

El archivo de plantilla incluye el nombre de las etiquetas de atributo, luego se incluye la letra C indicando que son datos de caracteres, si fuesen numéricos sería N, luego tres dígitos del ancho del campo (número de caracteres) y luego tres dígitos de lugares decimales en el campo si es numérico.

El archivo de atributos es una lista que contiene los atributos extraídos y que serán procesados por el programa de lenguaje C llamado HACERLP que generará un archivo de listado de partes.

Finalmente con la ejecución del programa de AutoLisp SUMLP se procede a crear el cuadro de lista de partes que será automáticamente incluido en el plano que hemos diseñado.

3. Conclusiones

Las ventajas que presenta esta técnica de D.A.C. sobre las técnicas tradicionales se pueden resumir como sigue:

- Homogeneidad en el diseño, en virtud del uso de símbolos normalizados en bloques.
- Pulcritud, al no existir roce del papel con herramientas de diseño se reducen las manchas sobre el papel.
- Rapidez, pues se optimiza la corrección de los diseños previamente realizados.
- Obtención rápida y eficiente de estadísticas del diseño. Es decir que podemos conocer el número de inserciones de un bloque en particular en un diseño.
- Generación automática de cuadros de listas de partes.

Referencias

- [1] Leigh, R. W.; AutoCAD. A Concise Guide to Commands and Features; 3ra ed.; Ed. Ventana Press, E.E.U.U. 1993; pp.447.
- [2] Shigley, J.E. y Mitchell, L.D.; Diseño en Ingeniería Mecánica; 5ta ed.; Ed. McGraw Hill, México 1997; pp.915.

MATLAB: La Herramienta

Ing. Fernando CASTILLO BALBOA, FIM, UTP

Para trabajar con el entorno MATLAB se requieren de conocimientos básicos de win95/98, programación y algo de creatividad.

¿Qué es MATLAB?

MATLAB es un lenguaje de alto rendimiento para cálculos técnicos. Este integra cálculos, visualización, y programación de un ambiente sencillo, donde los problemas y las soluciones se expresan en una notación matemática familiar y sencilla.

¿Cuáles son los usos que puedo darle a MATLAB?

A decir verdad MATLAB puede utilizarse para casi cualquier caso o problema que necesite de cálculos. Se puede utilizar para desarrollo de algoritmos, modelado, simulación y desarrollo de prototipos, análisis de datos, exploración y visualización, para graficar ensayos científicos e ingenieriles, para desarrollo de aplicaciones las cuales incluyen el desarrollo de interfaces gráficas para el usuario. (GUI's)

MATLAB: Entorno de trabajo

MATLAB provee una serie de instrucciones o herramientas que le facilita la labor al usuario. Provee herramientas que le facilitan la administración de las variables de su área de trabajo y además para la importación y exportación de datos. Además provee de herramientas para el desarrollo, manejo, corrección y enmarcar los archivos M, los cuales con la base de la aplicación MATLAB.

MATLAB viene con una extensa librería. Esta está compuesta de algoritmos computacionales que van desde los comandos sencillos de *sum*, *sine*, *cosine*, hasta funciones de aritmética compleja, y funciones sofisticadas como la inversa de una matriz, los valores característicos,

las funciones de Bessel y las transformadas rápidas de Fourier.

MATLAB: Primeros comandos

Escritura de matrices: MATLAB utiliza extensivamente los arreglos matriciales para sus cálculos. Esto se debe a que este fue desarrollado primordialmente para simplificar los cálculos de álgebra lineal que aparecen en muchas aplicaciones. La multiplicación matricial se denota por '*'. La escritura de una matriz en MATLAB es:

$$m = [1 \ 2 \ -4; \ 2 \ 1 \ -4; \ 1 \ -2 \ 3]$$

Una vez definida la matriz, se puede calcular su inversa por medio del comando *inv(m)*; su determinante *det(m)*; sus valores característicos *eig(m)* [utilizados extensivamente para calcular los modos normales de oscilación de un sistema dinámico], y así otros comandos como *diag(m)* que extrae la diagonal principal de la matriz *m*.

Otra forma para cargar en memoria una matriz de dimensiones muy grandes, es utilizando archivos precargados con los datos de la matriz. El comando *load* lee archivos binarios que contiene los elementos de la matriz previamente almacenados en secciones anteriores o lee archivos de formato texto, los cuales contienen los datos numéricos. En estos últimos la información debe organizarse como una tabla de datos de forma rectangular, separado por espacios en blanco o comas; cada línea representa una fila de la matriz y además debe haber la misma cantidad de elementos en cada fila. Como un ejemplo, almacene la matriz siguiente en un archivo llamado *matriz.dat*

7.2	3.3	6.4	9.0
4.0	1.0	2.4	-4.0
9.1	2.2	3.0	1.2
1.4	-1.1	-9.2	7.4

Para cargar la matriz, se emplea el comando
`load matriz.dat`

Este comando crea la variable *matriz* y le asigna los datos. Otra forma es utilizando un *archivo-M*, el cual es un archivo tipo texto que contiene código MATLAB. Como un ejemplo, utilice el programa NOTEPAD que está en las aplicaciones básicas del W95/98. Escriba:

```
m = [ 7.2 3.3 6.4 9.0 4.0 1.1 2.4...
      -4.0 9.1 2.2 3.0 1.2 1.4 -1.1 -9.2 7.4];
```

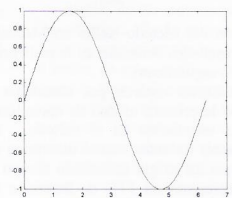
Guárdelo con el nombre *matriz1.m*. En MATLAB al escribir *matriz1* en la línea de comando, lee el archivo *matriz1.m* y crea la variable *m*.

Creando Gráficas

La forma de representar gráficas de MATLAB es bastante sencilla de aprender. El comando *plot* grafica una función predefinida. Veamos un ejemplo:

```
»t=0:pi/10:2*pi;
»f=sin(t);
»plot(f,t)
```

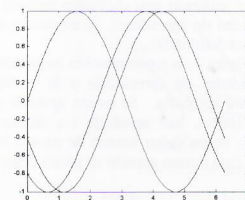
resulta en la grafica siguiente:



Si se quiere agregar más de una función a la figura anterior, se deben definir y posteriormente colocarlas dentro del comando *plot* como sigue:

```
»f2=sin(t-2.22);
»f3=cos(t+2);
»plot(t,f,t,f2,t,f3)
```

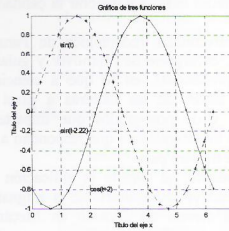
Lo que resultaría en:



Para ponerle el título y las etiquetas de los ejes y además una leyenda para cada función utilizo los comandos:

```
»plot(t,f1,'b:*',t,f2,'m-x',t,f3,'g-+');
»title('Gráfica de tres funciones');
»xlabel('Título del eje x');
»ylabel('Título del eje y');
»text(1.0,-0.2,'sin(t-2.22)');
»text(1.0,0.7,'sin(t)');
»text(2,-0.8,'cos(t+2)');
»grid on;
```

Para la creación de gráficas separadas dentro de una misma figura, MATLAB provee un comando que subdivide la figura en zonas administradas de forma matricial.



El comando en cuestión es:

```
subplot(n,m,p),fplot(func,lims)
```

Aquí *n* y *m* representan la cantidad de filas y columnas de arreglo matricial, y *p* representa la posición dentro de ese arreglo. El argumento *func* dentro del comando *fplot* es una función representada por medio de una sola variable, por ejemplo:

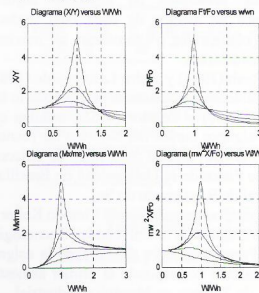
$$ff=2*x^2-2.24*x+2.2$$

El argumento *lims* representa los límites a los cuales la función va a ser evaluada y se representa de la siguiente forma:

$$lims=[y1,yd \ x1,x2]$$

El comando para representar varias funciones sería:

```
»subplot(2,2,1);xoverly
»subplot(2,2,2);ftf0
»subplot(2,2,3);mxmx
»subplot(2,2,4);mwxft0
```



Continuará.....