HSP: A Simple and Effective Open-Source Platform for Implementing Haptic Musical Instruments

Edgar Berdahl
CCRMA
Stanford University
Stanford, CA, USA

Günter Niemeyer
Telerobotics Laboratory
Stanford University
Stanford, CA, USA

Julius O. Smith III
CCRMA
Stanford University
Stanford, CA, USA

Abstract

When we asked a colleague of ours why people do not make more haptic musical instruments, he replied that he thought they were "too hard to program and too expensive." We decided to solve these perceived problems by introducing HSP, a simple platform for implementing haptic musical instruments. HSP obviates the need for employing low-level embedded control software because the haptic device is controlled directly from within the Pure Data (Pd) software running on a general purpose computer. Positions can be read from the haptic device, and forces can be written to the device using messages in Pd. Various additional objects have been created to facilitate rapid prototyping of useful haptic musical instruments in Pd. HSP operates under Linux, OS X, and Windows and supports the mass-produced Falcon haptic device from NovInt, which can currently be obtained for as little as US$150. All of the above make HSP an especially excellent choice for pedagogical environments where multiple workstations are required and example programs should be complete yet simple.

Keywords: haptic musical instrument, HSP, haptics, computer music, physical modeling, Pure Data (Pd), NovInt

1. Prior Work

Haptic musical instruments have been developed since the 1970's at the latest. Much of the work has focused on the design of accurate haptic devices and physical modeling of virtual musical systems [1][2]. More recently, Sinclair [3] has controlled a SensAble Omni haptic device from Pd by employing elements from the CHAI 3D [4] and the Open Dynamics Engine (ODE) [5] libraries. However, the haptic feedback loop is not closed directly around messages passed in Pd; rather, Pd sends OSC messages to a haptics server that computes the forces to be applied to the haptic device.

Figure 1. Musician holding the Falcon grip

Currently CHAI 3D is fully supported only on Windows, but there is also a beta release for GNU/Linux.

We decided to instead develop our own platform because we wanted a truly cross-platform solution, and we wanted to process the haptic signals directly in Pd due to its simplicity, instructional value, and the wide array of predefined objects for manipulating control messages in Pd. By employing this method for processing haptic control data, we cause ourselves to think using many of the computer music metaphors that motivated the development of Pd itself and its extended libraries.

2. HSP

High resolution haptic devices have been becoming more accessible to musicians due to advances in gaming and minimally invasive surgery. We currently recommend that musical instrument designers use the NovInt Falcon, as shown in Figure 1, which was originally designed for gaming and can deliver forces as large as 9 N. The position of the Falcon grip (see Figure 1) is transmitted to a computer, and in response, the computer can command forces acting on the grip in 3D space.

Machulis' new open-source Falcon driver operates under Linux, Windows, and OS X [6]. There is no fully cross-platform open source driver for any other mass-produced haptic device capable of providing kinesthetic feedback. In fact, the flext external object *np_nifalcon* allows the Falcon to be controlled directly from Max/MSP or Pd [6]. We provide a wrapper object called *Falcon* to hide some of the complexity of the underlying object, while still providing modularity for future updates of HSP allowing other haptic

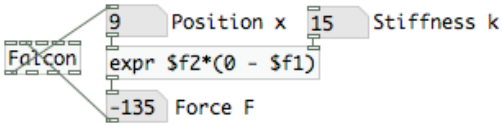

Figure 2. Spring implementation in HSP

devices to be accessed.

It is important that the reader note how easy it is to install HSP since in contrast with many other approaches, nothing needs to be compiled and only one driver needs to be installed. The only required elements are the Pd patches we provide [7], the Falcon external object [6], and Pd extended.

3. Examples

3.1. Spring

The spring patch demonstrates how HSP's simple philosophy of presenting the dynamics implementation directly leads to concise, easy-to-parse patches. Figure 2 shows how a one-dimensional spring ($F = -kx$) can be implemented in Pd using the *expr* object, where the rest position of the spring is 0. Notice how simple the Pd patch is—only one object carries out any computations.

A related algorithm implements a wall at $x = 0$:

$$F_{wall} = -kx \cdot (x < 0). \quad (1)$$

3.2. Haptic Drum

The patch *drum.pd* simulates a wall, and whenever the musician presses into the wall, a drum sample is played back with an amplitude scaling proportional to the velocity. If the musician enables "ASSISTANCE", then the instrument becomes unstable in a special manner enabling the musician to play especially fast drum rolls using only a single hand.

3.3. Force Profile

The graphical array object in Pd provides for a convenient interface for exploring nonlinear force functions of position. Using springs in the X-axis and Z-axis, the Pd example patch *force-profile.pd* restricts the motion of the Falcon grip to the Y-axis, causing the Falcon to behave like a 1DOF haptic device. The force in the Y-axis is programmed to be a function of the Y-position [8]. As illustrated in Figure 3, the function is stored in a user-editable graphical array that can be modified with the mouse. At the center of the array, a horizontal slider (hslider) shows the Y-position of the Falcon (see Figure 3, horizontal stripe in the middle). In this particular example, the user has drawn a force profile into the table resulting in two stable equilibrium points, toward which the Falcon grip is pushed by the control system. The stable equilibrium points have been marked in the image by way of small blue circles (see Figure 3).

3.4. More Examples

Documentation covering more intricate examples as well as actual musical instruments can be found on the project website [7].

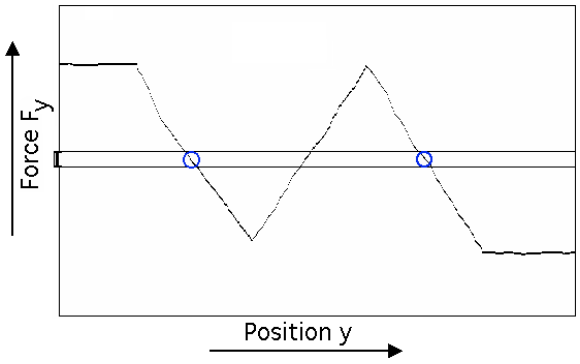

Figure 3. A force profile stored in a graphical *array* in Pd

4. Conclusions

HSP is an open-source platform that allows musicians to easily prototype haptic musical instruments. Since the haptic control signals are processed as messages within Pd, instrument designers are most likely to make use of the same metaphors, such as the graphical array, that motivated the development of Pd itself and its extended libraries.

We believe that HSP is simple to operate and that the examples included in the package demonstrate many of the ways in which haptics can be useful in musical instrument design. Accordingly, we have purchased six NovInt Falcons and created an instructional laboratory exercise around them for the CCRMA class entitled "Physical Interaction Design For Music" [9]. The example patches serve as excellent pedagogical material because they are simple and can be easily understood by students. In contrast with purpose-built platforms, HSP supports direct programming of the dynamics, so students learn to clearly comprehend the dynamics behind the physical interactions.

References

[1] C. Cadoz, A. Luciani, and J.-L. Florens. "Cordis-anima: A Modeling and Simulation System for Sound and Image Synthesis—The General Formalism," *Computer Music Journal*, 17(1):19–29, Spring 1993.

[2] J.-L. Florens. "Real-Time Bowed String Synthesis with Force Feedback Gesture Interaction," In *Proc. of the Forum Acusticum*, 2002.

[3] S. Sinclair and M. Wanderley. "A Run-Time Programmable Simulator to Enable Multi-Modal Interaction with Rigid-Body Systems," *Interacting with Computers*, 21(1-2):54–63, January 2009.

[4] http://www.chai3d.org/

[5] http://www.ode.org/

[6] http://sourceforge.net/projects/libnifalcon

[7] http://ccrma.stanford.edu/~eberdahl/Projects/HSP

[8] B. Verplank. "Haptic Music Exercises," In *Proc. of the Int. Conf. on New Interfaces for Musical Expression*, pages 256–257, Vancouver, BC, Canada, 2005.

[9] http://ccrma.stanford.edu/courses/250a/labs/Hapticslab