多重帰還型LPFの現実的な定数を求める方法

Ayumi's Lab.

2007 年 4 月 6 日
Revised 2009 年 8 月 2 日

多重帰還型フィルタは，歪みが少ないため，オーディオ用途に適しています．多重帰還型 LPF は，3 本の抵抗と 2 個のコンデンサで遮断角周波数 ω_0，Q，通過域のゲイン K が決定されますが，一般に，ゲインを 1 とし，3 本の抵抗を同じ値とし，Q の値によって 2 個のコンデンサの容量の比を定めます．しかし，一般に入手可能なコンデンサの種類は少なく，E12 系列が入手できることはまれで，E6 系列で設計できるとありがたいものです．望む Q の値を実現するには，複数のコンデンサを並列にすることが多いようです．コンデンサは 1 個あたりの価格が抵抗に比べて高いため，コンデンサの数を最小限にし，抵抗を複数使うほうがアマチュアにとっては現実的でしょう．E24 系列の抵抗を入手するのは容易なので，複数の抵抗を直列や並列につなげば，必要とされる値をかなり正確に得ることができます．ここでは，2 次のフィルタに 2 個のコンデンサのみを使用して，任意の ω_0，Q，K を持つフィルタの定数を求める方法を紹介します．

1　多重帰還型 LPF の特性

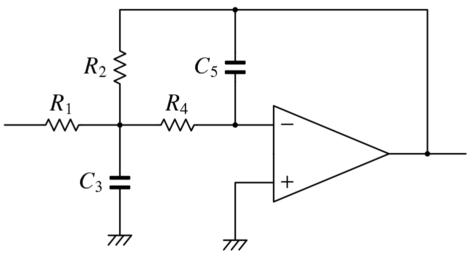

図 1: 多重帰還型 LPF の回路

図 1の回路の多重帰還型 LPF の伝達特性と ω_0, Q の値は,

$$T(s) = -\frac{R_2/R_1}{s^2C_3C_5R_2R_4 + sC_5(\frac{R_2R_4}{R_1} + R_2 + R_4) + 1} \quad (1)$$

$$\omega_0 = \frac{1}{\sqrt{C_3C_5R_2R_4}} \quad (2)$$

$$Q = \frac{\sqrt{C_3C_5R_2R_4}}{C_5(\frac{R_2R_4}{R_1} + R_2 + R_4)} \quad (3)$$

となります．通過域のゲインを $K = R_2/R_1$ とおけば,

$$T(s) = -\frac{K}{s^2C_3C_5R_2R_4 + sC_5(\frac{R_2R_4}{R_1} + R_2 + R_4) + 1}$$

$$Q = \frac{\sqrt{C_3 C_5 R_2 R_4}}{C_5[R_2 + (1+K)R_4]}$$

となります．

ここで，コンデンサおよび抵抗の比について，

$$\begin{aligned} R_2 &= m_r R \\ R_4 &= \frac{R}{m_r} \\ C_3 &= m_c C \\ C_5 &= \frac{C}{m_c} \end{aligned}$$

とおくと，

$$\omega_0 = \frac{1}{CR} \tag{4}$$

$$\begin{aligned} Q &= \frac{CR}{\frac{C}{m_c}[m_r R + (1+K)\frac{R}{m_r}]} \\ &= \frac{m_c}{m_r + \frac{1+K}{m_r}} \end{aligned} \tag{5}$$

となります．

さまざまな K の値について，m_r を変えながら Q の値を描くと，図 2のようになります．図 2は，

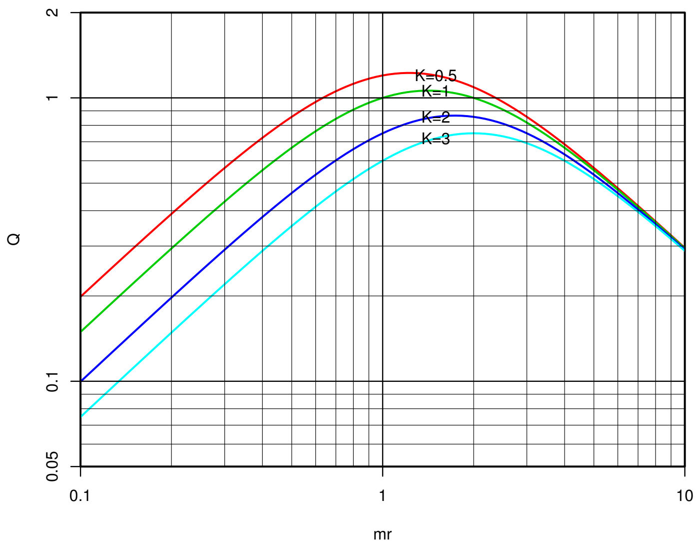

図 2: m_r に対する Q の値．($m_c = 3$ の場合)

$m_c = 3$ の場合ですが，Q は m_c に比例するので，異なった m_c の値に関するグラフは，図 2の形状をそのまま上下にずらしたものになります．

常識的な $R_1 = R_2 = R_4$ の場合，すなわち $m_r = 1$ における値よりも大きな Q が得られる m_r が存在することがわかります．

この最大の Q となる m_r を利用すれば，一定の Q に対して m_c の値を小さく済ませることができます．指定された Q に対して m_c が最小となるのは，式 (5) を m_c について解いた式の，m_r に関す

る導関数値が 0 となる m_r のときです.

$$\begin{aligned} m_c &= Q[m_r + (1+K)\frac{1}{m_r}] \\ \frac{dm_c}{dm_r} &= Q[1-(1+K)m_r^{-2}] = 0 \\ m_r &= \sqrt{1+K} \end{aligned}$$

したがって,

$$\min m_c = Q(\sqrt{1+K} + \frac{1+K}{\sqrt{1+K}}) \tag{6}$$

$$= 2Q\sqrt{1+K} \tag{7}$$

これ以上の m_c が確保されていれば，m_r を調整することによって，Q の値を目的の値に下げることができます．m_r の値は，式 (5) より，

$$\begin{aligned} m_r + \frac{1+K}{m_r} &= \frac{m_c}{Q} \\ {m_r}^2 - \frac{m_c}{Q}m_r + (1+K) &= 0 \\ m_r &= \frac{\frac{m_c}{Q} \pm \sqrt{\frac{m_c^2}{Q^2} - 4(1+K)}}{2} \end{aligned}$$

ここで，図 2より，m_r は 2 つの解を持ちますが，Q の極大値が必ず $m_r \geq 1$ にあるので，小さい方の解のほうが $m_r = 1$ に近くなります．抵抗の比は小さいほうが良いので，こちらの解を選びます．

2　サレンキー型 LPF の特性

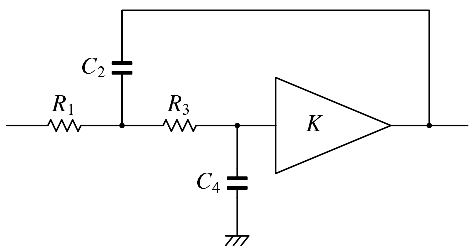

図 3: サレンキー型 LPF の回路

図 3の回路のサレンキー型 LPF の伝達特性と ω_0, Q の値は，アンプのゲインを K とすると，

$$T(s) = \frac{K}{s^2C_2C_4R_1R_3 + s[C_4(R_1+R_3) + (1-K)C_2R_1] + 1} \tag{8}$$

$$\omega_0 = \frac{1}{\sqrt{C_2C_4R_1R_3}} \tag{9}$$

$$Q = \frac{\sqrt{C_2C_4R_1R_3}}{C_4(R_1+R_3) + (1-K)C_2R_1} \tag{10}$$

となります．しかし，サレンキー型 LPF で K の値を大きくすると，Q の値が敏感になりすぎるので，ここでは $K = 1$ の場合のみを考えます．

ここで，コンデンサおよび抵抗の比について，

$$
\begin{aligned}
R_1 &= m_r R \\
R_3 &= \frac{R}{m_r} \\
C_2 &= m_c C \\
C_4 &= \frac{C}{m_c}
\end{aligned}
$$

とおくと，

$$
\begin{aligned}
\omega_0 &= \frac{1}{CR} && (11) \\
Q &= \frac{CR}{\frac{C}{m_c}(m_r + \frac{1}{m_r})R} \\
&= \frac{m_c}{m_r + \frac{1}{m_r}} && (12)
\end{aligned}
$$

となります．

Q は $m_r = 1$ のときに最大となります．指定された Q に対する m_c の最小値は，

$$\min m_c = 2Q \tag{13}$$

で，これ以上の m_c が確保されていれば，m_r を調整することによって，Q の値を目的の値に下げることができます．m_r の値は，式 (12) より，

$$
\begin{aligned}
m_r + \frac{1}{m_r} &= \frac{m_c}{Q} \\
{m_r}^2 - \frac{m_c}{Q} m_r + 1 &= 0 \\
m_r &= \frac{\frac{m_c}{Q} \pm \sqrt{\frac{m_c^2}{Q^2} - 4}}{2}
\end{aligned}
$$

ここで，m_r は 2 つの解を持ちますが，一方の解は他方の解の逆数となります．R_1 が大きいほうが前段の負荷が軽くなりますので，ここでは $m_r \geq 1$ の解を選びます．

3 定数を求めるプログラム

フィルタの定数を求める R のプログラムを，以下に示します．

lpfparam.r

```
‘lpfparam‘ <- function (w0, Q, K=1, type="SK", Rrange=c(5e3, 100e3), Cseries=E6)
{
    # LPF の定数を求める
    # w0: 遮断角周波数
    # Q: Q
    # K: 通過域のゲイン（サレンキー型の場合は無視される）
    # type: "SK"（サレンキー型）または "MF"（多重帰還型）
    # Rrange: 使用する抵抗値の範囲（これよりいくらか外れることもある）
    # Cseries: 使用するコンデンサの系列の値
```

```
    # 返値:
    #    type == "SK" の場合、6 列の行列
    #    各列の値は、R1, C2, R3, C4, mr, mc
    #    type == "MF" の場合、7 列の行列
    #    各列の値は、R1, R2, C3, R4, C5, mr, mc

    if (type == "SK")
        K <- 0
    mc.min <- 2 * Q * sqrt(1 + K)
    cat("mc.min=", mc.min, "\n", sep="")
    z <- combr(mc.min^2, series=Cseries, func="/", mult=0:3, n=100, comp=">")
    if (nrow(z) == 0)
        stop("指定された仕様は満たせません．Q や K を小さくしてください．")
    res <- NULL
    for (i in seq(1:nrow(z))) {
        Cm <- sqrt(z[i, 1] * z[i, 2])       # コンデンサの値の仮数部
        mc <- sqrt(z[i, 1] / z[i, 2])       # コンデンサの、基準値からの比率
        Ce <- log10(1 / (w0 * Rrange * Cm)) # 抵抗値が範囲に収まる
        Ce <- ceiling(Ce[2]):floor(Ce[1])   # コンデンサの指数を求める
        C <- Cm * 10^Ce                     # コンデンサの値
        R <- 1 / (w0 * C)                   # 対応する抵抗の値

        # 指定された Q となる抵抗の比を求める
        b <- -mc / Q                        # 2 次方程式の一次項
        c <- 1 + K                          # 2 次方程式の定数項
        if (type == "SK") {
            mr <- (-b + sqrt(b^2 - 4 * c)) / 2  # 2 次方程式の解
            for (j in seq(along=C)) {
                R1 <- R[j] * mr
                R3 <- R[j] / mr
                C2 <- C[j] * mc
                C4 <- C[j] / mc
                res1 <- c(R1, C2, R3, C4, mr, mc)
                if (is.null(res))
                    res <- matrix(res1, nrow=1,
                        dimnames=list(NULL, c("R1", "C2", "R3", "C4", "mr", "mc")))
                else if (match(signif(C2, 3), signif(res[, 2], 3), nomatch=0) == 0)
                    res <- rbind(res, res1)
            }
        } else if (type == "MF") {
            mr <- (-b - sqrt(b^2 - 4 * c)) / 2  # 2 次方程式の解
            for (j in seq(along=C)) {
                R2 <- R[j] * mr
                R4 <- R[j] / mr
                R1 <- R2 / K
                C3 <- C[j] * mc
                C5 <- C[j] / mc
                res1 <- c(R1, R2, C3, R4, C5, mr, mc)
                if (is.null(res))
                    res <- matrix(res1, nrow=1,
                        dimnames=list(NULL, c("R1", "R2", "C3", "R4", "C5", "mr", "mc")))
                else if (match(signif(C3, 3), signif(res[, 3], 3), nomatch=0) == 0)
                    res <- rbind(res, res1)
            }
        }
    }
    dimnames(res)[[1]] <- 1:nrow(res)
    res
}
```

遮断周波数 `w0` は，角周波数で与えることに気を付けてください．通過域利得 `K` は，省略時の値として 1 が与えられているので，通常は指定する必要はありません．フィルタの種類 `type` には，サレンキー型の場合は `"SK"` を，多重帰還型の場合は `"MF"` を指定します．使用する抵抗値の範囲 `Rrange` は，長さが 2 のベクトルで与えます．基準となる抵抗の値 (R_2 と R_4 の幾何平均) がこの範囲になるようなコンデンサの値を使います．結果として，抵抗値がこの範囲を外れることもあります．`Cseries` には，コンデンサの系列の仮数部の値のベクトルを指定します．`E6`, `E12`, `E24` という

ベクトルをあらかじめ用意してあります.

結果は 6 列 (サレンキー型の場合) または 7 列 (多重帰還型の場合) の行列で，各行が 1 つの定数の組になっています．コンデンサの比 mc (6 列目または 7 列目) が小さい順にならんでいます．最初のコンデンサの値が同じとなる定数の組は，最も mc が小さいもの以外は捨ててあります.

遮断周波数 1 kHz で Q が 1 の多重帰還型フィルタのパラメータを求めた例を，以下に示します.

```
> lpfparam(w0=2*pi*1e3, Q=1, type="MF")
mc.min=2.828427
          R1        R2      C3        R4      C5        mr       mc
1  19995.157 19995.157 2.2e-08 26174.000 2.2e-09 0.8740320 3.162278
2  43989.344 43989.344 1.0e-08 57582.799 1.0e-09 0.8740320 3.162278
3   4398.934  4398.934 1.0e-07  5758.280 1.0e-08 0.8740320 3.162278
4  29326.230 29326.230 1.5e-08 38388.533 1.5e-09 0.8740320 3.162278
5  13330.104 13330.104 3.3e-08 17449.333 3.3e-09 0.8740320 3.162278
6   9359.435  9359.435 4.7e-08 12251.659 4.7e-09 0.8740320 3.162278
7  64690.212 64690.212 6.8e-09 84680.587 6.8e-10 0.8740320 3.162278
8   6469.021  6469.021 6.8e-08  8468.059 6.8e-09 0.8740320 3.162278
9   2274.893  2274.893 1.5e-07 15793.931 4.7e-09 0.3795206 5.649327
10  1493.087  1493.087 2.2e-07 23367.842 3.3e-09 0.2527745 8.164966
```

最初の定数の組を使えば，$R_1 = 20\,\mathrm{k\Omega}$, $R_2 = 20\,\mathrm{k\Omega}$, $C_3 = 0.022\,\mu\mathrm{F}$, $R_4 = 24 + 2.2\,\mathrm{k\Omega}$, $C_5 = 2200\,\mathrm{pF}$ とすればよいことがわかります.