Disky: a DIY Rotational Interface with Inherent Dynamics

Karl Yerkes
University of California, Santa Barbara
yerkes@mat.ucsb.edu

Greg Shear
University of California, Santa Barbara
greg.shear@gmail.com

Matthew Wright
University of California, Santa Barbara
matt@create.ucsb.edu

ABSTRACT

Disky is a computer hard drive re-purposed into a do-it-yourself USB turntable controller that offers high resolution and low latency for controlling parameters of multimedia performance software. Disky is a response to the challenge "re-purpose something that is often discarded and share it with the do-it-yourself community to promote reuse!"

Keywords

turntable, dial, encoder, re-purposed, hard drive, scratching, inherent dynamics, DIY

1. INTRODUCTION

We describe *Disky*, a USB turntable controller, as a do-it-yourself project offering high resolution, low latency performance and built using household tools and inexpensive components. Firmware for the microcontroller and example software are available online further lowering the technical barrier for entry.

Consumer waste such as personal computers, VHS machines, and CRT monitors is discarded by the ton every year making these materials inexpensive and easy to find – ideal for do-it-yourself projects. Specifically, Disky utilizes the mechanical parts found in obsolete hard-disk drives meeting all of our design criteria: accessible, reliable, low-cost, and high-performance. In this, Disky is a response to the challenge "re-purpose something that is often discarded and share it with the do-it-yourself community to promote reuse!"

We begin by describing the device's hardware, software, and latency, then describe several applications including turntable scratching and non-obvious uses, and end with diagrams and step-by-step instructions for building a Disky.

2. THE DEVICE

Disky (Figure 1) is a rotatable control surface built from a common computer hard drive. When the surface is rotated, position, direction, speed, and presence of human touch are reported to the host computer.

Disky's rotating platters have the inherent ability to store physical energy that will dissipate through friction. The sound produced by many acoustic instruments decays over

Figure 1: A Disky

time; this behavior is commonly imitated in electronic synthesizers, e.g., with ADSR envelopes. Mapping Disky's rotational speed to loudness is one way to achieve this traditional behavior, and other mappings may also return interesting results. The tactile aspect of these dynamics is also important: the inertia of the system resists change to its state requiring proportional physical effort.

2.1 Hardware

There are several components used to modify a hard drive to create Disky. Two reflective photo interrupters located beneath the hard drive platters face upward to "read" a quadrature encoder pattern (Figure 9). This pattern is printed on paper and glued to the underside of the bottom platter. The photo interrupters come pre-assembled on a chip (Pololu enc01a 0J1216) with two Schmitt triggers that condition the signal for the digital inputs of a microcontroller. Though this is a rather expensive part (approximately \$15 USD), it greatly simplifies assembly.

An Atmel QT113 detects change in capacitance caused by human touch anywhere on the hard drive platters or chassis. The touch sensor and two Schmitt trigger digital outputs are connected to separate input pins on an ATMEGA based PRJC Teensy microcontroller package. All of these components are small enough to be secured to the hard drive platform eliminating long, vulnerable connecting wires and simplifying the form factor.

NIME2010, 15-18th June 2010 Sydney, Australia

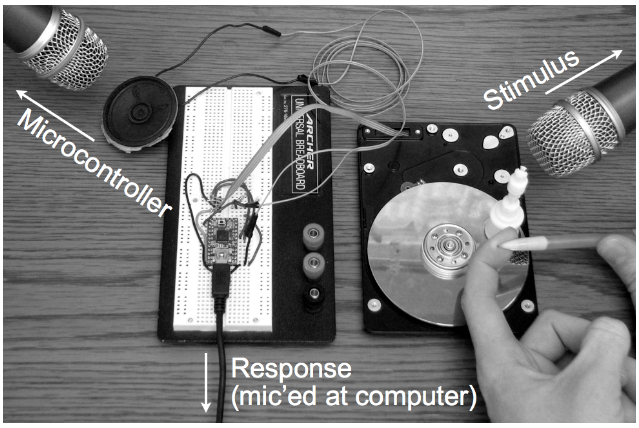

Figure 2: Latency test apparatus

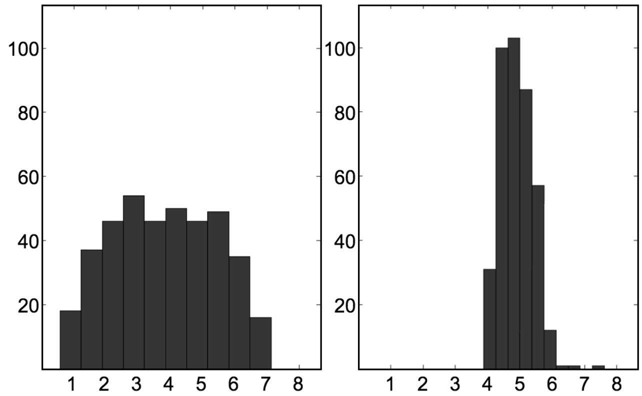

Figure 3: Histogram of latency in milliseconds from stimulus to microcontroller (left) and from microcontroller to audio output (right)

2.2 Software/Communication

Communication between the microcontroller and the host computer is accomplished with the PJRC rawhid library. This library provides fine control over queuing, timeouts, and bandwidth guarantees to allow low-latency communication.

2.2.1 *Microcontroller Firmware*

The main loop of the firmware checks the state of the pins that read the quadrature pattern and the touch sensor. This loop has a worst-case period of approximately 30 us. The system state (position, velocity, and presence of touch) is calculated and packed into a 16 byte packet. This packet is placed in a queue of size 1, which is checked by the host at a period of approximately 1 ms. The queue is overwritten with a new packet each time the system state changes, so only the most recent packets get to the host computer.

2.2.2 *Host Computer Software*

After opening the Disky device, the host-side software must check the USB HID queue for Disky packets at some period less than or equal to 1 ms or packets will accumulate in the queue. In the Max/MSP and Pd examples, this means banging the [disky] object with a [metro] object using period of 1 ms or less. The other examples use a thread that repeatedly reads from the USB HID queue and then sleeps for 900 us.

2.3 System Latency

Low latency is critical for a performer's ability to express through an interactive real-time instrument. A major design goal was to identify significant delay along the control signal path and minimize it wherever possible.

We measured the end-to-end latency using the stereo-digital-recorder paradigm [7] in which the acoustic sound from physically contacting an input device, the *stimulus*, is recorded to the first channel of a stereo audio track, and the audio output of the host application, the *response*, is recorded to the second channel. Both of these signals are recorded using standard microphones connected to a single audio interface such that error introduced by this technique affects all of the data equally and cancels differentially. A MATLAB script then finds the time between events in the two channels of a long stereo recording containing many stimulus/response pairs. This method provides "an indisputable measure of total system latency not based on any software's opinion of the current time" [7].

Manually setting a turntable in motion requires the user to touch the platter and thrust in the desired direction of rotation, relying on the friction between skin and the platter to transfer energy. This interaction is not inherently percussive and lacks the appropriate attack necessary for this analysis method. In order to produce a more percussive sound, a plastic spindle was glued to the top platter and used as a striking surface (Figure 2). A simple flick with a mechanical pencil was sufficient to set the disc in motion, the sound of which was easily recorded by a microphone into the *stimulus channel*.

In order to separate the latency of the Disky hardware from the latency of the USB connection and sound-producing application software, we created an intermediate signal by modifying the firmware to toggle the logic state of an output pin on the microcontroller whenever a signal was detected from the photo interruptors (immediately before sending a HID message to the host computer). The output pin was wired directly to a small audio driver that, with each state change, produced a definite click. This sound was also easily detected by another microphone and recorded into the *microcontroller channel*.

Finally, a lightweight Max patch output an impulse through the computer speakers with each HID message it received. This audio signal was recorded by a third microphone into the *response channel*.

From these three channels the MATLAB script determined the latency of the system as split in two stages. The first stage, between the stimulus and the microcontroller signal, contributed 4 ms latency on average with a standard deviation of 1.7. The second stage, between the microcontroller signal and the audio output from the host computer, contributed 5 ms latency on average with a standard deviation of 1.7 (Figure 3). The end-to-end latency of the entire system was 9 ms on average with a standard deviation 2.4. Out of the 401 data points there were 4 outliers in the first stage between 7 and 8 ms, and 8 outliers in the second stage between 9 and 24 ms. These points, all more than 2 standard deviations above the mean, were removed for visual clarity of the histogram, but their presence in these results calls for further investigation. The host computer was an Apple MacBook Pro 2.2 GHz Intel Core 2 Duo running Mac OS 10.6 and Max 5.0.6.

10 ms maximum latency and ± 1 ms maximum jitter are common target values for real-time computer-based instruments [6]. Disky's total average latency of 9 ms falls within this range, but the variability of the latency is well above the 1 ms target. We suspect that the outliers in the second stage are operating system anomalies and are beyond the current scope of this project. A comparison of the his-

tograms suggest that most of the jitter is introduced before the microcontroller signal and is likely a result of the varying starting position of the encoder pattern with respect to the optical sensor. Consider the situation in which an edge in the encoder pattern is immediately adjacent to the optical sensor: movement in one direction will produce a signal almost instantly, whereas movement in the opposite direction will produce no output until the platter rotates enough for the far edge to reach the sensor. It follows that the jitter of this stage is dependent on the speed of the gesture being performed, is only present when starting from rest, and may be reduced by increasing the resolution of the encoder pattern.

3. APPLICATIONS

This section presents a sampling of explorations that make use of Disky's defining attributes, both inherent and intentional: touch sensitivity, infinite rotation, and the ability to store and dissipate kinetic energy over time.

3.1 Scrubbed Media

Turntables were initially made popular as performance instruments by DJs, and Disky's most obvious use is in emulating an analog turntable for mix or scratch DJing. The user's interaction with Disky will certainly differ from that with a motorized turntable, however alternative modes of operation can compensate for this difference. The first is "Transport Mode" where the accumulative angular position of the platter maps directly to playback position in the recording. This mode allows the user to move quickly back and forth through the music in search of an appropriate transition point (otherwise known as "cueing"). Once this point has been found the user simply removes her hand from the conductive platter (and hence, the capacitive touch sensor) immediately resuming playback. Disky's low latency and responsiveness to subtle movements allows for scratching in this mode. We would like to perform user studies with turntable DJs to see how well their skills translate to Disky, particularly in light of the size difference between a Disky platter and a 12 inch record.

The second mode of operation is "Nudge/Drag Mode" in which platter speed temporarily alters playback rate. The act of mixing often requires the performer to re-align the rhythmic phase of two songs playing at the same tempo, and this is achieved on a traditional motorized turntable by nudging the record forward in the direction of rotation, or dragging a finger along the edge of the platter, momentarily increasing or decreasing playback rate. This interaction is approximated by mapping Disky's instantaneous speed to change in playback rate: clockwise motion temporarily increases the tempo by a small amount, and counterclockwise motion decreases it.

We acknowledge HDDJ, another turntable interface made from a re-purposed hard drive [5]. Instead of a printed encoder pattern and optical sensors, HDDJ determines the platter's speed from the oscillating voltage created when the electric motor is spun by hand. With fewer modifications necessary, HDDJ is an easier project and therefore more accessible to the DIY user. However the motor produces a signal only once every 30 degrees, a much lower resolution than Disky. According to the HDDJ developers this drawback makes the interface suitable only for mixing, not scratching.

Similar to the DJ application, Disky may be useful in audio/video editing workstations as a transport device. We envision a non-linear mapping of the accumulative angular position of the platter to cursor position, such that faster rotational speeds cause a larger change in cursor position per change in angular position.

3.2 Granular Synthesis

This instrument was designed for live granulation of a predominantly melodic source signal, e.g., piano or voice, to explore the space between harmonic and textural sounds. The audio engine continuously records the input signal into a 10-second circular buffer from which grains are read.

The qualities of a physical interface to a software synthesizer can easily come as an afterthought leaving the user to interact with the system via a QWERTY keyboard and pointing device or a generic bank of sliders and knobs. It was an important design goal to allow performance that resembled neither "office work" [8] nor "recording studio work," for the interface to visually reflect the state of the synthesis model, and for the output sound to decay to silence when the performer's hands leave the instrument.

Disky's own physical dynamics add sonic character that is unspecified by the software: platter speed is mapped to grain density such that high speeds produce many grains triggered in rapid succession. A spinning disc sustains the sound output absent persistent user input, but friction gradually removes energy from the system. As the disc slows, fewer and fewer grains are produced, their separation in time increasing until the disc stops and no more grains are played. Of course, the user controls the disk's motion by hand and may manually sustain or damp the motion in performance.

A long (50 cm) pressure-sensitive touch strip augments the hardware for this application: position on the touch strip is mapped to position within the source sample from where grains are sliced, and pressure determines both grain amplitude and attack time, where high pressure results in loud grains with fast attacks. Stereo position and width can also be adjusted through the GUI but are not intended for live manipulation during a performance.

Strengths of the system include interaction where the size and forcefulness of a gesture is reflected in the output making clear the relationship between the visual and the audible aspects of the performance, and minimal reliance on the GUI allows for eye contact between performers. It was difficult, however, to determine current position within the granulation buffer based on finger position on the strip before any sound was produced. Also, friction between the performer's finger and the touch strip during high pressure sliding gestures was problematic but was easily, though perhaps not elegantly, remedied by a cotton glove.

Inherent dynamics of a physical system were also used in a granular synthesis context by Hartman et al. in the Swing Set, an apparatus of three pendulums [3]. The motion of one pendulum was mapped to the motion of the cursor in a granulation buffer, and the position of the other pendulums were mapped to additional synthesis models, the outputs of which were all mixed together. The decaying oscillations of the pendulums and the resulting rhythms contrast to Disky's behavior and a system employing both may prove interesting with the proper mappings.

3.3 Rotational Scanning Synthesizer

Written in openFrameworks, this synth interprets images from a live camera (Figure 4). It scans through a circular section of the image along a radius whose angle is controlled by Disky. Intensity values along this radius are mapped to a wave-table which is played at a constant period. It is intended that a whiteboard, sketchbook, or some other drawing apparatus be used as the live image, so that this system can allow the composer/performer to map images

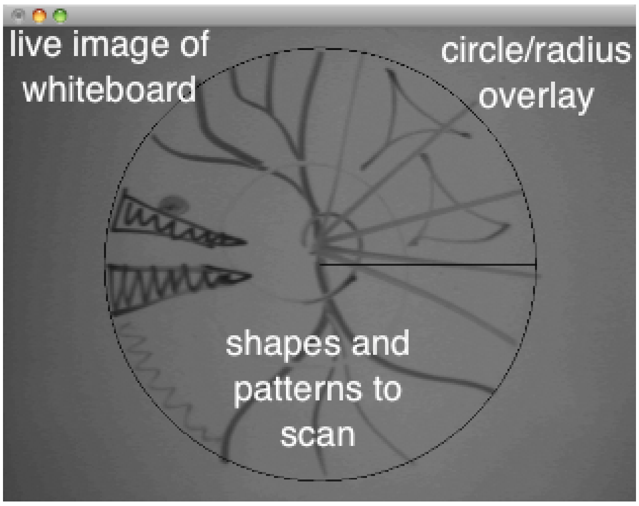

Figure 4: Live image scanning synthesizer

Table 1: Parts to buy

Description	Manufacturer	Part	Price
Wheel Encoder	Pololu	1217	$14.95
Microcontroller	PJRC	teensy	$18.00
Touch Sensor	Atmel	QT113	$4.00
Hard drive	–	–	$2.00
Total	–	–	$38.95

in a 2d polar space to sound. This is somewhat similar to the Circular Optical Object Locator [2] but with explicit control of angle rather than a constant rotation and with a different sound mapping.

Future work on this synth will include mapping radius color values to the amplitudes of partials (additive synthesis) and to filter coefficients (subtractive synthesis).

4. CONSTRUCTION

This section shows how to make a Disky as a DIY project.

4.1 Parts / Cost

Table 1 is a list of parts to buy and Table 2 is a list of tools, supplies and resources to secure in preparation for construction.

4.2 Find. Dismantle. Save.

The first step in building Disky is to find and dismantle several hard drives. Many parts in 3.5 inch computer hard drives are useful for Disky and other projects. Collect the bushings, platters, screws, magnets, spindle assemblies, motors, and circuit boards.

Extra platters are useful when gluing patterns to platters goes wrong. While gluing patterns, motors are used as guides. Bushings serve as spacers when adjusting the number of platters on a Disky and are also useful when adjusting the distance between the sensors and the pattern.

Future work on Disky will use the spindle assembly of the hard drive to produce sonic and tactile feedback. Consider leaving the spindle assembly in place.

Table 2: Materials, Tools, Equipment

Description	Manufacturer	Part	Price
Calipers	Harbor Freight	93293-1VGA	$19.99
Super glue	Devcon	29005	$2.00
Spray glue	Elmer's	E-451	$5.99
Soldering iron	–	–	–
Quality printer	–	–	–
Total	–	–	$27.98

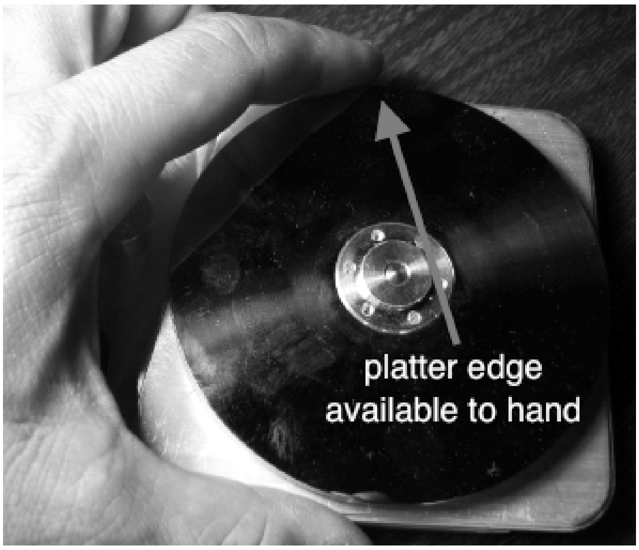

Figure 5: A hard drive that is good for Disky

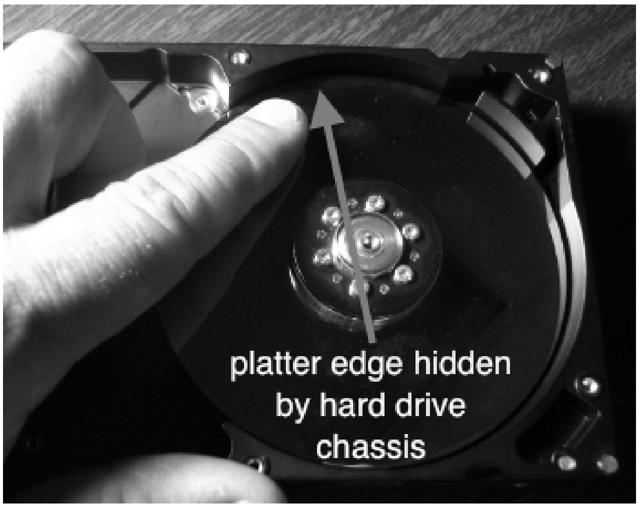

Figure 6: A hard drive that is not so good for Disky

4.3 Select the Best Hard Drive

While dismantling hard drives, look for qualities that would make a good Disky. Choose a drive that is pleasing to look at and feels nice to touch. Generally there are 2 forms of hard drive chassis. One form consists of a relatively flat platform covered by a sort of hat-shaped cap (Figure 5). These are good for Disky because they allow tactile access to the edge of the platters. The other form of hard drive chassis (Figure 6) blocks access to the edge of the platters and they generally have sharp protrusions that could injure the enthusiastic user.

Some hard drive motors have strong magnets that the user can feel as the platters are turned. These prevent smooth, uniform spin and are generally undesirable for Disky. Throw the platters and let them settle. If the platters rotate back and forth before they come to rest, consider using a different hard drive.

The ability to stack more or fewer platters on a Disky

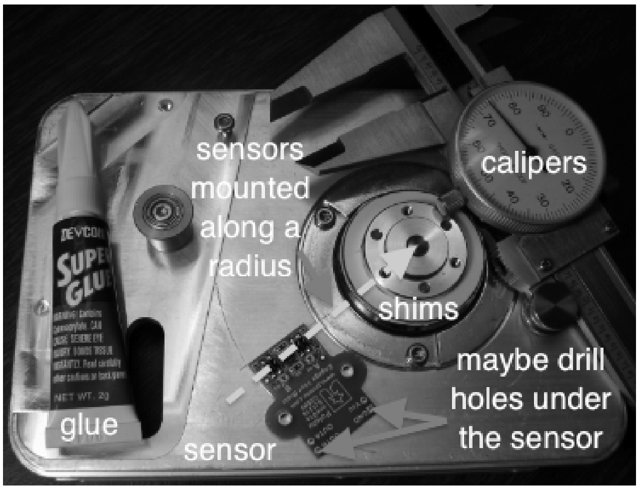

Figure 7: Mounting sensor

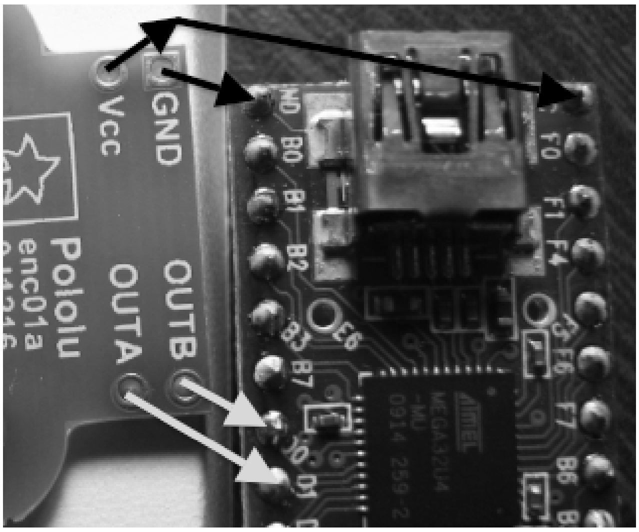

Figure 8: Soldering connections

allows the user to change the inertial characteristics. If this is desirable, look for a hard drive that can accept a variable number platters.

4.4 Steps for building

4.4.1 Mount Sensor Assembly

Inspect the hard drive chassis for a good place to glue the sensor. Position the sensor near the outer edge of the platter such that the photo-interrupters are in line with a radius as shown in Figure 7. If necessary, drill holes through the chassis to allow space for wiring the sensor to the microcontroller.

4.4.2 Mount/Wire Microcontroller

Solder microcontroller to sensor as seen in Figure 8.

4.4.3 Make/Mount Encoder Pattern

Download the encoder pattern tool from karlyerkes.com/disky. Use calipers to measure the distance from the outer edge of the platters to the center of the outer photo-interrupter as shown in Figure 11. The encoder pattern tool will use this measurement to create a printable pattern. Print the pattern on non-glossy, white paper using a high-quality printer with deep blacks. With a razor knife, cut out the center

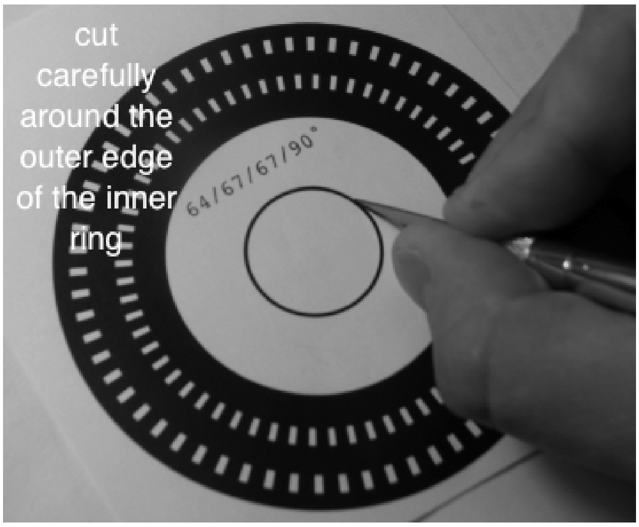

Figure 9: Cutting pattern

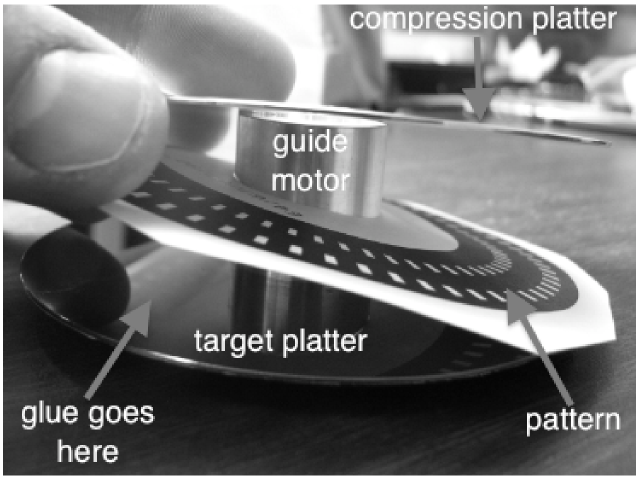

Figure 10: Gluing pattern

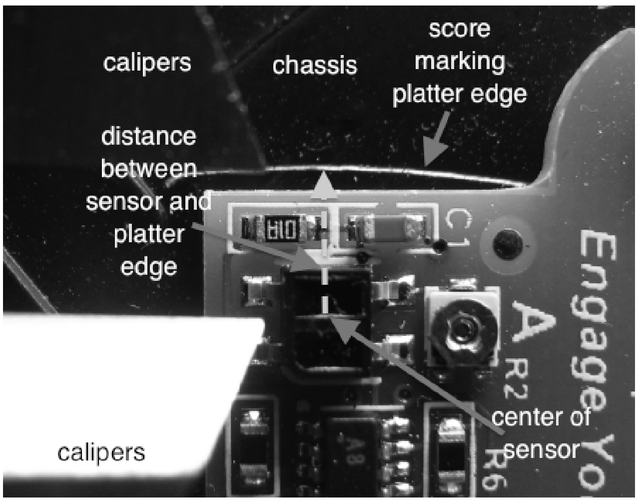

Figure 11: Measure from sensor center to platter edge

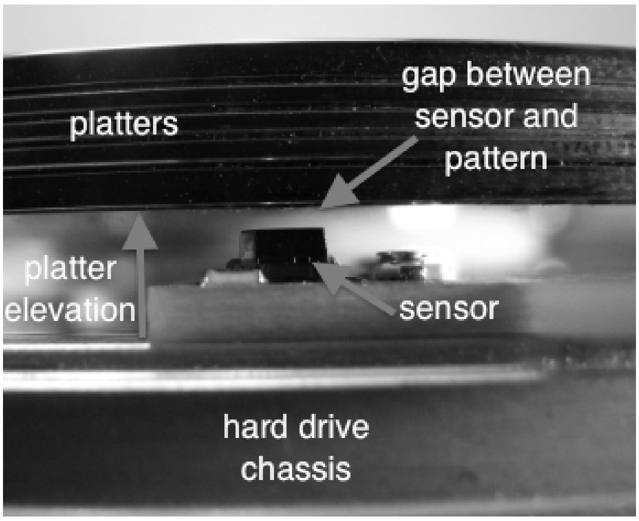

Figure 12: Platter elevation

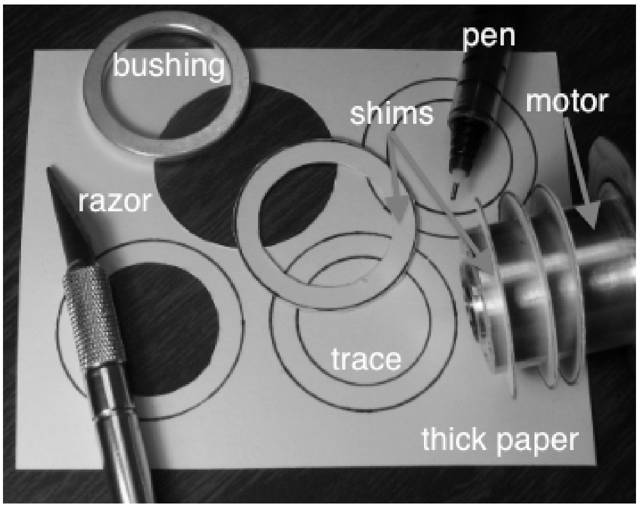

Figure 13: Making shims

of the pattern as shown in Figure 9. Cut carefully along the outside of the printed circle to get a uniform, centered circular cut.

Place a platter on a free motor as seen in Figure 10. Spray a very thin layer of glue on this platter, using a spent toilet-paper roll to protect the motor from the glue. Guide the prepared pattern onto the motor and then use another platter to compress it against the gluey platter.

4.4.4 Adjust Pattern Elevation

The reflective photo-interrupters work optimally when they are mounted approximately 0.6 mm from the encoder pattern (see Figure 12). Adjust this distance using shims made from thick paper (see Figure 13). Make sure the platters are tightened down, then measure the gap with your eye by comparing to calipers set to 0.6 mm. Take care not to tighten the bottom platter down onto the optical sensor.

4.4.5 Test

Connect Disky to your computer with a USB cable. Download and run one or more of the Disky examples found at karlyerkes.com/disky.

5. FUTURE WORK

In future work we will attempt to implement course and fine control of the variable reluctance motor to complete the turntable metaphor. We will try to implement haptic force-feedback using this motor in the style of D'Groove [1] and Verplank's PLANK [4]. We will increase the resolution of the pattern and plan a formal user evaluation.

6. REFERENCES

[1] T. Beamish, K. Van Den Doel, K. MacLean, and S. Fels. Dgroove: A haptic turntable for digital audio control. *Proc. of ICAD, Boston, MA*, 2003.

[2] T. Hankins, D. Merrill, and J. Robert. Circular optical object locator. In *Proceedings of the 2002 conference on New interfaces for musical expression*, page 2. National University of Singapore, 2002.

[3] E. Hartman, J. Cooper, and K. Spratt. Swing Set: Musical Controllers with Inherent Physical Dynamics. In *Proceedings of the 2008 conference on New interfaces for musical expression*, 2008.

[4] B. Verplank, M. Gurevich, and M. Mathews. The Plank: designing a simple haptic controller. In *Proceedings of the 2002 conference on New interfaces for musical expression*, page 4. National University of Singapore, 2002.

[5] N. Villar, H. Gellersen, M. Jervis, and A. Lang. The ColorDex DJ system: A New Interface for Live Music Mixing. In *Proceedings of the 7th international conference on New interfaces for musical expression*, page 269. ACM, 2007.

[6] D. Wessel and M. Wright. Problems and prospects for intimate musical control of computers. *Computer Music Journal*, 26(3):11–22, 2002.

[7] M. Wright, R. Cassidy, and M. Zbyszyński. Audio and Gesture Latency Measurements on Linux and OSX. In *Proceedings of the ICMC*, pages 423–429, 2004.

[8] D. Zicarelli. Music for mind and body. *Electronic Musician*, 7(2):154, February 1991.