TP6 - Procesamiento de Señales Transformada z

POLINOMIOS EN MATLAB

En matemática los polinomios son funciones formadas por suma de términos que son potencias de x.
Ejemplo: $y(x) = x^2 - 3x + 1$ (Este polinomio tiene nombre propio y se denomina parábola).
Se denomina orden del polinomio al mayor exponente de x. El ejemplo anterior es un polinomio de orden 2.
Obviamente la forma en que estén dispuestos los términos no importa, de modo que $y(x) = 1 - 3x + x^2$, es el mismo polinomio del ejemplo anterior.
En forma general un polinomio se expresa

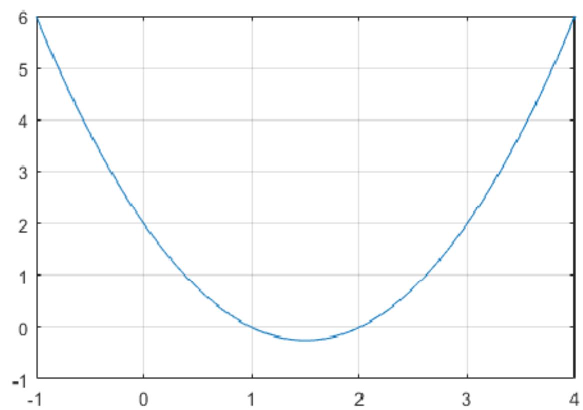
$$y = \sum_{k=0}^{n} a_n . x^n = a_0 x^0 + a_1 x^1 + a_2 x^2 + \cdots + a_n x^n$$

Un polinomio de grado 2 tiene tres términos (desde x^0 hasta x^2). Hay que recordar que cualquier valor elevado a la 0 es igual a 1, por lo que $x^0 = 1$, y no se suele escribir en el polinomio. Por su parte $x^1 = x$.
De modo que suele expresarse de la siguiente manera

$$y = \sum_{k=0}^{n} a_n . x^n = a_0 + a_1 x + a_2 x^2 + \cdots + a_n x^n$$

Grafiquemos el polinomio del ejemplo $y = x^2 - 3x + 1$

```
%% Graficar polinomio
x = -1:0.01:4;
y = x.^2 - 3*x + 2;
figure(1);
plot(x,y)
grid on
```



Como nos vamos a concentrar en los puntos donde corta al eje x, haremos una modificación en los ejes del gráfico. Esto en MATLAB es un poco extraño pero puede hallarse buscando en google "apariencia ejes MATLAB" o "diseño ejes MATLAB" y copiar los ejemplos.
Se utiliza una función especial llamada gca (GraphicCurrentAxes) y lo asigna a una variable de un tipo que aún no vimos y que se denomina "estructura" (en inglés es struct).
Si tipeamos lo siguiente (y evitamos el punto y coma para que haga eco en pantalla) veremos lo siguiente

```
>ejes=gca
ejes =
  Axes with properties:
XLim: [-1 4]
YLim: [-1 6]
XScale: 'linear'
YScale: 'linear'
GridLineStyle: '-'
Position: [0.1300 0.1100 0.7750 0.8150]
          Units: 'normalized'
```

Si ahora tipeamos lo siguiente (con eco en pantalla)

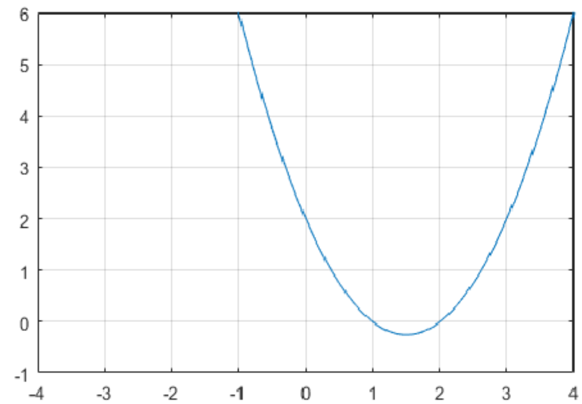
```
>>ejes.XLim
ans =
   -1     4
```

Nos devuelve los valores límites del eje x.

Una variable estructura es como un grupo de valores asociado a un nombre. Dimos el nombre ejes a esta estructura, y ella contiene varios valores que se especifican con el nombre de esa subvariable luego de un punto. Así ejes.XLim es un vector de dos elementos que especifica el mínimo y el máximo del eje x en el gráfico. Podemos ver su valor, pero también podemos modificarlo.

Si escriben

```
>>ejes.XLim=[-4 4];
```



verán que modifican los límites en el gráfico.

Si bien podríamos hacer hecho este mismo proceso con axis() nos servirá para tener una idea de lo que significa una estrctura.

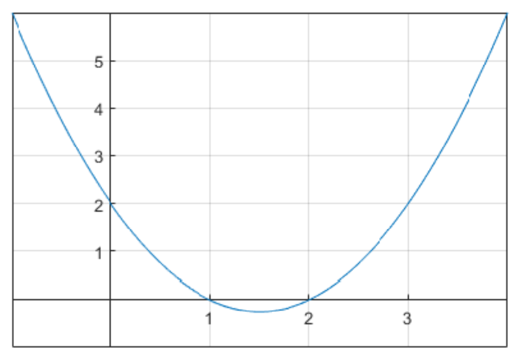
La estructura generada por gca tiene más propiedades que las que se muestran (da la posibilidad al final de mostrar todas las propiedades: "Show allproperties").

Nos interesarán ahora en particular las que indiquen la posición de los ejes, para que el gráfico se muestre más parecido a los gráficos tradicionales de matemática.

Agreguen las últimas tres instrucciones al script.

```
x = -1:0.01:4;
y = x.^2 - 3*x + 2;
figure(1);
plot(x,y)
grid on
ejes=gca;
ejes.XAxisLocation = 'origin';
ejes.YAxisLocation = 'origin';
```

NOTA: No olvidar que deben usar ".^" para la potencia.



Ahora regresamos al tema que nos interesa analizar. Esta parábola corta al eje x en dos puntos. El valor de cada uno de esos puntos se denomina "raíz". Se dice que un polinomio de orden 2 (una parábola) tiene dos "raíces".

Para hallar las raíces de una parábola existe una ecuación algo molesta de aplicar.

Dada la parábola $y = a.x^2 + bx + c$, entonces sus raíces serán

$$x_{1,2} = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

Aplicando esto al ejemplo $y = x^2 - 3x + 2$, vemos que a = 1, b = -3, y c = 2. Entonces

$$x_{1,2} = \frac{-(-3) \pm \sqrt{(-3)^2 - 4.2}}{2}$$

De donde se obtiene que la primera raíz es x1 = 2. y la segunda raíz es x2 = 1

Polinomios y Raíces en MATLAB

MATLAB tiene dos funciones denominadas poly() y roots() que facilitan el trabajo con polinomios y raíces. Pensemos que si es molesto hallar las raíces para un polinomio de orden 2 (parábola), es mucho peor para orden 3, y de hecho no tiene solución general con una simple ecuación para órdenes mayores y sólo se puede obtener un valor aproximado.
El trabajo de MATLAB con polinomios es generando un vector equivalente a sus coeficientes (los a_n del modo general o los a,b y c de la parábola).
Así, el polinomio $x^2 - 3x + 2$ se interpretará en MATLAB como [1 -3 2], para el trabajo de hallar las raíces.
El vector que representa al polinomio comienza con el coeficiente del término de mayor orden y va colocando los que siguen en orden, teniendo cuidado de colocar un cero si es que falta algún término en el medio.
Ejemplos:
El polinomio $x^2 - 1$ se convierte en el vector [1 0 -1]
El polinomio $4x^4 + x^3 + 2x^2 - x$ se convierte en el vector [4 1 2 -1 0]
Los coeficientes también pueden ser números decimales.
El polinomio $0.2 x^4 + 3x^3 - 0.5 x^2 + 7$ se convierte en el vector [0.2 3 -0.5 0 7]

Para hallar las raíces se aplica la función roots() a un vector con los coeficientes del polinomio.
Veamos el caso del ejemplo de parábola

```
>> parabola = [1  -3   2];
raices = roots(parabola)

raices =
2
    1
```

Es bastante más cómodo que resolver la ecuación anterior con a, b y c. MATLAB considera a los polinomios como vectores horizontales y a las raíces como verticales, aunque eso es sólo una curiosidad por el momento y no tendrá demasiada influencia en lo que haremos.
Ahora, supongamos que queremos saber qué polinomio tiene un conjunto de raíces previamente definido. Esto es, supongamos que un docente de matemática pidiera obtener el polinomio que corte al eje x en dos puntos (en x=1 y en x=2). Si tiene dos raíces, el polinomio será de segundo orden (el orden y la cantidad de raíces están relacionados).
Si quisiera obtener la respuesta manualmente, debería considerar lo siguiente:

$$y = (x - 1)(x - 2)$$

Este es un procedimiento general y consiste en escribir el polinomio como una serie de paréntesis que se multiplican entre si. Tantos paréntesis como raíces. Dentro de cada paréntesis se escribe la variable x y se le resta el punto que corresponde a cada una de las raíces. Para obtener el polinomio en su forma normal hay que aplicar la propiedad distributiva multiplicando cada elemento por los dos del otro paréntesis, sumar los términos y acomodarlos.
Créanme que al hacer eso se obtiene $x^2 - 3x + 2$. Podemos darnos el lujo de saltear este paso engorroso porque MATLAB nos resuelve la tarea mediante la función poly(). Esta función nos da el vector de coeficientes de un polinomio que tiene las raíces que se especifican como argumento (dentro del paréntesis).

```
>>raices = [1; 2];
>>poly(raices)
ans =
    1    -3     2
```

El orden de las raíces no es importante (pero sí lo es el orden de los coeficientes del polinomio). Esto quiere decir que si en el último ejemplo escribimos

```
>>raices = [2; 1];
>>poly(raices)
ans =
    1    -3     2
```

Pero si intentamos hallar las raíces y escribimos el polinomio al revés, obtendremos resultados diferentes

```
>>parabola = [2  -3  1];  %CUIDADO: El orden está invertido para mostrar que no da lo mismo
raices = roots(parabola)
```

```
raices =
   1.0000
   0.5000
```

Estas serían las raíces de la siguiente parábola $y=2x^2 - 3x + 1$

1. Obtener un polinomio que tenga tres raíces x1 = -1, x2 = 1, y x3 = 2

a. Generar una variable raices que contenga los tres valores de las raíces deseadas como vector vertical.

b. Obtener el vector con los coeficientes del polinomio

c. Generar una variable x (eje horizontal) que vaya desde -2 a 2, con saltos de 0.01

d. Escribir en papel el polinomio que se corresponda con el vector obtenido en el punto b. Generar una variable y que represente ese polinomio como función de x. Ejemplo: si el vector polinomio fuese [1 -3 -2 2], entonces escribiríamos y = x.^3-3*x.^2- 2*x + 2;

e. Graficar el polinomio con plot(x,y) y con los ejes en el origen (utilizando gca) para verificar si las raíces están donde se espera.

2. Hallar las raíces de un polinomio de grado 4

a. Graficar el siguiente polinomio y = x.^3 - 3*x.^2 - 2*x + 2 (eje x entre -2y 4)

b. Obtener el valor de sus raíces utilizando roots() y verificar que coincidan con los cruces del eje x en el gráfico.

Si tenemos un polinomio de orden 3, entonces tendrá 3 raíces. Uno de orden 2 (parábola) tendrá 2 raíces.

¿Todos los polinomios cortan al eje x en tantos puntos como indica su orden? En realidad no. Es perfectamente posible generar una parábola que no corte al eje x.

Por ejemplo, y = x.^2 - 3*x + 4.

Sin embargo, los matemáticos siguen diciendo que tiene 2 raíces (ya que aplican la misma ecuación para calcular las raíces de una parábola). Pero entonces, ¿cómo es posible que se obtengan raíces si no corta al eje? Bien, el tema es que al aplicar la ecuación de las raíces de una parábola el resultado da un número complejo. Resulta que en ese caso la cuenta dentro de la raíz cuadrada da negativo, y una raíz cuadrada de un número negativo no tiene solución en los números reales.

3. Hallar las raíces de una parábola que no corta al eje x

a. Graficar el siguiente polinomio y = x.^2 - 3*x + 4 (eje x entre -2 y 4)

b. Obtener el valor de sus raíces utilizando roots(). Se obtendrán 2 raíces (porque es un polinomio de orden 2) cuyos valores tienen componente real y componente imaginario.

¿Y qué sucede si corta al eje x en un solo punto? En ese caso se dice que tiene una "raíz doble". Los matemáticos siguen diciendo que hay dos raíces, pero que el valor de ambas raíces es el mismo. ¿Tiene sentido? En parte sí, ya que si se resuelve la ecuación de la raíz cuadrada para una parábola se verá que lo que está dentro de la raíz cuadrada da igual a cero y en ese caso x1 da el mismo resultado que x2.

4. Hallar las raíces de una parábola que corta al eje x en un solo punto

a. Graficar el siguiente polinomio y = x.^2 - 3*x + 2.25 (eje x entre -2 y 4)

b. Obtener el valor de sus raíces utilizando roots(). Se obtendrán 2 raíces (porque es un polinomio de orden 2) cuyos valores son exactamente iguales.

Lo visto hasta ahora nos permite trabajar con polinomios y raíces, pero hay algo más importante. A partir de lo trabajado podemos notar que para un polinomio dado (que en MATLAB puede representarse como un vector), es posible obtener

un conjunto de raíces (otro vector), que siempre será el mismo. Y se verifica que, si comenzamos con el vector del conjunto de raíces, podemos obtener el polinomio correspondiente (que será siempre el mismo).
Cada vector de coeficientes de polinomio se relaciona con un vector de raíces, y viceversa. La relación es uno a uno, y por lo tanto podemos utilizar al vector de raíces como una representación de un polinomio. Dicho en otras palabras, si nos hablan del polinomio que tiene las siguientes raíces [3 2 1], podremos saber que nos hablan de un polinomio en particular. Incluso lo podríamos graficar con unas pocas instrucciones de MATLAB.
Estas raíces pueden ser números complejos, por lo cual suelen representarse en el plano complejo (el eje "x" es la parte real y el eje "y" la parte imaginaria).

Ejemplo paso a paso
Supongamos que tenemos un gráfico donde vemos la ubicación de las raíces. Del gráfico leemos esos valores y los agrupamos en un vector que llamamos raíces. Luego obtenemos polinomio=poly(raices); y por último construimos el polinomio y(x).

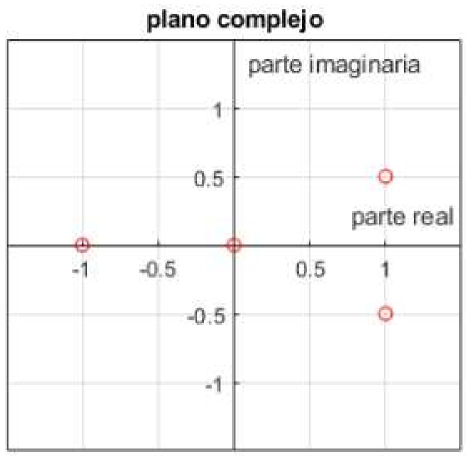


Figura 1: Raíces (ceros) de un polinomio

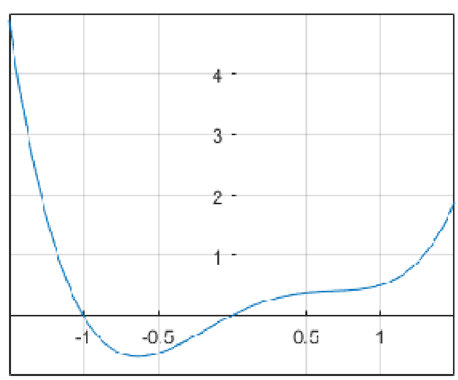
De la figura 1 obtenemos el vector raices, y calculamos poly(raices)

```
raices = [-1; 0; 1+0.5i;  1-0.5i];
p = poly(raices)
```

```
p =
   1.0000   -1.0000   -0.7500    1.2500        0
```

Graficamos el polinomio

```
>> x=-2:0.01:2;
>> y = x.^4 - x.^3 - 0.75*x.^2 + 1.25*x;
>>plot(x,y)
```



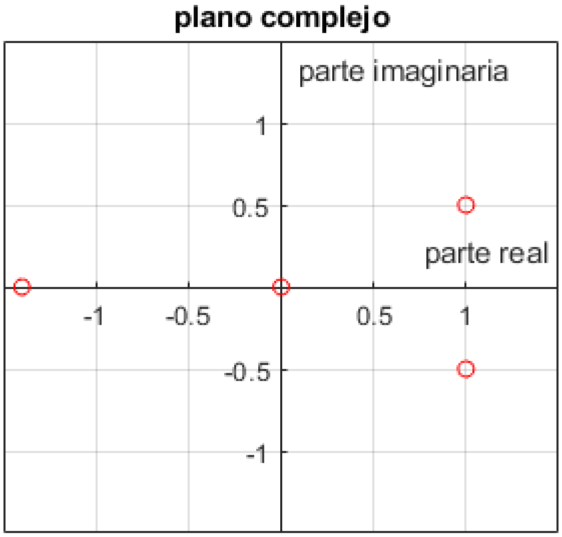
¿En cuántos puntos corta al eje horizontal? ¿Se podría haber contestado a esta pregunta mirando solamente el gráfico de las raíces?
Corta en 2 puntos (en -1 y en 0). Se podría haber contestado observando las raíces, porque si bien tiene cuatro raíces, solamente dos son raíces reales.

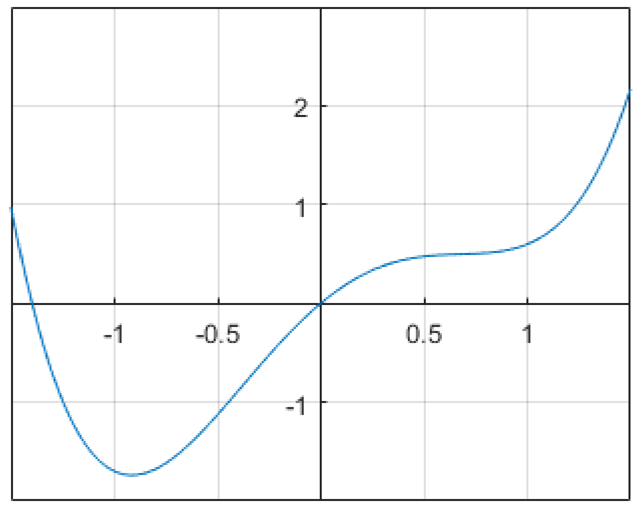
Lo más importante de esto que estamos viendo es la asociación fuerte que puede hacerse entre un conjunto de raíces (se llamarán "ceros", porque son los valores que hacen cero a la función del polinomio) y toda la ecuación del polinomio. Cada conjunto de "ceros" tiene asociado un polinomio que puede hallarse.

5. Hallar un polinomio conociendo el gráfico de dónde se ubican sus raíces en el plano complejo

a. Partiendo del gráfico que muestra las raíces, obtener el vector del polinomio correspondiente. (Es parecida a la anterior, pero tiene un pequeño cambio en la raíz de la izquierda, y eso modifica algo la curvatura del polinomio)

b. Graficar el polinomio y verificar que coincida con el que se muestra a la derecha





NOTA: Viendo el gráfico de la derecha anterior (el polinomio) no resulta sencillo saber el orden del polinomio ni su expresión matemática. Sin embargo, el gráfico de la izquierda (sus ceros) queda claro que es un polinomio de orden 4 y además, con solo un par de líneas de código se puede obtener la expresión matemática. Esto hace que en muchos casos resulte preferible concentrarse en el gráfico de ceros, para tener más claro de qué polinomio se está hablando.

Asociación entre una función temporal (audio) y un polinomio

Considerando que MATLAB asocia un polinomio con un vector, podríamos pensar qué pasaría si ese vector fuese un archivo de audio, por ejemplo. Es cierto que suena muy extraño al principio, porque si tenemos un archivo de audio de frecuencia de muestreo 48 kHz y duración 1 segundo, entonces estaríamos pensando en un polinomio de orden 48000. Suena algo loco, pero podremos encontrarle utilidad a este razonamiento.

Para hacerlo más comprensible trabajaremos en la explicación conarchivos de audio con muy pocas muestras totales. Utilizaremos además un tipo de representación gráfica común cuando se trabaja con señales digitales en las cuales se dibujan las muestras individuales (en lugar de trazar una curva continua). La figura 2 muestra la diferencia entre ambos modos de representación, con las funciones plot() y stem().

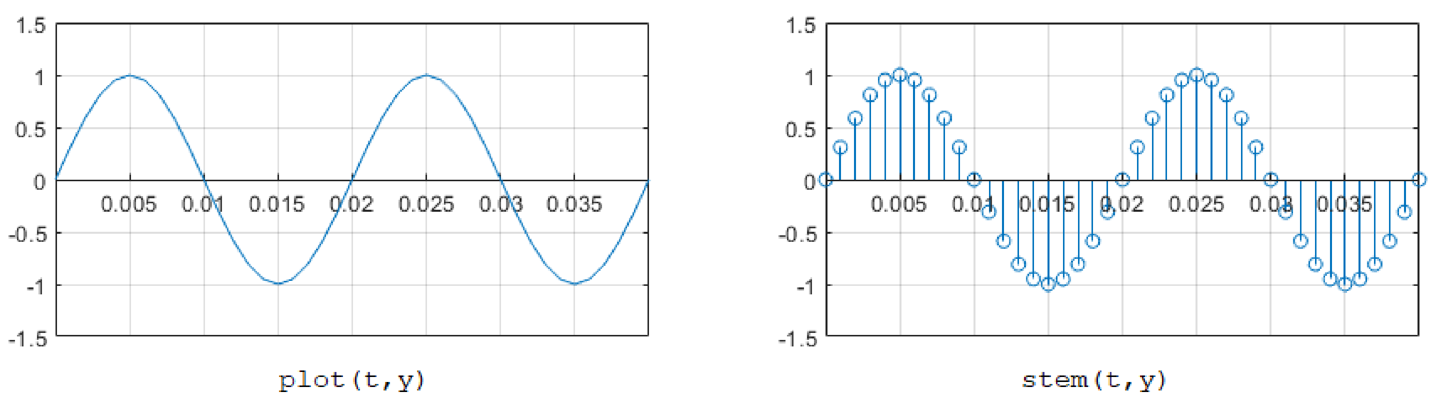
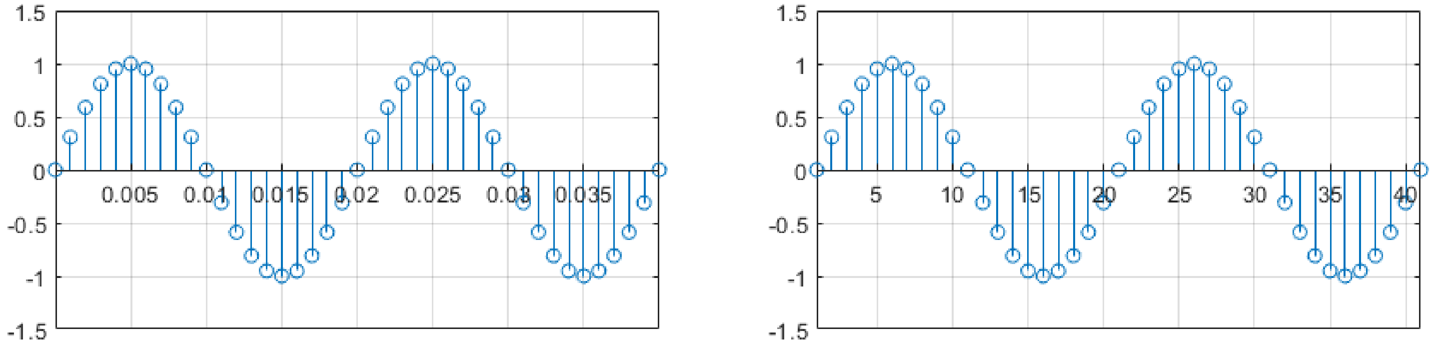


Figura 2 - Representación de una señal continua y una señal discreta.

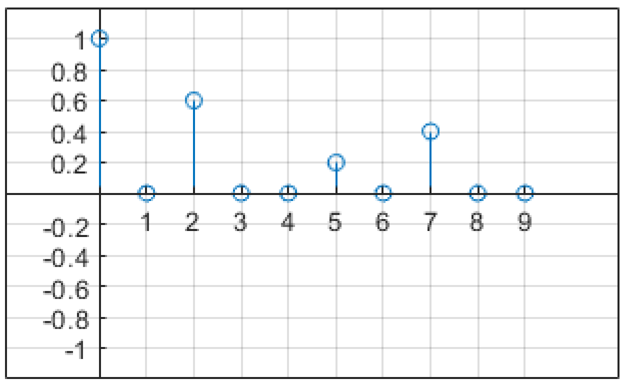
Es más, es común al representar señales discretas que se utilice como eje horizontal un número entero (correspondiente al número de muestra), como en la Figura 3.



stem(t,y) stem(n,y)

Figura 3 - Representación discreta en función del tiempo y del número de muestra (t = n.dt = n/fs)

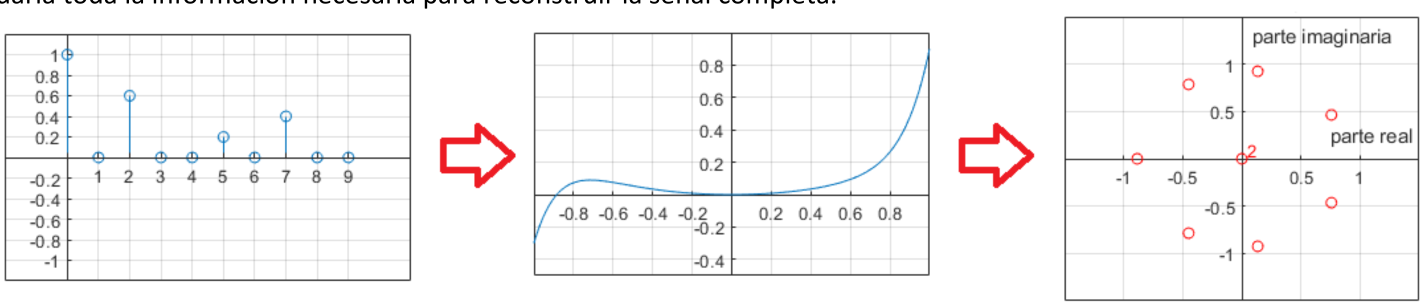
Supongamos que tenemos la siguiente señal discreta, formada por un puñado de pulsos.



El vector representado es y = [1, 0, 0.6, 0, 0, 0.2, 0, 0.4, 0, 0]

El polinomio que podríamos asociar con ese vector es polinomio = $x^9 + 0.6\ x^7 + 0.2\ x^4 + 0.4\ x^2$

¿Para qué podríamos querer hacer semejante cosa? Pues, porque si tenemos un polinomio asociado, eso quiere decir que tiene raíces (ceros) que pueden representarse en un plano complejo. La ubicación de esos "ceros" (el mapa) nos daría toda la información necesaria para reconstruir la señal completa.

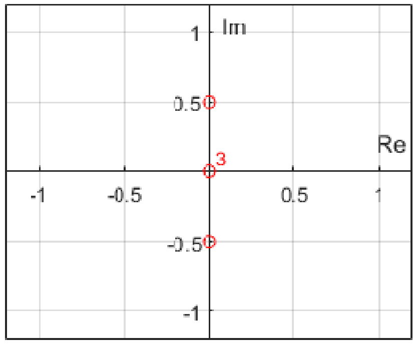


secuencia de muestras polinomio asociado mapa de ceros (raíces)

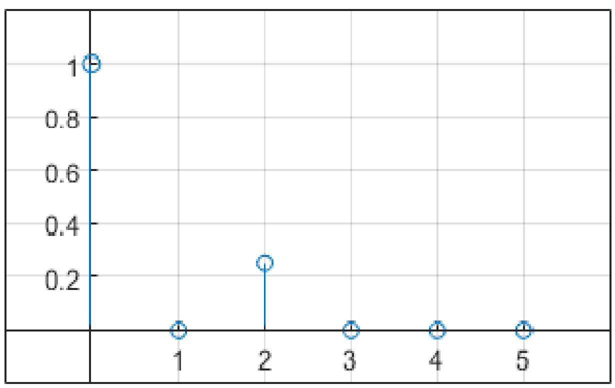
Puede notarse un "2" rojo en el origen. Es porque hay una doble raíz (cero doble) en ese lugar.

El polinomio no lo usaremos más, fue solo un "truco" matemático para asociar la secuencia de muestras con una representación de ceros en el plano complejo. Es verdad que por el momento no se puede notar ninguna ventaja, pero mantengamos eso en suspenso y aceptemos que, si tenemos el plano con la ubicación de los ceros, podemos reconstruir el vector del polinomio y este no es otra cosa que la secuencia temporal de muestras de nuestra señal. De hecho, podemos saltear el polinomio ya que sólo nos interesa el vector que lo representa (y ese vector es nuestra señal discreta).

Ejemplo paso a paso.
Hallar la señal discreta que queda representada por el siguiente diagrama de ceros.



Lo primero es escribir un vector que contenga los ceros (en cualquier orden, como vector columna)
>> ceros = [0; 0; 0; 0.5i; -0.5i];
Luego hacemos que MATLAB calcule el vector del polinomio que tendría esas raíces.
>> polinomio = poly(ceros);
Y en realidad, ese vector polinomio ya es nuestra secuencia.Para colocarla en forma ordenada desde la muestra n=0 hasta n=5, generamos el vector n.
NOTA: Tienen que ser 6 valores en total, porque había 5 raíces, y los polinomios tienen un término más. Una parábola tiene orden 2, tiene 2 raíces, pero tiene tres términos.
>> n=0:5;
>>stem(n,polinomio)



NOTA: En las figuras estamos utilizando la función cga para cambiar los ejes y la cantidad de líneas en la grilla para que la explicación quede más clara. No lo explicitamos con detalle por el momento para poder seguir con el razonamiento general. La aclaración es porque si repiten los pasos obtendrán idénticos valores pero el gráfico no se mostrará del mismo modo que la figura.

Conclusión: este "diagrama de ceros" contenía información de la señal mostrada. Y nos atrevemos a adelantar una idea: ¿No se le ve un cierto airea a "respuesta al impulso"? La figura mostrada podría ser la respuesta al impulso de un brevísimo eco. Vamos en camino de aclarar esos puntos.

NOTA: Quizás les sirva recordar los primeros momentos en que comenzaron a asociar una señal temporal con un espectro de Fourier. Al principio suena muy extraño, pero luego de un tiempo comenzamos a interpretarlo y se vuelve "casi natural".

Primera aproximación intuitiva a la Transformada z

Si volvemos a mirar el vector de las muestras con las que comenzamos esta explicación (señal original) y el polinomio, puede notarse que es bastante incómoda la relación entre los exponentes y la ubicación de las muestras.

vector y = [1, 0, 0.6, 0, 0, 0.2, 0, 0.4, 0, 0]
polinomio $y = x^9 + 0.6\ x^7 + 0.2\ x^4 + 0.4\ x^2$

Si por algún motivo tuviéramos la misma secuencia de muestras, pero con un poco más de duración (agregando, por ejemplo, un par de ceros al final), el comienzo del vector sería el mismo, pero el polinomio tendría un grado mayor y habría que modificar todos los exponentes de x.

Ahora, pensemos en hacer una pequeña modificación a nuestro polinomio, dividiéndolo completo por x elevado al mayor exponente.

$$\frac{x^9 + 0.6 \cdot x^7 + 0.2 \cdot x^4 \ + 0.4 \cdot x^2}{x^9} = 1 \ + 0.6 \cdot x^{-2} \ + 0.2 \cdot x^{-5} + 0.4 \cdot x^{-7}$$

El resultado que se muestra a la derecha surge de dividir dos polinomios. El original, y uno muy básico de orden 9 donde sólo el primer coeficiente vale 1 y el resto vale cero ($x^9 + 0.x^8 + 0.x^7 + ... + 0.x + 0$).

Si ahora agregamos ceros al final del vector de la señal, el nuevo "polinomio" no cambia.

Una cosa que se agrega es que ahora el polinomio de abajo de la división tiene sus propias raíces. Ese polinomio tan simple que usamos x^9 tiene nueve raíces en el origen (en "cero"). Quiere decir que si representásemos ese polinomio (toda la función para valores de x) resultaría que cuando x fuese igual a cero el resultado de esa cuenta daría infinito.

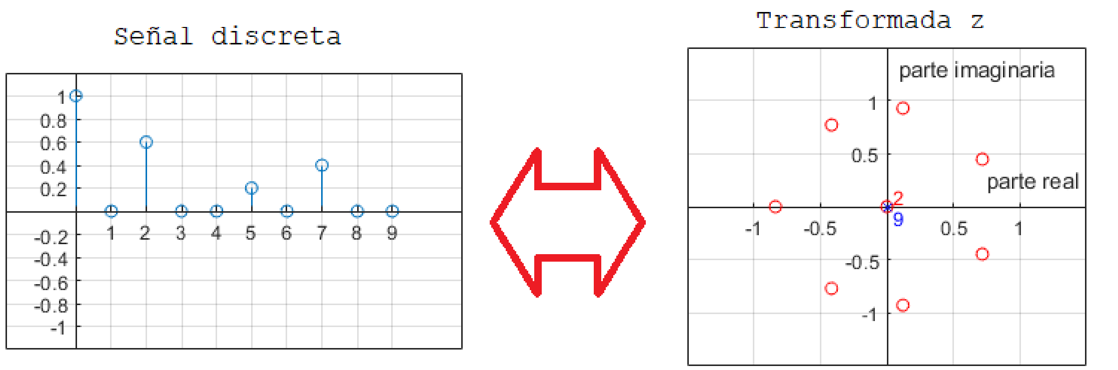
Eso asusta a cualquiera, pero como en realidad no representamos por el momento el polinomio no es un problema. Sin embargo, hay que tener estas raíces en cuenta para seguir el proceso de la teoría matemática que hay detrás de todo este análisis. Las raíces del polinomio de arriba se denominan ceros, y las raíces del polinomio de abajo se denominan polos.

Lo único que se modifica ligeramente respecto de lo que hicimos hasta ahora es que en lugar de tener un diagrama de ceros que representa una señal, tendremos un diagrama de polos y ceros. En el diagrama del plano complejo los ceros se representan con un pequeño círculo y los polos con cruces.

Debido a que los valores posibles de x que corresponden a raíces son números complejos, y a que los matemáticos suelen utilizar la variable z para los valores complejos, la expresión que se utiliza para representar la señal como resultado de la división de dos polinomios utilizará la variable z. Con esto comienza a aparecer una relación entre lo que venimos haciendo y el nombre de Transformada z.

Sabemos que una señal puede representarse mediante la Transformada de Fourier. Esto quiere decir que si tenemos la señal podemos obtener su espectro. Y si tenemos su espectro podemos obtener la señal que le corresponde.

Algo similar sucede con la Transformada z. Si tenemos una señal discreta (temporal), podemos asociarle una serie de términos utilizando la variable z elevada a exponentes negativos (división de dos polinomios). Esa división de polinomios tiene un diagrama de raíces (polos y ceros) que lo representa exactamente.Esa es la Transformada z. A cada señal se la puede asociar con un diagrama de polos y ceros, y si conocemos el diagrama de polos y ceros podemos obtener la señal que le corresponde. NOTA: Los polos están representados por una cruz azul.

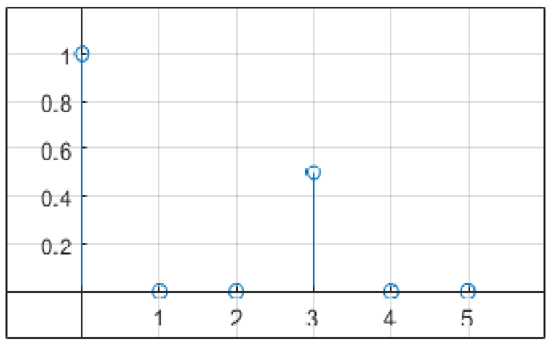


h[n] = [1,0,0.6,0,0,0.2,0,0.4,0,0]

$H(z) = 1 + 0.6 \cdot z^{-2} + 0.2 \cdot z^{-5} + 0.4 \cdot z^{-7}$

Ejemplo paso a paso

Obtener la Transformada z de la siguiente señal discreta (correspondiente a una respuesta al impulso)



El vector es

>> h = [1, 0, 0, 0.5, 0, 0];

La expresión H(z) será

$H(z) = 1 + 0.5\, z^{-3}$

Esta función es una división entre dos polinomios: el de ceros C(z) y el de polos P(z).

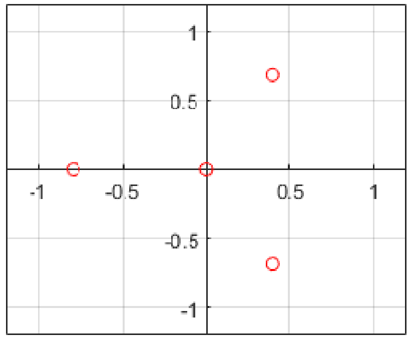
El polinomio de ceros es h (tal como está). Las raíces del polinomio de ceros serán los ceros de la Transformada. Escribimos lo siguiente con eco en pantalla

```
>> ceros = roots(h)
ceros =
   0.0000 + 0.0000i
   0.0000 + 0.0000i
  -0.7937 + 0.0000i
   0.3969 + 0.6874i
   0.3969 - 0.6874i
```

El polinomio de polos está constituido por 5 raíces, y no es necesario hacer ningún cálculo para saber que están en el origen.

Esto ya responde a lo pedido. Sólo falta graficarlo. Es muy sencillo de graficar a mano.

Aquí se muestra un gráfico ubicando los ceros en el plano complejo.

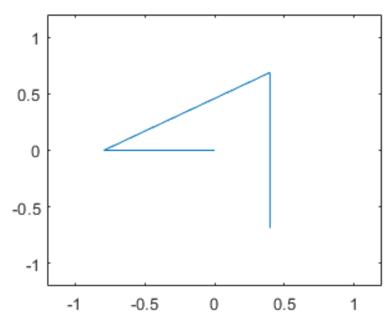


En el origen hay un cero doble y también un polo de orden 5.
NOTA: Estrictamente hablando, un polo y un cero en el mismo lugar se cancelan entre sí, por lo que en realidad en el origen quedaría un polo de orden 3. Por el momento no consideramos eso para no complicar innecesariamente la explicación paso a paso.

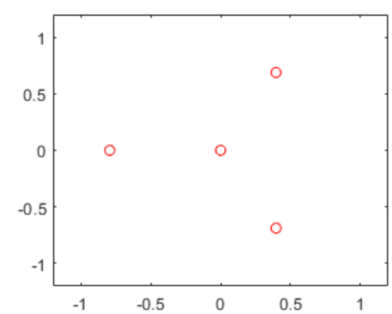
¿Cómo obtener el gráfico de ceros anterior en MATLAB?

Analicemos primero el uso de dos funciones de MATLAB llamadas real() e imag(). Estas funciones devuelven la parte real o la parte imaginaria de un número complejo, o de un vector de complejos.
Si se tiene el siguiente vector complejos = [2, 2+2i, 3i, 1+4i], resultará que
real(complejos) = [2, 2, 0, 1]
imag(complejos) = [0, 2, 3, 4]
Un modo de graficar los ceros es colocar como eje x del plot las partes reales del complejo y como eje y las partes imaginarias. Sin embargo, falta algo. La instrucción plot tiende a trazar una línea entre cada par de puntos que dibuja (para completar el contorno de lo que en realidad son puntos sueltos en cualquier gráfico de función). De modo que si hacemos lo anterior con un plot, dibujará los puntos, pero seguirá el contorno armando una especie de tela de araña.
Pero la instrucción plot tiene varios modificadores. Se puede cambiar el color de línea, por ejemplo, o si se desea graficar con línea punteada.
plot(x,y,'r') grafica en color rojo
plot(x,y,':') grafica en línea punteada
plot(x,y,'.') grafica colocando un punto en cada valor, sin unirlos entre sí
plot(x,y,'o') grafica colocando un círculo en cada valor, sin unirlos
plot(x,y,'x') grafica colocando una x en cada valor, sin unirlos

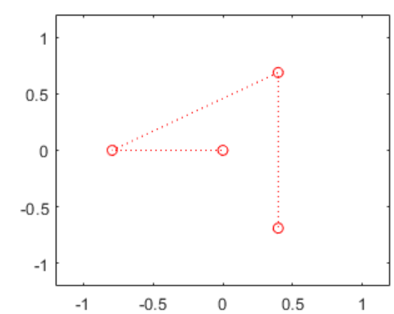
Estos modificadores pueden combinarse. Por ejemplo, plot(x,y,':or') graficará en rojo, colocando un círculo en el lugar de la función y unirá los distintos valores con línea de puntos.
Si tenemos la variable "ceros" del ejercicio de ejemplo anterior, podríamos obtener los siguientes gráficos. En todos ellos se utilizó luego la instrucción axis([-1.2 1.2 -1.2 1.2])



`plot(real(ceros),imag(ceros))`



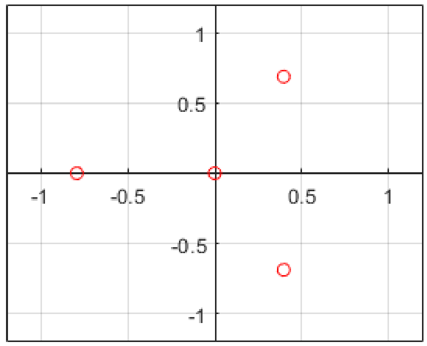
`plot(real(ceros),imag(ceros),'or')`



`plot(real(ceros),imag(ceros),':or')`

El que resulta preferible para graficar los ceros de la función es el del centro. A eso conviene agregarle los cambios en gca para que los ejes estén en el origen

```
>> ceros=roots(h);
>> plot(real(ceros),imag(ceros),'or')
>>axis([-1.2 1.2 -1.2 1.2])
>>ejes=gca; ejes.XAxisLocation = 'origin'; ejes.YAxisLocation = 'origin';
>> grid on
```



Factorización de polinomios

Consideremos la siguiente expresión matemática:$y = (x - 2).(x - 3)$

¿Para qué valores de x el valor de y será igual a cero?

Pues, justo cuando x sea igual a dos, el primer paréntesis valdrá cero, y si bien el segundo valdrá -1, resulta que cero por cualquier otro valor dará igual a cero. Sucederá lo mismo cuando x sea igual a tres. En ese caso el primer paréntesis valdrá uno, pero resulta que el segundo valdrá cero, y uno por cero será cero. No hay ningún otro valor de x que vuelva cero a esa función. Las raíces de esa ecuación son x1 = 2 y x2 = 3. Prestar atención a que si bien hay un signo menos dentro de los paréntesis, las raíces son positivas ya que indican el valor que debe tomar x para volver cero al paréntesis, y por consiguiente también a la función.

Ahora apliquemos la propiedad distributiva para desarrollar esa ecuación (multiplicando x por x, más x por -3, más -2 por x, más -2 por -3, y reordenemos luego los términos.

$$y = (x - 2).(x - 3) = x.x + x.(-3) + (-2).x + (-2).(-3)$$
$$y = x^2 - 3.x - 2.x + 6 = \ x^2 - 5.x + 6$$

Quedó un polinomio de grado 2 (una parábola), lo que no es sorprendente si asumimos antes que tenía dos raíces.

La ecuación es la misma que antes, sólo que está expresada de otra manera. Eso significa que sus raíces tienen que ser 2 y 3. Pongamos esto a prueba.

$$y(2) = 2^2 - 5\,.\,2 + 6 = 4 - 10 + 6 = 0$$
$$y(3) = 3^2 - 5\,.\,3 + 6 = 9 - 15 + 6 = 0$$

Podemos sacar algunas conclusiones de este análisis. Existen dos maneras de escribir la misma ecuación de una parábola. La manera normal (desarrollada) en la cuál se pueden ver claramente los coeficientes x^2 - 5x + 6, y la manera que se denomina "factorizada" (porque está dividida en factores que son las expresiones que están entre paréntesis) como la que utilizamos para comenzar el análisis (x-2).(x-3).

Factorizar un polinomio es hallar su expresión como producto de paréntesis, cada uno de los cuales deja claramente expresada una de las raíces de ese polinomio.

Con lo que sabemos hacer con MATLAB resulta sencillo factorizar un polinomio.

Por ejemplo, supongamos que se nos pide factorizar el siguiente polinomio$y = x^4 - 5.x^3 + 5.x^2 + 5.x - 6$

La tarea es encontrar las raíces x1, x2, x3, y x4, para expresar la función del siguiente modo

$$y = (x - x1).(x - x2).(x - x3).(x - x4).$$

Primero creamos el vector con los coeficientes del polinomio, luego obtenemos sus raíces

```
>> polinomio=[1 -5 5 5 -6];
```

```
>>roots(polinomio)
ans =
   3.0000
  -1.0000
   2.0000
   1.0000
```

La solución es entonces ***y = (x-3).(x+1).(x-2).(x-1)***. Es importante notar que el único signo + aparece en la raíz negativa. Eso se debe a que ese factor sería (x - (-1)) = (x + 1). Pensado de otro modo, cuando x sea -1 el factor (x+1) se vuelve cero.
Lo anterior es importante para la Transformada z porque permite expresar el polinomio de modo que queden bien claramente marcado cuáles son sus raíces.

6. Dado el siguiente polinomio en forma desarrollada, encuentre su expresión factorizada.

$$y = x^3 - 7.x^2 + 14. - 8$$

Rta; y = (x-1)(x-2)(x-4)

Multiplicación de polinomios

Analicemos ahora qué sucede cuando dos polinomios se multiplican, pensando sobre todo en qué podría esto significar cuando los polinomios representan señales temporales.

El caso más sencillo es multiplicar un polinomio cualquiera por otro de orden cero (un valor constante). En ese caso simplemente significa que amplificamos la señal en un factor igual al de la constante utilizada.
Tomemos como ejemplo el polinomio siguiente (que representa una señal de sólo 4 muestras.
$v(z) = 1 + 0.75\ z^{-1} + 0.5\ z^{-2} + 0.25\ z^{-3}$

¿Qué significa este polinomio? Que tenemos una señal que en el primer instante (t = 0) tiene un valor 1, en el siguiente instante (t = dt = 1/fs) tiene un valor 0.75, en el siguiente (t = 2.dt = 2/fs) vale 0.5 y en el último instante vale 0.25
Prestar atención al hecho de que representado de ese modo el exponente de z (cambiando el signo) se relaciona con un instante de tiempo. Así, si alguien me habla de un término $0.1z^{-6}$ debería entender que me está hablando de la muestra número 6 (que se produce en el instante t = 6.dt = 6/fs).

Si este polinomio se multiplica por otro de orden cero (por ejemplo w = 10). El resultado será

$v(z).w(z) = 10 + 7.5\ z^{-1} + 5\ z^{-2} + 2.5\ z^{-3}$

Lo que se relaciona con una señal 10 veces mayor que la original. Hasta acá, ninguna novedad ni nada llamativo.

¿Qué sucede si se multiplica por un polinomio de grado 1 muy simple $w(z) = z^{-1}$?
Quedará entonces como resultado

$v(z).w(z) = 0 + 1\ z^{-1} + 0.75\ z^{-2} + 0.5\ z^{-3} + 0.25\ z^{-4}$

¿Qué cambió? Es la misma secuencia, pero ahora sus valores se producen una muestra más tarde que antes.
Conclusión: Si se multiplica un polinomio de Transformada z por z^{-1}, esto equivale a producir un retardo de una muestra.

¿Y si multiplicamos el polinomio por otra potencia como z^{-12}?
Pues, resulta sencillo caer en la cuenta de que se obtiene una señal con un retardo de 12 muestras.

¿Y si multiplicamos el polinomio original por $w(z) = 1 + z^{-12}$?
Pues en ese caso sucede lo siguiente
$v(z).w(z) = v(z).(1 + z^{-12}) = v(z) + v(z).z^{-12}$

Esto es, todo el polinomio original v(z) se multiplica por cada uno de los términos del paréntesis y se suman los resultados.

Lo que se obtiene en este caso es la misma señal, sumada a una copia con un retardo de 12 muestras. Básicamente un eco con un retardo de 12 muestras.

Dicho en otras palabras, la función $h(z) = 1 + z^{-12}$podría considerarse como la respuesta al impulso de un eco de 12 muestras y la multiplicación de polinomios es equivalente a la convolución.

Revisemos esto desde otro punto de vista.
Si tenemos la multiplicación de dos polinomios (3x +5).(2x + 1), esta multiplicación requiere multiplicar cada término del primer paréntesis, con cada término del segundo y reacomodar los resultados.

$$(3x +5).(2x + 1) = 3x.2x + 3x.1 + 5.2x + 5 = 6 x^2 + 13 x +5$$

Ahora, si tuviésemos los polinomios expresados como vectores (tal como lo hace MATLAB);
v = [3, 5]
h = [2, 1]

¿Cuál es la operación que hacemos con esos dos vectores para obtener la multiplicación de polinomios?
No es igual a multiplicar los vectores.
Si hacemos la multiplicación punto a punto de los vectores obtendríamos

v.*h = [6, 5]

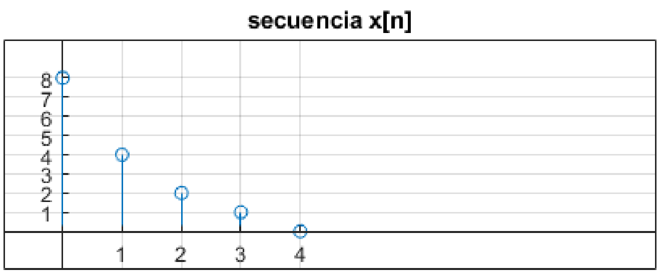
que no coincide con lo que habría pasado si multiplicábamos los polinomios. ¿Por qué esta diferencia? Porque no están las x en el polinomio, sino que sólo están representadas por el lugar que ocupan en el vector. Si multiplico la fila 1 de un vector por la fila 1 de otro vector, el resultado quedará en la fila 1. Sin embargo, si multiplico el coeficiente que está con x de un polinomio, con el coeficiente que está con x del otro polinomio, el resultado quedará en otra fila (x^2)

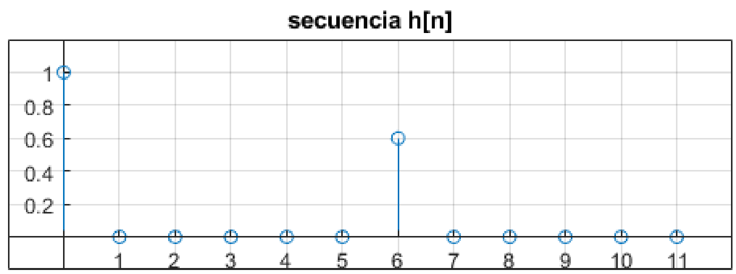
La operación que hay que hacer con los vectores para que el resultado coincida con la multiplicación de polinomios es justamente la convolución.

conv(v,h) = [6, 13, 5]

Un tema importante es que el resultado de la convolución es más largo que cada uno de los vectores. De hecho, su longitud es igual a la suma de ambas longitudes menos uno. Si bien el ejemplo usado tenía vectores de sólo dos componentes esta regla es general.

7. Ejercicio de multiplicación de polinomios - convolución
Dadas las siguientes secuencias de valores (señales) x[n] y h[n], obtener su convolución y[n] = x[n] * h[n] y graficarla.





NOTA; Lo anterior puede interpretarse como una señal de audio y la respuesta al impulso de un sistema que provoca un eco con un retardo de 6 muestras.

NOTA: Se muestra a continuación el código para representar el gráfico de la derecha, como ejemplo del uso de xticks e yticks, para elegir los lugares precisos donde estarán las marcas en la grilla y los valores de las escalas.

```
h = [1 0 0 0 0 0 0.6 0 0 0 0 0];
n = 0:length(h)-1;
stem(n,h)
ejes=gca; ejes.XAxisLocation = 'origin'; ejes.YAxisLocation = 'origin';
yticks([0:0.2:1])  % El argumento es un vector que comienza en 0, salta de a 0.2 y termina en 1
xticks(n)% El argumento es un vector que comienza en 0, salta de a 1 y termina en la longitud de h menos 1
axis([-1 12 -0.2 1.2])
grid on
```

División de polinomios

La división de polinomios genera un problema que requiere prestar atención.
Para entenderlo repasaremos lo que sucede con los números enteros (los que no tienen decimales).
Si solamente se conocieran los números enteros, hay divisiones que no podrían hacerse.
No hay problema en hacer 6/2, pero hacer 2/6 sería imposible.

Algo semejante sucede con los polinomios.
Algunas divisiones entre polinomios dan por resultado un polinomio normal, pero otras no podrían completarse, a menos que postulemos algo semejante a "polinomios con decimales".
Esto se entenderá dentro de pocas líneas, pero para ir anticipándolo mostramos un "polinomio normal" y un "polinomio con decimales" (NOTA: El nombre de polinomio con decimales es de fantasía y no se utiliza en la literatura técnica).

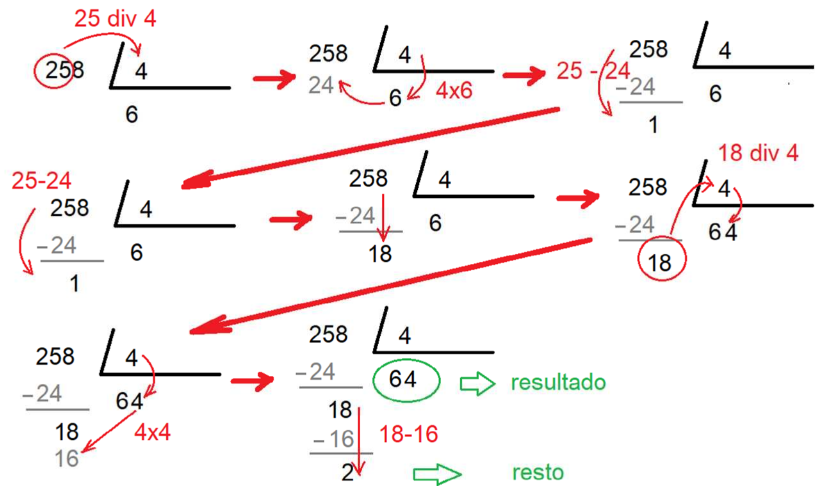
POLINOMIO "NORMAL" $P(x) = 6\,x^2 + 13\,x + 5$

POLINOMIO "CON DECIMALES" $P(x) = x + 5 + 4\,x^{-1} + 8\,x^{-2} + x^{-3}$

Incluso es posible pensar en polinomios "tipo números periódicos" en donde sus términos con exponente negativo de las x no finalicen nunca.

POLINOMIO "PERÍODICO" $P(x) = 5 + 4\,x^{-1} + 5\,x^{-2} + 4\,x^{-3} + 5\,x^{-4} + 4\,x^{-5} + \ldots$

Pero ... ¿Cómo se dividen los polinomios?
Para eso primero tenemos que recordar cómo se dividen los números en la escuela primaria.
Calculemos a mano la división 258/4



Se toma la primera cifra (2) y se intenta dividirla por 4. Como no da resultado entero se consideran las dos primeras cifras (25) y se repite el intento. Para anotar la primera cifra del resultado se analiza por qué valor habría que multiplicar al 4 para llegar cerca de 25 (no superarlo!). El valor es 6, ya que 6x4 = 24, con lo que sobrará 1 (ver Figura arriba a la derecha). Se escribe el resultado alcanzado 24 justo debajo del que queríamos conseguir 25, y se restan ambos valores, por lo que sobra 1. Como ese 1 no puede dividirse por 4 (con resultado entero), se agrega la siguiente cifra del número original (8), por lo que queda el número 18 para dividirlo por 4. Si bien no tiene resultado entero es posible aproximarse (por debajo) llegando a 16 = 4x4. Con lo cual se anota un 4 junto al 6 en el resultado (queda 64) y se resta el resultado que se buscaba 18 menos el alcanzado 16, lo que da un sobrante de 2. En primaria se dice que esa división tiene "resto 2".
El resto es la parte que no puedo dividir
El resultado final completo podría expresarse del siguiente modo

resto

$$\frac{258}{4} = 64 + \frac{2}{4}$$

resultado

Si el resto es igual a cero, se dice que se obtuvo una "división entera".

Luego de aprender los números decimales, se nos permite seguir con la división. Eso quiere decir que ese 2 que no podíamos dividir entre 4 lo consideramos un 20 (bajando el cero después de la coma de 258,0). Si probamos de dividir 20 entre 4, esta vez la división da entera y el resultado es 5 (ya que 5x4 = 20), pero ese valor 5 lo escribimos después de la coma en el resultado final que termina quedando 64,5.

Con los polinomios sucederá algo semejante.
Habrá situaciones en las cuales p(x)/q(x) dará un resultado R (con resto r = 0), pero habrá otras en que habrá resto distinto de cero y se dice entonces que no puede hacerse una "división entera".

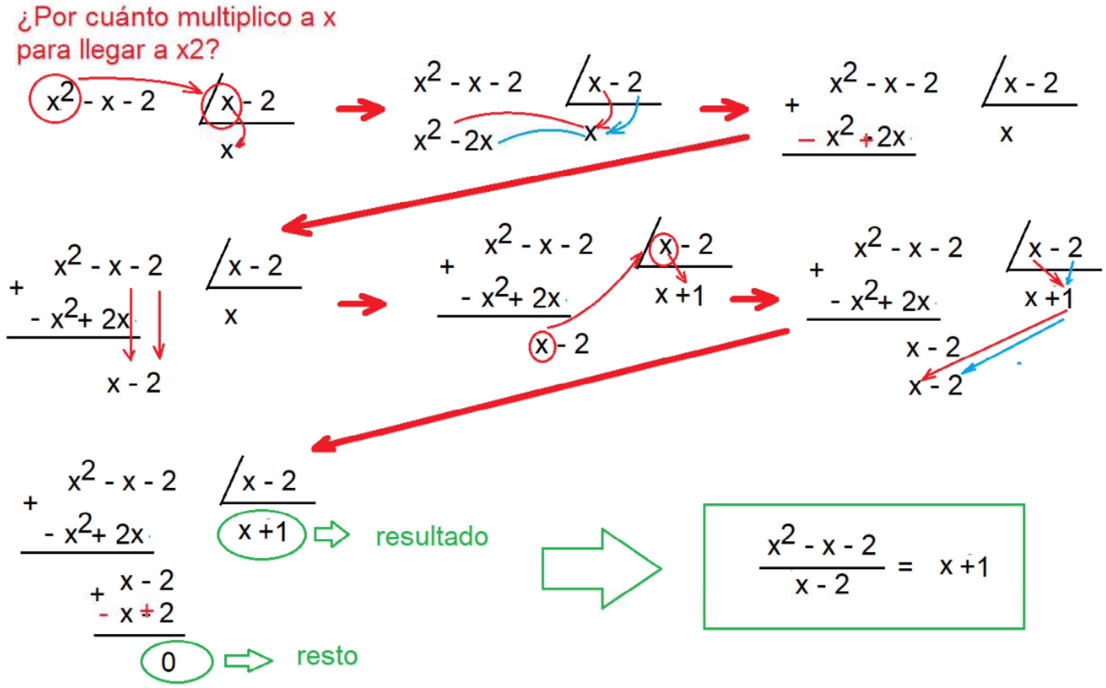
resto

$$\frac{p(x)}{q(x)} = R(x) + \frac{r(x)}{q(x)}$$

resultado

Veamos cómo sería la división de p(x) / q(x)
$p(x) = x^2 - x - 2$
$q(x) = x - 2$

En la figura puede verse que considero el primer término de p(x) y me pregunto: ¿por qué valor debería multiplicar al primer término de q(x) que vale "x" para que me de ese mismo valor, que vale "x^2"? Debo multiplicar por una x, por lo cual coloco ese resultado como "primera cifra" del resultado de la división. Pero ahora debo comprobar si llegué a cubrir lo que pretendía (ya que solamente me fijé en las "primeras cifras" de ambos polinomios.
Para comprobar a cuánto llegué (y cuánto me falta), multiplico esa "x" que acabo de colocar por todo q(x) y anoto el resultado debajo de p(x). Resto p(x) - q(x) para saber cuánto me falta, luego bajo una nueva "cifra" de p(x) y sigo el proceso.



En el ejemplo anterior, la división era entera, por lo que obtuvimos que el resultado es R(x) = (x+1)

Para comprobarlo podríamos verificar si es cierto que R(x).Q(x) = P(x)

Si hacemos (x+1).(x-2) podremos ver que se llega al polinomio P(x). Esto puede hacerse manualmente (aplicando distributiva) o bien convirtiendo los polinomios en vectores R = [1, 1] y Q = [1, -2], y aplicando conv(R,Q).

MATLAB permite dividir polinomios utilizando la función deconv(), que significa "deconvolution". Hay que tener presente que si bien en la convolución puede cambiarse el orden de los argumentos ya que conv(p,q) = conv(q,p), no sucede lo mismo con la deconvolución. En deconv() el primer argumento es el polinomio de arriba (numerador) y el segundo argumento es el de abajo (denominador).

Ejemplo de división en MATLAB, dejando la última línea con eco en pantalla

```
>> P = [1, -1. -2];
>> Q = [1. -2];
>>deconv(P,Q)
```

```
ans =
    1     1
```

El resultado de la división es R(x) = x + 1

¿Qué pasaría si la división no es entera?

```
>> P = [1, -1. -1];
>> Q = [1. -2];
>>deconv(P,Q)
```

```
ans =
    1     1
```

CUIDADO!!!! Generó el mismo resultado. Sin embargo falta algo. El tema es que deconv() devuelve dos variables. La primera es el resultado de la división (entera más cercana) y la segunda es el resto (r minúscula, podríamos usar).
Lo correcto sería hacer lo siguiente

```
>> P = [1, -1. -1];
>> Q = [1. -2];
>> [R r] = deconv(P,Q)
```

```
R =
    1     1
r =
    0     0     1
```

Esto significa que R(x) = (x + 1) y r(x) = $0.x^2 + 0.x + 1$

MATLAB entrega el resto con igual cantidad de términos que P(x) ya que es algo que habría que sumar al resultado entero de la división para obtener P(x).
Repetimos aquí una relación que se mostró más arriba.

$$\frac{p(x)}{q(x)} = \underbrace{R(x)}_{\text{resultado}} + \overbrace{\frac{r(x)}{q(x)}}^{\text{resto}}$$

Si multiplicamos todos los términos por Q(x) tendremos

$$P(x) = R(x).Q(x) + r(x)$$

Volvamos a calcular con MATLAB la división entera para verificar si el resto r(x) es cero.

```
>> P = [1, -1. -2];
>> Q = [1. -2];
>> [R r] = deconv(P,Q)
```

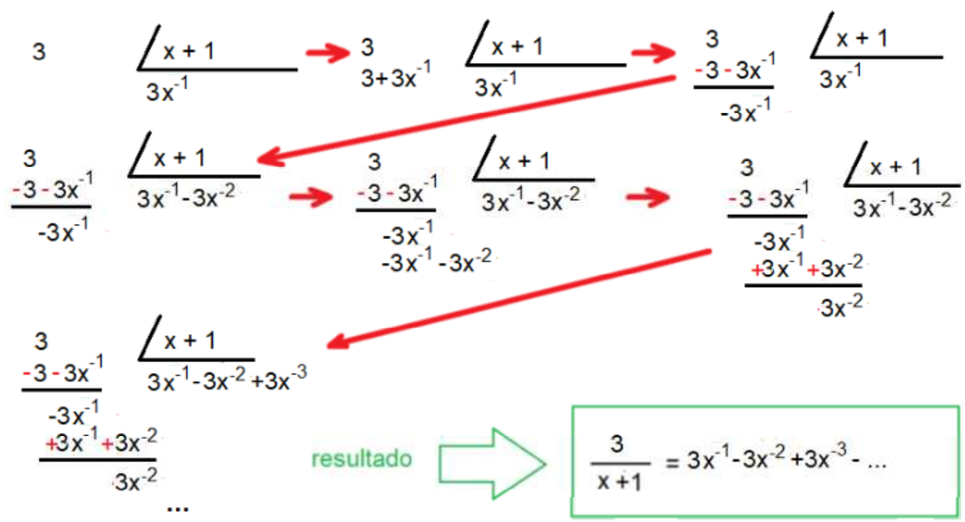
```
R =
    1     1

r =
    0     0     0
```

Vemos que r(x) es 0, y por lo tanto esa es una división entera.

NOTA: Si pensamos que la convolución entre un archivo de audio y una respuesta al impulso, nos da el audio procesado, podríamos entonces deconvolucionar el audio procesado con la respuesta al impulso para volver a obtener el audio original. Pero cuidado! Esto será cierto cuando la división se entera. Y si lo hago con archivos reales de audio, es sacarse la lotería conseguir que la división sea entera. Sería como elegir dos números de varias cifras al azar y al dividirlos pretender que la división de un resultado entero (sin decimales).

¿Cómo sería una división no entera hecha "a mano"?
Supongamos que P(x) = 3 y Q(x) = x +1
Queremos obtener P(x) / Q (x)
Tomamos el primer término (único en este caso), de P(x) y nos preguntamos por cuánto habría que multiplicar al primer término de Q(x) para igualarlo. Como el primer término de P(x) es 3 y el primero de Q(x) es x, deberíamos multiplicar a x por $3.x^{-1}$ (Esto es lo que podríamos considerar "decimales" en un polinomio: los términos con x con exponente negativo). Luego de escribir $3.x^{-1}$ como primer "cifra" del resultado, tenemos que comprobar si es exacta la división o si nos falta algo. Para esto multiplicamos efectivamente a $3.x^{-1}$ por todo Q(x), lo que nos da $3 + 3.x^{-1}$. Colocamos esto debajo del polinomio P(x) y restamos para ver cuánto "nos faltó". Al restar la primera columna queda 3-3 y da cero, pero al restar la columna tengo que considerar $0.x^{-1} - 3.x^{-1}$ y eso da $- 3.x^{-1}$. Por lo tanto, debo seguir con la división. Al continuar vemos que se obtiene un polinomio "periódico".



¿Y a qué se correspondería un polinomio "periódico" en audio?
A un archivo con extensión infinita (sus muestras no terminan nunca)
¿Es esto razonable?
En muchos casos lo es. Recordemos que uno de los usos que esto tendrá será para representar una respuesta al impulso. La respuesta al impulso "teórica" de un filtro real suele ser infinita.
¿Suena esto extraño? Lo típico de la respuesta el impulso es que cada nueva muestra sea una fracción de la muestra anterior.
Por ejemplo: [1, 0.5, 0.25, 0.125, 0.0625, ...]
En este ejemplo cada muestra es la mitad de la anterior. ¿Cuándo termina? A nivel teórico, jamás. Aunque a nivel práctico, cuando el nivel sea muy bajo podemos ignorar lo que sigue.

Lo normal en un filtro es que no disminuya tan rápido como a la mitad en cada muestra, sino por ejemplo al 90%. Esto es igual que multiplicar cada nueva muestra de la respuesta al impulso por 0.9 de la anterior, por lo que queda

$h = [1, 0.9, 0.9^2, 0.9^3. ...]$

Pues resulta que esto tiene una interesante aplicación. La transformada z del vector h (de respuesta al impulso) que mostramos recién como ejemplo es

$$H(z) = \frac{z}{z - 0.9}$$

Aún no tienen que saber por qué es ese resultado. Pero con lo que sabemos estamos en condiciones de comprobarlo. Lo que hay que hacer es dividir a mano el polinomio P(x) = z, por el de Q(x) = z - 0.9
Si lo hacen verán que queda

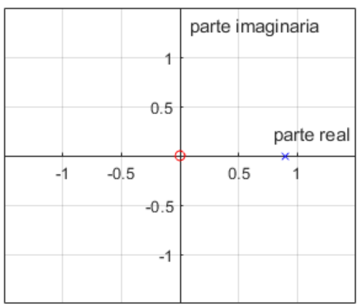
$$H(z) = \frac{z}{z - 0.9} = 1 + 0.9z^{-1} \ + 0.9^2 z^{-2} + \ +0.9^3 z^{-3} + \cdots$$

Mirando la expresión de la división de polinomios podemos saber cuál es su configuración de polos y de ceros. Recordemos que los ceros son los valores de z que vuelven cero al numerador (arriba) y los polos son los valores de z que vuelven cero al denominador (abajo). Esta función H(z) tiene un cero en z = 0, y tiene un polo en z = 0.9
La siguiente figura muestra la respuesta al impulso h[n] de ese filtro pasa bajos, el espectro de Fourier (suponiendo una frecuencia de muestreo de fs= 48 kHz) y la Transformada z con sus polos y ceros.

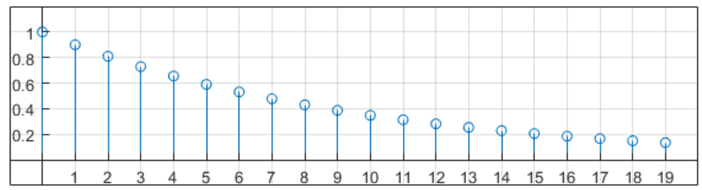
Función de transferencia como Transformada z

$$H(z) = \frac{z}{z - 0.9}$$

$$H(z) = 1 + 0.9z^{-1} \ + 0.9^2 z^{-2} + \ +0.9^3 z^{-3} + \cdots$$

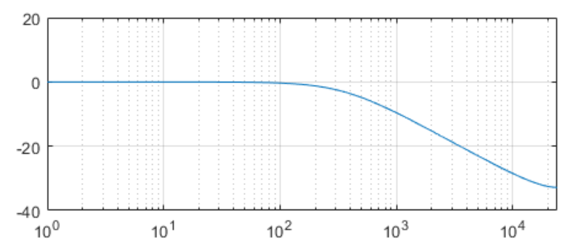


Respuesta al impulso h[n]



Función de Transferencia como espectro de Fourier

H(f)



Resumen hasta aquí

Una secuencia de números (audio digital, por ej.) puede representarse como un polinomio F(z) con exponentes negativos.
Algunos polinomios con infinitos términos pueden ser expresados como división de dos polinomios finitos F(z) = C(z)/P(z)
El polinomio numerador C(z) es el de "ceros"
El polinomio denominador P(z) es el de "polos"
Conocer los valores de los ceros y los polos, es equivalente a conocer toda la secuencia original

La Transformada z es una expresión matemática en función de z, normalmente descripta como división de dos polinomios escritos en forma factorizada de modo que puedan verse de modo sencillo sus polos y ceros.
Un caso de ejemplo es el siguiente, en el que se muestra una transformada y la señal discreta temporal que le corresponde.

Transformada	**Antitransformada**
(z) = z/(z-0.5)	$x[n] = 0.5^{n}$

La figura que sigue muestra una tabla de secuencias (señales) y sus transformadas. El ejemplo anterior es el tercero de la tabla con b=0.5. Hay dos símbolos en la tabla que merecen explicación. Uno es δ[n] que corresponde a un impulso, que tiene valor 1 para n=0 y es cero en todas las demás muestras. Su transformada z vale 1 (no tiene ningún polo ni ningún cero). La otra expresión que merece explicación es u[n] y es la señal que se denomina "escalón" (step en inglés). Se trata de una señal que para n=0 vale 1 y sigue valiendo uno para todas las muestras siguientes.

	Sequence	z - transform
1	$\delta[n]$	1
2	$u[n]$	$\frac{z}{z-1}$
3	b^n	$\frac{z}{z-b}$
4	$b^{n-1}u[n-1]$	$\frac{1}{z-b}$
5	e^{an}	$\frac{z}{z-e^a}$
6	n	$\frac{z}{(z-1)^2}$
7	n^2	$\frac{z(z+1)}{(z-1)^3}$
8	$b^n n$	$\frac{bz}{(z-b)^2}$
9	$e^{an}n$	$\frac{ze^a}{(z-e^a)^2}$
10	$\sin(an)$	$\frac{\sin(a)\,z}{z^2-2\cos(a)\,z+1}$
11	$b^n\sin(an)$	$\frac{\sin(a)\,b\,z}{z^2-2\cos(a)\,b\,z+b^2}$
12	$\cos(an)$	$\frac{z(z-\cos(a))}{z^2-2\cos(a)\,z+1}$
13	$b^n\cos(an)$	$\frac{z(z-b\cos(a))}{z^2-2\cos(a)\,b\,z+b^2}$

Table 9.1. z-transforms of some common sequences.