Large Group Musical Interaction using Disposable Wireless Motion Sensors

by

Mark Christopher Feldmeier

S.B., Massachusetts Institute of Technology (1996)

Submitted to the Program in Media Arts and Sciences,
School of Architecture and Planning,
in partial fulfillment of the requirements for the degree of

Master of Science in Media Arts and Sciences

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

February 2003

Author⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯
Program in Media Arts and Sciences
October 8, 2002

Certified by⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯
Joseph A. Paradiso
Associate Professor
Program in Media Arts and Sciences
Thesis Supervisor

Accepted by⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯
Andrew Lippman
Chair, Department Committee on Graduate Students
Program in Media Arts and Sciences
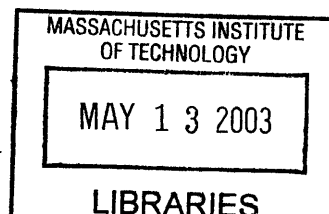
Large Group Musical Interaction using Disposable Wireless Motion Sensors

by

Mark Christopher Feldmeier

Submitted to the Program in Media Arts and Sciences,
School of Architecture and Planning,
on October 8, 2002, in partial fulfillment of the
requirements for the degree of
Master of Science in Media Arts and Sciences

Abstract

One of the difficulties in interactive music and entertainment is creating environments that reflect and react to the collective activity of groups with tens, hundreds, or even thousands of participants. Generating content on this scale involves many challenges. For example, how is the individual granted low latency control and a sense of causality, while still allowing for information retrieval from all participants so that the environment responds to the behavior of the entire group? These issues are particularly pertinent in the area of interactive dance.

To address these issues, a low-cost, wireless motion sensor has been developed. The sensor is inexpensive enough to be considered disposable, allowing it to be given away to participants at large dance events, enabling the dancers to participate concurrently in a real-time, interactive musical performance. The sensors are either worn or held by participants and transmit a short RF pulse when accelerated past a certain threshold. The RF pulses are received by a base station and analyzed to detect rhythmic features and estimate the general activity level of the group. These data are then used to generate music that can either lead or follow the participants' actions, thereby tightening the feedback loop between music and dancer.

Multiple tests of the system have been conducted, with groups ranging from fifteen to 200 participants. Results of these tests show the viability of the sensors as a large group interaction tool. Participants found the interface intuitive to use, effectively controlling such aspects of the music as style, tempo, voicing, and filter parameters. These tests also demonstrate the system's ability to detect both the activity level and dominant tempo of the participants' motions, and give considerable insight into methods of mapping these data to musical parameters that give participants direct feedback as to their current state. Furthermore, it is shown that participants, if given this direct feedback, will synchronize their actions and increase in activity level, creating a mutually coherent and pleasing outcome.

Thesis Supervisor: Joseph A. Paradiso
Title: Associate Professor, Program in Media Arts and Sciences

Large Group Musical Interaction using Disposable Wireless Motion Sensors

by

Mark Christopher Feldmeier

The following people served as readers for this thesis:

Thesis Reader _____

Rosalind W. Picard
Associate Professor
Program in Media Arts and Sciences

Thesis Reader _____

Loren Carpenter
Chief Scientist
Pixar Animation Studios

Acknowledgments

To **Abraham Farag**, **Tinsley Galyean**, and **Harry West**, who have inspired and challenged me throughout the years, and without whose recommendations i would not have arrived at the Media Laboratory.

To **Joe Paradiso**, for creating an incredible research environment, and whose belief and encouragement brought this project into reality.

To my research colleagues **Michael Broxton**, **Mateusz Malinowski**, **Mike Pihulic**, and **Josh Randall**, who contributed their time and intellect to ensure that this project was brought to completion.

To **Ari Benbasat**, **Josh Lifton**, and all members of the **Responsive Environments Group**, who supplied invaluable knowledge and a work environment unlike any other.

To **Loren Carpenter** and **Rosalind Picard**, my gracious readers, for their help and willingness to operate under deadline.

To the residents of **EAsT camPUS**, and **Senior Haus**, who provided a fertile ground for ideation and data collection. This work was completed because of you, and in spite of you, and for both i am grateful.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

There are very few systems that effectively enable a large number of participants to collaboratively control a real time, centralized interaction. This is particularly true in the area of interactive dance. Dance interfaces [1, 2] exist that allow a single or small group of dancers to control music with their actions, but these do not scale to allow for hundreds of participants to interact concurrently. The problems of cost, data communication bandwidth, and system responsiveness become increasingly more difficult as the number of participants increases. A system that could effectively give control to a large number of dancers offers the possibility of environments with extremely responsive music and lighting, engaging users to a heightened sense of expressiveness.

1.2 System Overview

To address these issues of large group musical mapping, we have developed a system [3, 4] that can effectively gather data over an essentially unlimited audience size. The system consists of wireless sensors that are given to audience members to collect rhythm and activity

information from the crowd. This information is then available to be mapped to sonic events or lighting control changes. The system is particularly well suited to very rhythmic dance music, such as electronic dance genres [5] house, techno, trance, and hardcore. A block diagram of the system can be seen in Figure 1.2.1. The sensors are small and lightweight, and can therefore be either worn or held by a participant. To detect the participant's motion, they have radio-frequency (RF) transmitters that send a short pulse of RF energy whenever they encounter acceleration greater than a predetermined level. Finally, they are inexpensive enough to be viable as disposable, give-away items for large crowds.
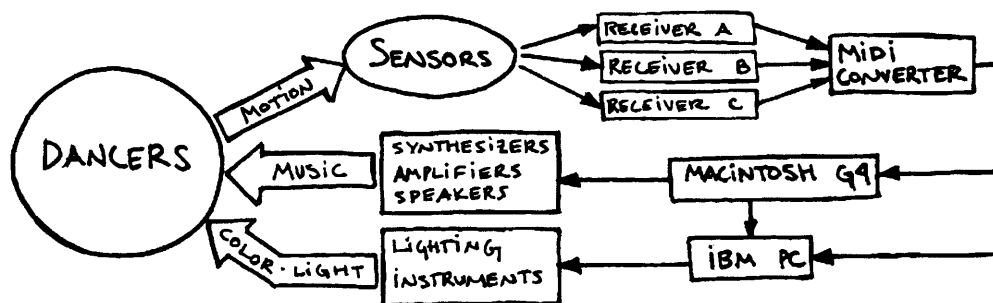
Figure 1.2.1: System Block Diagram

The sensors' RF pulses are collected by receiver base stations that have limited sensitivity, enabling the development of zones of interaction around each base station. In this manner, multiple base stations can be used in a venue to create distinct areas where the controller takes on new functions. This zoning information can also be used to direct the music and lighting to respond to the participants' actions in that area, localizing the response to a smaller group of proximate dancers. The pulses received at the base stations are then sent to the MIDI converter, which counts the amount of pulses received in each zone, and transmits this information, at a regular rate, as MIDI [6] serial communication note-on messages.

These MIDI signals are then received by a Macintosh G4 computer, where they are analyzed to detect activity levels and rhythmic features of the audience. These parameters are then available to be mapped to musical content or lighting control information. For our applications, all data analysis and musical mapping is done in the MAX [7] program-

16

ming environment, and sound generation is done off board with dedicated hardware music synthesizers. Lighting content is generated with an IBM-compatible personal computer, and control information is sent to the lighting instruments via DMX [8] serial communication. The sound and lighting changes are then played out to the audience, which in turn, the audience responds to, allowing the experience to build upon itself, giving the users an increased connection to the music.

Chapter 2

Background

2.1 Related Work

Current research in the area of large group interaction is dominated by interactive gaming. Networked computer games [9] can allow up to 100 physically distributed participants to compete with each other, with data delays on the order of 100 ms. For physically situated groups of less than a hundred participants, there exist fixed systems such as voting interfaces for game show audiences (usually pushbuttons located in the armrests of chairs). But, for participants in numbers over a hundred, hardwired solutions become costly, and do not allow the participants to be mobile. Some systems enable many participants to become engaged via wireless PDAs [10], but these are also quite costly and generally not real-time. Systems that look for cues from infrared cameras [11], microphones [12], or capacitive sensors [13] can gather bulk information over a large, mobile audience, but they do not lend themselves to direct control by an audience member. Generally, the participant has no sense of which action will dictate the desired response. For this to happen, there must be an effective way of measuring a particular action amongst each participant.

To date, the most effective methods of measuring and summing audience member's actions have been done via machine vision. The following gives the most pertinent examples of

systems that are capable of taking input from extremely large groups of people, and using these inputs as a controller for a real-time, central interaction.

2.1.1 Cinematrix Interactive Entertainment System

First demonstrated in 1991 at ACM SIGGRAPH, Loren Carpenter's Cinematrix Interactive Entertainment System [14] was the first system to enable an audience of 4,000 people to compete concurrently in an electronic game. The system functions by giving a paddle to each participant. One side of the paddle is red, while the other is green. By rotating the paddle, audience members can signal to a camera which direction they want an agent to move, or cast a vote between two outcomes. In this manner, the sum of red and green pixels in a given area determine the direction or outcome. The system has been used in competitive games where the audience is split into two groups, each controlling the direction of their agent in order to defeat the other, and in collaborative situations where the entire audience votes to control the agent to a positive outcome.

This system has been very successful, enough so to support a company based upon its technology. Each audience member is given a direct and causal input, and in turn feedback is given to inform the audience member that his or her response has been received. The nature of the input is very intuitive, and needs little explanation for its use. Finally, the reflective nature of the paddles add to the selectivity of the machine vision, allowing the system to be used under a variety of lighting conditions.

2.1.2 Galvactivator

Another method of summing an audience's actions is to give each participant an active device. This is demonstrated by Rosalind Picard's and Jocelyn Scheirer's glowing Galvactivator skin-resistance detectors [15], which have been used on an audience of 1200 people. The Galvactivator is a glove worn by participants that has two electrical contacts that measure the change in conductivity of the user's skin. This change in conductivity is displayed as brightness of a light emitting diode (LED) on the glove. The aggregate brightness of

20

the audience's Galvactivators can than be viewed with a camera, allowing the audience's excitement or stress level to be measured. The glove itself consists of a battery, two transistors, a resistor, and an LED. This minimal component set make it an inexpensive item, capable of being given away to participants. Direct feedback from each participant can be obtained, and except for calibration of the LED brightness, the user does not have to put any effort into the device or the interaction.

2.1.3 Maynes-Aminzade's audience interaction work

Dan Maynes-Aminzade's audience interaction work [16] is unique in its ability to sense the audience's actions without tethering its members to any input device. The machine vision merely looks for those motions made naturally by audience members. For example, to control a driving simulation, the audience leans in the direction it wishes the car to turn. Or, to control the motion of an agent, the system uses pixel differencing to detect the aggregate crowd motion, and increases the agent's motion accordingly. In this respect, the system is very intuitive to use, and requires no added cost besides for the vision system and associated processing hardware.

2.2 Limitations of Machine Vision for Dance Applications

As shown by the previous work, machine vision is an extremely low cost method of gathering data over a large group of people. It has the advantage of not tethering the audience, and being able to detect direct inputs by audience members. Despite these benefits, machine vision suffers from requiring a line-of-sight from camera to participant, and is susceptible to illumination effects and background lighting. Its ability to gather meaningful data would be limited in the common dance club setting of dark rooms with quickly changing colored lights. Also, many of the required input methods, such as voting paddles or glowing gloves, would restrict the motions of the dancer to those that placed the interface in view of the camera. For work such as Maynes-Aminzade's, which does not have a contact sensing

21

method, the user does not have a direct input to the system, and is merely a portion of the sum of activity.

In general, making these types of measurements using non-contact methods such as machine vision or machine listening is not as accurate as direct methods such as wearable or handheld sensors with RF links, which do not suffer from occlusion. One example of such a wearable device was used in the Sophisticated Soiree installation [17] at Ars Electronica 2001, where up to 64 participants were given wireless heart rate sensors that controlled a musical stream for an experiment in large-group bio-feedback. However, these systems, which measure autonomic responses such as heart rate and skin resistance (as in the Galvactivator), are generally not consciously controllable by participants. Cliff's HPDJ [18] project is an interesting hybrid, which proposes providing dancers with wireless accelerometers, heart rate, and perspiration sensors. This relatively expensive package would then be used to gauge the general activity level of a clubs crowd, and choose appropriate musical tracks via a genetic algorithm to keep the crowd dancing.

2.3   System Requirements

As shown by Carpenter's work, systems based on real-time kinetic inputs can provide for very causal control. For such a system to work, each participant must be supplied with a controller that has a consistent response, given a particular input. In this manner, the user can quickly learn what result each action will have. This controller must also be intuitive to use, allowing the participant to become engaged in the interaction, rather than in the use of the controller. Ideally, the device should not encumber the user, being unobtrusive enough that the participant quickly forgets that it exists.

For the controller to be viable on scales of hundreds to thousands of participants, it must be inexpensive, with a cost comparable to other give-away items seen at large venues (glowsticks, foam hands, baseball caps, calendars). And, to relay its data back to a central location, without the drawbacks of machine vision or communication tethers, it must also be wireless. This wireless data link needs to use a low latency, scalable communications

platform, allowing for thousands of non-synchronized inputs to be collected without significant delay or loss of data. Finally, its energy source should last many hours, if not weeks, to ensure that the device functions for the duration of an event

Chapter 3

Hardware Design

As described in Section 2.3, any controller intended to enable a large group of dancers to collaboratively control their musical experience must be consistent in response, intuitive and unobtrusive to use, inexpensive, wireless, with low latency and a scalable communications platform, and have a long battery life. To achieve these goals, the chosen design is a small, low-cost, wireless transmitter that sends a short pulse of RF energy whenever it senses acceleration greater than a predetermined level. These transmitters can be either worn or held by a participant, and are activated by motion. Both the strength and duration of the RF pulse are kept to a minimum for purposes of data collection and energy conservation. The short transmission radius creates a zone of interaction around the receiver base station, and the short RF pulse duration reduces the probability of collisions between signals. In this way, each participant's action is received, instantaneously, as a distinct event. And, the pulses in a particular area can be summed to give a sense of the rhythm and activity of the local participants.

These activity sums are performed at the MIDI converter, which receives input from multiple base stations, and sends each zone's sum to a central processing node via MIDI serial communication. This central processing node analyzes the data for relevant activity parameters, and responds with appropriate changes to the music and lighting. These control changes are communicated to the music and lighting instruments via MIDI and DMX serial

protocols, respectively. Although inexpensive MIDI interfaces are readily available for most computers, DMX interfaces are prohibitively expensive. For this reason, a custom designed, inexpensive DMX parallel port interface [19] is used.

3.1   Sensor

Cost is the main consideration for the design of the sensor. For this reason, inexpensive production parts are used. The first prototype, revision one, is seen in Figure 3.1.1. It consists of a 3-Volt lithium battery, an LM556 dual monostable-multivibrator, a vibration sensor, and a Ming TX-99 V3.0 300 MHz transmitter (an On-Off Keyed device similar to those used in electronic garage door openers). To minimize the power consumption and complexity of the sensor, the RF pulses are neither synchronized nor coded. As a result, if any two pulses are transmitted at the exact same time, they register as one event, resulting in loss of data. To minimize the probability of this occurring, the pulse width is kept as short as possible. The limit to how short the pulses can be is determined by the receiving
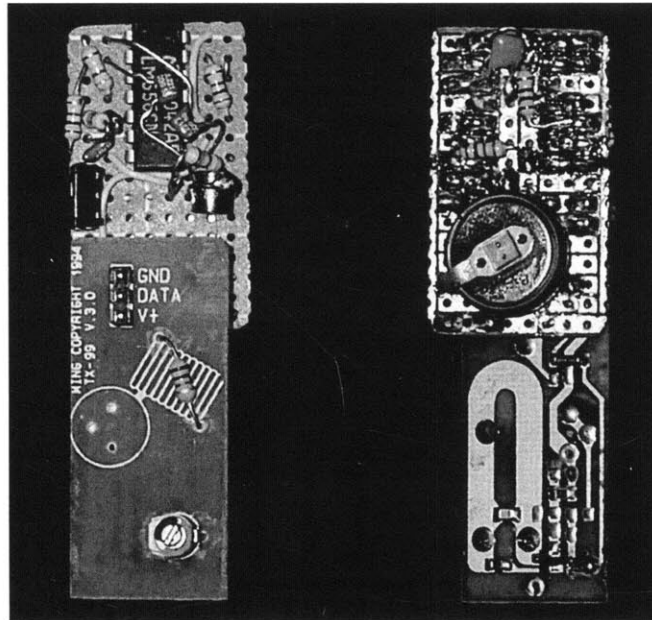
Figure 3.1.1: Front and Back Views of Revision One Sensor (actual size)

base stations, which have a finite sensitivity and response time. From testing with the base stations, this minimum width was found to be approximately $50\,\mu$s, as any pulses shorter than this would either not trigger the receiver, or not be discernible from background RF noise.

The vibration sensor used for revision one is a broken fuse ampoule. A detail of this fuse ampoule can be seen in Figure 3.1.2. The fuse ampoule's thin conductor acts as a vibration switch, triggering the LM556 whenever it contacts its surrounding metal ring. The input stage of the LM556 is arranged as a non-retriggerable one-shot timer, with a pulse width of 150 ms. This is done to reduce the probability of sending multiple RF transmissions per event, as the fuse ampoule's thin conductor continues to vibrate after excitation, contacting the surrounding metal ring multiple times. Through experimentation, the 150 ms pulse width was found to be long enough to eliminate this ring-down problem, but short enough to allow high frequency activation of the sensor. The initial edge of the 150 ms pulse is then used to trigger another non-retriggerable one-shot timer, creating a $50\,\mu$s pulse which is sent to the data pin of the Ming transmitter. In this way, a pulse of RF energy is transmitted for every crossing of the vibration sensor's threshold. A schematic for this circuit can be seen in Figure 3.1.3.
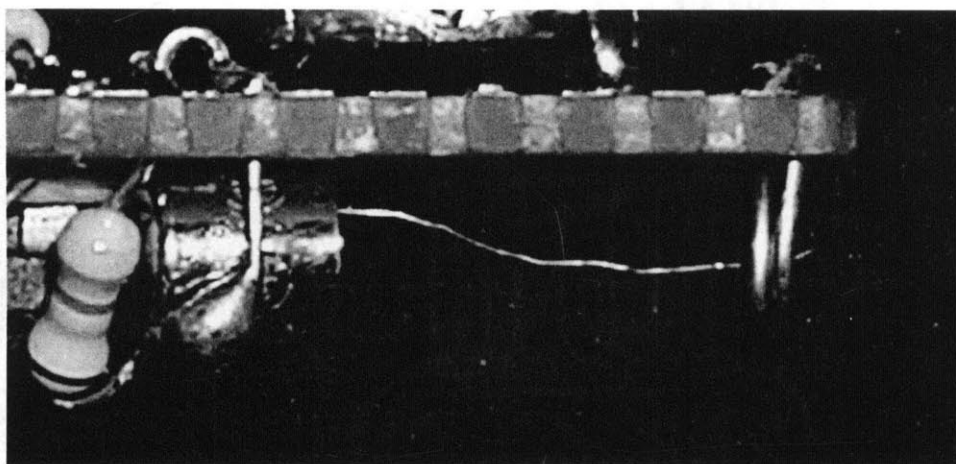
Figure 3.1.2: Detail of Revision One Vibration Switch
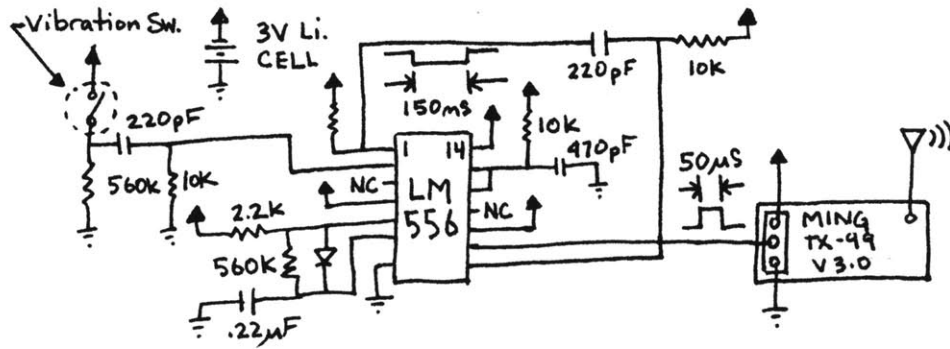
Figure 3.1.3: Schematic of Revision One Sensor

The fuse ampoule sensor has the advantage of sensing equally in all directions, but suffers from difficulty of manufacture and long vibration ring-down time. If the sensor experiences too great of an acceleration, the ring down lasts longer than the 150 ms pulse width, and a second 50 μs RF transmission occurs. The revision two vibration switch eliminates these problems, and a detail of it is shown in Figure 3.1.4. The new vibration switch consists of a bent piece of stainless steel shim stock. The length of the shim acts as a cantilever beam spring, and the end of the shim is bent upon itself multiple times to create a proof mass. Under increasing levels of acceleration, the proof mass pulls harder upon the cantilever beam spring, deflecting the end until it contacts the metal below it. This switch has the
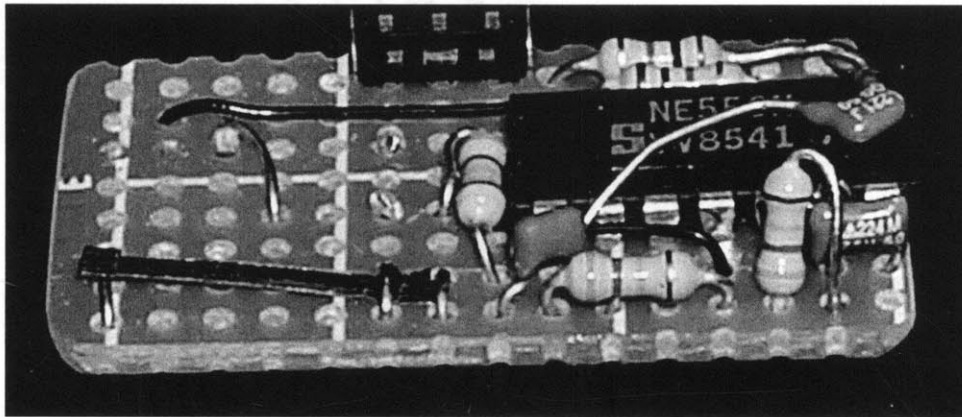
Figure 3.1.4: Detail of Revision Two Vibration Switch

advantage of allowing for variable mass, giving a variable ring-down time. By changing the amount of turns at the end, the trade off between sensitivity and ring-down time can be optimized. Despite these advantages, the shim stock vibration sensor is still time consuming to manufacture and has severe directional dependence, as the contacting surface is only on one plane.

These initial prototypes have excellent vibration sensitivity, and create stable RF transmissions, but consume too much power to be effective as a controller. The linear LM556 has a large quiescent current draw, and drains the battery within a few minutes. Also, The vibration switch ring-down is still too long, and occasionally creates multiple transmissions. Finally, The Ming transmitter is bulky, and costs 10.00$US, which dominates the cost of the sensor, making the price prohibitive for large scale distribution.

The first modification to these early prototypes was the replacement the LM556 with its CMOS equivalent, the 74HC221. The 74HC221 has a quiescent current draw of less than .01 μA, and requires fewer external parts to set the timing cycle. Next, the shim stock vibration switch was replaced with a 0.75$US piezo film sensor [20], manufactured by Measurement Specialties, Inc. (MSI) [21]. Again, a proof mass (wraps of electrical tape) is placed at the end of the sensor to set the sensitivity and ring-down time. The intrinsic damping of the piezo film is much greater than that of the shim stock, giving better ring-down response for the new sensor. Also, since the vibration sensor is no longer custom made, manufacturing time is greatly reduced. The results of these improvements, revision three, can be seen in Figure 3.1.5. This sensor still utilizes the Ming transmitter, so its cost is comparable to those before, but it no longer creates double hits, requires much less time to build, and can operate for years with intermittent (six hours a week) usage without draining its battery. Despite these improvements, this sensor is still too costly and bulky to be useful.

The next generation of prototypes, revision four, include the manufacture of application specific circuit boards, as shown in Figure 3.1.6. These boards incorporate the Ming transmitter, with slight modifications to accommodate reduced part count, less expensive parts, and a more dense layout. They maintain the same small battery, weighted piezo film sen-

29
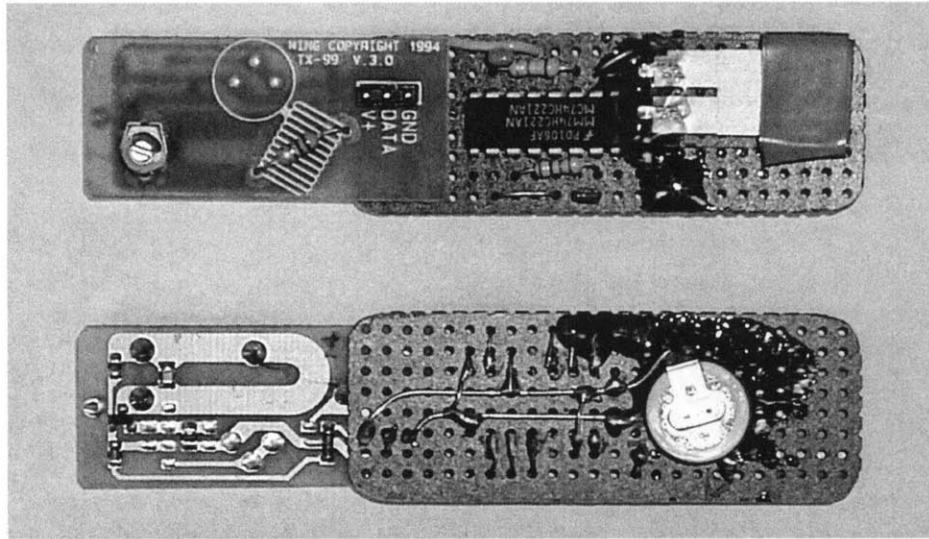
Figure 3.1.5: Front and Back Views of Revision Three Sensor (actual size)

sor, and 74HC221 (now in surface mount). These boards function extremely well, having excellent sensitivity, small size and weight, and long battery life. The transmission distance is slightly reduced with the modified transmitter, as a difficult and time consuming tuning process is required.

Twenty of these prototypes were produced and tested. They were then placed within plastic tubes to ensure that the user would not hit the vibration sensor with his hand, and to minimize heat and moisture effects on transmitter frequency drift. A sensor mounted inside of a tube can be seen in Figure 3.1.6. The tubing was specifically formed to encourage holding the sensor in its most sensitive direction (along the bending axis of the piezo film). These tubed sensors were used for the first data collection events, and worked exceptionally well.

To further reduce the manufacturing complexity, since hundreds are to be produced, a new vibration sensor [22] from MSI is used. This sensor has an integral proof mass, eliminating the need to add wraps of electrical tape. It is also smaller, enabling the entire sensor board to fit into a smaller tube, making it more comfortable to hold. The circuit board is redesigned, reducing part count and eliminating all through hole components except for the
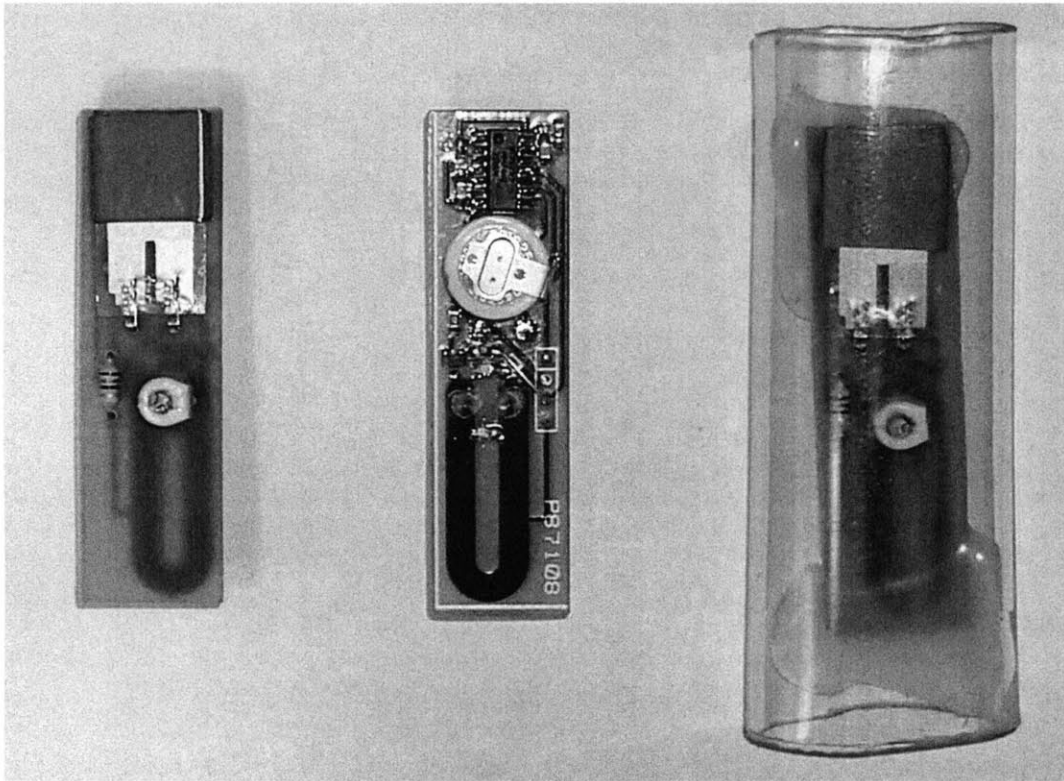
Figure 3.1.6: Revision Four Sensor (Front View, Back View, Tubed; actual size)

battery, sensor, and variable capacitor. Also, a smaller value variable capacitor is used to increase the sensitivity of the tuning. Finally, a $110\,M\Omega$ resistor chain is added to the piezo film sensor to set its sensitivity. A schematic of the sensor is shown in Figure 3.1.7. Five hundred of these revision five sensors were manufactured. An outside firm was contracted to stuff the boards, and final assembly, tuning, testing, and placement inside of tubes was done on site. A picture of the final product can be seen in Figure 3.1.8. These sensors were used for the final two data collection events. They have similar functionality to that of the previous revision, but are easier to manufacture, have a smaller form factor, and are easier to tune.

The cost breakdown for the revision five sensor, in quantities of 100, is shown in Table 3.1.1. Although the total cost is still 10$US, 5$US of this is due to labor costs, with the sensor,
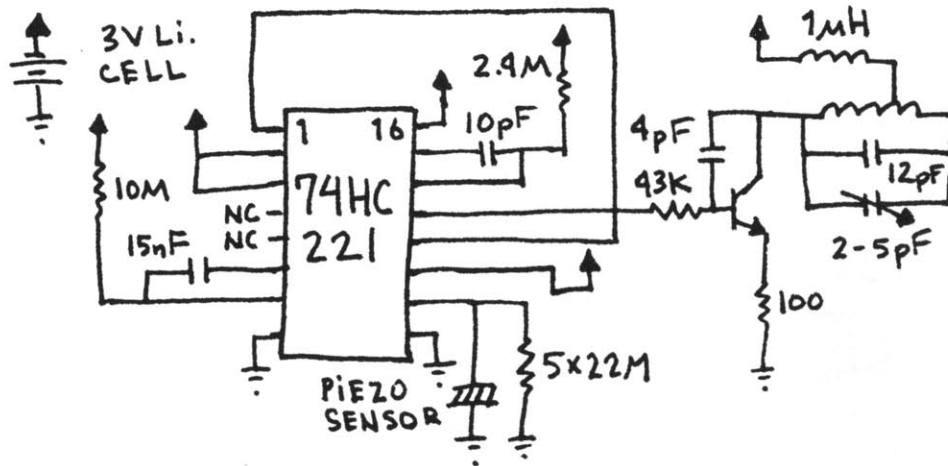
Figure 3.1.7: Schematic of Revision Five Sensor
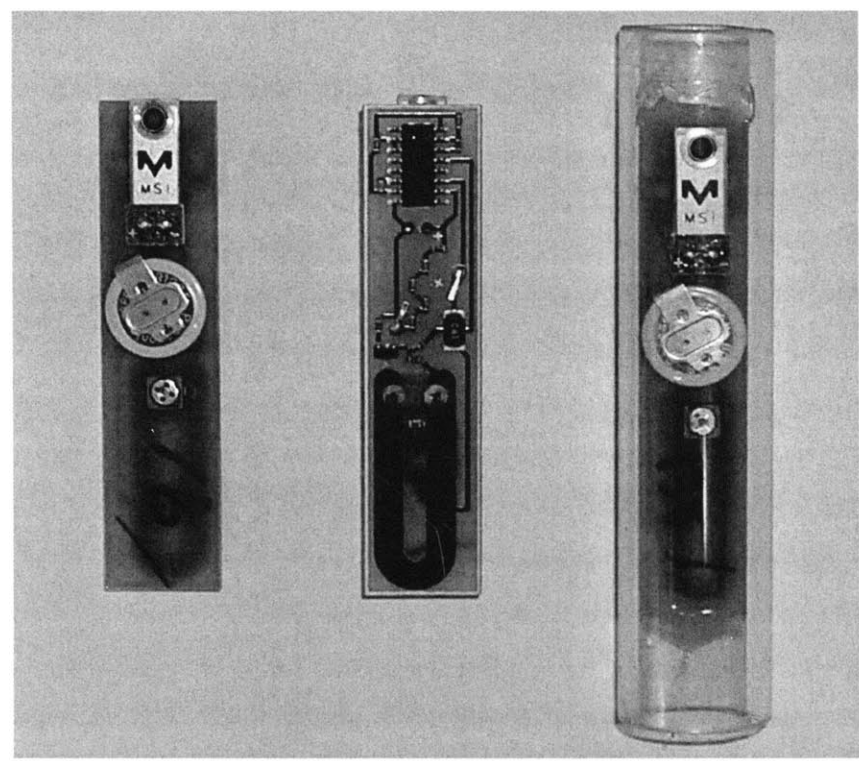
Figure 3.1.8: Revision Five Sensor (Front View, Back View, Tubed; actual size)

circuit board, and battery accounting for over 3.5$US of the remaining 5$US. With increased production, the parts cost could easily become less than 1$US, allowing them to be given away at large events with minimal cost to the event organizer.

Part Description	Cost [$US/sensor]
1 μH inductor	0.400
2–5 pF trimcap	0.180
74HC221	0.370
3 V battery	1.400
43 kΩ resistor	0.024
100 Ω resistor	0.024
2.4 MΩ resistor	0.024
10 MΩ resistor	0.024
22 MΩ resistor	0.120
piezo vibratab	1.250
NPN RF transistor	0.190
4 pF capacitor	0.045
10 pF capacitor	0.045
15 nF capacitor	0.045
12 pF capacitor	0.045
circuit board	1.000
stuffing	5.000
TOTAL	**10.19**

Table 3.1.1: Cost Breakdown for Revision Five Sensor

3.2 Receiver Base Station

The receiver base stations consist of a Ming RX-99 V3.0A receiver module, which plugs into a line driver board. This interconnectivity of parts allows for easy replacement of broken modules. The full assembly can be seen in Figure 3.2.1. Since the transmitter's pulses are not coded, the system is potentially susceptible to background RF noise. To reduce the possibility of RF interference, The Ming receivers are modified to reduce their sensitivity, eliminating spurious pulses created by background RF radiation. This modification involves removing C13, a 10 μF capacitor, and replacing it with a 2.2 nF capacitor in parallel with a 560 kΩ resistor. This has the effect of increasing the relative threshold and cut-off frequency
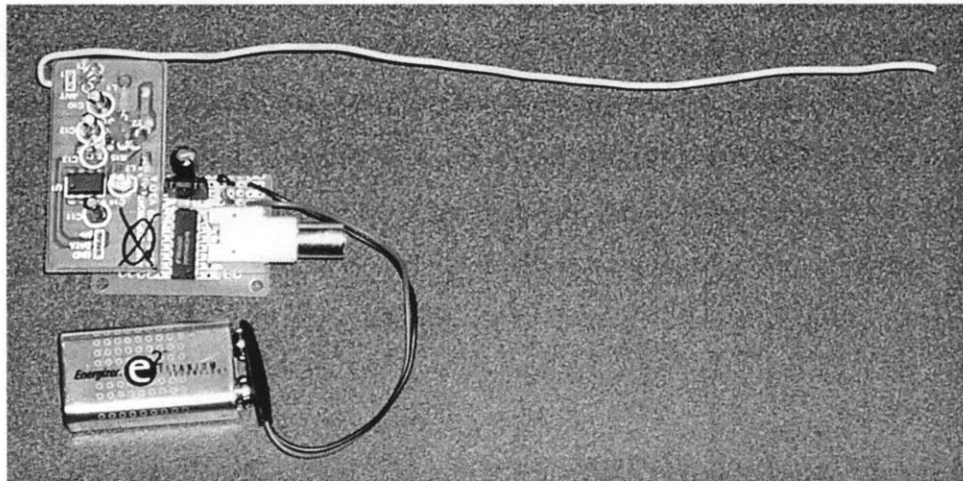
Figure 3.2.1: Receiver Base Station

of the data signal entering the comparator, allowing the detection of only sharp pulses of RF energy.

The line driver board serves two functions. First, it supplies power to the receiver module through a 9-Volt battery and a 5-Volt regulator. Second, it buffers the output stage of the receiver through a 74HC14 Schmitt trigger inverter. The hysteresis on the input stage of the inverter eliminates runt pulses that might occur due to the long transition time of the receiver output signal. The high current output stage of the inverter makes it appropriate for driving the long (25 m) coaxial cables that carry the signals to the MIDI converter, without causing significant increase in the signal's transition time. Schematics for the receiver base stations can be found in Appendix A.

3.3   MIDI Converter

The receiver base stations send their pulses to the MIDI converter, where they are summed at regular intervals for each zone. The full schematics for the MIDI converter can be found in Appendix A. As many as three receivers can connect to the MIDI converter's input stage, which performs logical operations on the received pulses. It does this for two reasons. First,

it allows for the position of each pulse to be more accurately sensed, as pulses arriving simultaneously at two different base stations indicate that the user is located somewhere between the two receivers. Secondly, it eliminates the possibility of double counting a single pulse seen at multiple base stations. Since the lag and sensitivity of the receiving base stations vary, a single sensor transmission can appear on two different receivers with a delay of up to $20\mu s$. To perform logical operations on these signals without creating runt pulses or double counting, the circuit receives a pulse and waits for at least the duration of that pulse ($50\mu s$), before deciding whether or not another pulse has arrived at another base station. It then sends out a zoned pulse to the appropriate counter. The zoning operations used are as follows. For receiving base stations A, B, and C, pulses are produced for C, for $A \cap B \cap \bar{C}$, $A \cap \bar{B} \cap \bar{C}$, and $\bar{A} \cap B \cap \bar{C}$. This scheme is used to maximize the area of coverage and number of zones, with a minimal amount of receivers and MIDI communication time. With each added zone, another $320\mu s$ of MIDI communication time is required. With four zones, the communication time is less than 2 ms, giving low latency and reducing consumption of the MIDI bandwidth.

These pulses are then sent to the counters, which accumulate the number of pulses received within a short and periodic window. By adjusting the position of a jumper, this window can be varied anywhere from 2 ms to 64 ms, in multiples of two. The pulse counts and a MIDI status byte are then loaded into a parallel to serial shift register, and clocked out at this same periodic rate. The MIDI status byte transmitted is selectable with two rotary dip switches that set both the command set and channel number. For most applications, this was set to command set note-on, channel fifteen. There is also a docking bay that allows extra circuit boards to be added to change the MIDI message. Two modules currently exist for this docking bay. One is a jumper block that sets the converter to send out its original status byte followed by four data bytes. The other is a module that sends out two status bytes, each followed by two data bytes. In this way, the differentiation of the four status bytes can be performed, even if the data arrives out of order.

The MIDI converter is packaged into a small, portable carrying case, as shown in Figure 3.3.1, making it easy to transport and ensuring that all necessary components are

present. The power supply, receiving base stations, and batteries all fit inside of the case. All power supply and data communication lines between boards are performed with non-interchangeable connectors. This ensures that the right wires connect to the right modules, but also allows for quick disconnect if new modules need to be installed. The MIDI converter has two data output lines, each with its own output drive circuitry to ensure stable communication and eliminate common mode failure.
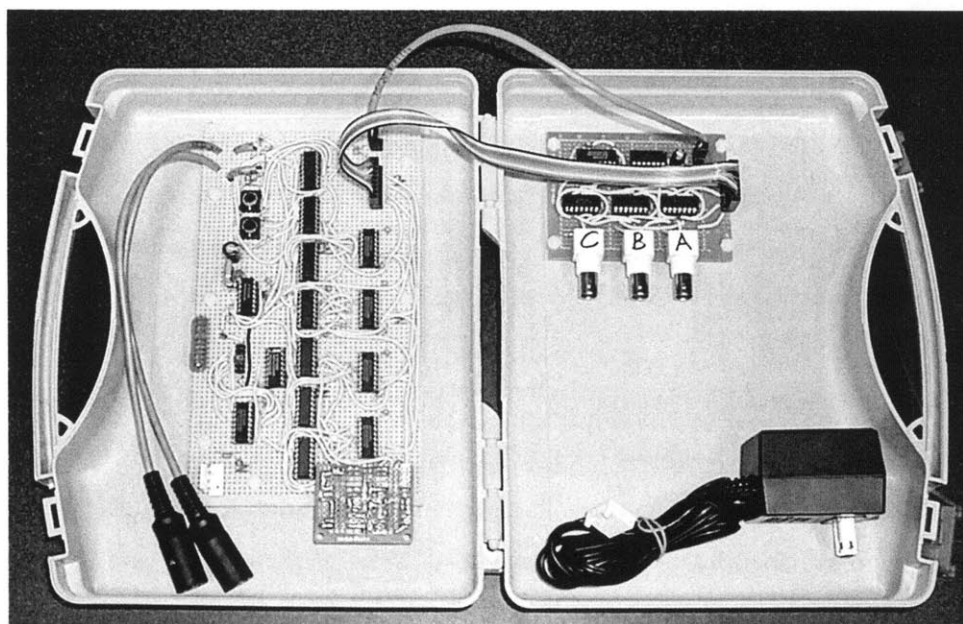
Figure 3.3.1: MIDI Converter Mounted in Portable Case

A 4 MHz crystal is used as the main oscillator for the board, and is divided down to 31.25 kHz for the standard MIDI baud rate. It is further divided down for the 2 ms to 64 ms data windowing. A high frequency crystal is chosen for its stability over temperature, time, and shock load, in comparison to a 31.25 kHz crystal. It is also mounted in a socket to prevent soldering related heat damage, and allow for ease of replacement in case of failure.

3.4 Lighting Controller

The MIDI converter data is transmitted to two separate computers. The first is a Macintosh G4, which performs the data analysis and musical mappings. The second is an IBM–compatible personal computer with an Intel 486 processor. This second computer performs lighting control mappings based upon both the raw sensor data from the MIDI converter, and relevant processed data from the Macintosh G4. Both of these data are received by a midiman MERGE 2×2 device which combines the signals, and outputs a single MIDI data line containing both sets of data.

After performing the required mappings from the received data, the lighting control computer sends out control information via the parallel port. This information is then converted to the DMX serial communications protocol via an interface [19] designed by Josh Randall. This interface utilizes an Atmel AV90S8515 microcontroller to convert the parallel data stream to DMX dimmer addresses and values, which are sent out via a MAX485 RS-485 line driver. This data stream then effects the appropriate changes in the lighting conditions.

Chapter 4

Interaction Design

4.1 Operational Platform

As the dancers move in response to the music, the sensors that they are holding transmit RF pulses in time with their motions. These pulses are summed at the MIDI converter, and the sums are sent to three separate computers. The first is a Macintosh G4, which performs the data analysis and musical mappings. This computer receives and sends MIDI data via an emagic Unitor8 MKII, MIDI to USB interface. This device has eight independent MIDI in and out ports, which merge to a single USB interface. The second computer is an IBM-compatible personal computer with an Intel 486 processor. This second computer performs lighting control mappings based upon both the raw sensor data from the MIDI converter, and relevant processed data from the Macintosh G4. Both of these data are received by a midiman MERGE 2×2 device which combines the signals, and outputs a single MIDI data line containing both sets of data. These data enter the computer via a built-in MIDI interface (MPU401 port) on its sound card. The third computer is another Macintosh G4, which receives a copy of the lighting controller computer input from the MIDI merger. These data are transferred to the computer via USB through a midiman MIDISPORT 4×4, MIDI to USB interface. This computer then logs all event data for future analysis.

The data analysis and musical mappings are performed in the MAX programming environment. MAX is a graphical data flow language designed to handle MIDI data events. Data flow and processing commands are stored as 'patches', which are combinations of functional blocks tied together by graphical lines that determine information pathways. For this reason, it is a natural choice for this application, since the computer needs to interface with hardware music synthesizers and the MIDI converter, both of which communicate via MIDI. Another advantage of MAX is its ability to allow for real-time mapping changes to be made without having to restart the program. Despite these advantages, MAX does have the drawback of being computationally and memory intensive to operate. If patches are not efficiently written, or too many patches are running concurrently, the data flow can take significantly longer than the time allotted between sonic events. In these cases, the music either stutters, or stops completely, as the computer struggles to finish its tasks.

Because of the aforementioned problem with MAX, external hardware music synthesizers were chosen as the means of creating sonic events, as compared to internal software synthesizers. The computationally intensive nature of the musical algorithms would leave little processing power for the software synthesizers to perform their required operations, and either MAX or the software synthesizers would cease to function. In either case, the effect would be the same; the music would stop playing. A fourth computer could have been enlisted to operate the software synthesizers, but since the data was already being converted to MIDI event messages, it was determined that the system would be more reliable if the data did not have to go through another two conversions, first to enter the fourth computer, and then finally to exit the computer and convert the synthesized notes to sound. A listing of all hardware synthesizers used can be found in the Interaction Testing and Results chapter, where the specifics of each event's mappings are located.

The hardware music synthesizers receive their MIDI commands from the emagic Unitor8 MKII. The Unitor8 is arranged so that no single port has both its input and output simultaneously connected to a data line. Also, each synthesizer has its own dedicated MIDI port. These arrangements are made to eliminate the possibility of data congestion on any one port. A MIDIComposer QuickShot keyboard controller is connected to a MIDI port, to

input commands to the MAX patches. During data collection events, if changes to patches are required, the act of clicking on windows and entering data in running MAX patches can cause the data processing to slow down while the new data is entered. To eliminate this potential problem, data commands are mapped to the keyboard controller, allowing them to be entered without having to interface with any of the MAX windows. Finally, within each synthesizer, each voice is given its own MIDI channel, allowing multiple voices to be played at the same time without voice overload and channel sharing issues.

The chosen musical style for all mappings is electronic dance music with a 4/4 time signature. This is done for several reasons. First, the music must be generated by electronic means for simplicity of control by a completely electronic system. Secondly, as a deejay of electronic dance music, this is where the author's compositional knowledge lies, allowing for mappings to be based upon an understanding of the typical listener's expectations and responses. Third, electronic dance music tends to follow a fairly simple set of rules for its composition, and can be easily scripted to data. The 4/4 time signature is chosen to facilitate a strong response in the FFT of the data signal, making the system very sensitive to tempo changes. Finally, the electronic dance music community quickly embraces new technology, making a system of this nature more viable in this venue as compared to others.

4.2   Goals

The main objective of the system, being primarily for entertainment, is the enjoyment of the participants. If they do not find the system sonically pleasing, it will not be used. The system must be intuitive to use, providing appropriate feedback to participants, so that their actions will naturally follow the expected behavior assumed by the mappings. The users should also be given a sense of causality, and a knowledge of what outcomes specific actions will create. This will allow them to dictate where the experience is directed, giving them a tool for sonic exploration, engaging them and encouraging them to continue using the system. The system must also be responsive, creating an environment that truly reflects the desires of its users.

41

Secondary goals include encouraging collaboration amongst participants, helping them to build new musical structures into their experience. As a test of the functionality of the system as a tool to direct behavior into certain patterns, tertiary goals of encouraging synchronization and increasing activity level of participants is chosen. These goals are selected due to natural human tendencies toward these behaviors [12]. If the audience can be schooled into these outcomes without prior training on the system, then intuitive feedback must have been given.

4.3 Heuristics

The nature of musical mapping for interactive instruments is not well documented. Each performance is determined by the experience of the interaction designer, and takes on his or her particular feel. Although this work is also heavily influenced by the experience of the author, it will try to show a logic behind the interaction, and explain the reasons for each section's functionality. Currently, there is very little research in the area of large group behavior while dancing; how dancers respond to music and what parameters are most effective or causal. However, there is no reason to believe that dance is too disparate from other human activities, and that human responses do not bear certain fundamental characteristics. A very interesting report [12] on crowds of people interacting involves the human clapping response. As anyone who has been to a public performance will note, the clapping of audience members will start chaotic and dispersed, then very quickly synchronize, so that the majority of members are clapping in unison. This combined clapping tempo will subsequently increase until it finally breaks back into chaos. This pattern, combining, building, breaking down, will then repeat for as long as the clapping continues.

What does this knowledge say about the nature of human interaction? It seems to imply a number of important things. First, humans will naturally school, if given appropriate feedback as to their current state. There is a herd instinct that encourages the individual to respond in accordance with the majority. Second, humans act in the manner of a positive feedback network. They drive each other to increase his or her activity, each time responding

to their neighbors' increases with an increase of his or her own, building to a level which is no longer stable or pleasing. Finally, it is at this point that order breaks and the users naturally return to a ground state, as there is no longer suitable feedback to denote what the majority is doing.

It is therefore assumed that, if given the appropriate feedback, humans will tend to create order from chaos, to become aligned with one another to generate a coherent and pleasing outcome. It is further assumed, that once this has occurred, given the appropriate feedback, they will build upon each other's activity to increase the general energy state of the group. Finally, it is assumed that the group will reach an energy state that is no longer maintainable, and will bring the energy back down, for it to build again. It is these three assumptions that will influence the philosophy of the design.

These assumptions pose two very difficult questions. What is appropriate feedback? And, what is the musical sonification of chaos, order, and energy? The first question is straightforward, but has multiple answers, depending upon the current state of the participants. This question can also be tested for validity by applying the chosen form of feedback and noting whether participant behavior follows the above pattern. The second question is not as straightforward, and is not as easily tested for validity. The answer will tend to vary from participant to participant, as each person has a different interpretation of music. But, there is some common ground which can be agreed upon between listeners [23], and this is the area which must be built upon in the musical mappings.

4.3.1 Appropriate feedback

Given a low participant energy state, in order for users to become aware of their actions, and to direct these actions into a coherent outcome, they must be given a direct response for each and every action. In this manner, they will realize what outcome their actions have, and be able to control not only the occurrence, but the timing of that outcome. They will then be able to make the decision whether or not to produce that outcome in time with the perceived average timing of all outcomes. A limitation to this scheme, especially with

43

this system, is that in order for it to work, the amount of responses must be limited, so that an average timing can be detected. With too many responses, as is the case for large crowds, the sum of all outcomes can easily stay in a cacophonic state, without building to coherency. The feedback mechanism employed in the clapping response does not translate to this system for groups of people greater than approximately twenty. With clapping, each response is created locally, and is heard locally. Each person clapping mainly hears his or her neighbors, and can quickly synchronize with those around him or her. With this system, all responses are summed globally, and in turn, effected globally. The sum of all responses, as a result, can easily become chaotic. To eliminate this problem, a secondary form of feedback can be given. The system has the ability to detect the average timing of all outcomes, and can not only give a direct response, but also a larger response that is timed to the computationally perceived average outcome. In this manner, participants can quickly sense the average outcome and chose to either coincide or syncopate.

Once the participants have become synchronized around a central outcome, the energy level of the music must increase with increased energy level of the participants. Through changing dancing patterns, the participants can then dictate whether or not to increase or decrease the perceived state of the interaction. This is a very intuitive interface for the dancers, as they will naturally increase their actions if the music becomes more energetic. If a negative feedback were to be applied, where increased activity decreased the energy level of the music, it is hypothesized that the music would tend to either stagnate, or oscillate between two states, depending upon the granularity of the energy transitions. If the dancers perceive the music to be interesting, they will become more active, decreasing the energy of the music, reducing their incentive to be active. As they become disengaged, the music would return to its previous state, re-engaging the participants. Once participants reach a consistent level, whether it be oscillatory or stagnant, they can quickly become bored due to the lack of progress in the music, and become disengaged completely. As a result of the negative feedback, this ultimate decrease in activity would turn the music into its highest energy state, becoming too disparate from the users' current state and discouraging further exploration. For this reason, it is assumed that, on a global scale, the music must be allowed to build with the participants' activity.

It is a fundamental law of nature that energy can not build indefinitely. As a result, the energy level of the music will reach a maximum at some point during the interaction. When this occurs, the participants will need to be given some method of transitioning to another, lower state. This can occur in a number of ways, but it must be done in a fashion that either maintains a global average outcome, or allows the users to build a new global average outcome. It is preferable that this highest energy state not merely return the participants to the previous energy state. If this were to happen, it would have the equivalent effect of negative feedback, and the crowd would settle into oscillatory behavior, before quickly becoming disinterested with the system. It as also preferable that this highest energy state become slightly chaotic, denoting to the users that it has been reached. This will have the combined effect of disrupting group behavior in a natural way, ensuring that the energy level decreases, and setting the users' expectations of what will occur next. They will then be able to build a new structure from the ground up, adding variety and a sense of exploration to the experience.

4.3.2 Sonic characteristics of chaos, order, and energy

For the particular style of music chosen here, definitions for such parameters as chaos, order, and energy can be more easily found than for other genres of music. Electronic dance music, especially techno and trance, tend to follow strict rules for the progression of the music, and variations from these progressions are quickly detected by the listener. The following characteristics will be defined based upon the author's observation of hundreds of dance events, and countless hours of listening to this particular style of music.

It is difficult to discuss the musical meaning of chaos without also discussing the musical meaning of order. The difficulty with defining chaos in this application, is that it is rarely employed in techno and trance, as it breaks the continuity of the experience. We choose to employ it in our mappings as a way of allowing the participants to control the breakdown sections of the music, rather than dictate the way in which energy will be redistributed, as it is done in deejayed events. The main exception to this rule is in the case of experimental hardcore, where chaotic sounds are used to denote the extreme of energy levels. Perhaps

the best way to define chaos is by first defining order, thereby creating a set of rules to be broken to create chaos.

As a way of discussing the various aspects of an electronic music composition, a track will be broken down into four distinct elements. The first is the rhythm structure, the second is arpeggiated lines, the third, drones, and the fourth, melodies. A composition may have all or only one of these elements. Some elements can also have multiple occurrences in the same track. The rhythm structure is the main controlling factor for electronic music. The structure must repeat at regular intervals, so that the next beat's position can be known by the dancer. This repetition creates familiarity and order. The more simple the repetition structure (the simplest being equally spaced kick drums), the more ordered the music will feel. The number of voices in the rhythm track, and each voice's pitch, also add to the sense of order. Fewer voices imply more order, and lower pitched voices have more influence over this sense or order.

The remainder of the elements follow similar rules. For arpeggiated lines and melodies, the note selection needs to follow a pattern that repeats with an integral number of measures, preferably one, two, or four. The voicing of these elements must also not vary too much during the course of the repeated section. Drones, for the most part, should repeat on longer time scales, with variations occurring less than four times a measure, and having long attack and decay times. Also, the drone should not dominate over the other voices. All of the voices must be in the same key, preferably harmonizing. Finally, the composition can not become too cluttered. Too many voices playing at the same time, especially conflicting melodies, create confusion and sensory overload in the average dancer. The opposite is also true. For example, if a piece becomes too thin, with only a rhythm structure, or only an arpeggiated line present, the listener's expectations will not be met, creating either apprehension or eventual boredom, which can only be released by insertion of the missing elements.

Chaos can therefore be created by violating any combination of the above rules. Playing random notes, activating rhythm elements at random points in time, or increasing the number of voices present are just a few of the ways this can be done. Many of these methods tend to create tension amongst the participants, which is a necessary element of

any composition. Without changing tension, there is no sense of progression, and no need to continue with the experience. The desire for release captures the interest of the listener, especially after starting with a well ordered section that invests the listener in maintaining that order. These elements can therefore be used to encourage changes in the participant's actions, only giving release when a new or desired outcome is reached.

The energy level of a musical piece is controlled by many factors. Perhaps the most apparent is the tempo. As tempo increases, so does the perceived energy level. The nature of the voices selected also has a strong influence. Typically, higher pitched and more complex voices, with sharper attack and decay, will accompany sections of music with increased energy. The presence of a strong beat is also characteristic of energetic music. Low energy levels may not have any beat at all, whereas high energy sections might be completely dominated by the sound of the drum kit.

The complexity of the composition has an interesting response with respect to energy level. Very minimal music is often termed ambient, because of its use as a backdrop, or in setting the sonic landscape. The number of voices, and the complexity of these voices increases with increased energy level, until a certain point, where the music again becomes less complex, and very focused. An example of this is in high energy melodic hardcore, which usually consists of little more than a kick drum and a modulated melody. As energy increases, these voices will tend to repeat on shorter time frames, and become more distorted, or modulated. Cleaner timbres are reserved for lower energy states, which generally have grand, sweeping sounds.

All of these effects can be combined in any number of ways to increase or decrease the perceived energy level. The order of the music can also be varied in accordance with the user's expectation of increasing chaos as energy increases. In this manner, the users can obtain feedback as to the average state of the crowd, and respond accordingly. The assumption that users will gravitate toward states of order and build toward higher energy levels, allows for the design of musical mappings that can either lead or follow the participants in any number of directions. It also allows for self exploration by the participants, as they determine the next sonic event by moving through different levels of music, deciding which

selections are more pleasing, and responding with increased activity.

4.4   Implementation

In order to provide the appropriate feedback, as discussed in the previous sections, the current state of the audience must be known. The aggregate receiver data is the only method available to determine this information, and as such, it becomes crucial to know how these data relate to such parameters as average crowd tempo and activity level. Since the amount of previous testing with the system is limited, some of these factors must be assumed at first, and then correlated to event data to determine whether the assumptions are true. The following section will describe these assumptions, and detail the data processing algorithms used to extract relevant parameters. The most recent algorithms will be discussed, and specifics of musical mappings used at various data collection events will be elaborated upon in the Interaction Testing and Results chapter.

The main framework for the musical mappings is fairly rigid. The music is scripted into various pieces, with each piece having a corresponding energy level associated with it. The users can then select which piece is played by changing their energy state. Within each energy state, various forms of real-time control are given to the participants, and this control increases with the energy level. This somewhat rigorous scripting of the data is done for two reasons. First, it is not yet known how to give a large group of untrained participants full control of sonic events, and still have a structured output. Secondly, since the primary goal of the system is for the enjoyment of the users, it was decided to err on the side of a more ordered and pleasing experience, rather than a chaotic and potentially displeasing event which receives little participation.

The perceived energy state of the audience is divided into five levels, with each level being proportional to the assumed maximum of four hits per person per measure. At levels less than one quarter of this maximum, *level1*, ambient pieces are played, consisting mostly of drones, which neither discourage nor encourage activity, and act more as a resting place. From one quarter to one half of the maximum, *level2*, the drone continues, and rhythmic

48

features and melodies are added. From one half to three quarters of the maximum, *level3*, more complicated rhythmic patterns emerge, and arpeggiated lines are added. From three quarters to 95% of the maximum, *level4*, more complicated arpeggiated lines appear, and the beat structure simplifies. The melodies and drones become less apparent, and the drum kit tends to dominate. At activity levels above 95%, *level5*, the music consists mainly of rhythmic tracks and arpeggiated lines, and the note selection and drum voicing is determined solely by user input. This creates a indeterminate musical experience which can quickly deteriorate into chaos, signaling to the users that the highest energy state has been reached.

In conjunction with these more scripted sonic events, there is also real-time controlled voicing. This is done to give a sense of responsiveness, and allow the audience to shape a portion of the musical experience. These real-time controllers are determined by parameters extracted from the received data. A listing of all currently used parameters can be seen in Table 4.4.1, and are described below.

Data Parameters used for Mappings
Arrival time of hit
Time since last hit
Value of each hit
10 s accumulation of hits
2 s accumulation of hits
1 s accumulation of hits
500 ms accumulation of hits
80 ms accumulation of hits
difference of 500 ms accumulation and 1 s accumulation
80 ms accumulation divided by 2 s accumulation
Percentage of hits in each of 16 steps of previous measure
FFT peak frequency
Magnitude of FFT peak frequency

Table 4.4.1: Data Parameters used for Mappings

The most basic piece of information available to the system is whether or not a sensor has sent an RF transmission (this condition is referred to as a 'hit'). Each hit indicates that a user has just crossed an acceleration threshold, deliberately giving input to the system. For low energy level conditions, when few hits are arriving, these hits can be mapped directly

to sonic events, giving prompt feedback to the users that the system is working, and that they are contributing to its input.

The next level of complexity involves calculating the time since the last hit arrived. If it has been a long time, it is more likely that the users are at a lower energy state, as little motion is occurring at the sensors. This is a direct response that can give an instantaneous feel for the state of the audience. It can be used to determine whether or not to give direct feedback on each hit, as the sound would be chaotic if too many hits were arriving at once. It can also be used to modify the sound of the sonic event occurring with each hit, so as to slowly fade out the direct response as more hits arrive.

Accompanying each incoming hit is a value of that hit, which represents how many people activated their sensors in the past windowing period. This windowing period is set at the MIDI converter, and is on the order of 16 ms. Since this is a relatively short time period, hit values greater than one will only be seen at high energy levels. Values greater than one imply tightly schooled behavior, and can be used to give positive feedback for this action by providing an instantaneous modulation to a voice, or creating a sonic event to signify that this has occurred.

The hits for any period of time can be accumulated, creating a moving average low-pass filter on the data. These averages can be used to give a more accurate view of the activity level of the crowd. The longer the time period of accumulation, the more representative the data will be, although it will introduce a time lag of one half the total integration time. From previous tests with the system, a ten second integration time is found to be the most appropriate to eliminate spurious activity, but still allow for the system to respond in a causal fashion. For this reason, the ten second moving average low-pass filter output is used to assess the current energy level of the participants. This can be used to select appropriate musical tracks to match the activity of the audience.

A two second moving average filter is then used to give a feel for the direction the activity level is taking, and can be used as a predictor for selecting appropriate musical tracks. For example, it is assumed that participants can decrease in activity level for two reasons. First,

they are no longer interested in the current musical track playing, but are still interested in maintaining that particular energy level. Second, they are becoming tired, and wish to reduce in energy level. The two second filter can be used to detect a sudden change in activity, and transition into a new track at the same energy level. If the activity continues to fall, such that the ten second average reduces in level, then the system can assume that lower energy is desired, and transition down to the next level accordingly. But, if the two second data returns back to its original energy level after the new track has started, it can be assumed that merely a new track was desired, and that the system should continue to stay at its current energy state.

This strategy can also carry over to increases in energy level. An increase in activity can signal that the participants are enjoying the current track, or that they wish to move to the next energy level. Upon transitions, the two second data can be used to signal the introduction of the next energy level, and if the ten second data follows this trend, then the transition is completed. If the two second data returns to the original energy level, then the original track is brought back in, and the new track is faded out. These sorts of transitions can occur in a seamless fashion if the music between energy levels is compatible, and forays into new energy levels do not last too long.

Shorter time averages can be very useful as real-time controllers. Both the half second and one second integrations undulate with the motions of the crowd, and often in time with the music, increasing with each beat. They act as excellent inputs to set the amplitude of low frequency oscillators (LFOs), affecting drone timbres or modulating cut-off frequency and resonance of arpeggiated lines. The overall level of the integrations rise with activity, and slight modulations on this level occur as people respond to the music. The half second integration allows more high frequency signals to pass through, and as a result works better on quickly changing lines, such as melodies. The one second integration responds slower, making it more suitable for LFO modulation.

Finally, an 80 ms integration is used to interrogate the data for clustered activity. The time of integration is chosen to capture as much simultaneous activity as possible, but still allow for a response which appears to have no delay. The human ear is not capable of sensing

delays less than a certain level, and 80 ms is just over this threshold. In this manner, if a group of people coincide within 80 ms of each other, they can be given immediate feedback as to the fact that this has occurred. Positive feedback can then be given to encourage more of this behavior in the future. These clustered hits can also be used to begin to build features into the music, and allow users to create their own rhythm structure.

A difference of the half second and one second data is also taken. This gives a rate of change of the current activity level. The value will increase if activity is increasing, stay at zero if activity has plateaued, and dip negative if activity is decreasing. This, too, makes an excellent controller for quickly changing sounds, such as arpeggiated lines or drum beats. It can be used to accent beats if activity is increasing, or slide between arpeggiated notes if activity is decreasing. It functions well as a musical controller, as it is constantly changing, making it musically interesting.

Another method of measuring the change in activity is by dividing the 80 ms accumulation by the two second accumulation. This will return a value representing the percentage of the current activity level due to the instantaneous contribution of activity. As before, this is useful as a continuous sound controller, and is constantly changing. It has the advantage, over the previous method, of being instantaneous, and as such will correlate to current sonic events. This means it can be used to give direct feedback as to the depth of the crowd's activity modulation, to encourage working together and synchronizing.

The system is also capable of looking at events that occurred in the recent past, and evaluating how well they correlated to sonic events played at that time. If the number of hits received over the last measure is accumulated, and that measure is divided into sixteen equal steps, the total percentage of activity that occurred in each step can be found. This will give information as to phase differences between drum beats and user responses. It is particularly useful for creating rhythmic structures or performing note selection for arpeggiated lines. For portions of higher activity, lower drum sounds or higher arpeggiated notes are selected. For steps with lower activity, high hats or snare drums can be triggered, and lower arpeggiated notes can be played. A threshold can also be set that triggers an event, such as a hand clap, if there is more than a certain percentage of activity in a

52

step. this is a very effective way of giving the audience control of the rhythmic structure. A histogram of the past eight measures can be taken to determine which steps have the highest probability of high activity levels. These steps can then be assigned certain drum voices, and patterns built in by the users will last for approximately four measures, and if new patterns develop, they can be quickly integrated.

The final control parameter is, perhaps, the most powerful. From previous testing, the peak frequency of a Fast Fourier Transform (FFT) of the data signal is shown to correlate strongly with the dominant tempo of the participants. A MAX object created by Michael Broxton, called Beatgrabber, takes a rolling FFT of the received data signal over a 30 s window, returning the peak frequency at one second intervals. The complete code for this object can be found in Appendix C. This peak frequency can be easily converted to beats per minute (BPM) to set the global tempo of the structured framework. In this way, the participants have direct control over the tempo of the music. For the designer to encourage faster tempos, the global tempo of the generated music is set to the perceived tempo plus one. To decrease tempo, it is set to the perceived tempo minus one. The Beatgrabber object can also be configured to output the magnitude of the peak frequency. This is a good indicator of the percentage of the audience dancing at the dominant tempo, and might be used to encourage more people to dance in time, by allowing the beats to become more prominent as the amplitude increases.

The tempo itself is a good indicator of energy level. As the audience speeds up, it can be assumed that they are becoming more energetic, even if the total hits per measure has not increased. A caveat to determining energy level is that the total amount of sensors present must be known, since the energy level is set by the number of hits received per unit time per person. An increase in hits could either be a result of more people entering the dance floor, or those currently on the dance floor becoming more active. Since there is no way of extracting the total number of people from the data stream, this information must be entered external to the system. In most cases, it can be done fairly effectively by counting the number of people currently on the dance floor, and manually entering this number into the system. To compensate for tempo effects on perceived energy level, a weighted average

53

of both tempo and number of participants is used to determine the assumed maximum value, which is then used to normalize the sensed activity level. The exact equation used for the most recent event mappings is as follows.

$$\text{Maximum Activity} \left[\frac{hits}{measure} \right] = 4 \times \left(\frac{1 + \frac{120}{BPM}}{2} \right) \times (number\ of\ people) \qquad (4.4.1)$$

This has the effect of reducing the perceived maximum value as tempo increases, allowing higher energy states to be achieved with lower amounts of activity. For example, *level3* starts when the hits per measure equal one half of the perceived maximum. For ten people dancing at 120 BPM, the perceived maximum would be 40 hits per measure. They would therefore enter *level3* when their activity produced 20 hits per measure. For the same group of ten people dancing at 240 BPM, the perceived maximum would be 30 hits per measure, allowing them to enter *level3* with only 15 hits per measure.

The majority of these mappings are based upon the author's experience as both a dancer and a deejay, incorporating a familiarity with what is both desired by the listener, and what are effective parameters to change as a controller. In a sense, the actions of a deejay have been duplicated within these musical mappings. Ultimately, the dancer needs constant change to maintain any state, and as such, the system also needs to be constantly changing. If the system can detect and respond to those changes that produce increased activity, it can begin to act as a deejay does. It would then need to elicit the particular parameter of the change that caused the activity to increase, and incorporate more of that parameter. Along with any modifications, it must be remembered that the system needs to be causal and intuitive to use. This means giving both immediate responses for a sense of control, and delayed responses that modify global parameters for a sense of responsiveness and complexity. In this manner, the system can respond effectively to its participants, creating an extremely captivating experience.

Chapter 5

Hardware Testing and Results

The following testing was done on the final revision of the sensors and receiver base stations.

5.1 RF Transmitters and Receivers

Each sensor's transmitter is tuned to a single, master frequency by rotating its variable capacitor until a single, master receiver base station receives its signal and triggers a piezo buzzer to sound. This tuning process is performed with a distance between the transmitter and receiver of at least 5 m. The tuning of the transmitters is a difficult and time consuming task. The capacitor sweeps through a range of approximately 2–5 pF, which correlates to frequency variations of $\pm 30\%$. The receiver requires a frequency tolerance of approximately $\pm 0.3\%$, which is 1/100 of the capacitor's full scale, which is only one half turn of the capacitor. To further complicate this task, the receiver gives no response if the transmission frequency is not within tolerance, and gives no variation in response as the transmission frequency varies about the center frequency. This creates a system with no feedback, making the tuning a somewhat random process, where the capacitor is merely turned until a sound is heard, and it is not known whether or not the transmitter is exactly at the center frequency. For this reason, the transmitters are all tuned to slightly different frequencies. Despite this fact, all transmitters are capable of making the receiver respond at a distance of 5 m. As

distance between the transmitter and receiver increases, those which are not precisely tuned to the center frequency cease to be detected. For this reason, the effective reception distance of each transmitter varies.

The maximum reception distance is tested by placing the master receiver in a fixed position, and moving the transmitter away from it, while activating the transmitter. In this way, the distance at which reception no longer occurs can be noted. Twenty sensors were tested in this fashion, and their reception distances varied from a maximum of 15 m in an open space, to a minimum of 10 m in the same space. Even at 10 m, the RF transmissions are adequate for the system to function.

The RF transmissions were then tested on five sensors in a crowded environment. The base station and transmitters were taken to the Park Street subway station in Boston, Massachusetts during rush hour on a business day. This is the busiest of public transit spaces in the Boston metropolitan area, and is usually filled with over one thousand people, all standing shoulder to shoulder. The master receiver was placed in a fixed position at about five feet above the ground. A transmitter, at about five feet above the ground, was then moved through the crowd, being activated at regular intervals. The distance at which the transmissions no longer arrived at the base station was approximated, since making distance measurements in such a crowded environment is difficult. The transmissions were consistently received at a distance of 3 m, and received 50% percent of the time at a distance of 5 m. Reception quickly ceased at distances greater than 5 m.

During testing, a number of factors were noted that influenced the reception of RF transmissions. The first is relative position of the transmitter to receiver. Not only is distance a factor, but also spatial orientation of the respective antennae. The way in which the transmitter is held has a significant impact. Often, while holding the transmitter, the user's hand shields the RF transmission. Finally, dense objects such as structural elements of buildings or other people, severely degrade performance by shadowing the receiver.

Since multiple base stations are used, each base station must also be tuned to the single, master frequency. This is done in a similar fashion to the tuning of the transmitters. A

tuned sensor is set to continuously transmit, and the receiver's tuning capacitor is varied until the signal is received. this procedure is repeated with multiple sensors and the same base station, to ensure that the base station is set to the average of the sensors' transmission frequencies. Despite tuning the receivers to respond to multiple sensors, all of the tuned receivers have reduced performance in comparison to the master receiver. This can be witnessed as a 30% reduction in maximum reception distance of the tuned receivers.

5.2 Sensor Battery Life

The current draw of the sensor was measured with a Hewlett Packard 974A Multimeter, which has a resolution of $0.01\,\mu A$. When the sensor's transmitter is off, the meter reads $0.01\,\mu A$, which suggests a current consumption of no more than $0.01\,\mu A$. At this rate, the $48\,mAh$ battery would last for over five hundred years. Of course, the actual battery life would be much less than this, due to its limited shelf life of ten years. The current consumption of the circuit was then measured for its debouncing phase and transmission phase. While the transmitter is on, it consumes $3\,mA$ of current, and the debouncing circuitry consumes $60\,\mu A$ of current. Since the transmitter is on for $50\mu s$ per transmission, and the debouncing circuitry is on for $150\,ms$ per transmission, this gives a total of $9.15\,\mu As$ of current consumption per transmission. At a rate of two transmissions per second, the $48\,mAh$ battery would last for 109 days. In practical application, the total transmission life would be approximately one half of the time calculated, as the voltage on the battery drops significantly enough after partial discharge to make the RF transmission too weak for reception. This gives the sensors a ten year shelf life, and at least a month of continuous usage before battery failure.

5.3 Sensor Sensitivity

The sensitivity of the sensors is not quantitatively assessed, as an exact numerical value is not germane to the project. However, each sensor is qualitatively tested for sensitivity, and

grouped accordingly. This test involves moving the sensor and listening for the audible chirp of the master receiver base station, at a distance of 5 m. The sensitivity of each transmitter is assigned a number from one to four. One represents a sensor that is too sensitive for practical use. If the sensor triggers when rotated slowly in the hand, or when gently picked up, it is assigned a one. The next level of sensitivity, level two, is the most useful. The sensor does not trigger under gentle motions, but consistently triggers when moved back and forth over a distance of 30 cm at a rate of 2 Hz. Assuming a sinusoidal motion, this equates to approximately 2.5 G. Since the level one sensitivity sensors trigger upon slow rotation in the hand, they are experiencing 2 G, or the reversal of the vibration sensor in earth's gravitational field. Level three sensors activate consistently at a back and forth motion of 35 cm at 2 Hz, or approximately 3 G. These are less useful, but required since the amount of level two sensors is limited. Finally, level four sensors only trigger under extreme acceleration. In some cases, the piezo film itself needs to be struck to force a transmission. These are not used, except for in destructive testing.

The wide variation in sensitivity of the sensors causes many problems. First, the average cost of a useful sensor increases as the number of useful sensors per batch manufactured decreases. Secondly, if the response of each sensor varies, all participants are not given the same level of input, as a motion by one participant may not have the same effect as a motion by another with a different sensor. Of the five hundred sensor boards assembled, 6% are level one, 47% are level two, 29% are level three, and 15% are level four. The remaining 3% are either not tunable or malfunctioning for another reason. This leaves only 47% of the sensors in the appropriate range for usage. For this reason, testing was performed to try to discover the primary cause of this variation in sensitivity.

The first possible mode of variation in sensor sensitivity is the difference in piezo vibration sensor sensitivity. Piezo sensors were removed from ten level four sensors. These were then connected to a FET input op-amp in unity gain buffer mode, and tested for voltage response. The same was done for a sensor with sensitivity level two. No noticeable difference was detected between any of the level four sensors and the level two sensor, eliminating vibration sensor variations as a possible cause of failure.

The next possible cause for loss of sensitivity is due to low input impedance as seen by the piezo vibration sensor. The piezo is connected to a $110\,M\Omega$ resistor chain to bleed off charge buildup, and to the input stage of the 74HC221. This gives two possible paths for current, and a third exists through oils and flux on the surface of the circuit board. Measuring the impedance seen between the pins of the piezo film shows a resistance of $1\,M\Omega$ or less on most of the level four sensors, as compared to $20\text{--}100\,M\Omega$ as seen on level two sensors. The variations in input impedance correlates strongly with the sensitivity variations, as piezo sensors have an extremely high output impedance at these excitation frequencies, on the order of $100\,M\Omega$, that cause their output voltage to decrease with loads less than $10\,M\Omega$. The decreased input impedance of the level four sensors degrades the piezo output voltage to the point where it no longer triggers the 74HC221.

Since surface oil variations are easier to modify than the other two current pathways, this parameter was analyzed first. Circuit board cleaner was used to clean ten level four boards around the piezo vibration sensor. Only two of the ten boards showed an increase in resistance when measured at the vibration sensor contacts, and four boards actually showed a decrease in resistance. The remaining boards showed little change in their resistance value. Next, the top layer of the circuit board, between the piezo contacts, was removed by scraping away the material with a knife. This decreased the resistance in all ten boards, most likely due to debris from the cutting operation conducting between the contacts. Further cleaning with circuit board cleaner showed little improvement. Although surface oils can reduce the impedance seen by the piezo film, it is apparent that the cleaning operation performed by the board stuffing firm is adequate to remove any of these oils.

Next, the resistor chain variations were analyzed. Ten level four circuit boards were modified by cutting the electrical leads between the piezo film and the resistor chain. In this manner, the resistor chain can be measured independent of the remainder of the circuit. Measuring the net resistance of these chains showed variations from $20\,M\Omega$ to $100\,M\Omega$, which, although out of specification for a chain of five $22\,M\Omega$ resistors, is still far greater than the impedance seen by the piezo film. This is also equivalent to the variations measured on level two boards. It is therefore concluded that resistor chain variations are not contributing to the

low impedance seen by the piezo film.

Finally, the input pin of the 74HC221 was electrically isolated from the rest of the circuit, and its input impedance was measured. For the ten level four sensors measured, the input impedance was on the order of 1 MΩ. Ten level two sensors were then modified and measured in the same manner. The level two boards show input impedances greater than 100 MΩ. By referencing the 74HC221 manufacturer's datasheet [24], it is noted that the input impedance is only guaranteed to be at least 1 MΩ. This confirms that the sensor sensitivity variations are a result of manufacturing tolerances on the input impedance of the 74HC221. Since this is a parameter which can not be controlled, no modifications were made, and the sensors are used in their current state.

Chapter 6

Interaction Testing and Results

Throughout the course of this work, many different stages of testing occurred, examining both the functionality of the hardware and the nature of the human response to the system. What follows is a summary of the most important tests and their findings.

6.1 Clapping Test

The very first test of the system was conducted to ensure that the transmitters and data collection systems functioned properly. This test consisted of seven people, each with a sensor, attempting to clap in unison. One person was singled out beforehand as the leader, and all other participants were encouraged to follow his lead. In this manner, a tighter synchronization of clapping could be obtained. The sensors used for this experiment are those described as revision two in the Hardware Design chapter. They consisted of a metal cantilever beam vibration detector, and a Ming Tx-99 V3.0 transmitter. The receiver base station was a Ming RX-99 V3.0, with a low-pass filter on the output stage. This low-pass filter acted to eliminate background noise on the receiver.

A graph of the receiver output voltage versus time can be seen in Figure 6.1.1. It shows that the transmitters did work, and that the sensors triggered with each clap. It also shows

that grouped activity can be detected, as the transmissions occur within a short time period of each clap. An expanded view of one of the clustered events is shown in Figure 6.1.2. As this graph shows, the apparent clustering of sensor hits actually occurs over a relatively large time window, in comparison to the $50\mu s$ transmitter pulse width. Although, to those clapping, the sound of the claps appeared to be in unison, they were actually spread out over approximately 200 ms. It is this result that proves the viability of the system. The random variation in user action reduces the likelihood of transmitter collisions to a negligible level. For this reason, the sensor transmissions of a very large crowd can be collected, without significant loss of data from collisions.
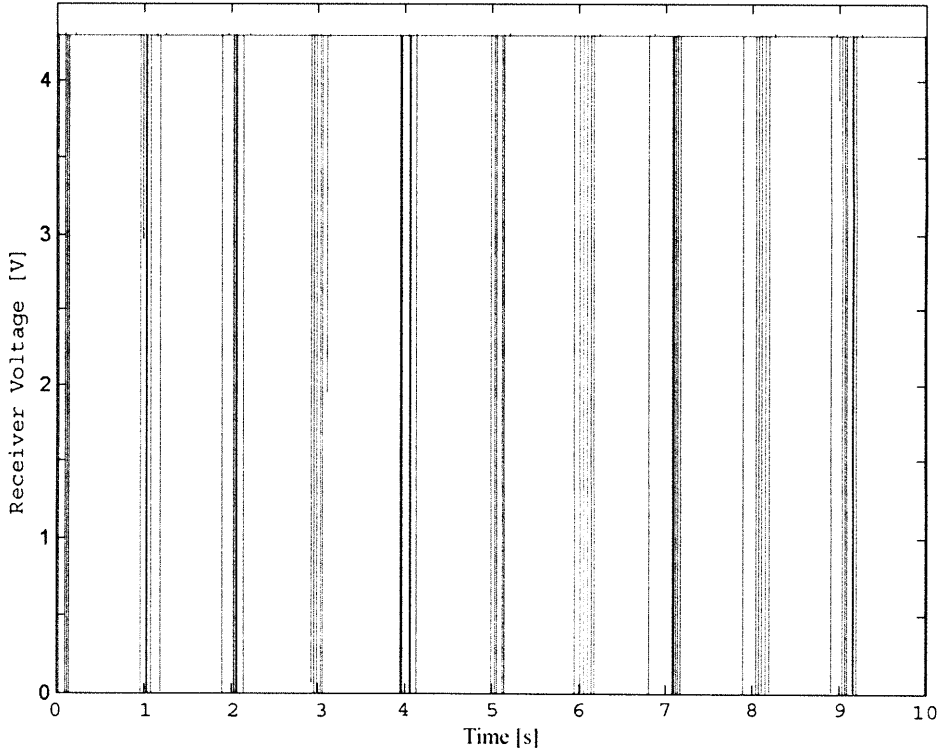

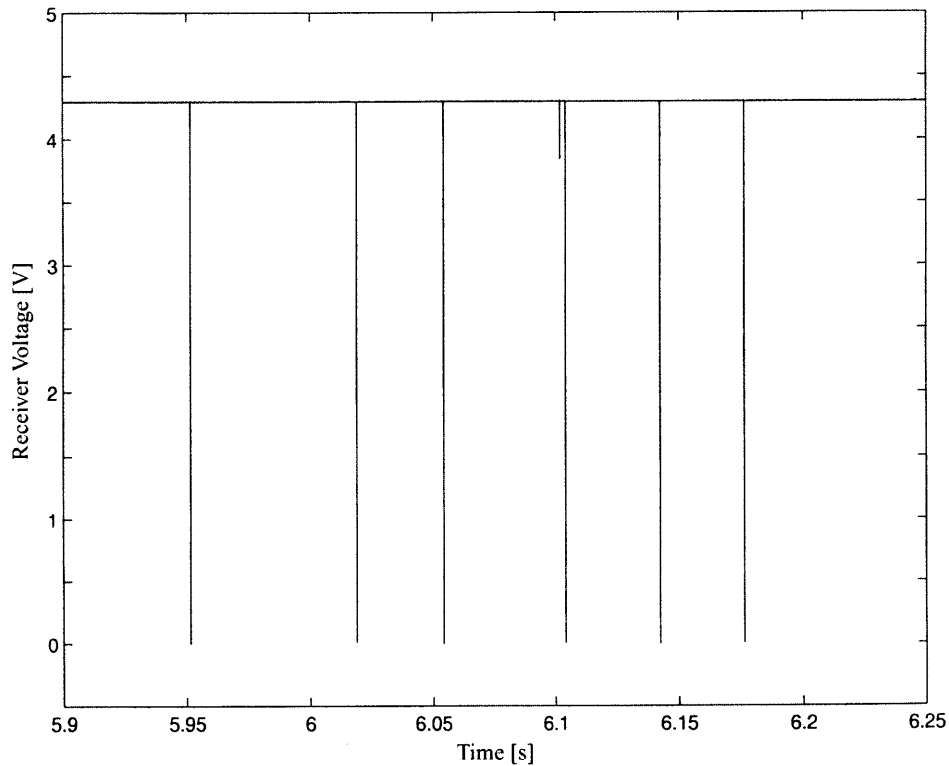Figure 6.1.1: Clapping Test: Receiver Voltage versus Time

Figure 6.1.2: Clapping Test: Expanded View of Single Clap

6.2 Sonic Tug of War

On September 30, 2001, at the ACM Ubiquitous Computing Conference in Atlanta, Georgia, the system [3] was demonstrated as a tool for large group gaming. For this demonstration, the game of *Sonic Tug of War* was developed. This game incorporated two base station receivers, each being placed at opposite sides of a room, approximately 10m apart. Revision four sensors, with weighted piezo film transducers and custom transmitters, were used. Groups of five or six people would locate themselves at each base station and use the sensor as an input device. In the center of the room was a circuit board that collected the received hits and integrated them over a two second period. As hits would arrive from one base station, the integration would be used to add to a common value. As hits arrived from the other base station, they would subtract from this common value. In this manner, the sum of the common value would represent which base station was detecting greater activity. This common value was then used to set the pitch of a tone generator. As hits increased on

63

one side of the room, the pitch would increase, and as hits would increase on the other side of the room, the pitch would decrease. If the pitch became either an octave higher or lower than the original pitch, this new pitch would lock in, signaling that the game was over, and who had won.

The system functioned well, and participants found it intuitive and fun to use. perhaps the most important result of this test was the ability to zone a space for data collection. Each side of the room was given a different functionality by placing a local base station as a node for interaction.

6.3   Talbot1

Since the system had not, to this point, been used for interactive dance, it was not known how people would interact with the system or what data parameters would be relevant. To answer these questions, and begin building effective musical mappings, sensor data was collected for a non-interactive dance event on November 12, 2001, held in Talbot Lounge in M.I.T.'s EAsT camPUS dormitory, Cambridge, Massachusetts. Fifteen participants, each holding a revision four sensor in a plastic tube, danced simultaneously for a half hour to a deejayed set of electronic dance music. Received data patterns were noted and compared to the music played at that time. The music had an average tempo of 154 BPM and varied musically with ambient sections, strongly rhythmic sections, and portions with syncopated rhythms. The received data gave insight into the way people respond to music, as detected by the sensors.

The sensor data was integrated over a ten second period to give the rate of pulse arrival. This integration, for the entire event, can be seen in Figure 6.3.1. A strong correlation between the rate of pulse arrival and the perceived energy level of the music was found. A listing of relevant musical events for the first ten minutes of the data can be found in Table 6.3.1, and are also annotated on Figure 6.3.1. At the start of the event, the music was ambient, and had no rhythm track. As a result, there was little motion by the dancers, and the rate of pulse arrival was subsequently low. When the rhythm track entered at 0:52, the
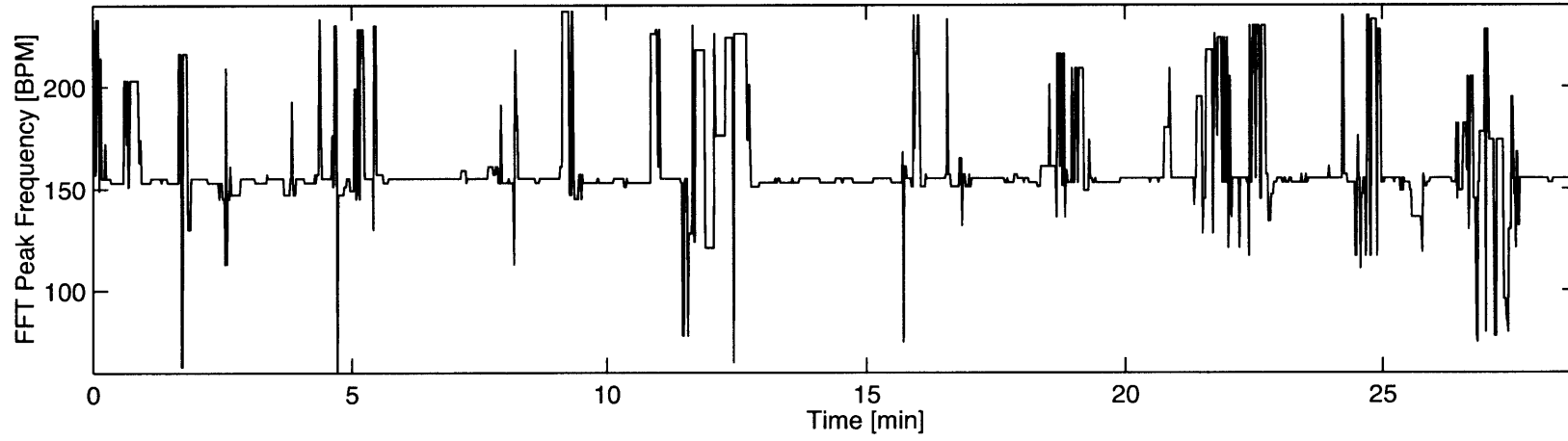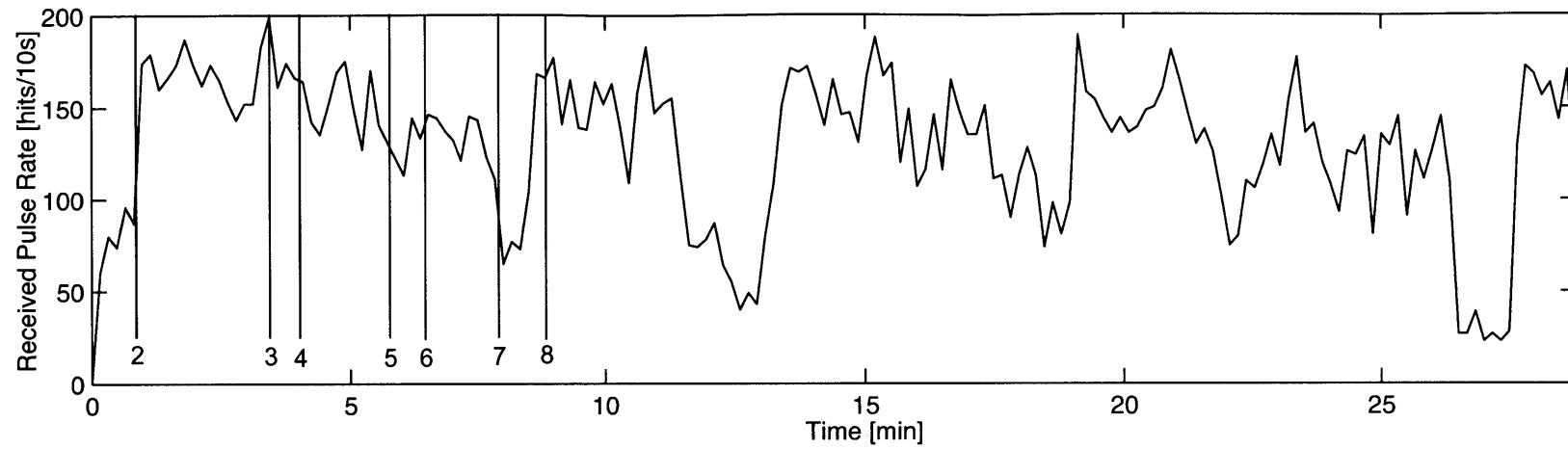
Figure 6.3.1: *Talbot1*: Rate of Pulse Arrival and Peak FFT frequency versus Time

rate of pulse arrival increased as the dancers began to respond to the music. This pattern continued throughout the event. In cases where the music had no rhythm track, or had a simple rhythm track and no melody, the rate of pulse arrival was one third that of sections with both a strong beat and melody.

Label	Time [min]	Musical Event
1	0:00	Start of event, music is ambient
2	0:52	Rhythm track start
3	3:38	Short crescendo followed by rhythm track end
4	4:03	Rhythm track start
5	5:45	Rhythm track end
6	6:25	Long build followed by rhythm track start
7	7:54	All voices end, except for rhythm track
8	8:56	Long build followed by all voices starting

Table 6.3.1: *Talbot1*: Relevant Musical Changes

A ten second sample of the received data signal can be seen in Figure 6.3.2. Although it appears to have very little rhythm information, an FFT of the received signal correlated strongly to the average tempo of the music, as shown in Figure 6.3.3. An FFT of the data, taken in thirty-second increments, returned a peak frequency of 2.57 Hz (154 BPM), for most sections of the music. This peak frequency is plotted along with the activity level in Figure 6.3.1. It can be seen that the FFT could not detect a dominant frequency during periods of low activity. Despite this shortcoming, it is concluded that the system, in its most rudimentary applications, can detect both the activity level and dominant tempo of its users.
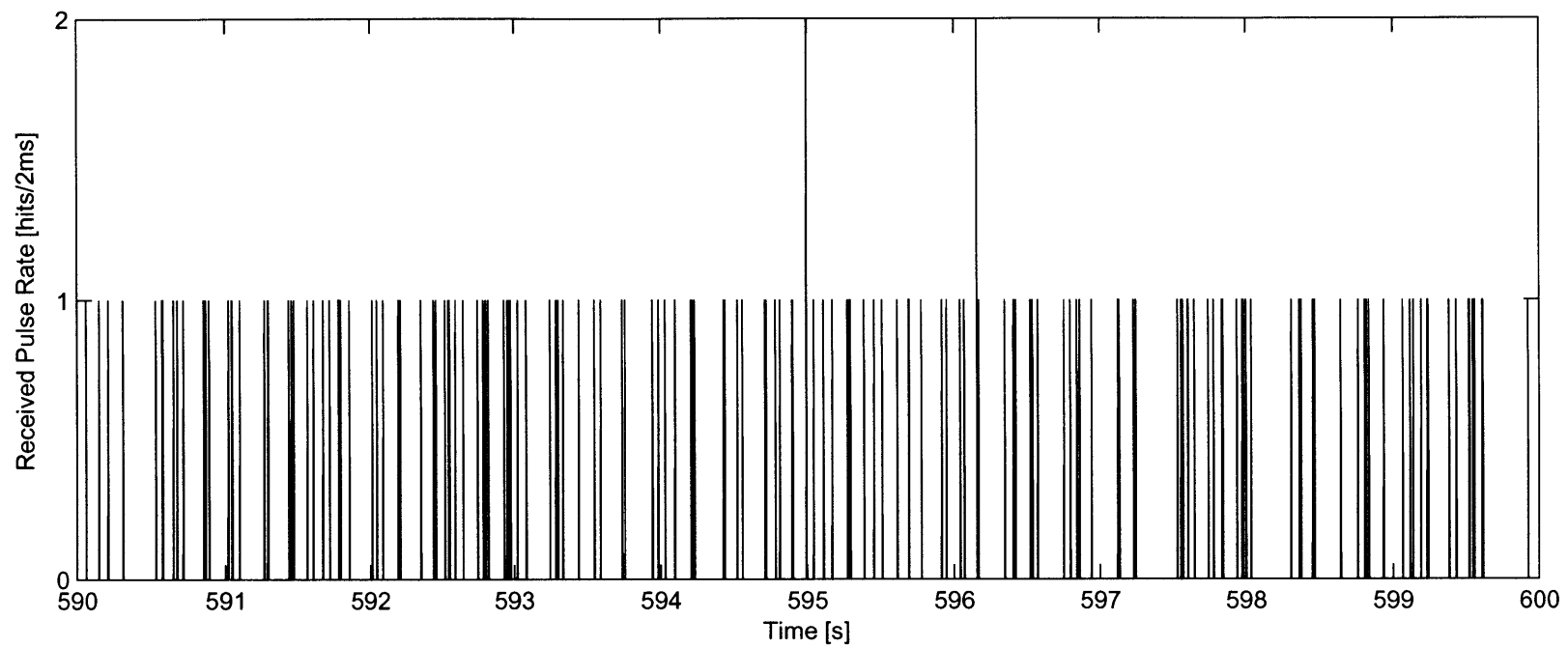
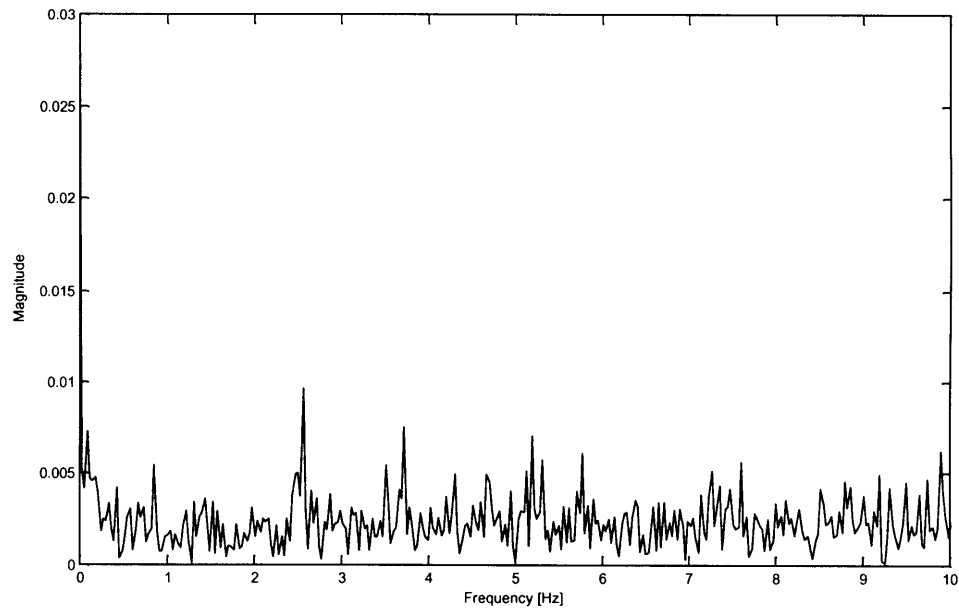Figure 6.3.2: *Talbot1*: Raw Sensor Data of 10 s Sample

Figure 6.3.3: *Talbot1*: FFT of 30 s Sample

6.4 Talbot2

With the knowledge gained from the previous test, the first musical mappings were written, and a fully interactive dance event was held in Talbot Lounge on February 18, 2002. Once again, fifteen participants, each holding a revision four sensor in a tube, danced to electronic dance music for a half hour. This time the music was not deejayed, but rather generated by the received data stream of the sensors. A Macintosh G4 computer was used to receive the MIDI data stream, store the event data, perform the FFT and other data processing algorithms, and send out a corresponding MIDI data stream to an E-mu Orbit V2 synthesizer, which generated the music.

The information from *Talbot1* was used to develop algorithms that detected features of the group behavior and generate matching musical pieces. The tempo of the music was set to one greater than the perceived group tempo, as determined by a smoothed output of the Beatgrabber object (see Section 4.4). The smoothing function would eliminate any values not within 10 BPM of the current tempo, and would pass any values within 10 BPM through a six step moving average low-pass filter. In this fashion, the tempo would maintain

68

its current value during low energy states when a dominant frequency could not be detected. A plot of both the raw and smoothed Beatgrabber data can be seen in Figure 6.4.1. The rate of pulse arrival was integrated over a ten second period and classified into one of five levels. The first represented the lowest activity state, for which there was no beat generated, and merely low drones played. The next level brought in a simple, soft beat, and changed the drone to a more complex, higher pitched voice. The third level increased the complexity of the beat and introduced an even higher drone. The fourth level added melody and a harder beat. The final level added another melody and began distorting the beats. In addition to these controls, each received hit triggered a subtle chime, and as the rate of arrival of hits increased, the chime was faded back into a low drone. If more than four hits were received within an 80 ms period, a louder and longer duration chime was triggered. Also, the rate of change of the received signal was used to modulate the cut-off frequency and volume of the melody voices.

A number of hypotheses was tested with these mappings. First, the effectiveness of giving each person a sonic response per hit was evaluated as a method of giving feedback to the functionality of the system, and as a tool to encourage synchronization of the participants. The participants quickly understood the causal nature of the interface as they moved their sensors, but the sound they were given had too long of a decay, and as a result, the combined sounds would merely blend into each other, rather than build to coherency. Despite this fact, at low energy levels, the participants would begin to synchronize, but the energy would soon build to a level where the hits would no longer sound. The 80 ms windowing of hits would then begin to produce sonic events, but these were also quickly drowned out by louder voices, which were still building. In this manner, only partial feedback was given, and, consequently, only partial synchronization would occur, although this test does show the tendency towards synchronization.

Second, the effect of giving positive feedback to increasing activity level was examined. As can be seen in the activity level plot of Figure 6.4.1, the activity would build to the highest level, and then collapse, usually to the lowest level, before building again. In this manner, the behavior followed the expected pattern as described in Section 4.3, implying
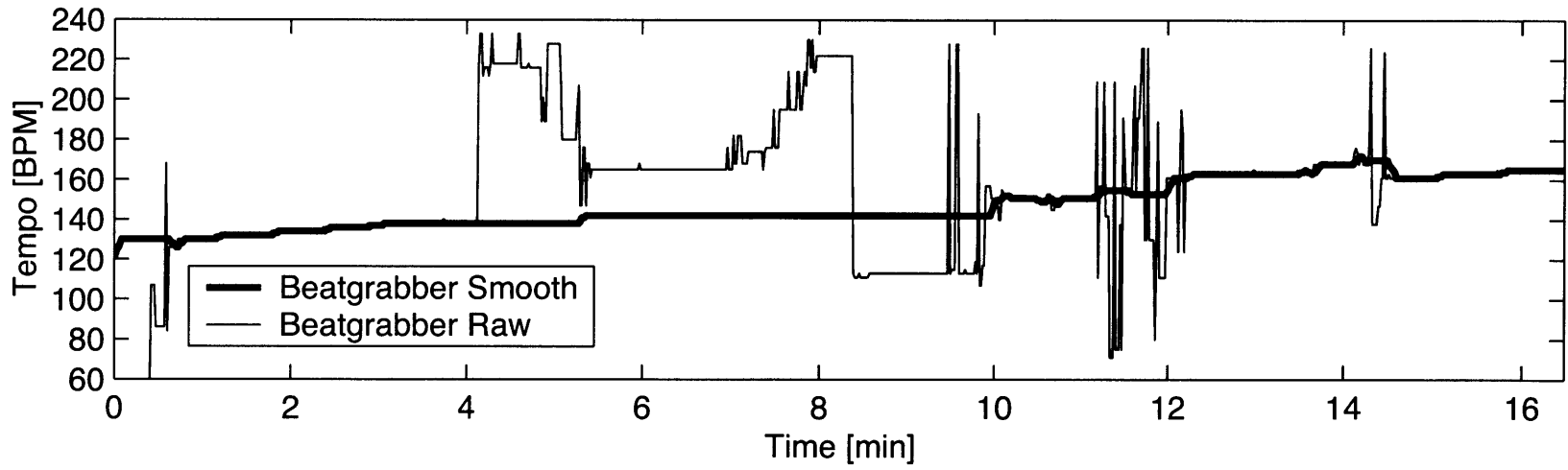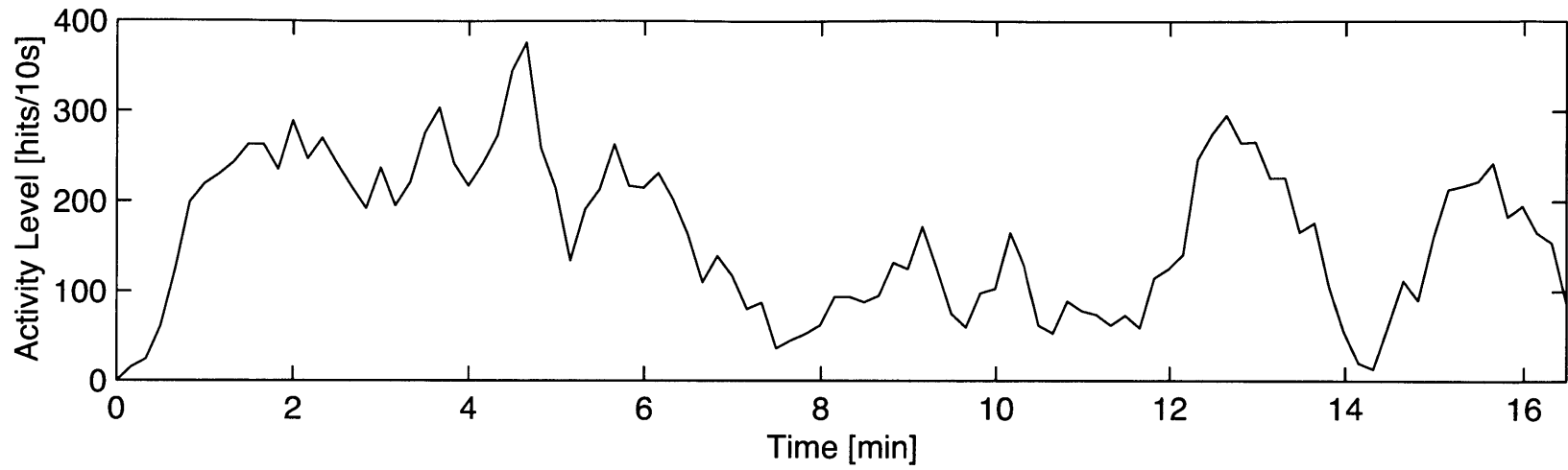
Figure 6.4.1: *Talbot2*: Activity Level and Peak FFT Frequency versus Time

that appropriate feedback was given. Finally, allowing the average motions of the group to set the tempo of the music showed that rhythmic coherency could be developed amongst the users, if given appropriate feedback as to their current state. Despite the tempo being set to the perceived rate plus one, it actually decreased in sections of the event, as shown in Figure 6.4.1, and did not merely increase indefinitely.

The received data from this fully interactive event varied significantly in a number of ways from the data for the non-interactive dance environment. The rate of pulse arrival was fifty-percent higher than for the non-interactive event, denoting a higher average activity level. Also, the occurrence of data clustering increased, and the received signal showed more rhythm information, as can be seen in a ten second sample shown in Figure 6.4.3. This increased synchronization of the dancers led to a higher magnitude of the FFT peak frequency, as shown in Figure 6.4.2. Finally, as shown in Figure 6.4.1, the tempo of the music varied from 120 BPM to 172 BPM, due to participant control.
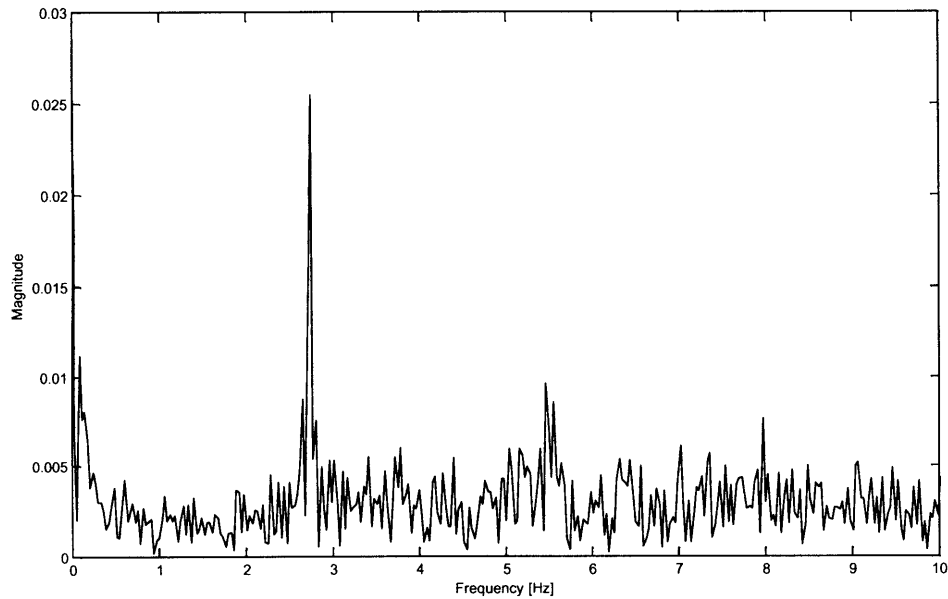

Figure 6.4.2: *Talbot2*: FFT of 30 s Sample

The dancers stated that they felt the music was responding to their motions, especially during the lower energy states when the more causal chimes could be heard, and a greater variation of music was occurring. They also felt as though they were controlling the tempo,
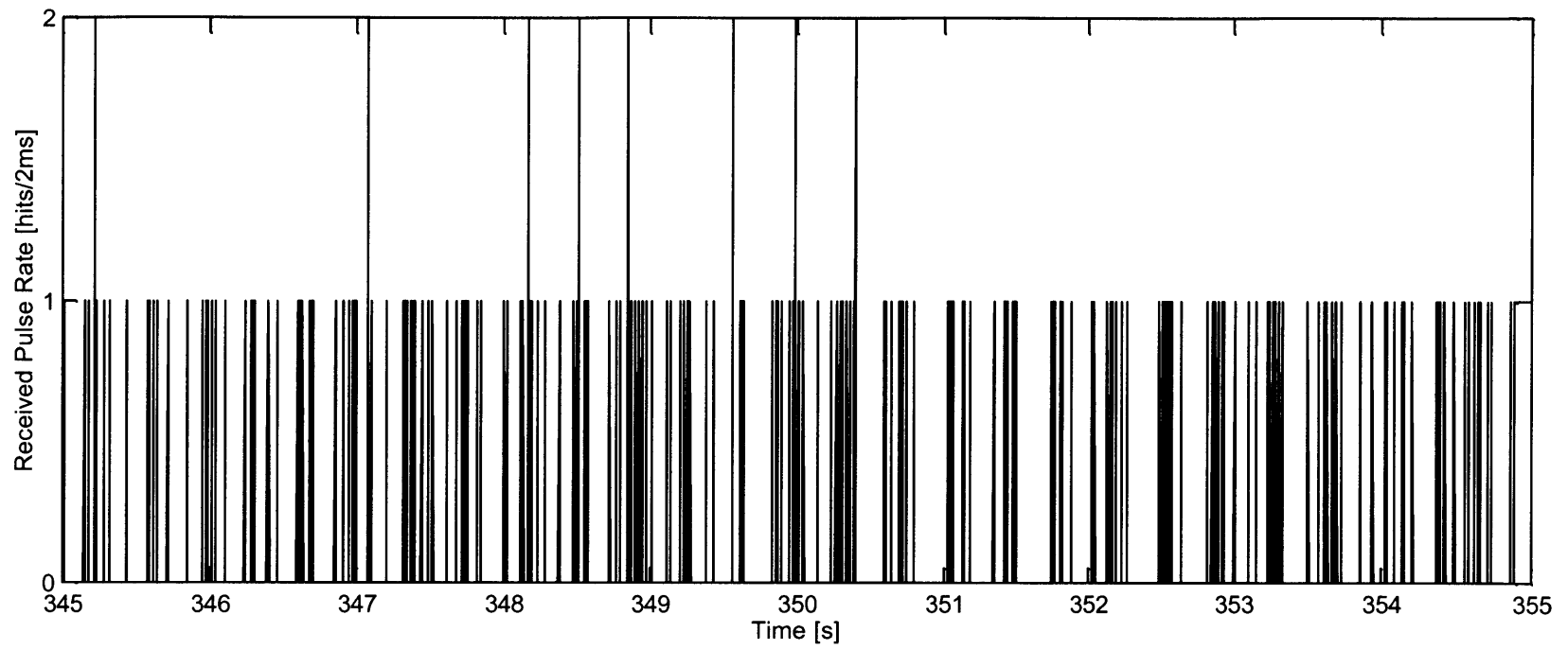
Figure 6.4.3: *Talbot2*: Raw Sensor Data of 10 s Sample

and in several instances, worked together to either raise or lower its level. Initially, they felt the music was engaging, but became disinterested after all of the various voices in the musical mappings had been exhausted. This is primarily due to the fact that the activity level was higher than in the non-interactive event upon which the mappings were modeled, so the music peaked much sooner, leaving no room for exploration.

6.5   CAMP

A music improvisation workshop for children was held at CAMP's Okawa Center outside of Kyoto, Japan on August 23 and 24, 2002. Three simultaneous workshops were conducted with 13–15 children in each group, the children ranging in age from 10–15 years old. The children used a combination of revision four and revision five sensors in tubes, contact microphones, voice microphones, and household materials to construct instruments. These instruments were then used to compose an improvised piece of music, which was performed separately by each of the three groups at the end of the workshop. Four separate base stations were given to each group, and each base station was tuned to a different frequency. Six to eight sensors were also given to each group, with anywhere from one to three sensors tuned to each frequency, and at least one sensor for each of the four frequencies. In this way, each sensor could be used as a trigger for various pre-recorded or MIDI sonic events.

Three groups practiced, in the same building, with their base stations and sensors on the first day. The performance was then given on the second day. Although 15–30 m apart, the base stations did not interfere with each other. The children quickly understood the interface, and each sensor triggered consistently enough to be useful as a musical instrument. This test of the system demonstrates its viability as a disposable musical controller, and points to the possibilities of identifying individual participants in a group event. By tuning the transmitters to different frequencies, a limited number of identified transmitters can cause different sonic events. Again, the intuitive nature of the interface, and the ability to zone a space into different nodes of interaction was shown.

73

6.6   Sidney1

Since the system had been tested with only fifteen dancers, a larger event was required to examine its viability as an interface for large crowds. An interactive dance event was therefore held at M.I.T.'s Sidney and Pacific dormitory in Cambridge, Massachusetts on September seventh, 2002. Approximately 200 revision five sensors in tubes (see Figure 6.6.1) were distributed to participants, with a glowstick taped to each sensor. Two receiver base stations were used, connected to channels A and B of the MIDI converter. Although multiple base stations were used, their data was not mapped to any sonic features, as the RF properties of the system are not well understood with respect to zoning in a relatively small space. Despite this fact, data was collected for each zone, to help build this understanding. The event was captured on both audio and video to allow for visual verification of participant activity level, and to correlate this activity with various data parameters and the music being played. Finally, at the end of the event, a questionnaire was distributed to assess the participants impressions of the experience.

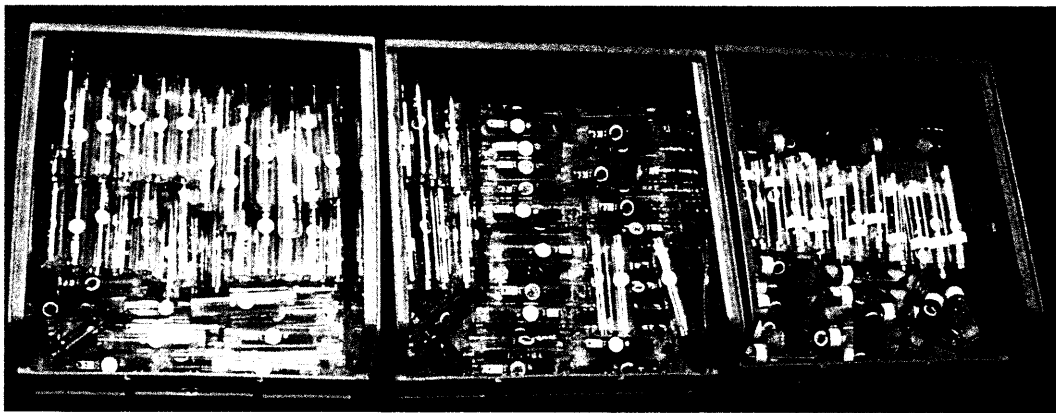Figure 6.6.1: *Sidney1*: Sensors Distributed at Event (sans glowstick)

Again, music was mapped to the data using a Macintosh G4 computer, and played out through MIDI to the synthesizers listed in Table 6.6.1. This time, a more complicated set of mappings was used to reduce the possibility of the music peaking too soon, as was seen in *Talbot2*. The music was mapped to the same five levels, but, within each level, the music

74

Manufacturer	Model
KORG	TR-Rack
E-mu	Audity 2000
E-mu	Planet Earth
E-mu	Orbit v2

Table 6.6.1: Synthesizers used at *Sidney1* and *Talbot3*

was broken into five sections: drones, melodies, high pitched beats, low pitched beats, and arpeggiated lines. For each section, there were from zero to three voices, and these voices could be indexed by changes in activity level. In some cases, a blank voice was inserted into a section, which would keep that section silent, increasing the period between repetitions of all other voices in that section. In this manner, the level changes must index through all possible combinations of sections before the music repeats, creating a more complex experience. Table 6.6.2 shows the number of voices allocated to each section, with blank voices indicated by the letter 'b'.

Energy Level	Drone	Beat Hi	Beat Lo	Arp.	Melody
1	2	0	1+1b	0	0
2	2	2+1b	2	0	2+1b
3	2	3	2	1+1b	3
4	2	3	2	2	2+1b
5	1+1b	3	3	3	0

Table 6.6.2: Number of Voices per Section for *Sidney1* and *Talbot3*

These transitions between levels could occur in a number of ways. The new track could either fade in over 16 measures, or cut in on the first beat of the next measure. In the same way, the old track could either fade out over 16 measures, or cut out on the first beat of the next measure. These transitions were fixed per voice and section, but distributed somewhat randomly amongst voices, creating an indeterminate transition state where the participants could either move the energy level up, or back down, before the next transition occurred. A delay was placed between level transitions to stabilize the audience response during these indeterminate states. After transitioning from one level to the next, the system would wait

for 32 measures before determining whether the activity level had changed. The exception to this rule were transitions into *level5*, which would occur instantaneously, regardless of the previous state.

The majority of the mapping tools described in Section 4.4 were employed for this event. The complexity of the music increased with activity level, as more voices were included. Lower energy level tracks were assigned voices that repeated on longer time scales, and had longer attack and decay. High energy tracks were higher in pitch, and had shorter attack and decay. As participants increased in activity, they were given more control over the voicing. At the lowest state, they could cause a chime to sound for every received hit, which would fade out as more hits arrived per unit time. A brighter chime was also employed if greater than 20% of the participants caused a hit within 80 ms of each other. At the next level, the drone was amplitude modulated by the one second integration value. Moving to the next level allowed the users to control the attack of the kick drum along with modulating the drone. As the activity increased to *level4*, the arpeggiated line's cut-off frequency was controlled by the one half second integration value. At *level5*, the dancers not only controlled cut-off frequency, but also resonance and note value of the arpeggiated line. They also triggered rhythm accents such as hand claps or high-hats if greater than average activity occurred at a particular step in the measure. In some cases, the voicing of the rhythm track was completely determined by the incoming data, as higher levels of activity would cause kick drums to sound, and lower levels would activate snare drums or cymbals. These mappings were intended to create a more energetic and chaotic experience as the activity level of the participants increased.

For the following reasons, a number of the mappings failed during the event. First, the computer that was running the mapping software malfunctioned less than two hours before the start of the event, and was unable to reload three of the four transition handling patches. For this reason, the remaining transition handling patch was remapped, directly before the event, to control all of the voices. This remapped patch, and other patches, had errors in them which prevented the majority of voices from activating. Finally, the Beatgrabber object had been modified to output a floating point value, rather than an integer, which

the smoothing function could not process, causing the tempo to remain constant.

As a result of these problems, the music did not respond to the participants' input for the majority of the event. A plot of the activity level and music tempo versus time for all three zones can be seen in Figure 6.6.2. From the audio and video taken at the event, these data can be correlated to circumstances occurring at that time. Table 6.6.3 lists relevant occurrences that are annotated on the activity plot. The beginning of the activity plot shows a low energy level, as people are entering the dance floor, and the music is extremely ambient. A slow increase follows, representing the crowd's attempt to shape the music, eventually just shaking their sensors wildly as the music does not respond. At four minutes into the event, the participants are told by the emcee that the music is controlled by their actions, encouraging the audience to increase their activity, in an attempt to force a response from the system. The energy level increases, and a new voice begins, capturing the audience's interest, until the voice fades out at seven and a half minutes. At ten and a half minutes into the event, the participants are once again encouraged to increase their activity. Again, this momentarily increases the perceived energy level, but this time a new voice does not enter, and the audience's activity fades between two and three minutes later.

Label	Time [min]	Musical Event
0	04:14	Audience encouraged to increase activity
1	05:06	New voice start
2	07:22	New voice end
3	10:38	Audience encouraged to increase activity
4	13:00	Audience ceases activity
5	15:43	First modifications to Beatgrabber, *level5* start
6	16:30	Tempo decrease
7	17:15	*level5* end
8	18:40	Beatgrabber fixed, *level5* start
9	20:00	Audience decreases tempo

Table 6.6.3: *Sidney1*: Relevant Musical Changes

New voices were not emerging, partially due to the fact that the perceived energy level is a function of tempo (see Section 4.4), and that the Beatgrabber object was not sending any tempo information. At 15:43 into the event, the first modifications to the Beatgrabber
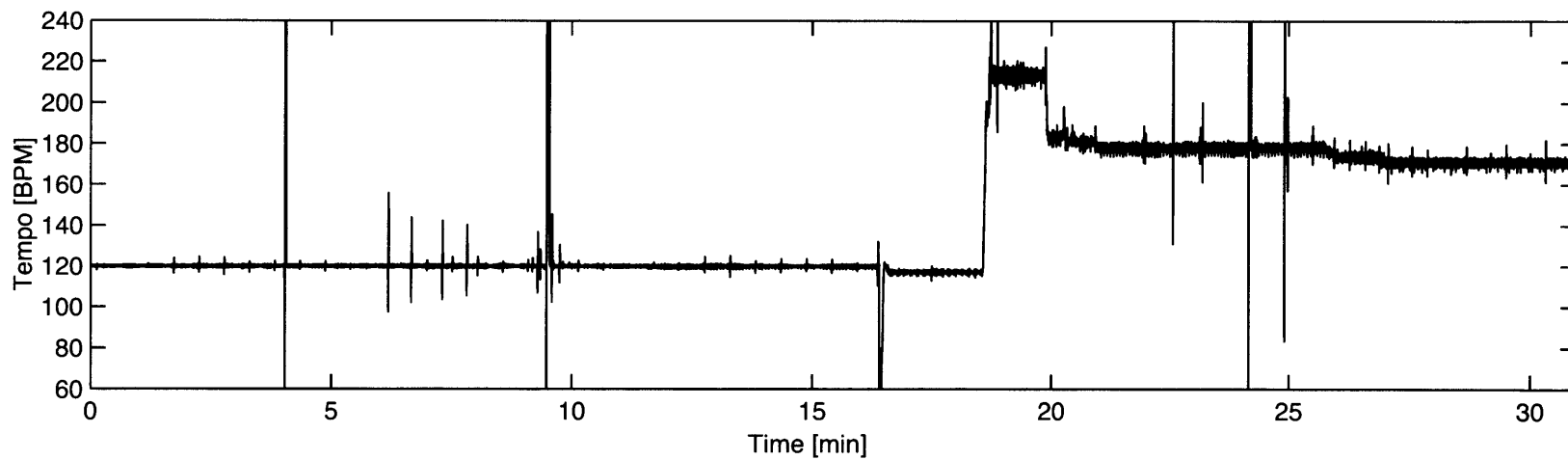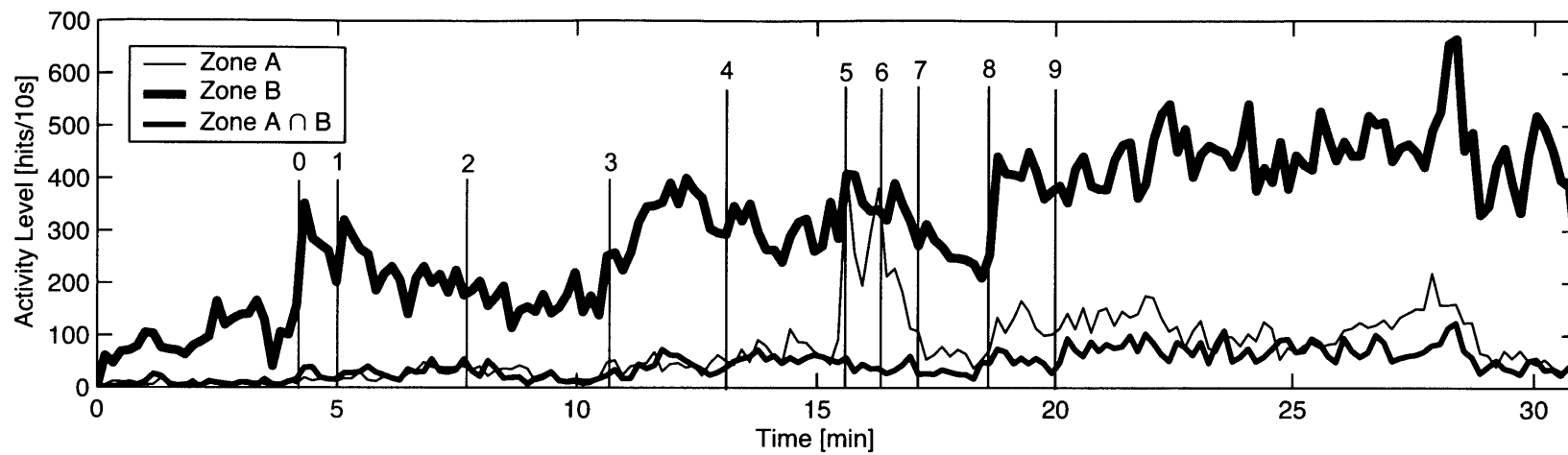
Figure 6.6.2: *Sidney1*: Activity Level and Tempo versus Time for Zones A, B, and A∩B

object were made, and the perceived energy level jumped quickly to *level5*. This was followed by a slight decrease in tempo as further modifications to the Beatgrabber were made, and the energy level slipped back down to *level4*. As the music began to change as a result of these modifications, the participants, once again, became interested, as can be seen in the activity plots. This interest soon faded when the music returned to *level4*, but was recaptured when the Beatgrabber object was finally operational.

As a result of the way the Beatgrabber object was fixed, at 18:40 the music tempo was set to 220 BPM. The energy level quickly rose to *level5*, causing new voices to begin playing, as the *level5* mappings were directly controlled by the participants, and not routed through the transition handling patch. The music began varying wildly, and the users responded with energetic motions. At 20 minutes into the event, the audience brought the tempo down to 180 BPM, and activity increased still further. This increasing of activity and decreasing of tempo continued until the end of the event at twenty eight minutes.

Throughout the entire event, the received data patterns were not only controlling the music, but also the lighting in the venue. The lighting instruments used, and their placement, are shown in Table 6.6.4. Of the data received from the MIDI converter and the data processing computer, three parameters were used to set the lighting levels. These parameters included the raw data signal containing the hit counts from both receiver A and receiver B, and a signal representing the position of each quarter note.

Position	Instrument	Color	Channel
Left Front	ETC Par NSP	Red	1
Left Front	ETC Par NSP	Blue	2
Left Front	ETC Par NSP	Green	3
Right Front	ETC Par NSP	Red	4
Right Front	ETC Par NSP	Blue	5
Right Front	ETC Par NSP	Green	6
Left Rear	ETC Par NSP	Magenta	7
Right Rear	ETC Par NSP	Lavender	8

Table 6.6.4: *Sidney1*: Lighting Color and Position

Each MIDI event triggered a change in lighting, dependent on the event's value. With each

event from the left receiver, the intensity level of the left rear instrument was changed. Likewise, on each event from the right receiver, the intensity level of the right rear instrument was changed. A static mapping between the intensity of the incoming signal and the intensity of the lighting instrument was adjusted at the beginning of the event, to allow the lighting to respond as dynamically as possible. The mapping used can be seen in Table 6.6.5.

Received Intensity (hit value)	Lighting Level (0–100%)
0	0
1	30
2	55
3	75
4	90
≥ 5	100

Table 6.6.5: *Sidney1*: Lighting Level versus Hit Value

The received values observed during the event varied between 0 and 8, with most of the activity between 0 and 3. There were periodic occurrences of 5 and 6, and 7 and 8 appeared rarely. The result was that a single transmitter pulse would momentarily light the instrument on the side closest to it, and multiple transmitters pulsing together could bring the light to full intensity.

On each quarter note, both the right front and left front instruments would randomly change color. This was done by setting one of channels 1, 2, or 3 (or 4, 5, or 6) to full intensity, depending on a pseudorandom number. At the same time, the remainder of the channels would be turned off, leaving only one color present. In this manner, the lighting dynamically changed with the dancers motions, providing them with immediate feedback as to their current state.

The participants' responses, as to the effectiveness of the system, were compiled using a questionnaire. This questionnaire can be seen in Figure 6.6.3, and the full results can be found in Appendix D. Relevant responses are shown in Figures 6.6.4, 6.6.5, and 6.6.6. From these data, it can be seen that the majority of people did not feel that the music and lighting

questionnaire

On a scale from 1 to 5, please state
whether you agree or disagree with
the following statements: disagree   neutral   agree

I am an avid dancer....................1   2   3   4   5
I often listen to electronic music...........1   2   3   4   5
I enjoyed this interactive dance
experience............................1   2   3   4   5
I spent my time actively dancing, rather
than just standing around................1   2   3   4   5
I danced by myself, rather than with
others................................1   2   3   4   5
I felt in control of the experience..........1   2   3   4   5
The music and lighting responded well to
my motions............................1   2   3   4   5
I would have liked to have more input....1   2   3   4   5
I felt that the sensors were well suited to
the way I dance........................1   2   3   4   5
I would have preferred a different interface
(please elaborate below)................1   2   3   4   5
The music was sufficiently varied to keep
the experience interesting................1   2   3   4   5
A different style of music would have
made the experience more enjoyable
(please elaborate below)................1   2   3   4   5

thanks.

Figure 6.6.3: *Sidney1*: Questionnaire

were very responsive, but enjoyed the experience anyways, and spent the majority of their time dancing.
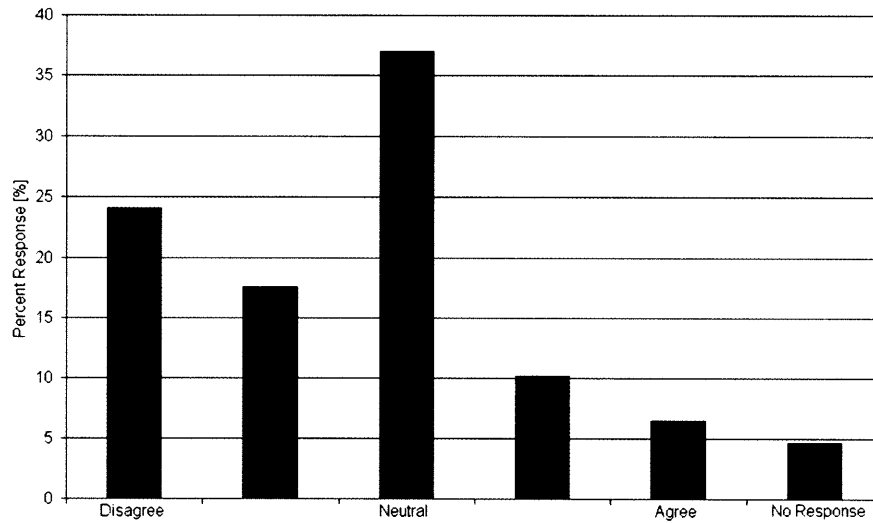

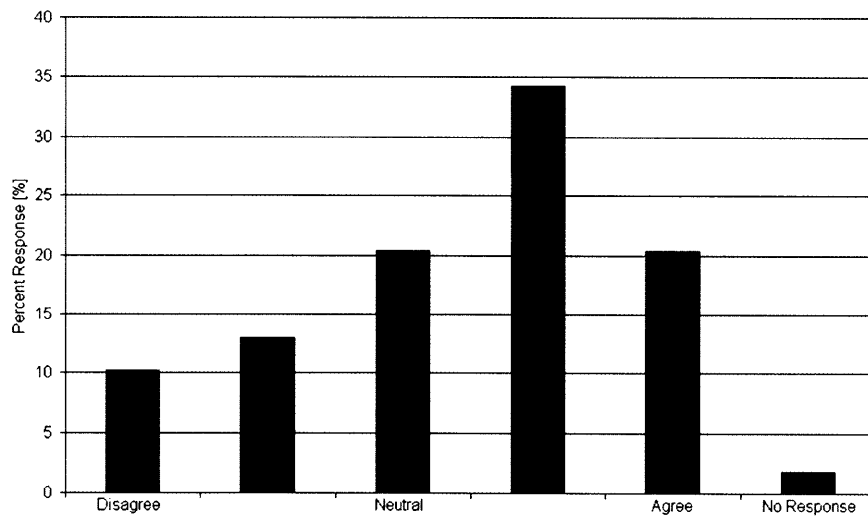Figure 6.6.4: *Sidney1*: Questionnaire Results for "The music and lighting responded well to my motions"

Figure 6.6.5: *Sidney1*: Questionnaire Results for "I enjoyed this interactive dance experience"
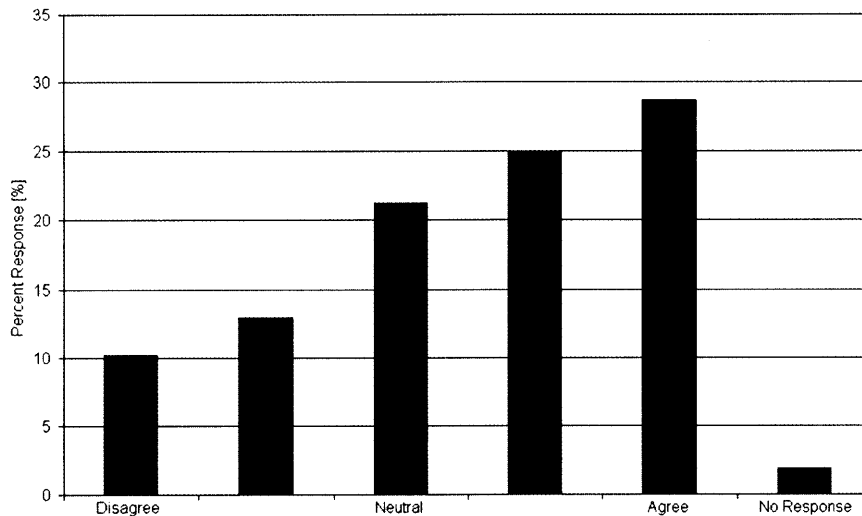
Figure 6.6.6: *Sidney1*: Questionnaire Results for "I spent my time actively dancing, rather than just standing around"

The data received from the location information was inconclusive as to the nature of the RF properties of zoning. As can be seen in Figure 6.6.1, the majority of activity detected by the system was in Zone B, with about one half as much activity in Zone A, and one quarter as much activity in Zone A∩B. This could be the result of a number of factors, each of which can not be decoupled from the others. First, the receiver used for Zone B was the master receiver, which has the greatest sensitivity, since all transmitters were tuned to its center frequency. Secondly, the Zone B receiver was adjacent to an outside wall with a patio on the other side. This patio area had many people present, some with sensors, all of whom were not dancing. It was also the area from which sensors were being distributed. These factors may have contributed to the increase in activity seen at Zone B, but are less likely than other causes due to the attenuation of the RF transmissions as they come in contact with the wall. This area was also directly coincident with the entry to the venue, which was crowded with people, increasing the total number of participants in Zone B, in comparison to Zone A. There was also a metal column near the Zone A receiver, possibly shadowing its antenna, and reducing the total number of hits received. Since the receivers were attached

83

to the ceiling, at a height of 3.2 m, the RF transmissions were only required to pass through a section of the crowd to reach the antenna. For this reason, it is assumed that the master receiver had an effective range of 7 m, one half of its non-crowded reception distance. The Zone A receiver, therefore, would only have a 5 m range, as secondary receivers showed a 30% decrease in maximum reception distance, as discussed in Section 5.1. These radii are shown in Figure 6.6.7, as are the receiver positions and other relevant data.
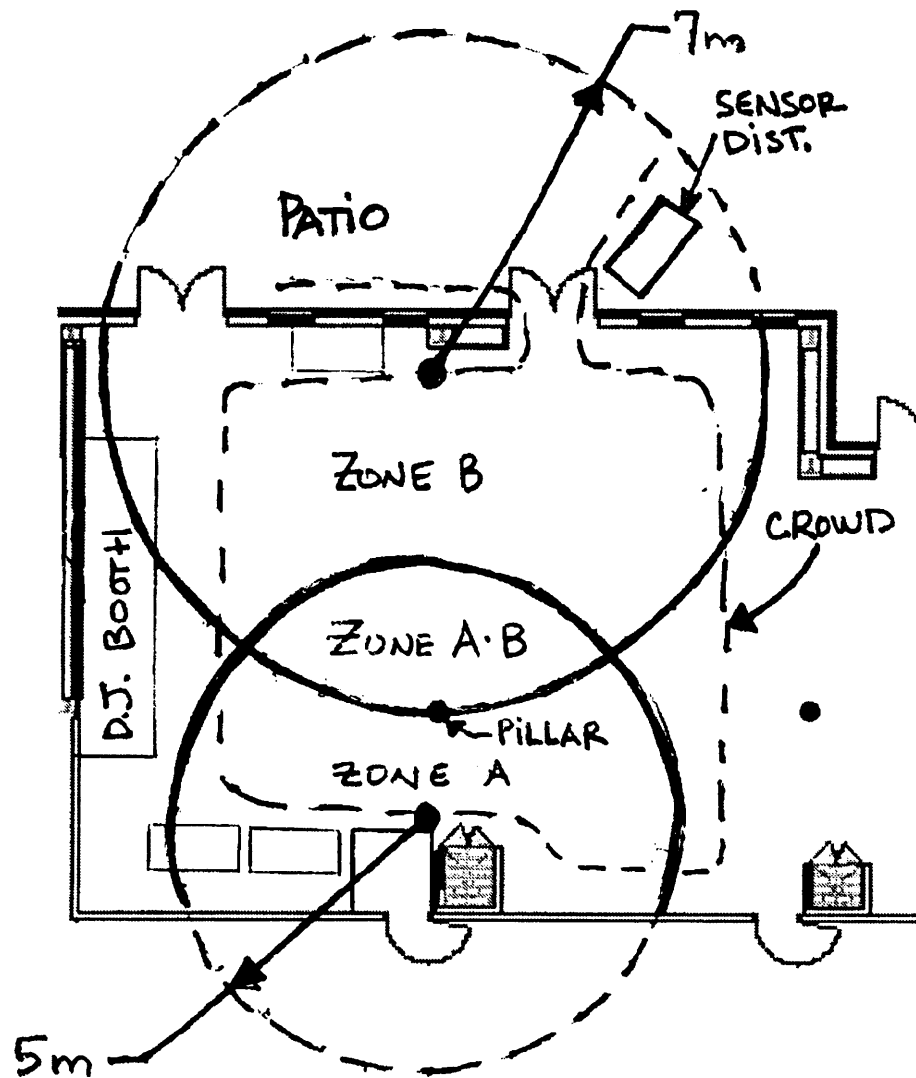

Figure 6.6.7: *Sidney1*: Receiver Positions and Reception Radii

These assumptions, although not quantitative, do account for the relative activity levels seen by the receivers. They do not, however, account for the low levels of activity seen by

84

all receivers. Assuming that all 200 sensors distributed were present on the dance floor, this would give a maximum activity level of 4,000 hits per ten second period. The actual activity level, summed over all receivers, only attained a maximum of 1,000 hits per ten second period. This is significantly lower than expected, and might be due to people holding multiple sensors, and not activating both at once. It is more probable, though, that the actual crowd never exceeded 100 people, and that the majority of these people were either shielded from the receiver, or stayed at a low activity level. The net effect, though, was an equivalent crowd size, as perceived by the system, of only 50 people.

6.7 Talbot3

Due to system malfunctions, the data collected at *Sidney1* did not accurately reflect the effectiveness of the mappings written for that event. For this reason, the system was promptly fixed, and another event was held in Talbot lounge on September 12, 2002. Although the system's functionality with a large crowd would not be assessed, the effectiveness of the new mappings could be evaluated. The mappings written for *Sidney1* were again used, this time with only one receiver, and 25 to 30 people dancing, each holding a tube containing a revision five sensor. The event audio and video was recorded, and the same questionnaire as used at *Sidney1* was distributed after the event.

The increase in the amount of music written for this event, in comparison to *Talbot2*, allowed for a greater degree of exploration by the participants. The ability to fade in and out of tracks also added to the complexity. Perhaps the only potential disadvantage of the new mappings involved the modifications made to the Beatgrabber object. The Beatgrabber object used in *Talbot2* had taken a rolling FFT over the past 30 seconds, with a 7 ms windowing function. This returned a very accurate value for the average tempo, with 2 BPM resolution, but had a 15 s delay, so that the current value did not reflect the current activity. The *Talbot3* and *Sidney1* Beatgrabber, in an attempt to improve the responsiveness of the tempo, had a windowing period of 16.384 ms, with a total sample time of 16.7 seconds. This had the effect of decreasing the delay to eight seconds, but, correspondingly, decreased the
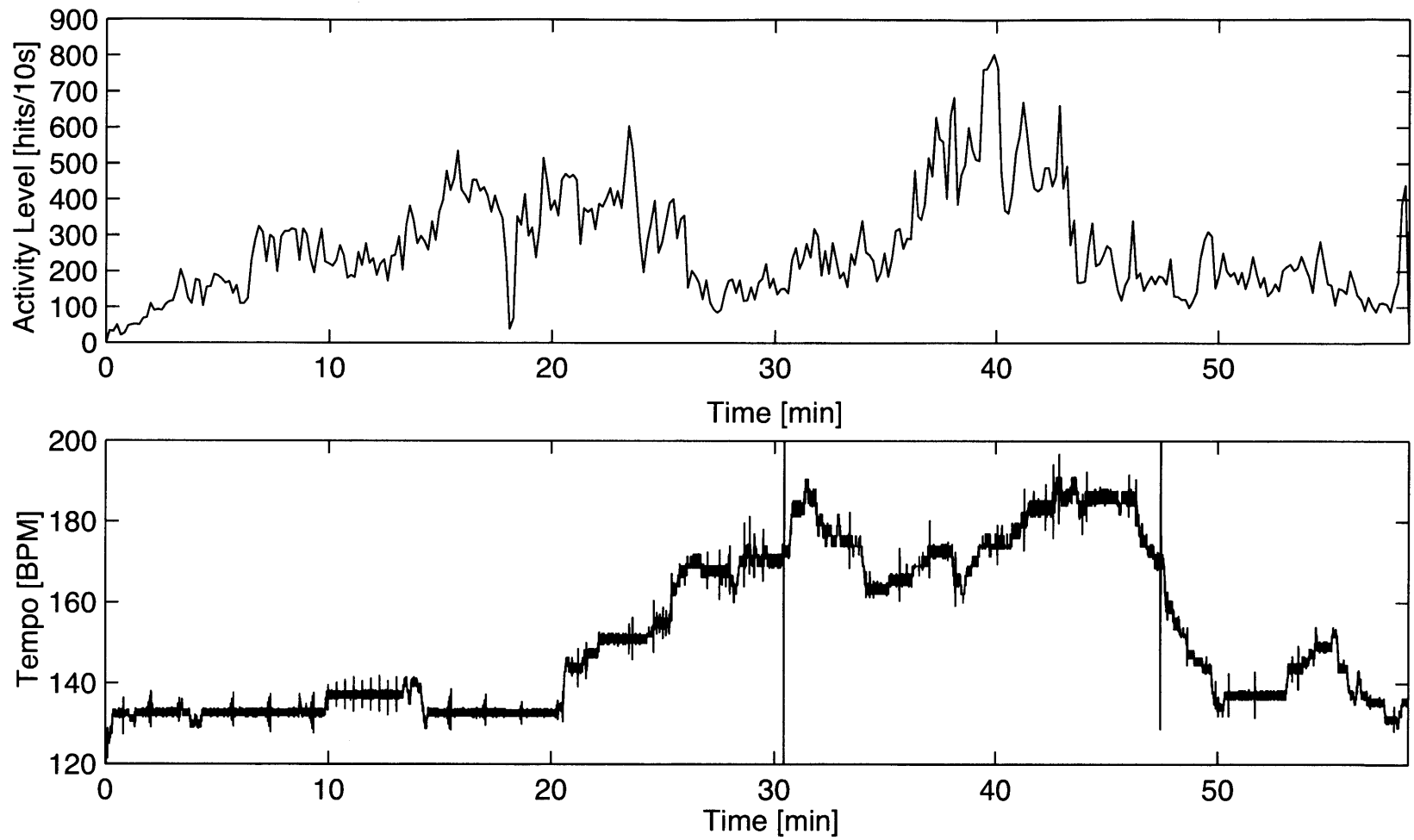
85

Figure 6.7.1: *Talbot3*: Activity Level and Tempo versus Time

resolution to 4 BPM. The result of this change can be seen in Figure 6.7.1, as the tempo varied insignificantly for the first 23 minutes of the event. This is due to the participants not being able to change the tempo significantly enough to overcome the 4 BPM binning. Instead, they remained at their current tempo, until the tempo setting function was varied to add a value of two to the current tempo, instead of just one. This was done at 23 minutes into the event, and the effect can be easily seen in Figure 6.7.1. The tempo began to ramp up quickly, as it led the dancers by a full 2 BPM. The function was then reset to plus one, and the tempo plateaued.

Throughout the remainder of the event, the tempo setting function was varied from minus one, to plus one, to plus two. The net effects of these changes can be seen in Figure 6.7.1. At every point where the function was changed to plus two, the tempo increased. For every point where the function was set to plus one, the tempo plateaued. And, for every point where the function was set to minus one, the tempo decreased. It seems expected that the tempo would increase when set to plus two, and reasonable that it would plateau, when set to plus one, especially given the 4 BPM Beatgrabber resolution. The interesting aspect of these results is how the tempo decreased when set to minus one. If the 4 BPM binning is keeping the participants from transitioning up in tempo with the function is set to plus one, then it would be expected that it would also keep the tempo from decreasing when the function is set to minus one, especially considering that it has been shown [12] that humans tend to increase their tempo when interacting as a group.

It is hypothesized, that people dancing to music greater than their desired tempo have an incentive to slow down their rhythms. The further they are away from this desired tempo, the greater this incentive becomes. Since the tempo was continuing to increase, seemingly out of control of the participants, due to the 4 BPM Beatgrabber resolution, it reached a level (190 BPM) that was completely disparate from their desired tempo. It is at this point that the desire to reduce the tempo overcame the 4 BPM tempo gap. A partial confirmation of this theory can be seen in the activity level plot of Figure 6.7.1. As the tempo increased, between 23 minutes and 33 minutes into the event, the activity level decreased. And, when the tempo decreased, between 33 minutes and 36 minutes into the event, the activity level

of the music increased. These increases and decreases in activity level tend to represent the participants' relative approval of the music. When the music slowed down, becoming more commensurate with their desired tempo, they would once again become engaged and raise the activity level.

The users' frustration with the high tempo eventually resulted in a complete disinterest with the system, as can be seen at 43 minutes into the event on Figure 6.7.1. At this point, the music collapsed and the tempo quickly decreased, as ambient sections began to play. Once the tempo was at a more comfortable level, the activity level stabilized. This hypothesis, of increasing incentives with increasing disparities between desired level and actual level, is corroborated by the tempo characteristics of *Sidney1* and *Talbot2*. As shown in Figure 6.6.2, when tempo control was given to the participants of *Sidney1* at 18:40 into the event, the users were able to lower the tempo from 220 BPM to 180 BPM, and activity levels increased. For the remainder of the event, the tempo continued to decrease, despite the fact that the tempo was set to plus one the entire time. This tempo plateaued at 175 BPM, a similar level at which participants of *Talbot2* (See Figure 6.4.1) and *Talbot3* became disinterested enough in the tempo to cause it to decrease. Finally, it is also shown by these tempo tests, that the feedback required to allow users to accurately set their tempo is in steps less than 4 BPM.

The results of *Talbot3*, in terms of activity level, are very similar to *Talbot2*, with one very important difference. They increase and decrease in energy, from *level1* to *level5*, corroborating the finding of *Talbot2*, that users will respond to positive feedback with increased activity, until they reach an unstable state, at which time they will decrease, only to build again. But, the activity levels of *Talbot3* do not vary as quickly, or as much as the activity levels of *Talbot2*. The *Talbot2* data, as seen in Figure 6.4.1, shows the activity levels varying on two to six minute time scales, and spending most of their time in either the highest or lowest energy state. The *Talbot3* activity levels, as shown in Figure 6.7.1, vary on 15 minute time scales, with only one sharp dip happening, due to a synthesizer which was momentarily set wrong, and playing a very dissonant voice. The activity also stays in the middle energy levels for much longer, with the average time spent per level being five minutes. This is

primarily due to the 32 measure transition delay used to bring in new tracks, keeping the participants from fully receiving feedback until the system was certain that a transition was desired. The increased voicing control of the *Talbot3* mappings resulted in more variations within a level, as participants shaped the music, no longer requiring a change in level for a change in the musical experience.

Perhaps the most encouraging result of this event, is the fact that it lasted for over two hours. The majority of people who had attended danced consistently for the first one and a half hours of the event, with the last half hour being mostly experimentation by those still dancing. As shown in Figure 6.7.2 and Figure 6.7.3, the results of the questionnaire indicate a marked increase, over *Sidney1* responses, in the percentage of people who enjoyed the event and felt that the music had responded well to their motions. It is more interesting, though, to compare this event to *Talbot2*, since the system was not fully functional at *Sidney1*. At *Talbot2*, the users were quickly disengaged as the music began to repeat, and no longer varied with their activity, ending the event within a half hour. From inspection of the audio and video recordings of *Talbot3*, many grouped patterns and repeated rhythms developed, generated solely by the users' input. These results seem to imply that, given the appropriate feedback and constraints, people will be able to assume considerable control over their musical environment, and tend to create a coherent and pleasing outcome.
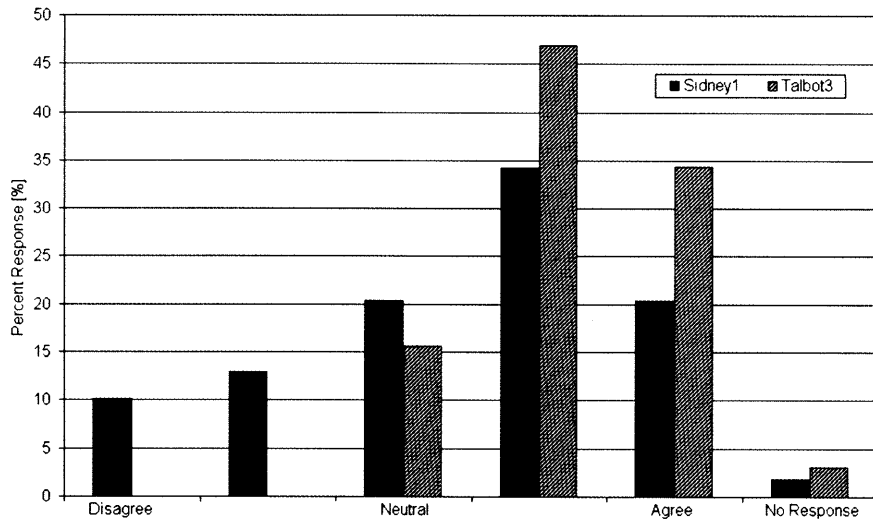
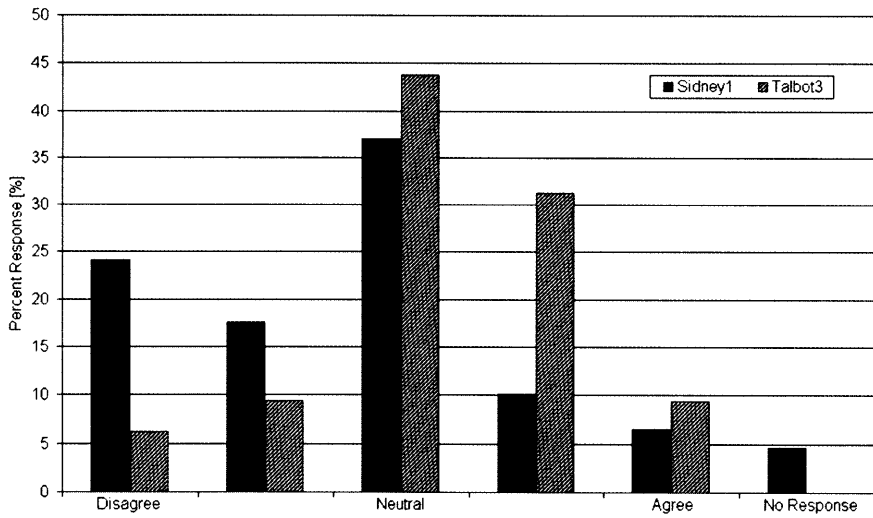Figure 6.7.2: *Talbot3*: Questionnaire Results for "I enjoyed this interactive dance experience"

Figure 6.7.3: *Talbot3*: Questionnaire Results for "The music and lighting responded well to my motions"

Chapter 7

Conclusions

7.1 Summary

In order to investigate the nature of musical interaction with large groups of dancers, a set of single bit, wireless motion sensors were developed. They employ a weighted piezo film to detect when the sensor exceeds an approximately 2.5 G acceleration threshold, transmitting a $50\,\mu s$ pulse of 300 MHz RF energy at each transition. This RF pulse is short enough to allow hundreds of non-synchronized transmissions to be received by a base station with a low likelihood of collision, and weak enough to limit the transmission radius (3–10 m), so that multiple base stations can be used to zone a venue. These sensors are small ($6\,cm \times 1.4\,cm \times 1\,cm$), inexpensive (5$US for parts in quantities of 100), and have a long battery life of over one month with continuous usage, making them viable as a disposable user interface that can be given away at large dance events. In this manner, the rhythm and activity level can be summed over all participants, per zone, and used as a real-time controller to effect music and lighting changes, allowing the dancers to control various aspects of the experience.

Five hundred of these sensors were fabricated, and multiple interactive and non-interactive dance events, varying in size from 15 to 200 participants, were held to test both the functionality of the system and the nature of the human response to musical feedback. It was

hypothesized, that groups of people, if given appropriate feedback as to their current state, would synchronize to produce a coherent and pleasing outcome. It was further stated that this outcome would then increase in energy until an unstable and chaotic state was reached, at which point the energy would return to a ground state and build again, repeating this process, for as long as the activity continued. Musical mappings were then written that were intended to convey to users their current state, both in terms of activity level and tempo.

Results of these events suggest that the previously stated hypothesis is correct, and that the musical mappings that were developed gave appropriate feedback. The nature of the human response was shown to follow the described pattern when feedback was given, as shown by the tests at *Talbot2* and *Talbot3*. In cases where either little or no feedback was given, as in the test at *Sidney1*, the activity level and tempo did not vary significantly. Musical mappings with negative feedback were not tested, except in the most rudimentary ways by the tempo settings at *Talbot3* and *Sidney1*, so it can not be stated conclusively that the human response follows the indicated pattern for only positive forms of feedback. But, if it can be assumed that the schooling behavior of the human clapping response carries over to other forms of sonic feedback, then the mappings developed can be considered appropriate positive feedback, and the musical definitions used in their derivation can be considered correct.

These results also indicate that the sensor is an effective and intuitive user interface for this application. The users required no training on the system, and merely responded naturally to the music. As shown by the *Sonic Tug of War* and Camp tests, the ability to effectively zone a venue for interaction is possible, but as the results from *Sindey1* reveal, the parameters controlling receiver reception distance are not well understood, and further work is required to make the system more robust. Despite the lack of zoning data, the musical mappings that were written displayed the system's ability to give a group of dancers causal control over the music to which they are dancing, changing, in real-time, aspects such as style, tempo, voicing, and filter parameters.

Overall, participant response to the system was positive. The majority of attendees at

92

Talbot3 enjoyed the experience, and spent most of their time dancing. Although quite a few of the attendees felt the music responded well to their motions, the majority desired more control over the experience. This poses a difficult question, which is still left unanswered by this thesis. How can a large group of anonymous individuals be given appropriate feedback, such that each individual has a sense of close control over the central interaction, while ensuring adequate structure so that all participants find the interaction pleasing? Indeed, for large groups, the possibilities may be quite limited.

7.2   Future Work

Although the system has been proven to function for relatively small groups of people, there are many areas where improvements can be made. The hardware is currently not robust enough to withstand repeated public usage, and the responsiveness of the mappings have left users with a desire for more control. Also, through the course of this work, methods of expanding the functionality of the system have become apparent, as detailed below.

7.2.1   Sensor improvements

The sensor currently performs to specification, except for the cost, which would decrease significantly at higher quantities. To help reduce this cost, the motion sensor could be replaced with an inexpensive mechanical tamper switch, similar to the ones used in the first revisions (see Section 3.1). Parts which perform this operation are currently in low cost products, such as toys, and take up very little space ($1\,cm \times 0.4\,cm \times 0.4\,cm$). They also have the advantage of directionally independent sensitivity. The piezo vibration sensor, although having directional sensitivity, is superior to a vibration switch, in that it can detect the amplitude of the acceleration it experiences. With the addition of a micropower op-amp, or FET, these data could be collected and used to modulate the transmission pulse width, giving an extra parameter for musical mappings. This added input stage would also have the benefit of standardizing the piezo sensitivity, as it presents a very high

input impedance to the piezo, unlike the variable impedance input stages of commercial CMOS logic. The pulse width modulation technique could also be employed to give each sensor one of multiple identities (IDs). The limit to this application would be the power consumption and increased likelihood of collision involved with long transmission times, along with the pulse width tolerance, as it is currently set with a simple resistor-capacitor network. In general, a small set of various IDs is more easily assigned by giving each ID its own frequency, as was demonstrated at CAMP (see Section 6.5).

To make the system viable for larger groups, a number of modifications should be made. First, it should be verified that the device meets the Federal Communications Commission's requirements for wireless devices. Second, the sensors need a more sensitive frequency tuning capacitor, and a more robust way of verifying that this frequency is set. For large production runs, a laser trimmed capacitor with feedback from a wideband receiver might be employed. Third, the pulse width should become shorter, to decrease the energy required per transmission, and to decrease the probability of collisions when hundreds of transmissions are being received at the same time. The sensor would also need to reduce in size, and preferably be attached to the body via a bracelet, pendant, or shoe clip. In this way, the user would not be encumbered by holding the device, which slips easily out of the hand while dancing. Finally, new sensing modalities, such as pressure, angular acceleration, or temperature might be incorporated, to allow for different types of input, for different types of events.

7.2.2 Receiver improvements

Although tests have been done that zone the transmitter locations via amplitude discrimination between multiple base stations, this was not robust enough to act as a reliable controller. Since few base stations are required to cover a standard dance venue, they could easily become more sophisticated without significant increase to the total cost of the system. For this reason, it would be advantageous to increase the base station's sensitivity and selectivity, to improve their ability to zone the incoming data. A more reliable method of tuning each receiver to a master frequency would also help in this regard. Ultimately, the

base stations might produce significantly better zoning results through time-difference-of-arrival (TDOA) [25] of sharp, ultra-wideband pulses. The sensors could be inexpensively modified to create this sort of transmission, enabling the system to become aware of the location of each transmission, perhaps tracking a user's motion throughout the venue.

7.2.3 Interaction improvements

Understanding the human response to the system, and converting this understanding into effective musical mappings, represents the largest area of opportunity for improvement. The system needs to be tested on much larger audiences, for both interactive and non-interactive venues. Through these data, mappings that either work with, or against the audience can be experimented with. The nature of the human response to negative feedback was also not rigorously tested, and could provide interesting results. Since the testing conducted gave limited control to the dancers, it is still unclear exactly how to give the users more real-time control, without having the experience degenerate into chaos. More music, in general, needs to be written for the system, since the users are capable of directing the experience in any number of directions. For each possible direction, there needs to exist a selection of mappings and tracks, demanding an order of magnitude more material than would be required for any non-interactive event.

A fully known experience is not generally considered an interesting one. Giving the system the ability to adapt and change to constantly changing states of the audience would create an experience as unique and unpredictable as its users. Dynamically optimized algorithms, trained to detect, from the sensor signals, which aspects of the music give the most favorable results, could eventually allow the users to shape their experience in more detail. The system could also benefit from more efficient algorithms, which could more promptly or robustly extract currently used variables. For example, autocorrelation of the data could allow for faster detection of tempos, and give information as to off rhythm activity. Likewise, the extraction of additional features from the received data could add the needed complexity required of a captivating musical mapping. Data from other possible modes of input, such as TDOA position sensing, acceleration level, or user ID could add this needed complexity.

7.3 Future Applications

Outside of interactive dance, there are many venues where this low-cost sensing technique could be employed. Perhaps the most apparent are also in the entertainment industry. Large groups of people gather every day, in every part of the world, for entertainment. Whether it be a football game, a rock concert, or an amusement park, the associated activities could easily be enhanced with this system. Crowds at a basketball stadium, instead of doing 'the wave', could shake their sensor, embedded into a large foam hand emblazoned with their favorite team's logo. The system could easily sum the activities of the competing teams' fans by assigning each a different frequency or pulse width. The relative excitement of the fans could then be displayed on the JumboTron, or control lighting and sound effects, rallying their team to victory.

As a more serious application, the sensor could be used to sum the responses of an audience at a conference or convention. It can be used to give 'yes' or 'no' votes by simply pulse width modulating the output of the transmitter. These votes could then be tallied over the entire audience to reach a consensus on how to proceed. They could also be used to give feedback to the event planners or public speakers, as to which portions of the event were most interesting. By being instructed to shake the sensor in approval of various exhibits or portions of a talk, the sum total of hits collected in a zone or during a speech could indicate a favorable response to what is being shown in a particular area, or said at particular time.

Finally, perhaps the most powerful insight this research gives into other venues, are the implications for distributed sensing [26]. Currently, all research in the area of wide-scale, ubiquitous, or distributed sensing involves the combination of a microprocessor, sensing unit, and relatively high bandwidth communication platform. These devices tend to collect data over a large group of fairly complex parameters, and then send these data, real-time, with fast update rates, over a wireless link to other sensor nodes, or to a central processing node. These systems have the advantage of being accurate, robust, and versatile, although they tend to suffer, correspondingly, from high power consumption and cost. In many cases, the same functionality could be achieved with a less robust and versatile system.

The sensors described in this work, for example, could easily be redesigned to incorporate a different low cost sensor, such as a light dependent resistor, photodiode, force sensitive resistor, thermistor, or magnetic field sensing loop. The cost of these sensors would be so low that they could literally by littered onto the desired sensing area. In this way, the object upon which they land becomes active, able to transmit whether its state had just crossed a certain threshold, or give periodic updates as to its current state via pulse width modulation. By detecting the position of these sensors via TDOA, a central processing node could be fully aware of the state of all objects in the sensing area. If a sensor breaks, or its battery drains, it is of little consequence, as a new sensor could easily be dropped into its place. Or, since the sensors were put in so densely to begin with, its neighbors' data would be adequate to perform the needed task.

The same methodology used to map parameters to music could be employed to discern the state of this disposable sensing network. By beginning with all known parameters of the sensed system, and building outward, the central processing node could quickly learn what incoming data patterns represent. For instance, if the current sensor were placed in a paper mill's machine room, a few on each motor, pump, valve, or even doorway, the system would become accustomed to seeing particular values of frequency response or activity level as various operations occurred. If, for example, it is known to the system that *pump1* has just turned on, as vibration is detected in the area of *pump1* and the system recently called for pulp to be moved from *vat1* to *vat2*, but *valve1* has not opened, since vibration was not detected at *valve1*, *pump1* can be shut down to prevent cavitation. In the same way, if *pump1* has been sending hits every half second for the past three weeks, but suddenly starts sending a hit every 100 ms, it might be assumed that a bearing has worn, and maintenance could be performed that would prevent motor winding failure and loss of operation time. Although the current sensor's uncoded pulses may not be robust enough for a noisy industrial environment, simple pulse coding or spectral allocation could alleviate these difficulties.

These are a few ways in which these disposable, wireless sensors might be used to collect data over large groups, whether they consist of people or objects. Ultimately, the limit to

their functionality lie, not within the hardware, but within the creativity of the designer; and the difficulty in their implementation, not within the software, but within the discerning of the information from the noise.
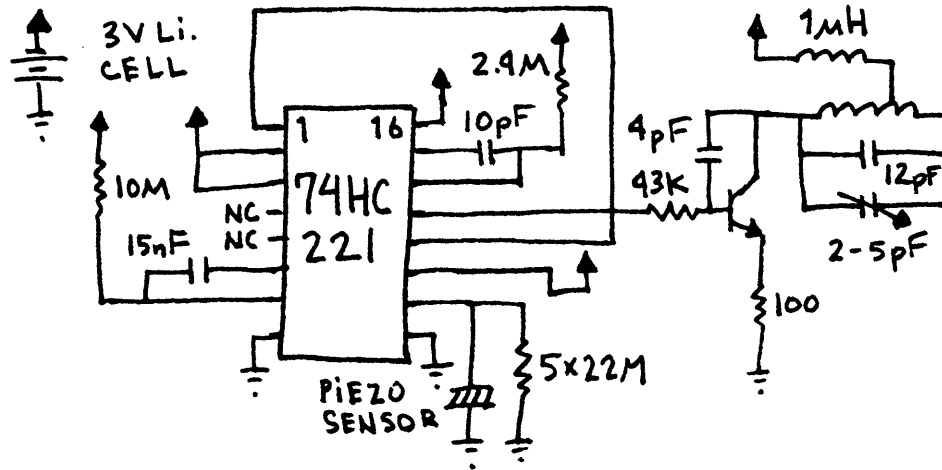
Appendix A

Hardware Schematics
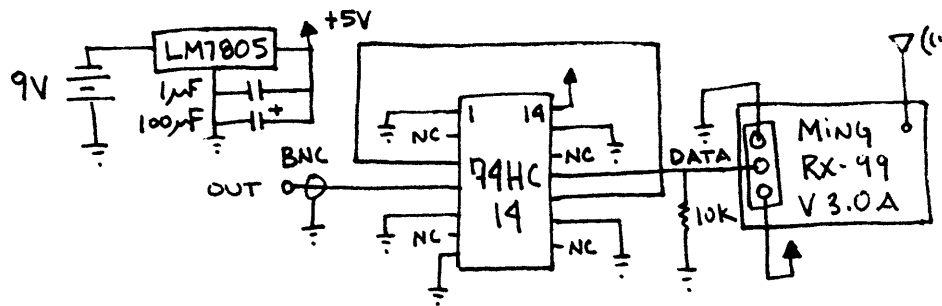
Figure A.1: Sensor: Revision Five Schematic

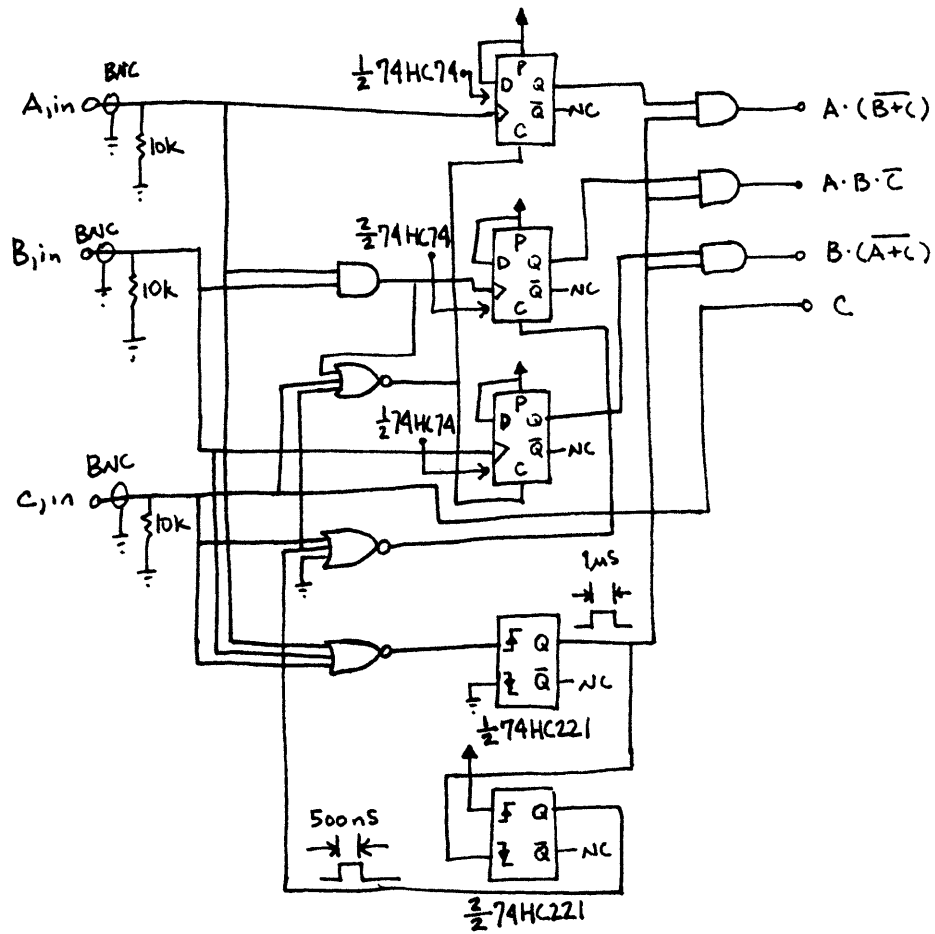Figure A.2: Receiver Base Station: Line Driver Board Schematic

Figure A.3: MIDI Converter: Logical Operations Board Schematic
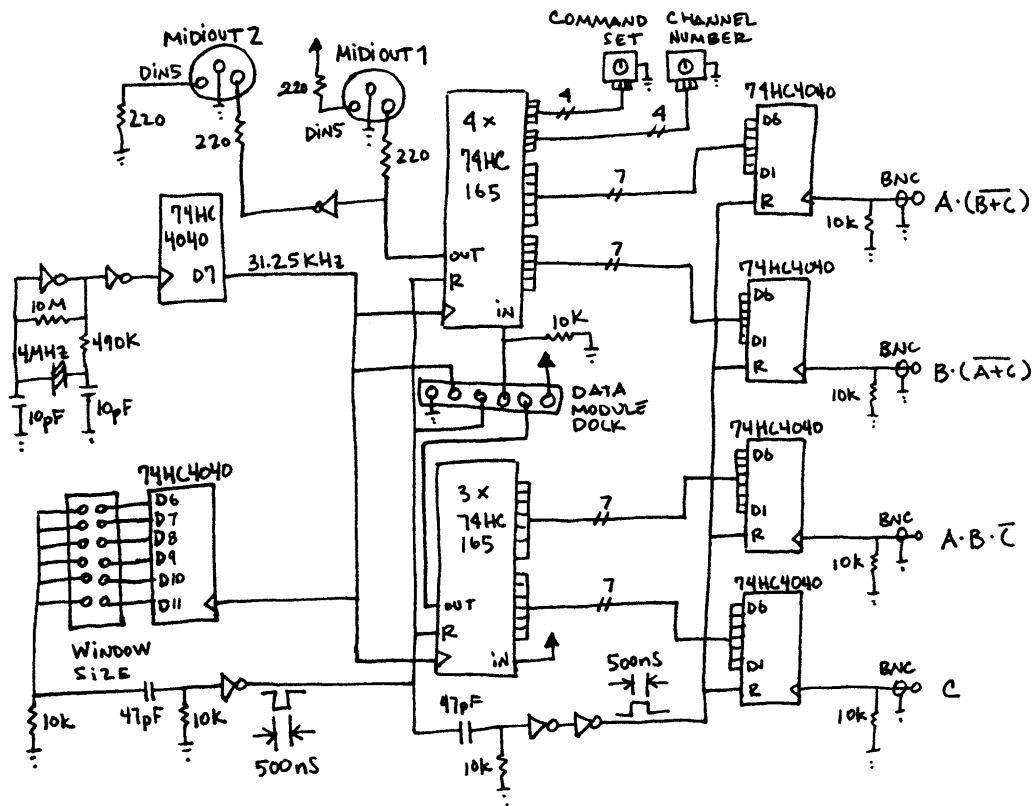
Figure A.4: MIDI Converter: Counter and Parallel to Serial Shift Register Board Schematic

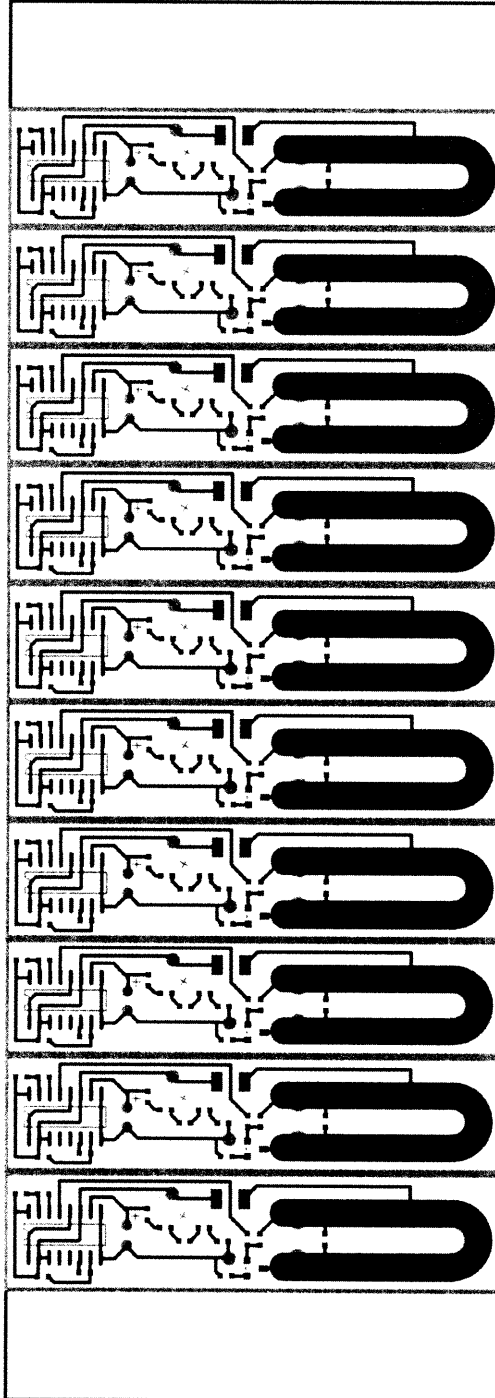# Appendix B

Sensor PCB Layouts

Figure B.1: Sensor PCB Layout for 10 Boards (actual size)

Appendix C

Beatgrabber Code

```c
/* beatgrabber.c -- output the beatgrabber of a group of numbers-------- */

#include "ext.h"
#include "fftw.h"
#include <math.h>
//   ** For more accurate timing
//   #include <time.h>

#define MAXSIZE 32
#define FFTWINDOW 4096        // Number of Samples in an FFTWINDOW
#define FFT_FREQUENCY 1       // Number of FFT's to comput in Hz
#define LOWER_BOUND 1         // Lower bound of frequencies we care about in Hz
#define UPPER_BOUND 4
#define SAMPLING_RATE 143     // Sampling rate for the data in Hz

#define VERBOSE 1            // Extra debugging
#define OPTIMAL 0            // An optimal plan (take more time to load patch initially) is computed if OPTIMAL==1


typedef struct beatgrabber
{
    struct object m_ob;
    Atom m_args[MAXSIZE];
    long m_index;          // The index in the input buffer.
    long m_fftstamp;       // Timestamp made the last time FFT was run.
    int m_bufferfull;      // 1 when the buffer has enough data to run an fft. (0 otherwise)
    short m_outtype;
    void *m_out;
    fftw_plan fftPlan;
} BeatGrabber;

fptr *FNS;
void *class;
fftw_complex *buffer;       // The data buffer we use to perform FFT analysis
fftw_complex *frequencyBuffer;  // Output buffer for the FFT

void add_Data(BeatGrabber *x);
float magnitude(fftw_complex a);
float max(fftw_complex* a, int *maxLocation);
void beatgrabber_bang(BeatGrabber *x);
void beatgrabber_int(BeatGrabber *x, long n);
void beatgrabber_float(BeatGrabber *x, double f);
//void beatgrabber_list(BeatGrabber *x, Symbol *s, short ac, Atom *av);
void beatgrabber_assist(BeatGrabber *x, void *b, long m, long a, char *s);
void beatgrabber_free(BeatGrabber *x);
void *beatgrabber_new(Symbol *s, short ac, Atom *av);

void main(fptr *f)
{

    FNS = f;

    setup(&class, beatgrabber_new,beatgrabber_free, (short)sizeof(BeatGrabber), 0L, A_GIMME, 0);
    addbang((method)beatgrabber_bang);
    addint((method)beatgrabber_int);
    addfloat((method)beatgrabber_float);
//   addmess((method)beatgrabber_list,"list",A_GIMME,0);
    addmess((method)beatgrabber_assist,"assist",A_CANT,0);
//   finder_addclass("Arith/Logic/Bitwise","beatgrabber");
}

void add_Data(BeatGrabber *x) {

    buffer[x->m_index].re = (float) x->m_args->a_w.w_long;     // Always fill the current position with the datum
    buffer[x->m_index].im = 0;

    // Now things get complicated.
    if (x->m_index >= FFTWINDOW) {                    // If we are in the second half of the buffer somewhere...
        buffer[(x->m_index - FFTWINDOW)].re = (float)(x->m_args->a_w.w_long);
// Fill the earlier part of the buffer as well.
        buffer[(x->m_index - FFTWINDOW)].im = 0;
        x->m_bufferfull = 1;
    }
    //   if(VERBOSE) {post("%ld : %ld   -- %f", x->m_index, (x->m_index - FFTWINDOW), buffer[x->m_index].re); }
    //if (x->m_index % 100 == 0) {
    //   post("        %d", x->m_index);
    //}

    if (x->m_index + 1 >= 2 * FFTWINDOW) {
        x->m_index = FFTWINDOW;
    } else {
        x->m_index++;
    }
}

float magnitude(fftw_complex a) {
    float out;
```

```c
    out = (a.re * a.re) + (a.im * a.im);
    return sqrt(out);
}


float max(fftw_complex *a, int *maxLocation) {
  float currentMax = 0;
  float q;
  int i, maxPos = 0;
  int upper, lower;
  // A quick note on frequency windowing
  // the frequency of the element i in an n element fft array
  // if f = samplingrate * i / n

  lower = FFTWINDOW * LOWER_BOUND / SAMPLING_RATE;
  upper = FFTWINDOW * UPPER_BOUND / SAMPLING_RATE;

  for (i = lower ; i < upper; i++) {
    q = magnitude(a[i]);
    if (q > currentMax) {
      currentMax = q;
      maxPos = i;
    }
  }
  *maxLocation = maxPos;
  return currentMax;
}


void beatgrabber_bang(BeatGrabber *x)
{
  register short i;
  int maxPosition;
  float bpm, maxValue;        // beats per second for bpm conversion

  long timeStamp = gettime();        // Make note of when the function was called.

//   if(VERBOSE) { post("%ld: %ld",timeStamp, (long)(x->m_args->a_w.w_long)); }
  add_Data(x);

  if ((x->m_fftstamp < timeStamp - (1000/FFT_FREQUENCY)) && (x->m_bufferfull)) {
    timeStamp = gettime();
    fftw_one(x->fftPlan, (buffer + (x->m_index - FFTWINDOW)), frequencyBuffer);
    if (VERBOSE) { post("FFT RUN IN TIME: %d", timeStamp - gettime()); }
    maxValue = max(frequencyBuffer, &maxPosition);
    if (VERBOSE) { post("frequencyBuffer max value : %f", maxValue); }
    if (VERBOSE) { post("frequencyBuffer max position  ::: %d", maxPosition); }
    x->m_fftstamp = timeStamp;
    bpm = (((float) maxPosition * (float) SAMPLING_RATE) / (float) FFTWINDOW) * 60.0f;
    outlet_int(x->m_out, round(bpm));
  }
}


void beatgrabber_int(BeatGrabber *x, long n)
{
  SETLONG(x->m_args,n);
  beatgrabber_bang(x);
}


void beatgrabber_float(BeatGrabber *x, double f)
{
  SETFLOAT(x->m_args,f);
  beatgrabber_bang(x);
}
/*
void beatgrabber_list(BeatGrabber *x, Symbol *s, short ac, Atom *av)
{
  register short i;
  if (ac > 31)
    ac = 31;
  for (i=0; i < ac; i++,av++) {
    if (av->a_type==A_LONG)
      SETLONG(x->m_args+i,av->a_w.w_long);
    else if (av->a_type==A_FLOAT)
      SETFLOAT(x->m_args+i,av->a_w.w_float);
  }
  x->m_count = ac;
  beatgrabber_bang(x);
}
*/
void beatgrabber_assist(BeatGrabber *x, void *b, long m, long a, char *s)
{
  assist_string(3008,m,a,1,3,s);
}


void beatgrabber_free(BeatGrabber *x)
{
  int i;
```

```
    if(VERBOSE) { post("Closing BeatGrabber and freeing memory..."); }

    fftw_destroy_plan(x->fftPlan);

    free(buffer);
    free(frequencyBuffer);

    if(VERBOSE) { post("Done.  Have a nice day."); }

}

void *beatgrabber_new(Symbol *s, short ac, Atom *av)
{
    BeatGrabber *x;
    int i;

    if(VERBOSE) { post("Loading beatgrabber!",NULL); }

    x = (BeatGrabber *)newobject(class);


    x->m_index = 0;      // Start indexing the buffer from position 0
    x->m_fftstamp = 0;
    x->m_bufferfull = 0;

    if (ac) {
      x->m_args[1] = *av;
      if (av->a_type==A_LONG) {
        x->m_outtype = A_LONG;
        x->m_out = intout(x);
        SETLONG(x->m_args+1,0L);
        SETLONG(x->m_args,0L);
      } else if (av->a_type==A_FLOAT) {
        x->m_args[0].a_type = x->m_outtype = A_FLOAT;
        x->m_out = floatout(x);
        x->m_args[0].a_w.w_float = 0;
      }
    } else {
      x->m_outtype = A_LONG;
      x->m_out = intout(x);
      SETLONG(x->m_args+1,0L);
      SETLONG(x->m_args,0L);
    }

    // Create a FFTW plan.  This makes transforms very fast later.
    if(VERBOSE) { post("Creating FFTW Plan."); }
    if (OPTIMAL) {
      x->fftPlan = fftw_create_plan(FFTWINDOW, FFTW_FORWARD, FFTW_MEASURE);
    } else {
      x->fftPlan = fftw_create_plan(FFTWINDOW, FFTW_FORWARD, FFTW_ESTIMATE);
    }
    if (x->fftPlan == NULL) {
      error("Warning: FFT Plan could not be created!"); }

    // Initialize the FFT buffer
    if(VERBOSE) { post("Allocating buffer space for %d sample transforms.",FFTWINDOW); }
    buffer = (fftw_complex *) malloc(2 * FFTWINDOW * sizeof(fftw_complex));
    frequencyBuffer = (fftw_complex *) malloc(FFTWINDOW * sizeof(fftw_complex));

    return (x);
}
```

Appendix D

Survey Results

Statement	Disagree 1 [%]	2 [%]	Neutral 3 [%]	4 [%]	Agree 5 [%]	No Response [%]
I am an avid dancer	12.0	21.3	18.5	25.9	21.3	00.1
I often listen to electronic music	17.6	22.2	21.3	15.7	21.3	01.9
I enjoyed this interactive experience	10.2	13.0	20.4	34.3	20.4	01.9
I spent my time actively dancing, rather than just standing around	10.2	13.0	21.3	25.0	28.7	01.9
I danced by myself, rather than with others	25.9	18.5	23.1	15.7	13.0	03.7
I felt in control of the experience	25.9	22.2	28.7	13.9	05.6	03.7
The music and lighting responded well to my motions	24.1	17.6	37.0	10.2	06.5	04.6
I would have liked to have more input	00.9	11.1	26.9	28.7	25.0	07.4
I felt that the sensors were well suited to the way I danced	16.7	18.5	36.1	13.9	07.4	07.4
I would have preferred a different interface	13.9	06.5	38.0	13.0	10.2	18.5
The music was sufficiently varied to keep the experience interesting	22.2	23.1	25.0	18.5	06.5	04.6
A different style of music would have made the experience more enjoyable	07.4	11.1	31.5	19.4	22.2	08.3

Table D.1: *Sidney1*: Questionnaire Results (108 respondents)

Statement	Disagree 1 [%]	2 [%]	Neutral 3 [%]	4 [%]	Agree 5 [%]	No Response [%]
I am an avid dancer	09.4	03.1	18.8	43.8	21.9	03.1
I often listen to electronic music	00.0	15.6	18.8	21.9	43.8	00.0
I enjoyed this interactive experience	00.0	00.0	15.6	46.9	34.4	03.1
I spent my time actively dancing, rather than just standing around	03.1	06.3	09.4	46.9	34.4	00.0
I danced by myself, rather than with others	12.5	18.8	15.6	28.1	25.0	00.0
I felt in control of the experience	06.3	21.9	40.6	21.9	09.4	00.0
The music and lighting responded well to my motions	06.3	09.4	43.8	31.3	09.4	00.0
I would have liked to have more input	03.1	09.4	28.1	25.0	34.4	00.0
I felt that the sensors were well suited to the way I danced	00.0	09.4	28.1	43.8	15.6	03.1
I would have preferred a different interface	15.6	12.5	34.4	21.9	00.0	15.6
The music was sufficiently varied to keep the experience interesting	00.0	12.5	15.6	43.8	25.0	03.1
A different style of music would have made the experience more enjoyable	06.3	21.9	37.5	18.8	09.4	06.3

Table D.2: *Talbot3*: Questionnaire Results (32 respondents)

Bibliography

[1] Mididancer. http://www.troikaranch.org/mididancer.html.

[2] J.A. Paradiso, K. Hsiao, A. Benbasat, and Z. Teegarden. Design and implementation of expressive footwear. *IBM Systems Journal*, 39(3&4):511–529, 2000.

[3] M. Feldmeier and J.A. Paradiso. Ultra-low-cost wireless motion sensors for musical interaction with very large groups. In *Online Proceedings for the Designing Ubiquitous Computing Games Workshop*, Atlanta, Georgia, Oct 2001. ACM UbiComp. See: http://www.playresearch.com/workshops/ubigame.ubicomp/papers/feldmeier.pdf.

[4] M. Feldmeier, J.A. Paradiso, and M. Malinowski. Large group musical interaction using disposable wireless motion sensors. In *Voices of Nature: Proceedings of the 2002 International Computer Music Conference*, pages 83–87, San Fransisco, Sep 2002. ICMC.

[5] Dance Dictionary. http://www.eurodancehits.com/dict.html.

[6] MIDI Manufacturers Association. http://www.midi.org.

[7] Cycling '74. http://www.cycling74.com.

[8] DMX512. http://www.usitt.org/DMX/DMX512.htm.

[9] S. Singhal and M. Zyda. *Networked Virutal Environments: Design and Implementation*. ACM Press, SIGGRAPH Series, New York, 1999.

[10] J. Falk, P. Ljungstrand, S. Bjork, and R. Hansson. Pirates: Proximity-triggered interaction in a multi-player game. In *Procedings of the CHI 01 Conference on Human Factors in Computing Systems, Extended Abstracts*, pages 119–120, New York, 2001. ACM Press.

[11] R. Ulyate. Interactive dance club. In *SIGGRAPH98 Conference Abstracts and Applications*, pages 230–235, New York, 1998. ACM Press.

[12] Z. Neda, E. Ravasz, Y. Brecht, T. Vicsek, and A.L. Barabasi. The sounds of many hands clapping. *Nature*, 403:849, 2000.

[13] D. Waxman. Digital theremins: Interactive musical experiences for amateurs using electric field sensing. Master's thesis, Massachusetts Institute of Technology, Cambridge, Massachusetts, 1995.

[14] Cinematrix. See http://www.cinematrix.com.

[15] R. Picard and J. Scheirer. The galvactivator: A glove that senses and communicates skin conductivity. In *Proceedings from the 9th International Conference on Human-Computer Interaction*, New Orleans, Sep 2001.

[16] D. Maynes-Aminzade, R. Pausch, and S. Seitz. Techniques for interactive audience participation. In *Proceedings of the Fourth International Conference on Multimodal Interfaces*. IEEE Computer Society Press, Oct 2002.

[17] E. Berger et al. A sophisticated soiree. In *Take Over, Proceedings of the 2001 Ars Electronica Festival*, pages 352–353, Vienna, 2001. Springer-Verlag.

[18] D. Graham-Rowe. Don't stop moving. *New Scientist*, 172(23):25, Nov 2001.

[19] J. Randall. Real-time lighting system for large group interaction. S.B. Thesis, Massachusetts Institute of Technology, Cambridge, Massachusetts, May 2002.

[20] Measurement Specialties, Inc. LDT0 solid state switch/vibration sensor. http://www.msiusa.com/ldt0.pdf.

[21] Measurement Specialties, Inc. http://www.msiusa.com.

[22] Measurement Specialties, Inc. Minisense 100 vibration sensor. Application specification 21-01, Feb 2002.

[23] M. Clynes. *Sentics: The Touch of the Emotions*. Doubleday and Company, New York, 1997.

[24] Fairchild Semiconductor. MM74HC221A dual non-retriggerable monostable multivibrator. Datasheet, Sep 2001. http://fairchildsemi.com/ds/MM/MM74HC221A.pdf.

[25] K. Pahlavan, L. Xinrong, and J.P. Makela. Indoor geolocation science and technology. *IEEE Communications Magazine*, 40(2):112–118, Feb 2002.

[26] J. Lifton. Pushpin computing: a platform for distributed sensor networks. Master's thesis, Massachusetts Institute of Technology, Cambridge, Massachusetts, 2002.