



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO
Programa de Maestría y Doctorado en Música

Facultad de Música
Instituto de Ciencias Aplicadas y Tecnología
Instituto de Investigaciones Antropológicas

UN MODELO DE PROGRAMACIÓN PARA OBRAS ELECTRÓNICAS EN VIVO
A PARTIR DEL ANÁLISIS DE DOS COMPOSICIONES REALIZADAS CON MAX/MSP

TESINA
QUE, PARA OPTAR POR EL GRADO DE
MAESTRO EN MÚSICA (TECNOLOGÍA MUSICAL)

PRESENTA
SALVADOR GOVEA VILLASEÑOR

TUTORA
MTRA. VIOLETA CANTÚ JARAMILLO
PROGRAMA DE MAESTRÍA Y DOCTORADO EN MÚSICA

CIUDAD DE MÉXICO, OCTUBRE 2021



Universidad Nacional
Autónoma de México



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

Agradecimientos

Quiero reconocer a quienes me han apoyado y acompañado en este viaje. En primer lugar, Violeta Cantú siempre me ha impulsado a desarrollarme en mi carrera y a fijarme nuevos objetivos a lograr. Ivonne Serment, mi psicóloga de cabecera, no paró nunca de arrojarme indirectas –y muy directas– invitaciones para terminar mi maestría. A mi querido sobrino Andrés Castro, le agradezco su bien cuidado y profesional trabajo con las ilustraciones del capítulo tres. Gracias también a Alexandra Fuentes por su entusiasmo para interpretar mi obra para tarola y por las sesiones de trabajo en que puso a prueba el *patch* creado para la pieza. A Jeffrey Zeigler le agradezco su talento y musicalidad para hacer realidad mi obra para violonchelo. Gracias a Esteban Chapela por su experta programación, por la entrevista concedida y por su siempre amable disposición para contestar mis dudas sobre su programa. Gracias a los profesores miembros del jurado por brindarme sus correcciones y observaciones para mejorar este texto. Y, por supuesto, agradezco la satisfacción y orgullo por haberme formado en la Facultad de Música de la Universidad Nacional Autónoma de México.

Contenido

Agradecimientos	2
Lista de tablas y figuras	5
Resumen	7
Introducción	8
Estado del arte	10
Breve descripción de las obras a analizar	13
<i>Subliminal</i> , para violonchelo eléctrico y electrónica	13
<i>Ocaso de Tenochtitlan</i> , para tarola y electrónica	13
Notas	14
1. Definiciones básicas	15
1.1 ¿Qué es la música electroacústica?	15
1.2 Electrónica en vivo: oportunidades y limitaciones	20
1.3 Software comercial versus <i>custom</i> software	22
1.4 ¿Qué es Max-MSP?	23
2. Análisis comparativo	27
2.1 Entrada y salida de audio	28
2.2 Generación, transformación y reproducción de sonidos	30
Generadores de sonido	31
Procesadores de sonido	31
Reproductores de sonido	32
2.3 Módulos de generación, transformación y reproducción sonora utilizados en cada obra	32
<i>Subliminal</i>	32
<i>Ocaso de Tenochtitlan</i>	37
2.4 Activación y desactivación de los módulos	41
Asistente a cargo de la electrónica	41
Intérprete	42
Automatización	48
Detectores de dinámicas, alturas y motivos musicales	52
3. Modelo de programación	57

4. Software, hardware y equipo de audio utilizado	65
4.1 Software	65
Notación musical	65
Secuenciador, audio digital e instrumentos virtuales	65
Editor de programas	66
Entrenamiento auditivo	66
4.2 Hardware y equipo de audio	67
Interfaz de audio	67
Interfaz MIDI	67
Teclado electrónico MIDI	67
SPY de Midiplus	67
Pedal de sustain	68
Pedal de expresión	68
Monitor de computadora	68
Micrófonos	68
Audífonos	68
Monitores de piso	68
5. El intérprete en la música electrónica en vivo	69
5.1 Colaboración entre compositor e intérprete	69
5.2 Control de la electrónica por el intérprete	70
5.3 Realización de ensayos y conciertos	71
Equipo mínimo necesario	72
6. Conclusiones	75
Referencias	79
Discografía	83
Apéndice: Partituras	84
<i>Subliminal</i>	85
<i>Ocaso de Tenochtitlan</i>	96

Lista de tablas y figuras

Tablas

Tabla 1	Categorización de obras electroacústicas de <i>México electroacústico (1960-2007)</i>	18
Tabla 2	Módulos utilizados en “Presagios”, primer movimiento de <i>Ocaso de Tenochtitlan</i>	39
Tabla 3	Módulos utilizados en “Soles de metal fino”, segundo movimiento de <i>Ocaso de Tenochtitlan</i>	39
Tabla 4	Módulos utilizados en “Pólvora, gérmenes y acero”, tercer movimiento de <i>Ocaso de Tenochtitlan</i>	40
Tabla 5	Valores de los presets 25 a 27 usados en el módulo PasaBanda de <i>Ocaso de Tenochtitlan</i>	47

Figuras

Figura 1	Ejemplo de programa (patch) en Max/MSP	25
Figura 2	Pantalla principal del patch para <i>Ocaso de Tenochtitlan</i>	29
Figura 3	Pantalla principal de <i>Live Patch</i> , usado para <i>Subliminal</i>	30
Figura 4	Sección de glissandi electrónicos (a partir del compás 197) de <i>Subliminal</i>	37
Figura 5	Programa que recibe las pulsaciones del pedal y lleva el conteo de las mismas	43
Figura 6	Selección del módulo a activar o desactivar en <i>Ocaso</i>	45
Figura 7	Selección del módulo a activar o desactivar en <i>Subliminal</i>	46
Figura 8	Automatización de procesamientos en <i>Discourse</i> de V. J. Manzo	49
Figura 9	Compases 38 a 39 de Presagios, primer movimiento de <i>Ocaso</i>	50
Figura 10	Implementación de fade in y fade out en <i>Live Patch</i>	52
Figura 11	Cambios de dinámica en los compases 22 a 29 de Presagios, primer movimiento de <i>Ocaso</i>	53

Figura 12	Volumen del módulo controlado por el nivel de audio de entrada	54
Figura 13	Entrada y salida de audio de la computadora	57
Figura 14	Procesadores de audio insertados entre la entrada y la salida	57
Figura 15	Los tres tipos de módulos dentro de un programa para una obra electrónica en vivo	58
Figura 16	Controles de encendido y apagado y regulación de volumen en los módulos	59
Figura 17	Adición de la salida directa del audio sin procesar y de la pantalla de preset en los módulos	60
Figura 18	Sección de activación, desactivación y modificación de los módulos	61
Figura 19	Modelo de programación para una obra electrónica en vivo	63
Figura 20	Colocación de equipo recomendada para la ejecución en concierto de <i>Ocaso de Tenochtitlan</i>	73

Resumen

A través del análisis y comparación de los programas diseñados para dos obras electrónicas en vivo –una para violonchelo eléctrico y otra para tarola–, se ofrece un modelo que guíe en la construcción de programas para la interpretación de este tipo de composiciones. Las dos piezas que se presentan fueron compuestas por el autor de este trabajo pero el programa necesario para su ejecución –que contiene los diferentes procesadores, generadores y reproductores de sonido, así como el sistema con el que estos se activan, modifican y desactivan– fue realizado en una de ellas por el compositor y en la otra por un programador independiente, de tal forma que se identificarán similitudes entre los dos programas –para establecer qué puntos son indispensables en un programa diseñado para este tipo de obras– y diferencias –para comparar estilos de programación y modos de utilización del software y explorar diferentes soluciones a un mismo problema técnico–. Se presentan ejemplos de las partes más importantes de los programas con una explicación detallada que espera ser suficientemente clara aún para aquellos que desconozcan la programación en Max/MSP. Es así que el modelo aquí presentado busca que los compositores vislumbren los alcances y posibilidades técnicas de este tipo de programas para la creación de obras electrónicas en vivo, sin importar que utilicen otros lenguajes y entornos de programación u otras herramientas tecnológicas.

Introducción

Empecé a estudiar música a los nueve años y electricidad a los siete. [...] Siempre quise de alguna manera conectar mi querida electricidad con la música. (Theremin, 1990, p. 5)

Graduado como violonchelista en el Conservatorio de Petrogrado el mismo año que se graduó como ingeniero militar de radio (Montague, 2001), Leon Theremin –inventor de uno de los primeros instrumentos musicales electrónicos– ilustra al personaje híbrido –mitad músico, mitad ingeniero– que a menudo se encuentra en la historia de la música electroacústica. Con tanto artista-ingeniero que ha existido en este campo de la música, cabe preguntarse si es indispensable tener ambas profesiones para desarrollarse en este ámbito, una pregunta que décadas atrás se hizo Raúl Pavón –él mismo, un ingeniero–, y a la que contestó que:

El músico seguirá músico y el ingeniero seguirá ingeniero, y aunque en esta fase del desarrollo la hibridación ha sido una necesidad, esto es más bien circunstancial y transitorio; los sistemas que requieran gran preparación técnica para ser operados irán desapareciendo dejando por selección natural a los equipos sencillos configurados para músicos. Creo, sin embargo, que el músico deberá conocer mejor la materia prima con la que trabaja, para poder obtener mejor partido del equipo electrónico, es decir: la física y electroacústica musicales y desde luego, familiarizarse completamente con el equipo que utilizará para sus composiciones musicales. (Pavón, 1981, p. 13)

A cuarenta años de ser escritas, estas palabras vaticinaron transiciones tecnológicas como el cambio ocurrido entre crear música electrónica cortando con navaja cintas de audio y uniéndolas con adhesivo, a la sencillez de la edición de las grabaciones digitales: cortando, pegando y transformando el audio de numerosas formas en programas que se pueden descargar gratuitamente de internet o que en algunas computadoras personales de uso general vienen ya instalados de fábrica –como si fueran una herramienta más del sistema operativo.

Los programas secuenciadores y los de audio digital –y las DAW (estaciones de trabajo de audio digital) que reúnen a estos dos tipos de programas– han simplificado enormemente la tarea de producción de música electrónica que décadas atrás se realizaba en cinta magnetofónica –y que por esos mismos cambios de tecnología dejaron de denominarse como obras “para cinta”–. Pero en cuanto a la música electrónica en vivo, aunque tenemos desde hace varias décadas lenguajes de programación para crear esta música –como Csound, Max o SuperCollider–, el largo tiempo requerido para el aprendizaje de los mismos –sobre todo si se le

compara con el necesitado para manejar una DAW– no ha favorecido la penetración de este software entre los usuarios músicos.

Personalmente, a pesar de que disfruto los desafíos de crear un programa y la emoción de encontrar el algoritmo que soluciona un problema, no podía dejar de advertir las ventajas que ofrecía el entorno gráfico de Max/MSP sobre un lenguaje como Csound –al que le había invertido largas horas de trabajo para obtener resultados muy modestos–. Así que comencé a experimentar con Max –la parte dedicada al MIDI– y sólo mucho más tarde con MSP –el bloque orientado al manejo del audio digital–. Mis primeros programas en Max fueron realizados para controlar los sonidos del conjunto de sintetizadores con los que tocaba en mi grupo de rock: para cada diferente pieza tenía un programa que mandaba el número de *preset* –a veces antecedido del número de banco– y el volumen respectivo a cada uno de los instrumentos –que iban de tres a cinco– que utilizaba para tocar. Si había cambios en el transcurso de la pieza, sólo necesitaba pulsar un pedal de *sustain* para que se enviara el nuevo preset con su respectivo volumen al teclado que lo necesitara. Cuando después surgió mi interés en componer una obra electroacústica, emprendí el estudio de los objetos para procesamiento sonoro de MSP pero me llevó tiempo entender la diferencia entre una instrucción MIDI –donde sólo basta un comando de *note-on* y uno de *note-off* para que suene una nota de tres segundos (o de cualquier duración)– y la forma de funcionamiento de un objeto de MSP para control de audio digital –donde el programa requiere mandar cada segundo 44100 números a la tarjeta de audio para que se mantenga sonando la nota deseada–. Por otro lado, siempre fue un misterio de qué forma conseguiría ir activando los procesos o efectos a lo largo de una pieza electrónica en vivo y cómo se podrían programar dentro de Max: mi confusión era enorme porque contaba con una descripción de los cientos de objetos del programa –creo que no he visto otra aplicación con tan amplia cantidad de tutoriales, ejemplos y sección de ayuda y referencia como Max– pero no atinaba a vislumbrar cómo hacer las conexiones correspondientes para lograr el avance y desarrollo de la pieza. Encontré varios libros que hablaban de la parte teórica de la música electroacústica, pero ninguno que me dijera cómo conectar los diferentes segmentos para crear una obra así.

Un libro fundamental que encaminó mis pasos hacia la creación de mi obra fue *Max/MSP/Jitter for Music* de V. J. Manzo (2011) que en su capítulo veinte, “Compositions and Perception Tools”, muestra una obra de él mismo –*Discourse* para clarinete en Mi bemol y

computadora–, donde discute y analiza el programa creado con Max para su pieza y presenta las formas en que el instrumentista puede interactuar con este. A partir de ese ejemplo de Manzo, logré comenzar la programación y composición –de manera simultánea– de mi pieza electroacústica.

Es mi deseo que las descripciones y explicaciones vertidas en este trabajo puedan arrojar luz sobre el proceso conjunto de programación/composición de este tipo de obras, sobre todo a los que –por las razones que sean– no han podido sumergirse en este campo. Las dos piezas analizadas no son ejercicios didácticos, sino ejemplos reales de composición con los problemas y situaciones que se pueden dar en el curso de este proceso.

Así como yo obtuve un gran impulso del análisis de la obra de Manzo, espero que este trabajo aporte ideas, guías y modelos que puedan facilitar y estimular la creación de más obras electrónicas en vivo.

Estado del arte

Si bien existe una amplia bibliografía sobre la música electroacústica –o electrónica–, pocos son los autores que se han abocado a profundizar sobre los problemas prácticos de la implementación técnica y ejecución de las obras electrónicas en vivo.

Rodrigo Sigal (2009), en su libro *Compositional Strategies in Electroacoustic Music*, presenta un análisis de cinco de sus obras, dos de ellas acusmáticas y tres para instrumento acústico con sonidos electrónicos en soporte fijo. Aunque una de sus obras utiliza Max/MSP para permitir que el intérprete –accionando un pedal– dispare archivos de audio en diferentes puntos de la partitura, el resultado es el mismo: una parte acusmática que acompaña la ejecución del instrumentista, pero, en este caso, con una pista electrónica dividida en varios archivos. Fuera de ese pequeño punto de contacto con la música electrónica en vivo, no hay mención de este tópico en su libro.

Por su parte, Curtis Roads (2015), en *Composing Electronic Music: a new Aesthetic*, ofrece valiosas ideas sobre la composición de esta música, pero es muy claro al señalar los puntos que no se cubren en el mismo: “Me gustaría hacer referencia a la omisión de tres temas: la interpretación en vivo y la improvisación, las representaciones gráficas y el timbre” (p. XV) y después subraya que “No tiene sentido para mí pretender que tengo algo particularmente

interesante que decir sobre la interpretación en vivo de la música electrónica. Esto lo dejo para otros” (p. XVII).

Por tanto, los libros y artículos consultados deben ser más específicos respecto al tema que nos interesa, como los referidos a continuación.

En su artículo *Live/Electro-Acoustic Music– a Perspective from History and California*, Barry Schrader (1991) expone una perspectiva histórica sobre la música electroacústica en vivo, lo que sirve como base para presentar sus desarrollos más recientes –para esa época, por supuesto–. Habla sobre las composiciones electroacústicas de estudio –otra forma de denominar a las obras acusmáticas– y sus problemas de recepción para el oyente, y brinda un panorama de las formas de electrónica en vivo que pueden existir. Explica brevemente los enfoques compositivos de diversas obras –a nivel de ideas y muestra de posibilidades–, pero no habla sobre los asuntos relativos a la configuración del equipo o a la programación del software.

Simon Emmerson (2007) plantea en *Living Electronic Music* una serie de reflexiones sobre la estética, teoría e historia de esta música. Acerca de la implementación de obras electrónicas en vivo, son importantes sus capítulos sobre el micrófono y el altavoz, de los que habla desde un nivel histórico, un poco del acústico, y bastante de aspectos técnicos –como características y especificaciones–; información provechosa para quien inicia en el campo pero que cualquier ingeniero de audio experimentado maneja sin problema alguno.

El libro *Live-electronic Music: Composition, Performance, Study*, editado por Bertolani, Burle, Sallis, y Zattra (2018), brinda muchos temas interesantes sobre teoría y conceptualización de la música electrónica en vivo, pero, nuevamente, muy poco que abone para el montaje de un sistema de hardware o software para la interpretación de estas obras. En el capítulo escrito por Agostino de Scipio, el autor habla sobre su obra *Two pieces of listening and surveillance*, para flauta y electrónica; sin embargo, su descripción se enfoca en la “puesta en escena” de la misma, la que comienza con:

Una flauta se encuentra sobre una mesita débilmente iluminada, frente al público, más o menos en el centro del escenario. Un cable delgado se abre paso a través del orificio de la embocadura y termina en un micrófono de pequeña cápsula en el interior de la cabeza de la flauta. No hay nadie cerca. Después de un largo silencio, se oye una débil y prolongada sonoridad procedente de dos altavoces un poco alejados en el fondo, un "siseo" que rememora el paso del aire a través de un pequeño tubo. (p. 20)

Su exposición sigue en el mismo estilo, sin que el lector se entere nunca de los programas, procesadores de audio y equipos utilizados: los aspectos técnicos son dejados fuera de la discusión. Aunque incluye en su artículo –y en la partitura de su obra– un *signal flow chart* donde se observan algunos de los procesos aplicados a la señal del instrumento, este esquema es particularmente confuso, no sólo para el intérprete sino también para el compositor/programador que quisiera tomarlo como modelo para sus obras.

El libro editado por Nick Collins (2007), *The Cambridge Companion to Electronic Music*, ofrece los capítulos “Live electronic music”, escrito por él mismo, y “Interactivity and live computer music”, por Sergi Jordá, con contenido primordialmente de carácter histórico.

En resumen, la mayoría de estos libros se concentran en plantear cuestiones estéticas, históricas, filosóficas y conceptuales sobre la música electrónica en vivo, y aunque los pocos estudios de caso descritos pueden aportar ideas sobre las posibilidades existentes en esta música, también dejan un vacío enorme sobre cómo llevarlas a cabo en la práctica. A todos estos textos podría aplicárseles el mismo comentario que hace Curtis Roads sobre su libro anteriormente citado: “un recorrido por hechos, historia, comentarios, opiniones y enlaces a ideas interesantes para considerar y explorar” (2015, p. X).

Machine Musicianship, de Robert Rowe (2001), trata de los sistemas musicales interactivos –parte esencial de la música electrónica en vivo –, y gran parte de su contenido se centra en los métodos de reconocimiento que puede usar la computadora para extraer de una secuencia musical MIDI una serie de alturas que formen acordes –con sus fundamentales e intervalos constitutivos–, tónicas y estructuras de escalas y modos, pulso y métrica, frases y patrones musicales, etc.; también contiene técnicas de generación de material musical por parte del software. Se incluyen numerosos ejemplos en los lenguajes C++ y Max, y muestra programas que trabajan con datos MIDI, pero no con audio digital.

Max/MSP/Jitter for Music, de V.J. Manzo (2011), aborda directamente la programación con fines musicales, desde el desarrollo de aplicaciones para entrenamiento auditivo, teoría musical, composición algorítmica, manejo de audio y video, hasta programas para música electrónica en vivo. El capítulo 20 de su primera edición, “Compositions and perception tools”, explora una composición del autor para clarinete en mi bemol y Max/MSP, donde muestra cómo se puede implementar una línea de tiempo que active procesadores en diversos momentos para modificar el audio del clarinete. El patch para esta pieza es brevemente discutido en mi trabajo,

puesto que se usó como modelo para la obra de tarola y electrónica. Sin duda, el estudio de los patches y las partituras que proporciona Manzo constituyen una ayuda enorme para la comprensión de la puesta en práctica de una obra electrónica en vivo.

Los dos volúmenes de *Electronic Music and Sound Design - Theory and Practice with Max 8*, de Cipriani y Giri (2019; 2020), son referencia obligada para todo el que quiera hacer uso del procesamiento en tiempo real en sus composiciones. Muchos de los módulos de procesamiento de mi obra para tarola fueron inspirados, tomados y modificados a partir de los ejemplos de estos libros, si bien no se presentan en ellos indicios de un esquema o modelo para obras electrónicas en vivo como el desarrollado en este trabajo.

Breve descripción de las obras a analizar

Subliminal, para violonchelo eléctrico y electrónica

Mi obra *Subliminal* fue compuesta desde mediados de 2016 hasta el primer trimestre de 2017 como parte del taller de composición para violonchelo eléctrico y electrónica impartido en la escuela privada “Núcleo Integral de Composición”. Fue estrenada por Jeffrey Zeigler el 27 de mayo de 2017 en la Sala Carlos Chávez del Centro Cultural Universitario de la UNAM y se grabó para el disco *Impulse Codes* (2018) del mismo violonchelista. El compositor Esteban Chapela –egresado del CIEM (Centro de Investigación y Estudios de la Música)– desarrolló en Max/MSP el programa *Live Patch* que todos los alumnos del taller utilizamos para crear la parte electrónica de las piezas a través de los procesadores de audio que nos proporcionó en el mismo y el control de activación y desactivación de los procesos que el intérprete operó en el concierto. Tiene una duración de 9:42 minutos. En la dirección <https://youtu.be/AMsayysl2j4> de mi canal de Youtube, puede escucharse la obra.

Ocaso de Tenochtitlan, para tarola y electrónica

Mi segunda obra a analizar en este trabajo es *Ocaso de Tenochtitlan*. Fue compuesta entre abril de 2020 y septiembre de 2021 y consta de tres movimientos: I. Presagios, II. Soles de metal fino y III. Pólvora, gérmenes y acero. En esta obra, la parte de programación en Max/MSP fue hecha por mí, al mismo tiempo que se componía la pieza, de tal forma que hubo una influencia en ambas direcciones entre los requerimientos de la pieza y las posibilidades del programa –y del programador–. Fue especialmente compuesta para este trabajo de titulación por lo que en el

momento que esto se escribe aún no ha sido estrenada. Tiene una duración aproximada de 16 minutos.

Notas

1. Todas las traducciones del inglés al español fueron realizadas por el autor de este trabajo.

2. Los objetos de Max/MSP y los subpatches dentro de los programas son escritos entre corchetes, por ejemplo: [makenote] y [p Flanger].

1. Definiciones básicas

1.1 ¿Qué es la música electroacústica?

La explicación de los conceptos *música electroacústica* y *música electrónica* es frecuentemente eludida, incluso en libros y artículos que ostentan en su título esas mismas palabras. Ejemplo de esto se observa en *The Cambridge Companion to Electronic Music*, donde se lee: “Existe cierta presión para definir y delimitar la música electrónica, lo que nosotros evitaremos” (Collins y D’Escriván, 2007, p. 256). Sin embargo, se puede determinar que ambos conceptos generalmente se entienden –por lo menos en el ámbito de la academia– como equivalentes. Por ejemplo, si se busca *Electronic Music* en el *New Grove Dictionary of Music and Musicians*, éste referirá directamente a *Electro-acoustic Music*. Precisamente, es la definición de este diccionario, formulada por Simon Emmerson y Denis Smalley (2001), la que se usará en el presente trabajo: “Música en la cual la tecnología electrónica, basada ahora principalmente en computadoras, es usada para acceder, generar, explorar y configurar materiales sonoros, y en donde los altavoces son el principal medio de transmisión” (p. 59).

De esta definición se desprende que el término *música electroacústica* abarca los conceptos de *elektronische musik*, música concreta, música para cinta (*tape music*), y música por computadora.

Emmerson y Smalley también distinguen dos tipos de obras electroacústicas. Primero se encuentran aquellas que existen únicamente en forma grabada, ya sea en cinta, disco compacto o algún tipo de memoria digital. Un concierto de este tipo de obras consiste en sentarse frente a dos o más altavoces y esperar a que alguien presione el botón de reproducción; no hay intérprete frente al público, lo que se escucha es el resultado sonoro final, fijo e inamovible que brinda el compositor. En la actualidad, a este género se le ha dado el nombre de música electroacústica en soporte fijo o de *acusmática*.

El término *acusmática* deriva de los Akousmatikoi, un grupo de discípulos de Pitágoras que escuchaban sus conferencias detrás de una cortina. El teórico francés Jerome Peignot usó después el término de Pitágoras para describir la separación, que existe en la música concreta, del sonido y su referencia visual. (Barret, 2007, p. 234)

El compositor Herbert Eimert da cuenta del concierto ocurrido el 19 de octubre de 1954 en los estudios de la Radio de Colonia donde se escucharon obras acusmáticas de Stockhausen, Eimert, Goeyvaerts, Poussuer y Gredinger, con siete piezas donde el “intérprete” era el

reproductor de cinta magnética: “En este concierto no sólo la música era algo nuevo, sino también el hecho de haber sido concebida para su transmisión por altoparlante” (Eimert, 1973, p. 25).

El segundo tipo de obras electroacústicas son aquellas en las que:

La tecnología es empleada para generar, transformar o disparar sonidos (o una combinación de éstas) en el momento de la interpretación; esto puede incluir la generación de sonidos con voces e instrumentos tradicionales, instrumentos electroacústicos, u otros dispositivos y controles enlazados a sistemas computarizados. (Emmerson y Smalley, 2001, pp. 59-60)

Esto describe lo que se denomina música electrónica en vivo. A continuación, se citan tres ejemplos. En un concierto de theremin, el sonido electrónico es generado en ese mismo instante por las manos del intérprete. Una guitarra acústica puede transformar su sonido –al tiempo que es tocada– a través de un pedal o dispositivo de *delay*, creando así una mezcla sonora electroacústica que sólo puede llegar al público por los altavoces. Se puede utilizar una batería electrónica o *pads* de percusión para que un músico, al golpear éstos, dispare eventos sonoros pregrabados.

Esta clasificación en obras acusmáticas y obras electrónicas en vivo puede ser una muy buena guía para entender el gran espectro de diferentes composiciones electroacústicas que se han creado –aunque, como en toda clasificación, puedan existir casos de excepción o que se resistan a entrar en una sola categoría.

Es posible, por supuesto, combinar una parte sonora acusmática con instrumentos acústicos o voces en vivo, y a esto se le denominará *obra electroacústica mixta*. Existen innumerables obras de este tipo, desde *Musica su due dimensioni I* (1952) de Bruno Maderna, para flauta, platillo y cinta, hasta *Temazcal* (1984), para maracas y cinta, del mexicano Javier Álvarez, por citar dos ejemplos muy separados en el tiempo y en la geografía.

A continuación, se aplicarán estas categorías de música electroacústica –acusmática, en vivo y mixta– a diversas obras de compositores mexicanos. Este ejercicio persigue dos objetivos: en primer lugar, busca ejemplificar las definiciones establecidas y demostrar su utilidad en la clasificación de obras electroacústicas de muy diversa índole y, segundo, proporcionará el contexto para la presentación y mejor entendimiento de las obras electroacústicas objeto del presente trabajo.

En la Tabla 1 se listan cronológicamente las veintiocho obras presentadas en los tres discos de *México electroacústico (1960-2007)*, editados bajo la dirección artística y producción de Manuel Rocha (2008), en la que se especifica la categoría en la cual las he incluido –tomando en cuenta la información del folleto que acompaña a los discos.

Es fácil identificar las piezas acusmáticas, puesto que la mayoría de ellas aparece con la denominación *para cinta*, etiqueta que no corresponde en muchos casos al soporte físico de la obra, la que puede estar en disco compacto u otro tipo de memoria digital: disco duro, memoria USB, reproductor de archivos de audio, e incluso en la memoria de un teclado electrónico.

En las piezas 14 y 25 de la tabla se tienen obras que son descritas de forma diferente. La pieza de Antonio Russek aparece como “para 8 canales (versión estéreo)”, y la pieza de Arturo Fuentes se señala como “para 4 canales (versión estéreo corta)”, por lo que es necesario utilizar la información del folleto de los discos para entender que son piezas acusmáticas.

Enseguida se describen las tres piezas que se clasificaron como electrónica en vivo. En estos casos, la sola mención de requerir intérpretes excluye a estas obras del ámbito de la acusmática, quedando las opciones de obra mixta o de electrónica en vivo.

Erótica II, para violín, globo y medios electrónicos, de Vicente Rojo Cama.

La clave para su clasificación la tenemos en el texto que dice: “Los sonidos producidos por ambos instrumentos fueron modificados electrónicamente en tiempo real durante su grabación en cuatro canales y posteriormente mezclados en dos canales” (Rocha, 2008, p. 49). Se tiene aquí una transformación sonora por medio de la electrónica que se da en el momento de la interpretación y no existe una pista pregrabada que se utilice en ese mismo lapso.

Nahual II, para arpa chamula y medios electrónicos, de Roberto Morales.

Este segundo caso es similar al primero: “El tempo, alturas, timbres, dinámicas y parámetros de control en el sintetizador y procesador *SPX 90*, se manejan de manera precisa y en tiempo real” (Rocha, 2008, p. 35).

En ambas descripciones aparece el término “en tiempo real”, el cual es indicador de que las transformaciones sonoras de los instrumentos ocurren al mismo tiempo que están siendo interpretados.

Tabla 1*Categorización de obras electroacústicas de México electroacústico (1960-2007)*

No.	Año	Autor	Obra	Categoría
1	1960	Carlos Jiménez Mabarak	<i>Paraíso de los ahogados</i>	Acusmática
2	1972	Mario Lavista	<i>Contrapunto</i>	Acusmática
3	1972	Manuel Enríquez	<i>Viols</i>	Mixta
4	1973	Héctor Quintanar	<i>Diálogos</i>	Mixta
5	1974	Manuel de Elías	<i>Non novo sed nova</i>	Acusmática
6	1980	Manuel Enríquez	<i>Misa Prehistórica</i>	Acusmática
7	1980	Julio Estrada	<i>Eua'on</i>	Acusmática
8	1986	Vicente Rojo Cama	<i>Erótica II</i>	Electrónica en vivo
9	1987	Francisco Nuñez	<i>Juegos sensoriales</i>	Acusmática
10	1987	Javier Álvarez	<i>Papalotl</i>	Mixta
11	1990	Roberto Morales	<i>Nahual II</i>	Electrónica en vivo
12	1992	Gabriela Ortiz	<i>Five micro études</i>	Acusmática
13	1993	Antonio Fernández Ros	<i>Con jícamo</i>	Acusmática
14	1998	Antonio Russek	<i>Babel de nuevo</i>	Acusmática
15	1999	Pablo García Valenzuela	<i>Gutura</i>	Acusmática
16	1999	Rogelio Sosa	<i>Maquinaria del ansia</i>	Acusmática
17	2000	Víctor H. Romero	<i>Voces (Canto I)</i>	Acusmática
18	2001	Sergio Luque	<i>Frío</i>	Acusmática
19	2001	Manuel Rocha	<i>Semi No Koe</i>	Mixta
20	2002	Guillermo Acevedo	<i>Punto de partida</i>	Acusmática
21	2002	Rodrigo Sigal	<i>Friction of things in other places</i>	Acusmática
22	2002	Juan Sebastián Lach Lau	<i>rwz</i>	Mixta
23	2006	Alejandra Hernández	<i>Sublingual</i>	Acusmática
24	2006	Guillermo Galindo	<i>Aura6</i>	Acusmática
25	2006	Arturo Fuentes	<i>FOSIL KV</i>	Acusmática
26	2006	Israel Martínez	<i>Exorcizio I</i>	Acusmática
27	2007	Alejandro Montes de Oca Torres	<i>Imaginario I</i>	Acusmática
28	2007	Miguel Hernández Montero	<i>Pout</i>	Electrónica en vivo

Aunque una obra acusmática también puede usar sonidos de instrumentos acústicos, estos son grabados en estudio y sólo tiempo después sus grabaciones son procesadas y ordenadas en la obra –esto es, se realiza en *tiempo diferido*–. Esta observación es importante, ya que muchas obras de electrónica en vivo son designadas con el calificativo de “obra electroacústica en tiempo real”.

La pieza 28 de la tabla, *Pout*, de Miguel Hernández, es la única de los tres discos que es nombrada “para electrónica en vivo”. Nuevamente aparece la descripción de procesamiento en tiempo real: “La pieza se construyó con pequeñas muestras de *Sea Feathered Flute* procesadas en tiempo real con un *Kaoss Pad*, algunas muestras de violín y una guitarra eléctrica interpretada por el guitarrista Pablo Gómez” (Rocha, 2008, p. 66).

Respecto a las cinco obras incluidas en la categoría mixta, éstas se denominan de la siguiente manera:

3. *Viols*, para violín y cinta.
4. *Diálogos*, para piano y cinta.
10. *Papalotl*, para piano y cinta.
19. *Semi No Koe*, para flauta y cinta.
22. *rwz*, para 4 canales (versión estéreo).

Las piezas 3, 4, 10 y 19 nos indican claramente que combinan la acusmática (la cinta) con un instrumento acústico. En el folleto también se describe la obra *Diálogos* como pieza “para piano y sonidos electrónicos” (Rocha, 2008, p. 26). La denominación “para sonidos electrónicos” se utiliza aquí como equivalente de “para cinta”, correspondencia que se da a menudo, pero que es completamente ambigua, ya que los sonidos electrónicos se integran en cualquiera de las tres categorías.

Por último, la pieza 28, *rwz*, tiene una denominación totalmente inadecuada. Cualquier obra electroacústica, sea acusmática, en vivo, o mixta, puede tener el número de canales de audio que desee el compositor. Es necesario ir al folleto para entender que esta obra requiere de un percusionista y de un elemento acusmático:

La pista electrónica fue generada en el estudio analógico a partir de una grabación multi canal de la percusión, y fue hecha en varias pasadas, procesando el sonido de los instrumentos en las secciones externas y disparando sonidos sintéticos para la parte intermedia. [...] La ejecución en vivo de esta pieza tiene a los sonidos electrónicos

siguiendo a la percusión, así como a la percusión siguiendo a la electrónica, lo que se asemeja al proceso por el que atravesó la composición. (Rocha, 2008, pp. 57-58)

La designación más adecuada para esta obra, y que no dejaría duda alguna, sería “para percusión y cinta” –o “para percusión y pista electrónica”.

Se derivan algunas conclusiones del análisis anterior:

1. Que existe todavía gran confusión, incluso entre los mismos compositores, en la forma de describir y clasificar sus obras.

2. Que el número de obras acústicas con relación al número de piezas electrónicas en vivo sigue siendo abrumadoramente mayor; en el caso de esta selección de *México electroacústico* es de veinte contra tres.

1.2 Electrónica en vivo: oportunidades y limitaciones

Como se observó en la sección anterior, las obras electrónicas en vivo suelen denominarse con etiquetas ambiguas o poco claras. Por ejemplo, el calificativo “para medios electrónicos” –o su variante: “para sonidos electrónicos”– puede aplicarse tanto a una obra acústica como a una obra electrónica en vivo. Por el contrario, la denominación “en tiempo real”, ya explicada anteriormente, es completamente correcta –aunque podría ocasionar confusión en las audiencias no especializadas.

Otra etiqueta que suele asociarse con la electrónica en vivo es la de “interactivo”, ya que esta característica se usa como elemento diferenciador entre este tipo de obras con las acústicas y las mixtas. En las obras acústicas la ausencia de un instrumentista elimina la interacción de este con la audiencia y en las obras mixtas el intérprete no puede interactuar demasiado con la parte electrónica al tener que seguir siempre a una inflexible pista sonora. En una pieza para instrumentista y pista electrónica no hay espacio para un *rallentando* espontáneo del intérprete o para una serie de adornos improvisados: la parte acústica no se puede adecuar a ello y cualquier cambio de este tipo que realizara el intérprete comprometería la sincronización entre el sonido en vivo y el sonido pregrabado. El compositor Marco Stroppa, en un artículo donde critica fuertemente a la electrónica en vivo, tiene que admitir que “Las mezclas rítmicas fuertemente entrelazadas entre un intérprete y una máquina son todo el tiempo engorrosas, y virtualmente inviábiles con cintas” (1999, p. 44). Por su parte, Miller Puckette subraya el objetivo fundamental de la electrónica en vivo: “El principal propósito para crear música electrónica que

se efectuara en tiempo real es el que un músico pudiera ejercer control en vivo sobre la música en ejecución” (1991, p. 68).

Aunque desde la década de 1920 era posible escuchar música electrónica en vivo en un concierto del recién inventado theremin, gran parte de la producción de esta música sólo se hizo viable cuando ocurrieron los cambios tecnológicos que permitieron el procesamiento del sonido en el mismo instante que el músico toca su instrumento. Esto último puede explicar la diferencia de veintiséis años entre la primera obra acusmática registrada en *México electroacústico* –de Carlos Jiménez Mabarak, *Paraíso de los ahogados* (1960)– y la primera pieza de electrónica en vivo en esa misma compilación –*Erótica II* (1986) de Vicente Rojo Cama–. También a esa dependencia tecnológica se debe el hecho de que sólo tres obras, de las veintiocho contenidas en *México electroacústico*, sean para electrónica en vivo, misma relación entre número de obras acusmáticas y cantidad de obras de electrónica en vivo que se puede observar en una gran parte de las compilaciones y ediciones discográficas de obras electrónicas. Y con respecto a los conciertos de música electroacústica, también en estos las piezas de electrónica en vivo suelen ser escasamente incluidas.

Desde el punto de vista técnico, es mucho más sencillo producir un concierto con obras acusmáticas y mixtas que con electrónica en vivo puesto que con un sistema de amplificación modesto y un reproductor de música –que puede ser desde un reproductor de DVD hasta una pequeña computadora en tableta– es suficiente para llevarse a cabo. Por su parte, la obra de electrónica en vivo puede requerir de modelos específicos de procesadores de audio –los que pueden ser inaccesibles, tanto por motivos geográficos como por la rápida obsolescencia del hardware–, de uno o más micrófonos –indispensables para procesar los instrumentos acústicos–, y en ocasiones hasta de un músico adicional –que muchas veces es el mismo compositor– para activar los diferentes tipos de procesamiento sonoro y manipular los diversos parámetros de estos según lo indique la partitura.

La misma problemática que ocurre al realizar un concierto de música electroacústica se da en el momento de preparación de la pieza por el intérprete. Ensayar una obra mixta demanda condiciones técnicas que en la actualidad cualquier instrumentista puede tener en casa, pero en una obra con electrónica en tiempo real la situación se complica en demasía. La solución frecuente es que el compositor, si es que es cercano, le proporcione todo el equipo al intérprete, le enseñe a instalarlo, configurarlo y operarlo; pero sin asistencia de este tipo, un intérprete

tendrá fuertes motivaciones –muy válidas y comprensibles– para sustituirla por otra pieza sin tantos requerimientos de equipo y de conocimiento tecnológico.

Es más que obvia la dependencia de esta música del desarrollo tecnológico imperante, pero esta situación que limita la interpretación y producción de música electrónica en vivo, también se dio con las obras acusmáticas y mixtas: "Hasta la década de 1970, pocas salas de conciertos tenían equipos de reproducción y amplificación adecuados para interpretar música con cinta" (Boutard y Féron, 2018, p. 118). El progreso tecnológico va borrando sus propias huellas, lo que ocasiona olvidar que antes no contábamos con instrumentos musicales eléctricos, así como siglos atrás no teníamos trompetas con pistones; tener una visión histórica nos permite entrever que las herramientas tecnológicas serán cada vez más accesibles y económicas –y su funcionamiento más comprendido– como para permitir en distintos ámbitos una mayor presencia de la música electrónica en vivo.

1.3 Software comercial versus *custom* software

Con tantos programas y aplicaciones existentes en el mercado, sería muy lógico preguntarse si no existe ya un programa diseñado para la creación de música electrónica en vivo, de tal forma que el compositor sólo tuviera que utilizar la aplicación y dedicarse únicamente a componer en lugar de invertir gran cantidad de tiempo aprendiendo un lenguaje de programación. Pero también surge la duda de si un programa de esa naturaleza pudiera adaptarse a las diferentes necesidades, formas de trabajo y lenguajes tan diversos que pueden tener los compositores. Este problema puede surgir en diferentes campos de la programación; por ejemplo, en el ámbito de los programas de solfeo y entrenamiento auditivo se pueden encontrar bastantes opciones comerciales, pero si ninguno de ellos se adaptara al método de enseñanza-aprendizaje que uno maneja no quedaría más remedio que programarlo uno mismo, como a mí me ocurrió alguna vez:

Cuando de estudiante aprendía solfeo con el libro de Georges Dandelot (1895-1975), me imaginé la posibilidad de tener una versión computarizada del mismo: un programa que presentara en pantalla notas aleatorias sobre un pentagrama, a las que el usuario respondería pulsando en un teclado electrónico las teclas correspondientes; además de tener las tres etapas de aprendizaje de dicho método, podría añadirse la medición del tiempo de respuesta y una modalidad donde existiera una alternancia entre dos claves

cada cierto número de notas. Por supuesto que era imposible encontrar una aplicación así, basada en un método tan específico como el Dandelot, por lo que me di a la tarea de programarla con el lenguaje Basic que venía incluido en una computadora Commodore 128 que me prestaban en la escuela. (Govea, 2020b, p. 22)

Casos como este son los que llevan a considerar el uso del *custom* software –también llamado *bespoke*, o *taylor-made*–: programas hechos “a la medida”, encargados por una compañía, institución u organización que no encuentra ninguno en el mercado que se adecúe a sus necesidades específicas; por supuesto, al ser desarrollados exclusivamente para estos clientes, tienen un costo económico mucho mayor y una considerable inversión en tiempo –que puede ser desde varias semanas hasta años– para la programación, pruebas, detección de errores y ajustes –además de sesiones regulares de mantenimiento.

Aunque es perfectamente posible realizar una obra electrónica en tiempo real usando software comercial –tal vez usando una DAW (*Digital Audio Workstation*) con un músico adicional manejando los procesadores en esta–, emplear un entorno de programación como Max/MSP permite tener un programa “hecho a la medida” de la obra que el compositor vaya imaginando, sin las restricciones o tendencias que pueda marcar un programa secuenciador o de audio digital que se orienta a la generalidad de los músicos, y con posibilidades de interacción entre el usuario y el programa que son indispensables para la electrónica en vivo. Las desventajas son las mismas del software custom: la enorme cantidad de tiempo que puede suponer el aprendizaje de un lenguaje de programación, así sea con las facilidades del ambiente gráfico de Max/MSP y su gran cantidad de tutoriales, manuales y soporte para el usuario. Como en muchas situaciones, hay que valorar los beneficios contra los inconvenientes que se obtienen de cada decisión.

1.4 ¿Qué es Max-MSP?

En su sitio web oficial, se define a Max como “Un lugar infinitamente flexible para crear software para medios interactivos. Con herramientas completas de audio, gráficos, interacción y comunicación, Max es un entorno para explorar y desarrollar tus propias ideas” (“What is Max?: Cycling '74”, 2020). Esta es una descripción orientada a despertar el entusiasmo de futuros usuarios pero que deja enormes dudas sobre su función.

Más esclarecedora es la explicación que ofrece Miller Puckette –el creador de Max– en el manual de referencia de 1988: "Max es un entorno gráfico de programación musical para gente que ha llegado a los límites de los habituales secuenciadores y programas de voces para equipos MIDI" (citado en Zicarelli y Taylor, 2006, p.3). Basado en el lenguaje Music500 que trabajaba desde 1982, Puckette cambió su nombre a Max en 1986 (Puckette, 2002, p. 33) –en tributo al pionero de la música por computadora, Max Mathews– cuando lo adaptó para poder controlar el sintetizador 4X del IRCAM (Institut de Recherche et Coordination Acoustique/Musique), en París. Con la posterior colaboración de David Zicarelli, en 1991 fue comercializado por Opcode Systems y desde el año 2000 es manejado por la compañía Cycling '74 (Zicarelli y Taylor, 2006, p.3).

En un principio, Max era un lenguaje de programación pensado para el control en tiempo real de dispositivos MIDI (Winkler, 1998, p.18) –así era como manipulaba al sintetizador 4X, con cientos de mensajes controladores MIDI– pero después se amplió para ofrecer el procesamiento de audio con la adición de MSP (Max Signal Processing) y después el manejo de imágenes y video al agregar Jitter (las rutinas de programación creadas para tal fin).

A diferencia de los lenguajes de programación que se basan en texto, Max emplea una interfaz gráfica, utilizando cajas –con texto o iconos– que se conectan unas con otras llevando mensajes o audio. Su funcionamiento evoca al de los primeros sintetizadores analógicos, como los modulares de Moog o Buchla:

Los primeros sintetizadores analógicos que surgieron, en la primera mitad de la década de los 60, producían su sonido a través de unidades independientes –cada una con funciones específicas de generación, mezcla o modificación sonora– que se interconectaban entre sí para producir un timbre. Su conformación con base en estas unidades o módulos derivó en el nombre de “sintetizador modular”, que lo diferencia de muchos de los sintetizadores que surgieron después donde estas unidades ya vienen internamente conectadas siguiendo un orden definido por el fabricante y que le quitan al usuario la posibilidad de experimentar con diferentes arreglos de módulos.

[...] Los cables que llevan la señal, de audio o de control, de un módulo a otro se llaman “patch cords”, de tal forma que a una disposición particular de conexiones (o cableado) se le llamó “patch”; así fue como esta palabra comenzó a utilizarse como

equivalente de un sonido creado en un sintetizador –a pesar de que muchos sintetizadores ya no utilicen estos “patch cords”. (Govea, 2020a, p. 16)

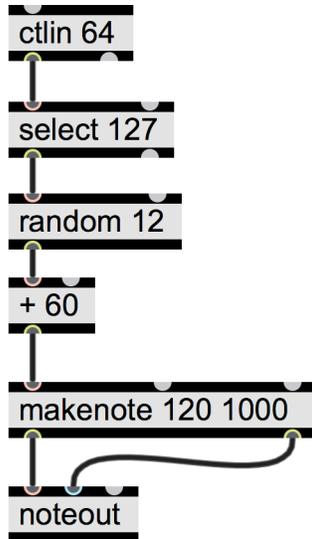
De similar manera, Max provee una estructura modular a través de estas cajas denominadas objetos, y a la disposición de los objetos y las conexiones entre ellos al formar un programa se le llama –de forma análoga al sintetizador– “patch” (Cipriani y Giri, 2010, p. 52) o “patcher”.

La mejor manera de entender cómo se programa en Max es a través de un ejemplo, como el que se observa en la Figura 1.

El siguiente patch manda a un sintetizador –u otro instrumento MIDI conectado a la computadora– la instrucción de sonar una nota cualquiera entre el do y el si de la octava central del piano –con duración de un segundo y una intensidad *forte*–, cada vez que se presiona el pedal MIDI de sustain.

Figura 1

Ejemplo de programa (patch) en Max/MSP



El patch utiliza los seis objetos siguientes:

1. [ctlin 64]: recibe mensajes MIDI llamados *controllers*; en este caso, el argumento “64” indica que los datos que le llegan son exclusivamente los del pedal de sustain. De la salida izquierda

(outlet) de [ctlin 64], se emite un número 127 cuando se presiona el pedal y un 0 cuando se suelta.

2. [select 127]: cuando el número que le llega de [ctlin] es 127, manda una señal ejecutora – “bang” es como le llaman en Max– al siguiente objeto al que está conectado.

3. [random 12]: al recibir el bang, genera un número aleatorio entre 12 posibles (del 0 al 11).

4. [+ 60]: el objeto de suma añade 60 al número recibido para que la nota a producir esté en el rango de do a si de la octava central del piano.

5. [makenote 120 1000]: el número recibido corresponde a una nota dentro del lenguaje MIDI, pero falta agregarle la información de la intensidad –el número 120 (de un rango de 0 a 127)– y la duración en milisegundos.

6. [noteout]: con la información de [makenote], hace sonar la nota requerida en el instrumento MIDI.

Esta forma gráfica de programación resulta más intuitiva y rápida de aprender que un lenguaje basado en texto, el que usa comandos y sintaxis que pueden resultar frustrantes para el músico –porque hay que recordar que este lenguaje está dirigido a los artistas, no a ingenieros ni a programadores–, características que fueron razones de gran peso en la selección de Max para la elaboración de las piezas electroacústicas –aunque es probable que con otro software como SuperCollider o Csound se hubieran logrado resultados parecidos.

En su conjunto, Max-MSP-Jitter cuenta con más de 600 objetos, lo que implica que llegar al dominio de los mismos requiere mucho más esfuerzo que el manejo de cualquier secuenciador o programa de edición de audio, esfuerzo que es compensado por la flexibilidad para adaptarse a los diferentes proyectos que pueden tener los compositores, educadores, intérpretes e investigadores musicales.

2. Análisis comparativo

Lo primero que hay que establecer es, de acuerdo con la clasificación presentada en el capítulo anterior, a qué categoría de la música electroacústica pertenecen cada una de las obras a analizar. *Subliminal* presenta en su primera y tercera parte –del inicio al minuto 2:26 y del minuto 6:41 hasta el final (indicaciones de tiempo según la versión grabada en el disco de Jeffrey Zeigler)– lo que se denominó como pieza mixta, ya que existe una parte electrónica en soporte fijo (o acusmática) al mismo tiempo que el instrumento eléctrico es tocado. La segunda parte de la obra (2:26 a 6:41), o sección central, está basada enteramente en procesamiento en tiempo real: la nota re más grave del violonchelo es grabada y después se procesa para crear un fondo electrónico y varios *glissandi* que ocurren simultáneamente a partir del minuto 5:34. Hay que añadir que en la primera parte también existe electrónica en vivo por medio de un armonizador que se aplica al cello (de 1:13 a 1:32 y de 2:10 a 2:13) al mismo tiempo que la pista electrónica está sonando.

En cuanto a *Ocaso de Tenochtitlan*, los tres movimientos hacen uso exclusivamente de la electrónica en vivo. No hay una pista electrónica establecida, aunque en el tercer movimiento se hace uso de dos sonidos percusivos pregrabados que son disparados a intervalos regulares de tiempo cuando el intérprete activa un metrónomo dentro del patch.

Ahora, en segundo lugar, explicaré la forma en que realizaré el análisis. Partiré de un ejemplo muy claro –aunque no muy reconocido, por hallarse fuera de la academia– de lo que es la música electroacústica en vivo: un músico que toca su guitarra eléctrica distorsionada en un concierto de rock. En ese escenario se ve al guitarrista conectar su cable a un pedal de distorsión y después, de la de salida de este, conectar un cable hacia el amplificador. Aquí el distorsionador es como una caja negra porque se desconoce su composición interna y su funcionamiento, aunque sí se sabe que como salida de este pedal se obtendrá el famoso sonido distorsionado de la guitarra de rock. Este modelo se puede aplicar a la programación del patch de Max/MSP para una obra electrónica en vivo, ya que debe contener una sección que reciba la entrada del audio, un bloque de procesamiento sonoro –que puede tener no sólo el distorsionador, sino también un armonizador, un reverberador, un flanger, o lo que se requiera– y la sección que manda el audio procesado, tal vez mezclado con el sonido sin procesar, a la salida de audio.

La definición de electrónica en vivo que se maneja en este trabajo dice que en estas obras “la tecnología es empleada para generar, transformar o disparar sonidos (o una combinación de

éstas) en el momento de la interpretación” (Emmerson y Smalley, 2001, pp. 59-60), por lo que aunado a los procesadores de audio –que llevan a cabo la transformación sonora– deben de existir otros dispositivos como generadores de ruido y osciladores –para generar sonidos– y reproductores de audio, secuenciadores y samplers –para disparar o reproducir sonidos.

Por consiguiente, el análisis de la programación de los patches para estas obras tiene que abarcar los siguientes elementos: 1) la entrada de audio, 2) la generación, transformación y reproducción de sonidos y 3) la salida de audio. Empezaré con la salida y entrada de audio porque son muy simples de comprender y dejaré el punto dos para después, ya que es el más extenso y minucioso. En cada caso, daré una breve explicación teórica y después mostraré cómo se presentan estas funciones en los patches de las composiciones examinadas.

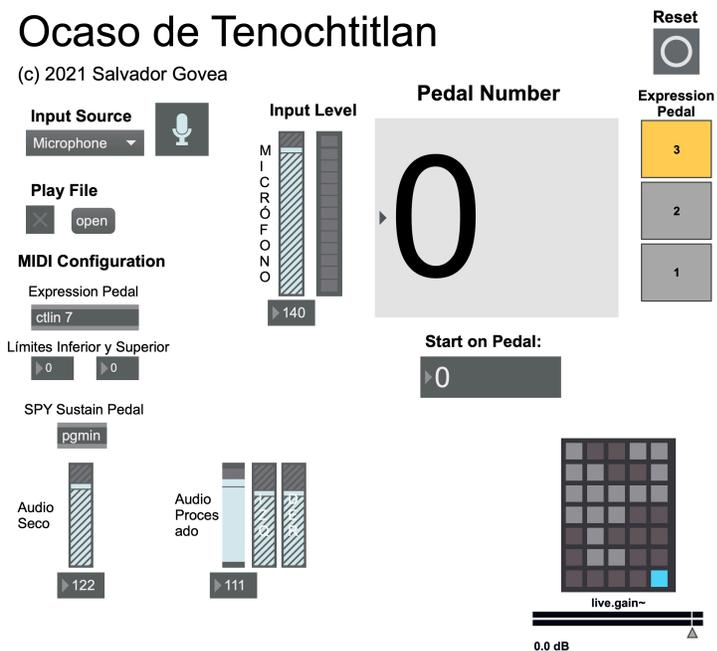
2.1 Entrada y salida de audio

La interfaz de audio es el dispositivo periférico que convierte la señal eléctrica a números –proceso llamado *digitalización*– para que la computadora pueda entenderlo y manejarlo. Esta señal puede provenir de un instrumento acústico, eléctrico o electrónico, y aunque la digitalización siempre ocurrirá de la misma forma independientemente de la señal que le llegue, la elección del tipo de instrumento presentará diferentes oportunidades para la creación de la obra electroacústica. Por supuesto que el proceso inverso, la conversión de los números del audio digital a una señal eléctrica, también lo realiza la interfaz de audio. Dicha interfaz es quien se comunica con el patch realizado en Max/MSP a través de los objetos que se muestran a continuación.

En la Figura 2 se observa la pantalla principal –en modalidad “Presentation”, para ocultar objetos y subpatches que no requiere visualizar el usuario– del patch de la pieza *Ocaso*. En la esquina superior izquierda se ubica un cuadro con el dibujo de un micrófono: este es la representación del objeto [ezadc~] –donde “adc” significa convertidor analógico a digital–, cuyo botón se prende y se apaga con un clic para que el audio del micrófono sea recibido o no por el programa. El rectángulo a la derecha de la palabra “micrófono” es un [gain~] a donde va el audio de [ezadc~] para controlar el nivel de la señal entrante –como un potenciómetro deslizable de una mezcladora–; de este [gain~] el audio irá a cada uno de los procesadores de sonido del patch, así como al [meter~] que se encuentra a su derecha, para indicar visualmente el nivel de entrada.

Figura 2

Pantalla principal del patch para Ocaso de Tenochtitlan



Debajo de las palabras “Input Source” se encuentra un menú donde se puede seleccionar entre “Off”, “Microphone” y “Audio File”, esto es, que no entre audio, que entre la señal del micrófono, o que entre el audio directo de un archivo de audio, opción última que usé para alimentar el programa con una versión MIDI de la tarola y probar con esta los diferentes procesamientos.

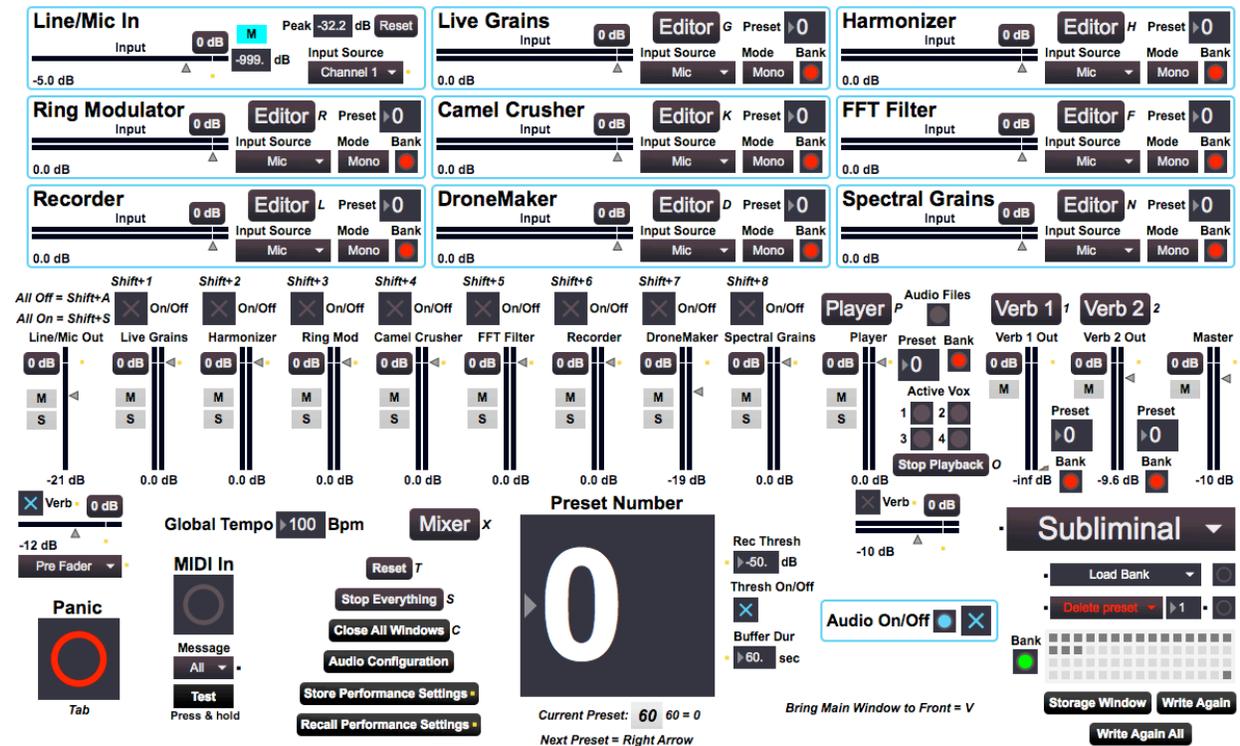
En cuanto a la salida, se utilizan un [gain~] para regular la salida de sonido no procesado; dos [gain~] más, controlados por un [slider], para la salida izquierda y derecha del audio procesado –a donde va el audio de todos los procesadores–, y un [live.gain~] –en el que se suman las señales del audio directo y el audio procesado.

En el patch de la pieza *Subliminal*, mostrado en la Figura 3, el recuadro superior izquierdo con el encabezado “Line/Mic In” utiliza un objeto [live.gain~], que tiene la misma función del [gain~] y el [meter~] de la entrada en el patch de *Ocaso*. Respecto a la salida del audio es muy útil visualizar en pantalla los niveles de salida de todos los procesadores y del módulo Player, así como de la salida del sonido directo y de la reverberación –para lo que usa en todos los casos el [live.gain~]–, lo que permite monitorear el flujo de las señales y detectar

posibles errores en la utilización del patch. Por supuesto, también cuenta con un interruptor para apagar y prender el audio: los botones al lado de las palabras “Audio On/Off” en la parte inferior derecha del patch.

Figura 3

Pantalla principal de Live Patch, usado para Subliminal



2.2 Generación, transformación y reproducción de sonidos

Siendo que la programación en Max/MSP es de tipo modular, es decir, trabaja con unidades independientes que se conectan o mandan información de una a otra, es muy provechoso usar el mismo concepto en esta parte del análisis. De hecho, en Max/MSP se crean módulos cuando un conjunto de objetos que cumplen una tarea se agrupan en la figura del *subpatch*, que visualmente es una pequeña caja rectangular con un nombre descriptivo y con diferentes lugares de entrada y salida de datos o señales –llamados *inlets* y *outlets*–. Por ejemplo, puedo crear un subpatch que saque el promedio de tres números, el que tendrá tres entradas –una para cada número– y una salida que dará el resultado; dentro del subpatch estarán los objetos de

suma y división para hacer el cálculo, aunque estos no estarán visibles hasta que se abra el subpatch –tal vez nombrado como “módulo de cálculo de promedio”.

Por consiguiente, hablaré de módulos de generación de sonido, módulos de procesamiento sonoro, y módulos de reproducción de sonido. Cada uno de ellos será una caja negra porque no hablaré de su constitución interna ni funcionamiento, pero sí del resultado sonoro que producen. Ahora detallaré las funciones de cada uno de estos grupos de módulos.

Generadores de sonido

Para la generación de sonido, Max/MSP ofrece objetos como [cycle~], [tri~], [saw~], [rect~], que son osciladores que producen, respectivamente, sonidos senoidales, de forma de onda triangular, de diente de sierra (*sawtooth*) y pulso. Hay generadores de ruido blanco y de ruido rosa con los objetos [noise~] y [pink~]. Por supuesto, se puede crear cualquier combinación de estos y conjuntarlos dentro de un subpatch generador de sonido o programar hasta un pequeño sintetizador de síntesis aditiva.

Procesadores de sonido

Aunque varias veces se ha hablado de los procesadores en este texto, es momento de establecer una explicación más extensa de los mismos. Miyara ofrece una definición general que dice que procesador es “todo dispositivo que opera algún cambio en una señal eléctrica” (2000, p. 298), la que coincide mucho con la definición de Gallagher, pero de la palabra “efecto”: “Un dispositivo de hardware o un procesador de software que modifique una señal. Los ejemplos incluyen el retardo, el eco, el desplazamiento de fase, el *flanging*, el coro, el trémolo y muchos otros” (2009, p. 64). Esto lleva a considerar si procesador y efecto son equivalentes –de hecho, en muchos contextos son utilizados como sinónimos–. Por otro lado, para Gallagher, un procesador es “un dispositivo que modifica o realza las señales de audio. Entre otros ejemplos se incluyen compresores, retardos, *flangers*, ecualizadores, reverberaciones y limitadores” (2009, p. 164), con lo que matiza un poco al añadir a los procesadores la función de realzar (*enhance*) la señal. De las definiciones de Gallagher, se podría establecer que los efectos son un subconjunto de los procesadores: todos los efectos son procesadores, pero no todos los procesadores son efectos –como los compresores y ecualizadores–. Pero hay que ver otras definiciones:

Los procesadores que hemos visto hasta el momento (por ejemplo, los compresores) tenían la particularidad de que procesaban la totalidad de la señal que los atravesaba.

Existe otro gran conjunto de procesadores denominados genéricamente **procesadores de**

efectos, o simplemente **efectos**, que se utilizan mezclando la señal procesada (señal “húmeda”) con parte de la señal sin procesar (señal “seca”). (Miyara, 2000, p. 174)

Paul White los describe en el mismo sentido:

Aunque ambos tipos son estrictamente hablando procesadores de algún tipo, he usado el término procesador aquí para describir aquellos dispositivos que tratan la totalidad de la señal y el término efecto para aquellos en los que hay una mezcla de sonido procesado y no procesado. (2000, pp. 64-65)

Por lo tanto, tenemos aquí dos usos de la palabra “procesador”, uno en sentido general – como todo aquel que modifica de alguna forma el audio– y otro para designar al grupo de dispositivos que actúan en toda la señal y que se distingue así de los efectos – donde sólo una parte se procesa y se mezcla con otra parte sin procesar–: este uso ambivalente de esa palabra es el que causa mucha confusión, pero no queda más que aceptarlo y tratar de entender qué uso se le da en cada situación. De tal forma, dentro del primer grupo se enlistan los siguientes: ecualizador, compresor, limitador, compuerta, expansor, atenuador, distorsionador, corrector de altura, entre otros; y en el segundo grupo se incluyen: reverberador, retardo, coro, flanger, desplazador de altura, armonizador y varios más (Izhaki, 2008, p. 101; Miyara, 2000, p. 175).

Sound manipulator es el término general que utiliza David Gibson (2019) para todo aquel dispositivo que procese la señal –ya sea del primer grupo enunciado o del segundo– y es de esa manera que yo uso la palabra “procesador” en este trabajo.

Reproductores de sonido

Aquí se incluyen las grabadoras y reproductores de audio, los secuenciadores –que son grabadoras, pero no de audio sino de información MIDI– y los samplers –instrumentos que permiten sonar muestras digitales de sonidos que han sido previamente grabados, editados y configurados para su reproducción.

2.3 Módulos de generación, transformación y reproducción sonora utilizados en cada obra

Subliminal

La pantalla principal de *Live Patch* –el programa diseñado por Esteban Chapela para estudiantes de diversos talleres y cursos de composición– muestra, en su versión 2.2, ocho recuadros que representan procesadores (no incluyo al rotulado como “Line/Mic In” puesto que es la unidad de entrada de audio). A continuación, se muestra una breve descripción de cada uno

de los procesadores proporcionados por el programador (más adelante se profundizará en aquellos que utilicé en la obra *Subliminal*).

1. Live Grains. Basado en el patch Granularized, que forma parte de los ejemplos del menú Extras de Max/MSP, utiliza procedimientos de síntesis granular para modificar el sonido. “La síntesis granular construye sonidos a partir de cortos segmentos de sonido llamados ‘granos’. Al igual que muchas imágenes de las revistas en color se componen de muchos puntos, la síntesis granular utiliza diminutos fragmentos sonoros para producir sonidos” (Russ, 2009, p. 294).
2. Harmonizer. Es un armonizador, esto es, crea copias del sonido entrante a diferentes intervalos hacia arriba y debajo de este, para crear una especie de acorde.
3. Ring Modulator. A partir de dos señales de entrada, produce de salida la suma y diferencia de estas. Por ejemplo, dos sonidos de frecuencia de 350 hertz y de 420 hertz que entran, producirá uno de 770 Hz –la suma– y otro de 70 Hz de salida –la diferencia.

El hecho de que los moduladores de anillo produzcan sonidos matemáticamente relacionados en lugar de sonidos relacionados armónicamente ha impedido su aceptación generalizada en la música popular, pero para producir efectos atonales y sonidos percusivos y parecidos al gong, o para añadir una estructura armónica compleja a un instrumento, el modulador de anillo es muy útil. (Anderton, 1980, p. 99)
4. Camel Crusher. Es un *plug-in* de distorsión producido por Camel Audio que Chapela incluyó en *Live Patch*, diseñándole una interfaz de usuario para controlar los parámetros del mismo. Ofrece dos tipos de distorsión, un compresor y un filtro pasa bajos con control de resonancia y frecuencia de corte. Es descargable gratuitamente y existe en versiones para Windows y Macintosh.
5. FFT Filter. Filtro creado a partir de *Forbidden Planet*, un patch proporcionado en los ejemplos de Max.
6. Recorder. Aunque una grabadora no es un procesador de sonido, este módulo añade la posibilidad de modificar la duración del audio grabado, de transportarlo y hasta de reproducirlo en ciclo.
7. DroneMaker. Plug-in gratuito creado por Michael Norris como parte de su colección *Sound Magic Spectral*. Su nombre original es Spectral DroneMaker y su función es “convertir cualquier sonido en un hermoso bordón de cambio lento basado en las características sónicas del sonido subyacente” (Norris, 2016, p. 6).

8. Spectral Grains. Adaptado también del grupo *Sound Magic Spectral* de Norris, el plug-in Spectral Granulation funciona así: “Los ‘granos’ o ‘pedazos’ del espectro, de un cierto rango de frecuencia y duración son tomados y retrasados por cierto tiempo” (Norris, 2016, p. 13).

De estos ocho procesadores, sólo utilicé cinco en *Subliminal*: Harmonizer, Recorder, DroneMaker, Spectral Grains y FFT Filter. También usé el módulo llamado Player –que clasifiqué como reproductor de sonido–, que es un importante agregado en *Live Patch* para permitir la reproducción de un archivo de audio que funcione como pista electrónica.

Enseguida se presenta una relación de los módulos usados desde el principio hasta el final de la obra, indicando los compases en que se activan y desactivan (puede consultarse la partitura en el Apéndice de este trabajo).

1. Compás 1.

La pieza *Subliminal* comienza con la reproducción de un archivo de audio a través del módulo Player: una pista con sonidos electrónicos con duración de 2:26 minutos.

2. Compás 65: Se activa el procesador Harmonizer.

3. Compás 81: Se desactiva el procesador Harmonizer.

El Harmonizer añade dos notas al sonido del violonchelo: una quinta y una octava arriba de la nota tocada –la octava con mayor volumen que la quinta–, con un paneo aleatorio en las notas agregadas. El Harmonizer va entrando poco a poco hasta llegar a su volumen máximo en un lapso de nueve segundos por medio de un *fade in* incorporado en el procesador.

4. Compás 107: Se activa el procesador Harmonizer.

5. Compás 117: Se desactiva el procesador Harmonizer.

Se aplica nuevamente el Harmonizer pero esta vez los parámetros son diferentes. Se añaden tres sonidos, uno a la octava inferior y dos sonidos a la octava superior. No hay fade in esta vez.

6. Compás 122: Comienza la grabación.

Al terminar la reproducción de la pista de audio, se utiliza el Recorder para grabar la nota re índice 3 que toca el violonchelo. Aunque la entrada de audio es una, la nota se graba en cinco localidades de memoria porque más adelante se aplicará un distinto procesamiento a cada una de estas grabaciones.

7. Compás 126: Termina la grabación y se activa el procesamiento sobre esta.

Al mismo tiempo que llega a término la grabación, se activa el DroneMaker, el que utiliza la nota grabada como materia prima a ser modificada –con el audio que recibe del módulo Recorder en lugar de la entrada directa del audio del violonchelo–. El DroneMaker se hace notorio poco a poco, a través de un aumento gradual del parámetro Hi bin cutoff –la frecuencia de corte de un filtro pasa bajos– que va desde un valor de 977 Hz hasta 10309 Hz en el curso de 13 segundos, lo que se traduce en un aumento de la brillantez del sonido y colateralmente un incremento de la intensidad. Se establece después un efecto de flanger, procesador que “mezcla la señal de entrada con su propia versión retrasada con una cantidad de retardo variable, normalmente inferior a 20 ms” (Escriván, 2012, p.37). La velocidad de cambio del retardo es de 750 milisegundos –esto es, una frecuencia de 1.33 Hz– estableciendo así un pulso aproximado de 80 golpes por minuto. El intérprete debe agrupar mentalmente estos pulsos para tener una métrica ternaria sobre la que se desarrollarán los siguientes motivos.

8. Compás 144.

Recorder sigue reproduciendo la nota grabada –multiplicando su duración por 17– pero nunca se envía su sonido a la salida de audio principal sino a las entradas de audio del DroneMaker y del Spectral Grains. Se prende el Spectral Grains para producir sonidos agudos que aparecen aleatoriamente en las bocinas izquierda y derecha.

9. Compás 154.

Continúa el mismo fondo electrónico, cambiando algunos parámetros del DroneMaker, como el Gain, que disminuye siete decibeles, y el Flanger Amount, que aumenta aproximadamente diez por ciento.

10. Compás 174.

Se activa el Harmonizer, que modifica el sonido del violonchelo haciendo que suene con una octava grave añadida. Hay también paneo aleatorio.

11. Compás 188.

Se apagan Harmonizer y Dronemaker, aunque el último llega al volumen cero en un tiempo de 12 segundos.

12. Compás 197.

Spectral Grains se apaga, tardando seis segundos en desaparecer su efecto. Se prende el FFT Filter, actuando sobre la nota que le manda el Recorder –que en ese momento ha sido

transportada 14 semitonos hacia arriba—. El intérprete mantiene esta sonoridad todo el tiempo que desea antes de activar el siguiente proceso.

Durante la siguiente sección, el FFT Filter siempre continuará sonando y el Recorder efectuará transposiciones y glissandi sobre la nota que se grabó en cinco diferentes localidades de memoria del mismo –y que llamaré voces 1 a 5.

13. Compás 199.

La voz 1 realiza un glissando descendente y las voces 2 y 3 se prenden, cada una con una nota diferente (en la Figura 4 se observa la altura a la que es transportada la nota original en cada voz, así como la duración de los glissandi y puntos de origen y destino de los mismos).

14. Compás 202.

La voz 1 se mantiene en la misma nota mientras voces 2 y 3 realizan glissandi descendentes. Se agregan las voces 4 y 5, cada una con diferente nota.

15. Compás 205.

La voz 1 asciende en glissando, las voces 2 y 3 se mantienen, y las voces 5 y 6 descienden en glissando.

16. Compás 207.

Las voces 1 a 3 se mantienen estables mientras las voces 4 y 5 efectúan glissandi ascendentes.

17. Compás 209.

Todas las voces se mueven en glissando: las tres primeras de forma ascendente y las restantes descendiendo. Se activa el DroneMaker pero su efecto entra gradualmente al cambiar el parámetro Gain de -25 dB hasta 0 dB en el transcurso de 12 segundos; la velocidad del flanger es de 4.33 Hz, lo que establece pulsaciones a 260 por minuto –donde el pulso es igual a una figura de octavo.

18. Compás 221.

El Player reproduce otro archivo de audio con una pista electrónica que sonará hasta el final de la pieza.

Figura 4

Sección de glissandi electrónicos (a partir del compás 197) de Subliminal

The image shows a musical score for five voices, labeled Voz 1 through Voz 5. The score is written in bass clef for all staves. Measures 12, 13, 14, 15, 16, and 17 are circled. Slurs with '4 segs.', '6 segs.', and '8 segs.' indicate glissandi durations. Voz 4 and Voz 5 have '8va-' and '8vb-' markings. Voz 1 has a sharp sign on the final note.

Ocaso de Tenochtitlan

En esta obra, los módulos de procesamiento, generación y reproducción de sonido fueron creados expresamente para la misma –si bien varios de ellos tomaron como base un patch de otro autor–. En ciertas ocasiones la idea musical sugirió el módulo necesitado, y en otras, la experimentación en un procesador insinuó una idea o pasaje musical. De cualquier forma, en el patch de *Ocaso*, a diferencia del de *Subliminal*, no existen más que los módulos necesarios para la obra. La lista siguiente presenta los módulos sonoros usados en la obra y la explicación del resultado sonoro que logran.

1. SeguidorEnv. Su nombre viene de “seguidor de envolvente” (*envelope follower*). La envolvente es la gráfica que describe la evolución de la amplitud de un sonido en el tiempo, un perfil que nos permite ver el ataque lento de una nota de un instrumento de cuerda frotada o el ataque rápido, casi instantáneo, de un sonido de percusión. El seguidor de envolvente puede extraer la envolvente de un sonido y utilizarlo como señal de control; en el caso de este procesador, controla la frecuencia de corte de un filtro y el tiempo de retraso de un delay. Esta tomado y adaptado de 07_01_envfollow.maxpat de Giri y Cipriani (2020, p. 376)

2. PasaBanda. Es un filtro Butterworth pasa banda que utiliza el objeto [vs.butterbpbw~] de Maurizio Giri (Cipriani y Giri, 2019, p. 396). La frecuencia central y el ancho de banda son los dos parámetros de este filtro que modifiqué para obtener muy diversos timbres al aplicarse al sonido de la tarola.
3. Delays. Un delay realizado con el objeto [tapin~], que contiene un filtro antes de llegar la señal de audio al delay.
4. Flanger. Un flanger muy simple hecho con [tapin~]. Incluye los parámetros de velocidad, delay mínimo, rango, profundidad y retroalimentación.
5. PB-Trans. Es una variante del procesador PasaBanda que permite el cambio continuo entre dos configuraciones –o presets– del mismo.
6. Paneo. Cambia la posición del audio gradualmente de la salida izquierda a la derecha a diferentes velocidades.
7. PB-Final. Otro procesador derivado de PasaBanda, que utilicé al final del segundo movimiento de la obra a manera de un “orquestador” electrónico.
8. Armon. Este módulo es un generador de sonido que incluye dos filtros, pero estos no procesan la señal de entrada de la interfaz sino el audio producido por un oscilador con onda triangular. La frecuencia de corte de los filtros –un pasa bajos y un pasa altos– se mueve gradualmente hacia las frecuencias correspondientes a algún armónico –escogido aleatoriamente– del sonido emitido por el oscilador. Hay también un *subpatch* para panear el sonido de izquierda a derecha según el número de armónico que se refuerza, de tal forma que los primeros armónicos –o más graves– se escucharán más por la bocina derecha y los armónicos superiores –o agudos– se emitirán cargados hacia la bocina izquierda.
9. Samp. Básicamente es un reproductor de dos pequeñas grabaciones, o *samples*, de sonidos percusivos, los que se disparan por un metrónomo interno para crear una métrica a cinco sobre la cual la percusión desarrollará sus frases.
10. Noise. Un generador de ruido blanco que a través de un filtro pasabanda se modificará cíclicamente para crear diversos efectos rítmicos en la obra.
11. Reson. Juego de ocho resonadores con diferentes afinaciones, intensidades y localizaciones, que producirán una sonoridad metálica al modificar el audio de entrada.
12. 5TET. Generador de sonido que usa ondas senoidales y ruido blanco filtrado para producir alturas sobre una escala de cinco sonidos con temperamento igual.

En las Tablas 2 a 4, se observa en qué compás de la obra se activan cada uno de los módulos (en el Apéndice de este trabajo puede consultarse la partitura).

Tabla 2

Módulos utilizados en “Presagios”, primer movimiento de Ocaso de Tenochtitlan

I. Presagios				
No.	Compás	Módulo	Preset	Observaciones
1	1	SeguidorEnv	16	
2	5	SeguidorEnv	16	
3	10	SeguidorEnv	17	
4	12	PasaBanda	19	Con fade in de 12 segs.
5	16	SeguidorEnv	19	
6	22	Delays y Flanger	2 2	
7	35	Se apagan Delays y Flanger	—	
8	38	SeguidorEnv	22	
9	39	SeguidorEnv	22	
10	40	Flanger	2	Con feedback después de 10 segs.
11	55	SeguidorEnv	17	

Tabla 3

Módulos utilizados en “Soles de metal fino”, segundo movimiento de Ocaso de Tenochtitlan

II. Soles de metal fino				
No.	Compás	Módulo	Preset	Observaciones
12	1	PasaBanda	25, 26 y 27	Se selecciona el preset con el pedal de expresión.
13	49	PB-Trans	17	
14	58	Se apaga PB-Trans Se activa Paneo	—	El pedal de expresión selecciona la velocidad lenta, media o rápida.
15	100	Paneo	—	Velocidad muy rápida.
16	111	Se apaga Paneo	—	
17	116	PasaBanda	25 a 27	
18	129	PasaBanda PB-Final	26 —	De aquí al 21 continúa prendido PasaBanda.
19	132	PB-Final	—	Abre 2ª. salida de audio.
20	135	PB-Final	—	Abre 3ª. salida de audio.
21	138	PB-Final	—	Abre 4ª. salida de audio.

Tabla 4

Módulos utilizados en “Pólvora, gérmenes y acero”, tercer movimiento de Ocaso de Tenochtitlan

III. Pólvora, gérmenes y acero				
No.	Compás	Módulo	Preset	Observaciones
22	1	Se prenden Armon y SeguidorEnv	2 y 4 ——	A los 35 segs. cambia a preset 4.
23	34	Samp	——	
24	63	Noise y Reson	2 y 6	
25	81	Reson Noise	Varios ——	Automatización de cambios de presets en Reson.
26	150	Se apaga Samp	——	
27	152	Sale Noise y entra 5TET	——	Con fade out y fade in.
28	166	5TET	——	Se añade oscilador de onda senoidal.
29	199	Se apaga 5TET	——	Con Fade out.
30	206	Noise	——	Genera pulso.
31	221	SeguidorEnv	——	
32	256	Noise	——	Genera ostinato rítmico.

Descritos los módulos de cada patch, es posible ahora establecer semejanzas y diferencias entre ambos.

1. Hay que hacer notar que tanto Esteban Chapela como yo estamos haciendo uso del trabajo de otros autores al incluir sus procesadores y otro tipo de patches en nuestros programas: como los de Michael Norris, el plug-in de distorsión “Camel Crusher”, los ejemplos de la ayuda de Max/MSP que se usan en *Live Patch*, y los módulos que yo programé con base en los ejemplos de Cipriani y Giri, de Manzo y de los tutoriales y ejemplos incluidos con Max-MSP. Es por esta razón que en este trabajo no se discute ni se estudia la programación de estos módulos, por la facilidad con que otros autores nos prestan su trabajo –citándolos, por supuesto– y porque hay mucha documentación, tutoriales y libros donde se examinan con detalle patches utilizables por todos.

2. Tanto en *Subliminal* como en *Ocaso* es posible activar varios módulos al mismo tiempo, sin ninguna restricción para ir aumentando o disminuyendo durante el transcurso de la obra la cantidad de módulos que operan simultáneamente.

3. En los patches para ambas obras es posible el almacenamiento de presets, esto es, tener lugares en la memoria para guardar diferentes combinaciones de los parámetros internos de cada módulo y obtener así distintas variedades de sonido a partir del mismo procesador o generador.

4. Posibilidad de fade in y fade out en los módulos y de automatización de algunos parámetros de los mismos.

5. Una característica muy valiosa de *Live Patch* –y que no implementé en mi programa– es la capacidad de seleccionar el origen del audio de entrada al módulo, ya sea del micrófono o de algún otro módulo. Obviamente, también se puede determinar a donde se manda el audio de salida de un módulo, ya sea al master de salida o a otro módulo.

2.4 Activación y desactivación de los módulos

Asistente a cargo de la electrónica

En la tesis de Alejandra Odgers *La música electroacústica en México*, se encuentran algunas descripciones interesantes de obras que agrupó bajo el apartado de “Instrumentos acústicos modificados en tiempo real”. En la obra *Para la diosa de la lluvia* (1983) de Arturo Salinas, para flautas y ocarinas prehispánicas “con sistema electroacústico espacial” (Odgers, 2000, p. 202) se indica explícitamente la necesidad de un intérprete y dos asistentes para su ejecución. Estos asistentes son los que están a cargo del procesamiento en vivo y ejemplifican una de las formas primeras en que este tipo de piezas se llevaban a cabo, donde a menudo era el mismo compositor quien operaba los procesadores para llevar a buen término la interpretación de dichas piezas. El mismo Arturo Salinas dice que “para mis obras en vivo y para las instalaciones considero que es prácticamente indispensable la presencia del compositor” (Odgers, 2000, p. 202).

Si el compositor no es el asistente, se requerirá otro músico –o un ingeniero de audio que pueda leer música– para que siga con precisión la partitura y las indicaciones en la misma para activar y desactivar los diferentes procesadores o algún cambio en cierto parámetro de los mismos, tal cómo pide Bernardo Feldman para la ejecución de su obra *¡Oh, cetecho!* (1987): “un músico o ingeniero de sonido manipula por medio de un procesador de efectos de sonido, alterándolo, cambiando el eco, la reverberación, haciendo transposiciones, etc.” (Odgers, 2000, p. 192).

Esta forma de trabajar pudiera parecer algo antigua, pero llega a ser muy efectiva, y no me parece muy diferente al caso de un organista que requiere de uno o dos músicos asistentes que, siguiendo las anotaciones hechas por el intérprete en la partitura, cambian los registros del órgano tubular en el momento de la ejecución –procedimiento que funciona muy bien–. Sin

embargo, en la música electrónica en vivo el asistente asume un papel de tan grande importancia que se le puede considerar como otro intérprete:

Los intérpretes incluyen instrumentistas, así como ingenieros de sonido y músicos electrónicos en vivo. El término "instrumentista" se refiere al agente humano que toca un instrumento musical "tradicional". El término "músico electrónico en vivo" designa al agente humano encargado de la electrónica durante la actuación. (Boutard y Féron, 2018, p. 104)

Por supuesto que este esquema tiene sus inconvenientes para la interpretación regular de estas obras ya que pocos músicos tienen el entrenamiento para operar y configurar el equipo electrónico necesario, por lo que si el compositor no puede estar auxiliando en el concierto tendrá que enseñar a alguien a manipular los electrónicos. Es por esto que el uso de un software como Max/MSP para el manejo de los procesadores y demás dispositivos ofrece importantes alternativas que hay que analizar.

Intérprete

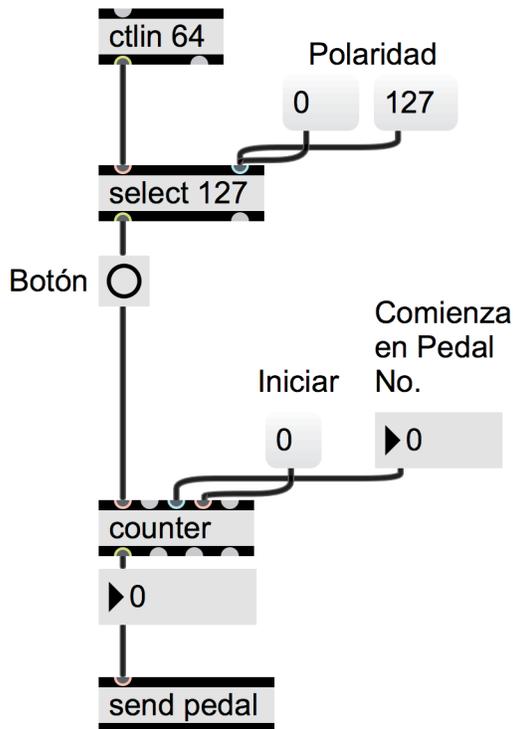
El dispositivo principal que se usa en *Subliminal* y *Ocaso* para transitar de un módulo a otro, es decir, para prender y apagar el que se requiere en cada momento de la obra, es un pedal MIDI de *sustain*. Este pedal también se conoce como *switch* pedal ya que sólo puede ubicarse en dos posiciones –presionado y suelto–, como el prendido y apagado de un interruptor (por cierto, también se le denomina como *dampers* pedal). Este pedal es accionado por el intérprete en los puntos especificados en la partitura, mientras el patch de Max/MSP lleva el conteo de las veces que se ha presionado el pedal, y según este número se activarán o desactivarán ciertos módulos (por supuesto que al inicio de la pieza el número del pedal se establece en cero).

En *Subliminal*, el violonchelista tiene un total de 18 indicaciones de pedal para toda la pieza, donde, por ejemplo, el pedal número dos prende el armonizador y el número tres lo apaga, el pedal seis inicia la grabación de una nota de violonchelo y el pedal siete detiene la grabación al mismo tiempo que prende el procesador DroneMaker –lo que ejemplifica la posibilidad de programar que un determinado número de pedal active o desactive cualquier combinación de módulos que se requiera–. En la relación de cómo usé los diferentes módulos en *Subliminal* –que se puede consultar más arriba– cada uno de los números de la lista equivale al número de pedal especificado en la partitura. Lo mismo puede observarse en las tablas de los movimientos de *Ocaso*, donde para toda la obra el intérprete debe presionar 32 veces el pedal.

En la Figura 5 se ve la programación requerida para captar las pulsaciones del pedal de sustain y llevar el conteo de estas.

Figura 5

Programa que recibe las pulsaciones del pedal y lleva el conteo de las mismas



1. [ctlin 64]: el nombre del objeto viene de *controller in*, porque en él entran mensajes MIDI llamados controladores; el argumento “64” se refiere a que recibirá información del controlador 64, que representa al pedal de sustain. Al presionar el pedal saldrá un número 127 de este objeto y un 0 al soltarlo.
2. [select 127]: si el número que le llega de [ctlin] es 127, mandará una señal ejecutora (“bang”) al siguiente objeto al que está conectado –si el número que le llega es 0, entonces no hará nada–.
3. [button]: Este es el cuadrado con un círculo a la derecha de la palabra “Botón”. Al recibir el bang, el botón parpadeará y mandará la misma señal de bang de salida (en realidad, este objeto no es indispensable, pero es un indicador visual para el programador de que se está recibiendo correctamente la señal).

4. [counter]: contará cada una de las señales de bang que le lleguen y desplegará en el objeto siguiente, llamado [number], cada uno de los números consecutivos, que corresponden a cada bang y que corresponden a su vez a cada pulsación del pedal.
5. [send pedal]: este objeto mandará cada número a un objeto llamado *receive* al que le siga la palabra “pedal” –palabra que sirve como identificador–, esto es a un [receive pedal].

Lo anterior describe el funcionamiento básico del patch, pero hay que agregar algunos detalles. Los números debajo de la palabra “Polaridad”, son mensajes para [select], lo que es necesario porque hay pedales de sustain que al presionarlos mandan el 0 y al soltarlos mandan el 127 –al contrario de cómo se describió anteriormente–, y en ese caso habrá que presionar el mensaje 0 para que ese tipo de pedales funcionen correctamente. En cuanto al mensaje 0 debajo de “Iniciar”, su función es restablecer el contador si se quiere retomar la pieza desde el inicio. Por otro lado, también se puede partir de cualquier pedal intermedio de la pieza; por ejemplo, si la pieza *Ocaso* quiere tocarse a partir del segundo movimiento, habrá que clicar en la caja [number] debajo del título “Comienza en Pedal No.”, escribir el número 12 –que es el pedal con que inicia el movimiento– y presionar la tecla *Enter* de la computadora: en cuanto se presione el pedal, el [counter] saltará directamente al número 12, entonces se activará el módulo PasaBanda y se podrá comenzar a tocar esa sección. Esta última función es muy importante para las sesiones de ensayo, porque el instrumentista tiene así la posibilidad de practicar la sección que quiera sin necesidad de partir desde el principio de la obra.

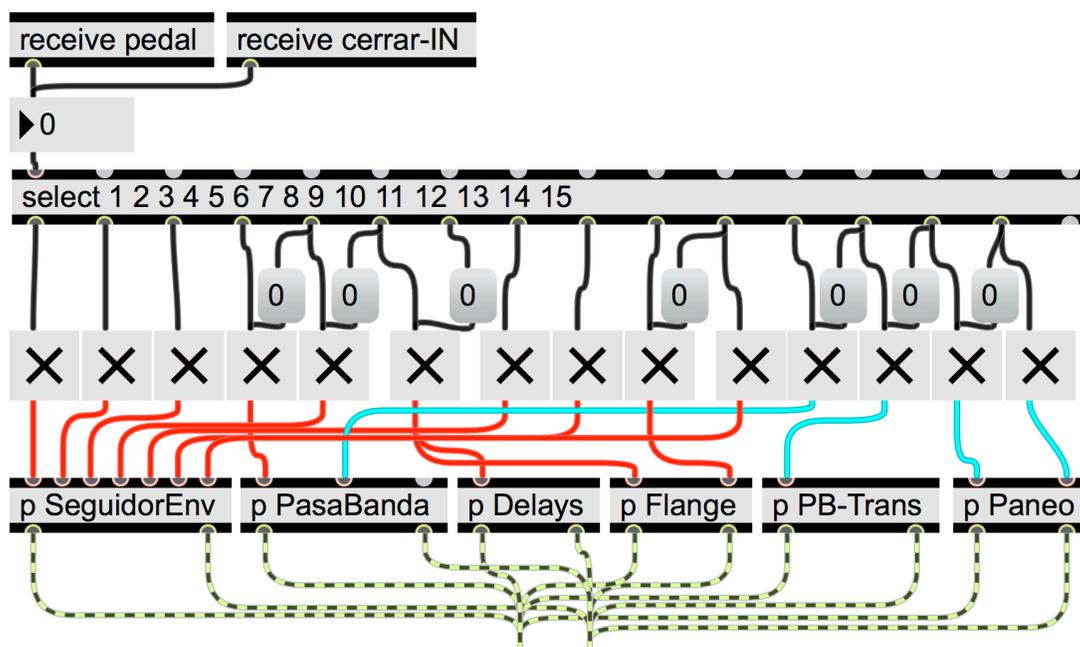
El siguiente punto importante en la programación, es el procedimiento mediante el cual el número de pulsación del pedal selecciona el módulo a activar o desactivar. La clave se encuentra en el objeto [select] que recibirá el número de pedal y mandará un bang de ejecución por la salida correspondiente a dicho número (ver la Figura 6). Presento a continuación la explicación detallada.

1. [receive pedal]: recibe de [send pedal] el número de pedal pulsado.
2. [number]: despliega aquí el número que llegó a [receive pedal] y lo envía al siguiente objeto –aquí [number] aparece únicamente para la visualización del programador.
3. [select]: manda una señal de ejecución por la salida que tiene el mismo número que recibió.
4. [toggle]: es el cuadrado con una “X” en él. Funciona como un interruptor que con un bang se prende –manda un 1 por su salida– y con otro se apaga –sale un 0.

5. [p SeguidorEnv], [p PasaBanda], [p Delays], etc.: la p significa que es un subpatch, esto es, dentro de él hay una serie de objetos que realizan una tarea específica con los datos que entran y dan salida a cierto resultado. Cada uno de los subpatches mostrados aquí es un módulo de procesamiento, generación o reproducción de sonido que utiliza la obra *Ocaso*, los cuales se activan en el momento que reciben un 1 del toggle y se apagan al recibir un 0 en su entrada.

Figura 6

Selección del módulo a activar o desactivar en Ocaso



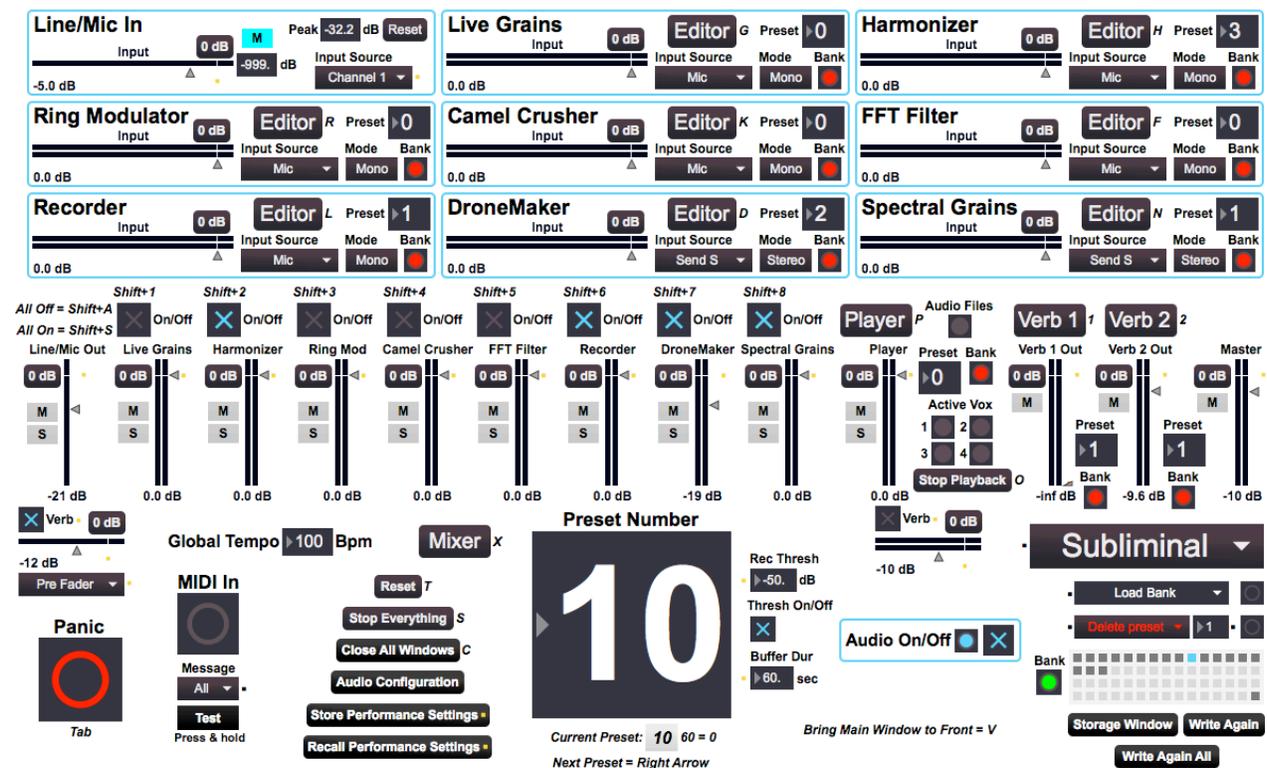
Puede notarse que en la parte superior de [p SeguidorEnv] existen siete entradas correspondientes a las siete veces que se utiliza en la obra. Dependiendo de la entrada, diferentes configuraciones de los parámetros de [p SeguidorEnv] –que no se ven aquí a menos que se abra esta “caja negra”– producirán diferentes resultados sonoros de salida.

En la obra *Subliminal*, el programador utiliza un método diferente para que el pedal seleccione los diversos módulos: haciendo uso del objeto [preset]. Dice Colasanto: “Al colocar un objeto *preset* dentro de un patcher, es posible memorizar el estado de todas las herramientas gráficas presentes” (2010, p. 239). En la Figura 7, se observa el [preset] en la esquina inferior derecha –después de la palabra “Bank”–. Dentro del rectángulo que forma el [preset], se tienen

60 cuadrados que representan las localidades de memoria –cada uno de ellos, de forma individual, también es llamado preset– para almacenar los estados de muchos de los objetos que están en la ventana principal del patch. Los cuadrados de color claro son presets sin usar, los oscuros tienen datos almacenados, y el que está seleccionado en ese momento –en la imagen, el número 10– utiliza un color diferente. En la figura se puede apreciar que el preset 10 fue seleccionado –en el momento de la décima pulsación del pedal–, y dicho preset activa los módulos de Harmonizer, Recorder, Dronemaker y Spectral Grains–lo que se puede identificar por el color del [toggle]–. También en el preset 10 está almacenada otra información, como los volúmenes de salida de estos procesadores, el origen de la entrada de audio a ellos, el preset interno que utilizan –por ejemplo, el 3 en el Harmonizer y el 2 en el DroneMaker–, la cantidad de reverberación que se le aplica al sonido del violonchelo, y las posiciones de múltiples *sliders* e interruptores.

Figura 7

Selección del módulo a activar o desactivar en Subliminal



Existe también la posibilidad de usar otros detonadores para activar los módulos en lugar del pedal de sustain. Una de las primeras opciones que se contempló para la pieza *Ocaso*, fue utilizar un “tambor” de batería electrónica (*drum pad*), al que podría adaptarse fácilmente el intérprete; sin embargo, la poca disponibilidad de estos pads –comparados con la de un pedal de sustain– me hicieron cambiar de opinión. De cualquier forma, es fácil hacer la adecuación en el programa para usar un pad, lo único que hay que cambiar es el [ctlin] por un [notein] y el número dentro del [select] tendría que ser el mismo del número de la nota MIDI que envía el tambor electrónico. De hecho, los dos patches de las piezas tienen la opción de usar como detonador una nota MIDI cualquiera en lugar de un mensaje de controlador 64 (pedal de sustain).

También existen pedales HID (*human interface device*) que se conectan directamente a un puerto USB. Funcionan como un teclado de computadora, ya que mandan números del código ASCII –como el número 32, que equivale a la tecla “enter”– cada vez que se presiona el pedal (Thierauf, 2015, p. 17).

En la interpretación del segundo movimiento de *Ocaso*, se usa un pedal de expresión MIDI –también llamado pedal de volumen MIDI– para cambiar los parámetros internos de los módulos PasaBanda y Paneo. Dentro de [p PasaBanda] existe también un objeto [preset] para cambiar tres parámetros del filtro: la frecuencia central, el ancho de banda y la ganancia. Dependiendo de su posición, el pedal de expresión selecciona los presets 25, 26 y 27, que tienen almacenados las diferentes variables del filtro mostradas en la Tabla 5.

Tabla 5

Valores de los presets 25 a 27 usados en el módulo PasaBanda de Ocaso de Tenochtitlan

Preset	Frecuencia central	Ancho de banda	Ganancia
25	252	20	50
26	880	30	35
27	3000	180	30

A diferencia del pedal de sustain que sólo tiene dos posiciones, el pedal de expresión tiene 128 posiciones que abarca en todo su movimiento. Dividí todo ese rango en tres secciones: de 0 a 29, de 30 a 100 y de 101 a 127. Después asigné a la primera sección la selección del preset 25, la segunda sección al preset 26, y la tercera al 27. Por medio de [ctlin 7] –que significa

entrada de controlador número 7– se reciben los datos MIDI cuando el pedal se mueve y en el subpatch [p Pedal3Zonas] se asignan al preset respectivo.

En la parte central del mismo segundo movimiento, el pedal de expresión cambia de destino, ahora es [p Paneo], módulo en el que cambiará entre tres velocidades de movimiento del sonido de izquierda a derecha dentro del campo estéreo.

Automatización

Se puede programar en el tiempo la activación y desactivación de los módulos, o la modificación de los parámetros dentro de los mismos, de tal forma que sólo baste una señal de inicio para que se desencadenen automáticamente estas acciones en el tiempo exacto previamente definido en el programa. V.J. Manzo utiliza este método en su composición *Discourse* para clarinete procesado en tiempo real, obra que describe brevemente –en aspectos tecnológicos más que compositivos– en su libro *Max/MSP/Jitter for Music*, proporcionando la partitura, el patch utilizado y un archivo de audio con una versión MIDI del clarinete para que el lector pueda experimentar cómo el patch procesa el audio del instrumento. Manzo lo describe así:

Quando se presiona la barra espaciadora, un *clocker* dentro del patch comenzará a disparar los eventos en el patch del Max; esto es como la partitura para la computadora. Estos eventos asumen que, dado que el usuario ha presionado la barra espaciadora, el patch puede esperar recibir las notas de la partitura tocadas al tiempo para coincidir con los diferentes procesamientos de audio que tienen lugar dentro del patch. (2016, p. 393)

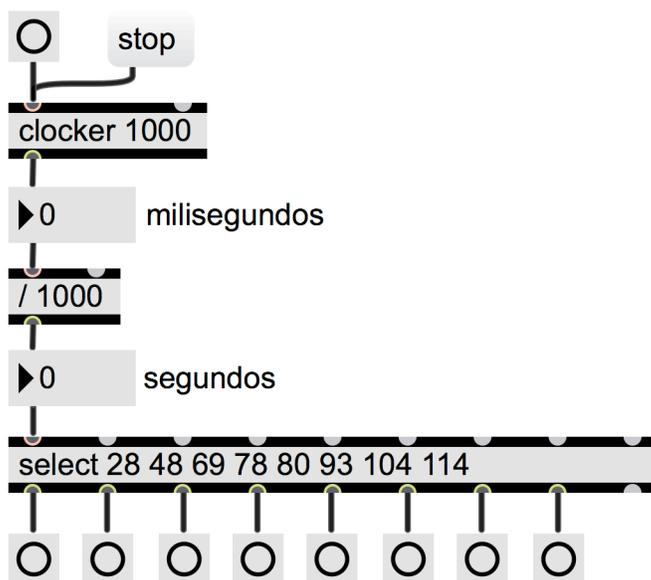
En la Figura 8, presento los objetos que llevan a cabo estos procesos automáticos en la obra de Manzo.

1. [button]: Dando clic a este botón se activa el [clocker] y pulsando el mensaje de “stop” se detiene.
2. [clocker]: es una especie de metrónomo que a lapsos iguales informa el tiempo transcurrido desde que se prende. El número que está dentro del objeto –en este caso, 1000–, es el tiempo en milisegundos entre cada reporte que envía de salida.
3. [number]: despliega el número que le llega de [clocker]; en este caso, cada mil milisegundos le llega un nuevo número.
4. [/]: divide el número que entra al objeto entre el número que tiene de argumento.
5. [number]: aquí despliega el valor de [clocker] en unidades de segundo.

6. [select]: si el número que recibe es igual a uno de los que tiene como argumento, mandará un bang por la salida correspondiente (hay una salida por cada número dentro del objeto).
7. Varios [button]: estos botones reciben una señal de ejecución (bang) en los tiempos exactos especificados en [select], que en este patch son 28 segundos, 48, 69, 78, etc.

Figura 8

Automatización de procesamientos en Discourse de V. J. Manzo



Discourse utiliza cinco módulos de procesamiento para toda la obra, los cuales recibirán la señal de los botones de [select] para prenderse, apagarse o cambiar sus parámetros –a veces a través de [preset]– en los tiempos programados por el autor. El intérprete sólo iniciará el [clocker] al pulsar la barra de espacio de la computadora, observará el reloj en el patch y cuando este marque cinco segundos comenzará a tocar; si el clarinetista sigue el tempo establecido los procesamientos coincidirán con la parte musical correcta –el compositor agregó en la partitura marcas de tiempo como guía para el intérprete–. En esta pieza no es necesaria una perfecta sincronización entre lo que toca el intérprete y los procesos, por eso es que puede funcionar muy bien.

Aunque esta sección se centra en el análisis de mis dos composiciones, me pareció importante examinar la pieza de Manzo porque esta fue la primera guía que tuve para empezar a

programar en Max/MSP mis ideas para una obra electroacústica –también de ahí mismo se deriva una de las ideas centrales para todo este trabajo– y porque representa un buen ejemplo de cómo una obra electrónica en vivo puede programarse con un patch completamente automatizado.

Ahora mostraré ejemplos de automatización en mis composiciones. En los compases 38 y 39 del primer movimiento de *Ocaso*, el intérprete presiona el pedal de sustain para activar un delay que sólo va a actuar en una nota para después apagarse de inmediato. En la línea inferior de la partitura –mostrada en la Figura 9–, se indica exactamente el punto en el que debe presionarse el pedal correspondiente. La desactivación del módulo de delay se lleva a cabo automáticamente para ahorrarle al percusionista la tarea de apagarlo con otro pedal y permitirle concentrarse en la parte musical que aquí es ininterrumpida.

Figura 9

Compases 38 a 39 de Presagios, primer movimiento de Ocaso

The image shows a musical score for two measures, 38 and 39. The upper staff is a piano part. Measure 38 begins with a triplet of eighth notes, followed by a quarter note with an accent (>) and a dynamic marking of *f*. Measure 39 starts with a triplet of eighth notes, followed by a quarter note with an accent (>) and a dynamic marking of *mf*. The final part of measure 39 consists of a quarter note with an accent (>) and a dynamic marking of *f*, followed by a quarter rest with a dynamic marking of *m*. The lower staff is a pedal line. It shows a pedal symbol (a rectangle) that is depressed at the start of measure 38. A circled number '8' is placed above the first quarter note of measure 38, with a line connecting it to the pedal symbol. A circled number '9' is placed above the first quarter note of measure 39, with a line connecting it to the pedal symbol. The word 'Delay' is written between these two circled numbers. The pedal symbol is released at the end of measure 39, indicated by a circled number '1'.

Esta desactivación automática se logra usando la misma señal que abre el módulo [p SeguidorEnv], la que lo cierra mandando ese mismo bang al [toggle] un segundo después –hay que recordar que [toggle] oscila entre prendido y apagado con cada bang que recibe–. Este procedimiento de cerrado automático del módulo también se utiliza en los primeros tres procesos del primer movimiento y en el último pedal (número 11) del mismo. El objeto [delay] es el que retiene por cierta cantidad de tiempo el bang que le llega, antes de enviarlo al siguiente objeto que se conecta (no hay que confundir este objeto con el procesamiento de audio que Max/MSP realiza con el llamado [delay~], donde el símbolo “~” que se agrega, es el indicador de que este objeto efectúa procesamiento de señal). El mismo [delay] se utiliza en el compás 40 de la pieza para retrasar por diez segundos la entrada del *feedback* del flanger.

Los citados objetos de [clocker] y [delay] permiten programar sucesiones temporales de acciones, al igual que los objetos [pipe] –que retrasa el envío de números y símbolos, aparte de bangs–, [timer] –que reporta el tiempo transcurrido entre dos eventos– y [metro] –que manda bangs a intervalos regulares de tiempo–. Precisamente, en el inicio del tercer movimiento de *Ocaso* se utiliza [metro] para cambiar cada 1120 milisegundos la frecuencia de corte de los filtros en el módulo [p Armon] y más adelante para disparar dos archivos de audio de sonidos percusivos –dentro del módulo [p Samp]– para crear una base métrica sobre la que la tarola va a enunciar diversas frases.

El fade in y el fade out son también formas de automatización, usados para la entrada y salida gradual de cierto procesamiento. Ejemplo de lo anterior se tiene en el primer movimiento de *Ocaso*, donde el pedal 4 acciona la apertura del filtro de forma progresiva en el transcurso de 12 segundos –efecto análogo al del pedal de volumen que mueve un organista para obtener un crescendo en un Hammond–. La programación de estos cambios se logra mediante el uso del objeto [line]:

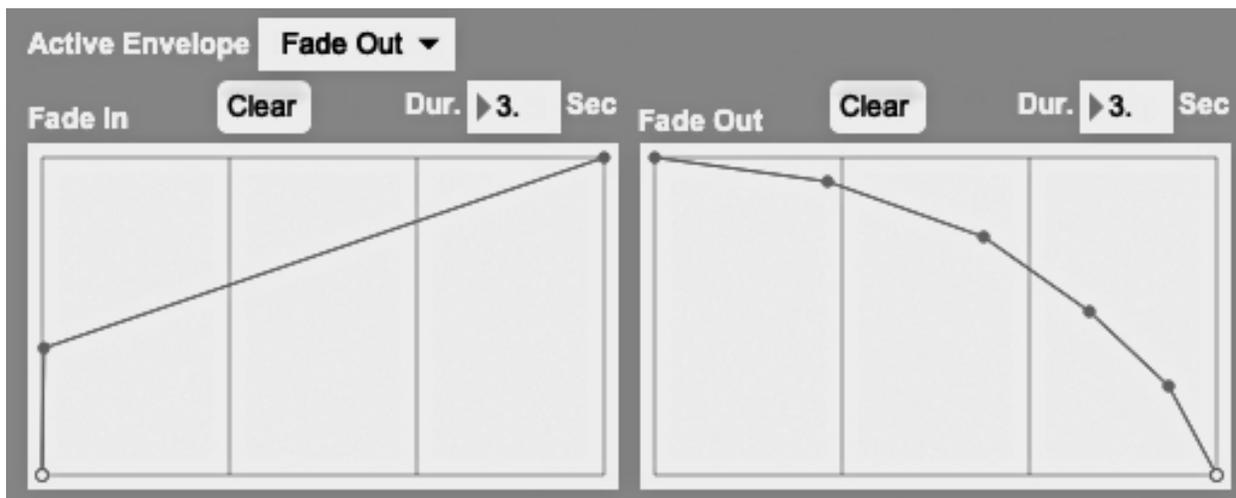
El objeto *line* genera una rampa de valores entre el valor actual y un valor de destino, en un tiempo determinado. Por ejemplo, si enviamos el mensaje con los valores “1 2000”, el objeto generará una serie creciente de números entre 0 (valor actual) y 1 (valor de destino), en 2000 milisegundos (tiempo para realizarlo). Y si luego enviamos la lista “0 3000”, decrecerá de 1 a 0 en 3 segundos. (Cetta, 2014, p. 29)

Con respecto a *Subliminal*, en las primeras versiones que utilicé del programa *Live Patch*, los procesadores se prendían o apagaban de inmediato produciendo saltos abruptos en la intensidad, por lo que se le pidió al programador una forma de enfrentar ese inconveniente, problema que resolvió a través de la incorporación de curvas de fade in y fade out dentro de los módulos, como las que se observan en la Figura 10.

Algunos ejemplos en *Subliminal* donde uso estos recursos son: el fade in con que entra el Harmonizer en el pedal 2, el incremento gradual del parámetro “Hi bin cutoff” del DroneMaker (pedal 7) y el fade out al apagarse el mismo con el pedal 11.

Figura 10

Implementación de fade in y fade out en Live Patch



Detectores de dinámicas, alturas y motivos musicales

Triggering gate, literalmente “disparador de compuerta”, es un algoritmo que dispara la ejecución de cierta acción o proceso cuando cierto nivel de amplitud es excedido. En *Ocaso* es utilizado en el último movimiento para modificar –dentro del módulo 5TET– la frecuencia central de un filtro aplicado a un generador de ruido blanco, a través de un pasaje donde la tarola alterna muchas notas en pianísimo con esporádicos acentos, los que provocarán el cambio en la afinación del ruido.

Otro mecanismo que emplea las dinámicas del instrumento interpretado para detonar acciones es el seguidor de envolvente (envelope follower), del que ya se habló brevemente en el listado de los módulos usados en *Ocaso* –por cierto, el procesamiento más utilizado en dicha obra.

El seguidor de envolvente (a veces llamado seguidor de amplitud pico o detector de envolvente) realiza la función de extraer la envolvente de un sonido por medio de la medición de la amplitud de los picos de su forma de onda. El seguidor de envolvente produce una señal de control basada en una serie de valores de amplitud extraídos de un determinado sonido. (Cipriani y Giri, 2020, p. 329)

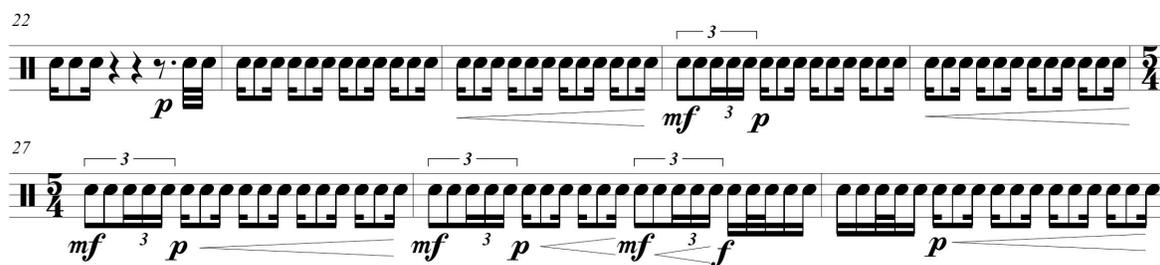
El módulo [p SeguidorEnv] utiliza el perfil extraído del audio entrante para determinar el tiempo de retraso en el delay incorporado y en la frecuencia de corte del filtro también incluido, con lo que la intensidad del sonido de la tarola puede crear un cambio tímbrico de más brillante –

cuando el sonido está en su pico de amplitud– a más oscuro –al disminuir la amplitud hacia el final de la nota– siguiendo el perfil de la envolvente extraída. En el primer movimiento de *Ocaso*, el proceso de seguidor de envolvente se utiliza siete veces, primordialmente para controlar el delay pero en el compás 17 también como control del filtro; y en el tercer movimiento se utiliza en tres ocasiones para modificar de forma más radical el audio, controlando la envolvente al delay y al filtro simultáneamente.

Una manera más de utilizar las dinámicas del audio entrante, se muestra en los compases 22 a 35 del primer movimiento de *Ocaso*, donde el rango dinámico de la tarola cambia de piano a forte en breves intervalos de tiempo, como se muestra en la Figura 11.

Figura 11

Cambios de dinámica en los compases 22 a 29 de Presagios, primer movimiento de Ocaso

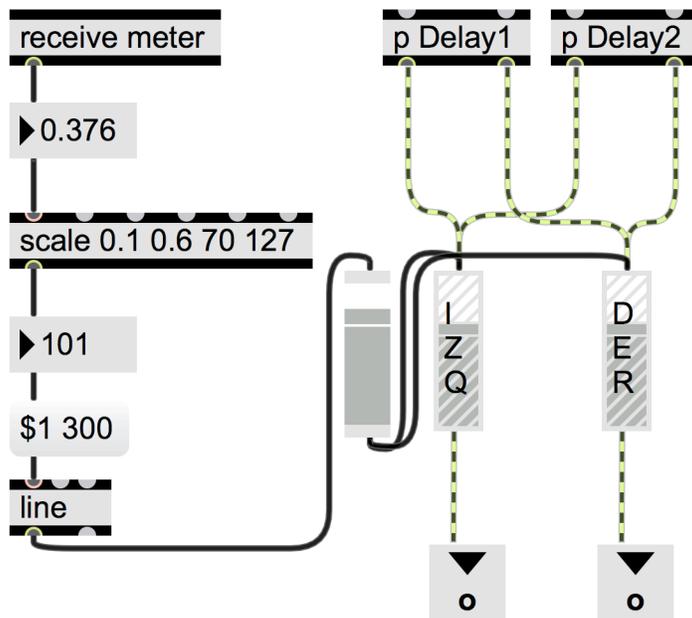


El audio del instrumento es procesado con un flanger estéreo que permanece durante toda la sección, pero al aumentar la intensidad de la tarola sube el volumen del módulo [p Delays], el que incorpora también un filtro pasa banda que refuerza las frecuencias alrededor de 11 Khz, resultando un sonido similar a unas maracas que sólo es audible en el rango de dinámicas de mezzoforte a forte.

En la pantalla principal del patch de *Ocaso* existe un monitor del nivel de entrada de audio llamado [meter] – con LEDs de diferentes colores que se iluminan según la intensidad de la señal entrante–; de la salida de dicho objeto se obtiene el valor máximo de la señal recibida en cierto intervalo de tiempo –en el caso de mi patch, 300 milisegundos–, valor que se utiliza para determinar el volumen de salida del módulo [p Delays] (ver Figura 12).

Figura 12

Volumen del módulo controlado por el nivel de audio de entrada



1. [receive meter]: recibe por medio de una conexión remota el valor de nivel de audio de [meter] y lo despliega en el siguiente [number].
2. [p Delay 1] y [p Delay 2]: son los subpatches donde se realiza el procesamiento; están conectados a dos objetos [gain~] por donde sale el audio del módulo.
3. [scale]: convierte cierto rango numérico de entrada –aquí, los valores de 0.1 a 0.6– a un rango diferente de salida –en este patch, de 70 a 127–.
4. [number]: aquí se visualiza la conversión realizada por [scale]; en este ejemplo, un nivel de 0.376 produce un número de salida de 101, valor que manda al siguiente mensaje.
5. [line]: genera una serie de números entre el valor actual y el que le llega en el mensaje de entrada (mensaje que contendrá, en este caso, los números 101 y 300) en un lapso de 300 milisegundos.
6. [slider]: recibe los números de [line] y mueve el valor de los dos objetos [gain~] para modificar el nivel de salida de audio del módulo.

Hay otros mecanismos de detonación de acciones que, aunque no fueron utilizados en ninguna de mis dos obras, es importante anotarlas y describirlas para tener un panorama completo de las posibilidades en la programación y creación de una obra electrónica en vivo.

La detección de alturas (*pitch detector* o *pitch tracking*) consiste en hacer un análisis del espectro del sonido para detectar la fundamental en un conjunto de frecuencias que forman un timbre. Comúnmente como resultado del análisis se obtiene, aparte de la frecuencia en Hertz, un número de nota MIDI –que se puede usar para disparar alguna acción–. Sin embargo, detectar la altura de un sonido que entra por la interfaz de audio no es una tarea fácil de ejecutar:

Se debe realizar un proceso conocido como transformada rápida de Fourier (FFT) en la señal de audio para descomponer la señal en frecuencias discretas. A partir de cada una de las frecuencias presentes, se tendría que identificar la frecuencia fundamental del sonido complejo y convertirla en una nota MIDI. (Manzo, 2016, p. 282)

Como es de esperarse, los resultados pueden variar dependiendo del objeto utilizado para tal propósito, así como del instrumento al que se le aplica el análisis; también es diferente que el software reconozca una nota aislada –rodeada de silencio– a que se pueda identificar una serie de sonidos que formen un motivo melódico.

Por otro lado, que el software reconozca una sucesión de alturas en un instrumento electrónico MIDI, es una tarea sencilla, muy rápida y que requiere un mínimo de cálculos de la computadora. Estos instrumentos no mandan sonido por su puerto de salida (MIDI Out) sino información de los mecanismos que se accionaron para producir tal sonido, esto es, qué teclas se activaron –en el caso de un instrumento MIDI de teclado–, con qué velocidad se pulsaron y por cuánto tiempo, el movimiento de pedales, ruedas de afinación y de modulación, etc., produciendo así una representación en números de la interpretación realizada en un piano electrónico, un controlador de aliento (*wind controller*), una guitarra MIDI, una batería electrónica o cualquier otro instrumento que pueda manejar este protocolo –o lenguaje– MIDI. De tal forma, si la computadora recibe del instrumento una serie de mensajes *note-on* acompañados de los números 60, 62, 64, 65, 67, 69, 71 y 72, podrá comparar esta serie de números con patrones numéricos previamente almacenados para identificarlos como una estructura de escala mayor con su tónica en la nota do.

Es así como la computadora puede reconocer diversas características de una interpretación musical llevada a cabo en un instrumento MIDI: alturas, dinámicas, duraciones, y

con un poco de programación se pueden reconocer intervalos (tamaño y dirección de estos), registro, contornos melódicos, ritmos, etc. Todo esto encuentra una aplicación en lo que se ha llamado *score following* donde una partitura que ha sido codificada a través de mensajes MIDI y almacenada en memoria puede ser comparada con lo que se está interpretando en un instrumento MIDI para poder determinar el tempo de la ejecución y sincronizarlo, por ejemplo, con una pista de acompañamiento electrónica que pueda seguir los ritardando, accellerando o pausas que realice el instrumentista. En el esencial libro de Winkler (1998) sobre la composición de música interactiva usando Max, se puede consultar al respecto.

3. Modelo de programación

Con el análisis realizado y la descripción de las partes más importantes que hacen funcionar los patches de las dos obras, se llega al siguiente modelo de programación propuesto para una obra electrónica en vivo. En el primer acercamiento a este modelo, se considera al programa –y a la computadora que lo alberga– como una caja negra, la que recibe el audio del micrófono para después ser dirigido a la salida de audio (ver Figura 13).

Figura 13

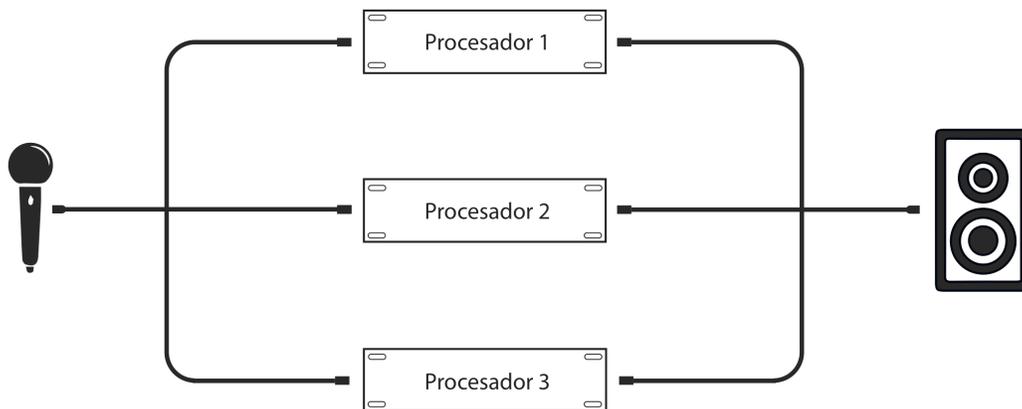
Entrada y salida de audio de la computadora



La modificación del sonido ocurre entre la entrada y la salida, para lo que habrá algunos procesadores de audio que recibirán la señal entrante, la transformarán y la mandarían a la salida, como se representa en la Figura 14. Cada caja –o módulo, como se le denomina en el análisis– puede contener un procesador de audio individual, pero también puede ser una combinación de ellos –como un delay cuya señal vaya a un filtro y después a la salida del módulo.

Figura 14

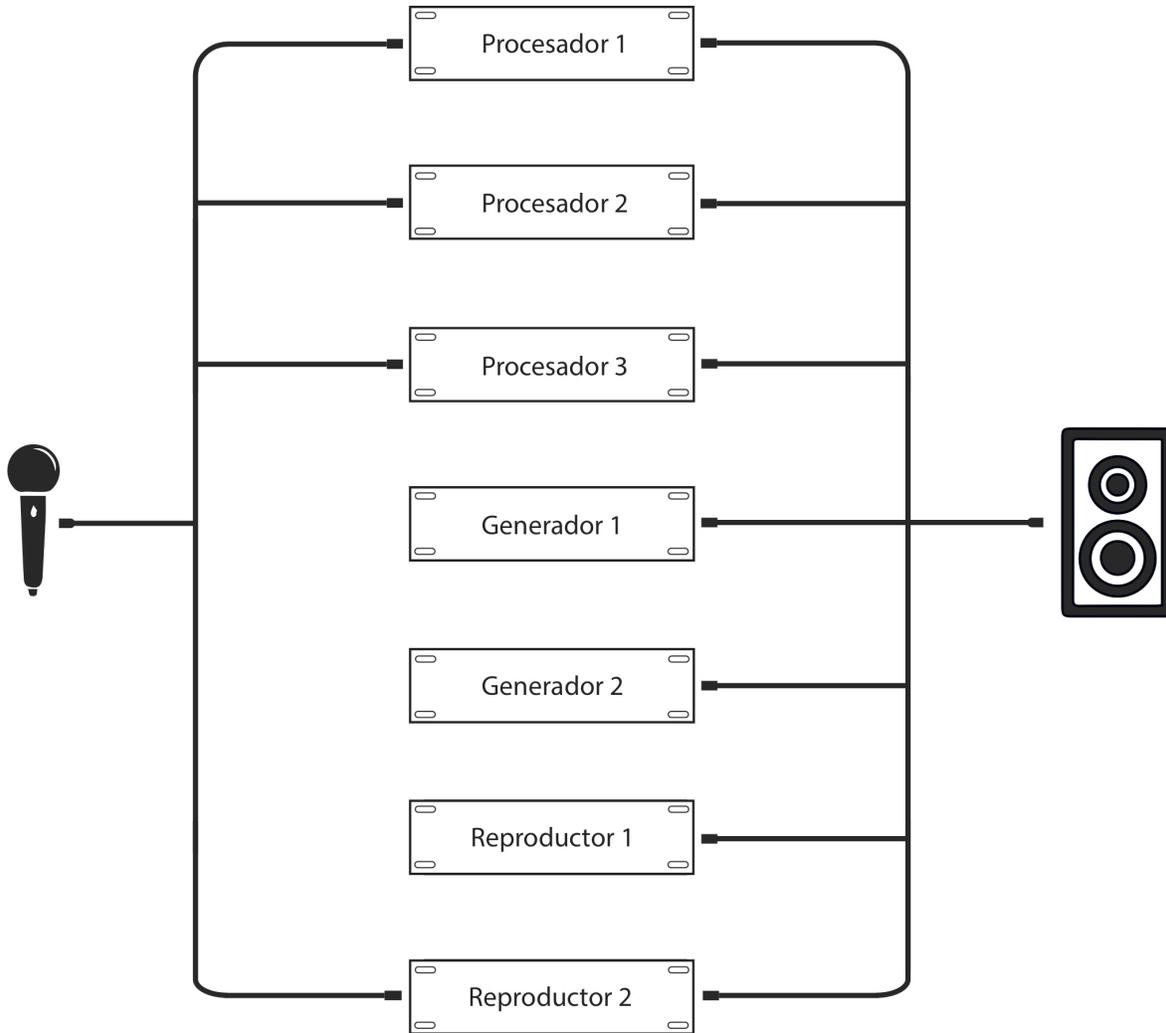
Procesadores de audio insertados entre la entrada y la salida



Como se presentó en la sección sobre los tipos de módulos incluidos en los programas, además de los procesadores existen generadores y reproductores de sonido, los que son añadidos en el esquema de la Figura 15.

Figura 15

Los tres tipos de módulos dentro de un programa para una obra electrónica en vivo



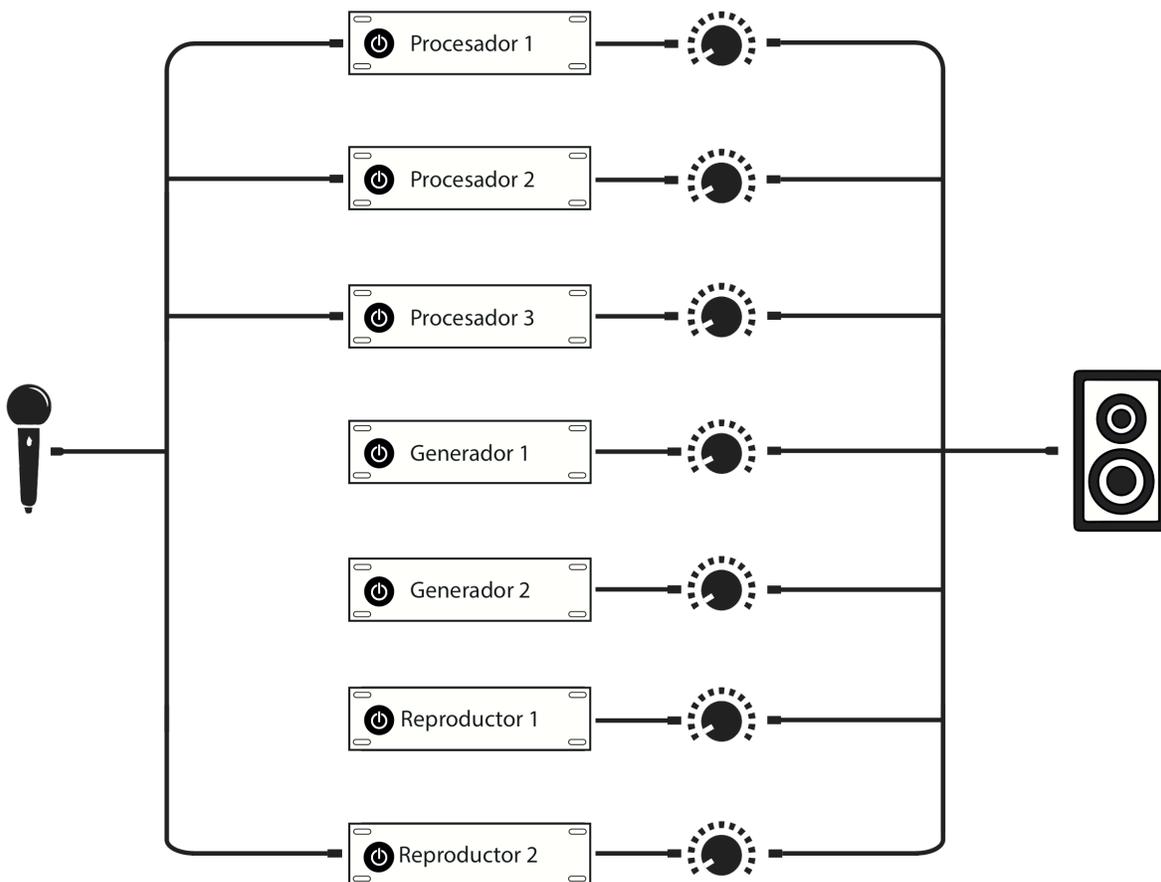
Aquí puede observarse que los módulos generadores no tienen entrada de audio, sólo salida, ya que producen el sonido a partir de las herramientas propias del software como osciladores con diferentes formas de onda o generadores de ruido. En este ejemplo, el reproductor 1 se utiliza para sonidos grabados y editados previamente a la ejecución de la obra –

almacenados como archivos de audio en la computadora— y el reproductor 2 se usa para grabar el sonido del instrumento en el momento que se interpreta la obra para ser reproducido posteriormente –por eso es que hay un cable de la entrada de audio de la computadora a dicho módulo.

Tal como se presenta la Figura 15, pudiera sugerir que todos los módulos mandan su señal al mismo tiempo a la salida general de audio, lo que sería el caso menos probable en una obra electrónica en vivo. Por lo tanto, es necesario añadir en cada módulo un interruptor de encendido y apagado para que entren en operación sólo los que se necesitan en cada sección de la obra; de igual importancia es la regulación del volumen del audio que sale de cada módulo, para lo que se agregan las perillas respectivas (ver Figura 16).

Figura 16

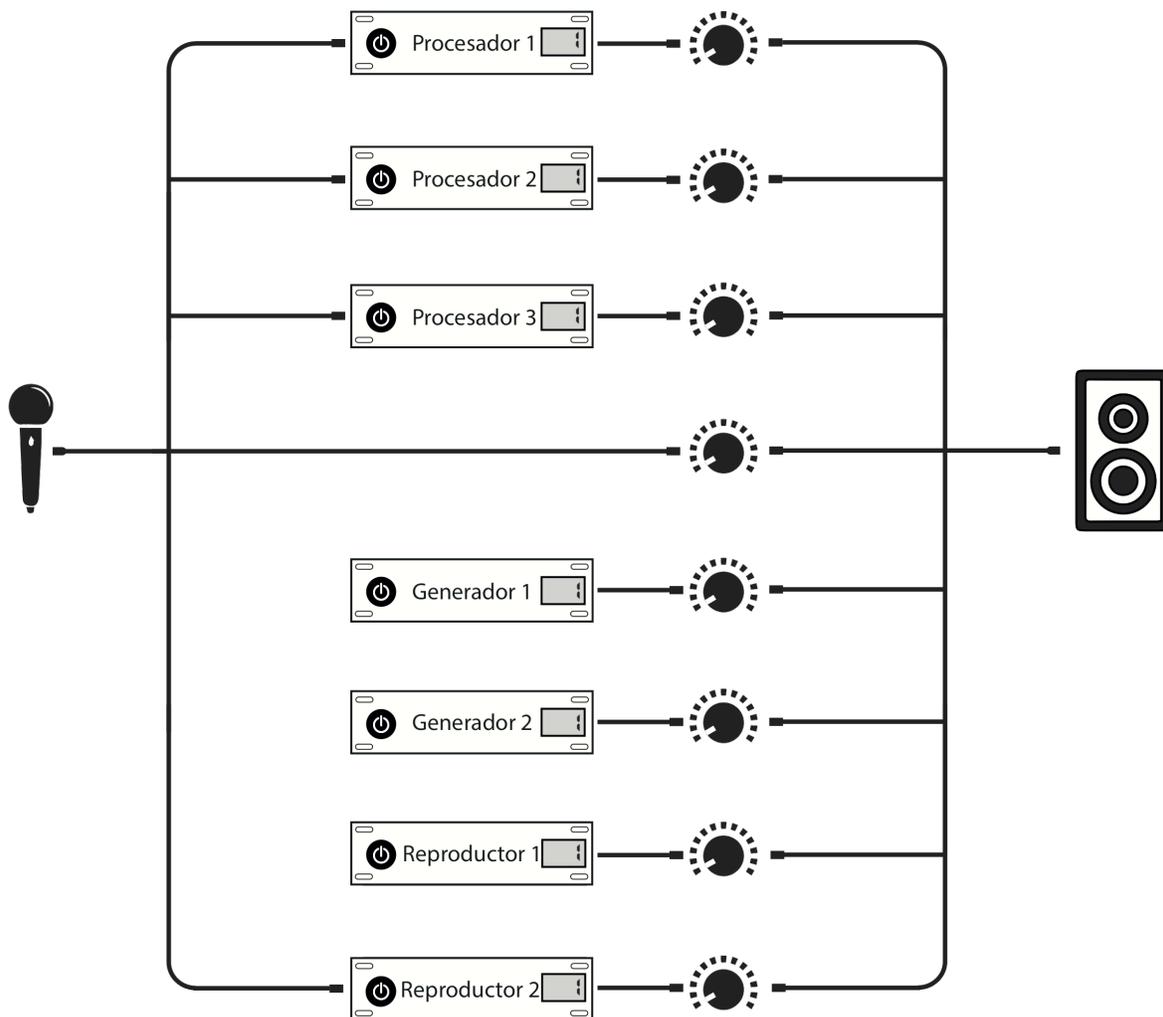
Controles de encendido y apagado y regulación de volumen en los módulos



En la Figura 17, se ha añadido una salida directa del audio sin procesar –con una perilla de volumen para el mismo–: en *Subliminal* se utilizó para la señal limpia del violonchelo eléctrico y en *Ocaso* para obtener una combinación del audio directo de la tarola con el procesado. También puede observarse que los módulos tienen ahora una pantalla con un número que representa el preset seleccionado en cada uno de ellos. El preset nos permite obtener diversos resultados sonoros de un mismo módulo al combinar de diferentes formas los parámetros internos del mismo: en un flanger, por ejemplo, el tiempo mínimo de retraso, el rango de variación del mismo, la velocidad, entre varios más.

Figura 17

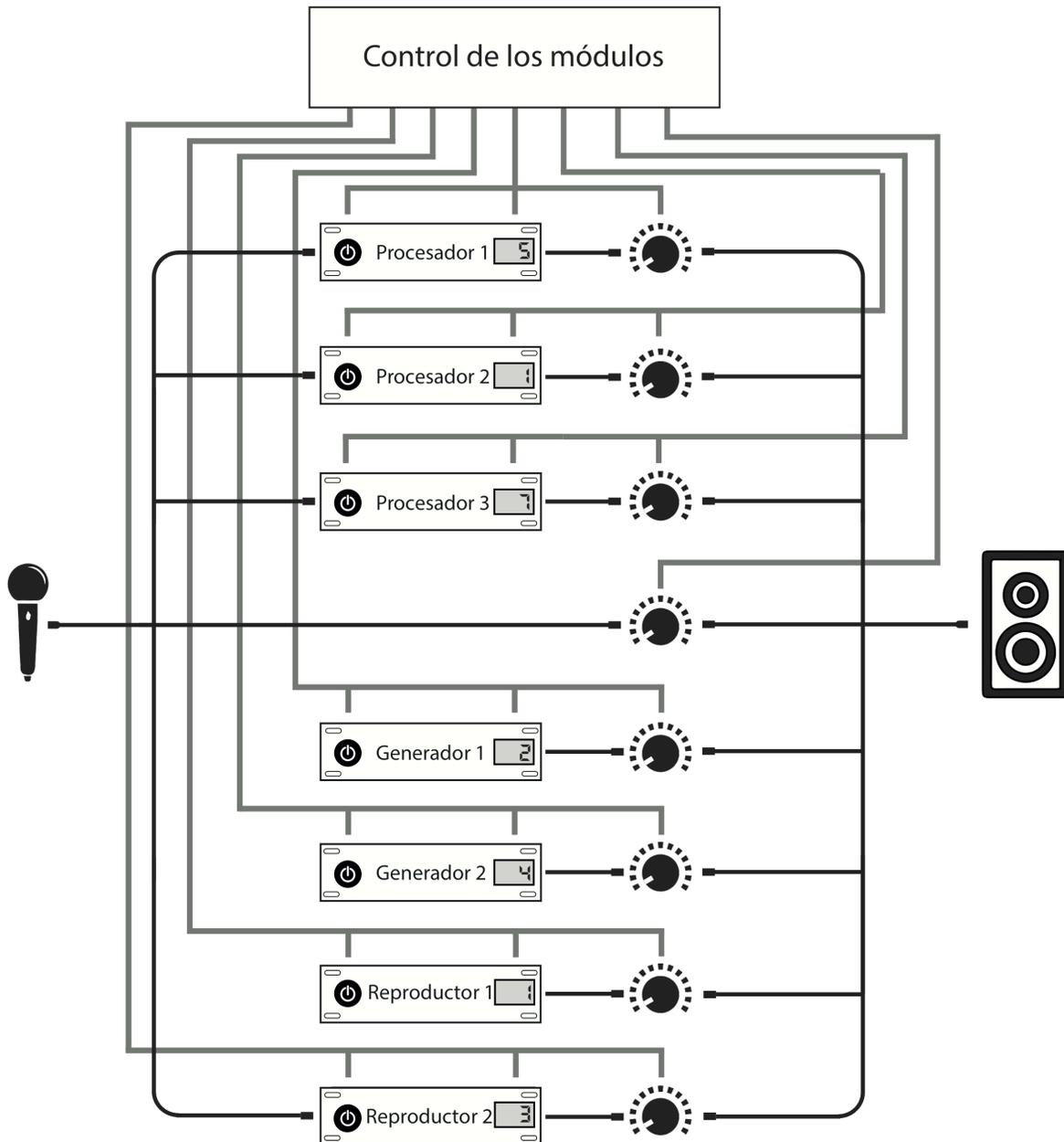
Adición de la salida directa del audio sin procesar y de la pantalla de preset en los módulos



La activación y desactivación de los módulos, las combinaciones de sus parámetros internos grabadas en los presets y sus niveles de salida son los aspectos a controlar durante la interpretación de la obra con la sección de “Control de los módulos” (Figura 18).

Figura 18

Sección de activación, desactivación y modificación de los módulos



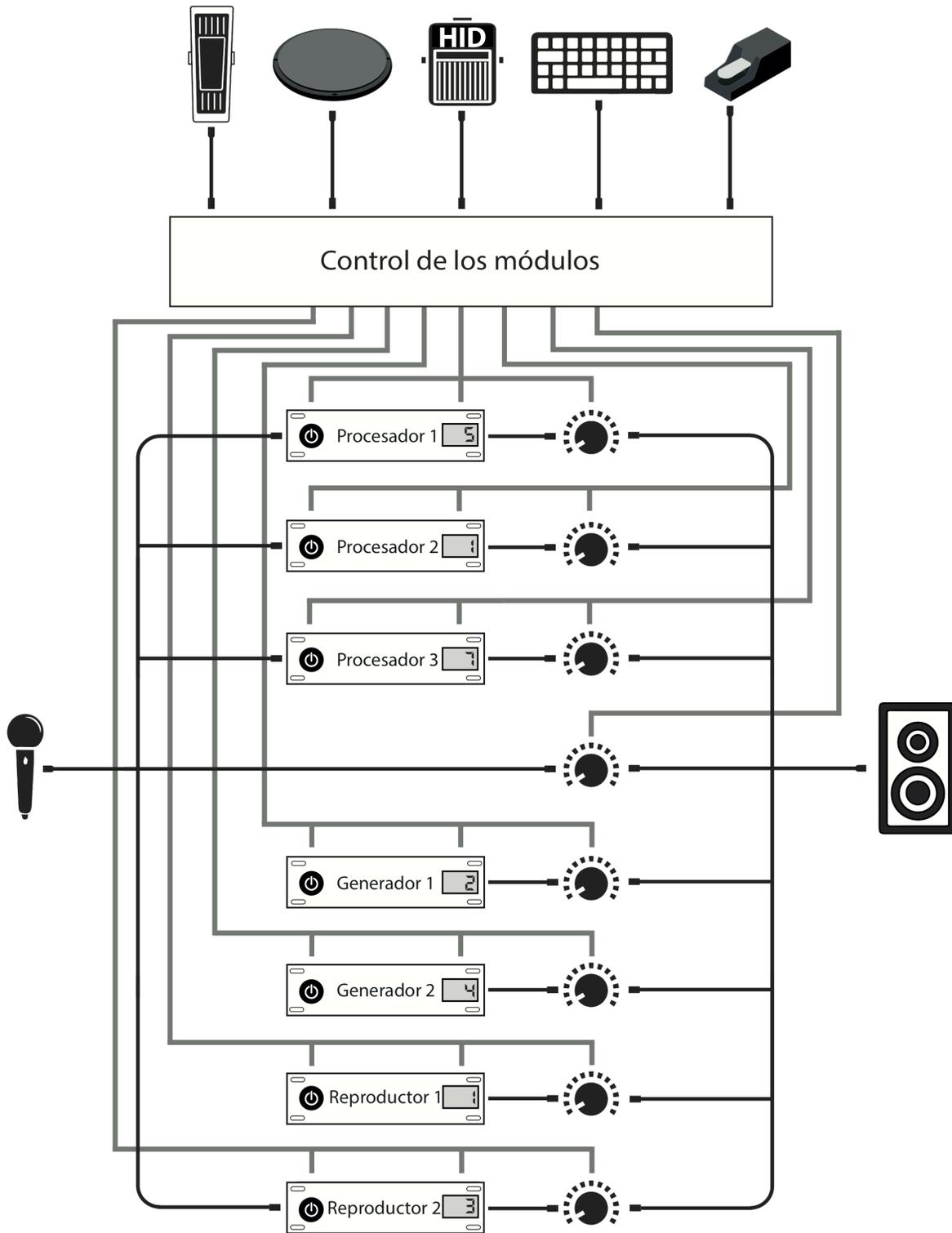
El control de los módulos dentro del programa puede ser llevado a cabo de diversas maneras –como se explicó en el capítulo anterior–. Un músico asistente puede prender y apagar los módulos y cambiar los volúmenes y los presets, ya sea con un ratón –dando clic directamente sobre los dispositivos visualizados en la pantalla–, con el teclado de la computadora –con la asignación de teclas a botones y perillas de control– o utilizando algún instrumento controlador MIDI, donde los potenciómetros, sliders e interruptores físicos del controlador pueden ser asignados a sus equivalentes virtuales dentro del programa. De cualquier forma, el programa debe de recibir una señal de entrada a través de diversos dispositivos, los que están representados en la Figura 19 con una conexión directa –que en este caso no es de audio– a la caja de “Control de los módulos”.

Enseguida se describen los dispositivos representados en la figura, en un orden de izquierda a derecha. El pedal de expresión en su movimiento de arriba a abajo envía una serie de números MIDI que puede utilizarse para realizar transiciones graduales en los parámetros de un módulo –tal como lo hace un potenciómetro–. El pad de batería electrónica se colocó en el esquema para representar a cualquier instrumento MIDI que al mandar un mensaje de nota controle o modifique algún aspecto de los módulos. Asimismo, se puede realizar un seguimiento de las notas MIDI entrantes para detectar patrones de alturas o rítmicos que desencadenen cambios en los módulos. Los pedales HID (*human interface device*) son una buena alternativa al pedal de sustain, ya que no requieren interfaz MIDI porque funcionan enviando un número de tecla –exactamente igual que el periférico discutido a continuación–. El teclado de la computadora manda números de tecla para dirigir la sección de control de módulos, y aunque es evidente que este dispositivo no lo podría manipular el intérprete de la pieza electrónica –excepto, al iniciar la pieza o en alguna pausa grande dentro de la misma–, un asistente podría hacerlo fácilmente. Para finalizar, el pedal de sustain, es otro dispositivo MIDI para controlar los módulos, cuyo uso se ha ejemplificado ampliamente en el análisis de las dos obras electrónicas.

Las señales de estos cinco dispositivos entran a la caja de control de los módulos donde se ubican mecanismos como el conteo de las pulsaciones del pedal (descrito en la Figura 5), o el que se utiliza para prender y apagar los módulos en un orden que responde al número de pulsación del pedal de sustain (detallado en las Figuras 6 y 7).

Figura 19

Modelo de programación para una obra electrónica en vivo



En cuanto a procesos de automatización, es en la misma sección de control de módulos donde se ubicaría un reloj o cronómetro, el que iniciaría su marcación al momento de recibir una señal de algunos de los cinco dispositivos mostrados. Otras automatizaciones, como el fade in, fade out, y el apagado automático del módulo después de haber transcurrido cierto tiempo –como se utiliza en el inicio de *Ocaso*– pueden estar programados dentro del mismo módulo.

Los mecanismos de detección de dinámicas sobre el audio de entrada fueron ubicados, en el ejemplo de la pieza *Ocaso*, dentro del mismo módulo sobre el que actuaban, aunque también podría ubicarse dentro de la caja de control de módulos si se agrega una línea de entrada de audio a esta sección –la cual no existe en los esquemas mostrados–. Lo mismo se aplica a la detección de alturas en la señal de audio.

Por último, la sección de control de los módulos, al recibir mensajes MIDI de nota, puede contener dentro de la misma el patrón de notas, intervalos o ritmos que al ser reconocidos en la información entrante detonen diversas acciones sobre los módulos.

4. Software, hardware y equipo de audio utilizado

4.1 Software

Si bien, el software principal usado para la realización de estas obras es Max/MSP, hay un conjunto de programas que han facilitado y acelerado la creación de las mismas. A continuación, se refieren estos programas auxiliares por categorías y no por el nombre específico del programa, porque no es realmente importante si se utilizó, por ejemplo, *Pro Tools*, *Logic* o *Cubase* sino el papel que jugó un programa de esta clase –en este ejemplo, de grabación y edición digital de audio– en este proceso.

Notación musical

Además del uso de este tipo de programa para hacer las partituras de las obras y las figuras musicales de este trabajo, como herramienta de composición la he usado siempre para probar ideas rítmicas y de arreglo y para revisar las estructuras de mis piezas. Un uso no menos importante fue el de realizar las versiones MIDI de las partes de violonchelo y de tarola, archivos que utilicé para alimentar el audio que entra a los patches de Max/MSP en lugar de la tarola microfoneada o del audio directo del violonchelo eléctrico –ya que no tenía ninguno de los dos instrumentos a mi alcance– y hacer pruebas con los procesamientos sonoros que se estaban programando. Fue importante para estas versiones MIDI que el programa elegido pudiera crear una buena simulación de los sonidos de pizzicato y glissandi del violonchelo y de los *rolls* y *flams* de la tarola.

Secuenciador, audio digital e instrumentos virtuales

El programa que integra estas tres funciones se le llama actualmente estación de trabajo de audio digital (DAW) y en varias de ellas se incluye también la opción de edición de partituras, aunque no es tan profesional como en un programa especializado en ello. La secuenciación de las pistas electrónicas de *Subliminal* se realizó aquí, a través de un archivo MIDI que importé de la partitura creada en el programa de notación musical. Una vez en el secuenciador, la información MIDI se modificó para cambiar ligeramente la duración de las notas y su posición en el tiempo, con el objetivo de darle cierta flexibilidad rítmica que evitara una interpretación muy exacta que pudiera sonar mecánica. La información de los *tracks* MIDI se envió a un sintetizador Motif-XS de Yamaha, con excepción de dos tracks que utilizaron uno de los instrumentos virtuales de la

DAW –el sintetizador Vacuum, de Air Music Technology– y un track que usó muestras de pizzicato de un ensamble de cuerdas del sampler Kontakt.

Se convirtió todo a audio digital –reproduciendo cada uno de los tracks MIDI de forma individual y grabándolos en la misma DAW– y se realizó una mezcla que después se vació en dos archivos estéreo de audio –para la pista inicial y la final de la obra–, que son los que dispara el patch con su módulo Player.

El programa Audacity, también de audio digital, fue usado cuando se requería editar un fragmento de la versión MIDI de la tarola o del violonchelo para probarlo con los procesadores que estaban desarrollándose en Max/MSP e ir afinando detalles en los mismos y para experimentar con diferentes combinaciones de sus parámetros.

Editor de programas

Programa es una palabra que también se utiliza para designar a cada uno de los timbres de un sintetizador, aunque también se le llama patch –la misma palabra con que denominamos a una combinación de objetos que se conectan en Max/MSP–, preset, *voice* o, simplemente, timbre. Para editar los sonidos del sintetizador Motif-XS se hizo un uso extensivo del editor de programas del mismo, no sólo porque es más fácil visualizar en la pantalla de la computadora las diferentes perillas, interruptores y botones del instrumento sino también porque hay algunos parámetros a los que no se puede acceder a su modificación con los botones de operación del Motif-XS, únicamente con el software de edición.

Entrenamiento auditivo

Los programas referidos aquí no son los de la práctica auditiva para músicos –con dictado de intervalos, acordes y escalas– sino los orientados a los ingenieros de audio, en los que se intenta entrenar el oído en la diferenciación de bandas de frecuencia que son acentuadas o atenuadas, en la cantidad de reverberación que tiene un sonido, el tiempo de retardo de un delay, la detección de la distorsión digital, el nivel de compresión del audio, entre otros aspectos importantes del sonido. Aunque no se utilizaron este tipo de programas directamente para la creación de las obras, me parece de importancia el trabajo de entrenamiento que llevé a cabo durante algunos meses con ellos –previos al trabajo de creación de mis piezas–. Si bien, no fue una práctica exhaustiva la que realicé, creo que me sensibilizó y abrió mis oídos a percibir detalles y sutilezas que nunca detectaba en las grabaciones musicales, por lo que recomiendo sobremanera esta preparación a quien quiera adentrarse en la composición electroacústica.

Dentro de esta categoría, sí daré referencia directa de estos programas porque no son muy comunes, ni conocidos. El principal programa con el que practiqué es el incluido en el disco compacto que acompaña al libro *Audio Production and Critical Listening* de Jason Corey (2010), software de entrenamiento creado, precisamente, con Max/MSP y que es ofrecido en versiones para Macintosh y Windows. Una versión actualizada de ese software puede encontrarse en la página web del libro, el cual está disponible para cualquier usuario.

También es recomendable el paquete de libro y disco compacto de *Critical Listening Skills for Audio Professionals* de F. Alton Everest y los programas en línea *Pro Audio Essentials* de Izotope y *SoundGym*.

4.2 Hardware y equipo de audio

Interfaz de audio

Este es el dispositivo a donde llega la señal eléctrica, ya sea de un micrófono –en el caso de la tarola– o de un cable de *plug* de un cuarto de pulgada –en el caso del violonchelo eléctrico– para ser digitalizado; también de aquí saldrá el sonido procesado de la computadora para ir a los audífonos o al sistema de amplificación.

Interfaz MIDI

Es necesaria para recibir las señales de los pedales de sustain y de expresión de un instrumento electrónico MIDI a donde estos están conectados. A veces viene incluida en la interfaz de audio.

Teclado electrónico MIDI

Se utiliza como un puente para mandar la información de los pedales de sustain y de expresión a la computadora. Si tiene puerto MIDI-Out, de aquí parte un cable a la interfaz MIDI; si manda y recibe datos MIDI por un cable USB, la información va directo a la computadora y se evita el uso de la interfaz. Hay algunos teclados que no tienen entrada para pedales o sólo para el de sustain.

SPY de Midiplus

Esta es una interfaz a la que se pueden conectar directamente un pedal de sustain y uno de expresión para mandar sus datos MIDI por el cable USB directamente a la computadora, sin la intermediación de un teclado electrónico o interfaz MIDI, por lo que su uso evita la necesidad de los dos dispositivos anteriores.

Pedal de sustain

También es llamado *switch pedal*. En las dos obras presentadas, es el componente principal que acciona los diferentes procesadores, generadores y reproductores de sonido. Algunos funcionan mandando un número 127 cuando se presionan y un cero cuando se sueltan, pero algunos trabajan de manera inversa –mandan un cero al presionarlos y 127 al soltarlos–, por lo que los dos patches tienen una manera de cambiar su forma de operación dependiendo del pedal utilizado. Esta característica de los pedales se llama “polaridad” y existen los que incorporan un interruptor para cambiar entre los dos tipos de funcionamiento.

Pedal de expresión

Conocido también como pedal de volumen, lo utilizo en el segundo movimiento de *Ocaso* para cambiar entre tres configuraciones de procesamiento y después entre tres velocidades de paneo.

Monitor de computadora

Es un monitor auxiliar que se utilizó en los conciertos donde se presentó *Subliminal*, para mostrar al intérprete de forma cercana la misma pantalla que presenta la computadora, ya que en ella se muestran los números de pedal cada vez que este se presiona, dando así una retroalimentación visual al músico de que el patch está recibiendo las señales del pedal y de que el flujo de las acciones en el patch está ocurriendo adecuadamente.

Micrófonos

En *Ocaso* se usan dos micrófonos para la tarola: uno dinámico arriba de ella para recibir el sonido del parche y uno de condensador dirigido hacia el entorchado para captar las frecuencias agudas del mismo.

Audífonos

Utilizados para que el intérprete de tarola reciba la mezcla sonora del sonido directo y procesado del instrumento, esto es, se utilizan para monitorear en lugar de los monitores de piso, ya que estos podrían ocasionar *feedback* al interactuar con los micrófonos. Es importante que estos audífonos aislen lo más posible del sonido externo, rodeando completamente la oreja del músico.

Monitores de piso

Se utilizaron en los conciertos de la obra *Subliminal* y son fundamentales para la escucha, por parte del intérprete, de los sonidos electrónicos de la obra y del violonchelo eléctrico.

5. El intérprete en la música electrónica en vivo

5.1 Colaboración entre compositor e intérprete

En su guía de composición para percusiones, Samuel Z. Solomon escribe: “Los compositores que están en comunicación con sus intérpretes casi siempre escribirán una música mucho mejor, más eficaz y más interpretada que los que no lo están” (2016, p. XV). Por supuesto, esta colaboración entre compositor e intérprete es recomendable en la composición para cualquier dotación instrumental, pero es particularmente importante cuando se intentan usos no convencionales del instrumento o recursos experimentales que intentan expandir los límites de lo probado –lo que suele ocurrir en las obras de música electrónica en vivo.

Aunque el intérprete es el destinatario último que hará realidad la pieza electrónica en vivo, su presencia se manifiesta durante todo el desarrollo de la obra y de la programación del software que permitirá la interpretación. Ejemplificaré esta idea a través de un breve recuento de lo acaecido en el proceso de realización de las dos obras aquí analizadas.

Subliminal fue resultado de un taller de composición para cello eléctrico y electrónica, en el que se contó con el auxilio de la violonchelista Iracema de Andrade para dar lectura a los primeros esbozos de las obras de cada uno de los participantes; asimismo, también ella grabó en estudio una serie de muestras de cello con diversas técnicas extendidas que no pueden producirse con un programa de notación musical y difíciles de conseguir en bibliotecas de sonidos de algún sampler. Más adelante, se tuvo una revisión de las partituras por parte de Jeffrey Zeigler, una videoconferencia donde interpretó las obras –ya con la parte electrónica controlada por él mismo con el *Live Patch*– y, en días previos al estreno de las obras, un ensayo con los compositores.

En la obra *Ocaso de Tenochtitlan*, la percussionista Alexandra Fuentes, estudiante de Licenciatura de la Facultad de Música de la UNAM, fue quien me auxilió para probar mis ideas musicales. Además de algunas consultas sobre la viabilidad de ciertos pasajes que estaba componiendo, fue de primordial importancia la verificación con ella del correcto funcionamiento del patch, de la claridad de la interfaz de usuario del mismo, y de la viabilidad del continuo manejo del pedal de expresión que ocurre en el segundo movimiento de la obra. Sólo cuando pude probar mi programa con el sonido real de la tarola en lugar del archivo de audio creado con MIDI, es que se logró ajustar en el patch las frecuencias de corte de los filtros usados en el

segundo movimiento y mejorar el balance entre el sonido de la tarola seca y la procesada que salen del programa.

5.2 Control de la electrónica por el intérprete

Ambas obras fueron compuestas pensando en que el intérprete sea quien desencadene o active los diversos módulos que crean la parte electrónica, por medio de un pedal que presiona en los puntos que le marca la partitura. De manera análoga al compositor que planea una obra de trompeta para que el intérprete pueda hacer fácilmente el cambio de sonido natural a instrumento con sordina, así deben de planearse los lugares de pulsación del pedal para que no interfiera con el discurso musical interpretado. En *Subliminal*, de las 18 ocasiones en que el intérprete debe pulsar el pedal, diez de ellas se dan cuando el violonchelo está en silencio; los pedales restantes están marcados en notas largas, después de algún calderón o de un ritenuto y en inicios de secciones y frases. Si bien, *Subliminal* también pudiera presentarse con la activación de los procesos electrónicos a cargo de un músico asistente –el cual tendría que estar atento a las señales del violonchelista en los ritenutos, calderones y lugares con tempo flexible–, la pieza de *Ocaso* no se podría adaptar a esa condición.

En la pieza para tarola existen varios puntos del primer movimiento en los que el momento de pulsación del pedal debe ser exacto –como en los compases uno, cinco y diez, donde el efecto de delay se aplica a sólo una nota del motivo–, por lo que el uso de un músico auxiliar para activar el efecto sería más conducente a errores, al igual que en el segundo movimiento, donde los movimientos del pedal de expresión –para cambiar los parámetros del filtro pasabanda– ocurren en cada compás y hasta cada dos pulsos. También en el segundo movimiento, el pedal de expresión pasa de funcionar con tres estados o posiciones a cambiar de forma continua los parámetros del filtro, para lograr tímbricas diferentes a través del movimiento continuo en toda la extensión del pedal, convirtiéndose así en un elemento más de expresión musical del intérprete, a la par de las dinámicas y del fraseo (ver compases 50 a 52 de la partitura).

Otro factor de peso para planear las obras con el control de los procesos de la electrónica por medio del intérprete en lugar de un músico asistente, fue la idea de lograr la autosuficiencia del intérprete no sólo en el concierto sino –tal vez, más importante– en los ensayos que realiza en

casa, para que los pueda llevar a cabo sin necesidad de otros recursos humanos y, como se verá más adelante, sin necesidad de recursos tecnológicos de difícil operación o instalación.

5.3 Realización de ensayos y conciertos

Una vez que el intérprete ha cumplido con el estudio necesario para dominar la partitura correspondiente a su instrumento, es esencial que lo más pronto posible interactúe con la sonoridad electrónica para tener una idea completa de la obra –en este sentido, puede ser muy similar a cuando forma parte de un dueto o un ensamble de cámara–. La parte electrónica de la obra puede ser desde limitada hasta muy amplia, como en la pieza de *Ocaso*, donde va aumentando gradualmente desde el primer movimiento –con un uso casi decorativo– hasta el tercer movimiento –con vastas secciones de pura electrónica y de importante complemento a la parte del instrumento–. De cualquier manera, aunque las partes de electrónica puedan ser mínimas, es necesario que se practique la interacción que se da con el instrumento: por ejemplo, en los compases dos, seis y once del primer movimiento de *Ocaso*, la duración de los silencios que debe hacer el instrumentista después de cada frase depende enteramente de la duración del efecto de delay en esos lugares, por lo que el percusionista debe estar atento para responder a la parte electrónica que está sonando.

Ya se habló en el capítulo uno de las formas en que puede ser más complejo el ensayo y la presentación de una obra electrónica en vivo en comparación con una obra electrónica mixta, pero ahora hablaré de algunos elementos y requerimientos técnicos que pueden favorecer la presentación exitosa de estas obras y motivar el ensayo frecuente de las mismas –centrándome, claro está, en los ejemplos de *Subliminal* y *Ocaso de Tenochtitlan*.

El programa creado con Max/MSP debe ser lo más claro posible en su interfaz de usuario: su pantalla principal debe mostrar sólo lo que sea necesario para el intérprete, con los recursos visuales de retroalimentación que necesita, como el número de pedal que se ha presionado, indicadores del nivel de señal de audio que entra y sale, la posición del pedal de expresión –en general, la entrada de datos MIDI– y algún botón para regresar los valores del programa a su estado inicial (ver Figuras 2 y 3).

Por otro lado, es de gran importancia que el intérprete tenga, en la medida de lo posible, la posibilidad de iniciar la obra desde cualquier sección particular que requiera ensayar sin tener que recorrer todas las secciones anteriores de electrónica desde el principio. Sin embargo, esto no

siempre es posible: en *Subliminal*, por ejemplo, el violonchelista puede comenzar a practicar sin problema desde el compás 122 (letra “E”, pedal seis), lugar donde activa el módulo de grabación para capturar la nota re índice tres, pero como esa nota es la que se procesa posteriormente para las sonoridades electrónicas que siguen, sería imposible empezar a ensayar a partir de los números de pedal ocho o nueve de la obra. Por el contrario, el primer movimiento de *Ocaso* puede ensayarse desde el inicio de cualquier sección –secciones que están delimitadas por las indicaciones del pedal.

Por último, aparte de la constante comunicación entre compositor e intérprete, la elaboración de un audio de demostración que proporcione el primero, donde se muestre la combinación de instrumento y electrónica, aclarará muchas dudas que la mejor explicación no podrá.

Equipo mínimo necesario

En cuanto al equipo necesario –hardware y software– para que el intérprete pueda practicar la pieza, aquí se da una simple lista con precios actuales y algunas indicaciones generales sobre sus características, puesto que varios de esos ítems ya se han descrito en el capítulo anterior.

1. Computadora personal. De preferencia laptop, para que este sistema sea portátil, con el programa Max/MSP instalado. El intérprete no necesita comprar el programa, sólo se descarga el demo, se instala y se abre el patch – un archivo con la extensión maxpat o mxp– que le proporciona el compositor. En el caso de *Subliminal* se requiere de software adicional: los plug-ins Spectral DroneMaker, Spectral Granulation de Michael Norris y el plug-in Camel Crusher, los que se pueden obtener gratuitamente. Es recomendable que la computadora esté situada cerca del intérprete para que este visualice en su pantalla el número de pedal que va presionando en cada momento o cualquier otra información que se requiera monitorear, aunque también es posible utilizar un monitor adicional para ello.

2. Una interfaz de audio con una entrada de audio (para *Subliminal*) o dos entradas de audio (para *Ocaso*). Los modelos siguientes, de dos entradas de audio, son sólo algunos ejemplos, pero cualquier otra interfaz equivalente puede usarse: Focusrite Scarlett 2i2 (\$170 USD) o Behringer U-Phoria UMC202H (\$117 USD).

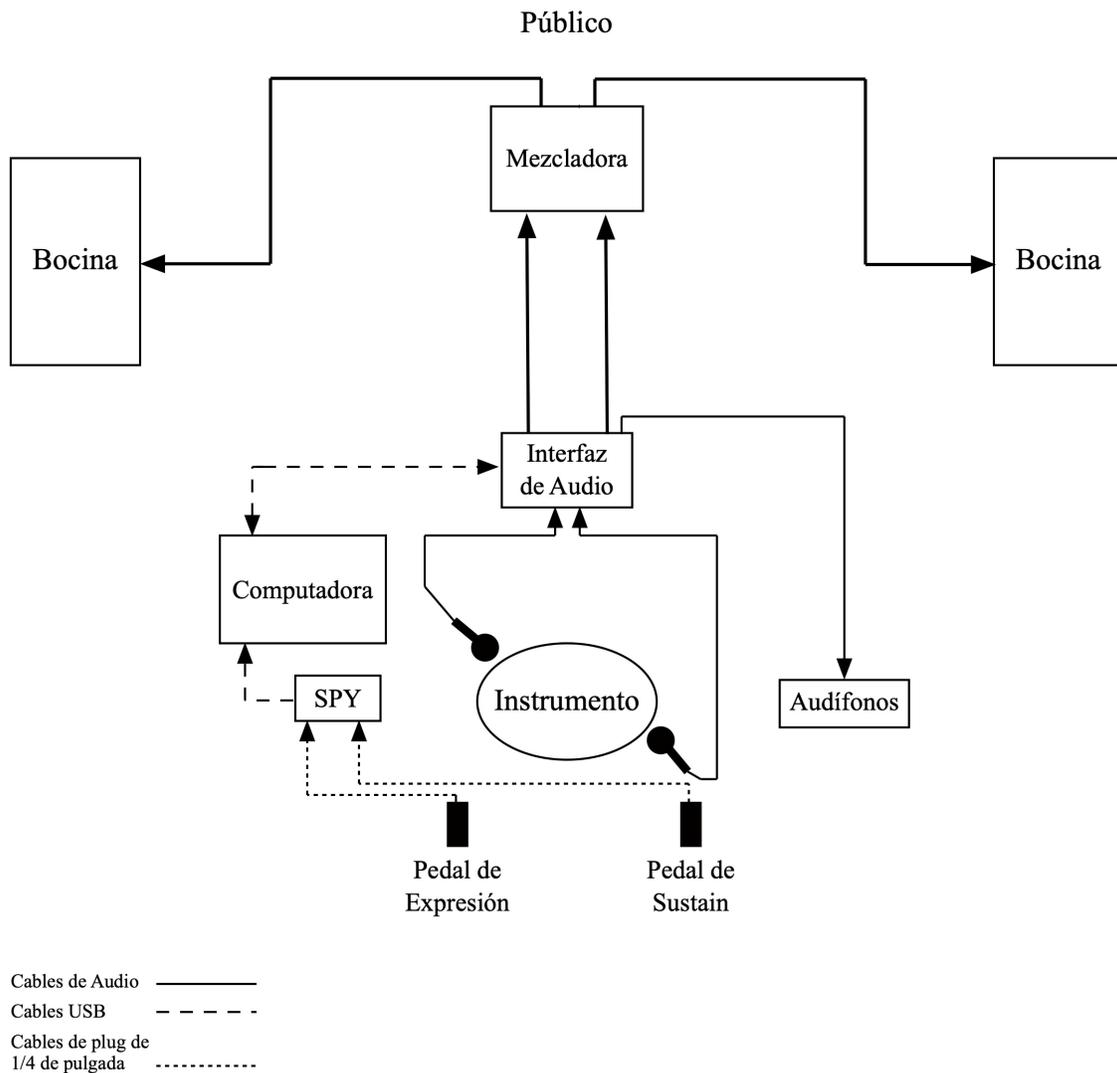
3. Un pedal de sustain (entre \$14 y \$30 USD) y un pedal de expresión (alrededor de \$30 USD) –este segundo pedal, sólo se requiere para *Ocaso*.

4. SPY de Midiplus (\$46 USD). Es aquí donde se conectan ambos pedales para enviar sus mensajes a la computadora. Recomiendo el uso de este dispositivo –o alguno equivalente que se encontrara en otra marca– porque evita el tener que cargar con un teclado y una interfaz MIDI (y es más económico).

Con este equipo, ya es posible practicar la pieza *Subliminal*, sólo hay que agregar unos audífonos o bocinas que sonarán el audio salido de la interfaz. Para el estudio de la pieza *Ocaso*, habría que añadir dos micrófonos, con sus atriles, para capturar el sonido de la tarola.

Figura 20

Colocación de equipo recomendada para la ejecución en concierto de Ocaso de Tenochtitlan



El equipo listado en los cuatro puntos anteriores es el que necesitaría llevar el intérprete para presentar estas piezas en diferentes recintos, los que posiblemente tendrán los aparatos de amplificación para que el audio que sale de la interfaz llegue al público: una mezcladora, amplificador, bocinas y monitores de piso. Incluso los micrófonos y atriles de los mismos pueden tenerlos ya algunos auditorios, así que básicamente, el ingeniero de audio del recinto –que siempre es bueno verificar que cuenten con uno– tendría que tomar la salida de audio de la interfaz para llevarla a su mezcladora. En la Figura 20, se puede ver el esquema de conexiones para la pieza *Ocaso*, que es el mismo que se usaría para *Subliminal*, pero sin pedal de expresión, sin los micrófonos –puesto que es violonchelo eléctrico el que se utiliza– y con monitores de piso en lugar de audífonos.

6. Conclusiones

Un hecho frecuente en el campo de la programación es que pueden encontrarse diferentes maneras para solucionar un problema dado. Dentro de las obras analizadas, puede observarse un ejemplo muy claro de distintos estilos de programación en los mecanismos que prenden y apagan los procesadores sonoros y demás módulos: en ambos casos se recibe el conteo de las pulsaciones del pedal de sustain pero en el *Live Patch* se utiliza el objeto [preset] y en el patch de *Ocaso* se usa la combinación de los objetos [select] y [toggle]. Otro ejemplo se observa en las secciones de entrada y salida de audio, donde el *Live Patch* utiliza el objeto [live.gain~] y mi patch usa la conjunción de [gain~] y [meter~]: el resultado es exactamente el mismo en ambos programas, y la elección de uno sobre otro puede ser, en ocasiones, enteramente fortuita, como en mi patch, donde simplemente seguí los ejemplos de los libros consultados cuando empecé a programar en Max/MSP –en ninguno de los cuales el [live.gain~] era usado.

Dentro del mismo Max/MSP se pueden encontrar diferentes objetos que llevan a cabo funciones similares, como en la reproducción de archivos de audio, que se puede realizar con [sfplay~], [play~], [index~] o [groove~], cada uno con sus particularidades que lo hará más o menos apropiado que otros dependiendo del contexto.

Al fin y al cabo, tener diferentes opciones de solución a un problema es una ventaja, y en ese sentido, el estudio de los dos acercamientos mostrados para la programación de estas obras es provechoso e ilustrativo y puede ayudar a una mejor comprensión de los elementos involucrados y su interacción.

Por otro lado, hay una diferencia importante en el diseño de los patches de las dos obras. El *Live Patch* de Esteban Chapela fue creado para ser usado por diversos compositores –con estilos y formaciones disímiles– a los que el programador deseaba proporcionar una herramienta lo suficientemente completa y flexible. El patch programado por mí, fue diseñado para uso exclusivo de mi pieza *Ocaso*, aunque incluí en él alguna rutina o subpatch de una composición previa –de la misma forma que, muy probablemente, tomaré este patch como base para futuras composiciones electroacústicas.

Los quince años que Chapela lleva trabajando con Max son claramente visibles en su *Live Patch*: la interfaz de usuario, la maleabilidad para mandar el sonido de un procesador a otro cualquiera –para lo que diseñó una mezcladora virtual– y la posibilidad de programar en un

reproductor de audio la secuencia de los tiempos de pulsación de los pedales para simular la interpretación en vivo, son sólo algunos aspectos sobresalientes de su programa. Aún así, siempre hubo alguien que pidió algo más: un procesador diferente, posibilidad de grabar más archivos de audio, entre otras peticiones. En mi caso, yo tenía algunas ideas musicales con efectos de delay y flanger, que tuve que descartar porque estos procesadores no estaban incluidos en *Live Patch* y porque tampoco se vislumbraba que se añadieran para quitarle el lugar a los granuladores –procesamiento que suele ser muy socorrido en las piezas electrónicas–. En cambio, en mi pieza *Ocaso* utilicé ampliamente el delay y el flanger, al igual que sistemas de control inexistentes en el *Live Patch*, como el apagado automatizado del delay, el monitoreo del nivel de audio de entrada para controlar el volumen del sonido procesado que sale del módulo y el detonador de compuerta que cambia la frecuencia central de un filtro pasa banda cuando el audio de entrada excede un nivel determinado. De alguna forma, el contraste mostrado aquí entre el patch de Chapela y el mío es muy parecido al que ocurre entre el software comercial y el custom: el primero es general y el segundo es específico.

Podría pensarse que las líneas anteriores son una crítica hacia el *Live Patch*, pero ese no es su objetivo, sino mostrar características de ambos patches que determinaron algunos aspectos del trabajo de composición. De varias maneras y en diferentes grados, todo programa presenta límites al usuario y puede dirigir las formas de trabajo con él –y hasta imponer lineamientos estéticos–, pero esto no es necesariamente un aspecto negativo. En mi experiencia, he visto que las restricciones y límites son buenos para la creación. El compositor William Russo afirma que “El control y las restricciones conducen a la creatividad y expansión [...] Algunas restricciones dan enfoque, concentrándote en los procedimientos a la mano –evitan que tengas tantas opciones que no puedas entrar en contacto con tus sentimientos e ideas” (1988, p.3). Puedo confirmar que esto último me ocurrió al componer *Subliminal*, y el *Live Patch* usado para esta obra sugirió nuevos caminos en mi forma habitual de composición e inspiró ideas que, tal vez, no hubieran surgido de no tener esas opciones disponibles.

En resumen, se presentan aquí dos formas de trabajo igualmente viables y provechosas. En la primera, se muestra una colaboración entre programador y compositor que puede ser muy conveniente, aunque, a diferencia del *Live Patch*, mi propuesta es que el programador trabaje específicamente para una obra determinada del compositor en lugar de hacer un patch de uso general. Este tipo de trabajo colaborativo debería ser alentado: ya se da en varias áreas de la

música, como cuando un compositor y un escritor participan en la elaboración de una canción. No obstante, para que esta relación sea fructífera, el compositor debe ser consciente de las posibilidades, alcances y límites del lenguaje de programación y, en general, de toda la tecnología implícita en el desarrollo de una obra electrónica en vivo. Es aquí donde el análisis realizado en este trabajo y el modelo propuesto aspiran a asistir al músico interesado en iniciarse en este campo de la composición. En la segunda forma de trabajo, el compositor crea las ideas musicales y programa el patch. A primera vista, pareciera que esta opción no tiene grandes ventajas: el aprendizaje de un lenguaje de programación requiere una enorme inversión de tiempo y ciertas habilidades matemáticas que, en muchas ocasiones, los músicos no hemos cultivado. En el caso anteriormente expuesto –donde se trabaja con un programa desarrollado por otro–, el compositor puede experimentar con los procesadores proporcionados sin necesidad de conocer cómo funcionan y puede mover perillas y apretar botones sin saber el porqué de la modificación sonora que resulta –únicamente es guiado por su oído, su intuición y su experiencia en la composición–. Pero si el compositor pasa de ser un usuario de un programa a un programador del mismo, el entendimiento profundo y preciso del flujo de audio de un módulo a otro y de las formas de controlar y manipular el sonido pueden hacerle descubrir oportunidades creativas que no podría vislumbrar de otra manera. Varias de las ideas musicales para mi pieza *Ocaso de Tenochtitlan* surgieron del estudio detallado de objetos de Max/MSP y de la experimentación con patches de los libros de Manzo y de Cipriani y Giri, patches que modifiqué, mutilé, amplié y adapté para mis propósitos compositivos.

El modelo de programación presentado puede proporcionar un panorama general que ayude a planear una ruta de trabajo en el proceso de programación y composición de una obra electrónica en vivo. Puede pensarse también como una estructura o esqueleto a rellenar con diferentes procesadores, generadores y reproductores de audio –para los que sí se encuentra abundante información, en libros y tutoriales, que enseñan a diseñarlos.

Este modelo también puede ampliarse para incluir opciones no mostradas en este análisis. Un *looper*, por ejemplo, es simplemente un reproductor de audio de algo previamente grabado en el mismo instante de la ejecución –de hecho, el módulo Recorder de *Live Patch*, permite hacerlo–. Se pueden incluir módulos generadores basados en composición algorítmica que utlicen el sintetizador interno de Max/MSP –o cualquier otro virtual o físico– para sonar el material musical creado por las fórmulas y directrices dadas al patch. También es posible, a

través del uso de instrumentos MIDI, que las funciones de procesamiento sonoro no se lleven a cabo directamente sobre el audio digital sino mandando mensajes MIDI que modifiquen los parámetros de los procesadores internos de los sintetizadores, esto es, nuestro modelo no utilizaría la entrada de audio sino únicamente la entrada de datos de un instrumento MIDI.

Para finalizar, espero que la aplicación de este modelo se haga extensivo a otros entornos de programación diferentes a Max/MSP, y que el lector que haya recorrido este trabajo obtenga ideas y motivación para utilizar un lenguaje de programación en la creación de obras electrónicas en vivo.

Referencias

Anderton, C. (1980). *Electronic projects for musicians*. New York, NY: Amsco Publications.

Barret, N. (2007). Trends in electroacoustic music. En N. Collins y J. Escrivan (Eds.), *The Cambridge companion to electronic music* (pp. 232-255). Cambridge: Cambridge University Press.

Boutard, G., y Féron, F. (2018). Instrumentalists on solo works with live electronics: Towards a contemporary form of chamber music? En V. Bertolani, J. Burle, F. Sallis, y L. Zattra (Eds.), *Live electronic music: Composition, performance, study* (pp. 101-130). Abingdon, Oxon: Routledge.

Cetta, P. (2014). *Captura y procesamiento de sonido*. Bernal, Buenos Aires: Universidad Virtual de Quilmes.

Cipriani, A., y Giri, M. (2010). *Electronic music and sound design - Theory and practice with Max/MSP* (Vol. 1). Rome: ConTempoNet.

Cipriani, A., y Giri, M. (2019). *Electronic music and sound design: Theory and practice with Max 8* (4ª ed., Vol. 1). Rome: ConTempoNet.

Cipriani, A., y Giri, M. (2020). *Electronic music and sound design: Theory and practice with Max 8* (3ª ed., Vol. 2). Rome: ConTempoNet.

Colasanto, F. (2010). *Max/MSP: Guía de programación para artistas*. Morelia: Centro Mexicano para la Música y las Artes Sonoras.

Collins, N., y Escrivan, J. (2007). *The Cambridge companion to electronic music*. Cambridge: Cambridge University Press.

Corey, J. (2010). *Audio production and critical listening: Technical ear training*. Burlington, MA: Focal Press.

Eimert, H. (1973). *¿Qué es la música electrónica?* (Vol. 14, Fichas). Buenos Aires: Nueva Visión.

Emmerson, S. (2007). *Living electronic music*. Aldershot: Ashgate.

Emmerson, S., y Smalley, D. (2001). Electro-acoustic music. En S. Sadie (Ed.), *The new Grove dictionary of music and musicians* (2ª ed., Vol. 7, pp. 59-67). London: Macmillan.

Escriván, J. (2012). *Music technology (Cambridge introductions to music)*. Cambridge: Cambridge University Press.

Everest, F. A. (2007). *Critical listening skills for audio professionals*. Boston, MA: Course Technology, Cengage Learning.

Gallagher, M. (2009). *Music tech dictionary: A glossary of audio-related terms and technologies*. Boston: Course Technology, PTR.

Gibson, D. (2019). *The art of mixing: A visual guide to recording, engineering, and production* (3ª ed.). New York, NY: Routledge.

Govea, S. (2020a). Bibliotecas y editores de patches. *Sound:check, Año 23(265)*, 16-18.

Govea, S. (2020b). Siete programas para crear música. *Sound:check, Año 22(261)*, 20-22.

Izhaki, R. (2008). *Mixing audio: Concepts, practices and tools*. Burlington, MA: Focal Press.

Manzo, V. J. (2011). *Max/MSP/Jitter for music: A practical guide to developing interactive music systems for education and more*. NY, NY: Oxford University Press.

Manzo, V. J. (2016). *Max/MSP/Jitter for music: A practical guide to developing interactive music systems for education and more* (2^a ed.). NY, NY: Oxford University Press.

Miyara, F. (2000). *Acústica y sistemas de sonido*. Rosario: Universidad Nacional de Rosario.

Montague, S. (2001). Termen, Lev Sergeyevich. Recuperado el 15 de diciembre de 2020, desde <https://doi.org/10.1093/gmo/9781561592630.article.45834>

Norris, M. (2016). *Soundmagic spectral: Real-time spectral plug-ins for creative audio processing - User documentation* [June 2016]. Recuperado el 1 de enero de 2021, desde <https://www.michaelnorris.info/software/soundmagic-spectral>

Odgers, A. (2000). *La música electroacústica en México* (Tesis de Licenciatura en Composición, Universidad Nacional Autónoma de México) (pp. 288). Distrito Federal, México: UNAM.

Pavón, R. (1981). *La electrónica en la música y en el arte*. Mexico City, Mexico: INBA.

Puckette, M. (1991). Combining event and signal processing in the MAX graphical programming environment. *Computer Music Journal*, 15(3), 68-77. doi:10.2307/3680767

Puckette, M. (2002). Max at seventeen. *Computer Music Journal*, 26(4), 31-43.

Roads, C. (2015). *Composing electronic music: A new aesthetic*. Oxford: Oxford University Press.

Rowe, R. (2001). *Machine musicianship*. Cambridge, MA: MIT Press.

Russ, M. (2009). *Sound synthesis and sampling* (3^a ed.). New York, NY: Focal Press.

Russo, W., Ainis, J., y Stevenson, D. (1988). *Composing music: A new approach*. Chicago: The University of Chicago Press.

Sallis, F., Bertolani, V., Burle, J., & Zattra, L. (2018). *Live electronic music: Composition, performance, study*. London Abingdon (Oxfordshire): Routledge.

Schrader, B. (1991). Live/electro-acoustic music — a perspective from history and California. *Contemporary Music Review*, 6(1), 91-106. doi:10.1080/07494469100640101

Sigal, R. (2009). *Compositional strategies in electroacoustic music: Generating materials and creating an effective musical language in electroacoustic music*. Saarbrücken, Germany: VDM Verlag Dr. Müller.

Solomon, S. Z. (2016). *How to write for percussion: A comprehensive guide to percussion composition* (2^a. ed.). New York: Oxford University Press.

Stroppa, M. (1999). Live electronics or...live music? Towards a critique of interaction. *Contemporary Music Review*, 18(3), 41-77. doi:10.1080/07494469900640341

Theremin, L. (1999). Recollections. *Contemporary Music Review*, 18(3), 5-8. doi:10.1080/07494469900640281

Thierauf, A. (2015). *Percussion and Max: A collection of short works for solo percussion and live electronics* (Master's thesis, University of Iowa, 2015) (p. 132). Iowa.

What is Max?: Cycling '74. (s.f.). Recuperado el 4 de agosto de 2020, desde <https://cycling74.com/products/max>

White, P. (2000). *Basic mixers*. London: Sanctuary.

Winkler, T. (1998). *Composing interactive music: Techniques and ideas using Max*. Cambridge, MA: MIT Press.

Zicarelli, D., y Taylor, G. (2006). *Max fundamentals* (Versión 4.6). San Francisco, California: Cycling '74.

Discografía

Jeffrey Zeigler. (2018). *Impulse Codes: Jeffrey Zeigler, cello and electronics* [CD]. Urtext Records. (2017)

Manuel Rocha Iturbide (comp.). (2008). *México electroacústico (1960-2007)* [CD]. Irradia.

Apéndice: Partituras

Subliminal
para violonchelo eléctrico
y electrónica

Salvador Govea

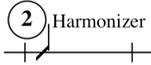
**Subliminal
para violonchelo eléctrico y electrónica**

Cada día nos preguntamos y damos respuesta a muchas preguntas sobre nuestros sentimientos y elecciones. Nuestras respuestas suelen tener sentido, pero a pesar de ello son del todo equivocadas.

Leonard Mlodinow

Creemos tener control de nuestros actos. Inventamos multitud de razones y armamos un complejo entramado de argumentos para justificar nuestras acciones. La mala noticia es que mucho de lo que hacemos y decidimos está por debajo del umbral de nuestra consciencia, es decir, es subliminal. Tendemos a actuar bajo esquemas preestablecidos y somos dirigidos a cada momento por nuestros prejuicios. ¿Por qué utilicé tal acorde?, ¿por qué la melodía tiene ese contorno?, ¿cómo llegué a estructurar la pieza de esa manera? Podría justificarlo de forma lógica y escribir veinte cuartillas sobre el tema... pero nos estaríamos engañando: la mayor parte de eso lo decidió mi inconsciente. Así que, mejor te invito simplemente a escuchar y conocer un poco de mi vida subliminal.

Nomenclatura

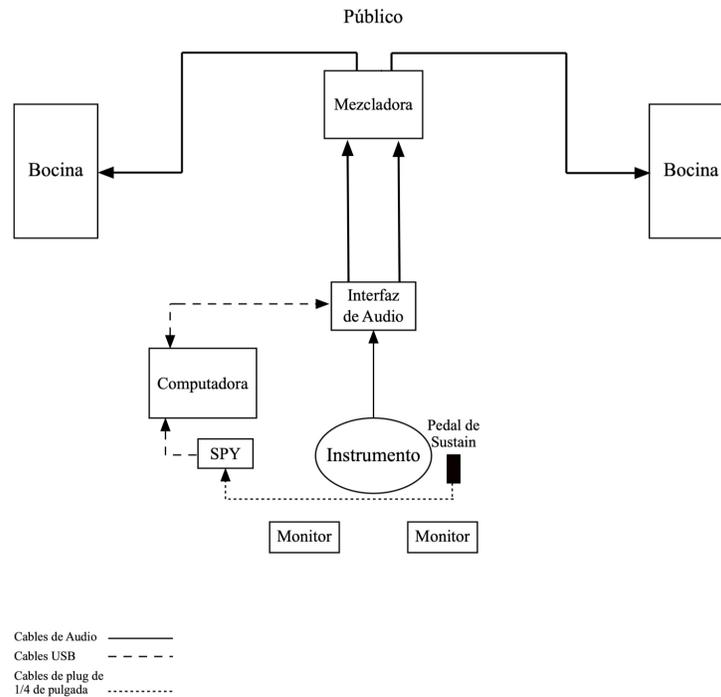


Un número encerrado en un círculo, en la parte de “Electronics”, representa el punto exacto de la partitura donde el intérprete debe pulsar el pedal de sustain. El número indica el número del proceso en la pieza –que se visualizará en la pantalla del programa–, proceso que suele describirse con una o dos palabras en la partitura.

Equipo necesario y configuración

1. Computadora personal con la versión 7.0 o posterior de Max/MSP (únicamente se requiere el demo del programa para poder ejecutar el patch con el que se realizará la interpretación).
2. Una interfaz de audio, con una entrada de plug de 1/4 de pulgada.
3. Un pedal MIDI de sustain.
4. Una interfaz SPY de Midiplus. Es posible usar en su lugar una interfaz MIDI y un teclado MIDI con entrada para pedal de sustain.

Disposición del equipo para el concierto



Subliminal

for Electric Cello and Electronics

Salvador Govea

♩ = 260 (♩ = 87)

Solo Cello

Electronics

① Player

Metallic Sound (continues)

3

3

f

9

16

23

3 **A**

3 *mf*

33

41

48

55 **B**

64 2

74

② Harmonizer

③ Stop Harmonizer

83 C

mp *mf*

④

90

f *f*

98

④

107

④ Harmonizer *f*

④

117 E

⑤ Stop Harmonizer *mf* ⑥ Recorder ⑦ Stop Recorder Flanger

④

Let the Flanger effect establish the pulse. Group this pulses in 3/4.
 The Multirest bars have an approximate duration,
 keep the proportion between them.

127

mp

⑦ ③ ③

142 *pizz.*
Gliss. *arco*
⑧ Random Grains *mf*

150 *arco*
p ⑨ Harmonics descending Strong Flanger *mf*

158 *p* *mf* *pizz.*

164 *arco* *pp*

172 *mf* ⑩ Harmonizer *pizz.*

178

184 *mf* *f* [F] ⑪ Stop Flanger *3* *3* $\frac{4}{4}$

192 *espressivo*
arco

mp *p* *mf*

197 $\bullet = 60$

pp *p*

12 Filter 13 Gliss.

202 - a tempo

mp

14

205 . a tempo poco a poco acell-

mf *f*

15 16

208 $\bullet = 74$ ca.

ff

17

211

18

213 $\bullet = 260$

ff

19

The electronic sounds establishes the pulse.
(The pulse is equal to 1/8 notes).
Multirest bars are approximate.

220 **G** $\text{♩} = 260 (\text{♩} = 87)$
 18 Player

229 *mf*
 Perc.

237 **H**
f

247 **I**
mf

257 *f*

266 **J**
mf
 8va
 Metallic Sound (continues)

273 K

279 *mf*

287

292 L *f* *f*

299 *molto rit.*

305 M *p* *molto accelerando* *molto rit.*

7

310 ----- $\text{♩} = 160 \text{ ca.}$ molto acclerando ----- poco rit.-----

315 ----- molto acclerando ----- \boxed{N} $\text{♩} = 290$ ($\text{♩} = 97$) a tempo

321 ----- pizz. -----

(continues) *mp* *mf*

329 ----- \boxed{O} arco -----

f

337 ----- \boxed{P} -----

ff

345 ----- \boxed{Q} -----

352 R S

Glissando

360 T

Ocaso de Tenochtitlan

para tarola y electrónica

Salvador Govea

I. Presagios

II. Soles de metal fino

III. Pólvora, gérmenes y acero

Ocaso de Tenochtitlan para tarola y electrónica

Ocaso de Tenochtitlan es una obra para tarola y electrónica en vivo, esto es, no existe una pista con sonidos electrónicos previamente grabados, sino que todos los sonidos son producidos –la mayoría de ellos con la modificación del audio de la tarola– en el mismo instante de la interpretación. El intérprete mismo es quien activa o dispara –a través de un pedal– los diferentes elementos que forman la parte electrónica.

Esta obra conmemora los 500 años de la caída de Tenochtitlan, ocurrida el 13 de agosto de 1521, en manos de los conquistadores españoles y sus aliados indígenas.

El primer movimiento, “Presagios”, trata sobre las señales y malos augurios que afirmaron ver los mexicas previo al arribo de los españoles, entre los que se cuentan una especie de llama de fuego que se veía en el cielo, una mujer que lloraba y gritaba fuertemente por las noches, y el ave con un espejo en la frente, en el que Moctezuma vio el cielo, las estrellas y jinetes en actitud de combate. En este movimiento la electrónica se combina poco a poco con lo acústico –en alusión a esa mezcla de realidad y fantasía que convivía en los citados presagios– para que el oyente empiece a adentrarse en el nuevo mundo sonoro que es la interacción entre el sonido que escuchará directo del instrumento y el de las bocinas.

“Soles de metal fino”, el segundo movimiento, hace referencia a los obsequios enviados por Moctezuma a Cortés, entre los que se encontraban dos grandes láminas circulares –una de oro y una de plata–, según se relata en *La visión de los vencidos* de Miguel León Portilla:

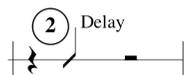
Por este motivo va a darle allá soles de metal fino, uno de metal amarillo y otro de blanco. Y un espejo de colgar, una bandeja de oro, un jarrón de oro, abanicos y adornos de pluma de quetzal, escudos de concha nácar.

En cuanto a la parte musical, este movimiento trata de emular, a través de varios filtros, tres sonidos acústicos de percusión que entablarán una conversación –como la que debió de haber existido en ese primer encuentro entre los mexicas y el conquistador Cortés.

El tercer movimiento se llama “Pólvora, gérmenes y acero”, título inspirado en el libro *Armas, gérmenes y acero* del geógrafo y biólogo Jared Diamond. Trata de representar el enfrentamiento militar que se dió entre las dos culturas y la importancia que tuvo para el mismo la superioridad tecnológica de los españoles –materializada en sus armas de fuego y de acero–, así como el virus de la viruela que trajeron consigo y que diezmo a la población mexicana. La parte final del movimiento, en cierto tono optimista, intenta representar esta innegable hibridación cultural de la que formamos parte los actuales mexicanos.

Indicaciones de interpretación y nomenclatura

Toda la obra se interpreta con entorchado.



Un número encerrado en un círculo, en la parte de “Electronics”, representa el punto exacto de la partitura donde el intérprete debe pulsar el pedal de sustain. El número indica el número del proceso en la pieza –que se visualizará en la pantalla del programa–, proceso que suele describirse con una o dos palabras en la partitura.

En “Soles de metal fino”:



Indica que el pedal de expresión debe colocarse en su punto mínimo, presionando con el talón hacia atrás.



El pedal de expresión debe colocarse en una posición media.



El pedal de expresión debe colocarse en su punto máximo, presionando con la punta del pie hacia adelante.



El pedal de expresión puede moverse hacia adelante y hacia atrás, al gusto del intérprete.



En los compases 1 a 49, 53 a 58 y 117 hasta el final del movimiento, la colocación de las notas en la primera, tercera o quinta línea del pentagrama, equivalen, respectivamente, a las posiciones baja, media y alta del pedal de expresión.



En la sección media del movimiento (compases 59 a 99) la posición del pedal de expresión panceará el audio entre las bocinas a velocidades lenta, media y rápida –en lugar de modificar el filtro que actúa

sobre el audio de tarola-. En esta sección, la escritura se limita al tercer espacio del pentagrama, porque el sonido de la tarola es natural.

En “Pólvora, gérmenes y acero”:

El número de compases de silencio en la sección inicial del movimiento (desde el compás 1 al 33) son sólo una sugerencia, el intérprete puede hacerlos tan extensos como quiera –lo que es posible porque la electrónica no reside en una pista pregrabada.



Alternancia entre golpes en la orilla del tambor –notas escritas arriba de la quinta línea– y golpes en el centro del tambor –notas escritas en el tercer espacio.



Cambiar gradualmente el área percutida desde la orilla hasta el centro del tambor.

Acerca de los puntos de ensayo de la obra

Aunque la ejecución de los procesos de la parte electrónica sigue un orden predeterminado en el patch creado con Max/MSP, el intérprete tiene libertad de poder practicar la pieza a partir de diferentes puntos de la partitura, generalmente en los lugares donde se presiona el pedal de sustain –sitios que se visualizan en la línea de “Electronics”.

Por ejemplo, si en el movimiento Presagios se desea practicar a partir de la anacrusa del compás 23, en la ventana principal del patch de Max –en el rectángulo debajo de las palabras “Start on Pedal”– hay que colocar el número 6 y pulsar Enter. Después, cuando el intérprete esté listo, sólo habrá que presionar el pedal y empezar a tocar desde ese punto: los procesamientos sonoros disparados en esa sección serán exactamente los requeridos. Este procedimiento funciona casi en toda la obra, con algunas excepciones que se detallan a continuación.

I. Presagios. Utiliza los pedales 1 a 11. Se puede practicar a partir de cualquiera de los puntos donde se marca un pedal en la partitura.

II. Soles de metal fino. Consta de los pedales 12 a 21. Es posible el ensayo desde cualquier punto marcado por los pedales 12 al 18. Por el contrario, como los pedales 18 a 21 van creando un efecto sonoro acumulativo, si se quiere iniciar la práctica desde el pedal 19 –o el 20, o el 21– el resultado sonoro no será el apropiado, puesto que depende de que se hayan pulsado previamente los números de pedal que anteceden.

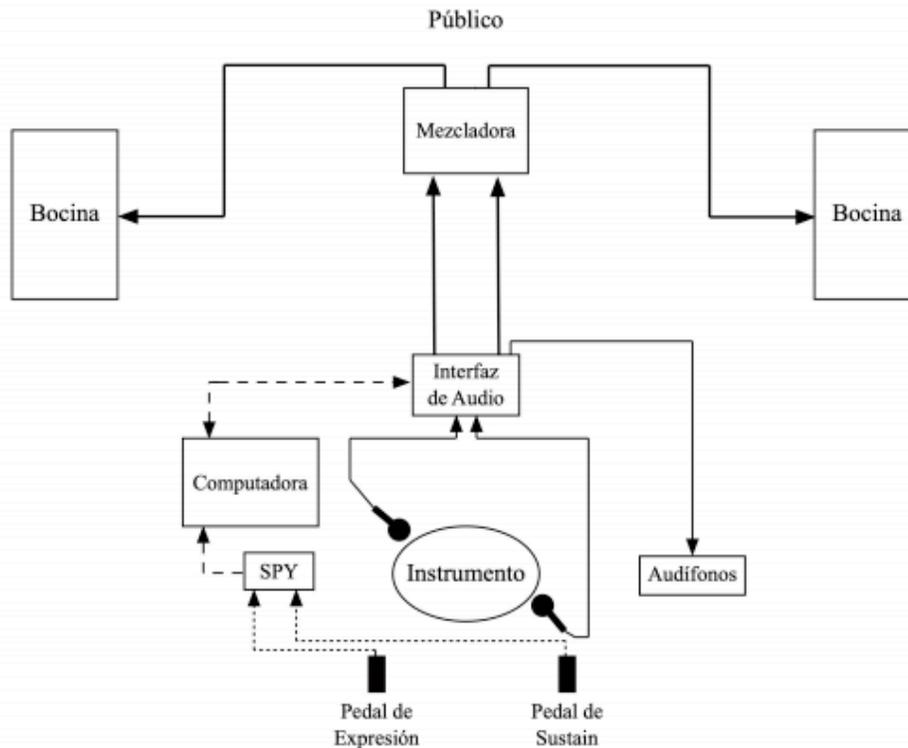
III. Pólvora, gérmenes y acero. Pedales 22 a 32. Los puntos de ensayo están marcados por los pedales 22, 27, 28, 30 y 32. Debido a que hay procesos que se van acumulando –como los armónicos aleatorios del inicio al que se le suman, con el pedal 23, las percusiones sampleadas–, existen pocos lugares de partida para la práctica. Sin embargo, es posible el siguiente método: digamos que se quiere estudiar desde el pedal 24 (compás 63), entonces hay que partir del pedal 22 y una vez que se escuchan los armónicos aleatorios se puede presionar el pedal 23 –para que entren las muestras de percusiones– y después el pedal 24, que cambiará el patrón de la muestra de percusión y activará el procesamiento que provoca el efecto metálico en la tarola.

Por último, es recomendable utilizar el botón “Reset” de la pantalla principal, cuando se quiera interrumpir un proceso sonoro o se requiera saltar a un pedal posterior o anterior en lugar de seguir el orden numérico de los pedales.

Equipo necesario y configuración

1. Computadora personal con la versión 7.0 o posterior de Max/MSP (únicamente se requiere el demo del programa para poder ejecutar el patch con el que se realizará la interpretación).
2. Una interfaz de audio, con dos entradas para micrófono (XLR).
3. Un pedal MIDI de sustain y un pedal MIDI de expresión.
4. Una interfaz SPY de Midiplus. Es posible usar en su lugar una interfaz MIDI y un teclado MIDI con entradas para los pedales de sustain y expresión.
5. Dos micrófonos con sus atriles. Se recomienda uno dinámico para captar el sonido del parche de la tarola y uno de condensador para el entorchado.
6. Audífonos Over Ear para el monitoreo del intérprete.

Disposición del equipo para el concierto



Cables de Audio —————
Cables USB - - - - -
Cables de plug de 1/4 de pulgada ······

Ocaso de Tenochtitlan

for Snare Drum and Electronics

Salvador Govea

I. Presagios

$\text{♩} = 80$

Snare Drum

Electronics

6

13

16

20

24

28

30

1 *Delay*

2 *Delay*

3 *Delay*

4 *Filter Fade In*

5 *Filter 2*
2nd. time only

6 *Flanger*

p *mp* *mf* *mf* *mp* *mf* *mp*

mf *mp* *mf* *p* *mf* *p*

mf *3* *p* *mf* *3* *p*

mf *3* *p* *mf* *3* *f* *p*

mf *3* *f* *3* *p* *mf* *3* *f*

Ocaso de Tenochtitlan

32 *p* *f* *f*

35 *p* *mp* *mf*

35 (7) Stop Flanger

38 *f* *mf* *f* *mp*

38 (8) Delay (9) Delay (10) Flanger

41 *mf*

43 *f*

45 > > > > > >

47 *ff*

50 *f*

53 *mf* (11) Delay

2

II. Soles de metal fino

$\text{♩} = 96$

The musical score is written for piano in 9/8 time. It consists of a single system with a grand staff (treble and bass clefs) and an Expression Pedal staff below. The score is divided into measures, with measure numbers 6, 11, 16, 21, 25, 30, 35, 40, and 45 indicated at the start of their respective lines. The music begins with a piano (*p*) dynamic and a circled '12' above the pedal staff, indicating a filter setting. The score features a variety of rhythmic patterns, including eighth and sixteenth notes, and rests. Dynamics range from piano (*p*) to fortissimo (*ff*). The word 'simile' is used in measure 10. The Expression Pedal staff shows the pedal being depressed and released throughout the piece.

p
Pedal 12 Filter
Expression Pedal

6

mp simile

11

16
mf

21
f *mf* *f* *mf*

25
f *mf* *f* *mf*

30
f *mf*

35
f

40
ff

45
mf *ff*

Ocaso de Tenochtitlan

49 4 times

mf (13) 1.*p* 2.*mp* 3.*mf* 4.*f* *f*

55 *mf* *mp* *pp* (14) Filter Off *ppp* Slow Panning

61 *pp* Slow Panning

67 Medium Panning Slow Panning

73 Medium Panning

79 Slow Panning Medium Panning Fast Panning

85 Slow Panning

90 Medium Panning

The image displays a musical score for 'Ocaso de Tenochtitlan'. It consists of eight systems of music, each with a piano staff and a MIDI controller staff. The piano staff contains melodic lines with various dynamics and articulations. The MIDI controller staff contains panning and filter instructions. The score is marked with measure numbers 49, 55, 61, 67, 73, 79, 85, and 90. Dynamics include *mf*, *p*, *mp*, *f*, *pp*, and *ppp*. Panning instructions include 'Slow Panning', 'Medium Panning', and 'Fast Panning'. A filter instruction 'Filter Off' is also present. A circled number '13' is in the first system, and '14' is in the second system. The first system is marked '4 times'.

Ocaso de Tenochtitlan

95

Fast Panning

15 Very Fast Panning

101

105

111

16 Stop *ppp* Processing

mf

17 Filter

117

mf

simile

122

f

127

18 Adds Percussion

132

19 Adds Percussion

20 Adds Percussion

137

21 Adds Percussion

III. Pólvora, gérmenes y acero

Freely 12 seconds 35 seconds $\text{♩} = 268$ ($\text{♩} = \text{c. } 89$)

22 Low Triangle Wave Harmonic Walk Random Harmonics *niente* *mf* *simile* (continues)

8 *p*

17 *pp* *p* *mp*

26 *p* *mp* *p*

34 *mf* (continues)

23 Sampled Percussions *mf* (continues)

44 *mp* *mp*

52 *mf* *mp* *mf*

58 *mp* *mf* *mp* *mf* (24) Filtered Noise Metallic Sound (continues)

66 *p*

Ocaso de Tenochtitlan

Freely

151 $\text{♩} = c. 80$

White Noise Reverb (continues)

27 *pp*

156 $\text{♩} = c. 84$

mf p

162 *mf p*

28 White Noise + Sine Wave

167 $\text{♩} = c. 84$

edge *ppp* *pp*

172 *pp* edge *pp*

mf center *mf* simile

177

182

187

192 *accel.*

197 $\text{♩} = c. 92$ from edge to center **2** *accel.* $\text{♩} = c. 100$ *accel.*

29 Fade Out

The musical score is written for piano and percussion. The piano part begins at measure 151 with a 'Freely' tempo marking and a tempo of approximately 80 beats per minute. It features a melodic line with notes and rests, and dynamic markings such as *pp* and *mf p*. A circled number 27 indicates the start of a 'White Noise Reverb' effect that continues through the piece. The percussion part starts at measure 156 with a tempo of approximately 84 beats per minute. It consists of a series of rhythmic patterns, primarily eighth and sixteenth notes, with dynamic markings like *ppp*, *pp*, *mf*, and *mf simile*. A circled number 28 indicates the start of a 'White Noise + Sine Wave' effect. The score concludes at measure 197 with a tempo of approximately 92 beats per minute, marked 'from edge to center', followed by a double bar line and a circled number 29 indicating a 'Fade Out' effect. The final tempo is approximately 100 beats per minute, marked 'accel.'.

Ocaso de Tenochtitlan

201 $\text{♩} = c. 108$ *accel-* ----- $\text{♩} = 120$

mp *mf* **30** Electronic Pulse (continues)

207 **3**

217 **31** Filter *mf* (continues)

226 *f* *mf*

234 *f* *mf* *f* *mf*

241 *f* *mf*

248 *f* *mf*

255 *f* *mf*

32 Electronic Rhythm *f* *mf*

262

Ocaso de Tenochtitlan

268

Musical notation for measures 268-272. The system consists of two staves. The upper staff is in treble clef with a key signature of one flat and a 4/4 time signature. It begins with a dynamic marking of *f*. The lower staff is in bass clef. The music features a mix of eighth and sixteenth notes, with some rests and accents.

273

Musical notation for measures 273-278. The system consists of two staves. The upper staff is in treble clef with a key signature of one flat and a 4/4 time signature. It features a mix of eighth and sixteenth notes. The lower staff is in bass clef. A 6/8 time signature change occurs at the beginning of measure 278.

279

Musical notation for measures 279-283. The system consists of two staves. The upper staff is in treble clef with a key signature of one flat and a 4/4 time signature. It features a mix of eighth and sixteenth notes. The lower staff is in bass clef. Annotations include "Random High Notes" pointing to a group of notes in measure 280, "Pulse" pointing to a note in measure 281, "Harmonics Gliss" pointing to a note in measure 283, and "niente" pointing to a note in measure 283.