



**UNA PROPUESTA DE CONTROLADOR MUSICAL PARA
LA GENERACIÓN Y PROCESAMIENTO DE AUDIO
DIGITAL MEDIANTE LA GESTUALIDAD DE LAS MANOS**

Ortega, Nicolás

Enicolas.ortega@gmail.com

Tesis de grado

Escuela Universitaria de Artes.

Licenciatura en Música y Tecnología

Director de la Carrera

Dr. Esteban Calcagno

Director de Tesis

Lic. Martin Matus Lerner

RESUMEN

La propuesta de ésta tesis consiste en el diseño y desarrollo de un instrumento musical digital; un guante que funcione como una interfaz para facilitar la creación de música electrónica o el procesamiento de audio digital en tiempo real, utilizando como controlador de los parámetros los gestos con las manos que realice el intérprete.

El diseño de dicha interfaz está basado en recursos *Open source* y la filosofía *Do It Yourself* (en inglés, *Hazlo tú mismo*), un movimiento con orígenes contraculturales proveniente de Inglaterra que alienta a los desarrolladores de cualquier ámbito creativo y niveles de experiencia a fabricar por cuenta propia todo lo que se pueda¹.

Para el desarrollo del guante se le incorporaron sensores de torsión que capturan los datos de los movimientos de la mano, el material del que están hechos cambia sus cualidades dieléctricas dependiendo de la presión que se le ejerza y, por ende, su conductividad varía al torcerse. Posteriormente, los datos recolectados por los sensores son convertidos en señales eléctricas interpretables por una placa microcontroladora *Arduino* y, luego, enviadas mediante comunicación serie al entorno de programación *Pure Data*. De ésta manera se pudo cuantificar y mapear los registros gestuales y utilizar los valores adquiridos para controlar distintos parámetros dentro del software.

Se decidió utilizar comunicación a través de puertos serie virtuales para interconectar los programas. Este tipo de comunicación es más lenta que la comunicación en “paralelo”, pero también permite un traspaso de datos bidireccional entre los programas y necesita menos líneas de transmisión, consumiendo menos recursos y evitando correr riesgos de interferencia o desincronización.

¹ Permanent Culture Now (2013). [«Introduction to DiY/counterculture»](#). Consultado en febrero de 2016.

Previo al proceso de fabricación se realizaron pruebas comparativas entre los sensores caseros y los industriales, los resultados determinaron la viabilidad del uso de los primeros; al arrojar datos estables, repetibles y fiables, se procedió a utilizar los sensores de fabricación casera para el armado del guante, como era la idea original.

Como parte del diseño de la interfaz se le agregó un módulo con un vibrador y un LED RGB, éste cumple la función de feedback para así poder monitorear lo que está ocurriendo en el patch con sólo atender los códigos de colores y vibraciones en la propia mano. De ésta manera el uso del guante se volvió más libre, al no tener el performer la necesidad de observar constantemente la computadora.

Introducción

En la actualidad, la *revolución del “Hazlo tú mismo” (do it yourself)*, permite que cualquier persona con una computadora tenga la oportunidad de desarrollar no solo su propia interfaz física, sino también el software adecuado para su funcionamiento.

La interfaz deberá cumplir con las características descritas por el autor Marcelo Wanderley en su artículo *“Gestural control of music”* sobre sistemas de control multiparamétricos. En él se explican los atributos que se deben considerar al momento de utilizar sensores, estos atributos son precisión, exactitud, sensibilidad, estabilidad y la capacidad de repetir los resultados.

La placa microcontroladora Arduino es una plataforma electrónica de código abierto basado en hardware y software fáciles de usar. Es una placa electrónica programable, de bajo coste, que sirve para la creación de prototipos electrónicos, interactivos y multidisciplinares.²

Pure Data es un lenguaje de programación visual de código abierto que permite a músicos, artistas visuales, investigadores y desarrolladores crear software de forma gráfica, sin necesidad de escribir líneas de código. Se utiliza para procesar y generar sonido, vídeo, gráficos 2D / 3D, sensores de interfaces, dispositivos de entrada y MIDI.³

Antecedentes y estado del arte

La creación del Theremin en 1919 marca el nacimiento de la música electrónica y, al mismo tiempo, el comienzo de una carrera por darle mayor gestualidad a la música, aunque es a partir de los años '50 que el arte comienza a vincularse más con la tecnología. Previo a eso los avances tecnológicos eran poco aplicables al arte y de difícil acceso, lo cual había mantenido a artistas de todas las disciplinas alejados de la idea de nuevas interfaces. (Kogan, 2008)

La performance en el escenario siempre fue el momento de mayor importancia para la expresión musical⁴, es cuando el músico hace llegar su obra al oyente sin intermediarios. Esto para la música electrónica siempre ha sido un desafío, dado que sus obras nacen en

² Hushner, David. "The Making of Arduino." *IEEE Spectrum*. Oct. 2011. Web. 09 Oct. 2012. Disponible en <http://spectrum.ieee.org/geek-life/hands-on/the-making-of-arduino/0>

³ PD-Community. *Pure Data*. Disponible en <http://puredata.info/>

⁴ Christopher Dobrian y Daniel Koppelman definen a la Expresión Musical como la "representación del humor o del sentimiento del interprete" en su trabajo *Musical expression with new computer interfaces*, presentado en la conferencia NIME del 2006.

estudios o laboratorios, y no era fácil reproducirlas en vivo, menos aún de manera expresiva y gestual.

Los avances tecnológicos han permitido el desarrollo de música electrónica en tiempo real, pudiendo así subirla a los escenarios. Uno de estos avances es la creación del protocolo MIDI, que proporcionó una manera de crear nuevos instrumentos que fueran accesibles y utilizables en el escenario para lograr que la música electrónica sea más gestual y realmente tangible.

A finales de la década del '70, aparecen los sintetizadores digitales, y junto con ellos un nuevo problema: la incompatibilidad entre los sistemas utilizados por cada compañía fabricante. Como solución a esta problemática se crea un lenguaje común, por encima de los parámetros de cada marca. En 1983 se publica e introduce el estándar MIDI (*Musical Instruments Digital Interface*), un protocolo diseñado para transmitir solamente datos de eventos y mensajes de control (en vez de señales de audio), que puedan ser interpretados arbitrariamente de acuerdo con la programación particular de cada dispositivo.

En 1984, Michael Waisvisz, hace uso de este nuevo sistema y da nacimiento a su primer interface experimental. *The Hands* utilizaba diversos sensores y botones, montados sobre unas estructuras de madera que sostenía el intérprete con sus manos. La información recibida por los sensores era convertida en mensajes MIDI.

Esta manera de capturar el movimiento de las manos, brazos y dedos lo volvió único en su tipo y dio nacimiento a una nueva generación de músicos performers, que buscaban darle gestualidad a una música de laboratorio. Interfaces así le brindaban al performer no solamente la posibilidad de moverse sobre el escenario, sino de expresar a través de su propio cuerpo lo que se estaba componiendo en tiempo real, pudiendo expresarse de manera más gestual, sensible e incluso sensual. Algo que hasta ese momento no se asociaba en absoluto

con la música electrónica debido a que hasta el momento se trabajaba con grandes sintetizadores analógicos inamovibles, atados a los primeros estudios.⁵

En 1991, Laetitia Sonami construye sus propios guantes, llamados *Lady's glove*. Un guante de goma de cocina al cual se le habían adherido transductores y un imán. Al tocar el imán, las variaciones en los voltajes eran convertidos en señales MIDI gracias a una placa Sensorlab de STEIM. Estas señales eran utilizadas para controlar sintetizadores y samplers

Junto con Sonami en la presentación estaba Paul DeMarinis, quien gracias a un Power Glove de Nintendo modificado (un guante diseñado como interfaz para videojuegos) realizaba síntesis sobre las voces de los espectadores.

Con el paso de los años, el proyecto de Sonami fue evolucionando, con el agregado de nuevos sensores (como los sensores de torsión, de proximidad y de presión) brindándole más gestualidad a la composición con el guante. Siempre convirtiendo vía Sensorlab la información eléctrica en eventos MIDI e interpretándolos con el programa MAX MSP.⁶

Sobre su trabajo Sonami afirma que “La intención de construir un guante era permitir movimientos sin referencias espaciales y, sobre todo, controles múltiples y simultáneos. Los sonidos son ahora “parte del cuerpo”, los controles intuitivos y la performance fluida. Se ha convertido en un instrumento muy bueno.”

En 2012, Imogen Heap y su equipo desarrollan *The Musical Gloves* (para recaudar el dinero necesario para llevar a cabo este proyecto recurrieron al *crowdfunding*, mediante campañas en internet apelaron a la cooperación colectiva). Este guante es una interface que

⁵ Waisvisz, Michael. *The Hands*. Disponible en <http://www.crackle.org/TheHands.htm>

⁶ MAX-MSP. Cycling '74 team. Disponible en <http://cycling74.com/products/max/>

consta no solamente de un guante controlador MIDI, ellos lo llaman un “*vocabulario gestual único*”⁷.

Este proyecto consta de un guante con sensores (al igual que el de Sonami) con el agregado de sistemas de captura de movimiento, en su caso el “instrumento” no es solo la mano, sino todo el cuerpo, permitiendo al artista usar sus propios movimientos para componer en tiempo real. La idea tras *The Musical Gloves* era que sean no solamente un controlador como los *Lady’s Gloves*, sino también un instrumento, diseñado para que el usuario se conecte de manera fluida con las herramientas que quiera, en este caso el software Ableton Live.

Existen proyectos similares en los que se han diseñado distintos tipos de interfaces con el propósito de agregarle gestualidad a la producción de música electrónica. Uno de ellos es el que, en 2004, Matthew Burtner presentó en la conferencia *New Interfaces for Musical Expression*. Sus “objetos Shamanicos” eran una máscara y unas manos de madera que convertían los movimientos del interprete en señales MIDI mediante sensores de proximidad.

Esta interfaz, basada en una máscara tradicional y los rituales de los shamanes de Alaska, así como otras que buscan convertir objetos comunes en objetos musicales, concuerdan con el concepto de *interfaces culturales* establecido en 2008 por los autores Christa Sommerer y Laurent Mignonneau para la creación de una nueva maestría en la Universidad de Arte e Industrias del Diseño de Linz, en Austria⁸.

⁷ Heap, Imogen. *Gestural musical ware by Imogen Heap*. Disponible en <http://www.imogenheap.co.uk/thegloves/>

⁸ El programa de estudio de la Maestría en Interfaces Culturales no se concentraba en el diseño de interfaces Humano-computadora en el sentido clásico de la ingeniería en computación, sino que se orientaba hacia los aspectos tanto sociales como artísticos (Mignonneau, Sommerer 2008)

Fundamentos teóricos de los instrumentos digitales

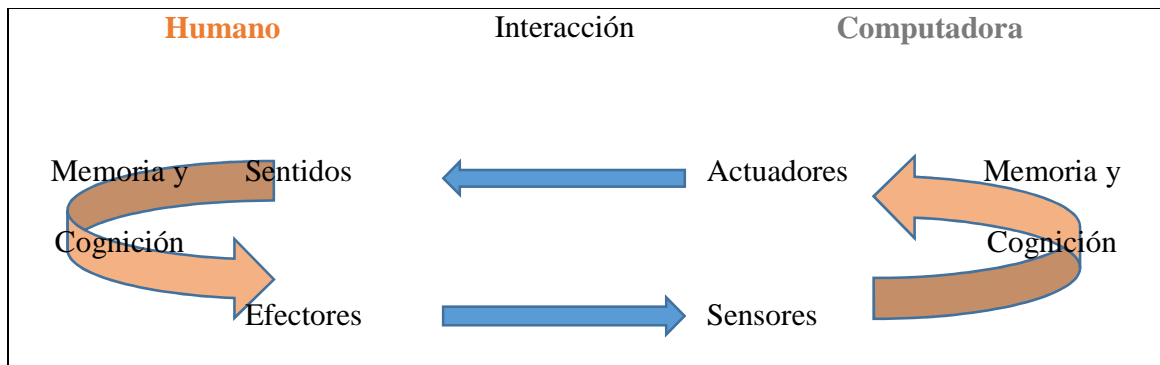
En la actualidad, la evolución de la computación aplicada a la música ha permitido una gran cantidad de nuevos métodos de síntesis y generación de sonidos. Esto junto con el desarrollo de tecnologías cada vez menos caras permite a la comunidad de compositores e instrumentistas acceder a estos sonidos en tiempo real, siendo utilizados en situaciones de conciertos. Este mismo avance tecnológico alcanza, a su vez, el campo de los dispositivos de entrada que capturan distintos movimientos humanos, generando sensores más precisos, versátiles y fáciles de manipular.

Teniendo ambos en cuenta, nos encontramos en una etapa en la que es posible combinarlos para crear nuevos instrumentos musicales basados en computación o *Instrumentos musicales digitales* (de aquí en adelante *DMI's* por sus siglas en inglés *Digital Musical Instruments*) que permitan controlar gestualmente sonidos generados por computadora, al mismo tiempo estos avances pueden extrapolarse a los instrumentos acústicos ya existentes aumentando sus capacidades generativas y abriendo todo un nuevo mundo de posibilidades musicales (Depalle, Wanderley. 2004).

A. Interacción Humano-Computadora

La HMI (en inglés *Human-Computer Interaction*) es el estudio de la manera en que la tecnología de la computación influye el trabajo humano, como explica John M. Carroll “Se encarga de entender cómo la gente hace uso de dispositivos y sistemas que incorporan o

se sumergen en el campo de la computación, y de cómo alguno es más útil o utilizable que otro.”⁹



Este gráfico (Bongers, 2000) nos permite entender como en este tipo de sistemas todas las partes están en constante comunicación e interactúan entre sí, el humano a través de los efectores (palabra que hace referencia a las células nerviosas encargadas de ejecutar respuestas a los distintos estímulos) controla los sensores y la computadora mediante sus actuadores (como el vibrador y el LED) altera los sentidos del humano. De ésta manera

Más sobre ésta temática se verá a continuación al momento de hablar de sistemas y la importancia del feedback y luego, al momento de hablar del mapeo de los gestos.

Diseño y Construcción de los guantes

La gestualidad de las manos tiene una gran importancia tanto a nivel artístico y musical como antropológico. La autora Justine Cassell describe sus características en la comunicación entre humanos y analiza cómo se han diseñado interfaces basadas en gestos naturales para mejorar la comunicación humano-computadora en su paper *A framework for*

⁹ Carroll J. (2003) "HCI Models, Theories and Frameworks: Toward a Multidisciplinary Science". Ed. Por Morgan Kaufmann, San Francisco, CA.

gesture generation and interpretation (Cassel, 2006) y como ya se ha visto en el capítulo de *Antecedentes y Estado del arte*, ya existen interfaces que apunten a utilizar esta gestualidad, estos guantes ofician de controladores para generación de sonido en tiempo real o *DMI's*, se han nombrado y explicado brevemente los de Michael Waisvisz, Laetitia Sonami, Imogen Heap y Paul DeMarinis. Cada uno tenía sus diferencias en cuanto a construcción, tipo de tecnología (sensores) utilizada, diseño estético y como generaban sonido (ninguno lo genera por sí sólo sino que utilizan diferentes medios para esto; Waisvisz enviaba señales MIDI a distintos samplers y sintetizadores, mientras que Sonami, Heap y DeMarinis recurrían a los softwares Max MSP o Ableton Live).

La principal desventaja de estos modelos es básicamente económica: los distintos sensores encargados de captar los gestos son todos fabricados industrialmente, siendo los sensores de torsión particularmente costosos, actualmente cada uno de ellos cuesta alrededor de USD 10 (diez dólares), mientras que tanto Max MSP como Ableton Live requieren una licencia paga. Este proyecto busca construir unos guantes propios con funciones similares a los vistos pero con métodos y materiales más accesibles: los sensores de torsión son de fabricación casera, y los softwares utilizados son de código abierto y libres de licencias.

A. Captura de gestos y su interpretación en Arduino

“Los sensores son los órganos de la máquina. A través estas entradas sensitivas ella se comunica con su ambiente y puede ser controlada. Los sensores convierten la energía física (mundo exterior) en electricidad (el mundo de la máquina).” - Bert Bongers, 'Physical interfaces in the electronic arts'.

El guante fabricado captura aisladamente los gestos de la mano mediante sensores de torsión en los dedos y la muñeca. Se utilizaron sensores de fabricación casera dado el alto

coste de los industriales. Para su funcionamiento están hechos a base de un material que se presenta para su comercialización en rollos de 30cm de ancho y cuesta USD 4 el metro, teniendo en cuenta que para fabricar un sensor se utiliza una tira de aproximadamente 5mm. X 35mm., queda clara la gran diferencia en costos, resultando en un costo aproximado de USD 0,20 si sumamos los demás materiales.

El material utilizado es Copolimero de polipropileno de alta conductividad volumétrica, también llamado Velostat¹⁰. Sus propiedades dieléctricas le permiten variar su conductividad dependiendo de la presión que se le ejerza, por eso es tan útil al momento de hacer los sensores, basta con poner una lámina de Velostat entre dos cintas de cobre (Figuras 1.) para tener un sensor de torsión ya que la presión en el punto de inflexión va a alterar la conductividad del material, dando como resultado una resistencia variable (Figura 2.). Estos sensores ofrecen un comportamiento unipolar ya que siempre disminuyen su resistencia nominal a medida que aumenta la deflexión y dentro de todos los tipos de sensores

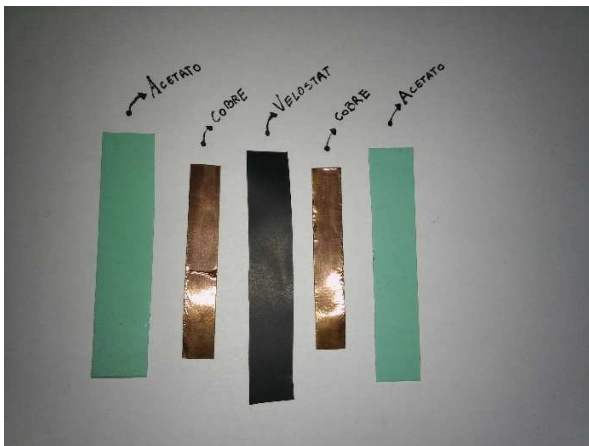


Figura 1. Disposición de los materiales para la fabricación del sensor de torsión



Figura 2. Sensor terminado

de torsión que existen son los más económicos (como ya se demostró) y presentan una gran resistencia a la humedad y cambios de temperatura (Dunne, Smith, Caulfield. 2007).

¹⁰ Producto fabricado por la empresa 3M

Luego de fabricados estos sensores, se los sometió a distintas pruebas comparativas junto con los de fabricación industrial para comprobar la viabilidad de su uso dentro de este proyecto. Para estas pruebas se tuvieron en cuenta los distintos parámetros establecidos por Andy Hunt y Ross Kirk en su trabajo “*Mapping strategies for musical performance*” (Hunt, Kirk. 2000) y por Marcelo Wanderley en “*Gestural control of Music*” (Wanderley. 2007) que son

- Rango de torsión
- Direccionalidad de la medida
- Polaridad de la medida
- Rango de resistencia

Al arrojar resultados dentro de los rangos aceptables y presentar una buena repetibilidad y certeza se decidió finalmente utilizar estos sensores de fabricación casera para el desarrollo del guante de este proyecto. Para eso se los adhirió a los dedos de los guantes más uno para registrar el movimiento de la muñeca. También se agregaron 5 pulsadores en el dorso de la mano para controlar otros parámetros en Pure Data. Por último en el diseño del guante, para facilitar el cableado entre éste y la placa Arduino se montó una PCB (Placa de circuito impreso) con conectores hembra, de ésta manera puede conectarse y desconectarse fácilmente (Figura 3.)

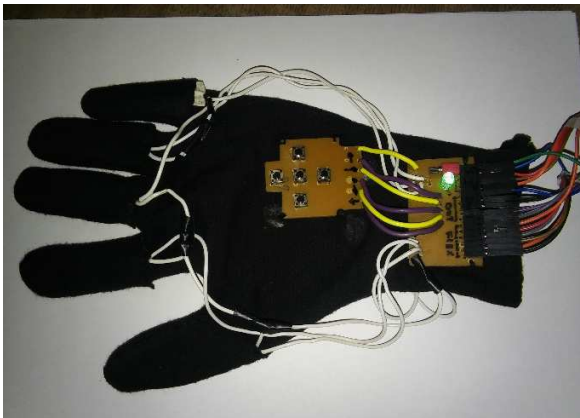


Figura 3. Guante terminado

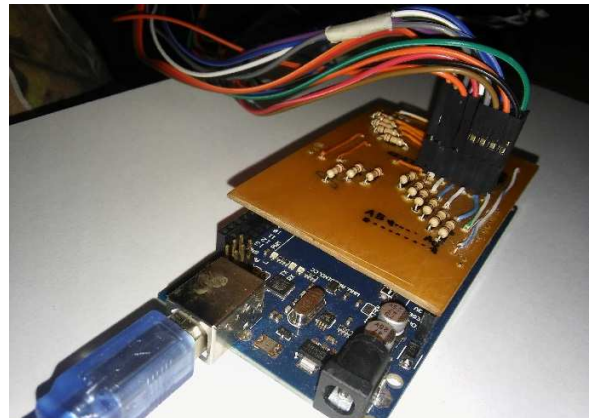


Figura 4. Placa Shield para el arduino

Como puede apreciarse en la Figura 3, el módulo en el que se montaron los conectores además cuenta con un LED RGB y un pequeño vibrador (reciclado de un celular). La idea de agregar ambos componentes surge de la necesidad de tener un feedback del sistema. La interacción entre el humano y el sistema sucede en dos etapas: Control y feedback. El control toma lugar a través de una interface que convierte las acciones del mundo real en señales eléctricas para ser interpretadas por el sistema que, luego, ofrece un feedback para ayudar al humano a articular el control (Bongers, 2000).

Dentro de los muchos tipos de feedback que pueden existir, éste sistema ofrece los tres más importantes para Bongers, el Visual (no sólo a través de lo que se ve en la pantalla de la computadora sino a través del LED), el Táctil (cuando el vibrador se mueve) y el Sonoro, a través de toda la composición musical que se va realizando mediante el uso del guante como controlador. Esto colabora a que el control del guante sea más intuitivo, fluido y rápido, al obtener instantáneamente una respuesta del sistema sin tener que estar constantemente observando la pantalla, con atender los códigos de colores y vibraciones y sumarlo a lo que está sonando el performer puede hacerse una idea de los parámetros que está controlando.

Una vez terminado el guante, se lo conecta a la placa Arduino a través del módulo antes explicado (se diseñó una placa *Shield* que se pueda montar directamente sobre el Arduino para así facilitar también esta etapa. Figura 4.)

En cuanto a la programación del Arduino, se utilizó como base un modelo de *patch* diseñado por Jiffer Harriman que luego fue modificado para adaptarse a las necesidades del proyecto. Este *patch* funciona enviando y recibiendo paquetes de datos (bytes) de Pure Data mediante las funciones `Serial.println()`; y `Serial.read()`. Cada uno de estos paquetes consta

de un array en el cual se “escanean” los pines previamente asignados y se establece el valor actual de ese pin (independientemente de su función o si es análogo o digital) en un lugar específico de la cadena, luego se envía el paquete anteponiendo una letra en mayúscula para indicar el comienzo y la misma en minúscula luego de todos los datos para indicar el final del array (“A [todos los datos] a” en el caso de las entradas analógicas y “D [todos los datos] d” en el caso de las digitales). Luego Pure Data se encarga de decodificar estas cadenas de datos y envía en un formato similar los datos que sean necesarios para obtener así una comunicación bidireccional.

B. Lectura en Pure data – Generación de sonidos

El *patch* encargado de interpretar las cadenas de datos enviadas al puerto serie por el software Arduino también se basó en el trabajo de Jiffer Harriman, con agregados de otro trabajo hecho por Alexandros Drymonitis; nuevamente modificados para su correcto funcionamiento en este proyecto.

La base de este *patch* se da en dos etapas, la primera es un objeto que el autor denominó [\[parse-start-stop\]](#), la palabra *parse* significa en inglés “Analizar gramaticalmente”, esto toma mucho sentido a la hora de ver el funcionamiento del objeto. En principio el objeto admite dos argumentos distintos, correspondientes al código ASCII (Código Estándar Estadounidense para el Intercambio de Información, un código de caracteres utilizado frecuentemente) de las dos letras que iniciarán y finalizarán la cadena de datos; p. ej: en decimal, el 68 corresponde a la letra ‘D’ mientras que 100 a la letra ‘d’, así que el bloque encargado de decodificar las entradas analógicas tendrá este formato: [\[parse-start-stop 68 100\]](#), los otros bloques serán simplemente ignorados. En definitiva, éste objeto se encarga de “elegir” las cadenas que correspondan y luego gracias a su funcionamiento interno ordenar

la información para que pase en bloques a la segunda etapa, utilizando su “análisis gramatical” para armar estos bloques. Una vez armados y ordenados los bloques pasan a la segunda etapa, donde los datos se decodifican para dejar de ser una gran cadena de bits y pasar a volver a ser números enteros separados entre sí y utilizables dentro del programa, correspondientes a los datos previamente obtenidos por los sensores, luego se “desempaca” y se envía a distintas salidas para obtener finalmente cada valor de cada sensor por separado.

A fin de obtener un feedback que dependa de lo que suceda en Pure Data (que será la parte más importante del proyecto, la parte musical) se utilizó un *patch* que permita una comunicación bidireccional, es decir que ambos programas estén intercomunicados para lograr esa relación más articulada de la cual habla Bongers. El envío de datos hacia Arduino es más sencillo y no requiere la preparación de bloques, simplemente se envía una sola cadena al puerto serie que Arduino lee cada cierto periodo de tiempo, luego estos datos obtenidos los imprime en los puertos asignados previamente mediante un array muy similar al que usa para lo demás. En el caso de este proyecto esta información se envía a los pines correspondientes al LED RGB y el vibrador.

C. Interconexiones, generación de sonidos y mapeo de gestos

Ya se establecieron las tres partes básicas de esta cadena:

1. Los sensores convierten los gestos de la mano en señales eléctricas (diferencias de voltaje)
2. La placa microcontroladora Arduino cuantifica, codifica, empaqueta y envía esta información al puerto Serie
3. El software Pure Data lee esa información, la decodifica, la organiza y queda a disposición del programador para darle la función musical que se quiera, es decir,

para controlar los parámetros de diferentes tipos de generadores o procesadores de sonido

A su vez, las tres partes están constantemente comunicándose entre sí, gracias a esta intercomunicación el usuario tiene una noción de lo que está pasando dentro del sistema a cada momento, por eso la importancia del feedback.

De esta manera la generación de sonidos dependerá del uso que se le quiera dar a la información obtenida por la cadena, la gran versatilidad que ofrece este proyecto radica en el software que utiliza para la parte musical. Existen varios ejemplos de posibles usos, partiendo de los más básicos a los más complejos.

- **Generación de sonido básica:** Se asignan osciladores seteados en distintos rangos a los sensores de torsión, al mover los dedos se irán generando los sonidos dependiendo de las posiciones de los dedos. Con los pulsadores se puede controlar la amplitud y asignar distintas envolventes.
- **Síntesis Aditiva:** En vez de utilizar los osciladores como sonidos individuales e independientes se pueden generar formas de onda complejas partiendo de otras más simples. P. Ej: Partiendo del ejemplo anterior con sus osciladores como fundamentales, se puede hacer que al apretar un botón particular se disparen una cierta cantidad de armónicos de esa frecuencia o emular el espectro de un órgano de tubos mediante un proceso similar.
- **Modulaciones (FM):** En lugar de hacer que cada sensor controle un oscilador que suene directamente (ya sea sólo o acompañado de sus armónicos), se pueden utilizar algunos de estos osciladores como

parámetro para generar distintos tipos de modulación: Por frecuencia, en Amplitud, Modulación en anillo, etc.

- **Control de multiefectos:** Como último ejemplo, se puede utilizar la información recibida para controlar diversos parámetros en un *patch* multiefectos. P. Ej: Los valores de los sensores se pueden asignar al pre-delay, el tamaño de sala, la difusión y el tiempo de decaimiento de una Reverb, mientras que con los pulsadores se controla la mezcla wet-dry y a la vez se alterna con otros efectos, dependiendo que pulsador se oprima se puede cambiar el control a un delay o una distorsión o un flanger, las posibilidades son enormes y dependen de la función que le quiera dar el programador.

Estos son algunos ejemplos concretos de distintos usos que se puede dar al sistema, los usos dependerán de cómo se asignen los valores registrados por los sensores. En este punto nos encontramos con una última problemática que es el mapeo de los gestos.

En el gráfico mostrado anteriormente cuando se habló de HCI o *Interacción Humano-Computadora* se explicó como todas las partes están en constante comunicación e interactúan entre sí, los efectores como equivalentes de los actuadores y los sentidos como equivalentes de los sensores. Entonces, el mapeo hace referencia a la organización de la información adquirida de la constante interrelación en el sistema en cada etapa. Es decir que implica asignar una respuesta a cada acción para que el sistema siga funcionando correctamente o como se espera que funcione. De manera más concreta: Como va a afectar un valor particular obtenido al resto del sistema.

Como ejemplos particulares: Que sonido va a generar cada oscilador asignado a un sensor, o que parámetro específico va a controlar en un efecto particular, o que va a suceder cuando presione cierto botón.

El mapeo es una de las partes más importantes a tener en cuenta al momento de querer llevar a cabo una performance y utilizar el guante como DMI, pero al mismo tiempo, éste depende exclusivamente del uso que se vaya a dar y, teniendo en cuenta la gran cantidad de posibilidades, para el correcto funcionamiento del guante es necesario primero pensar qué se quiere hacer y luego mapear los gestos para llegar a ese resultado de la manera más articulada y fluida posible.

Conclusiones

Se ha demostrado que con un costo mucho menor se puede fabricar sensores de torsión caseros viables y fiables, y con ellos fabricar una interfaz controladora en forma de guante capaz de llegar a resultados similares a los obtenidos por los autores mencionados, utilizando al mismo tiempo hardware y software de código abierto y libre de licencias.

Al mismo tiempo el uso del software Pure Data para la generación de sonido ofreció una diversa cantidad de posibilidades sonoras y musicales que dependen de las decisiones artísticas del usuario y no requieren *a priori* grandes conocimientos en programación, al presentar una interfaz gráfica intuitiva y fácil de manejar.

Bibliografía

Bongers, B. (2000). “Physical interfaces in the electronic arts”. *Trends in gestural control of music* (pp. 41-70). IRCAM – Centre Pompidou. Paris

Borchers, J., Herkenrath, G., and Karrer, T. (2008). “TWEND: Twisting and bending as new interaction gesture in mobile devices”. En Conferencia *CHI'08. Extended Abstracts on Human Factors in Computing Systems* (pp. 3819-3824). Association of Computer Machinery (ACM).

Burtner, M. (2004). “A Theory of Modulated Objects for New Shamanic Controller Design”. NIME'04. Procedente de la Conferencia de 2004 New Interfaces for Musical Expression (pp. 193-196). Hamamatsu. Japón.

Carrol, John M. (2003). “HCI Models, Theories and Frameworks: Toward a multidisciplinary science”. Editado por Morgan Kaufmann, San Francisco, CA.

Cassell, J. (1998). “A Framework for Gesture Generation and Interpretation”. *Computer vision in human-machine interaction*. Cambridge University Press (pp. 191-215). Cambridge University.

Depalle, P., Dubnov, S., Rovani, J. & Wanderley, M. (1997). “Instrumental Gestural Mapping Strategies as Expressivity Determinants in Computer Music Performance”. En *Kansei, The Technology of Emotion. Proceedings of the AIMI International Workshop* (pp. 68-73). Genoa: Associazione di Informatica Musicale Italiana.

Depalle, P. & Wanderley, M. (2004). “Gestural Control of Sound Synthesis”. IEEE, Vol. 92, N° 4 (pp. 632-644).

Dobrian, C. & Koppelman, D. (2006). “The **E** in NIME: Musical expression with new computer interfaces”. NIME’06. Procedente de la Conferencia de 2006 New Interfaces for Musical Expression (pp. 277-282). IRCAM – Centre Pompidou. Paris

Dunne, L., Smith B. and Caulfield, B. (2007). “A comparative evaluation of bend sensors for wearable applications”. En la 11° *IEEE International Symposium on Wearable Computers* (pp. 121-122). IEEE.

Freed, A. (2008). “Application of new fiber and malleable materials for agile development of augmented instruments and controllers”. NIME’08. Procedente de la Conferencia de 2008 New Interfaces for Musical Expression (pp. 107-112). Universidad de Genova. Italia.

Kirk, R. & Hunt, A. (2000). “Mapping Strategies for Musical Performance”. *Trends in Gestural Control of Music*, N° 21 (pp. 231-258). IRCAM – Centre Pompidou. Paris.

Kogan, R. (2008). “Brief History of Electronic and Computer Musical Instruments”. Texas A&M University. Texas. CA.

Mignonneau, L. Sommerer, C. (2008). “Interface Cultures - Artistic Aspects of Interface Design”. Editado por Dorothée King. University of Arts and Industrial Design. Linz. Austria

Wanderley, M. (2001). “Gestural control of music”. En *International Workshop Human Supervision and Control in Engineering and Music* (pp. 632-644).

Volker, K., and Waivisz, M. (1990). “The Hand in the web: An interview with Michel Waisvisz”. *Computer Music Journal*, Vol. 14, N° 2. (pp. 28-33).