

# Trabajo Práctico de Laboratorio N°4

## Filtros Activos.

**Profesor Titular:** DR. ING. PABLO FERREYRA

**Profesor Adjunto:** ING. CÉSAR REALE

**Ayudante alumno:** LUCAS HERALDO DUARTE

Facultad de Ciencias Exactas Físicas y Naturales

Universidad Nacional de Córdoba



Universidad  
Nacional  
de Córdoba

**Síntesis de Redes Activas**  
**Ingeniería Electrónica - Octubre 2024**

### OBJETIVOS

*En base a la planilla de requerimientos suministrada, sintetizar un circuito basado en amplificadores operacionales que satisfaga esos requisitos.*

## 1. METODOLOGÍA

En general, para cada uno de los casos particulares solicitados, se debe:

- a Realizar una sintética introducción teórica.
- b Analizar el circuito propuesto, su desarrollo numérico, todos los cálculos analíticos.
- c Realizar simulación en LTSPICE.
- d Armar el circuito y hacer las mediciones en laboratorio.
- e Finalmente comparar los valores calculados, simulados y medidos, y extraer conclusiones a cerca de las diferencias. Analizar las causas.
- f Presentar un informe digital y en papel.

## 2. DESARROLLO

### 1. CIRCUITO I

En base a la planilla de requerimientos de la figura 1, se pide:

- 1.1 Aproximar la función de atenuación mediante polinomios de Chebychev utilizando python o matlab (ver Anexo I y II).
- 1.2 Sintetizar un circuito que satisfaga los requerimientos del punto anterior utilizando topologías bicuadráticas de realimentación positiva o negativa, a elección.
- 1.3 Simular cada etapa y el filtro total con LTSPICE.

- 1.4 Calcular la sensibilidad de la frecuencia del polo de cada bicuadrática ( $\omega_p$ ) y del ancho de banda ( $\omega_p/Q_p$ ).
- 1.5 Analizar la peor desviación si todos los elementos tienen una tolerancia del 10%.
- 1.6 Realizar una simulación de Montecarlo de las desviaciones con LTspice.
- 1.7 Armar el circuito, medir experimentalmente las curvas de atenuación y desfase. Contrastarlas con las predicciones teóricas y las simulaciones.

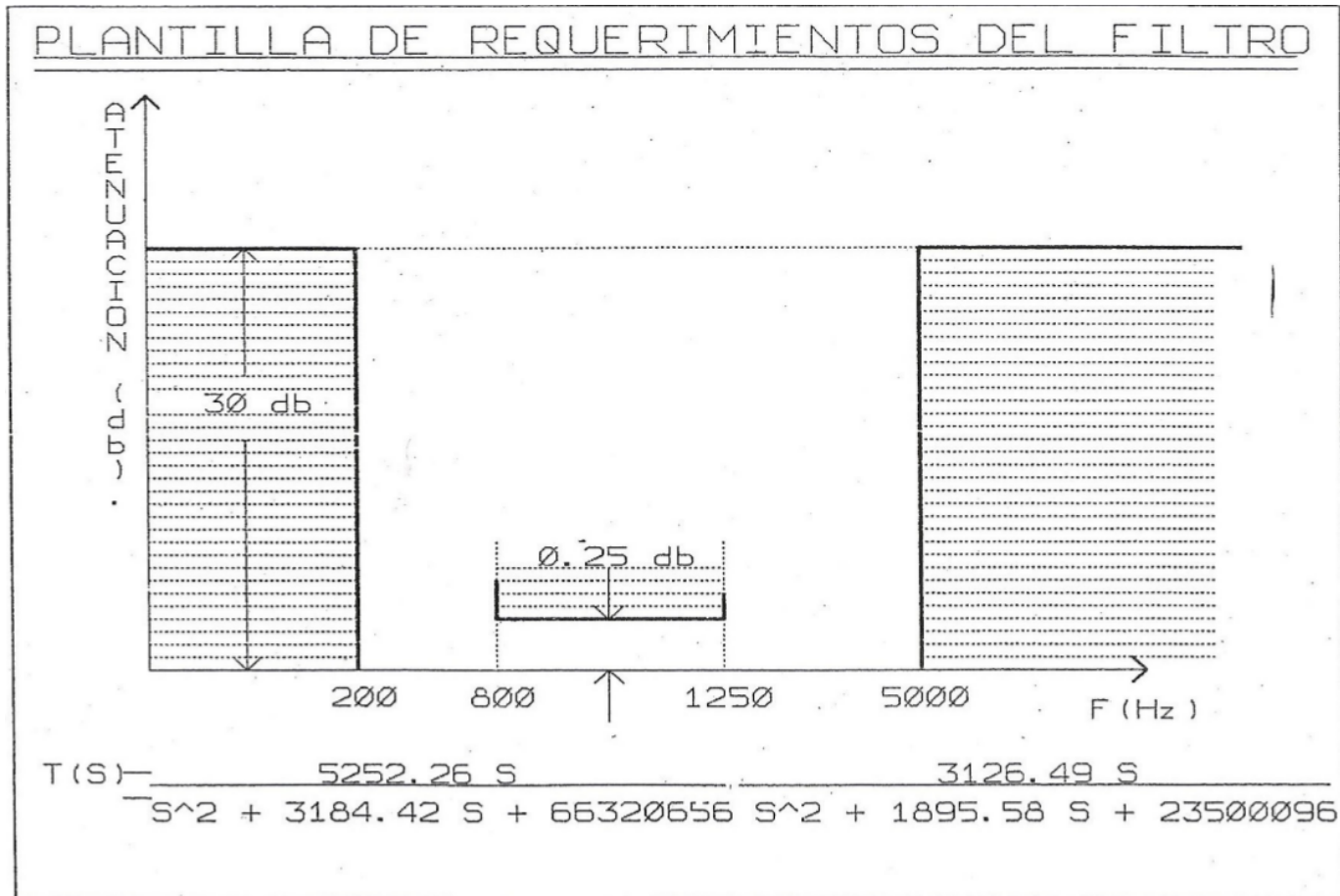


Figura 1: planilla de requerimientos, circuito I

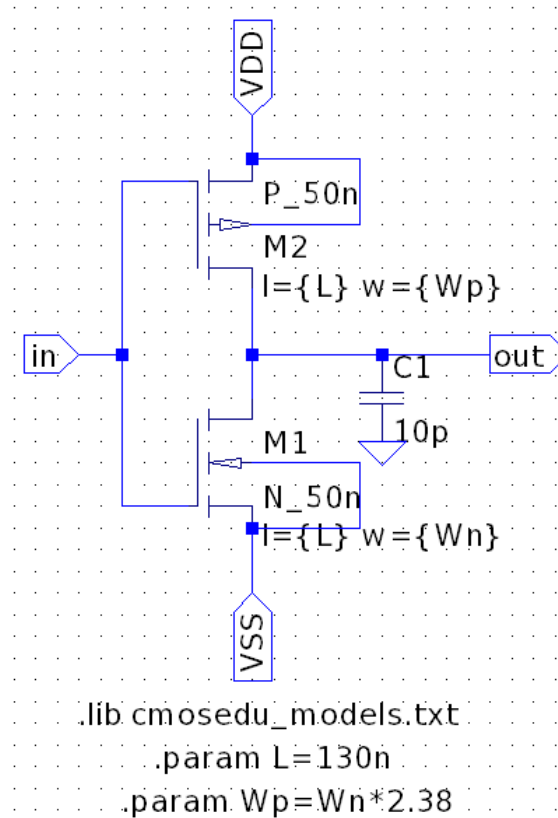
## 2. EJERCICIO ADICIONAL I: CIRCUITO II

Este ejercicio es una introducción a la microelectrónica.

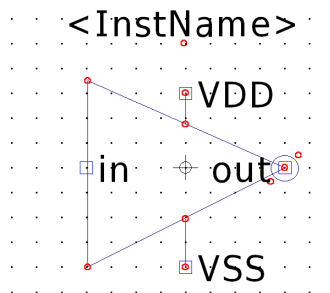
Primero diseñar un inversor (ver fig. 2 con  $t_p = 10[\text{ns}]$  criterio del 50 % y  $V_M = 0,5V_{DD}$  y con la salida se carga con un capacitor de  $10[\text{pf}]$ , crear un símbolo del inversor (fig. 3).

Luego conectar 13 inversores en serie realice un oscilador de anillo y determine, frecuencia y  $t_d$ . (Ver fig. 4 colocar al lazo de retroalimentación la condición de inicial de  $V=1.8V$ ).

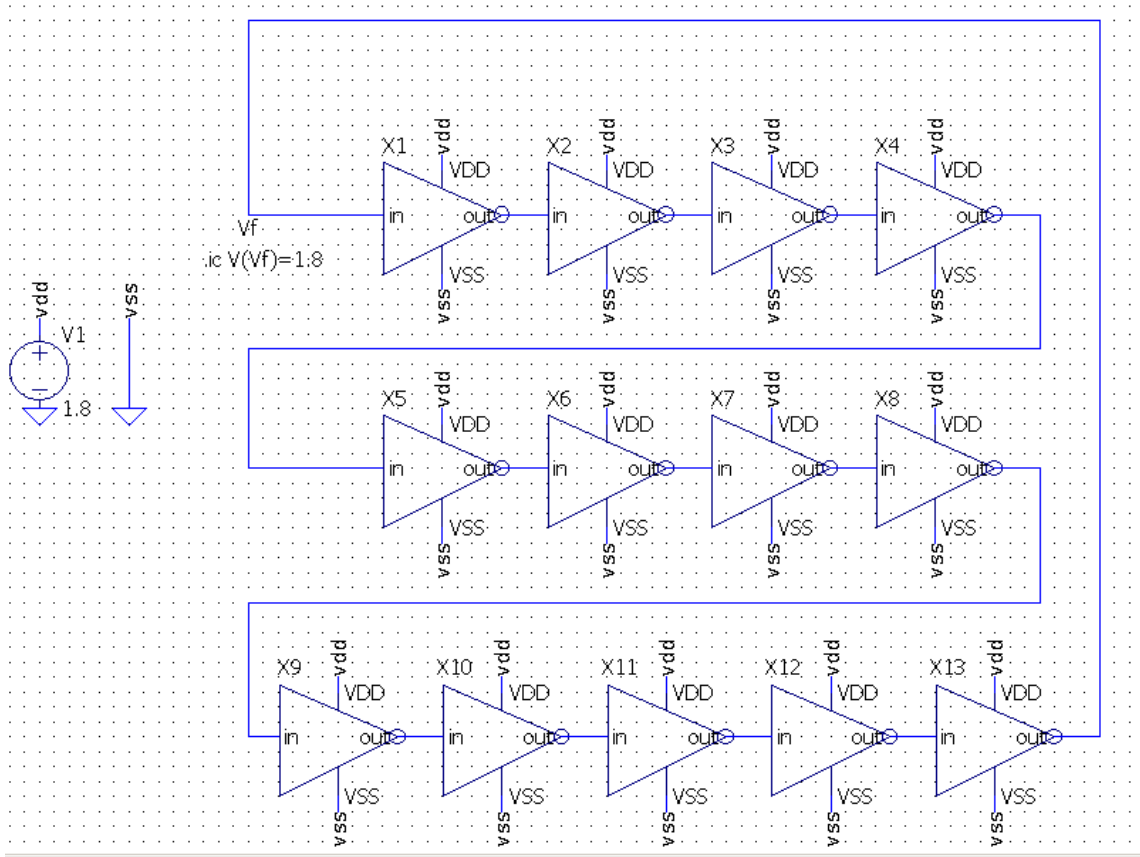
Encontrar todas las variables de forma perimétrica y utilizando el comando measure. Hacer simulaciones .dc, .tran y transformada de fourier. Utilizar modelo de cmosedu canal corto (www.cmosedu.com).



**Figura 2:** *Inversor CMOS*



**Figura 3:** *Símbolo de inversor*



**Figura 4:** Oscilador de anillo

### 3. EJERCICIO ADICIONAL II: CIRCUITO III

Dise ar un filtro NOTCH para ser utilizado en electrocardiograma (ECG), donde es fundamental filtrar la se al de 50Hz de la l nea de tensi n.

Especificaciones:  $F_n=50\text{Hz}$ ,  $Q=10$ , ancho de banda de ca da 3db de 5Hz y sin ganancia.

Ecuaci n del NOTCH:

$$H(\omega) = \frac{An(s^2 + \omega_n^2)}{s^2 + s\frac{\omega_n}{Q} + \omega_n^2} \quad (1)$$

Sintetizar filtro utilizando topolog a bicuadr tica general de 3 o 4 amplificadores operacionales.

### 3. ANEXO I

#### Codigo de PYTHON

```
from scipy import signal
from scipy.signal import freqs
import numpy as np #Importa libreria numerica
import sympy as sym #simbolica
import matplotlib.pyplot as plt #importa matplotlib solo pyplot

sym.init_printing() #activa a jupyter para mostrar simbolicamente el output
%matplotlib widget

# Parametros de Entrada
FS=5000*10;
fp=[800, 1250]; #Banda de Paso [Hz]
fs=[200, 5000]; #Banda de Rechazo [Hz]
Wp=np.dot(2*np.pi,fp); #Banda de Paso [rad/s]
Ws=np.dot(2*np.pi,fs); #Banda de Rechazo [rad/s]
Ap=0.25; #Atenuacion maxima en Banda de Paso [dB]
As=30; #Atenuacion minima en Banda de Rechazo [dB]

N, Wn = signal.cheb1ord(Wp, Ws, Ap, As, analog=True)
b, a = signal.cheby1(N, Ap, Wn, btype='bandpass', analog=True)
w, h = signal.freqs(b, a)

Filtro=signal.TransferFunction(b,a) #Funcion de transferencia calculada
#Implementacion como PasaAlto/PasaBajo
sos = signal.cheby1(N, Ap, Wn, btype='bandpass', output='sos', analog=True)
PasaBajo=signal.TransferFunction(2*sos[0,:3],sos[0,3:])
PasaAlto=signal.TransferFunction(1/2*sos[1,:3],sos[1,3:])

plt.figure()
plt.semilogx(w, 20 * np.log10(abs(h)))
plt.title('Chebyshev I ')
plt.xlabel('Frecuencia [rad/seg]')
plt.ylabel('Amplitud [dB]')
plt.grid(which='both', axis='both')
plt.show()

display(Filtro)
display(PasaBajo)
display(PasaAlto)
```

## 4. ANEXO II

### Codigo de MATLAB

```
%SRA - Trabajo Practico de Laboratorio N 4
clc;
clear;
close all;

%% Parametros de Entrada
fp=[800 1250]; %Banda de Paso [Hz]
fs=[200 5000]; %Banda de Rechazo [Hz]
Wp=2*pi*fp; %Banda de Paso [rad/s]
Ws=2*pi*fs; %Banda de Rechazo [rad/s]
Ap=0.25; %Atenuacion maxima en Banda de Paso [dB]
As=30; %Atenuacion minima en Banda de Rechazo [dB]

%% Calculo de FT
[n,Wp]=cheb1ord(Wp,Ws,Ap,As,'s');
[num,den]=cheby1(n,Ap,Wp,'s');
Filtro=tf(num,den) %Funcion de transferencia calculada
[sos,g] = tf2sos(num,den); %Descomponemos en bicuadraticas
%Implementacion como PasaAlto/PasaBajo
PasaBajo=tf(2*g*sos(1,1:3),sos(1,4:6))
PasaAlto=tf(1/2*sos(2,1:3),sos(2,4:6))

%% Graficos
figure;
hold on;
%Especificaciones Filtro
plot([fs(1)/10 fs(1) fs(1)],[-As -As -Ap],'Color','r','LineWidth',3);
plot([fs(2) fs(2) fs(2)*10],[-Ap -As -As],'Color','r','LineWidth',3);
plot([fp(1) fp(1) fp(2) fp(2)],[-As -Ap -Ap -As],'Color','g','LineWidth',3);

%Filtro
h = bodeplot(Filtro);
p = getoptions(h);
p.PhaseVisible='off';
p.FreqUnits='Hz';
p.Grid='on';
setoptions(h,p);
bode(PasaBajo);
bode(PasaAlto);
```