
電気電子情報一（前期）実験

P3. 回路シミュレーションとフィルタ設計

総合レポート

03-160441 土屋潤一郎（工学部 電子情報工学科）
共同実験者: 井上友貴、田中大幹、坂口達彦（第 28 班）
2016 年 6 月 2 日

1. 実験の目的

素子のパラメータや入力信号の条件を容易に変更可能で、数値計算によって回路の応答特性を求めるツールである回路シミュレータの使い方を学ぶ。また、これを用いて Butterworth フィルタと Chebyshev フィルタとを設計し、その応答特性を実際にシミュレートして学ぶ。

2. 実験の原理

2.1 フィルタ設計の正規化とインピーダンススケージング・周波数変換

フィルタ設計に於いては、主として出力側のインピーダンスと通過帯域の幅が個々の成分の値を決定する。従って、様々な仕様を与えられたときに、それらの値を簡単に決定できるように、正規化された設計法を考えておく。その上で、しかるべく変数の変換を行い、仕様を満たすフィルタを設計する。

2.1.1 インピーダンスの正規化

四端子回路によるフィルタは、伝達関数の出力/入力をどのように取るのかで、いくつかの形式が分別される（表 1）。

表 1 フィルタの形式

形式	出力	入力	備考
0-R 型	出力電圧	入力電圧	電源の内部抵抗を無視
R- ∞ 型	開放出力電圧	電源開放電圧	出力負荷抵抗を 0 とする
R-R 型	出力電圧	電源開放電圧	電源内部抵抗、出力負荷抵抗を考慮

表 1 で R と表される各種の抵抗を全て $R = 1\Omega$ に規格化した低域通過フィルタは、それぞれの抵抗の値が任意のインピーダンスを取る場合に改造できる。

2.1.2 通過域と減衰域の正規化

通過域と減衰域に関する正規化は、遮断角周波数を $\omega_c = 1[\text{rad s}^{-1}]$ とした低域通過フィルタの設計に帰着できる。

四端子回路の挿入による損失 $L(\omega)[\text{-dB}]$ を考えると、伝達関数 $F(j\omega)$ に対して、

$$L(\omega) = 20 \log_{10} |F(j\omega)| \quad (1)$$

なので、 $|F(j\omega)| = |F(-j\omega)|$ より周波数に対して偶関数である。さらに、 ω は関数 $L(\omega)$ の中に、 $L\omega$ 、 $C\omega$ 、 $M\omega$ のとその組み合わせで出現する。この性質を利用して、表 2 に示す（基準の規格化低域通過フィルタを $k = \omega$ に対して）周波

数の変数変換を行うことで、フィルタの種類に対応したインダクタンス及びキャパシタンスの構成とその値を定めることが出来る。

表 2 周波数変換

フィルタの種類	変数変換
低域通過フィルタ	$\frac{\omega}{\omega_c}$
高域通過フィルタ	$-\frac{\omega_c}{\omega}$
帯域通過フィルタ	$\frac{\omega_0}{\omega_b} \left(\frac{\omega}{\omega_0} - \frac{\omega_0}{\omega} \right)$
帯域除去フィルタ	$-\frac{\omega_b}{\omega_\infty} \bullet \frac{1}{\frac{\omega}{\omega_\infty} - \frac{\omega_\infty}{\omega}}$

但し、 ω_1 と ω_2 の間が通過域/除去域として、 $\omega_b = \omega_2 - \omega_1$ $\omega_0, \omega_\infty = \sqrt{\omega_2 \omega_1}$

2.2 Butterworth フィルタ

Butterworth 特性とは、最大平坦特性とも言われ、伝達関数の次数 n に対して通過帯域が実現可能な範囲で最も平坦であるような周波数特性である。この特性は、伝達関数 $F(\omega)$ に対して、

$$|F(\omega)|^2 = \frac{1}{1 + \omega^{2n}} \quad (2)$$

で与えられる。

安定な回路を作りたいので、 $s = j\omega$ としたときの $|F(s)|^2$ の極（複素平面の単位円を $2n$ 等分する点の集合となる）のうち、左反面に存在する n 個が極となるように $F(s)$ を定めれば良い。

2.3 Chebyshev フィルタ

Chebyshev 特性とは、実現したい伝達関数（の曲線）を近似するとき、その近似誤差の評価関数として Chebyshev ノルム

$$\|F\| = \max_{x \in \Delta} |F(x)| \quad (3)$$

を用いたフィルタの持つ特性である。このノルムを用いた近似を行う帯域では、近似関数は実現したい関数に対して正負の値を交互にとる。（符号を変える回数は、近似関数のパラメータ n 個に対して少なくとも $n-1$ 回である。）この特性を持つフィルタは近似を行う帯域によっていくつかの種類に分類できるが、本実験で設計するのは、通過域でこの評価関数を適用し（通過域でのリップルを許容し）減衰域に零点がない（伝送関数の減衰極が無遠点にある）無極形通過域 Chebyshev フィルタである。

この特性は、伝達関数 $F(\omega)$ に対して、

$$|F(\omega)|^2 = \frac{1}{1 + \epsilon^2 V_n^2(\omega)} \quad (4)$$

で与えられる。（但し、 ϵ はリップルの大きさの程度を決める変数である。） $V_n(\omega)$ はチェビシェフの多項式として知られる、

$$V_n(\omega) = \begin{cases} \cos(n \cos^{-1}(\omega)) & \text{for } |\omega| \leq 1 \\ \cosh(n \cosh^{-1}(\omega)) & \text{for } |\omega| \geq 1 \end{cases} \quad (5)$$

なので、 $s = j\omega$ としたときの $|F(s)|^2$ の極は短径 $\sinh((1/n) \sinh^{-1}(1/\epsilon))$ 、長径 $\cosh((1/n) \sinh^{-1}(1/\epsilon))$ の楕円上の、偏角が $2\pi[\text{radian}]$ を $2n$ 等分する点のうち、左反面にある点 n 個である。

3. 実験方法

3.1 回路シミュレータ LTSpiceIV での簡単な回路のシミュレートとその数値データのプロット

図 1 に示す RLC 直列回路 ($R_{out} = 50[\Omega]$, $R_0 = 4.7[\Omega]$, $R_1 = 50[\Omega]$, $R_2 = 4.7[\Omega]$, $L_1 = 47\mu\text{H}$, $C_1 = 470\text{pF}$) の周波数特性とステップ応答をシミュレートする。

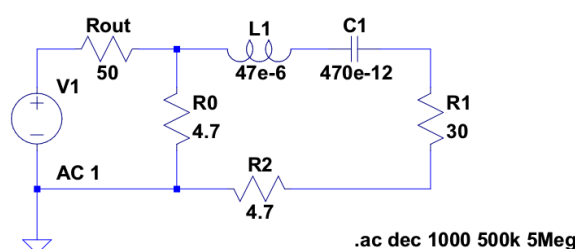


図 1 RLC 直列回路 (周波数特性の測定)

3.1.1 周波数特性のシミュレート

図 1 のように交流電圧源の電圧を 1V に設定し、周波数を 500 kHz から 5 MHz まで、周波数が 10 倍になる間に 1000 回の割合で計算を行う。

3.1.2 ステップ応答のシミュレート

図 2 のように矩形波電圧源の電圧を 1V、立ち上がり時刻を 0s、立ち上がり時間と立ち下がり時間を 10^{-50}s 、立ち上がっている時間を 1s、矩形波の周期を 2s に設定し、時刻 0s から 5 μs まで、50 ps に一回計算を行う。

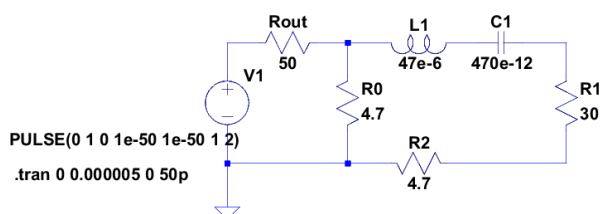


図 2 RLC 直列回路 (ステップ応答の測定)

3.1.3 数値データのエクスポート

LTSpiceIV では、回路を構成してシミュレート条件を指定して実行した後、file → export で数値データを .txt 形式で保存できる。

但し、完全な csv ファイルではないため、テキストエディタの全置換機能などで適宜プロットソフトが読みやすい形式にしてやる必要がある。また、位相差は $[-180(^{\circ}):180(^{\circ})]$ の範囲で数値データ化されてしまうため、グラフ化した際には値が一気に飛ぶように見えることもある。

3.1.4 数値データのプロット

数値データのプロットには gnuplot を用いる。CUI を用いて軸の範囲の設定や軸の目盛りの設定、グラフの出力形式などを設定できるが、スクリプトにすることも可能である。以下、本実験での設定を掲載する。

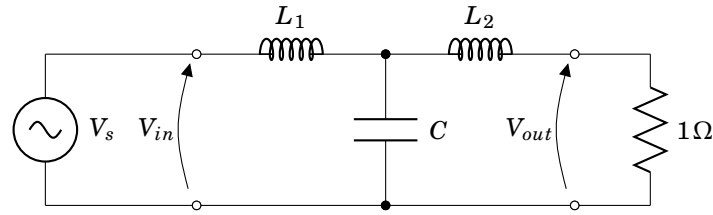


図3 3次規格化低域通過フィルタ

Listing 1 gnuplot の設定内容 (振幅特性の出力)

```
1 set terminal png
2 set output "1AC_Igraf.png"
3 set xrange [0:5000000]
4 set xlabel "Frequency[Hz]"
5 set ylabel "I_R2[A]"
6 plot "(1)(a).txt" using 1:2 with line
```

Listing 2 gnuplot の設定内容 (位相特性の出力)

```
1 set terminal png
2 set output "1AC_Agraf.png"
3 set xrange [0:5000000]
4 set yrange [-180:180]
5 set xlabel "Frequency[Hz]"
6 set ylabel "I_R2_Phase_difference[degree]"
7 plot "(1)(a).txt" using 1:3 with line
```

Listing 3 gnuplot の設定内容 (ステップ応答の出力)

```
1 set terminal png
2 set output "1stepgraf.png"
3 set xrange [0:5e-06]
4 set yrange [-0.0003:0.0003]
5 set xlabel "Time[second]"
6 set ylabel "I_R2[A]"
7 plot "(1)(a)step.txt" with line
```

3.2 3次の0-R型フィルタの設計と伝達特性のシミュレート

Butterworth フィルタと Chebyshev フィルタとでは、設計段階の最初の規格化低域通過フィルタの設計のみ、手順が異なる。

最初に、図3に示す3次規格化低域通過フィルタの伝達関数は、

$$T(s) = 1/F(s) = L_1 L_2 C s^3 + L_1 s^2 + (L_1 + L_2) s + 1 \quad (6)$$

である。

3.2.1 3次 Butterworth 規格化低域通過フィルタの設計

3次 Butterworth 規格化低域通過フィルタの伝達関数は、

$$T(s) = s^3 + 2s^2 + 2s + 1 \quad (7)$$

これと式(6)との右辺同士を比較して、 L_1 、 L_2 、 C の値を決めれば良い。

3.2.2 3 次 Chebyshev 規格化低域通過フィルタの設計

Chebyshev フィルタにはリップルの程度を決める値である ϵ があるが、これが 1.0 であるフィルタを設計する。3 次 Chebyshev 規格化低域通過フィルタの伝達関数を求めるために、2.3 にある法で極を求める。

$$a = (\pi/6)(2k + 1) \text{ for } k = 1, 2, 3, 4, 5, 6 \quad (8)$$

$$b = \pm(1/3)\sinh^{-1}(1) \quad (9)$$

$$(10)$$

の下で、極は

$$s = \sin a \sinh b + j \cos a \cosh b \quad (11)$$

で表される 12 個 (うち半分は重複するので実質は 6 個) のうち、複素平面上左反面にある 3 つである。この 3 つを数値的に求め、式 (6) との比較を行って L_1 、 L_2 、 C の値を決める python のコードを以下に示す。

Listing 4 極の数値解を求める

```
1 import numpy as np
2 import cmath
3 import math
4
5 def get_s(a, b):
6     return math.sin(a)*math.sinh(b) + 1j*math.cos(a)*math.cosh(b)
7
8 def get_a(k, n):
9     return (math.pi/(2.0*n))*(2.0*k+1)
10
11 def get_b_plus(n):
12     # print("%f" % np.arcsinh(1.0))
13     return (1.0/n)*np.arcsinh(1.0)
14
15 def get_b_minus(n):
16     return -(1.0/n)*np.arcsinh(1.0)
17
18 def L_1(a, b, c):
19     return (c-a/b)
20
21 def L_2(a, b, c):
22     return (a/b)
23
24 def C(a, b, c):
25     return (b/L_1(a,b,c))
26
27 klist = [0,1,2,3,4,5]
28 alist = [get_a(k,3.0) for k in klist]
29 blist = [get_b_plus(3), get_b_minus(3)]
30
31 s = [get_s(a, b) for a in alist for b in blist]
32
33 snum = [i if s[i].real < 0 else 'fizz' for i in range(12)]
34
35 for i in range(12):
36     if snum[i] != 'fizz':
37         print("snum_=%d, _s_=%f+_j_%f" % (snum[i], s[i].real, s[i].imag))
38
39 #以下は重複結果を排除するため、表示結果確認後の追記。追記後再び実行。
40 alpha = -1.0/(s[1]*s[3]*s[5])
41 beta = (s[1]+s[3]+s[5])/(s[1]*s[3]*s[5])
42 gamma = - (s[1]*s[3] + s[3]*s[5] + s[5]*s[1])/(s[1]*s[3]*s[5])
43
44 print("L1_=%f, _L2_=%f, _C1_=%f" % (L_1(alpha.real, beta.real, gamma.real), L_2(alpha.real, beta.real, gamma.real), C(alpha.real, beta.real, gamma.real)))
```

これを実行した後、最終行に出力される結果を用いる。

3.3 インピーダンススケーリングと周波数変換

3.3.1 低域通過フィルタ

遮断周波数 1.5 kHz、出力側インピーダンス $R = 50[\Omega]$ の低域通過フィルタを設計する。回路の要素の変換は表 1 に示した変数変換の下で、以下の通り行われる。

表 3 要素の置換

	インダクタンス	キャパシタンス
基準	$L = L_o$	$C = C_o$
変換後	$L = L_o R / \omega_c$	$C = C_o / \omega_c R$

但し、 ω_c が遮断角周波数

3.3.2 高域通過フィルタ

遮断周波数 1.5 kHz、出力側インピーダンス $R = 50[\Omega]$ の低域通過フィルタを設計する。回路の要素の変換は表 1 に示した変数変換の下で、以下の通り行われる。

表 4 要素の置換

	インダクタンス	キャパシタンス
基準	$L = L_o$	$C = C_o$
変換後	$C = 1 / L_o R \omega_c$	$L = R / C_o \omega_c$

但し、 ω_c が遮断角周波数

3.3.3 帯域通過フィルタ

中心周波数 1.5 kHz、出力側インピーダンス $R = 50[\Omega]$ の帯域通過フィルタを設計する。帯域幅は Butterworth フィルタが 1 kHz、Chebyshev フィルタが 500 Hz である。

回路の要素の変換は表 1 に示した変数変換の下で、以下の通り行われる。

表 5 要素の置換

	インダクタンス	キャパシタンス
基準	$L = L_o$	$C = C_o$
変換後	$L = L_o / R \omega_b$ と $C = \omega_b / L_o \omega_0^2 R$ の直列	$L = \omega_b R / \omega_0^2 C_o$ と $C = C_o / \omega_b R$ の並列

但し、 ω_0 が中心角周波数、 ω_b が帯域幅

3.4 伝達特性のシミュレート

3.4.1 周波数応答のシミュレート

振幅特性及び周波数特性のシミュレートは、1 V の正弦波に対する応答を見る。それぞれのフィルタのシミュレート条件を表 6 にまとめる。

3.4.2 ステップ応答のシミュレート

ステップ応答のシミュレートは、1 V の方形波に対する、0.01 s までの応答を見る。電源の設定を表 7 にまとめる。

表 6 シミュレートの条件（周波数応答）

フィルタ	周波数下限 [Hz]	周波数上限 [Hz]	計算頻度 *
LPF	100	5000	10000
ButterworthHPF	100	5000	10000
ChebyshevHPF	1000	30000	10000
ButterworthBPF	1000	22000	10000
ChebyshevBPF	100	10000	10000

* 周波数が 10 倍になる間に何回計算を行うか

表 7 ステップ応答解析における電源の設定

立ち上がり前電圧	0 V
立ち上がり後電圧	1 V
立ち上がり時刻	0 s
立ち上がり時間	10^{50} s
立ち下がり時間	10^{50} s
立ち上がり持続時間	1 s
周期	2 s

4. 使用器具

4.1 回路シミュレータ

シミュレータ LTSpiceIV ver.4.23

4.2 シミュレータ実行環境

PC Dell Latitude E6430

OS Windows 10 Education

CPU Intel(R) Core(TM) i7-3630QM CPU @ 2.40GHz

メモリ 16GB

4.3 プロットソフト

プロットソフト gnuplot 4.6 patchlevel 4

4.4 プロットソフト実行環境

OS Ubuntu14.04LTS

その他はシミュレータ実行環境に同じ。

5. 実験結果

5.1 回路シミュレータ LTSpiceIV での簡単な回路のシミュレートとその数値データのプロット

プロット結果はそれぞれ図 4,5,6 の通りであった。

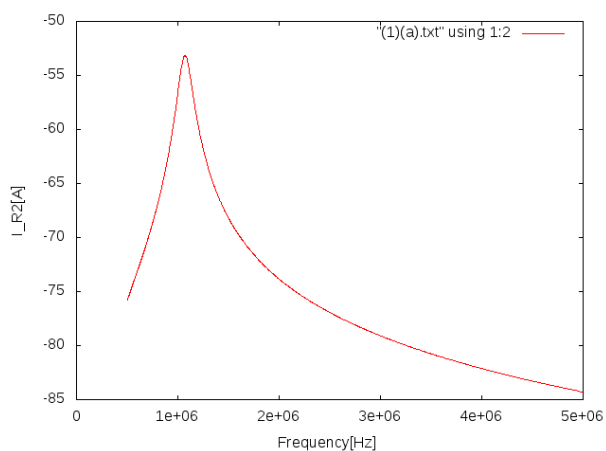


図 4 RLC 直列回路の周波数による振幅特性

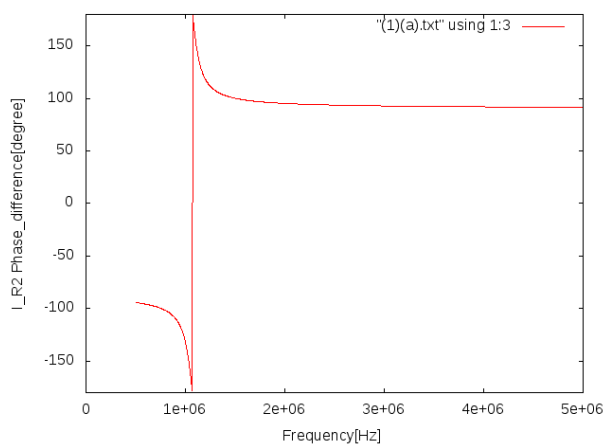


図 5 RLC 直列回路の周波数による位相特性

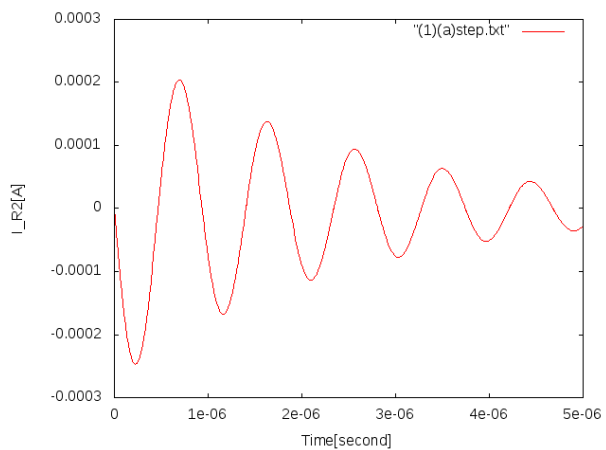


図 6 RLC 直列回路のステップ応答

5.2 Butterworth フィルタ

5.2.1 規格化低域通過フィルタの設計

設計の結果、

$$L_1 = 1.5 \quad (12)$$

$$L_2 = 0.5 \quad (13)$$

$$C = 4/3 \approx 1.333 \quad (14)$$

となった。この数値を基準として、以後の各種フィルタの設計を行った。

5.2.2 低域通過フィルタ

遮断周波数 1.5 kHz の Butterworth 低域通過フィルタは、図 7 (図中の u は μ である。以下同じ。) に示す通りとなった。そのシミュレート結果も、図 8 ~ 10 に示す。

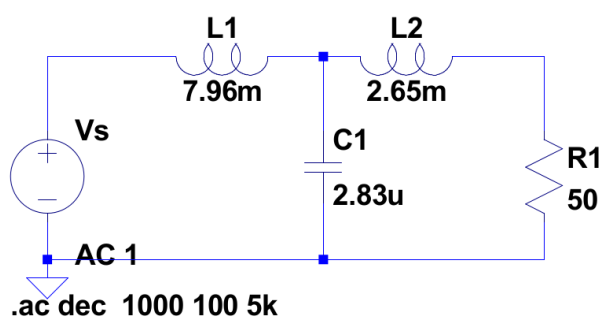


図 7 ButterworthLPF($f_c = 1500$)

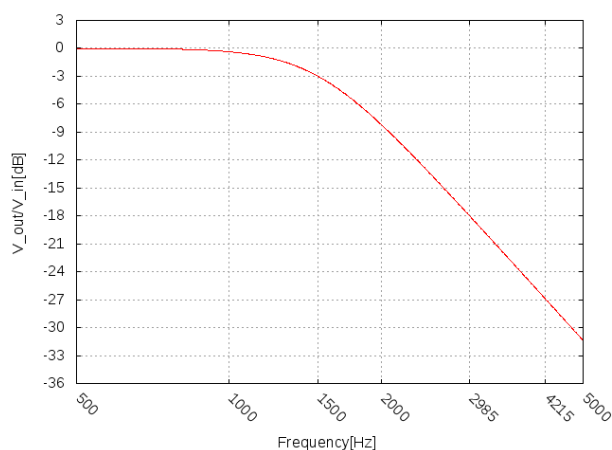


図 8 ButterworthLPF の振幅特性

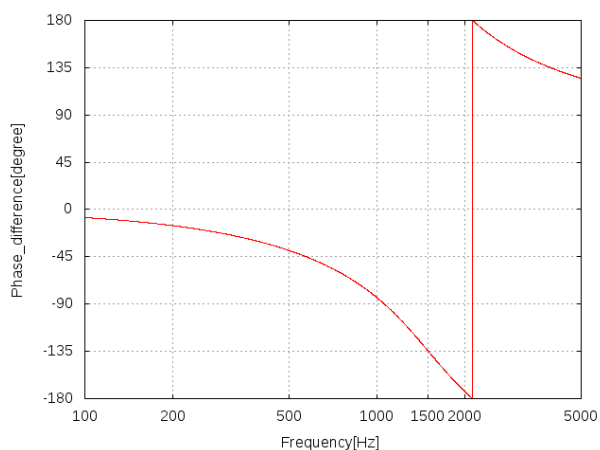


図 9 ButterworthLPF の位相特性

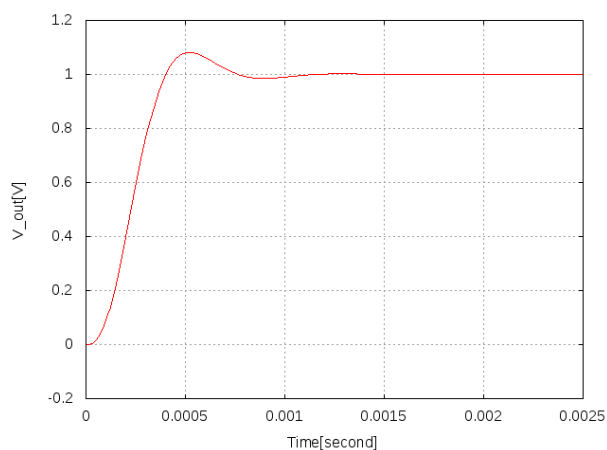


図 10 ButterworthLPF のステップ応答

5.2.3 高域通過フィルタ

遮断周波数 1.5 kHz の Butterworth 高域通過フィルタは、図 11 に示す通りとなった。そのシミュレート結果も、図 12 ~ 14 に示す。

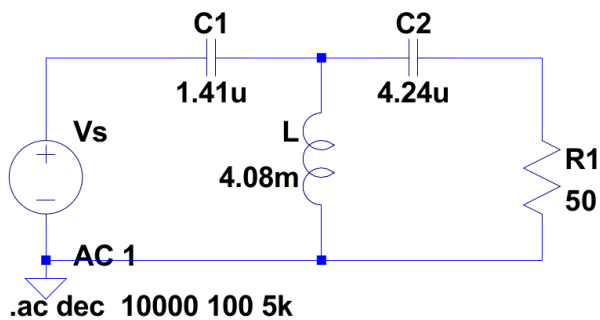


図 11 ButterworthHPF($f_c = 1500$)

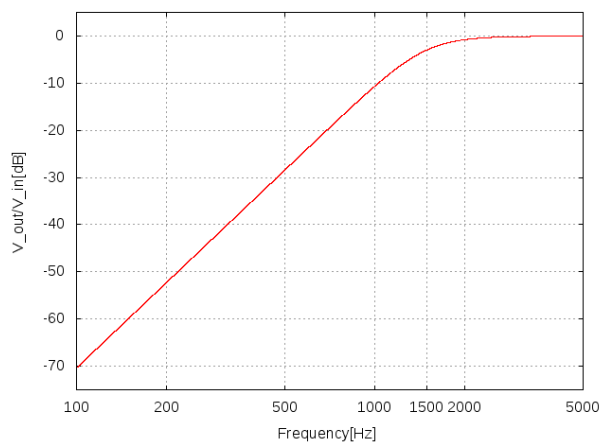


図 12 ButterworthHPF の振幅特性

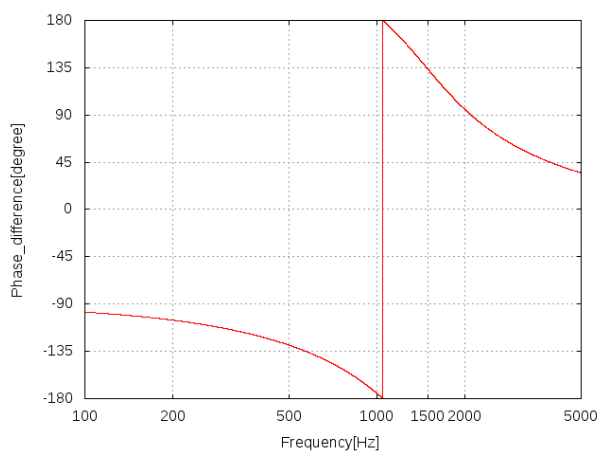


図 13 ButterworthHPF の位相特性

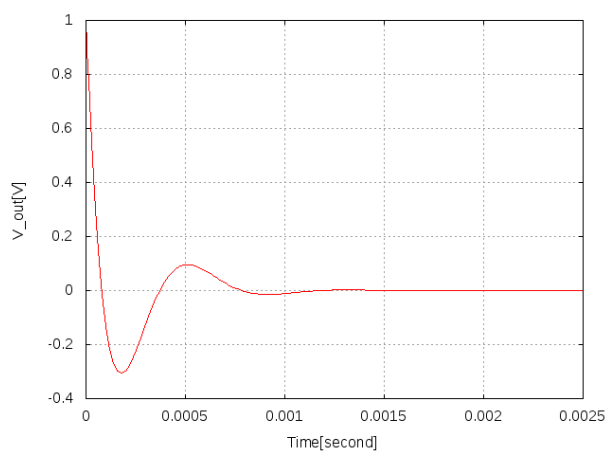


図 14 ButterworthHPF のステップ応答

5.2.4 帯域通過フィルタ

中心周波数 1.5 kHz、帯域幅 1 kHz の Butterworth 帯域通過フィルタは、図 15 に示す通りとなった。そのシミュレート結果も、図 16～18 に示す。

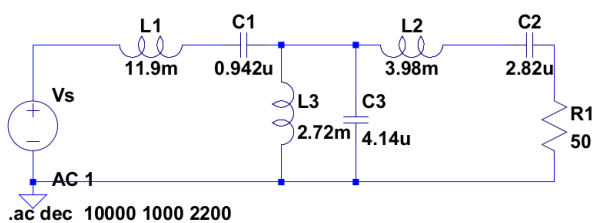


図 15 ButterworthBPF($f_0 = 1500$)

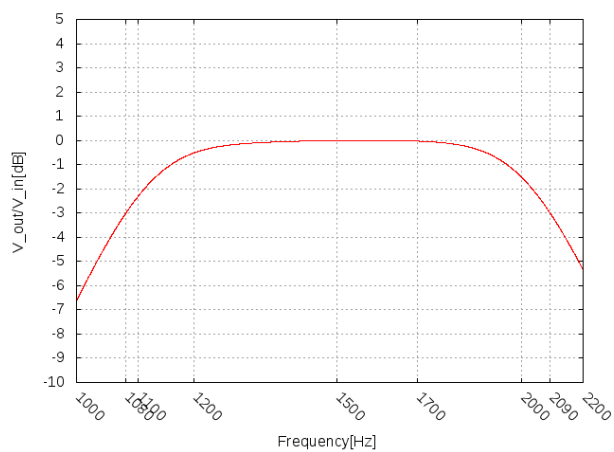


図 16 ButterworthBPF の振幅特性

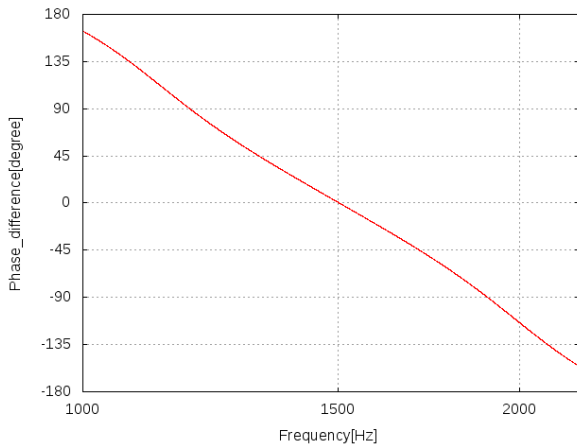


図 17 ButterworthBPF の位相特性

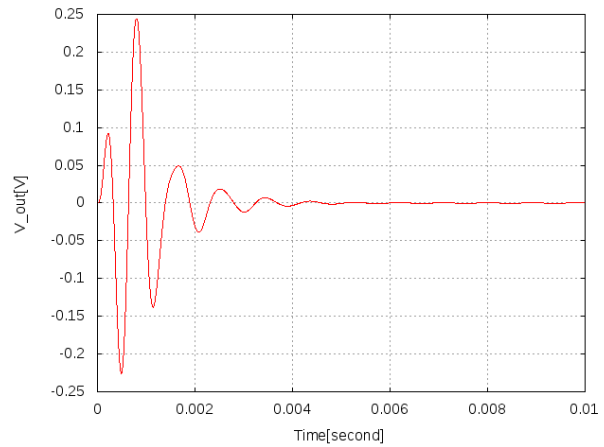


図 18 ButterworthBPF のステップ応答

5.3 Chebyshev フィルタ

5.3.1 規格化低域通過フィルタの設計

設計の結果、

$$L_1 = 2.033 \quad (15)$$

$$L_2 = 1.678 \quad (16)$$

$$C = 1.173 \quad (17)$$

となった。この数値を基準として、以後の各種フィルタの設計を行った。

5.3.2 低域通過フィルタ

遮断周波数 1.5 kHz の Chebyshev 低域通過フィルタは、図 19 に示す通りとなった。そのシミュレート結果も、図 20 ~ 22 に示す。

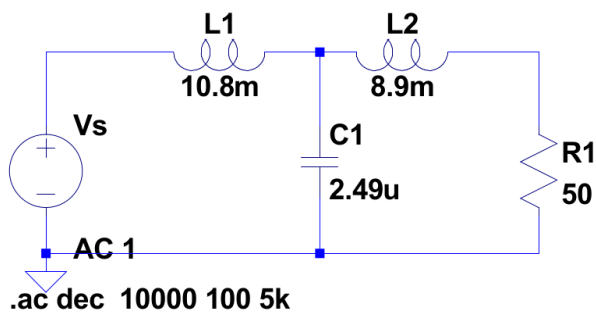


図 19 ChebyshevLPF($f_c = 1500$)

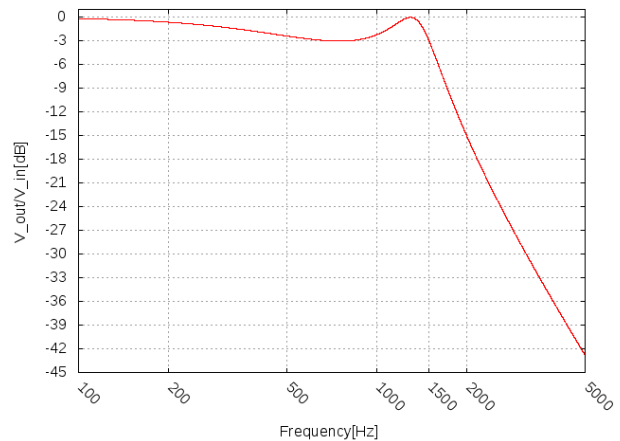


図 20 ChebyshevLPF の振幅特性

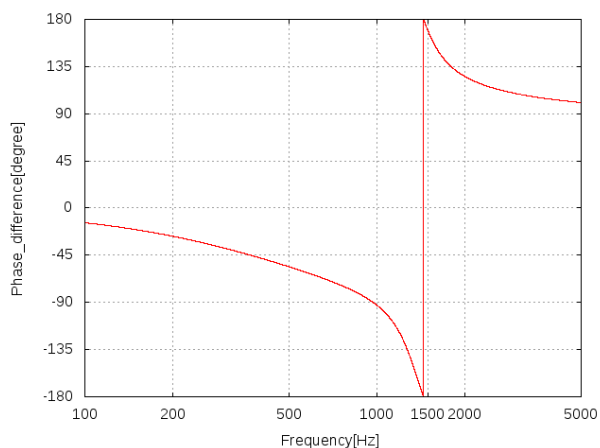


図 21 ChebyshevLPF の位相特性

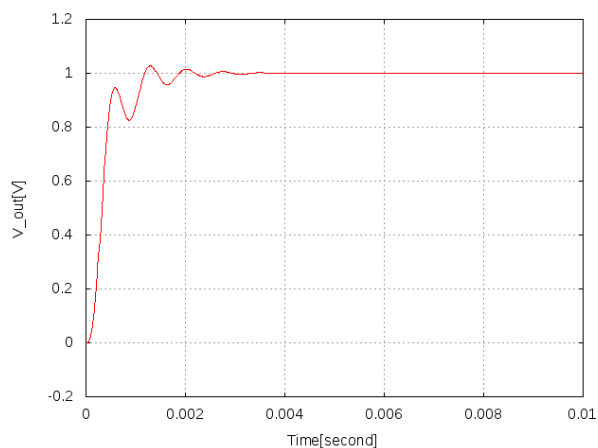


図 22 ChebyshevPF のステップ応答

5.3.3 高域通過フィルタ

遮断周波数 1.5 kHz の Chebyshev 高域通過フィルタは、図 23 に示す通りとなった。そのシミュレート結果も、図 24 ~ 26 に示す。

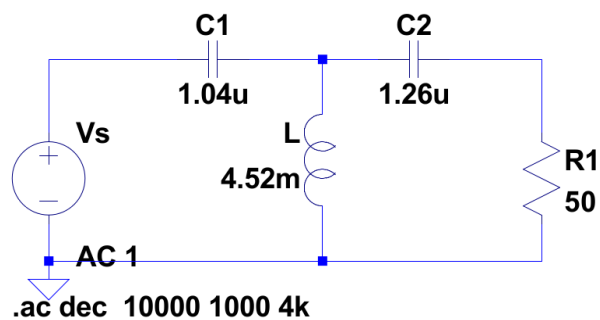


図 23 ChebyshevHPF($f_c = 1500$)

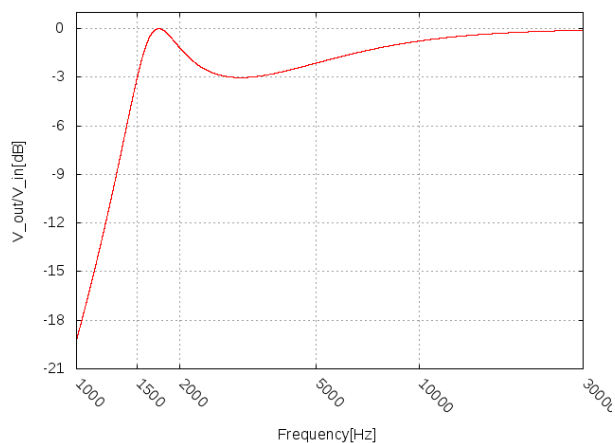


図 24 ChebyshevHPF の振幅特性

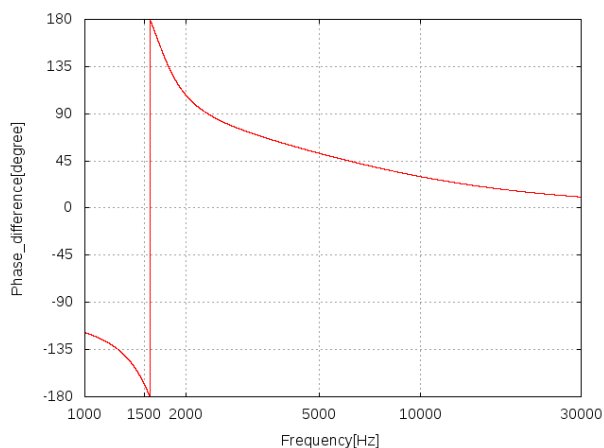


図 25 ChebyshevHPF の位相特性

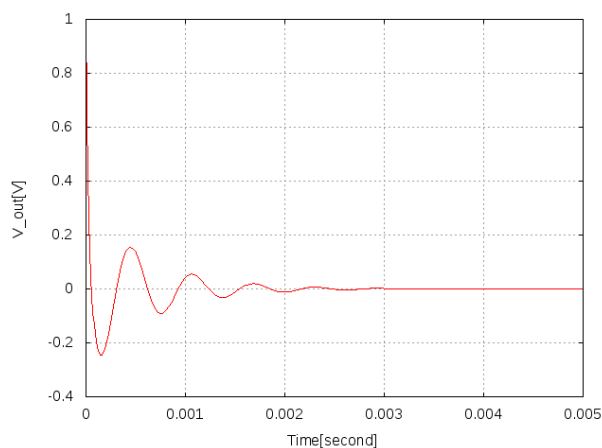


図 26 ChebyshevHPF のステップ応答

5.3.4 帯域通過フィルタ

中心周波数 1.5 kHz、帯域幅 500 Hz の Chebyshev 帯域通過フィルタは、図 27 に示す通りとなった。そのシミュレート結果も、図 28～30 に示す。

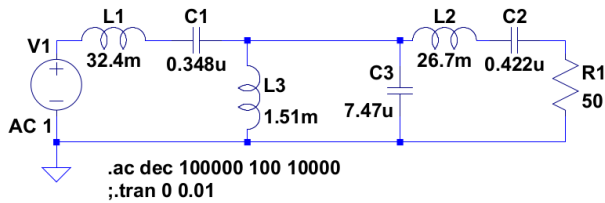


図 27 ChebyshevBPF($f_0 = 1500$)

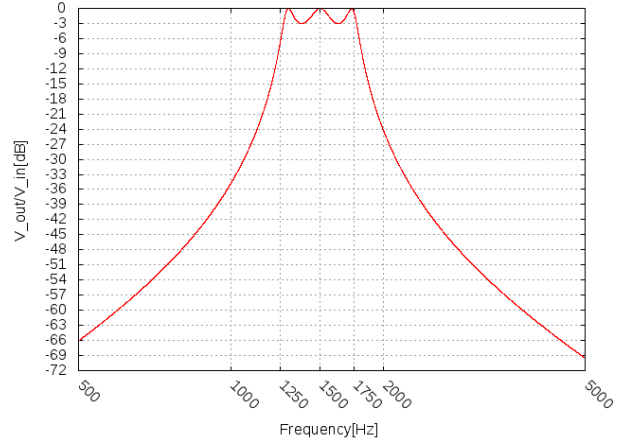


図 28 ChebyshevBPF の振幅特性

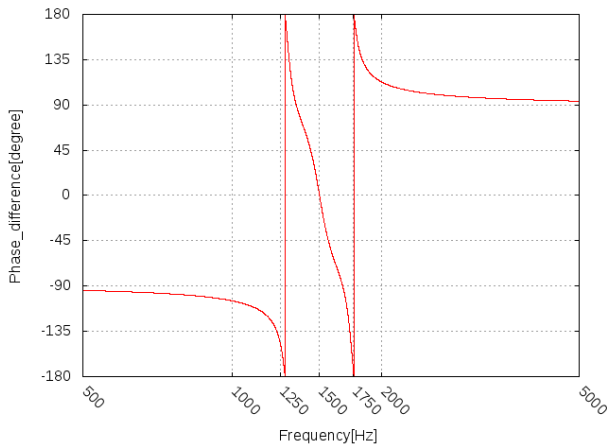


図 29 ChebyshevBPF の位相特性

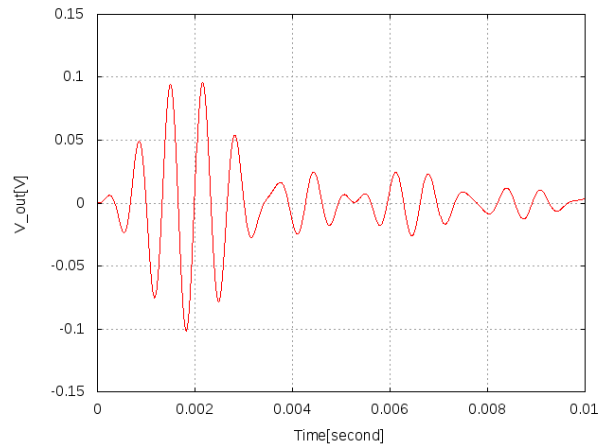


図 30 ChebyshevBPF のステップ応答

6. 考察

6.1 3次低域通過フィルタの伝達特性

3次フィルタとしての特徴は、主として振幅特性に現れている。これは、Butterworth 及び Chebyshev フィルタが主たる設計思想として通過域と減衰域に注目していることが大きな原因の一つであろうと思われる。

Butterworth フィルタでは、遮断域に n 次フィルタの特性が表れる。図 8 を見ると、2985 Hz から 4215 Hz まででおよそ -9 dB の損失が見られる。

$$\frac{-9}{\log_{10} 4215 - \log_{10} 2985} \approx -60.059 [\text{dB}] \quad (18)$$

より、周波数の常用対数が 1 増加する間に $20 \cdot 3 = 60 \text{ dB}$ の損失が発生するという理論値とほぼ等しい。

一方無極形通過域 Chebyshev フィルタでは、次数が最も顕著に表れるのは通過域でのリップルの回数である。 n 次の低域通過フィルタでは、少なくとも $n/2$ 回のリップルが発生する。図 20 を見れば、2 回のリップルが発生していることが

わかる。(或いは、BPFのリブルの回数を数えるのが妥当だろう[次節参照]。)

6.2 3次高域通過及び帯域通過フィルタの周波数特性

高域通過フィルタについては、前節にて述べた低域通過フィルタと大きく変わらない。Butterworth フィルタは遮断域での傾きが 60 dB/decade に、Chebyshev フィルタでは通過域で 2 回のリブルが発生している。

一方の帯域通過フィルタだが、振幅特性が左右対称なフィルタであるだけに、Chebyshev フィルタでは、リブルの数と次数の関係をより明確に読み取ることが出来る。図 28 では通過域に明らかに 3 つの山が立っている。

6.3 Butterworth 帯域通過フィルタでの誤差

図 16 を見ると、-3 dB を取る周波数は、およそ 1080 Hz と 2090 Hz で、その中心も 1585 Hz と、理論値から 6 % 弱の誤差が出ている。これは、周波数変換を行う際に電卓を用いて、幾らか数値を丸めているためであろうと考えられる。数値を有効数字三桁に丸めながら演算を行ったので、相対的な丸め誤差の最小単位は、 $u = 1/2 \cdot 10^{-2}$ である。構成要素の置き換えの計算だけでも二項演算子を 4 回使うから、その誤差は $\sqrt{4}u = 10^{-2}$ 程度発生する。そこから更に周波数と振幅の関係に記述しなおされると、ひと桁台%の誤差は妥当なように思われる。

7. 参考資料

1. 東京大学工学部 電子情報工学科・電気電子工学科 (2016) 『電気電子情報第一 (前期) 実験 テキスト』
2. R.M.Fano & A.W.Lawson (1948) 菅原英彦訳 (2009) 『マイクロ波フィルタの理論』丸善ブラネット
3. 古賀利郎 (1978) 『伝送回路』コロナ社
4. 幸谷智紀 (2005) 『数値計算における誤差について -数値微分を例に-』 http://na-inet.jp/na/na_error_diff.pdf