數位訊號處理、系統設計與模擬 P.1

系統設計名稱：可調分數階微積分器之設計

引用論文相關內容：

Signal Processing 89 (2009) 320–327

Contents lists available at ScienceDirect

Signal Processing

journal homepage: www.elsevier.com/locate/sigpro

ELSEVIER

SIGNAL PROCESSING

An iterative method for the design of variable fractional-order FIR differintegrators

Jong-Jy Shyu [a,*], Soo-Chang Pei [b], Cheng-Han Chan [c]

[a] Department of Electrical Engineering, National University of Kaohsiung, No. 700, Kaohsiung University Rd., Nan-Tzu District, Kaohsiung 811, Taiwan, ROC
[b] Department of Electrical Engineering, National Taiwan University, Taiwan, ROC
[c] Department of Computer and Communication Engineering, National Kaohsiung First University of Science and Technology, Taiwan, ROC

2. Problem formulation

For designing a VFO differintegrator, the desired response is given by

$$D(\omega,p) = e^{-jI\omega}(j\omega)^p, \quad p_s \leqslant p \leqslant p_f, \quad \omega_s \leqslant |\omega| \leqslant \omega_f, \tag{1}$$

where I is a prescribed delay and p is the variable order of the designed differintegrator. If a pure VFO differentiator is designed, $p_s \geqslant 0$ and $\omega_s \geqslant 0$, while $p_f \leqslant 0$ and $\omega_s > 0$ for designing a pure VFO integrator, and $p_s < 0 < p_f$, $\omega_s > 0$ for the VFO differintegrator design. Let

$$\begin{aligned}\hat{D}(\omega,p) &= (j\omega)^p \\ &= |\omega|^p\left[\cos\left(\frac{p\pi}{2}\right) + j\,\mathrm{sgn}(\omega)\sin\left(\frac{p\pi}{2}\right)\right],\end{aligned} \tag{2}$$

where $\mathrm{sgn}(\cdot)$ is a sign function, then Eq. (1) can be represented by

$$D(\omega,p) = e^{-jI\omega}\hat{D}(\omega,p). \tag{3}$$

For approximating the desired response, the used transfer function is characterized by

$$H(z,p) = \sum_{n=0}^{N} h_n(p) z^{-n}, \tag{4}$$

where the coefficients $h_n(p)$ are expressed as the polynomials of p

$$h_n(p) = \sum_{m=0}^{M} h(n,m) p^m, \tag{5}$$

hence Eq. (4) becomes

$$H(z,p) = \sum_{n=0}^{N}\sum_{m=0}^{M} h(n,m) p^m z^{-n}. \tag{6}$$

For simplicity, only even N is used in this section and the case for odd N will be given in Section 3. According to the symmetric and antisymmetric characteristics for the real part and imaginary part of (2), respectively, with respective to ω, the coefficients $h(n,m)$ in (6) are divided into even part and odd part by

$$h(n,m) = h_e(n,m) + h_o(n,m), \tag{7}$$

where

$$h_e\left(\frac{N}{2}+n,m\right) = \frac{1}{2}\left[h\left(\frac{N}{2}+n,m\right) + h\left(\frac{N}{2}-n,m\right)\right], \quad -\frac{N}{2} \leqslant n \leqslant \frac{N}{2}, \ 0 \leqslant m \leqslant M \tag{8a}$$

and

$$h_o\left(\frac{N}{2}+n,m\right) = \frac{1}{2}\left[h\left(\frac{N}{2}+n,m\right) - h\left(\frac{N}{2}-n,m\right)\right], \quad -\frac{N}{2} \leqslant n \leqslant \frac{N}{2}, \ 0 \leqslant m \leqslant M. \tag{8b}$$

So, the frequency response of the designed filter can be formulated into

$$\begin{aligned}H(e^{j\omega},p) &= e^{-j(N/2)\omega}\left[\sum_{n=0}^{N/2}\sum_{m=0}^{M} a(n,m)p^m\cos(n\omega) + j\sum_{n=1}^{N/2}\sum_{m=0}^{M} b(n,m)p^m\sin(n\omega)\right] \\ &= e^{-j(N/2)\omega}\hat{H}(\omega,p),\end{aligned} \tag{9}$$

where

$$a(n,m) = \begin{cases} h_e\left(\frac{N}{2},m\right), & n=0, \ 0 \leqslant m \leqslant M, \\ 2h_e\left(\frac{N}{2}-n,m\right), & 1 \leqslant n \leqslant \frac{N}{2}, \ 0 \leqslant m \leqslant M, \end{cases} \tag{10a}$$

$$b(n,m) = 2h_o\left(\frac{N}{2}-n,m\right), \quad 1 \leqslant n \leqslant \frac{N}{2}, \ 0 \leqslant m \leqslant M \tag{10b}$$

and

$$\hat{H}(\omega,p) = \sum_{n=0}^{N/2}\sum_{m=0}^{M} a(n,m)p^m\cos(n\omega) + j\sum_{n=1}^{N/2}\sum_{m=0}^{M} b(n,m)p^m\sin(n\omega). \tag{11}$$

Obviously, $I = N/2$ in (1) and (3).

Let $\mathbf{A}$ and $\mathbf{B}$ be $(N/2+1)\times(M+1)$ and $N/2\times(M+1)$ matrices defined by

$$\mathbf{A} = \left[a(n,m), \ 0 \leqslant n \leqslant \frac{N}{2}, \ 0 \leqslant m \leqslant M\right] \tag{12a}$$

and

$$\mathbf{B} = \left[b(n,m), \ 1 \leqslant n \leqslant \frac{N}{2}, \ 0 \leqslant m \leqslant M\right], \tag{12b}$$

respectively; the following objective error function is used in the paper:

$$\begin{aligned}e(\mathbf{A},\mathbf{B}) &= \sum_{i=0}^{K_\omega}\sum_{l=0}^{K_p} W(\omega_i)|D(\omega_i,p_l) - H(e^{j\omega_i},p_l)|^2 \\ &= \sum_{i=0}^{K_\omega}\sum_{l=0}^{K_p} W(\omega_i)|\hat{D}(\omega_i,p_l) - \hat{H}(\omega_i,p_l)|^2, \\ \omega_i &= \omega_s + \frac{i(\omega_f-\omega_s)}{K_\omega}, \quad p_l = p_s + \frac{l(p_f-p_s)}{K_p},\end{aligned} \tag{13}$$

where a $(K_\omega+1)\times(K_p+1)$ grid is chosen for the error evaluation, and $W(\omega)$ is a positive weighting function. In the paper, $K_\omega = K_p = 200$ is used. By Pythagorean law,

$$
\begin{aligned}
e(\mathbf{A},\mathbf{B}) &= \sum_{i=0}^{K_\omega}\sum_{l=0}^{K_p} W(\omega_i)\left[\omega_i^{p_l}\cos\left(\frac{p_l\pi}{2}\right) - \sum_{n=0}^{N/2}\sum_{m=0}^{M} a(n,m)p_l^m\cos(n\omega_i)\right]^2 \\
&+ \sum_{i=0}^{K_\omega}\sum_{l=0}^{K_p} W(\omega_i)\left[\omega_i^{p_l}\sin\left(\frac{p_l\pi}{2}\right) - \sum_{n=1}^{N/2}\sum_{m=0}^{M} b(n,m)p_l^m\cdot \right. \\
&\left. \times \sin(n\omega_i)\right]^2. \qquad (14)
\end{aligned}
$$

Eq. (14) can be expressed in matrix form as

$$
\begin{aligned}
e(\mathbf{A},\mathbf{B}) &= \mathrm{tr}[(\mathbf{D_A} - \mathbf{CAP}^\mathrm{T})^\mathrm{T}(\mathbf{D_A} - \mathbf{CAP}^\mathrm{T})] \\
&+ \mathrm{tr}[(\mathbf{D_B} - \mathbf{SBP}^\mathrm{T})^\mathrm{T}(\mathbf{D_B} - \mathbf{SBP}^\mathrm{T})] \\
&= e(\mathbf{A}) + e(\mathbf{B}), \qquad (15)
\end{aligned}
$$

where tr(·) denotes a trace operator, the superscript T denotes a transpose operator,

$$\mathbf{D_A} = \left[W^{1/2}(\omega_i)\omega_i^{p_l}\cos\left(\frac{p_l\pi}{2}\right),\ 0\leqslant i\leqslant K_\omega,\ 0\leqslant l\leqslant K_p\right], \qquad (16a)$$

$$\mathbf{D_B} = \left[W^{1/2}(\omega_i)\omega_i^{p_l}\sin\left(\frac{p_l\pi}{2}\right),\ 0\leqslant i\leqslant K_\omega,\ 0\leqslant l\leqslant K_p\right], \qquad (16b)$$

$$\mathbf{C} = \left[W^{1/2}(\omega_i)\cos(n\omega_i),\ 0\leqslant i\leqslant K_\omega,\ 0\leqslant n\leqslant \frac{N}{2}\right], \qquad (16c)$$

$$\mathbf{S} = \left[W^{1/2}(\omega_i)\sin(n\omega_i),\ 0\leqslant i\leqslant K_\omega,\ 1\leqslant n\leqslant \frac{N}{2}\right], \qquad (16d)$$

$$\mathbf{P} = [p_l^m,\ 0\leqslant l\leqslant K_p,\ 0\leqslant m\leqslant M] \qquad (16e)$$

and

$$
\begin{aligned}
e(\mathbf{A}) &= \mathrm{tr}[(\mathbf{D_A} - \mathbf{CAP}^\mathrm{T})^\mathrm{T}(\mathbf{D_A} - \mathbf{CAP}^\mathrm{T})] \\
&= \mathrm{tr}[\mathbf{D_A^\mathrm{T}D_A} - \mathbf{D_A^\mathrm{T}CAP}^\mathrm{T} - (\mathbf{CAP}^\mathrm{T})^\mathrm{T}\mathbf{D_A} + (\mathbf{CAP}^\mathrm{T})^\mathrm{T}(\mathbf{CAP}^\mathrm{T})], \qquad (17)
\end{aligned}
$$

$$
\begin{aligned}
e(\mathbf{B}) &= \mathrm{tr}[(\mathbf{D_B} - \mathbf{SBP}^\mathrm{T})^\mathrm{T}(\mathbf{D_B} - \mathbf{SBP}^\mathrm{T})] \\
&= \mathrm{tr}[\mathbf{D_B^\mathrm{T}D_B} - \mathbf{D_B^\mathrm{T}SBP}^\mathrm{T} - (\mathbf{SBP}^\mathrm{T})^\mathrm{T}\mathbf{D_B} + (\mathbf{SBP}^\mathrm{T})^\mathrm{T}(\mathbf{SBP}^\mathrm{T})]. \qquad (18)
\end{aligned}
$$

Differentiating $e(\mathbf{A},\mathbf{B})$ with respect to $\mathbf{A}$ [35],

$$
\begin{aligned}
\frac{\partial e(\mathbf{A},\mathbf{B})}{\partial \mathbf{A}} = \frac{\partial e(\mathbf{A})}{\partial \mathbf{A}} &= -(\mathbf{D_A^\mathrm{T}C})^\mathrm{T}(\mathbf{P}^\mathrm{T})^\mathrm{T} - \mathbf{C}^\mathrm{T}\mathbf{D_A}\mathbf{P} \\
&+ (\mathbf{PA}^\mathrm{T}\mathbf{C}^\mathrm{T}\mathbf{C})^\mathrm{T}(\mathbf{P}^\mathrm{T})^\mathrm{T} + \mathbf{C}^\mathrm{T}\mathbf{CAP}^\mathrm{T}\mathbf{P}, \qquad (19)
\end{aligned}
$$

which is then set to zero, and the coefficient matrix $\mathbf{A}$ can be obtained as

$$\mathbf{A} = (\mathbf{C}^\mathrm{T}\mathbf{C})^{-1}\mathbf{C}^\mathrm{T}\mathbf{D_A}\mathbf{P}(\mathbf{P}^\mathrm{T}\mathbf{P})^{-1}. \qquad (20)$$

Similarly, the coefficient matrix $\mathbf{B}$ can be achieved by differentiating $e(\mathbf{A},\mathbf{B})$ with respect to $\mathbf{B}$ and setting the result to zero, which yields

$$\mathbf{B} = (\mathbf{S}^\mathrm{T}\mathbf{S})^{-1}\mathbf{S}^\mathrm{T}\mathbf{D_B}\mathbf{P}(\mathbf{P}^\mathrm{T}\mathbf{P})^{-1}. \qquad (21)$$

Notice that the weighting function $W(\omega)$ has been incorporated in the relevant matrices, so that the peak absolute error of variable frequency response can be reduced by a proper iterative method, which will be shown in Section 3.

3. Numerical examples and discussions

To demonstrate the effectiveness and flexibility of the proposed method, several examples including a VFO differintegrator, two pure VFO differentiators and a pure VFO integrator are presented in this section. To evaluate the performance, the normalized root-mean-squared error of variable frequency response and the maximum absolute error of variable frequency response are defined by

$$\varepsilon_{rms} = \left[\frac{\int_{p_s}^{p_f}\int_{\omega_s}^{\omega_f} |D(\omega,p) - H(\mathrm{e}^{j\omega},p)|^2\,\mathrm{d}\omega\,\mathrm{d}p}{\int_{p_s}^{p_f}\int_{\omega_s}^{\omega_f} |D(\omega,p)|^2\,\mathrm{d}\omega\,\mathrm{d}p}\right]^{1/2} \times 100\% \qquad (22a)$$

and

$$\varepsilon_m = \max\{|D(\omega,p) - H(\mathrm{e}^{j\omega},p)|,\ \omega_s\leqslant\omega\leqslant\omega_p,\ p_s\leqslant p\leqslant p_f\}, \qquad (22b)$$

respectively. To compute the error ε_{rms}, the general trapezoidal rule is used [34] with step sizes $(\omega_f-\omega_s)/200$ and $(p_f-p_s)/200$ for ω-axis and p-axis, respectively. Also, the error ε_m is computed with the same sampling sizes as above.

Example 1. This example deals with the least-squares design of a VFO differintegrator with $N=40$, $M=5$, $\omega_s=(0.05)\pi$, $\omega_f=(0.95)\pi$, $p_s=-0.5$, $p_f=0.5$ and $W(\omega)=1$. Fig. 1(a) and (b) present the obtained magnitude response and the absolute error of variable frequency response, respectively, and the errors $\varepsilon_{rms}\approx 0.60277728\%$ and $\varepsilon_m = 0.1369375$.

It is noted that the phase difference between $\omega=\pi$ and $\omega=-\pi$ is $p\pi$, which is not an integer multiple of 2π for all p in the range $[p_s, p_f]$, so it is not recommended to set $\omega_f=\pi$. However, for comparing with the results of [22], the differintegrator is designed again with $\omega_s=(0.01)\pi$, $\omega_f=\pi$. If the computation of integration in [22] is implemented by using the trapezoidal rule with step sizes $(\omega_f-\omega_s)/200$ and $(p_f-p_s)/200$ for ω-axis and p-axis, respectively, both the method of [22] and the proposed method induce the exactly same results: $\varepsilon_{rms}\approx 10.01497046\%$ and $\varepsilon_m = 3.11763459$.

Example 2. For designing a pure VFO differentiator, $0\leqslant p_s<p_f$. For example, a VFO differentiator is designed with $N=30$, $M=6$, $\omega_s=0$, $\omega_f=0.9\pi$, $p_s=1$, $p_f=2$ and $W(\omega)=1$, the variable magnitude response and the absolute error of variable frequency response are shown in Fig. 2(a) and (b), respectively, and the errors $\varepsilon_{rms}\approx 0.166372\%$ and $\varepsilon_m = 0.03382684$ which are better than $\varepsilon_{rms}\approx 1.17212338\%$ and $\varepsilon_m = 0.11866149$ obtained with the method of [22] where the ill-conditioned problem will occur for the relevant matrix.

說明：學過微積分後自然對微分及積分不陌生，對多階(整數)微分亦了然於胸。但大部份同學可能就沒聽過 $\frac{1}{2}$ 階微分或 $1\frac{3}{4}$ 階積分了。此即分數階的微分器或積分器，其廣泛応用於自然及商業科學。本單元即要導入此觀念，並運用前一單元之技術來設計可調式分數階微積分器；相關公式推導如次。

Design of variable-fractional-order (VFO) FIR differ-integrators

desired frequency response:

$$H_d(\omega,p)=e^{-j\tau\omega}(j\omega)^p \qquad p_1\le p\le p_2 \quad \omega_1\le|\omega|\le\omega_2$$

$$=e^{-j\tau\omega}(\omega)^p\left[\cos\left(\frac{p\pi}{2}\right)+j\sin\left(\frac{p\pi}{2}\right)\right]$$

p代表微積分器之階數，

p=1代表一階微分器，即對輸入訊號做一次微分。

p=-1代表一階積分器，餘以此類推。

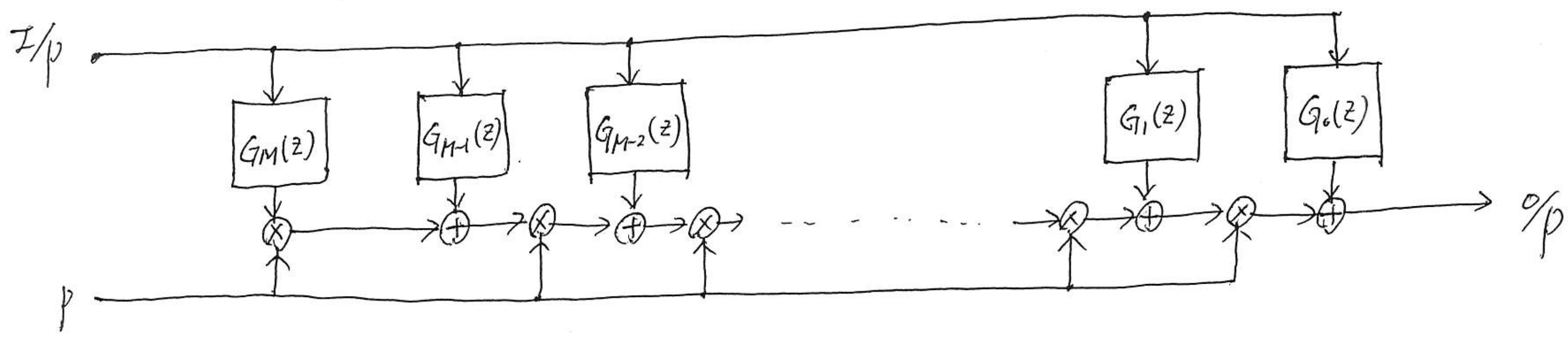
variable FIR digital filter:

$$H(z,p)=\sum_{n=0}^{N}h_n(p)z^{-n}$$

$$h_n(p)=\sum_{m=0}^{M}h(n,m)p^m$$

$$H(z,p)=\sum_{n=0}^{N}\sum_{m=0}^{M}h(n,m)p^m z^{-n}=\sum_{m=0}^{M}\left(\sum_{n=0}^{N}h(n,m)z^{-n}\right)p^m=\sum_{m=0}^{M}G_m(z)p^m$$

$$G_m(z)=\sum_{n=0}^{N}h(n,m)z^{-n}$$



$$h(n,m)=h_e(n,m)+h_o(n,m)$$

$$h_e\left(\frac{N}{2}+n,m\right)=\frac{1}{2}\left[h\left(\frac{N}{2}+n,m\right)+h\left(\frac{N}{2}-n,m\right)\right] \qquad -\frac{N}{2}\le n\le\frac{N}{2} \quad 0\le m\le M$$

N: even

$$h_o\left(\frac{N}{2}+n,m\right)=\frac{1}{2}\left[h\left(\frac{N}{2}+n,m\right)-h\left(\frac{N}{2}-n,m\right)\right] \qquad -\frac{N}{2}\le n\le\frac{N}{2} \quad 0\le m\le M$$

$$H(e^{j\omega},p)=e^{-j\frac{N}{2}\omega}\left[\sum_{n=0}^{N/2}\sum_{m=0}^{M}a(n,m)p^m\cos(n\omega)+j\sum_{n=1}^{N/2}\sum_{m=0}^{M}b(n,m)p^m\sin(n\omega)\right]$$

$$=e^{-j\frac{N}{2}\omega}\left[a^T c(\omega,p)+j\,b^T s(\omega,p)\right]$$

$$a(n,m)=\begin{cases}h_e\left(\frac{N}{2},m\right) & n=0 \quad 0\le m\le M\\ 2h_e\left(\frac{N}{2}-n,m\right) & 1\le n\le\frac{N}{2} \quad 0\le m\le M\end{cases}$$

$$b(n,m)=2h_o\left(\frac{N}{2}-n,m\right) \qquad 1\le n\le\frac{N}{2},\ 0\le m\le M$$

$$a=\left[\,a(n,m) \quad 0\le n\le\frac{N}{2},\ 0\le m\le M\,\right]^T$$

$$b=\left[\,b(n,m) \quad 1\le n\le\frac{N}{2},\ 0\le m\le M\,\right]^T$$

$$c(\omega,p)=\left[\,p^m\cos(n\omega) \quad 0\le n\le\frac{N}{2},\ 0\le m\le M\,\right]^T$$

$$s(\omega,p)=\left[\,p^m\sin(n\omega) \quad 1\le n\le\frac{N}{2} \quad 0\le m\le M\,\right]^T$$

let $I = \frac{N}{2}$

objective error function:

$$e(a,b) = \int_{p_1}^{p_2}\int_{\omega_1}^{\omega_2} |H_d(\omega,p) - H(e^{j\omega},p)|^2 d\omega dp$$

$$= \int_{p_1}^{p_2}\int_{\omega_1}^{\omega_2} |\omega^p \cos(\frac{p\pi}{2}) + j\omega^p \sin(\frac{p\pi}{2}) - a^T c(\omega,p) - j b^T s(\omega,p)|^2 d\omega dp$$

$$= \int_{p_1}^{p_2}\int_{\omega_1}^{\omega_2} [\omega^p \cos(\frac{p\pi}{2}) - a^T c(\omega,p)]^2 d\omega dp + \int_{p_1}^{p_2}\int_{\omega_1}^{\omega_2} [\omega^p \sin(\frac{p\pi}{2}) - b^T s(\omega,p)]^2 d\omega dp$$

$$= e(a) + e(b)$$

$$e(a) = s_a + r_a^T a + a^T Q_a a$$

$$e(b) = s_b + r_b^T b + b^T Q_b b$$

$$s_a = \iint (\omega^p \cos(\frac{p\pi}{2}))^2 d\omega dp \qquad s_b = \iint (\omega^p \sin(\frac{p\pi}{2}))^2 d\omega dp$$

$$r_a = -2\iint \omega^p \cos(\frac{p\pi}{2})\, c(\omega,p)\, d\omega dp \qquad r_b = -2\iint \omega^p \sin(\frac{p\pi}{2})\, s(\omega,p)\, d\omega dp$$

$$Q_a = \iint c(\omega,p)\, c^T(\omega,p)\, d\omega dp \qquad Q_b = \iint s(\omega,p)\, s^T(\omega,p)\, d\omega dp$$

$$\frac{\partial e(a,b)}{\partial a} = \frac{\partial e(a)}{\partial a} = r_a + 2Q_a a = 0 \Rightarrow a = -\frac{1}{2}Q_a^{-1} r_a$$

$$\frac{\partial e(a,b)}{\partial b} = \frac{\partial e(b)}{\partial b} = r_b + 2Q_b b = 0 \Rightarrow b = -\frac{1}{2}Q_b^{-1} r_b$$

Example: $N=40$ $M=5$ $\omega_1 = 0.05\pi$ $\omega_2 = 0.95\pi$ $p_1 = -0.5$ $p_2 = 0.5$

Example: $N=30$ $M=6$ $\omega_1 = 0$ $\omega_2 = 0.9\pi$ $p_1 = 1$ $p_2 = 2$

Example: $N=60$ $M=6$ $\omega_1 = 0.05\pi$ $\omega_2 = 0.9\pi$ $p_1 = -1.5$ $p_2 = -0.5$

```python
##
## Design of variable fractional-order (VFO) differintegrators
##
import numpy as np
import math
import matplotlib.pyplot as plt
from scipy import signal

N=40
M=5
w1=0.05*math.pi
w2=0.95*math.pi
p1=-0.5
p2=1
Nw=200
Np=60
#
NH=N//2
nma=(NH+1)*(M+1)
nmb=NH*(M+1)
deltaw=(w2-w1)/Nw
deltap=(p2-p1)/Np
Nwp=(Nw+1)*(Np+1)
NVa=np.arange(0,NH+1); NVa=NVa[:,np.newaxis]
NVb=np.arange(1,NH+1); NVb=NVb[:,np.newaxis]
##
ra=np.zeros((nma,1))
Qa=np.zeros((nma,nma))
for ip in range (0,Np+1):
    p=p1+ip*deltap
    for iw in range (0,Nw+1):
        w=w1+iw*deltaw
        cwp=np.zeros((nma,1))
        for im in range (0,M+1):
            cwp[im*(NH+1):(im+1)*(NH+1),0]=p**(im)*np.cos(w*NVa[:,0])
        ra=ra-2*w**p*np.cos(p*math.pi/2)*cwp
        Qa=Qa+cwp@np.transpose(cwp)
ra=(w2-w1)*(p2-p1)*ra/Nwp
Qa=(w2-w1)*(p2-p1)*Qa/Nwp
a=-0.5*np.linalg.inv(Qa)@ra
##
rb=np.zeros((nmb,1))
Qb=np.zeros((nmb,nmb))
for ip in range (0,Np+1):
    p=p1+ip*deltap
    for iw in range (0,Nw+1):
        w=w1+iw*deltaw
        swp=np.zeros((nmb,1))
        for im in range (0,M+1):
            swp[im*NH:(im+1)*NH,0]=p**im*np.sin(w*NVb[:,0])
        rb=rb-2*w**p*np.sin(p*math.pi/2)*swp
        Qb=Qb+swp@np.transpose(swp)
rb=(w2-w1)*(p2-p1)*rb/Nwp
Qb=(w2-w1)*(p2-p1)*Qb/Nwp
b=-0.5*np.linalg.inv(Qb)@rb
##
a2=np.reshape(a,(M+1,NH+1)); a2=np.transpose(a2)
he=np.zeros((N+1,M+1))
he[NH,:]=a2[0,:]
he[0:NH,:]=0.5*np.flipud(a2[1:NH+1,:])
he[NH+1:N+1,:]=0.5*a2[1:NH+1,:]
##
b2=np.reshape(b,(M+1,NH)); b2=np.transpose(b2)
ho=np.zeros((N+1,M+1))
```

```
ho[0:NH,:]=0.5*np.flipud(b2)
ho[NH+1:N+1,:]=-0.5*b2
##
h2=he+ho
MR=np.zeros((Nw+1,Np+1,1))
for ip in range (0,Np+1):
    p=p1+ip*deltap
    h=h2[:,0]
    for im in range (1,M+1): h=h+h2[:,im]*p**im
    rr=np.linspace(w1,w2,num=Nw+1); rr=rr[:,np.newaxis]
    MRR=np.absolute(signal.freqz(h,1,rr))
    MR[:,ip]=MRR[1]

for i in range (0,Np+1): plt.plot(rr/math.pi,MR[:,i])
plt.axis([w1/math.pi,w2/math.pi,0,3])
plt.xlabel('Normalized frequency')
plt.ylabel('Amplitude response')
plt.title('VFO')
plt.show()
```