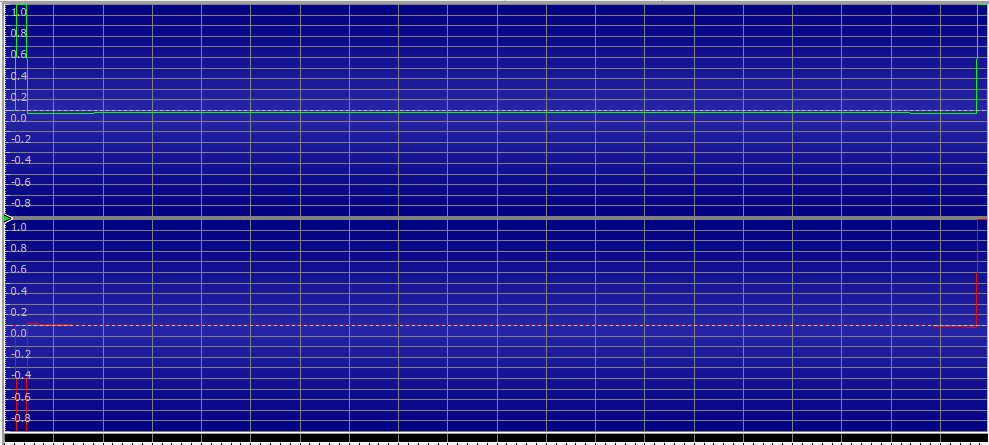
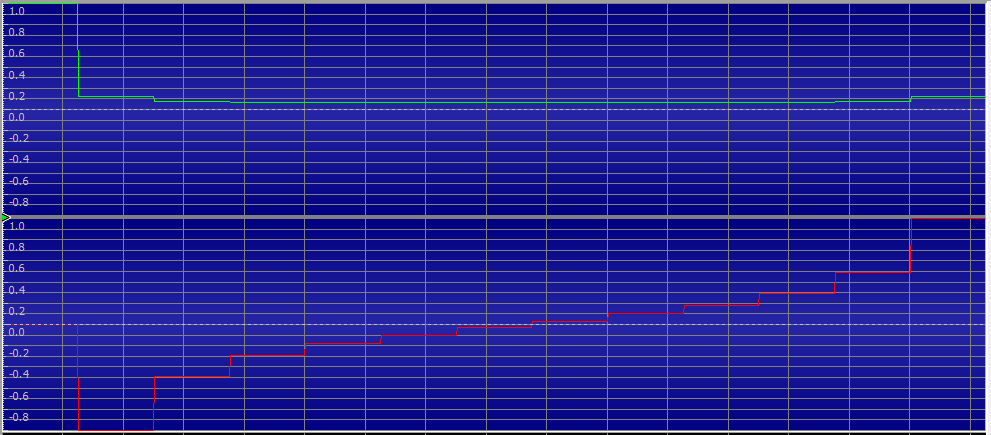
**Transformada de Fourier**

La transformada de Fourier , denominada así por Joseph Fourier, es una transformación matemática empleada para transformar señales entre el dominio del tiempo (o espacial) y el dominio de la frecuencia, que tiene muchas aplicaciones en la física y la ingeniería. Es reversible, siendo capaz de transformaciones de cualquiera de los dominios al otro. El propio término se refiere tanto a la operación de transformación como a la función que produce.

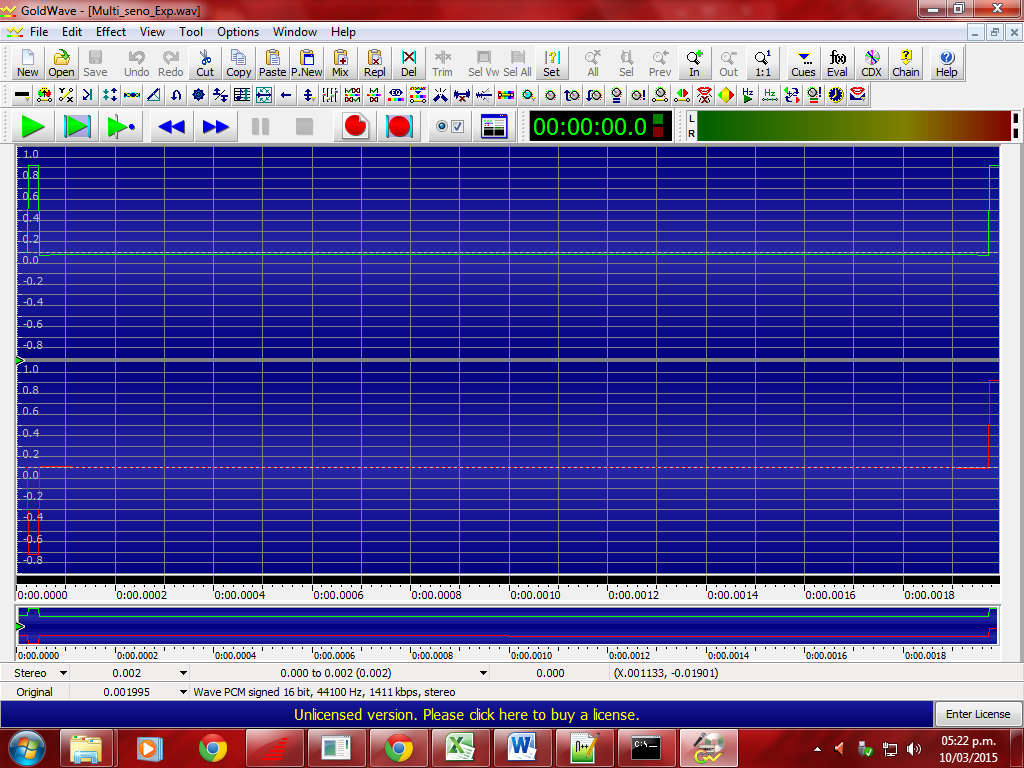
Veremos el proceso de hacer pasar una señal por un filtro pasa bajas pero esta vez la veremos desde el dominio de la frecuencia.

Primero transformaremos un seno y la respuesta al impulso

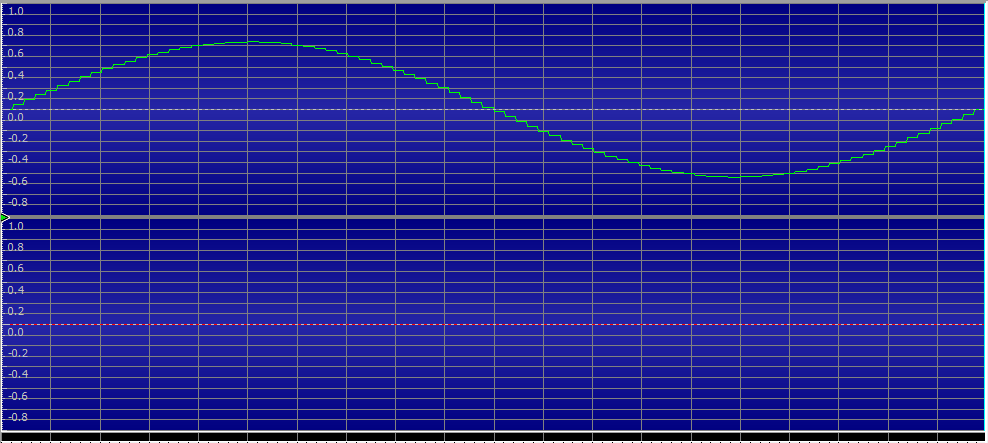




Estas señales cuando se convolucionan en tiempo lo que pasa en la frecuencia es que se multiplican, entonces multiplicaremos estas dos señales



Al hacerle la transformada inversa a esto deberíamos de poder ver el mismo resultado que en la convolucion entonces aremos transformada inversa



**Código**

**Transformada**

#include <stdio.h>

#include <math.h>

#include <time.h>

void tran(FILE \*s1, FILE \*res, int opc);

double mag(double x, double y);

double fase(double x, double y);

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, \*res;

int opc;

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");

}

printf("Ingrese un opcion del 1-3: ");

scanf("%d",&opc);

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

tran(s1, res, opc);

}

void tran(FILE \*s1, FILE \*res, int opc){

DESCRIPTOR cabS1, cabRes;

FILE \*hola;

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

int i,j;

unsigned short int aux;

unsigned int lenS1, bflong1, bflong2;

double \*argRes,\*argS1, escala, ctc;

double maxD = 0, maxI = 0;

double a,b;

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

cabRes = cabS1;

lenS1 = cabS1.ndatos/2;

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

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

argRes = (double \*) malloc((lenS1)\*sizeof(double));

fseek(s1, 44, SEEK\_SET);

bflong1 = lenS1/2;

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

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

if(aux<=0x7FFF){

escala = aux;

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

}else{

aux = (0xFFFF-aux);

escala = aux;

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

}

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

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

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

}

a = 0;

b = 0;

clock\_t start = clock();

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

ctc = (((-2\*M\_PI)\*i)/bflong1);

a = 0;

b = 0;

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

a += argS1[j]\*cos(ctc\*j);

b += argS1[j]\*sin(ctc\*j);

}

//fprintf(hola,"%f\t %f\n",a,b);

switch(opc){

case 1:

a = mag(a,b);

b = argS1[i];

maxD = maxD>=fabs(a)?maxD:fabs(a);

maxI = maxI>=fabs(b)?maxI:fabs(b);

break;

case 2:

escala = mag(a,b);

b = fase(a,b);

a = escala;

maxD = maxD>=fabs(a)?maxD:fabs(a);

maxI = maxI>=fabs(b)?maxI:fabs(b);

break;

default:

maxD = maxD>=fabs(a)?maxD:fabs(a);

maxI = maxI>=fabs(b)?maxI:fabs(b);

break;

}

argRes[i\*2] = a;

argRes[(i\*2)+1] = b;

}

printf("Tiempo transcurrido de la tranformada: %f", ((double)clock() - start));

fseek(res, 0, SEEK\_SET);

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

for(i=0;i<(bflong1\*2);i++){

if(argRes[i]>=0){

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

//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)/maxD);

aux = (0xFFFF-aux);

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

}

if(argRes[i+1]>=0){

fprintf(hola,"%f\t",argRes[i+1]);

aux = (unsigned short int)((argRes[i+1]\*0x7FFF)/maxI);

fprintf(hola,"%04x\n",aux);

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

}else{

fprintf(hola,"%f\t",argRes[i+1]);

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

aux = (unsigned short int)((argRes[i+1]\*0x7FFF)/maxI);

aux = (0xFFFF-aux);

fprintf(hola,"%04x\n",aux);

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

}

i++;

aux=0x0000;

}

fclose(hola);

fclose(s1);

fclose(res);

}

double mag(double x, double y){

return sqrt(pow(x,2)+pow(y,2));

}

double fase(double x, double y){

if(y!=0){

return atan(x/y);

}else{

return 0;

}

}

**Transformada inversa**

#include <stdio.h>

#include <math.h>

void tran(FILE \*s1, FILE \*res);

double mag(double x, double y);

double fase(double x, double y);

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, \*res;

int opc;

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");

}

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

tran(s1, res);

}

void tran(FILE \*s1, FILE \*res){

DESCRIPTOR cabS1, cabRes;

FILE \*hola;

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

int i,j;

unsigned short int aux;

unsigned int lenS1, bflong1, bflong2;

double \*argRes,\*\*argS1, escala, ctc;

double maxD = 0, maxI = 0;

double a,b;

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

cabRes = cabS1;

lenS1 = cabS1.ndatos/2;

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

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

argRes = (double \*) malloc((lenS1)\*sizeof(double));

fseek(s1, 44, SEEK\_SET);

bflong1 = lenS1/2;

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

argS1[i] = (double \*) malloc(2\*sizeof(double));

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

if(aux<=0x7FFF){

escala = aux;

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

}else{

aux = (0xFFFF-aux);

escala = aux;

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

}

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

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

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

if(aux<=0x7FFF){

escala = aux;

argS1[i][1] = (escala/0x7FFF);

}else{

aux = (0xFFFF-aux);

escala = aux;

argS1[i][1] = -(escala/0x7FFF);

}

}

a = 0;

b = 0;

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

ctc = (((2\*M\_PI)\*i)/bflong1);

a = 0;

b = 0;

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

a += argS1[j][0]\*cos(ctc\*j);

b += argS1[j][1]\*sin(ctc\*j);

}

a = mag(a,b);

b = a;

maxD = maxD>=fabs(a)?maxD:fabs(a);

maxI = maxI>=fabs(b)?maxI:fabs(b);

argRes[i\*2] = a;

argRes[(i\*2)+1] = b;

}

printf("MaxD: %f\t MaxI: %f\n",maxD, maxI);

fseek(res, 0, SEEK\_SET);

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

for(i=0;i<(bflong1\*2);i++){

if(argRes[i]>=0){

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

//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)/maxD);

aux = (0xFFFF-aux);

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

}

if(argRes[i+1]>=0){

fprintf(hola,"%f\t",argRes[i+1]);

aux = (unsigned short int)((argRes[i+1]\*0x7FFF)/maxI);

fprintf(hola,"%04x\n",aux);

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

}else{

fprintf(hola,"%f\t",argRes[i+1]);

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

aux = (unsigned short int)((argRes[i+1]\*0x7FFF)/maxI);

aux = (0xFFFF-aux);

fprintf(hola,"%04x\n",aux);

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

}

i++;

aux=0x0000;

}

fclose(hola);

fclose(s1);

fclose(res);

}

double mag(double x, double y){

return sqrt(pow(x,2)+pow(y,2));

}

double fase(double x, double y){

if(y!=0){

return atan(x/y);

}else{

return 0;

}

}