# Fourier

Mannem Charan AI21BTECH11019

Contents


**1 Periodic Function** 1

**2 Fourier Series** 1

**3 Fourier Transform** 3

**4 Filter** 5

**5 Filter Design** 6


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

## 1 Periodic Function

Let

$$x(t) = A_0 \left|\sin\left(2\pi f_0 t\right)\right| \tag{1.1}$$

1.1 Plot $x(t)$.
**Solution:** Download the python code for the plot of $x(t)$,

```
wget https://github.com/Charanyash/EE3900-
    Digital_Signal_Processing/blob/master/
    Fourier/Codes/1.1.py
```

Then run the following command,

```
python3 1.1.py
```

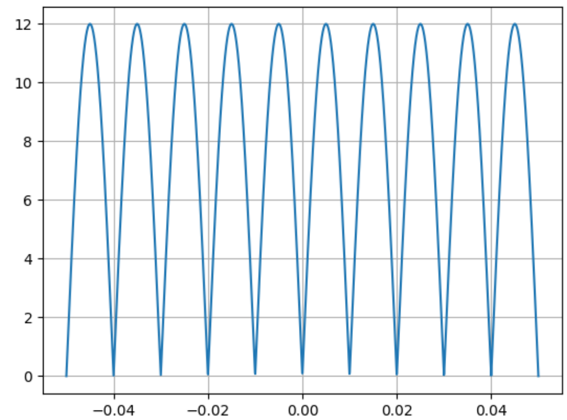


Fig. 1.1

1.2 Show that $x(t)$ is periodic and find its period.
**Solution:** We will say a function $f(x)$ is periodic if there exists a real number $T$, such that

$$f(x+T) = f(x) \tag{1.2}$$

So for the given $x(t)$ which is absolute sinusoidal function is also periodic which can be seen in the fig 1.1.For the period, consider the following

$$x(t + \frac{1}{2f_0}) = \left|\sin\left(2\pi f_0\left(t + \frac{1}{2f_0}\right)\right)\right| \tag{1.3}$$
$$= \left|\sin\left(2\pi f_0 t + \pi\right)\right| \tag{1.4}$$
$$= \left|-\sin\left(2\pi f_0 t\right)\right| \tag{1.5}$$
$$= \left|\sin\left(2\pi f_0 t\right)\right| \tag{1.6}$$

This shows that the $x(t)$ is periodic with period $\frac{1}{2f_0}$.

## 2 Fourier Series

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

2.1 If

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

show that

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

**Solution:** To show that,

$$c_k = f_0 \int_{-\frac{1}{2f_0}}^{\frac{1}{2f_0}} x(t) e^{-j2\pi k f_0 t}\, dt \tag{2.3}$$

Consider the RHS and use (2.1),

$$f_0 \int_{-\frac{1}{2f_0}}^{\frac{1}{2f_0}} x(t) e^{-j2\pi k f_0 t}\, dt = f_0 \int_{-\frac{1}{2f_0}}^{\frac{1}{2f_0}} \left( \sum_{n=-\infty}^{\infty} c_n e^{j2\pi n f_0 t} \right) e^{-j2\pi k f_0 t} \tag{2.4}$$

$$= f_0 \sum_{n=-\infty}^{\infty} \int_{-\frac{1}{2f_0}}^{\frac{1}{2f_0}} c_n e^{j2\pi(n-k) f_0 t}\, dt \tag{2.5}$$

And the definite integral evaluates to,

$$\int -\frac{1}{2f_0}^{\frac{1}{2f_0}} e^{j2\pi(n-k) f_0 t} = \begin{cases} \frac{1}{f_0}, & n = k \\ 0, & n \neq k \end{cases} \tag{2.6}$$

Using that,

$$f_0 \int_{-\frac{1}{2f_0}}^{\frac{1}{2f_0}} x(t) e^{-j2\pi k f_0 t}\, dt = f_0 \left( \frac{c_k}{f_0} \right) \tag{2.7}$$

$$= c_k \tag{2.8}$$

Hence proved.

2.2 Find $c_k$ for (1.1)
**Solution:** To find $c_k$ for given $x(t)$ we will use (2.2),

$$c_k = f_0 \int_{-\frac{1}{2f_0}}^{\frac{1}{2f_0}} x(t) e^{-j2\pi k f_0 t}\, dt \tag{2.9}$$

$$= A_0 f_0 \int_0^{\frac{1}{2f_0}} \sin(2\pi f_0 t) \left( e^{-j2\pi k f_0 t} + e^{-j2\pi k f_0 t} \right) \tag{2.10}$$

$$\left( \because \int_{-a}^{a} f(x) dx = \int_0^a f(a) + f(-a)\, dx \right)$$

$$= A_0 f_0 \int_0^{\frac{1}{2f_0}} \sin(2\pi f_0 t) \left( 2 \cos(2\pi k f_0 t) \right) \tag{2.11}$$

$$= A_0 f_0 \int_0^{\frac{1}{2f_0}} \left( \sin(2\pi f_0 t (1-k)) + \sin(2\pi f_0 t (1+k)) \right) \tag{2.12}$$

$$= A_0 f_0 \left[ \frac{\cos(2\pi f_0 t (k-1))}{2\pi f_0 (k-1)} - \frac{\cos(2\pi f_0 t (k+1))}{2\pi f_0 (k+1)} \right]_0^{\frac{1}{2f_0}} \tag{2.13}$$

$$= A_0 \frac{(\cos(\pi(k-1)) - 1)}{2\pi(k-1)} - \frac{(\cos(\pi(k+1)) - 1)}{2\pi(k+1)} \tag{2.14}$$

$$= \frac{A_0 \left( (-1)^{k+1} - 1 \right)}{2\pi} \left[ \frac{1}{k-1} - \frac{1}{k+1} \right] \tag{2.15}$$

$$= \frac{A_0 \left( (-1)^{k+1} - 1 \right)}{\pi (k^2 - 1)} \tag{2.16}$$

In other words,

$$c_k = \begin{cases} 0 & \text{, k is odd} \\ \frac{2A_0}{\pi(1-k^2)} & \text{, k is even} \end{cases} \tag{2.17}$$

2.3 Verify (1.1) using python.
**Solution:** Download the python code from the below link

```
wget https://github.com/Charanyash/EE3900-
    Digital_Signal_Processing/blob/master/
    Fourier/Codes/2.3.py
```

Then run the following command in terminal

```
python3 2.3.py
```

2.4 Show that

$$x(t) = \sum_{k=0}^{\infty} \left( a_k \cos j2\pi k f_0 t + b_k \sin \jmath 2\pi k f_0 t \right) \tag{2.18}$$

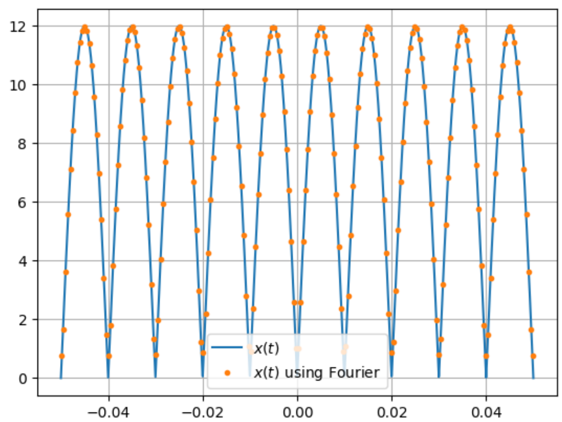


Fig. 2.3

and obtain the formulae for $a_k$ and $b_k$.
**Solution:** From (2.1),

$$x(t) = \sum_{k=-\infty}^{\infty} c_k e^{-j2\pi k f_0 t} \tag{2.19}$$

$$= \sum_{k=-\infty}^{\infty} c_k \left(\cos\left(2\pi k f_0 t\right) - \jmath \sin\left(2\pi k f_0 t\right)\right) \tag{2.20}$$

$$= c_0 + \sum_{k=1}^{\infty} \cos\left(2\pi k f_0 t\right)\left(c_k + c_{-k}\right)$$

$$+ \sum_{k=1}^{\infty} \left(c_k - c_{-k}\right) \sin\left(2\pi k f_0 t\right) \tag{2.21}$$

Now by comparing we can write $a_k$ and $b_k$ as
,

$$a_k = \begin{cases} c_0 & , k = 0 \\ c_k + c_{-k} & , k \neq 0 \end{cases} \tag{2.22}$$

$$b_k = c_k - c_{-k} \tag{2.23}$$

2.5 Find $a_k$ and $b_k$ for (1.1)
**Solution:** Using (2.17), we will get $a_k$ and $b_k$ as,

$$a_k = \begin{cases} 0 & , \text{k is odd} \\ \frac{2A_0}{\pi} & , k = 0 \\ \frac{4A_0}{\pi(1-k^2)} & , \text{k is even} - \{0\} \end{cases} \tag{2.24}$$

$$b_k = 0 \forall k \tag{2.25}$$

Note that $c_k = c_{-k} \forall k$.

2.6 Verify (2.18) using python.
**Solution:** Download the python code from the below link,

```
wget https://github.com/Charanyash/EE3900
    −Digital_Signal_Processing/blob/master
    /Fourier/Codes/2.6.py
```

Then run the following command,

```
python3 2.6.py
```

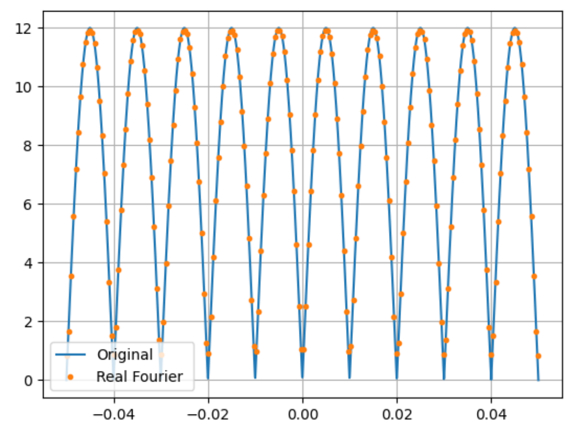


Fig. 2.6

## 3 Fourier Transform

3.1

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

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

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

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

3.3 Show that

$$g(t - t_0) \stackrel{\mathcal{F}}{\longleftrightarrow} G(f) e^{-j2\pi f t_0} \tag{3.4}$$

$$\tag{3.5}$$

**Solution:** Using the definition (3.3),

$$\mathcal{F}\left\{g(t-t_0)\right\} = \int_{-\infty}^{\infty} g(t-t_0)e^{-j2\pi ft}\,dt \tag{3.6}$$
$$= \int_{-\infty}^{\infty} g(k)e^{-j2\pi f(t_0+k)}\,dk \tag{3.7}$$
$$= e^{-j2\pi ft_0}\int_{-\infty}^{\infty} g(k)e^{-j2\pi fk}\,dk \tag{3.8}$$
$$= G(f)e^{-j2\pi ft_0} \tag{3.9}$$

Hence proved.

3.4 Show that

$$G(t) \xleftrightarrow{\mathcal{F}} g(-f) \tag{3.10}$$

**Solution:** We know that ,

$$g(t) \xleftrightarrow{\mathcal{F}} G(f) \tag{3.11}$$

So we can write,

$$g(t) = \int_{-\infty}^{\infty} G(f)e^{j2\pi ft}\,df \tag{3.12}$$
$$= \int_{-\infty}^{\infty} G(k)e^{j2\pi kt}\,dk \tag{3.13}$$
$$\implies g(-f) = \int_{-\infty}^{\infty} G(k)e^{-j2\pi kf}\,dk \tag{3.14}$$
$$= \mathcal{F}\left\{G(t)\right\} \tag{3.15}$$

Hence proved.

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

**Solution:** We know that,

$$\int \delta(t-t_0)f(t) = f(t_0) \tag{3.16}$$

So,

$$\delta(t) \xleftrightarrow{\mathcal{F}} = \int_{-\infty}^{\infty} \delta(t)e^{-j2\pi ft}dt \tag{3.17}$$
$$= e^{-j2\pi ft}|_{t=0} \tag{3.18}$$
$$= 1 \tag{3.19}$$

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

**Solution:** From above we know that,

$$\delta(t) \xleftrightarrow{\mathcal{F}} 1 \tag{3.20}$$
$$\delta(t-t_0) \xleftrightarrow{\mathcal{F}} 1.e^{-j2\pi ft_0} \tag{3.21}$$
$$\implies e^{-j2\pi tf_0} \xleftrightarrow{\mathcal{F}} \delta(-f-f_0) \tag{3.22}$$
$$\implies e^{-j2\pi tf_0} \xleftrightarrow{\mathcal{F}} \delta(f+f_0) \tag{3.23}$$

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

**Solution:** We know that,

$$\cos(2\pi f_0 t) = \frac{e^{j2\pi f_0 t} + e^{-j2\pi f_0 t}}{2} \tag{3.24}$$

Now if we apply fourier transform on both sides,

$$\mathcal{F}\left\{\cos 2\pi f_0 t\right\} = \frac{1}{2}\left[\delta\left(f-f_0\right) + \delta\left(f+f_0\right)\right] \tag{3.25}$$

$\therefore \cos(2\pi f_0 t) \xleftrightarrow{\mathcal{F}} \frac{\delta(f-f_0)+\delta(f+f_0)}{2}$

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

**Solution:** To find the fourier transform of $x(t)$ we can use the fourier series expansion,

$$x(t) = \sum_{k=-\infty}^{\infty} c_k e^{j2\pi kf_0 t} \tag{3.26}$$
$$\implies \mathcal{F}\left\{x(t)\right\} = \sum_{k=-\infty}^{\infty} c_k \mathcal{F}\left\{e^{j2\pi kf_0 t}\right\} \tag{3.27}$$
$$= \sum_{k=-\infty}^{\infty} c_k \delta\left(f-kf_0\right) \tag{3.28}$$

Now using (2.17),

$$\mathcal{F}\left\{x(t)\right\} = \sum_{k \text{ is even}} \frac{2A_0\delta\left(f-kf_0\right)}{\pi\left(1-k^2\right)} \tag{3.29}$$

The same can be seen in Fig 3.8. Download the python code from the below link,

```
wget https://github.com/Charanyash/EE3900
    −Digital_Signal_Processing/blob/master
    /Fourier/Codes/3.8.py
```

Then run the following command in terminal,

```
python3 3.8.py
```

3.9 Show that

$$\text{rect}\, t \xleftrightarrow{\mathcal{F}} \text{sinc}\, f \tag{3.30}$$

Verify using python.

**Solution:** The $rect(t)$ is defined as,

$$\text{rect}\, t = \begin{cases} 1 & , |t| \le \frac{1}{2} \\ 0 & , \text{otherwise} \end{cases} \tag{3.31}$$

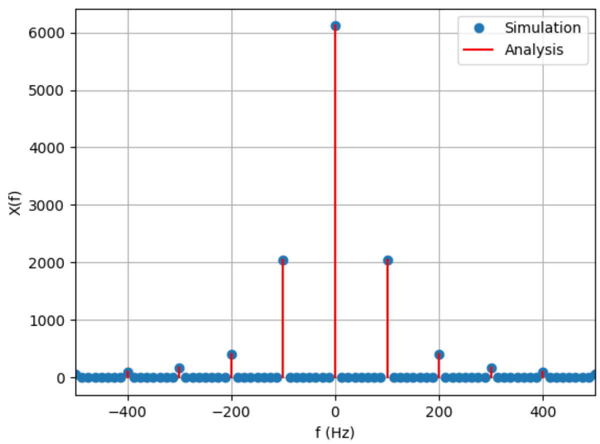


Fig. 3.8

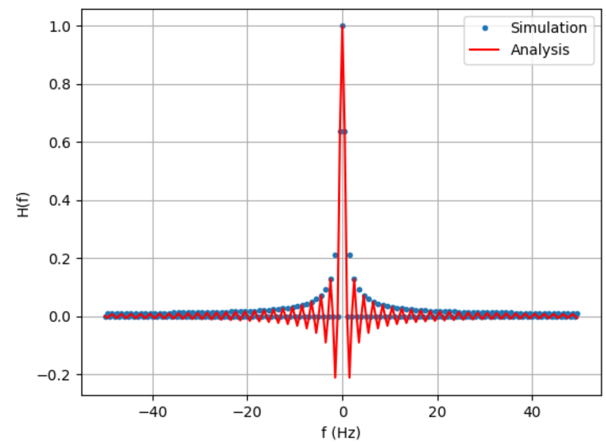


Fig. 3.9

So fourier transform will be,

$$\mathcal{F}\{\text{rect}\, t\} = \int_{-\infty}^{\infty} \text{rect}\, t e^{-j2\pi ft}\, dt \tag{3.32}$$

$$= \int_{-\frac{1}{2}}^{\frac{1}{2}} e^{-j2\pi ft}\, dt \tag{3.33}$$

$$= \int_{0}^{\frac{1}{2}} e^{-j2\pi ft} + e^{j2\pi ft}\, dt \tag{3.34}$$

$$= 2\int_{0}^{\frac{1}{2}} \cos(2\pi ft)\, dt \tag{3.35}$$

$$= 2\left[\frac{\sin(2\pi ft)}{2\pi f}\right]_0^{\frac{1}{2}} \tag{3.36}$$

$$= \frac{\sin(\pi f)}{\pi f} \tag{3.37}$$

$$= \text{sinc}\, f \tag{3.38}$$

Download the python code from the below link,

```
wget https://github.com/Charanyash/EE3900−
    Digital_Signal_Processing/blob/master/
    Fourier/Codes/3.9.py
```

Run the following command,

```
python3 3.9.py
```

3.10 $\text{sinc}\, t \stackrel{\mathcal{F}}{\longleftrightarrow} ?$. Verify using python.
**Solution:** From the duality property of Fourier Transform,

$$\text{rect}\, t \stackrel{\mathcal{F}}{\longleftrightarrow} \text{sinc}\, f \tag{3.39}$$

$$\implies \text{sinc}\, t \stackrel{\mathcal{F}}{\longleftrightarrow} \text{rect} - f \tag{3.40}$$

Since $\text{rect}\, t$ is an even function,

$$\text{sinc}\, t \stackrel{\mathcal{F}}{\longleftrightarrow} \text{rect}\, f \tag{3.41}$$

Download the python code from the following link,

```
wget https://github.com/Charanyash/EE3900−
    Digital_Signal_Processing/blob/master/
    Fourier/Codes/3.10.py
```

Then run the following command,

```
python3 3.10.py
```

## 4 Filter

4.1 Find $H(f)$ which transforms $x(t)$ to DC 5V.
**Solution:** Since we need a DC output of $5V$, we need a filter which will remove the higher frequencies i.e., a low-pass filter so that we can retrieve the zero frequency components. So we need a filter which only allows certain frequencies lower than a cuttoff frequency ($f_c$). One can use $rect(f)$ as a low-pass filter,

$$H(f) = krect\left(\frac{f}{2f_c}\right) = \begin{cases} k & , f \le f_c \\ 0 & , \text{otherwise} \end{cases} \tag{4.1}$$

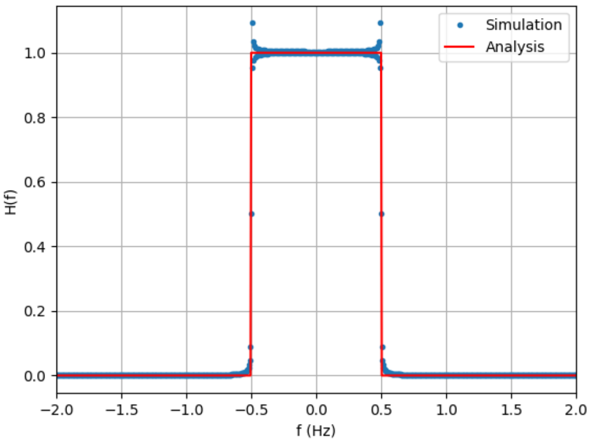


Fig. 3.10

And $k$ is amplification factor which makes $a_0$ of $x(t)$ as $5V$. We can evaluate the same as,

$$H(0) = \frac{Y(0)}{X(0)} \tag{4.2}$$

$$k = \frac{5}{\frac{2A_0}{\pi}} \tag{4.3}$$

This makes the transfer function $H(f)$ as,

$$H(f) = \frac{5\pi}{2A_0} rect\left(\frac{f}{2f_c}\right) \tag{4.4}$$

4.2 Find $h(t)$.
**Solution:** We know that,

$$sinc(t) \overset{\mathcal{F}}{\longleftrightarrow} rect(f) \tag{4.5}$$

$$sinc(2f_c t) \overset{\mathcal{F}}{\longleftrightarrow} \frac{1}{2f_c} rect(\frac{f}{2f_c}) \tag{4.6}$$

$$\therefore \frac{10f_c\pi}{2A_0} sinc(2f_c t) \overset{\mathcal{F}}{\longleftrightarrow} \frac{5\pi}{2A_0} rect\left(\frac{f}{2f_c}\right) \tag{4.7}$$

The impulse response will be,

$$h(t) = \frac{5f_c\pi}{A_0} sinc(2f_c t) \tag{4.8}$$

4.3 Verify your result using through convolution.
**Solution:** Download the python code from the following link,

```
wget https://github.com/Charanyash/EE3900−
    Digital_Signal_Processing/blob/master/
    Fourier/Codes/4.3.py
```

Then run the following command,

```
python3 4.3.py
```

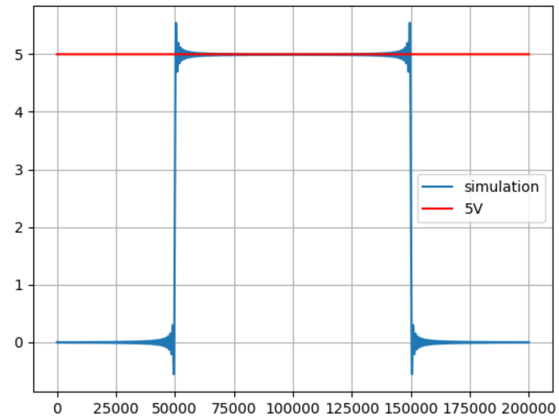


Fig. 4.3

## 5 Filter Design

5.1 Design a Butterworth filter for $H(f)$.
**Solution:** The Butterworth filter has an amplitude response given by

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

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

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

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

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 desriable order $n$ and cutoff frequency $f_c$ for the filter. From (5.3),

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

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

Thus,

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

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

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

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

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

$$f_{c1} = f_p\left[10^{-\frac{A_p}{10}} - 1\right]^{-\frac{1}{2n}} = 57.23\,Hz \tag{5.9}$$

$$f_{c2} = f_s\left[10^{-\frac{A_s}{10}} - 1\right]^{-\frac{1}{2n}} = 63.26\,Hz \tag{5.10}$$

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

5.2 Design a Chebyschev filter for $H(f)$.
**Solution:** The Chebyshev filter has an amplitude response given by

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

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[th] order Chebyshev polynomial, given by

$$c_n(x) = \begin{cases} \cos\left(n\cos^{-1}x\right) & |x| \le 1 \\ \cosh\left(n\cosh^{-1}x\right) & \text{otherwise} \end{cases} \tag{5.12}$$

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

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

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

$$\epsilon = \sqrt{10^{\frac{\delta}{10}} - 1} \tag{5.14}$$

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

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

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

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

We consider the following specifications:

a) Passband edge/cutoff frequency, $f_p = f_c = 60\,Hz$.
b) Stopband edge, $f_s = 100\,Hz$.
c) Passband ripple, $\delta = 0.5\,dB$
d) Stopband attenuation, $A_s = -20\,dB$

$\epsilon = 0.35$ 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

$$C'_k = \frac{C_k}{\omega_c} \qquad L'_k = \frac{L_k}{\omega_c} \tag{5.18}$$

De-normalizing these values, taking $f_c = 60$ Hz,

$$C'_1 = C'_5 = 1.64\mu F \tag{5.19}$$

$$L'_2 = L'_4 = 4.29\mu H \tag{5.20}$$

$$C'_3 = 5.31\mu F \tag{5.21}$$

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

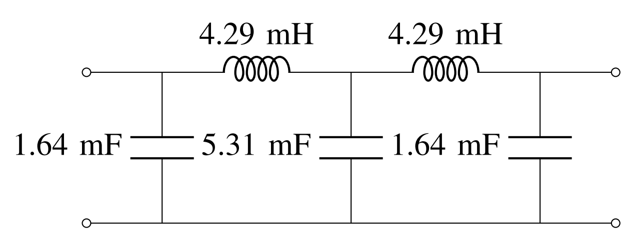


Fig. 5.3: L-C Butterworth Filter

Below python code plot the figure 5.3

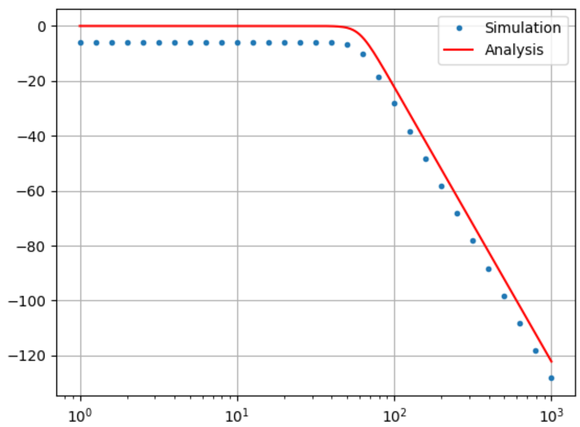


Fig. 5.3: Simulation of Butterworth filter.

```
wget https://github.com/Charanyash/EE3900−
    Digital_Signal_Processing/blob/master/
    Fourier/Codes/5.3.py
```

5.4 Design a circuit for your Chebyschev filter.
**Solution:** Looking at the table of normalized element values of the Chebyshev filter for order 3 and 0.5 dB ripple, and de-nomrmalizing those values, taking $f_c = 50\,Hz$,

$$C_1' = 4.43\mu F \tag{5.22}$$

$$L_2' = 3.16\mu F \tag{5.23}$$

$$C_3' = 6.28\mu F \tag{5.24}$$

$$L_4' = 2.23\mu F \tag{5.25}$$

The L-C network is shown in Fig. 5.4. Below

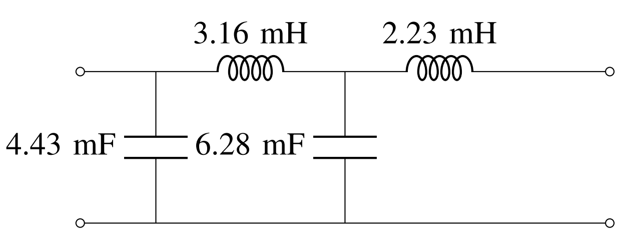


Fig. 5.4: L-C Chebyshev Filter

python code plot the figure 5.4

```
wget https://github.com/Charanyash/EE3900−
    Digital_Signal_Processing/blob/master/
    Fourier/Codes/5.4.py
```

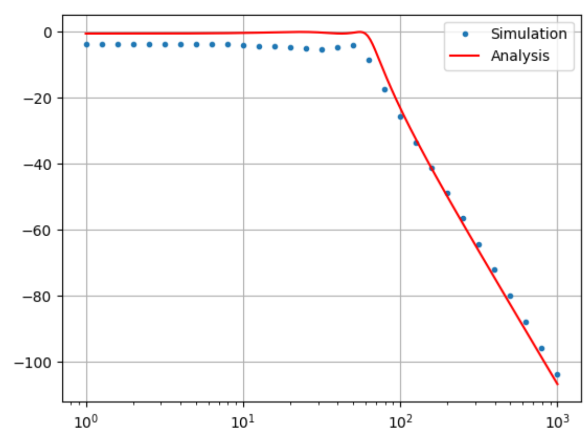


Fig. 5.4: Simulation of Chebyshev filter.