



UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
Escola d'Enginyeria de Barcelona Est

TREBALL FI DE GRAU

Grau en Enginyeria Elèctrica

**CREACIÓ D'UN SINTETITZADOR DIGITAL FENT ÚS DE
LABVIEW**



Memòria i Annexos

Autor: Roger Serra Gómez
Director: Francesc Xavier Roset Juan
Convocatòria: Juny 2020

Resum

En aquest TFG es descriurà de forma concisa el necessari per a poder dissenyar un sintetitzador digital en Matlab, des de la part teòrica fins a la part tècnica. No cal saber cap concepte musical per entendre el treball, tot i que es recomana, ja que hi ha apartats d'un nivell avançat. El resultat final és l'obtenció d'aquest sintetitzador de manera exitosa i innovadora.

El més interessant és que planteja una nova visió a l'indústria de la producció digital. Fins als últims anys, en comptes d'inventar diferents mòduls, ja siguin equalitzadors, compressors, etc, de forma digital el que es feia era intentar emular el component analògic al digital. La producció musical esta evolucionant cap a un entorn més gràfic, sense oblidar el so i precisament això és el que busca el projecte.

Resumen

En este PFG se describirá de forma concisa lo necesario para poder diseñar un sintetizador digital en Matlab, desde la parte teórica hasta la parte técnica. No hace falta saber ningún concepto musical para entender el proyecto, aunque se recomienda, ya que hay apartados de un nivel avanzado. El resultado final es la obtención de este sintetizador de manera exitosa e innovadora.

Lo más interesante es que plantea una nueva visión a la industria de la producción digital. Hasta los últimos años, en vez de inventar diferentes módulos, ya sean ecualizadores, compresores, etc, de forma digital lo que se hacía era intentar emular el componente analógico al digital. La producción musical está evolucionando hacia un entorno más gráfico, sin olvidar el sonido y precisamente eso es lo que busca este proyecto.

Abstract

The project describes in a concise way what is necessary to create a digital synthesizer in Matlab, from the theoretical part to the technical part. It is not necessary to know any musical concept to understand the project, although it is recommended, since there are sections of an advanced level. The principal objective is obtaining the synthesizer in a successful and innovative way.

The most interesting thing is that it presents a new vision to the digital production industry. Until recent years, instead of inventing different modules, like the equalizers, compressors, etc, what was done is trying to emulate the analog component to the digital one. Music production is evolving towards a more graphic environment, without forgetting, of course, the sound and precisely that is what this project is looking for.



*A la meva família per recolzar-me
en totes les decisions que he pres
i per creure en mi com a músic i enginyer.
Al Juan, mestre de taller,
per proporcionar-me informació essencial,
la qual ha fet possible aquest projecte.*





Glossari

DAW – Digital Audio Workstation

MIDI – Musical Instrument Digital Interface

FPGA – Field Programmable Gate Array

DSP – Digital Signal Processor

FFT – Fast Fourier Transform

ADSR – Attack, Decay, Sustain and Release

VCO – Voltage Controlled Oscillator

LPF – Low Pass Filter

HPF – High Pass Filter

VCA – Voltage Controlled Amplifier

LFO – Low Frequency Oscillator

CV – Controlled Voltage



Índex

RESUM	I
RESUMEN	II
ABSTRACT	III
GLOSSARI	VII
1. PREFACI	1
1.1. Origen del Treball.....	1
1.2. Motivació	1
2. INTRODUCCIÓ	3
2.1. Objectius del Treball	3
2.2. Abast del Treball	4
2.3. Estructura del Projecte	4
3. TEORIA DEL SO	7
3.1. L'espectre del So	8
3.2. Qualitats del So	11
3.2.1. El Timbre	11
3.2.2. El To	12
3.2.3. L' Intensitat	13
3.2.4. La Duració	13
3.2.5. La Ressonància.....	14
3.2.6. Eco i Reverberació	14
4. COM ES SINTETITZA UN SO?	15
5. SISTEMES DE SÍNTESIS MODULARS	23
5.1. CV	23
5.2. VCO.....	24
5.2.1. Senyal Sinusoidal	25
5.2.2. Senyal Quadrat	26
5.2.3. Senyal Triangular	29
5.2.4. Senyal Dent de Serra	31
5.3. Generador de Soroll.....	33
5.4. ADSR	34

5.5. VCA.....	35
6. DISSENY DEL SINTETITZADOR _____	37
6.1. Primer Plantejament.....	37
6.2. Plantejament del Algoritme	39
6.3. Disseny Gràfic del Sintetitzador.....	40
6.4. Desenvolupament de Algoritme	41
6.5. Resultat Final i Explicació de la Interfície	43
7. MIRADA AL FUTUR _____	45
8. ANÀLISI D'IMPACTE AMBIENTAL _____	47
CONCLUSIONS _____	49
ANÀLISI ECONÒMICA _____	51
BIBLIOGRAFIA _____	53
ANNEX A _____	55
A1. Codi “Matlab” del sintetitzador.....	55

1. Prefaci

De forma breu s'explicarà en aquest primer capítol la situació tant laboral, acadèmica i personal que m'han dut a fer aquest projecte.

1.1. Origen del Treball

Des de ben petit que la música produïa un efecte en mi especial, abans que tingues memòria, els meus pares diuen que ja cantava melodies, jo, tot i que m'hi esforço no soc capaç de recorda-ho. Fins que va haver un dia que em vaig decidir que era hora d'aprendre a tocar un instrument, això va ser als 10 anys d'edat. Des de llavors que la música només ha fet que créixer al meu voltant fins a tal punt que amb l'edat de 16 anys em vaig atrevir a aprendre instruments de forma autodidacta, actualment amb quasi 23 anys toco 8 instruments diferents, 4 dels quals els he tocat públicament en concert, veu, guitarra, baix i piano.

Tanmateix als 17 anys aproximadament em vaig interessar molt en la producció musical i les possibilitats que obria aquest món així doncs des de llavors que també m'ha acompanyat en el procés de creixement com a músic.

Arrel d'això he tingut experiència treballant amb tot tipus d'elements que intervenen en la producció, des de la producció fins al "mastering" i la post-producció. Familiaritzant-m'hi amb tot tipus d'efectes destinats a millorar una cançó o una mescla com és més conegut en el món de la música.

1.2. Motivació

Donada la meua experiència anterior en producció he treballat amb molts sintetitzadors, de varies marques, alguns més complexos, altres més simples, com per exemple els sintetitzadors de "Fabfilter", o "Kilohearts" entre molts altres, encara que si hagués d'escollir un sens dubte el "Massive X" de "Native Instruments" de fet és el meu sintetitzador preferit.

En tots ells veia el mateix error, sobretot al començament de tot que no existeix el suficient coneixement per tocar els paràmetres que aquests programes presenten i acabes que no saps ben bé el que estàs fent perdut dins la interfície.

L'empresa "Fabfilter" coneguda sobretot per el seu equalitzador revolucionari on va decidir apostar per un model gràfic d'equalitzador quan en aquell moment, els equalitzadors emulaven un producte analògic.

Vaig pensar que seria bona idea fer exactament el mateix però amb un sintetitzador per, sobretot, facilitar el procés d'aprenentatge i proporcionar als més inexperts una eina senzilla per començar a treballar.

2. Introducció

La tecnologia musical ha evolucionat ràpidament en l'últim segle, fet que ha permès entrar així en una nova etapa en la música on l'electrònica juga un paper fonamental. En un primer moment es van crear estudis de so, on només els privilegiats podien comprar material analògic per a poder fer gravacions, ja que el seu elevat cost no estava a l'abast de tothom. Mica en mica la tecnologia ha anat evolucionant i en l'actualitat fins i tot aquells amb un pressupost baix es poden permetre, el que abans era un luxe, gravar-se a casa. Fet que provoca, òbviament, una major accessibilitat per a les persones.

Aquells grans aparells analògics dels estudis de so, ara resulta que són digitals i que a més a més els pots executar amb un element petit com és un ordinador sense perdre qualitat de so. De fet aquest software que simula els aparells analògics ha anat evolucionant de tal forma que és un dels més utilitzats. Aquí s'introdueix el concepte de DAW (Digital Audio Workstation), que és la principal eina de desenvolupament musical actualment, el qual es desenvoluparà més endavant.

Gràcies a un DAW es poden fer servir elements com el seqüenciador el qual el ús més habitual és fer-lo servir per programar i/o reproduir "*samples*" gravats amb anterioritat de forma, com bé diu el nom, seqüencial. Tot això ho fa a través de fitxers d'àudio anomenats MIDI (Musical Instrument Digital Interface). Per a entendre millor el concepte MIDI s'ha de fer el símil a una partitura. El MIDI indica exactament el temps de duració d'una nota, la tonalitat d'aquesta nota i l'instrument que reproduirà. Més endavant s'entrarà a conèixer el concepte més a fons.

Per a que funcioni un sintetitzador digital, no es necessita dels conceptes anteriors, no obstant, dins d'un DAW i fent servir llenguatge MIDI és on es troba la majoria de vegades i es que és un element molt habitual avui dia.

L'objectiu es desenvolupar un sintetitzador fent servir el programa "Matlab" el qual és un dels més utilitzats al llarg del grau universitari, i té un llenguatge molt intuïtiu. Aquest sintetitzador posteriorment ha de treballar dins d'un DAW a temps real i com a MIDI. Principalment es farà servir el disseny d'aplicacions incorporat amb "Matlab".

2.1. Objectius del Treball

Seguidament es presenten els objectius principals a desenvolupar, i les seves modificacions degut a factors externs de força major.

Objectius inicials:

- Desenvolupar un sintetitzador en format digital fent servir el programa Labview.
- Treballar amb el sintetitzador a temps real fent servir un teclat MIDI o teclat d'escriure.
- Donar una nova visió al món de la producció musical introduint gràfiques.

A causa d'un factor econòmic aquest objectius inicials s'han hagut de substituir, ja que per a fer funcionar un sintetitzador a temps real, a través d'un teclat MIDI a Labview, fa falta el component FPGA i DSP creats per la pròpia empresa en format hardware, els quals tenen un preu elevat.

Per tant, s'ha canviat el plantejament inicial i els objectius finals són:

- Desenvolupar un sintetitzador en format digital fent servir el programa Matlab.
- Treballar amb el sintetitzador a temps real fent servir un teclat MIDI o teclat d'escriure.
- Donar una nova visió al món de la producció musical introduint gràfiques.

El programa Matlab ofereix la possibilitat de treballar amb FPGA i DSP amb la seva llicència en format software per tant, ho fa molt més accessible.

2.2. Abast del Treball

Aquest Treball de Fi de Grau pretén donar un suport al músic que s'està iniciant en el món de la producció musical. Un punt de partida, una eina senzilla i fàcil de fer servir i sobretot d'entendre, ja que en els primers passos pot arribar a ser un món confús.

Aquest projecte, ara per ara, no pretén competir contra les grans marques de software, principalment per la falta, segurament, de coneixement i equip. Però si donar una nova visió de com poder treballar de cara un futur en la producció musical. Reitero que l'abast és ajudar al productor principiant, aquells usuaris que tenen un petit estudi de so personal a casa i tenen ganes de produir música. Tanmateix aquest projecte seguirà creixent posterior a la finalització del TFG, això, tan sols és una primera versió, una fase beta.

2.3. Estructura del Projecte

Seguint els objectius marcats al apartat anterior el projecte començarà en la investigació de funcionament d'un sintetitzador analògic. Important començar pel analògic d'aquesta manera es diferenciarà d'una manera molt més clara les parts que componen el sintetitzador i els seu ús. Des del concepte més bàsic com per exemple, que és el so o com és genera el so, entre d'altres propietats d'aquest. Un cop estigui marcat s'hagi descrit d'una forma clara i concisa, aquest fenomen s'entrarà a veure que és un sintetitzador i les seves infinites possibilitats.

Posteriorment es desenvoluparan conceptes també importants per al sintetitzador, però que d'alguna manera no l'afecte directament. Conceptes com ara, que és un DAW i les seves característiques, que és i com funciona el llenguatge MIDI i la seva importància, entre d'altres conceptes altament necessaris per a un correcte ús d'aquest sintetitzador.

Un cop tota la teoria estigui explicada, es procedirà a explicar la interfície del sintetitzador creat a Matlab i el seu funcionament.

S'hi afegeix un últim apartat on s'explicarà quins plans de futur hi ha previst per aquest projecte.

3. Teoria del So

Segons el “Diccionari de la llengua catalana” de l’Institut d’Estudis Catalans el so *“és la impressió produïda en l’òrgan de l’oïda per les vibracions elàstiques d’un cos que es propaguen en tots els medis materials en forma d’ones. Un so agut o alt. Un so greu o baix. L’altura o to d’un so. La intensitat d’un so. El timbre d’un so.”* . D’aquesta definició es poden treure diverses conclusions de com és el so. La primera és que el so és una impressió produïda per la nostre oïda per les vibracions elàstiques, per tant, aquest so podria ser per exemple una ona ondulatoria, com ara una ona sinusoidal per exemple.

Aquesta primera definició ens proporciona altres conclusions sobre la nostre oïda. Com que és una impressió de la oïda cada persona respondrà de forma diferent a aquest estímul. Algunes persones escoltaran aquesta mateixa ona de forma diferent en diferents intensitats. Igual que amb les persones no hi ha tampoc dues orelles exactament igual. Aquest fet no te res a veure amb l’oïda absoluta. L’oïda absoluta esta relacionada amb la memòria auditiva, es a dir capacitat de recordar sons. Una persona amb oïda absoluta es capaç de reconèixer i reproduir estímuls auditius sense haver de escoltar una nota de referencia.

El fet de que cada persona escolti de diferent manera un mateix so te a veure amb l’espectre audible. L’esser humà per definició es capaç d’escoltar freqüències des de 20 Hz a 20 kHz. Cadascú per això, escolta aquestes freqüències a diferents intensitats, es podria dir d’alguna forma que cada orella esta equalitzada de forma diferent. Imaginem que tenim una ona sinusoidal amb una freqüència d’ 1 kHz, dues persones diferents amb la mateixa ona ho poden escoltar més alt o més baix respecte a altres freqüències, per exemple la persona 1 escolta les freqüències agudes (2 kHz cap endavant) més fortes que la persona 2, i en canvi la persona 2 escolta les freqüències mitges (256 Hz a 2 kHz) més fortes que la persona 1. Partint d’aquesta base ara imaginem que les dues persones els hi arriba el so d’un violí el qual esta tocant un La4, el La4 te una freqüència de 440 Hz afinat de forma estàndard, les dues persones escoltaran el so de forma desigual, depenent de l’espectre auditiu que tingui. Això pot sonar confús, i es que sempre s’ha estat parlant d’una ona, però en el cas d’un violí s’ha parlat d’un so. El so, en aquest cas, d’un violí tocant un La 4 esta compostat per una ona a 440 Hz però també existeixen altres ones dins d’aquesta principal degut als harmònics del instrument. Els harmònics no son més que ones complementaries a la ona principal a diferents freqüències amb diferents característiques per a cada instrument. És per aquest motiu que, i tornant al exemple de les dues persones, aquestes dues persones escolten el mateix so de forma diferent.

Acabant amb la definició de so anomena diverses qualitats per a poder definir un so com ara la intensitat, el timbre, el to i un últim que no anomena però l’afegeixo per la importància que te en la música i és la duració.

3.1. L'espectre del So

Com ja s'ha anomenat al primer apartat d'aquest capítol trobem que pensar en el concepte de so com a una simple ona és un error, ja que un únic so pot tenir múltiples ones, i en la majoria d'ocasions és així. En canvi si pensem que el so és la suma de múltiples ones a múltiples freqüències resultarà que d'aquesta manera es pot sintetitzar pràcticament qualsevol so.

Ajuda pensar en el so com si fos un raig de llum. Quan aquest raig de llum entra dins d'un prisma, aquesta es descompon en múltiples colors que componen aquest raig de llum. Cada color, és d'alguna manera el color pur, o en el cas objecte d'estudi, és una ona simple amb certa freqüència amplitud i fase. La transformada de Fourier en el cas d'estudi és el que actua com a aquest prisma. És la transformada de Fourier qui divideix aquesta ona de so, li podríem dir ona de so complexa, en múltiples més simples. Aquesta característica de la transformada de Fourier s'aconsegueix realitzant una Fast Fourier Transform, habitualment se la coneix com FFT.

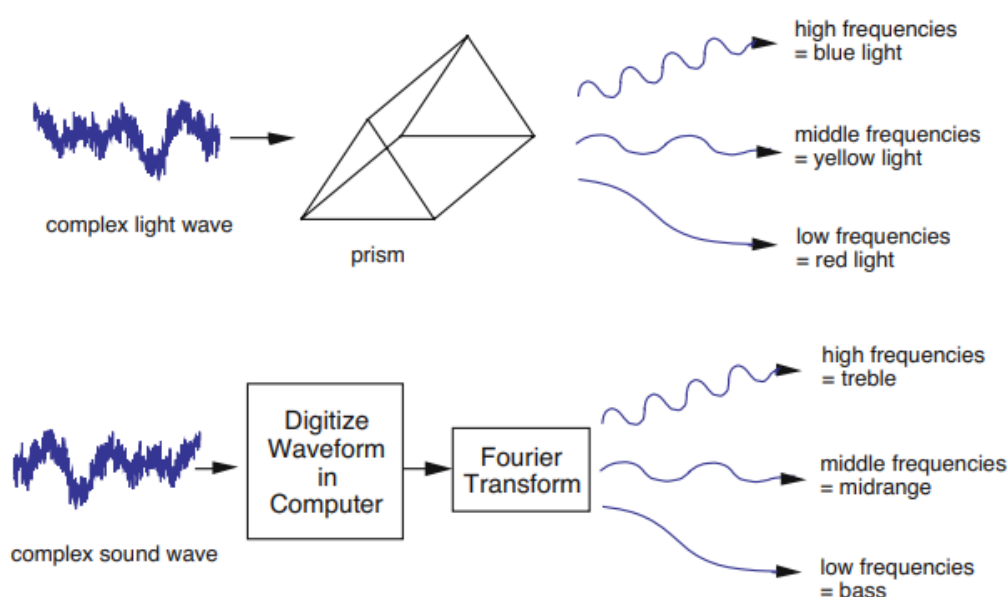


Figura 3.1. Similitud entre llum – prisma i Ona complexa – Transformada de Fourier (Font: Tuning, Timbre, Spectrum, Scale)

Procedim doncs a explicar un cas real. Molts dels instruments tenen ones molt més complexes que la mostrada anteriorment, per exemple una trompeta. Si agafem el so d'una trompeta i li fem un FFT s'obté el següent resultat.

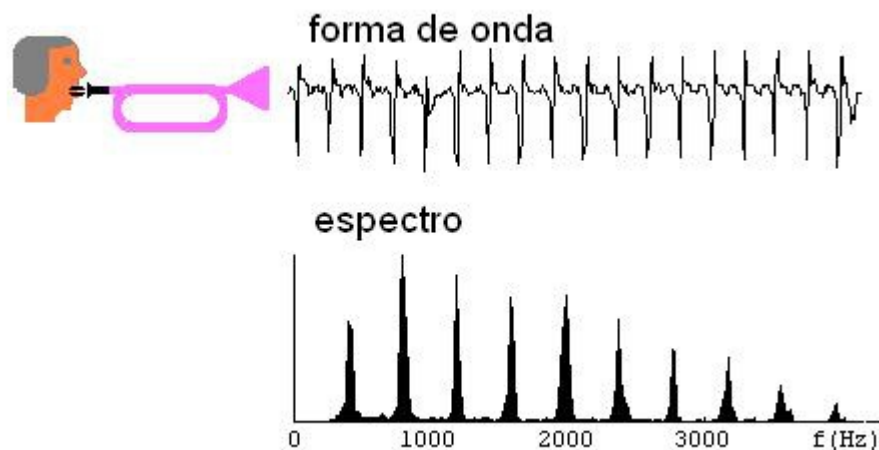


Figura 3.2. Espectre de so d'una trompeta (Font: Estructura armónica del sonido, Hispasonic)

Com s'observa a la figura 3.2 l'espectre de la trompeta és curiós i es que els harmònics fins arribar a 3 kHz pràcticament tenen una amplitud superior que la ona principal. És per aquest motiu que una trompeta sempre sona aguda, perquè els seus harmònics es troben a freqüències mitges i agudes.

Pot resultar difícil entendre com es pot generar so a través d'una trompeta quan existeix una excitació de vent al principi del tub. Per tant s'agafarà una guitarra com a exemple, que és molt més fàcil d'entendre, gracies a que una corda és un element molt més visual en comparació al vent. De totes formes segueixen el mateix principi, si que és cert que existeixen algunes diferències, principalment perquè la forma de generar aquesta ona és diferent, però el principi de funcionament és el mateix.

Bé, tal com s'ha demostrat anteriorment un so esta compostat per una ona fonamental i els seus harmònics. El cas d'una corda és molt clar perquè és un exemple molt visual. Imaginem ara que tenim una corda lligada entre dos punts i la fem vibrar a certa freqüència, la qual la corda vibri sense mostrar cap node i sense mostrar cap tipus d'irregularitat. En aquest punt, s'haurà trobat la freqüència fonamental de la corda. Si es volgués trobar el primer harmònic d'aquesta corda s'hauria doncs de multiplicar la freqüència fonamental per 2, obtenint així el que s'anomenarà com a $2f$ (2 vegades la freqüència fonamental). Si es volgués obtenir la tercera freqüència fonamental, s'hauria de fer $3f$ (3 vegades la freqüència fonamental) i així successivament. A la figura 3.3 es mostra clarament l'exemple, inclòs el FFT.

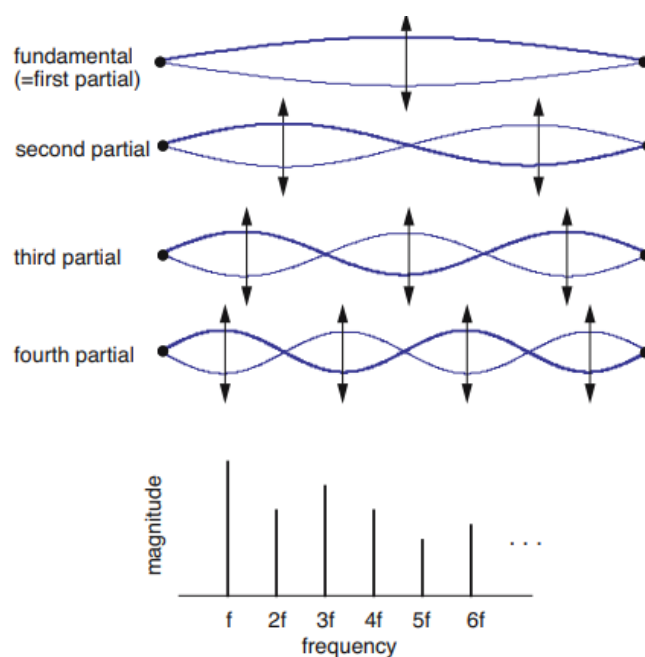


Figura 3.3. Cas ideal corda en vibració (Font: Tuning, Timbre, Spectrum, Scale)

Tot això és en un cas ideal. En el cas real intervien altres factors, intervien les qualitats del so, que s'anomenaven a la introducció del projecte. Aquestes són la duració, la intensitat, el timbre i el to. Totes aquestes característiques són mostrades dins del FFT, per això la importància d'aquest.

L'exemple doncs d'una guitarra que està tocant la corda Sol es mostra a la figura 3.4.

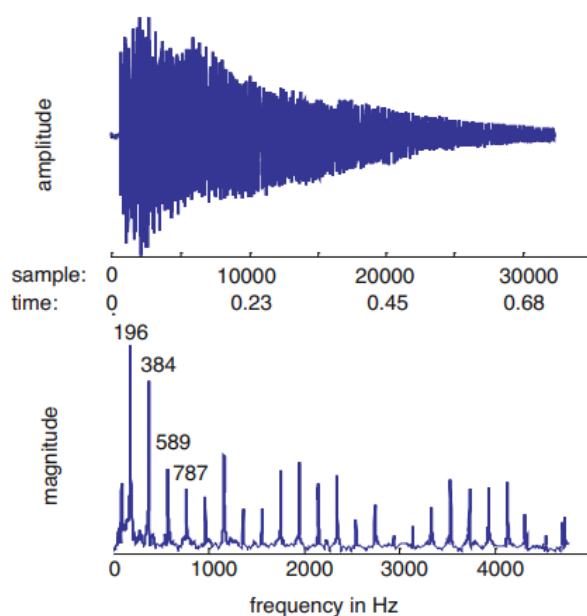


Figura 3.4. Cas real corda Sol (Font: Tuning, Timbre, Spectrum, Scale)

En el cas real d'una corda es poden trobar múltiples harmònics i no és tant perfecte com l'exemple ideal mostrat a la figura 3.3. A diferència de la trompeta l'ona fonamental amb els altres harmònics té una intensitat superior. Una altra diferència a destacar del cas real és que aquest té intensitat pràcticament a tots els harmònics amb molta poca intensitat en algun d'ells, però hi és.

La freqüència que es veu abans del 196 Hz és soroll, es molt a prop dels 60 Hz i això fa pensar que aquest experiment s'ha realitzat als Estats Units, ja que el corrent de línia va a aquesta freqüència allà.

3.2. Qualitats del So

Tal com s'ha definit anteriorment, el so es pot definir segons la seva intensitat, la seva duració, el seu timbre i els seu to. Aquests quatre paràmetres són els que defineixen un determinat so per definició. A banda d'aquests quatre paràmetres hi ha altres a tenir en compte com la ressonància, l'eco i la reverberació. Tot i que aquests tres últims no s'obtenen de la font directament, sinó que és un efecte del entorn.

3.2.1. El Timbre

Per a molta gent la definició més confusa de totes les llistades al paràgraf anterior, i és que, és habitual haver escoltat la seva definició de forma errònia. Es podria plantejar, si un arbre cau enmig del bosc, hi ha algun tipus timbre? Aquesta pregunta, que aparentment sembla simple, es troba dins de l'obra del filòsof George Berkeley "A Treatise Concerning the Principles of Human Knowledge" del 1710. Recentment l'American National Standard Institute ha respost que "no". Defineixen,

"El timbre és un atribut de sensació auditiva, el qual l'oient pot jutjar dos sons presentats de manera similar i tenint la mateixa intensitat i to com a diferència".

Aquesta definició és confusa, perquè no defineix el que és el timbre, més aviat, el contrari, defineix el que no és el timbre. A més a més, si el so no té to (per exemple el so d'un element de percussió o el "crack" d'un arbre caient) no pot ser "presentat de manera similar i tenint el mateix to". Per tant al veure aquesta definició, Pratt i Doak van suggerir una altra.

"El timbre és aquell atribut de sensació auditiva on l'oient pot jutjar que dos sons són diferents fent servir tan sols els criteris de tonalitat, sonoritat i durada."

Amb aquesta definició, ara, l'arbre té timbre quan cau, encara que tampoc s'ha solucionat res, la definició de timbre segueix sense estar clara.

Com ja s'ha demostrat definir el timbre no és del tot senzill. Afortunadament l'Arthur Jacobs amb la seva obra *A New Dictionary of Music* defineix que,

“En anàlisi, la diferencia entre els colors dels tons dels instruments esta en la diferencia dels harmònics representats en el so.”

Finalment s'obté una definició clara de que és el timbre. Com a totes les definicions també es pot treure una conclusió del contrari. Aquesta definició ens confirma que un so no tindrà timbre quan aquest no tingui harmònics. A més a més, aquesta definició uneix el concepte de timbre amb l'espectre del so a través dels harmònics. Tanmateix anomena al timbre com a color tonal. Per tant, tal com s'ha esmenat al apartat anterior, el timbre son els harmònics de una ona fonamental, és per aquest motiu que es distingeix entre dos instruments diferents tocant la mateixa freqüència.

3.2.2. El To

Breument, es defineix com aquella característica per la que un so sembla més agut o més greu que un altre. El to ve donat per el número de vibracions que es produeixen en un determinat període de temps, el que s'anomena freqüència, quan més alta sigui la freqüència més agut serà el to. El to s'expressa en Hertz (Hz).

Fins ara s'ha anat anomenant certes freqüències que generen ones, gracies a aquesta definició es troba el sentit a aquestes freqüències. En el cas de les notes musicals no totes les freqüències són valides tan sols les escollides en el sistema d'afinació internacional.

Una dada molt curiosa és que el sistema d'afinació ha anat variant al llarg dels segles. Aquest fet és molt important perquè significa que, per exemple, “La cinquena simfonia” de Beethoven, l'original no sona tal i com la coneixem sinó que lleugerament diferent.

Al 1936 una conferencia internacional va recomanar que el que s'anomena La4 fos afinat a 440 Hz i a partir d'aquesta freqüència generar la resta. No és, però, fins el 1955 que l'Organització Internacional d'Estandardització ho va acceptar.

Ara per ara en la majoria dels casos es fa servir el La4 440 Hz com a freqüència base, però s'estan començant a veure en orquestres de càmera i simfòniques una afinació a 442 Hz, que és una mica més brillant. No només s'altera l'afinació en orquestres, en cançons pop com ara “The Scientist – Coldplay” també es troba una lleugera modificació de l'afinació de referencia, en aquesta cançó en particular és de 437 Hz.

3.2.3. L' Intensitat

Quan es parla d'intensitat en la música, possiblement el primer que fa pensar és en la dinàmica, i es que així com en el cas del timbre, la majoria de població té el concepte d'intensitat erroni. La dinàmica és la variació d'enter un so més fort a un so més dèbil, o un so fort a un encara més fort. La intensitat és un paràmetre més complexa, i es que depèn de 4 factors que explicarem a continuació. En primer lloc la intensitat és la variació del so durant la seva emissió.

Es treballarà els conceptes d'aquests 4 factors en anglès, ja que és l'idioma estàndard a l'indústria de la música i son 4 conceptes molt treballats. A banda d'aquests 4 conceptes hi haurà altres que també es treballaran en anglès.

Bé, els 4 factors son l' "Attack", el "Decay", el "Sustain" i el "Release" o com se'ls coneix a l'indústria ADSR. Per exemple, un piano té un atac ràpid, després decau lentament o ràpidament depenent de si el pedal de "sustain" és premut o no.

Aquest paràmetre és el més comú en un sintetitzador, és principalment per on es modificaran les ones i serà els paràmetres que faran que el sintetitzador soni com una trompeta o soni com un violí. Més endavant s'entrarà al detall.

3.2.4. La Duració

La duració és el temps que una ona comença generar-se fins que para de fer-ho. Tal i com s'ha explicat en l'apartat de la intensitat en la duració intervé el concepte ADSR molt important per als sintetitzadors. Per la duració no només es important per els sintetitzadors, sinó que en les partitures la duració de les notes esta marcada en tot moment per el llenguatge musical. Per exemple, una nota negra, o una corxera.

Aquesta duració de la nota moltes vegades es pot veure allargada més enllà de quan es crea l'ona i quan s'acaba gràcies al fenomen de eco o reverberació que veurem amb més detall als següents apartats.

Això significa que si una nota té molta reverberació, i per tant sona molta estona, augmenta la duració d'aquesta? La resposta és "no". La duració es limita simplement on comença a generar-se ona i on s'acaba, la reverberació o eco són factors externs a la ona.

3.2.5. La Ressonància

La ressonància és un efecte pel que podem fer que un so es vegi reforçat en unes determinades freqüències. Aquest efecte es pot notar, sobretot, en instruments musicals mecànics quan a determinades notes l'instrument augmenta la seva intensitat. Pot ser molt interessant a la hora de reforçar harmònics de la ona original.

La ressonància és molt útil per l'acabat dels sons, com ara si es busca un so de flauta, per exemple.

3.2.6. Eco i Reverberació

Aquests efectes son altament coneguts, sobretot el d'eco. Qui no ha cridat algun cop "Eco!" en un espai tancat amb parets llises? L'eco es produeix exactament en aquestes condicions, quan les ones sonores son reflectides contra una paret llisa, la persona escoltarà el so directe i el reflectit, que és el mateix retardat "x" temps. Aquest temps serà més curt o més llarg depenen de la distancia a la paret.

L'efecte eco es degut al principi explicat, les ones quan reboten a la primera paret després ho fan cap a una altra i després cap a una altra, fins que s'atenuï el so, per sota el llindar d'audició. El més curiós és que a aquest efecte que se li ha anomenat eco, realment és l'efecte de reverberació.

En la producció musical l'efecte de reverberació és aquell efecte que fa rebotar les ones sonores però el so es difumina al llarg del temps, si per exemple, aquesta ona fos una paraula, en la reverberació no s'entendria. D'aquesta manera s'aconsegueix que el so inicial sembli més llarg del normal.

L'efecte eco en canvi, en producció musical és la repetició d'un so, però que s'identifica perfectament. És a dir, en el cas de una paraula, al cap de "x" segons sonaria exactament aquella paraula com si fos l'original.

4. Com es Sintetitza un So?

Bé, fins ara s'ha mostrat conceptes de so en general, sense concretar instrument. És moment ara de veure concretament el sintetitzador, que és l'element objecte d'estudi d'aquest treball de fi de grau. En capítols anteriors ja s'ha mostrat el que és una ona sinusoidal. A la següent figura es mostra l'ona sinusoidal d'un diapasó, L'ona d'un diapasó és "pura i neta" perquè no te cap tipus d'harmònics, l'ona i prou.

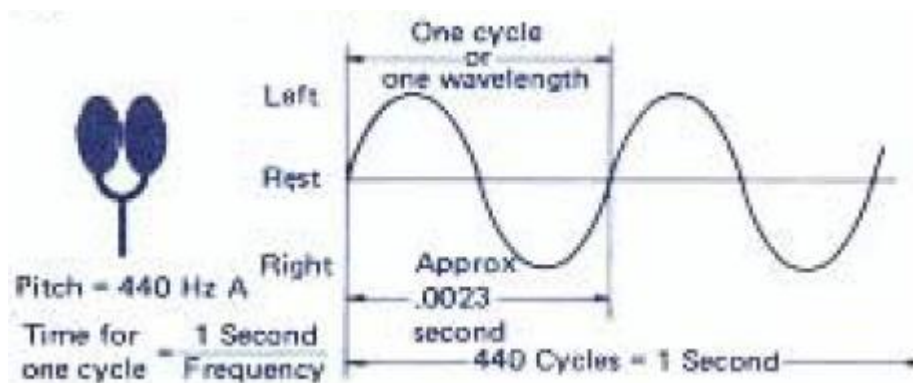


Figura 4.1. Ona sinusoidal d'un diapasó (Font: Abrana Studios)

A l'imatge es mostra també com la freqüència de l'ona és de 440 Hz, i correspon al La4 tal i com els estàndards marquen.

El matemàtic Fourier en el segle XVIII va determinar que qualsevol ona complexa podia ser descomposta en una suma infinita de ones sinusoidals de diferent amplitud i freqüència, denominant aquestes últimes com a harmònics. Aquest principi es coneix com a sèrie de Fourier, en aquest moment Fourier va establir el principi del sintetitzador sense voler-ho. Seguint aquest principi no hi ha cap forma d'ona complexa que no pugui ser sintetitzada a base d'ones sinusoidals simples de diferent magnitud. Tot el projecte es basa en aquest principi per això és tan important.

En aquest moment es troben dos escenaris diferents. El primer escenari és, partint d'una ona sinusoidal pura i simple afegir-li, sumant, altres ones sinusoidals de diferent freqüència i amplitud per així crear el so desitjat, una ona complexa.

D'altra banda es troba el cas contrari, es parteix d'una ona complexa, i aplicant diversos filtres, eliminar harmònics no desitjats fins aconseguir el so que interessa.

Aquests dos mètodes se'ls coneix com a síntesi per adició i síntesi per eliminació. En el disseny del sintetitzador s'ha escollit el mètode de adició.

En la següent figura es mostra com funciona una síntesi d'adició. En la primera ona (A), s'observa com a l'ona sinusoidal pura se li sumen dues ones sinusoidals harmòniques, amb la meitat i $1/3$ de l'amplitud original. El resultat d'aquestes fa recordar a una ona triangular tal i com es mostra al final de la figura.

En l'altre cas (B) es mostra una ona sinusoidal pura, i en aquesta se li sumen dues ones però aquesta vegada al harmònic 2 i al harmònic 4, amb una amplitud d' $1/3$ i d' $1/5$, en aquest cas es va formant una ona quadrada.

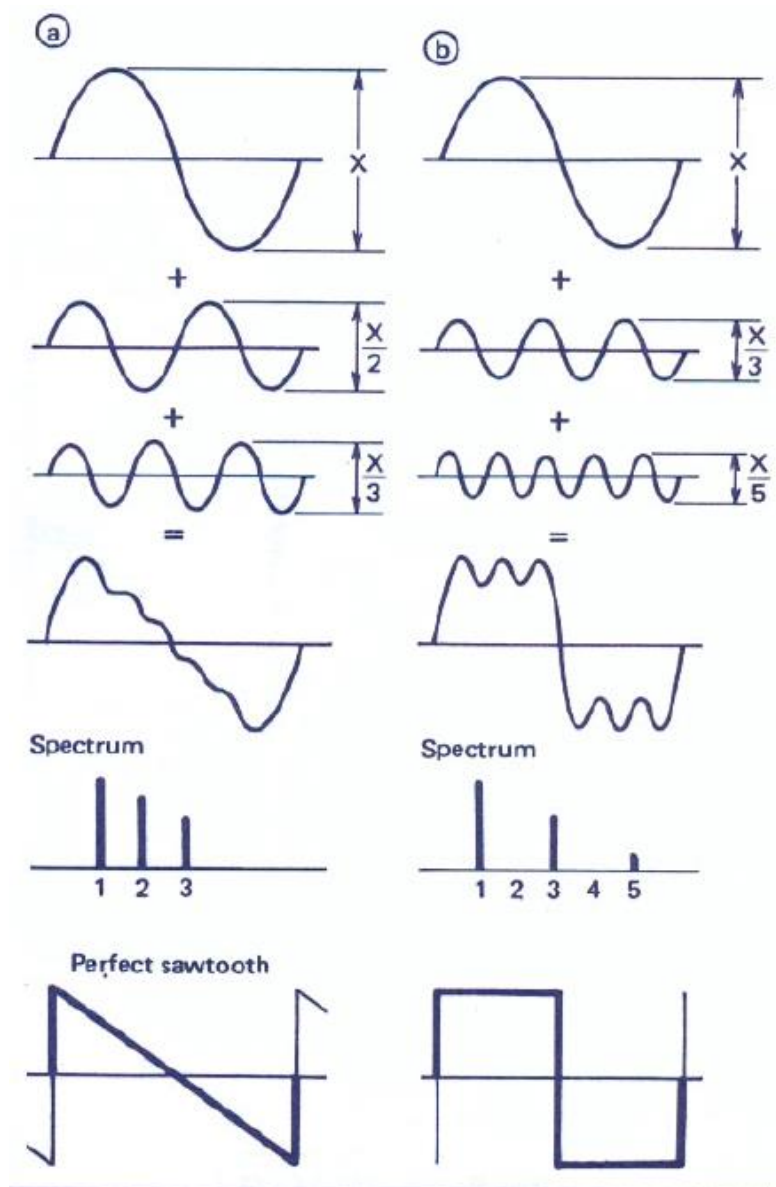


Figura 4.2. Creació d'una ona triangular i una ona quadrada (Font: Abrana Studios)

Quants més harmònics hi ha més perfecte seran les ones triangulars i quadrades. Realment aconseguir l'ona triangular i quadrada és pràcticament impossible, perquè faria falta un nombre infinit d'harmònics. Una molt bona aproximació es dona en els harmònics 5 i 7, per tant habitualment s'arriba fins aquests, ja que l'amplitud dels següents harmònics és tan petita que no es tenen en consideració, per a ser més acurats es tindrà en compte fins el 7é harmònic.

Amb l'anterior paràgraf en ment, ja es pot intuir com es llegeixen els gràfics d'harmònics, aquests són molt importants per saber com enfocar un determinat so, d'aquesta forma es mostren els harmònics més accentuats. Per exemple, si es vol treure el so d'una trompeta el més adequat és escollir una ona quadrada en comptes d'una ona sinusoidal. Un altra cas podria ser un instrument de corda, en aquest cas el més adequat seria una ona de dent de serra o triangular.

En canvi per a sons melòdics, suaus com ara el so d'una flauta l'ona més apropiada aquest cop si, és l'ona sinusoidal. Totes aquestes ones són generades per un VCO (Voltage Controlled Oscillator) aquest concepte s'explicarà en el proper capítol, però és important començar a introduir el concepte.

A continuació es mostra una ona triangular amb el seu espectre o gràfics harmònic.

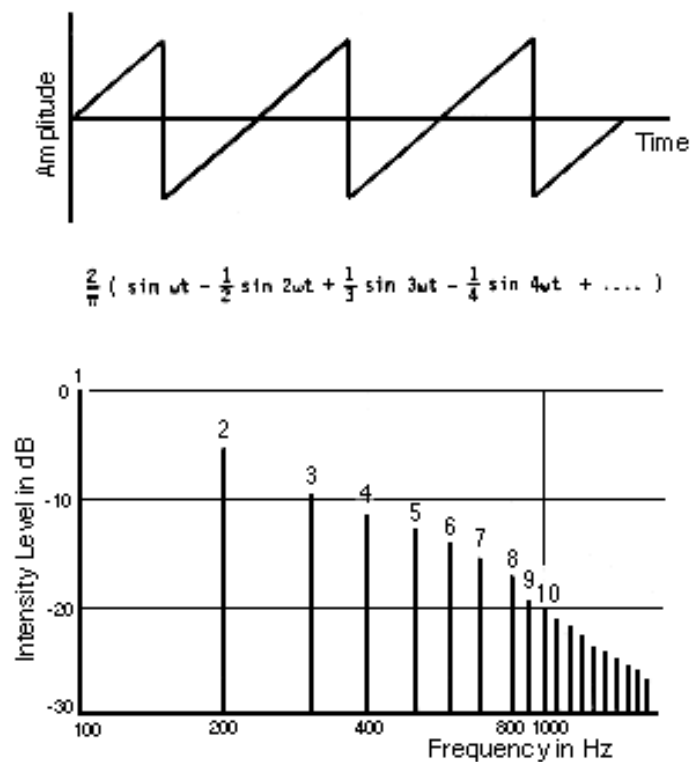


Figura 4.3. Espectre d'una ona triangular (Font: Simon Fraser University)

Tal i com s'ha explicat hi ha infinits harmònics, si s'observa la figura 4.3 l'harmònic 8, que correspon al numero 8 a la figura és l'últim per sobre de -20dB, a partir d'aquest habitualment es deprecien la resta d'harmònics tot i que el nombre d'ells és considerablement alt.

Seguint amb la pauta, a continuació es mostrarà, doncs, una ona quadrada i les seves característiques.

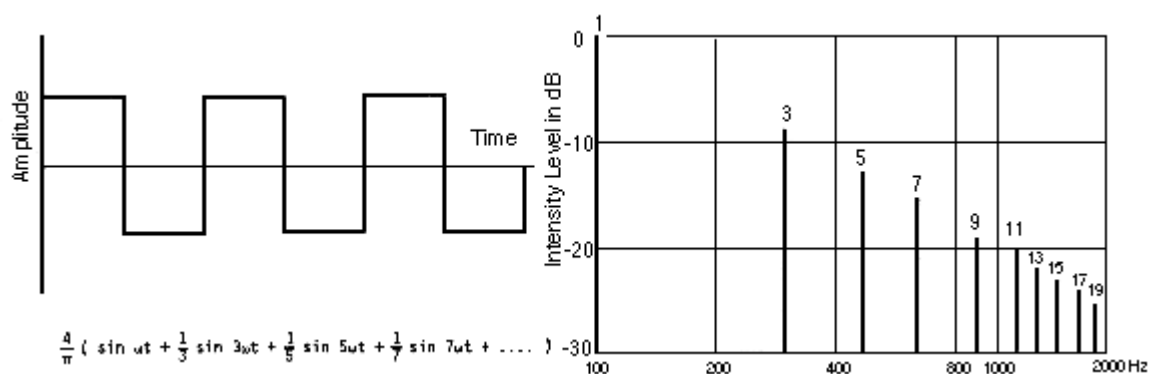


Figura 4.4. Espectre d'una ona quadrada (Font: Simon Fraser University)

En la figura 4.4 es mostra que l'ona quadrada compleix perfectament amb els paràmetres descrits anteriorment, veient l'eliminació dels harmònics parells. Tanmateix fins l'harmònic 7 tots ells tenen una amplitud força rellevant, a partir d'aquest cada vegada és inferior.

Llegir aquests gràfics és molt senzill, en els eixos de coordenades es troba la intensitat en dB en l'eix Y i la freqüència en l'eix X. Si ens fixem sempre s'interpreta a partir de l'harmònic 1 o harmònic fonamental que correspon al 0 dB. A partir d'aquest harmònic els altres tots es veuen atenuats en relació al fonamental. L'harmònic 3 correspon a una freqüència 3 vegades la fonamental i en ambdós casos es veu atenuat uns 10 dB. Al igual que amb l'harmònic número 5, la seva freqüència és 5 vegades la freqüència fonamental i es atenuat uns 13 dB.

En els sintetitzadors analògics el més habitual és fer servir la síntesis per eliminació vista en aquesta mateixa secció, els VCO generen ones enriquides en harmònics, en aquest moment s'apliquen filtres per limitar aquests harmònics fins aconseguir el so buscat.

Un bon exemple és la flauta.

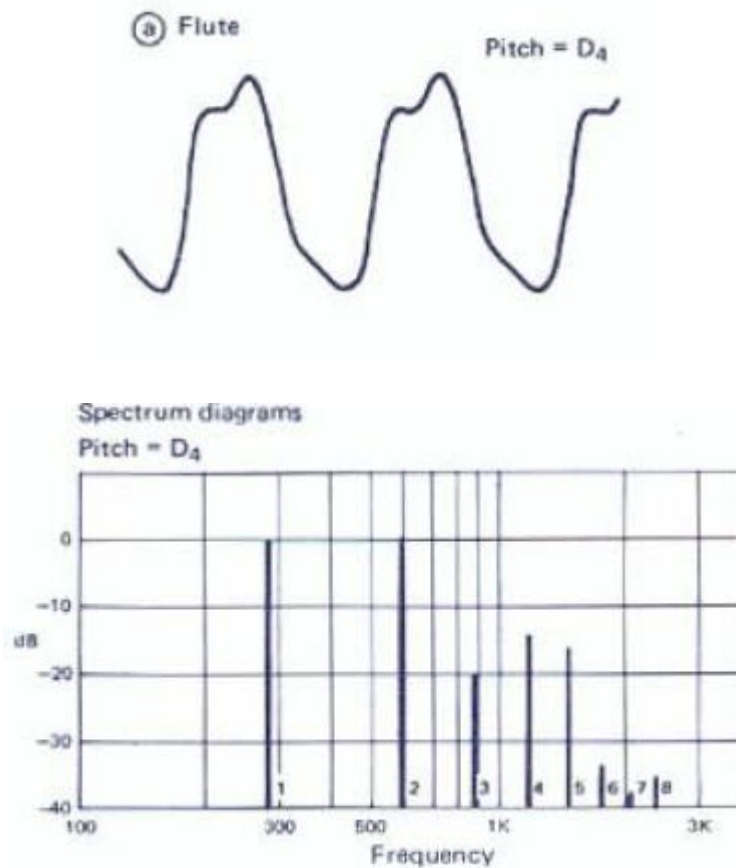


Figura 4.5. Ona i espectre d'una flauta (Font: Abrana Studios)

L'ona final recorda a una ona sinusoidal, per tant la pregunta ara és, com es sintetitza aquesta ona? Si s'observa l'espectre de l'ona es pot veure com te un harmònic, el número 2 que te pràcticament la mateixa amplitud que l'harmònic principal i posteriorment es troben 3 harmònics de més o menys la mateixa amplitud, la resta d'harmònic no son destacables, ja que tenen una amplitud molt baixa.

Primer farà falta que els VCO generin una ona que contingui aquests harmònics, a la figura 4.3 es comprova que l'ona dent de serra conté aquests harmònics.

Posteriorment amb els filtres s'han d'eliminar els harmònics a partir del 5, sense incloure aquest. Per poder realitzar-ho es fa servir un "Low-Pass Filter" (LPF), com ja s'ha comentat anteriorment els conceptes més utilitzats en l'industria es faran servir en angles. El LPF és possiblement el filtre més utilitzat en el món de la música sobretot en la producció cada instrument porta un LPF per a controlar harmònics. És un filtre que per sota del seu valor deixa passar totes les freqüències el cas contrari passa amb el valors per sobre el determinat, totes les freqüències per sobre el valor determinat seran eliminades. Existeix també el "High-Pass Filter" (HPF) és exactament el mateix però el funcionament contrari al LPF.

Amb el LPF es marca que a partir del 5é harmònic s'eliminin totes les freqüències superiors. Per últim si afegeix un filtre de ressonància per a poder arribar a generar l'ona que es desitja exactament, el filtre de ressonància fa que el harmònics guanyin importància.

Aquests son doncs tots els passos que s'han de seguir per a aconseguir sintetitzar un so. Hi ha molts elements que no són tant trivals però així és com funciona el mètode de síntesis per eliminació.

La conclusió que es pot treure d'aquí es que és important tenir un bon VCO, en el cas analògic, el cas digital aquest pas és més simple. Aquí es comença a veure doncs el cost d'un sintetitzador analògic, tots els elements que ha de portar com a mínim.

Malauradament això no és tot. Prèviament s'ha mencionat, també, el ADSR. Ara per ara es te un so que si es sosté sona com una flauta, però li falten tots els elements que fan que una flauta sigui una flauta.

D'aquest so de la flauta fa falta modelar el seu atac i la seva caiguda per a que soni com a tal. Aquest controls de ADSR s'anomena "Envelope". Un exemple mostrat podria ser el la guitarra. Per entendre millor el que és l' envelope a la següent figura es mostra com es genera aquest al llarg del temps en un guitarra.

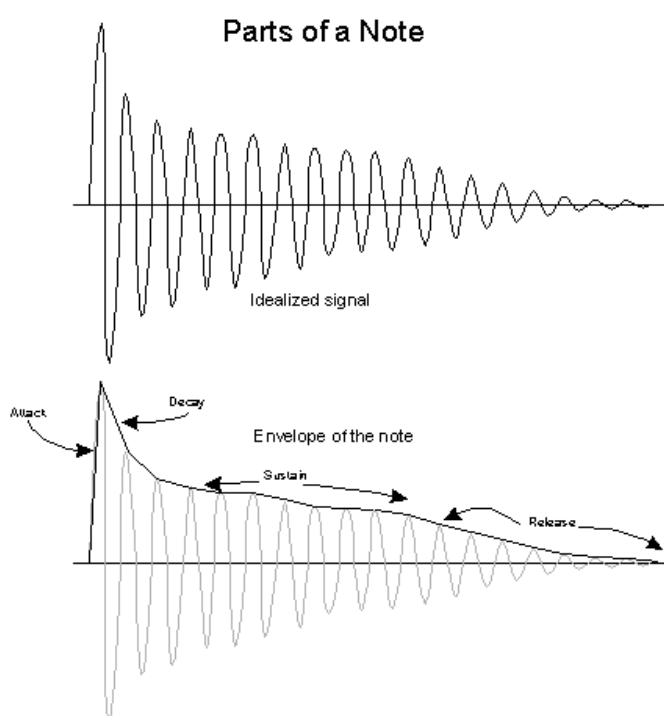


Figura 4.5. Envelope d'una guitarra (Font: Geo)

En la figura 4.5 es mostra com evoluciona la intensitat del so de la guitarra en el temps. Com es pot observar, fa un pic ràpidament i a partir d'aquí va decaient en el temps.

Cada instrument té el seu propi envelope, i cada un és diferent.

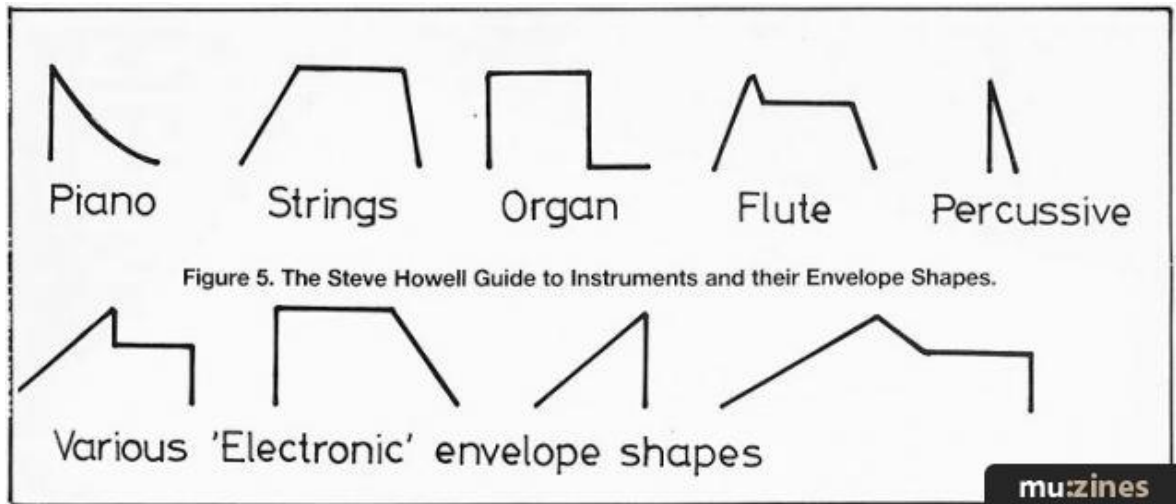


Figura 4.6. Envelope de varis instruments (Font: Music Magazine Archives)

Destacar que la figura on diu "Strings" no es refereix a tots els instruments de corda, sinó que es refereix als instruments de corda fregada, per exemple un violí. Al contrari que a la guitarra triga força temps en arribar al seu màxim i allà es manté fins que al final decau ràpidament.

Com ja s'ha mencionat en capítols anteriors aquestes son les amplituds que s'han de modelar amb el sintetitzador, amb el famós ADSR (Attack, Decay, Sustain i Release), que són els quatre temps d'un so musical.

En el cas de la guitarra i tal com mostra la figura 4.5, té un Attack molt pronunciat, un Decay pronunciat igual però en menys mesura, fins arribar a un llarg Sustain, fins que poc a poc, en el Release, arriba a zero. A la figura 4.5 mostra, també, que es poden obtenir totes les envelope que es desitgin modificant aquest 4 paràmetres.

Realment aquest mòdul el que fa es modificar el senyal electrònic, ha d'estar connectat a un amplificador anomenat VCA o "Voltatge Controlled Amplifier". L'únic que fa es modificar el guany (volum) directament proporcional a la tensió donada per el ADSR, és així com s'obté la intensitat del so desitjat.

Existeixen altres tipus de mòduls per tal d'aconseguir diferents tipus d'efectes i sons. Fins ara, s'ha mencionat que per generar un so, en un primer lloc fa falta un VCO. Bé aquesta afirmació no és del tot certa. A vegades interessa un generador de soroll, el qual, tal com diu el nom, generarà el que és

conegut com a soroll rosa i soroll blanc. Pot ser interessant per, amb els filtres adequats, poder sintetitzar ones de mar, vent i tots aquests sons ambientals.

Però, que és el soroll? En un sintetitzador el soroll es coneix com a un so el qual conté totes les freqüències, és un so format per tots els harmònics de totes de freqüències. Aquest so és caracteritzat per un “shh” quan l’escoltem, per exemple una radio sense sintonitzar cap emissora.

Anteriorment s’ha distingit entre dos tipus de soroll, el soroll rosa i el soroll blanc. El soroll rosa és aquell soroll que conté totes les freqüències de tots els harmònics audibles. El soroll blanc, en canvi, es com l’explicat anteriorment, conte totes les freqüències de tots els harmònics siguin audibles o no.

Tanmateix, el generador de soroll no es l’únic que pot substituir al VCO sinó que un LFO (Low Frequency Oscillator) també pot fer-ho. Amb aquest mòdul el que s’habitu a fer es modular l’amplitud per generar efectes de vibrato. Posteriorment es veurà cada un d’ells en profunditat.

5. Sistemes de Síntesis Modulars

En aquest capítol es farà una explicació dels diferents mòduls que es poden trobar en un sintetitzador, el seu funcionament. En el sintetitzadors comercials aquests mòduls venen incorporats i no son modificables. Hi ha la possibilitat però, de fer un sintetitzador per peces si es volgués, en aquest cas hi haurà total llibertat d'interconnexió entre els diferents components. El principal avantatge d'aquest mètode, és obvi, i es que es pot aconseguir una diversitat molt més gran de timbres i efectes, ja que sempre pots anar ampliant mòduls. El gran inconvenient es que es perd tota la simplicitat d'un sintetitzador comercial.

5.1. CV

El "controlled voltatge" o CV és una senyal que és transportada des del teclat. Primer es parteix de la base d'un teclat. Aquest teclat a cada tecla te una determinada informació relacionada amb la nota musical. Primer el temps de pulsació, no és el mateix si es prem la tecla durant 1s que si es prem durant 10s, i la segona és la freqüència de la nota musical.

Tot això es fa a través de senyals elèctriques de voltatge petit, tal com es mostra a la figura 5.1.

Note Name	Frequency (Hz)	Minimum Input Voltage	Maximum Input Voltage	Output Voltage
C2	131	0.958	1.042	1.000
C#2 (Db2)	139	1.042	1.125	1.083
D2	147	1.125	1.208	1.167
D#2 (Eb2)	156	1.208	1.292	1.250
E2	165	1.292	1.375	1.333
F2	175	1.375	1.458	1.417
F#2 (Gb2)	185	1.458	1.542	1.500
G2	196	1.542	1.625	1.583
G#2 (Ab2)	208	1.625	1.708	1.667
A2	220	1.708	1.792	1.750
A#2 (Bb2)	233	1.792	1.875	1.833
B2	247	1.875	1.958	1.917
C3	262	1.958	2.042	2.000
C#3 (Db3)	277	2.042	2.125	2.083
D3	294	2.125	2.208	2.167
D#3 (Eb3)	311	2.208	2.292	2.250
E3	330	2.292	2.375	2.333
F3	349	2.375	2.458	2.417
F#3 (Gb3)	370	2.458	2.542	2.500
G3	392	2.542	2.625	2.583
G#3 (Ab3)	415	2.625	2.708	2.667
A3	440	2.708	2.792	2.750
A#3 (Bb3)	466	2.792	2.875	2.833
B3	494	2.875	2.958	2.917
C4	523	2.958	3.042	3.000

Table 1: Part of the range of logarithmic frequencies and linear voltages that represent the notes between C2 and C4.

Figura 5.1. Voltatges de senyal CV (Font: Sound on sound)

Aquest senyal simbolitza la freqüència de la nota que correspon a la tecla premuda. Els valors de CV no són diferents per a cada sistema, sinó que són valors estandarditzats en el cas de la norma EURO els valors són 1V/octava a partir del “do”.

Aquests valors són fàcils de calcular, de “do” a “do” hi ha 1 octava per tant la resta de les dotze notes simplement es fa la divisió d’aquest volt entre 12. Aquest resultat es va sumant, d’aquesta manera es van trobant tots els voltatges.

El teclat que es fa servir, és un teclat MIDI (Musical Instrument Digital Interface), com es menciona a la introducció. El llenguatge MIDI segueix una lògica binària, interpretant així si una tecla està premuda o no, la interfície en que es connecta aquest teclat MIDI ha de ser una que pugui convertir aquesta lògica en senyals. Així ho fa doncs el CV. Depenen del teclat MIDI poden enviar altres tipus d’informació, com per exemple, la força en que aquesta tecla és premuda. En el llenguatge MIDI això és conegut com a “velocity”, també hi pot haver altres efectes com el “pitch bend” entre d’altres. Òbviament, quanta més informació es vol obtenir, més elevat és el cost del teclat MIDI. El rang de preus va des de els 50€ fins als d’última generació que poden costar 8000€ aproximadament.

	Entrada	Sortida
Necessari	MIDI	CV
Opcional	-	Velocity, pitch bend, etc.

Figura 5.2. Taula de característiques d’un mòdul de llenguatge MIDI a CV

5.2. VCO

L’objectiu del CV és actuar d’entrada per el mòdul més important del sintetitzador, el VCO (Voltatge Controlled Oscillator). Aquest mòdul proporciona senyals periòdiques amb una freqüència fonamental directament proporcional a la tensió d’entrada.

Com ja s’ha vist anteriorment les senyals de sortida més habituals són senyals sinusoidals, quadrades, dents de serra i triangulars. Igual que en el cas anterior hi ha VCO que proporcionen altres tipus de senyals, amb costos elevats. L’objectiu d’aquestes formes d’ona és contar amb senyals amb gran volum d’harmònics per poder obtenir així una gama de timbre elevada.

	Entrada	Sortida	Control
Necessari	1V/octava	Quadrada, Triangular	Transposició
Opcional	Senyal moduladora, Sincronització	Dent de Serra, Sinusoidal	Nivell de modulació

Figura 5.3. Taula de característiques d'un mòdul VCO

5.2.1. Senyal Sinusoidal

A la senyal sinusoidal també se la coneix com a "To Pur". Aquesta és la senyal més simple que ofereix un VCO i la seva expressió és la següent.

$$x(t) = A \cdot \sin(\omega \cdot t + \varphi) = A \cdot \sin(2 \cdot \pi \cdot f \cdot t + \varphi) \quad (\text{Eq. 5.1})$$

El seu espectre pot ser conegut a través de les series de Fourier tal com s'avançava en un dels capítols anteriors.

$$\begin{aligned} \hat{x}(\omega) &= \int_{-\infty}^{\infty} \sin(\omega_0 t) \cdot e^{-i\omega t} dt = \int_{-\infty}^{\infty} \left(\frac{e^{i\omega_0 t} - e^{-i\omega_0 t}}{2 \cdot i} \right) \cdot e^{-i\omega t} dt & (\text{Eq. 5.2}) \\ &= \frac{1}{2 \cdot i} \int_{-\infty}^{\infty} (e^{-i(\omega - \omega_0)t} - e^{-i(\omega + \omega_0)t}) dt \end{aligned}$$

Deixant com a resultat,

$$\hat{x}(\omega) = i \cdot \pi [\delta(\omega + \omega_0) - \delta(\omega - \omega_0)] \quad (\text{Eq. 5.3})$$

Descartant la informació de les freqüències negatives s'obté l'espectre corresponent a l'ona sinusoidal que és una delta, una sola freqüència. Tal i com es mostra a la figura a continuació.

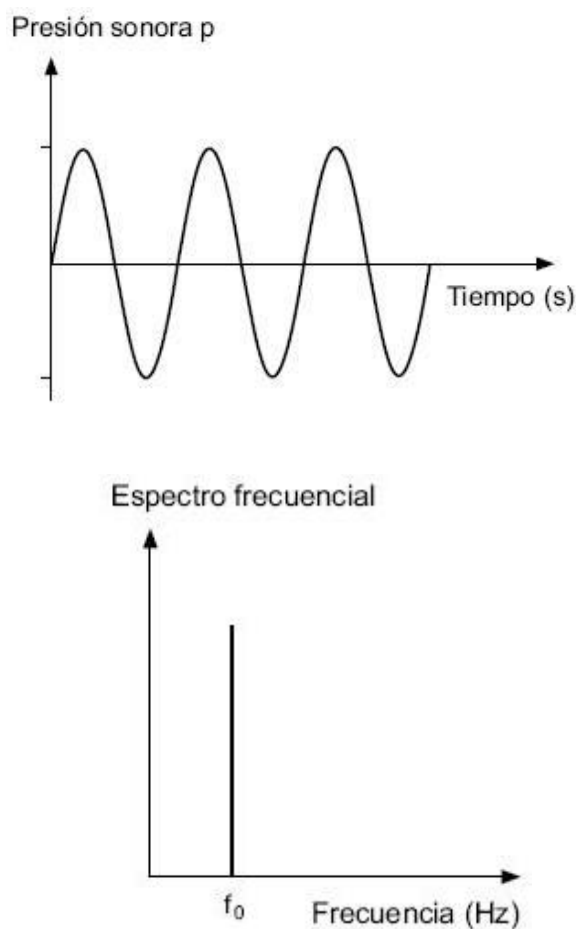


Figura 5.4. Senyal i espectre d'una ona sinusoidal (Font: Beatriz Helena García Uribe)

5.2.2. Senyal Quadrat

El senyal quadrat ja ha estat mencionada amb anterioritat, fins i tot s'ha mostrat com generar l'ona. Es donarà ara doncs una explicació més detallada d'aquest fenomen. Al llarg del temps, com bé s'ha mostrat, consisteix en dos valors de tensió de pendent infinita entre un altre.

$$x(t) = \begin{cases} -1, & -\frac{T}{2} < t < 0 \\ 1, & 0 < t < \frac{T}{2} \end{cases} \quad (\text{Eq. 5.4})$$

Contant que T és el període total.

S'ha de calcular els coeficients de a_0 , a_n , i b_n .

Coefficient a_0 :

$$a_0 = \frac{1}{T} \int_{-T/2}^{T/2} x(t) dt$$

$$a_0 = \frac{2}{T} \left[\int_{-T/2}^0 -dt + \int_0^{T/2} dt \right] = \frac{2}{T} \left[-t \Big|_{-T/2}^0 + t \Big|_0^{T/2} \right] = 0 \quad (\text{Eq. 5.5})$$

Coefficient a_n :

$$a_n = \frac{2}{T} \int_{-T/2}^{T/2} x(t) \cdot \cos(n \cdot \omega_0 \cdot t) dt$$

$$\begin{aligned} a_n &= \frac{2}{T} \left[\int_{-T/2}^0 (-1) \cdot \cos(n \cdot \omega_0 \cdot t) dt + \int_0^{T/2} 1 \cdot \cos(n \cdot \omega_0 \cdot t) dt \right] \\ &= \frac{2}{T} \left[-\frac{1}{n \cdot \omega_0} \cdot \sin(n \cdot \omega_0 \cdot t) \Big|_{-T/2}^0 + \frac{1}{n \cdot \omega_0} \cdot \sin(n \cdot \omega_0 \cdot t) \Big|_0^{T/2} \right] \\ &= 0 \end{aligned} \quad (\text{Eq. 5.6})$$

Coefficient b_n :

$$b_n = \frac{2}{T} \int_{-T/2}^{T/2} x(t) \cdot \sin(n \cdot \omega_0 \cdot t) dt$$

$$\begin{aligned} b_n &= \frac{2}{T} \left[\int_{-T/2}^0 (-1) \cdot \sin(n \cdot \omega_0 \cdot t) dt + \int_0^{T/2} 1 \cdot \sin(n \cdot \omega_0 \cdot t) dt \right] \\ &= \frac{2}{T} \left[\frac{1}{n \cdot \omega_0} \cdot \cos(n \cdot \omega_0 \cdot t) \Big|_{-T/2}^0 - \frac{1}{n \cdot \omega_0} \cdot \cos(n \cdot \omega_0 \cdot t) \Big|_0^{T/2} \right] \\ &= 0 \end{aligned} \quad (\text{Eq. 5.7})$$

Tenint en compte que,

$$w_0 = \frac{2\pi}{T} \quad (\text{Eq. 5.8})$$

S'obté,

$$b_n = \frac{1}{n\pi} \{ [1 - \cos(n \cdot \pi)] - [\cos(n \cdot \pi) - 1] \} = \frac{2}{n\pi} \cdot [1 - (-1)^n] \quad (\text{Eq. 5.9})$$

Finalment la sèrie de Fourier queda com,

$$x(t) = \frac{4}{\pi} \left[\sin(w_0 \cdot t) + \frac{1}{3} \cdot \sin(3 \cdot w_0 \cdot t) + \frac{1}{5} \cdot \sin(5 \cdot w_0 \cdot t) + \dots \right] \quad (\text{Eq. 5.10})$$

$$x(t) = \frac{4}{\pi} \sum_{n=1}^{\infty} \frac{1}{2 \cdot n - 1} \cdot \sin((2 \cdot n - 1) \cdot w_0 \cdot t) \quad (\text{Eq. 5.11})$$

Seguidament es mostrarà una figura, on es pot apreciar el que ja s'havia explicat amb anterioritat i la demostració feta en aquest apartat. A la figura es mostrarà fins al 7é harmònic, que tal com es justificava era l'últim harmònic amb importància.

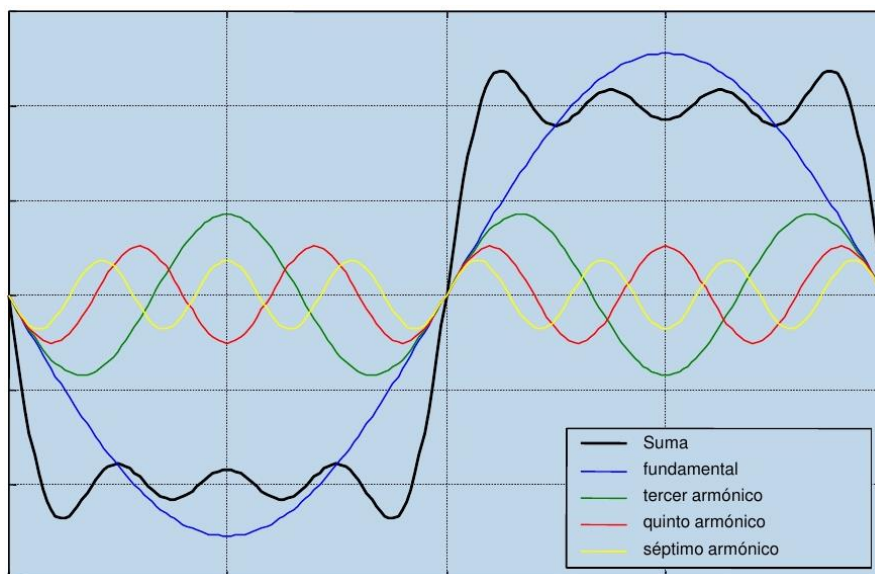


Figura 5.5. Formes d'ones de la sèrie de Fourier per a una ona quadrada (Font: Pato Villacis)

Como es pot comprovar només utilitzant 7 harmònics no és suficient per a l'obtenció d'un senyal quadrat. En el cas que es fes servir 50 harmònics resultaria el següent,

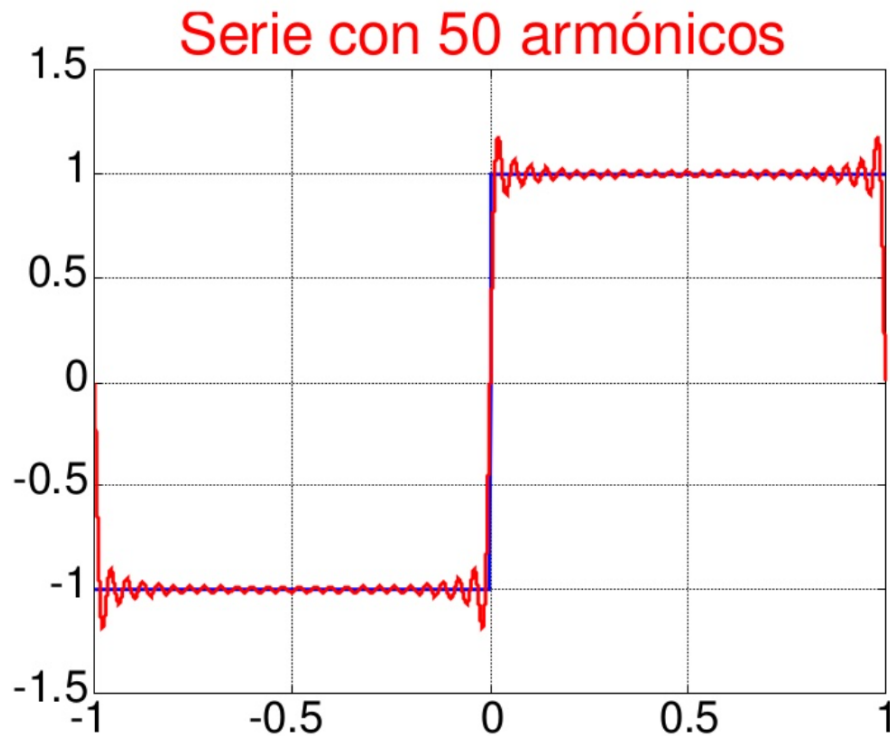


Figura 5.6. Ona quadrada utilitzant 50 harmònics (Font: Pato Villacis)

L'ona quadrada tot i comptar amb 50 harmònics es veu imperfecte, degut a la pendent infinita entre les amplituds, això provoca aquests rissats a les cantonades del pols. És conegut com a efecte Gibbs.

5.2.3. Senyal Triangular

L'ona triangular és una ona periòdica la qual està formada per cicles amb dos cantons d'un triangle isòsceles en el domini del temps. Aquest resultat s'aconsegueix, analògicament, carregant i descarregant un condensador.

Matemàticament,

$$x(t) = \begin{cases} \frac{A}{T/2} \cdot t, & 0 < t < \frac{T}{2} \\ 2 \cdot A - \frac{2 \cdot A}{T} \cdot t, & \frac{T}{2} < t < T \end{cases} \quad (\text{Eq. 5.12})$$

Aquesta sèrie de Fourier és una funció no par, per tant tots els números de "n" pars seran igual a 0. Tanmateix la funció b_n serà igual a 0, tant sols s'haurà de trobar a_0 i a_n .

Coefficient a_0 ,

$$a_0 = \frac{2}{T} \int_0^{T/2} x(t) dt = \frac{2}{A} \cdot \frac{2 \cdot A}{T} \int_0^{T/2} t dt = \frac{4 \cdot A}{T^2} \cdot \frac{t^2}{2} \Big|_0^{T/2} = \frac{4 \cdot A}{2 \cdot T^2} \cdot \left(\frac{T}{2}\right)^2 = \frac{2 \cdot A}{T^2} \cdot \left(\frac{T^2}{4}\right) = \frac{A}{2}$$

$$a_0 = \frac{A}{2} \quad (\text{Eq. 5.13})$$

Coefficient a_n ,

$$a_n = \frac{4}{T} \int_0^{T/2} \frac{2 \cdot A}{T} \cdot t \cdot \cos(n \cdot \omega_0 \cdot t) dt = \frac{8 \cdot A}{T^2} \int_0^{T/2} t \cdot \cos(n \cdot \omega_0 \cdot t) dt$$

$$= \frac{8 \cdot A}{T^2} \cdot \left[\frac{t}{n \cdot \omega_0} \cdot \sin(n \cdot \omega_0 \cdot t) \Big|_0^{T/2} - \int_0^{T/2} \frac{1}{n \cdot \omega_0} \cdot \sin(n \cdot \omega_0 \cdot t) dt \right]$$

$$a_n = \frac{8 \cdot A}{T^2} \cdot \left[\frac{T/2}{n \cdot \frac{2\pi}{T}} \cdot \sin\left(n \cdot \frac{2\pi}{T} \cdot \frac{T}{2}\right) + \frac{1}{n^2 \cdot \omega_0^2} \cdot \left[\cos\left(n \cdot \frac{2\pi}{T} \cdot t \Big|_0^{T/2}\right) \right] \right] \quad (\text{Eq. 5.14})$$

$$= \frac{8 \cdot A \cdot T^2}{T^2 \cdot n \cdot 4\pi^2} \cdot \left[\cos\left(\frac{n \cdot 2\pi \cdot T}{T \cdot 2}\right) - \cos(0) \right] = \frac{-2 \cdot 2 \cdot A}{n^2 \cdot \pi^2} = \frac{-4 \cdot A}{n^2 \cdot \pi^2}$$

Resultant,

$$a_n = \begin{cases} 0, & \text{par} \\ \frac{-4 \cdot A}{n^2 \cdot \pi^2}, & \text{impar} \end{cases} \quad (\text{Eq. 5.15})$$

Obtenint com a resultat final,

$$x(t) = \frac{A}{2} \left[\frac{-4 \cdot A}{\pi^2} \cdot \cos(\omega_0 \cdot t) - \frac{4 \cdot A}{3^2 \cdot \pi^2} \cdot \cos(3 \cdot \omega_0 \cdot t) - \frac{4 \cdot A}{5^2 \cdot \pi^2} \cdot \cos(5 \cdot \omega_0 \cdot t) - \dots \right] \quad (\text{Eq. 5.16})$$

Aquesta ona no s'ha estudiat prèviament, perquè es troba un pas abans que la de dent de serra. L'ona dent de serra és molt més important que la triangular, es fa servir força més, per aquest motiu no ha aparegut fins ara. La seva ona i espectre son els següents.

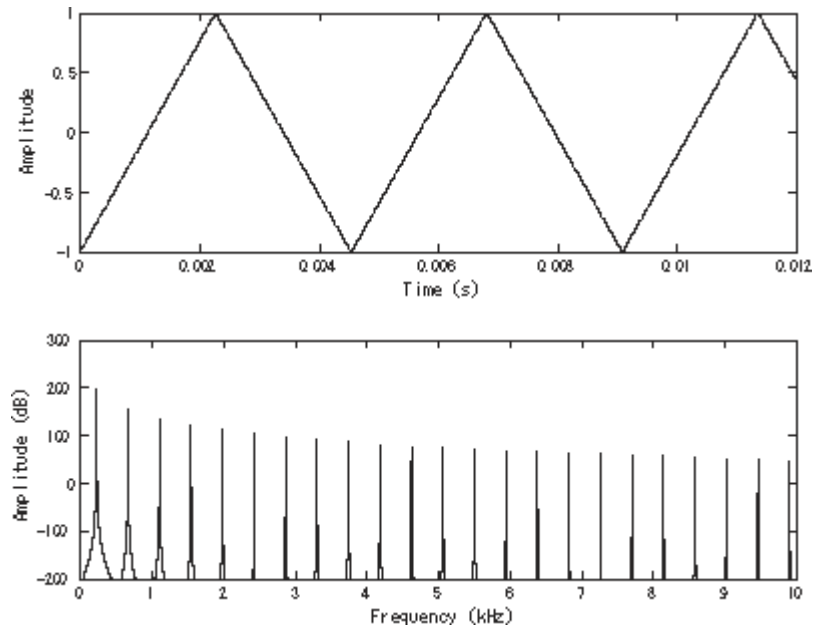


Figura 5.7. Ona i espectre de la senyal triangular (Font: Wikipedia)

Tal com s'ha mencionat, d'aquesta ona es pot anar un pas més enllà i fer la famosa senyal dent de serra.

5.2.4. Senyal Dent de Serra

Com ja s'ha reiterat, la dent de serra és una funció triangular amb asimetria, tenint una de les seves pendents el més abrupte possible, l'ideal seria una pendent infinita. Tanmateix aquesta ona ja ha estat estudiada anteriorment i s'han comentat les seves característiques. Per tant, es desenvoluparà la seva sèrie de Fourier, fins a l'obtenció d'aquesta senyal.

Al igual que amb la senyal triangular aquesta és una funció impar, la diferencia es que aquesta vegada els coeficients a_0 i a_n són 0.

Per tant,

$$x(t) = t \quad -\pi < t < \pi \quad (\text{Eq. 5.17})$$

Com que a_0 i a_n son igual a zero,

$$\begin{aligned}
 b_n &= \frac{2}{T} \int_0^T x(t) \cdot \sin\left(\frac{n \cdot \pi}{T} t\right) dx = \frac{2}{\pi} \int_0^{\pi} t \cdot \sin\left(\frac{n \cdot \pi}{T} \cdot t\right) dt & \text{(Eq. 5.18)} \\
 &= \frac{2}{\pi} \int_0^{\pi} t \cdot \sin\left(\frac{n \cdot \pi}{\pi} \cdot t\right) dt = \frac{2}{\pi} \int_0^{\pi} t \cdot \sin(n \cdot t) dt
 \end{aligned}$$

Resolent l'integral,

$$\begin{aligned}
 a_n &= \frac{2}{\pi} \cdot \left[\frac{\sin(nt) - nt \cdot \cos(nt)}{n^2} \right]_0^{\pi} = \frac{2}{\pi} \cdot \left[\frac{\sin(n\pi) - n\pi \cdot \cos(n\pi)}{n^2} - \frac{\sin(0)}{n^2} \right] & \text{(Eq. 5.19)} \\
 &= \frac{2}{\pi} \cdot \left(-\frac{n\pi \cdot \cos(n\pi)}{n^2} \right) = \frac{-2 \cdot \cos(n\pi)}{n}
 \end{aligned}$$

Obtenint un resultat final,

$$x(t) = \sum_{n=1}^{\infty} \frac{-2 \cdot \cos(n\pi)}{n} \cdot \sin(nt) = \sum_{n=1}^{\infty} \frac{-2}{n} \cdot \sin(nt) \quad \text{(Eq. 5.20)}$$

A la figura 5.8 es mostra com és una senyal dent de serra real on tant sols s'han tingut en compte els primers harmònics.

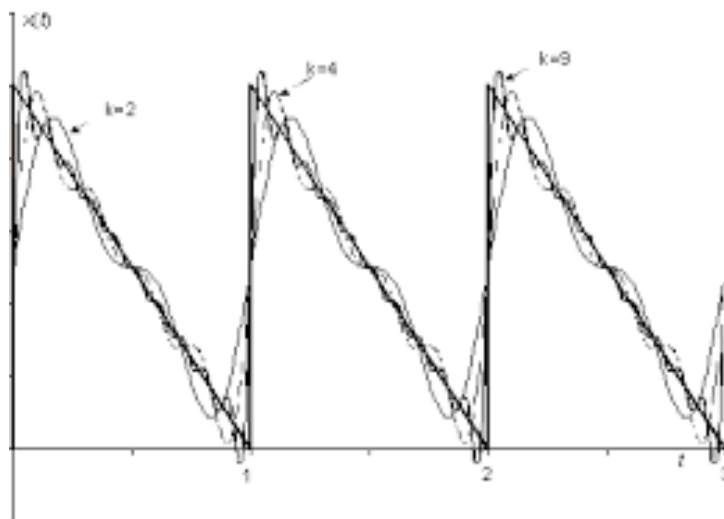


Figura 5.7. Senyal dent de serra real (Font: Rodas)

5.3. Generador de Soroll

Anteriorment s'ha parlat del que era un soroll rosa i un soroll blanc i que aquest es pot posar al lloc del VCO. Aquest mòdul es va dissenyar al principi dels 80, on es començava a descobrir la música electrònica.

El mòdul en si és molt simple, tant sols té una sortida de soroll blanc, una sortida de soroll rosa i habitualment també existeix una sortida de senyal aleatòria, de la qual encara no s'ha parlat. Tal com es mostra a la figura 5.8.

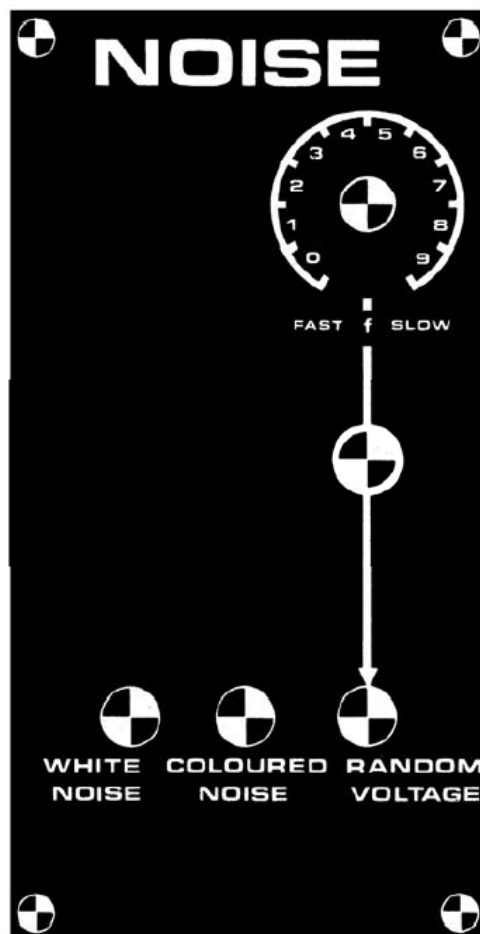


Figura 5.8. Mòdul generador de soroll (Font: Abrana Studios)

Els generadors de soroll a vegades son millors que un VCO depenent del que vulguis sintetitzar. Aplicant els filtres adequats es poden obtenir síntesis de sons com ara, sons de vent, cops metàl·lics, cops de plats, inclús de percussió.

El senyal aleatori és molt important també, encara que menys comú que els altres dos. Encara que no ho sembli que proporcioni una senyal que variï sense ningun patró i fluctuant en el temps de manera aleatòria fa que es puguin produir efectes de so de caire “espacial”, per exemple efectes de la naturalesa que no segueixen una pauta.

5.4. ADSR

Anteriorment s’ha explicat el significat de ADSR (Attack, Decay, Sustain i Release), que traduït seria alguna cosa semblant a atac, atenuació, sosteniment i amortiguació. Com ja s’ha mencionat en capítols previs, els noms es tractaran en anglès, ja que l’ADSR és un concepte molt comú molt utilitzat i és com si fos un nom propi. Amb els 4 paràmetres d’ADSR es pot controlar el VCA, que s’explicarà en el següent capítol, per aconseguir l’efecte desitjat.

El primer paràmetre és l’ “Attack”, aquest paràmetre marca quan de pressa el so arriba al seu pic quan es prem la tecla al sintetitzador. Per exemple, un piano o una guitarra tindran un atac ràpid. En canvi, una flauta o un violi, si es frega la corda, tindrà un atac lent.

Posteriorment hi ha el “Decay”, és un paràmetre de transició, marca quina atenuació tindrà el so després de fer el pic. El “decay” igual que l’ “attack” potser abrupte o pot ser una rampa llarga i suau en el temps.

El tercer element és el “Sustain”, tal com el nom indica, es definiria com quan temps el so aguanta en el temps, fa l’efecte com de pedal en un piano, que precisament també es diu “sustain”, la seva funció és aguantar la nota en el punt on el “decay” l’hagi deixat.

Finalment es troba el “Release”, que és la finalització, al igual que amb l’ “attack” i el “decay” es pot obtenir una finalització abrupte, on el so s’atura de cop, o una finalització suau, on el so disminueixi la seva amplitud fins que no sigui possible la seva audició.

En resum, quan es prem una tecla el control ADSR comença una rampa ascendent (“Attack”), fins a arribar al pic, en aquest moment comença una rampa descendent (“Decay”) la qual arriba a un llindar (“Sustain”), on l’ona és constant fins que es deixa de prémer la tecla. És llavors quan el so fa una última rampa de baixada (“Release”) fins que el so desapareix.

Per a una millor comprensió a la figura següent es mostra una “ADSR envelope”.

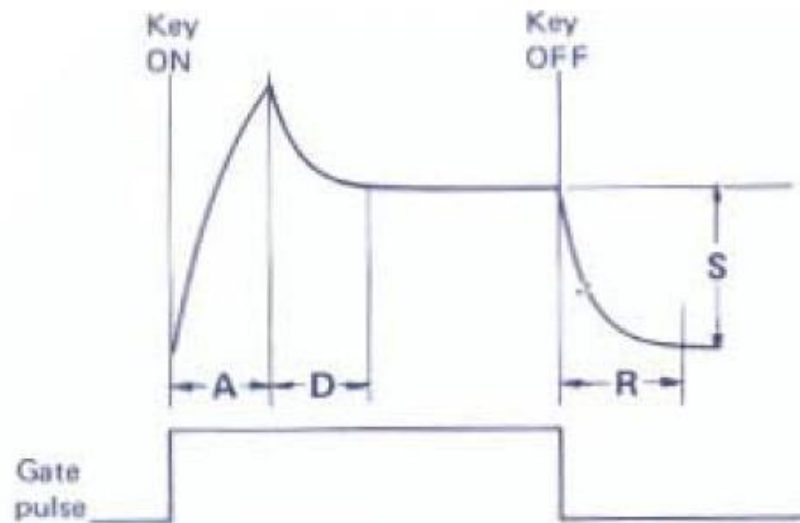


Figura 5.9. "ADSR envelope" (Font: Abrana Studios)

5.5. VCA

Les sigles de VCA signifiquen, "Voltage Controlled Amplifier" traduït amplificador controlat de voltatge. Simplement és un amplificador el qual el guany depèn de una tensió de control, vindria a ser el mateix que un VCO però en comptes de controlar freqüència, es controla el "volum".

D'aquesta manera es poden controlar els diferents mòduls, com el de soroll, o el més important d'ells, el ADSR. L'efecte del VCA és molt gràfica en la següent figura.

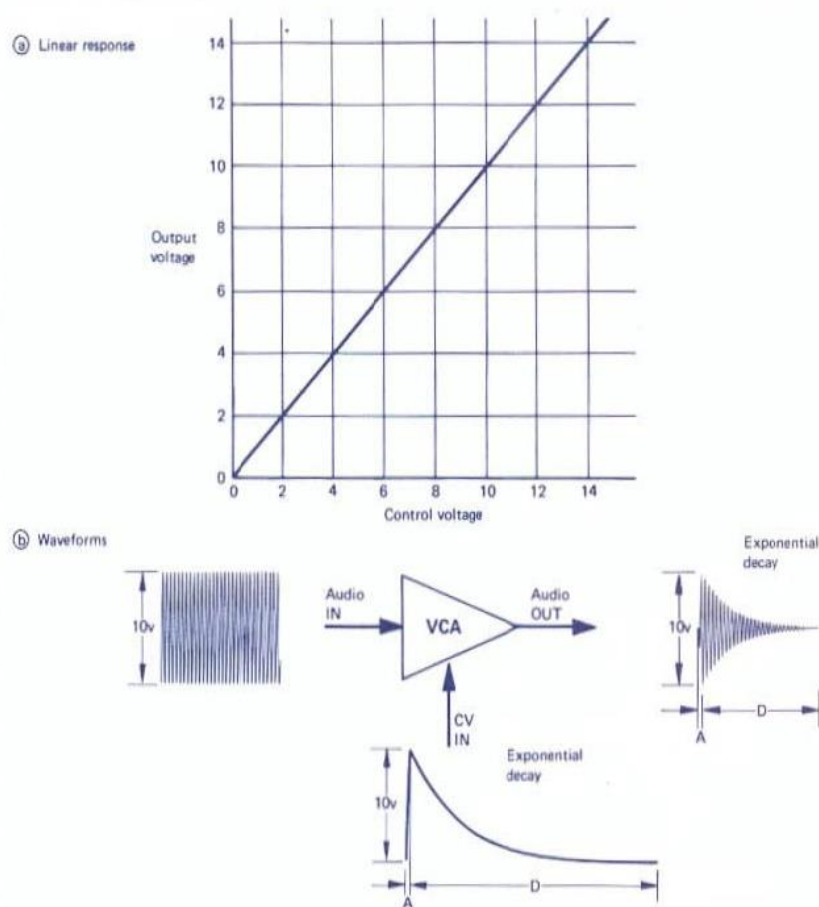


Figura 5.10. Esquema del control VCA (Font: Thomann)

Tal i com es veu a la figura 5.10 hi ha una ona d'entrada al mòdul continua en el temps i hi ha un ADSR d'entrada. El VCA transforma aquesta dues senyals per a obtenir la sortida desitjada.

6. Disseny del Sintetitzador

En aquest capítol s'explicarà com es va dissenyar el sintetitzador des de zero. Tant en la primera part de estudi del sintetitzador, com en l'estudi de dissenyar un algoritme el qual obtenir el resultat buscat i tot el seu procediment.

6.1. Primer Plantejament

Al començament es van plantejar diversos fronts, com per exemple, la manera de dissenyar el sintetitzador, analògic o digital? O quin programa per crear l'algoritme s'escull, *Phyton*, *Labview* o *Matlab*? Fins a, que es vol fer amb aquest projecte un cop acabat i presentat el TFG?

Bé, en un primer moment la intenció era construir un sintetitzador analògic per peces, les quals ja s'han estudiat, però per causa econòmica aquesta opció va ser descartada. Si es fa una mirada ràpida als preus d'un sintetitzador analògic, es veurà que són força elevats.






	<p>Erica Synths Techno System ★★★★★ 2 · Sistema Eurorack</p> <ul style="list-style-type: none">✓ Caja de ritmos modular completamente equipada✓ Compuesta de los siguientes módulos: Sample Drum, Bassline, Bass Drum, Snare...✓ Los módulos Drum están ya conectados internamente con 8 pistas de secuenciad... <p>Disponibilidad inmediata</p>	<p>4.666 €</p> <p>A la cesta Comparar</p>
	<p>Endorphin.es Shuttle System ★★★★★ 2 · Sintetizador modular Eurorack</p> <ul style="list-style-type: none">✓ Contiene 5 módulos Endorphin.es integrados en una carcasa Frap Tools✓ Dispone de dos osciladores cableados al estilo Buchla con Waveshaper✓ Dos generadores funcionales que pueden ser usados como envolventes, LFOs o c... <p>Disponibilidad inmediata</p>	<p>2.222 € 2.380 €</p> <p>A la cesta Comparar</p>
	<p>Erica Synths Black System II ★★★★★ 1 · Sistema modular Eurorack</p> <ul style="list-style-type: none">✓ Medidas (An x Pr x Al): 461 x 270 x 120 mm✓ Incluye 25 cables de Patch✓ Black Wavetable VCO <p>Disponibilidad inmediata</p>	<p>3.444 €</p> <p>A la cesta Comparar</p>
	<p>Erica Synths Pico System III Desktop ★★★★★ 4 · Sistema modular compacto</p> <ul style="list-style-type: none">✓ Construcción completamente analógica✓ Conexiones almacenables a través de Voice-Cards insertables (5 x Preset Cards y ...✓ Medidas (An x Pr x Al): 240 x 140 x 80 mm <p>Disponibilidad inmediata</p>	<p>535 €</p> <p>A la cesta Comparar</p>
	<p>Doepfer A-100BS2-P9 PSU3 ★★★★★ 17 · Sistema analógico modular en versión portátil con maleta</p> <ul style="list-style-type: none">✓ Sistema básico con fuente de alimentación PSU3 (+12V/2000mA, -12V/1200mA, 10...✓ Módulos y platinas Bus integrados✓ Medida exterior (An x Al x Pr - con tapa y pies de goma): 465 x 485 x 235 mm	<p>2.190 €</p> <p>A la cesta Comparar</p>

Figura 6.1. Rang de preus d'un sistema modular complet (Font: Thomann)

A la figura es mostra el rang de preus d'un sistema modular analògic complert a "Thomann". "Thomann" és una de les botigues més grans de música online en tot el món sinó la que més. És per això que s'ha agafat els seus rangs de preus com a vàlids per la quantitat increïble de diversitat de productes en stock i per la diversitat del material que es pot trobar.

Ara que el perquè de fer un sintetitzador digital ja està resolt, passem a resoldre el programari utilitzat. En un primer moment es va decidir fer amb el programa "Labview", perquè personalment havia tingut experiència anterior treballar i manipular ones amb aquest programari.

Fins i tot es van realitzar els cursos Core 1 i Core 2, va ser doncs quan es va decidir canviar el programari a Matlab, altra vegada per una raó econòmica. I és que, per a poder realitzar el sintetitzador i que treballés a temps real, feia falta un DSP i un FPGA.

El DSP (Digital Signal Processor) o processador digital de senyals és un sistema basat en un processador o microprocessador que posseeix un conjunt d'instruccions, hardware i software optimitzats per aplicacions que requereixin operacions numèriques a molt alta velocitat. És per aquest motiu que és ideal per el processat i representació de senyals a temps real.

L'FPGA (Field -Programmable Gate Array) o matriu de portes lògiques programables en camp. És un dispositiu programable que, com bé el nom indica, conté portes lògiques, on d'interconnexió i funcionalitat pot ser configurada en el moment, a temps real, mitjançant un llenguatge de descripció especialitzat.

L'empresa "National Instruments" la qual comercialitza "Labview" 2 ofereix aquests dos sistemes en software i hardware. L'UPC té un acord de programari amb l'empresa i per tant, com a estudiant, el DSP es pot obtenir de forma gratuïta. El problema està en el mòdul FPGA, que no entra dins aquest acord i el seu preu supera els 3000€.



Figura 6.2. Preu del mòdul FPGA de Labview (Font: National Instruments)

Aquesta opció doncs queda completament descartada. El programari finalment escollit és “Matlab”, ja que en el transcurs dels estudis dins la universitat s’hi ha treballat diverses vegades i a nivell personal estic familiaritzat amb l’entorn de treball.

A més a més l’acord de l’UPC amb “Mathworks”, l’empresa que distribueix “Matlab” permet descarregar “Add-Ons” al programa a banda del programa sencer, els quals inclouen el software DSP i el software FPGA, per tant, tots els requisits han estat complerts en aquest cas.

6.2. Plantejament del Algoritme

L’algoritme es va plantejar seguint el format de síntesis additiva, com s’ha mencionat en anterioritat. L’algoritme comença, doncs, premen una tecla la qual té una freqüència determinada assignada. L’assignació de les freqüències a les tecles ve donada per la figura 6.3.

En anterioritat no s’ha explicat de forma concisa aquesta síntesis, perquè no és habitual fer-la servir. La síntesis additiva tracta de generar el so des de zero, es a dir generant cada ona sinusoidal amb el harmònics corresponents fins a l’obtenció del so. Analògicament té un desavantatge molt clar i és que fa falta molts VCO per generar moltes ones i això encareix el preu d’una forma molt important. Com que en aquest cas el sintetitzador és digital no trobem aquest problema, ja que es poden generar tantes ones com es desitgin que el preu continuarà sent el mateix. Aquest és el principal avantatge entre analògic i digital.

Frecuencia (en Hertzios) de las notas musicales

x

	0	1	2	3	4	5	6	7	8	
n=1	do		32.7	65.41	130.81	261.63	523.25	1046.50	2093.00	4186.01
n=2	do#		34.65	69.30	138.59	277.18	554.37	1108.73	2217.46	4434.92
n=3	re		36.71	73.42	146.83	293.66	587.33	1174.66	2349.32	4698.64
n=4	re#		38.89	77.78	155.56	311.13	622.25	1244.51	2489.02	4978.03
n=5	mi		41.2	82.41	164.81	329.63	659.26	1318.51	2637.02	5274.04
n=6	fa	21.826	43.65	87.31	174.61	349.23	698.46	1396.91	2793.83	5587.65
n=7	fa#	23.125	46.25	92.50	185.00	369.99	739.99	1479.98	2959.96	5919.91
n=8	sol	24.50	49.00	98.00	196.00	392.00	783.99	1567.98	3135.96	6271.93
n=9	sol#	25.96	51.91	103.83	207.65	415.30	830.61	1661.22	3322.44	
n=10	la	27.50	55.00	110.00	220.00	440.00	880.00	1760.00	3520.00	
n=11	la#	29.14	58.27	116.54	233.08	466.00	932.33	1864.66	3729.31	
n=12	si	30.87	61.74	123.47	246.94	493.88	987.77	1975.53	3951.07	

Figura 6.3. Freqüència de les notes musicals (Font: Blog de Trobadoresco)

El rang de freqüències escollides són del Do3 fins el Si4. S'han escollit dues octaves perquè d'aquesta manera ja es pot començar a fer coses interessants amb el sintetitzador musicalment parlant, una octava seria limitar-lo en excés.

Posteriorment l'algoritme ha de generar una o varies ones d'acord els paràmetres seleccionats a la interfície. Paral·lelament a aquest procés, es genera el control ADSR. Finalment es crea l' "envelope" ADSR i es multiplica per la ona o ones creades, per a l'obtenció del so.

Com que s'ha fet servir "Matlab App" per fer el sintetitzador es comença primer pel disseny gràfic.

6.3. Disseny Gràfic del Sintetitzador

En una primera instancia el sintetitzador quedava de la següent manera.

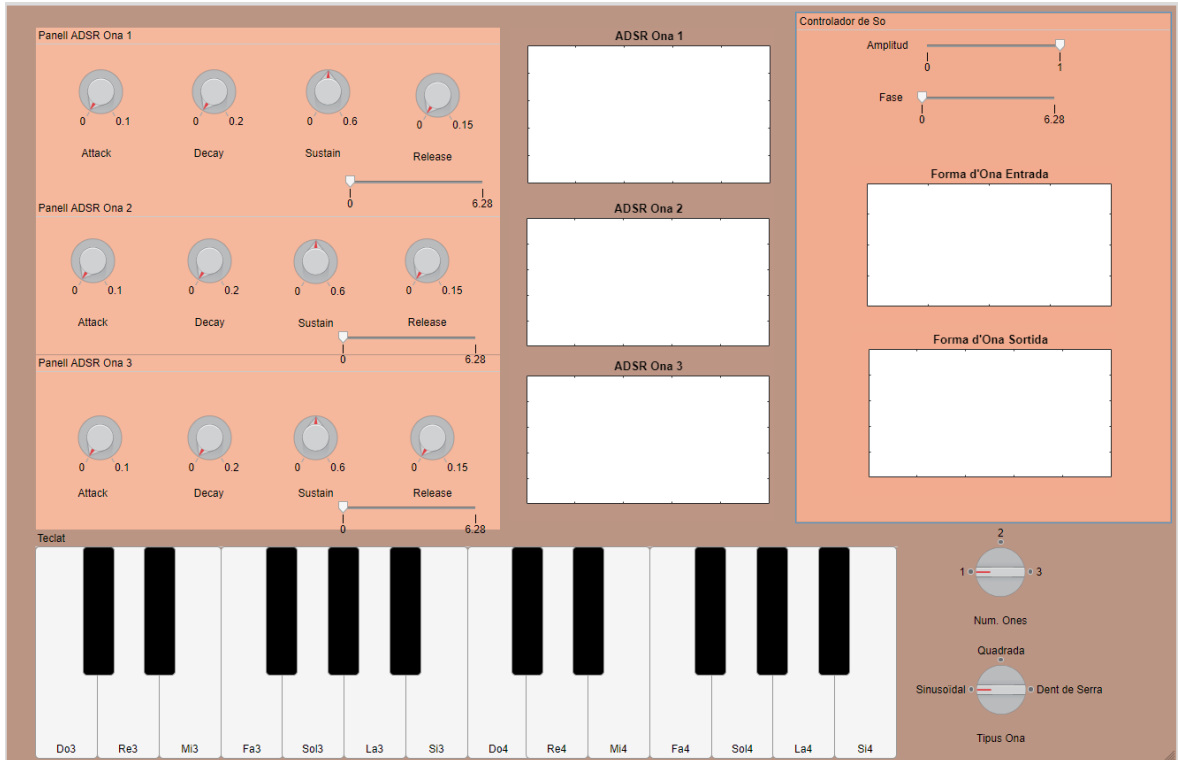


Figura 6.4. Versió número 1 del sintetitzador

La primera versió del sintetitzador, estudiada de forma detallada resulta ineficaç, però te coses molt interessants. Aquesta versió presenta 3 tipus d'ona. Ona sinusoidal, ona quadrada i ona dent de serra, cadascuna d'elles amb el ADSR adjunt. A banda d'això hi ha un "slider" a cada ADSR per modificar la Fase de la ona. A més a més d'un "slider" general que modifica tant l'amplitud final com la fase final.

Finalment es podia veure en els dos gràfics l'ona d'entrada i l'ona de sortida. És així doncs que a partir de l'explicació al apartat 6.1 i el disseny presentat en aquest apartat es comença a construir el sintetitzador.

6.4. Desenvolupament de Algorisme

L'algorisme final es divideix en 8 funcions. Per començar en un primer lloc hi ha la funció que descriu el comportament del sistema quan una tecla és premuda, la funció defineix que quan una tecla és premuda el vector ADSR i el vector AFF (Amplitud, Freqüència i Fase) exporten el so tal i com estiguin modificades les variables en un primer moment.

La segona funció, determina el comportament del sistema quan el ADSR és modificat un cop l'aplicació esta executada. Ja que sinó s'hauria de reiniciar l'aplicació cada vegada que es volgués fer un canvi.

La tercera funció defineix les ones de sortida, aquesta funció va ser molt modificada i es va decidir finalment que es treballaria en 3 casos diferents. El primer cas el més senzill, una ona sinusoidal. Una sinusoidal pura, és a dir, sense cap harmònic l'ona i res més, tot i així sempre amb el ADSR determinat.

El segon cas l'ona quadrada. L'ona quadrada finalment es va introduir de tal manera que fos una ona sinusoidal afegint harmònics, tal i com la síntesis additiva marca. Cadascun d'aquests harmònics pot ser modificat de la forma desitjada.

El tercer cas és l'ona dent de serra. Al igual que amb l'ona quadrada, s'ha generat a partir de ones sinusoidals afegint els seus harmònics, fins el setè. Al igual que en el cas de l'ona quadrada cadascun dels harmònics pot ser modificat de la forma que es desitgi.

En tots aquest casos a banda dels gràfics que s'havien mostrat en el disseny principal del sintetitzador s'hi afegeix un més on es mostrarà el FFT de l'ona final, de tal manera que es pot calibrar millor el so de sortida.

Seguint amb les funcions desenvolupades es troba una quarta on genera l'ona ADSR a partir dels valors establerts en la interfície.

En aquest punt és defineix el vector AFF, definit anteriorment en aquest mateix apartat i és defineix el vector ADSR que ja s'han vist que eren necessaris per a la primera funció.

Al igual que amb l'ADSR, l' AFF també ha de comptar amb una funció que determini el comportament d'aquest quan els valors son modificats a la interfície amb l'algoritme en marxa. Tanmateix hi ha una funció que mostra la gràfica de les ones inicials i finals.

Per últim es troba la funció de canvi de teclat, si és prefereix en comptes de fer sonar les tecles amb el ratolí es decideix fer-ho amb el teclat del ordinador, tant sols prement el "switcher" es podrà aconseguir.

A banda de les funcions hi ha altres aspectes a destacar, com per exemple les "callback", aquestes no es comentaran totes ja que seria molt llarg però si es comentarà la més important que és la "startupFcn". Aquesta és una "callback" on es determina les condicions inicials del sistema, la mesura de l'aplicació, llargada i amplada, el "Sample Rate" de 8192 Hz que és l'estàndard a "Matlab".

El codi sencer esta adjunt a l'annex 1 en format "Matlab .m".

6.5. Resultat Final i Explicació de la Interfície

A la figura 6.5, que es mostra a continuació, s'ensenya el disseny final del sintetitzador.

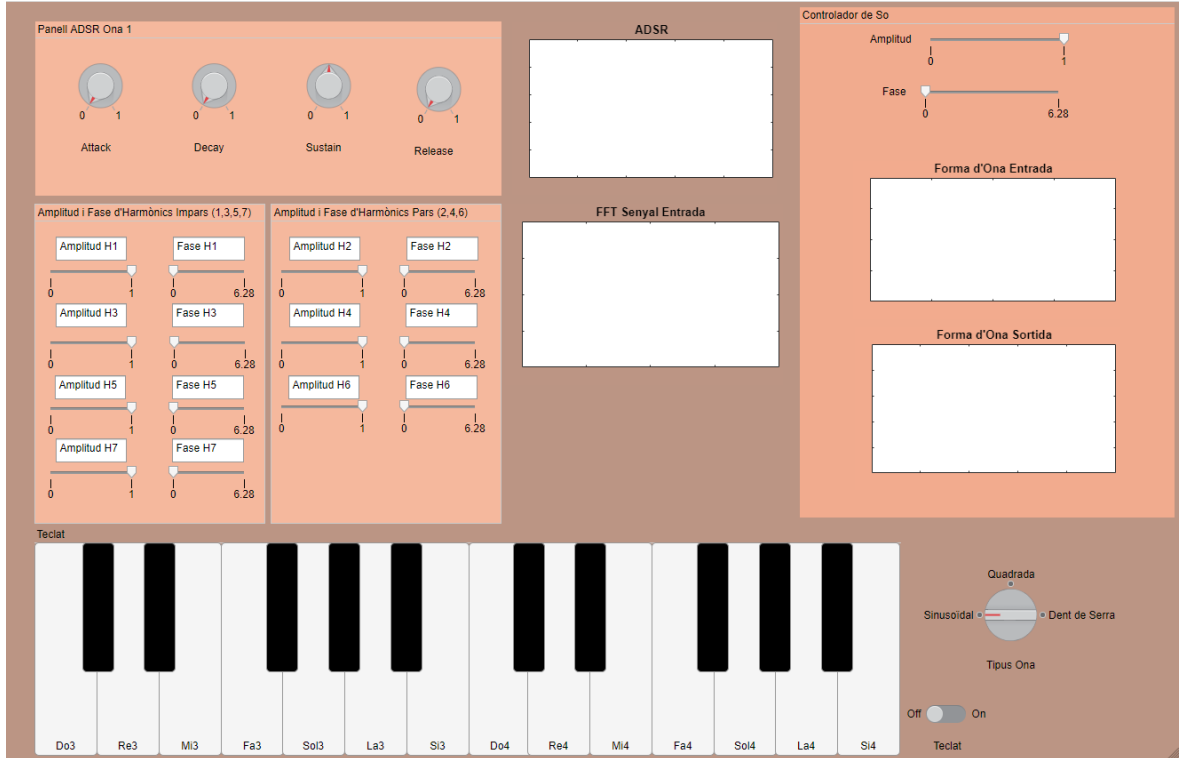


Figura 6.5. Versió final del sintetitzador

Aquest disseny és una millora de la versió principal, tot i que no serà l'última. En el pròxim capítol s'explicarà el que està previst de cara a un futur.

Hi ha un mòdul ADSR que serveix per cada una de les diferents ones, al costat d'una gràfica que mostra l'ona ADSR a temps real, això vol dir que si es modifiquen els paràmetres aquesta ona canviarà de forma segons els paràmetres ajustats.

Es distingeix doncs, un "switcher" amb tres tipus d'ones que poden ser seleccionades, sinusoidal, quadrada i dent de serra. Seria un error assumir que seleccionant una d'aquestes ones i no modificant res més, s'aconsegueix l'ona desitjada. Aquest seleccionador el que simplement fa és "desbloquejar" harmònics, tal com està explicat a la teoria. Es troba doncs, que si es selecciona una ona sinusoidal tant sols es desbloqueja el primer harmònic, el fonamental. En el cas de l'ona quadrada es desbloquejaria el 1r, 3r, 5é i 7é. El mateix passaria amb l'ona dent de serra afegint-li el 2n, 4t i 6é.

Per a la modificació d'aquests harmònics hi ha dos panells a la meitat esquerra, els quals ofereixen modificar l'amplitud i la fase de l'harmònic desitjat. D'aquesta forma es pot aconseguir de manera precisa les característiques adients. A més a més, hi ha afegit un gràfic FFT que mostra en tot moment l'amplitud d'aquests harmònics per així obtenir una visual més clara del que es vol aconseguir.

Finalment apareix un últim panell, el qual, mostra l'ona entrant i l'ona de sortida. De l'ona final se li pot modificar l'amplitud i la fase. Cada moviment es veuria reflectit a les gràfiques en aquell mateix panell.

El fet de poder tenir accés a les gràfiques és molt important, ja que és un plus que permet el digital enfront del analògic. El món de la producció musical esta agafant un rumb totalment cap a la digitalització i sobretot es treballa molt més amb gràfiques i no tant per so (que també te una rellevància molt important). Això es cobrirà al següent capítol on es parlarà del futur del projecte i com aquest es pot arribar a fer un lloc dins la producció musical.

A mode de resum s'adjunta a la següent figura un diagrama on es mostra el funcionament del algoritme dissenyat.

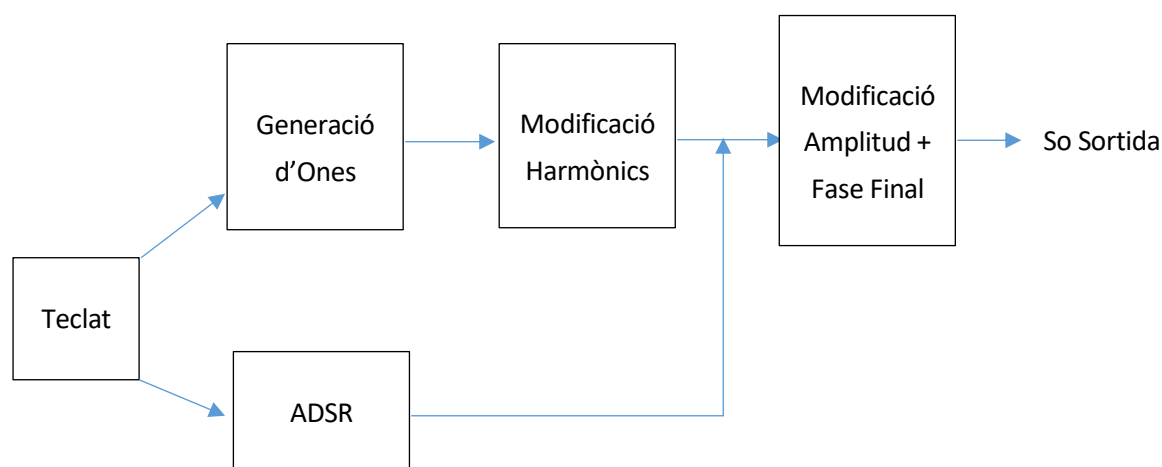


Figura 6.6. Diagrama del funcionament del algoritme

7. Mirada al futur

Tot i que hi ha molta feina feta aquest projecte no es pot limitar a quedar-se aquí. Encara s’ha d’anar un pas mes enllà. El següent pas serà el de produir el mateix codi que s’ha creat però amb alguna altre eina, ja que “Matlab Apps” no permet generar un codi compatible amb un plugin VST (Virtual Studio Technology) d’àudio. Generar aquest codi és molt important per a poder-lo importar dins d’un DAW (Digital Audio Workstation) o multipista.

Un DAW és el programa on es creen les cançons, on es tenen totes les pistes d’àudio i es poden fer tots els processos per a que una mescla soni de forma professional. El exemple de DAW més conegut és el Pro Tools. És curiós que de tots els DAWs existents el Pro Tools possiblement sigui el menys intuïtiu i el més complicat de fer servir, tanmateix el més car. Però com s’ha repetit diverses vegades, el Pro Tools emula molt bé el que és un estudi de so analògic i treballa exactament igual. És per aquest motiu que en aquest moment és el referent a l’industria. Tot i que aquest últims anys Ableton Suite Live 10 i Logic Pro X han trepitjat en el mercat d’una manera sorprenent i innovadora.

Personalment, he treballat amb varis DAWs com son, Cubase, Reaper, Pro Tools i Ableton, sens dubte aquest últim és el meu preferit i és amb el que estic treballant aquest últim any, per la facilitat de treballar amb el programa, això si amb la versió més costosa econòmicament parlant, perquè les altres dues es queden una mica curtes depenen del que es vulgui fer.



Figura 7.1. DAW Ableton Suite Live 10 (Font: Vice)

Un cop el codi es pugui exportar en un arxiu plugin VST, tot i que no és necessari, el següent pas és que des de Pro Tools el validin, ja que és el referent a l'empresa i si un plugin no compta amb la seva validació dona la sensació de que és de baixa qualitat.

A partir d'aquesta validació, el següent objectiu i ja l'últim i pot ser fins i tot ambiciós, serà seguir desenvolupant el sintetitzador i sobretot distribuir-lo de forma gratuïta, ja que l'objectiu és que els productors principiants es puguin beneficiar d'aquest.

8. Anàlisi d'Impacte Ambiental

Que el projecte sigui digital te com a conseqüència directe que tingui molt menys impacte ambiental, enfront a l'analògic. L'analògic te com a desavantatge que cada mòdul esta constituït per moltes peces més petites, ja sigui circuits impresos, díodes, resistències, etc.

Els sintetitzadors analògics, a més a més poden arribar a ser de de grans dimensions, degut a totes les funcions que pot arribar a desenvolupar. Això si, es parla dels casos més extrems. Tot i així fins i tot els sintetitzadors més senzills tenen molt més impacte ambiental que un sintetitzador digital.

Els sintetitzadors analògics estan exposats a possibles averies cosa que provoca més cost de material i més inversió, a banda de més components que son els que directament afecten al medi ambient, en quan a la construcció es refereix.

De la mateixa manera si es volgués ampliar el mòdul del sintetitzador analògic, augmentaria l'impacte inicial provocat per aquest, cosa que en el sintetitzador digital no passa.

En aquest aspecte el sintetitzador digital te un gran avantatge enfront l'analògic, ja que, l'impacte del sintetitzador en si, és nul. Si que és cert que per fer funcionar aquest tipus de sintetitzador, que per si sol no pot fer la funció per la qual esta destinat, fa falta un ordinador que aquest si que te un impacte important però aquest varia depenen de l'usuari.

Com s'ha comentat, doncs, el sintetitzador és molt més net que el sintetitzador analògic en el sentit de que no provoca un impacte ambiental directe a través del seu ús. No esta compost per peces sinó que és un algoritme, així que en el cas d'ampliació es mantindria el mateix impacte ambiental igualment, diferint amb el sintetitzador analògic. És un gran avantatge perquè es poden imitar els sintetitzadors analògics més grans del mercat, que son els que més impacte provoquen, i fer la versió digital on aquest impacte no és percebut.

Conclusions

El sintetitzador ha estat desenvolupat amb èxit i els objectius tot i tenir unes petites modificacions al començament s'han assolit de forma satisfactòria. L'objectiu principal de dissenyar un sintetitzador fent servir el programa "Matlab" s'ha assolit perfectament, si es volgués, es pot tocar a temps real i amb un teclat no cal que sigui amb el ratolí del ordinador.

S'ha comprovat també com amb només 7 harmònics es poden crear ones quadrades i ones triangulars, amb els paràmetres correctes, i que es podien arribar a intuir diferents sons amb l'aplicació creada tant sols coneixent l'espectre de l'ona.

La interfície és clara i concisa, un fàcil ús. Pot ser la part més complicada és la dels harmònics però no hauria de resultar un problema si es llegeix de forma diagonal el que fa cada cosa i perquè serveix. De totes formes, en un futur l'opció més adient és canviar el llenguatge de programació, tal com s'ha explicat "Matlab Apps" és limitat.

Tal com s'ha destacat en el últim capítol, el sintetitzador està lluny de poder competir amb altres sintetitzadors que es poden aconseguir per un preu econòmic, o inclús pels que estan ja incorporats dins d'un DAW quan s'adquireix.

Tot i que els resultats són satisfactoris encara queda molta feina per fer i petites coses a millorar. El treball ha estat molt productiu en quant a guany de coneixement, més de cara a conceptes i les formes de plantejar un mateix problema com és el de construir el sintetitzador, l'exemple més clar ha estat escollir si el sintetitzador seguia una síntesis additiva o una síntesis sostractiva.

Anàlisi Econòmica

Un enginyer de so o un productor musical, que en la majoria de casos una mateixa persona imparteix les dues professions guanya aproximadament entre 20 i 60 euros bruts l'hora, aquest valor serà més o menys elevat depenen del nivell de cada professional.

Per a fer aquest pressupost s'agafarà el llindar més baix, per tant, 20€ l'hora, ja que les hores dedicades no han estat les d'un professional sinó que les d'un estudiant inexpert en l'àmbit. En dissenyar l'algoritme es va trigar unes 70 hores en total aproximadament. Això resultaria d'un total de 1400 € que reitero és el mínim.

Donades les característiques del programa i fent una recerca de sintetitzadors digitals del mateix estil que el dissenyat en aquest projecte. Es podria vendre fàcilment a 40 – 50 € la unitat. Per tant es recuperaria la inversió no immediatament però si de forma ràpida. La majoria de vegades en aquests casos i més en una marca no coneguda, el que es fa és contactar amb un altre marca del món de la producció que amb cada compra de una de les seves unitats és regali el producte. D'aquesta forma les dues marques surten beneficiades.

Bibliografia

- Diccionari de la llengua catalana (Institut d'Estudis Catalans)
- Tuning, Timbre, Spectrum, Scale 2nd Edition (William Sethares)
- If a tree falls in the forest (American National Standard Institute)
- A New Diccionari of Music (Arthur Jacobs)
- Note Processing for Guitar Effects (Geo, R. G. Keen)
- Back to Basics (Steve Howells)
- From Sample & Hold to Sample – rate Converters (Sound on sound, Gordon Reid)
- Unidad IV (Conceptos básicos de Acústica Musical, Beatriz Helena García Uribe)
- Serie de Fourier (Pato Villacis)
- Fourier Series (Washington State University, Yuan Zhang)
- Desarrollo en sèrie de Fourier del movimiento periódico (Rodas)
- Procesador digital de señales (Wikipedia)
- Field-Programmable Gate Array (Wikipedia)
- Desarrollo de una aplicación de secuenciado MIDI en Matlab (Francisco José Molina García)
- Diseño e implementación de un sintetizador analógico modular (Gonzalo Recio Cervantes)
- Curso sobre sintetizadores analogico (Abrana Studios)

Annex A

A1. Codi “Matlab” del sintetitzador

```
2 classdef SinteTFG < matlab.apps.AppBase
3
4     % Properties that correspond to app components
5     properties (Access = public)
6         UIFigure                matlab.ui.Figure
7         PanellADSRona1          matlab.ui.container.Panel
8         AttackKnobLabel         matlab.ui.control.Label
9         AttackKnob              matlab.ui.control.Knob
10        DecayKnobLabel          matlab.ui.control.Label
11        DecayKnob               matlab.ui.control.Knob
12        SustainKnobLabel        matlab.ui.control.Label
13        SustainKnob             matlab.ui.control.Knob
14        ReleaseKnobLabel        matlab.ui.control.Label
15        ReleaseKnob             matlab.ui.control.Knob
16        ControladordeSoPanel    matlab.ui.container.Panel
17        UIAxes                  matlab.ui.control.UIAxes
18        AmplitudSliderLabel     matlab.ui.control.Label
19        AmplitudSlider          matlab.ui.control.Slider
20        FaseSliderLabel         matlab.ui.control.Label
21        FaseSlider              matlab.ui.control.Slider
22        FormaOna                matlab.ui.control.UIAxes
23        TeclatPanel             matlab.ui.container.Panel
24        Do3                     matlab.ui.control.Button
25        Re3                     matlab.ui.control.Button
26        Mi3Button               matlab.ui.control.Button
27        Fa3Button               matlab.ui.control.Button
28        Do4Button               matlab.ui.control.Button
29        Si3Button               matlab.ui.control.Button
30        La3Button               matlab.ui.control.Button
31        Sol3Button              matlab.ui.control.Button
32        Re4Button               matlab.ui.control.Button
33        Mi4Button               matlab.ui.control.Button
34        Fa4Button               matlab.ui.control.Button
35        Sol4Button              matlab.ui.control.Button
36        La4Button               matlab.ui.control.Button
37        Si4Button               matlab.ui.control.Button
38        Do3_2                   matlab.ui.control.Button
39        Re3_2                   matlab.ui.control.Button
40        Fa3_2                   matlab.ui.control.Button
41        Sol3_2                  matlab.ui.control.Button
42        La3_2                   matlab.ui.control.Button
43        Do4_2                   matlab.ui.control.Button
44        Re4_2                   matlab.ui.control.Button
```

```

45         Fa4_2                 matlab.ui.control.Button
46         Sol144_2             matlab.ui.control.Button
47         La4_2                 matlab.ui.control.Button
48         OnaADSR               matlab.ui.control.UIAxes
49         FreqLabel             matlab.ui.control.Label
50         TipusOnaKnobLabel     matlab.ui.control.Label
51         TipusOnaKnob          matlab.ui.control.DiscreteKnob
52         AmplitudiFasedHarmnicsImpars1357Panel
matlab.ui.container.Panel
53         FaseSlider_1         matlab.ui.control.Slider
54         FaseSlider_3         matlab.ui.control.Slider
55         FaseSlider_5         matlab.ui.control.Slider
56         FaseSlider_7         matlab.ui.control.Slider
57         AmplitudSlider_1     matlab.ui.control.Slider
58         AmplitudSlider_3     matlab.ui.control.Slider
59         AmplitudSlider_5     matlab.ui.control.Slider
60         AmplitudSlider_7     matlab.ui.control.Slider
61         TextArea              matlab.ui.control.TextArea
62         TextArea_2           matlab.ui.control.TextArea
63         TextArea_3           matlab.ui.control.TextArea
64         TextArea_4           matlab.ui.control.TextArea
65         TextArea_5           matlab.ui.control.TextArea
66         TextArea_6           matlab.ui.control.TextArea
67         TextArea_7           matlab.ui.control.TextArea
68         TextArea_8           matlab.ui.control.TextArea
69         AmplitudiFasedHarmnicsPars246Panel  matlab.ui.container.Panel
70         FaseSlider_2         matlab.ui.control.Slider
71         FaseSlider_4         matlab.ui.control.Slider
72         FaseSlider_6         matlab.ui.control.Slider
73         AmplitudSlider_2     matlab.ui.control.Slider
74         AmplitudSlider_4     matlab.ui.control.Slider
75         AmplitudSlider_6     matlab.ui.control.Slider
76         TextArea_9           matlab.ui.control.TextArea
77         TextArea_10          matlab.ui.control.TextArea
78         TextArea_11          matlab.ui.control.TextArea
79         TextArea_12          matlab.ui.control.TextArea
80         TextArea_13          matlab.ui.control.TextArea
81         TextArea_14          matlab.ui.control.TextArea
82         FFT                   matlab.ui.control.UIAxes
83         TeclatSwitchLabel    matlab.ui.control.Label
84         TeclatSwitch         matlab.ui.control.Switch
85     end
86
87
88     properties (Access = public)
89         VS;
90         VecADSR;
91         VecADSR2;
92         VecADSR3;
93         Ona;

```

```
94         NOnes
95         VecAFF;
96         T;
97         Fs;
98         TeclatPC;
99     end
100
101     methods (Access = public)
102
103         function handle_TeclaPushed (app,Hz)
104             [app.VecADSR] =
105             initADSR (app, app.AttackKnob.Value, app.DecayKnob.Value, app.SustainKnob
106             .Value, app.ReleaseKnob.Value);
107
108             app.FreqLabel.Text = num2str (Hz);
109             app.VecAFF =
110             initAFF (app, app.VecAFF (1), Hz, app.VecAFF (3));
111
112             Amp = app.VecAFF (1);
113             Freq = Hz;
114             Fase = app.VecAFF (3);
115
116             soundOut = getSound (app, Amp, Freq, Fase);
117             sound (soundOut, app.Fs)
118             plotSound (app, soundOut)
119
120         end
121
122         function handle_ADSRChanged (app)
123             [ADSR] =
124             initADSR (app, app.AttackKnob.Value, app.DecayKnob.Value, app.SustainKnob
125             .Value, app.ReleaseKnob.Value);
126             envADSR =
127             getADSR (app, ADSR (1), ADSR (2), ADSR (3), ADSR (4), app.T, app.Fs);
128
129             plot (app.OnaADSR, app.VS, envADSR)
130
131         end
132         function WaveOut = getSound (app, Amp, Freq, Fase)
133
134             switch app.Ona
135                 case 'Sinusoidal'
136                     envAmp =
137                     getADSR (app, app.VecADSR (1), app.VecADSR (2), app.VecADSR (3), app.VecADSR (
138                     4), app.T, app.Fs);
139
140                     FOna1 =
141                     app.AmplitudSlider_1.Value*sin (2*pi*(Freq) .*app.VS+app.FaseSlider_1.V
142                     alue)
143
144                     WaveOut = envAmp.*FOna1;
145                     plot (app.FormaOna, FOna1)
```

```

135         Y=fft(WaveOut);
136         P2 = abs(Y/10000);
137         P1 = P2(1:10000/2+1);
138         P1(2:end-1) = 2*P1(2:end-1);
139         fp = 1000*(0:(10000/2))/10000;
140         plot(app.FFT,fp,P1)
141
142
143         case 'Quadrada'
144
145
146             envAmp =
147             getADSR(app,app.VecADSR(1),app.VecADSR(2),app.VecADSR(3),app.VecADSR(
148             4),app.T,app.Fs);
149             FOna1 =
150             app.AmplitudSlider_1.Value*sin(2*pi*(Freq).*app.VS+app.FaseSlider_1.V
151             alue);
152             FOna3 =
153             app.AmplitudSlider_3.Value*sin(2*pi*3*(Freq).*app.VS+app.FaseSlider_3
154             .Value);
155             FOna5 =
156             app.AmplitudSlider_5.Value*sin(2*pi*5*(Freq).*app.VS+app.FaseSlider_5
157             .Value);
158             FOna7 =
159             app.AmplitudSlider_7.Value*sin(2*pi*7*(Freq).*app.VS+app.FaseSlider_7
160             .Value);
161             WaveOut =
162             (envAmp.*FOna1)+(envAmp.*FOna3)+(envAmp.*FOna5)+(envAmp.*FOna7);
163             plot(app.FormaOna,WaveOut)
164             Y=fft(WaveOut);
165             P2 = abs(Y/10000);
166             P1 = P2(1:10000/2+1);
167             P1(2:end-1) = 2*P1(2:end-1);
168             fp = 1000*(0:(10000/2))/10000;
169             plot(app.FFT,fp,P1)
170
171         case 'Dent de Serra'
172
173             envAmp =
174             getADSR(app,app.VecADSR(1),app.VecADSR(2),app.VecADSR(3),app.VecADSR(
175             4),app.T,app.Fs);
176             FOna1 =
177             app.AmplitudSlider_1.Value*sin(2*pi*(Freq).*app.VS+app.FaseSlider_1.V
178             alue);
179             FOna2 =
180             app.AmplitudSlider_2.Value*sin(2*pi*2*(Freq).*app.VS+app.FaseSlider_2
181             .Value);
182             FOna3 =
183             app.AmplitudSlider_3.Value*sin(2*pi*3*(Freq).*app.VS+app.FaseSlider_3
184             .Value);
185             FOna4 =
186             app.AmplitudSlider_4.Value*sin(2*pi*4*(Freq).*app.VS+app.FaseSlider_4
187             .Value);

```

```

167             FOna5 =
app.AmplitudSlider_5.Value*sin(2*pi*5*(Freq).*app.VS+app.FaseSlider_5
.Value);
168             FOna6 =
app.AmplitudSlider_6.Value*sin(2*pi*6*(Freq).*app.VS+app.FaseSlider_6
.Value);
169             FOna7 =
app.AmplitudSlider_7.Value*sin(2*pi*7*(Freq).*app.VS+app.FaseSlider_7
.Value);
170             WaveOut =
(envAmp.*FOna1)+(envAmp.*FOna2)+(envAmp.*FOna3)+(envAmp.*FOna4)+(envA
mp.*FOna5)+(envAmp.*FOna6)+(envAmp.*FOna7);
171             plot(app.FormaOna,WaveOut)
172             Y=fft(WaveOut);
173             P2 = abs(Y/10000);
174             P1 = P2(1:10000/2+1);
175             P1(2:end-1) = 2*P1(2:end-1);
176             fp = 1000*(0:(10000/2))/10000;
177             plot(app.FFT,fp,P1)
178
179             end
180
181         end
182         function y = getADSR(app,A,D,S,R,T,Fs)
183
184             N = T*Fs;
185             t = ([0:N-1]/Fs);
186             k = [0 (0.01+A) (0.12+D) T - (0.601-S) - R-0.02 T];
187             y = interp1(k,[0 1 0.4 0.4 0],t,"linear");
188
189         end
190         function [AFF] = initAFF(app,Amplitud,Frequencia,Fase)
191             Amp = Amplitud;
192             Frq = Frequencia;
193             Shf = Fase;
194             [AFF] = [Amp Frq Shf];
195
196         end
197         function handle_AFFChanged(app)
198             app.VecAFF =
initAFF(app,app.AmplitudSlider.Value,str2double(app.FreqLabel.Text),a
pp.FaseSlider.Value);
199             Amp = app.VecAFF(1);
200             Frq = app.VecAFF(2);
201             Fase = app.VecAFF(3);
202
203             soundOut = getSound(app,Amp,Frq,Fase);
204             plotSound(app,soundOut)
205         end
206         function [ADSR] =
initADSR(app,Attack,Decay,Sustain,Release)

```

```

207         Att = Attack;
208         Dec = Decay;
209         Sus = Sustain;
210         Rel = Release;
211
212         [ADSR] = [Att Dec Sus Rel];
213     end
214
215     function plotSound(app,y)
216         plot(app.UIAxes,app.VS,y)
217     end
218
219 end
220
221
222 % Callbacks that handle component events
223 methods (Access = private)
224
225     % Code that executes after component creation
226     function startupFcn(app)
227         app.UIFigure.Position = [25,25,1323,857];
228         app.Fs = 8192;
229         app.T = 1;
230         N = app.T * app.Fs;
231         app.VS = [0:N-1]/app.Fs;
232         app.Ona = 'Sinusoidal';
233
234
235         app.VecAFF =
            initAFF(app,app.AmplitudSlider.Value,str2double(app.FreqLabel.Text),a
            pp.FaseSlider.Value);
236
237         [ADSR] =
            initADSR(app,app.AttackKnob.Value,app.DecayKnob.Value,app.SustainKnob
            .Value,app.ReleaseKnob.Value);
238         app.VecADSR =
            getADSR(app,ADSR(1),ADSR(2),ADSR(3),ADSR(4),app.T,app.Fs);
239
240
241
242         plot(app.OnaADSR,app.VS,app.VecADSR)
243
244     end
245
246     % Button pushed function: Do3
247     function Do3ButtonPushed(app, event)
248         handle_TeclaPushed(app,130.81);
249     end
250
251     % Button pushed function: Do3_2

```



```
252     function Do3_2ButtonPushed(app, event)
253         handle_TeclaPushed(app,138.59);
254     end
255
256     % Button pushed function: Re3
257     function Re3ButtonPushed(app, event)
258         handle_TeclaPushed(app,146.83);
259     end
260
261     % Button pushed function: Re3_2
262     function Re3_2ButtonPushed(app, event)
263         handle_TeclaPushed(app,155.56);
264     end
265
266     % Button pushed function: Mi3Button
267     function Mi3ButtonPushed(app, event)
268         handle_TeclaPushed(app,164.81);
269     end
270
271     % Button pushed function: Fa3Button
272     function Fa3ButtonPushed(app, event)
273         handle_TeclaPushed(app,174.61);
274     end
275
276     % Button pushed function: Fa3_2
277     function Fa3_2ButtonPushed(app, event)
278         handle_TeclaPushed(app,185);
279     end
280
281     % Button pushed function: Sol3Button
282     function Sol3ButtonPushed(app, event)
283         handle_TeclaPushed(app,196);
284     end
285
286     % Button pushed function: Sol3_2
287     function Sol3_2ButtonPushed(app, event)
288         handle_TeclaPushed(app,207.65);
289     end
290
291     % Button pushed function: La3Button
292     function La3ButtonPushed(app, event)
293         handle_TeclaPushed(app,220);
294     end
295
296     % Button pushed function: La3_2
297     function La3_2ButtonPushed(app, event)
298         handle_TeclaPushed(app,233.08);
299     end
300
301     % Button pushed function: Si3Button
```

```
302     function Si3ButtonPushed(app, event)
303         handle_TeclaPushed(app,246.94);
304     end
305
306     % Button pushed function: Do4Button
307     function Do4ButtonPushed(app, event)
308         handle_TeclaPushed(app,261.63);
309     end
310
311     % Button pushed function: Do4_2
312     function Do4_2ButtonPushed(app, event)
313         handle_TeclaPushed(app,277.18);
314     end
315
316     % Button pushed function: Re4Button
317     function Re4ButtonPushed(app, event)
318         handle_TeclaPushed(app,293.66);
319     end
320
321     % Button pushed function: Re4_2
322     function Re4_2ButtonPushed(app, event)
323         handle_TeclaPushed(app,311.13);
324     end
325
326     % Button pushed function: Mi4Button
327     function Mi4ButtonPushed(app, event)
328         handle_TeclaPushed(app,329.63);
329     end
330
331     % Button pushed function: Fa4Button
332     function Fa4ButtonPushed(app, event)
333         handle_TeclaPushed(app,349.23);
334     end
335
336     % Button pushed function: Fa4_2
337     function Fa4_2ButtonPushed(app, event)
338         handle_TeclaPushed(app,369.99);
339     end
340
341     % Button pushed function: Sol4Button
342     function Sol4ButtonPushed(app, event)
343         handle_TeclaPushed(app,392);
344     end
345
346     % Button pushed function: Sol44_2
347     function Sol44_2ButtonPushed(app, event)
348         handle_TeclaPushed(app,415.30);
349     end
350
351     % Button pushed function: La4Button
```

```
352     function La4ButtonPushed(app, event)
353         handle_TeclaPushed(app,440);
354     end
355
356     % Button pushed function: La4_2
357     function La4_2ButtonPushed(app, event)
358         handle_TeclaPushed(app,466);
359     end
360
361     % Button pushed function: Si4Button
362     function Si4ButtonPushed(app, event)
363         handle_TeclaPushed(app,493.88);
364     end
365
366     % Value changed function: AttackKnob
367     function AttackKnobValueChanged(app, event)
368         handle_ADSRChanged(app);
369     end
370
371     % Value changed function: DecayKnob
372     function DecayKnobValueChanged(app, event)
373         handle_ADSRChanged(app);
374     end
375
376     % Value changed function: SustainKnob
377     function SustainKnobValueChanged(app, event)
378         handle_ADSRChanged(app);
379     end
380
381     % Value changed function: ReleaseKnob
382     function ReleaseKnobValueChanged(app, event)
383         handle_ADSRChanged(app);
384     end
385
386     % Value changed function: AmplitudSlider
387     function AmplitudSliderValueChanged(app, event)
388         handle_AFFChanged(app);
389     end
390
391     % Value changed function: FaseSlider
392     function FaseSliderValueChanged(app, event)
393         handle_AFFChanged(app);
394     end
395
396     % Value changed function: TipusOnaKnob
397     function TipusOnaKnobValueChanged(app, event)
398         app.Ona = app.TipusOnaKnob.Value;
399
400     end
401
```

```
402     % Callback function
403     function NumOnesKnobValueChanged(app, event)
404         app.NOnes = app.NumOnesKnob.Value;
405
406     end
407
408     % Callback function
409     function AttackKnob_2ValueChanged(app, event)
410         handle_ADSRChanged(app);
411
412     end
413
414     % Callback function
415     function DecayKnob_2ValueChanged(app, event)
416         handle_ADSRChanged(app);
417     end
418
419     % Callback function
420     function SustainKnob_2ValueChanged(app, event)
421         handle_ADSRChanged(app);
422     end
423
424     % Callback function
425     function ReleaseKnob_2ValueChanged(app, event)
426         handle_ADSRChanged(app);
427     end
428
429     % Callback function
430     function AttackKnob_3ValueChanged(app, event)
431         handle_ADSRChanged(app);
432     end
433
434     % Callback function
435     function DecayKnob_3ValueChanged(app, event)
436         handle_ADSRChanged(app);
437     end
438
439     % Callback function
440     function SustainKnob_3ValueChanged(app, event)
441         handle_ADSRChanged(app);
442     end
443
444     % Callback function
445     function ReleaseKnob_3ValueChanged(app, event)
446         handle_ADSRChanged(app);
447     end
448
449     % Value changed function: TeclatSwitch
450     function TeclatSwitchValueChanged(app, event)
451         value = app.TeclatSwitch.Value;
```

```
452     app.TeclatPC = figure('KeyPressFcn', @TeclatOn, 'Name',
'Key Press' , 'Position',[-40,-40,0,0]);
453     if strcmp(value , 'Off')
454         close all;
455     end
456     function TeclatOn(~,event)
457
458         switch event.Key
459             case 'z'
460                 handle_TeclaPushed(app,130.81);
461             case 's'
462                 handle_TeclaPushed(app,138.59);
463             case 'x'
464                 handle_TeclaPushed(app,146.83);
465             case 'd'
466                 handle_TeclaPushed(app,155.56);
467             case 'c'
468                 handle_TeclaPushed(app,164.81);
469             case 'f'
470                 handle_TeclaPushed(app,174.61);
471             case 'v'
472                 handle_TeclaPushed(app,185);
473             case 'b'
474                 handle_TeclaPushed(app,196);
475             case 'h'
476                 handle_TeclaPushed(app,207.65);
477             case 'n'
478                 handle_TeclaPushed(app,220);
479             case 'j'
480                 handle_TeclaPushed(app,233.08);
481             case 'm'
482                 handle_TeclaPushed(app,246.94);
483             case 'q'
484                 handle_TeclaPushed(app,261.63);
485             case '2'
486                 handle_TeclaPushed(app,277.18);
487             case 'w'
488                 handle_TeclaPushed(app,293.66);
489             case '3'
490                 handle_TeclaPushed(app,311.13);
491             case 'e'
492                 handle_TeclaPushed(app,329.23);
493             case '4'
494                 handle_TeclaPushed(app,369.99);
495             case 'r'
496                 handle_TeclaPushed(app,392);
497             case 't'
498                 handle_TeclaPushed(app,415.30);
499             case '6'
500                 handle_TeclaPushed(app,440);
```

```

501         case 'y'
502             handle_TeclaPushed(app,466);
503         case '7'
504             handle_TeclaPushed(app,493.88);
505         case 'u'
506             handle_TeclaPushed(app,523.25);
507         case 'p'
508             clear sound;
509         end
510     end
511
512     end
513 end
514
515 % Component initialization
516 methods (Access = private)
517
518     % Create UIFigure and components
519     function createComponents(app)
520
521         % Create UIFigure and hide until all components are
522         created
523         app.UIFigure = uifigure('Visible', 'off');
524         app.UIFigure.Color = [0.7412 0.5882 0.5216];
525         app.UIFigure.Position = [25 25 1323 857];
526         app.UIFigure.Name = 'MATLAB App';
527
528         % Create PanellADSROnal
529         app.PanellADSROnal = uipanel(app.UIFigure);
530         app.PanellADSROnal.BorderType = 'none';
531         app.PanellADSROnal.Title = 'Panell ADSR Ona 1';
532         app.PanellADSROnal.BackgroundColor = [0.9608 0.7294
533         0.6196];
534         app.PanellADSROnal.Position = [34 637 525 196];
535
536         % Create AttackKnobLabel
537         app.AttackKnobLabel = uilabel(app.PanellADSROnal);
538         app.AttackKnobLabel.HorizontalAlignment = 'center';
539         app.AttackKnobLabel.Position = [50 44 39 22];
540         app.AttackKnobLabel.Text = 'Attack';
541
542         % Create AttackKnob
543         app.AttackKnob = uiknob(app.PanellADSROnal,
544         'continuous');
545         app.AttackKnob.Limits = [0 0.1];
546         app.AttackKnob.MajorTicks = [0 0.1];
547         app.AttackKnob.MajorTickLabels = {'0', '1'};
548         app.AttackKnob.ValueChangedFcn = createCallbackFcn(app,
549         @AttackKnobValueChanged, true);
550         app.AttackKnob.MinorTicks = [];

```

```

547         app.AttackKnob.Position = [50 100 50 50];
548
549         % Create DecayKnobLabel
550         app.DecayKnobLabel = uilabel(app.PanellADSRon1);
551         app.DecayKnobLabel.HorizontalAlignment = 'center';
552         app.DecayKnobLabel.Position = [177 44 40 22];
553         app.DecayKnobLabel.Text = 'Decay';
554
555         % Create DecayKnob
556         app.DecayKnob = uiknob(app.PanellADSRon1,
'continuous');
557         app.DecayKnob.Limits = [0 0.2];
558         app.DecayKnob.MajorTicks = [0 0.2];
559         app.DecayKnob.MajorTickLabels = {'0', '1'};
560         app.DecayKnob.ValueChangedFcn = createCallbackFcn(app,
@DecayKnobValueChanged, true);
561         app.DecayKnob.MinorTicks = [];
562         app.DecayKnob.Position = [178 100 50 50];
563
564         % Create SustainKnobLabel
565         app.SustainKnobLabel = uilabel(app.PanellADSRon1);
566         app.SustainKnobLabel.HorizontalAlignment = 'center';
567         app.SustainKnobLabel.Position = [303 44 46 22];
568         app.SustainKnobLabel.Text = 'Sustain';
569
570         % Create SustainKnob
571         app.SustainKnob = uiknob(app.PanellADSRon1,
'continuous');
572         app.SustainKnob.Limits = [0 0.6];
573         app.SustainKnob.MajorTicks = [0 0.6];
574         app.SustainKnob.MajorTickLabels = {'0', '1'};
575         app.SustainKnob.ValueChangedFcn =
createCallbackFcn(app, @SustainKnobValueChanged, true);
576         app.SustainKnob.MinorTicks = [];
577         app.SustainKnob.Position = [307 100 50 50];
578         app.SustainKnob.Value = 0.3;
579
580         % Create ReleaseKnobLabel
581         app.ReleaseKnobLabel = uilabel(app.PanellADSRon1);
582         app.ReleaseKnobLabel.HorizontalAlignment = 'center';
583         app.ReleaseKnobLabel.Position = [425 40 50 22];
584         app.ReleaseKnobLabel.Text = 'Release';
585
586         % Create ReleaseKnob
587         app.ReleaseKnob = uiknob(app.PanellADSRon1,
'continuous');
588         app.ReleaseKnob.Limits = [0 0.15];
589         app.ReleaseKnob.MajorTicks = [0 0.15];
590         app.ReleaseKnob.MajorTickLabels = {'0', '1'};
591         app.ReleaseKnob.ValueChangedFcn =
createCallbackFcn(app, @ReleaseKnobValueChanged, true);

```

```

592     app.ReleaseKnob.MinorTicks = [];
593     app.ReleaseKnob.Position = [431 96 50 50];
594
595     % Create ControladordeSoPanel
596     app.ControladordeSoPanel = uipanel(app.UIFigure);
597     app.ControladordeSoPanel.BorderType = 'none';
598     app.ControladordeSoPanel.Title = 'Controlador de So';
599     app.ControladordeSoPanel.BackgroundColor = [0.949
    0.6784 0.5608];
600     app.ControladordeSoPanel.Position = [895 275 422 574];
601
602     % Create UIAxes
603     app.UIAxes = uiaxes(app.ControladordeSoPanel);
604     title(app.UIAxes, 'Forma d''Ona Sortida')
605     xlabel(app.UIAxes, '')
606     ylabel(app.UIAxes, '')
607     app.UIAxes.Box = 'on';
608     app.UIAxes.BoxStyle = 'full';
609     app.UIAxes.XTickLabel = '';
610     app.UIAxes.YTickLabel = '';
611     app.UIAxes.BackgroundColor = [0.949 0.6784 0.5608];
612     app.UIAxes.Position = [62 31 300 185];
613
614     % Create AmplitudSliderLabel
615     app.AmplitudSliderLabel =
        uilabel(app.ControladordeSoPanel);
616     app.AmplitudSliderLabel.HorizontalAlignment = 'right';
617     app.AmplitudSliderLabel.Position = [75 528 52 22];
618     app.AmplitudSliderLabel.Text = 'Amplitud';
619
620     % Create AmplitudSlider
621     app.AmplitudSlider =
        uislider(app.ControladordeSoPanel);
622     app.AmplitudSlider.Limits = [0 1];
623     app.AmplitudSlider.MajorTicks = [0 1];
624     app.AmplitudSlider.ValueChangedFcn =
        createCallbackFcn(app, @AmplitudSliderValueChanged, true);
625     app.AmplitudSlider.Position = [148 537 150 3];
626     app.AmplitudSlider.Value = 1;
627
628     % Create FaseSliderLabel
629     app.FaseSliderLabel =
        uilabel(app.ControladordeSoPanel);
630     app.FaseSliderLabel.HorizontalAlignment = 'right';
631     app.FaseSliderLabel.Position = [89 469 32 22];
632     app.FaseSliderLabel.Text = 'Fase';
633
634     % Create FaseSlider
635     app.FaseSlider = uislider(app.ControladordeSoPanel);
636     app.FaseSlider.Limits = [0 6.28];

```



```

637         app.FaseSlider.MajorTicks = [0 6.28];
638         app.FaseSlider.ValueChangedFcn = createCallbackFcn(app,
        @FaseSliderValueChanged, true);
639         app.FaseSlider.Position = [142 478 150 3];
640
641         % Create FormaOna
642         app.FormaOna = uiaxes(app.ControladordeSoPanel);
643         title(app.FormaOna, 'Forma d''Ona Entrada')
644         xlabel(app.FormaOna, '')
645         ylabel(app.FormaOna, '')
646         app.FormaOna.XLim = [0 200];
647         app.FormaOna.YLim = [-1 1];
648         app.FormaOna.Box = 'on';
649         app.FormaOna.BoxStyle = 'full';
650         app.FormaOna.XTickLabel = '';
651         app.FormaOna.YTickLabel = '';
652         app.FormaOna.BackgroundColor = [0.949 0.6784 0.5608];
653         app.FormaOna.Position = [60 224 302 179];
654
655         % Create TeclatPanel
656         app.TeclatPanel = uipanel(app.UIFigure);
657         app.TeclatPanel.BorderType = 'none';
658         app.TeclatPanel.Title = 'Teclat';
659         app.TeclatPanel.BackgroundColor = [0.7412 0.5882
        0.5216];
660         app.TeclatPanel.Position = [33 3 976 262];
661
662         % Create Do3
663         app.Do3 = uibutton(app.TeclatPanel, 'push');
664         app.Do3.ButtonPushedFcn = createCallbackFcn(app,
        @Do3ButtonPushed, true);
665         app.Do3.VerticalAlignment = 'bottom';
666         app.Do3.Position = [1 5 71 240];
667         app.Do3.Text = 'Do3';
668
669         % Create Re3
670         app.Re3 = uibutton(app.TeclatPanel, 'push');
671         app.Re3.ButtonPushedFcn = createCallbackFcn(app,
        @Re3ButtonPushed, true);
672         app.Re3.VerticalAlignment = 'bottom';
673         app.Re3.Position = [71 5 71 240];
674         app.Re3.Text = 'Re3';
675
676         % Create Mi3Button
677         app.Mi3Button = uibutton(app.TeclatPanel, 'push');
678         app.Mi3Button.ButtonPushedFcn = createCallbackFcn(app,
        @Mi3ButtonPushed, true);
679         app.Mi3Button.VerticalAlignment = 'bottom';
680         app.Mi3Button.Position = [141 5 71 240];
681         app.Mi3Button.Text = 'Mi3';

```

```
682
683     % Create Fa3Button
684     app.Fa3Button = uibutton(app.TeclatPanel, 'push');
685     app.Fa3Button.ButtonPushedFcn = createCallbackFcn(app,
    @Fa3ButtonPushed, true);
686     app.Fa3Button.VerticalAlignment = 'bottom';
687     app.Fa3Button.Position = [211 5 71 240];
688     app.Fa3Button.Text = 'Fa3';
689
690     % Create Do4Button
691     app.Do4Button = uibutton(app.TeclatPanel, 'push');
692     app.Do4Button.ButtonPushedFcn = createCallbackFcn(app,
    @Do4ButtonPushed, true);
693     app.Do4Button.VerticalAlignment = 'bottom';
694     app.Do4Button.Position = [490 5 71 240];
695     app.Do4Button.Text = 'Do4';
696
697     % Create Si3Button
698     app.Si3Button = uibutton(app.TeclatPanel, 'push');
699     app.Si3Button.ButtonPushedFcn = createCallbackFcn(app,
    @Si3ButtonPushed, true);
700     app.Si3Button.VerticalAlignment = 'bottom';
701     app.Si3Button.Position = [420 5 71 240];
702     app.Si3Button.Text = 'Si3';
703
704     % Create La3Button
705     app.La3Button = uibutton(app.TeclatPanel, 'push');
706     app.La3Button.ButtonPushedFcn = createCallbackFcn(app,
    @La3ButtonPushed, true);
707     app.La3Button.VerticalAlignment = 'bottom';
708     app.La3Button.Position = [350 5 71 240];
709     app.La3Button.Text = 'La3';
710
711     % Create Sol3Button
712     app.Sol3Button = uibutton(app.TeclatPanel, 'push');
713     app.Sol3Button.ButtonPushedFcn = createCallbackFcn(app,
    @Sol3ButtonPushed, true);
714     app.Sol3Button.VerticalAlignment = 'bottom';
715     app.Sol3Button.Position = [280 5 71 240];
716     app.Sol3Button.Text = 'Sol3';
717
718     % Create Re4Button
719     app.Re4Button = uibutton(app.TeclatPanel, 'push');
720     app.Re4Button.ButtonPushedFcn = createCallbackFcn(app,
    @Re4ButtonPushed, true);
721     app.Re4Button.VerticalAlignment = 'bottom';
722     app.Re4Button.Position = [556 5 71 240];
723     app.Re4Button.Text = 'Re4';
724
725     % Create Mi4Button
726     app.Mi4Button = uibutton(app.TeclatPanel, 'push');
```

```
727         app.Mi4Button.ButtonPushedFcn = createCallbackFcn(app,
@Mi4ButtonPushed, true);
728         app.Mi4Button.VerticalAlignment = 'bottom';
729         app.Mi4Button.Position = [626 5 71 240];
730         app.Mi4Button.Text = 'Mi4';
731
732         % Create Fa4Button
733         app.Fa4Button = uibutton(app.TeclatPanel, 'push');
734         app.Fa4Button.ButtonPushedFcn = createCallbackFcn(app,
@Fa4ButtonPushed, true);
735         app.Fa4Button.VerticalAlignment = 'bottom';
736         app.Fa4Button.Position = [696 5 71 240];
737         app.Fa4Button.Text = 'Fa4';
738
739         % Create Sol4Button
740         app.Sol4Button = uibutton(app.TeclatPanel, 'push');
741         app.Sol4Button.ButtonPushedFcn = createCallbackFcn(app,
@Sol4ButtonPushed, true);
742         app.Sol4Button.VerticalAlignment = 'bottom';
743         app.Sol4Button.Position = [766 5 71 240];
744         app.Sol4Button.Text = 'Sol4';
745
746         % Create La4Button
747         app.La4Button = uibutton(app.TeclatPanel, 'push');
748         app.La4Button.ButtonPushedFcn = createCallbackFcn(app,
@La4ButtonPushed, true);
749         app.La4Button.VerticalAlignment = 'bottom';
750         app.La4Button.Position = [835 5 71 240];
751         app.La4Button.Text = 'La4';
752
753         % Create Si4Button
754         app.Si4Button = uibutton(app.TeclatPanel, 'push');
755         app.Si4Button.ButtonPushedFcn = createCallbackFcn(app,
@Si4ButtonPushed, true);
756         app.Si4Button.VerticalAlignment = 'bottom';
757         app.Si4Button.Position = [905 5 71 240];
758         app.Si4Button.Text = 'Si4';
759
760         % Create Do3_2
761         app.Do3_2 = uibutton(app.TeclatPanel, 'push');
762         app.Do3_2.ButtonPushedFcn = createCallbackFcn(app,
@Do3_2ButtonPushed, true);
763         app.Do3_2.BackgroundColor = [0 0 0];
764         app.Do3_2.Position = [55 99 36 146];
765         app.Do3_2.Text = '';
766
767         % Create Re3_2
768         app.Re3_2 = uibutton(app.TeclatPanel, 'push');
769         app.Re3_2.ButtonPushedFcn = createCallbackFcn(app,
@Re3_2ButtonPushed, true);
770         app.Re3_2.BackgroundColor = [0 0 0];
```

```
771         app.Re3_2.Position = [124 99 36 146];
772         app.Re3_2.Text = '';
773
774         % Create Fa3_2
775         app.Fa3_2 = uibutton(app.TeclatPanel, 'push');
776         app.Fa3_2.ButtonPushedFcn = createCallbackFcn(app,
777 @Fa3_2ButtonPushed, true);
778         app.Fa3_2.BackgroundColor = [0 0 0];
779         app.Fa3_2.Position = [262 99 36 146];
780         app.Fa3_2.Text = '';
781
782         % Create Sol3_2
783         app.Sol3_2 = uibutton(app.TeclatPanel, 'push');
784         app.Sol3_2.ButtonPushedFcn = createCallbackFcn(app,
785 @Sol3_2ButtonPushed, true);
786         app.Sol3_2.BackgroundColor = [0 0 0];
787         app.Sol3_2.Position = [332 99 36 146];
788         app.Sol3_2.Text = '';
789
790         % Create La3_2
791         app.La3_2 = uibutton(app.TeclatPanel, 'push');
792         app.La3_2.ButtonPushedFcn = createCallbackFcn(app,
793 @La3_2ButtonPushed, true);
794         app.La3_2.BackgroundColor = [0 0 0];
795         app.La3_2.Position = [402 99 36 146];
796         app.La3_2.Text = '';
797
798         % Create Do4_2
799         app.Do4_2 = uibutton(app.TeclatPanel, 'push');
800         app.Do4_2.ButtonPushedFcn = createCallbackFcn(app,
801 @Do4_2ButtonPushed, true);
802         app.Do4_2.BackgroundColor = [0 0 0];
803         app.Do4_2.Position = [540 99 36 146];
804         app.Do4_2.Text = '';
805
806         % Create Re4_2
807         app.Re4_2 = uibutton(app.TeclatPanel, 'push');
808         app.Re4_2.ButtonPushedFcn = createCallbackFcn(app,
809 @Re4_2ButtonPushed, true);
810         app.Re4_2.BackgroundColor = [0 0 0];
811         app.Re4_2.Position = [609 99 36 146];
812         app.Re4_2.Text = '';
813
814         % Create Fa4_2
815         app.Fa4_2 = uibutton(app.TeclatPanel, 'push');
816         app.Fa4_2.ButtonPushedFcn = createCallbackFcn(app,
817 @Fa4_2ButtonPushed, true);
818         app.Fa4_2.BackgroundColor = [0 0 0];
819         app.Fa4_2.Position = [747 99 36 146];
820         app.Fa4_2.Text = '';
```

```
816         % Create Sol44_2
817         app.Sol44_2 = uibutton(app.TeclatPanel, 'push');
818         app.Sol44_2.ButtonPushedFcn = createCallbackFcn(app,
            @Sol44_2ButtonPushed, true);
819         app.Sol44_2.BackgroundColor = [0 0 0];
820         app.Sol44_2.Position = [817 99 36 146];
821         app.Sol44_2.Text = '';
822
823         % Create La4_2
824         app.La4_2 = uibutton(app.TeclatPanel, 'push');
825         app.La4_2.ButtonPushedFcn = createCallbackFcn(app,
            @La4_2ButtonPushed, true);
826         app.La4_2.BackgroundColor = [0 0 0];
827         app.La4_2.Position = [887 99 36 146];
828         app.La4_2.Text = '';
829
830         % Create OnaADSR
831         app.OnaADSR = uiaxes(app.UIFigure);
832         title(app.OnaADSR, 'ADSR')
833         xlabel(app.OnaADSR, '')
834         ylabel(app.OnaADSR, '')
835         app.OnaADSR.Box = 'on';
836         app.OnaADSR.XTickLabel = '';
837         app.OnaADSR.YTickLabel = '';
838         app.OnaADSR.BackgroundColor = [0.7412 0.5882 0.5216];
839         app.OnaADSR.Position = [570 637 300 196];
840
841         % Create FreqLabel
842         app.FreqLabel = uilabel(app.UIFigure);
843         app.FreqLabel.Enable = 'off';
844         app.FreqLabel.Position = [2 871 10 10];
845
846         % Create TipusOnaKnobLabel
847         app.TipusOnaKnobLabel = uilabel(app.UIFigure);
848         app.TipusOnaKnobLabel.HorizontalAlignment = 'center';
849         app.TipusOnaKnobLabel.Position = [1103 98 60 23];
850         app.TipusOnaKnobLabel.Text = 'Tipus Ona';
851
852         % Create TipusOnaKnob
853         app.TipusOnaKnob = uiknob(app.UIFigure, 'discrete');
854         app.TipusOnaKnob.Items = {'Sinusoïdal', 'Quadrada',
            'Dent de Serra'};
855         app.TipusOnaKnob.ValueChangedFcn =
            createCallbackFcn(app, @TipusOnaKnobValueChanged, true);
856         app.TipusOnaKnob.Position = [1104 136 59 59];
857         app.TipusOnaKnob.Value = 'Sinusoïdal';
858
859         % Create AmplitudiFasedHarmnicsImpars1357Panel
860         app.AmplitudiFasedHarmnicsImpars1357Panel =
            uipanel(app.UIFigure);
```

```
861         app.AmplitudiFasedHarmnicsImpars1357Panel.Title =
'Amplitud i Fase d''Harmònics Impars (1,3,5,7)';
862
app.AmplitudiFasedHarmnicsImpars1357Panel.BackgroundColor = [0.9608
0.7294 0.6196];
863         app.AmplitudiFasedHarmnicsImpars1357Panel.Position =
[33 268 260 360];
864
865         % Create FaseSlider_1
866         app.FaseSlider_1 =
uislider(app.AmplitudiFasedHarmnicsImpars1357Panel);
867         app.FaseSlider_1.Limits = [0 6.28];
868         app.FaseSlider_1.MajorTicks = [0 6.28];
869         app.FaseSlider_1.Position = [156 282 80 3];
870
871         % Create FaseSlider_3
872         app.FaseSlider_3 =
uislider(app.AmplitudiFasedHarmnicsImpars1357Panel);
873         app.FaseSlider_3.Limits = [0 6.28];
874         app.FaseSlider_3.MajorTicks = [0 6.28];
875         app.FaseSlider_3.Position = [157 202 80 3];
876
877         % Create FaseSlider_5
878         app.FaseSlider_5 =
uislider(app.AmplitudiFasedHarmnicsImpars1357Panel);
879         app.FaseSlider_5.Limits = [0 6.28];
880         app.FaseSlider_5.MajorTicks = [0 6.28];
881         app.FaseSlider_5.Position = [156 128 80 3];
882
883         % Create FaseSlider_7
884         app.FaseSlider_7 =
uislider(app.AmplitudiFasedHarmnicsImpars1357Panel);
885         app.FaseSlider_7.Limits = [0 6.28];
886         app.FaseSlider_7.MajorTicks = [0 6.28];
887         app.FaseSlider_7.Position = [156 56 80 3];
888
889         % Create AmplitudSlider_1
890         app.AmplitudSlider_1 =
uislider(app.AmplitudiFasedHarmnicsImpars1357Panel);
891         app.AmplitudSlider_1.Limits = [0 1];
892         app.AmplitudSlider_1.MajorTicks = [0 1];
893         app.AmplitudSlider_1.Position = [18 282 91 3];
894         app.AmplitudSlider_1.Value = 1;
895
896         % Create AmplitudSlider_3
897         app.AmplitudSlider_3 =
uislider(app.AmplitudiFasedHarmnicsImpars1357Panel);
898         app.AmplitudSlider_3.Limits = [0 1];
899         app.AmplitudSlider_3.MajorTicks = [0 1];
900         app.AmplitudSlider_3.Position = [18 202 91 3];
901         app.AmplitudSlider_3.Value = 1;
```

```
902
903     % Create AmplitudSlider_5
904     app.AmplitudSlider_5 =
        uislider(app.AmplitudiFasedHarmnicsImpars1357Panel);
905     app.AmplitudSlider_5.Limits = [0 1];
906     app.AmplitudSlider_5.MajorTicks = [0 1];
907     app.AmplitudSlider_5.Position = [18 128 91 3];
908     app.AmplitudSlider_5.Value = 1;
909
910     % Create AmplitudSlider_7
911     app.AmplitudSlider_7 =
        uislider(app.AmplitudiFasedHarmnicsImpars1357Panel);
912     app.AmplitudSlider_7.Limits = [0 1];
913     app.AmplitudSlider_7.MajorTicks = [0 1];
914     app.AmplitudSlider_7.Position = [18 56 91 3];
915     app.AmplitudSlider_7.Value = 1;
916
917     % Create TextArea
918     app.TextArea =
        uitextarea(app.AmplitudiFasedHarmnicsImpars1357Panel);
919     app.TextArea.Position = [24 296 80 27];
920     app.TextArea.Value = {'Amplitud H1'};
921
922     % Create TextArea_2
923     app.TextArea_2 =
        uitextarea(app.AmplitudiFasedHarmnicsImpars1357Panel);
924     app.TextArea_2.Position = [24 221 80 27];
925     app.TextArea_2.Value = {'Amplitud H3'};
926
927     % Create TextArea_3
928     app.TextArea_3 =
        uitextarea(app.AmplitudiFasedHarmnicsImpars1357Panel);
929     app.TextArea_3.Position = [23 140 80 27];
930     app.TextArea_3.Value = {'Amplitud H5'};
931
932     % Create TextArea_4
933     app.TextArea_4 =
        uitextarea(app.AmplitudiFasedHarmnicsImpars1357Panel);
934     app.TextArea_4.Position = [24 69 80 27];
935     app.TextArea_4.Value = {'Amplitud H7'};
936
937     % Create TextArea_5
938     app.TextArea_5 =
        uitextarea(app.AmplitudiFasedHarmnicsImpars1357Panel);
939     app.TextArea_5.Position = [156 296 81 27];
940     app.TextArea_5.Value = {'Fase H1'};
941
942     % Create TextArea_6
943     app.TextArea_6 =
        uitextarea(app.AmplitudiFasedHarmnicsImpars1357Panel);
944     app.TextArea_6.Position = [155 221 81 27];
```

```
945         app.TextArea_6.Value = {'Fase H3'};
946
947         % Create TextArea_7
948         app.TextArea_7 =
          uitextarea(app.AmplitudiFasedHarmnicsImpars1357Panel);
949         app.TextArea_7.Position = [155 140 81 27];
950         app.TextArea_7.Value = {'Fase H5'};
951
952         % Create TextArea_8
953         app.TextArea_8 =
          uitextarea(app.AmplitudiFasedHarmnicsImpars1357Panel);
954         app.TextArea_8.Position = [155 69 81 27];
955         app.TextArea_8.Value = {'Fase H7'};
956
957         % Create AmplitudiFasedHarmnicsPars246Panel
958         app.AmplitudiFasedHarmnicsPars246Panel =
          uipanel(app.UIFigure);
959         app.AmplitudiFasedHarmnicsPars246Panel.Title =
          'Amplitud i Fase d''Harmònics Pars (2,4,6)';
960         app.AmplitudiFasedHarmnicsPars246Panel.BackgroundColor
          = [0.9608 0.7294 0.6196];
961         app.AmplitudiFasedHarmnicsPars246Panel.Position = [299
          268 260 360];
962
963         % Create FaseSlider_2
964         app.FaseSlider_2 =
          uislider(app.AmplitudiFasedHarmnicsPars246Panel);
965         app.FaseSlider_2.Limits = [0 6.28];
966         app.FaseSlider_2.MajorTicks = [0 6.28];
967         app.FaseSlider_2.Position = [150 282 80 3];
968
969         % Create FaseSlider_4
970         app.FaseSlider_4 =
          uislider(app.AmplitudiFasedHarmnicsPars246Panel);
971         app.FaseSlider_4.Limits = [0 6.28];
972         app.FaseSlider_4.MajorTicks = [0 6.28];
973         app.FaseSlider_4.Position = [150 202 80 3];
974
975         % Create FaseSlider_6
976         app.FaseSlider_6 =
          uislider(app.AmplitudiFasedHarmnicsPars246Panel);
977         app.FaseSlider_6.Limits = [0 6.28];
978         app.FaseSlider_6.MajorTicks = [0 6.28];
979         app.FaseSlider_6.Position = [150 130 80 3];
980
981         % Create AmplitudSlider_2
982         app.AmplitudSlider_2 =
          uislider(app.AmplitudiFasedHarmnicsPars246Panel);
983         app.AmplitudSlider_2.Limits = [0 1];
984         app.AmplitudSlider_2.MajorTicks = [0 1];
985         app.AmplitudSlider_2.Position = [12 282 91 3];
```



```
986         app.AmplitudSlider_2.Value = 1;
987
988         % Create AmplitudSlider_4
989         app.AmplitudSlider_4 =
990         uislider(app.AmplitudiFasedHarmnicsPars246Panel);
991         app.AmplitudSlider_4.Limits = [0 1];
992         app.AmplitudSlider_4.MajorTicks = [0 1];
993         app.AmplitudSlider_4.Position = [12 202 91 3];
994         app.AmplitudSlider_4.Value = 1;
995
996         % Create AmplitudSlider_6
997         app.AmplitudSlider_6 =
998         uislider(app.AmplitudiFasedHarmnicsPars246Panel);
999         app.AmplitudSlider_6.Limits = [0 1];
1000        app.AmplitudSlider_6.MajorTicks = [0 1];
1001        app.AmplitudSlider_6.Position = [12 130 91 3];
1002        app.AmplitudSlider_6.Value = 1;
1003
1004        % Create TextArea_9
1005        app.TextArea_9 =
1006        uitextarea(app.AmplitudiFasedHarmnicsPars246Panel);
1007        app.TextArea_9.Position = [21 296 80 27];
1008        app.TextArea_9.Value = {'Amplitud H2'};
1009
1010        % Create TextArea_10
1011        app.TextArea_10 =
1012        uitextarea(app.AmplitudiFasedHarmnicsPars246Panel);
1013        app.TextArea_10.Position = [21 221 80 27];
1014        app.TextArea_10.Value = {'Amplitud H4'};
1015
1016        % Create TextArea_11
1017        app.TextArea_11 =
1018        uitextarea(app.AmplitudiFasedHarmnicsPars246Panel);
1019        app.TextArea_11.Position = [20 140 80 27];
1020        app.TextArea_11.Value = {'Amplitud H6'};
1021
1022        % Create TextArea_12
1023        app.TextArea_12 =
1024        uitextarea(app.AmplitudiFasedHarmnicsPars246Panel);
1025        app.TextArea_12.Position = [153 296 81 27];
1026        app.TextArea_12.Value = {'Fase H2'};
1027
1028        % Create TextArea_13
1029        app.TextArea_13 =
1030        uitextarea(app.AmplitudiFasedHarmnicsPars246Panel);
1031        app.TextArea_13.Position = [152 221 81 27];
1032        app.TextArea_13.Value = {'Fase H4'};
1033
1034        % Create TextArea_14
1035        app.TextArea_14 =
1036        uitextarea(app.AmplitudiFasedHarmnicsPars246Panel);
```

```

1029         app.TextArea_14.Position = [152 140 81 27];
1030         app.TextArea_14.Value = {'Fase H6'};
1031
1032         % Create FFT
1033         app.FFT = uiaxes(app.UIFigure);
1034         title(app.FFT, 'FFT Senyal Entrada')
1035         xlabel(app.FFT, '')
1036         ylabel(app.FFT, '')
1037         app.FFT.XLim = [0 200];
1038         app.FFT.YLim = [-1 1];
1039         app.FFT.Box = 'on';
1040         app.FFT.BoxStyle = 'full';
1041         app.FFT.XTickLabel = '';
1042         app.FFT.YTickLabel = '';
1043         app.FFT.BackgroundColor = [0.7412 0.5882 0.5216];
1044         app.FFT.Position = [562 424 315 204];
1045
1046         % Create TeclatSwitchLabel
1047         app.TeclatSwitchLabel = uilabel(app.UIFigure);
1048         app.TeclatSwitchLabel.HorizontalAlignment = 'center';
1049         app.TeclatSwitchLabel.Position = [1043 7 37 22];
1050         app.TeclatSwitchLabel.Text = 'Teclat';
1051
1052         % Create TeclatSwitch
1053         app.TeclatSwitch = uiswitch(app.UIFigure, 'slider');
1054         app.TeclatSwitch.ValueChangedFcn =
1055             createCallbackFcn(app, @TeclatSwitchValueChanged, true);
1056         app.TeclatSwitch.Position = [1038 44 45 20];
1057
1058         % Show the figure after all components are created
1059         app.UIFigure.Visible = 'on';
1060     end
1061
1062     % App creation and deletion
1063     methods (Access = public)
1064
1065         % Construct app
1066         function app = SinteTFG
1067
1068             % Create UIFigure and components
1069             createComponents(app)
1070
1071             % Register the app with App Designer
1072             registerApp(app, app.UIFigure)
1073
1074             % Execute the startup function
1075             runStartupFcn(app, @startupFcn)
1076
1077             if nargin == 0

```

```
1078         clear app
1079     end
1080 end
1081
1082     % Code that executes before app deletion
1083     function delete(app)
1084
1085         % Delete UIFigure when app is deleted
1086         delete(app.UIFigure)
1087     end
1088 end
1089 end
```