates. A widely studied example of cooperation of a community in an ecosystem is found in fireflies where, in the most studied species, many thousands of males gather on a tree flashing independently. Within hours, the community of the fireflies blink in unison. The organisms follow genetically encoded rules through a progression that results in synchronization. Early theories suggested the thousands of fireflies were following a leader [44,45] but these theories are no longer accepted. Buck [46] proposed leaderless methods whereby each individual synchronizes with its local neighbors. As all do this, the large community eventually synchronizes. Camazine et al. [47] conclude that individuals cannot be synchronizing by directly observing the current flash of any neighbors. It appears that, rather than directly following the flash of any neighbor, each firefly is adjusting its triggering level with each flash until it is synchronized with the neighborhood. Buck et al. [48] compare this behavior to studies of human synchronization. When a group is asked to clap following a leader, the closest followers were no better than 150 ms behind. Yet when asked to clap in unison, the slowest members lagged no more then 130 ms. behind the group. This biological behavior, where global action emerges from local interaction of individuals in a large group, inspired ideas for synchronization in massive sized sensor networks.

Simulation Using Cellular Automata

Cellular automata have been used successfully for several kinds of simulations that are detailed in this section. All of these simulations assign rules that, when followed by individuals, result in emergent behavior of the system. Study of these applications revealed the possibility of using cellular automata to simulate sensor networks where autonomous actions of sensor nodes could result in emergent behavior of the sensor system.

Mitchell et al. [49] used cellular automata to study the problem of one-dimensional density classification. They applied by genetic algorithms in an attempt to find a rule set that uses only local information to determine if an initial configuration contains a majority of 1s or 0s but concluded that it was not possible to find such a rule set. Epstein [50] also used cellular automata to investigate a similar one-dimensional problem, the evolution of norms and the conformance of individuals to the developing norm. In 2005, Cunha et al. [51] used cellular automata to simulate sensor networks for algorithm evaluation. They recognized a sensor network as autonomic, requiring much cooperation among nodes to execute global tasks. Their simulation seeks to use the minimum number of active nodes required to fully monitor a region at all times. Note that their simulation size was up to 1000x1000 cells (1 million sensor nodes), much larger than most other sensor network simulations to date.

In 1970, John Conway developed the *Game of Life* [52] to resemble real-life processes. Conway developed rules whereby cells appear to be born, indicated by setting the cell color to white, or die, indicated by changing the cell color from white to black. When the rules are applied quickly in a computer simulation, patterns appear animated as groups of white cells appear to move around the grid. Epstein [53] proposed a similar game where the rules are based on food supply and energy consumption. Cells represent both the environment (e.g., an initial amount of energy in sugar), and possibly an organism. Each organism is endowed at birth with an initial energy level (an amount of sugar in its cell), an energy consumption rate (the cost in sugar to *move*), a vision quality (how many cells can be *seen* in any direction) and a set of rules (e.g., look in all directions and move to the cell that has the most sugar). No two organisms can occupy the same cell simultaneously, so the first to arrive wins the sugar in that cell. Not surprisingly, the simulation mimics evolution with the strongest organisms surviving. As with the game of life, simulated organisms appear to move around the grid, but the cells do not move. These games display emergent behavior, which motivates us to design wireless sensor networks where global behavior emerges from the execution of simple rules within local neighborhoods.

Current Approach for Aircraft Noise Reduction

Anyone who has stood near a modern jet aircraft is aware of the noise produced by the engines. Further, some neighborhoods are acutely aware that the engine noise level is not constant: it increases dramatically during takeoff and landing. This has led to potentially dangerous noise abatement flight paths with steep turns and rapid altitude changes that are frequently required to minimize the noise for certain neighborhoods. Further, some airports charge airlines based on quantity of noise “pollution” created by their aircraft. Thus, it is desirable to *attenuate* the noise by maximizing absorption under changing conditions.

Obstacles to Noise Attenuation

Noise abatement in a jet engine is accomplished by the liner of the engine nacelle. As sound propagates through the nacelle, noise is reduced, or *attenuated*, by the liner. The amount of *attenuation* is determined by design characteristics of the liner. Designing liners for optimal noise attenuation is complicated by the following:

- The aeroacoustic environment changes throughout the flight regime with large discontinuities in sound frequencies between takeoff, cruising, and landing.
- The theoretical parameters specified in the design may not be met in the construction of the liner material.
- Physical changes over time in the nacelle or the liner (e.g., contamination) change the absorption properties.

Thus a design needs to account for a variety of changes during flight.

Current Technology: Passive Helmholtz Resonators

The *Helmholtz resonator* [54] works by changing incident acoustic energy into thermal energy as a result of resonance. Resonator configurations are used to accomplish acoustic absorption in liners of aircraft engine nacelles. A typical nacelle liner lies beneath the sidewall surface of the nacelle and contains a honeycomb core with a rigid backplate covered by a porous faceplate. The cells of the honeycomb are resonators (see

Fig. 1). This design results in liners containing thousands resonators.

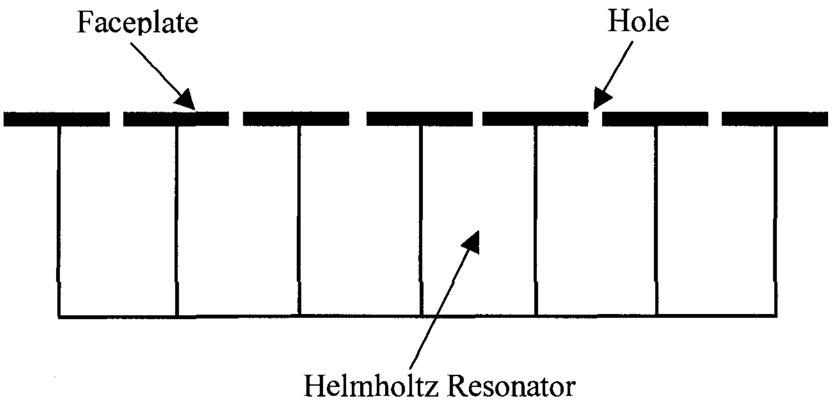


Fig. 1. Six adjacent Helmholtz resonators.

The purpose of the resonators is the attenuation of sound as it proceeds through the nacelle. Each resonator absorbs as much sound as possible. The ability of a resonator to reduce sound is determined by its *impedance*. The impedance is a complex quantity specifying the resonator's ability to absorb sound (the real component, resistance) and reflect sound (the imaginary component, reactance). Resonator design determines the impedance and a particular impedance results in different absorption for different frequencies. Current liner designs have uniform impedance for all resonators. As no impedance can be optimal for all acoustic conditions encountered during flight, a uniform liner is at best a compromise, resulting in acceptable, though not optimal, noise attenuation throughout a flight regime.

A Case for a Segmented Liner

Nacelles of modern airliners incorporate a liner of uniform impedance. Mani [55] demonstrated experimentally a significant improvement in attenuation by circumferentially *segmenting* the liner where segments are designed to have different impedances. Howe [56] and Watson [57] corroborated this result in theoretical studies.

Watson [58] extended the concept of liner segmentation in a theoretical study of what he defined as a *checkerboard* liner. The liner was divided into segments (see Fig. 2). He used a parallel genetic optimization algorithm to assign impedance values for the various segments. The best 4-segment liner could result in a non-uniform liner with 50% improved attenuation compared with the best uniform liner. The best 8-segment liner could improve attenuation by 214% when compared with the best uniform liner. These studies proved that a segment liner design is superior to a uniform liner design.

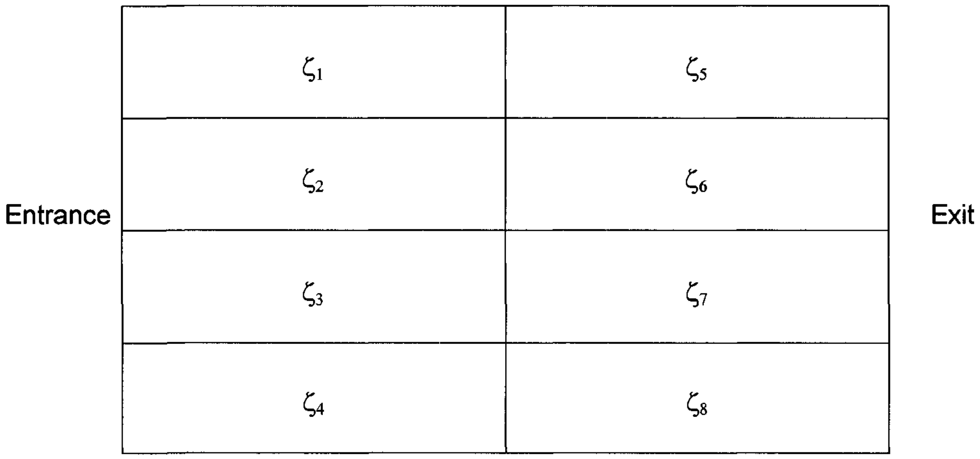


Fig 2. 8-segment liner. ζ = impedance.

An Adaptive Resonator

It would be advantageous to employ liners that are more effective over a wider frequency range. Some researchers [59, 60, 61, 62, 63, 64, 65, 66] have proposed dynamically altering the physical characteristics of a cell to change its impedance. Although this is conceivable, these techniques have been deemed impractical because of high construction and operation cost [67].

Researchers at University of Florida [67,68] have recently developed a tunable, electromechanical resonator. The size and shape of their resonator remains fixed but the *backplate* is composed of a compliant piezoelectric composite surface. The piezoelectric backplate (Fig. 3) can be dynamically changed to alter its absorbing characteristics, thus tuning the resonator for different sound absorption without changing its size and shape. This is a more practical approach in terms of construction and operation costs than techniques that physically change the shape of the cell.

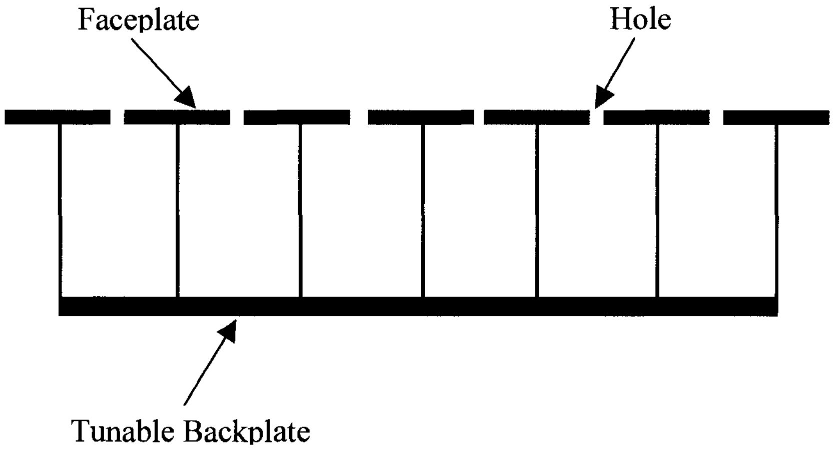


Fig. 3. Six adjacent adaptive resonators.

The challenge, as with all adaptive techniques, is to determine the best achievable tuning for the resonators in real time. As stated, a segmented liner can achieve greater attenuation than a uniform liner. But a static liner design is at best a compromise for all flight conditions. Adaptive resonators provide the capability of not only a segmented liner but of an adaptable segmented liner. Such an adaptive system is described in Section 4.

SECTION 3

PRELIMINARY WORK

Examination of existing wireless sensor network implementations revealed that all have a small number of nodes operating as sense and send to a central controller. Further, they employ communication protocols and other techniques that are adaptations of existing computer network protocols. While these work for the subject applications, they do not move toward the promises that massively distributed wireless sensor networks will provide ubiquitous and pervasive computing to autonomously control environments.

To realize the vision and full potential of sensor networks, designs must support:

- *Massive numbers of sensor/actuator nodes*: large numbers of sensor nodes are necessary to provide the high interaction per area and large area coverage required by many applications. Systems must be designed with many different kinds of nodes performing diverse functions of sensing and acting, yet cooperating for global behavior.
- *Autonomous and intelligent sensor/actuator nodes*: these systems will become too large and complex for centralized control. Acting autonomously, intelligent sensor nodes will need to make decisions and take action based on local information. From the combination of these local decisions and action, global behavior will emerge.
- *Sensor networks working as a cooperative community*: although these

> intelligent sensor nodes will act autonomously, they will not act independently. Instead, they will cooperate with neighboring sensor nodes.

Massive numbers of intelligent sensor nodes acting autonomously, yet collaborating within their neighborhoods, will result in self-sustained communities of machines able to adapt to changing environments. The great challenge ahead of us is to design for intelligence, autonomy, and cooperation, and engineering for *emergent behavior* (see Section 2) of these complex systems.

A key distinction between sensor networks and conventional computer networks is the extent to which the functionality of individual sensor nodes and the sensor network as a whole is tied to the physical environment. Conventionally, the network serves as a conduit for transfer of bits between computers and the physical location of the computer is seldom of consequence, provided that they are connected to the network. However, a sensor network exists to interact with its environment and the locations of the nodes are paramount to this interaction.

Perhaps wireless sensor network is misleading terminology as it indicates an architecture that is not desirable for all applications. The wireless sensor network community has lately found the term “sensor” to be limiting and begun to employ the paired term “sensor/actor” to more accurately describe the technology. Further, the function of this technology is not to network but to interact with the environment and networking is only one of many tools to facilitate this interaction.

Range of Wireless Sensor Network Applications

The sense and send architecture certainly has been useful in many applications (as described in Section 2) but it is easy to envision other applications where one might have each node using only its local data take actions independent of the data and actions of its neighbors and the rest of the community. There are other applications for sensor systems that will require a design that is somewhere between these two extremes, as we will demonstrate in this thesis, where cooperation is required among neighboring nodes, but it unnecessary to send all information to a central controller. A little terminology will be helpful.

A *global goal* represents the *desired* behavior of a deployment of sensor/actors in an environment, a *global behavior* is a result that emerges from the actions and cooperation of sensor/actor nodes deployed, and a *global view* refers to all information contained by the sensor/actor network including the network connectivity. A *local view* is information known to and actions taken by an individual node. A *neighborhood* is any subset of nodes and a *neighborhood view* is the subset of global information contributing a neighborhood decision. A neighborhood may be a *physical neighborhood* consisting of only nodes that can directly communicate or a neighborhood may be a *logical neighborhood*, a subset of nodes that cannot directly communicate but must rely on a multi-hop route to share information. But, regardless of their connectivity, neighborhoods make decisions and take actions based on a neighborhood view.

It is not a claim of this thesis that any one particular architectural configuration is inherently superior to any other configuration. Rather, it is an observation that designers should exploit the range of options to their advantage. It will be shown in this thesis that global goals may be realized by actions on neighborhood views without collecting and sharing a global view.

Biological Approach to Sensor Network Architecture

Because of the relationship of the sensor/actuators to their environment, a biological *ecosystem* (see Section 2) serves as an excellent model for a massively deployed sensor system. A key aspect of ecological systems is that there is no centralized control over the system. Further, it is not necessary for any single entity to attain all the information of the global view to produce the desired global behavior. Instead, the macro system maintains a balanced and stable state through solely localized interactions. In a similar manner, when viewing sensor/actors and their environment as an ecosystem, it is imperative for the designer to define actions and cooperative relationships among the sensor/actor community necessary to affect their local environments in such a way as to create the desired global behavior of the ecosystem. Similar to the biological ecosystem, a community of sensor/actors will have an effect on their environment which will emerge from the rules and cooperative arrangements among community members. We propose that using only the least expensive neighborhood view to make decisions and interact with the environment in order to achieve the desired global behavior can lead to more efficient algorithms. Following this model, system designers will *engineer for emergent* *behavior.*

The goal of this thesis is to demonstrate that a sensor/actor system modeled after a community of interacting organisms in an ecosystem can provide a more efficient and sustainable system than the conventional sensor network architecture for certain applications. Our designs employ neighborhood decisions from neighborhood views to affect a global behavior. Limiting decisions to those based on neighborhood views is important for two reasons in sensor networks:

- *Scalability*: If all sensor data must be accumulated at a central point, then the communication and data management overhead grow substantially with the number of sensor nodes in the deployment making the overhead for large deployments costly and problematic.
- *Autonomy and local intelligence:* Local decision distributes control, providing a natural redundancy affording robustness: if nodes fail, others will continue to make decisions). As each sensor node is operating autonomously, they may learn differently and adapt differently to their environment just as organisms do.

In Section 4, this biological model is applied to applications where a global goal is desired from a sensor network. Instead of using a sense and send, centralized approach, the systems are designed as sensor communities where autonomous sensor nodes cooperate within neighborhoods and take local actions such that emergent behavior achieves the global goal.

Simulating a Sensor Network Using Cellular Automata

A simulator was built in order to observe the global results that emerge from the individual actions of nodes operating autonomously, yet cooperatively, sharing local information with neighbors, and taking local actions. An analysis of simulators in use by the WSN research community [69,70,71,72] revealed shortcomings in these tools that marked them as inadequate for this task. All were designed following a paradigm of conventional computer networks, were overly complicated for the requirements, and were limited to simulating small networks.

This thesis uses cellular automata to simulate wireless sensor networks in a manner similar to the Game of Life [52] and Sugarscape [53]. A cellular automaton represents, in many ways, a distribution of sensor nodes throughout a geographic region and can be used to define neighborhoods in a natural way. A cell represents a sensor node, as illustrated in Fig. 4. For a one-hop neighborhood, we use the eight neighbor cells that surround each internal cell (border cells have 3 or 5 neighbors). Thus, the regularity of the grid provides a logical indication of physical proximity. Note that neighbors may not be limited to the eight adjacent cells, the two-hop neighborhood would include up to 24 neighbors, etc. The cells will be endowed initially with certain states and rules that determine both neighborhoods and the communication protocols in the neighborhoods.

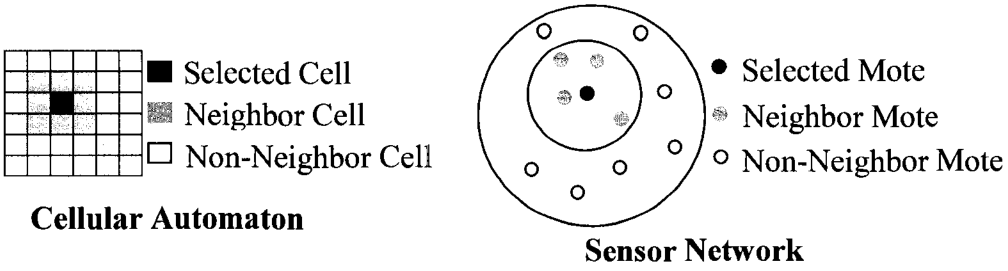


Fig. 4. Illustrating the neighborhood of a cell.

SECTION 4

RESEARCH RESULTS

The following subsections describe algorithms for attaining global goals using only cooperation of physical neighborhoods, analyze these algorithms for several properties and describe algorithms for determining liner impedance for optimal noise attenuation using an adaptive technique that exploits cooperation of logical neighborhoods.

Computing Average

The objective requires the sensor network to calculate an average value across the network without any individual sensor node, or any central authority, ever knowing all the data values, the global view. In this example, the sensor nodes are randomly distributed and each collects a local value of interest. From the local value an average value is to be calculated for the entire network and each sensor node is to learn this average value. The challenge is to attain this goal with all cells requesting/dictating information only from/to their neighbors. In particular, the following evaluation criteria are used to compare this method to conventional, centralized methods in terms of scalability, efficiency, and robustness. Success is positive responses to the following:

1. Can the function be accomplished using only local information and local decisions or must information be collected from the entire network? If

affirmative, the size of the network does not affect decisions and the network is scalable. Also, distributed decisions improve efficiency and robustness.

2. As the network grows, can the task be accomplished without an energy consumption increase for each individual node and is consumption distributed such that no sensor node unfairly assumes more of the burden? If both are affirmative, the network is scalable and efficient in that the longevity of the network is not affected by its size.
3. As the network grows, can the global task be completed without increasing the time required? If affirmative, the network is scalable as the size of the network does not affect the efficiency of task completion.

An informal algorithm to accomplish this task is now described. At deployment, the sensor nodes are endowed with genetic material (see [73,74] for details) that defines: a *status* (*active* or *inactive*), *value*, and a *transmission time*. The status for all nodes is initially active. The transmission time is assigned through a MAC protocol such as a TDMA scheme described by Rhee [75]. The value is assumed to be sensed from the environment. When an active node reaches its transmission time, it initiates an atomic process that is called a *Cell Value Change Operation* (CVCO). The cell initiating the CVCO, called the *leader cell*, begins a series of transactions. It broadcasts a request to its neighbors to participate in the CVCO. All neighbors that are not currently obligated to another CVCO agree to participate by sending their value and agree not to participate in other CVCOs until this one is complete. The leader cell computes the average for the neighborhood. If the leader cell determines that all values in the neighborhood are within a specified tolerance of the neighborhood average, it sets its status to inactive and will not lead another CVCO until it is reactivated. It then broadcast a release message to neighbors ending the CVCO. Otherwise, the leader cell sets its value to the average and broadcasts the average. Participating neighbors set their value to the average and send an acknowledgement to the leader cell ending the CVCO. During this value setting process, if any neighbor is inactive, it is reactivated as it may now have a value out of tolerance with some of its neighbors. The simulation continues until all nodes are inactive. Thus, the simulation ends using only local information; no global control is required.

In the following simulation, assume that a color observed by each sensor node is converted to a real number in the domain range [0, 100]. To simulate this collection upon initialization, values from the domain are randomly distributed throughout the cellular automaton. Following the algorithm described in above, cells begin the process of converting their neighborhood uniformly to its average value, a CVCO for this algorithm. The following results show that evaluation criteria for scalability, efficiency, and robustness are satisfied positively. The first criterion is clearly met in the definition of the algorithm. Only information local to the neighborhood is available and used by cells in the neighborhood. No cell ever directly asks for information that exists outside of the neighborhood. It appears that the average value "ripples" across the network even though communication is confined to localities.

Fig. 5 depicts an initial distribution in a cellular automaton with a 30 x 30 grid of cells showing a random distribution of colors. Fig. 6 depicts the color change after the first iteration (i.e. all cells have led one CVCO). Notice that after a few iterations the cells are much closer to the same color. Fig. 7 illustrates the color change after 10 periods. Soon thereafter, the color differences are indistinguishable to the naked eye, indicating that the value of all cells is approaching the average.

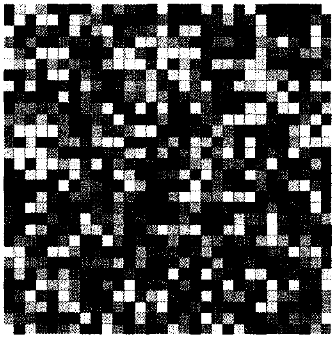


Fig. 5. Initial distribution of colors.

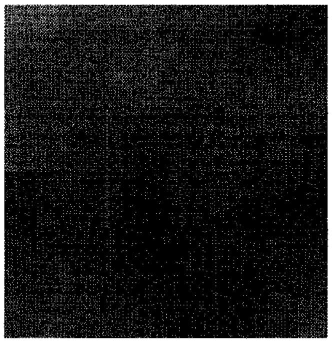


Fig. 6. Distribution of colors after the first iteration.

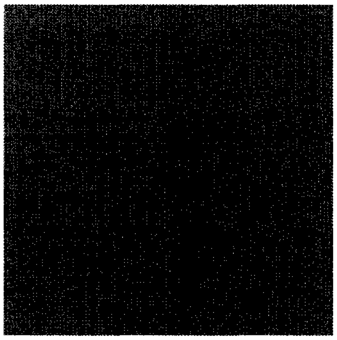


Fig. 7. Distribution of colors after 10 iterations.

A great advantage of this approach is in the distribution of energy expenditure, satisfying the second criterion. The "funnel effect" of multi-hop routing required for the centralized approach described above will deplete cells required for routing much faster as their distance to the sink decreases. Fig. 8 shows that, in this decentralized approach, the total workload for the network increase linearly, but as the second criterion states, this is acceptable since the total energy budget for the network also increases linearly with the size of the network. A much more important attribute is the distribution of the workload among the cells. Fig. 9 show that the workload per cell increases rapidly for small networks but as the network size reaches 300x300, the workload is evenly distributed, requiring between 200 and 300 CVCOs per cell to reach convergence regardless of the network size.

While the second criterion is concerned with the longevity of the network and the prevention of “dead zones” in the network while other zones are still active, the third criterion is concerned with how long it takes to complete a task. Scalability is limited if the amount of time required for a task increases with the increasing size of the network. Fig. 10 shows that the amount of time required for convergence is stable regardless of the size of the network. This is obviously due to the parallel nature of the algorithm: many more CVCOs can occur in parallel as the size of the network grows.

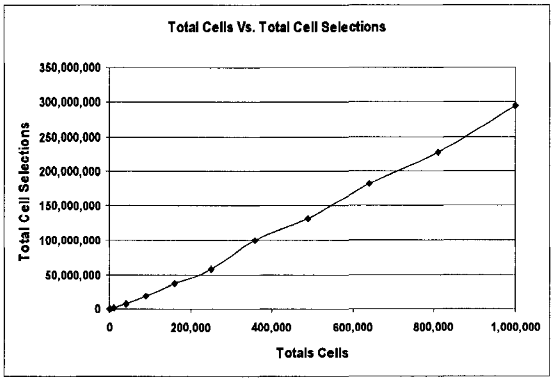


Fig. 8. Total CVC operations required for convergence.

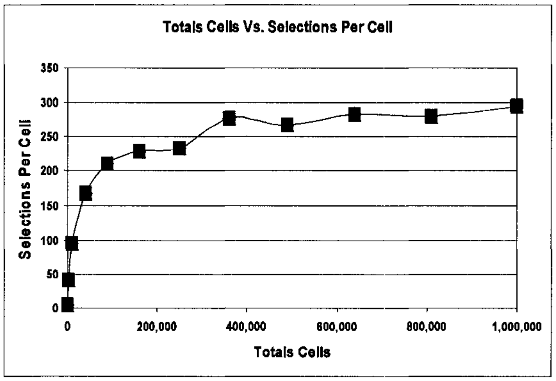


Fig. 9. CVC operations per cell required for convergence.

The distributed algorithm benefits greatly if we are willing to settle for a wide tolerance of values relative to the average instead of insisting on perfection. As stated before, some applications require a close agreement to a common value, while others will be satisfied with looser agreement. In the latter case a distributed consensus within a wider tolerance can occur quickly with relatively few transactions. Fig. 10 shows that most cases, regardless of network size, completed in about 300 iterations. Fig. 11 shows that 68% (1 standard deviation) of the cells are within the tolerance in as few as 22 iterations and 95% (2 standard deviations) of the cells are within the tolerance in less than 100 iterations. Fig. 12 shows a similar progression for CVCOs required to reach a percentage of cells within tolerance.

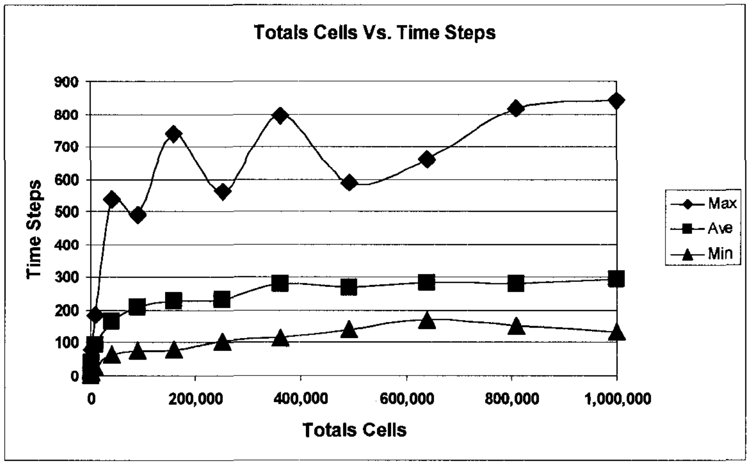
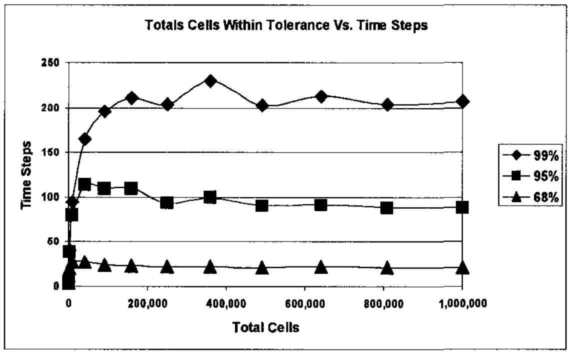


Fig. 10. Iterations required for convergence.



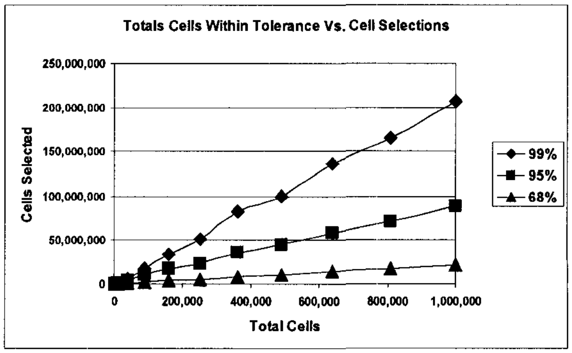


Fig. 11. Cells within tolerance in time.

Fig. 12. CVC operations for cells within tolerance.

More importantly, those cells that are not within the tolerance approach it very closely in very few iterations. As an example, a 200 x 200 grid begins with a color distribution depicted in Fig. 13. All 256 colors are represented in a similar number of cells with a standard deviation of 73.74: 49.6% of colors above the tolerance, and 49.95% of colors below the tolerance.

However, as shown by the visual display of the simulation depicted in Fig. 5, 6, and 7, the algorithm comes close to the solution very quickly. The simulation takes 271 iterations to come to a solution where all cells are within the specified tolerance of 0.5. Fig. 14 shows that the range of colors represented in the distribution after only 1 iteration (i.e., each cell has led one CVCO) has been reduced to 55. The standard deviation has reduced to 6.26. While the majority of colors are still outside of the tolerance, the average offset below the tolerance is only 4.9 and the average offset above the tolerance is only 4.7.

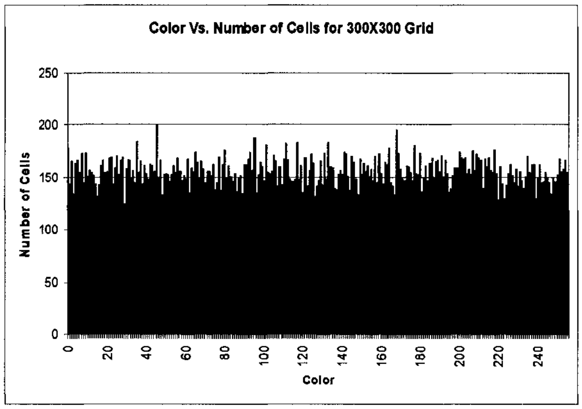


Fig. 13. Initial even distribution of colors.

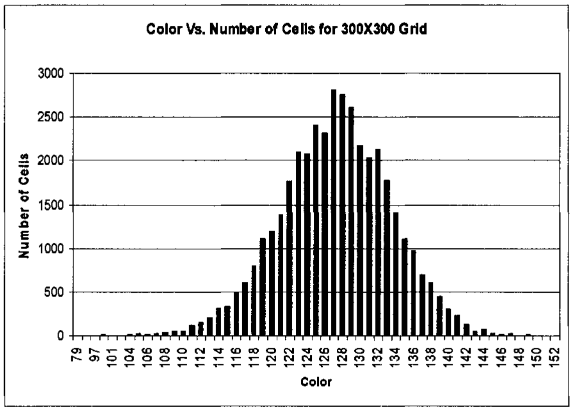


Fig. 14. Color distribution after one iteration.

Determining Majority Rule

Fig. 15 depicts an initial distribution in a cellular automaton consisting of a 30 x 30 grid of cells showing a random distribution of binary values, represented by the colors black and white. Fig. 16 illustrates the color change after the first iteration (i.e. all cells executed one CVCO). Notice that at the end of the first iteration, the cells are grouped in large blocks of the same color. Fig. 17 shows the color change at the end of four iterations. Soon thereafter, all cells are the same color indicating the majority has been determined.

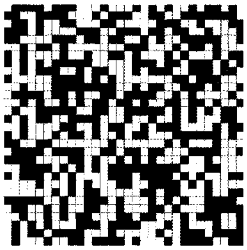


Fig. 15. Initial distribution of colors for majority rule.

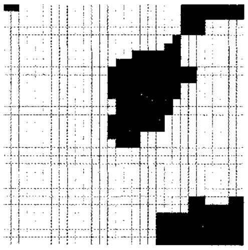


Fig. 16. Distribution of colors after the first iteration.

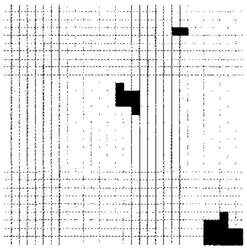


Fig. 17. Distribution of colors after 4 iterations.

Fig. 18 shows that the time required for consensus escalates quickly for smaller grid sizes but after 300x300, the acceleration slows and then stabilizes for massive grid sizes. Fig. 19 confirms that the same is true for the required number of transactions, an indicator of energy expenditure.

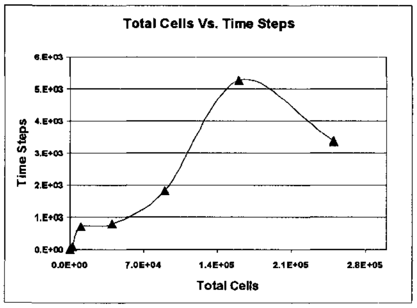


Fig. 18. Time required for majority rule.

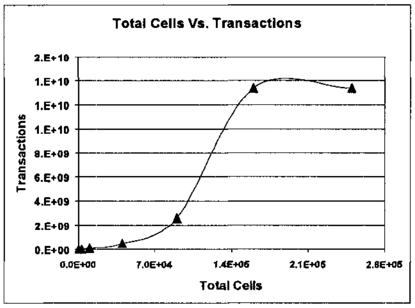


Fig. 19. Transactions required for majority rule.

This algorithm shares the great advantage described for the averaging approach, described in the previous subsection, in the distribution of energy expenditure. In this decentralized approach, the workload is not only evenly distributed, but the workload required of an individual sensor node actually decreases as the grid size increases, as shown in Fig. 20. Each time a cell leads a CVCO, it initiates transactions resulting in energy expense: the fewer the CVCOs, the less energy consumed. As Fig. 20 shows, the cost in energy per cell does not escalate for larger grids.

For some applications, it is not necessary for every sensor node to agree; it may be sufficient for most sensor nodes to be set to the majority value. Fig. 15-17 demonstrate the useful property of this algorithm for such cases, as many cells change to the majority value very rapidly. Fig. 21 shows that about 70% of the cells have the majority value with very few time periods and, furthermore, this time requirement stabilizes for larger grid sizes. Fig. 22 shows that while the number of transactions continues to increase for a final solution, the cost of 70% stabilizes for larger grid sizes. As with averaging, if this is acceptable, it can substantially reduce the cost in time and energy of the calculation.

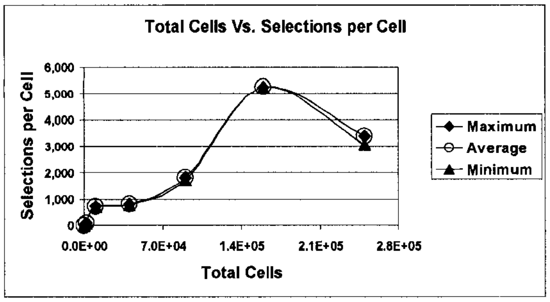


Fig. 20. Minimum, average, and maximum number of CVCOs.

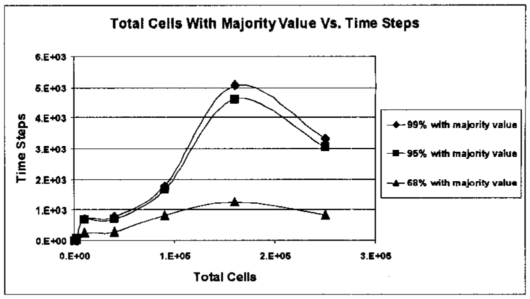


Fig. 21. Stabilizing time requirement for percentage of cells with majority value.

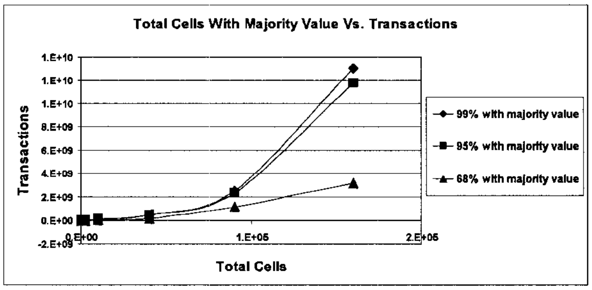


Fig. 22. Stabilizing transaction requirement for cells with majority value.

If carried to completion, this simulation results in the community agreeing on a single switch value. However it suffers from the same problems described by Mitchell et al. [49] and Epstein [50]. In some cases, particularly when there are large blocks within the initial distribution of the same color and especially if the distribution is close to equal, the final configuration may have a majority frequency opposite from the initial configuration, because the selection of cells to lead a CVCO is random and it is possible that initially the cells with minority color near the border of the color boundary are selected, which would turn cells of majority color in that neighborhood to minority color, thus incorrectly flip-flopping the majority-minority color.

The problem is solved by using averaging to incrementally calculate the majority frequency, using a real rather than an integer value. All cells are initially assigned genetic material for the switch value of 0.0 or 1.0. The algorithm proceeds as before, but now, instead of the cell leading a CVCO calculating a binary frequency for the neighborhood, it calculates the average of all neighbors and itself. This real value is stored as the switch value. Whenever the binary value of the switch is required this value is assessed by rounding up or down depending on whether or not the value is larger than or equal to 0.50.

Using this novel approach, the algorithm incrementally averages neighborhoods. Furthermore, the algorithm never breaks the rule that no cell can use any information outside of its neighborhood and the converged average is equal to the average of the initial distribution. Also, the average value of the entire community following each CVCO is also equal to the initial average of the community. This property is proven to be guaranteed in sub-section, "Proof of Consistent Average" (Page 59). If the initial distribution has a majority of 1s, then the initial average will be greater than or equal to 0.5 and if the majority is of 0s, the initial average will less than 0.5. Because the average remains constant throughout the process, the majority at the end will be the same as the majority at the beginning.

The majority rule technique shares many of the same properties as the averaging calculation. Fig. 23 shows similar requirements for CVCOs to determine the majority. Thus, the energy requirement per cell stabilizes for larger grids. Fig. 24 shows that, like the total transactions required for a solution, solutions with a large percentage of cells within the tolerance asymptotically slope towards zero as the grid size increases. Most importantly, solutions up to 95% with the major frequency value reach that asymptote quickly with few transactions regardless of grid size.

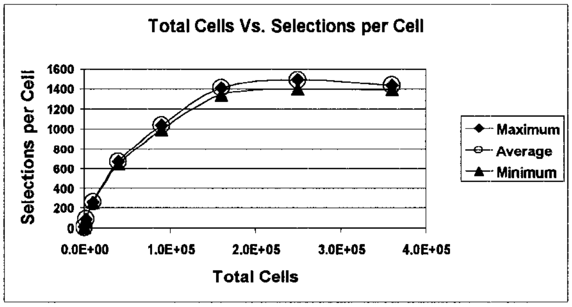


Fig. 23. Minimum, average, and maximum number of CVCOs.

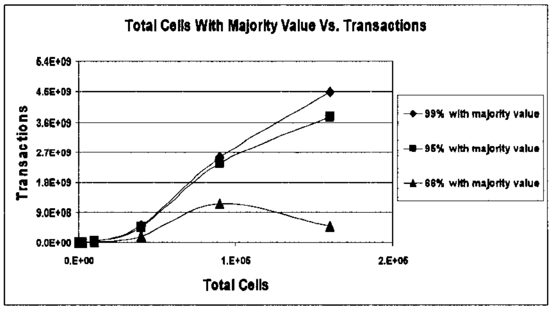


Fig. 24. Percentage of cells with majority value.

Synchronization

Ganeriwal et al. [76] designed the Timing-sync Protocol of Sensor Networks (TPSN) to synchronize all clocks of nodes in a sensor network. They provide a comprehensive explanation as to why this is necessary and details on how their protocol works. Only pertinent details will be provided here. TPSN is designed for Berkeley sensor nodes [77] and is included in the TinyOS distribution [78]. These sensor nodes have a 16 bit clock incrementing every 0.25 microseconds (μs). It is assumed that sensor nodes are distributed with random clock values. The objective of TPSN is to synchronize all sensor nodes with one selected sensor node, the root node. The protocol requires two sequential phases. The discovery phase begins with the hierarchical assignment of levels.

The root sensor node is assigned level 0 and other nodes are assigned a level relative to their position in hops from the root sensor node. The synchronization phase begins with a single sensor node of level 1 communicating with the root sensor node through a series of transmissions using the send-receive protocol to synchronize with the root sensor node. Thereafter, other sensor nodes use the send-receive protocol to synchronize with any other sensor node that has completed synchronization. Thus the process progresses hierarchically across the network.

Fig. 25 depicts an initial random distribution of clock values in the range of [0, 65,535] (normalized to 256 grayscale levels for visual display only). The root node is arbitrarily set to the cell in the upper left corner. Fig. 26, 27, and 28 show the status when 25%, 50%, and 75% of the cells are synchronized.

The designers of TPSN did reduce or eliminate many of the problems associated with centralized control. In fact, TPSN is not centrally controlled, but is only initiated from a particular location. Each sensor node acts somewhat asynchronously and autonomously, negotiating the send-receive protocol with another sensor node known to be synchronized. After neighbor sensor nodes are synchronized, a sensor node may die without affecting the completion of the process. The authors also point out that the error created through the send-receive process is not additive across the network and therefore does not limit the size of the network. Finally, each node is synchronized by interacting with only one synchronized sensor node and thereafter serves as the synchronizing host for at most the set of its neighbors. Thus, as shown in Fig. 29, the energy expenditure per sensor node is bounded and does not increase with the size of the network.

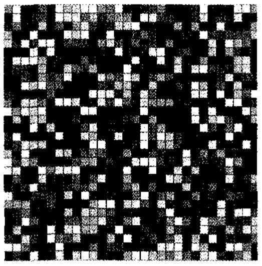


Fig. 25. Initial distribution of clock values for TPSN.

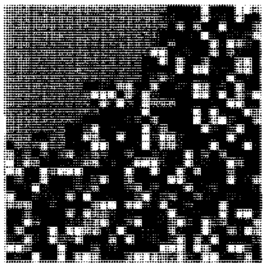


Fig. 26. 25% of cells are synchronized.

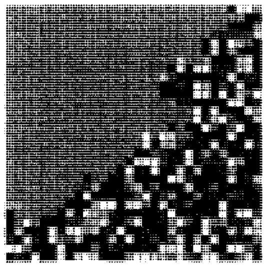


Fig. 27. 50% of cells are synchronized.

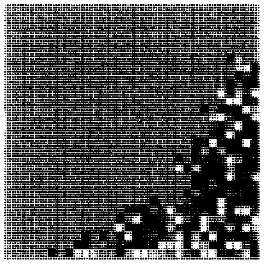


Fig. 28. 75% of cells are synchronized.

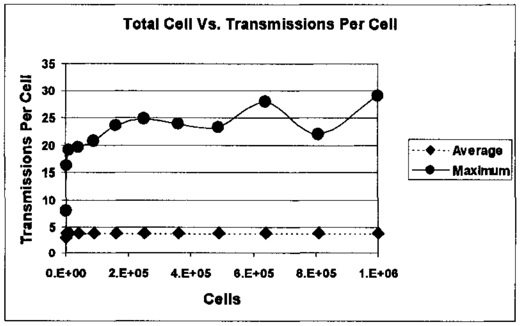


Fig. 29. Energy expenditure per sensor node in TPSN.

TPSN does display some undesirable characteristics of centralized control, suffering from two of its major problems: the requirements of "follow the leader" and "wait your turn." Notice in Fig. 25 through 28 that the process progresses from the top, left corner (root sensor node) to the bottom, right corner. Also notice that cells remain unchanged until it is their turn to synchronize. As the process is hierarchical, some parallel processing occurs, but as shown in Fig. 30, the total time required to reach a synchronized state increases substantially with the size of the network.

Also notice that cells remain unchanged until it is their turn to synchronize. The percentage of synchronized sensor nodes is near linear with time. In the simulation represented by Fig. 25 through 28, the number of iterations required for complete synchronization was 207. At iteration 80 (37% of the time), 25% of the sensor nodes are synchronized. At iteration 110 (53% of the time), 50% are synchronized. At iteration 139 (67% of the time), 75% are synchronized. Perhaps more important, sensor nodes in unsynchronized zones remain far from synchronized as the process proceeds. When 25% of the nodes are synchronized, the average offset from the root sensor node of the remaining 75% is 11,366. When 50% are synchronized, the average offset of the remaining sensor nodes is 10,904, and when 75% are synchronized, the average offset of the remaining 25% is 11,972. The decentralized approach greatly improves these numbers.

Using the averaging algorithm described above, sensor nodes act truly asynchronously and autonomously, yet cooperate, using only local information to make local decisions that affect a global result: synchronization. Upon deployment, the sensor nodes are powered on randomly resulting in unsynchronized clocks. As the algorithm is applied, neighborhoods synchronize their clocks. Eventually the entire network converges on a synchronized clock value.

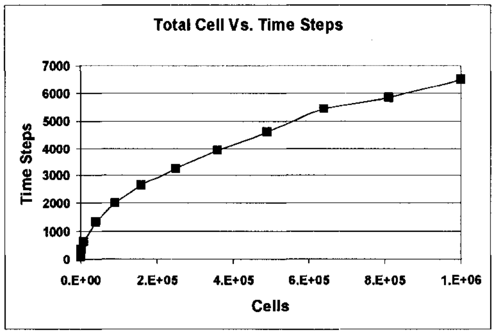


Fig. 30. Time required in TPSN.

Fig. 31 depicts an initial distribution in a cellular automaton with a 30 x 30 grid of cells showing a random distribution of initial clock values. Fig. 32 depicts the clock values after the first iteration (i.e. all cells led one CVCO). Notice that after only one iteration, the cells are much closer to the same time. Fig. 33 illustrates the time change after 10 iterations. Soon thereafter, the time differences are indistinguishable to the naked eye.

A major improvement in this approach is in the scalability of the time taken to reach a solution. Fig. 34 shows that the time to reach a solution does not substantially increase as the size of the network grows.

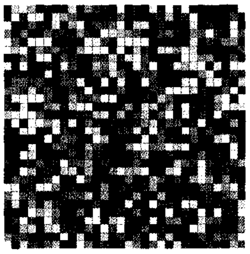


Fig. 31. Initial distribution of clock values for distributed synchronization.

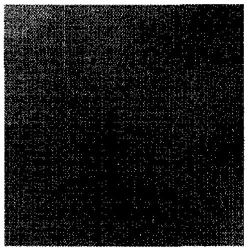


Fig. 32. Distribution of clock values after the first iteration.

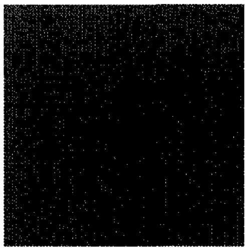


Fig. 33. Distribution of clock values after 10 iterations.

The decentralized approach presented is similar to TPSN in the distribution of energy expenditure. As shown in Fig. 35, the workload is evenly distributed as the grid size increases.

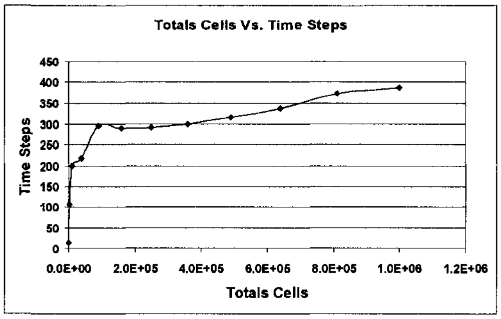


Fig. 34. Iterations required for a solution in decentralized synchronization.

Some applications require a close synchronization, while others may tolerate a much larger difference in clock value. In such cases, a distributed synchronization can occur quickly with relatively few transactions. As an example, assume a difference in clock value of 50 μs can be tolerated from a clock with 0.25 μs increment. A faster clock could exceed a slower clock by up to 200 increments and still remain within the 50 μs tolerance. Fig. 36 shows that solutions with a large percentage of cells within this tolerance are reached much quicker than the exact solution. For a specific example, a network of 1 million sensor nodes is synchronized in 386 iterations. But 99% of the sensor nodes are within the 50 μs tolerance in 207 iterations or approximately half the time for precise synchronization. But more importantly, of the 1% that are not within the tolerance, none are more than 67 μs from the average clock value. Compare this with TPSN: When the solution was half completed, approximately half of the cells were synchronized. The other half were on average 2,726 μs different from the synchronized half.

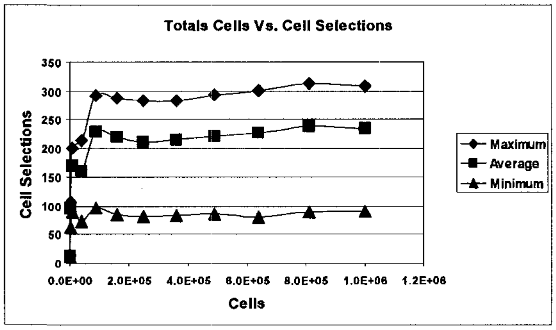


Fig. 35. Illustrating the minimum, average, and maximum number of CVCOs for distributed synchronization.

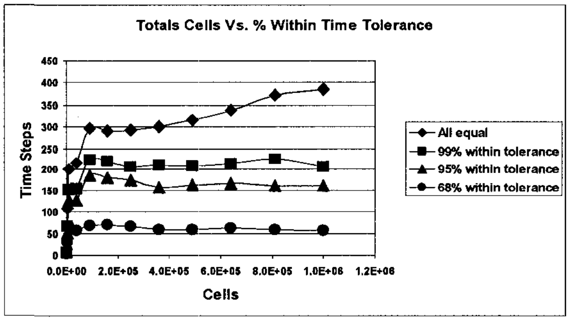


Fig. 36. Percentage of cells having clock values within tolerance.

Stochastic Wakeup

Suppose that instead of asking questions of neighbors and expecting a response, the sensor nodes behave and act asynchronously and autonomously. Each sensor node maintains its own clock independent of every other sensor node. A sleeping period, the length of the receiving time within a sleeping period, and the number of times the sensor node will awaken to transmit during a sleeping period is assigned as genetic material upon deployment. Upon activation, each sensor node decides for itself when it will awaken to transmit and when it will awaken to listen during the sleeping period. Furthermore, each sensor node makes this decision anew each sleeping period. This process requires dynamic irregular selections of transmit and receive times in order to perform well in a probabilistic sense. If fixed choices were made and neighbor receiving and transmitting times are aligned, the process will succeed, but if they are not aligned, the process will fail completely.

Transmissions are brief: a single 8 bit packet representing the current color constitutes each transmission. The MICA2 sensor node transmits at a rate of 40 Kbits/sec. Thus, a transmission burst of a single packet expends 10 mA for 0.2 msec resulting in an energy cost of 5.56E-7 mA-Hr. Receiving periods are assigned in multiples of msec. Thus, any sensor node listening when a neighbor makes a 0.2 msec transmission will hear that transmission. Notice that the clocks do not need to be synchronized. If a transmission spans the msec boundary of the listening sensor node, the received message will be incomplete and will be discarded.

The work done is similar to the averaging process described above. The objective is for each sensor node to gather from its neighbors their color values and average these with its own color. It is different from the averaging process in two important respects. Sensor nodes are never guaranteed to hear from all of their neighbors. A sensor node listens for its receiving period and averages all values it receives during that period with its own current color. Any of its transmissions subsequent to its receiving period will broadcast the new averaged color to any sensor nodes that are listening. Because all neighbors are not queried and set to the new average, the desirable property of the previous algorithm whereby the average of the neighborhood always remains equal to the initial average of the community is lost. But as will be demonstrated, the calculated average is close to the initial average, and in some applications this may be acceptable.

In a sample simulation for a 10 x 10 grid of 100 sensor nodes, each node is assigned the parameter, 20 msec, for duration of the sleep period. To access the optimal values for receiving duration and number of transmissions, these values are varied for each node as in Table 2. The first pair designates that during each sleep period of 20 msec, 1 msec is randomly selected for receiving and during every other msec period, a single transmission is made. Only one transmission can be made during a msec period. However, there can be periods during the sleep period when no transmissions occur as exemplified by the pair, {3,5}. In this case, there are 17 potential transmission periods, but during 12 of these, the sensor node remains sleeping. It would seem intuitive that the optimal results would be obtained by transmitting every period when a sensor node is not receiving but this is not the case. It is of no value to transmit when no neighbor is listening. Because each sensor node reassigns the transceiving ratio anew with each sleep period, it is difficult to predict the results. The simulation ends when all cells have a color value within the specified tolerance of the converged average (i.e., all colors are equal).

Three metrics are of particular concern to reaching consensus: the overall time required, the energy consumed, and the variance between the converged average and the average of the initial distribution. Fig. 37 compares the time required to reach consensus for all tested ratios.

Fig. 38 compares the energy cost to reach consensus for all tested ratios. Ratios with short receiving times performed at substantially lower cost. A number of top performers had ratios with components that add to less than 20, most notably 3-5 and 5-5. Fig. 39 compares the deviation of the converged average from the average of the initial distribution for all tested ratios.

A remarkable property of this approach is the even distribution of cost among the sensor nodes. All sensor nodes expended equal amounts of energy to perform the task. This attribute cannot be overemphasized, as the longevity of the network is maximized.

It is instructive to recall Crossbow's recommendation of a 1:99 ratio of transceiving to sleeping [77]. To test this recommendation, a simulation was executed using a sleep period of 100 with a receiving period of 1 and 1 transmission (a ratio of 49:1). The simulation ran for 12 hours and did not converge. It is understandable why this ratio did not work. If all sensor nodes within a neighborhood rarely transmit and rarely receive, it is unlikely that any will ever communicate. For all of the talk of the requirement for sensor nodes to sleep much of the time to save energy, it is not useful to sleep when there is work to be done. The objective is to design a sleep/transceive ratio that will get the work done as soon as possible with minimal energy consumption.

Table 2

Tested transceiving ratios.

Rec. Period	#Trans.	Rec. Period	#Trans.
1	19	14	6
2	18	15	5
3	17	16	4
4	16	17	3
5	15	18	2
6	14	19	1
7	13	3	15
8	12	3	10
9	11	3	5
10	10	5	10
11	9	5	5
12	8	10	5
13	7	15	3

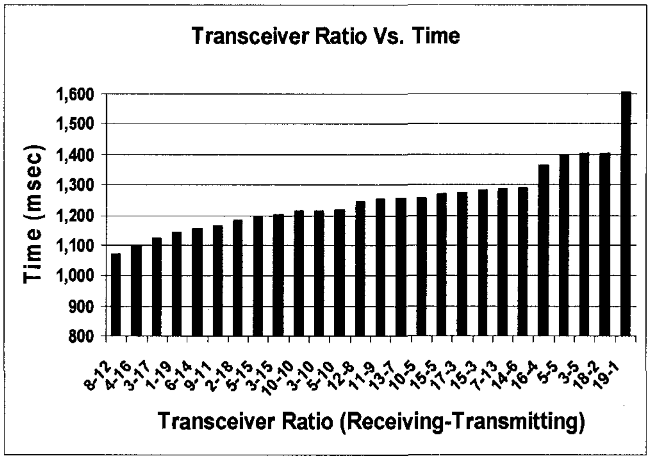


Fig. 37. Time required to reach consensus.

Some of the ratios that produced the best results were not expected. It seems that the best results should occur when all iterations are used for either receiving or transmitting. But this is not the case (e.g., 3-5, 3-10, 5-5). Although the transceiving periods are defined as genetic material, the choice of when these periods occur within the sleep period is made randomly at runtime. Thus, the choice of when transceiving periods occur by a sensor node is considered an environmental influence to its neighbors. This is an example of *emergent behavior* discussed in Section 2.

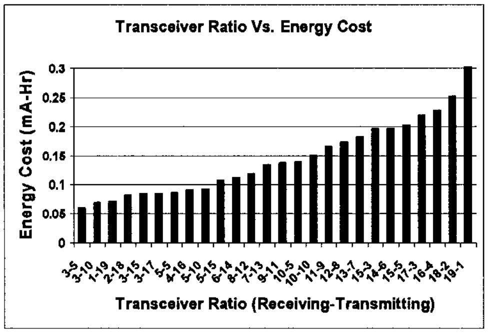


Fig. 38. Energy consumed to reach consensus.

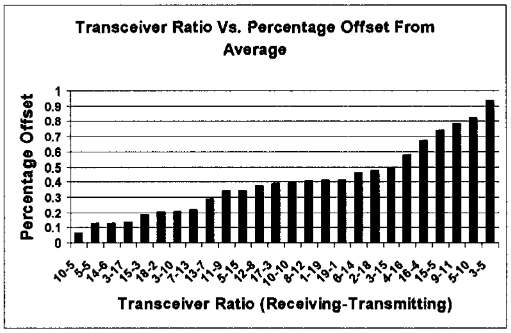


Fig. 39. Percent deviation of converged average from average of initial distribution.

Analyzing a Communal Sensor Network Ecosystem

The simulations presented thus far demonstrate the concept that local decisions based only on local information may lead to accomplishment of a global goal over a large sensor network. All simulations have converged such that all cells have a value within a specified tolerance. In subsequent subsections, properties of these simulations are analyzed. A proof is provided that will guarantee convergence in some finite time. The simulations not only demonstrate that, upon convergence, the value of each cell is within a specified tolerance, but that at each iteration in the simulation, the global average is the same as it was at the beginning of the simulation when cells had a wide range of values. A proof is provided that guarantees the initial global average is preserved with each step in the simulation. While it is proven that convergence is guaranteed in a finite time, the simulations demonstrate that convergence actually occurs in much less time than that guaranteed by the proof. Factors are presented that significantly reduce the time required for convergence compared with that predicted by the proof. While the convergence proof guarantees convergence to a local tolerance, a relationship is specified between the local tolerance and a global tolerance such that a local tolerance can be calculated that, when universally achieved, will guarantee all cells are within a global tolerance of the precise answer. Furthermore, factors are examined concerning implementation in a sensor network, demonstrating the algorithm proven to converge can be practically implemented, while also demonstrating some interesting implementation details.

Proof of Convergence

Here we give a formal proof that the averaging algorithm indeed converges. We define a neighborhood as a cell and all its neighboring cells. The process in which a leader cell computes the average value for the neighborhood and sets all cells in the neighborhood to that average value is described above for the averaging algorithm. For brevity, we define the term Cell Value Change Operation (CVCO) as that process, including all communication to gather neighborhood values and reset the neighborhood.

Theorem 1: Let N denote the total number of cells in the network. Let the cell value (e.g., color) be in the range $[\alpha, \alpha + Z]$. Let ε denote neighborhood tolerance (i.e., a CVCO is not required in a neighborhood if all cells in the neighborhood have a cell value within the range $\left[\overline{X} - \varepsilon, \overline{X} + \varepsilon\right]$, where $\overline{X}$ is the average cell value in the neighborhood). The distributed algorithm discussed above converges and stops with less than $\frac{N \times Z^2}{\varepsilon^2}$

CVCOs.

Proof: By examining the variance of the cell value distribution. We will show that the global variance for the network monotonically decreases toward 0 by more than a fixed amount whenever a CVCO occurs. The variance stops decreasing only when there is no cell needing to perform a CVCO.

We examine a specific CVCO of the algorithm. Suppose cell **A** finds that a CVCO is required (i.e., in its neighborhood having a average cell value, $\overline{X}$, there exists at least one cell having a value, X_i, where $|X_i - \overline{X}| > \varepsilon$). By the algorithm, cell **A** computes the average value of neighbors and itself, and sets the values of its neighbors and itself to this average value. We define the Table 3 for the ensuing discussion:

Table 3

Symbol Definitions for Theorem 1.

M	The mean or the global average of cell values in the network.[1]
L	Number of cells involved in the CVCO, including the operating cell and all of its neighbors.[2]
$X_1, \ldots, X_L$	Cell values of the operating cell and its neighbors.
$\overline{X} = \frac{\sum_{i=1}^{L} X_i}{L}$	Average of a neighborhood, $X_1, \ldots, X_L$
V_o	Global variance before the CVCO.
V_n	Global variance after the CVCO.

[1] Note that we do not need to specify the value of M. It is treated as an unknown in the proof.

[2] In our averaging algorithm, $L = 9$ (including the cell initiating a CVCO and the 8 neighbors). Here, we have generalized the number of cells in a CVCO to make the theorem more general (a later subsection discusses the case that L is not equal to 9).

By definition, V_o and V_n are calculated as follows:

$$V_o = \frac{Y + \sum_{i=1}^{L} (X_i - M)^2}{N} \tag{1}$$

$$V_n = \frac{Y + L \times (\overline{X} - M)^2}{N}, \tag{2}$$

where $Y = \sum_{k \notin \{1,\ldots,L\}} (X_k - M)^2$ is the part of variance contributed by other cells (other than the operating cell and its neighbors).

Next, we show $V_n < V_o$. To demonstrate this, it suffices to show that $\sum_{i=1}^{L} (X_i - M)^2$ in Eq. (1) is larger than $L \times (\overline{X} - M)^2$ in Eq. (2). For convenience, we denote these two quantities as:

$$C = \sum_{i=1}^{L} (X_i - M)^2 \tag{3}$$

$$D = L \times (\overline{X} - M)^2 \tag{4}$$

Subtracting Eq. (3) by Eq. (4), we get

$$\begin{aligned}
C - D &= \sum_{i=1}^{L} (X_i - M)^2 - L \times (\overline{X} - M)^2 \\
&= \left[\sum_{i=1}^{L} X_i^2 + L \times M^2 - 2 \times M \times \left(\sum_{i=1}^{L} X_i\right)\right] - \left[L \times \overline{X}^2 + L \times M^2 - 2 \times M \times L \times \overline{X}\right] \\
&= \sum_{i=1}^{L} X_i^2 - L \times \overline{X}^2 \qquad \text{(Note that } L \times \overline{X} = \sum_{i=1}^{L} X_i \text{)} \\
&= \sum_{i=1}^{L} X_i^2 + L \times \overline{X}^2 - 2 \times L \times \overline{X}^2 \\
&= \sum_{i=1}^{L} X_i^2 + L \times \overline{X}^2 - 2 \times \overline{X} \times L \times \overline{X} \\
&= \sum_{i=1}^{L} X_i^2 + L \times \overline{X}^2 - 2 \times \overline{X} \times \left(\sum_{i=1}^{L} X_i\right) \qquad \text{(Note that } L \times \overline{X} = \sum_{i=1}^{L} X_i \text{)} \\
&= \sum_{i=1}^{L} \left(X_i^2 + \overline{X}^2 - 2 \times X_i \times \overline{X}\right) \\
&= \sum_{i=1}^{L} (X_i - \overline{X})^2 .
\end{aligned} \tag{5}$$

According to the algorithm, if a CVCO is performed, there must exist a cell in the neighborhood with a cell value larger or smaller than the neighborhood average. In other words, among $X_1,\ldots,X_L$, there must exist X_k $(1 \le k \le L)$ with $|X_k - \overline{X}| > \varepsilon$. Thus we have:

$$C - D = \sum_{i=1}^{L} (X_i - \overline{X})^2 > \varepsilon^2 \tag{6}$$

Based on this, we have:

$$\begin{aligned} V_o - V_n &= \frac{Y + C}{N} - \frac{Y + D}{N} \\ &= \frac{C - D}{N} \\ &> \frac{\varepsilon^2}{N}. \end{aligned} \tag{7}$$

In other words, any CVCO decreases the variance by at least $\frac{\varepsilon^2}{N}$. Denote Δ_V as the minimum variance decrement. Clearly, we have:

$$\Delta_V > \frac{\varepsilon^2}{N} \tag{8}$$

Since the cell value is in the range $[\alpha, \alpha + Z]$, the maximum initial variance is:

$$\frac{\sum_{j=1}^{N} (X_j - M)^2}{N} < \frac{\sum_{j=1}^{N} Z^2}{N} = Z^2 \tag{9}$$

Thus the number of CVCOs performed by our algorithm for any initial color distribution is:

$$\leq \frac{Z^2}{\Delta_V} < \frac{N \times Z^2}{\varepsilon^2} \tag{10}$$

In other words, our algorithm converges and stops with less than $\frac{N \times Z^2}{\varepsilon^2}$ CVCOs for any initial cell value distribution.

While the above proof derives an upper bound on the number of CVCOs required for convergence, in practice, the algorithm may converge with several orders of magnitude less CVCOs and in much less time than indicated by the proof. There are three factors not accounted for in the proof. In the proof, the number of cells, N, and the domain range, $[\alpha, \alpha + Z]$, remain constant. In practice both values decrease rapidly, thus decreasing the number of CVCOs required for convergence. These two factors will be discussed in the subsection, “Reducing the Time for Convergence” (Page 60). Also, the number of CVCOs is not an indication of time required. Indeed, there may be potentially $O(N)$ CVCOs occurring in parallel in each time slot and thus the number of time slots is significantly less than the number of CVCOs. In subsections, "Maximizing Parallel CVCOs" (Page 65) and "Optimizing Neighborhood Radius" (Page 73), it will be shown that timing when CVCOs begin and the neighborhood radius (i.e., the value that determines the maximum number of neighbors in a CVCO) will affect the amount of parallelization of CVCOs and thus the time and energy required for convergence. Simulation results will be given to support these observations.

Proof of Consistent Average

An interesting and valuable attribute of this algorithm is that the global average after each CVCO is always equal to the initial calculated global average. Examining the process reveals why this occurs. A group of nine cells of different values (a neighborhood of radius 1) contributes to the average of the total grid. When these cells are averaged and each set to the same average value for the group, they contribute exactly the same to the average of the entire grid as they did with differing values.

Theorem 2: With each CVC, the global average of the network remains equal to the initial global average.

Proof: Let L denote the number of cells involved in the CVCO, including the operating cell and all of its neighbors. Denote the values of these cells as $X_1,\ldots,X_L$. We further denote the average value as:

$$\overline{X} = \frac{\sum_{i=1}^{L} X_i}{L} \tag{11}$$

It follows that $L\overline{X} = \sum_{i=1}^{L} X_i$. We can define the initial average, A_o, and the average after a CVC, A_n, as:

$$A_o = \frac{Y + \sum_{i=1}^{L} X_i}{N} \tag{12}$$

$$A_n = \frac{Y + (L\overline{X})}{N} \tag{13}$$

where Y is the sum of cell values outside of the neighborhood (i.e., cells other than the operating cell and its neighbors) and N is the total number of cells in the network. Since $L\overline{X} = \sum_{i=1}^{L} X_i$, it follows that $A_o = A_n$.

Reducing the Time for Convergence

This section details the effect of a rapidly decreasing domain range of values in practice. Theorem 1 guarantees that the averaging algorithm will converge and terminate in a finite number of CVCOs. The upper bound is:

$$\text{Maximum CVCOs} = \frac{N \times Z^2}{\varepsilon^2} \tag{14}$$

where N = number of cells in the grid, Z = the range of domain values to be averaged, and ε = the local tolerance such that all cells in the neighborhood have a cell value within the range $[\overline{X} - \varepsilon, \overline{X} + \varepsilon]$, where $\overline{X}$ is the average cell value in the neighborhood.

For example, for a 50x50 grid of cells, N = 2500, with a distribution of values in the range [0..255], Z = 255, and ε = 0.33, the maximum CVCOs would be:

$$\frac{2500 \times 255^2}{0.33^2} = 1.485E11 \tag{15}$$

It is important to note that the calculation is dependent on Z^2, so the number of CVCOs will increase geometrically with larger domain ranges.

However, simulation results converge much more quickly. For such a grid, domain range, and tolerance, the simulation converges in 262,266 total CVCOs through 137 iterations, where iteration is defined as a CVCO on all active cells. Examination of the simulation reveals why this happens: Z does not remain fixed, but reduces as the simulation progresses. With the square of Z as a factor, its reduction is significant. Z reduces because, with each CVCO, the variance of the neighborhood is reduced. For a neighborhood, if L is the number of cells in a neighborhood, C_i is the value of cell i and $\overline{X}$ is the average cell value in the neighborhood then:

$$Variance = (\sum_{i=1}^{L} (C_i - \overline{X}))^2 \tag{16}$$

Since the values of cells in neighborhood are not equal in a CVCO, the variance is greater than zero. Following a CVCO, the value of all cells in the neighborhood are equal and the variance is zero. When a CVCO occurs for another neighborhood containing one or more cells of this neighborhood, the variance temporarily increases. But with subsequent CVCOs of these neighborhoods the variance of each continues to decrease until the variance of the combined neighborhoods is zero.

In Theorem 1, the global variance also is guaranteed to decrease with each CVCO. Fig. 40 depicts simulation results that demonstrate how rapidly this decrease occurs. The key to how quickly the global variance decreases is in the rate of reduction of Z. Even for a worst-case neighborhood where cells of the maximum and minimum values are present, a CVCO reduces the variance of that neighborhood to zero as the domain of this neighborhood is a single value. With the reduction of variance in neighborhoods to zero with each CVC, the domain range decreases rapidly. Fig. 41 depicts the rapid reduction of Z. Initially, Z = 255, but after the first iteration (2500 CVCOs), Z has reduced to 41. By iteration 6 (< 15000 CVCOs), Z = 10. For the second half of the simulation, Z has reduced from 255 to 1.

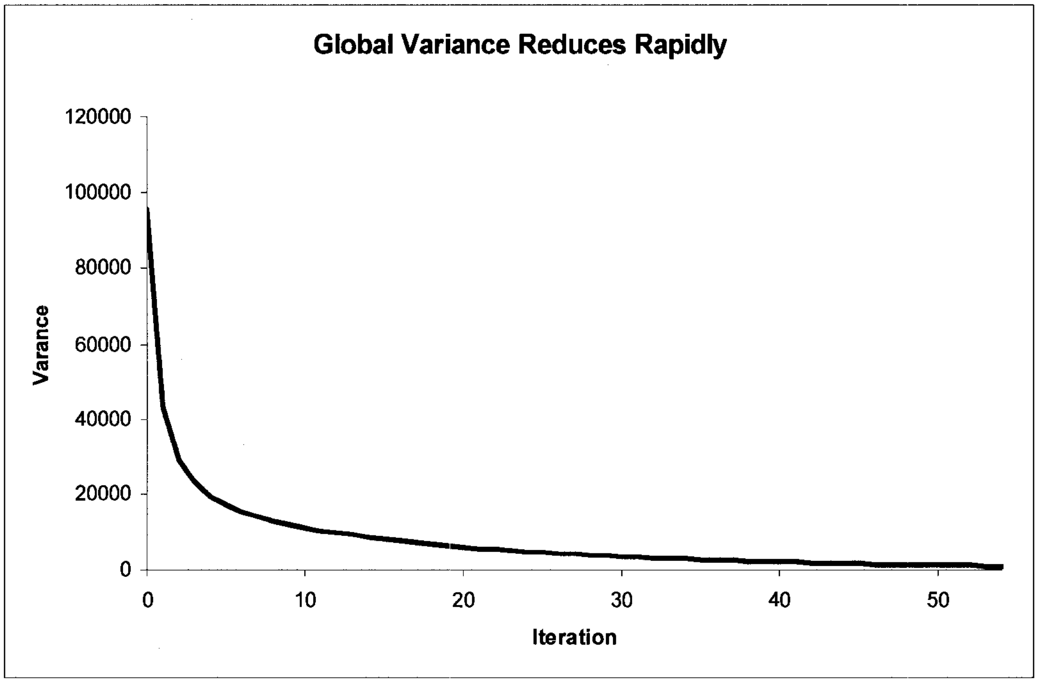


Fig. 40. Reduction in variance with each iteration.

If the observed values of Z at each iteration of the simulation are used as input to Theorem 1, the resulting prediction of CVCOs to convergence decrease so rapidly, a logarithmic scale is suitable in Fig. 42.

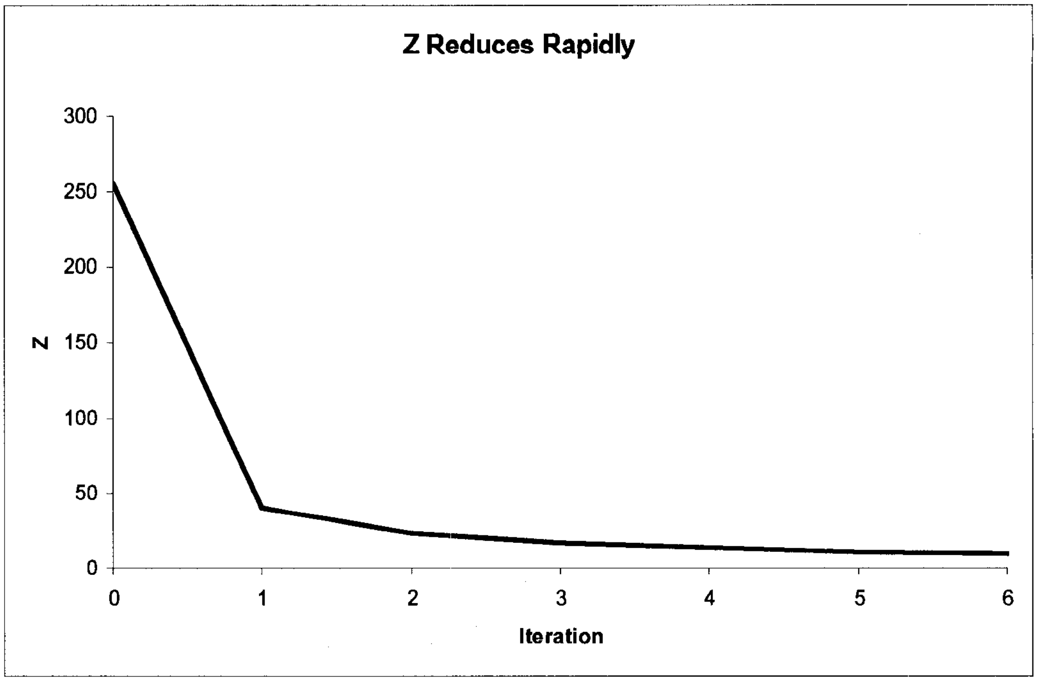


Fig. 41. Reduction in Z with each iteration.

Thus, a rapidly decreasing value of Z greatly reduces the number of CVCOs required for convergence. Fewer CVCOs require less time for convergence and also require fewer transmissions (i.e., less energy). In subsections, "Maximizing Parallel CVCOs" (Page 65) and "Optimizing Neighborhood Radius" (Page 73), the effect of parallel CVCOs and the size of the neighborhood for CVCOs will be examined and shown to have a great effect on both the time and energy required for convergence.

Local Versus Global Tolerance

The algorithm stops when there exists no neighborhood in the network requires a CVCO. This guarantees all cells in all neighborhoods are within tolerance of their neighborhood mean. We denote the local tolerance as ε_L, as in Theorem 1. However this alone does not guarantee that all cells have values within the tolerance ε_L of the global average cell value M, i.e., there may be a cell with a value X_i such that $|X_i - M| > \varepsilon_L$. Thus, satisfaction of an arbitrary local tolerance does not imply the satisfaction of the global tolerance. We can, however, determine an upper bound on the value of the local tolerance that will guarantee satisfaction of the global tolerance. If, at convergence, for

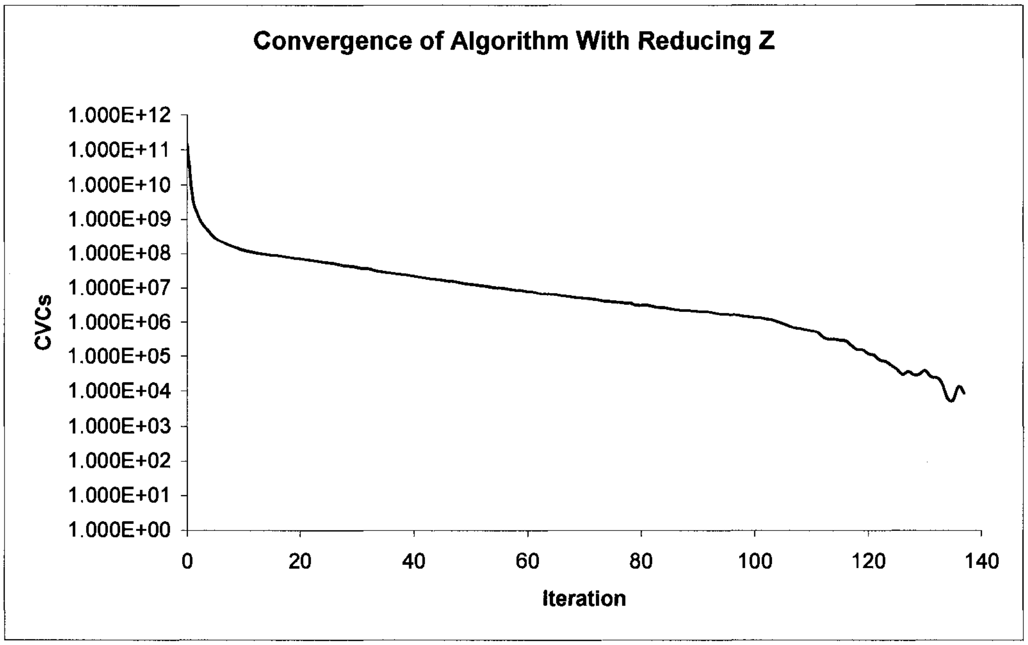


Fig. 42. Convergence of algorithm with reducing Z.

any neighborhood in the network having an average cell value $\overline{X}$, there exists no cell in the neighborhood having a cell value X_i such that $|X_i - \overline{X}| > \varepsilon_L$, then there cannot exist X_k, X_j in the same neighborhood with $|X_k - X_j| > 2\varepsilon_L$. That is, no two neighbors can differ by more than $2\varepsilon_L$. Assuming a square grid, the distance between any two cells is at most $\sqrt{N}$ number of cells, where N is the number of cells. Therefore, the maximum difference on the values of any two cells is $2\varepsilon_L\sqrt{N}$. Then to satisfy a specified global tolerance ε_G, we only require the local tolerance to satisfy $\varepsilon_L < \frac{\varepsilon_G}{2\sqrt{N}}$.

Maximizing Parallel CVCOs

The theorems proven thus far establish the concept that neighborhoods iteratively agreeing on a local average leads to all cells converging to the global average. The implementation of the simulator emulated MAC access by randomizing the order in which cells lead CVCOs. The section details an enhanced simulator implementing a MAC that quantifies the effect of simultaneous (i.e., parallel) CVCOs throughout the network and the effect of neighborhood size on the time, required number of transmissions, and the amount of energy required for convergence.

To maintain a consistent global average each CVCO must perform as an atomic operation. The leader cell broadcasts a request for participating neighbors. All neighbors that can participate report their current value to the leader cell. The leader cell then computes the neighborhood average, sets its own value to this average, and broadcasts the average value. All participating neighbor cells must receive from the leader cell the neighborhood average and set their value to that average. To assure completion of the atomic operation, sufficient acknowledgements must be sent to guarantee that the operation is complete. If the operation does not complete, all cells in the neighborhood must roll back to the value they had before the operation began.

When a cell is participating in a CVCO, it must commit to complete this operation before leading a CVCO or participating in another CVCO with any of its neighbors. Any cell may select itself to lead a CVCO autonomously at any time that it is not committed to participation in another CVCO. In previous simulations, the neighborhood radius was 1 resulting in a neighborhood of 8 neighbors plus the leader cell. Also, a cell could not lead a CVC unless all 8 neighbors could participate. This is not a realistic restriction as will be shown. As depicted in Fig. 43, Cell A, Cell B, and Cell C can lead a CVC simultaneously, but Cell D cannot lead a CVC until Cell A and B have completed their operations.

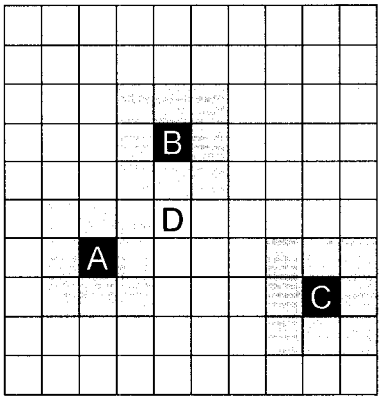


Fig. 43. Parallel CVCOs.

The simulator previously emulated this activity as follows. In each iteration of the simulation, all cells in the network are arranged in random order (a different order for each iteration). Then each cell in turn performs a CVCO sequentially by its order in the random sequence. This guarantees that all cells lead a CVCO, neighboring cells do not interfere with each other, but does not account for parallel CVCOs and does not provide an accurate assessment of time (i.e. with each CVCO, all cells not in the neighborhood wait). Such a series of CVCOs is depicted in Fig. 44 where one iteration requires 25 CVCOs.

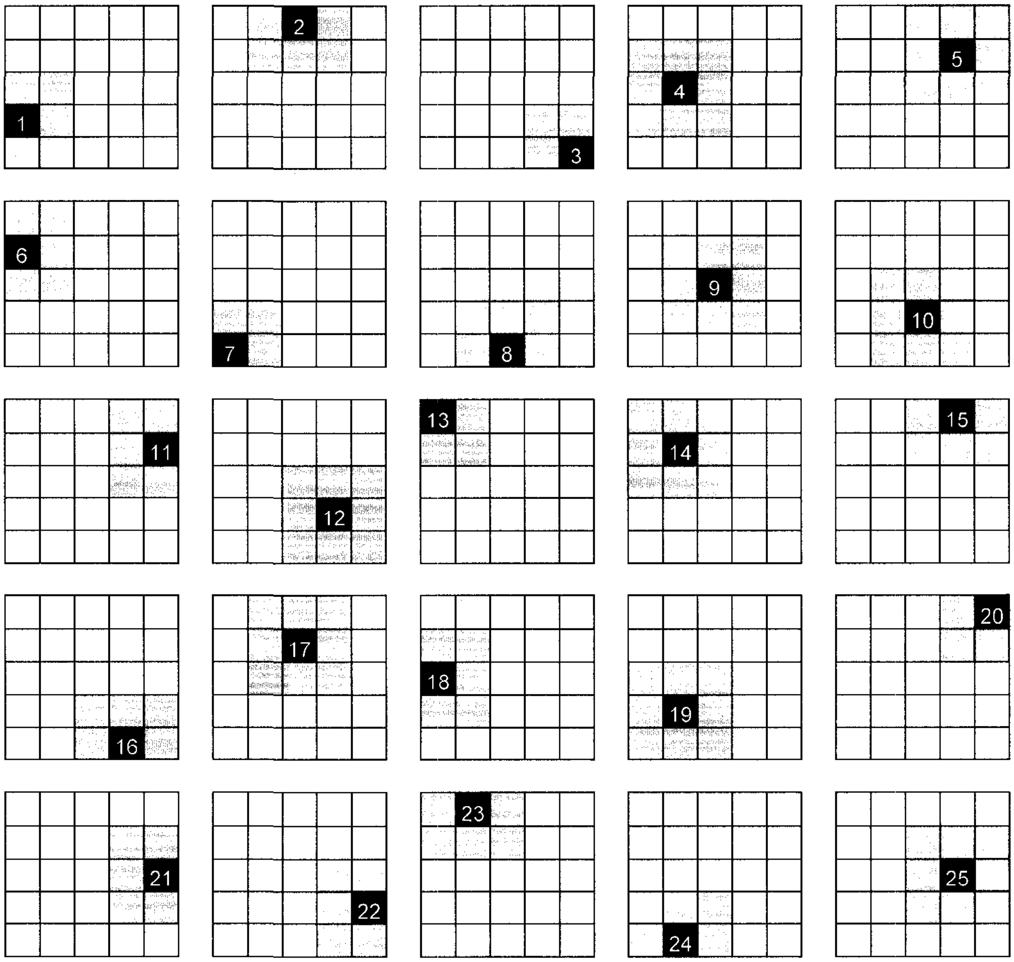


Fig. 44. One iteration requires 25 CVCOs.

A more realistic simulation would, for each iteration, select as large a set of cells as possible without interference to simultaneously lead CVCO operations. Thus the time for an iteration would be the time required for a CVCO on a neighborhood of maximum size. Such an iteration is depicted in Fig. 45 where 34 neighborhoods conduct a CVCO simultaneously. During the next iteration, a different set of leaders would be selected randomly simulating cells "capturing the wire" through Media Access Control. In the figure, black cells lead CVCOs, gray cells are neighbors participating in a CVCO with an adjacent black cell leader, and white cells are not participating.

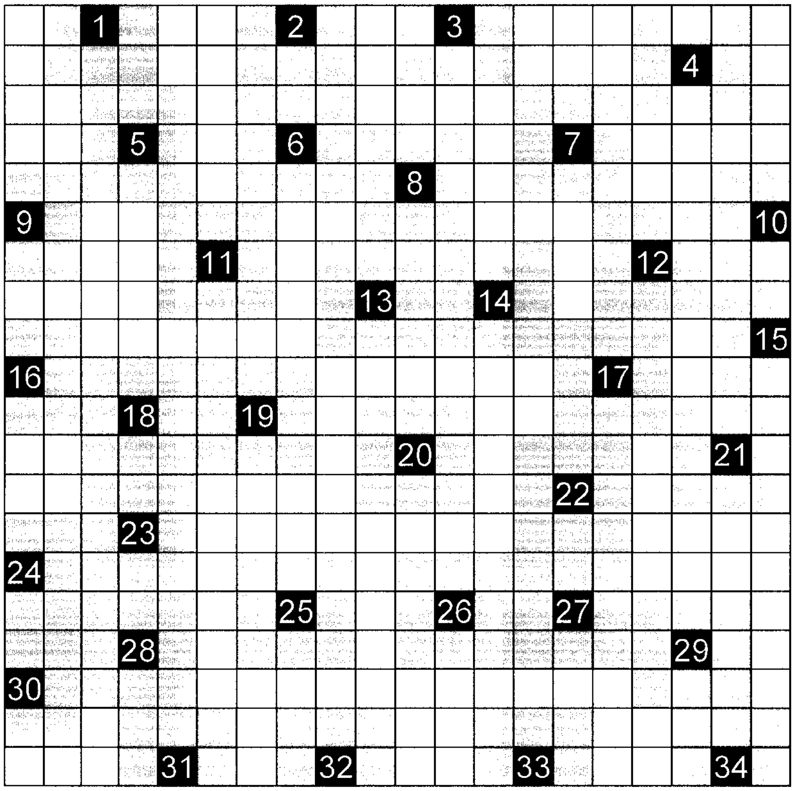


Fig. 45. One iteration consists of 34 parallel CVCOs.

Notice that 33% of cells in Fig. 45 are unable to participate in any CVCO during this iteration. This is because if a cell attempts to lead a CVCO and does not have eight neighbors that are not already participating in a CVCO, it does nothing. However, there is no requirement of Theorem 2 for neighborhoods to be of fixed size. A much more efficient, and realistic, method is, when a cell elects to lead a CVCO, it does so if at least one neighboring cell can participate (i.e., is not participating in another CVCO). Fig. 46 shows two such iterations. In iteration 1, Cell 1 secures media access first and captures 8 neighbors for its CVCO. Cell 2 captures 4 neighboring cells, Cell 3 captures 3 neighbors, Cell 4 captures 1 neighbor, Cell 5 captures 1 neighbor, and Cell 6 captures 1 neighbor. Only one cell is not selected to participate with another CVCO and has no neighbors to initiate a new CVCO; it remain inactive during this iteration. *Notice that this introduces a practical approach to implementation*: with no requirement that all neighbors participate, a cell can lead a CVCO even though some neighbors do not participate for whatever reason (e.g., they are asleep when the invitation request is made, they fail to hear the request, etc.). If a cell never participates, it is as if it were not a part of the network and will not affect the results. If it fails to participate in some CVCOs but does in others, it will have no effect on convergence except increasing the time required.

At all times, all cells want to transmit. Either they want to initiate a CVCO as leader or they want to respond to another CVCO leader. Therefore, a contention-based MAC such as Carrier Sense Multiple Access (CSMA) is not practical as there would be too many collisions. Rhee et al. [75] have developed DRAND, a method to establish a Time Division Multiple Access (TDMA) frame for local neighborhoods such that all cells in contention within a neighborhood are guaranteed a contention free time slot. Fig. 47 depicts a DRAND frame for a neighborhood of radius 1. Notice that, to prevent collisions, not only do the cells of the neighborhood have to be assigned a time slot, but cells of radius 2 must also be assigned a time slot in the frame. Thus, a time frame (TF) for neighborhood of radius 1 would be:

$$TF = \overset{\text{Leader}}{1} + \overset{\text{Neighbors}}{8} + \overset{\text{Adjacent to Neighbors}}{16} = 25 \text{ slots} \tag{17}$$

Cells participating in this CVCO would have to wait during slots for non-participating cells. At a minimum, two frames would be required for a CVCO.

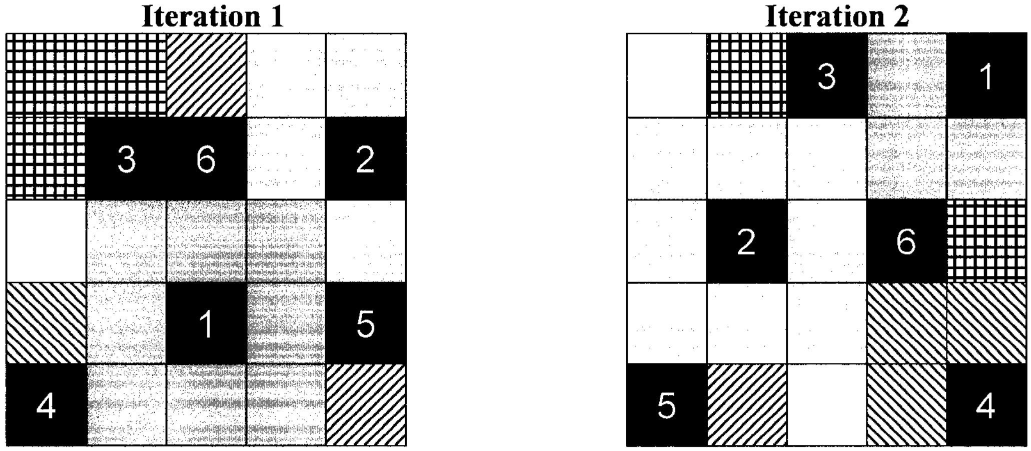
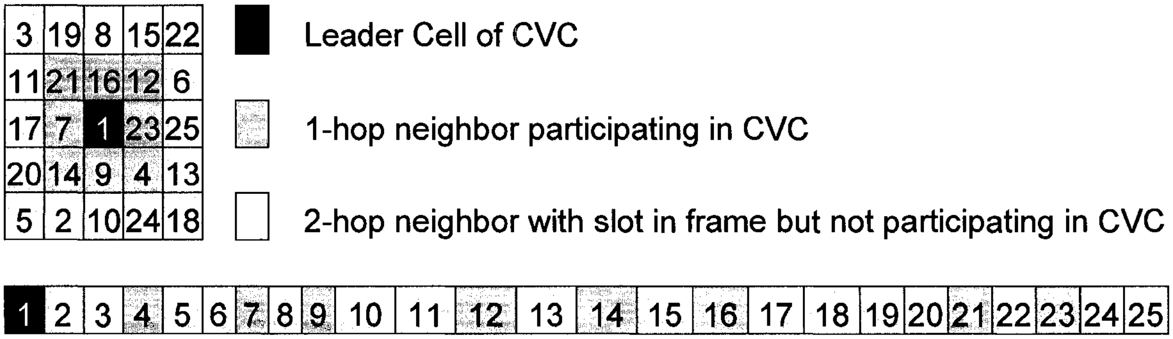


Fig. 46. Two iterations optimizing cell participation.

The simulator was modified such that a time step is the time for a TDMA time slot. As the neighborhood radius determines the time frame, a set of parallel CVCOs will complete for every two time frames. Fig. 48 shows time required for simulations of networks ranging in size from 40,000 to 1 million cells. For all of these network sizes, 1014 TDMA time slots are required (6 TDMA frames) to converge all cells within a tolerance of 10% of the initial domain range. Fig. 49 shows the minimum and maximum number of transmissions required of each cell for convergence. Notice that they are consistent regardless of network size. These results verify adherence to the desired properties: time and energy consumption required for convergence do not increase with increasing network size.



3	19	8	15	22
11	21	16	12	6
17	7	1	23	25
20	14	9	4	13
5	2	10	24	18

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25

■ Leader Cell of CVC
▨ 1-hop neighbor participating in CVC
□ 2-hop neighbor with slot in frame but not participating in CVC

Fig. 47. DRAND TDMA frame for a CVCO.

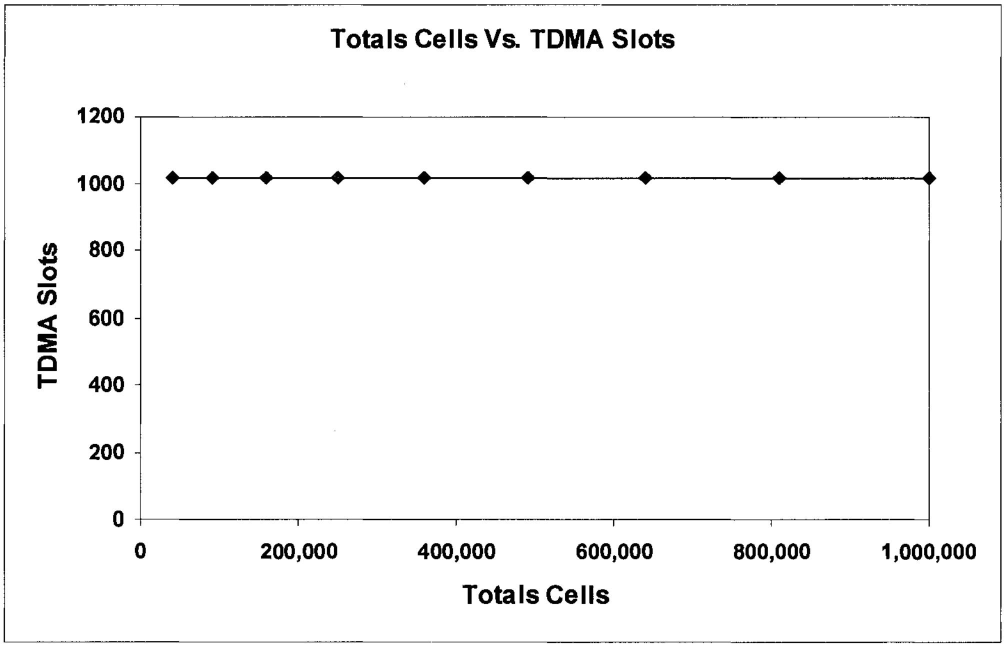


Fig. 48. TDMA slots required for convergence

This implementation of the simulator realistically models parallel CVCOs. Fig. 50 shows the distribution of number of CVCOs of various neighborhood sizes for a network of 1 million cells with neighborhood radius 1 (121 cells). 14% of CVCOs are full neighborhoods with the remaining CVCOs distributed among other neighborhood sizes (slightly more in smaller neighborhoods). Fig. 51 shows the distribution of transmissions executed by neighborhoods of various sizes. 29% of transmissions are executed by full neighborhoods while the remaining transmissions are evenly distributed among other neighborhood sizes (slightly more in larger neighborhoods).

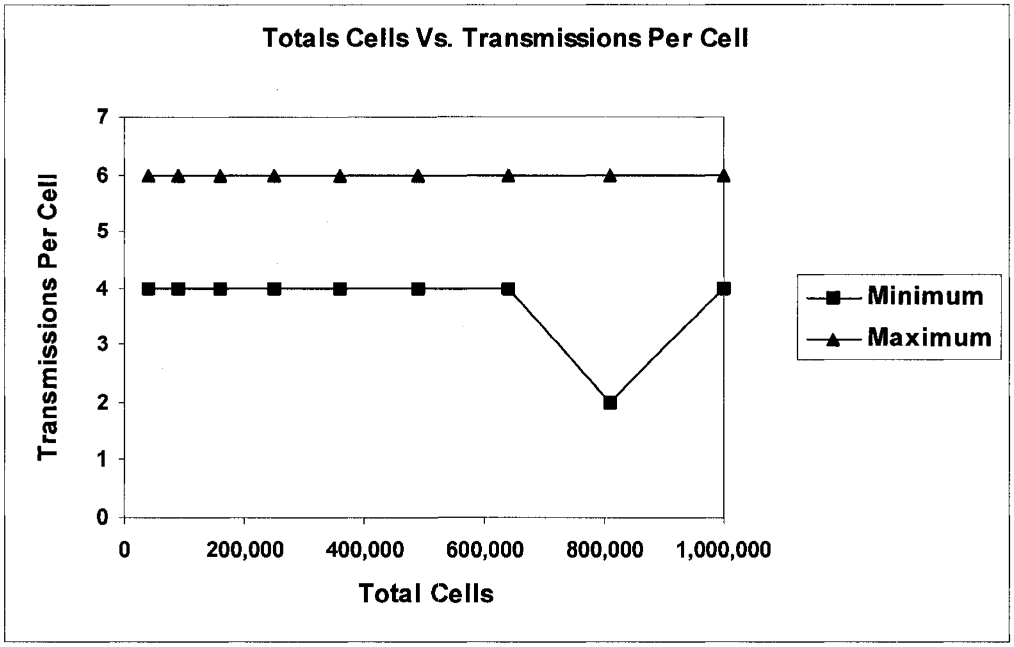


Fig. 49. Transmissions per cell required for convergence.

Optimizing Neighborhood Radius

In the previous section, the concern is maximizing the number of neighborhoods that operate in parallel. Next, the consideration is what is the optimal neighborhood size? For the examples given above, a neighborhood radius of 5 specified a neighborhood size of 121 cells. For the following, simulations are shown for a network of 250,000 cells where the neighborhood radius varies from 1 (9 cells) to 20 (1681 cells). Fig. 52 shows that the time required for convergence rises sharply for neighborhood radii greater than 6.

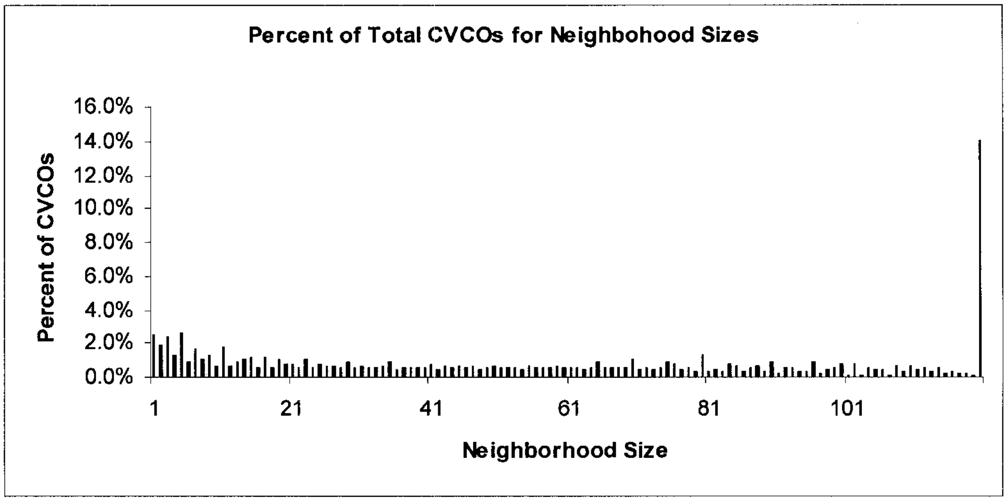


Fig. 50. Percentage of CVCOs for various neighborhood sizes.

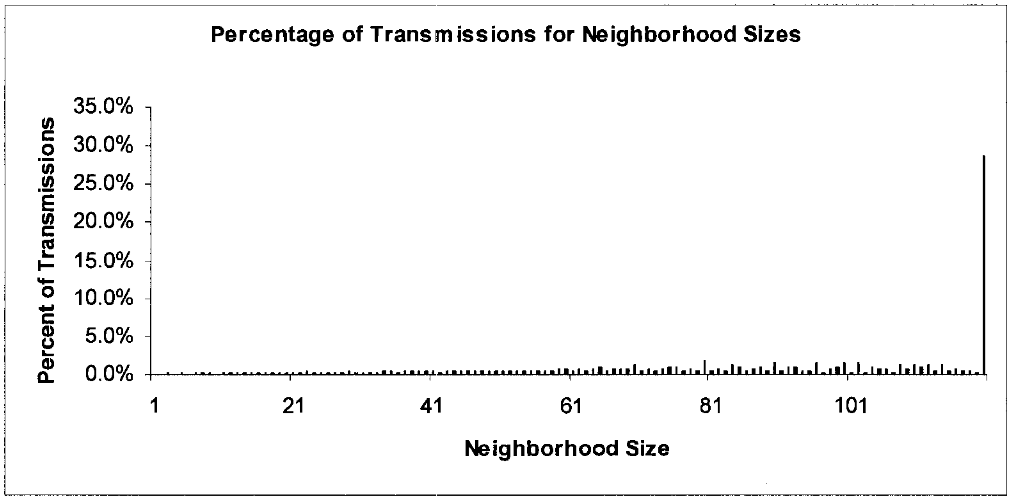


Fig. 51. Percentage of transmissions for various neighborhood sizes.

Fig. 53 shows that the number of transmissions required for convergence stabilizes and is minimal at and after neighborhood radius of 6. Fig. 54 shows the same for average number of transmissions per cell. Thus, the optimal choice of neighborhood size is 6 for all considerations.

Small neighborhoods increase the amount of parallel CVCOs, but reduce the rate at which Z, the domain range, decreases. Larger neighborhoods reduce the number of parallel CVCOs, but increase the rate of Z reduction. The figures reveal that, in this trade off of advantages versus disadvantages, a small, but not smallest, neighborhood is best. Energy consumption is most important. With an increase of 1 in neighborhood radius, time increases consistently (average increase of 16% and a maximum increase of 41%) relative to energy consumption. Transmissions decrease by 52% when the neighborhood radius is increased from 1 to 2; 41% from 2 to 3; 33% from 3 to 4; 11% from 4 to 5; 23% from 5 to 6; but an average of 1% from 6 to 20. It is clear from this that a neighborhood radius of less than 6 should not be used. Considering both time and energy, the optimal neighborhood size is 6.

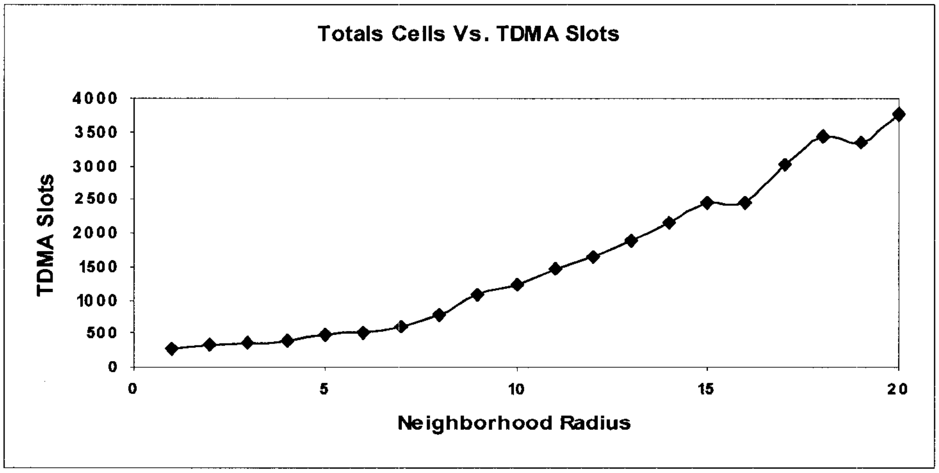


Fig. 52. TDMA slots required for convergence of various neighborhood sizes.

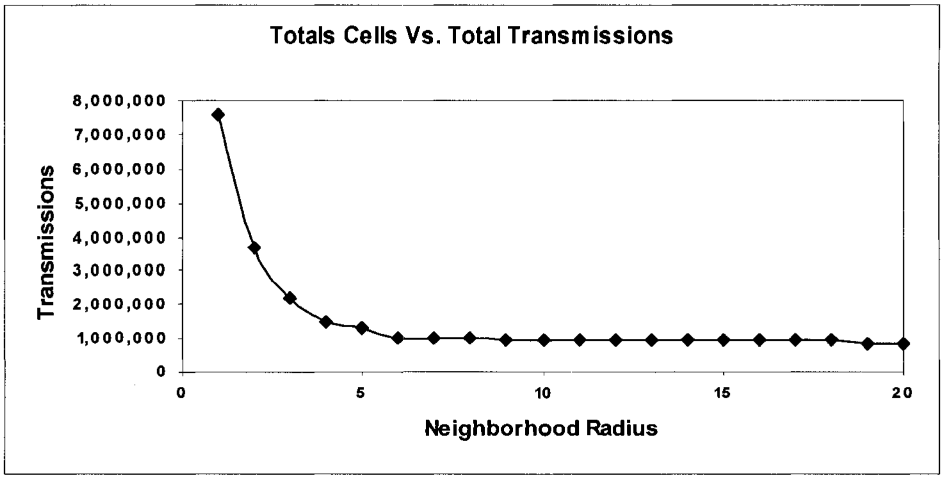


Fig. 53. Transmissions required for convergence of various neighborhood sizes.

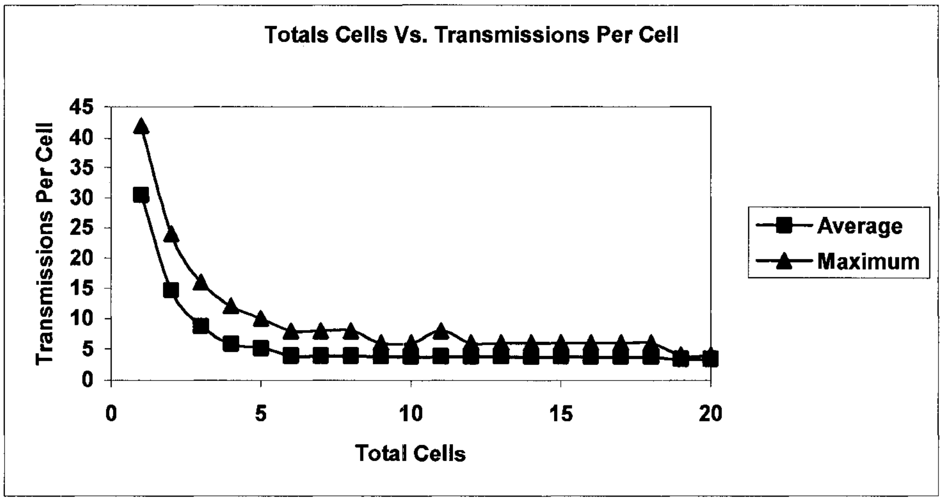


Fig. 54. Transmissions per cell required for convergence of neighborhood sizes.

Communal Sensor Network for Adaptive Noise Reduction

The goal of this research is to develop and analyze a replacement for the current passive liner of aircraft engine nacelles by an active, intelligent, and adaptive system that can adjust its impedance to account for changing aeroacoustic conditions resulting from changes in the operational environment. The architecture employed uses a community of cooperative, intelligent, and autonomous resonators instead of the more common hierarchical, centralized design approach.

We have developed an algorithm in which the individual liner resonators cooperate with their local neighbors to derive their own impedance tuning based on local information to improve sound reduction. The challenge was to produce a rule set for making this decision from which a maximum global attenuation emerges: engineering for desirable emergent behavior.

An implementation of dynamically tunable resonators to provide a uniform liner could be implemented by a central controller, which could respond to a changing input source by lookup of the optimum uniform liner impedance from a predetermined table. However, as Mani [55] and Watson [58] proved, segmented liners can be designed to outperform the best uniform liner for a given acoustic source. In the following sections, an algorithm is described that, through in situ adaptation, defines such a segmented liner configuration

To assess noise reduction, a metric is required. Noise attenuation results from a reduction in sound pressure exiting from the nacelle. Sound pressure is measured in decibels (dB), a logarithmic scale. Thus, a decrease in pressure of 10 dB decreases the pressure 10-fold, a decrease of 20 dB decreases the pressure 100-fold, etc. Pressure is not directly related to human hearing sensitivity. It is generally accepted that sound perceived by human ears is halved for every 6 dB decrease in pressure. Therefore, a decrease of a few dB in attenuation is significant.

In the following subsections, the duct used in this analysis is described including the liner, its placement, and the segmentation strategy. Also described are the input source conditions. The simulation and analysis software are detailed and the algorithm for selecting liner impedance values and evaluating in-duct attenuation is detailed. Finally, in the simulation results are presented.

Duct and Source Input Specifications

The geometry considered in this analysis is modeled after the NASA Langley Curved Duct Test Rig (CDTR) and is a rectangular duct with a 15" X 6" cross section and an axial length of 32". As depicted in Fig. 55, one entire wall of the duct (15" X 32")

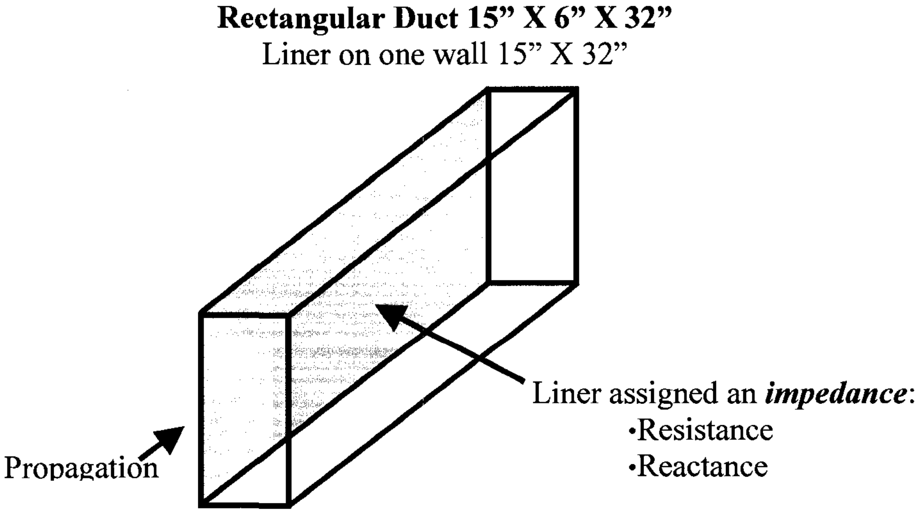

Rectangular Duct 15" X 6" X 32"
Liner on one wall 15" X 32"
Liner assigned an ***impedance***:
•Resistance
•Reactance
Propagation

Fig. 55. Rectangular duct for analysis.

is acoustically treated (the *liner*) along the full extent, while the other three sides are hardwall. The liner may be uniform or segmented. Acoustic source input consisted of the least attenuated mode at frequencies varying from 1000 to 2500 Hz in 500 Hz increments. These frequencies were chosen to provide a variety of aeroacoustic conditions requiring largely different optimal impedances (i.e., an impedance providing high attenuation for one frequency will provide low attenuation for other frequencies). Finally, the mean (background) flow Mach number was taken to be zero.

Simulation and Analysis Software

The impedance optimization approach used in this study began by identifying an "allowable" impedance domain, defined by a range of resistance and reactance values, over which the optimal impedance was desired. For each possible impedance value, in-duct acoustic propagation calculations were performed to predict the liner performance. The overall attenuation, defined below, was the metric (i.e., objective function) by which relative liner effectiveness was determined. The optimization process was terminated when an increase in attenuation could no longer be achieved. The algorithm used to control the aforementioned process is detailed in the next subsection.

The analysis software used to perform the in-duct acoustic propagation calculation and overall attenuation prediction for a given liner was CDUCT-LaRC [79], designed to study propagation within and radiation from a three-dimensional geometry such as the duct described above. Given a computational grid, mean flow solution, acoustic source, and, optionally, an acoustic liner positioned somewhere at the boundaries of the duct, CDUCT-LaRC will calculate the propagation within the duct geometry. Although CDUCT-LaRC is a general-purpose program with extensive flexibility for analyzing different geometries under various conditions, only options pertinent for use in this analysis will be discussed.

A three-dimensional geometry file containing a representation of the rectangular duct described above is required as input to CDUCT-LaRC to enable mean flow and propagation calculations. A preprocessor was developed to generate the input rectangular duct geometry based on given dimensions (width, height, and length) and grid resolution. Output is in the form of a Plot-3D file [80] that represents the duct as a series of grid points distributed equally in all dimensions throughout the duct. This file is input to CDUCT-LaRC and remains fixed throughout the analysis. The preprocessor is run only once unless the duct geometry or grid resolution is changed. A single geometry file was generated and used throughout this analysis matching the CDTR geometry described above with 31, 25, and 65 equally spaced grid points in the horizontal, vertical, and axial directions, respectively.

CDUCT-LaRC was next used with the resulting geometry file as input to compute a mean flow solution file required to compute acoustic propagation. The mean flow solution was assumed to be uniform flow with Mach number, M, equal to zero (no flow).

Because the liner begins at the duct's entrance, the acoustic source input to CDUCT-LaRC must be regenerated for each change in the impedance of the liner at the duct entrance. A preprocessor, *SW_MODES*, was developed to generate the acoustic source input file. For the cases considered, this acoustic source was taken to be the least attenuation mode for a rectangular duct with one wall lined (of given impedance) and zero mean flow. The modal amplitude was set to a constant value for all cases, which led to acoustic sources of varying acoustic power for each source plane impedance value. However, this did not affect the optimization approach, as the chosen metric was based only on the overall attenuation (i.e., reduction in acoustic power) from source to exit plane for each possible impedance value. Input to SW_MODES was then the impedance at the beginning of the liner along with other parameters defining the input source such as the geometry, frequency, flow speed, and mode specifications. Output of SW_MODES is a Plot3D function file [80] containing the acoustic source that is input to CDUCT-LaRC.

Input to CDUCT-LaRC includes the geometry file, the mean flow solution file, and the acoustic source file. Other input includes the acoustic liner definition in two files: the grid point boundaries of each segment (one segment or all segments of equal impedance for a uniform liner) are defined in a neutral map file and the liner is defined in a file that specifies the impedance value for each segment.

Pertinent output from CDUCT-LaRC includes a complex acoustic potential at each grid point. A post processor, ap2pressure, was developed to convert the acoustic potential value to a complex pressure value at each grid point. These values were converted to acoustic intensity and integrated for each segment to determine acoustic power for the segment. The acoustic power is used to make decisions on subsequent impedance choices. CDUCT-LaRC also outputs a global attenuation for the duct that is only useful in this analysis for corroboration of choices made by the sensor community.

A wrapper program, *MAXATTENUATION*, controls the simulation. It begins by accepting input information, including the frequency and impedance for the initial uniform liner. It then executes a loop for a series of candidate liner definitions wherein:

- The neutral map file and liner definition file are written.
- SW_MODES is executed defining the acoustic source.
- CDUCT-LaRC is executed producing an acoustic potential file.
- ap2pressure is executed to convert acoustic potential to pressure.

The candidates are compared and the liner with the highest attenuation is selected. This choice determines the next set of candidate selections and the loop repeats. The process terminates when no other candidates can provide a higher attenuation than the current liner. Details of candidate selection and attenuation calculation are provided in the next section.

Algorithm for Selecting Liner Segment Impedance

In the following sections, an algorithm is developed to study the possibility of adapting to a segmented liner capable of producing attenuation levels at least as high as those obtained with the best possible uniform liner. To understand the algorithm, first a global view of attenuation is used as a metric for adaptation. Next, the details of calculations to assess attenuation locally are discussed. Then an algorithm is detailed that uses a physical neighborhood view (see Section 3) to identify the uniform liner producing highest global attenuation. Finally, an algorithm is detailed that uses a logical neighborhood (see Section 3) view to identify a segmented liner producing attenuation that exceeds the best available from a uniform liner.

Evaluating Attenuation Using Global View

The objective of adaptive noise reduction is to determine a liner impedance distribution that provides the highest possible acoustic attenuation for a given geometry and flow conditions. As mentioned previously, impedance is represented by a complex number where the real component is the resistance and the imaginary component is reactance. The domain of "allowable" impedance values, **I**, can be represented as an Rs X Ra matrix with Rs possible resistance values and Ra possible reactance values. Obviously, the overall dimensions of this impedance domain are a function of the range of resistance and reactance values to be considered, as well as the relative spacing chosen between subsequent values. For a given frequency, the uniform liner providing highest attenuation can be determined by executing CDUCT-LaRC once for each impedance in the domain **I**. For all frequencies tested, the impedance domain **I** was composed of resistance values in the range, [0.1 .. 2.5], and reactance values in the range, [-2.0 .. 2.0], each incremented by 0.1. These were selected because values outside of these ranges are known to produce low attenuations for the selected frequencies. Fig. 56 depicts a subset of the domain **I** for 1000 Hz.

Execution of CDUCT-LaRC with a uniform liner of impedance = 0.2-1.3*i* results in matrix entry, **I**(0.2 , -1.3) producing global attenuation 3.00 dB. The maximum attenuation for this domain of 42.7 dB is at **I**(0.9, -0.8) attained with a uniform liner of impedance = 0.9-0.8*i*.

						Ra				
		-1.3	**-1.2**	**-1.1**	**-1**	**-0.9**	**-0.8**	**-0.7**	**-0.6**	**-0.5**
	0.1	1.06	1.12	1.18	1.23	1.28	1.31	1.32	1.25	1.13
	0.2	3.00	3.20	3.39	3.56	3.69	3.75	3.72	3.60	3.39
	0.3	4.93	5.29	5.63	5.93	6.16	6.27	6.22	6.02	5.69
	0.4	6.86	7.40	7.92	8.39	8.76	8.94	8.89	8.59	8.09
	0.5	20.36	9.51	10.26	10.96	11.54	11.86	11.83	11.38	10.61
	0.6	22.30	27.00	12.64	13.65	14.56	15.17	15.21	14.52	13.31
Rs	**0.7**	23.18	27.40	32.16	16.44	17.87	19.06	19.43	18.24	16.16
	0.8	23.26	26.97	31.19	36.00	41.96	23.66	25.73	22.74	18.86
	0.9	22.78	25.94	29.52	33.75	39.52	42.70	33.54	25.50	20.59
	1	21.90	24.48	27.24	30.13	32.69	32.90	29.39	24.77	20.87
	1.1	20.79	22.81	24.82	26.62	27.76	27.55	25.71	22.98	20.20
	1.2	19.58	21.13	22.56	23.70	24.27	24.01	22.87	21.11	19.13
	1.3	18.38	19.57	20.60	21.34	21.65	21.42	20.63	19.42	17.99

Fig. 56. Attenuation values for uniform liners of impedance values in domain **I**.

An *impedance neighborhood* in the domain is defined as matrix entry, **I**(Rs, Ra) and the eight entries adjacent to it and is displayed in Table 4:

Table 4

A neighborhood in the impedance domain **I**.

I(Rs-0.1, Ra-0.1)	**I**(Rs-0.1, Ra)	**I**(Rs-0.1, Ra+0.1)
I(Rs, Ra-0.1)	**I**(Rs, Ra)	**I**(Rs, Ra+0.1)
I(Rs+0.1, Ra-0.1)	**I**(Rs+0.1, Ra)	**I**(Rs+0.1, Ra+0.1)

Entries on a border of the matrix have fewer neighbors For example, **I**(0.1, -1.3) has three neighbors: **I**(0.2, -1.3), **I**(0.2, -1.2), and **I**(0.1, -1.2). Notice, in Fig. 56, that for the maximum attenuation at **I**(0.9, -0.8), all neighbors have less attenuation. Also, for any other entry, there is at least one neighbor with attenuation greater than that entry and only one neighbor with the greatest attenuation in the neighborhood. The algorithm for determining the entry with highest attenuation is as follows:

1. Given an initial impedance, examine the attenuation produced by impedances of all entries in its impedance neighborhood.
2. Select the neighbor producing the highest attenuation and make that the center of a new impedance neighborhood. (Hereafter steps 1 and 2 are referenced as a *neighborhood assessment*).
3. Continue this process until the impedance of the neighborhood's center entry produces the highest attenuation. This terminates the process as it has

identified the entry producing the highest attenuation in the domain.

As an example neighborhood assessment, a uniform liner with impedance, 0.6-1.0i produces attenuation of 13.65 dB (specified at **I**(0.6, -1.0)). If the liner tunes its impedance to that of each of its neighbors in **I**, and examines their resulting attenuation, the neighbor with the highest attenuation, 32.16, is entry **I**(0.7, -1.1). A new neighborhood is formed as the impedance adjusts to **I**(0.7, -1.1) as the center of the neighborhood. This may be described as a move: not a physical move but rather an adjustment to another entry in the impedance domain, **I**. For this neighborhood, its neighbor, **I**(0.8, -1.0) has the highest attenuation of 36.00. Adjusting to the impedance of **I**(0.8, -1.0) as the center of the neighborhood, its neighbor, **I**(0.8, -0.9) has the highest attenuation of 41.96. Adjusting to the impedance of **I**(0.8, -0.9) as the center of the neighborhood, its neighbor, **I**(0.9, -0.8) has the highest attenuation of 42.70. Adjusting to the impedance of **I**(0.9, -0.8), there are no neighbors with a higher attenuation, thus, the process terminates. The result of this adaptive process is plotted in Fig. 57. Importantly, there are only 5 points in this process where the attenuation is actually lower than the initial attenuation, 4 of these as the first four steps and the fifth within the first 24% of the iterations. There are two problems with this approach:

1. For this to work, the liner would have to know the global attenuation as it tunes its impedance to each entry in all neighborhoods.
2. The best that can be achieved is a uniform liner.

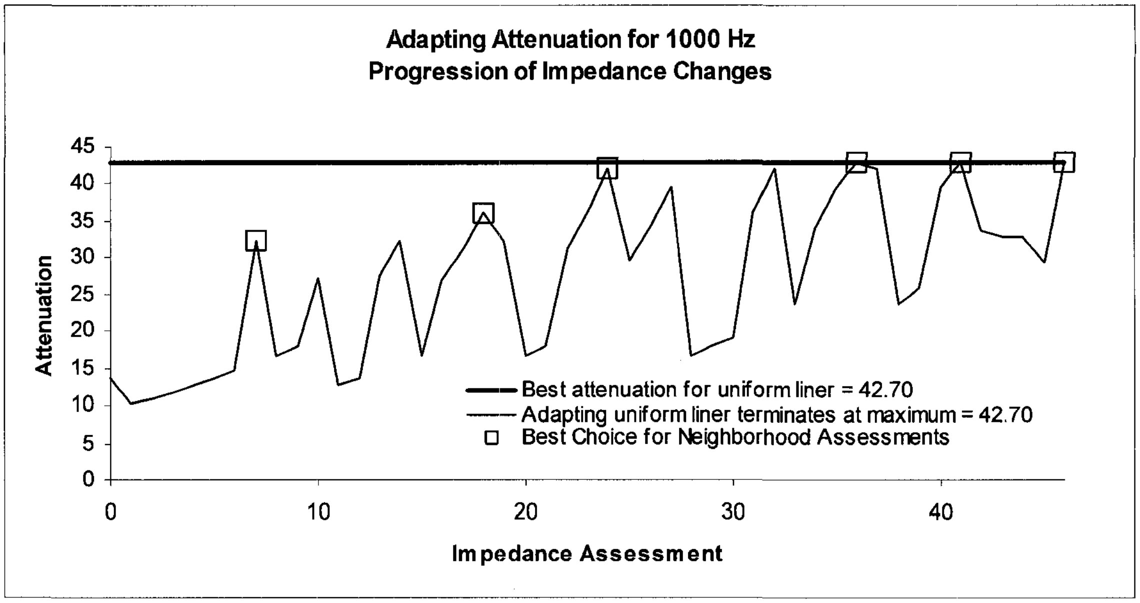


Fig. 57. Adapting attenuation from 0.6-1.0*i* to 0.9-0.8*i*.

Evaluating Attenuation Using Local View

Rather than evaluating global attenuation, it is desirable to evaluate local power within the duct to assess changes in impedance. This would reduce computation and communication costs, as well as distributing the decision process. An assessment of local power can be used to determine global attenuation.

As discussed above, in addition to global attenuation, CDUCT-LaRC computes locally for each grid point a complex value for acoustic velocity potential. From this value, pressure can be computed:

Constant values:

ρ = Ambient air density = 1.22 kg/m^3

C	=	Ambient speed of sound	=	340 m/s
M	=	Speed of air flow (Mach#)	=	0.0

Variables:

P	=	Pressure (complex)		
ϕ	=	Acoustic velocity potential (complex)		
F	=	Frequency		
W	=	Circular frequency	=	$2 * \pi * F$
K	=	Wave Number	=	W / C
D	=	Distance in gridpoints between values of ϕ (1 if G_i is at liner entrance or exit; else 2.		

For each grid point, G_k:

if G_k is on liner edge of flow entrance:

$$\frac{\partial \phi}{\partial D} = \frac{\phi_{k+1} - \phi_k}{1} \tag{18}$$

else if G_k is on liner edge of flow exit:

$$\frac{\partial \phi}{\partial D} = \frac{\phi_k - \phi_{k-1}}{1} \tag{19}$$

else:

$$\frac{\partial \phi}{\partial D} = \frac{\phi_{k+1} - \phi_{k-1}}{2} \tag{20}$$

Pressure for grid point, G_k computed as:

$$P_k = \rho * C * ((i * K * \phi_k) + (M * \frac{\partial \phi}{\partial D})) \tag{21}$$

From pressure, intensity can be computed:

P	=	Pressure (complex)
P^*	=	Conjugate of pressure (complex)
U	=	Acoustic particle velocity (complex)
U^*	=	Conjugate of acoustic particle velocity (complex)
ζ	=	Impedance (complex)
ζ^*	=	Conjugate of impedance (complex)
I	=	Intensity (power/ area)

Since:

$$\zeta = P / U \tag{22}$$

therefore:

$$U = P / \zeta \qquad (23)$$

and:

$$U^* = P^* / \zeta^* \qquad (24)$$

Since:

$$I = P * U^* \qquad (25)$$

therefore:

$$I = P * (P^* / \zeta^*) \qquad (26)$$

As described, pressure is computed indirectly by CDUCT-LaRC for each grid point. Impedance for each grid point is determined from the liner definition input to CDUCT-LaRC. Therefore, for each grid point, G_k, the intensity is computed from the pressure and impedance as:

$$I_k = P_k * (P^*_k / \zeta^*_k) \qquad (27)$$

Intensities of all grid points in the cross-sectional cube of the duct over the liner segment can be integrated over the duct area to calculate the acoustic power level, *PWL*, for the segment. Local assessment of attenuation for a segment is computed as a *delta power level* (ΔPWL), which is a function of the ratio of the two acoustic power levels, that closer to the entrance, PWL_{in}, and that closer to the exit of the duct, PWL_{out}:

$$\Delta PWL = \quad 10 * \log(PWL_{in} / PWL_{out}) \qquad (28)$$

In the following sections, acoustic power level will be used in varying localities to assess global attenuation.

Adapting to Best Uniform Liner

For this example, the 32" liner is segmented axially into 16 segments. The longest wavelength for the selected frequencies, that of 1000 Hz, is 13.4" (using c=340 m/s). Impedance changes for segments with a less than 1/8 of the wavelength (13.4 / 8 = 1.7) show minimal effect on the acoustic wave. Thus, the number of segments was selected to avoid segments narrower than 2". Although there are many tunable resonators within each segment, for this study, they act in concert as impedance values will only be determined for a complete segment. The liner is depicted in Fig. 58. Segments are numbered 0, 1, 2, …, 15 from entrance to exit.

The segment delimits an axial slice of the total duct length. The acoustic power level for each cube is used to compute delta power levels. For example, if PWL_i is the acoustic power level of the cube delimited by segment, S_i, then the delta power level, ΔPWL, between segment, S_3, and segment, S_4, is:

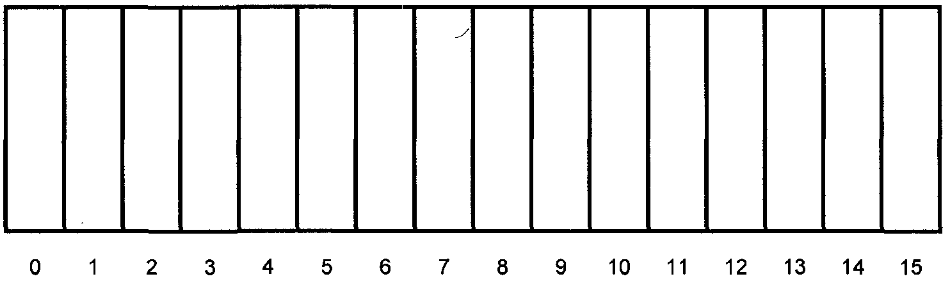


Fig. 58. Segmented liner.

$$\Delta PWL_{3\text{-}4} = 10 * \log(PWL_3 / PWL_4) \qquad (29)$$

To use local information instead of global attenuation, the algorithm is modified such that, instead of comparing global attenuation among neighboring impedance choices, each segment computes a local power assessment using its acoustic power level and that of the adjacent segment towards the exit (S_0 to S_1, S_1 to S_2, … S_{14} to S_{15}). This method is completely local to adjacent segments (i.e., no segment need ever communicate with any segment other than adjacent segments).

Adapting to Best Segmented Liner

A segmented liner with significantly higher attenuation than the best available from a uniform liner can be achieved at the cost of increased communal organization and neighborhood size. To accomplish this, two changes are required to the algorithm previously described.

First, a *leader* segment is assigned and all segments towards the exit from the leader serve as its slaves. That is, the leader decides on impedance changes and all resonators within that segment and the resonators of all slave segments accept that decision. Initially, segment, S_0, is assigned as the leader. It executes the loop performing a series of neighborhood assessments: examining all neighbors, selecting the best impedance, selecting a new neighborhood based on this best impedance, and terminating when the selected impedance is better than any of its neighbors. With each decision, it dictates to its resonators and to the resonators of all slave segments the new impedance. All of these resonators tune to this impedance. When the leader terminates, it freezes its impedance for the remainder of the process and turns leadership over to its adjacent segment towards the exit. This process continues until segment, S14, terminates. Segment, S15, assigns to its resonators the best impedance determined by the termination of segment, S14. Leadership is returned to the segment, S_0, which retains its current impedance (as do all other segments). The process terminates until segment, S_0, determines that the attenuation has dropped below a threshold and the process begins again. The result of this process is a segmented liner producing a significantly increased global attenuation.

It is important to note that while the initial leader assignment requires communication from opposing ends of the liner, this is not a requirement for global information: there are still only two segments providing information for a decision. With each transfer of leadership, the distance between the two opposing, communicating segments decreases.

Simulation Results

Fig.59 specifies the impedance providing maximum attenuation for the frequencies 1000 Hz, 1500 Hz, 2000 Hz, and 2500 Hz. These were determined by executing CDUCT-LaRC once for each impedance in the domain **I** with resistance and reactance ranges as specified above. Fig.59 depicts a subset of domain **I** containing highest attenuation impedances for these four frequencies. The frequency, 1000 Hz, yields an attenuation of 42.7 dB for a uniform liner of impedance = 0.9-0.8i. But this same uniform liner yields much lower attenuations of 2.83, 4.78, and 2.91 dB for 1500, 2000, and 2500 Hz respectively.

The simulations that follow represent this scenario. A uniform liner is arbitrarily set to impedance = 1.1-1.1*i*, which yields a relatively low attenuation for all four frequencies (24.82 dB for 1000 Hz, 14.40 for 1500 Hz, 6.85 dB for 2000 Hz, and 3.98 dB for 2500 Hz). The initial frequency is set to 1500 Hz. The liner will adapt until it determines the highest attenuation possible and holds that liner definition. The frequency changes to 1000 Hz and the attenuation decreases. The liner then adapts until it determines the highest attenuation possible and holds that liner definition. The frequency changes to 2000 Hz and the attenuation decreases. The liner then adapts until it determines the highest attenuation possible and holds that liner definition. The frequency then changes to 2500 Hz and the attenuation decreases. Finally, the liner adapts until it determines the highest attenuation possible and holds that liner definition. In the first simulation, local decisions and limited cooperation are used to adapt to the best uniform liner. Next, logical neighborhoods and increased communal cooperation are used to adapt to a segmented liner that produces higher attenuation than the best uniform liner.

	-1.8	-1.7	-1.6	-1.5	-1.4	-1.3	-1.2	-1.1	-1	-0.9	-0.8	-0.7
0.7												
0.8												
0.9											**42.7**	1000
1												
1.1												
1.2												
1.3								**25.01**	1500			
1.4												
1.5												
1.6												
1.7												
1.8				**17.62**	2000							
1.9												
2												
2.1		**14.81**	2500									
2.2												

Fig. 59. Maximum attenuation for uniform liner at selected frequencies.

Simulation of Adaptation to Best Uniform Liner

Segments tune their resonators in unison to each impedance in the domain neighborhood surrounding the initial impedance (i.e., CDUCT-LaRC is executed with a uniform liner of each impedance). Each segment records its acoustic power level for these 9 tunings. Each segment, except segment, S_{15}, in Fig. 58 (rightmost, exit segment), requests the 9 acoustic power levels of the adjacent segment towards the exit (to the right in Fig. 58). Using the acoustic power level of the adjacent segment and its own corresponding acoustic power level, it computes a delta power level for each impedance. For each neighboring impedance, *i*, acoustic power level for the segment, S_i, and acoustic power level of the adjacent segment A_i, ΔPWL_i is:

$$\Delta PWL_i = \quad 10 * \log(S_i / A_i) \tag{30}$$

Each segment then sets its impedance to that producing the largest delta power level for the neighborhood. If this impedance is different from the impedance of the center of the neighborhood, it forms the center of a new neighborhood and the process repeats. If not, the process terminates. Segment, S_{14}, dictates its decision to segment, S_{15}. Fig 60 depicts this process for the first neighborhood, where the columns "Rs" and "Ra" are the resistance and reactance for each neighbor, "G dB" is the global attenuation resulting from a uniform liner of that resistance and reactance, and "S_i" is the acoustic power level for segment, S_i. Note that even though each segment (except for segment,

Rs	Ra	G dB	S_0	S_1	S_2	S_3	S_4	S_5	S_6	S_7	S_8	S_9	S_{10}	S_{11}	S_{12}	S_{13}	S_{14}
1.0	-1.2	11.96	1.27	0.83	0.80	0.77	0.75	0.74	0.73	0.72	0.72	0.72	0.72	0.72	0.72	0.720	1.083
1.0	-1.1	12.17	1.30	0.85	0.82	0.78	0.76	0.75	0.74	0.73	0.73	0.73	0.73	0.73	0.73	0.727	1.091
1.0	-1.0	12.02	1.28	0.84	0.80	0.77	0.75	0.74	0.73	0.73	0.72	0.72	0.72	0.72	0.72	0.715	1.073
1.1	-1.2	13.89	1.48	0.97	0.94	0.90	0.88	0.86	0.85	0.84	0.83	0.83	0.83	0.83	0.83	0.831	1.251
1.1	-1.1	14.40	1.54	1.01	0.98	0.94	0.91	0.89	0.88	0.87	0.86	0.86	0.85	0.85	0.85	0.849	1.275
1.1	-1.0	14.29	1.53	1.01	0.97	0.93	0.90	0.89	0.87	0.86	0.86	0.85	0.84	0.84	0.84	0.838	1.255
1.2	-1.2	16.02	1.70	1.11	1.08	1.05	1.02	1.00	0.98	0.97	0.96	0.95	0.95	0.95	0.95	0.954	1.437
1.2	-1.1	17.26	1.86	1.23	1.19	1.14	1.11	1.08	1.06	1.04	1.03	1.02	1.01	1.01	1.00	1.000	1.500
1.2	**-1.0**	**17.72**	**1.93**	**1.27**	**1.23**	**1.18**	**1.14**	**1.11**	**1.09**	**1.07**	**1.06**	**1.04**	**1.03**	**1.02**	**1.01**	**1.008**	**1.504**

Fig. 60. Power levels for initial neighborhood surrounding impedance, 1.1-1.1*i*.

S_{15}) autonomously determines it new value after cooperating with only one adjacent neighbor, each segment comes to the same conclusion for best impedance, which also agrees with the maximum global attenuation. The choices of best acoustic power level for each segment and the best global attenuation are highlighted as bold characters.

The following describes the process for adapting to the best uniform liner for the four frequencies 1500, 1000, 2000, and 2500 Hz. The uniform impedance is initially set to 1.1-1.1*i* resulting in global attenuation of 14.4 dB. Tuning to all neighboring impedances, each segment examines its delta power level and selects the impedance resulting in the highest delta power level for the neighborhood: 1.2-1.0*i*. As stated, this selection also produces the highest global attenuation, 17.72 dB. Fig. 61 depicts that neighborhood and the global attenuation for each impedance. Each segment next sets its impedance to 1.2-1.0*i* resulting in a uniform liner producing attenuation of 17.72 dB. This represents the 2nd neighborhood also depicted in Fig 61. Selection of the best uniform liner of this neighborhood is at impedance, 1.3-1.1*i* resulting in global attenutation, 25.01. Each segment finally sets its impedance to 1.3-1.1*i* resulting in a uniform liner producing attenuation of 25.01 dB. The resulting 3rd neighborhood is also depicted in Fig 61. Attenuation in bold is that of the neighborhood center while that in bold with a gray background is the highest attenuation for the neighborhood. As there is no neighboring impedance that produces a delta power level for any segment higher than that of impedance, 1.3-1.1*i*, the process terminates and the uniform liner resulting in the highest attenuation has been determined.

With uniform liner impedance remaining at 1.3-1.1*i*, the frequency changes to 1000 Hz resulting in a reduction of the attenuation to 20.6 dB. The process begins again as each segment adjusts its impedance to all selections in the neighborhood and selects the impedance of the highest delta power level as the next neighborhood center. Fig 62 details the progression as the global attenuation increases from 20.60 dB to a high of 42.70 at a uniform liner impedance of 0.9-0.8*i*.

	-1.2	**-1.1**	**-1**
1	11.96	12.17	12.02
1.1	13.89	**14.40**	14.29
1.2	16.02	17.26	**17.72**

1st Neighborhood

	-1.1	**-1**	**-0.9**
1.1	14.40	14.29	13.37
1.2	17.26	**17.72**	15.48
1.3	**25.01**	21.09	16.92

2nd Neighborhood

	-1.2	**-1.1**	**-1**
1.2	16.02	17.26	17.72
1.3	18.05	**25.01**	21.09
1.4	23.55	21.97	19.53

3rd & Last Neighborhood

Fig. 61. Adapting to best uniform liner for 1500 Hz.

With uniform liner impedance remaining at 0.9-0.8*i*, the frequency changes to 2000 Hz resulting in a reduction of the global attenuation to 4.78 dB. The process begins again as each segment adjusts its impedance to all selections in the neighborhood and selects the impedance of the highest delta power level as the next neighborhood center. Fig 63 details the progression as the global attenuation increases from 4.78 dB to a high of 17.62 at a uniform liner impedance of 1.8-1.5*i*.

This process is repeated as the frequency changes to 2500 and the global attenuation for the uniform liner of impedance, 1.8-1.5*i*, reduces to 8.51 dB. The adaptation, depicted in Fig. 64, increases the global attenuation to 14.81 dB from a uniform liner of impedance, 2.1-1.7*i*.

The complete path from the initial uniform liner impedance of 1.1-1.1*i* producing attenuation of 14.4 dB at 1500 Hz, to a final uniform liner impedance of 2.1-1.7*i* producing attenuation of 14.81 dB is detailed in the plot of Fig. 65.

	-1.2	-1.1	-1
1.2	21.13	22.56	**23.70**
1.3	19.57	**20.60**	21.34
1.4	18.15	18.91	19.42

1st Neighborhood

	-1.1	-1	-0.9
1.1	24.82	26.62	**27.76**
1.2	22.56	**23.70**	24.27
1.3	20.60	21.34	21.65

2nd Neighborhood

	-1	-0.9	-0.8
1	30.13	32.69	**32.90**
1.1	26.62	**27.76**	27.55
1.2	23.70	24.27	24.01

3rd Neighborhood

	-0.9	-0.8	-0.7
0.9	39.52	**42.70**	33.54
1	32.69	**32.90**	29.39
1.1	27.76	27.55	25.71

4th Neighborhood

	-0.9	-0.8	-0.7
0.8	41.96	23.66	25.73
0.9	39.52	**42.70**	33.54
1	32.69	32.90	29.39

5th & Last Neighborhood

Fig. 62. Adapting to Best Uniform Liner for 1000 Hz.

	-0.9	-0.8	-0.7
0.8	4.40	4.22	4.02
0.9	5.00	**4.78**	4.55
1	**5.61**	5.34	5.07

1st Neighborhood

	-1	-0.9	-0.8
0.9	5.21	5.00	4.78
1	5.87	**5.61**	5.34
1.1	**6.55**	6.24	5.91

2nd Neighborhood

	-1.1	-1	-0.9
1	6.11	5.87	5.61
1.1	6.85	**6.55**	6.24
1.2	**7.64**	7.26	6.87

3rd Neighborhood

	-1.2	-1.1	-1
1.1	7.10	6.85	6.55
1.2	7.96	**7.64**	7.26
1.3	**8.91**	8.47	7.99

4th Neighborhood

	-1.3	-1.2	-1.1
1.2	8.21	7.96	7.64
1.3	9.24	**8.91**	8.47
1.4	**10.45**	9.97	9.36

5th Neighborhood

	-1.4	-1.3	-1.2
1.3	9.41	9.24	8.91
1.4	10.67	**10.45**	9.97
1.5	**12.26**	11.95	11.15

6th Neighborhood

	-1.5	-1.4	-1.3
1.4	10.61	10.67	10.45
1.5	12.02	**12.26**	11.95
1.6	13.66	**14.46**	14.01

7th Neighborhood

	-1.5	-1.4	-1.3
1.5	12.02	12.26	11.95
1.6	13.66	**14.46**	14.01
1.7	15.27	**17.25**	15.92

8th Neighborhood

	-1.5	-1.4	-1.3
1.6	13.66	14.46	14.01
1.7	15.27	**17.25**	15.92
1.8	**17.62**	16.71	15.02

9th Neighborhood

	-1.6	-1.5	-1.4
1.7	13.95	15.27	17.25
1.8	14.86	**17.62**	16.71
1.9	16.18	15.86	15.13

10th & Last Neighborhood

Fig. 63. Adapting to best uniform liner for 2000 Hz.

	-1.6	**-1.5**	**-1.4**
1.7	8.07	7.78	7.43
1.8	8.91	**8.51**	8.04
1.9	**9.92**	9.29	8.64

1st Neighborhood

	-1.7	**-1.6**	**-1.5**
1.8	9.16	8.91	8.51
1.9	10.30	**9.92**	9.29
2	**11.92**	11.17	10.08

2nd Neighborhood

	-1.8	**-1.7**	**-1.6**
1.9	10.28	10.30	9.92
2	11.60	**11.92**	11.17
2.1	13.1	**14.81**	12.3

3rd Neighborhood

	-1.8	**-1.7**	**-1.6**
2	11.60	11.92	11.17
2.1	13.1	**14.81**	12.3
2.2	14.5	13.6	12.1

4th & Last Neighborhood

Fig. 64. Adapting to best uniform liner for 2500 Hz.

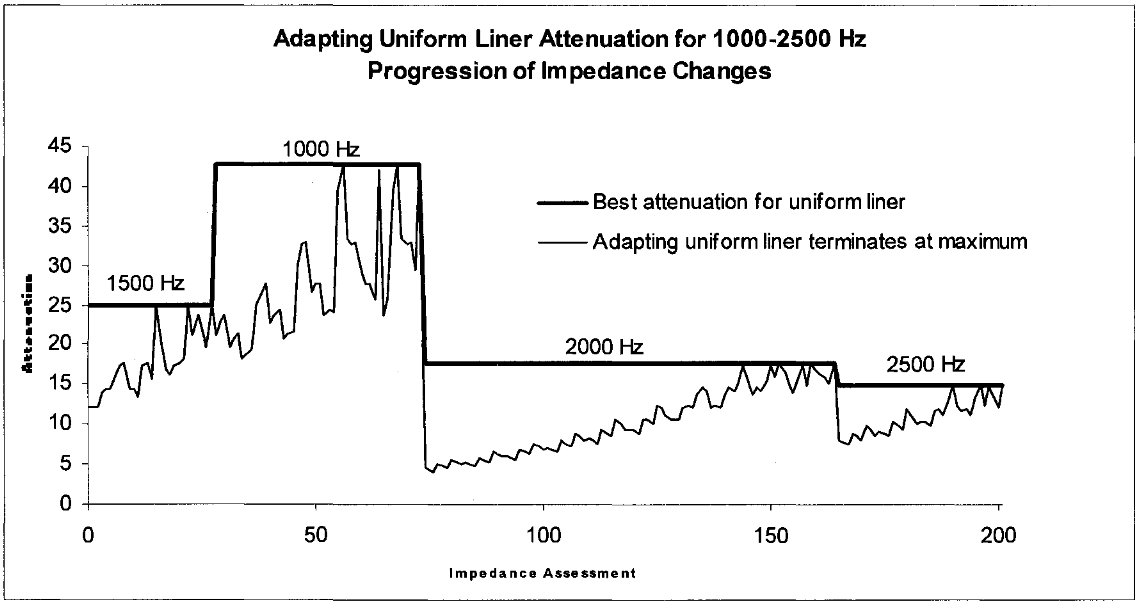


Fig. 65. Adapting uniform liner.

The advantages of this approach are:

- It determines the best uniform liner for a given frequency.
- It uses only local information (adjacent segments communicate but require no information from any other segments or any other source).

The limitations of this approach are:

- Most of the time during adaptation the global attenuation is significantly less than the maximum for a uniform liner.
- Although this works for a plane wave of one mode, the delta power levels of adjacent segments do not correlate with global attenuations when higher order modes are introduced.

Communication requirements of this approach are minimal. A sensor is assumed from which the acoustic power level may be computed at each segment. Although work could be more distributed, a simple design is to allocate a computation and communication node (CCN) for each segment. The CCN of each segment performs a series of impedance neighborhood assessments requiring only requests for the acoustic power level from its adjacent neighboring segment towards the exit (except the exit segment) and responses to requests for its acoustic power level from its adjacent neighboring segment towards the entrance (except for the entrance segment). Impedance neighborhood assessments must coordinated but occur in parallel. For each impedance change, a broadcast is made to all resonators in the segment. Transmission power of the CCN need only be strong enough to communicate with its adjacent segment's CCNs and its resonators.

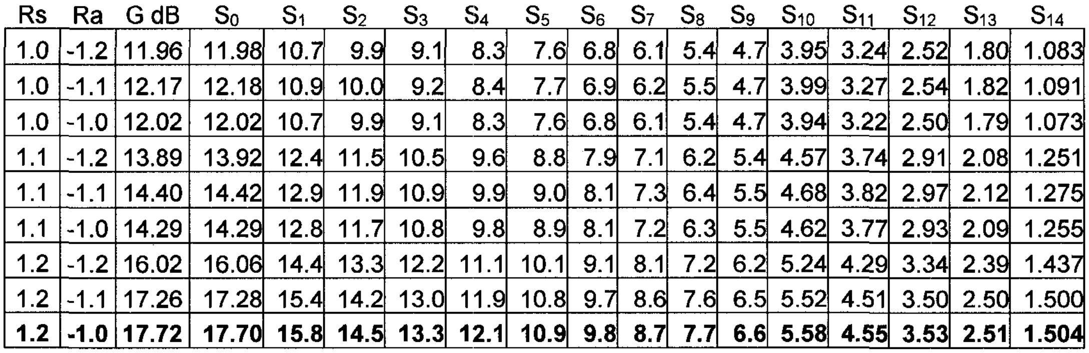
Simulation of Adaptation to Best Segmented Liner

In the last subsection, an algorithm was used whereby delta power levels among adjacent segments are used to determine the best uniform liner possible, but the algorithm will never generate a segmented liner. However, Section 2 showed that a segmented liner can produce higher attenuations than possible with a uniform liner. This section details modifications to the algorithm that will allow it to define a segmented liner providing higher attenuation than possible with the best uniform liner.

Fig. 60 shows that the delta power level between each adjacent segment correlates with the global attenuation for a given neighborhood of a uniform liner. However, with the adaptation to a segmented liner, higher order modes are introduced that alter this correlation such that the delta power level between each adjacent segment does not correlate with the global attenuation.

Fig. 66 details the relationship between global attenuation and delta power levels computed for each segment, S_i, and the exit segment, S_{15} for the same impedance neighborhood as in Fig 60. Notice that the impedance producing the highest delta power level for each segment again correlates with the global attenuation. Also, notice that the delta power level for segment, S_0, is very close to the global attenuation. It will be shown that this correlation is not destroyed by the introduction of a segmented liner and higher order modes. This relationship is still a locality as only two segments exchange information to determine the delta power level; no other information is required.

However, the distance for communication is increased.



Rs	Ra	G dB	S_0	S_1	S_2	S_3	S_4	S_5	S_6	S_7	S_8	S_9	S_{10}	S_{11}	S_{12}	S_{13}	S_{14}
1.0	-1.2	11.96	11.98	10.7	9.9	9.1	8.3	7.6	6.8	6.1	5.4	4.7	3.95	3.24	2.52	1.80	1.083
1.0	-1.1	12.17	12.18	10.9	10.0	9.2	8.4	7.7	6.9	6.2	5.5	4.7	3.99	3.27	2.54	1.82	1.091
1.0	-1.0	12.02	12.02	10.7	9.9	9.1	8.3	7.6	6.8	6.1	5.4	4.7	3.94	3.22	2.50	1.79	1.073
1.1	-1.2	13.89	13.92	12.4	11.5	10.5	9.6	8.8	7.9	7.1	6.2	5.4	4.57	3.74	2.91	2.08	1.251
1.1	-1.1	14.40	14.42	12.9	11.9	10.9	9.9	9.0	8.1	7.3	6.4	5.5	4.68	3.82	2.97	2.12	1.275
1.1	-1.0	14.29	14.29	12.8	11.7	10.8	9.8	8.9	8.1	7.2	6.3	5.5	4.62	3.77	2.93	2.09	1.255
1.2	-1.2	16.02	16.06	14.4	13.3	12.2	11.1	10.1	9.1	8.1	7.2	6.2	5.24	4.29	3.34	2.39	1.437
1.2	-1.1	17.26	17.28	15.4	14.2	13.0	11.9	10.8	9.7	8.6	7.6	6.5	5.52	4.51	3.50	2.50	1.500
1.2	**-1.0**	**17.72**	**17.70**	**15.8**	**14.5**	**13.3**	**12.1**	**10.9**	**9.8**	**8.7**	**7.7**	**6.6**	**5.58**	**4.55**	**3.53**	**2.51**	**1.504**

Fig. 66. Delta power level calculations for segments, S_i and S_{15}.

The algorithm in the last subsection terminates when each segment cannot find a neighboring impedance that provides higher attenuation. As all segments agree on this best impedance, the liner is always uniform. The following modifications produce an algorithm that does not terminate when the best uniform liner is determined, but continues to determine a segmented liner of higher attenuation. Initially, segment, S_0, is assigned as the *leader*, S_L. All segments towards the exit are slaves to this leader as they will accept as their impedance that which is determined by the leader. The leader segment, S_L, obtaining its own acoustic power level, PWL_L, and obtaining segment, S_0's acoustic power level, PWL_0, computes its delta power level, ΔPWL, as:

$$\Delta PWL_L \quad = \quad 10 * \log(PWL_L / PWL_0) \qquad (33)$$

The leader segment sets its impedance in turn to each within the current neighborhood of impedances. It computes the delta power level at each impedance and selects the impedance producing the highest delta power level. It sets itself to this impedance which forms the center of a new impedance neighborhood and dictates that each slave segment also set their resonators to this impedance. This process results in identical steps produced for the algorithm in the last subsection until the leader segment terminates by finding the best uniform impedance. At this point, leader segment, S_0, freezes its impedance and turns over leadership to segment, S_1. This process continues until segment, S_{14}, as leader, freezes its impedance and dictates this impedance to segment, S_{15}.

In the following, the progression described previously for adapting to the best liner for the four frequencies 1500 Hz, 1000 Hz, 2000 Hz, and 2500 Hz is demonstrated using this modified algorithm. Again, the uniform impedance is initially set to 1.1-1.1*i* resulting in global attenuation of 14.40 dB. After evaluating the third neighborhood, the best uniform liner is determined at impedance, 1.3-1.1*i* (resulting in global attenuation of 25.01 dB). As the leader segment can find no impedance in the neighborhood producing a greater attenuation, it freezes to the impedance, 1.3-1.1*i*, dictates this value to segments, S_1 .. S_{15}, and turns leadership over to segment, S_1. This state is depicted in Fig. 67. At the top of the figure is the liner of the duct with the entrance to the left and the exit to the right. The 16 columns in the image represent the 16 segments and are number as such at the bottom of the image (0 .. 15). Beneath the liner image is a subset of the impedance domain with entries highlighted and numbered if they are currently set in the liner. In the center of the liner segments are numbers corresponding the impedance value displayed in the domain. As all segments in the liner of Fig.67 are set to an equivalent impedance, this is a uniform liner.

With segment, S_0, frozen at impedance, 1.3-1.1*i*, segment, S_1, examines impedances in the neighborhood surrounding this impedance. The results are displayed in Fig. 68. Notice that the highest delta power level of 26.03 dB, which corresponds to the highest global attenuation of 28.62 dB, is produced when segment, S_1, and its slave segments, S_2 .. S_{15}, are set to impedance 1.3-1.2*i* while segment, S_0, is set to impedance, 1.3-1.1*i* (displayed in Fig. 69).

1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

	-1.4	-1.3	-1.2	-1.1	-1.0	-0.9	-0.8	-0.7
0.7								
0.8								
0.9								
1.0								
1.1								
1.2								
1.3				1				
1.4								

Fig. 67. Best uniform liner for 1500 Hz. S_0 is frozen, S_1 is leader.

This is already better than the best uniform liner (impedance = 1.3-1.1*i*, attenuation = 25.01), yet this is not the highest attenuation possible for leader segment, S_1. The process continues as segment, S_1, assesses a new neighborhood centered at impedance, 1.3-1.2*i*. It finds the highest delta power level (and therefore global attenuation) produced by impedance, 1.3-1.3*i*, and forms a new neighborhood at this center. For this neighborhood, it finds no neighbor producing a higher delta power level, freezes its impedance at 1.3-1.3*i*, and yields leadership to segment, S_2.

Rs	Ra	G dB	ΔPWL of S_1
1.2	-1.2	26.83	23.85
1.2	-1.1	24.05	21.15
1.2	-1.0	21.63	18.74
1.3	-1.2	**28.62**	**26.03**
1.3	-1.1	25.01	22.32
1.3	-1.0	22.25	19.50
1.4	-1.2	27.29	25.05
1.4	-1.1	24.49	22.01
1.4	-1.0	22.07	19.46

Fig. 68. Global attenuation and delta power for neighborhood centered at 1.3-1.0*i*.

Segment, S_2, assesses the neighborhood centered at impedance, 1.3-1.3*i*, and, in this first neighborhood for this leader, finds no neighbor producing a higher delta power level. It freezes its impedance at 1.3-1.3*i* and yields leadership to segment, S_2. This process continues until segment, S_{14}, freezes its impedance at its highest delta power level. The final global attenuation for this segmented liner is 33.5 dB, 34% higher than the best uniform liner. Fig. 70 displays the final segmented liner and the impedances assigned to its segments. Notice that the best impedance for segment, S_0, differs from that of S_1, but the impedance (and global attenuation = 31.92) remains fixed from S_1 .. S_5.

Segments, S_6 .. S_8, change slightly but significant improvements are made by segments, S_9 to S_{14}.

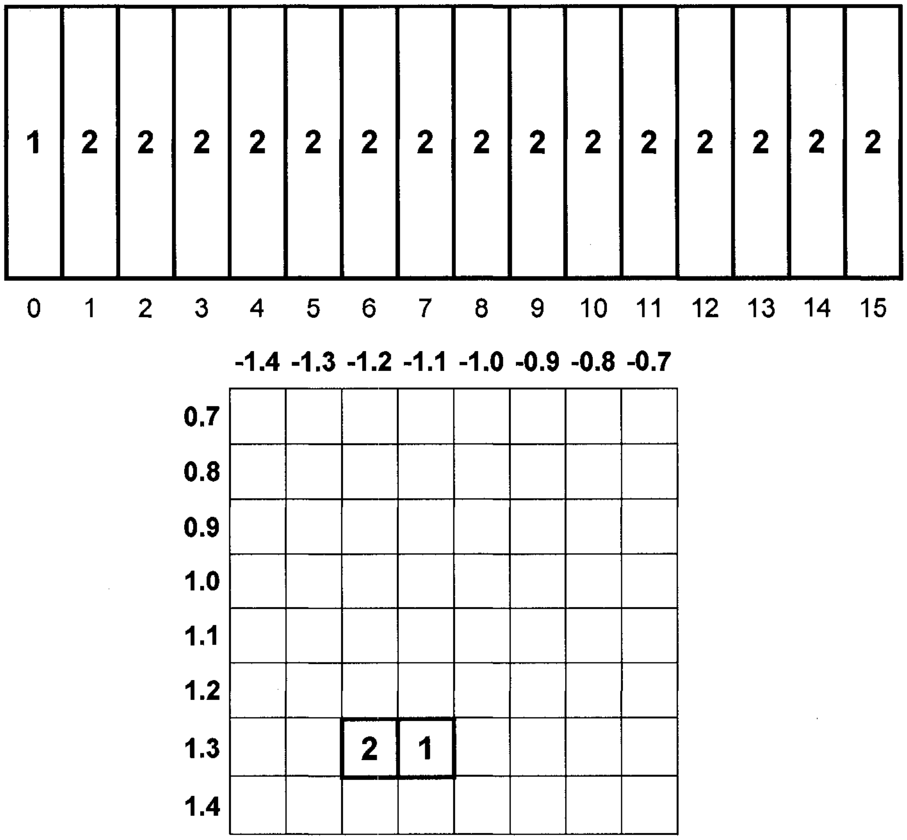


Fig. 69. Liner configuration for 1st neighborhood of segment, S_1, as leader.

The global attenuation resulting from the adapting impedance throughout this process is displayed in Fig. 71. Only 7 out of 243 adaptations (3%) produce lower global attenuations than the initial impedance and all of these occur in the assessment of the first two neighborhoods (of 27 = 7%). Also, only 13% of the assessments produce attenuations lower than the best attenuation of a uniform liner and all of these occur in the first five neighborhoods (of 27 = 19%).

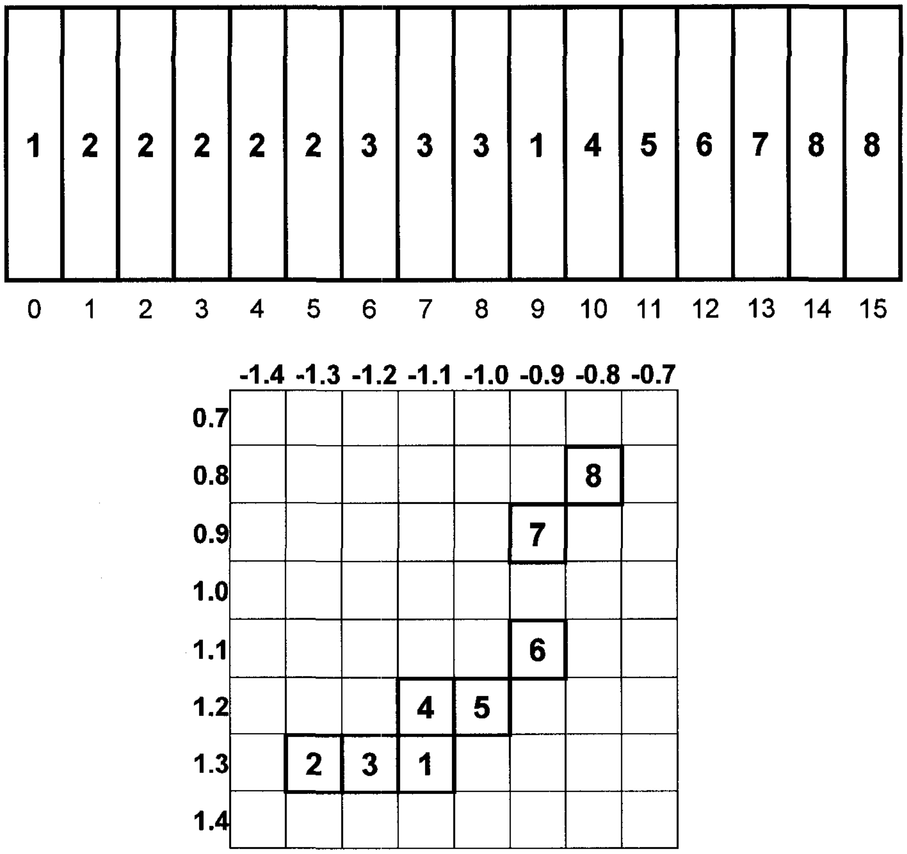


Fig. 70. Final segmented liner for 1500 Hz. Global attenuation = 33.5 dB.

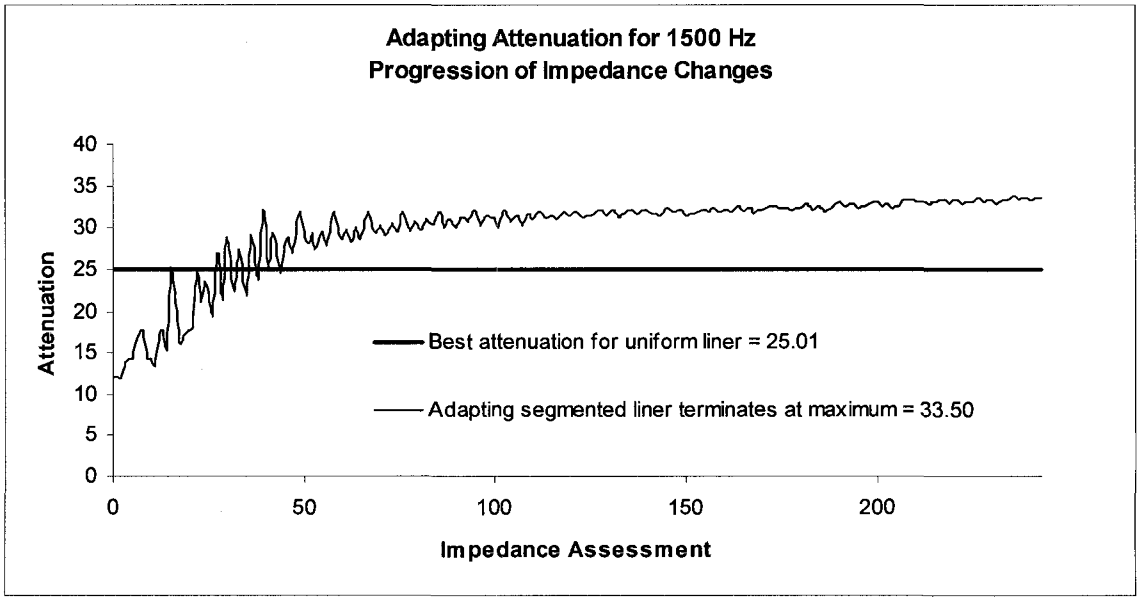


Fig. 71. Adaptation progression for segmented liner at 1500 Hz.

If the input frequency changes for the segmented liner displayed in Fig. 70 from 1500 Hz to 1000 Hz, the global attenuation drops from 33.5 dB to 21.11 dB. The process begins again. Segment, S_0, assumes leadership and dictates to segments, S_1 .. S_{15}, its impedance, 1.3-1.1*i*. The global attenuation decreases further to 20.6 dB. After 5 neighborhood assessments, segment, S_0, leads to the best uniform liner impedance for 1000 Hz of 0.9-0.8*i* yielding a global impedance of 42.7 dB. S_0 freezes at this impedance and turns leadership over to segment, S_1. Segment, S_1, assesses the neighborhood centered at impedance, 1.3-1.1*i*, and finds the highest delta power level still at impedance, 1.3-1.1*i*, freezes its impedance to this value and turns leadership to segment, S_1. This continues producing a global attenuation of 42.7 dB until segment, S_6, assumes leadership. It finds impedance, 0.9-0.9*i*, for itself and its slave segments, to produce a global attenuation of 43.35 dB. This process continues until segment, S_{14}, freezes its impedance resulting in the segmented liner displayed in Fig. 72 producing global attenuation of 46.57 dB.

The global attenuation resulting from the adapting impedance throughout this process is displayed in Fig. 73. This time it takes more neighborhood assessments (5 vs. 3) to achieve the best uniform liner as the starting impedance is farther across the impedance domain than the progression described for 1500 Hz. It also takes more neighborhood assessments (i.e., time) to increase the global attenuation above the best for a uniform liner because, for this frequency, the liner remains uniform for the first 6 segments. During this time, 8 neighborhood assessments out of the 9 in the neighborhood produce global attenuations below that of the best uniform liner. Nevertheless, the remaining ten segments modify their impedance to improve the global attenuation for the segmented liner from 42.7 to 46.57 dB.

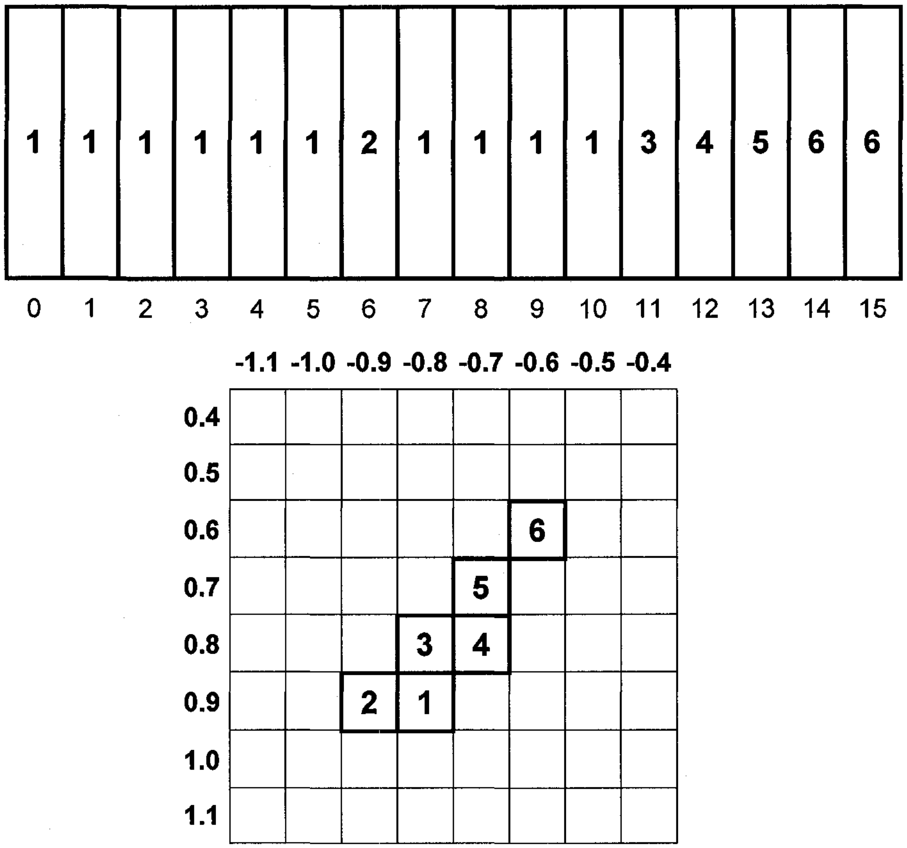


Fig. 72. Final segmented liner for 1000 Hz. Global attenuation = 46.57 dB.

If the frequency changes from 1000 Hz to 2000 Hz with the segmented liner displayed in Fig. 72, the global attenuation drops from 46.57 dB to 4.66 dB. The process begins again as segment, S_0, assumes leadership and dictates its impedance, 0.9-0.8*i*, to segments, S_1 .. S_{15}, which increases the global attenuation to 4.78 dB. The distance across the impedance domain between this initial impedance and the impedance for the best uniform liner for 2000 Hz, 1.8-1.5i, is greater than initially traversed for the best segmented liner for 1500 Hz to the best uniform liner for 1000 Hz. Nine neighborhood assessments are required for this adaptation. Attaining its best global attenuation of 17.62 dB at impedance, 1.8-1.5*i*, segment, S_0, freezes at this impedance, yields leadership to segment, S_1. Segment, S_1, requires 5 neighborhood assessments to freeze on the impedance, 1.6-1.9*i*, producing a global attenuation of 21.5 dB. Segments, S_2 and S_3, freeze on the first neighborhood assessment to this same impedance and, therefore, do not

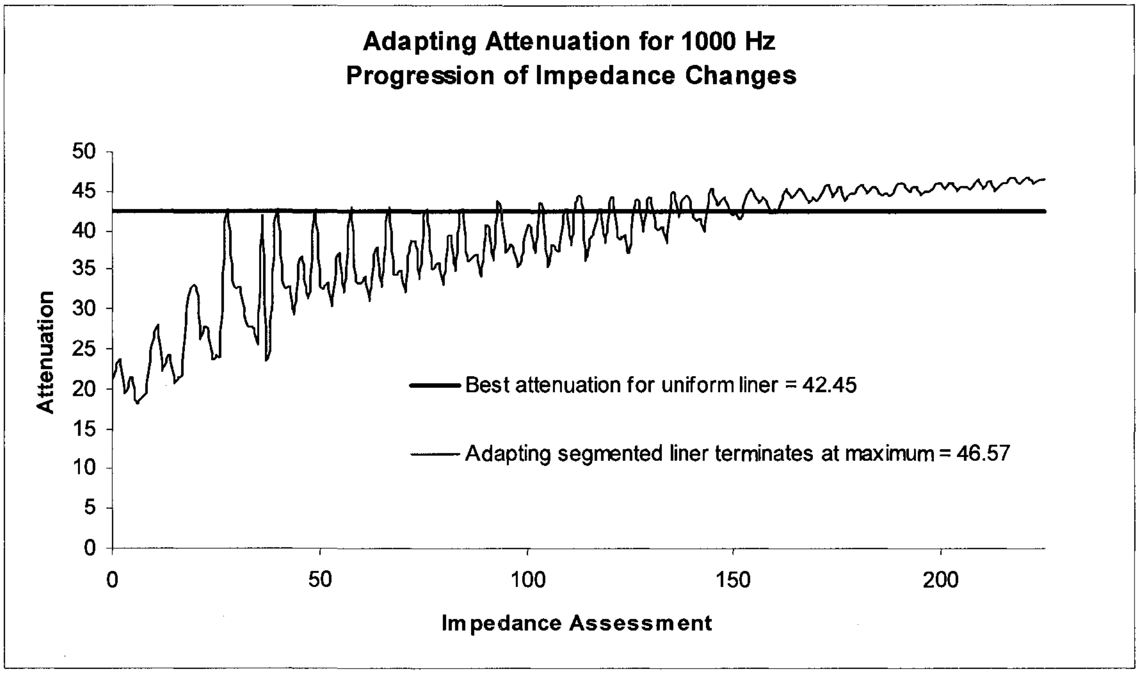
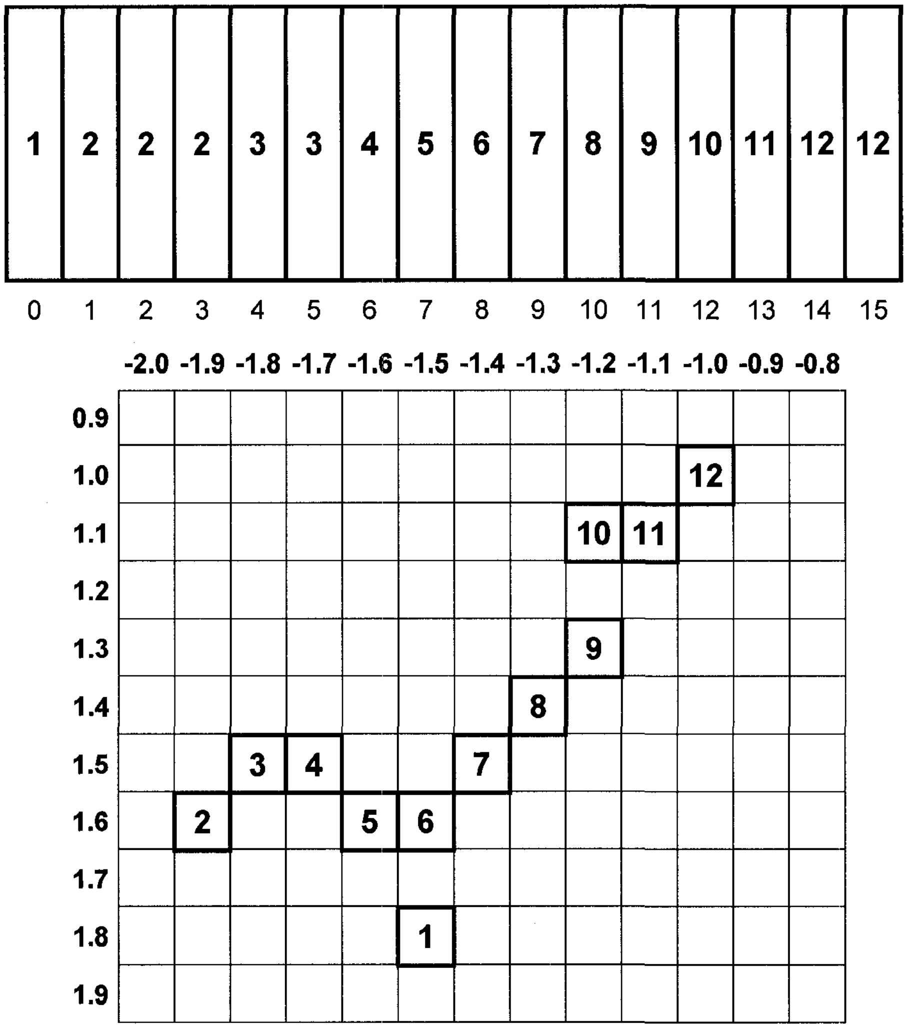


Fig. 73. Adaptation progression for segmented liner at 1000 Hz.

improve the global attenuation. In two neighborhood assessments, segment, S_4, improves the global attenuation to 21.87 dB, but segment, S_5, freezes at the same impedance an does not improve the global attenuation. From this point on, each segment improves the global attenuation until segment, S_{14}, freezes resulting in the final and best global attenuation of 23.33 dB. The liner configuration for this result is displayed in Fig. 74 and the global attenuation resulting from the adapting impedance throughout this process is displayed in Fig. 75.



	-2.0	-1.9	-1.8	-1.7	-1.6	-1.5	-1.4	-1.3	-1.2	-1.1	-1.0	-0.9	-0.8
0.9													
1.0											12		
1.1									10	11			
1.2													
1.3									9				
1.4								8					
1.5			3	4			7						
1.6		2			5	6							
1.7													
1.8						1							
1.9													

Fig. 74. Final segmented liner for 2000 Hz. Global attenuation = 23.33 dB.

The final frequency change is to 2500 Hz. At this frequency, the segmented liner definition in Fig. 74 produces a global attenuation of 8.31 dB. Segment, S_0, dictates its impedance, 1.8-1.5*i*, to segments, S_1 .. S_{15}, forming a uniform impedance which increases the global attenuation to 8.51 dB. After 4 neighborhood assessments, the best uniform liner is attained at impedance, 2.1-1.7*i*, providing global attenuation of 14.81 dB.

Leadership is passed to segment, S_1. After 7 neighborhoods assessments, segment, S_1, freezes on impedance, 2.7-1.9*i*, increasing the global attenuation to 16.26 dB. The process continues with each segment freezing on a different impedance to terminate with the configuration displayed in Fig. 76 producing final global attenuation of 17.33 dB (an improvement of 17% above the best uniform liner). The adaptation progression is plotted in Fig. 77.

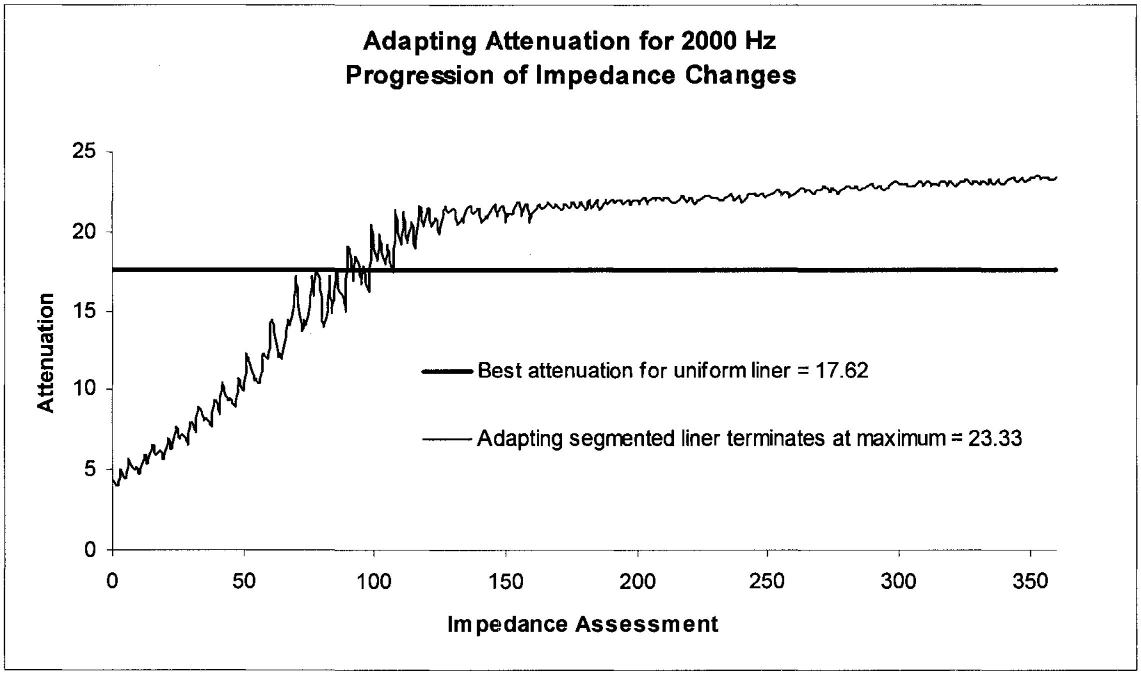
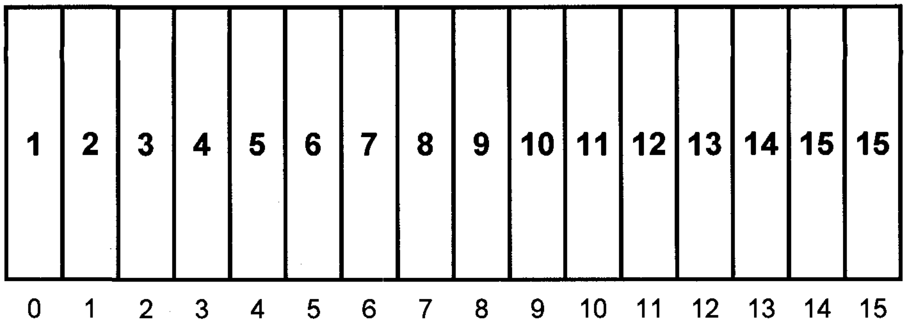


Fig. 75. Adaptation progression for segmented liner at 2000 Hz.



	-2.0	-1.9	-1.8	-1.7	-1.6	-1.5	-1.4	-1.3	-1.2	-1.1	-1.0	-0.9
1.1												
1.2											15	
1.3									14			
1.4							13					
1.5							12					
1.6								11				
1.7												
1.8								10				
1.9												
2.0								9				
2.1				1								
2.2							8					
2.3												
2.4						7						
2.5			5		6							
2.6			4									
2.7		2	3									
2.8												

Fig. 76. Final segmented liner for 2500 Hz. Global attenuation = 17.33 dB.

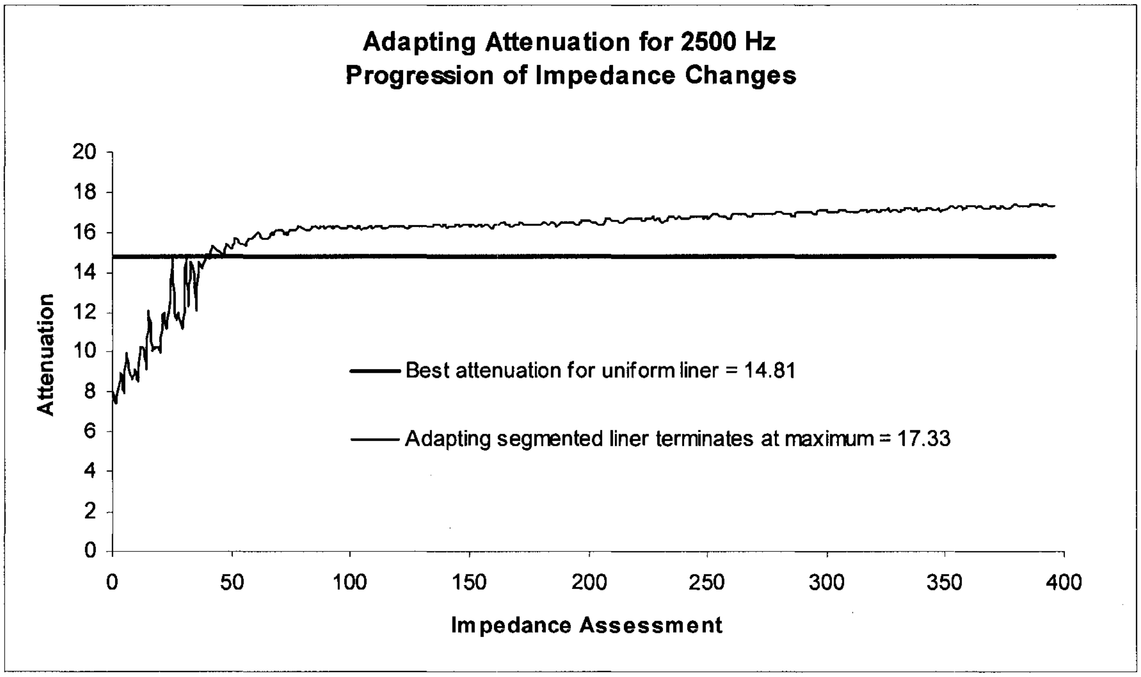


Fig. 77. Adaptation progression for segmented liner at 2500 Hz.

Fig. 78 combines the adaptation process sequentially for the frequencies 1500, 1000, 2000, and 2500 Hz. This plot exemplifies the problem with a static, passive liner: there is no uniform liner that provides the level of noise reduction that can be attained with an adaptable liner (e.g., a duct with the best uniform liner for 1000Hz, when faced with a change in frequency to 2000 Hz, reduces the global attenuation from 42.7 dB to 4.78 dB). Furthermore, it shows that a segmented liner is more effective in this instance than a uniform liner as, most of the time throughout the sequence, the global attenuation is higher than could be attained by a uniform liner. The times when the global attenuation falls below the best for a uniform liner are minimal correction periods. Fig. 78 is somewhat unrealistic as it depicts a change in frequency immediately following attainment of the best segmented liner for the previous frequency. In flight, the process for determining the best segmented liner for takeoff would take seconds followed by minutes of climbing, during which the acoustic signature remains relatively stable. During the transition from takeoff to cruising, the adaptation would occur in seconds followed by a much longer period of stability during cruising. The process could be configured so that it would only restart when attenuation fell below a threshold.

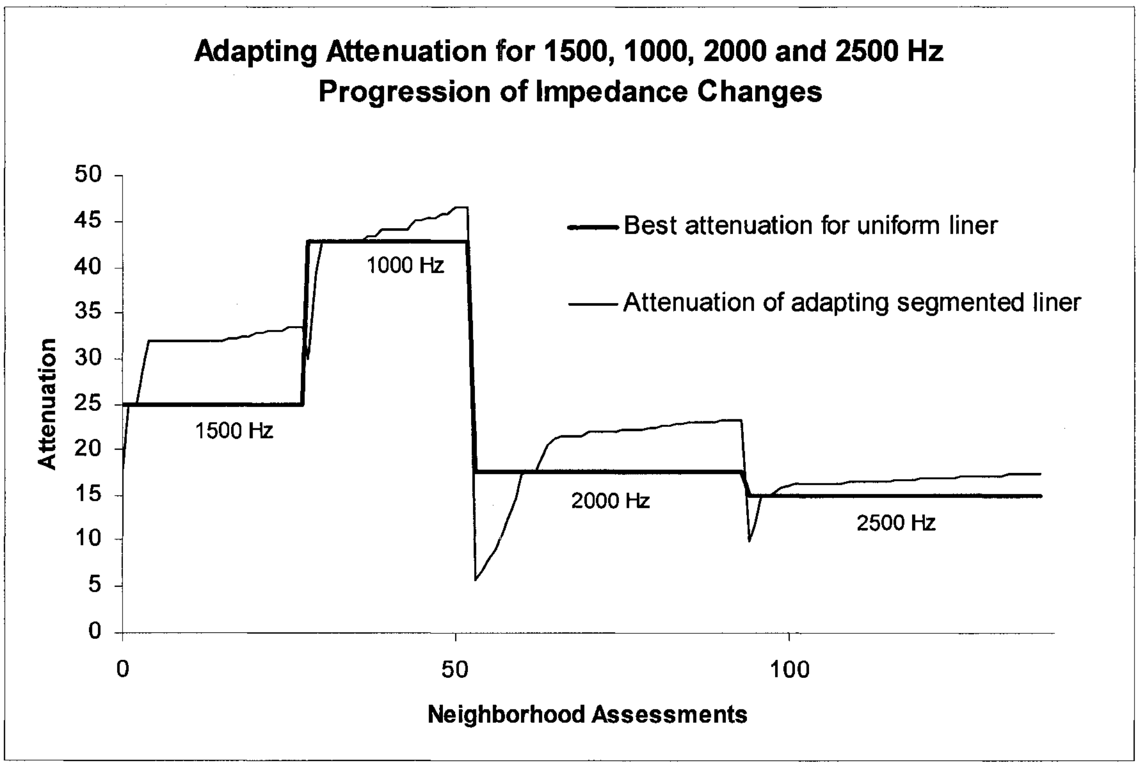


Fig. 78. Adaptation to best segmented liner for 4 frequencies.

Communication requirements of this approach are simple though over greater distances than the approach for the best uniform liner. A sensor is assumed that provides the acoustic power level for each segment. A computation and communication node (CCN) is allocated for each segment. The leader segment's CCN performs a series of impedance neighborhood assessments requiring only requests and responses for the acoustic power level of the exit segment. This request and response could be accomplished by a multi-hop route across the segments, but as the distance is relatively short (at maximum, the length of the liner), this could be a point-to-point broadcast. For each impedance change, the CCN of the leader segment broadcasts the value to resonators of the leader segment and all slave segments. Communication is not required with frozen segments.

Either of the two methods described could be used in design of a nacelle to predetermine a table of liner definitions for anticipated acoustic conditions. Simplistically, three liner definitions could be predetermined for an aircraft in take-off, cruise, and landing conditions. A centralized computer could determine the change in conditions, use the table to determine impedance definitions for that condition, and dictate these values to liner resonators. This would address the first challenge stated in Section 2, that acoustic conditions change over the flight regime, but would not address the other two: that theoretical characteristics of the design may not be met in the construction of the liner material or physical changes in the nacelle over time may change its absorption properties. It would also not account for other environmental factors such and atmospheric conditions. All of these could be addressed by placing sensors to determine the immediate conditions, but many sensors would have to be placed, collected from, and maintained to gather all necessary data. But the cost of construction and maintenance and the complexity of the installation and operation are unnecessary. The methods described here can accomplish the task by adapting the liner impedance in real time by collecting relatively little information from a local view.

SECTION 5

CONCLUSION

Research in wireless sensor networks has accelerated rapidly in recent years. The promise of ubiquitous control of the physical environment opens the way for new applications that will redefine the way we live and work. Due to the small size and low cost of sensor devices, visionaries promise smart systems enabled by deployment of massive numbers of sensors working in concert. To date, most of the research effort has concentrated on forming ad hoc networks under centralized control, which is not scalable to massive deployments. This thesis proposes an alternative approach based on models inspired by biological systems and reports significant results based on this new approach. This perspective views sensor devices as autonomous *organisms* in a *community* interacting as part of an *ecosystem* rather than as nodes in a computing network. The networks that result from this design make local decisions based on local information in order for the network to achieve global goals. Thus we must engineer for emergent behavior in wireless sensor networks.

First we implemented a simulator based on cellular automata to be used in algorithm development and assessment. Then we developed efficient algorithms to exploit emergent behavior for finding the average of distributed values, synchronizing distributed clocks, and conducting distributed binary voting. These algorithms are shown to be convergent and efficient by analysis and simulation. Finally, an extension of this perspective is used and demonstrated to provide significant progress on the noise abatement problem for jet aircraft. Using local information and actions, optimal impedance values for an acoustic liner are determined in situ providing the basis for an adaptive noise abatement system the provides superior noise reduction compared with current technology and previous research efforts.

Alternative Architecture for Sensor Networks Modeled After a Natural Ecosystem

Our research recognizes that the sense and send, centralized architecture severely limits the potential of wireless sensor networks. Instead, we developed an alternate, more versatile view of a sensor network. A wireless sensor node is an autonomous entity containing the ability to sense the environment, compute, remember, and communicate with its immediate neighboring nodes. Limiting use of the sensor node to sense and send data to a central controller or to route data from a neighboring node toward the central controller is merely a choice of the designer. Alternatively, we view these autonomous nodes as organisms dispersed within the environment to form an ecosystem. As autonomous organisms, they are free to make their own decisions based on local observations shared only with neighboring nodes. As neighborhoods iterate in this process of making local observations and taking local actions, a global behavior emerges. Our challenge was to develop algorithms specifying the iterations of local observations and actions, which result in desired global behavior.

To simulate algorithms implemented using this methodology, a simulator was developed using cellular automata. The cellular automaton was used because of its similarity to a wireless sensor network distribution. Neighboring cells represent sensor nodes that can directly communicate via a wireless transmission.

Initially, we developed an algorithm for calculating the global average assuming a random distribution of an observed value. Obviously, the average can be calculated if a central controller collects values from all nodes. But in our algorithm, no authority ever knows the values of all nodes. Nodes iteratively contract with any available neighbors to participate in an atomic operation, a Cell Value Change Operation (CVCO). This process continues until all neighborhoods recognize that the value of all neighbors is equivalent within a tolerance. Thus, each node is set to a value within a tolerance of the global average without any node, or central authority, ever knowing all values within the network.

This algorithm is modified to solve the problem of majority rule where the initial distribution is a binary value. The global goal is to determine which of the binary values is in the majority for the initial distribution. Again, rather than collect a global count of nodes set to each value, a CVCO iterates among nodes implementing the averaging algorithm. If the average is above 0.5, the node is assumed to contain the value of 1 else the value is assumed to be 0. Simulations demonstrate that the correct global decision is derived through sufficient iterations.

The algorithm is further modified for use in synchronization. The averaging technique is used to synchronize clock values and this approach is compared with another algorithm that propagates the synchronized value across the network.

The simulations prove that these algorithms demonstrate two properties rendering them far superior to a centralized approach. Using a centralized approach, both the time for convergence and the amount of transmissions (energy) required increase geometrically with the size of the network. Using the averaging algorithm, the time required for convergence and the amount of transmissions for large networks remains consistent. Another valuable property is that the range of values reduces and approaches the global average rapidly during the process.

A proof is provided that convergence will occur within a finite time. Simulations show that the time required for convergence is much less than that guaranteed by the proof. Another proof is provided that the global average remains the same with each CVCO and that upon convergence, the global average is the same as the initial global average. While each neighborhood terminates when all neighbors are within a local tolerance, this does not guarantee all nodes values are within that same local tolerance. A proof is provided defining the local tolerance required to guarantee a desired global tolerance.

One of the great advantages of this approach is that CVCOs can occur in parallel. A method is detailed to maximize this parallelization. Also, the neighborhood size can be modified to include more or less nodes in a neighborhood and results are presented to determine the optimal neighborhood size to minimize convergence time and required transmissions.

Real Time Determination of Optimal Acoustic Liner

Current technology for noise abatement in aircraft engines is dependent on acoustic liners composed of static, passive resonators of uniform impedance. New technology allows the impedance of resonators to be modified in situ permitting the adaptation of the liner impedance to changing acoustic conditions. Furthermore, research has determined that a liner of uniform impedance does not provide the optimal noise abatement. This dissertation addresses the issue of how to dynamically determine optimal segment impedances in situ as acoustic conditions change.

We produced an algorithm whereby an optimal uniform liner is determined for changing acoustic conditions using only local and limited cooperation among adjacent segments. The optimal uniform impedance emerges from local information and cooperation. This eliminates single point of failure and other problems created by a centrally controlled architecture. Expanding cooperation to non-adjacent segments, optimal, non-uniform segment impedances emerge from limited cooperation and communication. Superior impedance patterns emerge that could not be determined by prior design methods.

A Range of Cooperation in Sensor Communities

The major point of this dissertation is that wireless sensor nodes should not be viewed as nodes in a computing network but rather as the autonomous entities that are dispersed in and interact with a physical environment. It is not our belief that all applications can be accomplished from local cooperation and action. In some applications, a sense and send, centralized architecture is appropriate. In other applications, as demonstrated, global behavior can emerge from local information and action. In other applications, as demonstrated, more distant information and cooperation is required. The task of the designer is to determine the least amount of cooperation required to accomplish the global task

Future Research

It seems intuitive that distributing work among multiple computers would add robustness to the system resulting in higher reliability and more fault tolerance, but this is not guaranteed. Unlike the centralized approach where the central controller monitors the health of individual sensor nodes and compensates globally for failure of nodes, the compensation in a communal approach could emerge from the local decisions made by functional nodes. However, no work was done to prove this for any of the applications studied. It is anticipated that the simulators could easily be modified to introduce node failure forcing remaining functional nodes to compensate for this failure. A natural extension to this work is to introduce such failure and study how this affects both the averaging application and the adaptive noise suppression application.

BIBLIOGRAPHY

1. B. Warneke, M. Last, B. Leibowitz, and K. Pister, “SmartDust: communicating with a cubic-millimeter computer,” *IEEE Computer*, vol. 34, no.1, pp. 44-55, 2001.
2. V. Zhirnov and D. Herr, “New frontiers: self-assembly and nano-electronics,” *IEEE Computer*, vol. 34, no.1, pp. 34-43, 2001.
3. Xbow Technologies, Retrieved April 17, 2005 from http://www.xbow.com
4. P. Saffo, “Sensors, the next wave of innovation,” *Communications of the ACM*, vol. 40, no. 2, pp. 93-97, 1997.
5. I. Akyildiz, W. Su, Y. Sankarasubramanian, and E. Cayirci, “Wireless sensor networks: A survey,” *Computer Networks*, vol. 38, no. 4, pp. 393-422, 2002.
6. National Research Council, *Embedded Everywhere*, National Academy Press, 2001.
7. J. Kahn, R. Katz, and K. Pister, “Next century challenges: Mobile support for Smart Dust,” *Proc. ACM MOBICOM*, pp. 271-278, Seattle, WA, August 1999.
8. “29 Palms Fixed/Mobile Experiment”, Retrieved April 17, 2005 from http://robotics.eecs.berkeley.edu/~pister/29Palms0103
9. D. Lammers, “Embedded Projects Take a Share of Intel's Research Dollars,” *EE Times*, August 28, 2001. Retrieved April 5, 2004, from http://today.cs.berkeley.edu/800demo/eetimes.html
10. S. Park, I. Locher, A. Savvides, M. Srivastava, A. Chen, R. Muntz, and S. Yue, “Design of a Wearable Sensor Badge For Smart Kindergarten,” *Proc. 6th International Symposium on Wearable Computers*, Seattle, WA, October, 2002.
11. K. Ryokai and J. Cassell, “StoryMat: A play space for collaborative storytelling,” *Proc. CHI’99*, October 1999.
12. M. Srivastava, R. Muntz, and M. Potkonjak, “Smart Kindergarten: Sensor-based Wireless Networks for Smart Developmental Problem-solving Environments,” *Proc. ACM MOBICOM*, Rome, Italy, July 2001.
13. D. Estrin, R. Govindan, J. Heidemann, and S. Kumar, “Next Century Challenges: Scalable Coordination in Sensor Networks,” *Proc. MOBICOM*, Seattle, WA, August 1999.
14. J. Polastre, R. Szewcyk, A. Mainwaring, D. Culler, and J. Anderson, “Analysis of Wireless Sensor Networks for Habitat Monitoring,” in *Wireless Sensor Networks*, Raghavendra, Sivalingam, and Znati, Eds., Kluwer Academic, pp. 399-423, 2004.
15. R. Szewczyk, J. Polastre, A. Mainwaring, J. Anderson, and D. Culler, “An Analysis of a Large Scale Habitat Monitoring Application,” *Proc. 2nd ACM Conference on Embedded Networked Sensor Systems*, Nov. 2004.
16. R. Szewczyk, J. Polastre, A. Mainwaring, J. Anderson, and D. Culler, “An Analysis of a Large Scale Habitat Monitoring Application,” *Proc. 2nd ACM Conference on Embedded Networked Sensor Systems*, Nov. 2004.
17. D. Doolin, and N. Sitar, “Wireless Sensors for Wild Fire Monitoring,” *Proc. SPIE Symposium on Smart Structures & Materials (NDE 2005)*, San Diego, California, March 6-10, 2005.
18. D. Estrin, D. Culler, K. Pister, and G. Sukhatme, “Instrumenting the Physical World With Pervasive Networks,” *Pervasive Computing*, vol. 1, no. 1, pp. 59-69, 2002.

19. K. Martinez, J. Hart, and R. Ong, “Environmental Sensor Networks,” *IEEE Computer*, vol. 37, no. 8, pp. 50-56, 2004.
20. D. Doolin, S. Glaser, and N. Sitar, “Software Architecture for GPS-enabled Wildfire Sensorboard,” *TinyOS Technology Exchange*, University of California, Berkeley CA., Feb. 26, 2004.
21. Beckwith, R., Teibel, D., Bowen, P., "Unwired wine: sensor networks in vineyards", Proceedings of IEEE Sensors, Vol 2, pp. 561-564, Oct. 24-27, 2004, ISBN: 0-7803-8692-2, DOI: 10.1109/ICSENS.2004.1426227
22. K. Delin and S. Jackson, “The Sensor Web: a New Instrument Concept,” *Proc. SPIE Symposium on Integrated Optics*, San Jose, CA, January 2001.
23. Crossbow Technology, Inc. Home Page. Retrieved March 28, 2008 from http://www.xbow.com
24. Crossbow Technology’s MICA2 product page. Retrieved March 28, 2008 from http://www.xbow.com/Products/productdetails.aspx?sid=174
25. Nachman, L., Kling, R., Adler, R., Huang, J., and Hummel, V., "The Intel Mote platform: a Bluetooth-based sensor network for industrial monitoring," In *Proceedings of the 4th international Symposium on information Processing in Sensor Networks* (Los Angeles, California, April 24 - 27, 2005). Information Processing In Sensor Networks. IEEE Press, Piscataway, NJ, 61.
26. Adler, R., Flanigan, M., Huang, J., Kling, R., Kushalnagar, N., Nachman, L., Wan, C., and Yarvis, M. 2005. Intel Mote 2: an advanced platform for demanding sensor network applications. In Proceedings of the 3rd international Conference on Embedded Networked Sensor Systems (San Diego, California, USA, November 02 - 04, 2005). SenSys '05. ACM, New York, NY, 298-298. DOI= http://doi.acm.org/10.1145/1098918.1098963
27. Crossbow Technology’s Customer References. Retrieved March 28, 2008 from http://www.xbow.com/Industry_solutions/CustomerReference.aspx
28. “Catch a Sniper, Mote!” CrossBow Solutions, First Quarter, 2005, Volume 5. Retrieved March 12, 2008 from http://www.xbow.com/General_info/Info_pdf_files/XbowNewsletter_Q1-05.pdf
29. “Mote in Controlled Environment Monitoring,” CrossBow Solutions, Fourth Quarter, 2005, Volume 8. Retrieved March 12, 2008 from http://www.xbow.com/General_info/Info_pdf_files/Xbow_Newsletter_Q4_05.pdf
30. Kevan, T., “Irrigation Smarts Tee Up Savings,” Sensors Magazine, November 1, 2006. Retrieved 3/12/2008 from http://www.sensorsmag.com/sensors/article/articleDetail.jsp?id=383811&pageID=1
31. Crossbow Case Study, “Life Fitness Exercises Efficiencies with Wireless Mesh Networking,” Retrieved March 13, 2008 from http://www.xbow.com/General_info/Info_pdf_files/LF_Case%20Study.pdf
32. Jim Montague, “Wireless aids molding machine ERP,” Industrial Networking, Winter 2007. Retrieved March 13, 2008 from http://www.controldesign.com/articles/2006/176.html
33. I. Wokoma, L. Sacks, and I. Marshall, "Biologically Inspired Models for Sensor Network Design", *London Communications Symposium*, 2002.
34. P. Hammerstein, O. Leimar, “Theoretical Biology: Ants on a Turring Trail”, *Nature*, vol. 418 (6894), pp. 141-142, July 2002.

35. G. Theraulaz, E. Bonabeau, S. Nicolis, R. Sole, V. Fourcassié, S. Blanco, R. Fournier, J. Joly, P. Fernández, A. Grimal, P. Dalle, and J. Deneubourg, "Spatial Patterns in Ant Colonies", *Proc. National Academy of Sciences of the United States of America*, vol. 99, no. 15, July 2002.
36. I. Akyildiz and I. Kasimoglu, "Wireless Sensor and Actor Networks: Research Challenges," *Ad Hoc Networks Journal (Elsevier)*, vol. 2, no. 4, pp. 351-367, October 2004.
37. SANET 2007, First ACM Workshop on Sensor Actor Networks in conjunction with ACM MobiCom 2007, Montreal, Canada, Sept. 10, 2007. Retrieved March 26, 2008 from http://www2.lifl.fr/POPS/SANET2007
38. K. Jones, K. Lodding, S. Olariu, A. Wadaa, L. Wilson, and M. Eltoweissy, "Biomimetic Models for Wireless Sensor Networks," In *Handbook of Bioinspired Algorithms And Applications*, S. Olariu, and A. Zomaya, Eds., Chapman & Hall/CRC Computer & Information Science, Boca Raton, Florida, USA, 2004, ISBN 1-5848-8475-4.
39. I. Akyildiz, "Grand Challenges For Wireless Sensor Networks," Keynote Address at *8th International Symposium on Modeling, Analysis and Simulation of Wireless and Mobile Systems*, Montreal, Canada, October 2005.
40. R. Lachman, J. Lachman, and E. Butterfield, *Cognitive Psychology and Information Processing*, Lawrence Erlbaum Assoc, Hillsdale, NJ, 1979. ASIN: 047026649X.
41. R. Pfeifer, and C. Scheier, *Understanding Intelligence*, The MIT Press, Cambridge, MA, 2000. ISBN 0-262-16181-8.
42. E. Bonabeau, M. Dorigo, and G. Theraulaz, *Swarm Intelligence: From Natural to Artificial Systems*, Oxford University Press, 1999.
43. J. Turner, *The Extended Organism: The Physiology of Animal-Built Structures*, Harvard University Press, 2000.
44. E. Morse, "Fireflies Flashing in Unison", *Science*, vol. 44, issue 1133, 1916, pp. 387-388.
45. G. Hudson, "Concerted Flashing of Fireflies," *Science*, 48, pp. 573-575, 1918.
46. J. Buck, "Synchronous Rhythmic Flashing of Fireflies. II," *The Quarterly Review of Biology*, vol. 63, no. 3, pp. 265-289, Sep. 1988.
47. S. Camazine, J. Deneubourg, N. Franks, J. Sneyd, G. Theraulaz, and E. Bonabeau, "Self-Organization in Biological Systems", *Princeton Studies In Complexity*, Princeton University Press, 2001.
48. J. Buck and J. Case, "Control of Flashing in Fireflies. I. The Lantern as a Neuroeffector Organ", *Biological Bulletin*, vol. 121, pp 234-256, 1961.
49. M. Mitchell, J. Crutchfield, and R. Das, "Computer Science Application: Evolving Cellular Automata to Perform Computations," in *Handbook of Evolutionary Computation*, T. Bäck, D. Fogel, and Z. Michaelewics, Eds., Oxford University Press, 1997.
50. J. Epstein, "Learning to be Thoughtless: Social Norms and Individual Computation," *Center on Social and Economic Dynamics Working Paper No. 6*, revised January 2000.
51. R. Cunha, A. Silva, A. Loureiro, L. Ruiz, "Simulating Large Wireless Sensor Networks Using Cellular Automata", *Proc. 38th Annual Simulation Symposium (ANSS'05)*, pp. 323-330, 2005.

52. J. Conway, “Game of Life”, in Martin Gardner's Mathematical Games, *Scientific American*, October 1970.
53. J. Epstein and R. Axtell, *Growing Artificial Societies*, The Brookings Institution, Washington, D.C., 1996.
54. Helmholtz, H. and Ellis, A., "On the Sensations of Tone as a Physiological Basis for the Theory of Music," Dover Publications, New York, 1954.
55. Mani, R., “Acoustic Duct with Peripherally Segmented Acoustic Treatment. “U.S. Patent 3,937,590, February, 1976.
56. Howe, M., “The Attenuaiton of Sound in a Randomly Lined Duct,” Journal of Sound and Vibration, Vol. 87, No. 1, pp. 83-103, 1982.
57. Watson, W., Jones, M., Parrot, T., and Sobieski, J., “A Method for Optimizing Non-Axisymmetric Liners for Multi-Modal Sound Sources,” AIAA Paper 202-2516, Proceedings of the 8th AIAA/CEAS Aeroacoustics Conference and Exhibit, June 17-19, 2002.
58. Watson, W., Robinson, J., Jones, M., and Parrott, T., "Design and Attenuation Properties of Periodic Checkerboard Liners," AIAA-2003-3309, 9th AIAA/CEAS Aeroacoustics Conference and Exhibit, Hilton Head, South Carolina, May 12-14, 2003.
59. Little, E., Kashani, R., Kohler, J., and Morrison, F., "Tuning of an Electro-rheological Fluid-based Helmholtz Resonator as Applied to Hydraulic Engine Mounts," Transporation Systems, Vol. 54, pp. 43-51. 1994.
60. De Bedout, J., Franchek, M., Bernhard, R., and Mongeau, L., "Adaptive-passive Noise Control With Self-tuning Helmholtz Resonators," Journal of Sound and Vibration, Vol. 202(1), pp. 109-123, 1997.
61. Selamet, A. and Lee, I., "Helmholtz Resonator With Extended Neck," JASA, Vol. 113(4), pp. 1975-1985, 2003.
62. Birdsong, C., and Radcliffe, C., "A Smart Helmholtz Resonator for Tuning Acoustic Ducts," ASME International Mechanical Engineering Congress and Exhibition, Dallas, TX, 1997.
63. Utsunii, M., "Reduction of Noise Transmission in a Duct by Termination Impedance Control of a Sidebranch Resonator," Journal of Vibration and Acoustics, Vol. 123, Issue 3, pp. 289-296, July 2001.
64. Kostek, F. and Franchek, M., "Hybrid Noise Control in Duct," Journal of Sound and Vibration, Vol. 237(1), pp. 81-100, 2000.
65. Ahuja, K. and Gaeta Jr., R., “Active Control of a Liner Impedance by Varying Perforate Orifice Geometry,” National Aeronautics and Space Administration, CR-2000-210633, December, 2000.
66. Gaeta Jr., R., and Ahuja, K., "A Tunable 2DOF Liner," AIAA-1999-1852, AIAA/CEAS 5th Aeroacoustics Conference and Exhibit, Collection of Technical Papers. Vol. 1 (A99-27801 06-71), Bellevue, WA, May 10-12, 1999.
67. Taylor, R., Liu, F., Horowitz, S., Ngo, K., Nishida, T., Cattafesta, L., and Sheplak, M., "Technology Development for Electromechanical Acoustic Liners," paper a04-093, Active 04, Williamsburg, VA, September 2004.
68. Liu, F., Horowitz, S., Nishida, T., Cattafesta, L., and Sheplak1, M., "A Tunable Electromechanical Helmholtz Resonator," 9th AIAA/CAES Aeroacoustics Conference and Exhibit, AIAA 2003-3145, May 12-14, 2003.

69. L. Bajaj, M. Takai, R. Ahuja, K. Tang, R. Bagrodia, and M. Gerla, "GloMoSim: A Scalable Network Simulation Environment," Technical Report 990027, UCLA Computer Science Department.
70. Chen, G., J. Branch, M. J. Pflug, L. Zhu and B. Szymanski (2004). SENSE: A Sensor Network Simulator. Advances in Pervasive Computing and Networking. B. Szymanksi and B. Yener, Springer: 249-267
71. Ahmed Sobeih, Wei-Peng Chen, Jennifer C. Hou, Lu-Chuan Kung, Ning Li, Hyuk Lim, Hung-Ying Tyan, and Honghai Zhang, J-Sim: a simulation and emulation environment for wireless sensor networks, IEEE Wireless Communications Magazine, vol. 13, num. 4, pp. 104-119, August 2006.
72. Network Simulator 2 (NS2) Home Page, Retrieved March 4, 2008 from http://www.isi.edu/nsnam/ns
73. S. Olariu, A. Wadaa, L. Wilson, and M. Eltoweissy, "Wireless Sensor Networks: Leveraging the Virtual Infrastructure," *IEEE Network*, vol. 18, no. 4, pp. 51-56, 2004.
74. A. Wadaa, S. Olariu, L. Wilson, M. Eltoweissy, and K. Jones, "Training a Wireless Sensor Network," *Mobile Networks and Applications*, vol. 10, pp. 151-167, 2005.
75. Rhee, I., Warrier, A., and Xu, L., "Distributed Randomized TDMA Scheduling for Wireless Mesh Networks", *Proc. Mobihoc*, Florence, Italy, 2006.
76. S. Ganeriwal, R. Kumar, M. Srivastava, "Timing-sync protocol for sensor networks," *Proc. ACM Conference on Embedded Networked Sensor Systems (SENSYS 2003)*, 2003.
77. Crossbow Technology, Inc., *MPR/MIB User's Manual*, Retrieved May 5, 2005, from http://www.xbow.com/Support/Support_pdf_files/MPR-MIB_Series_Users_Manual.pdf
78. P. Levis, S. Madden, J. Polastre, R. Szewczyk, K. Whitehouse, A. Woo, D. Gay, J. Hill, M. Welsh, E. Brewer, and D. Culler, "TinyOS: An Operating System for Sensor Networks," in *Ambient Intelligence*, Jan Rabaey Ed, 2003, ISBN: 978-3-540-23867-6, DOI 10.1007/3-540-27139-2_7.
79. Nark, D. Farassat, F., Pope, D., Vatsa, V., "The Development of the Ducted Fan Noise Propagation and Radiation Code CDUCT-LaRC," AIAA Paper 2003-3242, 2003.
80. NASA NAS Plot 3D. Retrieved March 26 from http://www.nas.nasa.gov/Research/Software/swdescription.html#Plot3D

Appendix A: Publications Related to Sensor Networks

1. Jones, K., Lodding, K., Olariu, S., Wilson, L., and Xin, C., "Biology Inspired Approach for Communal Behavior in Sensor Networks", submitted to Special Issue of "Ad Hoc and Sensor Wireless Networks", Old City Publishing, Philadelphia, PA, 2007.
2. Jones, K., Lodding, K., Olariu, S., Wilson, L., and Xin, C., "Biology-inspired Architecture for Situation Management", submittted to 2nd Workshop on Situation Managment (SIMA 2006) at MILCOM 2006.
3. Jones, K., Lodding, K., Olariu, S., Wilson, L., and Xin, C., "Communal Cooperation in Sensor Networks for Situation Management", Proceedings of 9th International Conference on Information Fusion (Fusion 2006), Florence, Italy, July 10-13, 2006.
4. Jones, K., Lodding, K., Olariu, S., Wilson, L., and Xin, C., "Biology Inspired Approach for Communal Behavior in Sensor Networks", presented at Minitrack on Wireless Sensor Networks and Applications, Proceedings of 39th Hawaii International Conference on System Sciences (HICSS-39), Kauai, Hawaii, January 4-7, 2006.
5. Jones, K., Lodding, K., Olariu, S., Wilson, L., and Xin, C., "An Ecosystem Model for Massively-Deployed Sensor Networks", Proceedings of 2nd International Conference on Intelligent Sensors, Sensor Networks and Information Processing, Melbourne (ISSNIP 2005), Australia, December 5-8, 2005, ISBN: 0-7803-9399-6
6. Jones, K., Lodding, K., Olariu, S., Wilson, L., and Xin, C., "Energy Usage in Biomimetic Models for Massively-Deployed Sensor Networks", presented at 1st International Workshop on Mobile Ad-hoc and Ubiquitous Sensor Networks (MAUSN 2005), Proceedings of 3rd International Symposium on Parallel and Distributed Processing and Applications (ISPA'2005), pp. 434 - 443, Nanjing, China, November 2-5, 2005, ISBN: 3-540-29770-7
7. Jones, K., Lodding, K., Olariu, S., Wilson, L., and Xin, C., "Sensor Networks for Situation Management-A Biomimetic Model", presented at Workshop on Situation Management (SIMA 2005), Proceedings of IEEE Military Communications Conference (MILCOM 2005), Atlantic City, NJ, , October 17-20, 2005.
8. Jones, K., Lodding, K., Olariu, S., Wilson, L., and Xin, C., "Using Cellular Automata for Simulation of Biomimetic Models of Massively-Deployed Sensor Networks", Proceedings of 8th International Symposium on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM 2005), Montreal, Canada, October 10-13, 2005.
9. Jones, K., Lodding, K., Olariu, S., Wilson, L., and Xin, C., "Biology Inspired Distributed Consensus in Massively-Deployed Sensor Networks", Proceedings of 4th International Conference on AD-HOC Networks & Wireless (ADHOC-NOW 2005), Cancun, Mexico, October 6-8 , 2005, in Syroutiuk, V. and Chavez E. (Eds.), Lecture Notes in Computer Science, Springer Berlin / Heidelberg ISBN: 3-540-29132-6
10. Jones, K., Lodding, K., Olariu, S., Wilson, L., and Xin, C., "Biomimetic Models for an Ecological Approach to Massively-Deployed Sensor Networks", Proceedings of ISCA 18th International Conference on Parallel and Distributed Computing Systems (PDCS 2005), Las Vegas, Nevada, September 12-14, 2005.
11. Wadaa, A., Jones, K., Olariu, S., Eltoweissy, M., & Wilson, L., "A Scalable Solution

for Securing Wireless Sensor Networks“, in Wu, J (Ed.), Handbook on Theoretical and Algorithmic Aspects of Sensor, Ad Hoc Wireless, and Peer-to-Peer Networks, CRCnetBASE, Taylor and Francis Group, LLC, Florida Atlantic University, Boca Raton, USA, August 8, 2005, ISBN: 0849328322

12. Jones, K., Lodding, K., Olariu, S., Wadaa, A., Wilson, L., Eltoweissy, M., "Biomimetic Models for Wireless Sensor Networks" In Olariu, S. & Zomaya, A. (Ed.), Handbook of Bioinspired Algorithms And Applications. Chapman & Hall/CRC Computer & Information Science, Boca Raton, Florida, USA, 2004, ISBN 1-5848-8475-4
13. Wadaa, A., Olariu, S., Wilson, L., Eltoweissy, M., & Jones, K., "On providing anonymity in wireless sensor networks", Proceedings of 10th International Conference on Parallel and Distributed Systems (ICPADS 2004), Newport Beach, California, July 7-9, 2004, pp 411 - 418.
14. Wadaa, A., Olariu, S., Wilson, L., Eltoweissy, M., Jones, K., Sundaram, P., "Training a Sensor Network", in Special Issue of MObile NETwork (MONET) on "Algorithmic Solutions for Wireless, Mobile, Ad Hoc and Sensor Networks". C-edited by A. Bar-Noy, A. Bertossi, M. Pinotti, and C. Raghavendra, 2004.
15. Jones, K., Wadaa, A., Olariu, S., Wilson, L., & Eltoweissy, M., "Towards a New Paradigm for Securing Wireless Sensor Networks", Proceedings New Security Paradigms Workshop 2003, pp 115-122. Ascona, Switzerland. August 18-21, 2003.
16. Wadaa, A., Olariu, S., Wilson, L., K. Jones, & Xu, Q., "On Training a Sensor Network", 3rd International Workshop on Wireless, Mobile and Ad Hoc Networks (WMAN), 17th International Parallel and Distributed Processing Symposium (IPDPS 2003), Nice, France. April 26, 2003, pp 22-26. CD-ROM/Abstracts Proceedings available from IEEE Computer Society, ISBN 0-7695-1926-1.

Appendix B: Previous Publications

1. Jones, K. (1998). Improving Productivity and Value of Experimental Processes Through Application of Information Technology. Proceedings of the 20th AIAA Advanced Measurements and Ground Testing Technology Conference, A98-2718, June 15-18, Albuquerque, NM.
2. Jones, K., Olariu, S., Rowell, L., Schwing, J., & Wilhite, A. (1995). A CAD Database Integrity Monitor for EASIE.
3. Jones, K., (1994). TOPS On-Line – Automating the Construction and Maintenance of HTML Pages. Advance Proceedings of the Second International WWW Conference '94, Mosaic and The Web, Volume I, October 17-20, Chicago, IL.
4. Jones, K., Randall, D., Cronin, C., (1992). Information Management for a Large Multidisciplinary Project. AIAA/USAF/NASA/OAI Symposium on Multidisciplinary Analysis and Optimizations, AIAA 92-4720, September 21-23, Cleveland, OH
5. Jones, K., Randall, D., Rowell, L., Gates, R., Nichols, C., & Williams, S. (1992). Environment for Application Software Integration and Execution. Proceedings of the ACME International Computers in Engineering Conference and Exposition, American Society of Mechanical Engineers, August 2-6, San Francisco, CA.
6. Bowen, J., Hammond, D., & Jones, K., (1992). Graphics Kernel System/DI-3000 Interface Library (GLIB) User's Guide. NASA Technical Memorandum, May
7. Jones, K., Rowell, L., Schwing, J., & Wilhite, A. (1991). Environment for Application Software Integration and Execution. Proc. 7th ASCE Conf on Computing and Database.
8. Jones, K., Wilhite, A., Rowell, L., & Schwing, J. (1991). Environment for Application Software Integration and Execution. Proceedings of the Seventh Conference on Computing in Civil Engineering and Symposium on Data Bases, American Society of Civil Engineers, May 6-8, Washington D.C.
9. Jones, K., Olariu, S., Schwing, J., & Wilhite, A. (1990). A Database Consistency Checker for EASIE. Proceedings of the Fifth International Conference on CAD/CAM and Robotics, Springer-Verlag, Volume 1, pp 55-60.
10. Randall, D., Jones, K., & Rowell, L. (1988). The Environment for Application Software Integration and Execution (EASIE) Version 1.0, Volume IV, System Installation and Maintenance Guide. NASA Technical Memorandum 100576.
11. Jones, K., Randall, D., Stallcup, S., & Rowell, L. (1988). The Environment for Application Software Integration and Execution (EASIE) Version 1.0, Volume II, Program Integration Guide. NASA Technical Memorandum 100574.
12. Rowell, L., Schwing, J., & Jones, K., (1988). Software Tools for the Integration and Execution of Multidisciplinary Analysis Programs. AIAA/AHS/ASEE Aircraft Design, Systems, and Operations Meeting, AIAA-88-4448, September 7-9, Atlanta, GA.
13. Randall, D., Jones, K., von Ofenheim, W., & Gates, R. (1984). SMP – A Solid Modeling Program. NASA Contractor Report 172473.
14. Gates, R., Matthews, C., von Ofenheim, W., Randall, D., & Jones, K. (1984). Computer Generated Animation and Movie Production at LaRC: A Case Study. NASA Contractor Report 172375.

VITA

Kennie H. Jones

NASA Langley Research Center

Hampton, VA 23681

Contact

MS458, B1299, R236
NASA LaRC
Hampton, VA 23681

Work: 757-864-6720
Home: 757-868-7567

Education

Ph.D. in Computer Science, Old Dominion University, 2008

M.S. in Computer Science, Old Dominion University, 1981

B.S. in Biology, Christopher Newport College, 1977

Employment

1990 to Present	Computer Engineer, NASA, Hampton, VA.
1981 to 1990	Computer Scientist, Computer Sciences Corp., Hampton, VA
1977 to 1981	Science Teacher, York County Public Schools, York County, VA