DEPARTAMENTO DE ELECTRÓNICA Y AUTOMÁTICA

FACULTAD DE INGENIERÍA – UNIVERSIDAD NACIONAL DE SAN JUAN

Informe de Laboratorio N°3

Correlación y muestreo

Asignatura: Procesamiento Digital de Señales

Ingeniería Electrónica

Autor:

Avila, Juan Agustin – Registro 26076

1º Semestre

Año 2020

1 Introducción.

1.1 Correlación

La operación de correlación permite obtener información sobre el grado de semejanza que presentan dos señales, al ser comparadas entre sí. Una aplicación es la recuperación de señales periódicas inmersas en ruido. Otra aplicación común se encuentra en los radares y sonares, los cuales tras emitir una secuencia de datos conocida, compara la misma con las señales que recibe para determinar la presencia o no de un objeto, y poder también estimar la distancia a la que se encuentra.

El fundamento estadístico sobre el que se sustenta estas aplicaciones se basa en que la señal de interés y la señal de ruido que la contamina, se encuentran no correlacionadas. Recordando que la correlación entre dos variables aleatorias mide su parecido, como en este caso no se parecen la correlación cruzada tendería a ser nula.

1.1.1 Aplicación a la medida de distancias

Si consideramos que la señal x(t) recibida por el radar está conformada por una versión retardada de la señal emitida más el ruido que se adiciona durante el tránsito de la señal, es razonable pensar que el máximo de parecido entre las señales (variables aleatorias) s(t) y x(t) se obtendrá justo para el tiempo de retraso.

Teniendo en cuenta los fundamentos anteriores, una forma de medir la distancia entre el radar y un objeto es transmitir un pequeño pulso electromagnético y medir el tiempo que tarda el eco en volver. La distancia será la mitad del tiempo de transito multiplicado por la velocidad del pulso (300000km/s): d=(c x t)/2, siendo d la distancia estimada, c la velocidad de la luz y t el tiempo de transito.

1.2 Muestreo.

La frecuencia de muestreo debe exceder el doble de la mayor frecuencia de la señal para evitar el aliasing. Si esto no es así, una señal de frecuencia f0 mostrara como la menor frecuencia (aliasing) fa = f0 – NS donde N es un entero que coloca fa en el periodo principal -0,5S < f < 0,5S.

El comando alias permite evaluar si la relación entre una frecuencia analógica determinada y la frecuencia de muestreo elegida, cumplen con el criterio de Nyquist o no. Empleando como variables de entrada la frecuencia analógica a evaluar (f0) y la frecuencia de muestreo elegida (S), alias devolverá dos valores como resultado: fa y fd. Si -0.5S< f0 < 0,5S, entonces fa = f0 y fd = fa/S. Caso contrario fa < f0, siendo la frecuencia de aliasing corregida. La sintaxis será: >>[fa fd] = alias(f0,S);

2 Actividades.

2.1 Simulación de radar.

2.1.1 Implemente el código que permita la generación de las siguientes señales y sus autocorrelaciones. Graficar las señales en un intervalo temporal de 0 a 5 segundos, y las autocorrelaciones en un intervalo entre -5 y 5 (eje de retardo temporal).

$$x_1(t) = 6\sin 2\pi t \qquad x_2(t) = 3\cos 2\pi 6t$$

$$x_3(t) = 5 \qquad x_4(t) = 10$$

$$x_5(t) = rect\left[\frac{(t-2,5)}{3}\right] \qquad x_6(t) = rect\left[\frac{(t-2,5)}{0,5}\right]$$

$$x_7(t) = 3\,e^{-5t} \qquad x_8(t) = random$$

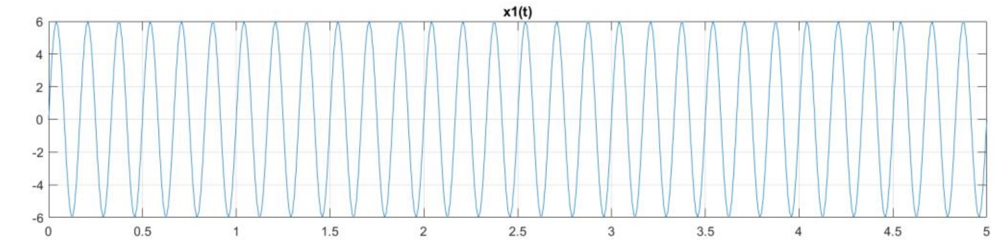
Para la graficacion de las funciones y sus respectivas autocorrelaciones, se utilizo el siguiente código de matlab:

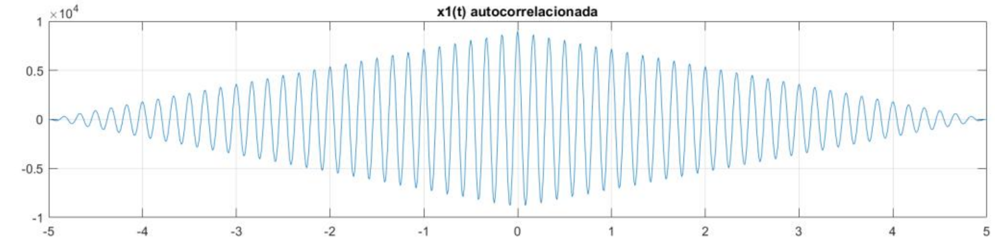
```
%% punto 1
%declaraciones temporales
F=100;
t=0:1/F:5;
tcorr=-5:1/F:5;
%declaracion de funciones
x1=6*sin(2*pi*6*t);
x2=3*cos(2*pi*6*t);
x3=5*ones(1,length(t));
x4=10*ones(1,length(t));
x5=urect((t-2.5)/3);
x6=urect((t-2.5)/.5);
x7=3*exp(-5*t);
x8=randn(1,length(t));
%% 1)a: obtencion y graficacion de correlaciones
xx1=grafcorr(x1,t,tcorr,'x1(t)');
xx2=grafcorr(x2,t,tcorr,'x2(t)');
xx3=grafcorr(x3,t,tcorr,'x3(t)');
xx4=grafcorr(x4,t,tcorr,'x4(t)');
xx5=grafcorr(x5,t,tcorr,'x5(t)');
xx6=grafcorr(x6,t,tcorr,'x6(t)');
xx7=grafcorr(x7,t,tcorr,'x7(t)');
xx8=grafcorr(x8,t,tcorr,'x8(t)');
```

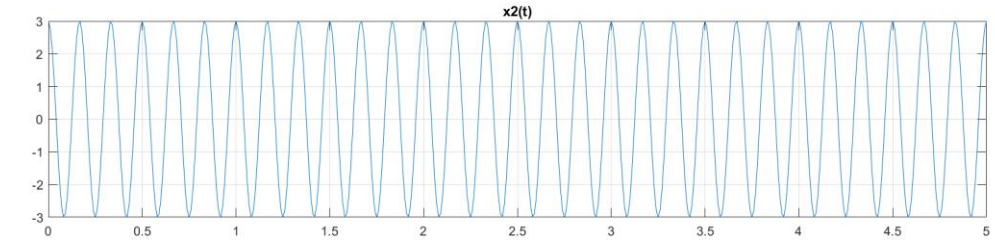
Siendo ‘grafcorr’ la siguiente función: (se utilizó una función para reutilizar código)

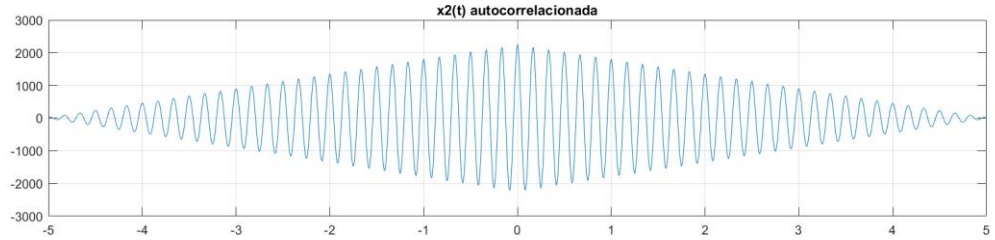
```
%% funciones auxiliares:
function xx=grafcorr(x,t,tcorr,nombre); %obtiene la correlacion y
grafica ambas señales
    xx=xcorr(x,x);
    figure('name', nombre+" autocorrelacionada");
    title("grafica de "+nombre+" y su autocorrelacion");
    subplot(2,1,1);
    plot(t,x);
    hold on;grid on;
    title(nombre);
    subplot(2,1,2);
    plot(tcorr,xx);
    hold on;grid on;
    title(nombre+" autocorrelacionada");
end
```

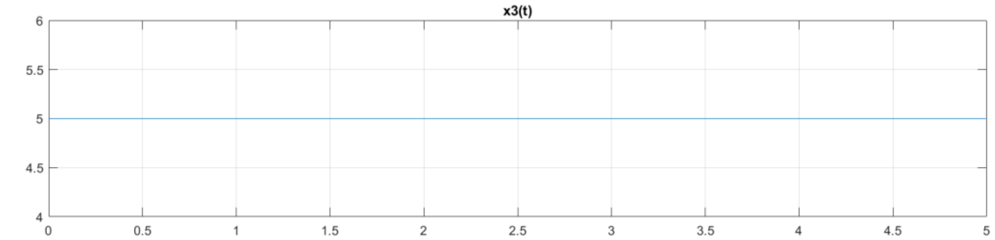
Las gráficas resultantes son las siguientes:

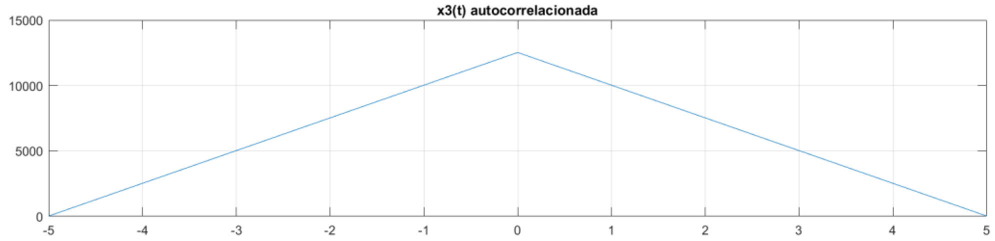

x1(t)

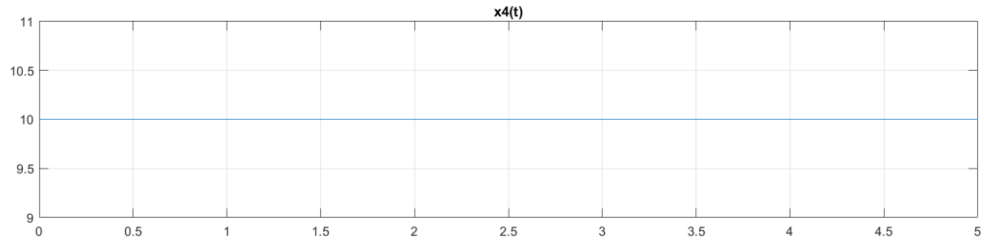

x1(t) autocorrelacionada

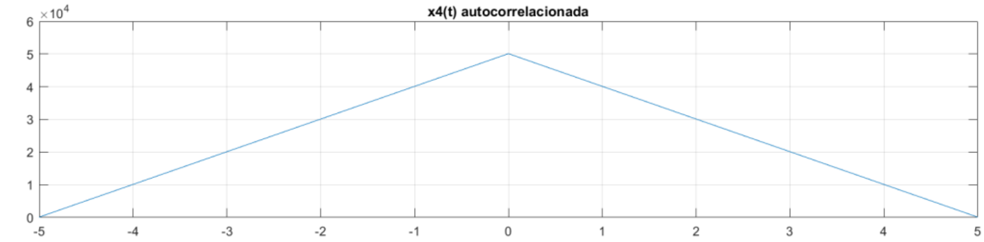

x2(t)

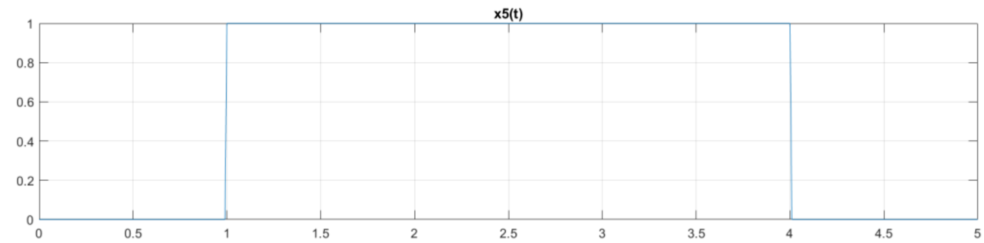

x2(t) autocorrelacionada

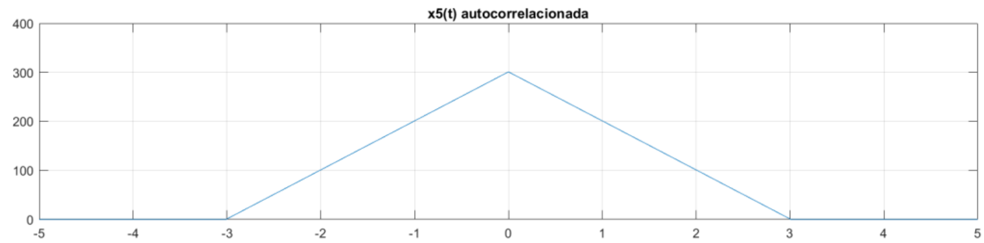

x3(t)

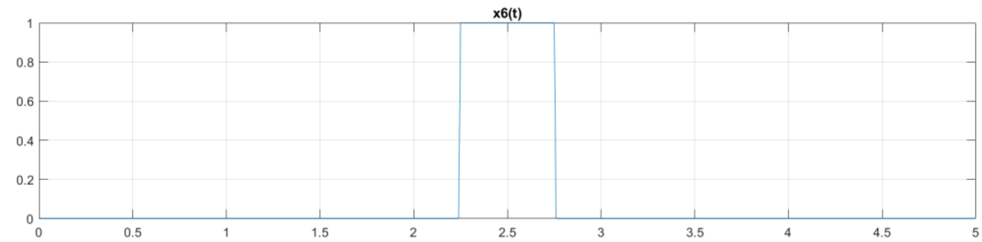

x3(t) autocorrelacionada

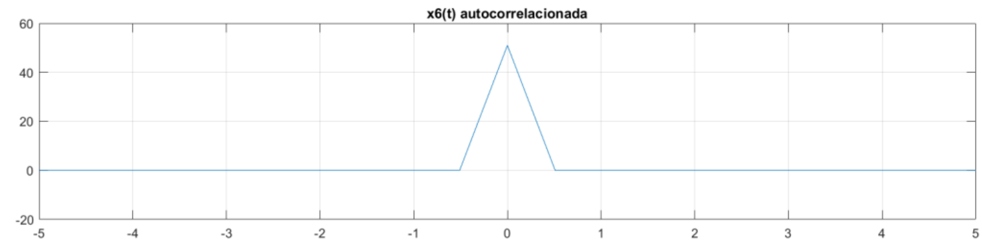

x4(t)

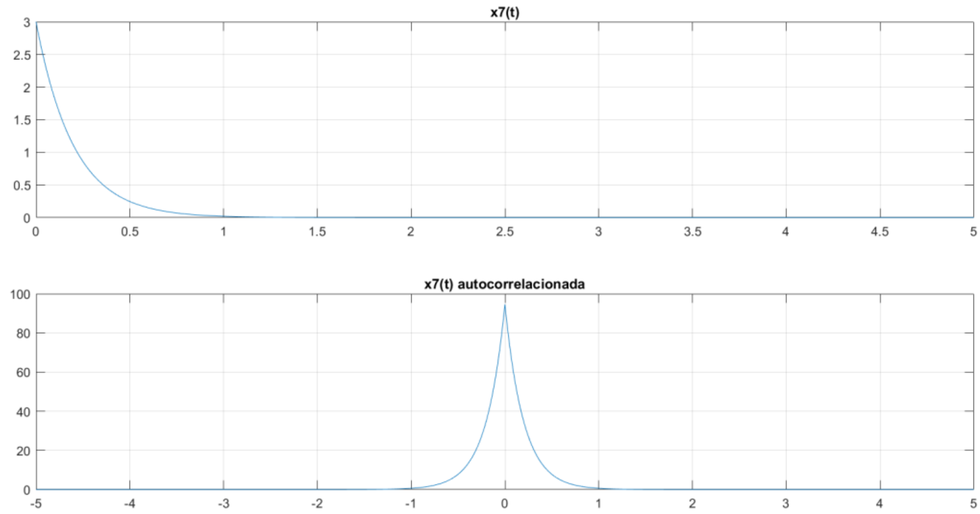

x4(t) autocorrelacionada

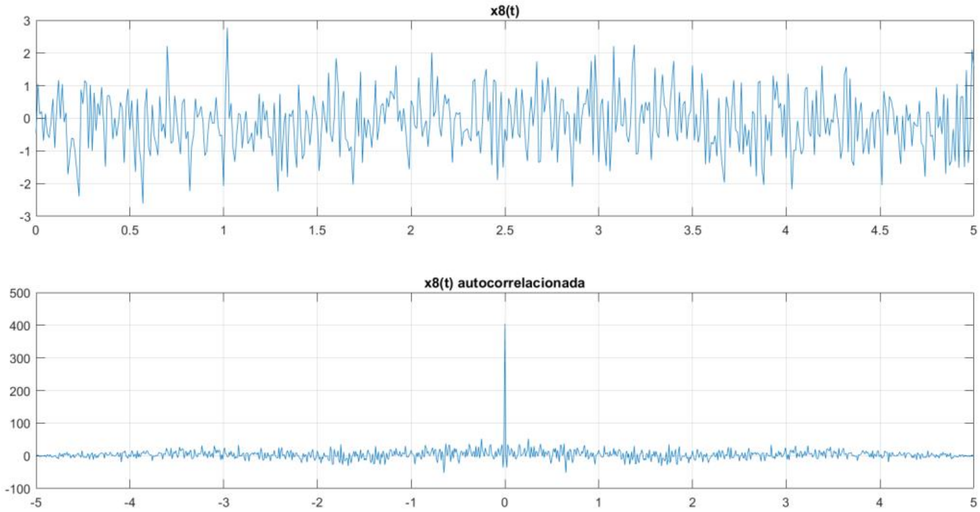

x5(t)


x5(t) autocorrelacionada


x6(t)


x6(t) autocorrelacionada





2.1.2 Realizar un breve análisis sobre la relación entre la naturaleza de cada señal y la autocorrelación obtenida en cada caso.

En las señales x_1 y x_2 se observan picos periódicos en la señal autocorrelacionada, siendo el mayor en t=0. Tambien se observa que la amplitud en x_1 (función con amplitud 6) es mucho mayor que en x_2(función con amplitud 3).

En el caso de las señales iguales a una constante (x_3 y x_4) se observa que su autocorrelación tiene una pendiente positiva constante para valores de desplazamiento temporal negativos, con un pico en t=0 y luego una pendiente negativa constante. Nuevamente se observa que ante una amplitud mayor, la autocorrelación tiene un valor máximo mayor, aunque ese aumento no es lineal, ya que cuando se duplica el valor de la función original, el valor de la autocorrelación aproximadamente se cuadriplica.

En el caso de x_5 y x_6 lo único que varía es el ancho del pulso, variando de la misma manera el ancho de la autocorrelación y su amplitud.

Para X_7, se observa que la autocorrelación genera un pico bastante marcado para t=0, con una varianza bastante baja.

En el caso de x_8, la autocorrelación genera un impulso de gran amplitud en t=0, y valores en comparación muy bajos para otros valores.

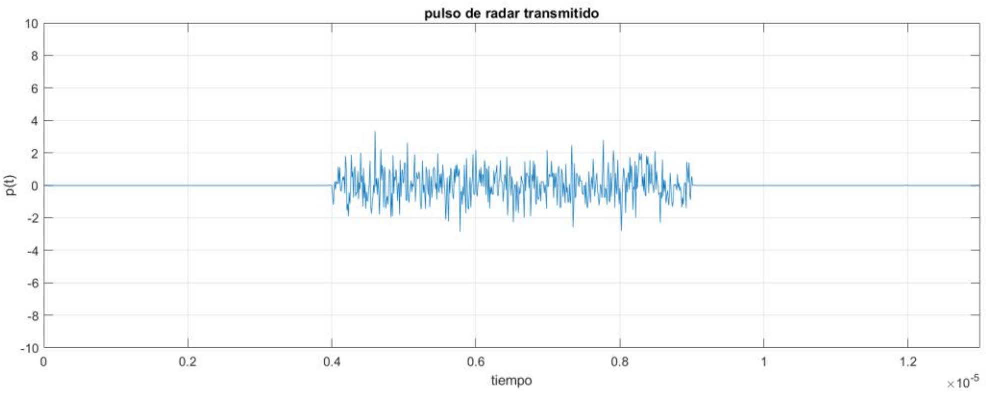
2.1.3 Luego de analizar las gráficas obtenidas, determine qué tipo de señal resultaría la más adecuada para ser usada como pulso de emisión de un radar. Tener en cuenta que es necesario que el tipo de señal elegida permita una detección precisa y con mínima dispersión. Justificar la elección.

Analizando la respuesta de todas las señales, la más adecuada seria la señal aleatoria, ya que la respuesta de su autocorrelación es mínima para todos los valores distintos de cero, generando un pico de gran amplitud en t=0. Esto asegura una mínima dispersión, y al ser tan pronunciado el pico (y con un ancho mínimo) asegura una detección precisa. Respecto a las otras señales, las señales periódicas tienen picos periódicos por lo cual sería prácticamente imposible detectar el máximo. Las funciones que son una constante no tendrían sentido ya que no se podría tomar ninguna referencia. Los pulsos cuadrados podrían ser utilizados pero tienen una gran dispersión respecto a su pico. Finalmente, la función exponencial da un pico bastante marcado con una baja dispersión, pero al ser solo un pico temporal sería difícil separarlo del ruido satisfactoriamente.

2.1.4 Implementación de la simulación de un radar.

Trabajando con el siguiente código como guía, genere un pulso de emisión con la señal elegida. La frecuencia del pulso emitido debe ser de 10MHz. Envíe luego por mail. Como respuesta al pulso emitido, recibirá un mail con el pulso reflejado contaminado (atenuado y retardado). Empleando el pulso emitido y el recibido en respuesta, y haciendo uso de la herramienta de correlación, determine el retardo en tiempo y la distancia al objetivo, tomando como guía lo presentado en la introducción teórica.

```
%% 1.e
F0=1e8;
T0=1/F0;

%%%generacion de señal
duracion=500; comienzo=400; resto=400;
N=comienzo+duracion+resto;
signal=randn(1,duracion);
p=[zeros(1,comienzo),signal,zeros(1,resto)];
figure();
plot((1:N)*T0,p);axis([0 N*T0 -10 10]);
xlabel('tiempo');ylabel('p(t)');
title('pulso de radar transmitido');
```



Obtener para cada caso las gráficas que muestran las distintas señales involucradas en el problema.

```
load('RespuestaRadarAgustinAvila.mat');
load('PulsoRadarAgustinAvila.mat');
%x=[zeros(1,10000),p,zeros(1,resto+24700)]; %prueba un retraso de 0.1
ms
ret=xcorr(x,p);                   % correlación entre x y p
tx=0:T0:T0*(length(x)-1);         % base temporal para graficar la
respuesta
```

```
tret=-T0*(length(x)-1):T0:T0*(length(x)-1);
%como p se rellena con ceros, la longitud total de la correlacion
%va desde -tx(maximo adelanto) hasta tx(maximo atraso)
[x,i]=max(ret);                    % valor y posicion del maximo de
correlacion
retardo=tret(i)*1000               %valor del retardo en ms
d=((3e5)*(retardo/1000))/2         %c en km, retardo en seg.
```

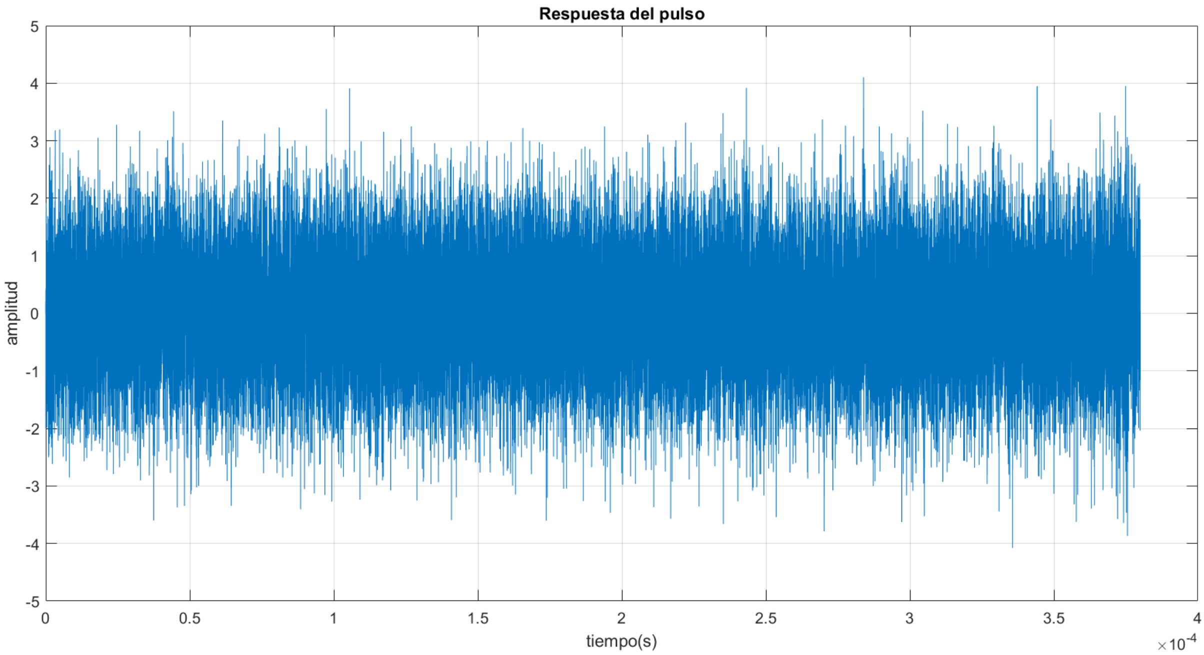
Las respuestas obtenidas son:

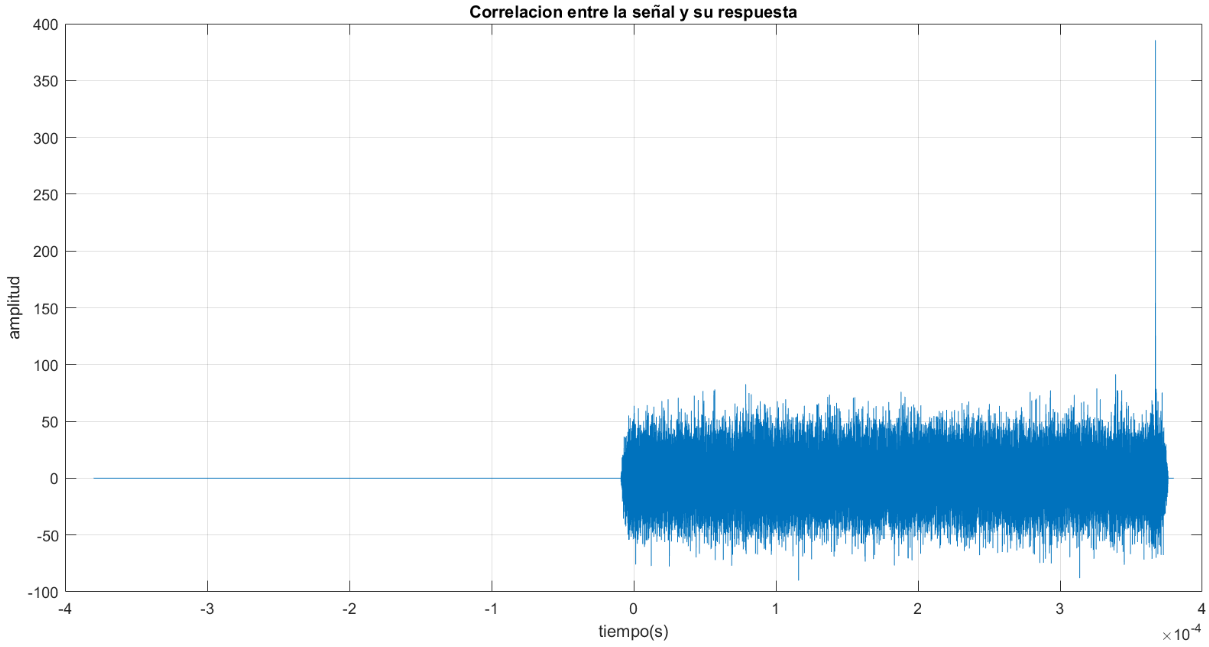
retardo = 0.3670

d = 55.0500

Por lo tanto, el retardo aproximado es de 0.367 ms y la distancia resultante es de aproximadamente 55km.

```
figure();
plot(tx,x);grid;title('Respuesta del pulso');
xlabel('tiempo(s)');ylabel('amplitud');
figure();
plot(tret,ret);grid;title('Correlacion entre la señal y su
respuesta');
xlabel('tiempo(s)');ylabel('amplitud');
```





2.2 Sea x(t) = cos (90πt) + cos(150πt). Obtener muestras de x(t) sobre el intervalo [0, 0.1]s a una tasa de muestreo de S1=200Hz, S2=100Hz, S3=50Hz.

Use el siguiente fragmento de código como una guía (los datos del mismo no necesariamente deben coincidir con los datos del ejercicio dado):

```
>> t=0:0.1/200:0.1;                                   % eje temporal par x(t) y xr(t)
>> x = cos(200*pi*t)+cos(400*pi*t);                   % señal CT
>> S=500;tn1=0:1/S:0.1;                               % eje la señal digital (S=500 Hz)
>> xn1=cos(200*pi*tn1)+cos(400*pi*tn1);               % señal DT
>>[fa1 fd1] = alias(100,S); [fa2 fd2]= alias(200,S);  % frecuencia recuperada
>> xr1=cos(2*pi*fa1*t)+cos(2*pi*fa2*t);               % señal recuperada
>> subplot(221),plot(t,x,t,xr1)                       % comienzo de la gráfica
>> hold on, dtplot(tn1,xn1), hold off
```

Para cada tasa de muestreo S, encontrar una expresión para la señal muestrada x[n].

Calcular f_1 y f_2 en la señal recobrada $x_r(t) = \cos(2\pi f_1 t) + \cos(2\pi f_2 t)$ (use el comando alias) para cada S. En tres pantallas diferentes (una por cada tasa de muestreo), graficar en tres subplot por pantalla, superponiendo x(t) y $x_r(t)$ en función del tiempo (usar plot con t=0:0.1/200:0.1) y x[n] en función de nt_s (usar dplot o stem con tn=0:1/S:0.1). Si consideramos a $x(t) = x_1(t) + x_2(t)$ y $x_r(t) = x_{r1}(t) + x_{r2}(t)$, graficar en los primeros subplot las dos componentes por separado, y en el tercero la señal completa ¿Coinciden los valores de x[n] con los de $x_r(t)$ para cada tasa de muestreo? Teóricamente ¿Deberían coincidir? explicar.

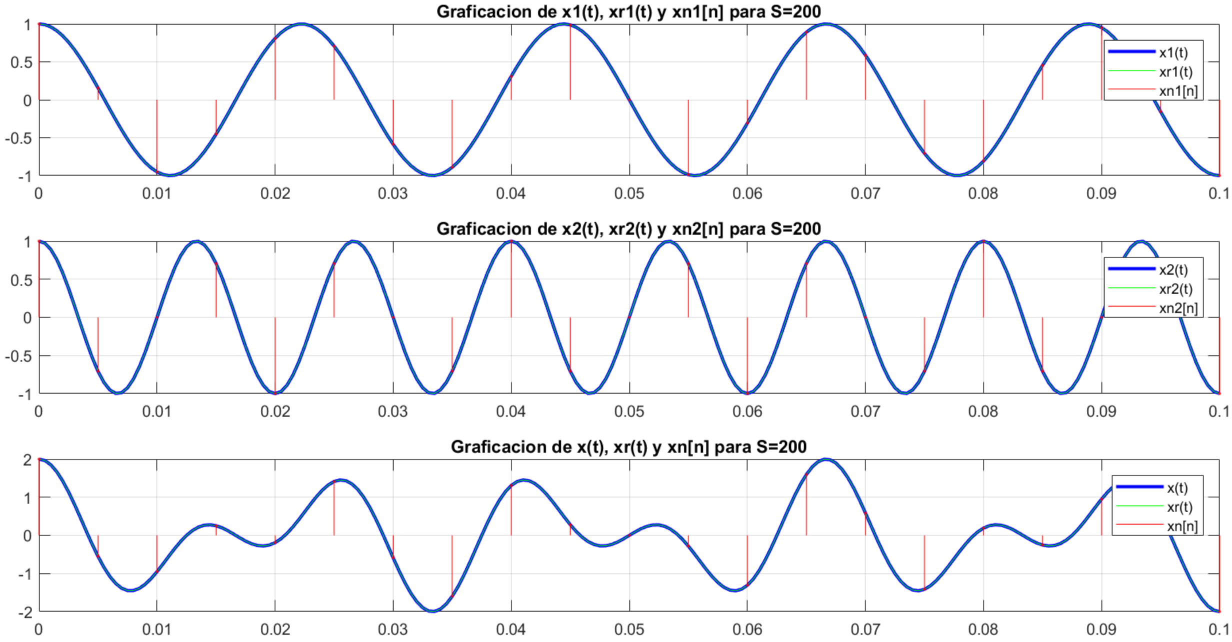
El código utilizado en matlab para este punto es el siguiente:

```
%% punto 2

t=0:0.1/200:0.1;
x1=cos(90*pi*t);
x2=cos(150*pi*t);
S1=200;S2=100;S3=50;
reconstruccion(x1,x2,t,S1);
reconstruccion(x1,x2,t,S2);
reconstruccion(x1,x2,t,S3);
```

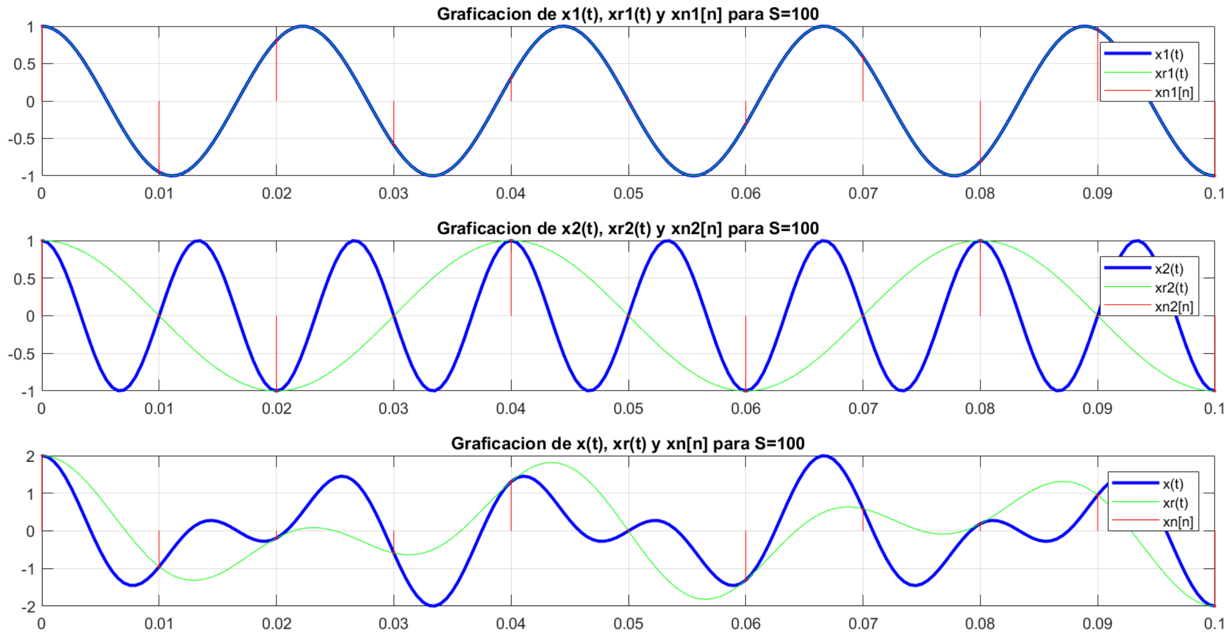
Siendo reconstrucción la siguiente función:

```
function reconstruccion(x1,x2,t,S);
    x=x1+x2;
    tn=0:1/S:0.1;
    [fa1 fd1]=alias(45,S); [fa2 fd2]=alias(75,S);
    xn1=cos(90*pi*tn);
    xn2=cos(150*pi*tn);
    xn=xn1+xn2;
    xr1=cos(2*pi*fa1*t);
    xr2=cos(2*pi*fa2*t);
    xr=xr1+xr2;

    %graficacion:
    figure();
    subplot(311);
    plot(t,x1,'b','LineWidth',2);
    hold on;
    plot(t,xr1,'g');
    dtplot(tn,xn1);
    grid;title("Graficacion de x1(t), xr1(t) y xn1[n] para S="+S);
    legend('x1(t)','xr1(t)','xn1[n]');

    subplot(312);
    plot(t,x2,'b','LineWidth',2);
    hold on;
    plot(t,xr2,'g');
    dtplot(tn,xn2);
    grid; title("Graficacion de x2(t), xr2(t) y xn2[n] para S="+S);
    legend('x2(t)','xr2(t)','xn2[n]');

    subplot(313);
    plot(t,x,'b','LineWidth',2);
    hold on;
    plot(t,xr,'g');
    dtplot(tn,xn);
    grid;title("Graficacion de x(t), xr(t) y xn[n] para S="+S);
    legend('x(t)','xr(t)','xn[n]');
end
```

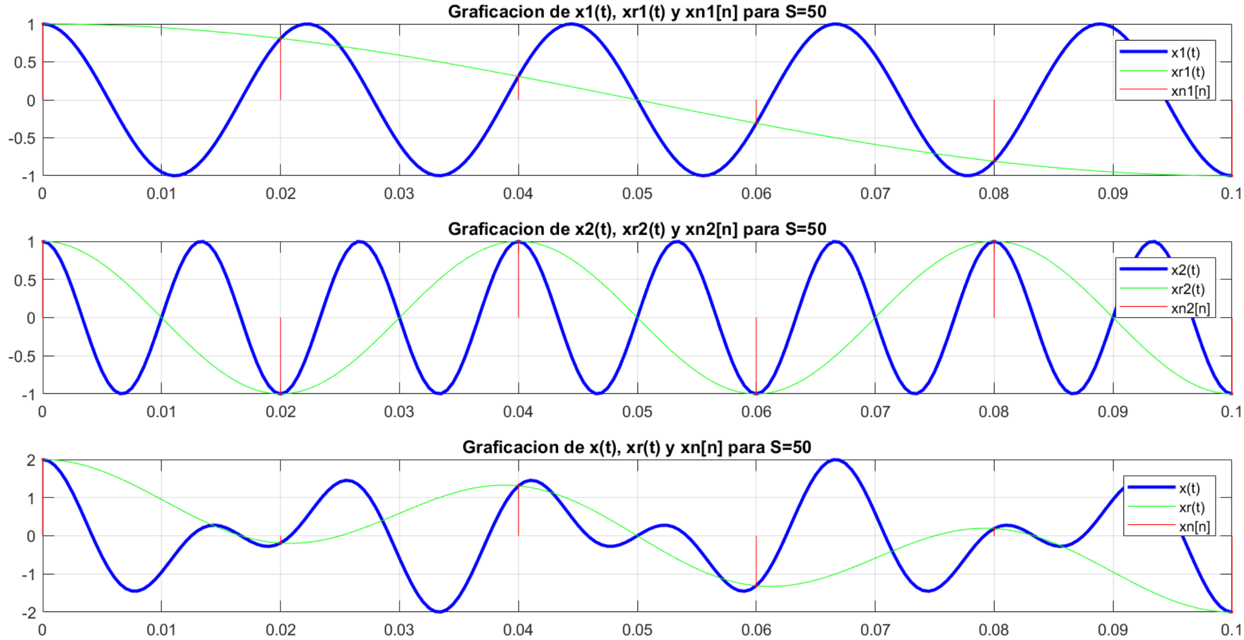
Los resultados para los distintos S son los siguientes:



Para S=200, se observa que ninguna de las tres señales se distorsiona (x1, x2 y x). Esto se debe a que S=200Hz es mayor que el doble de las dos frecuencias f1(45Hz) y f2(75Hz).



En el caso de S=100, se observa que la tasa de muestreo es suficiente para no distorsionar a x1, pero se observa aliasing en x2, lo cual termina distorsionando la señal xr. Esto se debe a que S=100 es mayor al doble de la frecuencia f1, pero no llega a ser el doble de la frecuencia f2. En este caso, fd2=25Hz.



En el caso de S=50, todas las señales se ven distorsionadas, ya que S=50Hz no cumple el criterio de Nyquist para ninguna de las frecuencias. En este caso, fd1=5Hz y fd2=25Hz.