

Laboratorio de Procesamiento Digital de Voz

Practica 1.

INTRODUCCION DE MATLAB Y MANEJO DE ARCHIVOS DE VOZ

Objetivo: Conocer las formas principales de archivar información correspondiente a señales de voz. Convertir los distintos formatos de archivos para ser usados en diversas aplicaciones de análisis.

Introducción a Matlab.

Matlab, Matrix laboratory, es un software con el cual se pueden realizar una gran cantidad de operaciones matemáticas; se pueden realizar desde operaciones sencillas como suma, resta, multiplicación, etc., hasta aplicar funciones para el manejo de filtros, transformada de Fourier, Procesamiento de Imágenes, Procesamiento de Voz, etc.

Por ejemplo, se puede calcular una sencilla multiplicación de la siguiente forma

```
>> 5*3
ans =
    15
```

o se puede calcular la transformada de Fourier de una señal $x(n)$ dada, utilizando el siguiente comando

```
>> y = fft(x)
y =
1.0e+002 *
1.27225135155731
0.02526963248074 - 0.05726891721053i
0.01410269133890 + 0.04244480168186i
-0.00209553328642 + 0.01159136563156i
...
```

En Matlab no es necesario declarar el tipo de variables, solo es necesario inicializar la variable con su valor y el tipo es automáticamente asignado. Por ejemplo:

```
>> y = 4
>> m = -3.4
>> str = 'Hola Mundo'
```

Todas las variables son almacenadas en el *espacio de trabajo base (workspace base)*, por lo que después de declarar una variable se puede hacer uso de su valor en cálculos posteriores.

Nota: Matlab crea un *espacio de trabajo* independiente para las llamadas a las funciones por los que las variables inicializadas en el *espacio de trabajo base* no se pueden visualizar dentro de una función (solo con funciones apropiadas por ej: *evalin*). Para mayor información consulte la ayuda de Matlab.

Para visualizar las variables de entorno en el espacio de trabajo actual, se utiliza el comando *who*. Por ejemplo, el siguiente comando muestra las variables de entorno en el espacio de trabajo base.

```
>> who
Your variables are:
ans h x y
```

En este caso se tienen tres variables (h , x , y), la palabra reservada *ans* se toma como otra variable que almacenan el último resultado de una operación sin asignar a una variable, por ejemplo cuando multiplicamos

```
>>5*3
ans = 15.
```

Las variables numéricas se pueden representar mediante escalares, vectores o matrices. De hecho, Matlab considera a las dos primeras también como matrices, e incluso también a las cadenas. A continuación se muestra la forma de asignar estos valores:

```
>> s = 4 %escalar
>> v = [-3.4, 2+i, 3, 1.2-0.2i] %Vector
>> m = [4, 3, 1; 3, 1.2, 0.2i] %Matriz
```

Matlab, tiene una sintaxis muy parecida a la de los lenguajes de programación conocidos, por lo que su uso es muy fácil de entender una vez conocido otro lenguaje. Además de las capacidades “normales” de los lenguajes de programación estándar, Matlab tiene operadores, funciones y recursos apropiados para realizar cálculos científicos. Por ejemplo, para asignar un vector a una variable y graficarlo, con solo dos instrucciones se puede hacer:

```
>> v = [-3.4, 2, 3, 1.2,-0.2];
>> plot(v)
```

o, si desea sacar el logaritmo a este vector se escriben los siguientes comandos

```
>> v = [-3.4, 2, 3, 1.2,-0.2];
>> plot(log(v))
```

Una de las ventajas de Matlab es el poder escribir un programa en un editor de texto y después correrlo en el *prompt* de la ventana de comandos. Para poder hacer esto debe seguir los siguientes pasos:

1. Abra el editor de Matlab, que viene incluido con la versión de Matlab. Se recomienda porque además de ser un editor también cuenta con un *debugger*, sin embargo, se puede utilizar cualquier otro editor de textos como: el *bloc* de notas o el editor de MSDOS.
2. Escriba el programa correspondiente, como ejemplo puede ser el siguiente código:

```
x = [-3.4, 2, 3, 1.2,-0.2];
y = 1:5
z = x .* y
plot(z)
```

3. Guarde el programa con extensión *.m* (en el editor de Matlab es la extensión por *default*). Hay que cuidar que el nombre no contenga espacios ni caracteres extraños. Recuerde que el *bloc* de notas de Windows agrega la extensión *txt* al final de cada documento que se guarda, por lo que el nombre del archivo junto con la extensión se deberá teclear entre comillas dobles (“ ”), de lo contrario será necesario corregirlo.
4. Una vez almacenado el programa, regrese a la ventana de comandos de Matlab, cambie de directorio de trabajo con el comando *cd*, de la misma forma que MSDOS o utilizando el icono de cambio de directorio (este último se encuentra disponible solo en versiones de Matlab superiores a la 6). Por último ejecute el programa escribiendo el nombre del archivo sin extensión.

Uno de los comandos mas usados es el comando de ayuda *help*, la forma de usar este comando es: se escribe el comando *help*, seguido por el nombre de la función a la que solicita pedir información. Por ejemplo: para solicitar la ayuda de la función *sinc*, se escribe.

```
>> help sinc
```

En respuesta a dicho comando se despliega lo siguiente:

```
SINC Sin(pi*x)/(pi*x) function.  
SINC(X) returns a matrix whose elements are the sinc of the elements  
of X, i.e.  
    y = sin(pi*x)/(pi*x)    if x ~= 0  
      = 1                  if x == 0  
...
```

Nota: Observe que Matlab en la ayuda despliega el nombre de la función en mayúsculas para resaltar su uso, sin embargo, se debe tomar en cuenta que la sintaxis de todas las funciones definidas en Matlab es con minúsculas.

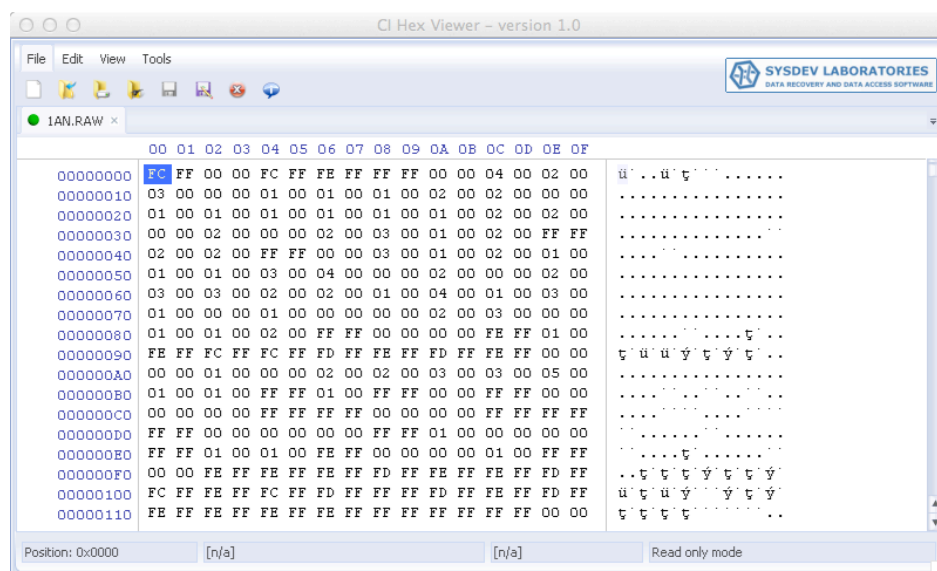
Matlab utiliza las sentencias de control tradicionales como `if`, `while`, `switch`, `for`, etc. El uso de estas funciones es muy similar a como se da en los lenguajes de programación convencionales, por lo que no es necesario describirlos, para obtener mayor información acerca de ellos utilice el comando de ayuda de Matlab.

También es posible usar el comando `doc`, seguido del nombre de la función o comando del que se desee obtener mayor información. Este comando abre una ventana que muestra la información dada en la ayuda junto con información más detallada acerca del comando que se quiere consultar.

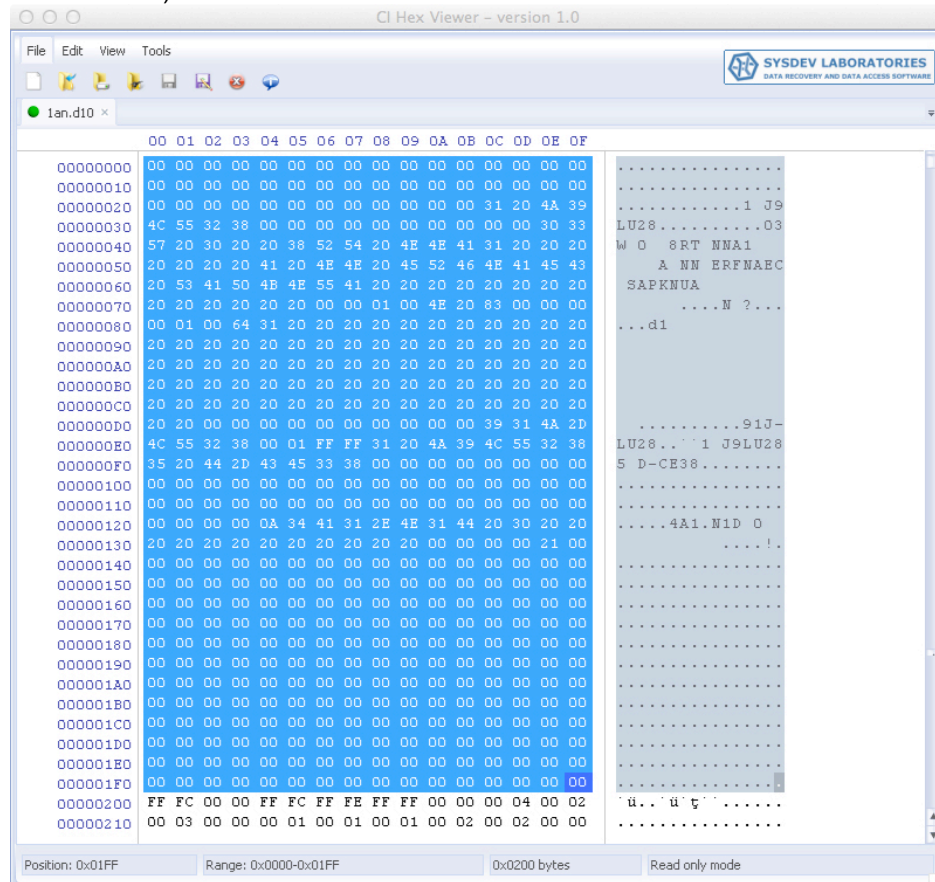
Manejo de archivos de voz.

Conceptos básicos: En las prácticas del laboratorio se maneja diferentes tipos de archivos como son: los archivos con formato RAW, archivos en código ASCII, archivos con formato RSM y archivos con formato WAV, todos ellos son archivos de sonido utilizados por diversos sistemas y programas.

1. *Archivos RAW:* La información proveniente de un cuantizador de 16 bits con formato little endian es almacenada en este tipo de archivos. Los datos se guardan sin encabezado y en un formato de números enteros en binario, cuya longitud es de 16 bits. Por ejemplo, la siguiente figura muestra el contenido del archivo *1an.raw* en forma binaria (el primer valor seleccionado corresponde al primer valor de la señal y así sucesivamente).



2. **Formato RSM o D10:** Las aplicaciones en *DSP's* de *Texas Instruments*, utilizan este tipo de formato. La información que se almacena en este archivo, consta de un encabezado de 512 bytes, seguido por muestras de un cuantizador de 16 bits con formato *Big-Endian*. La siguiente figura muestra el archivo *1an.d10* en formato binario en donde los primeros 512 (seleccionados) son el encabezado.



3. **Formato ASC:** Este formato es utilizado por Matlab. Son archivos de texto donde se almacenan las muestras de audio en forma secuencial, estos archivos no tienen encabezado y los números son representados como números arábigos en formato ASCII. Este archivo se puede observar en cualquier editor de texto.
4. **Formato WAV:** Es el formato de archivos de audio creado para la versión Windows 3.1 y posteriores, este formato consta de un encabezado seguido por las muestras de la señal, las cuales con ayuda de una aplicación Windows pueden variar en la forma en que se almacenan.