Designing the Haptic Turntable for Musical Control

Tim Beamish, Karon MacLean, Sidney Fels
University of British Columbia
{tbeamish,maclean}@cs.ubc.ca, ssfels@ece.ubc.ca

Abstract

In this paper, we discuss the design and implementation of D'Groove, an intelligent Disc Jockey (DJ) system that features the use of haptic force feedback to expand the expressive abilities of the traditional DJ setup.

We begin by describing the tasks of a DJ and defining some of the challenges associated with the traditional DJ process. We then introduce our new system, discussing how it alleviates these problems and at the same time introduces new performance possibilities. This is followed by a detailed description of some of the haptics-design-related problems that we solved in the course of building the system, including a method for accurately calculating low velocities. We conclude with a discussion of the role of haptics within the DJ domain.

1. Introduction

A Disc Jockey (DJ) plays prerecorded music at social events. In the past, this job meant simply selecting a music playlist and pressing a button to play it. A modern DJ show consists of the DJ mixing portions of numerous vinyl records together with two turntables and a mixing board to combine sounds into a new unified song. Portions of songs from the turntables are played simultaneously through the mixing board to produce a unique mixture of music.

Figure 1: A DJ mixing with traditional vinyl

Until now, the DJ used his or her hands only to control the physical artifacts associated with DJing: the vinyl records and various controls on the mixing board. No information about the musical content was provided through tactile feedback. Our new DJ system incorporates haptics into the DJ's setup, enabling information to flow in both directions with the sense of touch. Now we can both convey important musical information to the DJ through touch, and also create new methods, or haptic tricks, that a DJ can use in his or her performance. This broadens the realm of DJ expression and heightens the level of creativity in the music. The end of this paper will outline some of the new haptic performance possibilities that our system affords.

2. Tasks of the DJ

A DJ's primary goal is to entertain an audience by providing a musical experience, both auditory and visual. Today's DJs must be skilled in the art of mixing, where pieces of songs are blended together to create a single "opera" of music. In order to mix music into a seamless stream, the first and most important skill a DJ must learn is "beatmatching", where he must match the beat structure of the forthcoming song with that of the one that is currently playing. This is done by adjusting the tempo and phase of the forthcoming song, and it is a challenging task both cognitively and perceptually.

The turntable is a natural tool for this process as the rotational speed of the platter directly affects the playback speed (tempo) of the song being played on the platter, making it an intuitive "handle" for beatmatching. Turntable music comes in the form of vinyl records, physical artifacts that can be pushed or pulled to alter playback speed and direction. Adjusting the phase of a forthcoming song to coincide with the phase of a currently playing song simply involves using a hand to slow down, or even halt, the forthcoming song until its beats are inline with the current song. Currently, a DJ does this using his ears for cueing and his hands to control tempo and phase. He must listen carefully to the currently playing song in one ear and the forthcoming song in the other ear; then speed up or slow down the forthcoming song by physically adjusting the rotational velocity of its vinyl disk with his hands until its beat structure matches that of the currently playing song. Once the two songs are beatmatched, as shown in Figure 2a the DJ can begin to mix the two into a seamless piece of music. Examples of incorrectly matched beats are shown in Figures 2b and 2c.

Proceedings of the11th Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems (HAPTICS'03)
0-7695-1890-7/03 $17.00 © 2003 **IEEE**

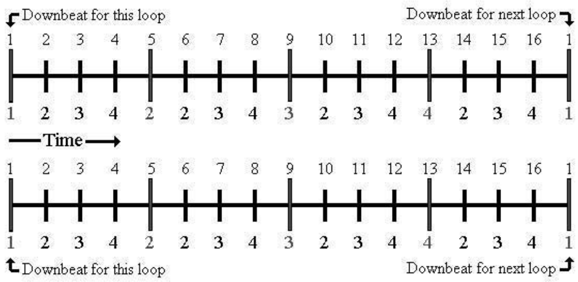

Figure 2a: Two correctly matched beats

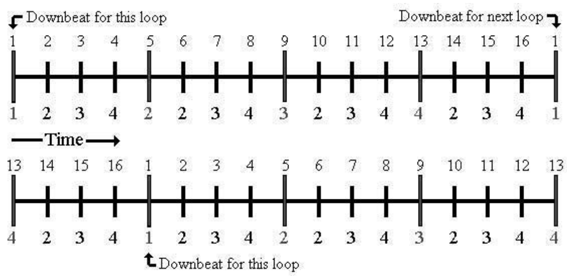

Figure 2b: Two beats that have the same tempo but are out of phase

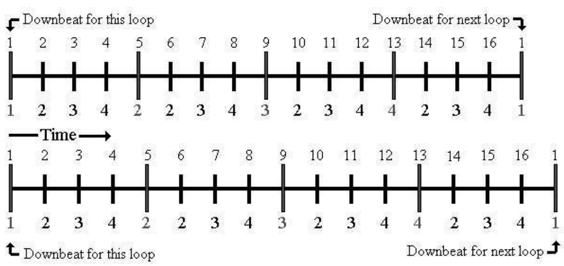

Figure 2c: Two beats that do not have the same tempo

Turntables are the leading controller for this task for two reasons: 1) they provide a physical medium in which the DJ can touch the music and get a direct response in its playback speed and direction; and 2) they create an important metaphor whereby the rotational movements of the turntable platter represents the progression of a piece of music. As a result, what might initially seem a trivial relationship is actually why turntables remain the leading controller for DJs. No other DJ controller provides a moving component and no other controller has lasted as long in the DJ domain as the turntable.

3. History of DJ Controllers

Most research concerning DJ controllers has been conducted by commercial companies, and thus remains unpublished. This section outlines some of the more notable commercial products and comments on some academic projects that are related to the field.

The CDJ-1000 [11] is a commercial CD player that is designed to emulate the turntable. While impressive, the CDJ-1000's turntable metaphor is incomplete because it lacks a motor. Thus, the unit does not rotate freely on its own and it loses out on the importance of rotation and the automatic progression of music.

Finalscratch [7] is another commercial product that uses standard turntables and specially encoded records to send a signal into a computer. Again, this is impressive technology, but does not provide a new means of control for DJs. They still perform as they would with a regular turntable and aside from permitting the DJ to access digitally encoded media, new kinds of performance are not enabled.

In 2001, four students from Stanford, Keatly Halderman, Daniel Lee, Steve Perella and Simon Reiff, under the supervision of Bill Verplank connected a turntable to a computer to control the playback speed of a song [13]. Their work is closely related to Finalscratch in that it uses the same components and does not support any new additions to the DJ domain.

Related to the study of turntable sounds, Kjetil Hanson is working towards a sound model of the scratching sounds produced by turntable musicians [8]. This work is interesting because it may enable us to produce new digital scratching sounds with our interface.

More closely related to haptics, Lonny Chu is working on a haptic dial used to navigate audio in an editing scheme [5].

4. Motivation

Our goal when designing D'Groove was to add simple, single degree of freedom haptics to a DJ setup. Since DJing is already a very hands-on task, we felt that it would be natural and valuable to provide more musical information through the sense of touch. Active haptics in the turntable permits the DJ to receive this information as well as control the musical progression; and at the same time, we believed we could increase the DJ's intimacy with the system and offload some perceptual load from the auditory and visual senses.

We also wanted to make it easier for DJs to beatmatch songs, to allow novice DJs to learn more quickly and experienced DJs to concentrate on higher level audio effects. The basic paradigm was to link the rotational velocity of the turntable platter to the song's tempo such that one revolution of the platter maps to n beats in a song (since we are primarily using modern 4/4 dance music, we choose $n = 4$). By placing visual cues on the platter, the DJ could then tell if two songs were beatmatched by noting that the markers travelled passed a corresponding point on each turntable at the same time. Figure 3 illustrates this idea with four turntables. With the rotation of the turntable coupled with the tempo of the song, we can visually see that the top two turntables are beatmatched. The lower left turntable is one beat out of phase with the top two and the lower right is half a beat out of phase. If the markers labeled 1 on each turntable pass by the markers labeled N on their respective turntable at the same time, we can say that the songs are beatmatched.

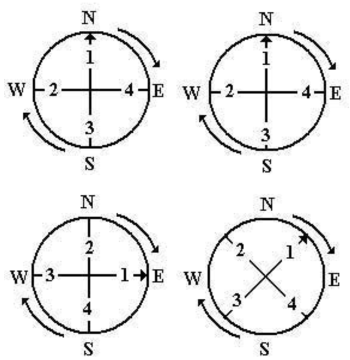

Figure 3: Markers on separate turntables help beatmatching with visual cues

5. D'Groove: The Digital Haptic Turntable

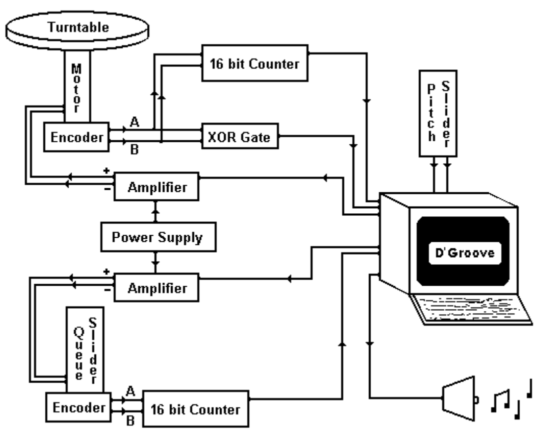

Figure 4: D'Groove's Configuration

The D'Groove system is composed of hardware and software components. Each component works together to create a richly expressive interface from simple low cost parts. The external physical hardware portion of D'Groove consists of three controllers: the turntable, the pitch slider and the queue slider. In this version, each is interfaced through a DAS-1602 Keithley MetraByte data acquisition board inside a 1.0 GHz Pentium III computer. Figure 4 outlines a complete schematic of the hardware setup.

5.1 The Turntable

The function of the turntable is to control the playback speed and direction of a piece of digital music. The turntable is composed of a brushless DC motor by Maxon with a small platter attached to one end. In the first iteration, we used a smaller, but readily available, platter configuration, which will be upgraded to full size in a subsequent iteration.

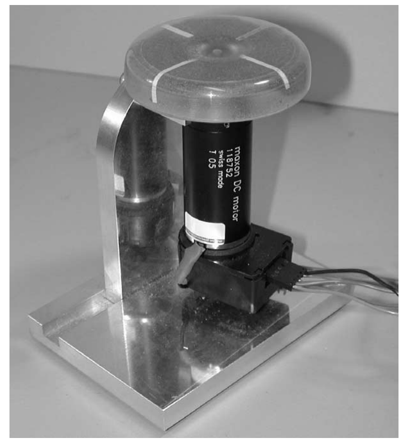

Figure 5: The Initial D'Groove Turntable

The motor is fitted with a dual-channel optical encoder boasting 4000 encoder ticks per revolution. The encoder is decoded locally using a 16 bit counting chip and then input to the host computer through a digital port on the data acquisition card. A DAC on our data acquisition card drives an Apex OPA544 amplifier chip, which in turn provides the motor's input current.

5.2 The Pitch Slider

D'Groove is equipped with a linear and non-actuated potentiometer, which sets the desired velocity of the turntable motor and platter. We call it the "pitch slider" because the platter's rotational speed directly affects the music's playback speed and hence its audible pitch (as with a conventional DJ setup, we do not pitch-correct the sound). The DJ can turn the pitch slider off: the position of an associated toggle switch determines whether variations in the slider's position affect the motor speed. This also permits use of an infinite sliding range by setting the slider to full speed, switching it off, resetting and switching it on again to further increase the motor speed in a manner similar to lifting a mouse and repositioning it to increase its "reach". The same applies in the opposite direction, allowing the motor to slow down and even rotate in reverse. Most conventional turntables are equipped with a similar sliding potentiometer for setting the pitch of the music; however, in their case one must trade off resolution to gain range. In our slider, we can set the resolution in the software, and with our toggle switch, range is not an issue. In future editions of D'Groove, we will turn this slider into a continuous knob with an encoder and avoid the toggle switch altogether.

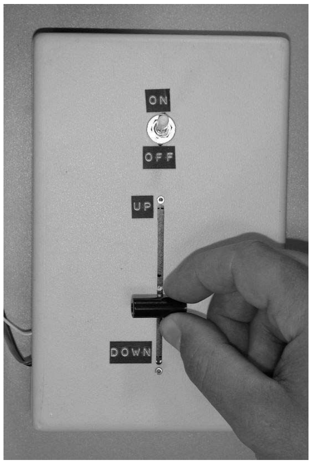

Figure 6: The Pitch Slider

5.3 The Queue Slider

D'Groove's third component, the queue slider (Figure 7), consists of a sliding potentiometer - like the pitch slider, but in this case it is motorized. The motor is made by Falhaber and also has a dual-channel optical encoder with a resolution of 400 CPR. The sliding portion of the potentiometer is made by Penny and Giles and is connected to the motor via a belt. We attached a pressure sensor on the slider's handle to sense when a user has grasped the slider. As with the turntable, we connected both encoder channels to a counting circuit that is fed into our data acquisition card. We control the current to the slider motor with another amplifier circuit connected to a second DAC on the Keithly card. We have two methods for determining the position of the slider, one from the encoder reading and one from the potentiometer reading. The encoder reading is more accurate; but the potentiometer reading is less vulnerable to slippage in the belt drive and thus we are using it in this iteration.

The queue slider replaces the needle and needle arm on a conventional turntable. The slider moves automatically along the track as the music progresses, under a tight position control loop. When the user squeezes the pressure sensor on the slider's handle, we put the slider into a passive mode, allowing the user to freely control its position, to locate the new play-point and meanwhile feeling position-based haptic feedback based on the track's content. When the user releases the slider (and stops squeezing the handle), we switch back to the driven (position-control) mode and resume movement of the slider along its track. When it reaches the end, it signifies the end of the song and the slider resets itself to the beginning of the track.

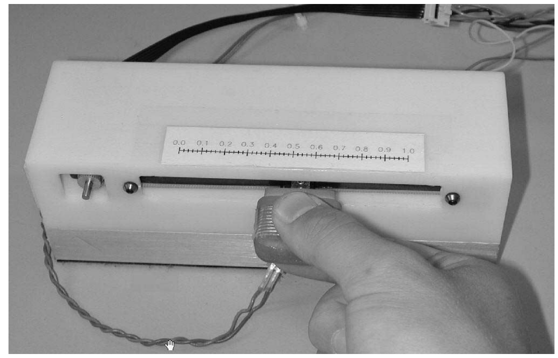

Figure 7: The Queue Slider

6. Our Software Environment

D'Groove's operating system and utilities environment is composed completely of Open Source components readily available from the Internet.

We required a realtime operating system to control the forces of the motor without hindering the playback of the music. After an analysis of various realtime operating systems, we chose Real Time Linux (RTLinux) [12].

RTLinux comes as a free patch for most Linux kernels and splits programs into realtime and non-realtime components. Our motor control loops are run as realtime components in the kernel. Our musical playback software and GUI is non-realtime and is run in regular user space.

We also installed Andrew Morton's low latency audio patch [3] to give the audio playback portion of the system its best chance for low latency. The patch provides near realtime audio playback with latency of 5 ms or less.

For musical playback, we used Alsaplayer [2]. This media player supports the playback of MP3, WAV and CD formats, and also supports the multi-speed playback of sound files.

Alsaplayer was designed to utilize the special features of the ALSA sound library [1], therefore in order to use the player to its full potential, we installed ALSA. This sound library is more efficient than Linux's traditional OSS platform and enables one to use fully modularized sound drivers.

Finally, we have installed COMEDI [6], a generic API for use with multiple data acquisition boards. COMEDI allows us to write board independent code so that the system can be run on any COMEDI compliant machine.

7. The Software

D'Groove is a realtime motor control and musical playback system. It uses two haptic control loops to invoke forces over the two motors as well as an audio engine to playback music at various tempos. The critical realtime code is run in the Linux realtime kernel and communicates with non-critical code in user space through shared memory. Thus, the software is comprised of two components, a hard-realtime motor controller and a soft-realtime user interface.

7.1 Controlling the Motors in Realtime

Within the kernel, we run our two haptics loops, both at 1 KHz. Each loop controls one of the motors in the system, either the turntable or the queue slider. Our main goal for the turntable was to maintain a constant but over-ridable velocity, just like a real turntable – i.e. the user can still feel resistance when over-riding, and in fact relies on it. For the queue slider, we wanted to tightly control position but completely turn off control when under manual control. And in both cases, we wanted to add haptic force feedback effects to each motor.

Within the turntable loop, we query the encoder counter and calculate a velocity based on the last three position samples. This averaging technique smoothes our velocity estimate and makes our control more stable.

The turntable encoder has 4000 ticks per revolution and the counter counts from 0 to 65535 before it wraps back to 0 again. Ideally, we want to know the position of the encoder (from 0 to 3999) so we can determine the position of the turntable but we can only read from the counter. We thus have a wraparound problem, which is not generally encountered in haptic interfaces, since most configurations do not allow continuous rotation. Further, as 4000 does not divide nicely into 65535, we also have a floating zero problem and must calculate the position in a range of 0 to 3999 based on the counter. The same problem occurs with the queue slider as it has 400 ticks per revolution. An algorithm to determine this was implemented and is inserted inside the haptic loops.

7.1.1 Measuring the Turntable's Velocity

We run into another problem when calculating the velocity of the turntable at the moment that the encoder flips from 3999 to 0. If we merely calculate the distance by subtracting the current encoder reading from the last encoder reading, we end up with an enormous distance at this point. The computed velocity becomes huge and the system becomes briefly unstable. To avoid this, we convert the position into radians and use the sine and cosine of the encoder reading to determine the distance. This gives us a continuous distance reading and removes the velocity jerk when the encoder flips.

Once the velocity is determined, we feed it into a proportional-integral-derivative (PID) control algorithm to calculate a force to send to the motor. The user specifies the desired velocity of the turntable using the pitch slider and the PID controller makes it happen. If the user touches the turntable, the PID controller senses an error and attempts to compensate. The integral component of the PID controller is necessary to reduce steady state error but causes windup when the user holds the turntable platter. The error between desired and actual velocity steadily increases, driving the force up. Therefore, we implemented an anti-windup cap that limits the amount of the integral component to a level that the user cannot perceive.

7.1.2. Emulating the Turntable with Constant Velocity

One of the more interesting problems that we dealt with while building the haptic turntable was to the need to achieve a truly constant velocity from the motor when in regular turntable mode. In our first design, we found that our velocity estimate (differentiated from the encoder signal) was sufficiently noisy, at the relatively slow velocities, to cause undesirable wavering sounds in the music whose rate it controlled.

The basic problem is that at typical playback speeds, relatively few encoder ticks are received per millisecond sample period, even with a relatively high-resolution encoder. For example, we might play a song with a tempo of 120 beats per minute (BPM) on our turntable at 30 rotations per minute (RPM). This equates to 4 beats of the song per one revolution of the turntable, or ½ rotation per second. As we are sampling at 1 KHz and using a 4000 CPR encoder, we expect only 2 encoder ticks to go by per sample. Occasionally we would count 1 or 3 ticks from the encoder. This translates into a huge distance offset of 50% and caused the PID controller to compensate for the offset. Even minimal compensation would cause the loop to become unstable for 5 to 10 samples as it tried to maintain 2 ticks per sample. This was apparent in the music as audible wavering, particularly noticeable when playing a pure tone through the turntable.

To solve the problem, we used the "1/T method" [4,9] and calculated the amount of time that occurs between each encoder tick instead the number of encoder ticks that occur in a given amount of time (in our case, 1 ms). This was implemented through the use of a callback routine and one of the features of COMEDI, our data acquisition API. We ran the two channels of the encoder through an XOR gate and sent that signal into the interrupt pin of the parallel port. Then we generated a small realtime callback routine that invoked itself each time the XOR'd signal went from low to high. The routine read a timer on the

PC's motherboard and calculated the number of nanoseconds that had elapsed since the last time this routine was called. It stored this value in shared memory so that it could be used by our 1kHz haptic loop to calculate the velocity. It was important not to do too much work inside the callback routine as it needs to be very fast in order to work properly. A simple subtraction between the current time and the previous time was all that was needed and proved to be fast enough. When playing the turntable at high speeds, we returned to our original method of calculating velocity as too many interrupts caused the system to freeze.

It should be noted that this callback routine was only invoked when the signal from the XOR gate went from low to high. The encoder actually advanced again when the same signal switched from high to low. We were unable to find a suitable method to create an interrupt for every encoder tick and settled for generating interrupts on every second encoder tick and dividing our result by two. Although this is not perfectly accurate, it is a significant improvement over estimating the velocity from the number of encoder ticks that go by in 1 ms as was attempted initially. This method successfully provided us with a constant velocity so as to remove any noticeable jitter in the resulting music.

7.1.3 Controlling the Queue Slider

We use a PID controller on position to set the queue slider's motor. We want the slider to travel along the track for the length of the song so we calculate where we are in the song as it plays and then drive the slider's handle to a certain position based on our song position. Calculating the desired position is done in user space as the song plays. This is stored in shared memory and the haptic loop simply feeds this value into its PID and causes the queue slider to update its position.

7.2 The User Interface in User Space

Within user space, we run our soft-realtime portion of the D'Groove system. A monitor program checks the velocity of the turntable by accessing the shared memory that is updated by the realtime kernel processes. The monitor is also connected to our media player, the Alsaplayer, through a socket connection. The monitor sends a playback value to the Alsaplayer based on the calculated velocity and the Alsaplayer takes care of playing back a song at the proper speed and direction.

A simple user interface allows the user to select various modes for the turntable, such as regular turntable mode, spring mode, sine wave bumps or muddy patches (see section 8 for more details). With each mode, the user can alter various parameters, such as spring forces and dampening. The user can also alter the PID parameters and get feedback on the current position and velocity of the turntable.

All musical playback aspects of the system are handled by the Alsaplayer. It features a playlist where songs can be loaded sequentially and selected with the mouse. It also features some standard components of a media player such as starting, stopping and pausing a song as well as adjusting volume and balance.

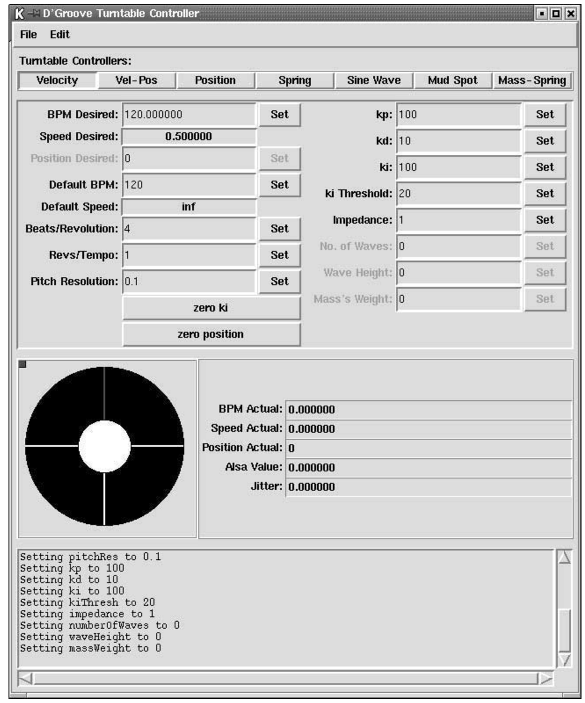

Figure 8: The D'Groove software interface

Before loading a song into the Alsaplayer, we run it through a scanning process to extract important information about the song, which will later be used in the realtime haptic effects. This includes the song's tempo as well as the location of any interesting musical events. The tempo is needed to set up the relationship between the rotational velocity of the turntable and the tempo of the song. The musical event information will be mapped haptically onto our turntable and queue slider. Thus, we run each song through a low band pass filter and conduct a Fourier analysis to ascertain these details. We followed the work of Masataka Goto and Yoichi Muraoka [10], who implemented a realtime beat tracking system to calculate tempo information. We combined their work with some of our own algorithms to retrieve the desired information.

8. Haptics Enabled Creative Effects

The use of haptic feedback in the turntable and the queue slider allows a wide variety of creative effects. We have constant access to the position and velocity of each motor at any given time, and can run these values through crafted controllers to create distinct textures and content-related effects on each motor. A few of these are described here as examples of the new possibilities afforded by this platform.

8.1 Bumps for Beats

We can produce sine waves on the turntable in such a way that they feel like bumps. We can control the height and number of bumps around the turntable as the user moves the platter. Each bump can correspond to a beat in the song, enabling the user to feel the beat structure as they move the platter. Using our visual markers (see Figure 3) and standard 4/4 dance music, beats and bumps often occur when the 1 marker is lined up with N, E, S and W markers. Finding a certain beat is as easy as spinning the platter until the desired bump is felt. Then the user can switch back into regular turntable mode and the velocity PID kicks in, causing the turntable to continue rotating on its own.

8.2 Scratching with a Spring

Scratching, a kind of performance that most DJs do with their regular turntable, involves altering the playback direction of a regular piece of vinyl so the needle travels along the groove, back and forth. The result is a unique sound. With haptics, we can create new scratching techniques. For example, turning the turntable into a rotational spring with different damping and inertial parameters allows the DJ to pluck at the platter instead of scratching it back and forth. A spring controller is easy to implement using the position PID and locking the desired position to a certain point along the platter. We can adjust the damping and spring forces of the platter to give the DJ various types of springs from which to choose. As the turntable is in spring mode, the music plays back and forth along with the spring actions, sounding either very much like professional fast scratching, or – with different model parameters – like something that couldn't be done on a conventional pro system.

8.3 Muddy Music

Again, using dampening forces with a velocity controller, we can make the turntable easy or difficult for the user to rotate. By incorporating the previously extracted musical information, heavily dampened "muddy" spots can correspond to musically "heavy" (or frequency rich) moments in a song. Lighter musical moments (breaks) can be accompanied with less dampening on the turntable motor. Thus the user can feel musical events in the song when in this mode.

8.4 Haptic Cueing

We also implemented haptics on the queue slider. As this slider is meant to replace the needle arm on the turntable, we needed a way for the DJ to select certain movements or events in a song. Using a needle on vinyl, the DJ can place the needle at a random area on the record to play back a sound. DJ vinyl is organized so that the pieces of a song that contain lots of music (runs) have grooves that are compressed together. Lighter moments in the song (breaks) have grooves spread further apart. DJs can see this and place their needle accordingly. Therefore, we added haptics to the queue slider so that they could feel these events along the track. Really musical (frequency rich) bits cause the slider to feel heavy and have lots of friction. Lighter bits make the slider feel lighter. Thus, DJs can move the slider along the track until they get to a point that feels right and then let go, causing the slider to resume its movement along the track as the song plays. The haptic force feedback is only enabled when the user is squeezing the pressure sensor on the slider.

9. Response from the DJs

Upon completion of the system, we were eager to have some DJs try it out and give us feedback. DJs are known to be quite stubborn when it comes to new DJ technology, most refusing to give up their turntables for new controllers such as the line of DJ CD players that are available on the market today. This is mostly due to the high level of control that a conventional turntable affords.

We hoped that with a controller that mimics the functionality of a turntable and provides some new features, we could attract the professional DJ community.

After demonstrating D'Groove and allowing the audience to experiment with it, we were met with an overall excited round of applause. People were impressed that we were able to playback (and scratch) digital music in the same manner as a regular turntable. The haptics impressed the DJs as it provides something new to the turntable interface. The spring and sine wave bumps were the most popular haptic tricks, although for beatmatching, most DJs still preferred to use our turntable in standard turntable mode.

Our users also enjoyed the infinite range of the pitch slider. This feature allowed them to play a song back at any speed, which is not possible with a regular turntable. The queue slider was of interest but most people wanted their needle arm back. We sensed that a lack of visual grooves on the turntable was a deterrent in the system. Cueing was a little difficult with the queue slider as accuracy was an issue. The users did, however, like being able to see how much time remained in a song by viewing the position of the queue slider along its track.

Finally, all users agreed that the turntable platter size was too small. We had anticipated this but due to time constraints, we were unable to find a suitable motor to drive such a large platter and still maintain stability. With the system now running smoothly, we can upgrade to larger motors (and platters) and work on the cosmetic appearance of D'Groove.

10. Conclusions and Future Work

With the encouraging response from our users, we are confident that we have created a unique and expressive tool for DJs to mix and scratch music. The haptics adds a new level of sensory inclusion and binds more of the user's abilities with the tool.

In future iterations, we plan to upgrade the size of our turntable so it looks more like a standard turntable. We would also like to find a way to project virtual grooves on the turntable platter so DJs can get a visual indication of their music from the platter. Creating the grooves on a nearby monitor is easy but we feel that this information needs to be contained within the unit itself. A discussion of projecting grooves from above was discussed although our real wish (with an extensive budget), is to have a backlit LCD platter with colored grooves displaying musical information.

To summarize, D'Groove is significant for the following reasons:

1) It is interactive and performance driven.
2) It combines the user's senses of touch, sight and sound for a multi-sensory experience.
3) It adds a new channel of information and communication for the user.
4) It is complete and self contained.
5) It is a rich, immersive and expressive tool.
6) It is highly entertaining.

11. Acknowledgements

We would like to thank Ben Forsyth for his help on the beat tracking and Fourier analysis of the music. We would also like to thank the Natural Sciences and Engineering Research Council of Canada (NSERC) for providing the funding necessary to work on this project.

12. References

[1]ALSA, http://www.alsa-project.org, August 28, 2002.
[2]Alsaplayer, http://www.alsaplayer.org, August 28, 2002.
[3]Andrew Morton's Low Latency Audio Patch, http://www.linuxdj.com/audio/lad/resourceslatency.php3, August 28, 2002.
[4] Bhatti, P., Hannaford, B., "Single Chip Velocity Measurement System for Incremental Optical Encoders", *IEEE Transactions on Control Systems Technology*, November, 1997
[5] Chu, Lonny., "Using Haptics for Digital Audio Navigation", *Proceedings of ICMC 2002*, ICMA, pp 118-121.
[6] COMEDI, http://stm.lbl.gov/comedi, August 28, 2002.
[7] Finalscratch, Stanton, http://www.finalscratch.com, August 28, 2002.
[8] Hansen, Kjetil, "Playing the Turntable. An Introduction to Scratching". *Speech Music and Hearing Quarterly Progress and Status Report*. Stockholm. 2001, Volume 42. pp 69-79.
[9]Lemkin, M., Yang, P. H., Huang, A. C., Jones J., Auslander, D. M., "Velocity Estimation From Widely Spaced Encoder Pulses." *American Control Conference*, Seattle, Washington, June, 1996, pp. 998-1002.
[10] Masataka Goto, Yoichi Muraoka. "Parallel Implementation of a Beat Tracking System – Real-time Musical Information Processing on AP1000", *Transactions of Information Processing Society of Japan*, July 1996, Volume 37 No 7, pp 1460-1468.
[11] Pioneer CDJ-1000, Pioneer Audio, http://www.pioneerprodj.com/products/cdj1000/cdj1000.mv, August 28, 2002.
[12] Real Time Linux. http://www.fsmlabs.com. August 28, 2002.
[13] Verplank, Bill, "A Course on Controllers", *NIME Worklshop at AIGCHI 2001*, Seattle, WA, Online at http://www.nime.org