

INTRODUCCIÓN AL PROCESAMIENTO DE SEÑALES - AÑO 2022

Práctica con Utilitario 1

Introducción

En esta materia se ha puesto énfasis en la utilización de la herramienta informática Matlab u Octave como complemento del espacio de práctica, considerando que en primer término la misma brinda grandes ventajas en el aprendizaje de los temas de la materia, y en segundo término el manejo de la herramienta en sí misma es de provecho para un profesional vinculado al procesamiento de señales. Por otra parte, consideramos que la capacidad de presentar resultados de una manera ordenada, para que puedan ser interpretados por un tercero no debe ser ajena a la formación de todo profesional, y mucho menos de un profesional de la Ingeniería.

Este trabajo con utilitario se enfoca en la utilización de Matlab u Octave como herramienta para el análisis y descripción de sistemas discretos, en particular de tipo lineal e invariante al desplazamiento (o en el tiempo).

Para ello se provee una serie de funciones en archivos de tipo .m. Para ver la ayuda de los mismos, en la ventana de comandos se debe escribir `help nombreArchivo`. Los archivos provistos son:

- `senal.m`: Señal del Ejercicio 1, y sus correspondientes instantes. Para ver los detalles, en la ventana de comando escribir `help senial`
- `hcanald.m`: Respuesta impulsional del canal discreto del Ejercicio 2. Para ver los detalles, en la ventana de comando escribir `help hcanald`

Lxs estudiantes deben resolver las consignas planteadas y entregar:

- Un breve informe con los procedimientos utilizados (por ejemplo listado de comandos), los resultados conseguidos (por ejemplo gráficas de las señales) y las conclusiones obtenidas. El informe debe tener el lenguaje y la estructura de un documento de entrega, por lo que debe dedicarse atención al formato y a la presentación del mismo. La entrega deberá hacerse en formato digital, **respetando la fecha límite**. Ver condiciones de entrega debajo.
- **TODOS** los scripts de Matlab utilizados en la resolución, de manera que los resultados obtenidos puedan ser reproducidos por un tercero. Ver condiciones de entrega debajo.

Condiciones de Entrega

El trabajo puede realizarse de manera individual o de grupos de hasta dos estudiantes. En caso de detectarse copias los trabajos serán **desaprobados**.

La fecha límite de entrega es el día **10 de Noviembre de 2022 a las 17 hs**. El trabajo será corregido para el día 22 de Noviembre de 2022. En caso de existir correcciones, la fecha límite para la entrega del trabajo corregido es el **29 de Noviembre de 2022 a las 17 hs**.

La entrega del informe y los scripts de Matlab debe realizarse en la plataforma Moodle, subiendo dos archivos:

1. El informe en formato PDF. El nombre del archivo debe respetar lo siguiente: **IPS-pu1-APELLIDO.pdf** si el informe se realiza de forma individual e **IPS-pu1-APELLIDO1-APELLIDO2.pdf** si se realiza entre dos.
2. Un archivo comprimido (.zip o .7z) que contenga todos los scripts de MATLAB utilizados. Debe haber un script principal para que el ayudante que corrija el informe pueda ejecutar todos y cada uno de los incisos del laboratorio de manera simple y ordenada. El nombre del archivo debe respetar lo siguiente: **IPS-pu1-scripts-APELLIDO.zip** si el informe se realiza de forma individual e **IPS-pu1-scripts-APELLIDO1-APELLIDO2.pdf** si se realiza entre dos.

Se creará una tarea dentro del aula virtual donde cada estudiante deberá subir individualmente los archivos.

Entrega de Correcciones

En caso de ser necesarias correcciones en el informe y/o en los scripts, las versiones corregidas deberán subirse en una nueva tarea, respetando el formato en los nombres de los archivos ya consignado pero agregando al final de cada uno **_Corregido**. Los archivos correspondientes a la primera entrega deberán permanecer en la plataforma, para que los docentes puedan contrastarlos con las nuevas versiones y verificar las correcciones pertinentes. Cabe destacar que esta segunda entrega no tendrá devolución, por lo que la nota definitiva del laboratorio se dará en esa instancia.

Consejos

Para insertar gráficos hechos con Matlab en un editor de textos, un método rápido y relativamente seguro es el siguiente:

1. En Matlab ingrese el comando `print -f1 -dpng figurita`, esto generará en el directorio actual el archivo gráfico `figurita.png`. El número que sigue a la opción `-f` es el número de la figura de Matlab que se exportará.
2. Desde el editor de texto que utilice, inserte la figura. En general es conveniente que la figura no flote sobre el texto.

Ejercicio 1

La función `senal.m` retorna una señal de variable independiente discreta, y los instantes n en los que está definida la misma.

1. Grafique esta señal. Obtenga la TFTD de la misma y gráfiquela. ¿Qué componentes de frecuencia identifica?

Se tienen cuatro sistemas descritos por las siguientes ecuaciones en diferencias

$$\begin{aligned}
 y[n] &= \frac{1}{2}x[n] + \frac{1}{2}x[n-1] \\
 y[n] &= \frac{1}{2}x[n] - \frac{1}{2}x[n-1] \\
 y[n] &= \frac{1}{4}x[n] + \frac{1}{4}x[n-1] + \frac{1}{2}y[n-1] \\
 y[n] &= \frac{1}{4}x[n] - \frac{1}{4}x[n-1] - \frac{1}{2}y[n-1]
 \end{aligned}$$

Con los cuales se quiere procesar la señal anterior.

2. Obtenga la respuesta impulsional y la respuesta en frecuencia de cada uno de los sistemas (de manera analítica). Gráfiquelas. ¿De qué tipo de sistemas se trata?

3. Codifique funciones que implementen cada uno de los sistemas. Como ejemplo, puede basarse en la función **tri1** que se muestra en el TP 1. Verifique que la respuesta impulsional de cada uno de los sistemas así implementados coincide con la obtenida analíticamente. Verifique que la TFTD de la respuesta impulsional coincida con la respuesta en frecuencia obtenida analíticamente.
4. Utilizando sus implementaciones del inciso anterior, obtenga la señal de salida para cada sistema, cuando se aplica a la entrada la señal del inciso 1. Grafique en cada caso.
5. Obtenga la TFTD de las salidas. ¿Qué componentes de frecuencia identifica? ¿Qué conclusiones puede sacar?

Ejercicio 2

La señal $h_A[n]$ retornada por la función **hcanald.m** representa la respuesta impulsional de un sistema SLID FIR que modela el comportamiento de un canal digital con dos ecos.

1. Grafique esta respuesta al impulso e interprete en qué consiste el modelo mencionado. Escriba la ecuación en diferencias que representa el sistema. ¿Qué puede decir acerca de la estabilidad del sistema?
2. Escriba en MATLAB una función que implemente un sistema con esta ecuación en diferencias. Verifique que este sistema tiene la misma respuesta impulsional que la obtenida en 1.

Por otra parte, el comando¹

```
[x, fs] = audioread('audio.wav');
```

permite cargar un archivo de audio (en este caso, el provisto entre los archivos de la presente práctica) en la variable **x**. La variable **fs** indica la frecuencia de muestreo de la señal, concepto que estudiaremos más adelante. Como cualquier otra variable, se puede graficar utilizando el comando **plot(x)**. Es posible escuchar la señal mediante el comando:

```
sound(x,fs);
```

3. Usando la función escrita en 2. halle la salida del sistema (que llamaremos **y**) cuando se aplica a su entrada la señal de audio **x**. ¿Cómo se escucha esta señal? Grafique la señal de entrada y de salida en el tiempo.

Una forma de intentar corregir el efecto que introduce el canal consiste en colocar un filtro FIR en cascada con el canal intentando que la respuesta impulsional total se aproxime lo más posible a una única delta de Kronecker. ¿Por qué queríamos lograr esto? Por ejemplo, si la respuesta impulsional del canal fuese $h_c[n] = \delta[n] + \alpha_1 \delta[n - n_0]$, un filtro con una sola rama de retardo sería de la forma $y[n] = x[n] - \alpha_1 x[n - n_0]$. ¿Qué efecto tiene este sistema? ¿Por qué la corrección no es perfecta? ¿Cómo añadiría una segunda rama de retardos que mejore esto?

4. Con esta idea implemente un sistema que intente corregir el efecto del canal. Pruebe con 2 y 3 ramas de retardo. Explícite la ecuación en diferencias. En cada caso grafique la respuesta impulsional de la cascada (canal + filtro).
5. Filtre la señal y con estos sistemas y escuche como suena. Compare cada una de estas señales con la señal **x** original.

¹En versiones antiguas de Matlab, no existe el comando “**audioread**”. En su lugar, es posible utilizar el comando:
`[x, fs] = wavread('audio.wav');`