

PRÁCTICA 1.

Adquisición de datos computerizada en el entorno *MathWorks*.

1. 1. Introducción.

En esta practica diseñaremos un sistema computerizado de adquisición de datos para obtener la característica de salida de transistores de unión de efecto de campo (JFET) para poder optimizar posteriores diseños eléctricos que tengan como base dichos dispositivos.

Las características operativas de los JFETs son hasta cierto punto impredecibles, debido a su dependencia de la geometría interna de cada dispositivo. Por otro lado, considerar incluso pequeñas diferencias en los dispositivos a la hora del diseño puede ser importante. Por ejemplo, cuando se diseñan amplificadores de audio, es crucial determinar exactamente cuál es el rendimiento que cabe esperar para las diferentes entradas del amplificador. Consecuentemente, se deben realizar pruebas sobre los transistores a utilizar para determinar sus características particulares. En dichas pruebas, se miden las corrientes de salida sobre cierto rango de voltajes de entrada. Después, el diseño del sistema completo puede optimizarse a partir de esos datos. Lamentablemente, las pruebas reseñadas son muy tediosas y consumen mucho tiempo, de modo que resulta atractivo disponer de algún sistema que permita automatizar el test.

En esta práctica, vamos a utilizar hardware standard para preparar un sistema computerizado que nos permita obtener las características operacionales de cierto número de JFETs, visualizarlas, compararlas y almacenar los resultados. En primer lugar, prepararemos el montaje hardware para proceder después a escribir el código MATLAB para adquirir y analizar los resultados. Finalmente, comprobaremos el funcionamiento de todo el sistema, comparando la respuesta de varios JFET provenientes de la misma partida de producción. Veremos que, a pesar de estar producidos bajo las mismas circunstancias, son eléctricamente diferentes debido a las pequeñas diferencias en la geometría interna.

Antes de seguir nos interesa recordar algunas cosas sobre las características típicas de los JFET. Tenemos en la **figura 1** la representación esquemática de un dispositivo de este tipo.

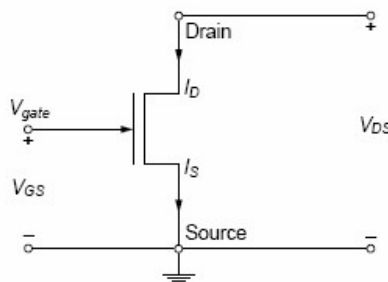


Figura 1. Transistor de canal N de unión de efecto de campo (JFET).

La forma habitual de explicar el funcionamiento de este dispositivo consiste en dibujar la corriente de drenador I_D contra el voltaje drenador-fuente V_{DS} , o sea, la curva característica I_D - V_{DS} . Para generar dicha curva, un voltaje constante se aplica a la puerta-fuente V_{GS} , mientras que se aplica un voltaje variable al drenador-fuente y se mide la corriente I_D . Cuando se incrementa V_{DS} , I_D también crece, como puede verse en la **Figura 2**. Eventualmente, I_D alcanzara un punto de saturación donde se convierte en constante. Dicho punto es una importante característica de un JFET y se denomina

corriente de saturación drenador-fuente I_{dss} . Después de ese punto, incrementar V_{ds} no causará incrementos en I_d , y la curva se queda plana. Para completar la caracterización del JFET, repetiremos dicha operación para valores ligeramente distintos del V_{gs} , alcanzando siempre la saturación de I_d .

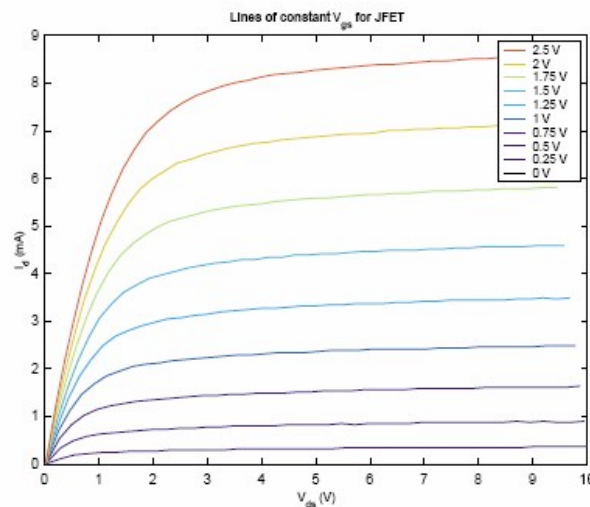


Figura 2. Curva característica típica de un transistor JFET de canal N.

1. 2. Hardware.

El hardware que usaremos en esta practica consta de un PC, una tarjeta de adquisición y ciertos dispositivos de acondicionamiento, además de los propios transistores a caracterizar. En concreto, vamos a estudiar JFETs MPF102 que han sido específicamente diseñados para amplificadores de audio-VHF y aplicaciones de mezclado.

Para obtener todo el rango de salida de los JFETs, necesitamos variar el voltaje drenador-fuente V_{ds} y el voltaje de puerta V_{gs} , sobre los rangos de 0 a 10 V. y -5 a 0 V., respectivamente. Dicha excursión nos proporciona el máximo rango de salida de los dispositivos y nos permite obtener las características relevantes. Como el MPF102 tiene una corriente de puerta de hasta 10 mA, la saturación ocurrirá aproximadamente a ese nivel y debemos ser capaces de proporcionárselo. La tarjeta de adquisición que vamos a usar solo puede proporcionar hasta 1 mA por canal, de modo que tendremos que usar un amplificador de corriente. Utilizando el circuito de la **figura 3**, podremos proporcionar el voltaje necesario para V_{ds} (nodo 2) e inyectar los 10 mA necesarios. El voltaje de puerta V_{gs} y el voltaje de entrada V_{in} se controla con los canales de salida analógica de la tarjeta. Las medidas de corriente se obtienen a partir del voltaje diferencial entre los nodos 1 y 2 dividido entre el valor de la resistencia R_2 .

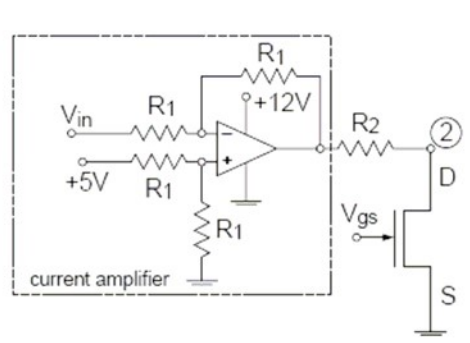


Figura 3. Esquema del circuito de test.

Los valores para R1 y R2 son 47 k y 100 Ω , respectivamente. La tarjeta de adquisición proporciona en una de sus salidas +5 V DC, que pueden utilizarse como entrada para el terminal positivo en el amplificador operacional LM324. La fuente de +12V DC para alimentar el AO puede obtenerse de un convertidor AC/DC.

1. 3. Código de adquisición.

Una de las características de la *Data Acquisition Toolbox* es que está basada en objetos. Esto quiere decir que todas las propiedades asociadas a un mismo dispositivo hardware se recogen en un único objeto que además permite acceder a sus diversos subsistemas de hardware. Los objetos pueden ser de entrada analógica (AI), de salida analógica (AO) y de I/O digital. En esta práctica, utilizaremos objetos AI y AO. El objeto de la salida analógica se utiliza para mandar los correctos voltajes de puerta y fuente desde el PC. Por su lado, el objeto de la entrada analógica recoge el voltaje diferencial de salida del circuito que se convierte posteriormente en una corriente.

Comenzamos entonces creando los objetos de entrada y salida analógica. Para ello, debemos especificar el hardware que estamos utilizando en la realización:

```
ai=analoginput('nidaq',1);
ao=analogoutput('nidaq',1);
```

Ahora necesitamos configurar los objetos para nuestra aplicación. Comenzamos configurando el objeto de entrada. Nuestro montaje necesita de dos entradas, una para la corriente de drenador y otra para el voltaje de drenador. Entonces, añadimos primero dos canales al objeto:

```
ic=addchannel(ai,[1 0]);      % Adding the first two differential
                              % inputs to the input object
```

A continuación, damos el rango de entrada de cada canal y les asignamos nombre y unidades:

```
ic.inputrange=[0 10]; % Setting the sensor and input
ic(1).SensorRange=ic(1).InputRange;      % ranges from 0 to 10 volts
ic(1).UnitsRange=ic(1).InputRange;
ic(1).ChannelName='Drain-Source Voltage';  % Naming channel 2 (1st index)
ic(2).UnitsRange=ic(2).InputRange*10;     % Measuring current indirectly:
ic(2).SensorRange=ic(2).InputRange;       % I = V/R, where R is 100 $\Omega$  and
ic(2).Units='mA';                         % 1000 mA = 1 A
ic(2).ChannelName='Drain-Source Current';  % Naming channel 0 (2nd index)
```

Hemos realizado asimismo la conversión de voltaje a corriente, considerando que $R=100\ \Omega$. Podemos verificar la configuración tecleando ic en la línea de comandos:

»ic

Index:	ChannelName:	HwChannel	InputRange:	SensorRange:	UnitsRange:	Units:
1	'Drain-Source Voltage'	2	[0 10]	[0 10]	[0 10]	'Volts'
2	'Drain Current'	0	[0 10]	[0 10]	[0 100]	'mA'

Ahora, para configurar el objeto de salida:

```
oc=addchannel(ao,[0 1]);      % Adding 2 channels to the output object
oc.UnitsRange=oc(1).OutputRange; % The unit range on all channels will
                                % be the same as the default output
                                % range (-5V to 5V)
```

```
oc(1).ChannelName='Vin'; % Naming the output channels, according
oc(2).ChannelName='Vgs'; % to circuit schematic
```

Finalmente, comprobamos las propiedades del objeto oc:

```
» oc
```

Index:	ChannelName:	HwChannel:	OutputRange:	UnitsRange:	Units:
1	'Vin'	0	[-5 5]	[-5 5]	'Volts'
2	'Vgs'	1	[-5 5]	[-5 5]	'Volts'

1. 4. Código de análisis y representación.

Una vez que hemos configurado nuestro sistema para la adquisición, debemos escribir el código para hacer el barrido en los voltajes drenador-fuente. Lo haremos mediante un lazo *for* anidado. Como estamos usando un amplificador de corriente para generar V_{ds} , utilizaremos V_{in} para generar los distintos valores de interés (ver **figura 3**). Par cada voltaje de puerta, queremos barrer el voltaje de entrada, V_{in} desde -5 a $+5$ voltios en pasos de 0.25 voltios. Finalmente, creamos una matriz de voltajes V_{ds} y corrientes I_d . Un posible código para hacer esto sería:

```
Vds=[]; % Initializing Drain Voltage &
Id=[]; % Drain to Source Current
VgsList = [-4.5:.5:-2 -1.75:.25:0];
for Vgs = VgsList
    data=[];
    for Vin = -5:0.25:5
        putsample(ao,[Vin,Vgs]); % Output drain-source, gate
    voltages
        data=[data; getsample(ai)]; % Acquire vector of all inputs
    end % for Vgs
    Vds = [Vds data(:,1)]; % Setting up voltage matrix
    Id = [Id data(:,2)]; % Setting up current matrix
end % for Vin
```

Como puede verse, utilizamos la función *putsample* para poner los voltajes de salida en el objeto *ai*, mientras que *getsample* adquiere el vector de voltajes de entrada de los canales definidos en el objeto de entrada, *ao*. Como estamos interesados en este caso en los valores permanentes de salida del circuito debido a una entrada constante, utilizamos las funciones de muestra única *putsample* y *getsample* para enviar la entrada y captar la respuesta. Si estuviésemos interesados en analizar la reacción del sistema a una entrada variable en el tiempo, deberíamos usar las funciones de adquisición en buffer *putdata* y *getdata*. Pero como hacemos adquisición estática (valores únicos), en este caso no debemos preocuparnos por la sincronización y la velocidad de adquisición. Según vamos adquiriendo los voltajes, estos se van concatenando para formar unas matrices que contienen los voltajes V_{ds} (primera columna de datos) y la corriente del drenador I_d (segunda columna de datos). Eso simplifica el análisis y representación del resultado de experimento.

Para representar los resultados, utilizamos las rutinas de Matlab. Primero haremos un sencillo diagrama 2-D de los resultados:

```
legendStr={}; % Build up a legend string
for VgsCt=1:length(VgsList),
    legendStr{VgsCt}=[num2str(VgsList(VgsCt)) 'V'];
end
h=plot(Vds,Id);
legend(h(5:end),legendStr(5:end));
xlabel('V_{ds} (V)');
ylabel('I_{d} (mA)');
title('\bf{Lines of constant V_{gs} for JFET}');
```

De donde obtenemos la Figura 4, que responde exactamente a lo que cabía esperar:

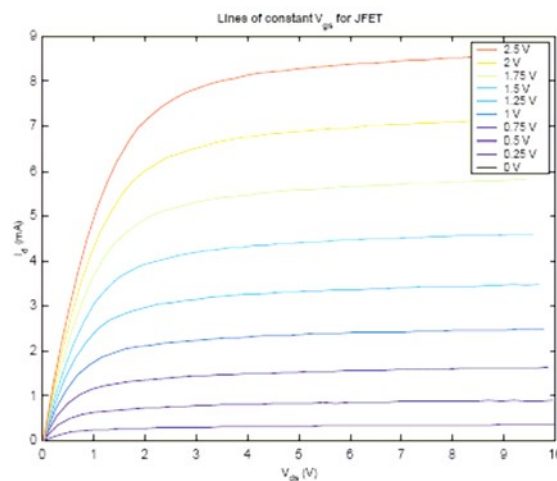


Figura 4. Curva característica I_d versus V_{ds} del JFET MPF102 de canal-n.

Para un análisis más detallado de los resultados, podemos utilizar otras rutinas de Matlab que permiten ver los resultados en tres dimensiones. Para hacerlo, creamos un *array* de voltajes de puerta correspondientes a los voltajes de drenador:

```
Vgs = repmat(VgsList,[length(Vds(:,1)),1]); % Duplicates Vgs by rows to
                                              % have as many rows as Vds
```

De donde se obtienen matrices de igual tamaño para V_{gs} , V_{ds} e I_d (ver **figura 5**), permitiéndonos después generar representaciones en 3D de dichos datos.

```
Vgs =
Columns 1 through 7
-4.5000 -4.0000 -3.5000 -3.0000 -2.5000 -2.0000 -1.7500
-4.5000 -4.0000 -3.5000 -3.0000 -2.5000 -2.0000 -1.7500
-4.5000 -4.0000 -3.5000 -3.0000 -2.5000 -2.0000 -1.7500
-4.5000 -4.0000 -3.5000 -3.0000 -2.5000 -2.0000 -1.7500
-4.5000 -4.0000 -3.5000 -3.0000 -2.5000 -2.0000 -1.7500
...
-1.5000 -1.2500 -1.0000 -0.7500 -0.5000 -0.2500 0
-1.5000 -1.2500 -1.0000 -0.7500 -0.5000 -0.2500 0
-1.5000 -1.2500 -1.0000 -0.7500 -0.5000 -0.2500 0

Name      Size      Bytes      Class
Id         41x14      4592      double array
Vds        41x14      4592      double array
Vgs        41x14      4592      double array
```

Figura 5. Resultado del uso de la función *repmat* para generar matrices de igual dimensión.

Finalmente, usando la función *surf*, generamos una superficie (ver **figura 6**) en 3D de V_{ds} , V_{gs} y I_d , colocándolos en los ejes x, y, z, respectivamente. Asimismo, añadimos etiquetas a los ejes:

```
surf(Vds, Vgs, Id);           % Plot 3-D surface
xlabel('V_{ds} (V)');
ylabel('V_{gs} (V)');
zlabel('I_{d} (mA)');
title('\bf{Operating Characteristics for JFET}');
```

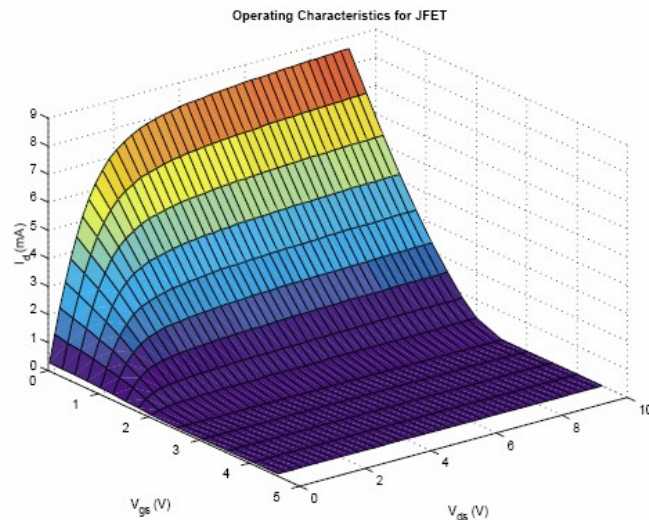


Figura 6. Superficie de la característica I_d versus V_{ds} de los JFET analizados.

Queda por obtener la caracterización de otros dispositivos de la misma partida para hacer las comparativas que establezcan sus diferencias operativas. Para ello, sencillamente debemos reemplazar el JFET en el montaje de la **figura 3** y ejecutan el código de nuevo. Después, comparamos las características obtenidas en ambos casos. Como puede verse, hay diferencias sustanciales en las características operativas. Nuestro sistema nos permitirá tener en cuenta dichas diferencias a la hora de implementar diseños electrónicos con los dispositivos caracterizados.

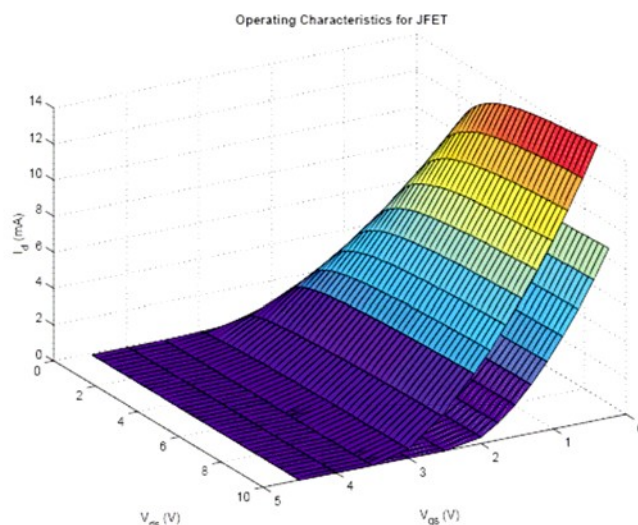


Figura 7. Comparativa de los JFET analizados.