* **myFilterFIR():** misma aplicación que “filter()” (ver Sección 5.1.2). Permite filtrar una secuencia de datos utilizando un filtro digital.Sólo es válido si mi filtro es FIR (Ya que el denominador es 1); si fuera IIR no sería válido.

De esta manera, la función de transferencia de nuestro filtro FIR será:

**\*Código en Scilab de myFilterFIR():**

**function y=myFilterFIR(VN, VD, x)**

*//Recibo 2 vectores REALES (VN:VectorNumerador y VD:VectorDenominador) y la señal de entrada (x).* *La salida de myFilter() es la señal x filtrada.*

*//Sacamos longitudes:*

**LongDen=length(VD);** *//Longitud del vector del DENOMINADOR.*

**LongNum=length(VN);** *//Longitud del vector del NUMERADOR.*

**LongX=length(x);** *//Longitud de nuestra función de entrada x.*

**Maximo=max(LongNum,LongDen);** *//Obtenemos el valor máximo entre ambas longitudes.*

**x2=[zeros(Maximo-1,1);x'];** *//Usamos x2 como auxiliar. zeros(2,3) me devuelve una matriz de ceros de 2 filas x 3 columnas. x' es la traspuesta de x. El ";" es para concatenar ambos vectores. Esto lo hacemos porque en el for se abajo (al sacar y[n]) se necesita que los dos vectores tengan el mismo tamaño; entonces agrandamos el tamaño de x rellenándolo con 0s.*

*//Ahora hacemos un bucle for para obtener nuestra salida y[n]. Este for está basado en nuestra ecuacion H(z)=B(Z)/A(Z) --> ver help filter(). Va de n=Maximo HASTA Maximo+Nx-1.*

**for n=Maximo:LongX+Maximo-1**

**y(n)=( (VN(LongNum:-1:1))\*(x2(n-LongNum+1:n)) ) / (VD(1));**

*//Al hacer: 8:-1:4 (ejemplo) nos devuelve 8,7,6,5,4 (disminuimos el 8 en 1 hasta llegar al 4.*

**end**

*//Salida...*

**y2=y(Maximo:LongX+Maximo-1);** *//Nos sirve para eliminar los 0s.*

**y=y2';** *//La trasponemos y obtenemos nuestra señal x ya filtrada = señal y.*

**endfunction**