

GENERACIÓN DE TONOS DIGITALES ADSR EMPLEANDO UN PROCESADOR DIGITAL DE SEÑALES

Eddie Raúl Muñoz Vara
Sección de Postgrado, Facultad de Ingeniería Eléctrica y Electrónica
E-mail: emunoz@tp.com.pe

RESUMEN

Este artículo desarrolla dos herramientas de síntesis de formas de onda de sonido, a saber; un sintetizador de tono y un generador de envolvente ADSR (Attack, Decay, Sustain and Release) para formar la amplitud del tono. Tales herramientas son codificadas en Matlab y programadas luego en la tarjeta procesadora de señal DSP56811 de Motorola. Empleando la herramienta de síntesis musical de Matlab, previamente instalado en una computadora personal, nosotros podemos variar el período la forma de onda correspondiente a un instrumento, hasta dar con el tono preciso de dicho instrumento. Cabe anotar que las señales musicales así sintetizadas, se puede transmitir, por ejemplo, empleando modems.

ABSTRACT

This paper develops two synthesis tools for sound waveforms, namely, tone synthesizer and a ADSR (Attack, Decay, Sustain and Release) generator to form the tone amplitude. Such tools are coded in Matlab and programmed in a Motorola DSP568. By employing the musical synthesis tool of Matlab, previously installed into a personal computer, we are able to modify the period of a waveform sound corresponding to an instrument. It is worth to mention that musical sounds synthesized in this way can be transmitted by means of modems.

INTRODUCCIÓN

El sonido es un fenómeno físico que percibimos al producirse una perturbación en el medio en el cual estamos; son ondas que en este caso, se propagan por aire, sólidos o líquidos. El cuerpo humano percibe el sonido como un cambio en la presión del aire en el tímpano. De este modo, por ejemplo, si alguien aplaude, ese aplauso desplaza aire que hace vibrar a las sensibles membranas de nuestros oídos. Esa vibración es decodificada por el cerebro como sonido. El sonido es una forma de energía mecánica que se representa por una onda sinusoidal que muestra vibraciones a lo largo del tiempo. Este tipo de onda viene caracterizado principalmente por dos parámetros: amplitud y frecuencia

Nosotros percibimos la amplitud como la intensidad (o volumen) del sonido y la frecuencia como su tono (o altura). La medida de la amplitud de una onda

es importante porque informa de la fuerza, o cantidad de energía, de una onda, la cual se traduce en la intensidad de lo que oímos. Su unidad de medida es el decibel. Un decibel, abreviado como dB, es la unidad de medida de la fuerza de la señal y es útil en la comparación de la intensidad de dos sonidos.

La frecuencia se mide en Hertz (Hz), y corresponde al número de vibraciones por segundo de un determinado sonido. Cualquier forma de onda periódica puede descomponer en múltiples ondas sinusoidales, cuyas relaciones de frecuencia son múltiplos enteros de una frecuencia denominada fundamental. A estas ondas sinusoidales que tienen frecuencias que son 2, 3, 4, ... veces la frecuencia fundamental, se denominan armónicos. Las intensidades relativas y el contenido en armónicos dan a cada onda un sonido único y característico que

se conoce como timbre (o color tonal). Interviniendo sobre estas tres características: intensidad, tono y timbre, no se logra una síntesis fiel del sonido real. Todavía hay que considerar los ruidos inherentes a cualquier sonido (arrastre de los dedos, sibilancia de los labios, impacto de los mazos, etc.) que son las señas de identidad imprescindibles para caracterizarla. Tal caracterización se denomina la envolvente. Los procedimientos de síntesis de sonido intervienen sobre las cuatro características de la envolvente denominada ADSR (Fig.1) por las palabras en inglés Attack, Decay, Sustain y Release (ataque, decaimiento, sostenimiento y relajación) [1, 2].

Si nuestro convertidor (analógico-digital o digital-analógico) toma una muestra cada segundo, las variaciones en el sonido que se produzcan en el intermedio no serán registradas.

El número de lecturas que se hace por unidad de tiempo es la frecuencia o tasa de muestreo (sampling rate); esto es, el número de veces por segundo que se realiza la conversión de una señal para obtener/colocar el valor de su tono.

La generación de tonos ADSR con señales sinusoidales y empleando un DSP (Digital Signal Processor) requiere de un esfuerzo en programación.

AUDIO O SONIDO

El sonido se percibe con un cambio en la presión del aire en el tímpano. Se representa por una onda sinusoidal que muestra vibraciones en el tiempo y se caracteriza por dos parámetros: amplitud y frecuencia. La amplitud es el volumen del sonido y la frecuencia es el tono.

La combinación de diferentes amplitudes y frecuencias en un sonido lo caracteriza como único y a esto se le conoce como timbre.

Con estos tres parámetros no se logra una síntesis fiel del sonido real. Falta la envolvente ADSR del sonido, cuyos componentes se describen a continuación (ver Fig. 1).

El Ataque es la forma de iniciar el sonido, el cual puede comenzar pulsando una nota o aumentando el

volumen gradualmente en el transcurso del tiempo. Decaimiento es la forma en que el sonido varía en volumen hasta la etapa de Sostenimiento.

El Sostenimiento es el volumen (dinámica) que tiene el sonido durante el tiempo que mantengamos la tecla pulsada.

Por último, Relajación es la variación de la dinámica del sonido una vez soltada la nota, la cual puede ser gradual (hasta que se extinga el sonido). Se ha graficado la envolvente ADSR para los instrumentos guitarra $t < 30$ ms. y piano $t < 125$ ms. (ver Fig. 2).

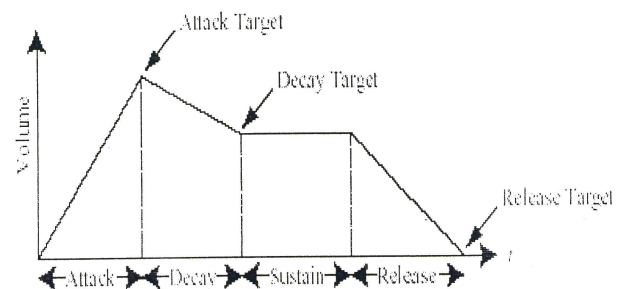


Fig.1 Perfil ADSR de la función envolvente.

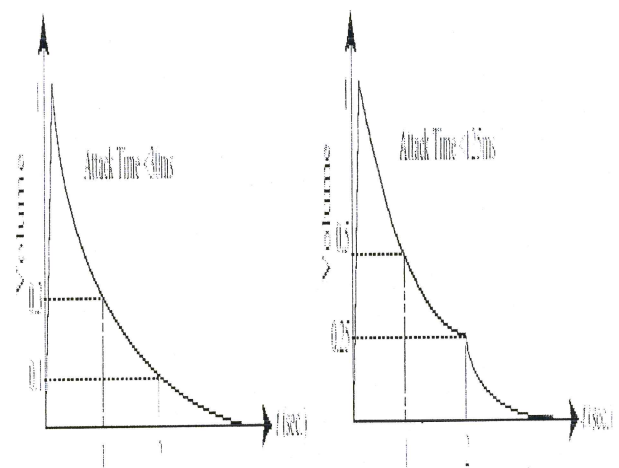


Fig.2 Envolturas ADSR (volumen vs. tiempo).

Para guitarra ($t < 30$ ms. y piano $t < 125$ ms.).

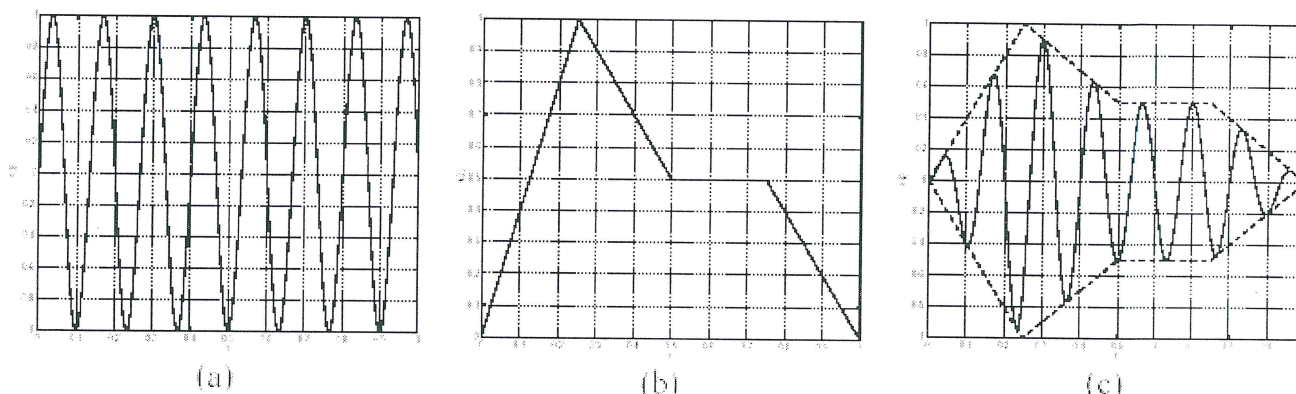


Fig. 3 Generación de la envolvente ADSR, (a) señal sinusoidal (volumen vs.t), (b) envolventeADSR, (c) señal modulada.

IMPLEMENTACIÓN Y MODELO DE LA ENVOLVENTE

Sintetizaremos formas de Ondas compuestas por señales sinusoidales de la forma:

$$x(t) = A \cos(\omega t + \phi) \quad (1)$$

donde $x(t)$ es la señal sinusoidal, A es la amplitud, ω es la frecuencia, t es el tiempo y ϕ es la fase a muestrear y reconstruir para poder escuchar el sonido. Utilizando combinaciones lineales de señales de la forma dada en la Ec. (1), se pueden construir tonos monocromáticos de frecuencia especificada. Generaremos una senoide de 3 segundos de duración, sin aplicar el ADSR, de una nota musical que posee los valores predeterminados para T_s (periodo de muestreo) y f_s (frecuencia de muestreo). Luego en Matlab [3], se puede escribir una función que genere dicha nota durante cada cierto tiempo. Tal función tiene la forma siguiente:

```
function x= singen(f,fs,duración)
%f=frecuencia de la señal
%fs=frecuencia de muestreo
%duración=duración segundos
n=[1:fs*duración]';
x=sin(2*pi*f/fs*n);
```

Ahora escribimos el programa en Matlab para la generación del envolvente (Fig 3). Dado la onda sinusoidal con un volumen característico (amplitud),

la cual es la aproximación para imitar un instrumento real. Los valores de la envolvente correspondientes al piano están dados por un vector entre el vector de la senoide y la envolvente dando la amplitud modulada de la senoide. La envolvente es construida en un segmento (A, D, S y R) en el tiempo.

```
clc
clear
A=input('\n\n Número de Tonos : ');
k=0;
fs=16000;
duración=1;
num=16000*duración; %muestras totales
ga=[0;0.999;0.5;0];
pe=[125;375;125;375];
for a=1:A,
f0(a)=input('\n introd freq Pulso de 250-800);
end;
for x=1:A
nota=singen(f0(x),fs,duración);
aa=adsr2(ga,pe,fs);
yy=aa .* nota;
for b=1:num, %equivalente muestras / nota
Ysin(k+b)=nota(b);
Yadsr(k+b)=yy(b);
end;
k=k+num;
end;
%escuchamos efectos generados %
sound(Ysin,fs,16);
pause(5)
sound(Yadsr,fs,16);
```

```

%guardamos archivos%
archivo1=input('\n archivo sin efecto: ','s');
archivo2=input('\n archivo con efecto: ','s');
wavwrite(Ysin,fs,16,archivo1);
wavwrite(Yadsr,fs,16,archivo2);
%datos hallados para instrumentos%
%% FLAUTA %%
%ga=[0;0.999;0.5;0];
%pe=[125;375;125;375];
%%PIANO%%
%ga=[0;0.999;0.5;0.5];%%cerca
%pe=[10;600;140;250]%%
%ga=[0;0.8;0.6;0.3];***el mejor
%pe=[10;600;140;250];
%ga=[0.5;0.9;0.4;0.2];%* sintetizado
%pe=[50;400;50;500];%suma duración msg

```

A continuación la sentencia en lenguaje assembler [4]:

```

;* DSP56L811 *
;Puertos Considerados
PCC EQU $FFED ; Port C Reg control
IPR EQU $FFFB ; Reg de Interrupc.
SCRRX EQU $FFD4 ; SSI Rx Reg control
SCRTX EQU $FFD3 ; SSI Tx Reg Control
SCR2 EQU $FFD2 ; SSI Reg de Control 2
SSR EQU $FFD1 ; SSI Registro de estado
STSR EQU $FFD1 ; SSI Reg Time Slot
SRX EQU $FFD0 ; SSI Registro Rx
STX EQU $FFD0 ; SSI Registro Tx
;Constantes
PBASE EQU $200 ; inicio memoria
cero EQU $0; dirección mem donde va
T EQU $3E7 ;buffer m01(2-16384)= M-1;2^k>=M
pr0 EQU $800
ENVOL EQU pr0; muestras envolvente
;Tabla de Vectores de Interrupción
org p:$0022 ;definiciones para los saltos
JSR REC ; subrutinas de transmisión y
org p:$0026; definiciones para los saltos
JSR TRA
;Tabla de datos de seno
org x: ENVOL
include 'adsr.asm'
;Programa Principal
org p:$0 ;Inicio del Programa luego reset
RESET jmp START; salto a START

```

```

org p:PBASE ;$200 escribe el programa
START
movep #$613f,x:SCRRX ;config de registro
;PSR=0,WL1=1,WL0=1 (16BPW)
,FRD=1,PSM= 15
movep #$613f,x:SCRTX ;de control Rx,Tx
;PSR=0,WL1=1,WL0=1 (16BPW)
,FRD=1,PSM= 15
movep #$F312,x:SCR2 ;config SCR 2
;RIE=1,TIE=1,RE=1,TE=1,RXD=1,TXD=1,SSIEN=1,FSL=1
bfclr #$0200,SR ;config reg SR I1=0 I0=1
BFSET #$0200,X:IPR;Habil GPIO interr
IPL6=1
movep #$3F00,x:PCC;config PCC PIN 8-13
para SSI
;Llenado de Buffer circular valores 0 inicial
move #pr0,r0 ;Posición de la tabla en buffer
move #T,m01 ;Tamaño de buffer circular
main
jmp main ;bucle infinito
; Rutina de Recepción
REC move x:SRX,x0 ;x0 dato Line In
; clr b
; move x0,b ;coloca en mem dato recibido
;jsr ADSR
rti ; retorno del salto
; Rutina de Transmisión
TRA
move x0,x:STX; tx el dato a line out
lea (r0)+ ; Incremento puntero
rti ;retorno del salto
; Rutina de efecto ADSR
ADSR
clr a
move x:(r0), y0
mpyr y0,x0,a ; a= x(n)*y(n)
nop
move a,X0 ;Transfiero efecto salida
nop
lea (r0)+
asl x0 ;Multiplico por 2
rts
end

```

APLICACIONES

En el gráfico de la Fig.4 se muestran los pulsos sinusoidales ADSR para el instrumento Flauta y para las siguientes frecuencias: 250-440-550-700 Hz.

En la Fig. 5 se grafica la respuesta en frecuencia ADSR de la Flauta (db vs. Hz), donde g_a es la amplitud y P_e es el período, para los valores: $g_a = (0; 0,999; 0,5; 0)$; $P_e = (125; 375; 125; 375)$.

Luego podemos graficar el espectro ADSR de la Flauta (ver Fig. 6).

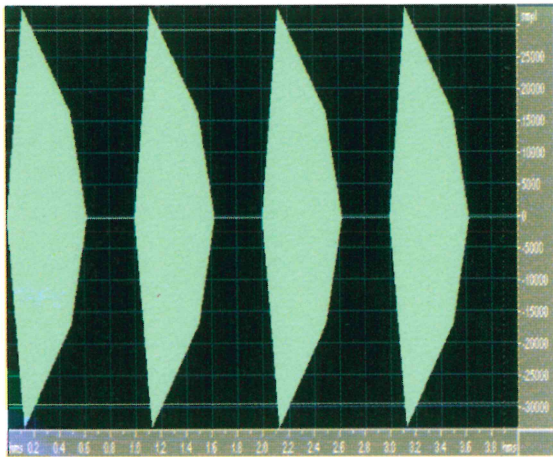


Fig. 4 Señal ADSR de la Flauta (volumen vs tiempo).

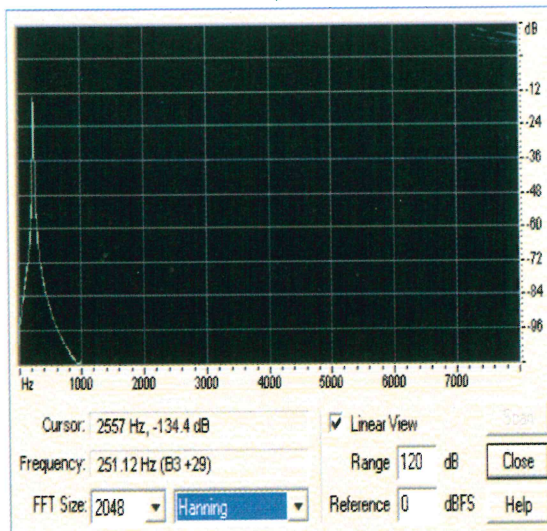


Fig. 5 Respuesta en frecuencia ADSR de la Flauta (db vs. Hz).

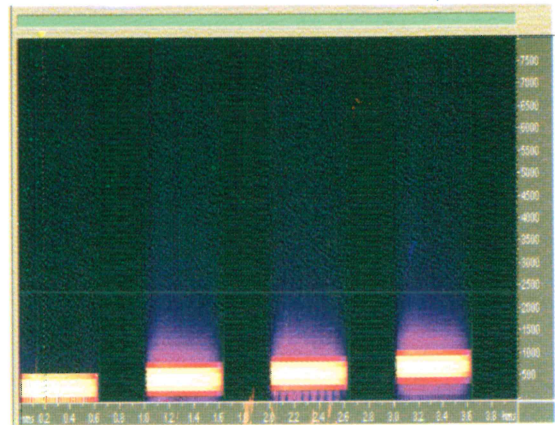


Fig. 6 Espectro ADSR de la Flauta (ms).

CONCLUSIONES

En este trabajo se ha demostrado como se puede generar y procesar digitalmente un tono para un instrumento digital. Se ha escogido el tono para el instrumento flauta, el cual depende de la amplitud de diseño de la envolvente generada. Los tipos de tonos a escuchar dependen también del tiempo que se asigne a cada envolvente. Este varía para cada instrumento.

Específicamente, un tono musical se puede obtener multiplicando la señal original $x(t)$ (ver Ec.1) con la envolvente ADSR (ataque, caída, sostenimiento, relajación) (ver Fig.1).

El familiarizarse con la generación y la reproducción de señales sonoras en el entorno del DSP56811 y con el lenguaje ensamblador y el establecer la conexión entre notas musicales, frecuencias y sinusoides, nos facilita la creación de efectos musicales. Así, podemos modificar la entonación hasta llegar a la ideal y crear otros efectos mediante operaciones matemáticas sobre las señales.

Se ha generado una representación en tiempo vs. frecuencia de las señal de sonido de trabajo. La información que contiene esta representación (su espectrograma) nos puede informar de los cambios en el dominio del tiempo que producen los efectos externos.

En el mercado existe una variedad de equipos con tales características; instrumentos de diferentes variedades con conexión a tarjeta de sonido de alta performance, las cuales pueden ser simuladas con el hardware y software aplicado anteriormente.

AGRADECIMIENTOS

Mi reconocimiento al Dr. Arturo Rojas por la revisión de este artículo y al MSc. Javier Donayre por incentivar el uso del DSP.

REFERENCIAS

1. **J. Chowning**, "The synthesis of complex audio spectra by means of frequency modulation," *Journal of the Audio Engineering Society*, Sept. 1973, Vol. 21, No. 7, pp.526–534.
2. **S. Orphanidis**, "Introduction to Signal Processing, Prentice-Hall, Upper Saddle", NJ, 1996.
3. **Toollbox** "Communicator for use with Matlab". The Math Works Inc. User's guide version 1.3.
4. **User's Manual DSP56L811EVM**, Motorola INC. "Evaluation Module".