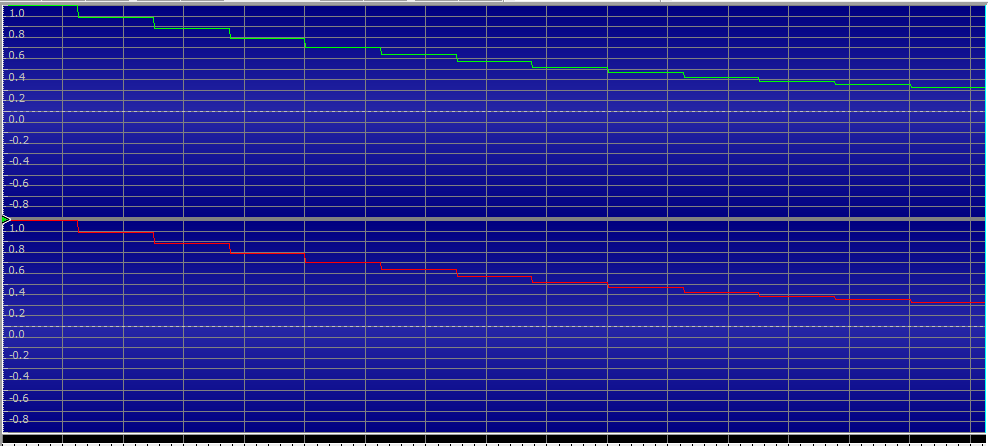
**Convolución**

La convolución es un operador matemático que transforma dos funciones f y g en una tercera, la convolución toma importancia en el análisis de señales ya que es una operación muy común en la mayoría de procesos de señales. Una convolución es un tipo muy general de [media móvil](http://es.wikipedia.org/wiki/Media_m%C3%B3vil)

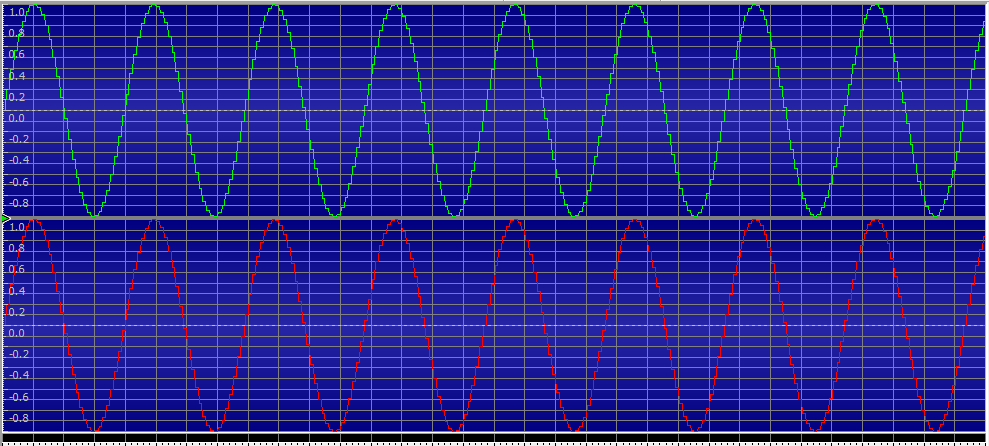
**Practica**

En nuestro caso tenemos la respuesta al impulso de un circuito pasa bajas y para comprobar el funcionamiento del circuito pasa bajas convolucionaremos esta con un seno que nos dará como resultado otro seno con un volumen mas bajo.

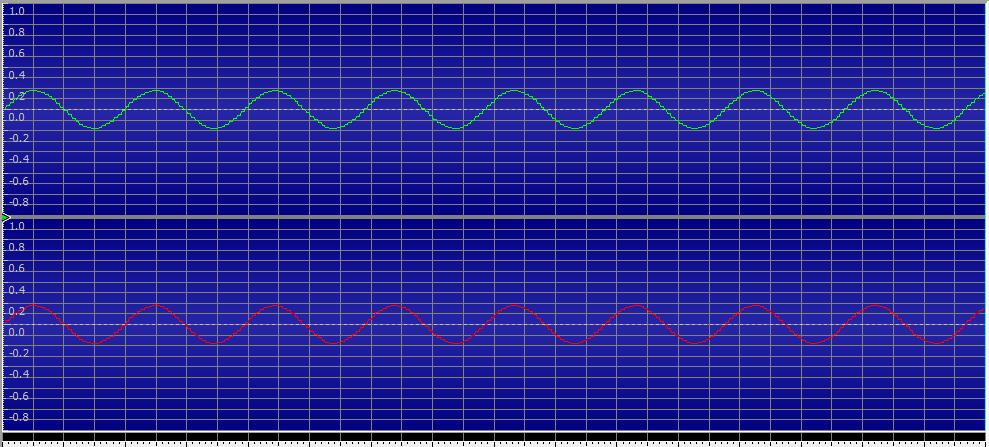
Respuesta al impulso



Sen



Convolución



Lo que pasa es que cuando se convoluciona la respuesta al impulso con el seno es como si pasáramos el seno por un circuito pasa bajas, en este caso la frecuencia pasa por el filtro y al ser una frecuencia mas o menos alta esta pierde amplitud después de pasar por el filtro pasa bajas, si pasáramos un seno con una frecuencia aun mas alta este se aria aun mas pequeño y si aumentamos mas la frecuencia del seno tendería a 0.

**Código:**

#include <stdio.h>

void conv(FILE \*s1, FILE \*s2, FILE \*res);

typedef unsigned short int WORD;

typedef unsigned int DWORD;

typedef unsigned char CHAR;

typedef struct

{

CHAR riff[4];

DWORD tamano;

char wave[8];

DWORD nbytescabeza;

WORD formatopcm;

WORD monoestereo;

DWORD frecnyquist;

DWORD bytesporseg;

WORD bytesporcap;

WORD bitsmuestra;

CHAR data[4];

DWORD ndatos;

} DESCRIPTOR;

main(int argc, char \*\* argv){

FILE \*s1, \*s2, \*res;

char nom[50];

s1 = fopen(argv[1],"rb");

while(s1==NULL){

printf("Error. Archivo no encontrado!!!\n");

printf("Ingrese el nombre del archivo wav1: ");

scanf("%s",nom);

s1 = fopen(nom,"rb");

}

s2 = fopen(argv[2],"rb");

while(s2==NULL){

printf("Error. Archivo no encontrado!!!\n");

printf("Ingrese el nombre del archivo wav2: ");

scanf("%s",nom);

s1 = fopen(nom,"rb");

}

res = fopen("resultado.wav","wb");

conv(s1, s2, res);

}

void conv(FILE \*s1, FILE \*s2, FILE \*res){

DESCRIPTOR cabS1, cabS2, cabRes;

FILE \*hola;

hola = fopen("hola.txt", "w");

int i,j;

unsigned short int aux;

unsigned int lenS1, lenS2, lenRes, bflong1, bflong2;

float \*argRes,\*argS1, \*argS2, escala, max;

fread (&cabS1, sizeof(cabS1), 1, s1);

fread (&cabS2, sizeof(cabS2), 1, s2);

cabRes = cabS1;

lenS1 = cabS1.ndatos/2;

lenS2 = cabS2.ndatos/2;

lenRes = lenS1+lenS2;

cabRes.tamano = (lenRes\*2)+36;

cabRes.ndatos = lenRes\*2;

printf("Longitud: %d\t %d\t %d\n",lenS1,lenS2,lenRes);

if(lenS1>=lenS2){

argS1 = (float \*) malloc((lenS2/2)\*sizeof(float));

argS2 = (float \*) malloc((lenS2/2)\*sizeof(float));

argRes = (float \*) malloc((lenRes/2)\*sizeof(float));

fseek(s1, 44, SEEK\_SET);

fseek(s2, 44, SEEK\_SET);

bflong1 = lenRes/2;

bflong2 = lenS2/2;

for(i=0;i<bflong2;i++){

fread(&aux, sizeof(aux),1,s2);

if(aux<=0x7FFF){

escala = aux;

argS2[i] = (escala/0x7FFF);

}else{

aux = (0xFFFF-aux);

escala = aux;

argS2[i] = -(escala/0x7FFF);

}

//fprintf(hola,"%f\n",argS1[i]);

//fprintf(hola,"%f\n",argS2[i]);

fread(&aux, sizeof(aux),1,s2);

argS1[i]=0;

}

for(i=0;i<bflong1;i++){

if(!feof(s1)){

fread(&aux, sizeof(aux),1,s1);

//fprintf(hola,"%d\n",aux);

if(aux<=0x7FFF){

escala = aux;

argS1[0] = (escala/0x7FFF);

}else{

aux = (0xFFFF-aux);

escala = aux;

argS1[0] = -(escala/0x7FFF);

}

}else{

argS1[0]=0;

}

fread(&aux, sizeof(aux),1,s1);

argRes[i]=0;

for(j=bflong2-1;j>=0;j--){

argRes[i] += argS1[j]\*argS2[j];

//fprintf(hola,"%d\t%f\t%f\t%f",i,argRes[i],argS1[j],argS2[j]);

argS1[j] = j==0?0:argS1[j-1];

}

max = max>argRes[i]?max:argRes[i];

}

}else{

argS1 = (float \*) malloc((lenS1/2)\*sizeof(float));

argS2 = (float \*) malloc((lenS1/2)\*sizeof(float));

argRes = (float \*) malloc((lenRes/2)\*sizeof(float));

fseek(s1, 44, SEEK\_SET);

fseek(s2, 44, SEEK\_SET);

bflong1 = lenRes/2;

bflong2 = lenS1/2;

for(i=0;i<bflong2;i++){

fread(&aux, sizeof(aux),1,s1);

if(aux<=0x7FFF){

escala = aux;

argS2[i] = (escala/0x7FFF);

}else{

aux = (0xFFFF-aux);

escala = aux;

argS2[i] = -(escala/0x7FFF);

}

//fprintf(hola,"%f\n",argS1[i]);

//fprintf(hola,"%f\n",argS2[i]);

fread(&aux, sizeof(aux),1,s1);

argS1[i]=0;

}

for(i=0;i<bflong1;i++){

if(!feof(s1)){

fread(&aux, sizeof(aux),1,s2);

if(aux<=0x7FFF){

escala = aux;

argS1[0] = (escala/0x7FFF);

}else{

aux = (0xFFFF-aux);

escala = aux;

argS1[0] = -(escala/0x7FFF);

}

}else{

argS1[0]=0;

}

fread(&aux, sizeof(aux),1,s1);

argRes[i]=0;

for(j=bflong2-1;j>=0;j--){

argRes[i] += argS1[j]\*argS2[j];

//fprintf(hola,"%d\t%f\t%f\t%f\n",i,argRes[i],argS1[j],argS2[j]);

argS1[j] = j==0?0:argS1[j-1];

}

max = max>argRes[i]?max:argRes[i];

printf("%f\n",max);

}

}

fseek(res, 0, SEEK\_SET);

fwrite(&cabRes,sizeof(DESCRIPTOR),1,res);

for(i=0;i<bflong1;i++){

if(argRes[i]>=0){

aux = (unsigned short int)(argRes[i]\*0x7FFF)/max;

fprintf(hola,"%04x\t%f\n",aux,argRes[i]);

fwrite(&aux,sizeof(short),1,res);

}else{

argRes[i] = (-1)\*argRes[i];

aux = (unsigned short int)(argRes[i]\*0x7FFF)/max;

aux = (0xFFFF-aux);

fwrite(&aux,sizeof(short),1,res);

}

aux=0x0000;

fwrite(&aux,sizeof(short),1,res);

}

fclose(hola);

fclose(s1);

fclose(s2);

fclose(res);

}