

# Fourier Series

Omkar Raijade

## CONTENTS

1	Periodic Function	1
2	Fourier Series	1
3	Fourier Transform	3
4	Filter	4
5	Filter Design	5

**Abstract**—This manual provides a simple introduction to Fourier Series

## 1 PERIODIC FUNCTION

Let

$$x(t) = A_0 |\sin(2\pi f_0 t)| \quad (1.1)$$

1.1 Plot  $x(t)$ .

**Solution:** The following code plots  $x(t)$ :

```
wget https://github.com/omkar30122001/
Fourier-Series/blob/main/1.1.py
```

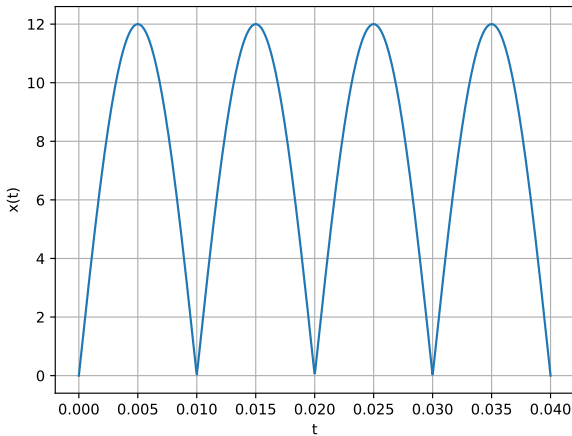


Fig. 1.1:  $x(t)$

1.2 Show that  $x(t)$  is periodic and find its period.

**Solution:** Note that,

$$x\left(t + \frac{1}{2f_0}\right) = A_0 \left| \sin\left(2\pi f_0 \left(t + \frac{1}{2f_0}\right)\right) \right| \quad (1.2)$$

$$= A_0 |\sin(2\pi f_0 t + \pi)| \quad (1.3)$$

$$= A_0 |\sin(2\pi f_0 t)| \quad (1.4)$$

Hence the period of  $x(t)$  is  $\frac{1}{2f_0}$ .

## 2 FOURIER SERIES

Consider  $A_0 = 12$  and  $f_0 = 50$  for all numerical calculations.

2.1 If

$$x(t) = \sum_{k=-\infty}^{\infty} c_k e^{j2\pi k f_0 t} \quad (2.1)$$

show that

$$c_k = f_0 \int_{-\frac{1}{2f_0}}^{\frac{1}{2f_0}} x(t) e^{-j2\pi k f_0 t} dt \quad (2.2)$$

**Solution:** For some  $n \in \mathbb{Z}$ ,

$$x(t) e^{-j2\pi n f_0 t} = \sum_{k=-\infty}^{\infty} c_k e^{j2\pi(k-n)f_0 t} \quad (2.3)$$

But

$$\int_{-\frac{1}{2f_0}}^{\frac{1}{2f_0}} e^{j2\pi k f_0 t} dt = \frac{1}{f_0} \delta_{0k} \quad (2.4)$$

where  $\delta_{ij}$  denotes the Kronecker delta. Thus,

$$\int_{-\frac{1}{2f_0}}^{\frac{1}{2f_0}} x(t) e^{-j2\pi n f_0 t} dt = \frac{c_n}{f_0} \quad (2.5)$$

$$\Rightarrow c_n = f_0 \int_{-\frac{1}{2f_0}}^{\frac{1}{2f_0}} x(t) e^{-j2\pi n f_0 t} dt \quad (2.6)$$

2.2 Find  $c_k$  for (1.1)

**Solution:** Using (2.2),

$$c_n = f_0 \int_{-\frac{1}{2f_0}}^{\frac{1}{2f_0}} A_0 |\sin(2\pi f_0 t)| e^{-j2\pi n f_0 t} dt \quad (2.7)$$

$$= f_0 \int_{-\frac{1}{2f_0}}^{\frac{1}{2f_0}} A_0 |\sin(2\pi f_0 t)| \cos(2\pi n f_0 t) dt$$

$$+ j f_0 \int_{-\frac{1}{2f_0}}^{\frac{1}{2f_0}} A_0 |\sin(2\pi f_0 t)| \sin(2\pi n f_0 t) dt \quad (2.8)$$

$$= 2f_0 \int_0^{\frac{1}{2f_0}} A_0 \sin(2\pi f_0 t) \cos(2\pi n f_0 t) dt \quad (2.9)$$

$$= f_0 A_0 \int_0^{\frac{1}{2f_0}} (\sin(2\pi(n+1)f_0 t)) dt - f_0 A_0 \int_0^{\frac{1}{2f_0}} (\sin(2\pi(n-1)f_0 t)) dt \quad (2.10)$$

$$= A_0 \frac{1 + (-1)^n}{2\pi} \left( \frac{1}{n+1} - \frac{1}{n-1} \right) \quad (2.11)$$

$$= \begin{cases} \frac{2A_0}{\pi(1-n^2)} & n \text{ even} \\ 0 & n \text{ odd} \end{cases} \quad (2.12)$$

2.3 Verify (1.1) using python.

**Solution:** The following code plots Fig. 2.3:

```
wget https://github.com/omkar30122001/
Fourier-Series/blob/main/2.3.py
```

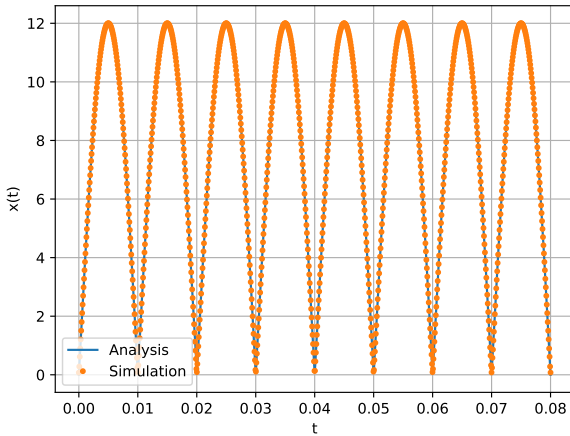


Fig. 2.3: Verification of (2.1).

2.4 Show that

$$x(t) = \sum_{k=0}^{\infty} (a_k \cos j2\pi k f_0 t + b_k \sin j2\pi k f_0 t) \quad (2.13)$$

and obtain the formulae for  $a_k$  and  $b_k$ .

**Solution:** From (2.1),

$$x(t) = \sum_{k=-\infty}^{\infty} c_k e^{j2\pi k f_0 t} \quad (2.14)$$

$$= c_0 + \sum_{k=1}^{\infty} c_k e^{j2\pi k f_0 t} + c_{-k} e^{-j2\pi k f_0 t} \quad (2.15)$$

$$= c_0 + \sum_{k=1}^{\infty} (c_k + c_{-k}) \cos(2\pi k f_0 t) + \sum_{k=0}^{\infty} j(c_k - c_{-k}) \sin(2\pi k f_0 t) \quad (2.16)$$

Hence, for  $k \geq 0$ ,

$$a_k = \begin{cases} c_0 & k = 0 \\ c_k + c_{-k} & k > 0 \end{cases} \quad (2.17)$$

$$= \begin{cases} f_0 \int_{-\frac{1}{2f_0}}^{\frac{1}{2f_0}} x(t) dt & k = 0 \\ 2f_0 \int_{-\frac{1}{2f_0}}^{\frac{1}{2f_0}} x(t) \cos(2\pi k f_0 t) dt & k > 0 \end{cases} \quad (2.18)$$

$$b_k = \frac{c_k - c_{-k}}{j} = 2f_0 \int_{-\frac{1}{2f_0}}^{\frac{1}{2f_0}} x(t) \sin(2\pi k f_0 t) dt \quad (2.19)$$

2.5 Find  $a_k$  and  $b_k$  for (1.1)

**Solution:** From (2.1), we see that since  $x(t)$  is even,

$$x(-t) = \sum_{k=-\infty}^{\infty} c_k e^{-j2\pi k f_0 t} \quad (2.20)$$

$$= \sum_{k=-\infty}^{\infty} c_{-k} e^{j2\pi k f_0 t} \quad (2.21)$$

$$= \sum_{k=-\infty}^{\infty} c_k e^{j2\pi k f_0 t} \quad (2.22)$$

where we substitute  $k := -k$  in (2.21). Hence, we see that  $c_k = c_{-k}$ . So, from (2.17) and

(2.19), for  $k \geq 0$ ,

$$a_k = \begin{cases} \frac{2A_0}{\pi} & k = 0 \\ \frac{4A_0}{\pi(1-k^2)} & k > 0, k \text{ even} \\ 0 & \text{otherwise} \end{cases} \quad (2.23)$$

$$b_k = 0 \quad (2.24)$$

2.6 Verify (2.13) using python.

**Solution:** The following code plots Fig. 2.6

```
wget https://github.com/omkar30122001/
Fourier-Series/blob/main/2.6.py
```

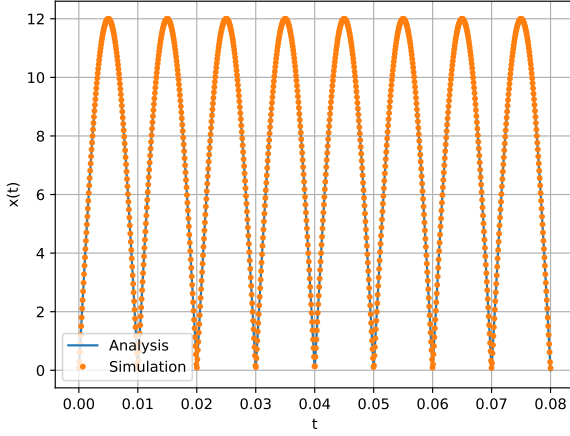


Fig. 2.6: Verification of (2.13).

### 3 FOURIER TRANSFORM

3.1

$$\delta(t) = 0, \quad t \neq 0 \quad (3.1)$$

$$\int_{-\infty}^{\infty} \delta(t) dt = 1 \quad (3.2)$$

3.2 The Fourier Transform of  $g(t)$  is

$$G(f) = \int_{-\infty}^{\infty} g(t) e^{-j2\pi ft} dt \quad (3.3)$$

3.3 Show that

$$g(t - t_0) \xleftrightarrow{\mathcal{F}} G(f) e^{-j2\pi ft_0} \quad (3.4)$$

**Solution:** We write, substituting  $u := t - t_0$ ,

$$g(t - t_0) \xleftrightarrow{\mathcal{F}} \int_{-\infty}^{\infty} g(t - t_0) e^{-j2\pi ft} dt \quad (3.6)$$

$$= \int_{-\infty}^{\infty} g(u) e^{-j2\pi f(u+t_0)} du \quad (3.7)$$

$$= G(f) e^{-j2\pi ft_0} \quad (3.8)$$

where the last equality follows from (3.3).

3.4 Show that

$$G(t) \xleftrightarrow{\mathcal{F}} g(-f) \quad (3.9)$$

**Solution:** Using the definition of the Inverse Fourier Transform,

$$g(t) = \int_{-\infty}^{\infty} G(f) e^{j2\pi ft} df \quad (3.10)$$

Hence, setting  $t := -f$  and  $f := t$ , which implies  $df = dt$ ,

$$g(-f) = \int_{-\infty}^{\infty} G(t) e^{-j2\pi ft} dt \quad (3.11)$$

$$\implies G(t) \xleftrightarrow{\mathcal{F}} g(-f) \quad (3.12)$$

3.5  $\delta(t) \xleftrightarrow{\mathcal{F}} ?$

**Solution:** We have, from the definition of  $\delta(t)$ ,

$$\delta(t) \xleftrightarrow{\mathcal{F}} \int_{-\infty}^{\infty} \delta(t) e^{-j2\pi ft} dt \quad (3.13)$$

$$= \int_{-\infty}^{\infty} \delta(0) dt \quad (3.14)$$

$$= \int_{-\infty}^{\infty} \delta(t) dt = 1 \quad (3.15)$$

3.6  $e^{-j2\pi f_0 t} \xleftrightarrow{\mathcal{F}} ?$

**Solution:** Suppose  $g(t) \xleftrightarrow{\mathcal{F}} G(f)$ . Then,

$$g(t) e^{j2\pi f_0 t} \xleftrightarrow{\mathcal{F}} \int_{-\infty}^{\infty} g(t) e^{-j2\pi (f-f_0)t} dt \quad (3.16)$$

$$= F(f - f_0) \quad (3.17)$$

Using (3.10) in (3.15),  $1 \xleftrightarrow{\mathcal{F}} \delta(-f)$ . Hence, applying (3.17),

$$e^{-j2\pi f_0 t} \xleftrightarrow{\mathcal{F}} \delta(-(f + f_0)) = \delta(f + f_0) \quad (3.18)$$

3.7  $\cos(2\pi f_0 t) \xleftrightarrow{\mathcal{F}} ?$

**Solution:** Using the linearity of the Fourier

Transform and (3.18),

$$\cos(2\pi f_0 t) = \frac{1}{2} (e^{j2\pi f_0 t} + e^{-j2\pi f_0 t}) \quad (3.19)$$

$$\xleftrightarrow{\mathcal{F}} \frac{1}{2} (\delta(f + f_0) + \delta(f - f_0)) \quad (3.20)$$

3.8 Find the Fourier Transform of  $x(t)$  and plot it. Verify using python.

**Solution:** Substituting (2.12) in (2.1),

$$x(t) \xleftrightarrow{\mathcal{F}} \sum_{k=-\infty}^{\infty} c_k \delta(f + kf_0) \quad (3.21)$$

$$= \frac{2A_0}{\pi} \sum_{k=-\infty}^{\infty} \frac{\delta(f + 2kf_0)}{1 - 4k^2} \quad (3.22)$$

The following code plots Fig. 3.8

```
wget https://github.com/omkar30122001/
Fourier-Series/blob/main/3.8.py
```

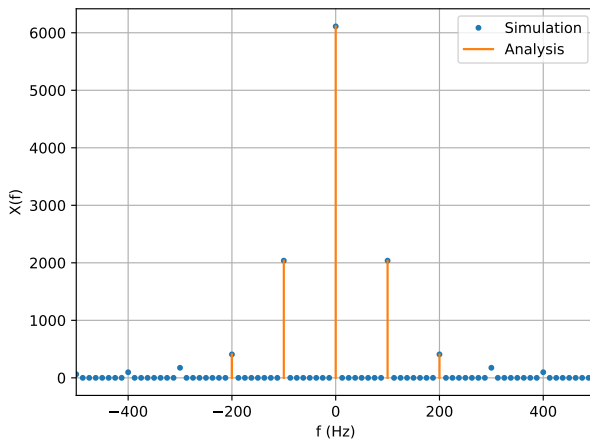


Fig. 3.8: Fourier Transform of  $x(t)$ .

3.9 Show that

$$\text{rect}(t) \xleftrightarrow{\mathcal{F}} \text{sinc}(f) \quad (3.23)$$

Verify using python.

**Solution:** We write

$$\text{rect}(t) \xleftrightarrow{\mathcal{F}} \int_{-\infty}^{\infty} \text{rect}(t) e^{-j2\pi ft} dt \quad (3.24)$$

$$= \int_{-\frac{1}{2}}^{\frac{1}{2}} e^{-j2\pi ft} dt \quad (3.25)$$

$$= \frac{e^{j\pi f} - e^{-j\pi f}}{j2\pi f} = \frac{\sin \pi f}{\pi f} = \text{sinc}(f) \quad (3.26)$$

The following python code plots Fig. 3.9

```
wget https://github.com/omkar30122001/
Fourier-Series/blob/main/3.9.py
```

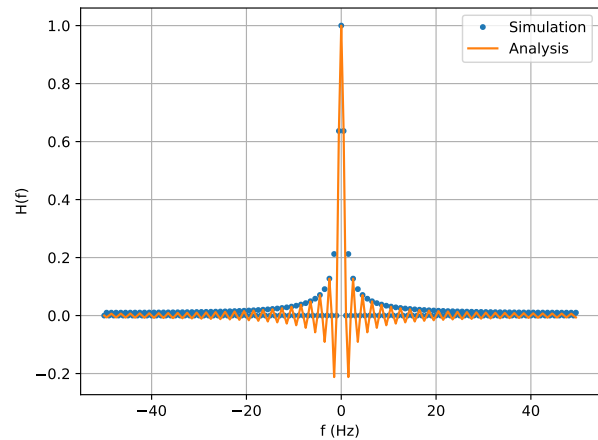


Fig. 3.9: Fourier Transform of  $\text{rect}(t)$ .

3.10  $\text{sinc}(t) \xleftrightarrow{\mathcal{F}} ?$ . Verify using python.

**Solution:** From (3.10), we have

$$\text{sinc}(t) \xleftrightarrow{\mathcal{F}} \text{rect}(f) = \text{rect}(f) \quad (3.27)$$

Since  $\text{rect}(f)$  is an even function. The following python code plots Fig. 3.10

```
wget https://github.com/omkar30122001/
Fourier-Series/blob/main/3.10.py
```

## 4 FILTER

4.1 Find  $H(f)$  which transforms  $x(t)$  to DC 5V.

**Solution:** The function  $H(f)$  is a low pass filter which filters out even harmonics and leaves the zero frequency component behind. The rectangular function represents an ideal

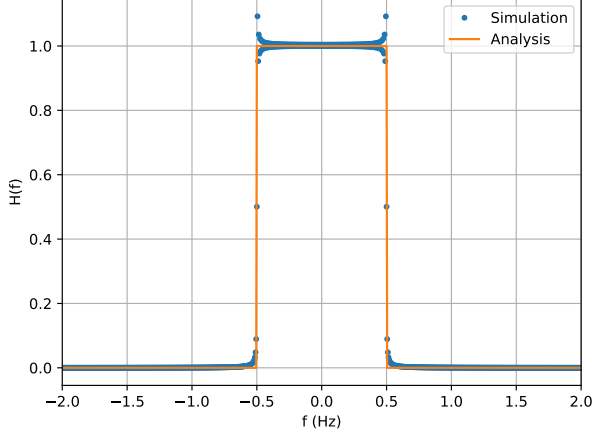


Fig. 3.10: Fourier Transform of  $\text{sinc}(\cdot)t$ .

low pass filter. Suppose the cutoff frequency is  $f_c = 50$  Hz, then

$$H(f) = \text{rect}\left(\frac{f}{2f_c}\right) = \begin{cases} 1 & |f| < f_c \\ 0 & \text{otherwise} \end{cases} \quad (4.1)$$

Multiplying by a scaling factor to get DC 5V,

$$H(f) = \frac{\pi V_0}{2A_0} \text{rect}\left(\left(\frac{f}{2f_c}\right)\right) \quad (4.2)$$

where  $V_0 = 5$  V.

4.2 Find  $h(t)$ .

**Solution:** Suppose  $g(t) \xleftrightarrow{\mathcal{F}} G(f)$ . Then, for some nonzero  $a \in \mathbb{R}$

$$g(at) \xleftrightarrow{\mathcal{F}} \int_{-\infty}^{\infty} g(at) e^{-j2\pi ft} dt \quad (4.3)$$

$$= \frac{1}{a} \int_{-\infty}^{\infty} g(u) e^{-j2\pi \frac{f}{a} t} dt \quad (4.4)$$

$$= \frac{1}{a} G\left(\frac{f}{a}\right) \quad (4.5)$$

where we have substituted  $u := at$ . Using (4.5) of the Fourier Transform in (4.1),

$$h(t) = \frac{2\pi V_0}{A_0} f_c \text{sinc}(2f_c t) \quad (4.6)$$

4.3 Verify your result using through convolution.

**Solution:** The following code plots Fig. 4.3

```
wget https://github.com/omkar30122001/
Fourier-Series/blob/main/4.3.py
```

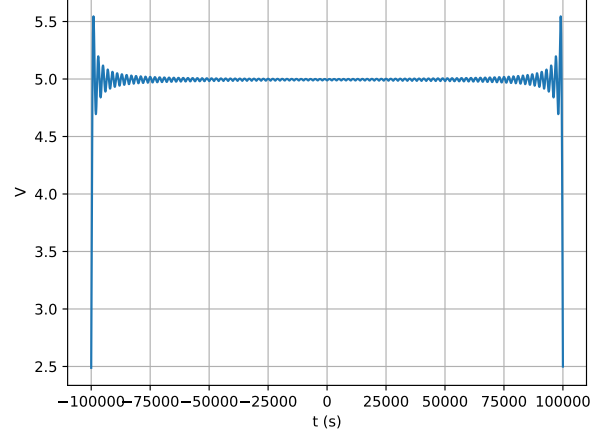


Fig. 4.3: Convolution of the two signals.

## 5 FILTER DESIGN

5.1 Design a Butterworth filter for  $H(f)$ .

**Solution:** The Butterworth filter has an amplitude response given by

$$|H(f)|^2 = \frac{1}{1 + \left(\frac{f}{f_c}\right)^{2n}} \quad (5.1)$$

where  $n$  is the order of the filter and  $f_c$  is the cutoff frequency. The attenuation at frequency  $f$  is given by

$$A = -10 \log_{10} |H(f)|^2 \quad (5.2)$$

$$= -20 \log_{10} |H(f)| \quad (5.3)$$

We consider the following design parameters for our lowpass analog Butterworth filter:

- a) Passband edge,  $f_p = 50$  Hz
- b) Stopband edge,  $f_s = 100$  Hz
- c) Passband attenuation,  $A_p = -1$  dB
- d) Stopband attenuation,  $A_s = -20$  dB

We are required to find a desirable order  $n$  and cutoff frequency  $f_c$  for the filter. From (5.3),

$$A_p = -10 \log_{10} \left[ 1 + \left(\frac{f_p}{f_c}\right)^{2n} \right] \quad (5.4)$$

$$A_s = -10 \log_{10} \left[ 1 + \left(\frac{f_s}{f_c}\right)^{2n} \right] \quad (5.5)$$

Thus,

$$\left(\frac{f_p}{f_c}\right)^{2n} = 10^{-\frac{A_p}{10}} - 1 \quad (5.6)$$

$$\left(\frac{f_s}{f_c}\right)^{2n} = 10^{-\frac{A_s}{10}} - 1 \quad (5.7)$$

Therefore, on dividing the above equations and solving for  $n$ ,

$$n = \frac{\log\left(10^{-\frac{A_s}{10}} - 1\right) - \log\left(10^{-\frac{A_p}{10}} - 1\right)}{2(\log f_s - \log f_p)} \quad (5.8)$$

In this case, making appropriate substitutions gives  $n = 4.29$ . Hence, we take  $n = 5$ . Solving for  $f_c$  in (5.6) and (5.7),

$$f_{c1} = f_p \left[10^{-\frac{A_p}{10}} - 1\right]^{-\frac{1}{2n}} = 57.23 \text{ Hz} \quad (5.9)$$

$$f_{c2} = f_s \left[10^{-\frac{A_s}{10}} - 1\right]^{-\frac{1}{2n}} = 63.16 \text{ Hz} \quad (5.10)$$

Hence, we take  $f_c = \sqrt{f_{c1}f_{c2}} = 60 \text{ Hz}$  approximately.

## 5.2 Design a Chebyshev filter for $H(f)$ .

**Solution:** The Chebyshev filter has an amplitude response given by

$$|H(f)|^2 = \frac{1}{\left(1 + \epsilon^2 C_n^2\left(\frac{f}{f_c}\right)\right)} \quad (5.11)$$

where

- a)  $n$  is the order of the filter
- b)  $\epsilon$  is the ripple
- c)  $f_c$  is the cutoff frequency
- d)  $C_n = \cosh^{-1}(n \cosh x)$  denotes the  $n^{\text{th}}$  order Chebyshev polynomial, given by

$$c_n(x) = \begin{cases} \cos(n \cos^{-1} x) & |x| \leq 1 \\ \cosh(n \cosh^{-1} x) & \text{otherwise} \end{cases} \quad (5.12)$$

We are given the following specifications:

- a) Passband edge (which is equal to cutoff frequency),  $f_p = f_c$
- b) Stopband edge,  $f_s$
- c) Attenuation at stopband edge,  $A_s$
- d) Peak-to-peak ripple  $\delta$  in the passband. It is given in dB and is related to  $\epsilon$  as

$$\delta = 10 \log_{10}(1 + \epsilon^2) \quad (5.13)$$

and we must find a suitable  $n$  and  $\epsilon$ . From (5.13),

$$\epsilon = \sqrt{10^{\frac{\delta}{10}} - 1} \quad (5.14)$$

At  $f_s > f_p = f_c$ , using (5.12),  $A_s$  is given by

$$A_s = -10 \log_{10} \left[1 + \epsilon^2 c_n^2\left(\frac{f_s}{f_p}\right)\right] \quad (5.15)$$

$$\Rightarrow c_n\left(\frac{f_s}{f_p}\right) = \frac{\sqrt{10^{-\frac{A_s}{10}} - 1}}{\epsilon} \quad (5.16)$$

$$\Rightarrow n = \frac{\cosh^{-1}\left(\frac{\sqrt{10^{-\frac{A_s}{10}} - 1}}{\epsilon}\right)}{\cosh^{-1}\left(\frac{f_s}{f_p}\right)} \quad (5.17)$$

We consider the following specifications:

- a) Passband edge/cutoff frequency,  $f_p = f_c = 60 \text{ Hz}$ .
  - b) Stopband edge,  $f_s = 100 \text{ Hz}$ .
  - c) Passband ripple,  $\delta = 0.5 \text{ dB}$
  - d) Stopband attenuation,  $A_s = -20 \text{ dB}$
- $\epsilon = 0.35$  and  $n = 3.68$ . Hence, we take  $n = 4$  as the order of the Chebyshev filter.

## 5.3 Design a circuit for your Butterworth filter.

**Solution:** Looking at the table of normalized element values  $L_k, C_k$ , of the Butterworth filter for order 5, and noting that de-normalized values  $L'_k$  and  $C'_k$  are given by

$$C'_k = \frac{C_k}{\omega_c} \quad L'_k = \frac{L_k}{\omega_c} \quad (5.18)$$

De-normalizing these values, taking  $f_c = 60 \text{ Hz}$ ,

$$C'_1 = C'_5 = 1.64 \text{ mF} \quad (5.19)$$

$$L'_2 = L'_4 = 4.29 \text{ mH} \quad (5.20)$$

$$C'_3 = 5.31 \text{ mF} \quad (5.21)$$

$$(5.22)$$

The L-C network is shown in Fig. 5.3. This cir-

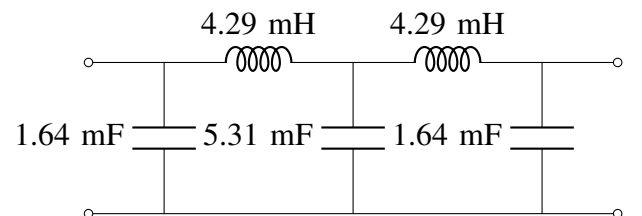


Fig. 5.3: L-C Butterworth Filter

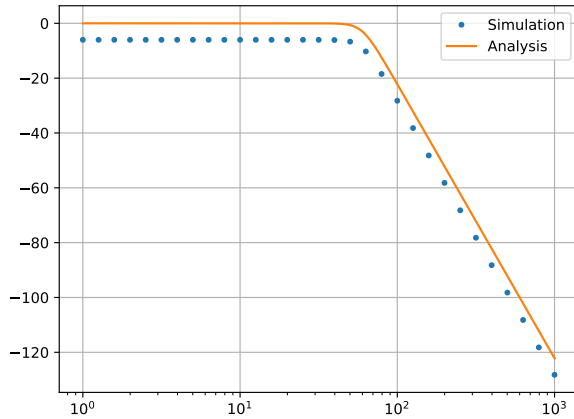


Fig. 5.3: Simulation of Butterworth filter.

cuit is simulated in the ngspice code 5.3.cir. The Python code 5.3.py compares the amplitude response of the simulated circuit with the theoretical expression.

#### 5.4 Design a circuit for your Chebyshev filter.

**Solution:** Looking at the table of normalized element values of the Chebyshev filter for order 3 and 0.5 dB ripple, and de-normalizing those values, taking  $f_c = 50$  Hz,

$$C'_1 = 4.43 \text{ mF} \quad (5.23)$$

$$L'_2 = 3.16 \text{ mH} \quad (5.24)$$

$$C'_3 = 6.28 \text{ mF} \quad (5.25)$$

$$L'_4 = 2.23 \text{ mH} \quad (5.26)$$

The L-C network is shown in Fig. (5.4). This circuit is simulated in the ngspice code

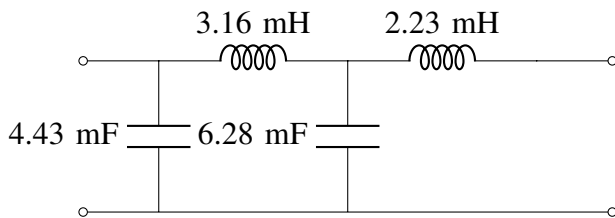


Fig. 5.4: L-C Chebyshev Filter

5.4.cir. The Python code 5.4.py compares the amplitude response of the simulated circuit with the theoretical expression.

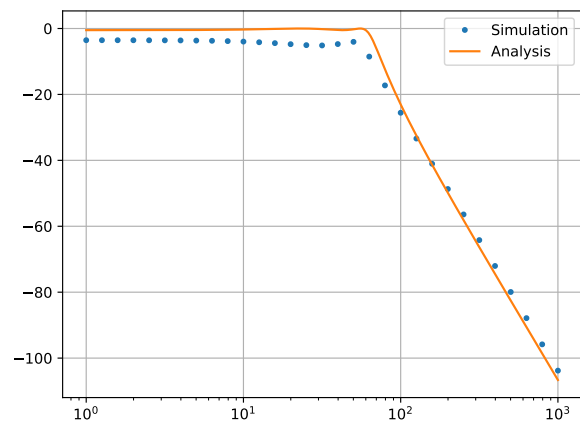


Fig. 5.4: Simulation of Chebyshev filter.