One-Dimensional Force Feedback Slider: Going from an Analogue to a Digital Platform

Ali Shahrokni, Julio Jenaro, Tomas Gustafsson, Andreas Vinnberg, Johan Sandsjö, Morten Fjeld
Tabletop Interaction Lab (www.t2i.se), IDC, CSE, Chalmers TH, SE-412 96 Göteborg
{ali, julio, tomas, andreas, johan, morten}@t2i.se

ABSTRACT
This paper examines the use of motorized physical sliders with position and force as input and output parameters for tangible human computer interaction. Firstly, we present an analogue platform. It was used to realize two proof-of-concept applications: one for learning system dynamics as part of physics education and the second for interaction with music loops. Based on the insight gained with the analogue platform and the two applications, we took the first steps towards a digital platform, also presented here. More generally, the paper presents so-called haptic modes, which may be generated using force feedback control of motorized sliders. The paper also briefly presents parts of the underlying software and hardware which was designed and realized as part of this project.

Author Keywords
HCI, User interface design, physical prototyping, haptic interface, force feedback, slider, TUI

ACM Classification Keywords
H.5.2 User Interfaces: Tangible user interfaces (TUI), Haptic interface, prototyping

INTRODUCTION

It is known that human beings have a very fast operating sense of touch [1] and several research projects [2][3] have capitalized on this fact. Interactive systems with motorized sliders have been suggested in the iStuff [4] and Phidgets [5] projects, as well as in commercial joysticks with force feedback. Haptic display for so-called 3D haptic widgets has been examined [6]. Most of the existing approaches have proposed motorized sliders for output only, with no dynamic control of output force or registration of user input force. These devices with force feedback are not attuned to human tactile sensing and the feedback is mostly simulated by vibrations of different frequency and intensity.

To explore a wider use of force feedback with one-dimensional motorized sliders, we have started a project called Force Feedback Slider (FFS). Motorized physical sliders with position and force as input and output parameters lend themselves well to tangible interaction (Fig. 1). Their capacity to support direct [7] tangible interaction can be proved at several levels. While the directness of 3D interaction has been widely explored, there is a need for more direct interaction with multivariate models. For instance, the externalization of abstract mathematical models was studied by Tweedie et al. [8], as they examined a yield enhancement function of upper and lower bounds.

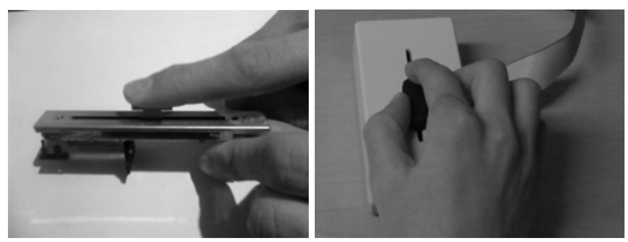

Figure 1. Motorized slider (left) and application (right).

From a human's perceptual point-of-view, the absence of tactile feedback in GUI interfaces may overload our visual senses and lead to increased physical strain. While GUI input is largely limited to discrete operations such as typing and selecting, a mouse affords some degree of continuous control. However, since mouse interaction occurs mainly without physical resistance and tactile response it is not very effective. End-users who draw, sculpt, or create continuous audio lines for long periods of time using a mouse often become frustrated and experience pain. There are negative health consequences to the fixed posture and small repetitive movements required by working with a GUI interface [3]. We admit that one-dimensional sliders will not solve this problem. However, the combination of alternative task-specific UIs, including actuated sliders, may be an improvement.

The next section presents an analogue FFS platform, including its software, followed by two proof-of-concept applications and some discussion. This is followed by a section on a digital FFS platform, including its hardware and software. The final section discusses the results, relates them to the other projects [4][5], and envisions the next steps in the FFS project.

ANALOGUE PLATFORM

The first version of the FFS is a layered analogue system which is connected to the computer through a USB interface. The software is written in Java which maximizes compatibility with different operating systems and Java Applets. Using this device as a proof of concept, two distinct applications were realized: the first for direct interaction with a projectile motion model, *Catapult* (Fig. 2) [2], and the second for direct manipulation of sound during playback in a program, *FeelTheBeat* (Fig. 3) [9].

Software of the Analogue Platform

The slider is designed in several layers which have been previously described in a separate paper [10]. The physical layer consists of an external USB sound card and the slider board. The sound card is used to control the slider board and send commands to the slider. It also functions as an AD/DA converter. To stabilize the movement of the slider a P-regulator is attached to the slider board.

The driver layer is in direct contact with the sound card and sends and receives the commands directly from the analogue hardware. Depending on which haptic mode[1] the application requests a slider to run, the driver uses different lookup tables (LUT) to control the behavior of the slider. Each LUT contains the force to be applied by the slider depending on the user force and the position of the slider.

The slider should be compatible with major operating systems and be easy to develop new applications for. To achieve this goal an application program interface (API) layer is provided in Java. This API allows the writing of an application for the FFS without any knowledge of how the slider fundamentally works. Most operating systems support Audio-Class-USB and have available device drivers. Also, the API is executable in the Java Virtual Machine. Thus the portability between the different systems is successfully achieved.

Applications with the analogue platform

As mentioned in the previous sections, two applications were realized to prove the functionality of the analogue system.

Catapult [2]

This is a secondary education (K-12) application allowing for direct manipulation and testing of a catapult (Fig. 2). Physics education often relies on the visualization of theoretical laws to enhance the learning experience, and while Java animations are widespread, they generally lack user interaction. So, FFS could be advantageous to any application used to interact with the law of physics. Here, the users receive both tactile and visual feedback using the software. We conjecture that a UI calling upon two perceptual channels at the same time may help users to more easily construct a mental model of the subject-matter content. This application used the API.

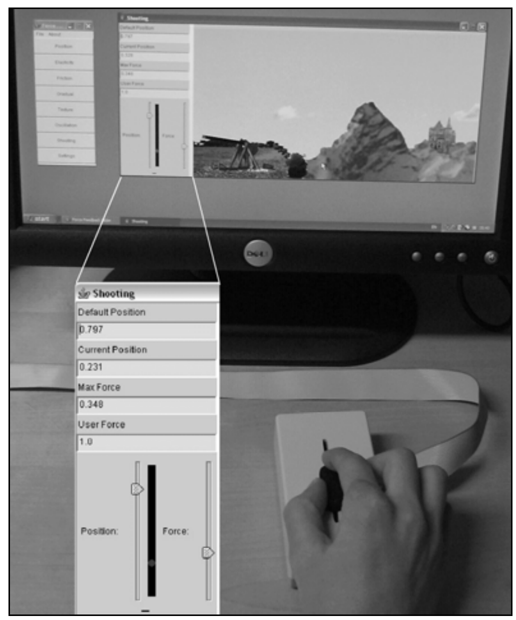

Figure 2: The FFS in operation with catapult.

FeelTheBeat [3]

This is an application allowing for direct manipulation of sound during playback (Fig. 3). Computers are widely used in music performance and production. DJs increasingly use computers rather than analogue turntables and mixers [11]. Musicians use sequencing software in composition and ubiquitously employ computers in their productions. Sequencing software

[1] FFS offers five haptic modes: In *position,* the slider is used only as input device, the motor is switched off, and the user can move the slider without FF. In *elasticity*, a default position and a variable resisting force are defined. The user's fingers have to overcome the resisting force. When the handle is released it returns to the default position. *Gradual* offers a number of discrete steps into which the handle can snap. In *texture*, high-frequency low-intensity vibrations are applied to the handle, thus giving users an impression of a rough surface. In *oscillation*, the handle comes to rest after a damped sine movement. These five haptic profiles are abstract descriptions of elementary capacities; other applications can be composed from these modes. Any finite function that can be written as a mathematical expression is allowed [2].

[2] http://www.t2i.se/pub/media/ShootingDemo.avi

[3] http://www.t2i.se/pub/media/FeelTheBeat.mpeg

offers the ability to arrange and transform music, primarily in an offline situation, but with notable exceptions such as Ableton Live, which is designed for live performance. The aim of this application is to develop a tangible user interface for common sequencing operations such as the looping of a sound. Samples of duration between 1 to 8 beats corresponding to 0.5 to 8 seconds are used in this case. The interface allows the display and modification of sound during playback and should operate in a direct way [7]. Furthermore, the slider moves with different accelerations depending on the amplitude of the sound that is played, generating different perceivable forces. The proposed interface employs a loudspeaker and a motorized slider [12] [9], thus offering continuous audio, visual, and haptic cues during playback. The slider handle moves according to a predefined temporal audio parameter and thus gives immediate and continuous feedback corresponding to the current playback state. When the user holds or moves the handle, the audio parameter changes and the audio playback is altered accordingly. As the sound software applied here was written in C++, this application communicated directly with the sound card, bypassing the Java API.

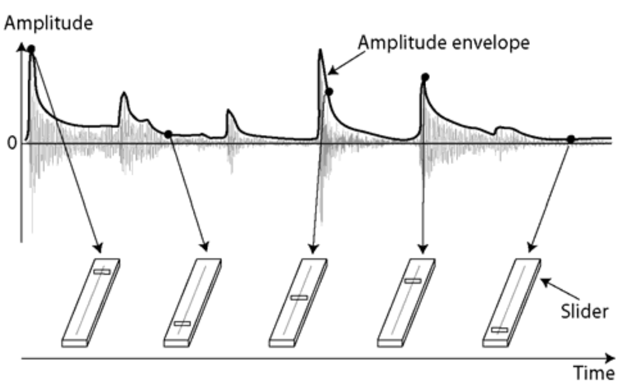

Figure 3. Amplitude envelope of sound and slider position.

Analogue platform: discussion

The analogue slider proved the validity of the idea. By implementing a prototype system and testing its operation, the way for improving the system opened up.

However, the analogue system exhibited a high latency due to the fact that the data was processed through the operating system, an external sound card, and the slider board. Conversion between analogue and digital data and the operating system's sound mixer were responsible for the largest parts of the delay. The design was not sufficiently stable and depended on low-level operative system settings. Nevertheless, the analogue platform served as proof-of-concept. The next step was digitalizing the system and combining several sliders to perform a task.

DIGITAL PLATFORM

Based on previous experience and project aims, we formulated a set of *use requirements* for the digital FFS platform: i) low latency, ii) high stability, iii) platform independency, iv) extendable to maximum of 16 sliders mounted into a compact box, v) programmable via a standardized API, vi) remote haptic collaboration enabled.

The goal of digitalizing the system was to preserve its advantage, solve the shortcomings presented above, and thereby meet the use requirements. The most critical problem is the high latency which causes the user to experience the force change as discrete and not continuous. Another major problem is system instability. An additional requirement is to make the system independent of operating system configurations and settings. The digital solution is still under construction but the tests conducted so far have shown promising performance and much shorter delays. In the following section we present and discuss hardware and software issues of the digital platform.

Hardware of the digital platform

The digital version of the slider is designed in a modular way and each slider can operate either individually or within a combination of several sliders. Each slider has a µcontroller operating at a clock frequency of 16 MHz. It has several ADC converters digitalizing the analogue data much faster than the sound card in the analogue version. This reduces the latency of the slider considerably as compared to the analogue system. The discreteness should be undetectable by the human sense of touch.

The platform should allow the simultaneous operation of several sliders side by side (Fig. 4). If each slider was directly connected to the computer it would complicate the system and increase the latency. Therefore, a mainboard was designed that can control up to 16 sliders simultaneously. It operates with the same kind of µcontroller as each slider and at the same frequency. The mainboard and the sliders are connected by an I2C bus [13] which operates at a frequency of 400 KHz. The sliders operate independently and therefore the bus is only used to read the force or position of each slider or to send LUTs to each slider. Thus, the amount of data transferred by the bus is relatively low and should not slow down the sliders.

Using in-house designed technology, the µcontroller which drives the slider motor is able to determine approximate forces applied by the user's hand and to generate different resisting or active/driving forces. A specific force is generated by sending the corresponding current to the slider motor.

Software of the digital platform

Similar to the analogue design, the digital design has two different layers of software: the µcontroller code and the API code, which is the interface for the application developer.

µcontroller programming

Each slider has a LUT stored in its memory according to which it operates. Depending on the position of the handle and some other factors, the slider generates a specified force by applying the suitable current required by the slider motor.

The mainboard µcontroller keeps track of the current position and force of each slider. It can also send a new LUT to the sliders using the I2C bus whenever a slider's operating mode changes.

Application Programming Interface (API)

As in the analogue version, the API simplifies the development of new applications. Due to the same reasons mentioned for the analogue system, the API is written in Java. To make its use easier, the API inherits the standard class (component) JSlider [14] which is the standard soft slider class in Java library. To program, the developer either chooses a standard operation mode or provides a force function to be simulated by the slider. The slider can either be resistive, active/driving, or both. This means that it may resist against the user force (up to a certain level given by the motor used), automatically move with different forces, or combine these two depending on the position of the slider handle.

Digital platform: discussion and outlook

We have achieved positive results in the design and realization of the digital platform. In the next steps, we foresee improvements in usability and robustness of the hardware, the protocol, and the API. We also intend to examine system performance related to human perceptual factors such as haptic profile resolution, system feedback, latency, and stability.

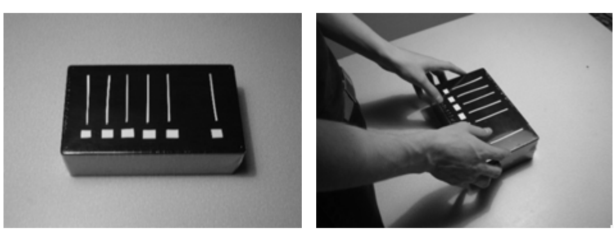

Figure 4. Mock-up of multi-slider interactive device (left) and the same mock-up device in use (right)

SUMMARY

We have presented an analogue and a digital realization of the Force Feedback Slider (FFS). Most of the features offered by iStuff [4] and Phidgets [5] such as USB interface, modularity, and use of physical sliders are or will be offered by the FFS. In contrast to commercial joysticks with force feedback, FFS is planned to offer accurate customized force feedback matching real world settings, such as mechanics and music, more realistically. In summary, we expect the FFS to be a versatile force feedback device for developers and end-users.

ACKNOWLEDGEMENTS

Analogue platform realized by Adjan Kretz and Remo Huber. FeelTheBeat realized with Tue Haste Andersen. Sliders partly sponsored by Fournier Electronic (CH). Thanks to http://www.eta.chalmers.se for cooperation and support in hardware assembly.

REFERENCES

[1] Verplank, B., Gurevich, M., and Mathews, M., "The PLANK: Designing a simple haptic controller", Proc. of NIME, 2002.

[2] Kretz, A., Huber, R., and Fjeld, M., "Force Feedback Slider: An interactive device for learning dynamic system behavior", Proc. ICALT05, 2005, pp. 457-458.

[3] MacLean, K.E., Shaver, M.J., and Pai, D.K., "Handheld Haptics: A USB Media Controller with Force Sensing," Proc. Symp. on Haptic Interfaces for Virtual Environment and Teleoperator Systems, 2002, pp. 311-318.

[4] Ballagas, R., Ringel, M., Stone, M., and Borchers, J. 2003. "iStuff: a physical user interface toolkit for ubiquitous computing environments", Proc. of CHI '03. pp. 537-544.

[5] Greenberg, S. and Fitchett, C. 2001. "Phidgets: easy development of physical interfaces through physical widgets", Proc. of UIST '01, pp. 209-218.

[6] Miller, T. and Zeleznik, R. 1999. „The design of 3D haptic widgets", Proc. SI3D '99. ACM, pp. 97-102.

[7] Shneidemann, B., *Designing the User Interface*, 1993.

[8] Tweedie, L., Spence, R., Dawkes, H., and Su, H. "Externalizing Abstract Mathematical Models", Proc. CHI96, 1996, pp. 406-412.

[9] Andersen, T.H., Huber, R., Kretz, A., and Fjeld, M. (2006). "Feel the Beat: Direct Manipulation of Sound during Playback". Proc. TableTop2006, pp. 123-124.

[10] Kretz, A., Huber, R., and Fjeld, M., "Architecture of force feedback slider", ETH E-Collection, 2005. Available at http://e-collection.ethbib.ethz.ch/show?type=semarb&nr=65

[11] Beamish, T., Maclean, K., and Fels, S. "Manipulating music: multimodal interaction for DJs", Proc. CHI, 2004, pp. 327-334.

[12] Newton-Dunn, H., Nakano, H., and Gibson, J., "BlockJam", Abstract, Proc. SIGGRAPH, 2002.

[13] http://www.esacademy.com/faq/i2c/general/i2cproto.htm

[14] http://java.sun.com/j2se/1.4.2/docs/api/javax/swing/JSlider.html