Fundamentos de las Comunicaciones
Año 2023

Laboratorio 1: Demodulación digital de señal real de FM

Fecha límite de entrega: 14/11/23

1 Introducción

Este laboratorio, previsto como una de las actividades obligatorias de la materia, consiste en la utilización de un *Dongle* USB para analizar e implementar algunos aspectos de lo que se conoce como *Software-Defined Radio* (SDR) o Radio Definida por Software. Este término refiere a un elemento de un sistema de comunicaciones (receptor o transmisor) en el cual parte del procesamiento que habitualmente se realizaba de manera analógica (filtrado, conversión de frecuencia, demodulación) se implementa de manera digital mediante un software operando en una PC o sistema embebido.

Las actividades del laboratorio se realizarán en el tiempo y lugar elegidos por los alumnos ya que se entregará un Dongle USB a cada uno. Para la calificación del Laboratorio cada estudiante deberá entregar un informe que detalle el proceso de resolución de las tareas del laboratorio junto con los resultados obtenidos.

2 Objetivos

Los objetivos de este laboratorio, vinculados con lo que concebimos como parte de las cualidades de un buen profesional de la ingeniería y/o con los objetivos de formación de nuestra asignatura son:

- Presentar un caso de procesamiento digital aplicado a las comunicaciones (el procesamiento digital tiene mucho para aportar a las comunicaciones, saber de temas de comunicaciones es muy útil para hacer procesamiento digital).
- Analizar las ventajas que ofrece, así como las dificultades que presenta la implementación de receptores SDR.
- Vincular diferentes temáticas estudiadas en la materia: sistemas de modulación lineal y exponencial, modelo pasa-banda de señales, densidad espectral de potencia.
- Fomentar la utilización de herramientas de procesamiento para complementar la formación en los temas incluidos en la currícula.
- Fortalecer las habilidades para presentación de informes y resultados.

3 Breve descripción del Dongle SDR

El Dongle se encuentra formado por dos integrados principales:

- Un sintonizador de señales **R820T2**: Puede sintonizar un rango de frecuencias de 24 MHz a 1700MHz. Posee una figura de ruido de 3.5 dB y consume alrededor de 180 mA.
- Un DSP y controlador USB **RTL2832U**: A pesar de que la hoja de datos de este chip no es accesible para usuarios convencionales, se sabe que posee un ADC y un DSP que realiza la conversión de frecuencia intermedia a banda base a través de mezcladores en fase y cuadratura (I/Q), el filtrado pasa bajos, el re-muestreo en fase y cuadratura y el envío de dichas muestras por el puerto USB.

En la Fig. 1 se muestra el circuito impreso del Dongle junto a un diagrama esquemático de conexión.

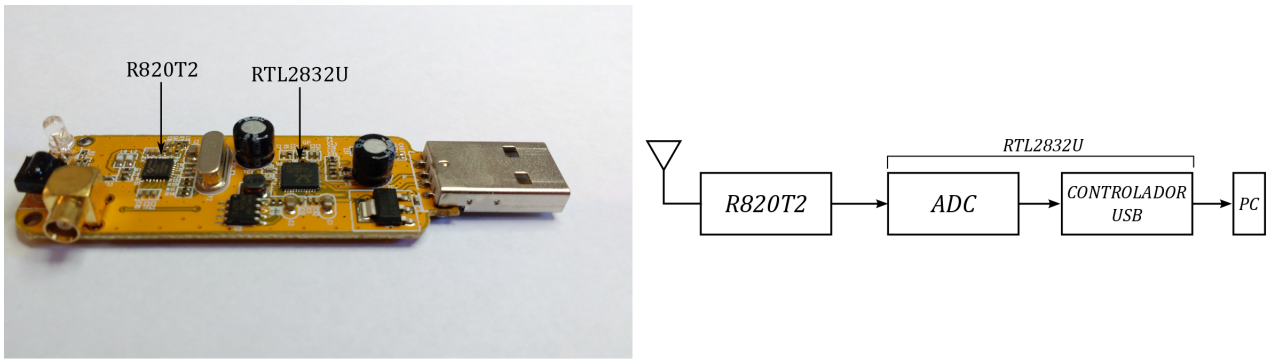

Fig. 1: PCB del Dongle. Diagrama en bloques simplificado.

4 Ejercicios

4.1 Primera vuelta

Este ejercicio será para introducir nociones básicas del Dongle. Se utilizará el software **SDR#**, disponible en la web de la cátedra. La idea será sintonizar algunas estaciones comerciales de FM y notar algunas características básicas de las mismas.

1. Antes de comenzar, instale los drivers del Dongle utilizando el programa **zadig.exe** como se explica a continuación:
   (a) Conectar el dongle a algún puerto USB.
   (b) Ejecutar **zadig_ 2.2.exe** .
   (c) Ir a **opciones**, **list all devices**.
   (d) Aparecerá listado **RTL2838UHIDIR**, o bajo el nombre de **Bulk-In, Interface**. Seleccionar al lado de la flecha verde **WinUSB**.
   (e) Finalmente **Replace Driver**.

   Ahora abra el SDR# teniendo el Dongle conectado a la PC y conectese utilizando la opción RTL-SDR(USB). Debería ver un gráfico de la DEP estimada en pantalla en tiempo real.
2. Haga un barrido por el espectro, desde el comienzo de la banda de FM comercial hasta el final, dando saltos de a 100 kHz. Las distintas estaciones de radio deberían distinguirse a simple vista. Coloque la opción WFM en la pestaña radio y elija las estaciones que posean mayor potencia. Utilice los parlantes o auriculares para escuchar.

Al elegir la opción WFM, el programa comienza a ejecutar un código que demodula la señal de FM comercial. Por defecto, supone a las estaciones con un ancho de banda de 250 kHz.

3. Note que puede modificar el ancho de banda de la señal colocando el cursor sobre el límite del ancho de banda del filtro actual haciendo click y arrastrando. Verifique con los auriculares que al achicar el ancho del filtro empieza a "escucharse mal".

A su vez, tanto el chip que hace de sintonizador como el chip que muestrea y controla el USB posee ganancia regulable y hasta control automático de ganancia. Desde el SDR# (ícono de engranaje) es posible controlar los mismos.

4. Verifique las diferencias en el espectro si se activa/desactiva el control automático de ganancia. Cambie los valores de ganancia manualmente y observe los cambios. Quizás note que a veces "desaparecen" algunas estaciones.

Escriba una observación o conclusión de cada punto anterior, adjunte fotos si lo cree necesario.

5 SDR + Matlab

Ahora se utilizará al Dongle como proveedor de muestras para procesar digitalmente en Matlab. Para ello, obtendremos un registro de muestras de un largo aproximado de 10 segundos de una estación de radio conocida y la tarea será escuchar el audio proveniente de la misma.

Para llegar al resultado final de este ejercicio, que es escuchar el audio, conviene construirse algunas funciones que sirvan como herramientas para ir depurando y visualizando los resultados y avances.

Tener en cuenta que para poder utilizar el Dongle desde Matlab deberá seguir los pasos indicados en la guía [1] disponible en la web de la cátedra. En caso de que no pueda instalar el driver en Matlab por alguna razón, puede seguir las instrucciones de [2] para instalar la librería rtl-sdr que permite obtener un registro de muestras de largo querido, guardarlas en un archivo y luego procesarlas en Matlab o similar.

1. A modo de ejemplo, explicaremos cómo generar un vector de muestras de 15 segundos de duración, tomadas a una frecuencia de muestreo de 2.048 MHz provenientes de una estación en 103.7 MHz utilizando el driver de Matlab. Para ello, crearemos un objeto en Matlab que posee la configuración deseada para nuestro Dongle:

```
fs = 2.048e6; % Frecuencia de muestreo
fc = 103.7e6; % Frecuencia de la estación de radio deseada
spf = 256*64; % Muestras por frame. Este valor es adecuado.

hSDR = comm.SDRRTLReceiver('CenterFrequency', fc,...
                           'SampleRate', fs,...
                           'OutputDataType','double',...
                           'SamplesPerFrame',spf,...
                           'FrequencyCorrection',0);
```

2. Una vez creado el objeto (**hSDR**), podemos capturar un vector de duración 15 segundos con el siguiente código:

```
dur = 15;                    % Duración en segundos de la toma de muestras
frames = floor(dur*fs/spf);  % cantidad de frames para
                             % obtener un vector de dur segundos.
hLogger = dsp.SignalSink;
for counter = 1:frames
  data = step(hSDR);         % Toma un frame de muestras del Dongle
  step(hLogger, data);       % Concatena los frames en un logger
end

x = hLogger.Buffer;          % Vector de muestras (complejo)
```

   El vector x posee muestras en fase y cuadratura en banda base de la señal centrada en 103.7 MHz.

3. Ahora que tiene el vector de muestras, grafique el espectro a ver qué tal se ve. Contemplando que a lo largo del trabajo habrá muchas señales a las cuáles se requiere observar el espectro, es sumamente útil construirse una función que realice la estimación del mismo y grafique el resultado detalladamente. Construya una función que realice la estimación de la DEP utilizando lo aprendido durante el TP1 de la materia. Puede consultar el apunte [3] que se encuentra en la web de la Cátedra.

4. Si en el gráfico anterior pudo distinguir más de una estación, realice un filtrado conveniente para quedarse con la estación que eligió en un principio. Para ello puede utilizar un filtro tipo butterworth de orden 5. Tenga en cuenta el ancho de banda de Carsson para FM comercial para realizar el filtrado. Utilice

```
>> help butter
```

   para aprender a generar un filtro digital conveniente y utilice

```
>> help filter
```

   para realizar el filtrado. Para convencerse que el filtrado fue exitoso, grafique el espectro de la señal filtrada utilizando la función DEP que diseñó previamente.

5. Con el fin de escuchar el mensaje modulado en la señal adquirida debemos construir un demodulador digital. Un esquema de un demodulador típico se muestra en la Fig. 2.

   Siguiendo el diagrama de la Fig. 2, lo primero será generar un vector x de N muestras a una tasa f_s de una estación de radio FM centrada en f_c, como se explicó en el ejemplo anterior.

   Debido a que en general para la frecuencia fs por defecto en las muestras hay más de una estación de radio, se debe realizar un filtrado pasa bajos con un ancho de banda acorde B_1 tal que sólo deje pasar la estación querida. Ahora bien, en procesamiento digital es muy importante procesar el mínimo número de muestras posible, para ahorrar

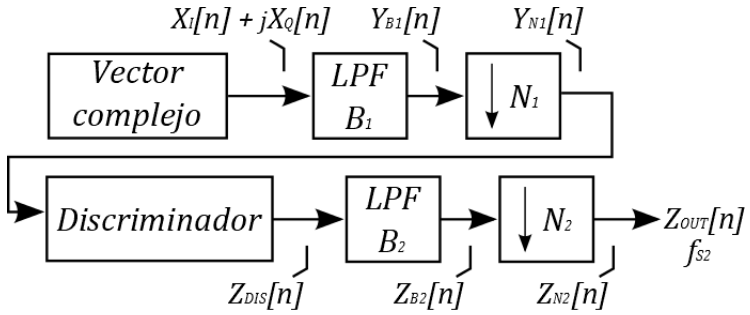

Fig. 2: Esquema de un demodulador de FM por procesamiento digital.

operaciones y tiempo de ejecución. Por ejemplo: si se toman muestras a $f_s = 2.048$ MHz (frecuencia de muestreo por defecto del Dongle) durante 10 segundos, se tienen $N = 20.48 \times 10^6$ muestras. El espectro de estas muestras irá de $-fs/2$ a $fs/2$, es decir de -1.024 MHz a 1.024 MHz. Pero el ancho de banda B_1 será bastante menor que este rango, por lo que podría bajarse la tasa de muestreo en un factor N_1 tal que nuestra señal filtrada quede dentro del rango $-\frac{f_s}{2N_1}$ a $\frac{f_s}{2N_1}$ con la ventaja de que ahora hay que procesar N/N_1 muestras pero no olvidando que hay una *nueva* frecuencia de muestreo $f_{s1} = f_s/N_1$. Esta operación se conoce como **diezmado** y puede ejecutarse utilizando en matlab:

```
>> y_N1 = decimate(y_B1,N1,'fir');
```

El valor de N_1 lo tiene que elegir usted, teniendo en cuenta lo explicado en el párrafo anterior.

Ahora viene la parte más importante del receptor de FM, que es el discriminador de frecuencia. Hay muchos esquemas posibles para realizar la demodulación. El objetivo en principio es hallar la fase de la señal. Sabemos que en dicha fase $\phi[n]$ se encuentra el mensaje "integrado". Por lo tanto, debemos realizar una "derivada" de la fase para obtener el mensaje. Es decir, hay que aproximar:

$$m(t) = \frac{1}{2\pi f_d} \frac{d\phi(t)}{dt} \tag{1}$$

donde f_d es la desviación de frecuencia.

La ventaja de contar con muestras en fase y cuadratura, es que obtener la fase es algo bastante sencillo, ya que es como encontrar la fase de un número complejo o de un fasor[1]. Para aproximar la derivada puede realizar simples diferencias del vector de fase obtenido.

$$m[n] \approx \frac{1}{2\pi f_d} \frac{(\phi[n] - \phi[n-1])}{T} \tag{2}$$

donde T es el tiempo entre muestras de la secuencia $\phi[n]$. Para realizar la aproximación de la derivada puede utilizar la función **diff** de Matlab:

```
>> help diff
```

y tenga en cuenta que para realizar la derivada de una función de fase "continua", debe sumar o restar 2π para que no haya saltos de ciclo antes de derivar. La función **unwrap** evita los saltos en un vector de fase.

```
>> help unwrap
```

En este paso, obtendríamos la señal marcada en la Fig. 2 como $Z_{dis}[n]$. Dibuje el espectro de esta señal a ver qué contenidos frecuenciales ve. Debería obtener un espectro de "audio". Dependiendo la estación de radio que haya elegido, puede que vea un tono (piloto) en 19 kHZ. ¿A qué se debe? Busque en internet. ¿Para qué es utilizado este piloto?

Finalmente, es posible realizar un filtrado más apropiado para nuestra señal $Z_{dis}[n]$ eligiendo un filtro pasa bajos de ancho B_2. Para verificar el correcto funcionamiento del algoritmo hasta aquí, debería poder escuchar el audio

[1]Recuerde como hallaba la fase de un fasor. Para un vector complejo es exactamente igual.

obtenido utilizando la función **sound** de Matlab. El problema que puede presentarse aquí es que las placas de sonido no reproducen cualquier frecuencia de muestreo, pero se sabe que funcionan bien para frecuencias cercanas a 48 kHz. Por lo tanto, quizás necesite hacer un segundo diezmado por un factor de N_2. Ahora sí, escuche el mensaje! Quizás sea bueno normalizar el mensaje por su máximo para que la amplitud máxima sea 1 antes de enviar la señal a la placa de sonido.

```
>> sound(Z_out,f_s2);
```

Deberá empaquetar todo este receptor de la Fig. 2 en una función de Matlab como la siguiente:

```
function [z_out, z_N2, z_B2, z_dis, y_N1, y_B1] = FM_DEMOD_ApellidoAlumno(x, B1, N1, B2, N2, fs)


    % [z_out, z_N2, z_B2, z_dis, y_N1, y_B1] = FM_DEMOD_ApellidoAlumno(x, B1, N1, B2, N2, fs)
    %
    % INPUTS:
    %
    %    x  = señal de entrada
    %    B1 = ancho 3dB filtro pasa bajos de la estacion de radio.
    %    N1 = valor de diezmado luego del primer filtrado pasa bajos.
    %    B2 = ancho 3dB filtro pasa bajos de la señal pasada por el discr.
    %    N2 = valor de diezmado luego del segundo filtrado pasa bajos.
    %    fs = frecuencia de muestreo de x.
    %
    % OUTPUTS:
    %
    %   y_B1 = señal luego del primer filtro pasa bajos.
    %   y_N1 = señal y_B1 diezmada en N1.
    %   z_dis = señal que sale del discriminador.
    %   z_B2 = z_dis filtrada pasa bajos en un ancho B2.
    %   z_N2 = señal z_B2 diesmada en N2.
    %   z_out = z_N2.

        % Código a completar por usted!
end
```

acompañada por un script que le de valores a los parámetros de entrada y la llame.

6 Entregables obligatorios:

- **Dongles:** A través de los canales de comunicación de la cátedra se anunciará la fecha acordada para la devolución de los dispositivos.
- **Informes: (fecha límite 14/11/23)**
  (a) Deberá entregar a través de la plataforma moodle un archivo **.zip** con el informe en formato pdf y el código hecho por usted y funcionando[2].

7 Entregables opcionales: (fecha límite 06/02/24)

Estos entregables no son obligatorios, pero se alienta a que lo resuelva para profundizar los conceptos y lograr una implementación más completa a nivel práctico.

- Realice una función que demodule una estación de FM en tiempo real y permita escuchar el audio hasta que un usuario pulse una tecla. Puede presentarlo con una interfaz gráfica. Tenga en cuenta que la frecuencia que se pretende sintonizar debe ser un parámetro de entrada.

[2]El trabajo es personal. En caso de detectar copias, los trabajos serán desaprobados.

Bibliografía

[1] Instalación Driver RTL-SDR para Matlab. *Web de la cátedra.*

[2] Instalación de Drivers Dongle SDR 820T2. *Web de la cátedra.*

[3] Estimación práctica de DEP. *Cátedra de Comunicaciones 2018.*

[4] Lab 6 Software Defined Radio and the RTL-SDR USB Dongle. *ECE 4670 Spring 2014.*