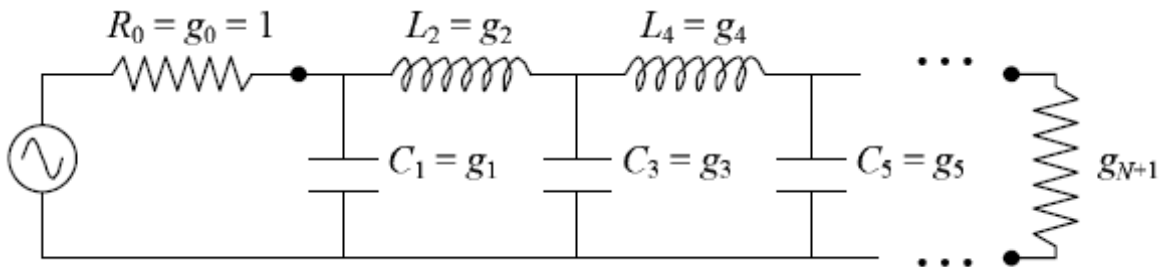


Anexo

Tablas para el diseño de filtros

A continuación se presentan unas tablas en las que se encuentran los valores tabulados de los g_k que aparecen en el diseño de cualquier filtro. Pero primeramente debemos recordar que representan los valores g_k .



Viendo lo anterior podemos escribir los valores de las inmitancias:

Respuesta máximamente plana ($g_0=1, \omega_c=1$)									
N	g_1	g_2	g_3	g_4	g_5	g_6	g_7	g_8	g_9
	g_{10}	g_{11}							
1	2.0000	1.0000							
2	1.4142	1.4142	1.0000						
3	1.0000	2.0000	1.0000	1.0000					
4	0.7654	1.8478	1.8478	0.7654	1.0000				
5	0.6180	1.6180	2.0000	1.6180	0.6180	1.0000			
6	0.5176	1.4142	1.9318	1.9318	1.4142	0.5176	1.0000		
7	0.4450	1.2470	1.8019	2.0000	1.8019	1.2470	0.4450	1.0000	
8	0.3902	1.1111	1.6629	1.9615	1.9615	1.6629	1.1111	0.3902	1.0000
9	0.3473	1.0000	1.5321	1.8794	2.0000	1.8794	1.5321	1.0000	0.3473
10	0.3129	0.9080	1.4142	1.7820	1.9754	1.9754	1.7820	1.4142	0.9080
	0.3129	1.0000							

Respuesta igual rizado 0.5dB ($g_0=1, \omega_c=1$)									
N	g_1	g_2	g_3	g_4	g_5	g_6	g_7	g_8	g_9
	g_{10}	g_{11}							
1	0.6986	1.0000							
2	1.4029	0.7071	1.9841						
3	1.5963	1.0967	1.5963	1.0000					
4	1.6703	1.1926	2.3661	0.8419	1.9841				
5	1.7058	1.2296	2.5408	1.2296	0.7058	1.0000			
6	1.7254	1.2479	2.6064	1.3137	2.4758	0.8696	1.9841		

7	1.7451	1.2583	2.6381	1.3444	2.6381	1.2583	1.7372	1.0000	
8	1.7451	1.2647	2.6564	1.3590	2.6964	1.3389	2.5093	0.8796	1.9841
9	1.7504	1.2690	2.6678	1.3673	2.7239	1.3673	2.6678	1.2690	1.7504
	1.0000								
10	1.7543	1.2721	2.6754	1.3725	2.7392	1.3806	2.7231	1.3485	2.5239
	0.8842	1.9841							

Respuesta igual rizado 3dB ($g_0=1, \omega_c=1$)									
N	g_1	g_2	g_3	g_4	g_5	g_6	g_7	g_8	g_9
	g_{10}	g_{11}							
1	1.9953	1.0000							
2	3.1013	0.5339	5.8095						
3	3.3487	0.7117	3.3487	1.0000					
4	3.4389	0.7483	4.3471	0.5920	5.8095				
5	3.4817	0.7618	4.5381	0.7618	3.4817	1.0000			
6	3.5045	0.7685	4.6061	0.7929	4.4641	0.6033	5.8095		
7	3.5182	0.7723	4.6386	0.8039	4.6386	0.7723	3.5182	1.0000	
8	3.5277	0.7745	4.6575	0.8089	4.6990	0.8018	4.4990	0.6073	5.8095
9	3.5340	0.7760	4.6692	0.8118	4.7272	0.8118	4.6692	0.7760	3.5340
	1.0000								
10	3.5384	0.7771	4.6768	0.8136	4.7425	0.8164	4.7260	0.8051	4.5142
	0.6091	5.8095							

Retardo máximamente plano ($g_0=1, \omega_c=1$)									
N	g_1	g_2	g_3	g_4	g_5	g_6	g_7	g_8	g_9
	g_{10}	g_{11}							
1	2.0000	1.0000							
2	1.5774	0.4226	1.0000						
3	1.2550	0.5528	0.1922	1.0000					
4	1.0598	0.5116	0.3181	0.1104	1.0000				
5	0.9303	0.4577	0.3312	0.2090	0.0718	1.0000			
6	0.8377	0.4116	0.3158	0.2364	0.1480	0.0505	1.0000		
7	0.7677	0.3744	0.2944	0.2378	0.1778	0.1104	0.0375	1.0000	
8	0.7125	0.3446	0.2735	0.2297	0.1867	0.1387	0.0855	0.0282	1.0000
9	0.6678	0.3203	0.2547	0.2184	0.1859	0.1506	0.1111	0.0682	0.0230
	1.0000								
10	0.6305	0.3002	0.2384	0.2066	0.1808	0.1539	0.1240	0.0911	0.0557
	0.0187	1.0000							

Programa MATLAB para filtro con acopladores

```
function filtros(fp1,fp2,fs1,fs2,Ap,As)
```

```
fprintf('Esta aplicacion permite calcular filtros de microondas con respuestas Butterworth y Chebyshev tabulada tal y como aparecen en los apuntes de microondas \n');
```

```
a = input('Por favor introduzca el valor de Z0 \n','s');
Z0 = sscanf(a,'%f');
```

```
%Solicitamos el valor de Z0
```

```
fprintf('Que tipo de filtro desea diseñar: \n')
fprintf(' 1.-Butterworth \n')
fprintf(' 2.-Chebyshev (0.5dB de rizado) \n')
fprintf(' 3.-Chebyshev (3dB de rizado) \n')
a = input('','s');
tipo = sscanf(a,'%d');
```

```
%Solicitamos el tipo del filtro
```

```
f0 = sqrt(fp2*fp1);
fa = (f0^2)/fs2;
fb = (f0^2)/fs1;
```

```
%Calculamos f0 y los bordes de
%las bandas de rechazo
```

```
if(fa > fs1)
    fs1 = fa;
else
    fs2 = fb;
end
```

```
%Creamos simetria en la banda de rechazo
%respecto a f0
```

```
ws=(fs2-fs1)/(fp2-fp1);
```

```
%Calculamos la frecuencia de corte en el
%equivalente paso de baja
```

```
epsilon = sqrt(10^(0.1*Ap)-1);
delta = sqrt(10^(0.1*As)-1);
```

```
%Calculo de epsilon
%Calculo de delta
```

```
if (tipo == 1)
    A = ((log10((delta^2)/(epsilon^2)))/(2*log10(ws)));
else
    A = acosh(sqrt((delta^2)/(epsilon^2)))/acosh(ws);
end
```

```
%Calculamos el orden del filtro
%segun la eleccion determinada
```

```
N = floor(A);
```

```
%Tomamos la parte entera de A
```

```
if((A - N) > 0)
    N = N + 1;
end
```

```
%Comprobamos si A tiene decimales
%en cuyo caso tomamos el entero superior
```

```
fprintf('El orden del filtro seleccionado es: %d',N)
```

```
%Presenta por pantalla el orden del filtro
```

```
fprintf('Los filtros Chebyshev de orden par presentan problemas de adaptacion a la salida.')
```

```
%Advierte de los
```

```
a = input('Introduzca el valor del filtro que desee (recomendamos un orden impar)\n','s');
```

```
%problemas que
```

```
%presentan los
```

```
%filtros Chebyshev
```

```
%de orden par
```

```
N = sscanf(a,'%d');
```

```
DELTA = (fp2-fp1)/f0;
```

```
%Calculo de DELTA
```

```
g = coeficientes(tipo,N);
```

```
%Tomamos los coeficientes de la base de datos
```

```
for i = 1:N + 1
```

```
%Calculamos los valores de los transformadores
```

```
if (i == 1)
    J(i) = sqrt(pi*DELTA/(2*g(i)))/Z0;
    Zoe(i) = Z0+J(i)*(Z0^2)+(J(i)^2)*(Z0^3);
    Zoo(i) = Z0-J(i)*(Z0^2)+(J(i)^2)*(Z0^3);
elseif (i == N+1)
    J(i) = sqrt(pi*DELTA/(2*g(i)*g(i-1)))/Z0;
    Zoe(i) = Z0+J(i)*(Z0^2)+(J(i)^2)*(Z0^3);
    Zoo(i) = Z0-J(i)*(Z0^2)+(J(i)^2)*(Z0^3);
else
    J(i) = (pi*DELTA/(2*sqrt(g(i-1)*g(i))))/Z0;
    Zoe(i) = Z0+J(i)*(Z0^2)+(J(i)^2)*(Z0^3);
    Zoo(i) = Z0-J(i)*(Z0^2)+(J(i)^2)*(Z0^3);
end
end

Zoo          %Presenta los resultados por pantalla
Zoe
```