Applications of Expressive Footwear

by

Eric Hu

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Master of Engineering in Electrical Engineering and Computer Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

May 1999
[June 1999]

Author ....
Department of Electrical Engineering and Computer Science
May 21, 1999

Certified by ....
Joseph Paradiso
Principal Research Scientist, MIT Media Lab
Thesis Supervisor

Accepted by ....
Arthur C. Smith
Chairman, Department Committee on Graduate Students

Applications of Expressive Footwear

by

Eric Hu

Submitted to the Department of Electrical Engineering and Computer Science
on May 21, 1999, in partial fulfillment of the
requirements for the degree of
Master of Engineering in Electrical Engineering and Computer Science

Abstract

The design of Expressive Footwear centered on creating as dense a sensing module as possible. Originally, the motivation came from creating a dance performance in which the dancer is participating directly in the creation of music in the dance performance. In order to accomplish this, a wide variety of sensors and sensing techniques are used to quantify physical parameters of the foot. Once the sensing is accomplished, the resultant data is transformed to music in real time. Recent work has also included using Expressive Footwear in gesture recognition. The development of this system and each of its parts is elaborated upon in this thesis.

Thesis Supervisor: Joseph Paradiso
Title: Principal Research Scientist, MIT Media Lab

Acknowledgments

I would like to acknowledge and thank Joe Paradiso for all of his guidance, support and vision throughout this project.

I would like to acknowledge and thank Kai-yuh Hsiao for all of his resourcefulness and creativity.

I would also like to thank and acknowledge Andrew Wilson for his innovation and selflessness.

Others I would like to thank for their help and technical expertise are Matt Gray, Joshua Strickon, Rehmi Post, Matt Reynolds, Ari Adler, Ari Benbasat and Zoe Teegarden.

For their help in performance matters, I would like to thank David Borden and Byron Suber of Cornell University as well as Yuying Chen, Mia Keinanen, Sarah Brady and Brian Clarkson.

Jack Memishian from Analog Devices deserves thanks as does Kyung Park from AMP Sensors (now Measurement Specialists). The Things That Think consortium at the Media Lab has my gratitude for their support.

I would also like to dedicate this to the memory of Phil Gale.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Input devices are no longer keeping up with the power of computers. In many fields, ranging from gaming to data acquisition, the processing power of the computer is no longer the bottleneck. The way we get information about the world to our computers is becoming the limiting factor. This is particularly true in real-time gaming, where the human players' slower interface can no longer be compensated for by superior strategy.

Expressive Footwear is an attempt to open up this bottleneck. By sensing a myriad of aspects about what humans do physically, we can begin to break the barrier between human thought and computational action. Though Expressive Footwear and its applications are not an attempt to help solve the gamers' plight, the idea of very dense sensing, the center of Expressive Footwear, might some day do so.

In this chapter, some motivation for Expressive Footwear is given. Some rather abstract ideas about the human condition are discussed along with the very practical side of industrial demands. After this, the design of the core hardware and sensing technology is revealed in Chapter 2. The applications we devised for Expressive Footwear are explained in Chapter 3 and some conclusive remarks about our results as well as the future of such technology are mentioned in Chapter 4. Throughout this paper, the term "Expressive Footwear" refers to the shoe and base station (all but the two rightmost blocks of Figure 1-1).

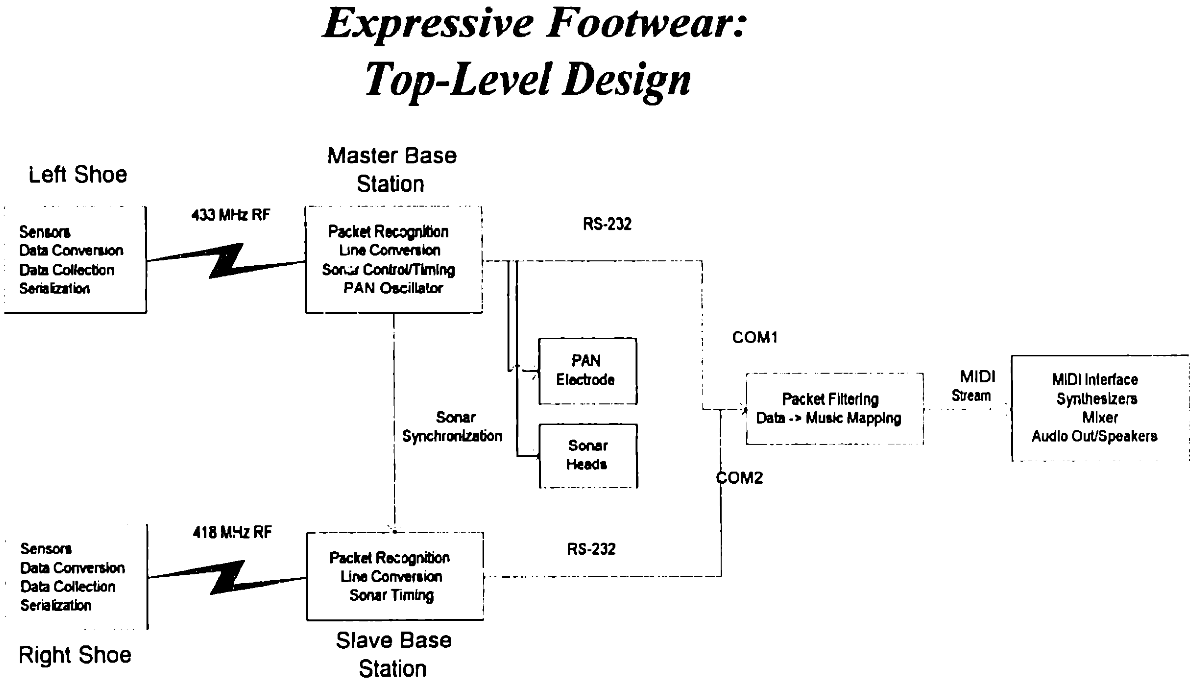


Figure 1-1: Block diagram of Expressive Footwear

1.1 Pure Input Device

Expressive Footwear is a very flexible sensing system. It gathers a lot of data about its environment. Already, this data gathering is being exploited for uses other than footwear (see [7] and Section 3.2.2). This is just the beginning of the possibilities. The world of The Lawnmower Man, in which a full body virtual experience opens up the pure creative force of the human mind, is taking shape. Of course, the ultimate goal in all of this is to create the Holodeck from Star Trek, albeit with much more advanced technology.

Expressive Footwear is just one piece of the epic puzzle for the realization of such an environment. But subsets of this immense, new virtual world can start piecing themselves together with dense sensing and its implications on input devices. The more useful information a computer can get, the closer we can come to achieving meaningful interactions.

Despite the fact that Expressive Footwear was not designed with gaming in mind, the ability to create these virtual worlds would aid the real-time gamer tremendously. Interfaces that mirror everyday actions provide smoother control and avoid clumsy limitations in today's hardware.

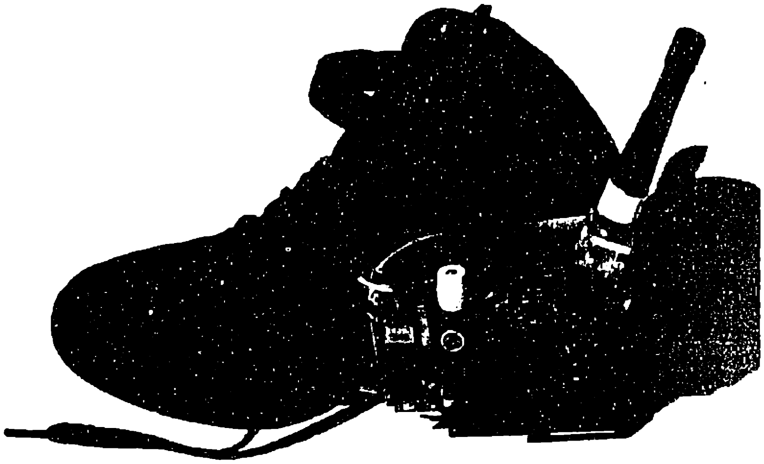


Figure 1-2: Picture of the Original Shoe

1.2 Human Expression

When personal computers first began to find their way into schools and homes, the keyboard was the only interface mechanism. After some time, improved graphics cards and processors allowed the much more intuitive mouse to come into widespread use. These kinds of steps are the right ones to make because it takes advantage of what humans do and understand naturally. As we improve devices, they should merge how humans think of problems with the procedures that computers use to help solve them. Or, even more generally, input devices should encourage a coexistence of human and computational resources at all levels.

Specifically with regard to dance, Expressive Footwear is allowing dancers to communicate through two senses rather than one. Traditional dance, barring specialties like tap dancing, gives the dancer full control over the visual aspect of the performance, but little to no control over the aural aspect. With a very attentive conductor and orchestra in many varieties of staged dance, some subtle changes in tempo and perhaps dynamic are possible; but none of the phrasing or sounds are under the dancer's control. In an environment with Expressive Footwear, such control becomes possible.

The principle problems with such an approach is the inherent coupling between the visual and the aural. The very motions that control the sound are not necessarily the same ones that are visually interesting or expressive. One of the great challenges of the project was creating a sound mapping from the sensor values on the shoe that made sense in both realms.

This problem, again, comes back to merging the human and computational world. We have to create a reaction from the computational system that fits within some framework of human expectation. In particular, the reactions elicited from the audience by the dancer's movement should be somewhat consonant with the reaction elicited by the sounds they hear. Of course, there is a wide spectrum of expressive possibility in toying with the audience's expectations and intentionally jarring them as well.

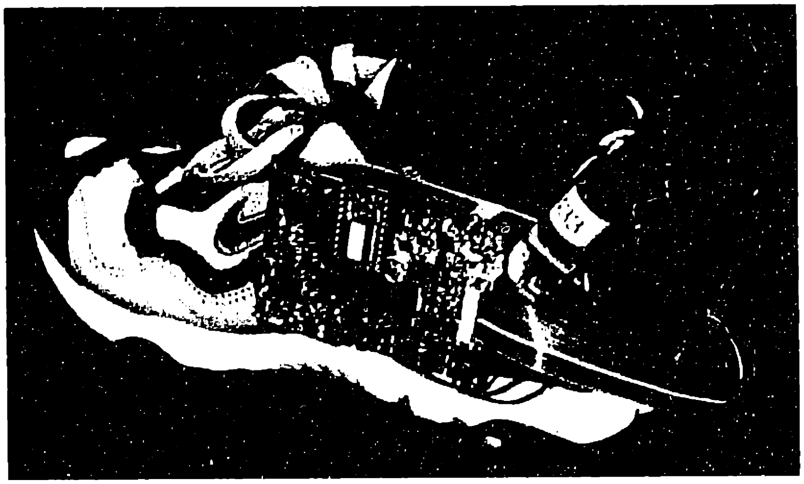


Figure 1-3: Recent Picture of the Shoe

1.3 Medical Applications

The actual implementation of such a setup is quite complex. Even when Expressive Footwear, which is at the heart of the system, performs ideally, a computer still needs to process the data. Sound synthesis systems are needed to produce the audio signals. If multiple synthesizers are used, a mixer is needed and, of course, speakers create the actual sound. Though implementing this system is a lot of fun and opens up a unique palette of expressive qualities, there are many applications for which such an elaborate setup is overkill.

A wide array of medical applications would need only Expressive Footwear and some sort of data logging device, presumably a computer. After a patient performs some set of motions designed to diagnose foot problems, the doctor can look at the data and prescribe treatment. Some work has already been done with very dense force-sensing arrays at footwear manufacturer laboratories and pediatric treatment facilities [2]. Even more time and cost effective would be preventive care for patients. As part of a regular physical examination, doctors could use devices like Expressive Footwear to observe high-risk patterns of everyday physical motion before problems arise.

One major problem with these systems is that they tend to be hardwired and stationary, making it very awkward for use by the patient. Some systems are emerging (e.g., www.clevemed.com) that sacrifice sensor density for portability. However, much of this work looks only at the pressure along the bottom of the foot and miss many other characteristics. Expressive Footwear could easily augment the power of such systems with added data.

Athletic footwear is also a good place for Expressive Footwear applications. Already, some systems have been developed for training (e.g., www.pro-balance.com) purposes, while others use inertial sensors to make pedometry measurements [6]. Training, with a little development, seems to have great promise and is a good example of the right applications for such electronics. However, it is rare for these systems to exploit more than a couple of physical parameters. It is possible and likely that these very specific applications do not require any more than a small number of parameters, a system that provides more information would be much more flexible with very little extra cost involved.

It would also be very beneficial in athletics to use Expressive Footwear for the same preventive purposes as described above. Many sports injuries occur over long periods of time, particularly joint injuries (knees, back, etc.). If there is a repetitive motion that continuously wears on a joint and it can be detected, the user can be given a warning, providing a possibility of injury prevention and extending athletes' careers.

Chapter 2

Design

The complete Expressive Footwear system consists of four primary components–sensing, data acquisition, communication and interpretation. Sensing involves the devices that interact with the physical world and give information about it. Data acquisition gathers and digitizes the information the sensors give back. These two stages are done in the shoe and can be seen as the two leftmost blocks in Figure 1-1. Communication takes this data and makes it available to whoever speaks the protocol (the jagged lines through the RS-232 lines in Figure 1-1). Once the data is transferred, a variety of different high-level interpretations are possible. Our use of the system as a musical system for dance is shown by the last two boxes in Figure 1-1. The first three stages are elaborated in this chapter. We also show some of the data taken in Appendix B.

Though it is easy to speak of the first three stages as separate, they begin to blur into each other, particularly data acquisition and communication. This chapter details the three stages and presents some of the design issues in integrating the components into an operational system.

2.1 Sensing

Many different sensors were used to extract local physical parameters of the foot. One of the challenges was deciding which aspects of the foot to sense. Both a discussion of the decision process and a detailed description of each of the sensors follows. It should be emphasized that, though the discussions focus on dance performance, many of these parameters are very useful for other applications, particularly medicine and wearable computing [12]. Dance, however, can take sensing to the extreme, emphasizing the degrees of freedom and sensitivity that are required.

2.1.1 Important Parameters of the Foot

As this was a dance shoe, finding useful pressure points along the bottom of the foot was important, particularly around the ball of the foot and under the heel. Much of a dancer's balance throughout their entire body effects these pressure points thus there is useful information to be gained there. A similar parameter we wanted to detect was how a dancer's foot would either flex or point during a performance. This was a natural choice of parameters since both flexing and pointing are relatively easy for a dancer to control.

A full, 3-axis characterization of orientation was also important. Many of the most graceful and expressive dance gestures involve slow arcs of motion through the air. Though this is particularly dramatic with a dancer's arms, the foot accomplishes similar effects in a more subtle manner. Spins and twirls are also very important dance movements. A successful characterization of orientation will also yield good data about angular rate.

Other gestures we felt important to a dance performance were fast-moving gestures. In particular, we wanted to capture kicks and jumps, where the body is going through rapid changes of direction. As a contrast to the graceful movements of the previous two classes of sensing, we also wanted to detect jerky, jagged motions.

A few different sensing techniques were also used to extract more global parameters. One of these was the height of the dancer's foot above the stage. This could augment either the fast or slow motions mentioned above or could be used on its own.

Another global parameter that required more sophisticated sensing techniques was trying to find the dancer's position on stage. As the dancer would move across the stage, one could imagine he/she to be moving through various different zones. In the approximation that the height is much less than the distance from the positional sensors, it is also possible to reconstruct all three axes of absolute position from at least two different range measurements to the foot.

2.1.2 Sensors Used to Measure the Different Parameters

The one characteristic we looked for in all of the sensors is that they were small. Keeping the overall goals of "dense sensing" in mind, the sensors should be as unobtrusive as possible. This meant that, where possible, we chose surface-mount devices and components. Also, if we found two sensors that were similar in function, we chose the smaller of the two. This was a rather important consideration throughout the project. It may be helpful to refer to Figure 2-1 to see how each of the sensors fits in to the overall system on the shoe.

Pressure

For the purpose of embedding into a shoe, the pressure sensors must be extremely flat and quite flexible. Two basic types of pressure sensors fit this description. One type is made of polyvinylidene difluoride (PVDF), a piezoelectric polymer. The other type is a force sensing resistor (FSR). In the Expressive Footwear shoe, both are used.

PVDF is an active material. When force is applied to the polymer, an AC potential difference is created across its two terminals. Due to this property, it is much better suited to measuring dynamic force than static force, making it ideal as a trigger. For dance purposes, placement of such a trigger in the heel is best because the dynamic characteristics expressed in this part of the foot capture the most gestural information. Static characteristics in the heel are difficult for a dancer to control, also making dynamic measurements adequate. In addition, PVDF film is very thin and flexible as well as being quite physically robust.

FSRs are two terminal devices that measure continuous pressure over the area of the sensor. The resistance across the terminals changes as force is applied. Structurally, FSRs have two flexible, conductive surfaces with regular, finger-like protru-

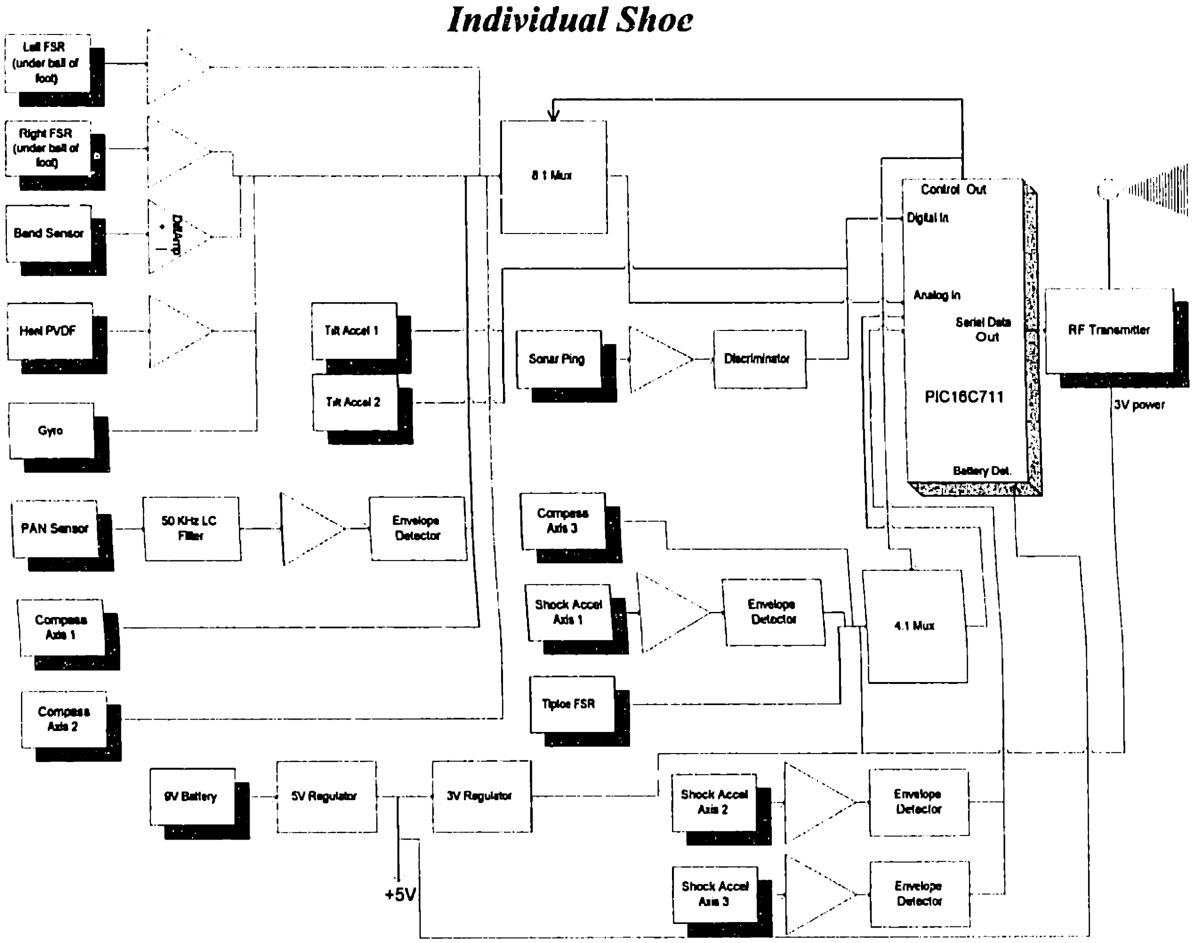


Figure 2-1: Block Diagram of the Shoe's Electronics

sions in the surface. These two surfaces are placed parallel to each other with the protrusions facing each other. This creates a middle section with interleaving fingers. As the parallel surfaces are pressed together, more and more fingers make contact, creating an increased number of parallel paths for current to flow. Thus, the resistance of the device is reduced with applied pressure. Because FSRs measure continuous pressure, we put them where the dancers have the most control, at the tip of the toe and at two spots under the ball of the foot.

Due to differing physical demands, we used two different brands of FSR, one from Interlink, the other from Flexiforce (see Table A.3 for more information). The ones from Interlink are thicker, thus stiffer and less flexible. Flexiforce FSRs, as the name implies, are quite flexible and very thin. We used the Interlink FSRs under the ball of the foot, where the extra stiffness would result in robustness. The Flexiforce FSRs were used at the tip of the toe, where the sensor was more likely to be bent around. In fact, we damaged some Interlink FSRs by trying to use there.

FSRs are not very good for taking precise measurements over large ranges of pressure since they become very nonlinear in extreme pressure ranges. However, for the relatively small pressure ranges we are operating in, we can approximate linearity.

Bend Sensor

The bend sensor from Abrams-Gentile (see Table A.3) used is quite similar to the FSRs described above when viewing the terminals. As the sensor is angularly displaced, the resistance of the device changes. The structure of this sensor, however, is somewhat different. Rather than using substrates with interleaved fingers, the bend sensor uses a conductive ink with carbon particles in a binder. As the sensor is bent outward, the inter-particle distance increases. This also results in an increase in resistivity.

Orientation

Though orientation is local in the sense that it is independent of absolute position within our projected operating space (about 30 feet), it is global in the sense that it requires some reference to absolute direction. When the foot is not being accelerated quickly, using gravity is a natural choice, but another absolute directional reference is required to fix all axes. With gravity alone, any spinning about the vertical axis will remain undetected by our sensors. The other natural choice was use of the earth's magnetic field, as in a compass. With three axes of orientation about the local magnetic field as reference, we can resolve the final singularity, provided that it is not aligned vertically.

To detect the pull of gravity, we used a 2-axis, 2g accelerometer from Analog Devices (see Table A.3). It is a micromachined accelerometer with two sets of MEMs springs in two perpendicular axes. Each set controls the deflection of a suspended proof mass that moves the plates of a parallel plate capacitor. By inertially moving the capacitor plate, physical acceleration is converted to an electrical signal. Two different types of outputs are available from the sensor, a digital, pulse-width modulated (PWM) rectangular wave and a continuous analog signal. Due to the analog I/O limitations of the on-shoe processor (Microchip 16C711), we selected the PWM wave. Its duty cycle is directly proportional to acceleration. The PWM wave output is also gives better accuracy, stability and resolution.

For measuring the earth's magnetic field, we tried two types of electronic compasses. The first one used mechanical deflections of a magnetic compass to drive an electrical output signal from two Hall sensors. In a very small package, such as the Analog Devices accelerometer described above, mechanical coupling works quite well because your bandwidth is limited by the mass of much smaller objects. However, it was quite clear early on that a mechanically coupled compass was too slow for our needs. Additionally, the particular compass we used from Dinsmore (seen as the white cylinder atop the circuit card in 1-2) also ran into mechanical and longevity problems in a dance performance, becoming unusable after a few hours of performance.

Our second compass was a solid-state compass from Honeywell (see Table A.3). It used permalloy magnetoresistive transducers to sense the strength of magnetic field along three axes. In terms of the transducers themselves, their resistance changes with the parameter they are measuring, like the FSRs and bend sensors. However, the changes are very slight and the sensor uses an on-board bridge circuit followed by a stage of amplification to pull up its output response to reasonable levels. Permalloy has a tendency to drift over time, so Honeywell includes a reset pin to strap the permalloy back into magnetic saturation.

Though we could deduce angular rate from the compass measurements, a numerical differentiation can introduce delay and noise. To make a real-time dance system, it is much easier to simply use a sensor that directly measures spin of the ankle, where it is most often expressed. For this, we used a miniature vibrating-read gyro from Murata (see Table A.3). It also uses piezoelectric material to sustain vibration, while taking advantage of Coriolis forces to measure angular rate. A rate along one axis orthogonal to the vibration is precessionally coupled into the other, where it is measured.

Rapid Foot Motion

Since rapid motions are best characterized by their acceleration, we picked another accelerometer rated in the range of ±5 – 10g to detect them. Though this provides some redundant data from our previous accelerometer, we still need both to adequately detect the full range of activity (e.g. generic foot motion and tilt as well as stomps and kicks). Our processor's on-board analog-to-digital converter (A/D) has 8 bit resolution. Even if we work as low as ±5g, finding tilt from acceleration (±1g) is compromised by more than two bits; and, of course, things only get worse if we try to handle even higher levels of acceleration.

The accelerometer we settled on is a three-axis package made by the former AMP Sensors (now Measurement Specialists, see Table A.3). It also uses piezoelectric material. For each axis, a cantilever beam is used, with one end fastened while the other end is allowed to flex in response to acceleration. This flexing excites the piezo material, yielding the desired electrical response.

Absolute Position

To gain information about absolute position, we took advantage of two existing technologies and combined them. The first is electric-field sensing [14], which we use for height (above a conducting platform on the floor), and the second is sonar, which we use for positioning.

Our utilization of electric-field sensing places an electrode receiver in the sole of the shoe. This electrode acts as one side of a parallel plate capacitor while an electrode on or under the stage acts as the other end. The base station (discussed in Section 2.3), generates a sine wave that is connected to an electrode on stage. The shoes are tuned to 50kHz and the signal generation at the base station is adjusted to match this. Perhaps a better way to think of the electrodes are as capacitive antennae, the one on stage being a transmission antenna and the one in the shoe a receiving antenna. Based on the strength of the signal received by the shoe, it is possible to determine

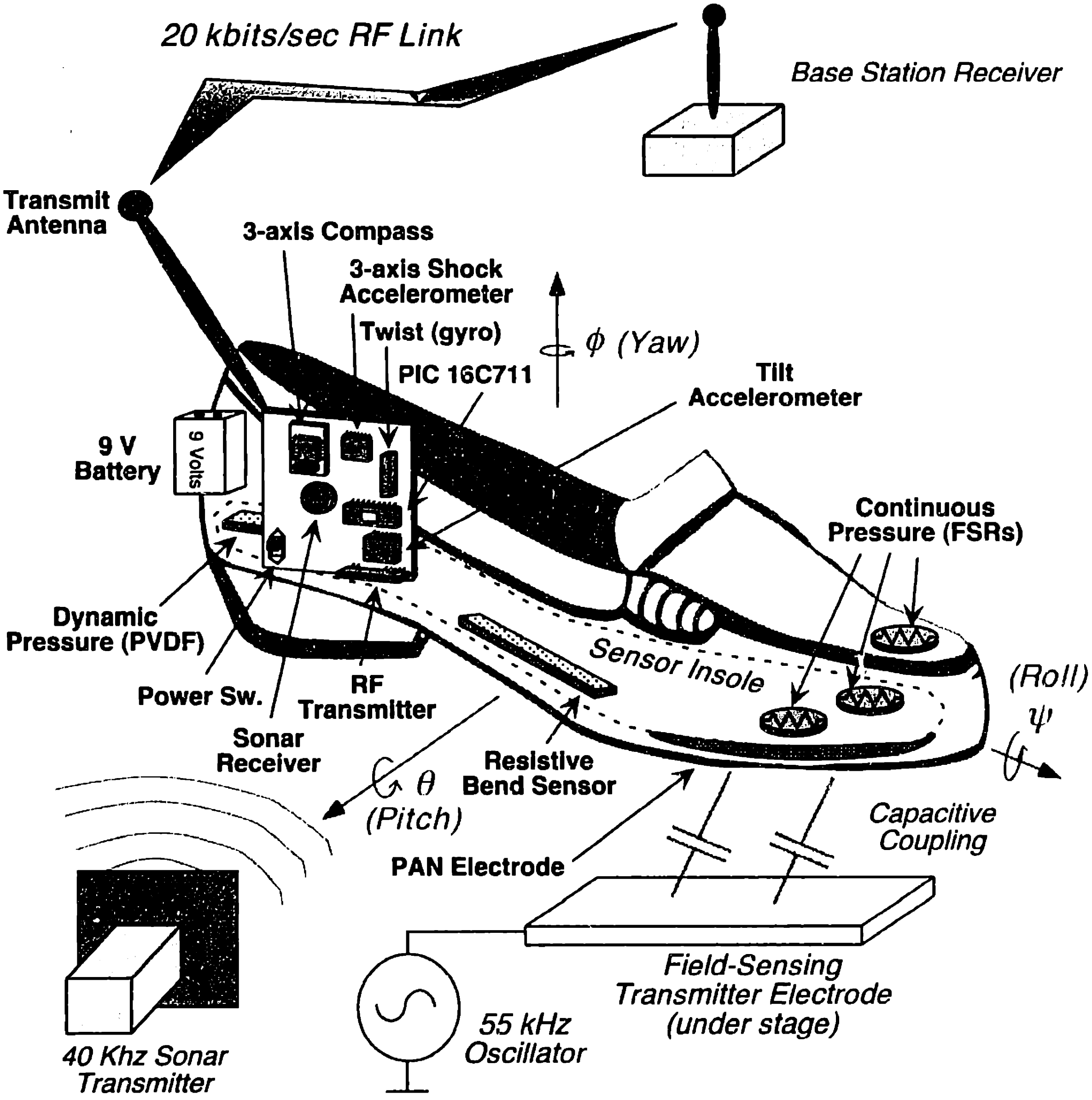


Figure 2-2: Sensors in the Final Version of Expressive Footwear

the distance between the two electrodes. With a wide, flat electrode on the dance stage, this distance equates to height. It should be noted that the electrode could cover any arbitrary section of the stage and could be in multiple pieces if so desired, though second-order effects can arise from variation in the capacitive grounding of the performer that may be undesirable.

The base station also controls the sonar measurements. It can send pulse signals to up to four different pinging stations, which generate $1ms$ long, 40kHz pulses that feed piezoceramic ultrasonic transducers from Polaroid (see Table A.3). We have configured our processor to enable pinging in one of four modes: pulsing all four stations simultaneously, pulsing two at a time in sequence, pulsing one at a time in sequence or pulsing one at a time.

To extract the data, we have devised the following algorithm. When the base station sends a pulse, it begins to count at 0.4ms intervals. The shoe receives the ping and begins counting itself at the same rate. When the shoe transmits its data packet, it sends back the value of its count, which the base station subtracts from its own count, giving us the number of counts between transmission from the base station and receipt of the ping in the shoe. Since the speed of sound can be approximated as constant (1129 ft/s or about 344 m/s) in air, each count corresponds to about 14 cm. With a range of 250 counts, we get about 34m of range.

2.2 Data Acquisition

This section details the electronics involved in translating sensor behavior into numerical data. In addition to the design of analog circuits for each individual sensor, efficient code had to be written for the processor to integrate all of the data into a single serial stream. This is one example of how communication constraints influenced our data acquisition methods. Throughout this section, it will probably be helpful to refer to Figures B-2, B-1 and B-3.

2.2.1 Processor

This stage of data acquisition actually comes after the electronics described in the following sections. However, since the electronics are very flexible and our choice of processors is more constrained, it makes sense to pick a processor first.

In an attempt to keep all of our devices as small as possible, we decided on a peripheral interface controller (PIC) from Microchip, specifically the 16C711. It provides sufficient execution speed and program memory along with 13 I/O pins (5 analog, 8 digital) and an on-board 8-bit A/D. The PIC runs at 16MHz (evaluating instructions at 4MHz) with 1kB of ROM, 68 8-bit registers, and also comes in a compact, 18-pin form factor (see Table A.1).

The large number of I/O pins was essential for a system with such heavy data requirements. However, with so few analog inputs, we had to utilize two multiplexers (one 8:1, the other 4:1) to expand the number of possible digitized analog signals from four to fourteen. This, of course, used up 3 of our digital pins as address lines. The lack of analog lines also made the decision to use the PWM output of the ADXL202 trivial. Newer versions of this circuit card use two 8-input multiplexers, accommodating another four analog channels, useful in other applications (see Table A.1).

Moving the A/D into the processor saved a lot of space and configuration difficulty. With a separate A/D, several more digital lines would have been needed, to say nothing about increases in power and space requirements.

The PIC was the only component where we picked a through-hole model rather than a surface-mount model. All of the available surface-mount PICs are only one-time programmable. Using these would have made the process of debugging the embedded code intolerable. Not only would we have wasted a large number of PICs in trying different versions of code, but we would have wasted a lot of time detaching the PICs from the board as well. Future versions of this circuit board, which can use the code developed and debugged on the prototypes, could be designed around a surface-mount PIC and save considerable space.

2.2.2 Embedded Code

The PIC's job is to collect sensor data and create a serial bitstream. In order to get all of the data needed, we had to coordinate all of the different features of the PIC: analog and digital I/O as well as A/D conversion. We did the programming for the control sequence with the Custom Computer Services, Inc. (CCS) C compiler (http://www.ccsinfo.com/picc.html). This saved much time since we could program in a familiar, high-level language. The reader may find it useful to refer to Figure 2-3 throughout this section.

In order to get the maximal data rate, we programmed the shoe to stream its data as fast as it can; the data rate was limited to 20 kbaud by the RF transmitter. To accomplish this, the algorithm for the code sits in a single loop that collects data, then creates the bitstream. There is also an interrupt routine to handle the sonar timing.

Measuring the analog values was relatively easy. In the loop, the appropriate multiplexer (MUX) address pins are first asserted and deasserted. After letting the A/D value settle (45μs in our case), we instruct the PIC to write the A/D output to the proper register.

A slightly more complicated process takes place for the tilt accelerometer outputs. To obtain duty-cycle information about the PWM output, we count in time increments much smaller than the period. The first step is to reset the counter values to zero. Then we look for a rising edge in the input. We count until the signal falls to zero and store that value in our data packet. For the $1ms$ period of the accelerometer's PWM output, we chose a time increment of 4μs. This also occurs in the main loop.

We encountered problems with the output of the accelerometer staying saturated and slowing down packet transmissions. To prevent this from happening, we added code that would time out of the counter loop. We set an upper limit to the value of the accelerometer and if the shoe counted up to the threshold, it stops counting and sends back a predetermined maximum value.

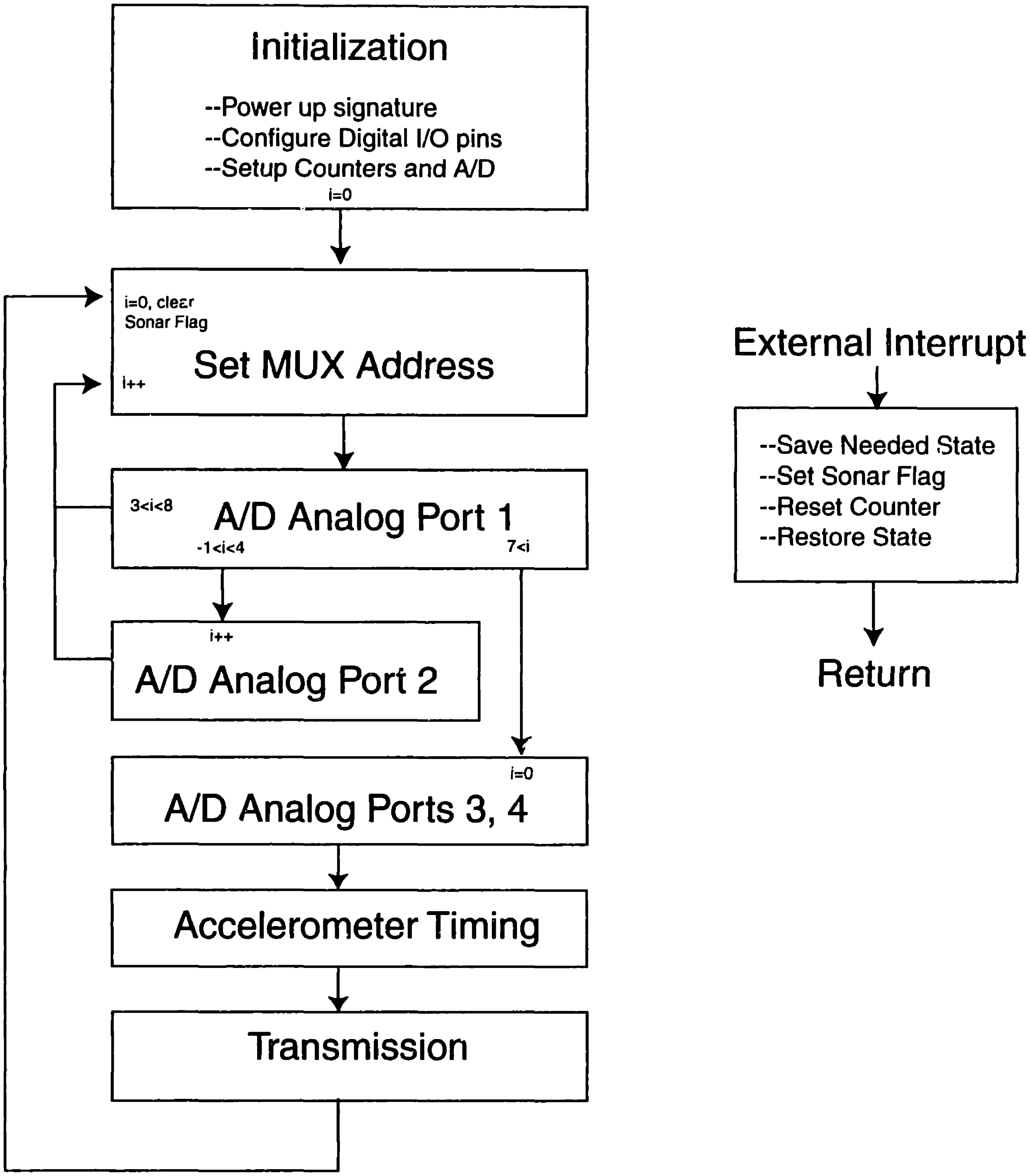


Figure 2-3: Flowchart of Embedded Code

The final piece of digital data is the sonar measurement. For this, we had to take advantage of the interrupt routines that come with the PIC compiler. The overall scheme begins with resetting a counter to zero in the base station (see Section 2.5) when a ping is sent. When the shoe receives the ping, it interrupts its current process to reset a counter of its own to zero. After waiting until the transmission of the sonar data byte, the shoe's counter value is saved and sent as the byte of sonar data. This represents the latency in the shoe.

When the base station receives the data, its counter value is the sum of two components. The first is the time-of-flight (t_f), the second is the time between the shoe receiving the ping and transmitting the data. This ping-to-transmission time is the value being sent back from the shoe, so we find t_f rather easily by subtracting the shoe's returned value from the base station's counter value.

This algorithm works well in theory, but we ran into a couple of problems in its implementation. The first, and simplest to fix, was an outright hardware problem. Only one pin of the PIC acts as a general interrupt. As a matter of chance, the sonar circuit was not connected to that pin. A simple swap of the sonar pin with the interrupt pin took care of this problem.

Another problem we ran into involved the CCS compiler. When our C code was compiled, the interrupt routines took anywhere between 30 and 40 assembler instructions. Because interrupts can be problematic, most compilers assume a worst-case scenario when dealing with interrupt routines, preserving all aspects of the PIC's state. Given the simplicity of our routine, however, the size, thus duration, of the assembled code was unacceptable. The problems these unnecessarily large interrupt routines caused were skewed tilt accelerometer data and faulty data transmission.

Both of these problems relate to timing. Since the interrupt can occur at any point in time, anything sensitive to timing would be adversely affected. If, while in the middle of a timing routine, a long (i.e. time-consuming) interrupt is triggered, the timing routine no longer returns reliable results. Both the tilt accelerometer's PWM output and the baud rate of the transmission line are very sensitive to timing.

The solution to this was to write large portions of our interrupt routine in assembly code. In our interrupt, we simply set a flag, indicating that a sonar ping has been received, store the status byte (S) and the accumulator value (W), reset the counter, then restore S and W. Storing and restoring S and W is necessary because resetting the counter is destructive to these values and the interrupt should be transparent to the routine it is interrupting.

One final problem is more fundamental. Sound travels at about 1ft/ms. If we want 50Hz update rates, the sonar measurement must take place within 20ms, which translates to 20ft. Our desired range of operation is 30ft, perhaps further. Linking our update rate to our sonar has the other undesirable effect of coupling our data rate to the location of the dancer with respect to the pinging station. The only way to obtain a 50Hz update rate was to not insist on a sonar measurement every transmission cycle. Therefore, the shoe transmits a zero for the sonar value every transmission cycle during which no sonar ping was received. This means that the base station receiving the data must store its count each time it receives a zero in the sonar byte, waiting for a nonzero byte before it resets its counter. At present, we are using four pingers on a station, all of which ping simultaneously. They ping at 10Hz, which is adequate for our applications.

2.2.3 Pressure

Again, we had to deal with two very distinct sets of signals for the two different kinds of pressure sensors.

The PVDF strip acts like a voltage source in series with a capacitor. This means that any load you put across the device produces a high-pass voltage across the pins. Any resistive load thus defines the cutoff frequency of the high-pass output. For the size of PVDF we were using, the capacitance is about 0.14nF. Our communications link allowed data packet update rates to approach 50Hz, meaning that we would like the analog PVDF output to be preserved for at least 20ms. The equation for the desired resistance becomes

$$R \geq \frac{t}{2\pi C} = \frac{20ms}{2\pi 0.14nF} = 22.7M\Omega$$

To be very safe, we used $40M\Omega$ shunt resistors on the output. In order to keep the impedance seen by the PVDF high, we also fed the output into a source-follower JFET circuit for current amplification.

The FSRs were a little more straightforward. We simply put the sensor in the bottom half of a two-resistor divider. To buffer these signals before the PIC A/D stage, we add a single stage emitter-follower BJT circuit. It turns out that 220Ω is a good operating value for the upper resistor in the divider for the Interlink FSRs, obtaining nearly the full dynamic range of 0-5V throughout the full range of foot motion. For the Flexiforce, $50k\Omega$ works well. These values were determined experimentally.

2.2.4 Bend Sensor

Initially, we were using the Capezio Dansneaker, which the sole underneath the arch of the foot cut away for easy pointing. Since the bend sensors can only sense bend in one direction, we used two back to back. Each of the two are connected to the lower half of a voltage divider. The voltages in the middle of the dividers were fed to a MAX492 (see Table A.2) in a differential configuration.

In a standard differential amplifier setup, the positive input terminal is connected through a resistor to the output of the voltage divider. A second connection is made through another resistor to ground. Since we were operating with a single-ended power supply, we had to bias up the ground connection to 2.5V, half our supply voltage to maximize our dynamic range.

2.2.5 Orientation

Three different sensors were used to extract orientation and angular rate, the low-g accelerometer, the compass and the gyro. Descriptions of the associated electronics follows.

The electronics for the ADXL202 were quite straightforward. One resistor was used to set the period of the duty-cycle output. The period is related to this resistor by

Period (sec)

$$= \frac{R_{set}}{125M\Omega}$$

Once the period is set, the device outputs a pulse wave, whose duty-cycle is proportional to acceleration. These outputs are sent to two digital pins (one for each axis) on the PIC. The counting measurement is detailed in the Embedded Code section above.

The electronics for the compass were quite complex. Initially, it looked as though the daughtercard that the compass was mounted on would provide all of the necessary electronics for the dynamic ranges of the outputs to be 0-5V. However, we ran into two major problems with the electronics provided. First, the power supply was rated at 6-15V and we were operating with a 5V source. Second, the sensitivity of the outputs was too low. It is nominally set to 1V/gauss. We wanted to measure the earth's field which is about half a gauss with our 8 bits running the full 0-5V.

After reverse-engineering the daughtercard, we found ways around both of these problems. We also discovered that the only thing on the daughtercard that needed the higher power supply were three instrumentation amplifiers. So, we simply replaced them with similar but more modern amplifiers (OP623 from Analog Devices, see Table A.2) that accept lower power supply voltages and run rail to rail.

Tracing the sensitivity problem led us to the amount of gain on the instrumentation amplifiers. So, to adjust the sensitivity, we pulled out the gain resistors they were using and replaced them to fit our sensitivity requirements. The original resistor value was $1.2k\Omega$. The expression for gain through a standard instrumentation amplifier is

$$Gain = 1 + 100k\Omega/R_G$$

We changed the value of the gain resistors to 330Ω. After these modifications, the output of the compass was always in the desired range.

The electronics for the gyro were trivial. A simple connection from the output pin to the multiplexer was all that was necessary. The dynamic range was about 4V over the ranges of angular change that we were interested in, so no further biasing, filtering or amplifying were necessary.

2.2.6 Rapid Foot Motion

The AMP accelerometer needed lots of signal conditioning. Measuring the output of the device with no external biasing, we saw a very low sensitivity with a DC offset, as this package only buffers the piezoelectric sensors with JFETs, giving no voltage gain. For measuring the true acceleration profile, this offset is very desirable since acceleration is generally centered around 0g. However, we wanted to use this sensor as a trigger for higher-g stomps, jumps and kicks. All of these motions exhibit high acceleration in both directions, so we pulled off the DC offset with a capacitor before amplifying the signal.

The accelerometer's sensitivity is around 2mV/g. We were looking to detect acceleration of at least 5g. If we wanted 5V to correspond to 5g, we would need gain of around 500. A single stage of gain that large would compromise the bandwidth of our signal too much. The output would not rise fast enough to keep up with the input. On the MAX494 opamps we were using, the gain-bandwidth product is 500kHz (see Table A.2). So, our output bandwidth would have been down to 1kHz. Stomps can generate pulses quite a bit shorter than 1ms. However, due to both space and power constraints, we did not want to add a second stage of gain. Instead, we reduced the amount of gain in the circuit, dropping it to 200, so that our output bandwidth would be around 2.5kHz.

After looking at the output, we were still unsatisfied. The MAX494 simply could not keep up with acceleration resulting from stomping. So, after shopping around, we found the OP462 from Analog Devices (see Table A.2). With a rail-to-rail output, low power requirements and 15Mhz gain-bandwidth product, we ended up with plenty of headroom (the output bandwidth now being 75kHz). What we also saw at the output was that 5g acceleration was not all that much for a stomp or kick. Though we never quite saturated the output, we were getting signals up to 4.5V. This was perfect, so we left the gain at 200.

Our next problem was our data sample rate. At 50-60Hz, there is no chance of catching a peak from a true acceleration profile. We needed to stretch our output to fit within our sample rate. For this we put a diode in series with a capacitor to make a peak detector. Also, we needed the output of the stretching circuit to decay, otherwise it would just provide the maximum accelerometer value over time. A resistor was placed in parallel with the capacitor to bleed off charge, slow enough for the PIC to sample the signal, but fast enough to bleed away within two or three sample periods. The time constant we settled on was 16ms, using a $1M\Omega$ resistor and a 0.1μF capacitor. An emitter-follower buffered this high impedance before the signal was presented to the PIC's A/D converter. Although this rectification lost the sign of net acceleration, it was necessary to insure reliable detection of transient events, such as jumps and kicks.

2.2.7 Absolute Position

In order to avoid contamination from acoustic or electric-field background, both of the circuits used here rely on tuning of some sort. The sonar relies more upon the intrinsic resonant tuning of the ultrasonic transducer (the sensor itself), whereas the electric-field sensing uses a tuned receiver circuit to detect the transmission from the electrode on the stage.

The sonar circuit goes through a few stages. When the signal comes off of the transducer, it is first biased, then passes through an active high-pass filter. This filter's purpose is to block DC gain in the amplification stage. Again, the signals are not long enough for PIC detection at this point, so a stretcher similar to the one used for the AMP accelerometer is used. Finally, this signal is fed to a comparator whose output is fed to one of the digital pins of the PIC. The stretcher circuit's function for the sonar is somewhere between an actual stretcher and an envelope filter. Though we are detecting a pulse, it is a wide pulse. Taking the envelope of the pulse also prevents the subsequent comparator from chattering, guaranteeing a good digital bit on the input pin of the PIC.

Initial biasing of the transducer is accomplished with a simple voltage divider. At the common node, changes in current generated by the sensor will adjust the voltage around the bias point. For maximal swing, we chose the bias point to be half our 5V supply, produced by a $1.1M\Omega$ and a $1M\Omega$ resistor, together with leakage through the op-amp and sensor. The high values of the resistors avoided loading of the sensor.

Active high-pass filtering was accomplished with an op-amp in a non-inverting amplifier configuration. The standard setup for such an amplifier has the input going to the positive input terminal of the op-amp and two resistors from the negative input of the op-amp, one to ground and the other to the output. By putting a capacitor in series with the resistor to ground, a high-pass response is created. The transfer function is given by

$$V_{out} = \frac{s(1 + R_1/R_2) + 1/R_2C}{s + 1/R_2C} V_{in}$$

For low frequencies, we achieve only unity gain and for high frequencies, our gain is $1 + R_1/R_2$. The 40kHz signal we are detecting fits in neither of these regions. Low frequencies (determined by $(R_1 + R_2)$C) for our circuit are less than 160Hz and high frequencies (determined by R_2C) are above 300kHz. The roll-off of the op-amp will attenuate frequencies much earlier than 300kHz. The gain at 40kHz, where we are operating, is very high, near the maximum capacity for this stage.

Following this, we have an envelope filter (again, almost identical to the stretching RC circuit as discussed in Section 2.2.6). Since we have the sonar ping driving a PIC digital input, the time constant on the envelope, 160μs, is much lower. Following the envelope filter is a simple comparator. When the input breaks a threshold, the output of the comparator goes high. The threshold on this comparator is made variable with a potentiometer. Since the sonar response varies with many different factors (e.g. pinging transducers, venue acoustics, etc.), the optimal threshold value will change. This generated bit drives the interrupt on the PIC and triggers the events detailed in Section 2.2.2 above.

The electric field sensor is slightly different. Before biasing the input, current flowing in from the electrode is bandpass filtered with a passive RLC circuit to roll off signals away from the electric-field transmit frequency. After the same biasing, high-pass filtering and envelope filter used in the sonar circuit, a final stage inverts, filters and amplifies the signal before passing it to the PIC, achieving the full 0-5V dynamic range. It should be noted that the RC on the output of the high-pass filter is truly an envelope filter in this circuit. We are no longer interested in a triggering pulse, but the continuous amplitude of the waveform which reflects the proximity of the foot.

Connected to the electrode are a resistor, inductor and capacitor in parallel to ground. The current to voltage (transresistance) transfer characteristic is given by

$$\frac{V_{out}}{I_{in}} = \frac{s(1/C)}{(s - s_+)(s - s_-)}$$

where s_+ and s_- take the plus and minus signs, respectively, of the expression

$$\frac{1}{2RC} \pm \sqrt{(1/2RC)^2 - 1/LC}$$

When we take $1/LC \gg (1/2RC)^2$, the poles of the system come very close to the $j\omega$ axis. This makes $\sqrt{LC}$ a very good approximation to the resonant frequency of the the filter in radians. We have a 2mH inductor with a 4.7nF capacitor, giving us about 52kHz as a resonant frequency.

Biasing in this circuit is also a little bit different. The output voltage of the tuning filter is connected to a capacitor in series with the input of the active high-pass filter. This eliminates any DC signal relative to the input node, which is biased with a separate 2.5V source, created with a voltage divider and connected to the high-pass filter input via a $1M\Omega$ resistor, preventing any loading on the filter's feedback network.

The active high-pass filter is almost identical to the one used by the sonar transducer. Only the gain resistor, R_1, changes. This is due strictly to the natural phenomena that govern the magnitude of our input signals. The envelope filter is also very similar. Its time constant is even smaller than before (around 8μs). This makes our next stage (an inverting low-pass filter) cleaner.

The subsequent stage starts once again with the non-inverting amplifier circuit. Two changes are made, one to invert and one to filter. A resistor from the negative input terminal of the op-amp to our power supply (5V) strips the intrinsic 2.5V bias from the signal and gives a gain of two, allowing access to the full 0-5V range. By putting a capacitor in parallel with the gain resistor, we create a low-pass response to further smooth the signal.

The inverted output is desirable due to the inverse relationship between height and field strength. Naturally, as the foot moves higher, it is farther away from the transmission electrode. The strength of the signal is, therefore, smaller. Inverting the signal through the low-pass filter compensates for this.

For both of these circuits, the initial active filtering was done with MAX474s (see Table A.2). The electric-field signal's final stage of amplification used the second of the dual op-amps on the MAX492 used for the bend sensor.

2.3 Communication

In order to make the shoe system as unobtrusive as possible, we had to make the communications wireless. A bus of wires to the dancer would have been more like a leash than a communications port. After we settled on the appropriate hardware, we devised a communications protocol.

2.3.1 Hardware

In looking for a wireless system, our main objective was size. We were looking to minimize the size of any part that was to be attached to the shoe. However, we also wanted to minimize power and maximize data rate. We found a good combination of these things in an RF transmitter/receiver pair made by Radiometrix [4]. It has a maximum data rate of 20kbps and consumes at most 10mA of current and delivers 0.25mW of radiated power. The transmitter (TXM-4xx-F-3) is also very small, measuring about 1cm x 3cm. The part comes in three different frequencies, of which Expressive Footwear uses 418MHz and 433MHz, one per shoe. Radiometrix also makes a 403MHz version for foreign use as well as transceivers at the two frequencies the shoe uses. Streaming at these frequencies is not legal in the United States. We are currently researching other protocols, including channel-sharing schemes. Such protocols would also allow more than two transmitters to be used at the same time within the same range.

Another attractive feature of this RF module is the simplicity of use. Once the module receives the appropriate power supply, the bitstream generated by the PIC (see Section 2.3.2 for bitstream information) can be fed directly into the input pin of the module. The module's operating range is 30-50m, depending on the antenna used and the local RF environment. Under optimal conditions with a whip antenna, ranges of 400m have been achieved.

One minor problem we had with this module was the power supply. The different models of the transmitter operate on either a 2.7-3.6V supply or a 6-9V supply. Our 5V supply needed to be adjusted before we powered the transmitter. For this, we used a micropower series regulator from Maxim. It takes our 5V supply and regulates it down to 3V. This addition is rather minor, given the simplicity of the RF module.

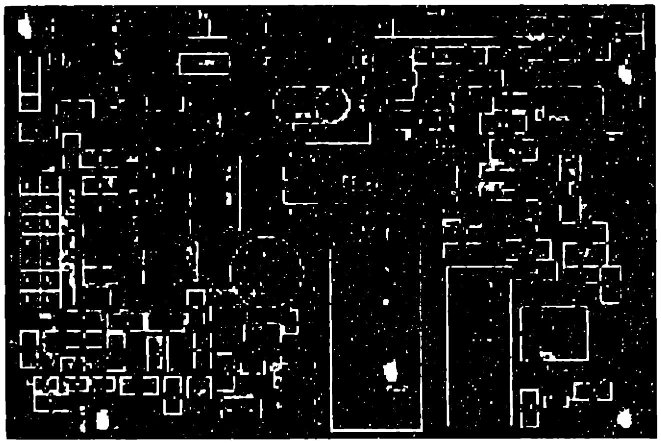


Figure 2-4: Shoe Circuit Board (actual size)

2.3.2 Software

The software for our communications system consisted mainly of devising a workable protocol for data packets. There were two main design issues, framing our data packets and zero-balancing our data stream.

The data packets are framed with a start byte. A potential problem with this method is the loss of states. In order to generate a unique byte, that byte must be removed from your possible data values. With 8 bits, however, you have 256 values. Sacrificing one for packet framing was a reasonable trade off for us. We actually have two different possible start bytes. This way, we can encode a single bit of information in our start byte while still retaining 254 possible values for our data. The bit encoded comes from our 5V regulator. It has a battery-low flag that can be useful in diagnosing problems during performance. The two values, 255 and 254, represent battery-ok and battery-low respectively. We have the regulator (see Section 2.4.2) set to indicate a battery-low at 4.5V. On the receiving end, you can just look for these two values on the incoming line and then consider all the following bytes as data.

In any RF communications scheme, there is a need for zero-balancing. The only way a receiver can distinguish between two signals is by comparing the demodulated incoming signal to a reference level stored internally. It generates this value by taking an average of the received values. This method works, provided that the incoming signal has a duty-cycle close to 50%. Just before the shoe transmits, it initializes the receiver by sending a 'U', whose binary representation is alternating ones and zeros. This helps to fulfill the necessary condition for the receiver to properly distinguish the bits as ones or zeros at the beginning of the data stream. As the values get transmitted, the receiver's internal voltage level is skewed. In general, the direction and amount of skew is unpredictable. To keep the receiver's voltage level accurate, we also send the complement of the full data packet with each transmission, neutralizing the skew.

We also developed a few software routines for robustness. A couple of causes of bad packets kept arising in the early stages of development. One of the bytes we send is the regulated 3V power supply. This signal serves as a more advanced battery-low than the single bit indicated in the start byte. In addition to monitoring the power levels on the shoe, sudden disturbances in this signal indicate bad packets. Our code currently checks this and the AMP accelerometer before sending the data to music software. Spikes in the accelerometer that are not accompanied by an exponential decay also indicate bad packets. In future versions, we will probably also send a checksum (e.g. the total number of 1's in the packet) for additional error checking.

Another check would take advantage of the complementary bytes in our data stream used for zero-balancing. The base station (discussed below) could simply compare the true bytes with the complementary ones. A bitwise XOR that returns anything other than all ones would indicate a corrupted data packet.

2.4 Power

To make the system wireless, a battery was used as the power supply. For stability at our desired level of 5V, we used a regulator. One additional stage of regulation was used to obtain the 3V supply needed for the transmitter.

2.4.1 Batteries

In the area of power storage, there is an inherent trade-off between size and capacity. The technology driving the semiconductor industry that made the miniaturization of the rest of Expressive Footwear possible is scaling down much faster than the charge density of materials used for batteries.

Initially, in the interest of size, we selected with a small, 6.2V lithium camera battery (200mAh capacity) that was attached to the sensor card. Lithium batteries drop off very sharply in time. That is, they use up most of their charge and then sharply degrade, maximizing the use of charge above a given threshold (in our case, 5V). The shoe draws roughly 50mA continuously, giving us at most 4 hours of operation. In practice, we saw just under 3 hours of operation. The principle reason for this is that we did not run the batteries to their full capacity. As soon as the voltage dropped significantly below 5V, we would replace the battery. This is not strictly necessary as most of the sensors are ratiometric, but we did not want the battery to die in the middle of a performance. The battery loading also increased with activity. For example, the current being drawn by the FSRs (toe pressure sensors and bend sensors, see Section 3.1.1) and the current being drawn by the op-amps varied with sensor stimulus.

Even at 3 hour intervals, switching the batteries became very cumbersome and inconvenient, so we moved the battery off the sensor card and onto its own housing behind the heel. This allowed larger batteries to be considered; and we eventually decided to use standard 9V batteries. Both alkaline and lithium batteries have between two and three times the life of the 6V batteries. Moving the battery off the sensor card also gave us more room on the card to put electronics. The second multiplexer and circuitry for the toe sensor on the tip of the shoe ended up using this space.

Of course, a larger battery needs a very strong housing. We had many different 9V battery housings to choose from. To minimize weight, we picked a plastic case. The lower side of the case we chose had a lip, which grabs onto the bottom of the battery, which often also has a lip. Unfortunately, the lithium batteries we chose had no such lip and had a smooth, metallic bottom as well, eliminating any friction that may have helped to hold the battery. We ended up using wire-ties to hold the battery in place, especially important when the dancer leaps and kicks, as we learned with experience.

With the 6.2V lithium camera batteries, there was a significant danger of placing the battery into its holder backwards. To protect the sensor card against such mistakes, we put a 1N4001 rectifier between the two power terminals, with the negative end on the positive end of the power supply. This way, if the battery were placed in backwards, current would flow through the rectifier rather than in the reverse direction through all of the components. For the 9V batteries, this is not such a problem, but we left the diode in for safety's sake.

2.4.2 Regulators

To regulate the different batteries down to 5V, we used Maxim's MAX883 low-dropout series regulators. It can source up to 200mA of current and keeps its output to within 0.25V. Quiescent current is held at 11μA and its very low dropout occurs at 0.11V. These specifications are summarized in Table A.1.

Additionally, it has the capability of setting a battery-low flag (LBO). This flag goes high when the low battery input (LBI) drops below 1.2V. We control the voltage at LBI by using a voltage divider. To obtain a 4.5V threshold, we set the top half of the divider to be 270kΩ and the lower half to be 100kΩ.

For the transmitter's 3V supply, a similar Maxim chip, the MAX8873 was used. Its voltage accuracy is 0.1V and can source 120mA. With no load applied, it draws 73μA and has an 55mV dropout level. These specifications are also summarized in Table A.1.

We could have saved significant power drain by using a switching regulator when we moved to the 9V source. However, we were getting enough operational time (about 6 hours per battery) with the series regulator, thus we elected not to change the layout to incorporate the switcher and its associated components (e.g. inductor). In addition, switching supplies produce significant amounts of electrical noise, which may contaminate some of our high-gain sensor signals.

2.5 Base Station

The base station serves a few different functions. For communication, it translates the data coming off of the RF receiver into an RS-232 compatible signal. This is done with a PIC looking at the output of the receiver and a line converter, a very standard MAX233. The PIC on the master base station also controls triggering of the sonar pingers. A separate module in the base station generates the 50kHz signal required for electric-field sensing.

2.5.1 Data Conversion

There are two stages to our data conversion process. First, the PIC monitors the bitstream coming off of the RF receiver until it sees a coded start byte (255 or 254). Since our data packets are fixed in length, the PIC can just write the start byte and the next 17 bytes (16 physical parameters and 1 to monitor the shoe's power supply) of the input stream to the RS-232 converter.

The second stage is a simple line conversion from the TTL levels (0V for low and

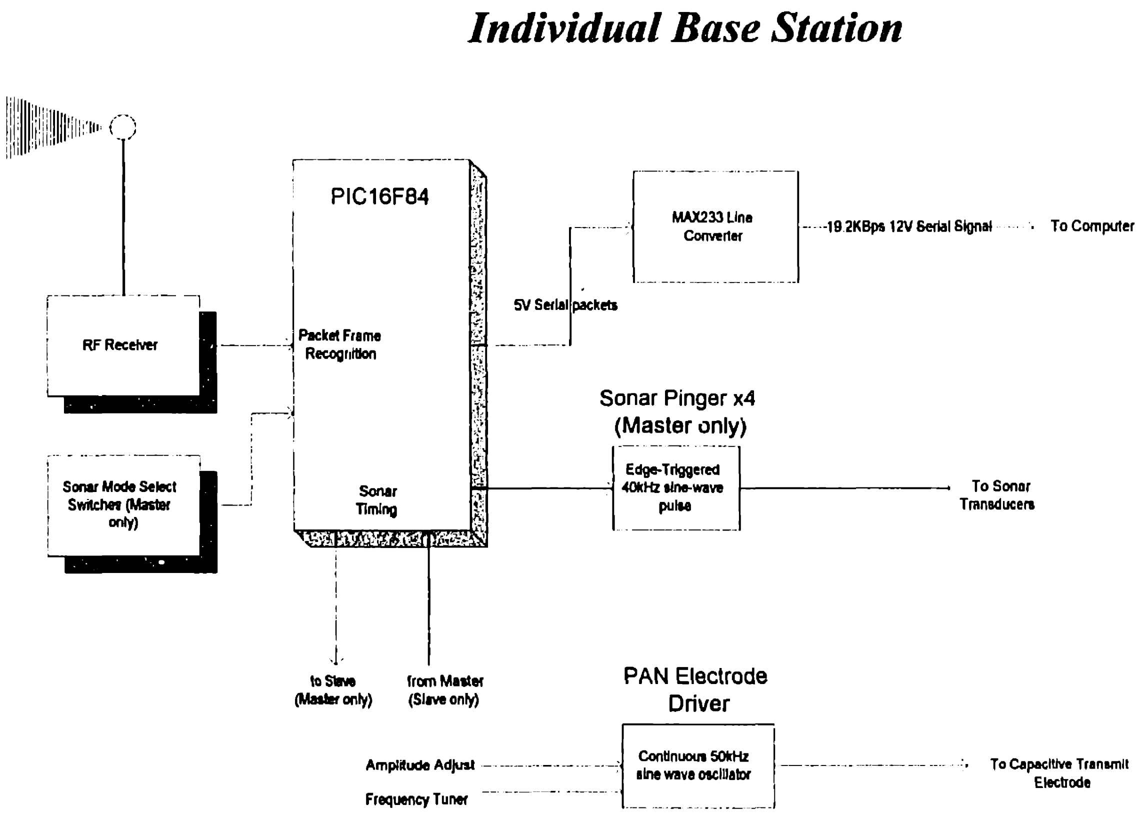


Figure 2-5: Block Diagram of the Base Station

5V for high) of the PIC to the RS-232 levels (12V for low and -12V for high) that a standard PC serial port accepts. This line conversion is accomplished with a MAX233. The base station also does the handshaking required by an RS-232 compatible serial port.

2.5.2 Sonar Pinging

Hardware

The signal that triggers the sonar ping is a digital low-to-high transition. A 555 timer chip takes this single clock-cycle pulse and creates a wider pulse. This is fed as the gate to a sample-and-hold amplifier (LF398) acting as an analog gate, whose input is a 40kHz square wave. Finally, a pair of high current-output op-amps buffer and provide gain for the square wave, creating a differential signal for the transducer.

Since the trigger signal comes from the PIC (0-5V), our first step was to add a transistor as a saturated switch to boost it up to the 12V supply of the 555. Also, the 555 trigger is an active low signal. To get both the inverted signal, we setup the transistor in a grounded-emitter configuration. The output is fed to the 555 trigger input.

When the trigger goes high, the 555 begins charging a capacitor through a resistor. While the capacitor is charging, the 555's output is held high. When the capacitor voltage has reached two thirds the supply voltage, the chip is reset and the output returns to the low state. The equation for the pulse width is given by $t = 1.1RC$. In order to make the pulse approximately 1ms wide, we used a $56k\Omega$ resistor and a 0.015μF capacitor.

The pulse generated by the 555 timer chip controls the gate of the LF398. Though this is a sample-and-hold amplifier, we have configured it strictly as an analog gate. When the gate is high, the input is seen on the output. With the placement of a capacitor, the input value can be held when the gate goes low. However, we simply wish to pass the 40kHz signal at the output, so we leave out this capacitor, grounding this pin through a resistor so the "off" voltage is quiescently at ground.

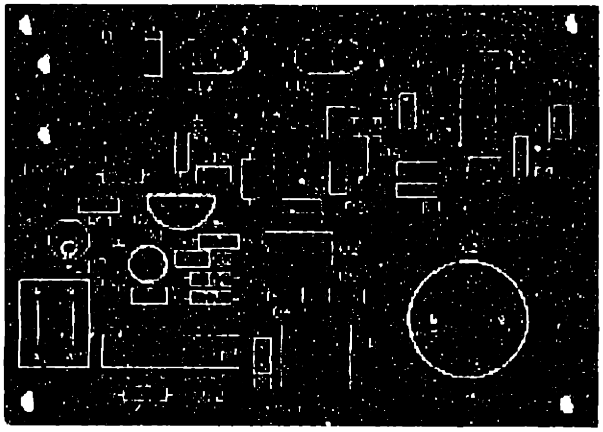


Figure 2-6: Sonar Pinging Board

To generate the input signal, we use a 40kHz crystal. Connected across a digital inverter, the crystal makes a ring oscillator and a square wave is produced. To avoid a direct connection from the crystal to the output, a second inverter stage buffers the 40kHz square wave before it enters the LF398. A single stage of passive high pass filtering (a simple RC with low frequency cutoff at about 200Hz) centers the previously single-ended signal around 0V. This filter also gives us the opportunity to use a potentiometer in place of the resistor to adjust amplitude.

The output of the LF398 is then used to drive two different amplifiers with the same gain, one in an inverting configuration, the other in a non-inverting configuration. Because the Q of the piezoceramic transducer is high, a 40kHz sine wave is transmitted. Both amplifier configurations also give about a factor of 2 DC gain to increase the range of the transducer. Because the drive is differential, a net gain of 4 is produced, giving us 24V peak-to-peak across the piezo element. The TLE2082, a high-current output op-amp, is used for the driver stage.

Software

Most of the software involved with counting out the sonar value was described in Section 2.2.2. One added feature needed to accommodate two asynchronous shoes is the use of a master base station and a slave base station. The master station is physically connected to the pinger units. When it pings, it also sends a pulse to the slave base station so that it knows when to start counting. Since the data cycle of the two shoes will be different, each base station needs to take its own sonar measurement. This arrangement allows separate, asynchronous sonar measurements at each shoe.

The master base station also contains two switches that configure the sonar pinging. Four modes of pinging are selectable, pinging one station at a time sequentially, pinging two at a time in sequence, pinging all four at the same time and pinging just one station. Sending the ping is simply a digital pulse for the PIC, so the code for these different modes just selects different output pins to assert (each pin triggering a different pinging station) for a single clock cycle. Currently, we have the pings occuring from a single station once every 5 transmission cycles. As mentioned above, when a ping is sent, the master base station also needs to trigger the slave base station. In addition, the master sends an additional code in the sonar byte (0-3) to indicate which output gave the last ping. This byte is sent when there is no valid sonar data. As a result, the allowable sonar range is now 4 to 255.

2.5.3 Electric Field Generation

To generate the electric field on the stage, an electrode is connected to a base station, driven by an oscillator. In order to get optimal performance in different working environments, we made both the frequency and the amplitude of the oscillator adjustable. This is done with two potentiometers at the base station, where all of the other electronics reside. Once the signal is generated, a wire is connected from the base station to the electrode, where the appropriate electric field is transmitted. At the heart of the oscillator is an 8038 function generator chip. It can generate a square wave, triangle wave or sine wave with adjustable symmetry and frequency (up to 100kHz). Our needs only demanded a symmetric, 50kHz sine wave.

The symmetry of the waveform is determined by two resistors. By making the two resistors equal, we achieve a symmetrical waveform. The frequency is determined by four different inputs to the 8038. Both of the resistors that determine symmetry contribute, as well as a timing capacitor and the voltage seen at the FM pin. Since our symmetry resistors were equal, the frequency of our output waveform is given by

$f = 0.33/RC$, where C is the capacitor mentioned above. We also had to consider the amount of current going into the symmetry circuit. It is best to have this in the range of 10μA to 1mA. For our supply voltage (+12V), the current is 2.5/R, so we selected 10$k\Omega$ as the value of each symmetry resistor. To give ourselves plenty of headroom, we set the capacitor to 270pF, yielding a 120kHz maximum frequency. From the FM pin, we have the freedom we need to adjust down to the desired 50kHz. The voltage at the FM pin is set with a potentiometer connected to the power rail in series with a resistor to ground.

The output swing coming directly off of the 8038 ranges from 4V to 8V. For current gain (or, equivalently, to lower the sine wave's output impedance), a JFET source-follower was used. With too much loading, the quality of the sine wave can be adversely affected. Voltage increases are achieved by AC coupling the JFET's output to an op-amp (±12V supply) in an inverting amplifier configuration. A 100$k\Omega$ potentiometer is put in the feedback path to adjust gain, allowing the output to range up to ±10V. The input resistor is 13$k\Omega$. Finally, the electrode is connected to this output via a 330Ω buffer resistor, transmitting the desired electric field.

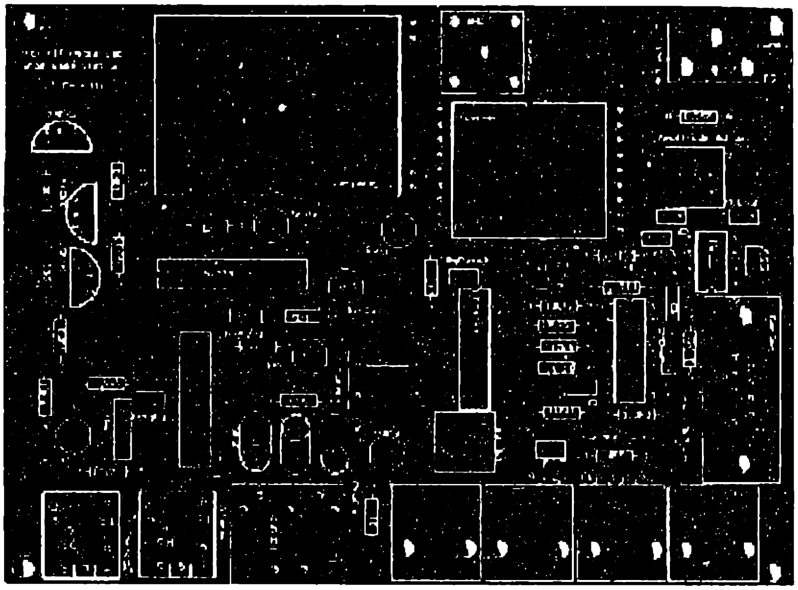


Figure 2-7: The Base Station Circuit Card

Chapter 3

Applications

Acquiring measurements and data from the shoe is a challenging task, but more important is how we use it. In this chapter, the interpretation problem is discussed. Since interpretation is such a broad topic, only a relatively brief discussion of how interpretation relates to the shoe is given. It is followed by an outline of the music applications that we created. Our most recent work with Hidden Markov Models (HMMs) is also included.

3.1 Music

Our original intent for Expressive Footwear was for interactive dance. Because of this, the first application we created for the shoes was a musical application. It used a set of rules and conditions to "map" gesture onto synthesized sound. Other possible applications for dance might vary different aspects of the visual stage presentation or somehow combine visual and aural effects.

Many different versions of the music were created. This was reflective of the different versions of the shoe as they were revised and improved. What follows outlines the different mappings and what version of the shoe they were associated with.

Figure 3-1: YuYing Chen with an Early Version of Expressive Footwear

3.1.1 Dancers and Initial Mappings

Yuying Chen

In our first version of the shoe, Yuying Chen (see Figure 3-1) was our dancer. At the time, Chen was a student and fellow Media Lab researcher. This was perfect for us then; as a student, she was open to the experimentation of the early stages and, as a researcher, had the patience to bear with us through debugging. Also, since she was also at the Media Lab, collaboration was straightforward.

Chen wore the shoes for their premiere at the Wearable Computing Fashion Show at MIT in October 1997. That initial version of Expressive Footwear (see Figure 1-2) was a single shoe, rather than a pair. Some sensors had also not yet been implemented and only one base station was completed. We were still using the two-axis, mechanical compass and the pressure sensor at the tip of the shoe was not present. Also, all of the pressure sensors at this time were PVDF, measuring dynamic pressure. That is, we were not getting steady-state pressure measurements from FSRs yet. The embedded code also had not yet developed sufficiently to include the sonar measurements. At this point, we were using the E-Mu Morpheus as our primary synthesizer.

The mapping that Chen worked with was a very simple one composed by Kai-yuh Hsiao with "Rogus McBogus" [3] (discussed below). There were three voices in the music, a drum line, a bass voice and a melody voice, along with a few miscellaneous sound effects. The drum line was continuous and its rhythmic elements were only slightly controlled by the dancer. Small changes in tempo could be brought about by the bend sensor. An industrial, techno beat was used to match the rest of the Fashion Show. Both the bass voice's volume and the bass drum's volume were controlled by the pitch angle of the foot position (θ in Figure 2-2). The volume of the remaining drum voices' was controlled by the electric-field sensor [9].

Both the bass and melody voice would be switched off as the AMP accelerometer picked up high-g impulses. This made sonic room for the sound effect of an explosion, also triggered by large amplitudes returned from the AMP accelerometer. When an impulse was seen from the piezo sensor in the heel, the bass would shift octaves. The bass voice was not strictly a bass line, but included harmony as well. Yaw rotation (ψ as depicted in Figure 2-2) selected among different harmonizations. Quick rotations also triggered another sound effect, this one of sweeping wind. The melody was in an upper register and the pitch range of the notes for the melody were dictated by the pressure sensors under the ball of the foot. Both the bass and melody voice would pan and flange depending on the compass value [11].

Mia Keinanen and Brian Clarkson

Mia Keinanen (see Figure 3-2) was the first to wear a complete pair of working shoes for Expressive Footwear. Like Chen, she was also both a student and researcher at the Media Lab. She brought all of the same positive attributes to our team. The version of Expressive Footwear she performed with had all of the sensors, including the pressure sensor on the tip of the toe and the 3-axis, solid-state compass. Measurement

Figure 3-2: Mia Demonstrating an Recent Model of Expressive Footwear

of the sonar values had still not yet come online and, due to concerns about shielding of the earth's magnetic field in our operating space, the compass was not used.

The most significant change we made to the system that Keinanen worked with was the removal of the continuous techno drum track. While this gives the dancer much more freedom and flexibility (especially with regard to tempo), it also gives much more responsibility since the dancer must create rhythmic regularity where it is desired.

Melodic notes were still controlled by the various pressure sensors in the toe and heel. With no tempo to change anymore, we instead used the bend sensor over to selecting the underlying harmonic chord. This harmony's volume is changed by the value of θ (see Figure 2-2) from the tilt sensor. The other tilt value (ψ in Figure 2-2) faded a second chord's volume up and down [10].

We kept the bass voice, but used it as a pedal tone for this mapping. Its volume increased as the shoe got closer to the transmitting electrode of the electric-field sensor. The high-g movements detected by the AMP accelerometer still triggered sharply attacked sounds, this time taken from percussion sounds in the synthesizer rather than explosions. Values from the gyro were mapped to a string of piano notes, providing a nice coloristic contrast.

Figure 3-3: Brian Clarkson at the Tokyo Wearable Fashion Show

Our next effort was at the Tokyo Wearable fashion show. Here, MIT gymnast Brian Clarkson (see Figure 3-3) demonstrated a shoe with a different mapping. The AMP accelerometer was used to produce orchestra hits and change the key of an ongoing sequence. Notes were also triggered by the pressure sensors and runs of notes were launched by the the gyro. The compass crossfaded melodic notes and when the tilt accelerometers detected a handstand, a drum roll would ensue with a crash on impact. This mapping formed the basis for the mapping we created with Byron Suber of Cornell University.

Byron Suber

After settling on a final version of the shoe with all of the sensors shown in Figure 2-2. Suber is a choreographer at Cornell and works with many different types of performance that incorporate recent technology. The mapping we created with him is discussed in detail below in Section 3.1.3.

3.1.2 Code

Before any kind of mapping could be developed, an infrastructure for the code had to be written. Our first goal was writing code to collect the data off of two separate serial ports (one for each shoe). The PC we used was configured for data acquisition and contained a Rocket Port, which added 8 serial ports to the two that come with the motherboard. Microsoft Visual C++ comes with a variety of libraries, one of which provides the classes necessary to handle incoming data on the serial ports. This section of the code is rather straightforward and simply requires knowledge of the function arguments within the class.

As it turns out, the code for the second serial port is very similar to the code for the first. The only changes that need to made are the arguments to the previously mentioned class functions. In particular, each instance of a serial port object must be assigned to a different port. The other arguments, which affect baud rate, flow control, etc., were not altered for the shoe system, although for other applications, such alterations can be made rather easily.

The next step we took was to create some data checking routines that would recognize bad data packets and keep them from entering the music creation code. Though our protocol of recognizing start bytes at the base station is effective, it is not guaranteed. Occasionally, due to RF noise in the ambient environment, a false start byte (one that was not sent by the shoe) at 433MHz or 418MHz will appear on the receiver. These are easy to recognize from the abnormal length of the remainder of the transmission.

Bad packets also result from battery connections that are loosened by the activity of the dancer and from poor placement of the base stations. Monitors, for instance, are a large source of RF noise. When base stations are placed very near them, false start bytes can be generated. The power glitching that results from unreliable power connections is actually taken care of in the shoe. When the PIC on the shoe is powered up, it sends alternating saturated-high and saturated-low values on every byte of the first several data packets before moving into normal operation. This is easily recognized by the computer. The AMP accelerometer's post-signal conditioning output also serves as a good "glitch detector." If, after a rapid increase, the signal does not die off exponentially, we know we have a bad packet since the time constant on the stretching circuit (see Section 2.2.6) does not change. Again, error detection packets would increase the reliability of data transmission. For our applications, however, the added reliability was not necessary.

The last step before deciding on the actual mapping was to create objects and structures that make quick transitions from data to music. For this, we took advantage of the an existing library of classes and functions developed at the Media Lab known as Rogus McBogus [3]. These libraries were designed for the Brain Opera [8] and serve as an interface between raw data and MIDI commands. Though MIDI commands should never be mistaken for music, the interface Rogus McBogus provides is very convenient because the vast majority of commercial synthesizers take MIDI commands.

The audio demands of the system included two synthesizers, a mixer and a pair of powered speakers. Our application sent MIDI commands through the sound card of the PC to the synthesizers. For the Clarkson and Suber mappings, we chose the Korg N1R and the E-mu Orbit as our synthesizers.

3.1.3 Professional Dance Mapping

Our final mapping of sensor values to sounds ended up creating an interesting set of dependencies. We started by grouping sensor values into like categories. For instance, we thought about the pressure sensors as one group of values distinct from the others. The three values of the AMP accelerometer were similarly grouped. As we worked with Suber, it became clear that such independence was not ideal. The following mapping was a result of our collaboration with Suber and incorporated a choreographer's perspective. Each sensor includes a description of the direct mapping of the sensor to sound as well as that sensor's dependence on others.

Pressure and Bend

Because the front of the foot is an area of potentially very delicate control, the two pressure sensors under the ball of the foot control rather high melody notes. These notes are randomly selected from a predetermined set. The mapping we made with

Figure 3-4: Byron Suber Dancing with Our Most Recent Mapping

Suber used picked from a set of five notes: tonic, dominant, subdominant and both the major and minor third. This note selection process was also used for the heel sensor. However, since the control there is not quite as delicate, the notes are an octave lower.

For the particular shoe that the dancer requested (Nike Terra Kimbias), pointing the toe was not very feasible. This means that most of the time the bend sensor is showing a non-zero value, the dancer is flexing their foot and very likely standing up on the balls of their feet. For this reason, it was very natural to have the bend sensor control the octave of the melody notes. As the bend increases, the octave increases in pitch. There is the potential of a three octave increase triggered from the bend sensor.

One problem with this setup is that the melody notes can never be turned off. Even when the dancer is standing still and wants to create a more serene tone, the melody notes continue to create a sparkling effect. In order to give the dancer the ability to turn the melody off for periods of time, we link the melodic activity to both shoes. That is, if only one shoe's toe sensors are active, the melody notes are switched off. Both shoe's toe sensors need to be active to get the melody notes. The dancer thus has two primary methods of turning off the melody, either by standing on one foot or standing sufficiently still on both feet.

The last pressure sensor right at the very tip of the toe was added for kicks when the toe is pointed down and pressure against the toe of the shoe. A common dance step is to kick or press the floor with the tip of the toe in an area just behind the dancer. This type of kicking is quite controlled and rather different from the high-impact kicking the AMP accelerometers would detect. Still, it is a sharp motion, so we assigned a cymbal crash to that sensor. When the value breaks a threshold, the cymbal sounds.

We also control the volume of the melody notes triggered by the toe sensors with the activity of the sensors. The notes are soft when the sensor values are static and loud when the values are changing a lot.

Gyro

The gyro proved to be one of the more compelling sensor-to-sound mappings. Successful mappings relate the dancer's intuitive sense of the motion detected to an aural quality that elicits the same intuitive reaction. In the case of the gyro, we associate the spinning with a glissando. The speed and direction of the glissando are directly related to the gyro values. For example, a quick spin to the right might initiate a loud, upward glissando while a slow turn to the left would initiate a soft, downward glissando. On one foot, a string ensemble sound is used and on the other, a bell-like sound is used.

Compass

Once we break the stage into different sections based on absolute position measurements, we can create multiple uses for the compass. This partitioning of the stage is further discussed in Section 3.1.3. The use for the compass in this context is also described there.

When the compass is not in the zone over the electric-field sensor, it has a rather intuitive function. Its axes are used to determine the spatial characteristics of some of the other sounds. Our setup only uses two channels, left and right; and the relative volume of each channel is adjusted based on the compass readings. With additional speakers and audio channels, more sophisticated spatialization can be performed.

In the small area that we were working in (roughly 8' x 8'), this effect was barely noticeable. However, in a large stage production, very interesting effects are possible. For instance, with speakers set up on the extreme left and right sides of the stage, the dancer could play music, in effect, for just one side of the audience or the other. Emphasizing what you are doing to the people you are facing is a very natural human idea and integrates very well into the dancing scheme.

Accelerometers

Our interpretation of the ADXL202 output was mainly as a tilt reading because of its low-g operating range. This shoe value was used to control the background of the music. The notes controlled by these tilt parameters were rather high in pitch, but are quite fleeting compared to the notes controlled by the pressure sensors, both temporally and timbrally. An airy, flute-like sound is used by the tilt accelerometer rather than the spongy, bassoonish sounds controlled by the pressure sensors. Since we have two axes, one is used to control the volume of these notes and the other controls the frequency of notes. In order to avoid too much conflict with the pressure sensors, the volume is loudest and number of notes most frequent when the shoe is tilted such that the pressure sensors are disengaged.

Assigning a sound to the AMP accelerometer was quite easy. Both abstractly as an engineer and intuitively for the dancer, kicks and stomps are associated with short, pulsing, chaotic sounds. We ended up with a fairly high orchestra hit for one shoe and used a sound similar to those associated with the laser blasters from the movie Star Wars for the other.

Absolute Position

One possibility we recognized from the start was the partitioning of the stage into different zones depending on the absolute position readings. It would be possible to use each of the sensors in a completely different way for each of the regions. This lends a great deal of expressive flexibility, as well as taking advantage of perhaps the most basic of human intuitions, the idea of where an object (in this case, a dancer) is in space. Explorations to this end are currently being made.

Using both the sonar and electric-field sensing, we created a couple of different regions, one close to the sonar pinging station, one over the electric-field sensor and one which was neither. The physical bounds of these regions overlap, which forces us to do one of two things. We could either have the effects of the different zones blend or have one take precedence over the other. In order to create distinct zones, we chose the latter. With control over multiple pinging stations, it is possible to have many regions. If a setup with distinct zones is desired, a more complete hierarchy would need to be worked out.

For the sonar region, we wrote a drum track. The playback volume is determined by the sonar value calculated at the base station. As you get further away, the drum track fades out. In an otherwise unrhythmicized mapping, this feature stands out quite well. It is the only time the dancer can remain absolutely still yet maintain an active soundscape.

With the shoes typically so close to the ground, the electric-field sensor acted more like a switch, turning on when the dancer was over the electrode and turning off when the dancer moved away from it. Though we extracted height information from one of the shoes while the other was on the ground over the electrode, the electric-field sensor was very effective at being a zone boundary.

Outside the zone, all of the music continued as described. Inside the zone, however, we turned all of the sound off except for two chords. We associated a different chord to each of the shoes. One was a major triad, the other a dominant seventh without the fifth. Naturally, the dominant seventh, in the context of so many diatonic notes, sounded very unstable and the reaction from the people present reflected this.

Because of how we control the volume of the two chords inside the zone, they do not usually sound simultaneously. The volume of the chord is controlled by the height of the other shoe over the transmitting electrode. Compass readings are the only other values controlling the sound at this point. They are used to control the relative volumes of each of the notes in the chords.

Expressive Footwear was shown at the International Dance and Technology 99 conference [10].

3.2 Gestural Recognition

Though gestural recognition was not taken advantage of in our musical application, it has been successful in other applications of hardware derived from Expressive Footwear. One of the original papers on this can be found in [1], while the most recent developments are in [7]. We have also done some preliminary work with gesture recognition via both cluster-weighted modeling [5] and Hidden Markov Models (HMMs), though it has not yet been integrated into our musical application. The work with cluster-weighted modeling successfully used the shoe to distinguish between a waltz and a tango. Here, a description of the HMM algorithm and some possible uses are discussed.

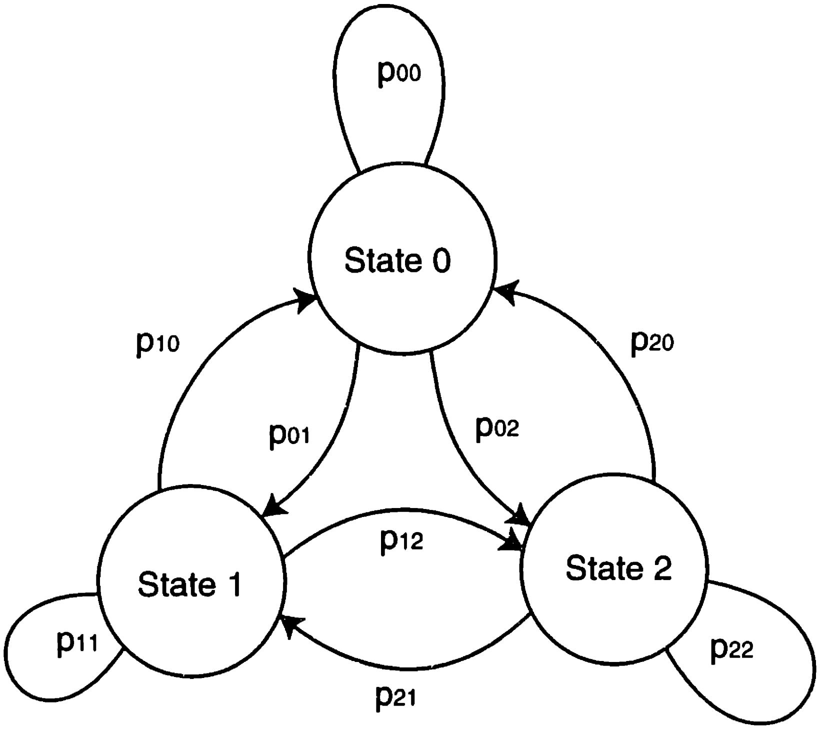
3.2.1 Hidden Markov Models

Hidden Markov Models try to infer gestures from a set of observations. A Markov process is one in which the present state is dependent, in a probabilistic sense, only on the state immediately preceding it. That is, when a particular state is reached, the probability of transitioning to a given state is not changed by knowledge of the path taken to get to the state. A generic Markov model describes the states and transition probabilities.

The reason our particular problem is "hidden" is that we do not have access to the actual state we are interested in. Specifically, we wish to deduce the dancer's gestures. There is not currently any sensor that can do this directly. So, we do the next best thing and try to infer gestural information from data that is well-correlated to the gesture of interest. Thus, each HMM is associated with its own gesture and returns a scalar value representing the probability of that gesture. In our case, the data is coming from Expressive Footwear. A brief introduction to HMM theory is given below, followed by some of the specifics of our implementation. Much of theory is taken from [13].

HMM Theory

There are two phases to the hidden Markov modelling problem. First, there is a training phase, in which we try to determine the state transition matrix. For such a matrix, the ij^{th} entry is the probability that a transition from the i^{th} state to the j^{th} state happens in consecutive samples (see Figure 3-5). Typically, such matrices have fairly large diagonal components and small off-diagonal components. This is a result of the time step of data acquisition getting faster while human motion stays at about the same rate. For example, a pirouette may have four states associated with it, one per quadrant of rotation. The amount of time spent in a particular quadrant may be very brief to human perception (a quarter of a second or so). However, Expressive Footwear updates at 50Hz, meaning that the dancer stays in a given state for 12 or 13 sample periods.



$$\begin{bmatrix} p_{00} & p_{01} & p_{02} \\ p_{10} & p_{11} & p_{12} \\ p_{20} & p_{21} & p_{22} \end{bmatrix}$$

Figure 3-5: Generic State Transition Diagram and Matrix

The second phase occurs during the receipt of data. Here, the problem is deciding which state sequence most likely produced the data vector. At first glance, this seems contradictory to the basis of Markov processes, where the statistics of the next state are not dependent upon previous states, but only the current one. However, a distinction must be made between the process of a given, known gesture and the data from a process whose gestural content is unknown. While it is true that for a given gesture, the associated Markov model has the stated probabilistic structure, this is a posterior model. If we are trying to detect a given process from what might be arbitrary data, sequences of state are not only allowable, but necessary.

As an example of the decision procedure, consider a two dimensional case where one can think of the data vector as a coordinate in a plane. Each of the states in the HMM has a value associated with it for each point in the plane that corresponds to the value of the density function of the given state evaluated at that point.

A particularly simple case arises when we look at a single feature vector and a model with two states whose vector components are independent, Gaussian random variables with unit variance and mean vectors x_1, y_1 and x_2, y_2. The final condition for this simple case is that the transition matrix is all 1/2. In this case, our decision would correspond to the state whose value is larger at the point of the data vector. One can easily imagine this process generalizing to many dimensions with arbitrary prior probabilities.

From a signal processing standpoint, examining state sequences could be seen as analogous to matched filtering, where your incoming data encodes state and your filter is designed from a given gesture. To take the example of the pirouette further, one could think of the states as the quadrants discussed above and the state sequence as (1, 2, 3, 4). If this state sequence is extracted from the data, the HMM returns a high probability. However, if the sequence is (1, 2, 1, 2), the output of the HMM should remain low even though the two sequences have the first two states in common.

It should also be noted that HMMs are much more flexible temporally than matched filters. Since data vectors are interpreted as being in states rather than as an element in a raw sequence, HMMs can interpret time-warped versions of gestures. Even with temporal scaling, many pattern recognition techniques fail when a particular gesture's temporal relationships are not scaled in a similar fashion.

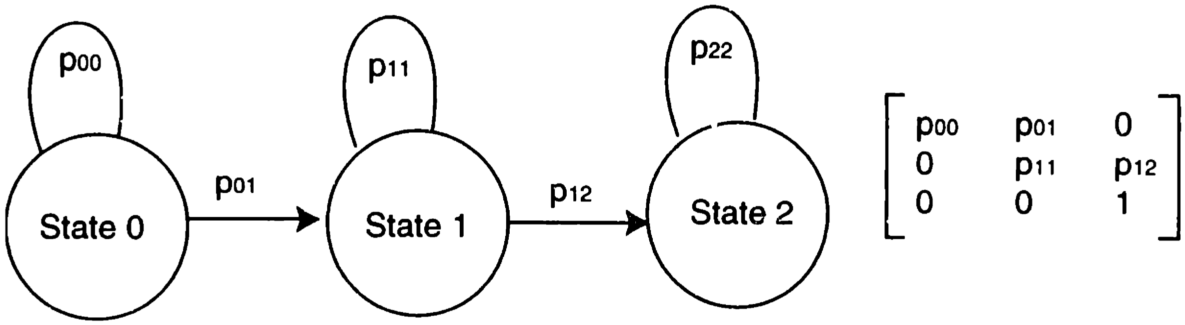
Our HMM Algorithm

There are three steps in performing the training algorithm. The first tells the HMM which sensor values to pay attention to. In theory, every HMM for eve~y gesture could always be looking at all of the sensor values and extract the necessary information from them. This would actually yield the optimal model in some sense. However, this would extend the number of training trials (discussed below) required for recognition and be very computationally expensive.

The second step in the training algorithm is deciding how many states the model of a particular gesture should have. Obviously, the more complex the gesture, the more states it will require. Physical states do not necessarily translate well into virtual states. For instance, a dancer may be thinking of a pirouette in the four states mentioned above, but the HMM may only require three: one corresponding to an initial state, one in which the magnitude of the gyro value is very high and one where the gyro value returns to its bias value.

The final step is where the real training happens. First, a trigger is sent to an HMM, indicating that its gesture is about to happen. A second trigger indicates the end of the gesture. From the data taken by the sensors specified, the HMM computes the probabilities in the transition matrix. We use the Baum-Welch reestimation formulas [13] iteratively on a set of observation sequences. Our probabilistic model for each state is a vector of jointly Gaussian random vectors, so we our iteration continues until the means and variances converge. That is, when the difference between the old values and the new values drops below a threshold, we stop iterating and use the new values in our HMM.

In order to effectively code the algorithm, a few modifications were made to the structure outlined above. First, for a given HMM, only forward transitions are used (see Figure 3-6). That is, all states but the initial and final ones have exactly one transition coming in, one transition going out and one transition leading to itself. In terms of the state transition matrix, only the diagonal elements and the entries immediately to their right are nonzero. Of course, the initial has no transition in and the final has no transition out. This means that the lower-right entry of the transition matrix is one.



$$\begin{bmatrix} p_{00} & p_{01} & 0 \\ 0 & p_{11} & p_{12} \\ 0 & 0 & 1 \end{bmatrix}$$

Figure 3-6: Simplified Transition Diagram and Matrix

Using this model greatly simplifies a number of problems. Obviously, the state transition matrix becomes much easier to deal with. The Baum-Welch formulas have the property that when a transition matrix entry is 0, all of the further iterations of that entry are 0. This knowledge allows for quite a bit of computational optimization.

Another benefit is the simplification of the state sequence problem. If you are in a given state, the only possible way to have reached it was through all of the previous states sequentially. The main problem to be solved at this point is determining where the transitions between different states happen. Here, we use the Viterbi algorithm [13], which is equally useful for generic hidden Markov models. In broad terms, the algorithm does a search through the state sequence space and returns the path with the highest probability.

The Baum-Welch formulas require the multiplication of a large number of probabilities. Exactly how many multiplications is dependent upon the length of the gesture in question or, equivalently, the size of the observation window. Many of the multiplications involve the off-diagonal transition probabilities, which, as explained above, can be quite small. Even with modestly sized observation windows, underflow can, very rapidly, have adverse effects on the HMM. Since the drop off is exponential, our algorithm computes the log of the likelihoods and adds them rather than multiplying all of them out.

Another interesting problem arises with HMMs that have states whose mean vectors are very close to each other. In judging whether or not a gesture has occured, we look at the probabilities of state sequences and pick the most likely one. Assume for the moment we have three states in our sequential model (s_1, s_2 and s_3). Now assume s_1 and s_3 have means very near each other and s_2 is quite distant in relation. The only way to distinguish between s_1 and s_3 is by looking at where we are in the state sequence. What ends up happening is that any vector path that leavcs s_1 and enters s_3 will cause the HMM to construct a state path through s_2 even if the data path goes directly from s_1 to s_3. The probability of the path at the point in time where the switch occurs between s_1 and s_3 is very low, but the summation of the probabilities of the state sequence over the observation window is still very high, making it statistically viable.

One possible solution is to add more states. Similar problems arise in creating a state sequence. With N additional states, you simply have N additional time steps during which you progress through all of the intermediate states. However, the transition probabilities cause these pathological paths to be discarded. Though this solution would work, it is undesirable since the transition matrix grows like N^2 as does the number of calculations for both the Baum-Welch and Viterbi algorithms.

Instead, we simply recognize the paths which have isolated instances in time of a very low probability. That is, we add the further constraint that only state sequences whose probabilites are above a threshold at every instance in the observation window are valid. Mathematically, we simply take any probability that ends up below our threshold and map it to $-\infty$. This allows the algorithm to run unperturbed with a single special case.

3.2.2 HMM Applications using Expressive Footwear

As noted earlier, Expressive Footwear has many possible applications. However, the sensor card is even more flexible. There is no reason why it has to be attached to a shoe. On the card itself sits three axes of magnetometers, one axis of gyroscopic measurement, three axes of high-g accelerometers, two axes of low-g accelerometers and a sonar transducer. In addition, there is a header on one edge of the card that allows five more single ended inputs and one differential input. Looked at this way, the sensor card has an enormously wide range of uses.

The Synthetic Characters Group at the Media Lab has already explored using the sensor card as an interface to a video game [1] and is continuing development of the card as a virtual marionette, while another group at the Media Lab is trying to incorporate it into kites. Jerry Bowskill from British Telecom (a Media Lab sponsor) has also tried using it in bicycle helmets for gestural recognition and context segmentation.

Again, we have yet to integrate HMMs into our music applications[1]. We have, however, recognized some simple gestures. With a 5 state HMM, we were able to detect the sequence of someone pointing their toe forward, then up and then forward again. A similar gesture, with the toe pointed forward initially, spun clockwise and spun back, was also recognized with a 5 state HMM. Both of these gestures demonstrate that the alterations to the traditional algorithm were effective for detecting gestures whose initial and final states have similar mean-vectors.

[1] The integration would be straightforward to do if we simply substitute our current data input vector with an HMM probability vector. Corresponding changes to the mapping could be made based on the HMMs used. Of course, using both the direct data and probabilities from various HMMs is also a possibility.

Chapter 4

Conclusions

Many positive results came about from working on Expressive Footwear. As important as the results themselves, however, is the trajectory that has emerged. Not only is the technology already capable, but it has a promising future. Both the benefits of the current state of the project and the promise of the future are discussed in this final chapter.

4.1 The Past and Present

Expressive Footwear brought together many sensing technologies and utilized a wide array of sensing techniques. It has not been until recent years that all of this sensing technology has reached a size where it could be integrated into such a compact package.

Perhaps the most notable examples of technological advancement leading to size reduction are the accelerometer. Just a few years ago (perhaps while I was still in high school), MEMs was considered a black art. People did not understand all of the implications of creating such fine mechanical structures. Now, the ADXL202 is mass-produced by Analog Devices. The AMP accelerometer is also a fine example of miniaturization. Piezo polymers have revolutionized aerospace and civil engineering with its remarkable charging properties; and sensors have taken advantage of these properties as well.

Of course, most representative of the miniaturization process is the PIC. From the beginning of the semiconductor industry, microcontrollers have been a target product. Every year, companies are competing to make faster and more powerful microcontrollers. Luckily, in recent years, the industry has recognized the need for programmable devices that do not carry the baggage of full processors like the Alpha or Pentium. The PIC is a result of producing devices to fill this niche.

After having created such a small input device with so much advanced hardware, developing the next layer of content also integrated some very advanced software. Both the musical applications and the gesture recognition incorporated software that is on the cutting edge of their respective fields.

The Rogus McBogus library was at the very heart of the Brain Opera, perhaps the most ambitious use of interactive, electronic music to date. Of course, our musical demands were nowhere near the proportion of the Brain Opera, but the smooth, efficient interface made collaborating with dancers and making modifications on the fly a breeze.

Integrating HMMs into the Expressive Footwear project was a perfect match. Though the basics of HMMs have been around for 10-15 years, it has only been recently that computers have been able to keep up with the computation. The main body of work in HMMs is also relatively narrow. It focuses on speech recognition and control systems, the bread and butter of the signal processing industry. Expressive Footwear, in contrast, has given HMMs a chance to demonstrate their power in solving a high-dimensionality problem in multimodal human-computer interface.

4.2 The Future

There are many directions where Expressive Footwear could go. Its role in wearable computing is, of course, the most obvious. As wearables begin to permeate into everyday life as natural entities, the interfaces to them should permeate just as naturally. The three applications already mentioned (medical, musical and gestural) also show great promise.

The goals of wearable computing are to make our environments friendlier and more responsive. Foot sensing, such as in this project, has all sorts of possible uses in this context. If you want the light in the room over your head to be brighter, tap your foot. If you want it dimmer, tap twice. Is the room too cold? Point to the thermostat with one shoe and stomp the floor with the other. Though these particular ideas may not be viable, similar ones certainly are.

Medical applications for Expressive Footwear have yet to be explored at the Media Lab. Given the benefit to the preventive medical community Expressive Footwear might bring, it is unfortunate that applications down this path have not yet been explored. The medical community, however, has seen our system at conferences and in the press, and have visions for how it might be useful in medicine. A unique, multi-sensor platform like Expressive Footwear lends itself well to podiatric research, and the potential applications are numerous and span many levels of development.

Gestural recognition is also moving along at a fast pace. Already, the limitations of HMMs are being recognized and research is being pursued to extend its capabilities. Flexible learning algorithms that do not need to be told how many states to use or what the transition matrix is are being developed. Once these fall into place, the possibilities are endless. There's no telling what might be learned by a device that is adapting to people all the time. And, if it really does get to be as intrusive as people have a tendency to fear, devices can always be turned off.

What the author would love to see is to finally integrate the gestural recognition into the musical application. The boundary between the two is already blurring and should be nonexistent if possible. Learning and identification algorithms are running faster on ever more powerful hardware, reducing latency concerns that plagued much of their early applications to real-time performances. Both sides of the integration are already very well developed and ready to exist as a cohesive unit. The compositional implications are very intriguing, as are the performance aspects. It would be possible to simply make an instrument and treat the dancer as a player in an orchestra. It would be possible to write a full piece and simply have the shoe trigger the start. It would also be possible to make the mapping an improvisatory one, with certain constraints to lend some sort of coherence and unity to the music. With the gestural recognition, the improvisatory mapping could even learn the nuances of its user with enough trials.

These are only the musical possibilities. Dancers' and choreographers' imaginations could certainly add more to the mix. Our work with Byron Suber has definitely been a testament to this. There is no reason to think that such success would be curtailed by further involvement of the dance community.

4.3 Final Remarks

Working on Expressive Footwear has been a very rich learning experience. It operates on many different levels of complexity and is rather innovative on many of them. Everything from the physics of electric-fields up through MEMs hardware and cutting-edge gesture recognition software to the philosophical ideas of composers, choreographers and dancers are touched upon and explored beyond their normal bounds. My hope is that the technology continues to be developed and its applications continue to challenge human sensibility.

Appendix A

Tables

Table A.1: Specs of ICs

Chip	Vendor	Part #	Contact	Relevant Specs
PIC	Microchip	16C711	www.microchip.com	16MHz, 1K ROM, 68 bytes RAM, 8 bidirectional digital I/O lines, 4 analog inputs, 8 bit A/D, 13mA
Mux	Fairchild	CD4051	www.farichildsemi.com	500ns switching, $< 160\mu$A
5V Reg	Maxim	MAX883	www.maxim-ic.com	200mA continuous current, $\pm$.25V accuracy, 0.11V dropout, 11μA quiescent current
3V Reg	Maxim	MAX8873	same as above	120mA continuous, $\pm$0.1V, 55mV dropout, 82μA quiescent

Table A.2: Specs of Amplifiers Used

Vendor	Type	Part #	Gain-BW	Current/Op-Amp
ADI	Quad	Op462	15MHz	500μA
ADI	Instrumentation	Op623	–	575μA
Maxim	Dual	MAX474	10MHz	2mA
Maxim	Dual	MAX492	500kHz	150μA
Texas Instruments	Dual	TLE2082	10MHz	160mA

Table A.3: Sensor Data

Sensor	Contact	Relevant Specs
Interlink 400 FSR	www.interlinkelec.com	$r_{min} = 100\Omega$
FlexiForce ELF4200 FSR	www.cooperinstruments.com	$r_{min} = 20k\Omega$
AMP Sensors[a] DT-1 (PVDF)	www.msiusa.com	Stress constant (g_{31}) = 216V/N
Abrams-Gentile (Bend)	www.infomall.org/home/ tenants/age.html	50μA
ADI[b] ADXL202 (low-g)	www.analog.com	0.5mg res @ 100Hz, dual-axis, 0.6mA
Murata ENC Gyro	www.murata.com	±90 deg/s range, 3.5mA
Honeywell compass (HMC2003)	www.ssechoneywell.com	40μGauss res, 3-axis, < 20mA
AMP Sensors[a] (high-g)	www.msiusa.com	0.2mg res @ 100Hz, < 10mA
Polaroid 40LT16 (Sonar)	www.polaroid-oem.com	40kHz Freq, ±1kHz BW, 119dB sensitivity

[a] Now Measurement Specialists

[b] Analog Devices

Appendix B

Schematics and Data

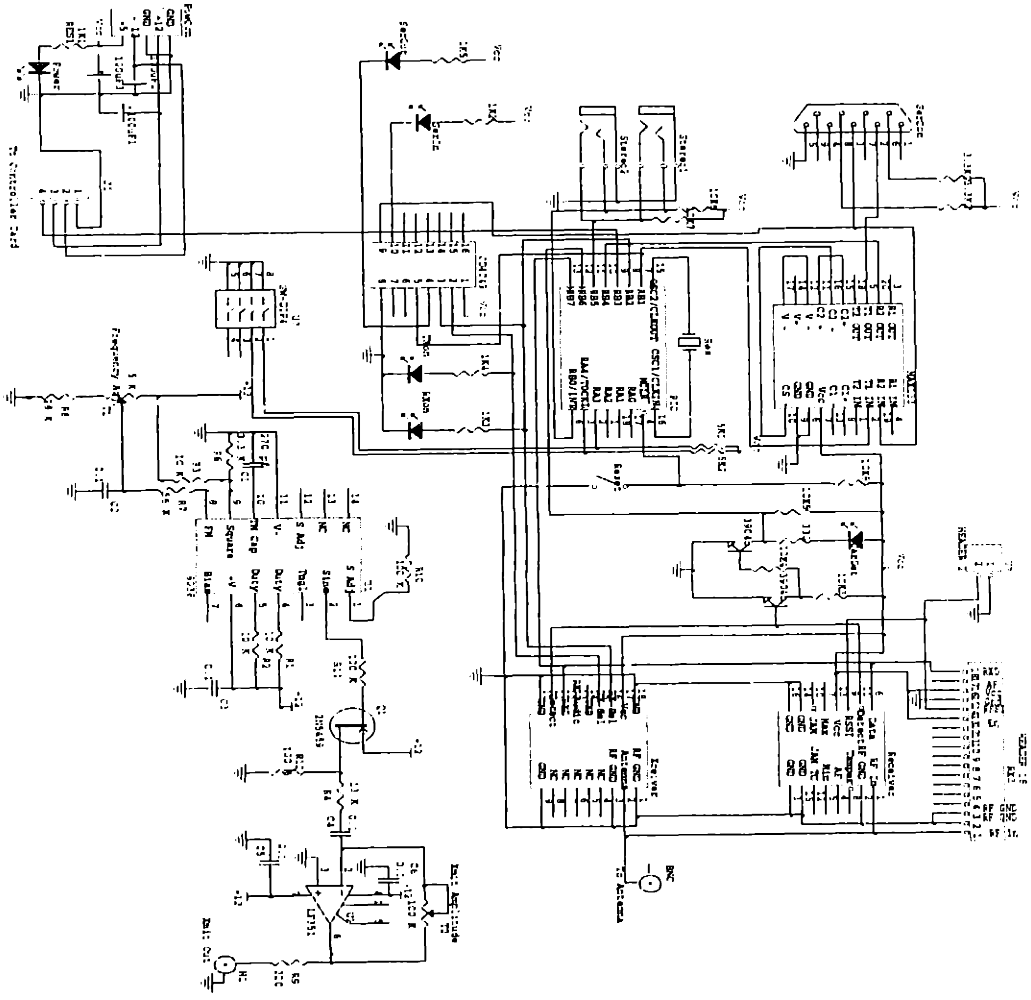


Figure B-1: Base Station

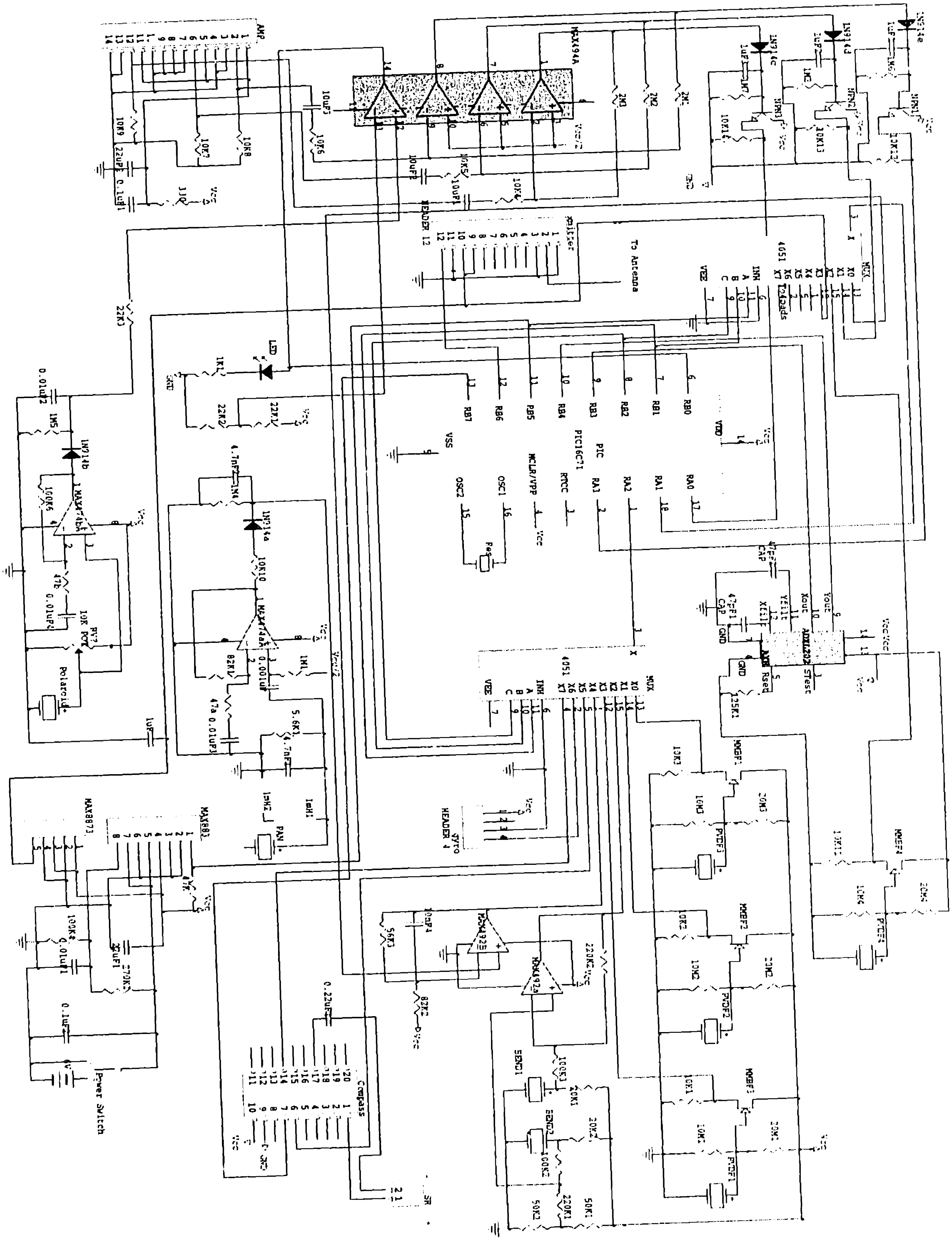


Figure B-2: Shoe Electronics

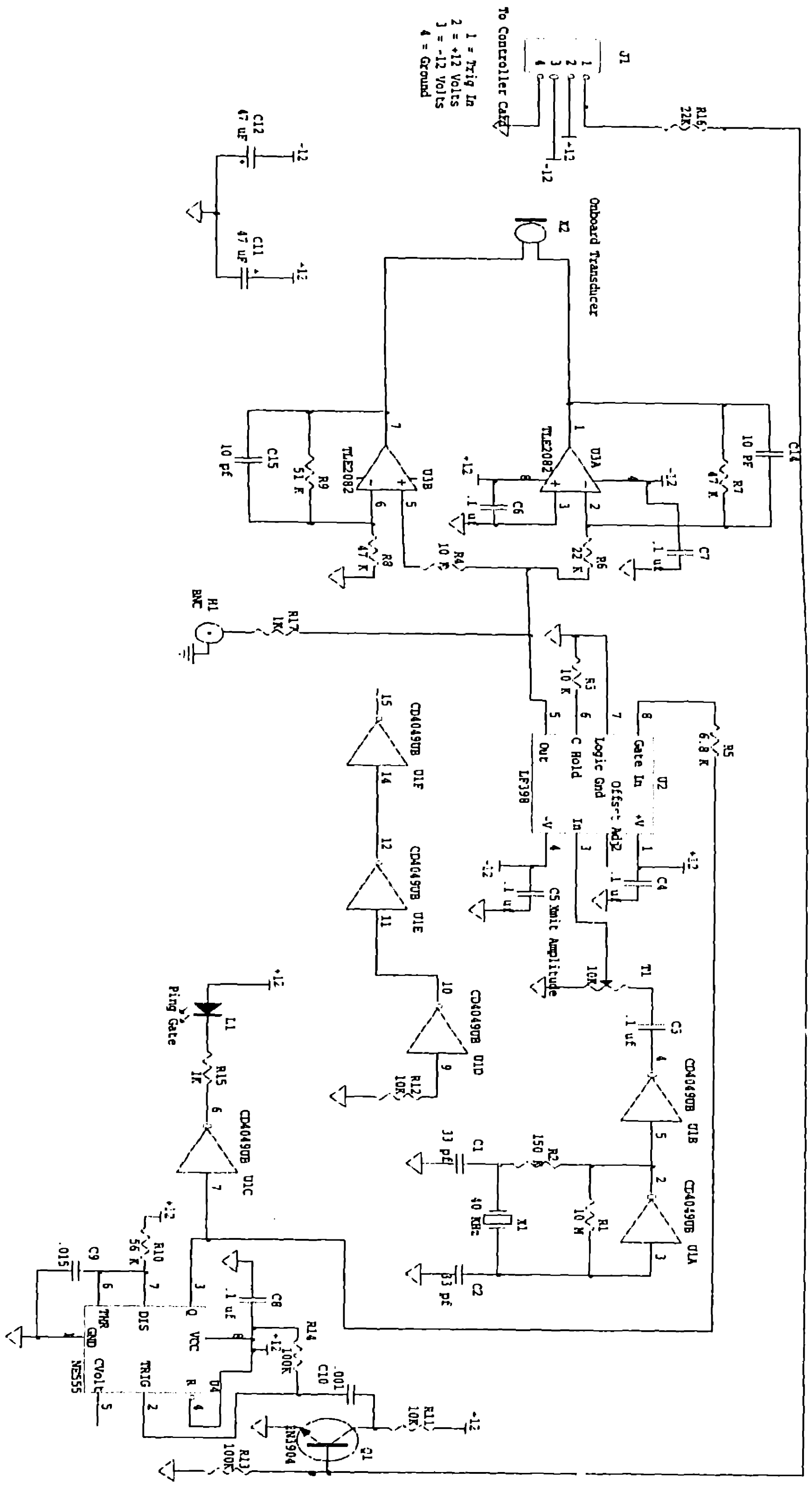


Figure B-3: Sonar Pinger

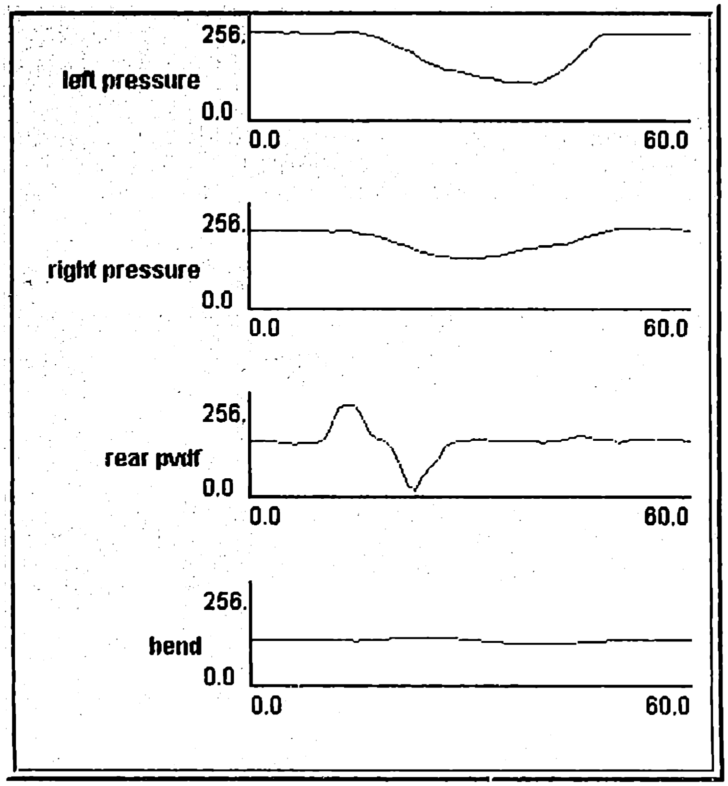


Figure B-4: The In-Sole Sensors During a Single Walking Step

In Figure B-4, you can see the pressure sensors working together to detect a person walking over a 1s interval of time. The "rear PVDF" shows the initial impact with the ground from the heel and the "left pressure" and "right pressure" sensors show the shift of pressure from the heel to the ball of the foot. As the foot flexes, the "bend" sensor changes slightly. When calibrated, the effects of the "bend" sensor are quite dramatic.

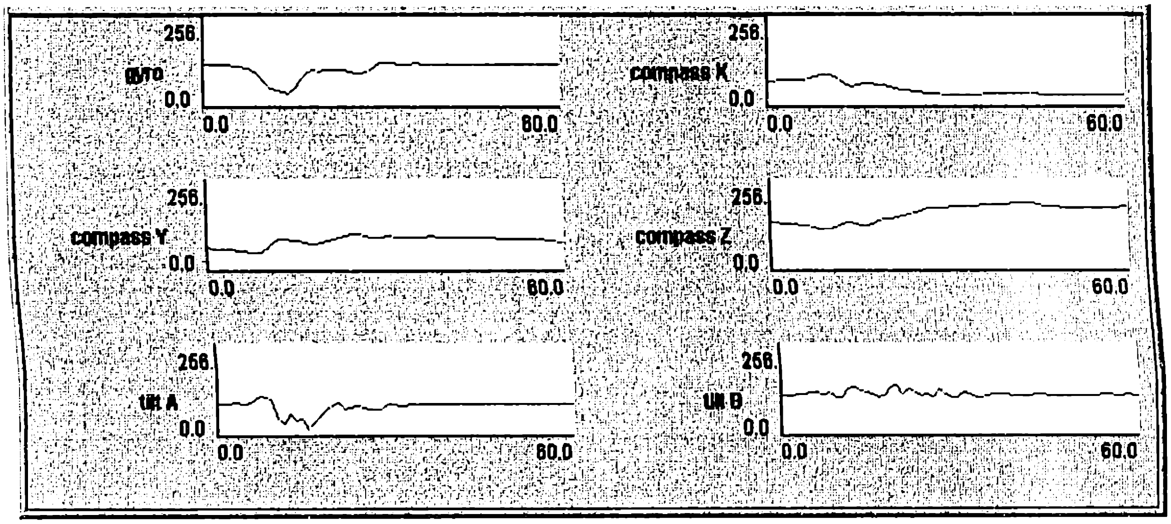


Figure B-5: The Orientation Sensors During a Spin

In Figure B-5, you can see the orientation sensors working while a dancer does a spin. Both "ti!t A" and "tilt B" wobble around due to the fast, dynamic motion. The dancer's foot was tilted up somewhat during the spin and that can be seen in "tilt A". The "gyro" behaves as expected. All of the compass sensors show a stable beginning and end, indicating that the dancer did not end up in quite the same final position as initial. These sensors also show the prepatory motion of the spin as evidenced by the transients before the "gyro" changes value.

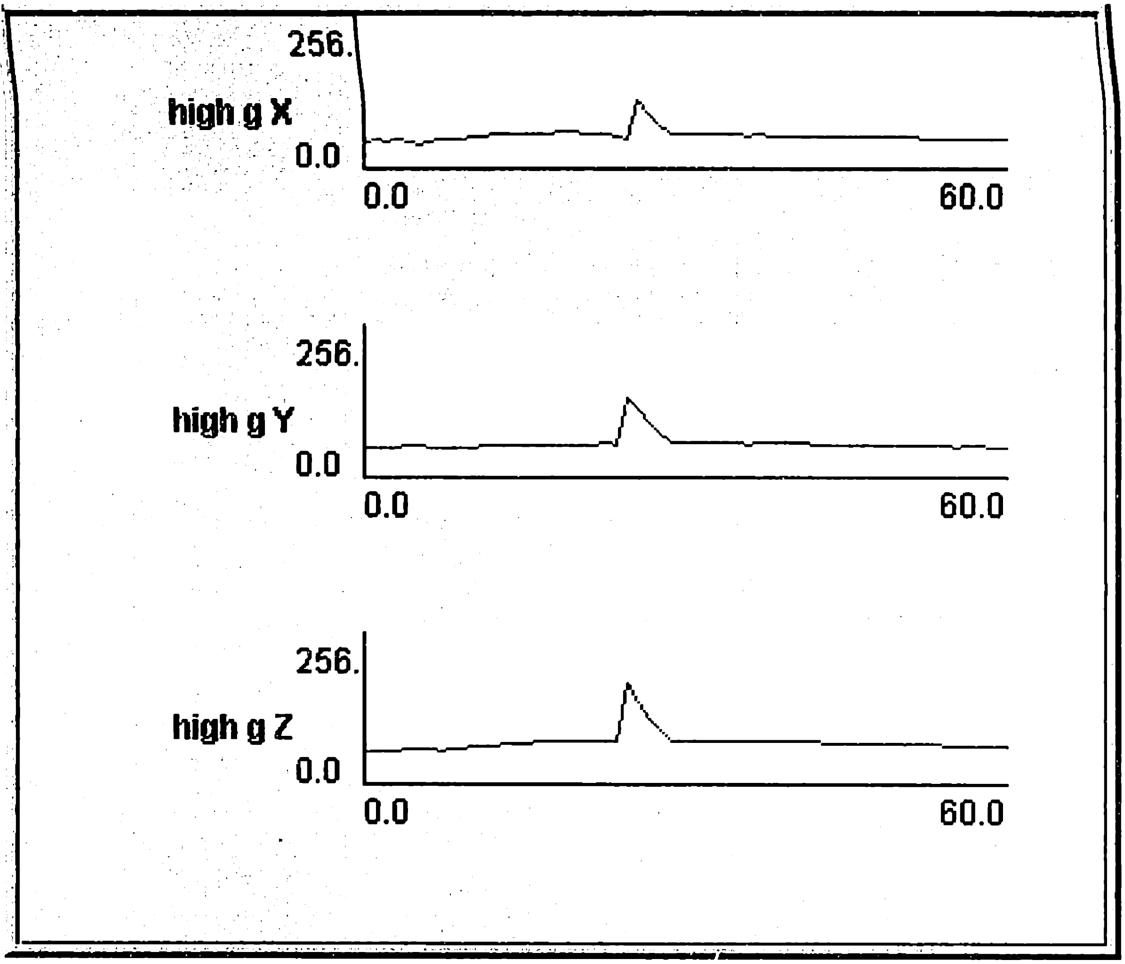


Figure B-6: The High-g Accelerometers during a Stomp

Here, in Figure B-6, you can see the quick rise in the "high-g" accelerometers after a foot stomp, as well as the characteristic exponential decay that we use to remove bad packets that spike and return instantly to the baseline.

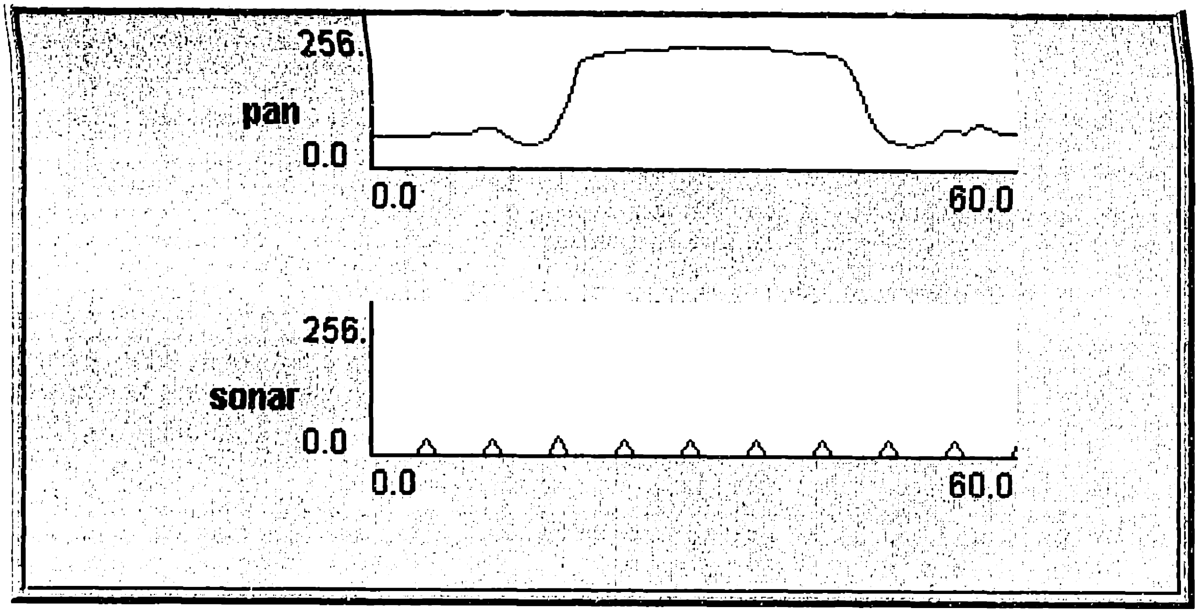


Figure B-7: Global Sensors

Here, we see data from the global sensors. The "pan" electric-field sensor shows the shoe being moved into and, subsequently, out of the zone over the transmitting electrode. On the "sonar" plot, you can see the values transmitted back every 5 data packets (roughly 100ms). Because the window of the plot was so small, it was difficult to show dramatic changes in this plot.

Bibliography

[1] B. Blumberg. Swamped! Using Plush Toys to Direct Autonomous Animated Characters. *SIGGRAPH 98 Conference Abstracts and Applications*, page 109, 1998.

[2] P.R. Cavanagh, F.G. Hewitt Jr., and J.E. Perry. In-shoe Planar Pressure Measurement: a review. *The Foot*, 2(4):185–194, 1992.

[3] B. Denckla and P. Pelletier. The technical documentation for 'Rogus McBogus', a MIDI library. http://theremin.media.mit.edu/rogus.

[4] F. Eady. RF Telemetry: Part I: Theory and Implementation. *Circuit Cellar Ink*, (90):61–67, January 1998.

[5] M. Gray. Cluster Weighted Modeling: A mixture of E-M Experts Applied to Analysis of Biometric Data. http://www.media.mit.edu/~mkgray/army/main.html.

[6] L.J. Hutchings. System and Method for Measuring Movement of Objects. US Patent No. 5724265, March 3, 1998.

[7] M. Johnson, A. Wilson, C. Kline, B. Blumberg, and A. Bobick. Sympathetic Interfaces: Using a Plush Toy to Direct Synthetic Characters. *SIGCHI 99 Conference Abstracts and Applications*, 1999.

[8] T. Machover. The Brain Opera and Active Music. *Proc. of the Ars Eledtronica 96 Festival*, pages 300–309, 1996.

[9] J. Paradiso and E. Hu. Expressive Footewear for Computer-Augmented Dance Performance. *Proc. of the First International Symposium on Wearable Computers*, pages 165–166, October 1997.

[10] J. Paradiso, E. Hu, and K.Y. Hsiao. The Cybershoe: a Wireless Multisensor Interface for a Dancer's Feet. *Proc. of the International Dance and Technology 99*, 1998.

[11] J. Paradiso, E. Hu, and K.Y. Hsiao. Instrumented Footwear for Interactive Dance. *Proc. of the XII Colloquium on Musical Informatics, Gorizia, Italy*, pages 89–92, September 1998.

[12] A. Pentland. Wearable Intelligence. *Scientific American Presents*, 9(4), Winter 1998.

[13] L. Rabiner and B.H. Juang. An Introduction to Hidden Markov Models. *IEEE ASSP Magazine*, pages 4–16, January 1986.

[14] T.G. Zimmerman. Personal Area Networks: Near-field Intrabody Communication. *IBM Systems Journal*, 35(3):609–617, 1996.