UNIVERSIDAD NACIONAL DE CÓRDOBA

FACULTAD DE CIENCIAS EXACTAS, FÍSICAS Y NATURALES

CÁTEDRA DE SÍNTESIS DE REDES ACTIVAS

TRABAJO PRÁCTICO DE LABORATORIO N°4: FILTROS ACTIVOS

Grupo N°10

Alumnos:

Fernández Segovia, Sergio Andrés CI: 10728127

Richter, Juan Bautista DNI: 45375789

Profesor:

Ing. Ferreyra, Pablo

Noviembre / 2025

Índice

1. Introducción

La *Teoría de Filtros Activos de Segundo Orden* aborda aspectos fundamentales del procesamiento analógico de señales. Mediante una Etapa Bicuadrática, es posible ajustar parámetros importantes, tales como la Frecuencia del Polo (w_p) y el Factor de Calidad del Polo (Q_p), para así obtener respuestas Pasa Bajos, Pasa Altos o Pasa Banda, según la estructura utilizada.

Para obtener la *Función del Filtro* a sintetizar mediante una bicuadrática, existen distintas aproximaciones; entre las más utilizadas se pueden mencionar las de *Chebyshev* y *Butterworth*. La aproximación por polinomios de *Chebyshev* se distingue por presentar, a igual orden, una pendiente de rechazo más pronunciada respecto a la de *Butterworth*, a cambio de una cierta ondulación o rizado en la banda de paso. Estas propiedades teóricas condicionan las decisiones de diseño y la selección de topologías cuando se busca cumplir especificaciones prácticas.

2. Objetivos

El presente trabajo de laboratorio tiene como objetivo sintetizar un *Filtro Activo*, cuya función de atenuación se debe obtener mediante una *Aproximación de Polinomios de Chebychev*, para posteriormente evaluar la *Sensibilidad* del mismo ante variaciones de los componentes seleccionados para su implementación. El filtro a sintetizar debe cumplir con las especificaciones correspondientes a la plantilla que se muestra a continuación.

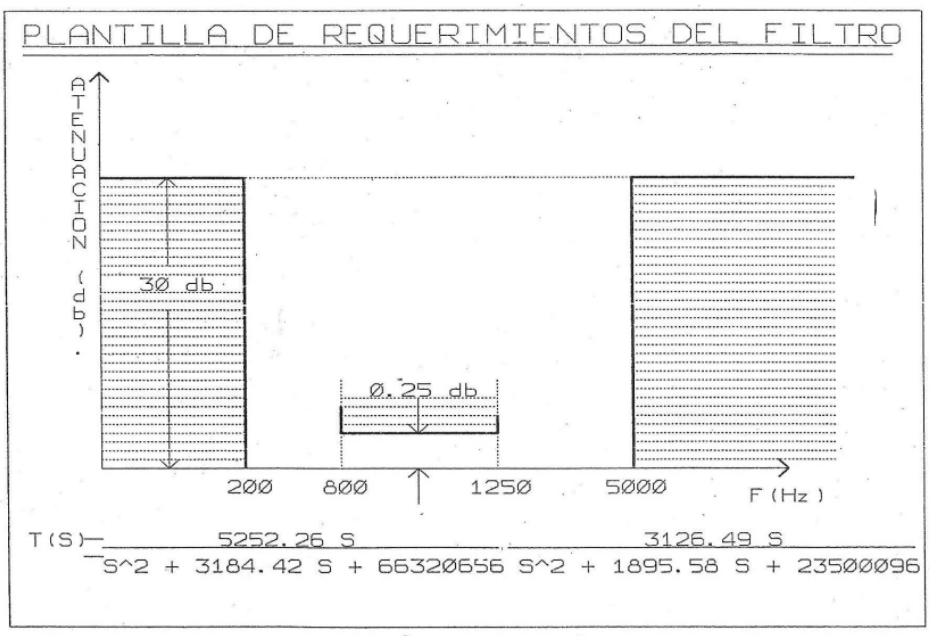

Figura 1: Plantilla de Requerimientos

3. Desarrollo

3.1. Aproximación de Chebyshev

De la plantilla proporcionada, se puede observar que el filtro a desarrollar es un *Pasa Banda* y la *Banda de Paso* está comprendida entre las frecuencias de 800 [Hz] y 1250 [Hz]. Asimismo, se detallan las *Bandas de Rechazo*, conformadas por las frecuencias inferiores a 200 [Hz] y superiores a 5000 [Hz].

Primeramente, se debe encontrar la *Función de Transferencia* del filtro en cuestión; para ello, se utiliza como herramienta de cálculo el entorno de programación que ofrece *Matlab*. A continuación, se muestra el código realizado; en el mismo, luego de definir las *Bandas de Paso y de Rechazo*, se determina el orden mínimo del filtro, parámetro requerido para que, a través del comando *cheby1*, se pueda obtener la *Función de Transferencia* buscada.

```
clc; clear; close all;

%%
% ======================
%     ESPECIFICACIONES
% ======================
fp1 = 800;          % Frecuencia inferior Pasabanda
fp2 = 1250;         % Frecuencia superior Pasabanda
fs1 = 200;          % Frecuencia inferior Rechazabanda
fs2 = 5000;         % Frecuencia superior Rechazabanda

Ap = 0.25;          % Atenuacion en la Banda de Paso
As = 30;            % Atenuacion en la Banda de Rechazo

%%
% ==================
%     DESARROLLOS
% ==================
% Convertir las frecuencias de [Hz] a [rad/seg]
Wp = 2 * pi * [fp1 fp2];
Ws = 2 * pi * [fs1 fs2];

% Calcular Orden Minimo del Filtro (n) y Frecuencia Normalizada (Wc)
[n, Wc] = cheb1ord(Wp, Ws, Ap, As, 's');
fprintf('Orden minimo requerido: n = %d\n', n);

% Disenar Filtro de Chebyshev Tipo 1
[num, den] = cheby1(n, Ap, Wc, 'bandpass', 's');
Filtro     = tf(num, den);
fprintf('La Funcion de Transferencia obtenida es:\n');

% Descomponer la Funcion de Transferencia en Bicuadraticas
[sos, g] = tf2sos(num,den);

PasaBajo = tf(2 * g * sos(1,1:3),sos(1,4:6));
PasaAlto = tf(1 / 2 * sos(2,1:3),sos(2,4:6));
```

```
                    1.642e07 s^2
-----------------------------------------------------
s^4 + 5080 s^3 + 9.586e07 s^2 + 2.006e11 s + 1.558e15
```

Figura 2: Función de Transferencia

Otro detalle que se observa en el código es que se descompone la *Función de Transferencia* mediante el comando *tf2sos*, ésto con el objeto de poner en evidencia que dicha función puede expresarse como el producto de secciones bicuadráticas, concretamente, en el producto de un *Filtro Pasa Bajo* y un *Pasa Alto*.

Una vez cumplido este punto, se procede a verificar si el filtro calculado cumple con las especificaciones planteadas inicialmente. En ese sentido, se presenta una tabla en la cual se detallan las magnitudes para las frecuencias de interés, así como el *Diagrama de Bode* elaborado por *Matlab*, para facilitar la visualización de las *Bandas de Paso y de Rechazo* y las *Secciones Bicuadráticas*.

Frecuencia [Hz]	**Magnitud [dB]**
200	-34.832
800	-0.25
1250	-0.25
5000	-34.832

Cuadro 1: Tabla Resumen

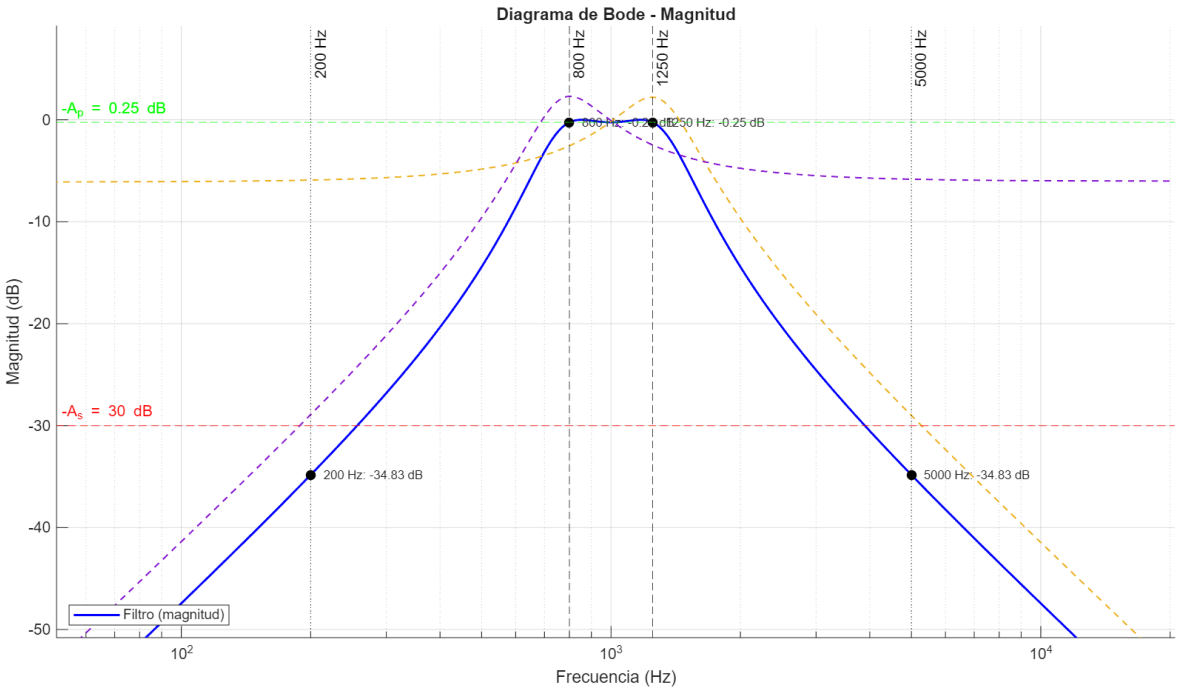

Figura 3: Diagrama de Bode

3.2. Topología Bicuadrática

El siguiente punto a tratar es sintetizar el filtro utilizando las *Topologías Bicuadráticas* estudiadas en la asignatura. Para ello, se toma como punto de partida la descomposición en secciones bicuadráticas provista por la plantilla, la cual se muestra a continuación. En la misma, se puede notar que la *Función de Transferencia* resulta del producto de dos *Filtros Pasa Banda*.

```
T(S)= 5252.26 S                    3126.49 S
      S^2 + 3184.42 S + 66320656 S^2 + 1895.58 S + 23500096
```

Figura 4: Función de Transferencia Descompuesta

Por otro lado, se sabe que dentro de las *Topologías Bicuadráticas* existen las de *Realimentación Positiva* y *Realimentación Negativa*. En este caso, se optó por aplicar la *Realimentación Positiva*; seguidamente, se muestra el esquema circuital correspondiente a esta topología que permite sintetizar un *Filtro Pasa Banda* y es el modelo a seguir para la realización de ambos términos bicuadráticos de la expresión anterior.

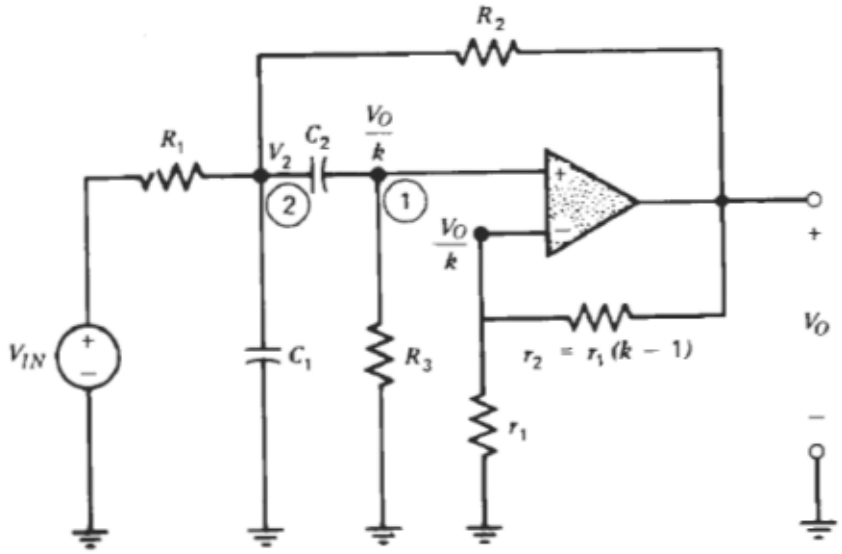

Figura 5: Topología Bicuadrática de Realimentación Positiva

A través del paquete de *Matemática Simbólica* de *Matlab*, se procede a hallar la *Función de Transferencia* de este sistema. A continuación, se muestra el código realizado, en el cual se plantea el sistema de ecuaciones siguiendo el *Método de los Nodos*, teniendo como incógnitas las tensiones en los nodos especificados como *'1'* y *'2'*, de acuerdo con el circuito. Sin embargo, la tensión incógnita de interés es la '1' debido a que se trata de la *Tensión de la Entrada No Inversora* y es sobre éste parámetro en el que se definen las funciones de transferencia denominadas como *Feedforward (TFF)* y *Feedback (TFB)*; ambas son necesarias, junto con la relación de las resistencias que conforman la realimentación negativa *(k)*, para poder hallar la *Función de Transferencia*.

$$T_v = \frac{k \cdot T_{FF}}{1 - k \cdot T_{FB}} \tag{1}$$

$$T_{FF} = \left.\frac{V_o}{V_{in}}\right|_{V_{o'}=0} \tag{2}$$

$$T_{FB} = \left.\frac{V_o}{V_{o'}}\right|_{V_{in}=0} \tag{3}$$

$$k = 1 + \frac{r_2}{r_1} \tag{4}$$

```
clc; clear; close all;

%%
% ================================================================
%     TOPOLOGIA BICUADRATICA DE REALIMENTACION POSITIVA
% ================================================================
% Definir Variables, siendo V1: Entrada No Inversora de AO    (V-)
%                           V2: Tension de Entrada            (Vin)
%                           V3: Tension de Realimentacion     (Vout')
syms V1 V2 V3 Vx C1 C2 R1 R2 R3 k s

% Definir Sistema de Ecuaciones:
%
    ------------------------------------------------------------------------

% eq1 = (Vx - V2) / R1 + Vx * s * C1 + (Vx - V3) / R2 + (Vx - V1) * s *
    C2
% eq2 = V1 / R3 + (V1 - Vx) * s * C2
%
    ------------------------------------------------------------------------

% Hallar TFB (V2 = 0):
eq1 = Vx / R1 + Vx * s * C1 + (Vx - V3) / R2 + (Vx - V1) * s * C2;
eq2 = V1 / R3 + (V1 - Vx) * s * C2;

% Solucion:
sol = solve(eq1, eq2, V1, Vx);
TFB = simplify(sol.V1 / V3);

% Hallar TFF (V3 = 0)
eq1 = (Vx - V2) / R1 + Vx * s * C1 + Vx / R2 + (Vx - V1) * s * C2;
eq2 = V1 / R3 + (V1 - Vx) * s * C2;

% Solucion:
sol = solve(eq1, eq2, V1, Vx);
TFF = simplify(sol.V1 / V2);

% Hallar Tv (Funcion de Transferencia):
Tv = simplify(k * TFF / (1 - k * TFB));
```

```
                                    C2 R2 R3 k s
-------------------------------------------------------------------------------------------
                                                                                        2
R1 + R2 + C1 R1 R2 s + C2 R1 R2 s + C2 R1 R3 s + C2 R2 R3 s - C2 R1 R3 k s + C1 C2 R1 R2 R3 s
```

Figura 6: Función de Transferencia de la Bicuadrática

3.3. Selección de Componentes

En este apartado, se busca dimensionar adecuadamente los componentes a utilizar. Se debe recordar que se está tratando la *Función del Filtro* como el producto de dos bicuadráticas; por lo tanto, se cuenta con dos bloques fundamentales interconectados entre sí, cada uno asociado con una sección. Por otro lado, se sabe que las bicuadráticas presentan la siguiente estructura:

$$T(s) = K \cdot \frac{s^2 + \frac{w_z}{Q_z} \cdot s + {w_z}^2}{s^2 + \frac{w_p}{Q_p} \cdot s + {w_p}^2} \tag{5}$$

En este caso particular, dado que se trata de *Filtros Pasa Banda* se tiene la siguiente expresión simplificada, donde se define w_p como la *Frecuencia del Polo*, Q_p como el *Factor de Calidad del Polo* y finalmente, la relación w_p/Q_p como el *Ancho de Banda*:

$$T(s) = \frac{K \cdot s}{s^2 + \frac{w_p}{Q_p} \cdot s + {w_p}^2} \tag{6}$$

La estrategia a seguir para la selección de los componentes se basa en la *Técnica de Igualación de Coeficientes*, la cual, como su nombre lo indica, consiste en igualar los términos de la estructura general con los correspondientes a la *Función del Filtro*. De esta manera, se construye un sistema de ecuaciones en el cual intervienen todos los componentes que conforman dicho filtro. Sin embargo, se presenta el inconveniente de que hay más incógnitas que ecuaciones, lo que hace que existan infinitas soluciones. Es por este motivo que se toma el criterio de considerar que todas las resistencias y capacitores tienen el mismo valor.

Esta solución viene acompañada de otro concepto importante conocido como la *Técnica de Escaleo de Impedancias*, la cual permite simplificar aún más el diseño mediante la normalización de los componentes y, posteriormente, mediante un *Factor de Escala* a elección, obtener los valores de los mismos en verdadera magnitud, es decir, llevar los resultados a valores prácticos. A continuación, se muestran las expresiones finales que resultan de tener en cuenta las técnicas mencionadas.

$$C_1 = C_2 = 1 \tag{7}$$

$$R_1 = R_2 = R_3 = R = \frac{\sqrt{2}}{w_p} \tag{8}$$

$$k = 1 + \frac{r_2}{r_1} = 4 - \frac{\sqrt{2}}{Q_p} \tag{9}$$

A partir de estas expresiones, se procede con la *Síntesis de las Bicuadráticas*, siendo necesario primeramente identificar la *Frecuencia del Polo (w_p)* y el *Factor de Calidad del Polo (Q_p)* en sus correspondientes *Funciones de Transferencia*. Con respecto al *Factor de Escala*, se optó por el valor de 10^7 para ambas etapas. Dicho ésto, seguidamente se muestra el código elaborado para este punto.

3.3.1. Síntesis del Bloque N°1

```
clc; clear; close all;

%%
%==========================
%     SINTESIS BLOQUE N1
%==========================

%================================
%     Funcion de Transferencia:
%              5252.26s
%    ------------------------------
%     s^2 + 3184.42s + 66320656
%================================

% De la FT se observa que:
wp1    = sqrt(66320656);
Qp1    = wp1 / 3184.42;

% Escalero de Impedancias:
C_b1      = 1;
R_b1      = sqrt(2) / wp1;
r2_r1_b1 = 3 - sqrt(2) / Qp1;

% Considerando 10^7:
C_b1 = C_b1 / 1E7;
R_b1 = R_b1 * 1E7;

% Por otro lado:
r1_b1 = 1E3;
r2_b1 = r2_r1_b1 * r1_b1;

% Resultados:
fprintf('=========================================\n');
fprintf('      Componentes Bloque N1       \n');
fprintf('=========================================\n');

fprintf('Capacitores:  C1 = C2 = %.0f [nF]\n', C_b1 * 1E9);
fprintf('Resistencias: R1 = R2 = R3 = %.2f [kOhm]\n', R_b1 / 1E3);
fprintf('Resistencias Realimentacion: r1 = %.0f [kOhm]; r2 = %.3f [kOhm]\
    n', r1_b1 / 1E3, r2_b1 / 1E3);
```

```
=================================
     Componentes Bloque N°1
=================================
Capacitores:  C1 = C2 = 100 [nF]
Resistencias: R1 = R2 = R3 = 1.74 [kΩ]
Resistencias Realimentación: r1 = 1 [kΩ]; r2 = 2.447 [kΩ]
```

Figura 7: Componentes Bloque N°1

Como se puede observar, las resistencias necesarias para implementar esta bicuadrática poseen valores no comerciales, excepto la de 1 [kΩ]; sin embargo, es posible aproximarse a los mismos mediante los valores que sí lo son. En ese sentido, la selección de los componentes para este bloque queda de la siguiente manera.

Componente	**Valor**
R	$1{,}8[k\Omega]$
C	$100[nF]$
r_1	$1[k\Omega]$
r_2	$2{,}2[k\Omega] + 220[\Omega]$

Cuadro 2: Tabla Componentes Bloque N°1

Con esto resuelto, se realiza la simulación del circuito obtenido utilizando el entorno de *LTspice*. Además, cabe aclarar que el mismo cuenta con la opción de incluir *Funciones de Transferencia* mediante el componente especificado como *"Laplace"*. Esta funcionalidad es muy interesante, ya que se puede cargar en dicho componente la *Función del Filtro* a sintetizar y tener su *Respuesta en Frecuencia* como referencia. Por lo tanto, esto permite comparar la *Salida del Circuito* obtenido con la *Original*.

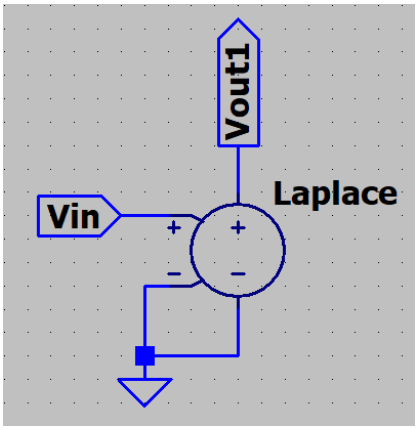

Figura 8: Componente "Laplace"

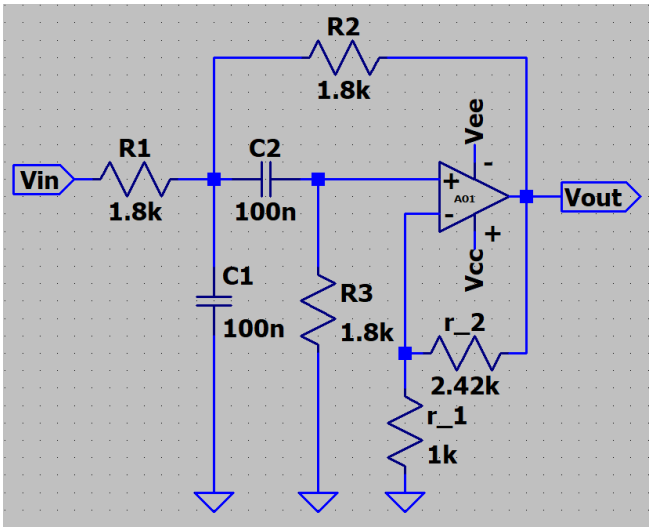

Figura 9: Circuito Bloque N°1

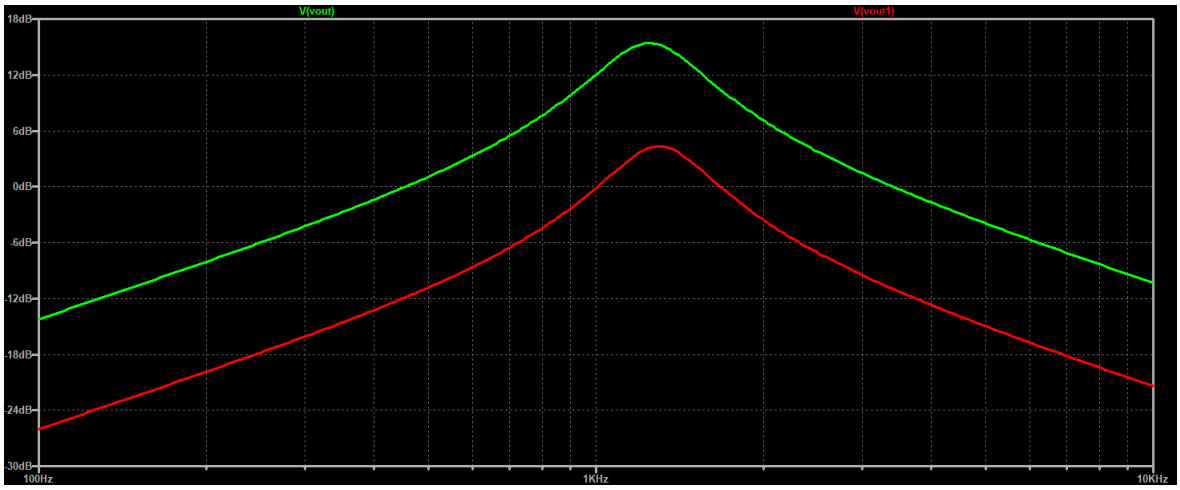

Figura 10: Diagrama de Bode Bloque N°1

3.3.2. Síntesis del Bloque N°2

```
clc; clear; close all;

%%
% ================================
%     SINTESIS BLOQUE N2
% ================================

% ================================================
%     Funcion de Transferencia:
%              3126.49s
%     ---------------------------------
%     s^2 + 1895.58s + 23500096
% ================================================

% De la FT se observa que:
wp2    = sqrt(23500096);
Qp2    = wp2 / 1895.58;

% Escalero de Impedancias:
C_b2      = 1;
R_b2      = sqrt(2) / wp2;
r2_r1_b2 = 3 - sqrt(2) / Qp2;

% Considerando 10^7:
C_b2 = C_b2 / 1E7;
R_b2 = R_b2 * 1E7;

% Por otro lado:
r1_b2 = 1E3;
r2_b2 = r2_r1_b2 * r1_b2;

% Resultados:
fprintf('=======================================\n');
fprintf('      Componentes Bloque N2          \n');
fprintf('=======================================\n');

fprintf('Capacitores:  C1 = C2 = %.0f [nF]\n', C_b2 * 1E9);
```

```
fprintf('Resistencias: R1 = R2 = R3 = %.2f [kOhm]\n', R_b2 / 1E3);
fprintf('Resistencias Realimentacion: r1 = %.0f [kOhm]; r2 = %.3f [kOhm]\
    n', r1_b2 / 1E3, r2_b2 / 1E3);
```

```
================================
     Componentes Bloque N°2
================================
Capacitores:  C1 = C2 = 100 [nF]
Resistencias: R1 = R2 = R3 = 2.92 [kΩ]
Resistencias Realimentación: r1 = 1 [kΩ]; r2 = 2.447 [kΩ]
```

Figura 11: Componentes Bloque N°2

Siguiendo la misma idea del Bloque N°1, es decir, aproximarse a las resistencias calculadas mediante valores comerciales, la selección de los componentes queda de la siguiente manera.

Componente	Valor
R	$2{,}7[k\Omega] + 220[\Omega]$
C	$100[nF]$
r_1	$1[k\Omega]$
r_2	$2{,}2[k\Omega] + 220[\Omega]$

Cuadro 3: Tabla Componentes Bloque N°2

Con estos valores, se realiza la simulación correspondiente y, nuevamente, utilizando la funcionalidad "Laplace", se compara la *Respuesta en Frecuencia* del Circuito con la de la *Función de Transferencia Original*.

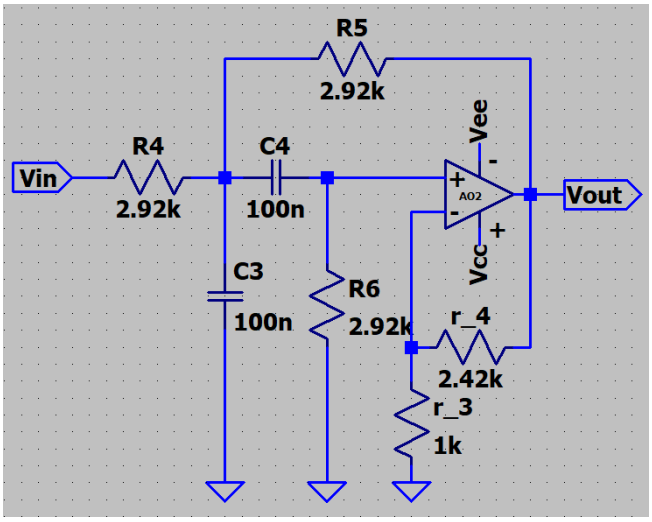

Figura 12: Circuito Bloque N°2

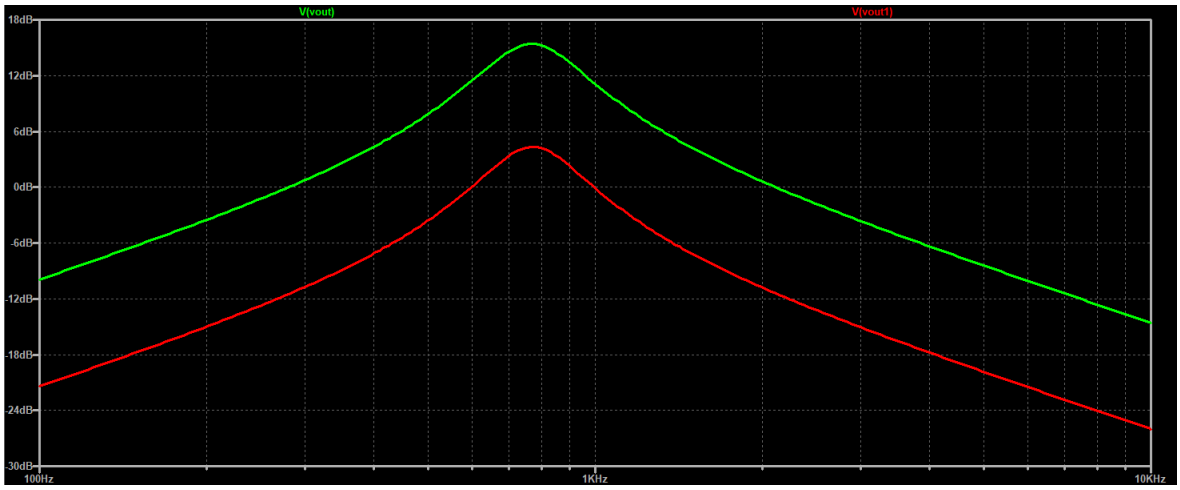

Figura 13: Diagrama de Bode Bloque N°2

3.3.3. Filtro Final

Una vez implementadas las bicuadráticas, lo que sigue es acoplarlas en cascada para llegar a la *Función Final del Filtro Pasa Banda*. A continuación, se muestra el esquema circuital logrado y se compara la *Salida* del mismo con la referencia.

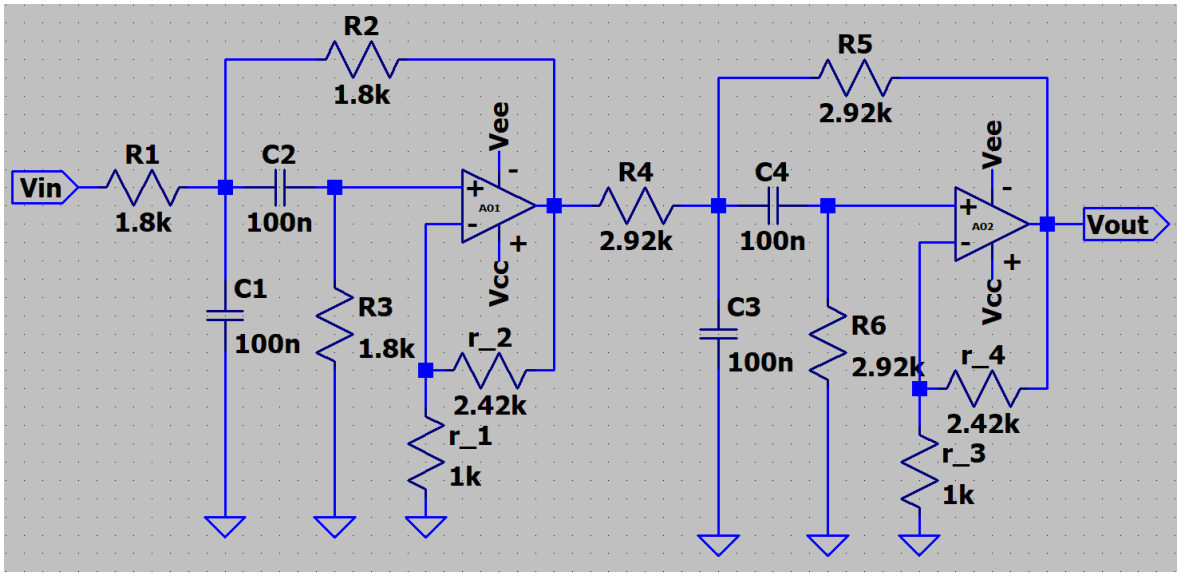

Figura 14: Filtro Final

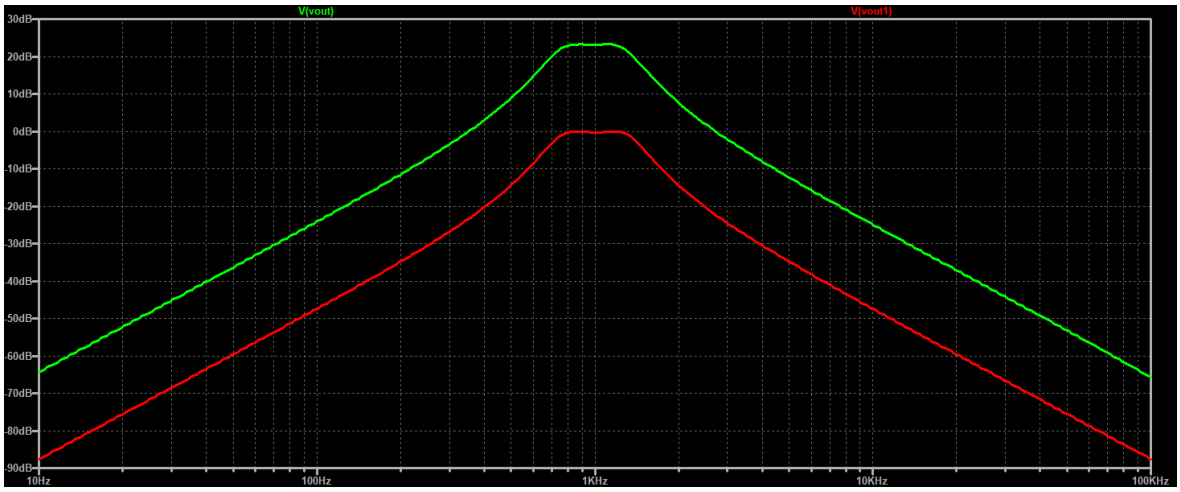

Figura 15: Diagrama de Bode Filtro Final

3.3.4. Ajuste de Ganancia

Como se puede observar en la gráfica anterior, existe una discrepancia notable entre ambas curvas; concretamente, la *Función de Transferencia del Circuito* se encuentra desplazada hacia arriba aproximadamente *23 dB*. Por este motivo, se consideró conveniente incluir una *Etapa Adicional* que permita realizar un *Ajuste de Ganancia* en la *Salida del Circuito*. Para ello, se prevé una etapa conformada por un *Amplificador Operacional* en *Configuración Inversora*. En las siguientes líneas de código se detalla el procedimiento numérico para dimensionar los componentes necesarios.

```
clc; clear; close all;

%%
% ============================
%     AJUSTE DE GANANCIA
% ============================
% Se tiene que:
Ganancia_dB     = 23;                       % Diferencia en dB
Ganancia_veces = 10^(Ganancia_dB / 20);     % Diferencia en veces
Atenuacion      = 1 / Ganancia_veces;       % Atenuacion buscada

% Entonces, si R1 = 100 [kOhm]:
R1 = 100E3
R2 = Atenuacion * R1

% Resultados:
fprintf('=========================================\n');
fprintf('      Componentes Bloque N3       \n');
fprintf('=========================================\n');

fprintf('Resistencias: R1 = %.0f [kOhm]; R2 = %.0f [kOhm]\n', R1 / 1E3,
    R2 / 1E3);
```

```
================================
    Componentes Bloque N°3
================================
Resistencias: R1 = 100 [kOhm]; R2 = 7 [kOhm]
```

Figura 16: Componentes Bloque N°3

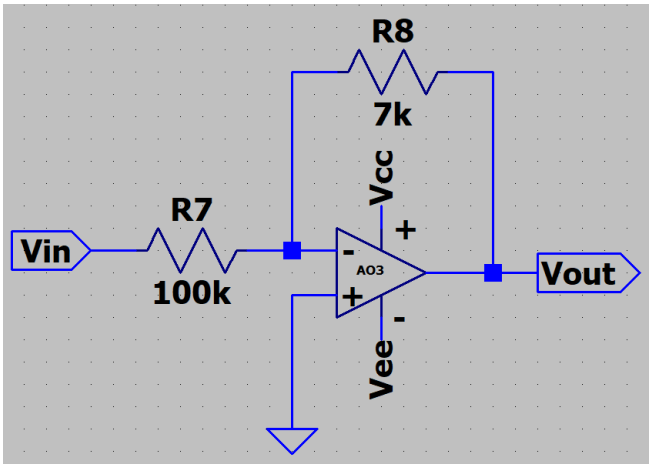

Figura 17: Circuito Bloque N°3

3.3.5. Filtro Final Corregido

A continuación, se muestra el esquema circuital final, que incluye la *Etapa Adicional*, así como la *Respuesta en Frecuencia* del mismo, observándose una mayor aproximación a la *Función del Filtro Original* y, por lo tanto, un mejor resultado en el filtro obtenido.

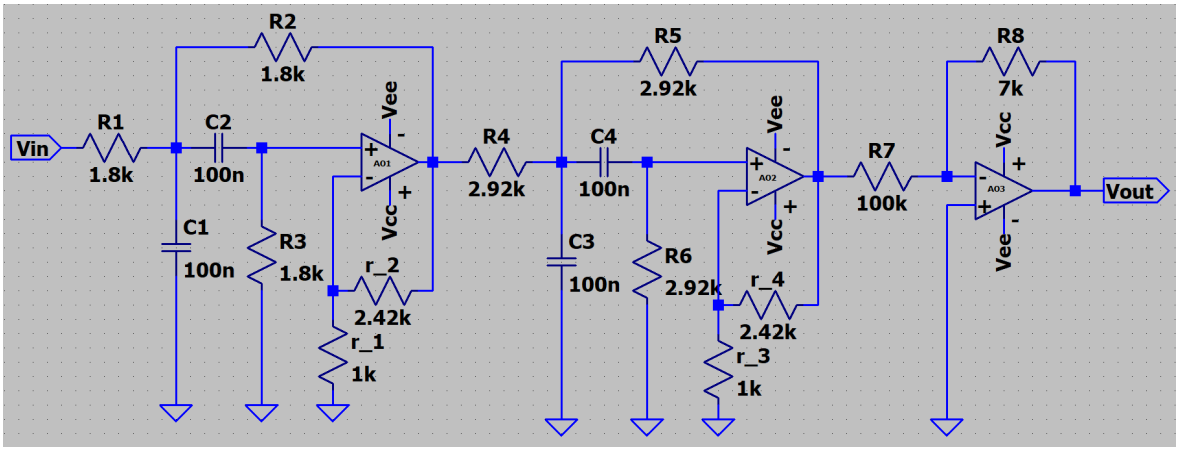

Figura 18: Filtro Final Corregido

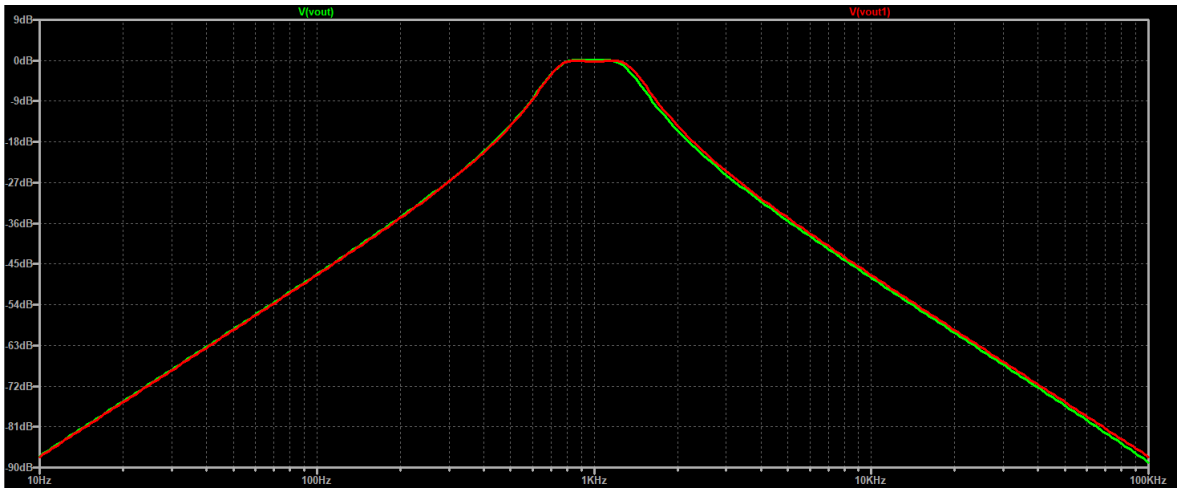

Figura 19: Diagrama de Bode Filtro Final Corregido

3.4. Sensibilidad

Otro punto importante a tratar tiene que ver con la *Sensibilidad*, es decir, qué tanto se ve afectado un determinado parámetro del sistema ante posibles variaciones en el valor nominal de los componentes utilizados. La necesidad de realizar este análisis se debe a que los diferentes componentes no tienen exactamente el mismo valor que indican, sino que cuentan con un cierto grado de incertidumbre, es decir, una tolerancia.

La *Sensibilidad* se calcula de acuerdo con la siguiente expresión, donde se define como *'p'* el *Parámetro del Sistema* y como *'x'* el *Elemento* que influye en dicho parámetro.

$$S_x^p = \frac{x}{p} \cdot \frac{\partial p}{\partial x} \tag{10}$$

A partir de la estructura general de las bicuadráticas que se presentó anteriormente, se observa que los parámetros del sistema en cuestión son dos: la *Frecuencia del Polo* (w_p) y el *Ancho de Banda* (w_p/Q_p). De la *Función de la Bicuadrática* se extraen las siguientes expresiones de los parámetros mencionados.

$$w_p = \sqrt{\frac{R_1 + R_2}{C_1 \cdot C_2 \cdot R_1 \cdot R_2 \cdot R_3}} \tag{11}$$

$$\frac{w_p}{Q_p} = bw_p = \frac{1}{C_1 \cdot R_1} + \frac{1}{C_2 \cdot R_3} + \frac{1}{C_1 \cdot R_3} + \frac{1-k}{C_1 \cdot R_2} \tag{12}$$

Utilizando estas expresiones, se procede a calcular la sensibilidad de los mismos en relación con cada uno de los componentes a través del siguiente código realizado en *Matlab*. Por otro lado, es necesario recordar que las dos bicuadráticas desarrolladas son de *Realimentación Positiva*; por lo tanto, los cálculos siguientes aplican para ambas etapas.

```
clc; clear; close all;

%%
% ==================
%     SENSIBILIDAD
% ==================
% De la FT se tiene que:
wp  = simplify(sqrt((R1 + R2) / (C1 * C2 * R1 * R2 * R3)));
bwp = simplify((C1 * R1 * R2 + C2 * R1 * R2 + C2 * R1 * R3 + C2 * R2 * R3
    - C2 * R1 * R3 * k) / (C1 * C2 * R1 * R2 * R3));

% Sensibilidad de cada Componente:
S_wp_R1 = simplify(diff(wp, R1) * R1 / wp);
S_wp_R2 = simplify(diff(wp, R2) * R2 / wp);
S_wp_R3 = simplify(diff(wp, R3) * R3 / wp);
S_wp_C1 = simplify(diff(wp, C1) * C1 / wp);
S_wp_C2 = simplify(diff(wp, C1) * C1 / wp);

S_bwp_R1 = simplify(diff(bwp, R1) * R1 / bwp);
S_bwp_R2 = simplify(diff(bwp, R2) * R2 / bwp);
S_bwp_R3 = simplify(diff(bwp, R3) * R3 / bwp);
S_bwp_C1 = simplify(diff(bwp, C1) * C1 / bwp);
S_bwp_C2 = simplify(diff(bwp, C2) * C2 / bwp);

% Recordar que: R1 = R2 = R3 y C1 = C2:
syms R C

% Resultados:
fprintf('=================================\n');
fprintf('         Sensibilidad          \n');
fprintf('=================================\n');

fprintf('wp:\n');
fprintf('S_wp_R1 = %.2f\n', double(subs(S_wp_R1, [R1, R2, R3, C1, C2], [R
    , R, R, C, C])));
fprintf('S_wp_R2 = %.2f\n', double(subs(S_wp_R2, [R1, R2, R3, C1, C2], [R
    , R, R, C, C])));
fprintf('S_wp_R3 = %.1f\n', double(subs(S_wp_R3, [R1, R2, R3, C1, C2], [R
    , R, R, C, C])));
fprintf('S_wp_C1 = %.1f\n', double(subs(S_wp_C1, [R1, R2, R3, C1, C2], [R
    , R, R, C, C])));
fprintf('S_wp_C2 = %.1f\n', double(subs(S_wp_C2, [R1, R2, R3, C1, C2], [R
    , R, R, C, C])));
fprintf('---------------------------------\n');
fprintf('bwp:\n');
fprintf('S_bwp_R1 = %.2f\n', double(subs(S_bwp_R1, [R1, R2, R3, C1, C2, k
    ], [R, R, R, C, C, r2_r1_b1 + 1])));
fprintf('S_bwp_R2 = %.2f\n', double(subs(S_bwp_R2, [R1, R2, R3, C1, C2, k
    ], [R, R, R, C, C, r2_r1_b1 + 1])));
fprintf('S_bwp_R3 = %.2f\n', double(subs(S_bwp_R3, [R1, R2, R3, C1, C2, k
    ], [R, R, R, C, C, r2_r1_b1 + 1])));
fprintf('S_bwp_C1 = %.2f\n', double(subs(S_bwp_C1, [R1, R2, R3, C1, C2, k
    ], [R, R, R, C, C, r2_r1_b1 + 1])));
fprintf('S_bwp_C2 = %.2f\n', double(subs(S_bwp_C2, [R1, R2, R3, C1, C2, k
    ], [R, R, R, C, C, r2_r1_b1 + 1])));
```

```
============================
        Sensibilidad
============================
wp:
       S_wp_R1 = -0.25
       S_wp_R2 = -0.25
       S_wp_R3 = -0.5
       S_wp_C1 = -0.5
       S_wp_C2 = -0.5
----------------------------
bwp:
      S_bwp_R1 = -1.81
      S_bwp_R2 = 4.43
      S_bwp_R3 = -3.62
      S_bwp_C1 = 0.81
      S_bwp_C2 = -1.81
```

Figura 20: Sensibilidad

De acuerdo con estos resultados, se puede interpretar lo siguiente:

- El incremento de un 1 % de *R1* produce un decremento del 0.25 % de w_p y un decremento del 1.81 % de w_p/Q_p.
- El incremento de un 1 % de *R2* produce un decremento del 0.25 % de w_p y un incremento del 4.43 % de w_p/Q_p.
- El incremento de un 1 % de *R3* produce un decremento del 0.5 % de w_p y un decremento del 3.62 % de w_p/Q_p.
- El incremento de un 1 % de *C1* produce un decremento del 0.5 % de w_p y un incremento del 0.81 % de w_p/Q_p.
- El incremento de un 1 % de *C2* produce un decremento del 0.5 % de w_p y un decremento del 1.81 % de w_p/Q_p.

Esta información permite definir la tolerancia mínima que deben cumplir los diferentes componentes a utilizar para que la *Respuesta del Filtro* se ajuste lo suficiente a las especificaciones indicadas en la plantilla.

Asimismo, en la consigna del presente laboratorio se solicita evaluar la peor desviación posible de los *Parámetros del Sistema*, suponiendo que todos los elementos tienen una tolerancia del 10 %. Esto se resuelve mediante la sumatoria del producto de dos términos: la sensibilidad correspondiente al parámetro elegido y la tolerancia del componente asociada a dicha sensibilidad, ésta última expresada en decimal.

$$\frac{\Delta w_p}{w_p} = \sum_{i=0}^{i} T_{x_i} \cdot S_{x_i}^{w_p} \tag{13}$$

$$\frac{\Delta bw_p}{bw_p} = \sum_{i=0}^{i} T_{x_i} \cdot S_{x_i}^{bw_p} \tag{14}$$

- Para el parámetro w_p, tanto en la primera como en la segunda etapa, se advierte un desvío total del 20 %
- Para el parámetro w_p/Q_p o bw_p, tanto en la primera como en la segunda etapa, se advierte un desvío total del 20 %.

3.5. Simulación de Montecarlo

El *Análisis de Montecarlo* es una herramienta útil que permite evaluar cómo influye la variación de los componentes en el rendimiento de un sistema. Este método consiste en realizar múltiples simulaciones y, en cada una de ellas, modificar los valores de los componentes dentro de un rango definido por una tolerancia. De esta manera, se puede determinar qué tan sólida es la respuesta del sistema ante dichas variaciones.

Esta simulación fue realizada en *LTspice*, para la cual se definieron como valores centrales de los componentes los valores calculados en los desarrollos anteriores. Asimismo, para el rango de variación de los mismos se consideró una tolerancia del 10 %; finalmente, se optó por realizar un total de 50 simulaciones. A continuación, se muestra el circuito preparado para el *Análisis de Montecarlo*, así como las *Respuestas del Filtro* obtenidas de este análisis.

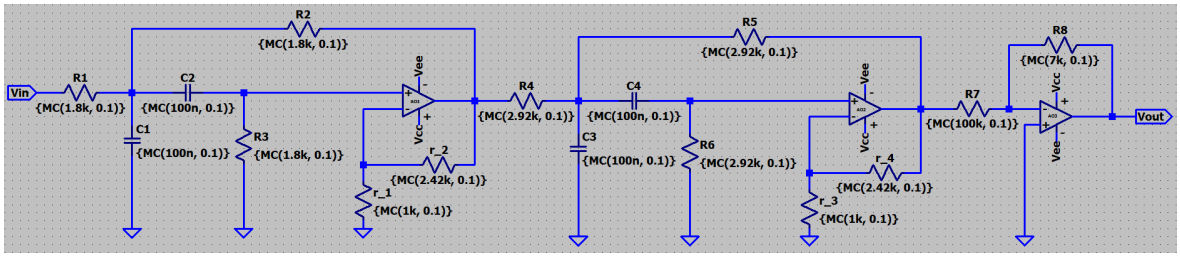

Figura 21: Filtro Final Montecarlo

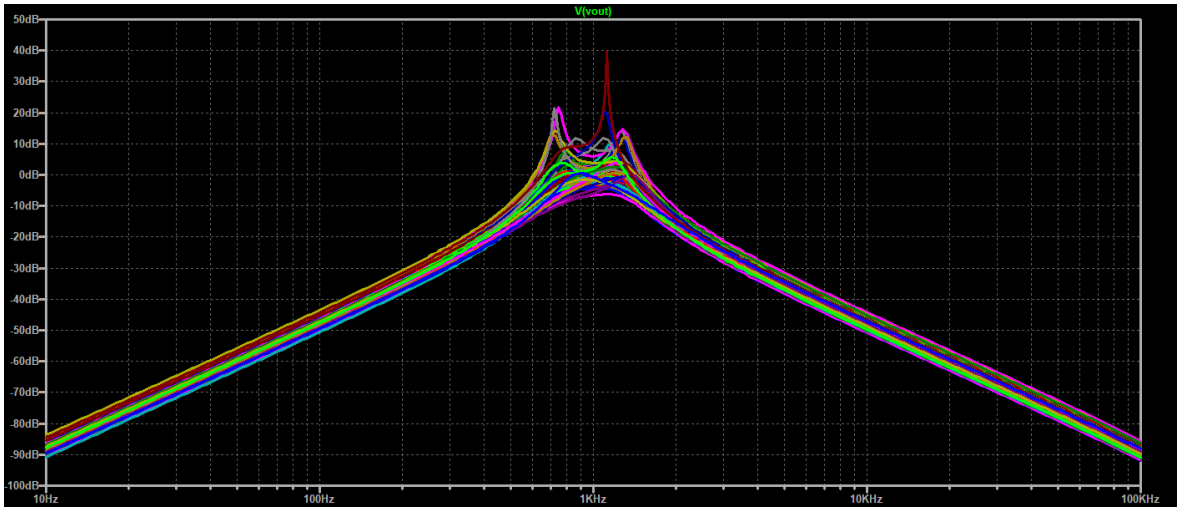

Figura 22: Simulación de Montecarlo

De la gráfica anterior se concluye que una tolerancia del 10 % en los componentes no genera la confiabilidad suficiente en el filtro en cuanto a los requerimientos. En ese sentido, se realiza nuevamente la simulación, teniendo en cuenta ahora una tolerancia del 1 % y se obtienen los siguientes resultados, los cuales presentan una mejor respuesta.

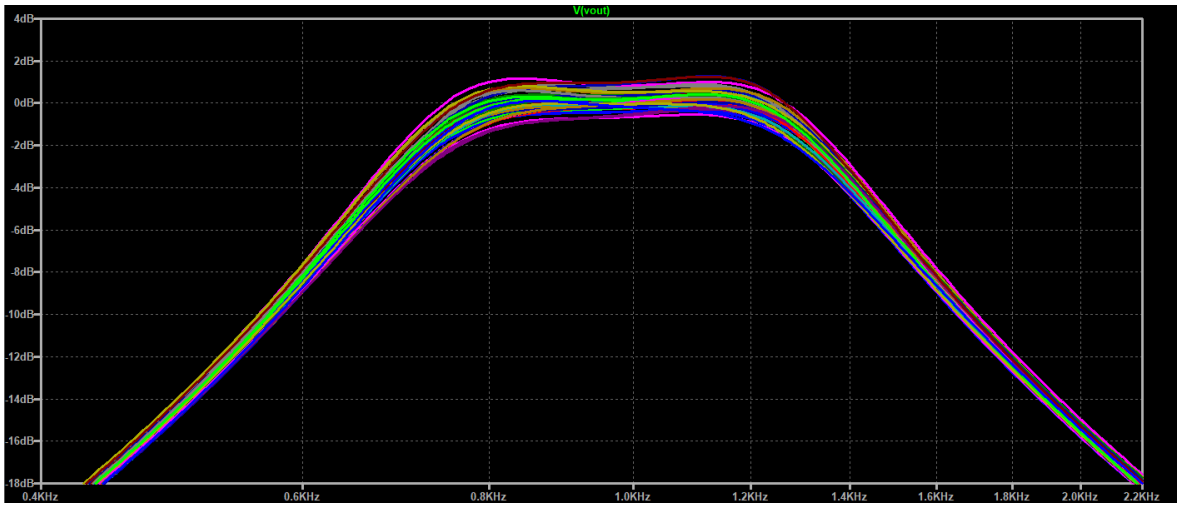

Figura 23: Simulación de Montecarlo

4. Conclusión

En el presente laboratorio, se aplicaron conceptos relacionados con la *Síntesis de Filtros Activos*. Durante su desarrollo, se profundizó en las *Topologías Bicuadráticas* basadas en el *Método de Sallen-Key*, lo que constituye un paso importante para implementar la *Función de Transferencia del Filtro*. Para lograrlo, se pusieron en práctica las técnicas aprendidas, como la *Igualación de Coeficientes* y el *Escaleo de Impedancias*.

Además, se realizó el estudio de la *Sensibilidad* del filtro en función de los parámetros que lo conforman, permitiendo comprender cuánto afectan al rendimiento esperado del circuito. Esto, a su vez, introduce un nuevo concepto: *Simulación de Montecarlo*. A través de esto, se logra establecer la tolerancia que se debe exigir a los diferentes componentes para mantener la coherencia con las especificaciones.