**Sistema de Comunicaciones Digitales Integrantes:**

**Jose Andres Hernández Hernández ie704453**

**Práctica** **2 Robin Salgado de Anda ie686481**

**Fase 1 Isaac Yael Velázquez González ie703092**

**Objetivo**. El objetivo de esta práctica es evaluar el canal de comunicación entre dos computadoras, para identificar las bandas de frecuencia donde es posible transmitir sin distorsión. Para evaluar la calidad del canal utilizaremos los diagramas de ojo. Esta práctica es para realizarse en equipos de dos personas.

La primera parte de la práctica consiste en encontrar la respuesta en frecuencia del canal de comunicación, formado por una tarjeta de sonido para transmitir, un cable, y otra tarjeta de sonido para recibir.

**Procedimiento**:

1. Conectar la salida de audio a la entrada de audio de otra computadora y transmitir una señal sinusoidal.
   1. De preferencia utilizar las conexiones line-out y line-in, porque este tipo de interfaces tienen las impedancias bien acopladas. Algunas computadoras tienen una sola entrada compartida entre line-in y mic; otras computadoras no tienen line-out, o la comparten con la salida de audífonos, o la salida de micrófono se puede configurar como line-out o como line-in (investigue las capacidades de su tarjeta). Si es imposible utilizar estas interfaces, entonces utilice la salida de Audífonos y la entrada de micrófono.
   2. En todo lo que sigue, utilizar una frecuencia de muestreo fs = 96KHz (también puede ser a otra frecuencia, pero siempre usar la misma) y 16 bits por muestra. Vamos a utilizar un solo canal de audio (mono).
   3. Configurar los niveles de sonido tanto de la salida como de la entrada es esencial para que no haya distorsión. Haga un programa que genere una señal sinodal de amplitud 1 (es decir, amplitud pico a pico igual a 2), frecuencia igual a 5,000 Hz y duración un segundo. Reproduzca la señal de audio en una computadora y al mismo tiempo grábela en la otra. Se pueden utilizar los comandos sound() para reproducir; audiorecorder() para la creación de un objetos y record() para grabar. La idea es comenzar con niveles de volumen y sensibilidad muy bajos, e irlos subiendo paulatinamente. Si la señal grabada no se ve bien, puede ser útil conectar la salida de la tarjeta de sonido a un osciloscopio para distinguir si el problema está en la generación o en la grabación de la señal.
   4. Observar la señal grabada (puede utilizar un software para grabar la señal y posteriormente analizarla en Matlab: http://www.digitaltrends.com/computing/best-freerecording-software/). Debe verse la misma sinodal, sin cambios de frecuencia, pero sobre todo sin saturación (debe observarse claramente la curva donde la señal cambia de creciente a decreciente).
   5. Apuntar o guardar los niveles de volumen y sensibilidad que den los mejores resultados.

**Imágenes:**

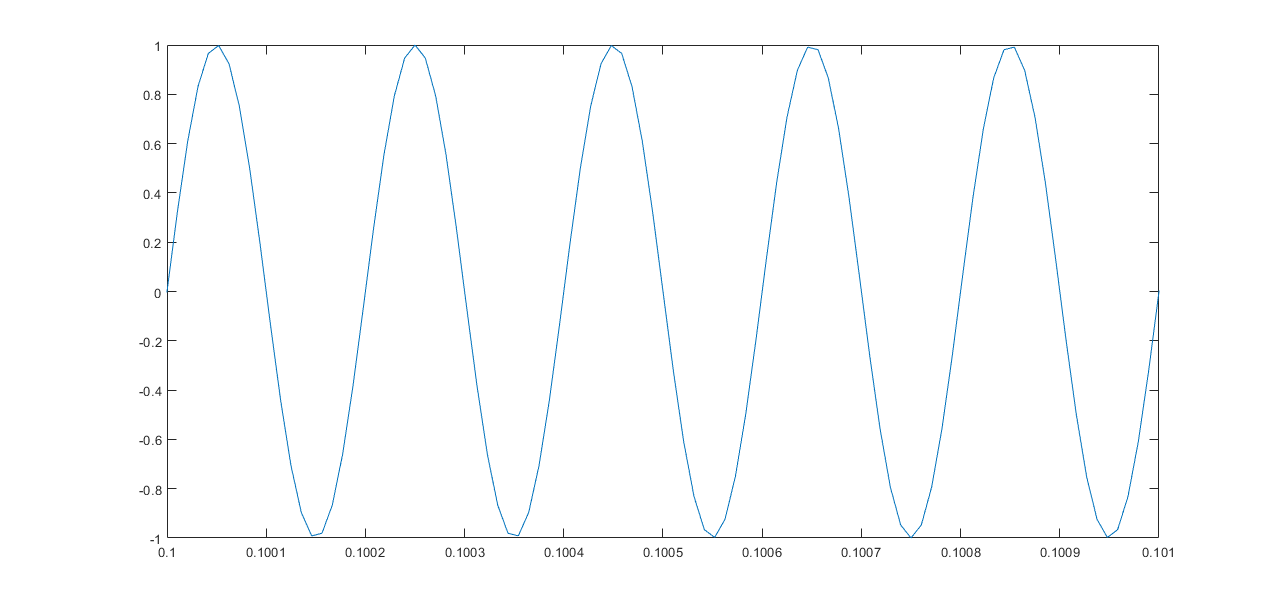
****

Figura 1a. Senoidal a transmitir de 5kHz con 70% de volumen

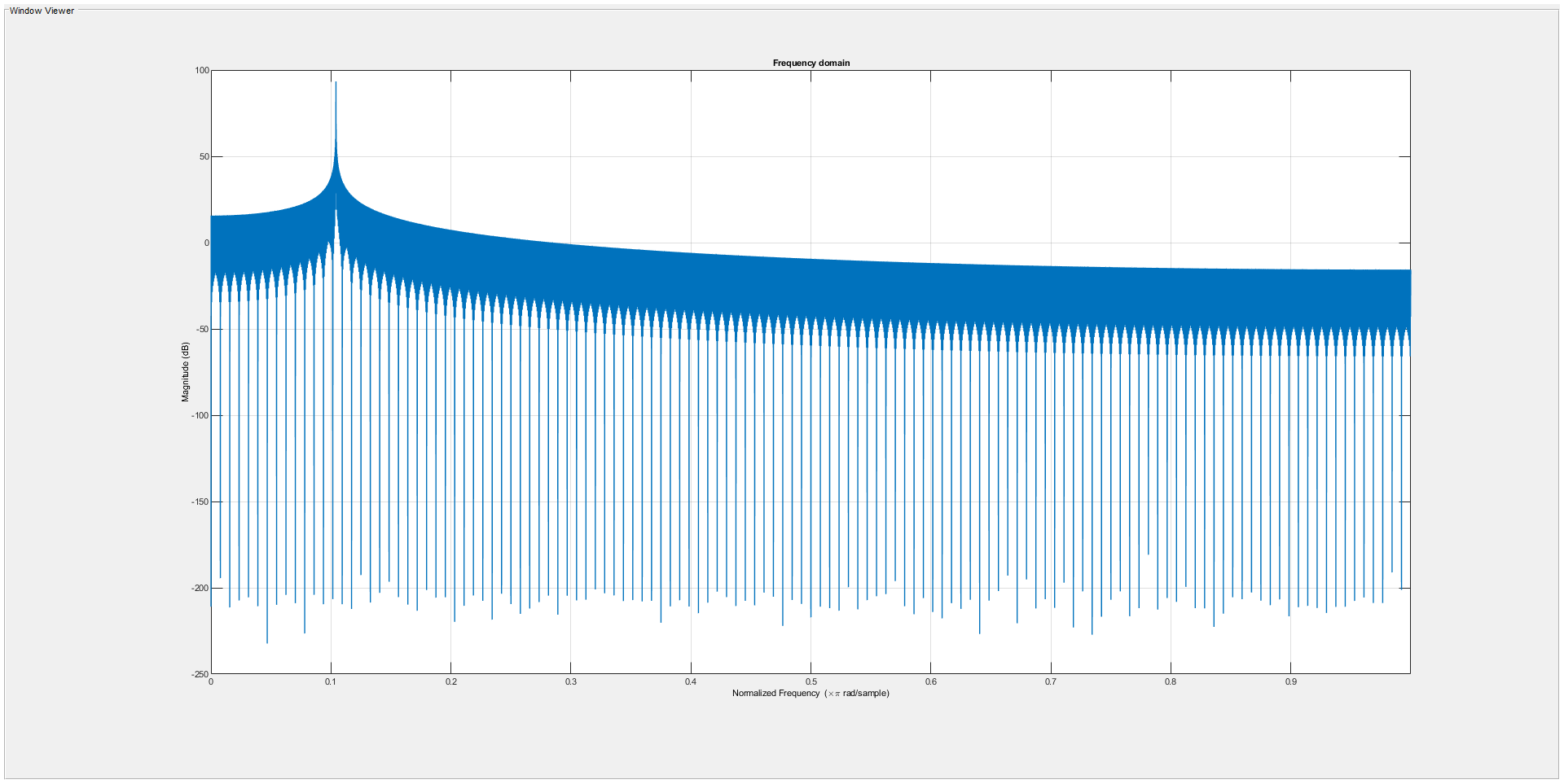
****

Figura 1b. Espectro de frecuencia de senoidal a transmitir de 5kHz de amplitud 1

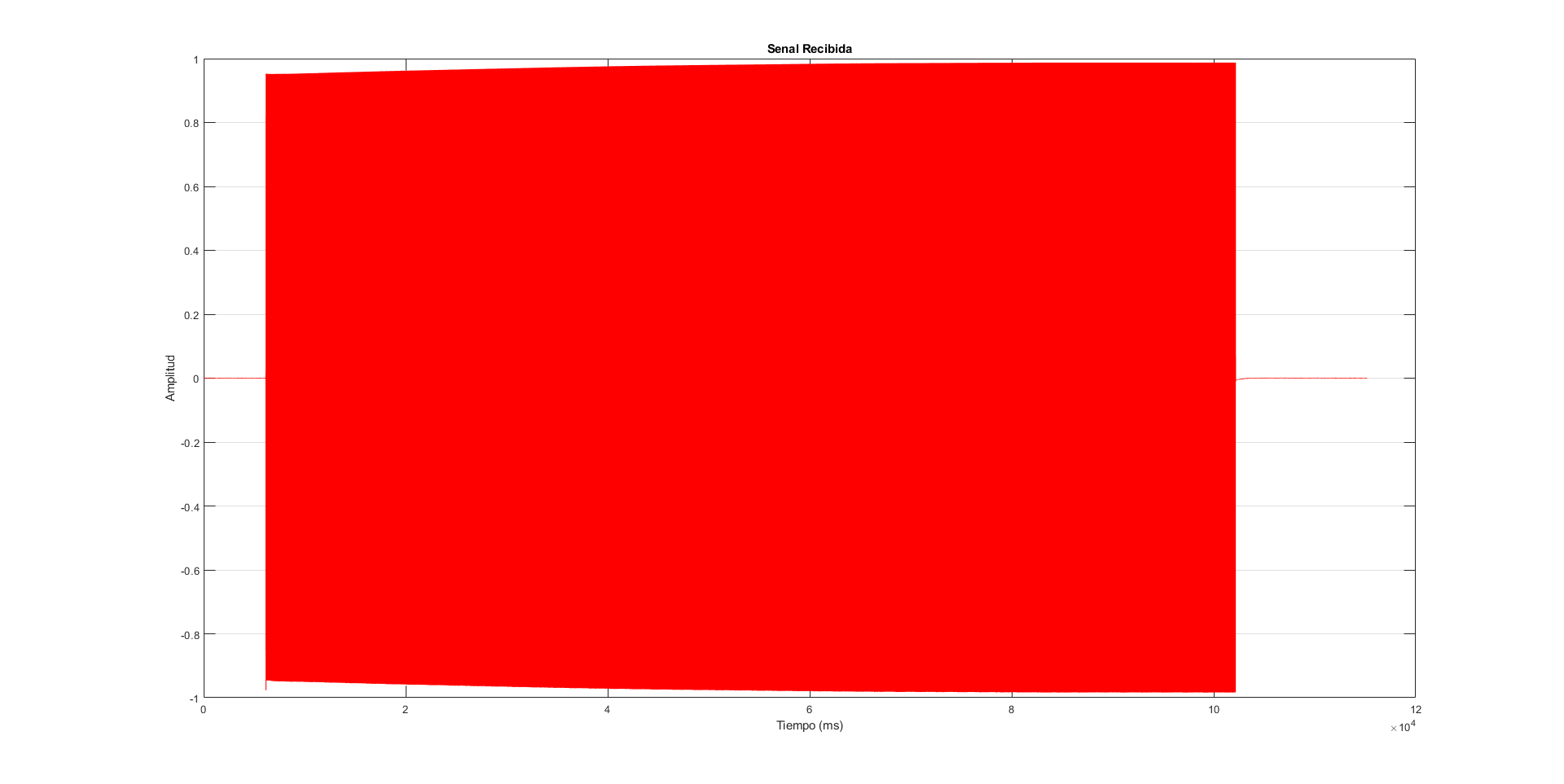


Figura 1c. Señal senoidal recibida con 70% de sensibilidad

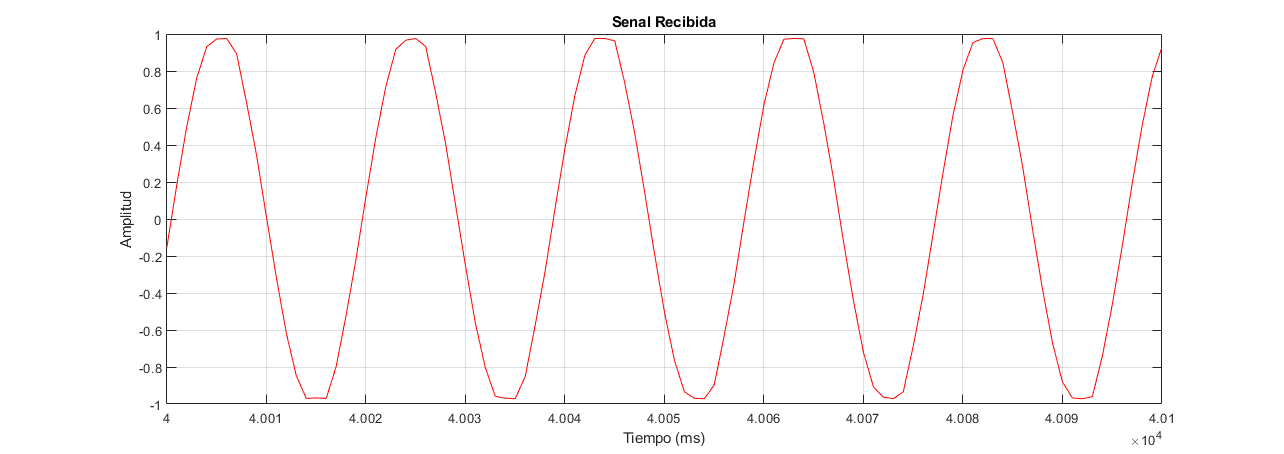


Figura 1d. Senal senoidal recibida con 70% de sensibilidad con zoom

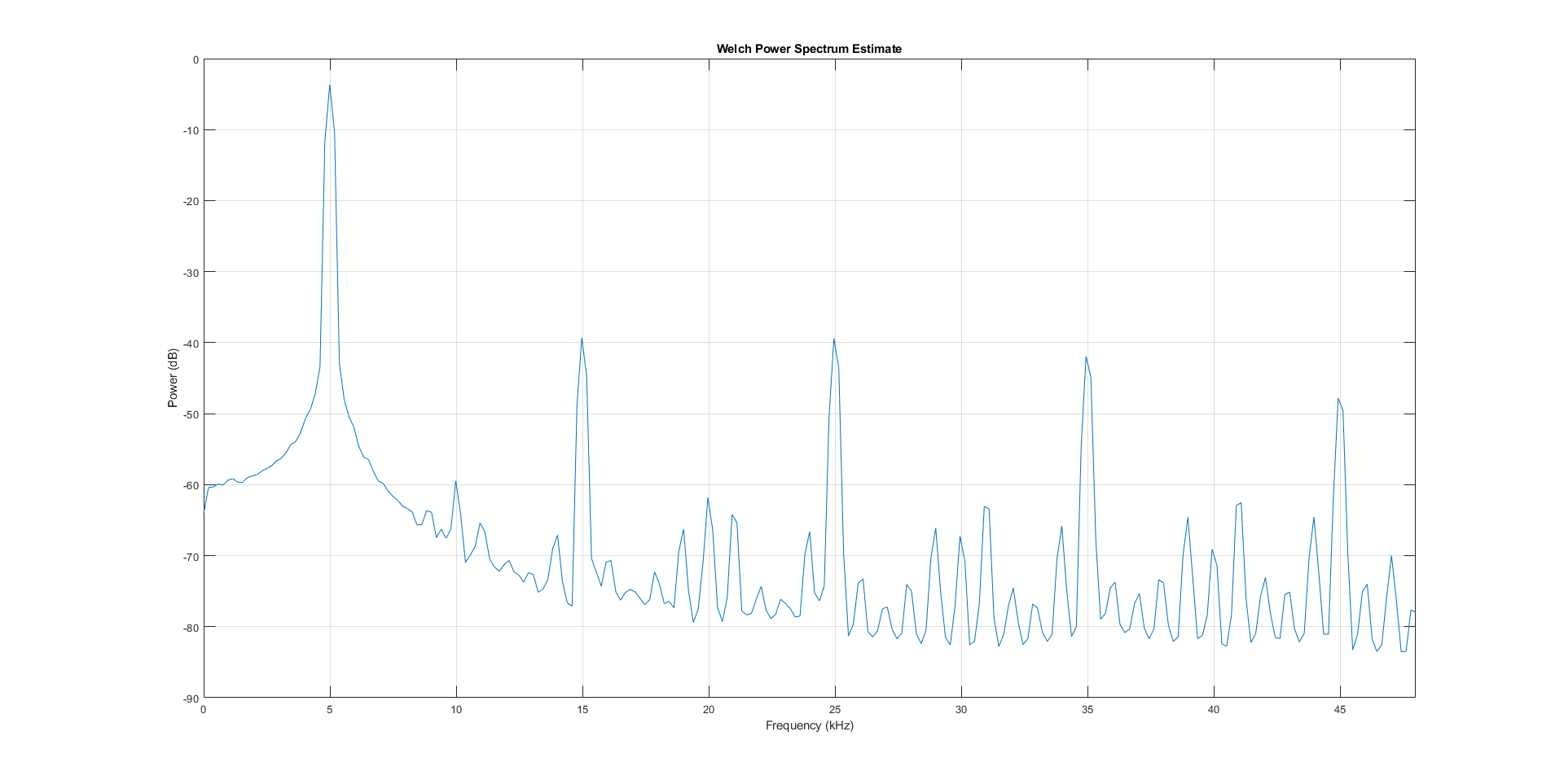


Figura 1e. Espectro de frecuencia de señal senoidal de 5kHz recibida

1. Identificar el canal, es decir, obtener su respuesta en frecuencia. Vamos a emplear tres técnicas diferentes.
   1. Primero transmita un impulso, es decir, una señal formada de ceros (digamos, un segundo), después un uno, y después otro segundo de ceros. Según la teoría, la señal grabada es la respuesta al impulso del canal, y su transformada de Fourier es su respuesta en frecuencia (función de transferencia en el dominio de Z).
   2. Grafique la señal recibida en el tiempo y su transformada de Fourier.
   3. Como segunda técnica, utilizaremos una señal que tiene el mismo espectro que el impulso: el ruido gaussiano. Generar una señal de ruido de cinco segundos de duración (ver tutorial sobre el ruido). Utilizando el comando pwelch (u otro comando en Matlab relacionado a la estimación espectral como se vio en clase), estimar la densidad espectral de potencia de la señal. Debe ser muy cercana a plana entre 0 y 48,000 Hz, esto es, fs/2.
   4. Transmita y grabe esta señal (puede usar el software Audacity), y grafique el espectro de la señal grabada. Esa es la función de transferencia del canal.
   5. Genere una señal que sea la suma de sinodales de la misma amplitud y de frecuencias 500:500:20000. Grafique el espectro de esta señal y de la señal recibida.
   6. Genere una señal “chirp” (-1 volt a 1 volt) de frecuencias 500:500:20000. Basándose en el siguiente ejemplo: t=0:0.001:2; % 2 secs @ 1kHz sample rate y=chirp(t,0,1,150); % Start @ DC, cross 150Hz at t=1sec
   7. Grafique el espectro de esta señal y de la señal recibida en tiempo y frecuencia.

**Imágenes:**

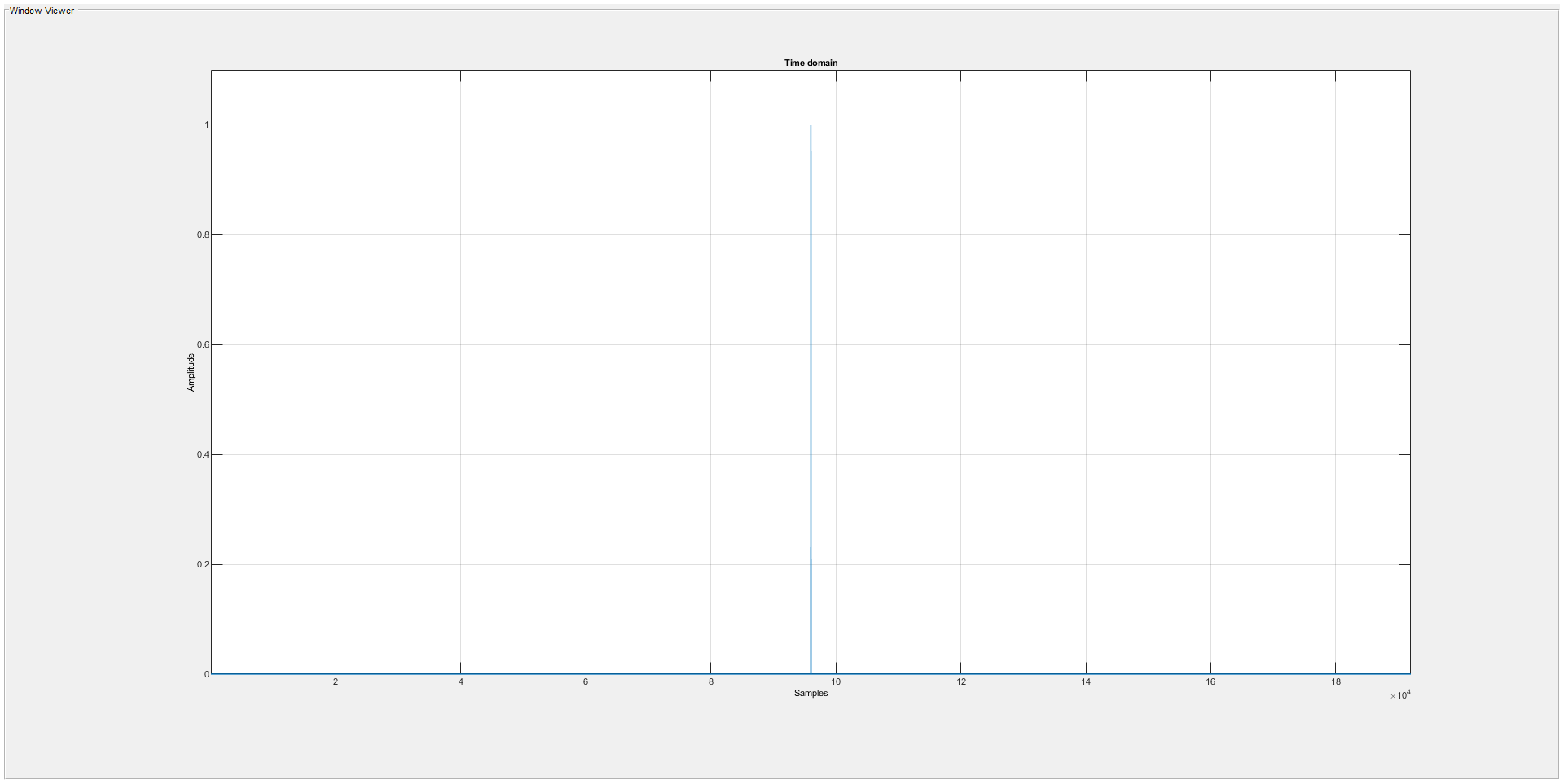
****

Figura 2a. Impulso en el tiempo desde Tx

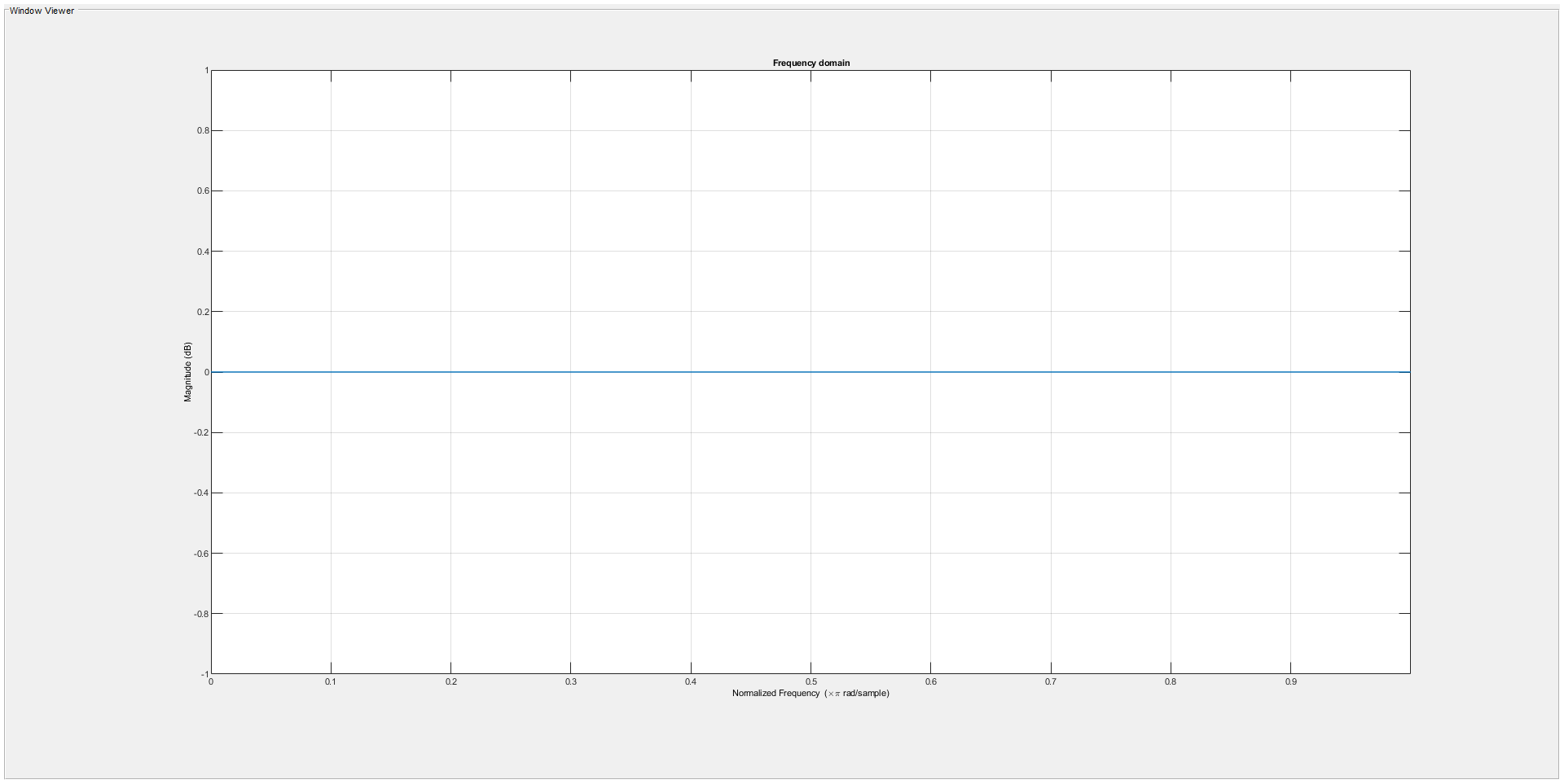
****

Figura 2b. Impulso en frecuencia desde Tx

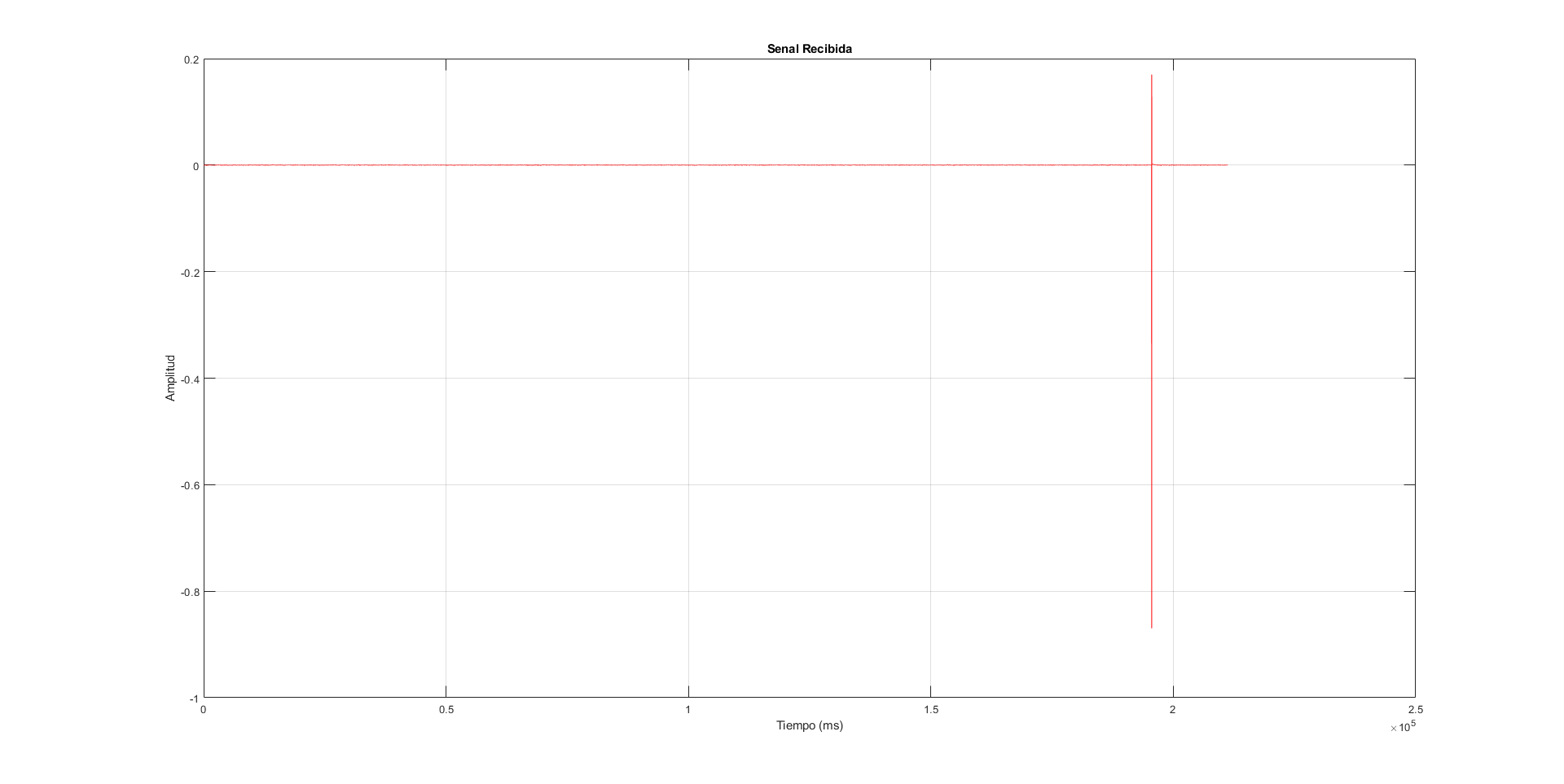


Figura 2c. Impulso en tiempo desde Rx

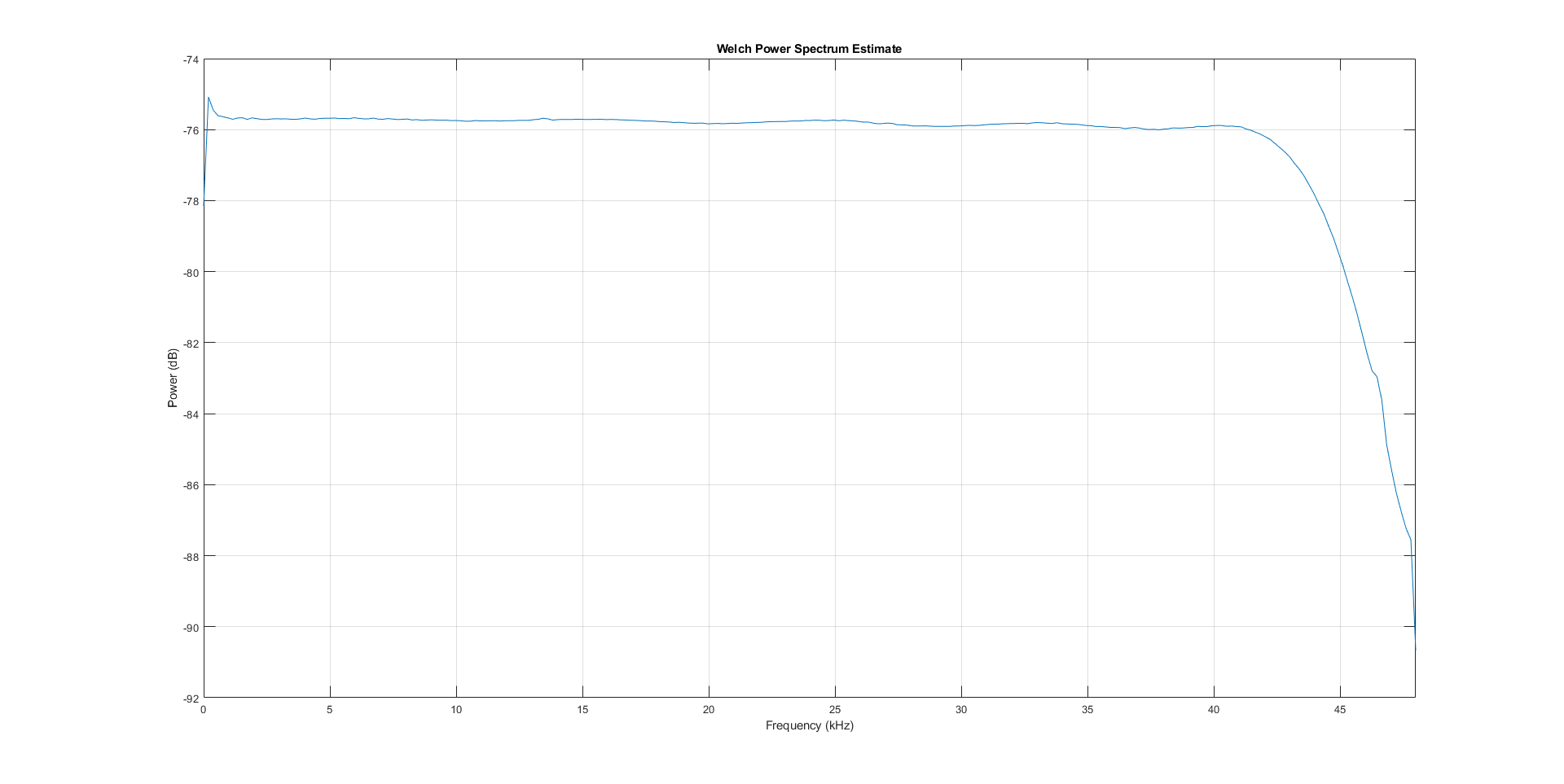


Figura 2d. Impulso en frecuencia desde Rx

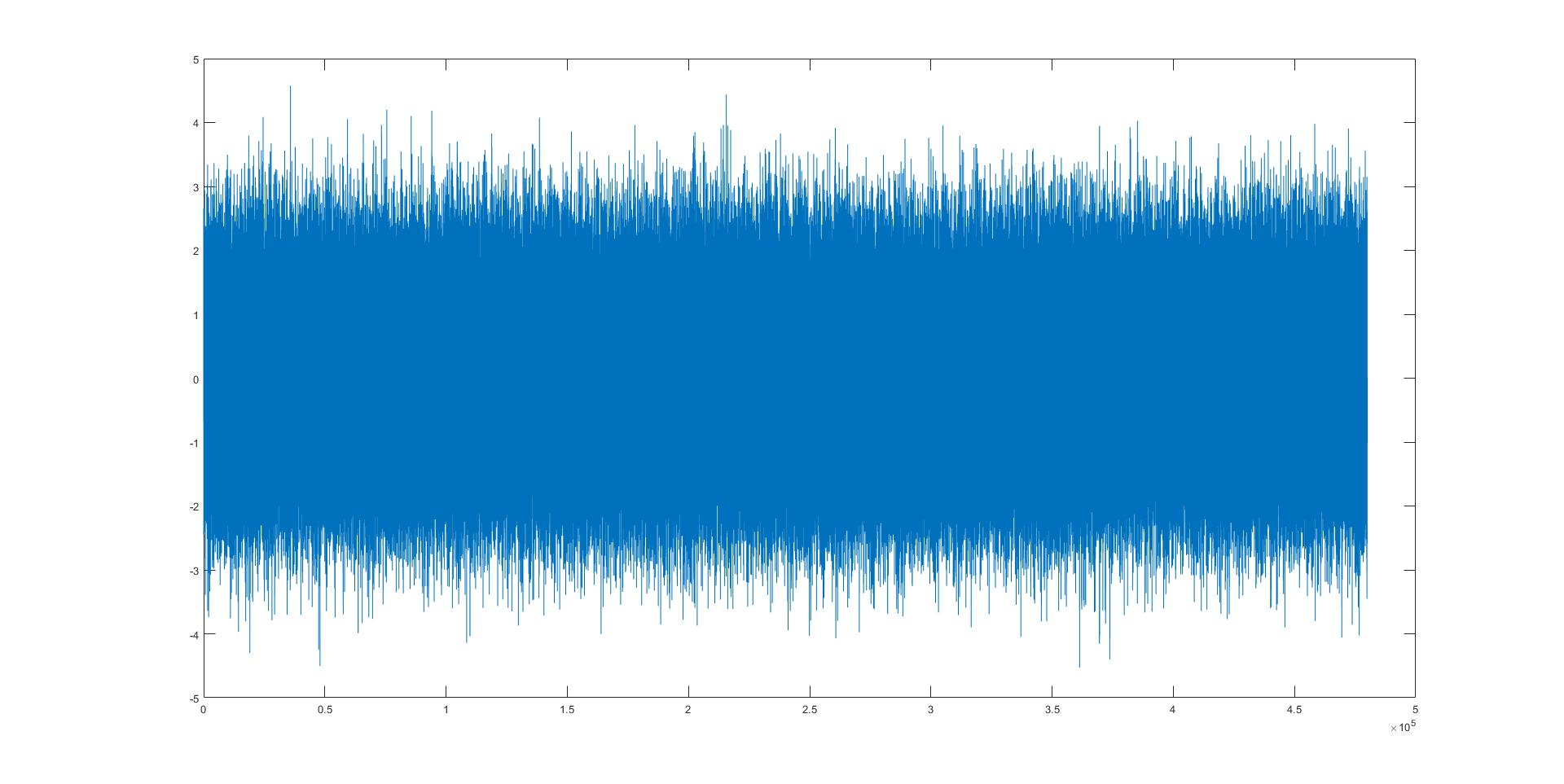


Figura 2e. Ruido gaussiano en tiempo desde Tx

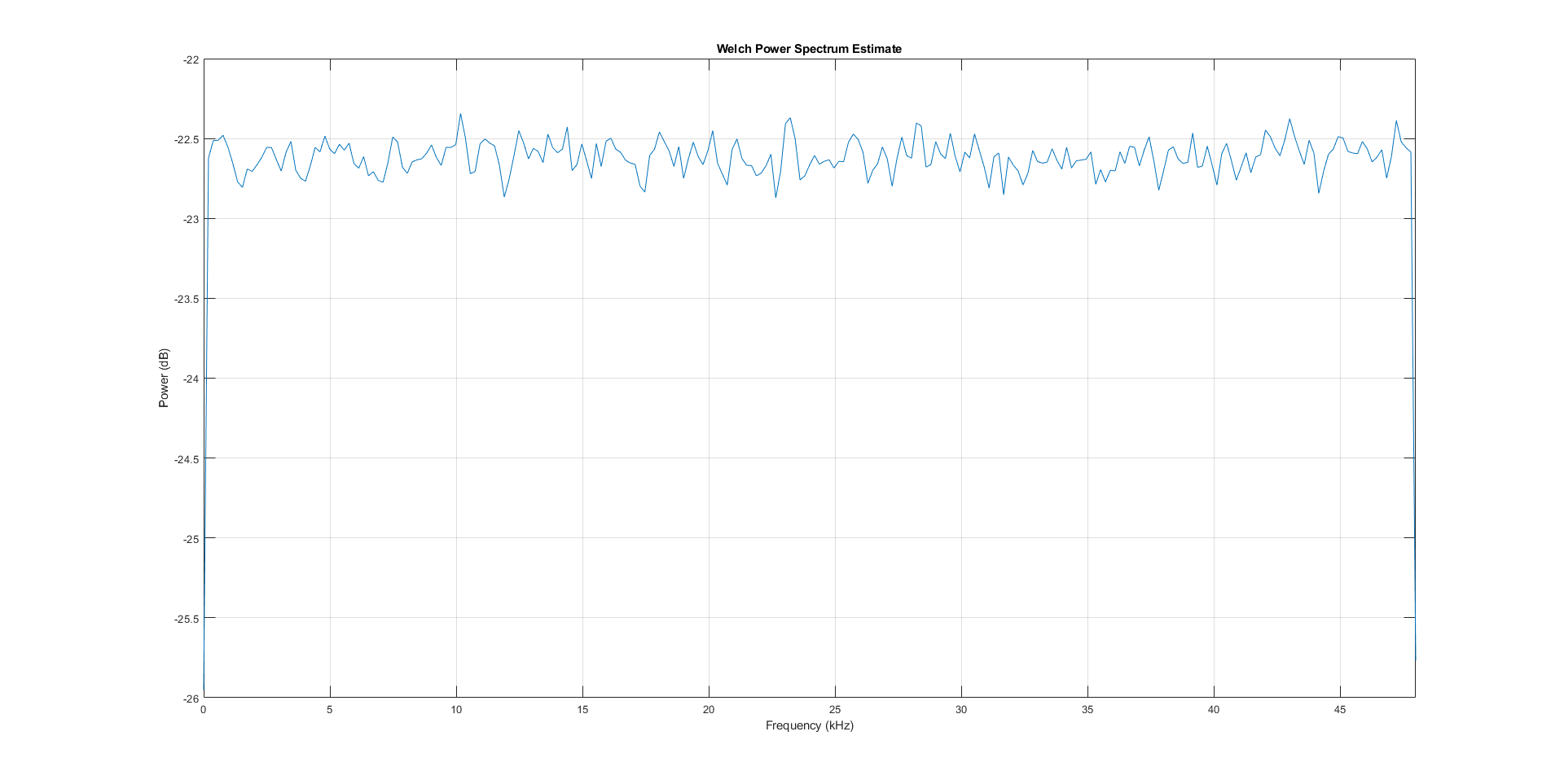


Figura 2f. Ruido gaussiano en frecuencia desde Tx

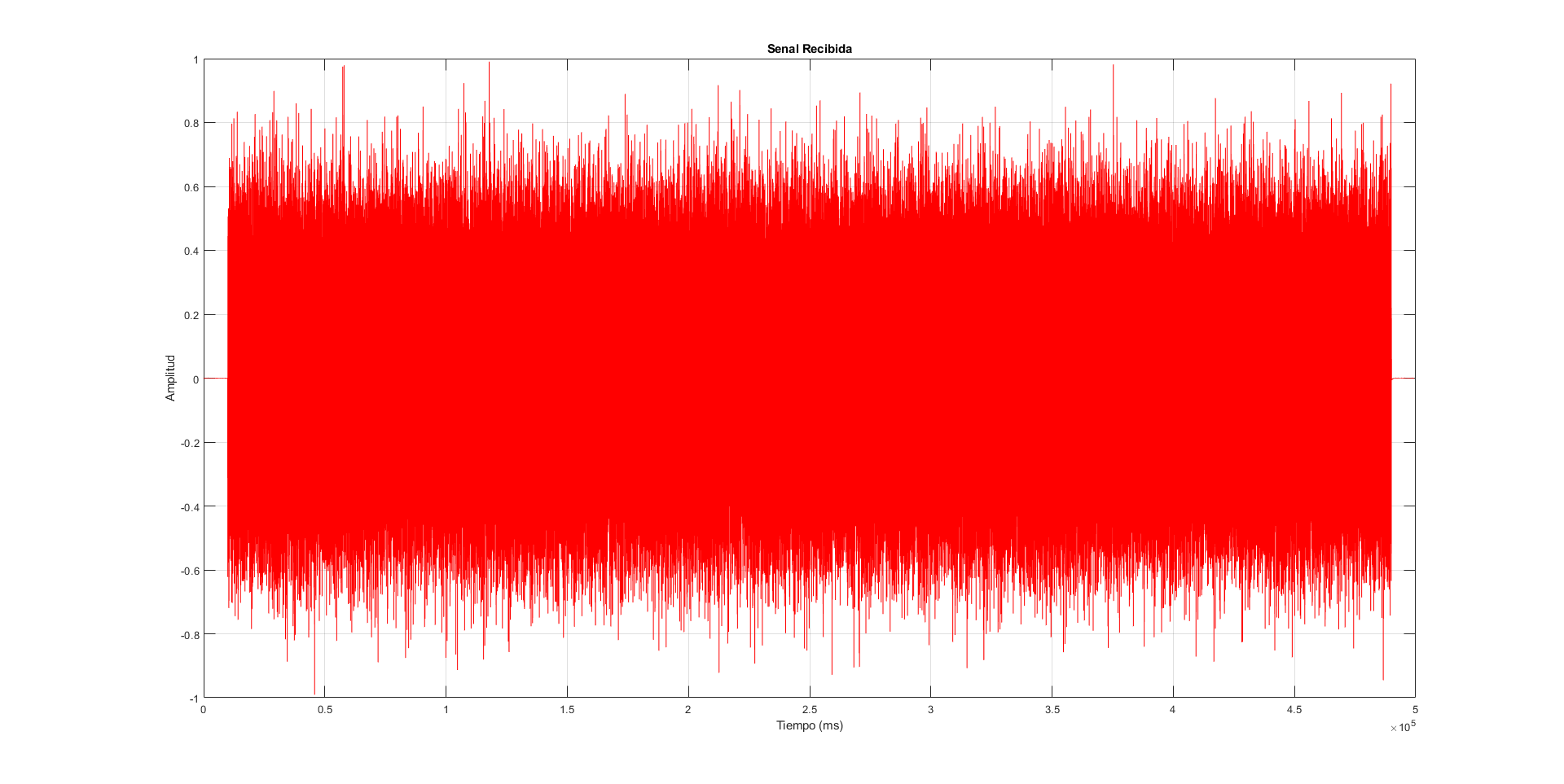


Figura 2g. Ruido gaussiano en tiempo desde Rx

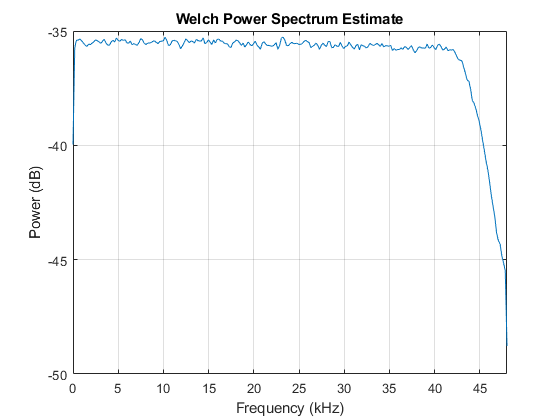


Figura 2h. Ruido gaussiano en frecuencia desde Rx

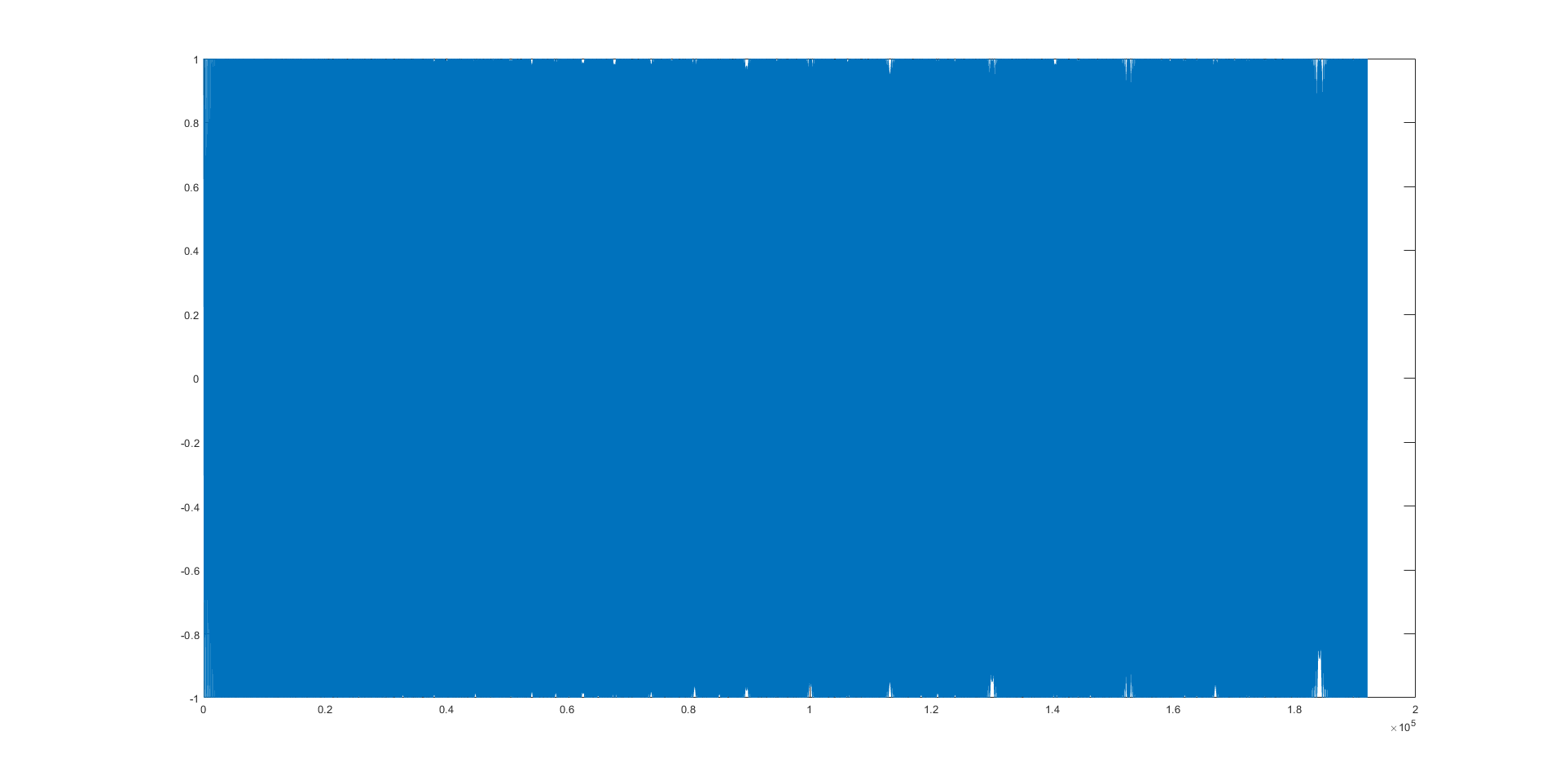


Figura 2i. Senal Chirp en tiempo desde Tx

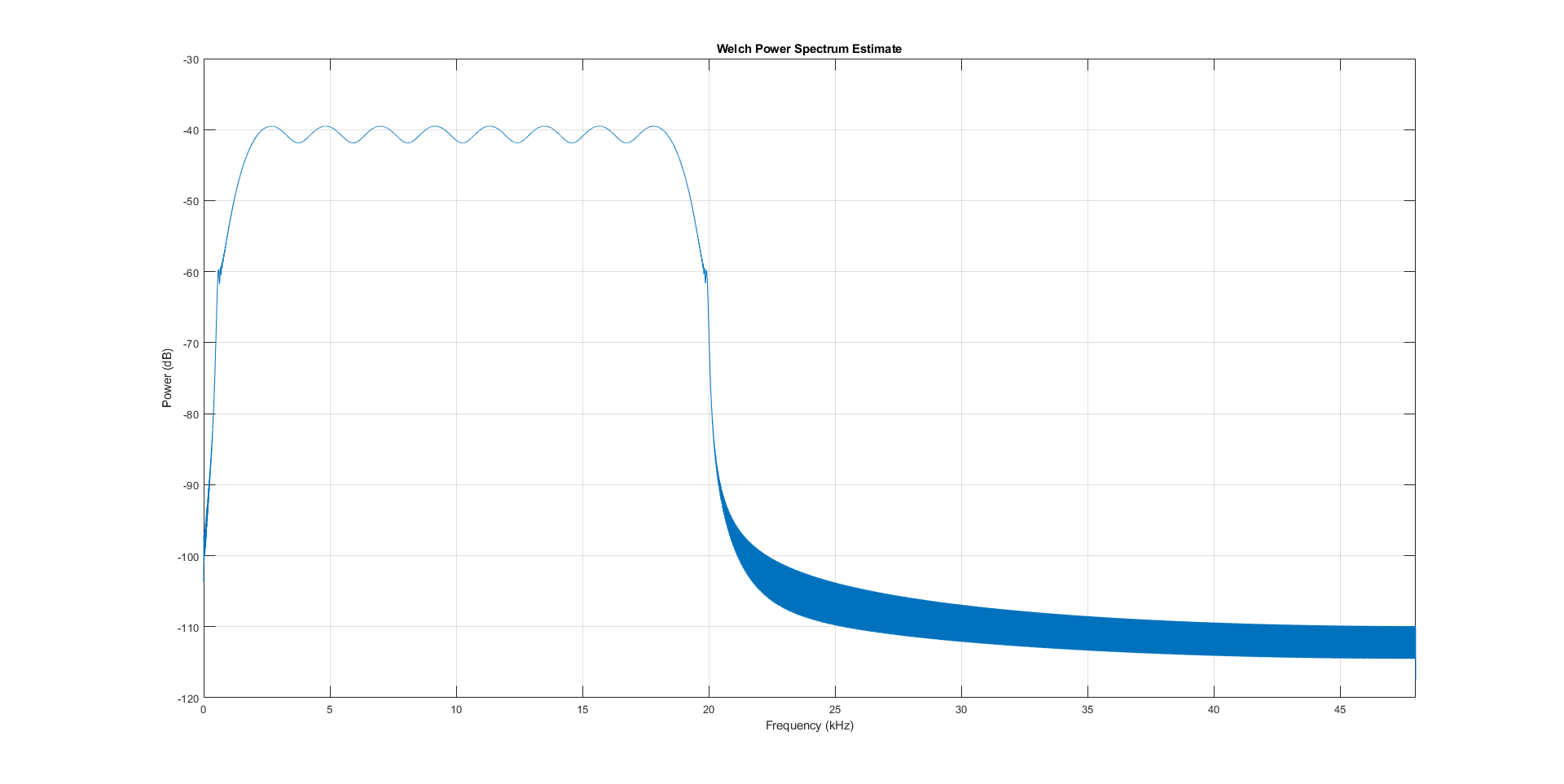


Figura 2j. Senal chirp en frecuencia desde Tx

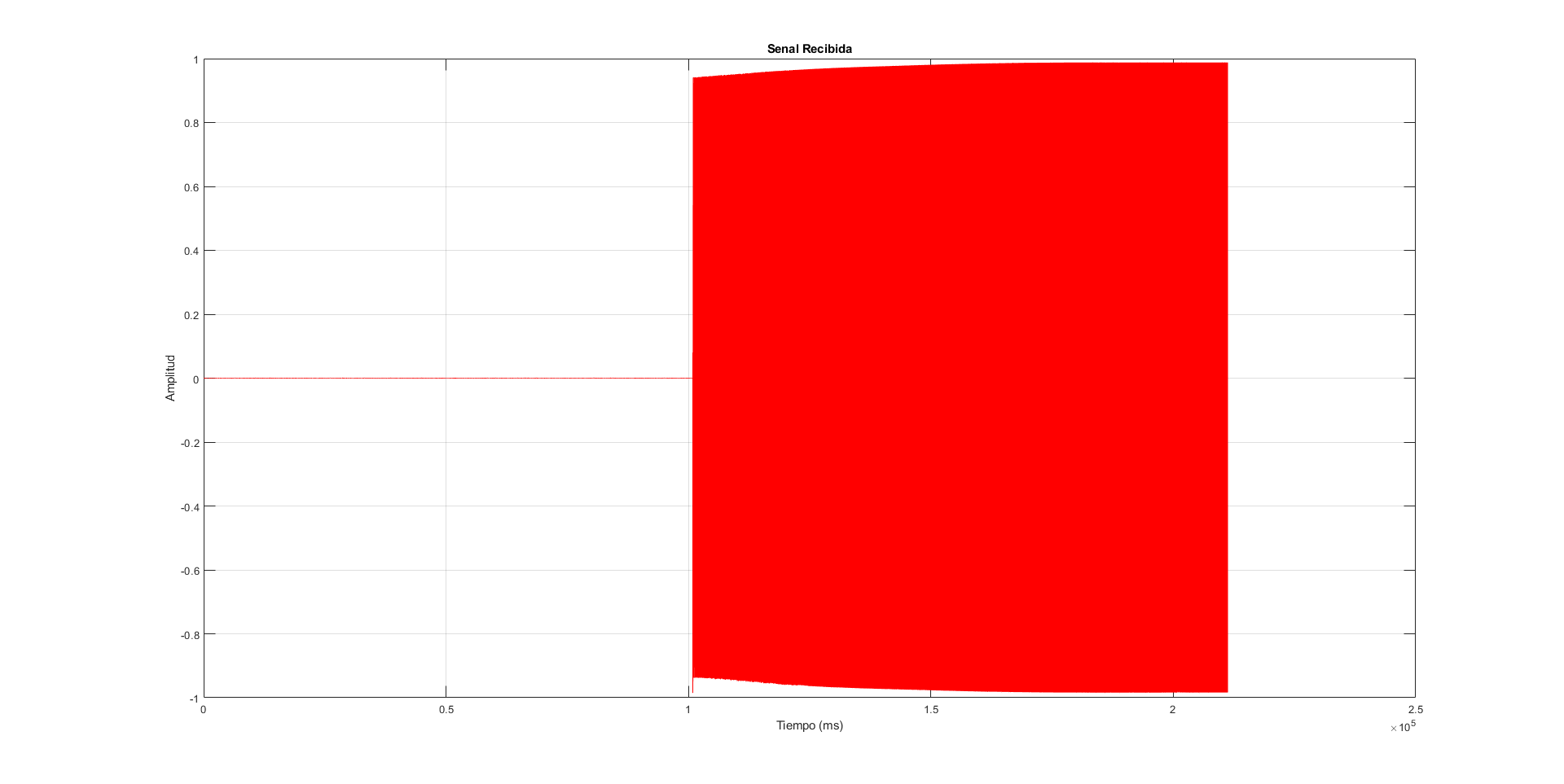


Figura 2k. Senal chirp en tiempo desde Rx

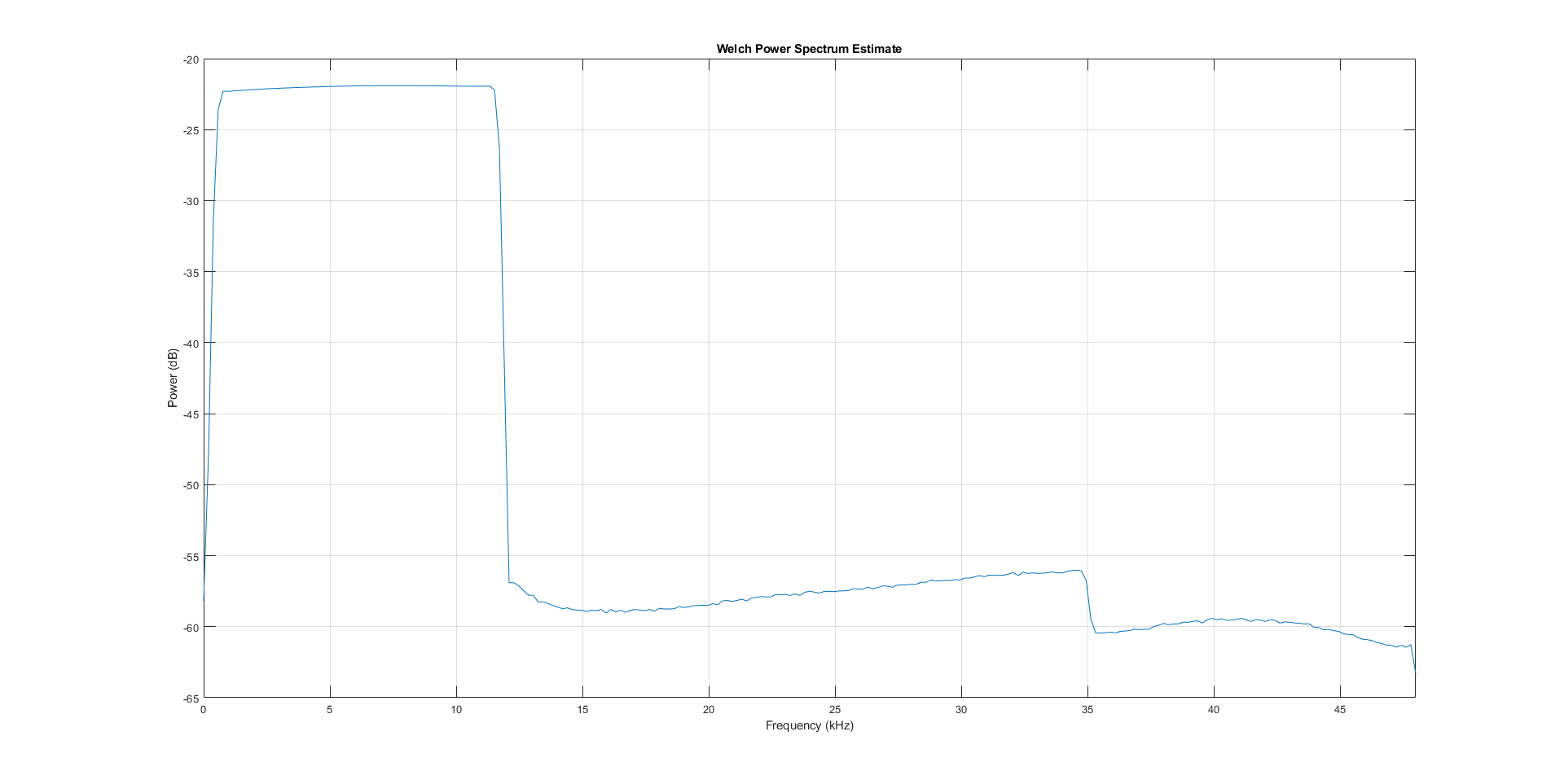


Figura 2L. Senal chirp en frecuencia desde Rx

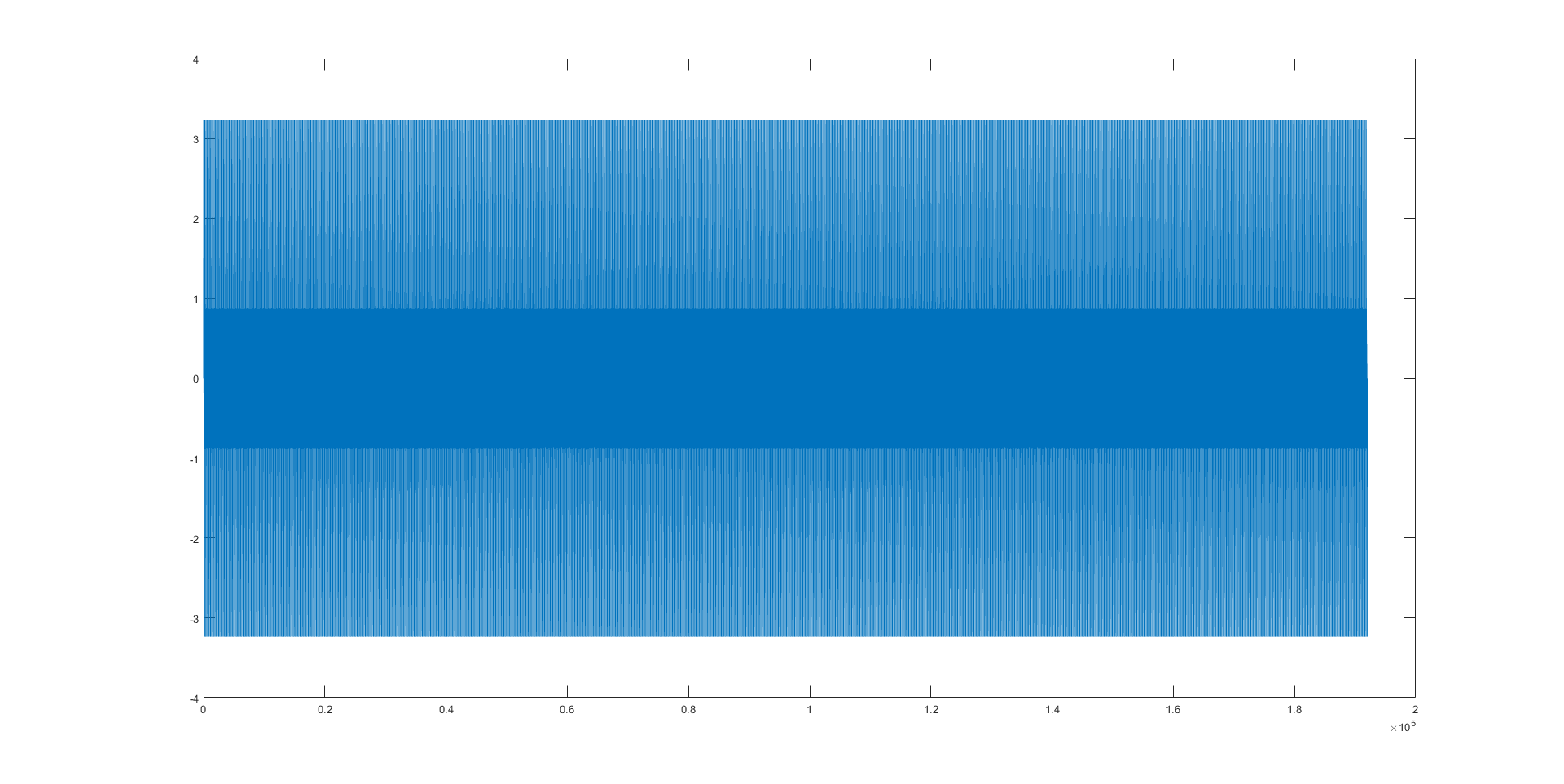


Figura 2m. Suma de senoidales en tiempo desde Tx

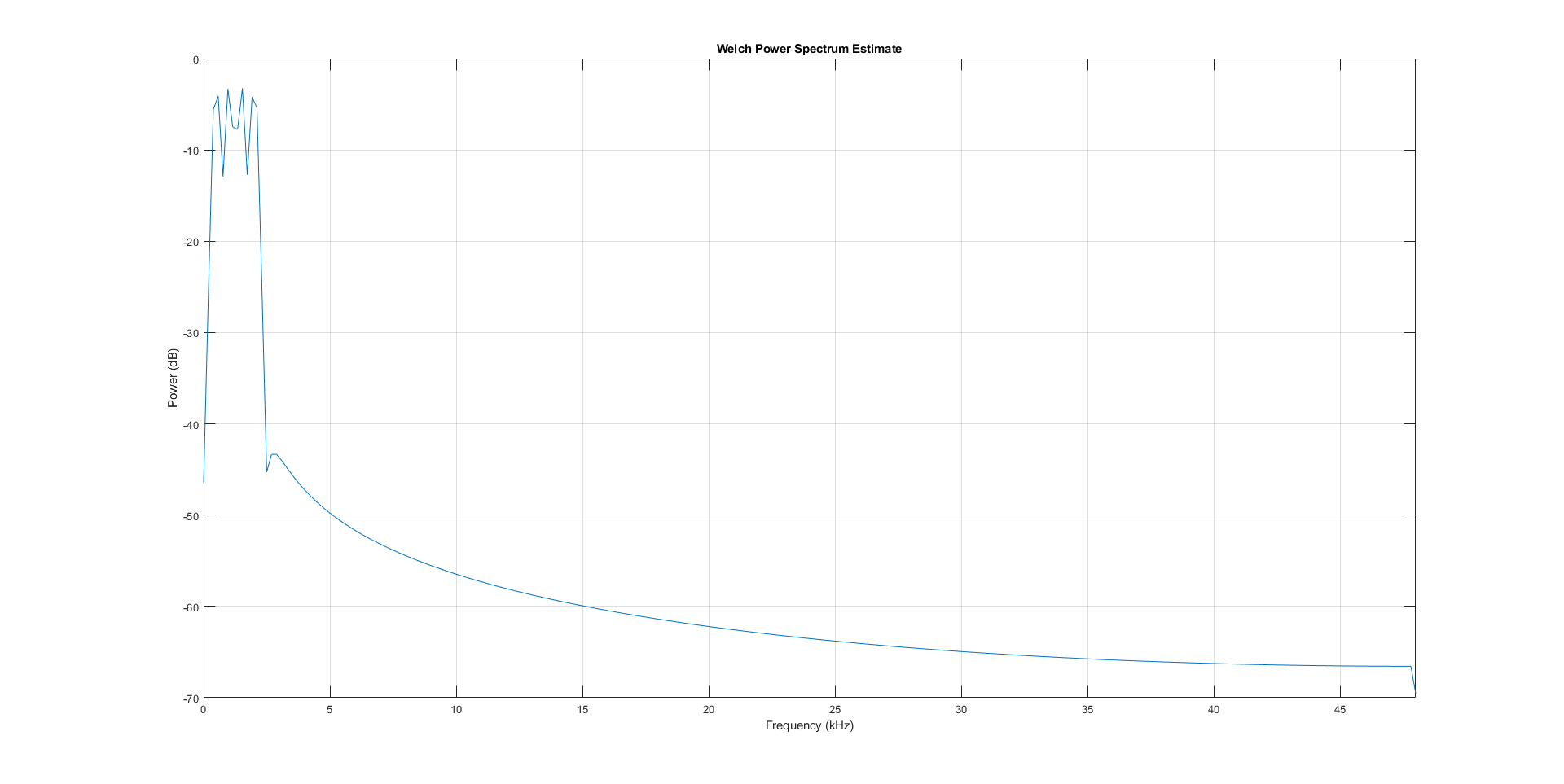


Figura 2n. Suma de senoidales en frecuencias desde Tx

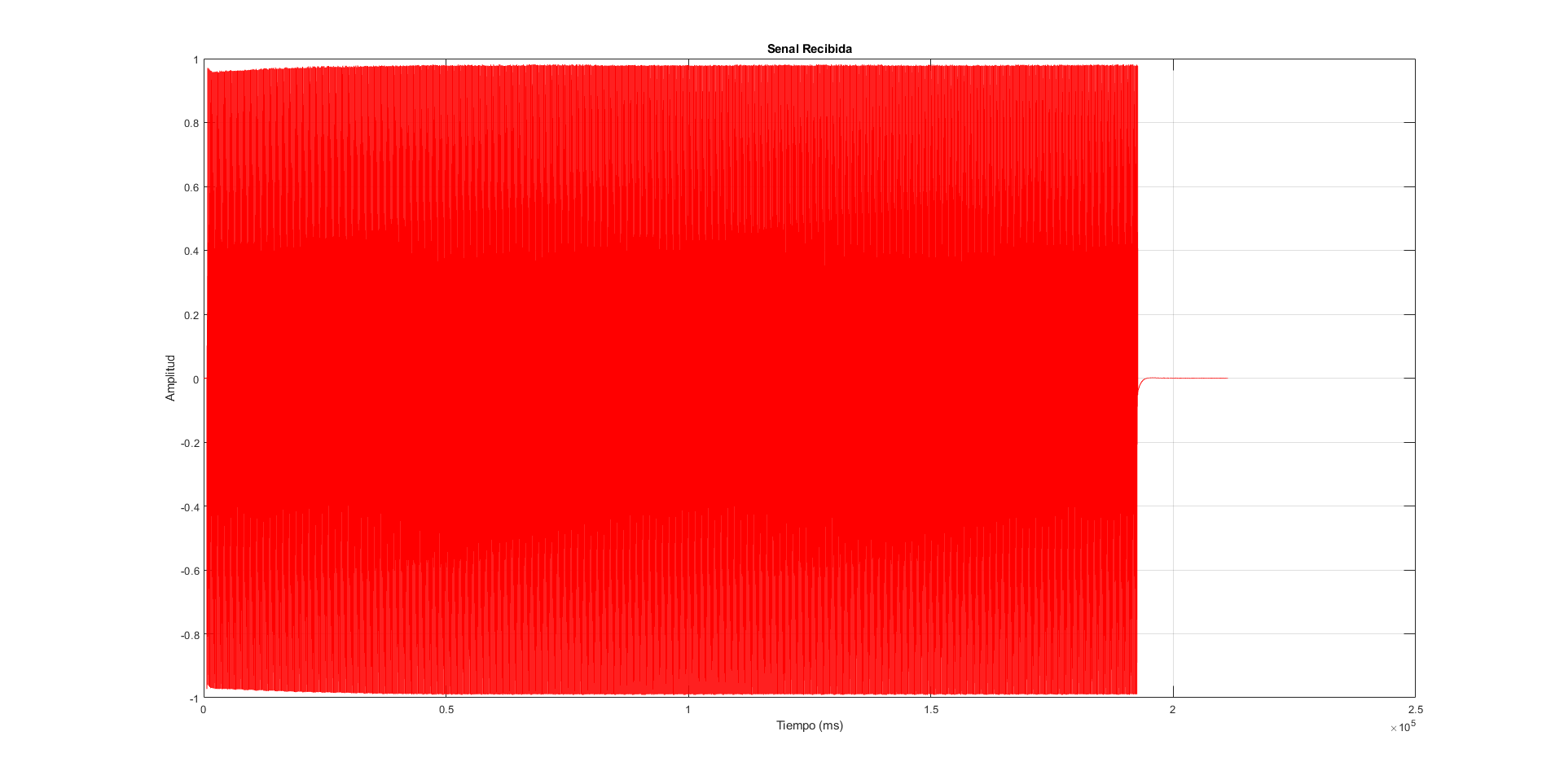


Figura 2o. Suma de senoidales en tiempo desde Rx

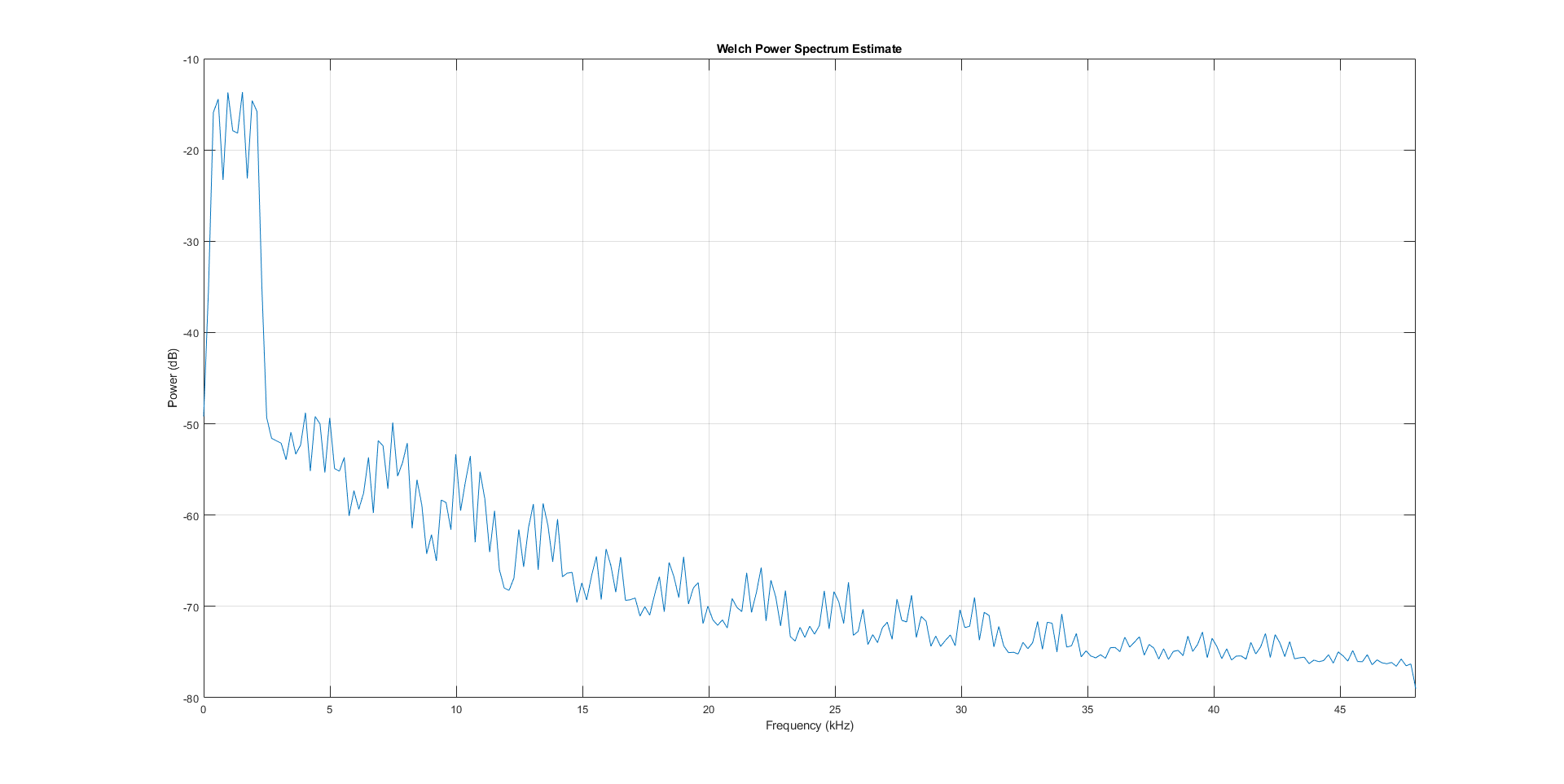


Figura 2o. Suma de senoidales en frecuencia desde Rx

1. Compare las respuestas en frecuencia obtenidas con cada método.
   1. ¿Qué conclusiones pueden obtenerse?

El primer método del impulso fue el método más difícil de detectar debido a que la recepción captaba principalmente ruido. El segundo método fue el ruido gaussiano el cual tuvo una mejor caracterización del canal de comunicación debido a que se comportaba como un filtro pasa-bajas con frecuencia de corte de 44.54kHz de acuerdo al ancho de banda con caída de -3dB. El tercer método con la señal chirp se pudo observar que el canal de comunicación reacciona como un pasa-bajas, concordando con el anterior método.

* 1. ¿Cuál es la banda más plana y más ancha que es posible encontrar en este canal?

La banda de paso del método de chirp fue la más plana y ancha fue la del método chirp, tomando en cuenta que la señal transmitida fue de 20kHz.

**Código Transmisor:**

%% Paso 1: Generacion de una senal senoidal de 5kHz de frecuencia y 1 de

% amplitud. Reproducir el sonido de manera que se transmita al

% receptor de manera que se encuentren los valores de volumen y

% sensibilidad adecuados para el sistema

close all; clc; clear all; % Borramos todo

fs = 96e3; % Frecuencia de muestre a 96kHz

mp = 16; % 16-bits por muestra

Amp = 1 % Amplitud de senoidal

f = 5000; % Frecuencia de senoidal

T = 1; % Cantidad de segundos limite

t = 0:1/fs:T; % Vector de tiempo

y = sin(2\*pi\*f\*t); % Funcion de senoidal

plot(t,y) % Grafica de la senal senoidal

soundsc(y,fs) % Reproduccion de audio

%% Paso 2: Identificar el canal, es decir, obtener su respuesta en

% frecuencia. Vamos a emplear tres técnicas diferentes.

%

%% a) Impulso conformado por un segundo de 0 1 0

Pulse = zeros(1,2\*fs+1) % Vector de ceros con duracion de 2s

Pulse(fs) = 1; % A la mitad del vector ponemos un solo pulso

soundsc(Pulse,fs); % Reproducimos con una frecuencia de 96kHz

%% c) Como segunda técnica, utilizaremos una señal que tiene el mismo

% espectro que el impulso: el ruido gaussiano

Ruido = randn(1,5\*fs); % Vector de ruido gaussiano de 5s

pwelch(Ruido,[],[],[],fs,'power'); % Lo analizamos con pwelch

soundsc(Ruido,fs); % Lo reproducimos

%% e) Genera una senal que sea la suma de senoidales de misma amplitud y

% con frecuencias de 500:500:2000

time = 0:1/fs:2; % Vector de tiempo de 2s

sen1 = sin(2\*pi\*500 \*time); % Senoidal de 500Hz

sen2 = sin(2\*pi\*1000\*time); % Senoidal de 1kHz

sen3 = sin(2\*pi\*1500\*time); % Senoidal de 1.5kHz

sen4 = sin(2\*pi\*2000\*time); % Senoidal de 2kHz

senTotal = sen1 + sen2 + sen3 + sen4; % Suma de senoidales

soundsc(senTotal,fs); % Lo reproducimos

%% f) Genere una señal “chirp” (-1 volt a 1 volt) de frecuencias

% 500:500:20000

t = 0:1/fs:2; % Vector de tiempo de 2s con pasos de 1/fs

y = chirp(t,500,2,20e3); % Senal chirp de 2s segundos de 500 a 20kHz

pwelch(y,[],[],[],fs,'power'); % Analisis con pwelch

soundsc(y,fs); % Reproducimos

**Código Recepción:**

%% Definicion de parametros y creacion de objeto para grabacion

clear all; clc; close all; % Limpieza del entorno

Fs = 96e3; % Frecuencia de muestreo a 96kHz

mpbits = 16; % 16-bits por muestra

nChannels = 1; % Utilizacion de un canal de audio

ID = -1; % Dispositivo de entrada de audio default

time = 1.2; % Tiempo expresada en segundos

% Creacion de objeto para guardar la grabacion de audio

recObj = audiorecorder(Fs, mpbits, nChannels, ID);

%% Grabacion de audio

disp('Grabando Señal...'); % Mensaje de inicio de grabacion

recordblocking(recObj, time); % Grabacion de audio determinada por time

disp('Fin de la grabacion'); % Mensaje de final de grabacion

%% Recuperacion de los datos

%play(recObj); % Primer testeo de audio recibido

signal\_received = getaudiodata(recObj); % Obtencion de senal recibida

plot(signal\_received,'r') % Graficacion de senal recibida

ylabel('Amplitud') % Eje Y como amplitud de la senal

xlabel('Tiempo (ms)') % Eje X como tiempo en ms

title('Senal Recibida') % Titulo de la senal recibida

grid on