Fortissimo: Force-Feedback for Mobile Devices

Tae Hong Park, Oriol Nieto
Music and Audio Research Lab
New York University
New York, NY, USA
{tae.hong.park, oriol}@nyu.edu

ABSTRACT

In this paper we present a highly expressive, robust, and easy-to-build system that provides force-feedback interaction for mobile computing devices (MCD). Our system, which we call *fortissimo* (*ff*), utilizes standard built-in accelerometer measurements in conjunction with generic foam padding that can be easily placed under a device to render an expressive force-feedback performance setup. *fortissimo* allows for musically expressive user-interaction with added force-feedback which is integral for any musical controller – a feature that is absent for touchscreen-centric MCDs. This paper details *ff* core concepts, hardware and software designs, and expressivity of musical features.

Keywords

force-feedback, expression, mobile computing devices, mobile music

1. INTRODUCTION

One of the most interesting developments in the mobile technology area occurred with the advent of robust multi-touch mobile computing devices. These devices created a paradigm shift where mobile phones were no longer phones but rather *smart phones* with seemingly limitless software possibilities and a multitude of on-board hardware features – indeed, there always seemed to be "an app for that." Almost immediately, musicians began to experiment and use MCDs to explore their potential in musical contexts leading to two main areas of interest: software application development such as stand-alone instruments [15] and exploration of MCDs as a musical controller [5]. For most of the applications, the primary means of interaction is through the multi-touch screen (an exception is the Ocarina [15] which also uses mic input to drive the synthesis engine). The touch-screen interface, however, lacks a feature that is very important in any machine-based musical interaction scenario: force-feedback. This type of haptic feedback plays an essential role in musical expressivity for most traditional acoustic instruments (pianos, guitars, percussion instruments, etc.), but its absence in touchscreen-based interfaces, greatly diminishes musical expressivity potential [1].

These observations motivated us to explore adding force-feedback to existing MCDs while keeping four key philosophies in mind: (1) avoidance of physical alteration to the device itself, (2) easiness in adding force-feedback to the device, (3) flexibility in modifying expressivity via hardware and/or software, and (4) exploitation of polysensory features of modern MCDs. *ff* essentially augments expressivity by providing force-feedback for MCDs. This is achieved through simple combination of foam padding and exploiting on-board accelerometers found in many standard MCDs including the iPad. The name *ff* itself has duality in meaning: on one hand it functions as an acronym (force-feedback) and on the other, it has symbolic and expressive musical significance (*fortissimo*).

NIME'13, May 27 – 30, 2013, KAIST, Daejeon, Korea.

1.1 Related Work

For the majority of commercial musical instrument development, a clear effort is made in trying to mimic traditional instruments both in design and functionality. This is especially true in products like Akai's Electronic Wind Instrument (EWI), various MIDI guitars and basses, the V-Accordion by Roland, and numerous flavors of digital drums including the Korg Wavedrum to name a few. In typical academic and "research" circles, however, design criteria have rarely been driven by mechanisms of economic return, but rather, the focus has been on developing "novel" controllers using sensors, microcontrollers, and signal processing/sound synthesis design paradigms. The classic *Radio Baton*[1] developed by Max Mathews (and the *Theremin* before that by Theremin), Michael Waiswisz's *The Hands*[2], Ron Buchla's *Thunder*[3], Dan Trueman's *BoSSA* system [14], the *SBass* developed by Curtis Bahn, Weinberg's Beatbug [16] and Peter Leonard's *Hoola* [7] represent merely a very small sample of new instruments/controllers [13]. The majority of these controllers (with the exception of the Theremin and Radio Baton), allow force-feedback interaction, whether via flex-sensors, force-sensing resistors, or simple potentiometers.

With the ubiquity of MCDs and robust multi-touchscreen technology, a great many number of musical apps have emerged in recent years. Examples are a aplenty [5] and include software for PocketPCs such as iPAQs [4] enabling integrated sound synthesis using a port of Pure Data (PD); the pioneering work of Schiemer and Havryliv with the Pocket Gamelan [12]; and more recently the Ocarina, one of the "All-Time Top 20" iOS apps by Smule [15]. The app and smartphone revolution is seemingly not slowing down any day soon but is, rather, continuously pushing the boundaries of not only what a user can *do* with these devices [11], but what users can *do together* in an ensemble and orchestra setting – a setting where musical interaction and sharing is encouraged [3]. For the majority of the touchscreen-based

[1] https://ccrma.stanford.edu/radiobaton/
[2] http://www.crackle.org/TheHands.htm
[3] http://www.buchla.com/

interactions, force-feedback is absent although it is essential for musical expressivity – whether in the context of live performances with electronic instruments in general [9] or when interacting with physically modeled instruments [2].

2. HARDWARE DESIGNS

Musical controllers are primarily *built* in two ways: (1) using off-the-shelf sensors and microcontrollers [10] by either attaching them, permanently or semi-permanently, to existing instruments [6, 8] and (2) by building entirely new designs from "scratch" [7, 14, 17]. These types of controllers have the advantage of allowing for force-feedback but also suffer from issues including fragility and maintenance, mass-production difficulties, accessibility, and cost. With the advent of modern tablets and smartphones, a third method can now be added to the way controllers are "built:" exploiting the inherent polysensory capabilities of an MCD, battery life, wireless networking capabilities, computing power, and its mass appeal. The great majority of MCDs include a reasonably precise built-in accelerometer, alongside with buttons and cameras. Furthermore, as MCDs are mass-produced, robust, and serve many different purposes on a daily basis, its usage is ubiquitous. However, one of the (many?) issues for touchscreen-based MCDs as an expressive musical controller is the unavailability of force-feedback. *ff* provides a simple, yet surprisingly expressive force-feedback solution for touchscreen-based MCDs, ultimately augmenting musical expressivity and control.

2.1 Fortissimo Foam Blocks

The foam paddings we used in *ff* are materials one might find in a bathroom, kitchen, or leftover foam padding designed to protect computers during shipping. We constructed *ff* using foam pads from a new shoebox that we just happened to have at our disposal.

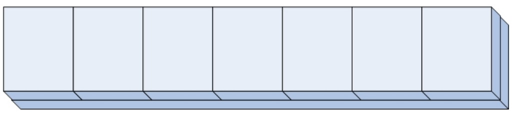

Figure 1: Fortissimo: combined blocks of foam paddings

As seen in Figure 1, *ff* consists of two layers of foam. The top layer is made up of smaller, equal-sized modular foam blocks firmly placed in series on a single foundational foam pad. The smaller and modular top blocks are kept securely affixed to the bottom foundation with velcro as shown in Figure 2. The top foam block configuration allows for quick force-feedback adjustment and feel: removing foam blocks will result in less resistance and smoother force-feedback (and vice-versa). Figure 2 shows three blocks removed to allow for "softer" force-feedback feel – a type of "subtractive-resistance" design approach so-to-speak.

2.2 Typical Configuration

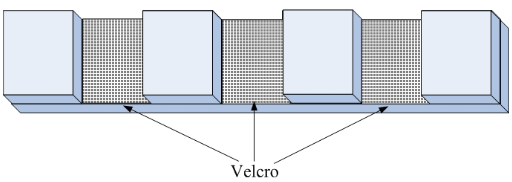

Figure 2: Fortissimo: detachable foam with velcro

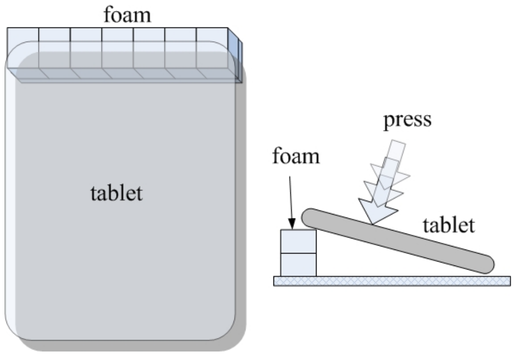

Figure 3: Typical fortissimo setup, from a top view (left) and side view (right)

A typical *ff* setup is shown in Figure 3 and 4 where the block is placed at the edge of the tablet resulting in an angled tablet setup. When pressing on the touchscreen, force-feedback is rendered by the foam's absorption of pressure which is applied by the user during touchscreen interaction. The built-in accelerometer is used to measure the amount of "tilt" with respect to its rest position when no pressure is applied. The accelerometer readings are then mapped to musical parameters such as tremolo, vibrato, and velocity.

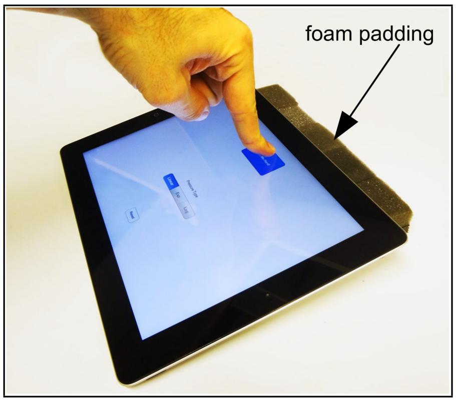

Figure 4: Fortissimo in action

A number of different foam configurations have been explored by taking into account sensitivity, resistance of materials, and tablet angles to render a robust and expressive "default" configuration. Our default configuration ended up as in Figure 4. This configuration rendered the most expressive setup as tilt amount (angle of tablet) was directly proportional to expressive range. However, other configurations are also possible depending on the desired application as further discussed in subsection 2.3.

Other observations of this setup are that the accelerometer's side-to-side/xy axes contribute the most expressive control whereas the "z-axis" is not as sensitive when producing gradual movements. Furthermore, the GUI objects that are closest to the *ff* blocks (highest point on the screen) yield the greatest range of expression and sensitivity due to the maximum possible displacement range rendered by the tilt of the tablet – GUI objects closest to the bottom end are least pressure sensitive.

2.3 Additional Configurations

To allow for more uniform sensitivity, we configured *ff* as shown in Figure 5. This configuration allows for a more

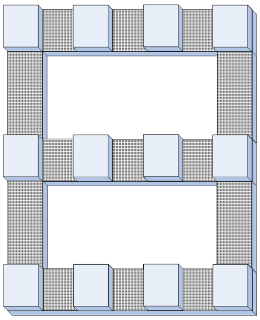

Figure 5: Fortissimo: flat surface configuration

distributed pressure setup providing expressive 2D mapping possibilities. If the pressure is applied at the exact geometric centroid, however, accelerometer readings will yield no change – except during high transient response interactions (e.g. quick strike). In this scenario, the accelerometer's "z-axis," can be used to detect acceleration-type interactions.

A final configuration that we found useful was a flat surface foam padding (e.g. "sponge") secured to the MCD with rubber bands at the top and bottom. This setup enabled the user to comfortably hold the MCD (e.g. a smaller smartphone) in one hand while interacting with the device with the free hand. This performance configuration not only provides force-feedback interaction but also allows great mobility as the user is no longer tied to a flat surface.

2.4 Foam Padding and Sensitivity

We tested a number of foam padding materials which resulted in subtle variances in sensitivity. In general, we noted two key characteristics when using foam padding for force-feedback purposes: foam type (1) determined the amount of resistance which in turn affected user sensitivity, and (2) affected the amount of time the foam took to return to its uncompressed position/shape. We noted that more resistance generally lead to slower recovery time and vice-versa. Softer foam resulted in softer touch, while more rigid foam added resistance and thus "harder" sensitivity. We also found that less resistive paddings rendered wider movement range when coupled with the standard configuration and placement of the *ff* blocks as presented in section 2.2. In our current study, we decided to concentrate on one type of foam padding material that was reasonably rigid while exploring possibilities of quickly modifying "touch sensitivity" via placement and change in configuration as outlined above.

3. SOFTWARE DESIGNS

Our current *ff* software is written for the 3rd generation Apple iPad, which essentially measures accelerometer readings to mimic force-feedback-based user interaction. Before using *ff*, we first set up the system as shown in Figure 3. Once this is established, we calibrate the system for optimal accelerometer readings by pressing the reset button in the GUI. This is a necessary step as the accelerometer in its rest position will vary during each setup procedure. *ff* also provides a second optional calibration procedure to allow for improved expressivity – this entails capturing the maximum displacement of the tablet (at rest to fully depressed) in order to set the maximum accelerometer range for a given session. It is, however, possible to bypass this additional calibration sequence and use the default fixed movement range instead.

3.1 Pressure Sensitivity Options

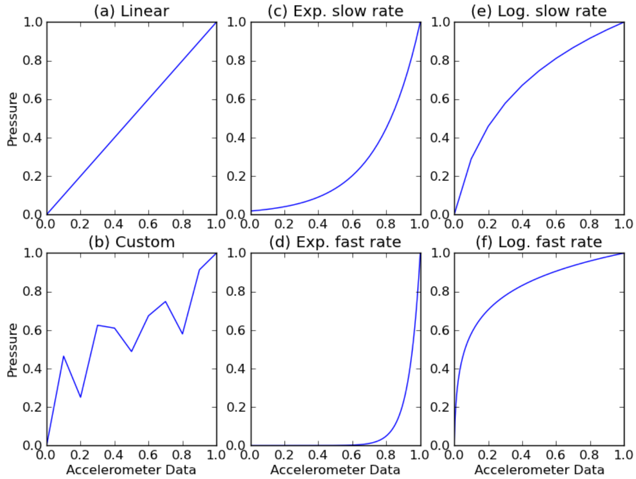

Figure 6: Different pressure types that we have experimented with Fortissimo.

In lieu of physically changing the sensitivity of *ff*, a number of software pressure settings are also available. This is easily accessible as part of our main GUI interface. Figure 6 shows the different types of pressure settings we experimented with where a represents the normalized user input accelerometer data ($a = 0.0$ represents tilt amount at rest and $a = 1.0$ at maximum pressure) and p the output mapping, where $p = 0.0$ denotes zero pressure and $p = 1.0$ the maximum pressure that the system is able to track. Tests were conducted on the Apple iPad app that mapped pressure to the amplitude of an audio signal.

3.1.1 Linear

The linear mode is the simplest mapping function with a $1 : 1$ relation: $p = a$. We found this setting to be particularly expressive in rendering tremolo performance techniques. This is simply achieved by rubbing the GUI button which is similar to guitar playing techniques and akin to channel aftertouch on the Yamaha DX-7. This mode can also easily be used for vibrato performance techniques.

3.1.2 Exponential

The exponential mode is expressed as shown below where λ is the exponential rate and b is the scaling factor.

$$p(\lambda) = b \exp(a\lambda)$$

The lower the λ, the slower the growth will be (see Figure 6c) – smaller λ will be similar to the linear mode. When we increase λ as seen in Figure 6d, a lot of pressure will have to be applied in order to perceive a change in sound. An extreme case is when λ is so high that the *handle* will behave like an *button*.

3.1.3 Logarithmic

This pressure type is the inverse to the exponential mode and is expressed as:

$$p(\lambda) = b \log(a\lambda)$$

where λ is the logarithmic rate and b is the scaling factor. In this case, a small amount of input pressure will result in a dramatic change of output control value. The less the logarithmic rate, the more it will resemble the linear mode (see Figures 6e and 6f). Setting λ sufficiently high will accordingly increase the sensitivity.

3.1.4 Custom

It is also possible to create custom user functions as shown in Figure 6b. One illustrative example of this pressure type would be to map different amounts of pressure to specific changes in pitch, thus creating different melodic structures as pressure is varied.

3.2 Accelerometer

When using the 3rd generation iPad and its on-board accelerometer we were able to get an average sampling rate of 30 Hz and floating point values ranging from approximately -1.271802 to +1.215422 (for full range). A typical reading when using our *ff* system in its default configuration (Figure 3) resulted in a range of 0.0 and 0.2 providing approximately 200,000 possible discrete values of accelerometer readings. This, plus the fact that we get one measurement every 33 milliseconds, makes *ff* very accurate and sensitive in real-time musical applications.

4. DISCUSSION AND FUTURE WORK

Although we have not yet used *ff* in a full-concert performance setting, we have had some promising initial results especially when used in conjunction with TouchOSC which is available for both iOS and Android operating systems. One of our goals is to write a number of short compositions for *ff*. The expressivity that is rendered has been quite surprising as it provides a dynamic, "physical" and "visceral" interaction experience in both the table-top (default) and the hand-held configurations. Rather than creating custom iOS and Android apps with GUI interfaces such as TouchOSC, our immediate future plans are to develop software to capture OSC control data, remap the stream using our *ff* software, and transmit to a software synthesizer on the computer side. We anticipate that this will enable users easy access to pressure sensitive sliders, pan-pots, buttons, and other standard GUI interface objects for greater musical expression. We are also planning to integrate velocity trigger modes with the "handle" modes for seemless expressivity and control exploiting the z-axis of the accelerometer. This will essentially allow for *handle+button* interaction for every GUI object.

Other areas in which *ff* could potentially be useful is in computer games (e.g. car racing games in which the acceleration would be measured by the amount of pressure of a specific button), or other utility applications (e.g. paint brush where the amount of pressure determines the width of the brush).

Finally, we are in the midst of exploring various GUI interfaces for *ff* and plan to do a user evaluation of our proposed systems.

5. CONCLUSIONS

We have presented *fortissimo* (*ff*) which provides force-feedback for touch-screen-based mobile computing devices with a simple, yet effective set of foam paddings that are placed under a typical MCD with a built-in accelerometer. The accelerometer readings effectively capture user exerted pressure on the touchscreen in real-time. The foam padding and accelerometer combination allows for robust force-feedback interaction with MCDs resulting in a more expressive and physically meaningful interaction setup. We analyzed a number of mappings between user interaction (pressure) and different musical parameters such as tremolo and vibrato, while exploring various foam padding materials, placements, shapes and sizes.

6. ACKNOWLEDGMENTS

The authors would like to thank Langdon Crawford in helping test *ff*, *Fundación Caja Madrid* for the funding, and Dr. Dennis Shasha for the iPad.

7. REFERENCES

[1] C. Bahn and D. Trueman. Physicality and Feedback : A Focus on the Body in the Performance of Electronic Music. In *Proc. of the International Computer Music Conference*, pages 44–51, La Habana, Cuba, 2001.

[2] E. Berdahl, J. O. Smith, and G. Niemeyer. Feedback Control of Acoustic Musical Instruments : Collocated Control Using Physical Analogs. *The Journal of the Acoustical Society of America*, 131(1):963–73, Jan. 2012.

[3] N. Bryan, J. Herrera, J. Oh, and G. Wang. Momu: A mobile music toolkit. In *Proc. of the International Conference on New Interfaces for Musical Expression*, Sydney, Australia, 2010.

[4] G. Geiger. PDa : Real Time Signal Processing and Sound Generation on Handheld Devices. In *Proc. of the International Computer Music Conference*, Singapore, Singapore, 2003.

[5] G. Geiger. Using the Touch Screen as a Controller for Portable Computer Music Instruments. In *Proc. of the International Conference on New Interfaces for Musical Expression*, pages 61–64, Paris, France, 2006.

[6] A. Kapur, A. J. Lazier, P. Davidson, R. S. Wilson, and P. R. Cook. The Electronic Sitar Controller. In *Proc. of the International Conference on New Interfaces for Musical Expression*, pages 7–12, Hamamatsu, Japan, 2004.

[7] P. Leonard. *HOOLA: A Circular Digital Musical Interface*. Master's thesis, Tulane University, 2011.

[8] M. Levin and A. Woo. Tactile-Feedback Solutions for an Enhanced User Experience. *Information Display*, 25:18–21, 2009.

[9] T. Michailidis and J. Bullock. Improving Performers' Musicality Through Live Interaction with Haptic Feedback : A Case Study. In *Proc. of the 8th Sound and Music Computing Conference*, pages 227–232, Padova, Italy, 2011.

[10] T. H. Park. Introduction to Music Performance Systems. *Journal Society of Electro-Acoustic Music in the United States*, 19(1):2–9, 2006.

[11] M. Rohs, G. Essl, and M. Roth. CaMus : Live Music Performance using Camera Phones and Visual Grid Tracking. In *Proc. of the International Conference on New Interfaces for Musical Expression*, pages 31–36, Paris, France, 2006.

[12] G. Schiemer and M. Havryliv. Pocket Gamelan : a Pure Data interface for mobile phones. In *Proc. of the International Conference on New Interfaces for Musical Expression*, pages 156–159, Vancouver, Canada, 2005.

[13] S. Sinclair. *Force-Feedback Hand Controllers for Musical Interaction*. Master's thesis, McGill University, 2007.

[14] D. Trueman and P. Cook. BoSSA: The Deconstructed Violin Reconstructed. In *Proc. of the International Computer Music Conference*, Beijing, China, 1999.

[15] G. Wang. Designing Smule's Ocarina : The iPhone's Magic Flute. In *Proc. of the International Conference on New Interfaces for Musical Expression*, pages 303–307, Pittsburgh, PA, USA, 2009.

[16] G. Weinberg, R. Aimi, and K. Jennings. The beatbug network: A rhythmic system for interdependent group collaboration. In *Proc. of the International Conference on New Interfaces for Musical Expression*, pages 107–111, Dublin, Ireland, 2002.

[17] Y. Yang, Y. Zhang, S. Member, Z. Hou, Z. Chen, and B. Lemaire-semail. FingViewer : A New Multi-Touch Force Feedback Touch Screen. In *Proc. of the IEEE International Conference on Consumer Electronics*, pages 837–838, Las Vegas, NV, USA, 2011.