Tema III: Transformación de Laplace

1 Introducción

La transformación de Laplace permite analizar circuitos utilizando las propiedades del dominio *s*. Si se desea expresar el resultado de un análisis en dicho dominio en el habitual dominio temporal, tal resultado ha de ser sometido a la transformación inversa de Laplace.

Los cálculos correspondientes a la transformación de Laplace, tanto directa (formular una expresión temporal en el dominio *s*) como inversa, son relativamente sencillos en términos conceptuales, pero suelen entrañar bastante complejidad matemática. En ellos aparecen números complejos, polinomios para los que la determinación de sus raíces entraña la realización de cálculos iterativos, variaciones con la frecuencia de módulos y fases de funciones, etcétera. De ahí que MatLab haya demostrado ser una herramienta muy adecuada para abordar problemas que implican la realización de tales cálculos. El objetivo de este tema es, precisamente, presentar algunas nociones relativas a la utilización de MatLab en cálculos referidos a la transformación de Laplace.

En su mayor parte los cálculos aludidos en este tema implican el manejo de polinomios. Por ello, antes de hablar explícitamente de la transformación de Laplace, presentaremos una serie de consideraciones referidas al manejo de polinomios en MatLab.

2 Manejo de polinomios en MatLab

Un polinomio (véase la figura III.1) es la suma algebraica de *n* términos. Cada uno de ellos está formado por un coeficiente y una potencia de la variable independiente. La potencia es siempre un número natural (entero, positivo) y la variable independiente suele ser *s* en problemas relativos a la transformación de Laplace. El orden del polinomio es el mayor de los exponentes de *s* que figuran en el polinomio. Sólo hay un término por cada exponente (si hubiera dos términos con el mismo exponente, se sumarían los coeficientes). Uno o más coeficientes pueden ser nulos. Como puede observarse, la definición del polinomio en MatLab se reduce a la lista de los coeficientes, separados por blancos y encerrados entre corchetes. No deben olvidarse los coeficientes nulos, que son los que acompañarían a potencias no existentes en la definición del polinomio; si se prescinde de tales coeficientes, MatLab interpretará el polinomio como otro distinto del que realmente se desea formular.

En general la variable independiente es un número complejo, que puede ser expresado en MatLab como la suma de sus partes real e imaginaria. Esta última se denota multiplicando su valor por *i* o por *j*, lo cual impide que estas letras puedan ser utilizadas aisladas para designar otras variables. Es indiferente emplear una u otra, pero ambas han de ir precedidas por el número 1.

Se denominan raíces del polinomio aquellos valores de la variable independiente que hacen nulo el valor del polinomio. Un polinomio tiene tantas raíces como su orden. Algunas de estas raíces pueden ser múltiples; es decir, varias raíces tienen el mismo valor. Las raíces de un polinomio se obtienen en MatLab utilizando la instrucción `roots`. Dicha instrucción no puede ir seguida del símbolo **;** ya que en ese caso el resultado de utilizarla no sería accesible al usuario. El resultado de emplearla aparece automáticamente en la ventana de comandos de MatLab.

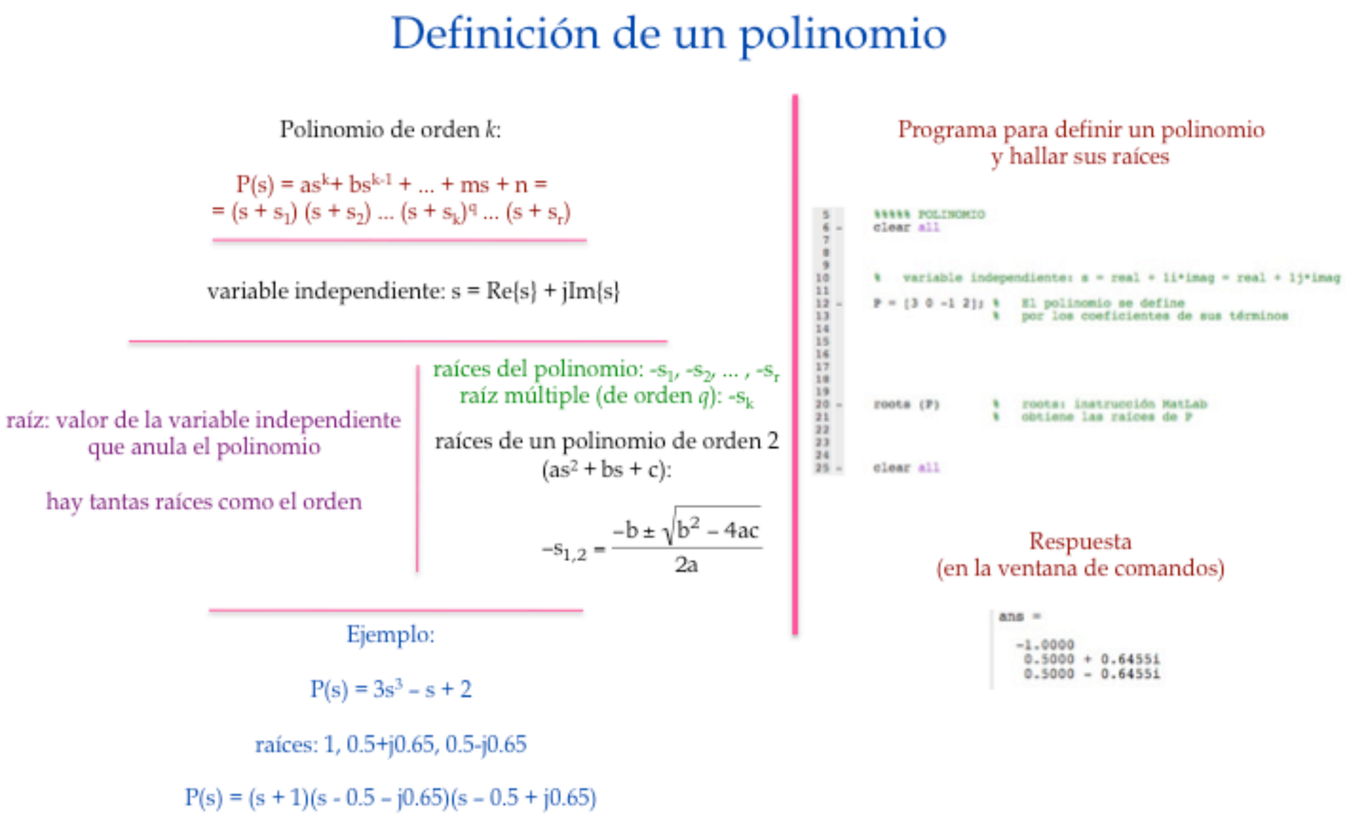

Figura III.1. Definición y raíces de un polinomio.

En el caso de polinomios de orden 2 las raíces se obtienen directamente utilizando la expresión indicada en la figura III.1. Para órdenes superiores no se dispone de expresiones analíticas sencillas que permitan calcular las raíces directamente.

En ocasiones es importante determinar el valor que toma un polinomio cuando la variable independiente tiene un valor determinado (no necesariamente el correspondiente a una raíz). Ese valor del polinomio puede calcularse utilizando la instrucción específica de MatLab que se muestra en la figura III.2. El resultado de aplicar esta instrucción de nuevo es presentado en la ventana de comandos.

Se sugiere al lector que utilice esta instrucción para determinar los valores que toma el polinomio de la figura III.1 cuando la variable independiente tiene los valores correspondientes a las raíces de dicho polinomio. Si esta acción se realiza correctamente, es obvio que el polinomio deberá tener un valor nulo en los tres casos.

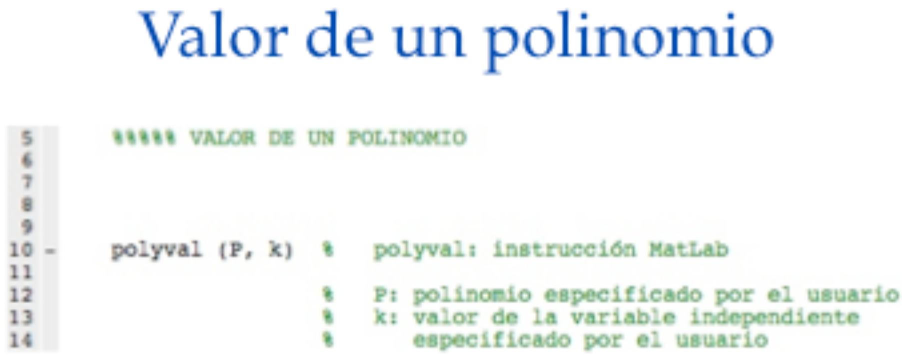

Figura III.2. Instrucción para hallar el valor de un polinomio correspondiente a un valor dado de la variable independiente.

Otro aspecto importante del manejo de polinomios en MatLab es el relativo a las operaciones que es posible realizar con ellos. La suma (algebraica) de polinomios se realiza como si se tratara de variables cualesquiera, pero sólo puede efectuarse cuando los órdenes de los polinomios son iguales (se sugiere al lector que compruebe este extremo por sí mismo).

La multiplicación y la división se realizan como se indica en la figura III.3. Hay que advertir que las instrucciones correspondientes a ambas operaciones no están sujetas a la misma restricción que la suma algebraica. Los resultados de estas operaciones, al igual que el de la suma, se presentan en la ventana de comandos de MatLab.

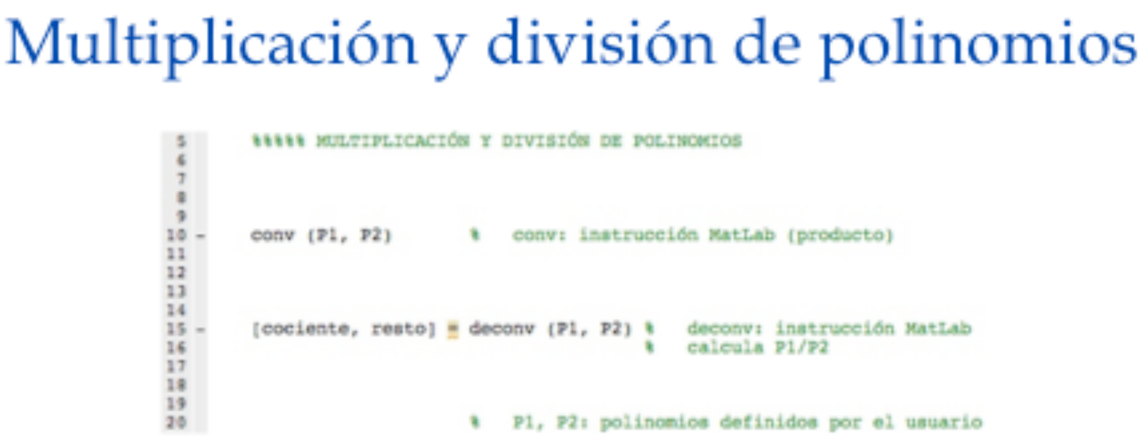

Figura III.3. Multiplicación y división de polinomios.

Obsérvese que las instrucciones relativas al producto y la división de polinomios recuerdan el concepto de *convolución*, ampliamente utilizado en tratamiento de señales. De hecho, tales instrucciones deben ser interpretadas en ese sentido cuando se hace referencia a convoluciones de señales discretas. Ahora bien, en la teoría correspondiente a este texto únicamente se trata el caso de convolución de señales continuas, con lo que los nombres de las instrucciones deben ser interpretados como indicativos de las operaciones apuntadas.

Téngase en cuenta que con anterioridad ya habíamos hecho referencia a productos y divisiones de vectores utilizando los símbolos `*`, `.*`, `/` y `./`; tampoco debe olvidarse que un polinomio es un vector fila (es decir, una matriz de una sola columna) particular (es decir, compuesto por términos exponenciales). Luego las cuatro instrucciones que acabamos de mencionar también podrían utilizarse en el caso de los polinomios. Sin embargo, cada juego de instrucciones (las indicadas en la figura III.3 y las cuatro a las que acabamos de aludir) resulta más adecuado en los contextos en los que se definen. En las operaciones con vectores se indican específicamente los elementos que los componen, mientras que en las realizadas con polinomios sólo se detallan los coeficientes. A cambio, los resultados de las operaciones son más explícitos cuando se utiliza la nomenclatura propia de los polinomios.

Una corroboración de lo que acabamos de indicar la tenemos en el hecho de que la división de polinomios puede proporcionarnos tanto el valor del cociente, como el del resto que queda una vez efectuada la operación, lo cual resulta de interés en numerosos casos prácticos.

3 Descomposición de un cociente en una suma de fracciones simples

El procedimiento convencional para obtener la transformada inversa de Laplace de una función expresada como un cociente de polinomios pasa por descomponer dicho cociente en fracciones simples (véase la figura III.4).

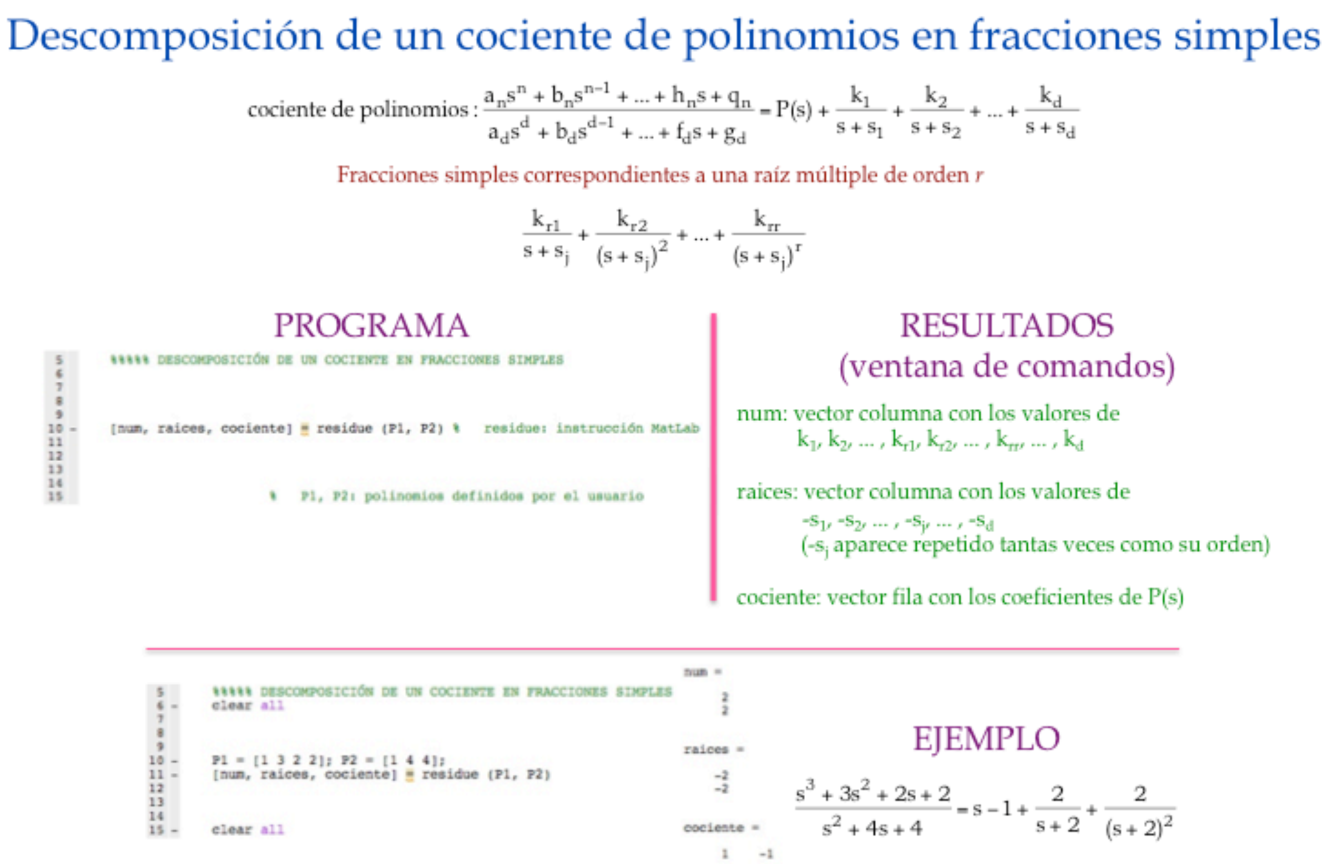

Figura III.4. Descomposición de un cociente de polinomios en MatLab.

Los denominadores de las fracciones son las sumas algebraicas de la variable independiente y las distintas raíces del denominador. Estas sumas están elevadas a la unidad excepto en el caso de que se trate de raíces múltiples; en esta situación hay que considerar tantas fracciones como el orden de la raíz múltiple. El denominador de cada una de ellas está elevado a una potencia que va creciendo de unidad en unidad hasta que se alcanza el orden de la raíz. Los numeradores de las fracciones simples son siempre números, que pueden ser calculados de acuerdo con un determinado algoritmo. La ventaja de usar MatLab estriba en que este algoritmo es transparente para el usuario. Evidentemente, si el orden del numerador del cociente es mayor que el del denominador, a las fracciones simples habrá que sumarles un polinomio, que representa el cociente de la división de polinomios.

MatLab permite obtener con una sola instrucción los numeradores de las fracciones simples, las raíces del denominador y el cociente de la división. Todos estos resultados aparecen automáticamente en la ventana de comandos. Los dos primeros se presentan en forma de vectores columna (una sola fila). Ambos vectores se corresponden término a término, de modo que, por ejemplo, el primer elemento del vector de numeradores es el que hay que utilizar en la fracción simple en la que figura la raíz que se muestra en el primer elemento del vector de raíces. Una raíz es múltiple si aparece repetida en el vector de raíces. El cociente se expresa como un vector fila integrado por los coeficientes del polinomio que constituye dicho cociente.

En ocasiones es de interés recomponer los polinomios cuyo cociente ha sido descompuesto en fracciones simples. Esta recomposición se hace igualmente con la instrucción `residue`, utilizándola como se indica en el ejemplo mostrado en la figura III.5, en la que se han empleado los mismos datos que en la figura III.4. Puede observarse que, tal y como era de esperar, los polinomios que se obtienen (definidos por sus coeficientes) son los mismos de los que se partió en los cálculos de la figura III.4.

Figura III.5. Reconstrucción de los polinomios de un cociente cuya descomposición en fracciones simples es conocida.

4 Aspectos relacionados con la respuesta en frecuencia

Como se indicó anteriormente, s es una variable compleja en general, con lo que suele ser denominada frecuencia compleja. Aún así, puede ser utilizada, en combinación con las transformaciones de Laplace y de Fourier para analizar cuestiones relacionadas con la variación de distintas magnitudes con la frecuencia en un circuito. En estas variaciones se da por sentado que la frecuencia angular, ω, es una variable real. De hecho, ω puede ser interpretada como una particularización de s para el caso de que la parte real de esta última sea nula. En relación con esta consideración es preciso tener en cuenta que ω es una variable continua, de modo que puede tomar cualquier valor en un rango determinado.

Los problemas de respuesta en frecuencia suelen estar asociados al manejo de lo que se denomina función de transferencia de un circuito. En un gran número de casos de interés práctico la función de transferencia adopta la configuración indicada en la figura III.6.

Tratamiento de la función de transferencia

$$H(s) = \frac{P_1(s)}{P_2(s)} \Rightarrow H(\omega) = H(s)\Big|_{s=j\omega} = \frac{P_1(\omega)}{P_2(\omega)} = \frac{|P_1(\omega)|_{\angle\theta_1(\omega)}}{|P_2(\omega)|_{\angle\theta_2(\omega)}} = |H(\omega)|_{\angle\theta(\omega)}$$

$$|H(\omega)| = \frac{|P_1(\omega)|}{|P_2(\omega)|} \qquad \theta(\omega) = \theta_1(\omega) - \theta_2(\omega)$$

Figura III.6. La función de transferencia como el cociente de dos polinomios dependientes de la frecuencia angular.

En los problemas aludidos es preciso definir el rango en el que varía la frecuencia angular, lo cual puede hacerse utilizando las instrucciones definidas en el tema II. Obsérvese que tales instrucciones definen vectores discretos cuando un poco más arriba hemos indicado que la frecuencia angular es una variable continua. Sin embargo, no se comete un error significativo si se elige el número de elementos del vector suficientemente alto o el intervalo entre dos elementos consecutivos suficientemente pequeño. Las representaciones gráficas suelen ser del tipo semilogarítmico, también definido en el tema II.

El hecho de que la función de transferencia sea, en general, una función compleja obliga a tener cierta familiaridad con algunas instrucciones de MatLab relacionadas específicamente con números complejos. Tales instrucciones son las mostradas en la figura III.7.

Instrucciones relativas a números complejos

```
%%%%% INSTRUCCIONES PARA EL MANEJO DE NÚMEROS COMPLEJOS



real(z)      % obtiene la parte real de z
imag(z)      % obtiene la parte imaginaria de z
abs(z)       % obtiene el valor absoluto (módulo) de z
angle(z)     % obtiene la fase (en rad) de z

             % z: número complejo definido por el usuario
```

Figura III.7. Instrucciones de MatLab relacionadas directamente con números complejos.

La figura III.8 muestra un ejemplo típico de determinación de la respuesta en frecuencia de un circuito caracterizado por su función de transferencia.

Respuesta en frecuencia

Se trata de obtener la respuesta en frecuencia de un circuito caracterizado por la función de transferencia

$$H(s) = \frac{0.5s}{s^2 + 2s + 2}$$

```
%%%%% RESPUESTA EN FRECUENCIA
clear all



w = logspace(-2, 2, 10000);
s = 1j*w;



Hs = 0.5*s./(s.^2 + 2*s + 2);
moduloH = abs(Hs); faseH = (180/pi)*angle(Hs);



semilogx (w, moduloH);  %   Desactivada cuando se representa la fase
semilogx (w, faseH);    %   Desactivada cuando se representa el módulo



clear all
```

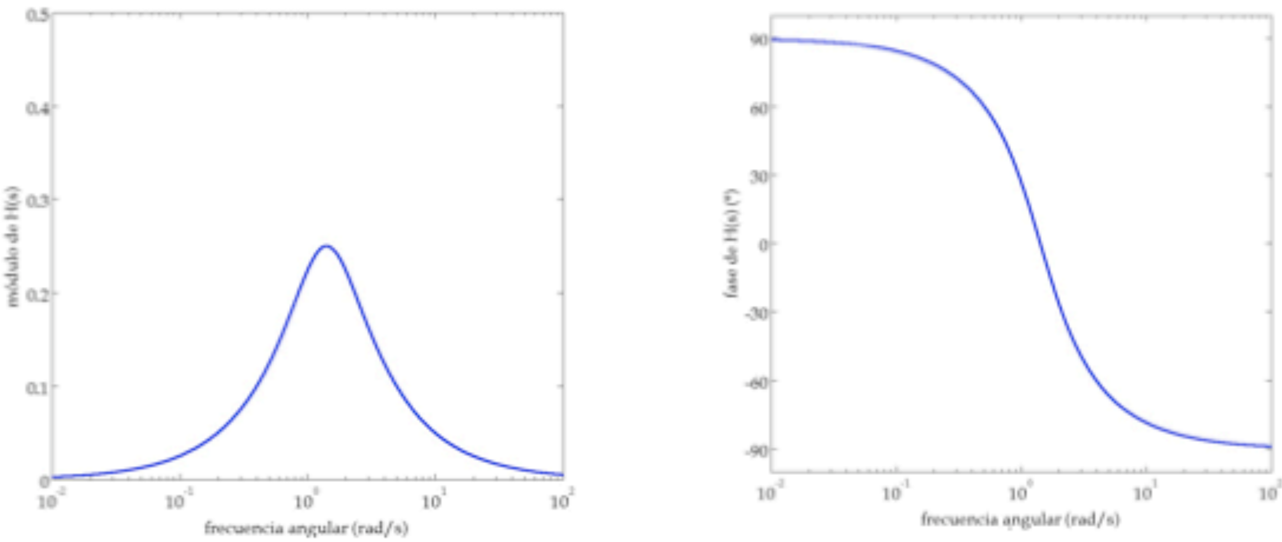

Figura III.8. Respuesta en frecuencia de un circuito caracterizado por su función de transferencia.

Dos instrucciones de interés en problemas relacionados con la respuesta en frecuencia son las que se muestran en la figura III.9.

Valores máximo y mínimo de un vector

```
%%%%% RESPUESTA EN FRECUENCIA




maximo = max(y)     %   obtiene el valor máximo del vector y
minimo = min(y)     %   obtiene el valor mínimo del vector y

                    %   y: vector definido por el usuario

                    %   max, min: instrucciones MatLab
```

Figura III.9. Instrucciones de MatLab aplicables en problemas relativos a la respuesta en frecuencia.

Como puede observarse, se trata de dos instrucciones que permiten obtener los valores más elevado y más bajo entre los contenidos en un vector. Los resultados de la ejecución de estas instrucciones aparecen en la ventana de comandos.

Un aspecto importante en la determinación de la respuesta en frecuencia en un circuito está relacionado directamente con la variación con la frecuencia de la fase de dicha respuesta. En algunas ocasiones la representación de la fase muestra una discontinuidad (por ejemplo, pasando de -180º a 180º) a una frecuencia dada. Esta circunstancia puede deberse a que el circuito se comporta de forma que realmente se produce tal discontinuidad, o bien puede ser originada por las manipulaciones matemáticas internas que MatLab lleva a cabo. A este respecto debe recordarse que, a fin de cuentas, los ángulos α y 180º+α tienen la misma tangente y que es la tangente la función que suele utilizarse en la determinación de ángulos en calculadoras y ordenadores.

Una forma de determinar a qué se debe la discontinuidad (y de eliminarla en el caso de que sea causada por los cálculos internos de MatLab) consiste en utilizar la instrucción indicada en la figura III.10. En la figura se muestra, además, a título de ejemplo, el efecto que tiene utilizar o no tal instrucción en la representación de la variación con la frecuencia de la fase de una función de transferencia.

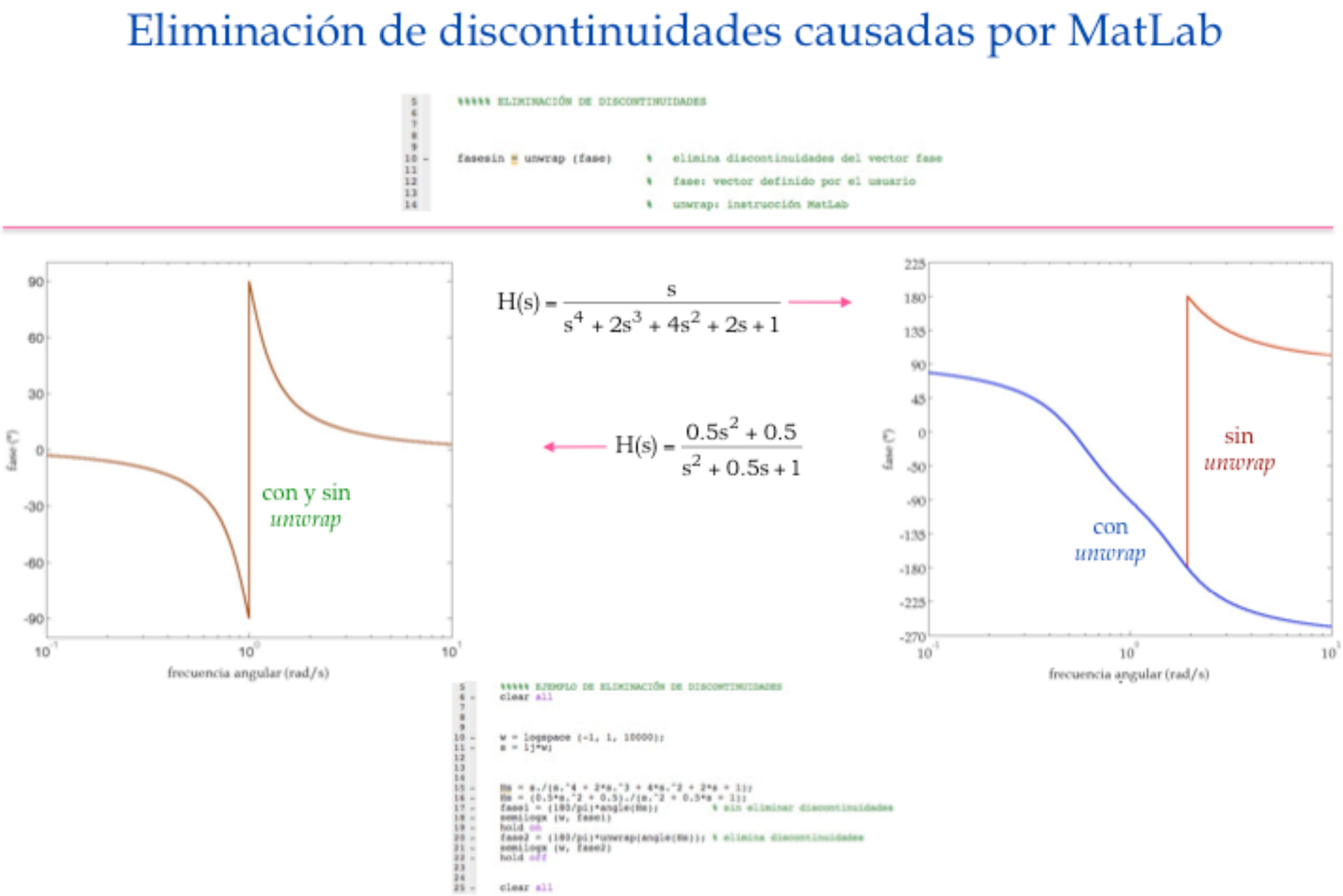

Figura III.10. Eliminación de discontinuidades originadas por cálculos internos de MatLab.

A la vista de las representaciones gráficas mostradas en la figura puede concluirse que la función de transferencia representada en el lado izquierdo tiene una discontinuidad propia, mientras que en la indicada en el lado derecho la discontinuidad se debe a las operaciones internas de MatLab.

Anteriormente se mencionó el hecho de que la función *tangente* suele estar directamente relacionada con el ángulo al que corresponde en programas de calculadora y ordenador. Ahora bien, la circunstancia de que en un arco de 360º haya parejas de ángulos que tienen las mismas tangentes puede introducir cierta ambigüedad en la determinación de ángulos por parte de un programa. Esto es lo que ocurre en MatLab, donde, como se indica en la figura III.11, hay dos instrucciones que permiten obtener el valor de un ángulo a partir de su tangente.

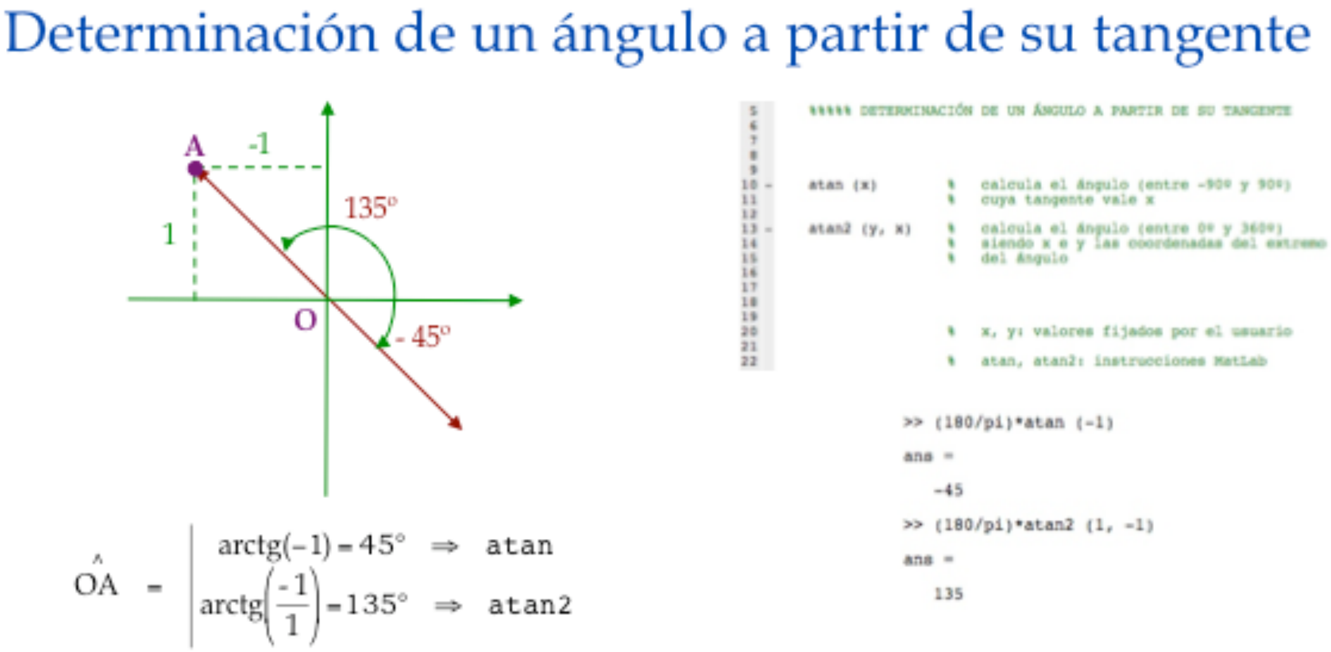

Figura III.11. Dos formas posibles de determinar un ángulo a partir de su tangente.

Una de las instrucciones únicamente tiene en cuenta el valor de la tangente y calcula el ángulo comprendido entre -90º y 90º que presenta dicho valor. La otra tiene en cuenta, además, el cuadrante afectado por el cálculo, por lo que la instrucción requiere que el usuario proporcione las coordenadas del punto que corresponde al extremo final del vector cuyo ángulo se pretende determinar. En la figura pueden verse los distintos resultados que se obtienen según se utilice una u otra instrucción.

Para concluir esta sección volveremos al caso representado en la figura III.8, en la que se utilizan un programa y dos gráficas independientes para determinar la respuesta (en módulo y fase) de un circuito ante las variaciones de la frecuencia. Si no se desea tener tantos detalles como los que pueden conseguirse recurriendo al programa, puede utilizarse la instrucción indicada en la figura III.12, la cual proporciona directamente dos gráficas con las variaciones con la frecuencia del módulo y la fase de una función de transferencia.

Representación simultánea de las variaciones con la frecuencia del módulo y la fase

```
%%%%% REPRESENTACIÓN SIMULTÁNEA DE LAS VARIACIONES DE MÓDULO Y FASE




freqs (P1, P2, w)          %   P1, P2: polinomios, definidos por el usuario,
                           %   que determinan la función de transferencia

                           %   w: vector que determina
                           %   el rango de variación de la frecuencia

                           %   freqs: instrucción MatLab
```

EJEMPLO

$$H(s) = \frac{0.5s}{s^2 + 2s + 2}$$

```
%%%%% REPRESENTACIÓN SIMULTÁNEA DE LAS VARIACIONES DE MÓDULO Y FASE
clear all



w = logspace (-3, 3, 10000);
s = 1j*w;



freqs ([0.5 0], [1 2 2], w)




clear all
```

Las figuras han sido modificadas utilizando la opción `Figure Properties` del menú `Edit`

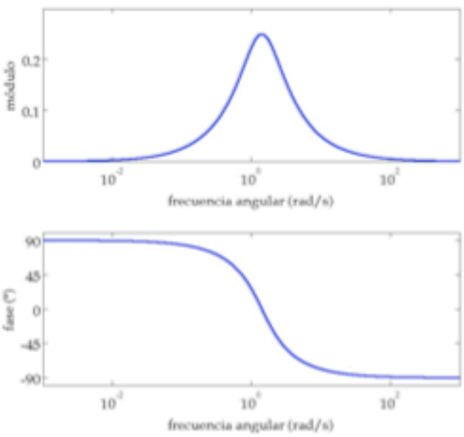

Figura III.12. Representación simultánea de las variaciones con la frecuencia del módulo y la fase de una función de transferencia.

La instrucción considerada permite obtener simultáneamente las dos curvas de interés. En la figura se muestran las curvas correspondientes a una función de transferencia dada (se muestran las versiones editadas). Obsérvese que la curva de la fase está dimensionada directamente en grados sexagesimales aunque MatLab opere por defecto en radianes. En la confección del programa no se indican explícitamente los polinomios cuyo cociente determina la función de transferencia; por el contrario, se definen directamente, mediante sus coeficientes, en la instrucción `freqs`.

5 Ejercicios propuestos

Para comprobar hasta qué punto se ha familiarizado con los procedimientos detallados en este tema el lector puede intentar resolver los ejercicios que se proponen seguidamente. Las soluciones de los mismos han sido incluidas en el apéndice 1.

Ejercicios propuestos

1 Descomponed en fracciones simples la función $H(s) = \dfrac{2}{s^2 + 2s + 2}$

2 Descomponed en fracciones simples la función $H(s) = \dfrac{s^4}{s^3 + 10s^2 + 25s}$

3 Dada la función $H(s) = \dfrac{0.5s}{s^2 + 2s + 2}$, hallad la frecuencia para la que la fase es nula.
¿Cuánto vale el módulo de la función para dicha frecuencia?

Figura III.13. Ejercicios propuestos relacionados con la transformación de Laplace y la respuesta en frecuencia.