

# UNIVERSITAT DE BARCELONA

#### Trabajo de fin de máster

#### MÁSTER EN ARTE SONORO

FACULTAD DE BELLAS ARTES

#### Synthi GME Modular Emulator

Una propuesta de emulación digital del sintetizador EMS Synthi 100 del Gabinete de Música Electroacústica de Cuenca

Autor:
Carlos Arturo Guerra Parra

Director:
Dr. José Manuel Berenguer Alarcón

13 de mayo de 2020

Con la colaboración de:





A mi hija, Ana Magdalena, que lo mismo toca el clave que el theremín.

#### **Abstract**

The present paper describes the development of the "Synthi GME" software, which digitally emulates the modular synthesizer EMS Synthi 100 located at the Gabinete de Música Electroacústica de Cuenca. This software is implemented using the SuperCollider language (sclang). It is meant to be an educational tool for the dissemination and discovery of the analog synthesizer, which is currently being restored and which fills a significant gap in the history of electroacoustic music around the world. Released with a free software license, it is accessible to all for musical or educational purposes. The dissertation provides a detailed description of how each emulated module functions, as well as each of the design decisions made throughout the process.

 $\textbf{\textit{Keywords}} - \text{synthesizer, emulator, electroacoustics, SuperCollider, computer music}$ 

#### Resumen

El presente trabajo describe el desarrollo del software «Synthi GME», un emulador digital del sintetizador modular EMS Synthi 100 del Gabinete de Música Electroacústica de Cuenca. Este software está implementado en el lenguaje de SuperCollider (sclang), y su destino es el de ser una herramienta pedagógica para la difusión y el conocimiento de este sintetizador analógico, en actual proceso de restauración y con un importante hueco en la historia de la música electroacústica mundial. Lanzado bajo una licencia de software libre, está a disposición de la comunidad para cualquier propósito musical o de estudio. En la memoria se recoge una descripción pormenorizada del funcionamiento de los módulos emulados, así como cada una de las decisiones de diseño tomadas.

Palabras clave— sintetizador, emulador, electroacústica, SuperCollider, música con ordenador

## Índice general

Ín	ndice de figuras		
Ín	dice	de cuadros	XII
1.	Intr	roducción	1
	1.1.	El sonido, la herramienta y el gesto	1
	1.2.	Esta memoria y Synthi GME	5
	1.3.	Objetivos	6
		1.3.1. Objetivos generales	6
		1.3.2. Objetivos específicos	7
	1.4.	Mi relación con el Synthi 100	8
2.	El C	Gabinete de Música Electroacústica de Cuenca (GME)	9
3.	Ant	secedentes del proyecto	15
	3.1.	Lenguajes de programación sonora	15
	3.2.	Conectando cables virtuales	16
	3.3.	La era de los <i>plugins</i>	17
	3.4.	Synthi GME, ¿un emulador más?	17
4.	Met	todología	19
	4.1.	El proyecto en su conjunto	19
		4.1.1. Toma de contacto con los responsables del GME en la	
		universidad de Castilla–La Mancha	19
		4.1.2. Investigación sobre los sintetizadores de EMS	20

		4.1.3.	Inicio del desarrollo de la aplicación	20
		4.1.4.	Fotografiado exhaustivo del Synthi 100 en el GME	20
		4.1.5.	Sesiones de pruebas y testeos en el GME	21
	4.2.	La pro	ogramación	22
<b>5.</b>	Des	cripció	on de la aplicación	24
	5.1.	Caract	terísticas de SuperCollider	24
		5.1.1.	Orden de ejecución en el servidor	24
		5.1.2.	Audio rate y control rate	26
		5.1.3.	Los Synths ocupan recursos	27
		5.1.4.	Aliasing	28
	5.2.	Estruc	etura de Synthi GME	29
	5.3.	GUI d	le SuperCollider	30
	5.4.	Interfa	az OSC	32
	5.5.	Oscillo	ators	36
		5.5.1.	El módulo <i>Oscillator</i> en el Synthi 100	37
			5.5.1.1. Los diales del módulo	38
			5.5.1.2. Entradas y salidas del módulo	39
		5.5.2.	Los osciladores de la aplicación Synthi GME	39
	5.6.	Filters	3	40
		5.6.1.	Tipos de filtros	40
		5.6.2.	Los filtros resonantes de Synthi 100	42
		5.6.3.	Implementación en Supercollider	43
	5.7.	Octave	e Filter Bank	43
		5.7.1.	Implementación en Synthi GME	44
	5.8.	Envelo	ope Shapers	45
		5.8.1.	Las fases de una envolvente típica	45
		5.8.2.	Control de la envolvente: Gate y trigger	47
		5.8.3.	Otros tipos de envolventes	48
		5.8.4.	$Envelope\ Shaper\ en\ Synthi\ 100\ \dots\dots\dots$	48
			5.8.4.1. Del generador de trapezoides del Synthi 100	
			al $ADSR$ del Synthi 100 del GME	48
			5 8 4 2 Decisiones de implementación en SuperCollider	49

_		
INDICE	CENERAL	

#### ÍNDICE GENERAL

	5.8.4.3. Diales de salida	52
	5.9. Noise Generators	52
	5.10. <i>Echo</i> A. D. L	53
	5.10.1. Implementación en Synthi GME	54
	5.11. Entradas y salidas	55
	5.12. Input Amplifier Level	58
	5.13. Output Channels	59
	5.13.1. Los parámetros del módulo	59
	5.13.2. Salidas y <i>busses</i> internos	59
	5.13.3. Implementación en SuperCollider	60
	5.14. Ring Modulators	60
	5.15. Random C. V. Generator	62
	5.15.1. La aleatoriedad en el Synthi $100$	62
	5.15.2. Los diales del módulo	62
	5.15.3. Las salidas de voltaje	64
	5.15.4. Implementación en Synthi GME	64
	5.16. Slew Limiters	64
	5.16.1. Implementación en Synthi GME	65
	5.17. Matrices de conexiones	65
	5.17.1. Implementación en Synthi GME	66
	$5.17.1.1.$ Diseño de las matrices en el $\emph{GUI}$ de Synthi	
	GME	70
<b>6.</b>	Futuro de la aplicación	73
	6.1. <i>«To Do»</i>	73
	6.1.1. Sistema de versionado	75
	6.2. Continuación del desarrollo	76
	6.2.1. Mejoras en el código	76
	6.2.2. Interacción con el usuario	77
	6.2.3. Difusión, formación e información	77
7.	Conclusiones	78
8	Agradecimientos	79

ÍNDICE	CONTRACT
INDICH	GENERAL

#### ÍNDICE GENERAL

Anexos	80
A. Instalación y ejecución de Synthi GME	
A.1. Requisitos	81
A.2. Instalación del quark	82
A.2.1. Instalación con la interfaz gráfica de Quark	82
A.2.2. Instalación con código	83
A.3. Ejecutar Synthi GME	84
B. Synthi 100 Sheet Data	87
C. Folleto del EMS Synthi 100	89
Bibliografía	99

## Índice de figuras

2.1.	Aspecto del GME de Cuenca en 1983	10
2.2.	Doble teclado EMS en Cuenca	11
2.3.	El Synthi 100 en el GME recién creado en 1983	12
2.4.	«Contador» de frecuencia	13
2.5.	Synthi 100 en proceso de restauración	14
2.6.	El Synthi 100 en las instalaciones actuales del GME	14
3.1.	Patch en Pure Data	16
3.2.	Arturia Synthi V	18
4.1.	Aplicación Synthi GME ejecutándose junto al Synthi 100 en el	
	GME	21
5.1.	Orden de ejecución en el servidor de SuperCollider	25
5.2.	Gráfica de aliasing	28
5.3.	Árbol de directorios de Synthi GME	30
5.4.	Ventanas de Synthi GME	31
5.5.	Páginas de ejemplo del $\mathit{layout}$ creado en $\mathit{TouchOSC}$ para Synthi	
	GME	34
5.6.	Ondas periódicas simples	37
5.7.	Osciladores del Synthi 100 del GME	38
5.8.	Tipos de filtro	41
5.9.	Dos de los 8 filtros del Synthi 100 del GME	42
5.10.	Módulo Octave Filter Bank	44
5.11.	. Esquema de una envolvente tipo ADSR	46

#### ÍNDICE DE FIGURAS

5.12. Envolvente del Synthi VCS3
5.13. Las envolventes del Synthi 100 del GME
5.14. Uno de los dos generadores de ruido del Synthi 100 del GME. 53
5.15. <i>Echo</i> A. D. L
5.16. Entradas y salidas al sistema
5.17. Algunas salidas XLR del Synthi 100
5.18. Input Amplifier Level
5.19. Output Channels
5.20. Ring Modulators
5.21. Random Voltage Control Generator
5.22. Slew Limiters
5.23. Dope Sheet conservado en el GME 67
5.24. Tipico cableado de un sintetizador modular
5.25. Matriz Audio Control del Synthi 100 del GME 69
5.26. Matriz <i>Voltage Control</i> de la aplicación Synthi GME 71
5.27. Matriz Audio Control de la aplicación Synthi GME
6.1. Módulos y nodos de <i>audio</i> en funcionamiento
6.2. Módulos y nodos de <i>control</i> en funcionamiento
A.1. Instalación gráfica del quark Synthigme

### Índice de cuadros

5.1.	Atajos de teclado	32
5.2.	Estructura de los mensajes $OSC$	36
5.3.	Comparación de envolventes del Synthi 100	52
5.4.	Entradas y salidas de la aplicación	56

#### 1 | Introducción

#### 1.1. El sonido, la herramienta y el gesto

No es infrecuente encontrar folletos publicitarios de sintetizadores de las décadas de los 70 y 80 en los que se resalta la capacidad ilimitada de creación sonora de estos artefactos. En la introducción del manual de uso de Synthi AKS (Grogono, 1972), por ejemplo, se alaba especialmente el hecho de que el sintetizador, a diferencia de otros instrumentos, no tiene un sonido característico. Esta idea de universalidad del sonido sintetizado también se puede encontrar en revistas especializadas. Así, Pérez-Arroyo describía el funcionamiento del sintetizador como «capaz de producir cualquier tipo de sonido existente o no, por medios electrónicos» (Pérez-Arroyo, 1984).

Pero igual que reconocemos un sonido producido con un instrumento clásico, no tengo duda de que también podemos reconocer los sonidos procedentes de un sintetizador. A pesar de ser ilimitadas las posibilidades y de no tener un timbre característico al modo de un instrumento clásico, existen ciertos indicios reconocibles: sonidos puros, frecuencias moduladas, formas «matemáticas» de los glissandos, reverberaciones «artificiales», etc., incluso cuando sus combinaciones llegan a gran complejidad. De hecho, me aventuro a decir que no es más ilimitado el abanico de sonidos que se pueden arrancar de un sintetizador respecto al que se puede extraer de una orquesta clásica. Esta discusión me trae a la mente la cuestión matemática de que aunque el conjunto de los números impares es mayor que el de los números primos, en ambos casos se trata de conjuntos infinitos. Y el hecho de que uno sea más grande no nos permite suponer que uno sea superconjunto de otro. De hecho

no lo son. Algo análogo ocurre, a mi modo de ver, con los sonidos sintetizados respecto a los acústicos. Cada uno tiene sus propios patrones reconocibles, sus señas de identidad sonora. En ambos casos las posibilidades son infinitas, puede que mayores los del sonido sintetizado electrónicamente, pero ello no le confiere el derecho de erigirse en ningún caso en una generalización del sonido de origen físico. Aquel pensamiento respondía más a un cambio de paradigma que a una realidad comprobada.

El desarrollo de las computadoras en el mundo sonoro parecía prometer una definitiva desencarnación del sonido respecto a las limitaciones. Este optimismo no se circunscribe a los inicios de los ordenadores —que bañaba a otros muchos campos más, como la inteligencia artificial—, sino que sigue siendo común en nuestros días. No hace muchos años, un profesor mío de música electroacústica me mostraba el potencial de CSound, como si en ese archivo en blanco antes de escribir código alguno, estuvieran en potencia ante nosotros «los infinitos sonidos posibles». Esta idea de infinitud me cautivó y confieso que me pareció creíble y sugerente durante mucho tiempo. Era algo así como estar ante la Biblioteca de Babel<sup>1</sup> del sonido. La lectura de este libro siempre ha hecho que mi imaginación vuele hacia otra biblioteca imaginaria, la del sonido digital. No hay más que delimitar el formato de los archivos de esta biblioteca y dejar que existan todas las combinaciones posibles de unos y ceros en un disco duro ideal. ¡Esto sí que sería tener posibilidades sonoras delante! El problema es que dejar abiertas todas las posibilidades arroja principalmente «ruido». Revisar al azar un libro de la biblioteca de Babel no es muy diferente que pedir a un ordenador que nos entregue un libro con letras al azar. No importa las veces que hagamos click para ver un libro nuevo, con toda seguridad encontraremos letras sin sentido. Lo mismo ocurre cuando generamos un segundo de audio con bits al azar. Cualquiera que se dedique al audio digital sabe que el resultado será «ruido blanco», y en caso negativo, se debe, con toda probabilidad, a algún fallo en el algoritmo. Un sintetizador digital no tiene ante sí todas las posibilidades

<sup>&</sup>lt;sup>1</sup>La Bioblioteca de Babel es un afamado libro de José Luis Borges, en el que plantea la idea de infinitud unida a una biblioteca imaginaria en la que existen simultáneamente todos los libros que es posible escribir con todas las combinaciones del alfabeto y los signos de puntuación.

ni mucho menos. Ni siquiera el ruido que genera es absolutamente azaroso. Todo lo que puede hacer un ordenador a nivel sonoro, es tan reconocible, como lo es el sonido creado con un sintetizador o el creado con una orquesta sinfónica. Solo nuestra particular Biblioteca de Babel sonora contiene todos los sonidos posibles, pero, precisamente por ser tantos, nos es infinitamente más práctico «crearlo» que buscarlo.

Otro elemento que considero esencial, junto a la «herramienta», en toda creación es el «gesto». Lo que un artista puede crear está determinado fuertemente por lo que el instrumento puede hacer potencialmente, por una parte, y del gesto humano, por otra. «Herramienta» y «gesto» son inseparables en la creación de cualquier arte, y el sonoro —en el sentido más amplio posible: sonido, música, ruido...— no es una excepción. Es, sin duda, muy tentador pensar que las herramientas más modernas son las que definitivamente están desencarnadas de todo gesto, que son potencialmente ilimitadas y que engloban a las ya obsoletas. Pero un producto sonoro (producido con intencionalidad humana) sin gesto no existe; siempre lleva consigo la impronta del gesto productor. Ningún sintetizador analógico consiguió imitar a la perfección un instrumento musical clásico. Este hecho se puede apreciar muy claramente en la evolución de los órganos electrónicos. En mi experiencia como organista, he podido tocar todo tipo de instrumentos electrónicos que prometen un sonido idéntico al de un órgano de tubos, y hasta la fecha no he encontrado ninguna solución que sea totalmente convincente. Finalmente, solo los órganos que usan alguna suerte de sampleado en sus sonidos son los que más se aproximan a los salidos de tubos de órgano. No quiero entrar en una discusión sin fin de si una muestra sonora es un sonido digital. Desde luego que bajo algún aspecto sí que lo es, pero descarto este tipo de sonido por ser una copia obvia de otro sonido generado físicamente. De hecho, basta procesar estos samples, aplicarles ciertas técnicas propias de la música concreta o electroacústica para que, de nuevo, nuestros oídos reconozcan que el producto ha salido de un ordenador o que ha pasado por procesos de cinta magnética. Nuestra mente «oye» el gesto y lo entiende, porque el gesto es humano. Oímos un violín y parece que vemos las arcadas, oímos un cantante y el fraseo nos recuerda el gesto de respirar...

Pero a veces el gesto no es obvio. Oímos sonidos y no los podemos asociar a su fuente, ni comprender la intención del que los genera, porque no «vemos» el gesto, nos es imposible identificarlo. No me estoy refiriendo con esto al sonido «desencarnado» del «objeto sonoro» de Pierre Schaeffer, que le sirve de ladrillo para crear, sino, más bien, al producto sonoro ya terminado. No cabe duda de que la música de índole electrónica o digital puede desconcertar o sorprender al oyente ya que no se le comunica siempre de forma obvia el gesto que la ha producido. Pero el gesto ha estado ahí: la escritura de un código, el deslizamiento de unos *sliders*, o el giro de una perilla. De hecho, basta escuchar varias obras que han utilizado la misma herramienta, con sus gestos asociados, para que nuestra mente aprenda a reconocerla.

Las posibilidades siempre vendrán determinadas por dicho gesto y por lo que la herramienta —sea software, sea sintetizador— nos permita. Aquella infinitud que se me prometía en el estudio de Csound, representa más un cambio de paradigma que la panacea de la creación sonora. Solo cambia el gesto, que ya no es el de un arco de un violín o una perilla de un sintetizador. Ahora puede ser el tecleo de código, el movimiento del ratón o de cualquier dispositivo que se nos ocurra.

El corolario de toda argumentación es que tan burda es la imitación de un órgano o un violín por un sintetizador analógico como lo es la de este por medios computacionales. Quizás el resultado sonoro en este segundo caso sea más convincente que en el primero, pero no hay que olvidar que el gesto ha cambiado indefectiblemente. No es lo mismo mover el ratón que estirar el brazo para girar una perilla al tiempo que se conecta con la otra mano la salida de un modulo. Es claro que no podemos hacer lo mismo con un sintetizador que con su emulación en una pantalla, pero aun así siempre han existido emulaciones: sintetizadores que buscan imitar sonidos de instrumentos acústicos, sintetizadores digitales que emulan a sintetizadores analógicos, sintetizadores digitales que emulan a instrumentos acústicos. Unas veces son un curioso juguete, otras una forma pretendida de «sustituir» a lo emulado. Se puede imitar la herramienta, pero el gesto ya será otro.

En todo caso, veo las emulaciones como una ocasión para descubrir la originalidad de cada técnica y cada fuente, su esencia y su valor artístico.

Vuelvo a mi oficio de organista; me encantan los emuladores de órganos, los uso y me interesa su funcionamiento. Pero nunca he sentido que ya no fuese necesario mantener los caros y delicados instrumentos acústicos. Muy al contrario, la emulación me remite a lo emulado y cuanto más se consigue el propósito de imitación, más valoro lo imitado y su originalidad.

Esta es la razón de fondo de este trabajo. Emular un sintetizador analógico supone todo un reto multidisciplinar: acústica, psicoacústica, síntesis analógica, electrónica, computación, sonido digital, historia, interfaz y su gesto, etc. En el proceso aparecen indefectiblemente conceptos triviales para un sintetizador analógico que son muy complejos para uno digital, y viceversa, lo cual redunda en la idea de que cada disciplina es independiente y que una no engloba a la otra.

Difícilmente voy a reinventar la rueda diseñando un emulador —ni lo intento—, pero confío en que si pongo el conocimiento del proceso, su código y la posibilidad de uso, en manos de la comunidad, quizás el trabajo adquiera entonces sentido y pueda despertar el interés de otras personas, como lo hizo conmigo, por la síntesis analógica y los instrumentos «históricos», que en ocasiones se han visto como chatarra inservible. Diseñar un dispositivo como este, objeto del trabajo, es una forma muy gratificante de entender por qué un sintetizador es como es, por qué tiene unas características y no otras, por qué fue diseñado con unos controles y no con otros que quizás esperaríamos, cuáles son sus posibilidades y, en definitiva, el porqué de la herramienta y sus gestos inherentes.

#### 1.2. Esta memoria y Synthi GME

El presente texto recoge el proceso de creación de un emulador digital del sintetizador modular EMS Synthi 100 del Gabinete de Música Electroacústica de Cuenca. La memoria consta principalmete de una exhaustiva descripción de los módulos implementados, la decisiones tomadas en cada caso, así como la explicación de su funcionamiento para el usuario.

El dispositivo de software se ha programado para PC, en el lenguaje de SuperCollider. Se ha diseñado con un fin principalmente pedagógico y por ello ha sido licenciado bajo los términos de GNU GPLv3 (2016).

La aplicación se llama «Synthi GME», nombre recursivo siguiendo la tradición de muchos programas de software libre (*GNU*, *KDE*, etc.). En él, «GME» son las siglas de «GME Modular Emulator», donde «GME» vuelve a significar de nuevo «GME Modular Emulator», y así *ad infinitum*, haciendo un guiño al Gabinete de Música Electroacústica, que es lo que significa *GME* en su última iteración...

#### 1.3. Objetivos

Por claridad, he dividido los objetivos del trabajo en dos secciones. En la primera, los *objetivos generales* describen las metas y motivaciones más globales y remotas que hay tras el proyecto. La segunda seccion, los *objetivos específicos*, enumera los objetivos más inmediatos y concretos en la elaboración de este proyecto.

#### 1.3.1. Objetivos generales

- 1. Promover el conocimiento de la síntesis sonora analógica, como medio para comprender las nuevas técnicas digitales.
- Revalorizar y dar a conocer el patrimonio del arte sonoro español, tangible e intangible, en muchos casos relegado a su almacenamiento, con vistas a fomentar su restauración y puesta a punto y publicación.
- 3. Fomentar el uso del software libre tanto en el ámbito técnico como artístico, como medio idóneo de transmisión y creación de conocimiento.
- 4. Contribuir al desarrollo de herramientas pedagógicas libres y colaborativas en el ámbito universitario y profesional.

- 5. Poner a disposición de la comunidad educativa en general, y del GME de Cuenca en particular, una herramienta didáctica y de promoción del patrimonio instrumental del arte sonoro.
- 6. Alentar la formación multidisciplinar y multicompetencial en el ámbito de la creación artística.

#### 1.3.2. Objetivos específicos

En cuanto a la creación de la aplicación informática, se destacan los siguientes objetivos:

- 1. Diseñar y crear un programa informático que emule la interfaz y el funcionamiento del sintetizador EMS Synthi 100 del Gabinete de Música Electroacústica de Cuenca.
- 2. Poner el foco en el uso pedagógico y artístico del dispositivo de software.
- 3. Conferirle una arquitectura escalable con vistas a constantes mejoras y ampliaciones.
- 4. Ser publicada como software libre en los medios habituales de este momento.
- 5. Implementar una interfaz de usuario abierta al uso de otros dispositivos tanto de software como de hardware.

La presente memoria sobre el diseño y programación de esta herramienta de software tiene por objetivos:

- 6. Ofrecer una descripción del funcionamiento de los principales módulos del sintetizador, atendiendo tanto a las similitudes como a las diferencias con versiones anteriores del mismo modelo.
- 7. Servir de manual de la aplicación informática, sin perjuicio de la creación de los archivos de ayuda propios de la misma.

- 8. Recoger información y contribuir a la escasa documentación sobre el funcionamiento del EMS Synthi 100 del GME.
- 6. Poner a disposición de la comunidad educativa en general, y del GME de Cuenca en particular, una herramienta didáctica y de promoción del patrimonio instrumental del arte sonoro.

#### 1.4. Mi relación con el Synthi 100 del GME de Cuenca

La primera vez que oí hablar del Synthi 100 del Gabinete de Música Electroacústica de mi ciudad fue en una grata conversación con el compositor Julio Sanz Vázquez en enero de 2019. No solo me hizo un magistral resumen de la historia del gabinete y de las personas más importantes con él relacionadas, sino que me habló del elemento estrella del estudio, su flamante sintetizador. En menos de un año tras la conversación ya pude comenzar a realizar el presente trabajo, momento que coincidiría precisamente con los trabajos de restauración que la Universidad de Castilla—La Mancha en Cuenca, comenzaba en ese momento. En este contexto mi trabajo adquiría para mí un especial interés, ya que podría servir como un elemento divulgador y pedagógico en torno a la recuperación del Synthi.

Durante los meses en los que me he dedicado a la investigación e implementación de este emulador, he contado con el apoyo de la universidad, que me ha abierto de par en par las puertas del GME y me ha permitido llegar hasta las perillas del Synthi 100. Por desgracia, la crisis que se está viviendo en España durante los últimos meses ha paralizado todo este contacto, por otra parte, fundamental, con el sintetizador. No obstante, ha habido suficiente tiempo como para hacer un exhaustivo reportaje fotográfico del Synthi 100, probar y experimentar con ciertos módulos, y preguntar muchas dudas ante la ausencia de manuales detallados de la época.

### 2 | El Gabinete de Música Electroacústica de Cuenca (GME)

El GME aparece en Cuenca integrado en todo un planteamiento de apuesta por el arte en general y el arte contemporáneo en particular. Este nace con la intención de poner en el panorama nacional e internacional no solo un conjunto de herramientas tecnológicas a disposición de los compositores, sino, con una clara dirección pedagógica y formativa, ser el primer centro público de enseñanza, creación, investigación y difusión de la Música Electroacústica en España (Molina, Osona, Sanz Vázquez, del Saz, y Alcázar, 2018, p. 317). Fue inagurado el 23 de abril de 1983, situándose en el contexto español con personalidad propia respecto a otros grandes centros de esta música coetáneos como el estudio Phonos de Barcelona, el Laboratorio de Música Electrónica de la Escuela de Música Jesús Guridi de Vitoria o el Laboratorio de Informática y Electrónica Musical del Centro para la Difusión de la Música Contemporánea (LIEM-CDMC), por citar unos de los más emblemáticos (Supper, 2012).

Fue inaugurado en 1983, dirigido por Pablo López de Osaba, director por aquel entonces del Conservatorio Profesional de Música, el Museo de Arte Abstracto Español y el festival de la Semana de Música Religiosa de Cuenca. El GME nació sin compositor asociado, siendo su primer técnico Leopoldo Amigo (entre los años 1983 y 1990). En 1989 se contrata a Gabriel Brnčić como profesor de «Teoría de la Composición» y «Teoría y Práctica de la Música con Medios Electroacústicos e Informáticos». Su labor se prolongará

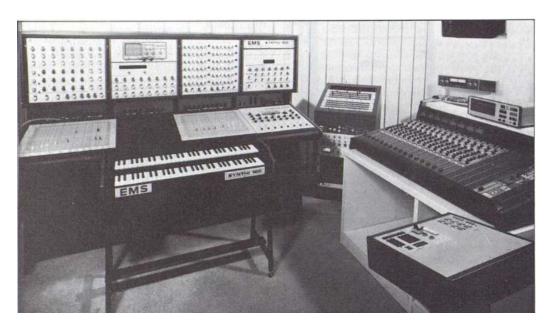


Figura 2.1: Aspecto del GME de Cuenca en 1983.

hasta 1998. Entre 1990 y 2006, el trabajo de Leopoldo Amigo será relevado por el del compositor Julio Sanz Vázquez.

Los años en los que Gabriel ejerció su labor musical y magisterio coincidieron con los de mayor florecimiento de esta institución. Se organizaron los primeros cursos desde la oficialidad de un Conservatorio Profesional en España especializados en música electroacústica, así como conciertos mensuales en diversos espacios conquenses. Por sus instalaciones han pasado los más importantes compositores del momento, tanto por invitación como por iniciativa propia, para crear, dar conferencias o estrenar obras.

Tras la escisión del contrato de Gabriel Brnčić en 1998, el GME entró en un periodo de decadencia institucional (por motivos que no son objeto de este trabajo) que lo haría desaparecer a pesar de los esfuerzos llevados a cabo por su técnico Julio Sanz. Desde septiembre de 2006, momento en el que sus instalaciones fueron desmanteladas, un gran trabajo de recuperación y catalogación llega hasta nuestros días de la mano de la asociación AVADI (Audio Vídeo Arte Digital Interactivo), que nace con el objetivo de preservar el ingente material creado en el GME y luchar por la continuidad de la institución de cara al futuro.



Figura 2.2: Doble teclado EMS en Cuenca.

El elemento estrella del GME fue desde el principio el sintetizador Synthi 100 de *Electronic Music Studios* (1998). Este modelo, heredero tecnológico del VCS3, alcanzó mucha popularidad en la década de los 70 y 80. Se caracteríza por la gran consola que lo alberga, compuesta de 4 paneles verticales y 3 horizontales. Va acompañado de un doble teclado (fig. 2.2), si bien, según narra su técnico, Julio Sanz, apenas ha sido usado históricamente, ya que los compositores han utilizado el sintetizador principalmente para «diseñar» con él materiales sonoros que serían utilizados posteriormente en las mezclas y otros procesados.

En él, todo es conectable con todo por medio de dos matrices de conexiones activadas por pines<sup>1</sup>. Llama la atención la gran cantidad de perillas que alberga, todas ellas numeradas con 10 niveles, quedando ocultos al usuario datos como el voltaje, frecuencia, amplitud, y otros parámetros a los que actualmente estamos acostumbrados a conocer de inmediato. La única forma de saber estos datos es analizándolos en el espectroscopio y el «contador» de frecuencia (fig. 2.4), pero estos módulos no dejan de ser «analizadores» de las señales que a ellos se les dirigen. No es posible conocer todos los

<sup>&</sup>lt;sup>1</sup>Sobre esto me extenderé más a la hora de describir la emulación de estas matrices en 5.17.



Figura 2.3: El Synthi 100 en el GME recién creado en 1983.

datos de todos los módulos simultáneamente. Estas características han contribuido a crear un cierto halo de misterio alrededor del uso de este tipo de sintetizadores, pareciendo mágico el manejo de estos ante el espectador. Un inconveniente claro de este sistema es la dificultad en repetir un mismo resultado en dos momentos distintos. La naturaleza caótica de muchos de los procesos que implican la creación sonora hacen que pequeñas desviaciones de los valores deseados —incluso aquellas procedentes de agentes ajenos al Synthi y al usuario, como la temperatura o la humedad— desemboquen en resultados sonoros completamente distintos.

El sintetizador Synthi 100 está, en los momentos en los que se escribe esta memoria, en proceso de restauración, gracias al apoyo del Patronato Universitario Cardenal Gil de Albornoz (fig. 2.5). Con su recuperación se devuelve a la comunidad un instrumento histórico de un valor imponderable y que forma parte privilegiada de la historia del arte sonoro español. Actualmente, el GME está acogido por la Universidad de Castilla—La Mancha en

#### 2. EL GABINETE DE MÚSICA ELECTROACÚSTICA DE CUENCA



Figura 2.4: «Contador» de frecuencia.

Cuenca. Así, continua su andadura como la comenzó, en un centro público y disposición de todos.



Figura 2.5: Synthi 100 en proceso de restauración. (Foto: Fuzzy Gab 4, 2020)

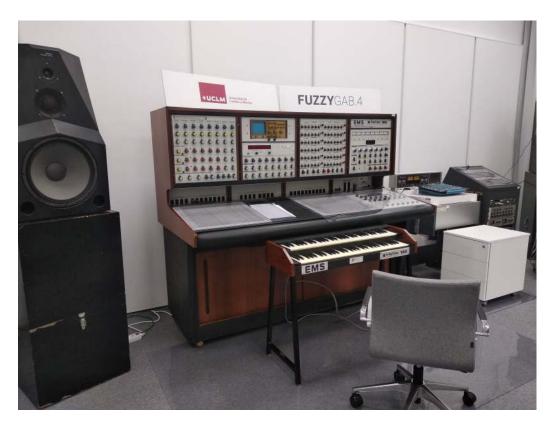


Figura 2.6: El Synthi 100 en las instalaciones actuales del GME, en proceso de restauración.

#### 3 | Antecedentes del proyecto

#### 3.1. Lenguajes de programación sonora

La emulación digital de dispositivos analógicos es tan antigua como la misma computación. En concreto, el diseño de programas y lenguajes de programación expresamente destinados a la síntesis sonora tiene sus raíces en los años 60 del pasado siglo, con lenguajes como Music 1, Music 2, Music 3 y Music 4, predecesores de CSound (Boulanger, 2000, p. xxvii) que, a pesar de su aspecto primitivo, ha llegado hasta nuestros días y sigue siendo una referencia en el ámbito de la programación sonora. Aunque este lenguaje fue pensado para «compilar» los archivos en uno de audio como salida, actualmente es posible usarse en tiempo real gracias al crecimiento del poder de los ordenadores actuales. Precisamente es SuperCollider (s.f.) uno de los lenguajes que podemos denominar «modernos» (Wilson, Cottle, y Collins, 2011, p. ix), ya que nace para ser ejecutado en tiempo real, además de asimilar herramientas de programación de más alto nivel que CSound, como la «orientación a objetos», así como la separación entre lenguaje, intérprete y servidor de sonido, que le confieren una gran versatilidad.

Existen otros muchos lenguajes pensados para la síntesis sonora, como *Common Music, Kyma, Nyquist*, and *Patchwork*, entre otros, pero son *CSound y SuperCollider* los que han ejercido en mí un gran influjo en la forma de pensar la programación sonora. Algo que no deja de sorprenderme es que estos lenguajes, a pesar de tener un status y coherencia interna propios, usan explícitamente conceptos que provienen de la síntesis analógica realizada con sintetizadores. Sus unidades fundamentales de generación de

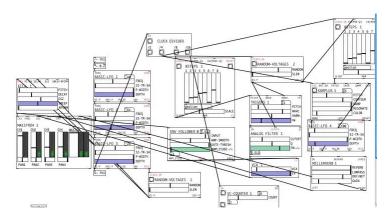


Figura 3.1: Patch de cierta complejidad diseñado en Pure Data.

señal (Opcode en Csound y UGen en SuperCollider) utilizan terminología claramente heredada del mundo analógico: oscilador, envolvente, filtro, control, modulación, etc. Si bien, se introducen otros conceptos que solo pueden ser propios de la síntesis digital por su naturaleza computacional: tabla de onda, aliasing, tasa de audio o de control, etc.

#### 3.2. Conectando cables virtuales

Existen otros sistemas digitales de generación y tratamiento de audio que heredan características propias e interesantes de los sintetizadores modulares y que actualmente tienen una salud de hierro. Me refiero al paradigma de programas como Pure Data (s.f.) o MAX (s.f.). No cabe duda que su principal acierto ha sido el de tomar el concepto de patching de la síntesis modular como elemento básico de su funcionamiento. Se basan en una interfáz gráfica en la que todo son objetos que el usuario une con cables virtuales a golpe de ratón. A diferencia de los lenguajes de programación basados en texto plano, estos sintetizadores digitales tienen un aspecto visual intuitivo y aún más próximo a lo que significa «tocar» un dispositivo analógico.

Las generaciones más modernas de músicos y artistas sonoros nos hemos educado principalmente con este tipo de software, y nuestro contacto con la síntesis analógica ha sido aprendida a través de estas herramientas, que «imitan» el funcionamiento de los viejos sintetizadores.

#### 3.3. La era de los *plugins*

Un grupo de programas que han ejercido un influjo más concreto en la creación de Synthi GME son los plugins de las modernas DAW (Digital Audio Workstation). Los plugins son unidades de generación de señal o de efectos que se aplican arbitrariamente a las pistas de sonido de las «estaciones de audio digital». Típicamente traen una interfaz gráfica que suele imitar con detalle todo tipo de dispositivos analógicos. Existe todo un mercado en torno a estos productos, que vienen a sustituir en muchos casos el uso de aparatos electrónicos, con diseños de una gran fidelidad a los originales. En la interminable lista de plugins¹ que podemos encontrar por la red, tienen un hueco especial aquellos que emulan sintetizadores de todo tipo.

Existen, por supuesto, emuladores de productos míticos de EMS. Entre los productos de uso libre que podemos encontrar, se puden citar las siguientes emulaciones de EMS Synthi VCS3 y AKS: Cynthia VST, KX-Synth-X16 VST, Synthia 2 o SynthiAKS, VCS5 $^2$ . En este punto caben destacar los esfuerzos por ofrecer productos de pago y de alta calidad por empresas como Ableton o Arturia. Quizás Arturia Synthi V (s.f.) sea el software más completo y elegante del mercado que emula al sintetizador EMS Synthi VCS3.

#### 3.4. Synthi GME, ¿un emulador más?

El proyecto de diseño e implementación de Synthi GME toma características de distintos productos, ninguno de los cuales abarca todas ellas:

Es software libre Synthi GME está licenciado bajo los términos de GNU GPLv3 (2016). Esto es esencial si no se quiere limitar la información a la que puede acceder el usuario, así como su libertad de modificar y usar el software.

<sup>&</sup>lt;sup>1</sup>Como muestra, en esta web se encuentran unos 200 plugins de libre uso, unos con licencias de código abierto y otros simplemente freeware (The 200 Best Free VST Plugins Ever. s.f.).

<sup>&</sup>lt;sup>2</sup>Se pueden encontrar listados todos ellos con links de descarga en *Free EMS VCS3 and Synthi AKS VST emulations*, s.f.



Figura 3.2: Aspecto del *plugin* de *Arturia*, *Synthi V*, emulación del Synthi VCS3.

Emula un sintetizador de grandes dimensiones No es común encontrar un emulador tipo plugin con tantos módulos como el Synthi 100. De hecho, sus dimensiones no están exentas de ciertos problemas de manejabilidad y de visionado, que se han intentado resolver de la manera más sencilla y práctica posible.

Emula un sintetizador concreto Más allá de imitar un modelo concreto de sintetizador, el objetivo está puesto en una unidad concreta, la del Gabinete de Música Electroacústica de Cuenca.

**Tiene un fin pedagógico** Su origen es académico y también su destino. La elaboración de este trabajo ha coincidido con la restauración del Synthi 100 del GME de Cuenca, de forma que este *software* ha sido pensado para ser utilizado tanto para la difusión y formación en torno al GME en la Universidad de Cuenca, como para la educación en general.

Está escrito en lenguaje de SuperCollider sclang es un lenguaje diseñado expresamente para SuperCollider, lleno de características de propósito general que facilitan la escritura de scripts para el diseño de audio. El hecho de ser software libre, su simplicidad y su compilación just in time lo hacen apto para live coding y para la creación de programas realmente complejos, como el caso de Synthi GME. La elección de este lenguaje está muy unida a su destino pedagógico.

# 4 | Descripción metodológica del trabajo

El proceso de trabajo llevado a cabo para el presente proyecto, así como sus tiempos de planificación y ejecución, se exponen aquí desde dos perspectivas: la del proyecto global desde su concepción (investigación, contactos, visitas al GME, etc.), y la del desarrollo del software en sí, que lleva su propio tiempo y calendario.

#### 4.1. El proyecto en su conjunto

Una vez definida la idea general del proyecto, se procedió a dar los siguientes pasos:

## 4.1.1. Toma de contacto con los responsables del GME en la universidad de Castilla-La Mancha

Este era el primer paso a dar tanto desde el punto de vista cronológico como lógico. Las personas con las que puede entrevistar entre octubre y noviembre fueron precisamente las encargadas actualmente del Gabinete, Sylvia Molina Muro<sup>1</sup> y Julio Sanz Vázquez, compositor y técnico del Gabinete. Ambos acogieron con gran entusiasmo el proyecto y pude acceder con ellos a las instalaciones de estudio actual en varias ocasiones.

 $<sup>^{1}</sup>$ Prof. contratado doctor (UCLM). Investigador principal del grupo I+D+I «Fuzzy Gab 4» (2020), encargado de la recuperación y dinamización de los fondos GME en el ITCT (Cuenca–UCLM).

#### 4.1.2. Investigación sobre los sintetizadores de EMS

No se han encontrado manuales de usuario del Synthi 100. Sí que existe un «manual de servicio», cuya finalidad es la de ofrecer datos técnicos para ingenieros en electrónica, no la de ser una guía de uso para músicos o artistas sonoros. Sin embargo, EMS sí que publicó manuales de usuario para modelos de sintetizadores más «domésticos», como el Synthi AKS (1972), Synthi CVS3 (s.f.). Existen, por fortuna, muchas similitudes entre estos sintetizadores y el Synthi 100, a pesar de las diferencias de tamaño. En todos los casos el sistema de conexiones es el de una matriz, por lo que la filosofía de funcionamiento es muy similar entre ellos. Por otra parte, el uso de emuladores ya implementados fue una gran ayuda para comprender el funcionamiento de muchos módulos, más allá de la intuición y de la relación con otros la sección sintetizadores documentados².

#### 4.1.3. Inicio del desarrollo de la aplicación

El desarrollo de la aplicación de software, Synthi GME, objeto de esta memoria, no podía ser pospuesto, por razones de tiempo, al momento de tener toda la información posible acerca del sintetizador, así como el comportamiento particular del Synthi 100 del GME. Al contrario, era imprescindible plantear desde el inicio la estructura de su código, el estilo de programación, etc., de modo que la implementación de los diversos módulos del sintetizador no estuviera fuertemente condicionada por la arquitectura general del programa. La elaboración del programa en SuperCollider fue siempre paralela a la de toda investigación y recogida de información del Synthi 100 o del GME.

#### 4.1.4. Fotografiado exhaustivo del Synthi 100 en el GME

Esta tarea fue crucial para conocer al detalle la interfaz del sintetizador, por una parte, y como base de la interfaz gráfica del programa, por otra. Esta sesión tuvo lugar a mediados de diciembre. Había prevista una nueva sesión para obtener fotografías de mayor calidad en el mes de abril que, lógicamente,

<sup>&</sup>lt;sup>2</sup>Sobre los plugins y diversos emuladores, véase la sección 3.3, pág 17.



Figura 4.1: Aplicación Synthi GME ejecutándose junto al Synthi 100 en el GME, realizando testeos y pruebas en los diferentes módulos implementados. Febrero de 2020.

no pudo llevarse a cabo. Estas nuevas fotografías, que serán realizadas en otro momento en el futuro, estarán destinadas a la interfáz gráfica de usuario.

#### 4.1.5. Sesiones de pruebas y testeos en el GME

En enero de 2020 comenzó a estar operativo el Synthi 100 en su proceso de restauración. En ese mismo mes pude experimentar con el comportamiento de varios de sus módulos (fig. 4.1). Es evidente que este trabajo de campo con el sintetizador se puede traducir en una ingente cantidad de horas de experimentación y registro de materiales sonoros que exceden los límites de este trabajo. En todo caso, se trata de un trabajo que es imprescindible para una futura evolución de la aplicación.

#### 4.2. La programación de Synthi GME

Todo desarrollo de software requiere de un gran esfuerzo organizativo así como una buena estructuración de las diversas etapas. Consumir una de ellas antes de tiempo puede provocar problemas muy difíciles de resolver en el futuro. Desde el inicio, el código ha estado bajo el control de versiones *Git*, y alojado en la plataforma *Github*. Este sistema permite registrar todos los cambios de código realizados momento a momento, con descripciones de las mismas. Además, permite el versionado y es imprescindible para poder ser instalado y utilizado como un paquete, de modo que el usuario no necesita trabajar directamente con el código del programa para poder ejecutarlo.

Las diversas etapas de desarrollo pueden observarse directamente en el repositorio en *Github*, y pueden resumirse en los siguientes puntos:

Establecimiento de una estructura global del código Se trata de un proceso iterativo, en el que es necesario realizar toda una serie de pruebas de concepto. El resultado de dichas pruebas van aportando información muy valiosa a la hora de tomar decisiones que quedarán en la estructura de la aplicación. Una mala arquitectura y jerarquía en las clases, sus métodos, su interfaz, etc., pueden acarrear problemas más adelante, cuya resolución es muy costosa en cuanto a tiempo de trabajo. El primer mes de trabajo fue esencial, en este aspecto, para la continuación exitosa hasta el día de hoy.

Implementación de una interfaz gráfica externa Los primeros elementos de la aplicación no podían manejarse desde ninguna interfaz gráfica. Todo se debía de manejar con código (algo natural, por otra parte, en SuperCollider). En un primer momento se prentendía separar completamente la interfaz gráfica del motor del sonido, de modo que dicha interfaz pudiera ser programada en cualquier lenguaje y en cualquier plataforma y su comunicación con el motor se realizaría íntegramente con mensajes  $OSC^3$ . Pero una vez implementado este protocolo, incluso diseñada en parte una interfaz para móvil y tablet, se vio oportuno

 $<sup>^3</sup>$ Véase la sección 5.4 dedicada al protocolo OSC (pág. 32).

que la aplicación tuviera una interfaz gráfica por defecto en el propio ordenador.

Implementación de la interfaz gráfica en SuperCollider Hasta enero de 2020 aproximadamente, todo el código estaba dirigido a una interfaz OSC, arriba descrita, y ciertos módulos de prueba. La creación de una interfaz gráfica para el propio ordenador fue un punto de inflexión en el desarrollo de la aplicación. El Synthi 100 es demasiado grande como para ser contenido en pantallas móviles, y poco práctico si se plantea como diversas páginas de una aplicación móvil. La pantalla del ordenador permitía ver todos los módulos de un solo vistazo. Su inmediated, por otra parte, agilizó mucho el desarrollo de la aplicación. El lenguaje de programación, por otra parte, es el mismo que el del resto de los módulos, sclang, y, en beneficio de un aspecto más realista, fue posible usar fotografías del Synthi 100 para realizar su diseño.

Implementacion del resto de módulos Una vez que interfaz gráfica y módulos estaban al mismo nivel de desarrollo, la evolución de ambos caminó a la par hasta hoy. Módulo a módulo fue siendo implementado y testeado en su conjunto. Con cierta frecuencia aparecían bugs, por suerte corregidos exitosamente. Esta etapa es la que llega hasta el presente y se proyecta hacia el futuro. El esqueleto de la aplicación está terminado, con una gran parte de sus módulos implementados y funcionando de forma lógica e intercomunicada.

#### 5 Descripción de la aplicación

## 5.1. Características importantes de SuperCollider y sus limitaciones

Synthi GME ha sido escrito enteramente en SuperCollider. Antes de entrar en detalles de su funcionamiento conviene describir brevemente algunas de las características estructurales de SuperCollider que se han tenido muy en cuenta a la hora de programar la aplicación.

#### 5.1.1. Orden de ejecución en el servidor

En un sintetizador modular analógico no hay limitación alguna en el orden en el que se conecta sus módulos. Todos son independientes entre sí ya que se «ejecutan» al mismo tiempo. Cualquier señal de salida puede ser conectada a cualquier entrada. Esta es una de las características más interesantes y sugerentes de estos sintetizadores. En el manual de instrucciones de EMS Synthi VCS3 (*Synthi Users Manual*, s.f.) —antecesor del Synthi 100 y de características muy similares— se invita al usuario a experimentar libremente con cualquier conexión. La posibilidad de que todo pueda ser conectado con todo hace de los sintetizadores modulares en general, y de los Synthi en particular, un banco de experimentación sonora sin precedentes.

Los sistemas informáticos actuales son multihilo, es decir, pueden ejecutar varias tareas a la vez. El tiempo de ejecución es repartido en todas ellas a velocidad suficiente como para crear en el usuario la ilusión de que ocurren las tareas simultáneamente, cuando en realidad solo una se lleva a cabo en un

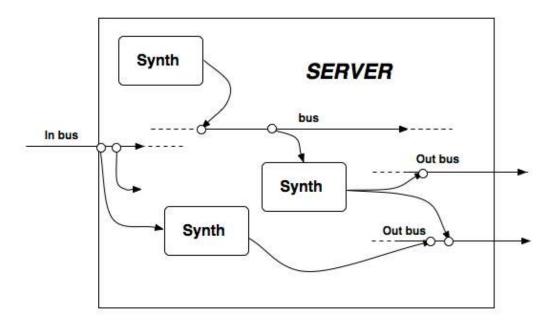


Figura 5.1: Orden de ejecución en el servidor de SuperCollider. Cada ciclo de audio los *Synths* presentes en el servidor son ejecutados en orden (de izquierda a derecha en esta respresentación). Ninguna información puede viajar por los *busses* en sentido contrario, ya que su contenido es eliminado tras la finalización de cada ciclo.

instante concreto. Los sintetizadores digitales aprovechan esta característica, pero sus módulos no se ejecutan simultáneamente, sino que siguen un orden preciso de ejecución entre ellos. En el caso de SuperCollider, cada Synth en el servidor ocupa un lugar en una lista, la cual es ejecutada en orden a la frecuencia de audio (audio rate). Puesto que todos los módulos de Synthi GME están construidos a base de Synths, no será raro que la salida de uno de ellos sea conectada a la entrada de otro que ya ha sido ejecutado en el ciclo presente. El contenido de los busses de audio es eliminado al terminar un ciclo completo, por lo que ninguna señal llegará al módulo de destino. La figura 5.1 representa esta característica de unidireccionalidad en el orden de ejecución de los nodos presentes en el servidor. Ningún Synth podrá enviar señal alguna a otro situado a su izquierda (el orden de ejecución es de izquierda a derecha en esta representación).

Existe un *UGen* que permite obtener información de ciclos pasados de los *busses*. Se trata de InFeedback. La limitación frente a In es que la señal que recibe es la correspondiente a un ciclo de control previo. Aunque en la mayoría de los casos esto no tiene mayor importancia, es una característica importante a tener en cuenta, especialmente cuando se produce un *feedback* en la señal, ya que su comportamiento distará mucho del que puede tener un sistema analógico.

#### 5.1.2. Audio rate y control rate

No todo ocurre a la misma velocidad en SuperCollider. La producción de señales de audio requiere de la creación de un número de muestras por segundo igual a la «tasa de audio». Pero no siempre es necesaria una demanda tan alta de computación. En el caso de los efectos como curvas relativamente lentas en los cambios de los parámetros sonoros pueden realizarse con una tasa mucho menor. Por ejemplo, un glissando frecuencial entre 100 Hz y 1000 Hz de un segundo de duración, no necesita tener una resolución similar a la «tasa de audio»<sup>1</sup>. Es posible realizarlo a una tasa («de control») bastantes veces menor a la de audio sin verse afectado auditivamente el resultado sonoro. Por defecto, la tasa de control en SuperCollider es 64 veces menor que la de audio.

Existe una clara analogía entre una señal de audio analógica y audio rate, así como entre una señal de voltaje y control rate. En una primera aproximación, podría parecer obvio que las señales de voltaje del Synthi 100 pueden ser emuladas digitalmente a la tasa de control. El problema de una decisión así es que limita considerablemente el uso que podemos hacer con esta señales «de control». En un sintetizador analógico, una señal de control de voltaje puede ser utilizada como señal de audio y viceversa. No hay ninguna diferencia real entre ambos tipos de señal, más allá de una división puramente funcional. No hay ninguna razón que nos obligue a limitar las compotentes frecuenciales de una señal de voltaje en un sintetizador analógico. Por esta

<sup>&</sup>lt;sup>1</sup>Sin entrara aquí en detalles, la frecuencia mínima de audio utilizada un sistema digital normal es de 44.1 kHz, la correspondiente a un CD de audio estándar.

razón, Synthi GME prescinde por completo de las señales en «tasa de control». Todas las señales en la aplicación son del mismo tipo, con «tasa de audio». Esto es posible, evidentemente, gracias a la capacidad de computación de los ordenadores modernos. A cambio de eficiencia se gana en versatilidad y, por supuesto, en aproximación al funcionamiento de un sintetizador modular.

#### 5.1.3. Los *Synths* ocupan recursos

Puede resultar una obviedad decir que tanto los *Synths* como los *busses* digitales ocupan memoria y tiempo de ejecución, pero cuando estos elementos se empiezan a contar por miles, como en el caso de Synthi GME, este es un factor que puede comprometer la eficiencia y la escalabilidad de la aplicación.

En el diseño de Synthi GME se ha prestado una especial atención en la eficiencia. Sclang es un lenguaje de alto nivel e interpretado just in time, por lo que necesita de un consumo de recursos de computación mayor que el de lenguajes de bajo nivel y compliados, como ocurre con C o C++. Los Synth son básicamente unas funciones que se ejecutan una vez cada ciclo de audio. Si cada módulo tiene entre uno y cinco Synths y otro cada nodo de las matrices, contaríamos por varios miles los Synths que se ejecutarían al mismo tiempo, lo cual comprometería el correcto funcionamiento del sistema en su conjunto.

Para que esto no ocurra, todos los *Synths* están «pausados» por defecto. Solo se ejecutan cuando cumplen ciertas condiciones. Cada módulo y cada *Synth* tiene implementadas unas condiciones distintas para ejecutarse, pero, en términos generales, si la salida de un módulo no está conectada en la matriz a otro modulo, no producirá ninguna señal hasta que se «le comunique» que está conectado. Lo mismo ocurre si su salida tiene ganancia 0. Otros factores, en función de las características de cada módulo, se sopesan a la hora de decidir su pausa o ejecución, como si recibe o no alguna entrada (en el caso de un efecto) o de si se espera que siga ejecutándose incluso sin entradas y salidas (el módulo *Echo*, 5.10), para que siga procesándose la información que almacena temporalmente en su *buffer*.

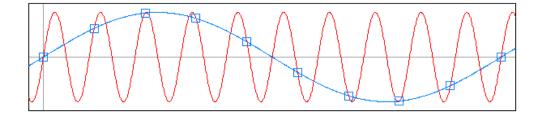


Figura 5.2: Gráfica de aliasing. (CC BY-SA 3.0, https://commons.wikimedia.org/w/index.php?curid=275973)

#### 5.1.4. Aliasing

Esta es una característica propia de cualquier sistema de audio digital, y que más pronto que tarde aparece como un obstáculo en la emulación de la síntesis analógica ya que esta última es ajena a esta característica.

Cuando una señal senoidal periódica de frecuencia f es muestreada a una frecuencia s mayor que f/2, la señal reconstruida con dichas muestras corresponde a una frecuencia inferior a f. Para evitar este efecto en señales producidas digitalmente, la frecuencia de muestreo ha de ser al menos 2s. Este es el «criterio de Nyquist» (Lavry, 2004), en honor a su descrubridor.

La figura 5.2 muestra gráficamente este proceso. En rojo se representa la señal original, y en azul, la señal reconstruida a partir de la muestras dibujadas como cuadrados. La frecuencia de la señal azul es mucho menor que la que se pretende reproducir. Solo una frecuencia mayor de muestreo (mayor densidad de cuadrados azules en la figura), al menos la mitad que la frecuencia de la señal roja, permitiría que la señal azul conservara la frecuencia.

Esta característica de las señales digitales no es compartida por los sintetizadores analógicos, razón por la cual se convierte en no deseable a la hora de emularlos. SuperCollider posee un gran abanico de *UGens* cuya implementación intenta minimizar este efecto. Un criterio importante a la hora de elegir entre varios *UGens* en el diseño de los *Synths* de los módulos de Synthi 100 ha sido el que incorporasen algún tipo de sistema *antialiasing*. Aun así, es impredecible el uso que se puede hacer de Synthi GME: una simple frecuencia modulada producida con dos generadores sinusoidales producirá fácilmente

aliasing, ya que las frecuencias laterales producidas frecuentemente incumplirán el criterio de Nyquist.

La única forma efectiva y práctica de limitar la presencia de *aliasing* es aumentar en la medida de lo posible la tasa de audio (*sample rate*). Aun así, es probable encontrar combinaciones de módulos y sus parámetros que superen el límite de Nyquist, por alto que este sea.

# 5.2. Estructura del quark Synthi GME

A continuación se muestra el árbol de directorios en los que se organiza el Quark de Synthi GME. En la representación de la figura 5.3 no se muestran los archivos concretos que albergan las definiciones de las clases, a excepción del de la clase principal SynthiGME y los que están alojados en el directorio raíz del programa.

El directorio más importante de todos es /clases, ya que en él se contienen todas las clases que SuperCollider entiende y compila al iniciarse o cuando se le pide expresamente. Estas clases se organizan por su función. Así, /classes/GUI contiene las clases relativas a la interfaz gráfica de usuario, mientras /classes/modules, las que representan a los diversos módulos del Synthi 100. No es preciso que exista ningún orden ni jerarquía de directorios en el quark. El orden actual tiene como fin la inteligibilidad y mantenibilidad del código.

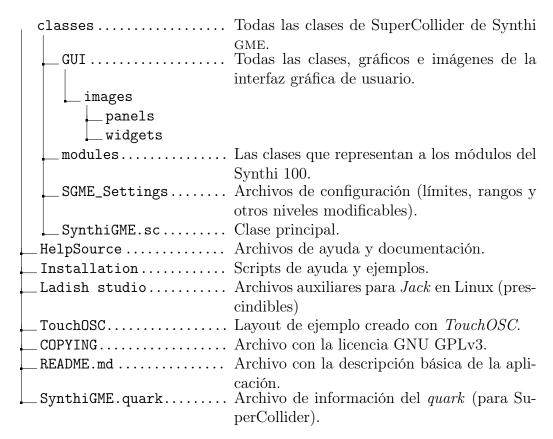


Figura 5.3: Árbol de directorios de Synthi GME.

## 5.3. Interfaz gráfica de SuperCollider

Cuando instanciamos la clase SynthiGME, lo primero que encontramos es una serie de ventanas que se corresponden con los diversos paneles de sintetizador, repartidos a lo largo y ancho de la pantalla del ordenador (fig. 5.4), manteniendo una disposición análoga a la de aquel. Cada ventana tiene como fondo una fotografía del Synthi 100, albergando en su capa superior toda una serie de mandos que provee SuperCollider para el diseño de interfaces gráficas, «sliders» y «knobs» principalmente. Los colores de estos han sido tomados de la misma fotografía para darle una apariencia lo más integrada posible con el conjunto.

Cada una de las ventanas puede ser cambiada de posición y de tamaño de forma independiente. El tamaño por defecto de las ventanas, si bien nos



Figura 5.4: Ventanas de Synthi GME. Cada ventana contiene uno de los paneles del Synthi 100 y llevan el nombre de «Panel 1», «Panel 2», y así sucesivamente de derecha a izquierda y de arriba a abajo.

permite tener una visión general del conjunto e identificar rápidamente cada panel del sintetizador, no nos permite ver los detalles suficientes como para trabajar sin hacer zoom. En este punto se hace totalmente necesario cambiar el tamaño de cada ventana sobre la que se desee trabajar. Por esta razón se han creado una serie de atajos de teclado (cuadro 5.1) y de ratón para conseguir hacerlo con cierta agilidad. Con un poco de práctica es posible usarlos con mucha soltura tanto en demostraciones como en la creación sonora.

Se ha tenido especial cuidado en usar solo Views compartidas por las tres plataformas en las que actualmente se puede compilar SuperCollider (Linux, Windows y Mac OS)<sup>2</sup>. Usando exclusivamente estas clases, se garantiza la compatibilidad hacia el futuro del diseño de la interfaz gráfica de usuario.

En cada uno de los diferentes *widgets* cuya función es enviar al programa un valor entre un rango continuo —como es el caso de los diales («Knobs») y los controles deslizantes («Sliders»)—, podemos variar su valor con diferentes

 $<sup>^2</sup>$ La lista de clases, los frameworks en los que están escritas y su compatibilidad entre plataformas se puede encontrar en la ayuda de SuperCollider:

http://doc.sccode.org/Overviews/GUI-Classes.html (consultado: 16 de febrero de 2020).

Hace visibles o invisibles los mandos de la ventana en foco Lleva al frente la ventana del panel 1 2 Lleva al frente la ventana del panel 2 3 Lleva al frente la ventana del panel 3 Lleva al frente la ventana del panel 4 5 Lleva al frente la ventana del panel 5 6 Lleva al frente la ventana del panel 6 7 Lleva al frente la ventana del panel 7 Desactiva y activa los pines sin función en las matrices Lleva al frente todas las ventanas Lleva a al tamaño y posición original la ventana en foco Lleva a al tamaño y posición original todas las ventanas Ctrl+0 Establece una nueva posición de ventanas «origen» Aumenta el tamaño de la ventana en foco (o doble click izquierdo) Disminuye el tamaño de la ventana en foco (o doble click derecho) Ctrl+C Cierra la aplicación

Cuadro 5.1: Atajos de teclado de la Interfáz gráfica de Supercollider

valores de precisión con la ayuda de diversas teclas pulsadas simultáneamente al movimiento del ratón o a la pulsación de las flechas. Ordenados desde el movimiento más grueso al más fino, la teclas a pulsar para alterar la velocidad son: Shift, Ctrl, sin tecla de función (ratón solo), y Alt.

Es importante notar que la forma por defecto en la que se cambia el valor de un dial no es haciendo un gesto rotatorio con el ratón. Se ha optado, en cambio, por un movimiento vertical. Si bien el movimiento rotatorio puede resultar más intuitivo, tiene la característica que un solo click de ratón sobre el dial cambia su valor de golpe al correspondiente al radio sobre el que el puntero se encuentra. Esto hace imposible hacer click sobre el dial sin alterar su valor drásticamente.

# 5.4. Más allá de la pantalla: interfaz OSC

Open Sound Control (s.f.), conocido por sus siglas OSC, es, tal como lo define la propia CNMAT (Center for New Music and Audio Technologies),

«un protocolo para la comunicación entre computadoras, sintetizadores y otros dispositivos multimedia, optimizado para la tecnología en red». Nacido en el CNMAT, OSC busca ser un MIDI mejorado, especialmente en lo que se refiere a la facilidad de implementación, precisión de los datos y a la expresividad de su sintaxis. Un mensaje OSC está formado por una cadena de texto en estilo URL, y los valores formados por 32 bits, tanto enteros como de coma flotante.

Por todo esto, y por su fácil uso en SuperCollider, este protocolo ha sido escogido, no solo para que la clase SynthiGME se comunique con interfaces remotas, sino cualquier comunicación con ella se produce a través de este protocolo. Se ha creado un único método en la clase con el que el usuario puede interactuar de forma segura: SynthiGME.setParameterOSC (string, value, addrForbidden), donde string es la cadena de texto del mensaje y value, el valor. Ambos forman el mensaje OSC que se envía al sintetizador. Tanto si los valores se introducen desde la red, como desde la interfaz gráfica por defecto, o a través de mensajes de texto de código, el método recomendado es precisamente este.

Todo mensaje recibido por el sintetizador es devuelto, a modo de eco, a todas las direcciones con las que el sintetizador ha sido «emparejado» previamente. El emparejamiento de los diversos dispositivos al sintetizador se realiza por medio del método SynthiGME.pairDevice(), reconociendo cualquier dispositivo en red que envíe el mensaje /ping en los siguientes 4 segundos<sup>3</sup>. El argumento addrFrobidden del método de envío de mensajes OSC contendrá opcionalmente la dirección del dispositivo remitente, y está pensado para bloquear dicha dirección a la hora de hacer eco del mensaje.

Synthi GMEtrae un ejemplo de interfaz *OSC* implementada en *TouchOSC* (fig. 5.5), en la carpeta [raíz]/TouchOSC. Contiene algunos ejemplos de módulos así como las dos matrices de conexiones (sección 5.17).

Esta interfaz única de comunicación de la clase SynthiGME con el exterior extiende las posibilidades de uso de este dispositivo más allá de la propia

 $<sup>^3</sup>$ Este mensaje se ha elegido para mantener compatibilidad con el software TouchOSC (s.f.), una App para dispositivos móviles que permite diseñar interfaces eficientes y muy funcionales, usando el protocolo OSC

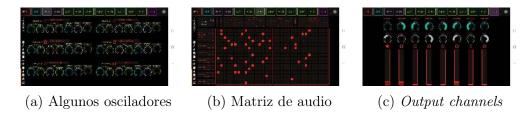


Figura 5.5: Páginas de ejemplo del layout creado en TouchOSC para Synthi GME. Los widgets se pueden deslizar y girar con el dedo en pantallas táctiles. El layout se encuentra en el directorio /TouchOSC de la aplicación.

interfaz gráfica por defecto. Podemos imaginar interfaces en red creadas ad hoc, escritas en cualquier lenguaje de programación y framework o, quizás algo mucho más sugerente y creativo, interfaces de hardware controladas con microchips tipo Arduino. Por esta razón se ha cuidado que los mensajes OSC necesarios para comunicarse con el sintetizador sean autoexplicativos. En la tabla 5.2 se recoge la formación de todos los mensajes OSC que entiende la clase SynthiGME, asociados a cada uno de los módulos.

OSC Type Tag String

Módulo	Número	Parámetro	Level	Módulo de Synthi 100
		/selector	[1-5]	
		/delay	[0.0 - 10.0]	
		/attack	[0.0-10.0]	
		/decay	[0.0-10.0]	
/env	/[1-3]	/sustain	[0.0-10.0]	Envelope Shapers
		/release	[0.0-10.0]	
		/envelopeLevel	[-5.0-5.0]	
		/signalLevel	[-5.0-5.0]	
		/gate	[0-1]	
/ring	/[1-2]	/level	[0.0-10.0]	Ring Modulators
		/pulse/level	[0.0-10.0]	
		/pulse/shape	[-5.0-5.0]	
		/sine/level	[0.0-10.0]	
/osc	/[1 10]	/sine/symmetry	[-5.0-5.0]	Oscillators
	/[1-12]	<pre>/triangle/level</pre>	[0.0-10.0]	Oscillators

## 5. DESCRIP. DE LA AP.

			F		
		/sawtooth/level	[0.0-10.0]		
		/frequency	[0.0-10.0]		
		/range	[0-1]		
/noise	/[1-2]	/color	[0.0-10.0]	Noise Generators	
7110150		/level	[0.0-10.0]	11003C Generators	
		/mean	[0.0-10.0]		
		/variance	[-5.0-5.0]	Random Control Voltage Generator	
/random		/voltage1	[0.0-10.0]		
		/voltage2	[0.0-10.0]		
		/key	[-5.0-5.0]		
		/filter	[-5.0-5.0]		
/	/[4 0]	/pan	[-5.0-5.0]	Outnot Channels	
/out	/[1-8]	/on	[0-1]	Output Channels	
		/level	[0.0-10.0]		
/in	/[1-8]	/level	[0.0-10.0]	Input Amplifiers	
		/frequency	[0.0-10.0]		
/filter	/[1-8]	/response	[0.0-10.0]	Filters	
		/level	[0.0-10.0]		
		/63	[0.0-10.0]		
		/125	[0.0-10.0]		
		/250	[0.0-10.0]		
/C:31 D 1		/500	[0.0-10.0]	Filter Bank	
/filterBank		/1000	[0.0-10.0]		
		/2000	[0.0-10.0]		
		/4000	[0.0-10.0]		
		/8000	[0.0-10.0]		
		/delay	[0.0-10.0]		
/ 1		/mix	[0.0-10.0]		
/echo		/feedback	[0.0-10.0]	$Echo\ A.D.L.$	
		/level	[0.0-10.0]		
/slew		/rate	[0.0-10.0]	Slew Limiters	
/patchA		/[67-128]/[1-66]	[0.0-1.0]	Audio Control	

/patchV /[67-128]/[1-66] [0.0-1.0] *Voltage Control* 

Cuadro 5.2: Estructura de la cadena de texto (*OSC Type Tag String*) y valor de los mensajes *OSC* para comunicarse con Synthigme.

## 5.5. Oscillators

Los módulos más importantes de cualquier sintetizador son aquellos que son capaces de generar algún tipo de señal, sea esta audible o no. La forma más simple de generador de señal es el oscilador, que produce una continua repetición de cambios periódicos de la señal en el tiempo. Dependiendo de la finalidad de dicha señal, en términos generales, esta puede ser de audio, cuando su frecuencia está dentro del rango audible, o de control de voltaje, normalmente con frecuencias por debajo del umbral de audición, y cuya función es la de modular otras señales en diversos parámetros. Esta última es la función de los llamados LFO (Low Frequency Oscillator, por sus siglas en inglés).

Los parámetros fundamentales de una señal periódica en cualquier sintetizador son la frecuencia y la amplitud. Otro parámetro, que puede ser variable o fijo, en función de su diseño, es la forma de onda. Esta determina en alto grado el timbre o color del sonido producido, por lo que existen muchas y muy diferentes técnicas de moducación de la misma. A este respecto, cabe señalar que existen una formas de ónda básicas, por sus características matemáticas, que las convierten en básicas y comunes dentro del mundo de la síntesis sonora. Estas son la onda sinusoidal, la triangular, la rectangular (también denominada pulso), y la diente de sierra o rampa (fig. 5.6). No solo son posibles las transiciones entre una y otra forma de onda, como mecanismo de modulación, sino también la variación de ciertos parámetros de cada una de ellas, siendo la simetría entre sus partes la más importante.

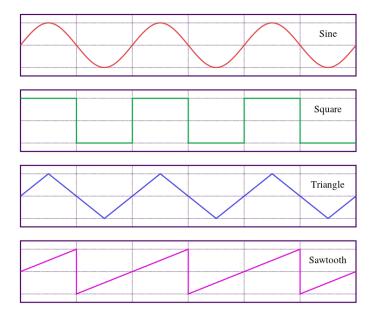


Figura 5.6: Ondas periódicas simples más comunes en la síntesis sonora. (Omegatron / CC BY-SA)

## 5.5.1. El módulo *Oscillator* en el Synthi 100

El Synthi 100 original de EMS estaba provisto de tres osciladores sinusoidales y de diente de sierra, tres triangulares y rectangulares y, por último, tres LFO, osciladores de baja frecuencia. El cambio introducido en el Synthi 100 de Cuenca es de gran importancia ya que pasa de 9 a 12 osciladores, cada uno de los cuales es capaz de generar cualquiera de las cuatro formas de onda, pudiéndose todos ellos commutar entre High Frecuency y Low Frecuency con un sencillo interruptor. A pesar de no conservar la antigua división entre Low Frecuency Oscillator y High Frecuency Oscillator, llama la atención que solo los tres últimos osciladores (los númerados como 10, 11 y 12) tienen una salida de voltaje, como reminiscencia de aquella división, ya que, cualquiera de las salidas del resto de osciladores (como de cualquier módulo del sintetizador) puede ser utilizada como control de voltaje gracias a la posibilidad de usar los canales de salida (Output Channels), como busses entre la matriz de audio y la de control de voltaje.



Figura 5.7: Tres de los 12 osciladores con los que cuenta el Synthi 100 del GME de Cuenca. A diferencia de versiones anteriores, este cuenta con todas cuatro formas de onda en cada uno de sus módulos, pudiendo ser utilizados tanto para bajas como altas frecuencias.

#### 5.5.1.1. Los diales del módulo

Cada oscilador del Synthi 100 de Cuenca está compuesto un interruptor LFO/HFO, denominado *range*, y 7 diales (fig. 5.7).

Pulse Level Amplitud de una señal con forma de onda rectangular.

Pulse Shape Anchura de la sección positiva de la onda.

Sine Level Amplitud de la señal sinusoidal.

Sine Symmetry Simetría entre las partes positiva y negativa. A menor simetría la señal adquiere progresivamente en uno de sus polos una

forma más puntiaguda, rompiendo, por tanto, la simetría con el polo sinusoidal.

Triangle Level Amplitud de una señal triangular.

Sawtooth Level Amplitud de una señal con forma de diente de sierra.

Frequency Este valor define la frecuencia de cada uno de las formas de onda. Su rango oscila entre 0.025 Hz y 500 Hz o entre 1 Hz y 10 KHz, dependiendo de si range está en posición LFO o HFO.

#### 5.5.1.2. Entradas y salidas del módulo

Tal como se observa en las matrices de conexiones de audio y de control de voltaje, cada oscilador tiene dos salidas: una para la forma de onda sinusoidal y de diente de sierra, Sine/Saw, y otra para la forma de onda rectangular y triangular, Pulse/Triangle. Así, aunque todas ellas dependen de un mismo parámetro de frecuencia, sus niveles de salida y los propósitos de cada una pueden ser totalmente diversos dentro de la red de la matriz.

Como entradas, en la matriz de control de voltaje contamos con una por cada oscilador de control de la frecuencia, con el claro fin de permitir la técnica de síntesis por frecuencia modulada (FM). En la matriz de señal de audio, existe una entrada por oscilador Synchronisation inputs, y una entrada (en los 6 primeros osciladores) de Pulse Width Modulation, para la modulación de la forma de onda Pulse.

## 5.5.2. Los osciladores de la aplicación Synthi GME

A pesar de ser uno de los módulos más elementales, es uno de los que más líneas de código tiene tras de sí, ya que aglutina a 4 osciladores, si contamos por tales cada una de las formas de onda que lo componen. Sine está compuesto por al menos 2 UGens de SuperCollider para, haciendo morphing, poder imitar el cambio en la simetría de la onda. Las formas de onda Triangle y Pulse están también formadas por dos UGens que hacen crossfade entre ellos para evitar el efecto de aliasing. Son muchos UGens en total, que pueden

comprometer la eficiencia en el funcionamiento del *software* en su conjunto y, por tanto, uno de los elementos a mejorar en el futuro.

A día de hoy, las entradas de audio Synchronisation inputs y Pulse Width Modulation están sin implementar.

#### 5.6. Filters

En el campo de la síntesis sonora, tan importante es la generación de una señal a partir de la suma de otras, como la capacidad de filtrar componentes espectrales de una señal dada. Ambos caminos, basados en el hecho de que cualquier señal periódica imaginable puede ser descompuesta en una suma de señales sinusoidales, constituyen la síntesis aditiva y la síntesis sustractiva respectivamente. Son los filtros los módulos imprescindibles en la síntesis sustractiva, ya que permiten elegir un grupo de frecuencias de entre todas las que ofrece la señal de su entrada, generando cambios tímbricos y dinámicos en la señal de salida.

## 5.6.1. Tipos de filtros

No es este el lugar para profundizar mundo del filtrado de señales, por su complejidad y amplitud. Simplemente se darán unas pinceladas básicas para poder describir el comportamiento del módulo *filter* de Synthi 100.

De igual modo que una señal luminosa puede ser descompuesta en diferentes longitudes de onda, formando líneas o bandas espectrales, el sonido también se puede representar en el dominio de la frecuencia, de manera que es susceptible de ser descrito como la suma de un número ilimitado de componentes sinusoidales, con su propia amplitud y fase. Los diferentes timbres o colores de un sonido están fuertemente determinados por esta composición frecuencial. Filtrar un sonido o señal no será otra cosa que dejar pasar ciertas frecuencias de una señal y cancelar otras, consiguiendo un producto sonoro diferente de la señal original a nivel tímbrico.

A grandes rasgos, los tres tipos de filtros más importantes (fig. 5.8) son el de paso bajo (LPF), el de paso alto (HPF) y el de paso banda (BPF). Los dos

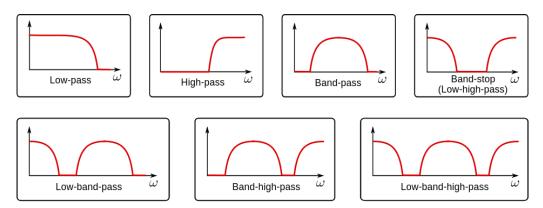


Figura 5.8: Diferentes tipos de filtro. De izquierda a derecha en la fila superior: pasa bajo, pasa alto, pasa banda y rechaza banda. En la fila inferior, diversos filtros producto de combinaciones de otros más simples (imagen original de SpinningSpark, CC BY-SA 3.0, modificada por el autor).

primeros tiene la característica común de dejar pasar solo las componentes frecuenciales de un lado del espectro, bien del lado de los graves (bajos), bien del lado de los agudos (altos), teniendo como principal parámetro la frecuencia de corte, a partir de la cual se atenúan o cancelan el resto de frecuencias. El filtro de paso de banda tiene como variables una frecuencia central y una anchura de la banda que deja pasar. Otros parámetros importantes determinan el tipo de curva y de pendiente que tiene el filtro en su frecuencia de corte, lo que queda determinado, principalmente, por su orden.

A partir de estos filtros teóricos, la tecnología en torno al filtrado de señales ha combinado de múltiples maneras todos sus parámetros en función de sus objetivos. Se pueden encontrar filtros en los que puede variar uno o varios de sus parámetros (frecuencia de corte, anchura de banda...), combinados en filtros multibanda, filtros en serie que provocan un cambio en el orden del filtrado, filtros de rechazo de banda (como una combinación de LPF y HPF), etc.

Por último, conviene indicar la exitencia de filtros en los que la frecuencia de corte puede tener una amplitud adicional, formando un pico más o menos pronunciado. Esta amplitud es conocida como *resonancia*. En los sintetizadores analógicos, si esta resonancia es lo suficientemente elevada, se puede producir un sonido sinusoidal en la frecuencia de corte incluso en la ausencia



Figura 5.9: Dos de los 8 filtros del Synthi 100 del GME.

de señal de entrada. Esta característica es utilizada por EMS en varios de sus dispositivos, entre ellos el Synthi 100.

## 5.6.2. Los filtros resonantes de Synthi 100

El Synthi 100 cuenta en su primer panel con 8 módulos denominados filter, cada uno de los cuales permite variar tres parámetros: Frequency, Response y Level (Fig. 5.9). Estos 8 módulos están divididos en dos grupos de 4: Low pass/Resonator/Osc, con los filtros del 1 al 4, y High pass/Resonator/Osc, con los numerados del 5 al 8.

Para comprender mejor el comportamiento de este módulo, a continuación se describe el comportamiento del dial *Frequency* en función de los diversos valores de *Response*:

- $Low/Hith\ pass\ (Response \gtrsim 0)$  El filtro se comporta como  $Low\ pass$  (filtros 1 al 4) o  $High\ pass$  (filtros 5 al 8). Frequency varía la frecuencia de corte del filtro.
- Resonator (Response  $\approx 5$ ) En torno al valor de 5, el filtro se comporta como Low o High pass con amplificación de la frecuencia de corte.
- Osc (Response ≤ 10) A medida que el valor de Response se acerca a 10, el resultado se acerca a un sonido sinusoidal cuya frecuencia es frequency. El filtro se comporta prácticamente como un oscilador incluso cuando no existe una señal de entrada.

#### 5.6.3. Implementación en Supercollider

Existen diversos *UGens* que pueden emular perfectamente el comportamiento de un filtro *resonante*, es decir, del tipo *de paso bajo* o *de paso alto* con la adición de la amplificación de la frecuencia de corte. Se ha optado por usar los filtros BLowPass y BHiPass respectivamente. Estos forman parte del conjunto de filtros *BEQSuite*, incluidos en las distribuciones estándar de SuperCollider.

Existe, sin embargo, cierto comportamiento de los filtros resonantes analógicos que no es posible emular con estos filtros digitales: la posibilidad de convertirse en un oscilador cuando el valor de su resonancia se acerca al máximo sin contar con la existencia de una señal de input. Habitualmente, la salida de un filtro digital en estas circunstancias será nula. Para emular esta característica analógica, se ha añadido un ruido blanco constante de una amplitud casi insignificante (en torno a -60 dB). Este nivel de ruido es suficiente como para ser amplificado como para convertirse en señal audible en en la frecuencia de corte cuando el valor de response se acerca a 10.

## 5.7. Octave Filter Bank

Octave Filter Bank es un módulo compuesto por 8 filtros paso de banda cuyas frecuencias centrales están a distancia de octava, dentro del rango



Figura 5.10: Vista parcial del módulo *Octave Filter Bank*. Las frecuencias centrales de cada filtro están separadas entre sí por un intervalo de octava.

62.5 Hz–8 kHz (Estas frecuencias son algo más bajas que la nota do del sistema temperado). Según las especificaciones de EMS, estos filtros son resonantes. Aunque no trae más aclaraciones, podemos suponer que es precisamente el valor de dicha resonancia la que puede ser variada con cada dial. La función de este banco de filtros sería, por tanto, no la de hacer una ecualización multibanda, sino, más bien, la de colorear de una forma pronunciada el sonido de su input.

Esta configuración, sin embargo, no está libre de cuestiones difíciles de explicar: Llama la atención, por ejemplo, que estas frecuencias resonantes sean fijas en lugar de poder ser variadas por voltaje. Tampoco es claro qué anchura tiene la banda de cada filtro y cuál es su amplitud independientemente del valor de la de la frecuencia resonante. Por otra parte, el panel del Synthi 100 trae la siguiente descripción en el centro superior del módulo: Band Pass Centre Frequency (Hz) (Fig. 5.10), lo cual no da ningún indicio de que se trate de un conjunto de filtros resonantes. Sin duda, estas cuestiones solo pueden dirimirse con la experimentación en el mismo sintetizador.

## 5.7.1. Implementación en Synthi GME

En el diseño del módulo en SuperCollider se ha optado por crear un filtro multibanda no resonante. Esta solución deja para estudios ulteriores el comportamiento real del módulo *Octave Filter Bank* en el Synthi 100. Por otra parte, aunque no sea resonante, no cabe duda de que tiene la capacidad de sustraer de la señal del su input determinados rangos frecuenciales. En la versión presente de Synthi GME, cada *Synth* contiene como *UGen* principal

a BBandPass, del conjunto de filtros BEQSuite, un filtro paso de banda de segundo orden.

# 5.8. Envelope Shapers

El «generador de envolvente» (conocido por sus nombres en inglés, Envelope Shaper o Envelope Generator), es uno de los módulos, junto con el de los osciladores, que no suelen faltar en los sintetizadores analógicos. Pero esta omnipresencia no es paralela a su estandarización. Los parámetros que controlan el generador de envolvente dependen de cada fabricante e, incluso, de cada modelo. Por si fuera poco, precisamente una de las grandes diferencias entre el Synthi 100 de Cuenca y el resto de Synthi 100 son los parámetros del módulo Envelope Shaper que este permite variar, lo cual da fe de la velocidad a la que el concepto de envolvente se desarrollaba por aquellos años.

#### 5.8.1. Las fases de una envolvente típica

Una «envolvente» puede ser descrita como la forma que se da al desarrollo en el tiempo del nivel de un parámetro de una señal, habitualmente la amplitud. De esta forma, una señal continua generada, por ejemplo, con un oscilador, puede adquirir una evolución dinámica, así como un inicio y un final. Puesto que una envolvente puede «envolver» a cualquier tipo de señal, sea esta de audio como de voltaje, podemos hacer variar con ella otros parámetros como la frecuencia de un oscilador, la frecuencia de corte de un filtro, el nivel de salida de otra envolvente, etc.

Históricamente, la envolvente surge inspirada por el comportamiento natural dinámico de los instrumentos musicales, de ahí que el tipo de envolvente más popular hasta nuestros días es el ADSR, siglas de los términos en inglés attack, decay, sustain y release (Fig. 5.11), que es el que mejor se adecua (de forma simplificada) a la evolución dinámica de una nota de un instrumento musical clásico:

Attack Tiempo transcurrido entre el nivel 0 y el punto álgido (convencionalmente 1). Es el equivalente al ataque de un instrumento acústico

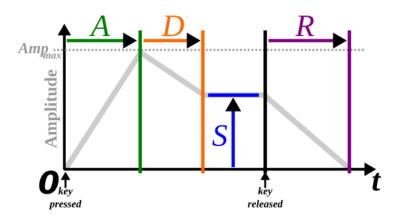


Figura 5.11: Esquema de una envolvente tipo ADSR (Abdull, CC BY-SA 3.0).

tipo, en el que los sonidos transitorios pueden producir un rápido pico dinámico.

**Decay** Tiempo transcurrido entre el final del ataque, con nivel 1 y el nivel de *sustain*. En instrumentos acústicos, tras alcanzar el nivel máximo en el periodo de ataque, existe un progresivo *decaimiento* dinámico del sonido hasta alcanzar un nivel estable.

Sustain Nivel en el que la señal puede ser mantenida por tiempo indefinido (entre 0 y 1). A diferencia del resto de parámetros, el sustain no es un parámetro temporal, sino de nivel. De hecho, es el único parámetro de nivel de toda la envolvente, ya que el nivel inicial y final siempre son 0 mientras que el nivel alcanzado en el ataque siempre es 1 (entendiendo estos valores como factor de cualquier unidad de medida). En los instrumentos acústicos, este nivel es mantenido a voluntad por el instrumentista. En los sintetizadores (analógicos o digitales) se utiliza el control de gate para mantener la señal en sustain.

Release Tiempo transcurrido entre el final del mantenimiento de la señal (sustain), y su extinción.

#### 5.8.2. Control de la envolvente: Gate y trigger

Como se ha indicado, attack, decay y release son duraciones, con lo que su inicio y fin están determinados por los valores elegidos por el usuario (o por el control desde otro módulo), pero ¿cómo «sabe» el sintetizador cuándo ha de terminar sustain y comenzar el periodo de release? Y otra pregunta muy relacionada, ¿cómo se da comienzo al periodo de attack? La analogía de un instrumento de tecla, como el órgano, puede darnos una respuesta muy intuitiva: Cuando el organista baja la tecla se pone en marcha el ciclo completo de la nota musical con su propia envolvente dinámica. En ese preciso instante comienza un periodo de crecimento sonoro producido por el propio aire entrando a los tubos como por las sucesivas reflexiones del sonido recién comenzado en las paredes de la sala. Tanto el ataque como el decaimiento tienen lugar con un tiempo definido para cada tubo, con controlable por el intérprete. Una vez estabilizado el sonido en un nivel determinado (sustain) el sonido continuará este nivel tanto tiempo como quiera el intérprete. Solo cuando levante la tecla a su posición inicial será cuando se de comienzo a la fase de release, en la que el sonido, dependiendo de las características de la sala, requerirá de un periodo de tiempo mayor o menor hasta dejar de ser escuchado.

La tecla bajada y subida tiene su análogo dentro del mundo de los sintetizadores en el concepto de gate («puerta»), la cual se puede encontrar, del mismo modo, en dos estados: abierta (valor mayor que 0) y cerrada (valor de 0 o inferior). El valor de este gate no es otra cosa que un nivel de voltaje en el caso de los sintetizadores analógicos o el valor digital de 1 o 0 en el caso de un dispositivo de software. En ambos casos, gate puede ser considerada otra señal, cuya función es la de «dirigir» las etapas de la envolvente.

Si la señal gate viene delimitada por su inicio y su fin —puntos con los que se ponen en marcha los diferentes periodos de una envolvente— el trigger es cualquier señal cuando pasa de un valor de 0 o negativo hasta un valor positivo. Es el inicio de dicha señal (momento en el que pasa a ser mayor que 0) el que puede ser usado como trigger. Para el desarrollo de una envolvente en la que no existe sustain bastaría una señal trigger para dar el «pistoletazo

de salida» y recorrer sucesivamente todas sus etapas de principio a fin, con una duración total suma de todas ellas: attack, decay y release.

#### 5.8.3. Otros tipos de envolventes

No hay nada que nos limite a tres duraciones y transiciones (attack, decay y release) y a un nivel variable controlado con gate, sustain. De hecho, ADSR no es más que un caso especial de toda una infinidad de tipos de envolvente imaginables, especialmente en el mundo digital, donde podemos diseñar tantos periodos de transición como se quiera, asociados también a valores arbitrarios. Esta característica la encontramos en la clase Env de SuperCollider, si bien es común encontrarla en cualquier sintetizador digital moderno. Sin embargo, en los sintetizadores analógicos no hay más remedio que delimitar el número de parámetros de la envolvente, ya que cada uno de ellos será un dial. Por ello, podemos encontrar cierta variedad de envolventes que, en definitiva, podrían ser explicadas por configuraciones ADSR con los valores apropiados.

#### 5.8.4. Envelope Shaper en Synthi 100

# 5.8.4.1. Del generador de trapezoides del Synthi 100 al ADSR del Synthi 100 del GME

Según el folleto de especificaciones técnicas de Synthi 100 publicado por EMS, el módulo *Envelope Shaper* posee 4 diales para diseñar la forma de onda: delay, attack, on y decay. Atendiendo a las respectivas descripciones, así como a las gráficas (fig. 5.12) que EMS ha publicado en otros folletos de sintetizadores (Synthi Users Manual EMS, 1970), los cuatro parámetros controlan duraciones, sin posibilidad de variar el nivel de la envolvente al modo en el que lo hace sustain en el tipo ADSR. Delay es el tiempo inicial en el que la envolvente se mantiene en valor 0; attack, el tiempo en el que progresivamente asciende hasta su valor más alto; on el tiempo en el que se mantienen en este valor; y decay, el tiempo de descenso progresivo hasta el valor 0. La forma de onda es siempre un trapecio en el que podemos variar a la

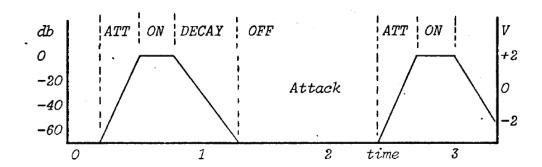


Figura 5.12: Envolvente del Synthi VCS3.

inclinación de sus lados no paralelos (attack y decay), y la longitud de su base menor (on). La altura del trapecio —el valor alcanzado tras attack— puede ser variada por otros dos diales: Trapezoid level para su salida de voltaje, y Signal level, para su salida de audio.

El comportamiento de la envolvente en función de los controles de voltaje puede ser variado por el dial Triggering mode, con cinco modos: Signal Threshold, en el que una señal a partir de un determinado nivel (variado por el dial Threshold level) inicia un ciclo de la envolvente; En Hold la envolvente comienza cuando recibe un valor positivo, no comenzando el tiempo de decay hasta recibir un valor negativo; Single Shot inicia un ciclo de la envolvente cada vez que recibe un valor positivo; El modo Free run inicia en bucle la envolvente; y Gated free run hace lo mismo que Free run pero solo si se recibe una señal gate positiva.

#### 5.8.4.2. Decisiones de implementación en SuperCollider

El generador de envolventes que posee el Synthi 100 del GME muestra una clara evolución respecto a las primeras unidades que EMS fabricó de este sintetizador. La forma de onda es generada por cinco diales: delay, attack, decay, sustain y release, que no es otra cosa que un ADSR ampliado por un delay inicial, parámetro existente en Synthi 100 desde sus orígenes, muy útil principalmente para las envolventes en bucle. Los modos que permite el dial Triggering mode nos recuerdan a los originales, si bien tienen algunas diferencias. Free run y Gated free run se mantienen con un significado similar,

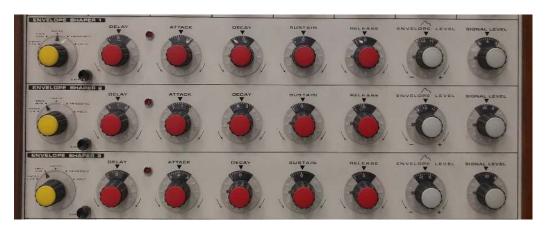


Figura 5.13: Los tres módulos de envolventes del Synthi 100 del GME de Cuenca. El dial de la izquierda permite seleccionar entre diversos modos de funcionamiento. Los diales rojos permiten modular la envolvente con diversas duraciones y niveles, y los diales blancos controlan los niveles de las señales de salida.

el de crear bucles de la envolvente. Sorprende la existencia de dos modos que «sostienen» la señal mientras que la señal control gate es positiva: Hold —que ya existía en Synthi 100— y Gated. A falta de unas descripciones por parte de EMS de su funcionamiento, podemos inferir que Hold mantiene la señal en su punto más alto tras el ataque (tal como en los Synthi 100 primitivos), mientras que Gated lo hará en el nivel de sustain, como cabría de esperar en una envolvente standard ADSR. Sin embargo, experimentos realizados ad hoc en el Synthi 100 del GME muestran un comportamiento diferente del esperado, tanto en los modos Hold y Gated, como en Gated free run.

Se observa un comportamiento aparentemente anómalo en los generadores de envolventes del Synthi 100 del GME. Esto puede ser debido a una falta de comprensión de su funcionamiento a la hora de realizar este estudio, a un funcionamiento aún incorrecto en su proceso de restauración, o incluso a cuestiones de fabricación originales. En cualquier caso, al carecer de un manual de instrucciones publicado por EMS, se han tomado ciertas decisiones de implementación para la aplicación informática del presente trabajo, reflejando en ella un comportamiento que no excluya el del propio Synthi 100 del GME, sino que lo amplíe y elimine redundancias. Esta decisiones se han

tomado para la presente implementación, siendo fácilmente reformable, si es necesario, en sucesivas versiones.

En la tabla 5.3 se presenta de forma sintética el paralelismo del comportamiento —en términos ADSR— del Synthi 100 original, el del GME y el implementado en SuperCollider, en función de los modos de *Triggering mode*. Las anomalías observadas en el GME y las soluciones abordadas son las siguientes:

Comportamiento idéntico de Gated free run y Triggered No se observa diferencia alguna: en ambos casos se procuce un ataque (attack) y un release hasta el nivel 0 (llamado decay) seguido de un tiempo de espera (en este caso, release). Se ha optado por la solución — aparentemente más lógica— de que, por una parte, Gated free run funcione en bucle mientras que gate tenga un valor positivo, algo que parece exigir el término Gated free run y que aparece en las descripciones del Synthi 100 original. Por otra parte, se permite que sustain aporte un nivel de llegada tras decay y que, por tanto, relase tenga la función de hacer progresar el nivel de sustain hasta 0. Triggered realiza lo mismo que Gated free run, excepto que no entra en ningún caso en bucle.

Comportamiento idéntico de *Gated* y *Hold* En ambos casos la envolvente «se para» en *sustain* mientras que *gate* introduce un valor positivo. Puesto que el Synthi 100 del GME se diferencia del original —entre otras cosas— por la presencia del nivel *sustain*, se ha optado por que *Gated* «se pare» en el nivel de *sustain* hasta que *gate* tenga valor negativo, mientras que en *Hold*, la parada se produzca tras *attack*, algo que ya hacía el Synthi 100 original.

Excepto en bucles, delay no tiene función Se ha optado por hacer que delay aporte un tiempo inicial antes de attack en todos los casos (en bucle o sin él), tal como parece indicar EMS para el Synthi 100 original. Si se quiere anular su efecto, basta con poner el dial a 0.

$Triggering\ mode$	Synthi 100 original	Synthi 100 GME	SuperCollider
$Gated\ free\ run \ (gate>0)$	bucle DAR	AD(0)R	bucle DADSR
Free run	bucle DAR	bucle DADR	bucle DADSR
$\overline{Gated} \ (gate{>}0)$	DASR	ADSR	DADSR
$Triggered \\ (Single shot)$	DAR	AD(0)R	DADR
Hold	DAOR	ADSR	DADSR

Cuadro 5.3: Comparación del comportamiento de las envolventes entre versiones del Synthi 100 y de la implementación en SuperCollider.

Es importante notar que el comportamiento observado en el sintetizador del GME puede ser imitado por completo desde la implementación de Super-Collider, manteniendo la posibilidad de una gama mayor de posibilidades. Por otra parte, las funcionalidades extendidas por esta implementación de software no solo son coherentes con las que encontramos en la síntesis analógica de la época, sino que están basadas principalmente en las descripciones que el propio EMS publicó sobre Synthi 100.

#### 5.8.4.3. Diales de salida

Envelope Shaper ofrece dos salidas, una de audio —Signal Level— y otra de voltaje —Envelope level—. En ambos casos se ofrece la posibilidad de invertir la polaridad de la señal, haciendo que sus diales permitan valores entre -5 y 5. Ambas salidas son independientes. En el caso de la salida de voltaje, la señal es continua con fines de control de otros módulos, y en el caso de la salida de audio, su salida es una señal de audio recibida y modulada por la envolvente.

## 5.9. Noise Generators

El EMS Synthi 100 posee tres generadores de ruido. Este módulo es probablemente el más sencillo de los que podemos considerar «fuentes» de sonido.



Figura 5.14: Uno de los dos generadores de ruido del Synthi 100 del GME.

No se encuentra ninguna diferencia en cuanto al diseño de la interfaz entre diferentes versiones del Synthi 100. Está compuesto de dos diales, ambos con los valores entre 0 y 10 (Fig. 5.14):

Colour Altera la «coloración» del ruido hacia el grave o hacia el agudo. Atendiendo a la descripción de EMS, en su posición central (5), su salida es la de ruido blanco, actuando un filtro pasabajos cuando se aumenta su valor, y un filtro pasabaltos cuando se disminuye.

#### Level Nivel de salida.

La implementación de Synthi GME responde precisamente a esta descripción. Un ruido blanco es filtrado en serie por un filtro pasaltos y un filtro pasabajos, cuyas frecuencia de corte entran dentro del espectro audible cuando el valor de Colour es respectivamente, mayor o menor que 5.

Este módulo consta de una única salida de audio, ninguna de control de voltaje. No contiene entrada de ningún tipo.

## 5.10. Echo A. D. L.

Este módulo, incluido en el Synthi 100 del GME de Cuenca, no forma parte de la configuración inicial del sintetizador. Debido a la ausencia de



Figura 5.15: Vista del módulo *Echo* A. D. L..

información publicada por EMS sobre sus últimos modelos del sintetizador, no se ha podido hallar el significado de las iniciales que incluye el nombre del módulo: «A. D. L.». Sin embargo, los diales que posee dejan poco margen de duda de su funcionamiento:

Delay Tiempo transcurrido entre la señal recibida y su eco.

Mix Su valor varía desde Dry, en el que solo se escucha la señal original (sin tratar), hasta Echo, en el que tiene por única salida la señal retardada. Los valores intermedios permiten mezclar las amplitudes relativas de ambas señales.

**Feedback** Nivel de rentrada o retroalimentación del módulo. Permite crear una cola de ecos.

Level Nivel de la amplitud de salida del módulo.

## 5.10.1. Implementación en Synthi GME

Existe un *UGen* en SuperCollider que emula precisamente un módulo de características similares, SwitchDelay. Para producir el retardo de la señal, este *UGen* posee un *buffer* donde esta se almacena para ser leída en un punto variable dependiendo del tiempo de *delay*. Una limitación de este sistema es, precisamente, que la variación de la cantidad de *delay* en tiempo de ejecución produce artefactos debidos a los saltos abruptos del puntero de lectura. La forma encontrada de minimizar estos sonidos residuales ha

consistido en hacer variar la variable de control delay suavemente de un valor a otro con la frecuencia de muestreo. Esto suaviza notablemente los cambios de posición del puntero de lectura del buffer, traduciéndose en una notable disminución de los ruidos, aunque no ha sido posible una eliminación absoluta de este. Es interesante seguir trabajando en la resolución de este problema ya que el delay es uno de los parámetros controlables por voltaje dentro del sintetizador.

## 5.11. Entradas y salidas del sistema

El Synthi 100 posee una serie de entradas y salidas de audio que permiten la recepción y el envío de señales. En la aplicación se ha procedido de un modo análogo al sintetizador, aprovechando la capacidad de SuperCollider de crear cualquier cantidad de puertos desde el servidor de sonido, quedando a disposición del sistema operativo. Estos canales pueden recibir audio –en este caso siempre digital— y enviar audio desde y hacia cualquier otro software instalado. SuperCollider nombra automáticamente estos puertos numerándolos. La tabla 5.4 muestra la correspondencia entre estos y los del Synthi.

Como se puede observar, se han reservado las dos primeras entradas y las dos primeras salidas para el propio sistema operativo. Por defecto, SuperCollider conecta las entradas 1 y 2 al micrófono del sistema y las salidas 1 y 2 a los altavoces (fig. 5.16). Por comodidad y practicidad, estos canales de salida, 1 y 2, contienen por defecto una mezcla stereofónica y panoramizada de los 8 canales de salida de Synthi GME. Dependiendo del sistema operativo o del software controlador del audio utilizado, los puertos pueden ser renombrados para ser utilizados de forma más intuitiva.

Cabe destacar, entre las salidas, dos estereofónicas que combinan varios canales de salida, *Pan Outputs*. Una combina a los canales del 1 al 4, mientras que la segunda los del 5 al 8 (fig. 5.17). Estos canales, como se verá en su sección (5.13), tienen un dial para panoramizar la señal. Esto solo tiene efecto en estos puertos combinados.

	Channels	Left	SuperCollider:out_3
Pan	1-4	Right	SuperCollider:out_4
outputs	Channels	Left	SuperCollider:out_5
	5-8	Right	SuperCollider:out_6
	Individual channels outputs	1	SuperCollider:out_7
		2	SuperCollider:out_8
		3	SuperCollider:out_9
		4	SuperCollider:out_10
		5	SuperCollider:out_11
		6	SuperCollider:out_12
		7	SuperCollider:out_13
		8	SuperCollider:out_14
	Send to device	1	SuperCollider:out_15
		2	SuperCollider:out_16
D 4 1		3	SuperCollider:out_17
External		4	SuperCollider:out_18
treatment devices		1	SuperCollider:in_3
devices	Return from device	2	SuperCollider:in_4
		3	SuperCollider:in_5
		4	SuperCollider:in_6
	Input amplifiers	1	SuperCollider:in_7
		2	SuperCollider:in_8
		3	SuperCollider:in_9
		4	SuperCollider:in_10
		5	SuperCollider:in_11
		6	SuperCollider:in_12
		7	SuperCollider:in_13
		8	SuperCollider:in_14
	Mic amp	1	SuperCollider:in_15
	Mic amp	2	SuperCollider:in_16

Cuadro 5.4: Entradas y salidas de audio del Synthi 100 y su correspondencia con las de Synthi  ${\tt GME}.$ 

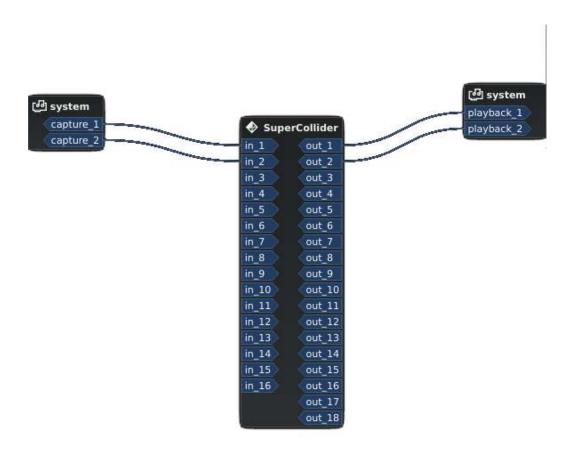


Figura 5.16: Entradas y salidas de SuperCollider al sistema operativo con Synthi GME. Los dos primeros puertos de entrada y de salida, están reservados por defecto a la entrada y salida estándar (micrófono y altavoces del ordenador). Por comodidad y practicidad, estos canales de salida, 1 y 2, contienen por defecto una mezcla stereofónica y panoramizada de los 8 canales de salida de Synthi GME. Los puertos están numerados, y respetan el mismo orden que el que se encuentra en el Synthi 100.



Figura 5.17: Algunas salidas XLR del Synthi 100. A la izquierda dos pares de salidas *stereo* que mezclan la salida de 4 canales del sintetizador respectivamente. A la derecha, dos de las 8 salidas monofónicas correspondientes a *Output Channels*.



Figura 5.18: Array de las ganancias de los 8 canales de entrada

# 5.12. Input Amplifier Level

Este módulo, compuesto únicamente por 8 diales de ganancia (fig. 5.18), controla los 8 canales de entrada del Synthi 100<sup>4</sup>. Por una simetría que no parece casual, este es el mismo número de canales que tiene para la salida de audio.

Las señales que por estos canales se reciben son ambivalentes, pudiendo ser tanto de audio como de voltaje, con lo que son conectables con cualquiera de las entradas de cualquier módulo del sintetizador. Esta característica hace el sintetizador modular un módulo en sí mismo capaz de ser controlado por cualquier señal exterior.

<sup>&</sup>lt;sup>4</sup>En total son 16 las entradas si contamos, además de los 8 *Input Amplifier Level*, las 4 entradas de retorno desde otros dispositivos y los 2 canales específicos para micrófonos.

# 5.13. Output Channels

#### 5.13.1. Los parámetros del módulo

**Filter** Colorea el sonido hacia grave o el agudo, a modo de combinación de un filtro paso de altos y otro paso de bajos. Solo funciona cuando está conectado hacia el exterior.

**Pan** Paneo del sonido. Funciona para los puertos *stereo* que combinan a los canales del 1 al 4 y del 5 al 8.

Off Interruptor que activa y desactiva la salida del canal hacia los puertos del sintetizador. Cuando su estado es Off, el canal puede seguir siendo utilizado a modo de bus.

Level Ganancia de salida del canal. Solo tiene efecto hacia los puertos del sintetizador.

## 5.13.2. Salidas y busses internos

Los 8 canales tienen como destino natural los puertos del sintetizador, bien hacia salidas stereofónicas, bien monofónicas. En todo caso, siguiendo el principio de ductilidad que lo caracteriza, estos canales pueden ser utilizados como simples busses internos capaces de reenviar la señal de entrada a cualquier módulo, como audio y como voltaje. El interruptor Off que incluye cada canal, permite activar o desactivar la salida a los puertos exteriores, pero su función de bus permanece en todo momento, si bien no tienen efecto sobre él los diales o el control deslizante. Devuelve la señal tal como la ha recibido. Esta función de bus resulta interesante para hacer que cualquier señal producida como audio sea utilizada como control de voltaje. De este modo, el patchbay o matriz de Audio Control puede ser comunicado con el de Voltage Control.



Figura 5.19: Vista del módulo *Output Channels*, 8 canales de salida independientes. Se aprecia en la foto el estado del Synthi 100 del GME, al que le falta uno de los controles deslizantes debido al repetido uso del mismo.

## 5.13.3. Implementación en SuperCollider

Existen en el caso de *Output Channels*, existen ciertas decisiones tomadas con más o menos arbitrariedad respecto a su comportamiento general. ¿Los controles de ganacia, paneo o filtro afectan a la señal portada como si de un *bus* se tratase, o se comporta en este caso como un simple *bypass*? ¿El interruptor *Off* desactiva solo la salida hacia los puertos del sintetizador o también al *bus*? Estas cuestiones no tienen una respuesta evidente sin una comprobación empírica. Por lo pronto, se ha optado por tomar las decisiones que más posibilidades permitan, las cuales han sido descritas en el epígrafe anterior.

# 5.14. Ring Modulators

El modulador en anillo o *Ring Modulator*, fue inventado por Frank A. Cowan en 1934 con una finalidad práctica en el campo de la telefonía. Su nombre



Figura 5.20: Los tres moduladores en anillo del Synthi 100 del GME.

procede de la forma del circuito analógico, la de un *anillo* de diodos. El producto de la modulación en anillo de dos señales cualesquiera es una señal compuesta por la suma y la diferencia de las frecuencias presentes en aquellas. La aproximación más sencilla al procedimiento analógico de este efecto es el de la multiplicación de ambas señales de entrada en el dominio del tiempo. La diferencia con la *amplitud modulada* es que las frecuencias de las señales de entrada no están presentes en su producto.

Este procesado de la señal adquiere interés sonoro cuando una de las dos señales de entrada —típicamente llamada «portadora» $^5$ — tiene una forma sinusoidal u otra simple.

El Synthi 100 tiene 3 moduladores en anillo, y su único mando de control es un dial de ganancia (*Level*). Sus entradas y salidas son exclusivamente de audio, aunque, gracias a la posibilidad de comunicar *busses* con entradas de audio hacia salidas de voltaje (fig. 5.13), la salida de *Ring Modulator* es susceptible —como el salidas de cualquier módulo— de ser utilizada como un voltaje de control.

Synthi GME implementa una simple multiplicación de las dos señales de entrada, con lo que se trata de uno de los módulos de menor coste de computación de toda la aplicación, al no intervenir ningún *UGen* de procesado de señal.

<sup>&</sup>lt;sup>5</sup>Ya que el *output* del *ring modulator* equivale a la multiplicación de dos señales, ambas son equivalentes en términos matemáticos debido a la propiedad conmutativa de la multiplicación. Los términos «señal moduladora» y «señal portadora» son intercambiables.

# 5.15. Random Control Voltage Generator

## 5.15.1. La aleatoriedad en el Synthi 100

Uno de las cualidades que hacen diferentes e interesantes las ejecuciones musicales humanas es el mayor o menor grado de impredicibilidad de sus movimientos, que se traduce en ligeras variaciones aleatorias. De este indeterminismo se ha hecho gala frecuentemente, quizás haciendo de tripas corazón, en los sintetizadores analógicos en general, y en el Synthi 100. Sus circuitos son sensibles a los cambios de humedad y temperatura hasta el punto de que un mismo patch y valores de los controles de los diversos módulos puede tener resultados sonoros completamente diferentes en función de las diversas condiciones ambientales en las que se le haga funcionar.

Pero no todo es aleatoriedad incontrolada. Este módulo (junto al de *Noise Generators* en 5.9), permite «inyectar» cierta dosis de aleatoriedad al sistema de forma intencionada. Su disposición de salidas y diales, como se verá más abajo, está inspirado en el funcionamiento de los teclados de la época, que enviaban información —por voltaje— de la nota pulsada (frecuencia), velocidad de pulsación (intensidad) y la información de subida y bajada de la tecla (*gate*). De algún modo, este módulo equivale a la ejecución aleatoria de teclas atacadas con diferentes velocidades.

## 5.15.2. Los diales del módulo

Los dos primeros diales, *Mean* y *Variance* bastan para establecer la frecuencia del cambio de los voltajes de salida. La media temporal entre los cambios es controlada por *Mean*. Los valores reales de tiempo entre cambios serán más próximos a esta media en tanto en cuanto *Variance* sea más cercana a 0, y más dispersos aleatoriamente de esta media cuanto más se aleje de 0. Los tres diales de la derecha tienen como única función la de controlar la amplitud del voltaje de salida. Aunque el cambio de voltaje se establece al mismo tiempo en *Voltage 1* y *Voltage 2*, el valor de cada uno de ellos es aleatorio y el rango en el que estos valores se mueve es independiente y



Figura 5.21: Unidad del generador aleatorio de voltaje de control del Synthi 100 del GME, uno de los pocos módulos que no tiene más que una instancia.

variado por ambos diales. Por último, Key, a modo de «tecla» cumple con la función de abrir un gate y cerrarlo brevemente cada vez que se produce un cambio en los voltajes.

El hecho de que los cambios sean simultáneos permite eventos sonoros con dos parámetros aleatorios e independientes. Típicamente, es posible crear notas musicales en los que *Voltage 1* controla, pongamos por caso, la frecuencia, mientras que *Voltage 2*, la intensidad. *Key* comunica a una envolvente el inicio y fin de cada nota. Evidentemente, su uso no está reducido a este ejemplo musical, pero su simplicidad permite comprender mejor el funcionamiento de este módulo.

A modo de resumen, estos son los parámetros que se nos permite modificar:

*Mean* Tiempo medio entre cambios de voltaje.

Varianza en torno al tiempo medio.

Voltage 1 Voltaje de salida 1.

Voltage 2 Voltaje de salida 2.

**Key** A modo de *gate*, esta salida de voltaje tiene un valor distinto de 0 entre cambios, mientras que tiene valor 0 durante un tiempo breve en cada cambio.

## 5.15.3. Las salidas de voltaje

Haciendo honor al propio nombre del módulo, las salidas de este son únicamente de voltaje. Las tres salidas coinciden con los tres diales de ganancia, Voltage 1, Voltage 2 y Key.

# 5.15.4. Implementación en Synthi GME

Este es el único módulo cuyo generador de señales no está dirigido por una instancia de Synth sino por un Routine. Un Synth es utilizado únicamente para convertir los valores numéricos del Routine a valores de audio y de control. Una «rutina» en SuperCollider puede explicarse como una serie de eventos que se suceden uno tras otro y que pueden ejecutarse asíncronamente. Una característica interesante es la posibilidad de situar en el tiempo estos eventos, haciendo esperar a la rutina un número determinado de segundos con el método wait(Integer). La rutina que gobierna el módulo Random Control Voltage Generator se compone de un bucle de valores aleatorios que se suceden con esperas de duración variable. Los valores aleatorios son generados por métodos del propio lenguaje de programación.

Existen ciertas decisiones arbitrarias tomadas a la hora de crear el código de Synthi GME. Como en todos los módulos, el rango de los valores se ha tomado dentro de unos límites razonables para poder trabajar con él, siempre que no se haya encontrado ninguna indicación técnica en las fichas de los 70 sobre EMS Synthi 100. En Variance se ha ignorado el hecho de que su rango en el dial esta comprendido entre -5 y 5, haciendo que la varianza sea mayor o menor según su valor también lo es, sin que 0 tenga un significado especial. Esta cuestión ha de ser resuelta empíricamente en el propio GME.

## 5.16. Slew Limiters

Un *Slew Limiter*, también conocido como *rate limiter*, limita la velocidad de cambio de una señal y, por tanto, su pendiente. Su efecto es el de suavizar la señal al redondear sus cambios bruscos. Cuando una señal de entrada tiene



Figura 5.22: Los tres *Slew Limiters* del Synthi 100 del GME.

componentes frecuenciales mayores a la del *Slew limiter*, estas son eliminadas en la señal de salida. Synthi 100 posee tres de estos módulos, cuya único parámetro, controlable por voltaje, es la «tasa de cambio» (fig. 5.22).

# 5.16.1. Implementación en Synthi GME

Existen varios *UGens* que pueden de variar la «tasa de cambio» de una señal de entrada, si bien solo uno, Slew, está optimizado hacerlo con una de audio. El resto (Var, Var2, VarLag o LagUD) están más recomendados para las señales de control. La implementación del *Synth* es totalmente trivial debido a que los parámetros de Slew coinciden con los del Synthi 100.

## 5.17. Matrices de conexiones modulares

Un sintetizador modular típico, comunica sus módulos por medio de cables que el usuario conecta (fig. 5.24). Los términos patch y patcher se usan habitualmente para designar un conjunto de conexiones entre módulos, no solo en el caso del audio analógico, sino también del mundo digital. Sintetizadores de propósito general como Pure Data o MAX, siguen este paradigma de «patchear» entradas y salidas con cables virtuales (fig. 3.1). Sin embargo, si hay algo que caracteriza a los sintetizadores de EMS, es la forma particular en la que se realizan las conexiones entre sus módulos.

No es necesario el uso de un solo cable para ordenar las entradas y salidas. En lugar de tener un único conector para cada salida o entrada, Synthi 100 tiene dos matrices bidimensionales cuyas ordenadas representan a las salidas y las abcisas las entradas de todos los módulos. Cada uno de sus nodos representa, por tanto, una conexión posible, a modo de coordenada. Las matrices del Synthi 100 del GME de Cuenca tienen unas dimensiones de 66 columnas por 60 filas, si bien algunas de estas no tienen ninguna conexión en este momento. En la matriz de *Audio Control* existen 59 filas y 64 columnas, haciendo un total de 3776 conexiones individuales posibles. La de *Voltage Control*, asimismo, 56 filas por 65 columnas, con un total de 3640 individuales posibles.

La conexión se realiza por medio de unas clavijas que, al introducirse en la conexión hembra, cierran el circuito. La limpieza visual de las conexiones es indiscutible respecto al paradigma del sintetizador modular por cables, especialmente cuando el número de conexiones es significativa; es muy fácil de reflejar en tablas sobre papel (fig. 5.23), incluso es posible escribirlo con coordenadas numéricas, ya que cada salida o entrada tiene un número único asignado (fig. 5.25).

# 5.17.1. Implementación en Synthi GME

El diseño de la clase SGME\_Patchbay y sus herederas SGME\_PatchbayAudio y SGME\_PatchbayVoltage, ha sido uno de los que más cuidado ha requerido a la hora de crear el código. El concepto de su funcionamiento, sin embargo, es bastante simple. Cada módulo dispone de tantos busses como entradas y salidas tenga. Al conectar un nodo de una matriz, se crea un Synth cuya única misión es recibir la señal de un bus y comunicarlo a otro. De este modo, el circuito queda cerrado, y solo se abre cuando se desconecta el nodo, y con él se destruye el Synth que unía los busses.

Es muy importante tener en cuenta el orden de ejecución de todos los Synths de la aplicación (véase la sección 5.1 para más detalles). Una conexión entre un módulo que se ejecuta más tarde que su receptor no comunicará a este ninguna señal ya que el ciclo terminará antes de que el bus pueda conducirla a su destino, siendo eliminada antes de comenzar el siguiente ciclo. Es precisamente en las matrices donde hay que asegurar que la señal llega a su destino. Para ello se chequea la situación relativa en el servidor de los

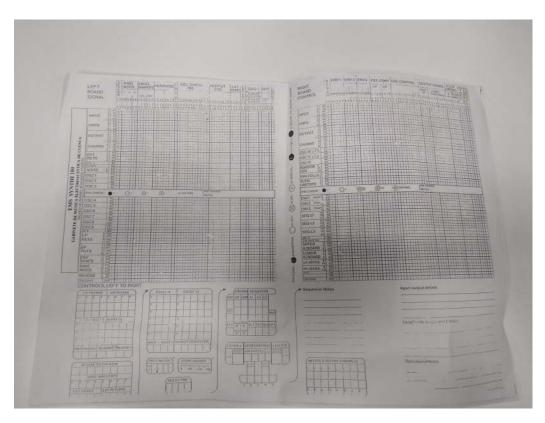


Figura 5.23: *Dope Sheet* conservado en el GME, hoja en la que el compositor o el técnico encargado del GME tomaba nota esquemática de todas las conexiones y de los valores de ciertos módulos con el fin de poder repetirlos en el futuro.

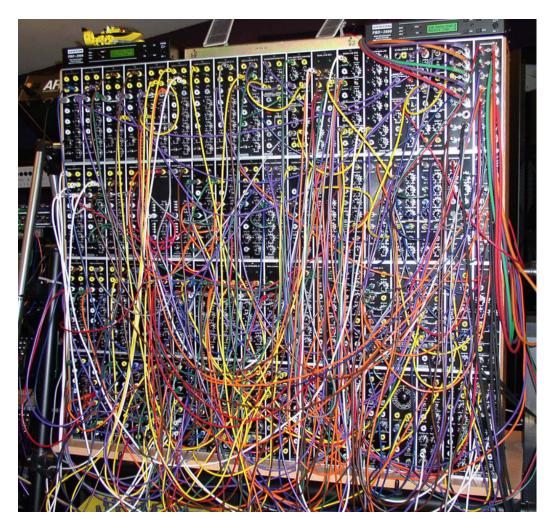


Figura 5.24: Tipico cableado de un sintetizador modular. Las salidas de un módulo se conectan por medio de cables con las entradas de otros módulos. La configuración del cableado determina en parte los resultados sonoros del conjunto. A pesar de lo intuitivo del sistema y su extendido uso, Los cruces de cables y la dificultad en su seguimiento le confieren una apariencia de gran complejidad.

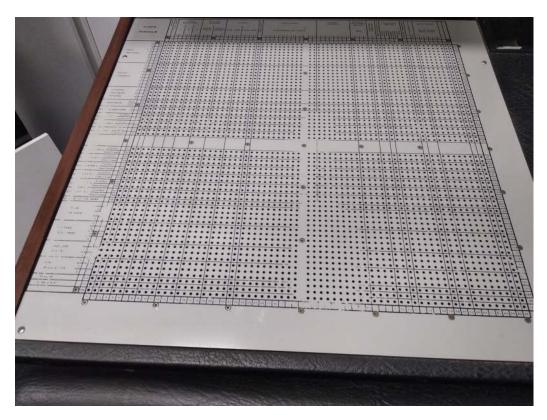


Figura 5.25: Matriz *Audio Control* del Synthi 100 del GME. En vertical están ordenadas todas las salidas de audio de todos los módulos, mientras que en horizontal se encuentran todas las entradas. Cada nodo representa una conexión única de una entrada con una salida. Los espacios que cruzan por el centro, así como los diversos grosores de las líneas dibujadas, ayudan visualmente a situar el nodo en las coordenadas deseadas. En la fotografía se observa sensiblemente el deterioro de los textos impresos debidos al uso continuado.

5. DESCRIP. DE LA AP.

Synths de origen y de destino, decidiendo si se ha de leer del bus de origen con el UGen In o InFeedback

## 5.17.1.1. Diseño de las matrices en el GUI de Synthi GME

Si bien se ha intentado que la apariencia gráfica de Synthi GME tuviera como base fotografías del Synthi 100 de Cuenca, incluyendo sus imperfecciones en el caso de haberlas, no ha sido viable seguir este principio en el caso de las matrices, debido a que su superficie brillante y su posición horizontal hacía muy difícil la toma de fotografías sin reflejos. Además, la franja superior que contiene los textos de las entradas queda oculta en la vista cenital por los paneles verticales, ya que las matrices están ligeramente metidas bajo estos.

Se optó por dibujar ambas matrices vectorialmente (fig. 5.26). El dibujado de los nodos, tanto los negros como los blancos, se realiza por código desde SuperCollider. Estos han sido creados desde cero, tanto su dibujado como su comportamiento ante los clicks del ratón. En todo caso, se ha seguido escrupulosamente las proporciones de las matrices originales, así como los diversos tipos de grosores de línea, espaciados, etc. El tipo de letra no es el mismo, si bien se ha buscado uno similar sans serif.

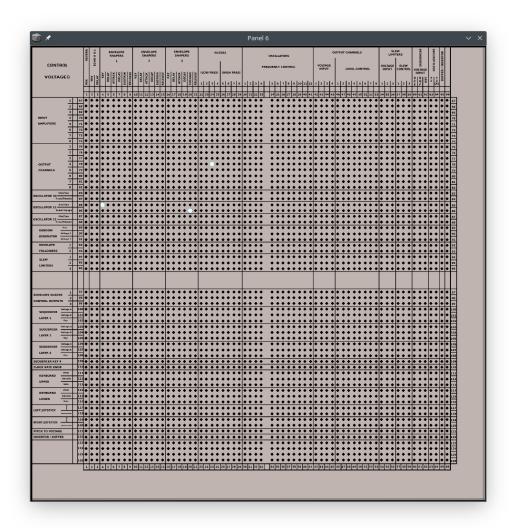


Figura 5.26: Matriz *Voltage Control* de la aplicación Synthi GME. Las matrices son las únicas ventanas cuya diseño no corresponde a fotografías del Synthi 100 de Cuenca, sino que son dibujos vectoriales copia del original. En la imagen se pueden apreciar tres conexiones representadas con puntos blancos.

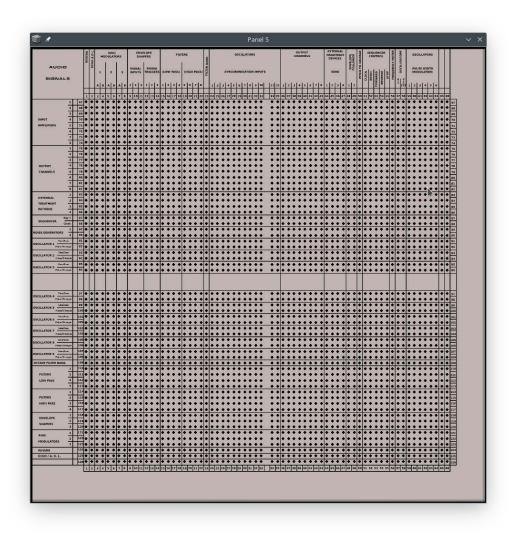


Figura 5.27: Matriz Audio Control de la aplicación Synthi GME.

# 6 | Futuro y perspectivas de la aplicación

## 6.1. «To Do»

«To Do» es el nombre que tradicionalmente se da en el mundo del software a la lista tareas futuras por realizar en el desarrollo de un proyecto, y simboliza el hecho de que todo programa informático está sometido a un interminable ciclo de vida. De un software existen diferentes versiones pero ninguna se considera un trabajo «terminado». Synthi GME se encuentra en una primera etapa en la que tiene una arquitectura definida, tiene la capacidad de aceptar más módulos, es eficiente y su ejecución no lanza graves errores ni excepciones.

La forma más sencilla de ver el estado de implementación en el que se encuentra Synthi GME respecto al Synthi 100 analógico es ejecutándolo (ver A.3) y observando los paneles 5 y 6, correspondientes a las matrices de *audio* y *voltaje* respectivamente. Por defecto, los nodos funcionales aparecen en un color más oscuro que los que aún no tienen ningún efecto. Estos paneles ofrecen así una información muy clara de qué módulos, qué entradas y qué salidas están implementadas y cuáles no. Las figuras 6.1 y 6.2 muestran el estado de implementación de Synthi GME en su versión 1.0.0.

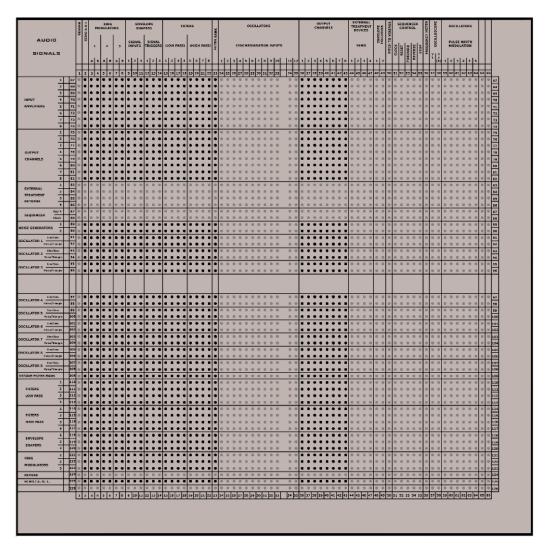


Figura 6.1: Módulos y nodos de *audio* en funcionamiento. Solo los nodos de color oscuro tienen una función asignada en la aplicación. Según se van implementando nuevos módulos, el número de nodos activos crece.

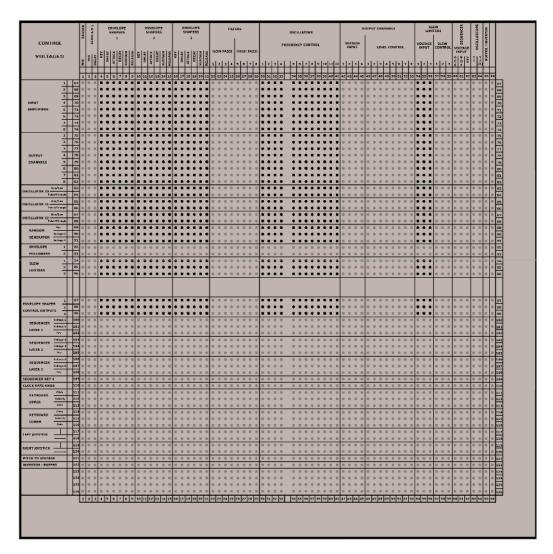


Figura 6.2: Módulos y nodos de control en funcionamiento.

# 6.1.1. Sistema de versionado

El número de las versiones de Synthi GME está compuesto de tres grupos de cifras separadas por un punto: [mayor].[menor].[micro]:

*mayor* Este número cambia cada vez que hay un cambio significativo en el software. También puede variar por la acumulación de muchos cambios menores o por razones puramente tácticas de cara a su difusión.

*menor* Se modifica cuando el software experimente cambios significativos pero que no llegan a merecer un nuevo número *mayor*, como la creación de un nuevo módulo, la implementación de conexiones de audio o voltaje a un módulo, una nueva implementación de un *Synth*, etc.

**micro** Este número indica cambios no significativos: refactorizaciones pequeñas, correcciones de errores no críticos, etc.

Synthi GME está en la versión 1.0.0 en el momento de entrega de esta memoria. En este caso, la razón por la que *mayor* es 1 no es otra que la de unir esta instantánea del programa con la presente memoria.

# 6.2. Continuación del desarrollo de Synthi GME

## 6.2.1. Mejoras en el código

Estos son los objetivos en los que se centrará el desarrollo del código de la aplicación:

- 1. Implementar todos los módulos del Synthi 100 del GME de Cuenca.
- 2. Crear un sistema de memorias tanto de combinaciones estáticas como de combinaciones de eventos en el tiempo.
- 3. Completar todas las entradas y salidas de los módulos.
- 4. Crear una interfaz de teclado MIDI para emular los dos teclados originales.
- 5. Mantener el código libre de errores.
- 6. Mejorar y refactorizar el código siempre que sea necesario, buscando la legibilidad y la eficiencia.
- 7. Simplificar al máximo las dependencias de software, como sc3-plugins.
- 8. Acceder al Synthi 100 de Cuenca con el fin de obtener muestras para estudiar el comportamiento de los diferentes módulos de cara a una implementación más fiel.

## 6.2.2. Interacción con el usuario

De cara a al usuario y la usabilidad de la aplicación:

- 1. Crear de ejecutables *standalone* para las tres plataformas de escritorio (Mac Os, Windows y Linux).
- 2. Testar en profundidad en las diferentes plataformas y obtener feedback de los usuarios.
- 3. Mejorar la interfaz para que sea intuitiva.
- 4. Sustituir las fotografías actuales por unas de alta calidad.

## 6.2.3. Difusión, formación e información

Este apartado es muy importante de cara al éxito de la aplicación. Los esfuerzos se centrarán eventualmente en los siguientes puntos:

- 1. Creación de un conjunto de videotutoriales de uso de Synthi GME.
- 2. Ofrecer la posibilidad de charlas y talleres divulgativos en torno a Synthi GME o la programación en SuperCollider.
- 3. Completar la documentación (Help) en SuperCollider.

# 7 | Conclusiones

Como final, una confesión. . . El primer objetivo de este trabajo no ha sido otro que el de aprender, y la motivación, la curiosidad; el mismo motor que siempre me ha movido en todas las direcciones en la vida. Y debe de haberse cumplido en el presente trabajo, a juzgar por tantos momentos de satisfacción que me ha entregado las incontables horas delante del ordenador, leyendo libros de síntesis analógica y digital o, de un modo muy especial, visitando el GME.

Entre los objetivos expuestos al principio de la memoria (sección 1.3), se incluye el de aportar a la comunidad una herramienta fruto de este trabajo. De nuevo he de decir, no sin cierta satisfacción, que en el camino recorrido he recibido más que he podido dar. Me he encontrado con personas muy generosas y de gran valía durante todo el máster, y, en particular, durante la elaboración del trabajo final. En la introducción reflexionaba en la importancia de la herramienta y en su gesto asociado para la producción artística, pero en la conclusión quiero fijarme en quien hace precisamente el gesto. Al fin y al cabo, los artefactos y sus productos son de y para las personas. Detrás de un sintetizador, de una composición sonora, hay personas que crean y personas que escuchan.

Mi proyecto no es más que un grano de arena en torno a la recuperación del GME de Cuenca y de todos sus fondos. Tanto si este trabajo resulta útil a alguien —ojalá— como si no, espero al menos haber levantado un poco de polvo con él. Hay mucha gente trabajando porque el GME vuelva a la vida y ahora, de algún modo, me siento parte de ese gran equipo.

# 8 | Agradecimientos

En primer lugar quiero agradecer su labor de dirección de este trabajo a José Manuel Berenguer Alarcón, quien desde el principio ha encauzado sabiamente mis ideas, me ha ayudado a darles forma, pero, sobre todo, por su actitud positiva, respetuosa y humana en todas nuestras comunicaciones. Gracias de corazón.

Un hueco muy especial en este trabajo lo ocupan Sylvia Molina Muro y Julio Sanz Vázquez. Sin ellos, simplemente este trabajo hubiera versado sobre otro tema. A ellos les agradezco el entusiasmo y generosidad con el que han apoyado este proyecto desde el principio.

También quiero dar las gracias al compositor Gonzalo Lacruz Esteban por sus continuas sugerencias bibliográficas de diverso tipo. Nos conocimos en el GME y ambos compartimos la admiración por éste y sus fondos. Gonzalo, jmucho éxito, muy merecido, con tu TFE!

A Kate Jetmore por la traducción del *abstract* al inglés. Más rápido difícilmente alguien hubiera hecho este gran favor; con más cariño y profesionalidad, sin duda, nadie.

Pero a quienes más debo, y nunca podré agradecer lo suficiente su amor y apoyo incondicional, es a mi mujer, Lucie, y mi hija, Ana Magdalena. Gracias por entregar vuestras horas —porque no solo eran mías— a este trabajo; por comprender mis ausencias en el proceso y, principalmente, por el apoyo en los momentos más difíciles.

# Anexos

# A | Instalación y ejecución de Synthi GME

Synthi GME está distribuido como un quark de SuperCollider, es decir, como un conjunto de clases empaquetadas y listas para ser instaladas en el sistema y ser usadas desde sclang, el lenguaje de programación de SuperCollider. Puesto que SuperCollider se distribuye para ser compilado en las plataformas más importantes de escritorio, a saber, Linux, Mac OS y Windows, por la misma razón Synthi GME es también multiplataforma. Las instrucciones que aquí se encuentran son comunes a todos los sistemas operativos, si bien, determinadas acciones, como la instalación o la compilación de SuperCollider, pueden variar en sus detalles, para lo cual se ha de acudir a los documentos de ayuda pertinentes en cada caso.

# A.1. Requisitos

git Si no está instalado ya en el sistema, ha de ser instalado para poder descargar y manejar los quarks: https://git-scm.com/

**SuperCollider** Es el programa principal. Contiene tanto el servidor de sonido como el intérprete del lenguaje *sclang*, en el que está escrito Synthi GME. La versión más baja en la que se ha probado que Synthi GME funciona es la 3.8. Puede ser compliado desde las fuentes:

https://github.com/supercollider/supercollider o desde los binarios ya compilados disponibles: https://supercollider.github.io/download. sc3-plugins Se trata de una colección de *UGens* de SuperCollider mantenidos por la comunidad. Puede ser compilado desde las fuentes:

https://github.com/supercollider/sc3-plugins o desde los binarios precompliados para cada sistema operativo desde el mismo repositorio.

# A.2. Instalación del quark de Synthi GME

Synthi GME se encuentra alojado en Github:

https://github.com/mesjetiu/SynthiGME, desde donde puede ser descargado todo su código fuente. No existen binarios instalables ya que es el propio intérprete de SuperCollider quien se encarga de compilar todas las extensiones y quarks instalados en el sistema en cada inicio.

## A.2.1. Instalación con la interfaz gráfica de Quark

Una vez abierto el editor de SuperCollider, abrir el administrador de quarks en el menú Language/Quarks (fig. A.1). Buscar en la lista alfabética el quark llamado SynthiGME. A continuación, basta con hacer click en el botón con el símbolo «+» que hay justo en la columna anterior al nombre. En la parte inferior de la ventana se ofrecen datos relevantes del quark, como su origen, una descripción, así como información sobre la instalación. Una vez instalado, es necesario recompilar las clases para poder usarlo, pulsando el botón Recompile class library. Con esto Synthi GME estará listo para ser utilizado.

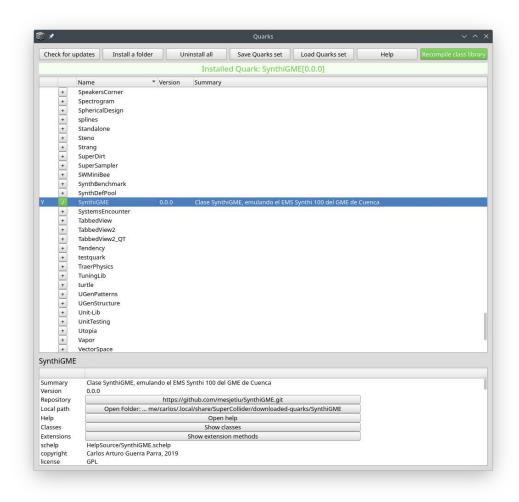


Figura A.1: Instalación gráfica del quark SynthiGME.

# A.2.2. Instalación con código

La forma más sencilla de instalar el Synthi GME en su última versión es ejecutando la siguiente línea de código en SuperCollider<sup>1</sup> (asegurarse de que se tiene conexión a internet):

```
Quarks.install("SynthiGME");
```

<sup>&</sup>lt;sup>1</sup>Para ejecutar un código en SuperCollider basta con poner el cursor sobre la línea (o hacer un bloque con las líneas que interesan) y pulsar [Control + Entrar]. Esta combinación podría variar en función del sistema operativo y la versión de SuperCollider, con lo que es conveniente informarse en cada caso.

SuperCollider buscará en la base de datos de *quarks* y localizará el paquete, que se encuentra en el repositorio de *Github* correspondiente. Si todo ha ido bien, se verá la siguiente salida en *Post window*:

```
-> Quarks
Installing SynthiGME
Adding path: /[user path]/.local/share/SuperCollider/
downloaded-quarks/SynthiGME
SynthiGME installed
-> Quark: SynthiGME[0.0.0]
```

En la última línea aparecerá entre corchetes la versión de Synthi GME instalada, que coincidirá con la última release en la rama master del repositorio de la aplicación.

Una vez instalado, es necesario recompilar todas las clases para poder usar las del *quark* recién instalado. Ello se puede hacer con la combinación [Control+Mayúsculas+L]<sup>2</sup> o ejecutando el siguiente código:

```
thisProcess.recompile;
```

# A.3. Ejecutar Synthi GME

Synthi GME es, en esencia, un conjunto de clases, de entre las cuales, solo una está diseñada para ser instanciada por el usuario: SynthiGME. Esta clase contiene el método run, que pone en marcha todas las rutinas de arranque de la aplicación: crea los busses de audio que comunican todos los módulos entre sí, todos los Synths, o «sintetizadores» de SuperCollider, las rutinas, etc. Configura el servidor de audio para que acepte el número adecuado de entradas y salidas requerido.

El siguiente código muestra varias formas de ejecutar Synthi GME:

```
// Se instancia la clase SynthiGME
// y se ejecuta su método "run":
    ~synthi = SynthiGME();
```

<sup>&</sup>lt;sup>2</sup>Esta combinación funciona en Linux, pero podría variar en función del sistema operativo o de la versión de SuperCollider.

```
4 ~ synthi.run();
5
6 // o, ambas sentencias en una:
7 ~ synthi = SynthiGME().run;
8
9 // Al introducir el objeto en una variable, podemos cerrar el programa ejecutando el método "close":
10 ~ synthi.close;
```

Si todo el proceso se ha producido satisfactoriamente, *Post window* mostrará una salida similar a esta:

```
Conexión de salida stereo canales 1 a 8...OK
1
2
  Conexión de salida stereo canales 1 a 4...OK
  Conexión de salida stereo canales 5 a 8...OK
3
   Conexión de salida de cada canal individual...OK
4
  Output Channels...OK
5
  Filters...OK
6
7
  Octave Filter Bank...OK
8
  Ring Modulators...OK
9
  Echo A.D.L...OK
10 | Noise Generators...OK
  Random Voltage Generator...OK
11
   Slew Limiters...OK
12
13
  Oscillators...OK
14 | Envelope Shapers...OK
15
   Input Amplifier Level...OK
16
  Conexiones en Patchbay de audio...OK
17
   Conexiones en Patchbay de voltage...OK
   Conexión de entrada Input Amplifiers, canales 1 a 8 a puertos
18
       de SC...OK
  SynthiGME en ejecución
19
```

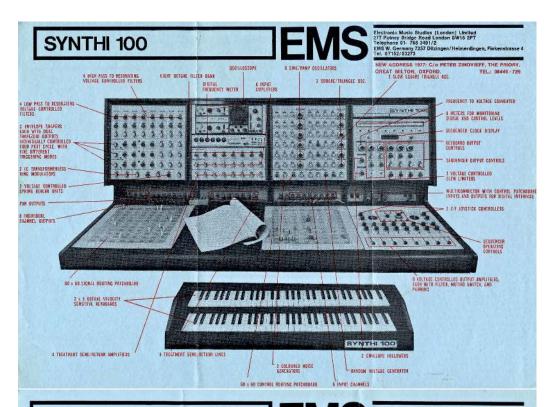
Es importante notar que el proceso de arranque reiniciará automáticamente el servidor de sonido para asegurar que todos los cambios en la configuración surten efecto. Esto significa que si se requiere ejecutar más código de SuperCollider simultáneamente, este ha de ser ejecutado después que Synthi GME.

En pocos segundos se abrirán 7 ventanas ordenadas y adaptadas al tamaño de la pantalla, representando a los 7 paneles del Synthi 100 del GME de Cuenca. A partir de aquí, si *Post window* no señala ningún error o mal funcionamiento en la ejecución, el programa ya está listo para ser usado.

# B | Synthi 100 Sheet Data

Anuncio original del model Synthi 100 de EMS, con las especificaciones técnicas más importantes.

 $Extraído\:de\:https://whitefiles.org/rwz/zxe/1977\_synthi\_100\_sheet.pdf$ 



# SYNTHI 100

The SYNTHI 100 is the most formidable electronic music system ever devised. It has as its heart a digital sequencer, which is in fact a small special purpose computer, complete with analog-to-digital and digital-to-analog converters. This device enables the operator to load, in his own time, up to six independent tracks of control voltage data, plus attack and switching pulses, then hear it played back, forwards or in reverse, at any speed, All events can be individually examined by stopping the clock, and edited or erased. This sequencer is years ahead of its time.

The SYNTHI 100 is easy to set up (two pin the controls, but to save space this is not mentioned in the following descriptions of each item.

examined by stopping the clock, and earned or erased. This sequence is years ahead of its time.

\*\*Mith All devices have output level controls, but to save space this is not mentioned in the following descriptions of each item.

\*\*Mith All devices have output level controls, but to save space this is not mentioned in the following descriptions of each item.

\*\*Mith All devices have output level controls. Out of save space this is not mentioned in the following descriptions of each item.

\*\*Mith All devices have output level controls. Out of save space this is not mentioned in the following descriptions of each item.

\*\*Mith All devices have output level controls. Out of the following from the controls. Output specificate when vectorially of the controls on all signal producing devices to facilitate accurate balancing, and phase locking on all oscillators for additive as well as subtractive "filler synthesis." As above but shaping on both outputs, giving square — rectangular — pulse, and rising ramp — triangle — falling ramp plusted controlled, make possible amazingly souther studies equipment, using eight voltage from voltage controlled line output channels, eight line input channels, as well as four send and return lines for treatments outside the synthesises. This actionaling machine can also be operated entirely automatically by a computer. The system, known as the COMPUTER SYNTHI, can store, process and deliver all the control voltage information which manipulates the various devices.

At it is simplest it is a vast mutulilayer, highly intelligent sequencer. This opens up the

THREE ENVELOPE SHAPERS WITH DUAL TRAPEZOID OUTPUTS:
Two manually controlled outputs from Four section trapezoid cycle: Delay, Attack, stick.

On, Decay

V-Controllabletime (each section): 2m5-2OS.

Five Triggering Modes: Signal Threshold,
Hold On, Single Shot, Free Run, Gated
Free Run.

Voltage Outputs: As well as modulating an
audio signal, each shaper can produce two
different control voltages.

# FOUR LOW PASS TO RESONATING FILTERS:

FILTERS: Frequency Range: 5Hz-20KHz. Voltage Control: 0.5V/Octave. Maximum 0: 20. Cut-off Rate: 12dB/Octave for first octave, then 18dB/8ve.

FIGHT VOLTAGE CONTROLLED OUTPUT
AMPLIFIERS:
Each with slide fader, output panning,
variable first order low to high pass litter,
output disconnect switch, and meter
switchable to read AG or DC level.
Voltage Control: 0.5V/Octave.

TWO 60 x 60 PIN MATRIX PATCHBOARDS:

# EIGHT AC/DC INPUT AMPLIFIERS: Maximum Line Input: 1.8VAC (rms) or ±2.5VDC.

FREQUENCY-TO-VOLTAGE CONVERTER

# TWO ENVELOPE FOLLOWERS:

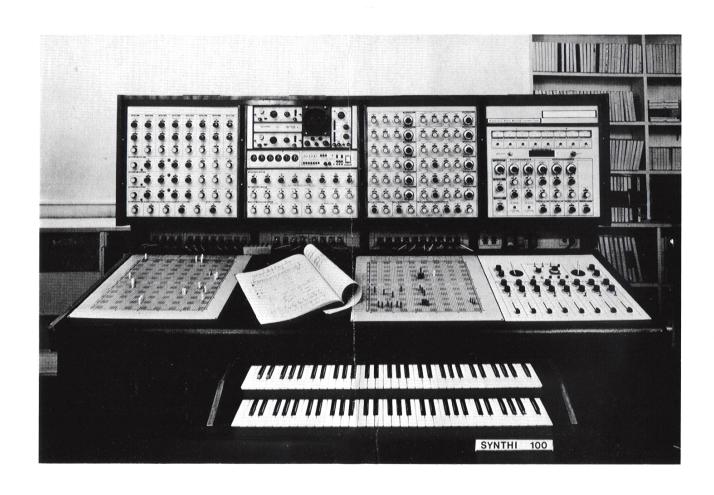
DOUBLE BEAM OSCILLOSCOPE: DIGITAL FREQUENCY METER/TIMER/ COUNTER:

# FOUR EXTERNAL TREATMENT SEND AND RETURN LINES: For interfacing with other studio against

For interfacing with other studio equipi MULTIWAY PLUG FOR EXTERNAL CONNECTION OF CONTROLS All control patch connections are avail at these sockets, making possible e.g. connection to a computer. All line connections are to Cannon sockets.

# C | Folleto del EMS Synthi 100

Folleto con especificaciones técnicas detalladas del modelo Synthi 100, publicado por EMS (1971).



#### The Synthi 100

#### **Professional Electronic Music Studio**

The SYNTHI 100 studio has at its heart a digital sequencer, which makes conventional analogue controllers, with their gigantic array of knobs, seem old-fashioned, inaccurate and cumbersome. With a solid state storage capacity of 10,240 bits, the new sequencer is capable of precisely controlling 6 different simultaneous parameters over a sequence of 256 successive events. There are several modes of operation and full, easy to operate editing facilities, so that any or all of the 256 stored items and their time relationships may be changed without difficulty.

For example, two five-octave dynamically proportional keyboards are included, to operate the studio in real time, on six tracks, with the sequencer remembering what is played. This performance can then be played back backwards or forwards, at any speed, and edited to any degree of precision, prior to recording on magnetic tape.

magnetic tape.

The SYNTHI 100 also contains new electronic devices exclusive to EMS, such as voltage controlled slew limiters, a frequency to voltage converter and a two-output random staircase generator with controllable time and amplitude variances. Also included are a very full complement of 12 drift-free oscillators, eight dynamic filters and three transformerless i.c. ring modulators (which can be cascaded for double and triple modulation), as well as eight voltage controlled output channels with full panning facilities, eight input amplifiers, two X-Y joystick controllers, a filter bank, three elaborate envelope shapers and followers, noise generators and reverberation units. A double-beam oscilloscope and six-digit crystal-controlled counter/timer/frequency meter ensure accurate setting up and logging of parameters, and patching is by cordless pin matrices (two boards each with 60 x 60 locations).

In the same price range as the larger voltage controlled synthesizers, the SYNTHI 100 offers far more, and is intended for really exacting composition and realisation work in professional recording and electronic music studios, broadcasting companies and universities. First customers include the BBC, Radio Belgrade and the University of Wales.

This photograph shows the EMS Synthi 100. The patching matrixes are on the main desk; the right hand matrix is used for control voltages and is connected to the Digital Interface via two 60 way junctions. The knobs above the desk are used as presets when the SYNTHI is controlled by computer.

#### **General Points**

#### **Appearance**

The SYNTHI 100 is presented in an afrormosia case, with padded black leather. A cupboard stand and seat are available as extras. The working surfaces are illuminated by dimmer lights, and all outputs and inputs are available behind a sliding screen in the front of the unit, while the input/output cables are channelled on to the rear. The SYNTHI 100 is designed for computer interface, and therefore all control inputs are available for direct connection to digital/analog converters on multiway connectors. The overall size base of the SYNTHI 100 is 79 in, length by 37½ in, depth, the height is 33 in, and the height from the underside to the ground is 30 in.

#### Signal Levels

The levels of all signal outputs available at the patchboard are controlled by panel mounted potentiometers. In fact, these potentiometers provide control voltages which operate on voltage controlled amplifiers. The advantages of this system include:

Constant low output impedence.

No 'fader scratch' as levels are changed

Less conveying of signals at high impedance (where they are susceptible to crosstalk)

Much simplified wiring system

Ideal logarithmic control of level

In general, signal levels are about  $\pm 1V$  p-p, although most outputs can deliver much more than this. Most devices which have signal inputs are adjusted for optimum operation at this level, although it can be usually considerably exceeded without serious distortion.

#### Control Voltage Levels

All EMS control voltage outputs are bipolar, that is to say they go both positive and negative. This allows control signals and audio signals to be interchanged, and also allows the magnitude of a control output to be adjusted without altering the mean level of the parameter controlled.

### Input Impedances

All input impedances are approximately 10 KOHM. This comparatively high figure was chosen as a compromise between the conflicting requirements of 'fan out' (i.e. the number of inputs an output may drive before it runs out of power) and susceptibility to crosstalk when an input is left open circuit. Screened cable is used for all signal paths in the studio, and impedance at certain central inputs is made infinitely low so that secondary control parameters can be added without readjusting the magnitude of the primary ones.

#### Noise Levels

The signal to noise ratio of all voltage controlled amplifiers is better than 74 dB for an input signal level which produces a T.H.D. of 3%. This ratio is constant over the upper 30 dB of the control range. In other words, as the amplifier gain is reduced, the noise is proportionally reduced as well. When the amplifier is cut off, the noise at its output is immeasurable.

#### Specification

#### Source Modules

#### Three Voltage Controlled Audio Waveform Generators, Sine and Ramp

Manual Frequency Range: Greater than 1Hz - 10 KHz (extendible by voltage controls in both directions — to 0.25 Hz and 20 KHz).

Sine Purity: Better than 50% total distortion between 10 Hz and 10KHz.

Ramp Output Linearity: Departure from linearity  $\pm$  1% of best straight line between 10Hz and 10KHz.

Voltage Control: 5v/octave. Accuracy 0.3% departure from best straight line between 100 and 2000 KHz. Frequency Stability: Generally better than 2% from month to month, but the oscillators usually hold their

setting to within 2 cycles in a thousand during a working session.

A sine shaper is included by which variable amounts of even harmonic distortion may be added.

#### Three Voltage Controlled Audio Waveform Generators, Triangle and Square.

These can be varied from triangle to sawtooth ramp, and from symmetrical square to short pulse, in either

Manual Frequency Range: Greater than 1 Hz to 10 KHz. Triangle Symmetry: ±5% rise time to fall time equality. Other specifications as for sine/ramp oscillator.

# Three Voltage Controlled Low Frequency Waveform Generators

Same details as before, but oscillators are twenty times

Frequency range: Greater than 0.025 Hz (40 secs. per cycle) to 500 Hz.

Voltage Control: .5v/octave.

These three oscillators are intended mainly as control sources, but can be used for tone generation at the upper end of the range.

All nine of the above oscillators have synchronisation inputs so that they can operate at an integral multiple of another oscillator, providing a huge variety of waveforms which can be used in additive synthesis.

#### Three Noise Generators.

Variable from white (central position of colouration control) to dark or light positions (low or high pass filters).

Distortion: In white position, frequency content is flat  $\equiv$ 3 dB from 100 Hz - 10 KHz. We recommend several noise sources, because with

We recommend several noise sources, because with different filterings more than one can be used for different purposes.

#### Dual Output Random Control Voltage Source

This device produces two control voltages which move abruptly from one level to another. The distribution of levels is rectangular rather than Gaussian, and the two outputs are uncorrelated in level, but synchronous in time. The mean time between changes, and the variance about that mean are manually controllable. The distribution of times is rectangular, and in common with all other time controls devices in the studio, a control range of at least 1000:1 is available.

#### Controls:

Amplitude Variance. Up to 2.5v symmetrically positive and negative (x2, each output separately controllable).

Time Range: From approximately 10 secs to 10 ms if no time variance is applied.

Time Variance: From equal steps to 1000:1 variance. Effect of variance control is obviously limited when time range control is at either entrance.

#### Three Voltage Controlled Trapezoid Generators with Integral Envelope Shapers

These devices might be described as voltage programmable four segment curve synthesizers, the output being available as two control voltages and as the modulation of an audio signal.

The basic waveform produced (at each control output) has four stages:

- Delay. The output remains constant and negative for a controllable time after the generator is triggered.
- 2 Attack. The output rises to a fixed positive value at a controlled rate.
- 3 On. The output remains constant and positive for a controllable time after the completion of 'attack'.
- 4 Decay. The output falls at a controlled rate to its initial value.

In addition, there is a second output which lags behind the first by one quarter of a complete trapezoid cycle. Thus the time set for, say, 'on' in respect of output 1, becomes the time for 'attack' from output 2, and so

The amplitude and polarity of both outputs may be adjusted independently so that if they are summed (on the patchboard) any continuous four line function which ends at the value at which it starts may be produced. This arrangement gives an extraordinary flexible range of envelopes.

The envelope shaper portion consists of a logarithmic voltage controlled amplifier permanently connected to one of the trapezoids.

The overall control of each cycle of operation may be in a number of ways selected by a switch, having the following positions:

- Signal threshold. Any signal above a certain level initiates a single cycle.
- Hold sequence starts when control is positive. Decay does not start until control goes negative.
- 3 Single shot positive zero crossing initiates a single
- Free run.
- Gated free run positive level allows the sequence to free run. Sequence stops at the end of a cycle when level goes negative.

Initial delay time 2 ms to 20 sec. Attack time On time 2 ms to 20 sec. 2 ms to 20 sec. Decay time

(these parameters may also be voltage controlled over their entire range.)

Trapezoid phase 1 output level (centre zero knob) Trapezoid phase 2 output level (centre zero knob) Signal level control

Push button (to initiate cycle)

Trigger mode selector switch

Envelope shaper is logarithmic to within 3 dB over its 60 dB range.

Voltage control function of time parameters is ideally exponential to within 10% (of dependent parameter) over a range of 1000:1: Departure from ideal is gradual beyond these limits. This permits a single voltage applied to all inputs to compress the time scale.

#### Four Voltage Controlled Filter/Oscillators (Low Pass to Resonating)

Operation as a sine source.

Frequency Range: Greater than 5 Hz-20 KHz.

Sine Purity:

Better than 3% total distortion between 10 Hz and

General Noise Figure for Oscillators: Spurious outputs not greater than 0.1%.

As filters, they are adjustable from Low Pass to Resonating Filters, covering the entire sonic range.

Frequency Range: Greater than 5 Hz to 20 KHz.

Low Pass Position.

Cut off rate 12 dB for first octave and 18 dB per octave thereafter.

Resonator Position

Maximum stable 'Q' factor-20.

Accuracy of Exponential Voltage Control Function: = 1% between 100 Hz and 2000 Hz.

Note: Operation as a voltage controlled oscillator is limited by the time taken to respond to an abrupt change. Maximum slew rate is about 2 ms per octave.

#### Four Voltage Controlled Filter/Oscillators (High Pass to Resonating)

Similar, but complementary to Low Pass Filters.

#### One Octave Filter Bank

This consists of eight resonating filters, fixed-tune one octave apart, in the range 62.5 Hz-8 KHz, seperately

#### Two Voltage Controlled Reverberation Units

Each spring unit has two elements with delays of 35 and 40 ms.

Maximum Reverberation Time: 2.4 seconds.

Useful Frequency Range:

30 Hz-12 KHz.

Voltage Control Range: ±2v from no reverberation to maximum reverberation.

## Three Voltage Controlled Slew Limiters

This device is a unity gain amplifier in which the output exactly follows the input at a rate whose maximum (slew) is defined by a control voltage. One application might be to interpose the device between the pitch control voltage from a keyboard and the oscillator whose pitch is to be controlled. If the key velocity voltage were then applied to the slew control input, the player could produce a glissando between any two notes that he played, the rate of glissando being controlled by his touch.

Steady State Gain: Steady State Linearity:

1 ± 1% = 0.05% (BSL)

Range of Slew Control: 1 ms to 10 sec.

Voltage control of slew is exponential.

Note: Unlike all other devices, no output level control is provided as the device has unity gain.

#### Three Integral Circuit Transformerless Ring Modulators

These very efficient modulators also include amplifiers, and can therefore be used in series for double or triple modulation.

Maximum Input for Undistorted Output: 1.5v p-p to each input Breakthrough with 1.5v on one input only: 5mV p-p (—60 dB)

# One 256 Event, 6 Simultaneous Parameter Digital Sequencer

This machine is, in fact, a small special purpose digital computer, complete with analog to digital and digital to analog converters. It provides a sequence of control voltages which may be used on any of the devices in the studio.

The operation of the sequencer may most easily be described in terms of conventional music, although it must be remembered that the design by no means limits it to this kind of operation.

The sequencer stores 256 'notes' and plays each note at the correct time and for the correct duration. It simultaneously provides two voltages, one of which might be used to define pitch and the other loudness. It is capable of controlling three voices, each with duration, pitch and loudness. The 256 note storage may be distributed to each voice in any proportion, for instance, 254 notes may go to one voice and one each to the others. In fact, the second and third voices need not be used as such, their voltages could be used to control parameters (filtering, decay time, etc.) of the first voice.

The information which is to be stored and subsequently reproduced is presented to the machine as control voltages, which are most easily supplied from the keyboards.

keyboards.

All timing data is entered by playing the keys. To record a sequence the composer sets the speed of a clock and starts it running. As he plays each note, the machine at that instant records how many clock pulses have elapsed since the start of the sequence, and how many during the time that the note was held down. It simultaneously remembers which note on the keyboard was pressed and with which velocity it was struck. (The second parameter, or, indeed, the first, could equally be derived from any voltage source.) The composer continues until he has recorded perhaps sixty notes. He then restarts the clock and turns a switch which tells the machine that the next notes he plays will be directed to the second envelope shaper. While recording each sequence, he can simultaneously hear the results of what he has previously recorded.

The machine may then be set to the edit mode. In this mode the sequence may be advanced at any speed, or step by step, so that each note may be modified or erased. A special feature allows time to be reversed. A control rather like the spooling knob on studio tape recorders allows one to go forwards or backwards at any speed. Unlike a tape recorder, however, there is no inertia in the system, so that one can quite easily 'zero in' on a particular note.

#### Sequencer Controls

Ten controls are provided to adjust the amplitude of the sequencer's output voltages, and a further ten supervise the actual operation.

#### 1 Range of Layer 1 Output Voltage A

This control is a slow motion dial calibrated 0-100. If Voltage A were used to control the pitch of an oscillator, then this control could be used to define the musical interval for each step of the output. At 25, for instance, a range of sixty-four quarter tones (covering about 2½ octaves) will be available

#### 2 Range of Layer 1 Output Voltage B

The range of the second parameter for each event may similarly be adjusted with this control.

#### 3 Range of Layer 1 Keying Voltage

This voltage, which is positive for the duration of each note in Layer 1, would normally be used to control the envelope shaper. It can also be used to assist in the synthesis of certain instrumental sounds. For this reason, a centre zero control is provided. This inverts the polarity of the voltage when it is counterclockwise.

4 5 and 6 are controls identical to 1 2 and 3, except that they apply to the second layer. Likewise 7, 8 and 9 which apply to the third layer.

#### 10 Range of 'Key 4'

A fourth kind of event may be recorded which is similar to the three layers, except that there are no parameter control voltages available with it—just the keying voltage. It is primarily intended to stop or reset the sequencer's clock, allowing one to produce a single finite sequence or a repeating pattern. If not used for this purpose it might be used in conjunction with a slew limiter and a voltage controlled amplifier to initiate a crescendo or a number of other things.

Note that all controls, 1-10, can be adjusted after the sequence is entered, without changing the basic data.

#### 11 Clock Rate

This slow motion dial is a centre zero control in a rather special sense. When it is less than halfway, the sequence runs backwards. It controls the clock rate over a range of  $\pm 1000:1$ . That is to say, with the control near its centre position, the clock pulses occur at about 2 per second, allowing a total sequence length of over six minutes. In this case, however, the resolution in time of each event is only half a second. The control has a distinct dead space around half-way, which prevents the clock from 'drifting' during editing. A voltage proportional to the absolute clock speed is available at the control patchboard, so that it can simultaneously control all time variant parameters — envelope shapers, slew limiters and even oscillator frequencies — as the clock rate is adjusted.

#### 12 Note Distribution

This control is a four way switch; it tells the machine which of the three layers is being recorded, so that on replay the voltages will appear at the appropriate output. The fourth position denotes 'Key 4' as described above.

#### 13 Stop at each note

When this toggle switch is down, the sequencer clock stops at the start of each note that it reproduces.

#### 14 Stop at end of note

Similar to 13, except that the sequencer stops at a time corresponding to when the key was released as the note was recorded.

These switches operate in conjunction with the note distribution switch, in that they only stop at a note in the layer defined by the latter.

The purpose of 13 and 14 is to facilitate the editing of events after a tentative sequence has been entered.

#### 15 Erase note button

While this button is pressed, any notes in the layer selected by the note distribution switch which start at the time shown by the clock display, will be erased.

#### 16 Clear memory button.

This is the 'bulk erase' button.

#### 17 Reset button

This button sets the clock to zero, and holds it there as long as it is pressed. It does not stop the clock, it simply restarts the sequence from the beginning.

#### 18 Start button

This button allows the clock counter to start or continue counting. It would be used continually during editing, to advance the sequence note by note.

#### 19 Stop button

This button stops the clock from counting.

Note that 17 18 and 19 are momentary action pushbuttons, not switches. They roughly correspond to the controls on a stop-watch.

When the sequencer clock is driven from pulses previously recorded on tape, or, indeed, from any external source, it will ignore them until the start button is pressed. Remote operation of the sequencer is facilitated by electrical inputs at the signal patch-board.

#### 20 Rewrite B, D, F

When this switch is down, the second parameter voltages in each layer may be rewritten without disturbing the first parameter, or the event timing.

Note: In this section, 'Note' is used for musical convenience, but, it must be remembered, can be used for any parameter which has been selected.

### Summary of Specification

Total storage capacity: 10,240 bits (of which 9,216 bits are normally used).

#### Organisation of Data

36 bit words — each word representing one event.

Start-of-event time (referred to start of sequence) 10 bits

End-of-event time (referred to start of sequence) 10 bits.

Selection of one of three envelope shapers and one pair (out of three pairs) of digital analog converters. Also internal functions 4 bits.

Data, for digital analog converters 2 x 6 bits.

#### Details of Coding

The 10 bit event time allows the start of each event to be defined to an accuracy of 1 part in 2 to the power 10 (viz. 1024). Thus, if the clock is set to a rate of, say, one hundredth pulses a second, each event may be adjusted forwards or backwards in increments of one hundredth of a second. The total sequence length would be ten seconds.

The 'end of event' time, i.e. the time at which the key is released, is similarly recorded. Thus three control signals are reproduced, each being positive during the duration of a note intended for one of the three layers of the sequence. They are available at the patchboard as switching voltages which would normally go to the supervising inputs of the envelope shapers.

#### **Digital Analog Converters**

Of the six converters, three are of accuracy appropriate to exact control of pitch on the diatonic scale. Six bits give a range of 64 notes. If greater range and/or finer resolution is required, then the output of the second converter may be added to that of the first. In this case, the player might use one keyboard to define a note on the diatonic scale, and the second to raise or lower that note by increments of one thirty-second of a tone.

The precise converters are accurate to  $\pm 0.15\%$  (BSL). The second parameter converters are accurate to  $\pm 0.78\%$  (BSL).

## Eight Multifunction Output Amplifiers

These amplifiers are primarily intended to be the last link in the signal chain before the tape recorder or monitor, but they provide certain subsidiary functions which will make them otherwise useful. All eight are voltage controlled (0.5V per 6 dB).

#### Controls

Level: Slider type fader.

Pan: A knob which distributes the output to between the left and right bus, these being common to four of the eight amplifiers.

Filter: A single knob providing continuous transition between first order low pass and first order high pass. Off Switch: Totally disconnect output from the pan control, allowing the amplifier to be used earlier in the signal chain.

Meter Switch: The meter may be used as a centre zero DC voltmeter, or as an AC level meter.

## Two X-Y Joystick Controllers

These give continuous control of two parameters together, which is very useful in live performance. The control sticks have a range of 2 x  $\pm$ 2V DC.

#### Two Five Octave Dynamically Proportional Keyboards

Five octave keyboards giving precise divisions of pitch or any other controllable parameter. In the case of pitch, the range would give anything between 4 and 40 notes per octave. This is useful for microtonal work. By setting 12 notes per octave, the keyboard can be used as a normal melodic source.

A second voltage output is proportional to touch — actually the velocity with which a key is struck.

A third voltage switches positive when one or more keys are pressed. Note that the keyboard produces only one pitch voltage at any instant; when several notes are pressed, the voltage of the highest appears. Both the pitch voltage and dynamic voltage are 'remembered', even when a key is released.

Keyboard voltage: 0.5V per octave maximum, accurate to better than 0.15% at all points.

Dynamic voltage: ±1.5V depending on key velocity. Output function was synthesized to be a compromise between a strict proportionality to velocity over a range of about 100:1, and a function that would distribute seven subjectively equal increments of playing force evenly over the output voltage range. The keyboard feels most natural when the dynamics voltage is used to control a modulator over a 40 dB range.

Eight Way Fading/Panning Console (See Multifunction

60 x 60 Pin Matrix Patchboards (7,200 Pin Locations) These patchboards allow any input to be connected to any output by the insertion of a single cordless pin. Each output is connected to a row of sixty horizontal holes. The holes appear as a square array of 3,600 (x2) cross-points, in any of which a jack may be inserted. The jacks contain resistors so that several outputs may be mixed into a single input. All device outputs are fed to the board at a low impedance, blocking any reverse signal paths.

Two patchboards are provided, one intended for control signals and one for audio signals. A small number of interconnection patches between the patchboards are hard wired, as some signals can be used in both domains.

It is also possible to route external signals to the patchboards by using the jacks in the conventional way. All contacts, including the jacks have a surface coating of silver.

## Eight AC/DC Input Amplifiers

Maximum distortion at rated inputs: 0.1%.

Input Sensitivities: Line Input: maximum 1.8V AC (rms) or ±2.5V DC

These amplifiers convert input signals to a suitable ritiese amplifiers convert input signals to a suitable level and impedance to feed treatment devices. The line inputs are directly coupled and are therefore suitable for both signal and low frequency or DC control inputs. Two separate microphones amplifiers are supplied, which can feed any two of the above channels.

## Four External Treatment Send and Returns

Provision for sending out to external echo plates and other equipment, and returning to the Studio.

## One Frequency to Voltage Converter

This device accepts inputs from a variety of sources, including acoustical instruments (via a microphone or pickup and pre-amplifier) and produces a voltage proportional to the fundamental pitch of the note played. Sophisticated analog circuitry is incorporated to remove overtones, provided that their energy constitutes no more than 90% of the total signal.

Unlike conventional frequency measuring techniques, which count the number of zero crossings of a wave-form in a fixed interval of time, the converter measures the period of the signal and transforms this data to a voltage which is compatible with the other devices in the studio. The advantage of this method is that an accurate measure of the pitch can be made in a much shorter time. The output is gated into a track and hold buffer by a discriminator, which suppresses spurious outputs when the signal is dying away.

A single output control adjusts the range and polarity of the output voltage.

#### Two Envelope Followers

These devices produce a voltage proportional to the mean level of an audio signal. The output is passed through a second order low pass filter to remove ripple while keeping a fast response. Cut off is about 50Hz. Output amplitude is adjustable by a centre zero knob to give positive or negative excursions of up to 1 volt per 6 dB.

#### Items from other Manufacturers Dawe 3000 AR/6 Digital Frequency Meter

Brief details as follows:

Crystal frequency:

100 KHz + 0.002% Range — 0-1 MHz Frequency Measurement:

Accuracy —  $\pm 1$  digit  $\pm$ 

crystal accuracy Gating Time — 1 ms to 10

seconds

Period Measurement:

Range — 0-300 KHz Time Units 1 10 s to 10 ms Gating Period — 1/7 to

1,000 cycles of input frequency

Time Measurement:

- 10 s - 10 secs Range -(nearly four months)

#### Telequipment Double Beam Oscilloscope, D43R

Rack mounted double gun laboratory oscilloscope, with 6 x 8 cm display area.

Electronic Music Studios (London) Ltd. reserve the right to vary the specification and/or price of the SYNTHI 100 studio without notice, should it be necessary or desirable.

#### Why a Computer?

In recent years the most important single problem in designing an Electronic Music Studio has been finding a way in which reasonable complexity is achieved without an absurd proliferation of manual controls. Essentially, the problem is one of storing and manipulating data, and for a number of reasons a small computer is suited to this task.

A large quantity of information can be stored. Using a 12 bit word, the EMS configuration can store 4,000 events in the computer's core store, 32,000 events on the magnetic Disc and over a quarter of a million on each reel of magnetic tape.

There is no redundant information. 64 levels of control are adequate for most devices and EMS use 6 data bits as standard (the other 6 bits are used to specify the device). Controls, such as oscillator frequencies for which precision is required, use converters in pairs giving a resolution of 10 bits, corresponding to 0.1 semitone over the audio range, which is adequate for glissando.

The computer is fast enough for all musical requirements. When necessary, it will perform calculations during the performance without the risk of degrading the music

Composers can do much of their preparatory work away from the studio. They can write the score as a computer program and realise it as sound very quickly when they come to the studio.

When a composer leaves the studio, all control settings are stored digitally on his own tape, and when he returns he can start from exactly where he left off regardless of who has used the studio in the meantime. This is particularly advantageous in a University studio, where many students will be allowed short periods in the studio.

A composition stored digitally does not deteriorate as an audio recording does. A piece can be re-recorded at any time from the original program.

## What Equipment is needed?

A computer studio has conventional voltage controlled equipment (oscillators, filters, modulators, etc.), a Computer to control them, and an interface which converts the digital computer information to voltages for device control. In addition, programs for the computer are required, and these are referred to as the software of the studio: EMS provide a studio consisting of an EMS SYNTHI 100 Synthesizer; a set of digital to analogue (DAC) and analogue to digital (ADC) converters; a crystal clock; and a computer system. The system can be enlarged at any time by adding analogue or digital equipment. The equipment provided by EMS does not include tape-recorders, monitoring amblifiers or loudspeakers. EMS will negotiate a price and delivery time based on the following equipment:

EMS Synthi 100

This photograph shows the original computer studio in Putney. The computer and its typewriter are at the left hand side and the Digital Interface is on the fourth equipment rack, underneath the oscilloscope. In this studio the computer is used to control devices in the centre racks which were built before the SYNTHI.

EMS Digital Interface and Clock PDP8 Processor, 4K Core, 32K Disc and Teletype EMS Software

#### What can it do?

The Synthi 100 is a comprehensive e.m. studio by itself

The MUSYS software provided by EMS enables compositions to be stored, modified and performed by the computer

Miscellaneous chores such as tuning oscillators and measuring frequency responses can be performed by the computer.

Appropriate software is provided by EMS and additional programs can be written by the user, using the DEC assembly system.

An added bonus is that several hundred programs are available free from the computer manufacturer (DEC) which enable the computer to be used for non-musical tasks such as calculation, accounting, circuit-design, or even playing games.

#### Why Musys?

EMS bought its first computer four years ago, and the first public appearance of the computer was in the Royal Festival Hall in January, 1968, performing 'Partita for Unaccompanied Computer' by Peter Zinovieff. Later in the same year, the system was the most spectacular exhibit at the Cybernetic Serendipity exhibition at the ICA, where it improvised and accompanied melodies whistled to it by visitors. At this time, each piece realised ir, the studio required a separate program, and it became apparant that each program had many things in common which could be incorporated into a single generalised program. Peter Grogono proposed that a translation program should be available to relieve the composer of the task of writing down long strings of numbers, and designed MUSYS. MUSYS 1 and MUSYS 2 were both short-lived, but they provided valuable experience and led to MUSYS 3 which was completed in early 1970 and is still the principal program used. The first two compositions which used MUSYS 3 were "Medusa" by Harrison Birtwistle and "A Lollippo for Papa" (variations on a sonata by Hadyn) by Peter Zinovieff, and the language was taught to fifteen musicians who attended the BSEM course at Putney in June 1970. Since then, MUSYS 4 has been written to control the studio's Filter Bank, and MUSYS 5 is being prepared for Autumn 1971 to allow larger compositions to be realised in one piece.

#### How is it used?

The system can be used as a Sequencer, in which case no programming is necessary: the Digital Interface has controls which permit data for each device in the SYNTHI 100 to be stored, examined, altered and played back. Alternatively, MUSYS 3 can be used, in which case the composer writes a program which constitutes the "score" of his piece. Very short "scores" may be typed directly on the computer's typewriter, but this technique is normally used only to try out ideas or test the devices. The score is compiled by the MUSYS Compiler, a program which translates the composer's

score into a string of numbers for delivery to the SYNTHI 100. At each step, the process can be closely controlled by the composer or interpreter, using both hardware and software.

The requirements for a programming language for a system of this kind are simplicity, efficiency and flexibility; EMS feel that MUSYS meets these requirements adequately. In a MUSYS score, the "instruents" are given abbreviated names, so that, for example, 02., E3., F7., refer to Oscillator 2, Envelope Shaper 3 and Filter 7 respectively. In the simplest case, other is exert to a degree simple by writing a number Shaper 3 and Filter / respectively. In the simplest case, data is sent to a device simply by writing a number after the device name — 02.15., F7. 30., etc. The effect of the number depends on the device — for an oscillator it represents a pitch, for an Envelope Shaper the length of Attack or Decay, and so on. Arithmetic and algebra are possible, using letters to denote variables; these have an obvious use in compositions of a mathematical nature, but are also necessary for quite mundane tasks, such as transposition, repetition, calculating rests, etc. MUSYS also has structural facilities, allowing any sound or group of sounds to be facilities, allowing any sound or group of sounds to be given a name, and subsequently referred to by that name. This means that scores can use as much conventional terminology as the composer desires. For example, a crescendo can be called CRESC and a glissando called GLISS. There are six simultaneous systems, each with its own time control, so it is not necessary to work out time interrelationships for complex rhythms. There is a random number program which generates randomly distributed numbers in a repeatable sequence, which is useful in aleatoric composition. composition

The studio can be used immediately, without learning MUSYS, since the computer can be used simply as a large sequencer.

However, a unique advantage offered by the studio is programmed scoring, and for this it is necessary to learn MUSYS. The course given last year demonstrated that composers without previous knowledge of programming do not find MUSYS a difficult language to learn. EMS offer free courses in MUSYS hardware and software, each lasting one week, to purchasers of computer studios, and the computer manufacturers offer courses in computer programming.

#### What Software does EMS provide?

If you buy a computer studio from EMS, all the currently available EMS software and documentation is given to you as part of the studio. New programs is given to you as part of the studio. New programs and program enhancements will also be issued free for a minimum period of two years from the date when the studio is commissioned. In addition, as the owner of a PDP8 computer, you are entitled to all software issued by DEC, and to membership of DECUS, the DEC Users Society which issues information about new and improved sofeware. EMS will also act as a pool for programs written by owners of EMS computer studios, so you will receive software written by other insers and may contribute your own to the pool. Comusers, and may contribute your own to the pool. Com-plete pieces will also be pooled by EMS and made available to other users, copyright permitting.

If you do not buy an EMS computer studio, you may purchase programs and documentation from EMS at the listed prices. EMS will not maintain programs run on machines they did not provide.

#### **Specifications**

#### Synthi 100

The SYNTHI 100 is described fully in the first half of this brochure. It contains the voltage controlled analogue equipment of the studio: the outputs are at line level for feeding to amplifiers or tape recorders

The devices included are these:

- 6 sine/ramp oscillators
- 6 square/triangle oscillators
- 2 coloured noise generators
- 8 swept filters
- 3 envelope shapers 3 ring-modulators
- 1 random voltage generator
- 2 envelope followers
- 1 pitch-voltage converter
- 3 slew-limiters
- 8 input amplifiers
- 8 output amplifiers (8 line outputs, 4 panel outputs)
- 1 octave filter bank
- 6 track, 256 event Sequencer and Clock
- 2 keyboards

The SYNTHI 100 has a patching matrix for signals and a patching matrix for control voltages, each 60 x 60; when used with a computer, the 60 control inputs are provided by the Digital Interface.

#### The Digital Interface

The Digital Interface has 64 6-bit DACS and 4 6-bit ADCs; the digital to analogue converters enable the computer to control all of the SYNTHI 100 devices, of which 4 have high precision (10 bit) control, ADCs enable the analogue equipment to signal the computer, for cueing, etc. It also contains a clock, which can be controlled by a crystal oscillator, by an external oscillator, or manually. The external oscillator is usually a prerecorded pulse train on one track of a multitrack (ape, permitting exact synchronisation between tracks in a piece which is realised in several passes, and manual control is used in editing and

The Digital Interface has a number of illuminated pushbuttons which enable any device to be selected and controlled manually, and entire pieces can be constructed on it by using the Sequencer program and Keyboards. When using MUSYS, the manual controls are used for monitoring and making detailed changes, large scale changes being done in terms of the original score.

#### The Computer

The basic configuration as provided by EMS enables scores of up to 6,000 characters to be compiled, and the composition may have up to 20,000 events. Longer compositions can be realised in several sections. The clock has a maximum rate of 400 Hz, giving time dis-crimination to 2.5 msec which is adequate for musical applications. The basic software comprises a Text Editor, for the preparation and correction of the MUSYS scores, a Compiler, which translates the scores into lists of numbers to the devices under the control of the clock. There is also a program called Sequencer, which simply uses the storage units of the computer to store data provided by the SYNTHI 100 (from the keyboards, for example).

# Bibliografía

- The 200 best free vst plugins ever [página web]. (s.f.). https://blog.landr.com/best-free-vst-plugins/.
- Arturia synthi v [página web]. (s.f.). http://www.futuremusic-es.com/arturia-synthi-v-sinte-plugin.
- Boulanger, R. (Ed.). (2000). The Csound book. perspectives in software synthesis, sound design, signal processing, and programming. Massachusetts: The MIT Press.
- Cycling '74 [página web]. (s.f.). https://cycling74.com/products/max/. Electronic muscic studios [página web]. (1998). http://emssynthesisers.co.uk/.
- Free ems vcs3 and synthi aks vst emulations [página web]. (s.f.). https://blog.wavosaur.com/free-ems-vcs3-synthi-aks-vst-emulations/.
- Fuzzy gab 4 [página web]. (2020). http://fuzzygab.uclm.es/.
- Gnu general public license [página web]. (2016, noviembre). https://www.gnu.org/licenses/gpl-3.0.html.
- Grogono, P. (1972). The synthi educational handbook [folleto]. Electronic Music Studios of American Inc.
- Lavry, D. (2004). Sampling theory for digital audio [artículo en PDF]. Descargado el 10 de mayo de 2020, de http://lavryengineering.com/pdfs/lavry-sampling-theory.pdf
- Lieb, R.-D., y Kaiser, U. (2019). A guide to modular worlds (1ª ed.). Rellingen: SynMag-Verlag.

Bibliografía Bibliografía

Molina, S., Osona, J., Sanz Vázquez, J., del Saz, D., y Alcázar, A. J. (2018). Las colecciones del gme: presente, pasado y futuro (1ª ed.). Cuenca: Ed. de la Universidad de Castilla—La Mancha.

- Open sound control [página web]. (s.f.). opensoundcontrol.org.
- Pure data [página web]. (s.f.). https://puredata.info/ ..
- Pérez-Arroyo, R. (1984). El sintetizador: el instrumento de la era visual. Ritmo, 542, 12-16.
- Roads, C., Strawn, J., Abbott, C., Gordon, J., y Philip, G. (1996). *The computer music tutorial*. Cambride: The MIT Press.
- Supercollider [página web]. (s.f.). https://supercollider.github.io/.
- Supper, M. (2012). Música electrónica y música con ordenador. Historia, estética, métodos, sistemas. Madrid: Alianza Música.
- The synthi 100 [folleto]. (1971).
- Synthi 100 music synthesizer (Inf. Téc.). (2016). Descargado el 25 de abril de 2020, de https://portal.engineersaustralia.org.au/system/files/engineering-heritage-australia/nomination-title/Synthi%20100.Nomination.V7.27%20May%202016\_0.pdf
- Synthi users manual [folleto]. (s.f.). Electronic Music Studios of American Inc. Descargado de https://whitefiles.org/rwz/zxe/1970\_ems\_synthi\_users\_manual.pdf
- Synthi users manual EMS. (1970). Londres: Electronic Music Studios. Descargado el 25 de abril de 2020, de https://whitefiles.org/rwz/zxe/1970\_ems\_synthi\_users\_manual.pdf
- Touchosc [página web]. (s.f.). https://hexler.net/products/touchosc.
- Vail, M. (2014). The synthesizer. A comprehensive guide to understanding, programming, playing, and recording the ultimate electronic music instrument. New York: Oxford University Press.
- Wilson, S., Cottle, D., y Collins, N. (Eds.). (2011). *The SuperCollider book*. Massachusetts: The MIT Press.