APPUNTI SUL FILTRO 1-POLO

CARMINE EMANUELE CELLA

SOMMARIO. Forma differenziale e simmetrica; funzione di trasferimento e risposta in frequenza; calcolo dei coefficienti mediante condizioni sull'equazione del guadagno.

1. CARATTERISTICHE PRINCIPALI

Può assumere comportamento tipo *passa-basso* o *passa-alto*; toglie 6 dB per ottava. È possibile mettere questo filtro in serie fino ad ordini piuttosto alti, mantenendo un discreto comportamento frequenziale.

2. EQUAZIONI

Lo schema a blocchi è rappresentato in figura 1 (si noti l'uso del coefficiente a_1 col segno cambiato) e la relativa equazione differenza è la seguente:

$$y[n] = b_0 x[n] - a_1 y[n-1] \tag{1}$$

la cui forma simmetrica è:

$$y[n] + a_1 y[n-1] = b_0 x[n]. \tag{2}$$

L'operatore *ritardo unitario*, definito dal simbolo z^{-1}, restituisce il valore precedente nella sequenza rispetto al valore a cui è moltiplicato, immettendo nel sistema un ritardo pari ad un campione. Moltiplicando dunque z^{-1} al valore n-esimo otteniamo il valore (n-1)-esimo: $z^{-1}y[n] = y[n-1]$. È possibile, quindi, riscrivere l'equazione simmetrizzata 2 nella seguente forma:

$$y[n] + a_1 z^{-1} y[n] = b_0 x[n] \tag{3}$$

da cui si ottiene, raccogliendo $y[n]$ e dividendo per $x[n]$:

$$\frac{y[n]}{x[n]} = \frac{b_0}{(1 + a_1 z^{-1})} = H(z). \tag{4}$$

L'equazione 4 è comunemente nota col nome di *funzione di trasferimento* del filtro digitale ed è generalmente indicata col simbolo $H(z)$; tutte le formulazioni date fin'ora sono equivalenti. Tuttavia, dalla funzione di trasferimento è possibile ricavare importanti informazioni sul comportamento del filtro; assumendo che z sia un numero complesso[1] e che valga $e^{j\omega t}$, si ottiene:

[1] Si veda la sezione 3.

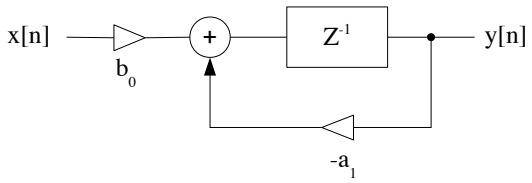

FIGURA 1. Schema a blocchi per il filtro 1-polo

$$H(e^{j\omega t}) = \frac{b_0}{(1 + a_1 e^{-j\omega t})} \tag{5}$$

L'equazione 5 è nota come *risposta in frequenza* del filtro: attraverso essa è possibile conoscere come si comporta il filtro ad una data frequenza ω. Prendendo il modulo dell'equazione 5 si ottiene l'andamento dell'ampiezza del filtro:

$$\begin{aligned} G(\omega t) &= |H(e^{j\omega t})| \\ &= \frac{|b_0|}{|1 + a_1 e^{-j\omega t}|} \\ &= \frac{|b_0|}{\sqrt{(1 + a_1 e^{-j\omega t})(1 + a_1 e^{j\omega t})}} \\ &= \frac{|b_0|}{\sqrt{1 + a_1 e^{-j\omega t} + a_1 e^{j\omega t} + a_1^2 e^{-j\omega t} e^{j\omega t}}} \\ &= \frac{|b_0|}{\sqrt{1 + a_1^2 + 2a_1 cos(\omega t)}} \end{aligned} \tag{6}$$

Calcolando l'angolo della medesima equazione, invece, si ottiene l'andamento di fase $\Theta(\omega t) = \angle H(e^{j\omega t})$ che tuttavia non verrà discusso oltre.

Per calcolare i coefficienti b_0 e a_1 in rapporto alla frequenza di taglio scelta, è possibile seguire il procedimento seguente:

(1) determinare il massimo della funzione guadagno (eq. 6);
(2) imporre la condizione di uguaglianza a 1 del massimo trovato (*normalizzazione*) per ricavare il coefficiente b_0;
(3) imporre la condizione di ugualianza a $1/\sqrt{2}$ del guadagno, dopo aver sostituito b_0 (ciò deriva dalla definione di *frequenza di taglio*);

Per il caso *low-pass* il massimo della funzione guadagno per questo filtro i trova a 0 Hz, in cui $cos(0) = 1$; quindi:

$$G(0) = \frac{b_0}{\sqrt{1 + 2a_1 + a_1^2}} = 1 \tag{7}$$

da cui si ricava che $b_0 = 1 + a$. Sostituendo b_0 in 6 e imponendo l'uguaglianza con $1/\sqrt{2}$ si ottiene:

$$
\begin{aligned}
&\frac{1+a_1}{\sqrt{1+2a_1cos(\omega t)+a_1^2}} = 1/\sqrt{2} \\
&\Rightarrow 2(1+a_1)^2 = 1+2a_1cos(\omega t)+a_1^2 \\
&\Rightarrow 2+2a_1^2+4a_1-1-2a_1cos(\omega t)-a_1^2 = 0 \\
&\Rightarrow a_1^2+2a_1(2-cos(\omega t))+1 = 0 \\
&\Rightarrow a_1 = \frac{-2(2-cos(\omega t)) \pm 2\sqrt{(2-cos(\omega t))^2-1}}{2} \\
&\Rightarrow a_1 = -(2-cos(\omega t)) \pm \sqrt{(2-cos(\omega t))^2-1}.
\end{aligned} \tag{8}
$$

Scartando la soluzione negativa per a_1, i coefficienti ottenuti sono:

$$
\begin{aligned}
&\mathbf{a_{1LP}} = cos(\omega) - 2 + \sqrt{(2-cos(\omega))^2-1} \\
&\mathbf{b_{0LP}} = 1 + a_{1LP}.
\end{aligned} \tag{9}
$$

Per il caso *high-pass*, invece, il massimo si trova a π in cui $cos(\pi) = -1$; quindi:

$$
G(\pi) = \frac{b_0}{\sqrt{1-2a_1+a_1^2}} = 1 \tag{10}
$$

da cui si ricava che $b_0 = 1 - a$. Sostituendo b_0 in 6 e imponendo l'uguaglianza con $1/\sqrt{2}$ si ottiene:

$$
\begin{aligned}
&\frac{1-a_1}{\sqrt{1+2a_1cos(\omega t)+a_1^2}} = 1/\sqrt{2} \\
&\Rightarrow 2(1-a_1)^2 = 1+2a_1cos(\omega t)+a_1^2 \\
&\Rightarrow 2+2a_1^2-4a_1-1-2a_1cos(\omega t)-a_1^2 = 0 \\
&\Rightarrow a_1^2-2a_1(2-cos(\omega t))+1 = 0 \\
&\Rightarrow a_1 = \frac{-2(2-cos(\omega t)) \pm 2\sqrt{(2-cos(\omega t))^2-1}}{2} \\
&\Rightarrow a_1 = (2-cos(\omega t)) \pm \sqrt{(2-cos(\omega t))^2-1}.
\end{aligned} \tag{11}
$$

Scartando in questo caso la soluzione positiva per a_0, i coefficienti ottenuti sono:

$$
\begin{aligned}
\mathbf{a_{1HP}} &= 2 - cos(\omega) - \sqrt{(2-cos(\omega))^2-1} \\
&= -1.(a_{1LP}) \\
\mathbf{b_{0HP}} &= 1 - a_{1HP}.
\end{aligned} \tag{12}
$$

Dall'analisi effettuata appare evidente che il coefficiente a_1 dipende da ω e dunque determina la frequenza di taglio del filtro ed anche la tipologia: se assume un valore positivo il filtro sarà passa alto, altrimenti passa-basso. Si noti, inoltre, che ciò che differenzia i due comporamenti è solo il segno di a_1, ovvero: $a_{1HP} = -a_{1LP}$. Il coefficiente b_0, invece, determina il guadagno del filtro: imponendo la condizione di uguaglianza ad 1 per il massimo, si otterrà un filtro il cui guadagno sarà quasi unitario[2].

[2]Per alcuni ritardi di fase tuttavia, potrebbe ancora verificarsi una saturazione in alcune circostanze.

3. Riferimenti matematici

Equazione di Eulero (in cui $j = \sqrt{-1}$): $e^{j\omega t} = cos(\omega t) + jsin(\omega t)$.

Si noti che $sin^2(\omega t) + cos^2(\omega t) = 1$ e dunque $sin^2(\omega t) = 1 - cos^2(\omega t)$; per questa ragione la relazione di Eulero può essere scritta anche nei seguenti modi:

$$\begin{aligned} e^{j\omega t} &= cos(\omega t) + \sqrt{-1 sin^2(\omega t)} \\ &= cos(\omega t) + \sqrt{-1(1 - cos^2(\omega t))} \\ &= cos(\omega t) + \sqrt{cos^2(\omega t) - 1}. \end{aligned} \tag{13}$$

Da tale equazione si ricava anche che $2cos(\omega t) = e^{-j\omega t} + e^{j\omega t}$.

Numeri complessi: se z è un numero complesso esso si può esprimere in forma rettangolare $z = a + jb$ o in forma polare $z = r[cos(\phi) + jsin(\phi)]$ dove $a = rcos(\phi)$ e $b = rsin(\phi)$. È possibile calcolare il valore assoluto di z nei seguenti modi: $|z| = \sqrt{(a^2 + b^2)} = z\overline{z}$, in cui $\overline{z}$ è il complesso coniugato di z, mentre il suo *argomento* si ottiene con $\angle(z) = arctan(b/a)$. Si noti, infine, che:

$$\begin{aligned} \mathbf{e^{j\omega t} e^{-j\omega t}} &= \mathbf{e^{j\omega t}\overline{e^{j\omega t}}} = \\ [cos(\omega t) + jsin(\omega t)][cos(\omega t) + jsin(\omega t)]] &= \\ cos^2(\omega t) - jsin(\omega t)cos(\omega t) + jsin(\omega t)cos(\omega t) + sin^2(\omega t) &= \mathbf{1}. \end{aligned} \tag{14}$$

4. Calcolo alternativo del guadagno

È possibile calcolare il guadagno del filtro sostituendo e^{jw} immediatamente, mediante la relazione di Eulero:

$$\begin{aligned} G(\omega t) &= |H(e^{j\omega t})| \\ &= \frac{|b_0|}{|1 + a_1[cos(\omega t) - jsin(\omega t)]|} \\ &= \frac{|b_0|}{\sqrt{[1 + a_1 cos(\omega t)]^2 + [-a_1 sin(\omega t)]^2}} \\ &= \frac{|b_0|}{\sqrt{1 + a_1^2 cos^2(\omega t) + 2a_1 cos(\omega t) + a_1^2 sin^2(\omega t)}} \\ &= \frac{|b_0|}{\sqrt{1 + a_1^2(cos^2(\omega t) sin^2(\omega t)) + 2a_1 cos(\omega t)}} \\ &= \frac{|b_0|}{\sqrt{1 + a_1^2 + 2a_1 cos(\omega t)}} \end{aligned} \tag{15}$$

5. Implementazione in C++

Il seguente listato in C++ mostra l'implementazione del filtro descritto nella classe *OnePole* e un suo uso esemplificativo come low-pass con un impluso unitario:

Listing 1. Implementazione del filtro 1-polo

```
// onePoleTest.cpp - LP filter
```

```
//

#include <stdexcept>
#include <iostream>
#include <fstream>
#include <complex>
#include <cmath>

using namespace std;

const int BSIZE = 4096;
const double TWOPI = 8. * atan (1.);
const double PI = 4. * atan (1.);

const double SR = 44100;

#define DEBUG

template <typename T>
class OnePole {
public:
  OnePole (T sr, T cutoff, int order, bool lp = true) : m_y1 (0) {
    T ST = 1.  / sr;
    T omega = TWOPI * cutoff;
    T b = 2. - cos (omega * ST);

    if (lp) {
      m_a1 = cos (omega * ST) - 2. +
        sqrt ((b * b)  - 1.);
      m_b0 = 1. + (m_a1);
    } else {
      m_a1 = b - sqrt ((b * b)  - 1.);
      m_b0 = 1. - (m_a1);
    }

    m_order = order;
    m_y1 = new T[m_order];

#ifdef DEBUG
    cout << "b0 = " << m_b0 << ", a1 = " << m_a1 << endl;
#endif
  }
  virtual ~OnePole () {
    delete [] m_y1;
  }
  T* process (const T* input, T* output, int size) {
    T y = 0;
    T* tmp = (T*) input;
    for (int j = 0; j <  m_order; ++j) {
      for (int i = 0; i < size; ++i) {
        y = (m_b0 * tmp[i]) - (m_a1 * m_y1[j]);
        output[i] = m_y1[j] = y;
      }
      tmp = output;
```

```
    }
    return output;
  }
private:
  T* m_y1;
  T m_a1;
  T m_b0;
  int m_order;
};

int main (int argc, char* argv[]) {
  try {
    if (argc != 4) {
      throw runtime_error (
        "syntax is 'onePoleTest cutoff order lp [=0|1]'"
      );
    }


    float cutoff = atof (argv[1]);
    int order = atoi (argv[2]);
    bool lp = (bool) atoi (argv[3]);

    float* pulse = new float[BSIZE];
    float* out = new float[BSIZE];
    for (int i = 0; i < BSIZE; ++i) pulse[i] = 0.;
    pulse[0] = 1.;

    OnePole <float> opol (SR, cutoff, order, lp);
    opol.process (pulse, out, BSIZE);
    ofstream outf ("h_n.raw", ios::binary);
    for (int i = 0; i < BSIZE; ++i) {
      outf.write ((char*) &out[i], sizeof (float));
    }
    outf.close ();

    delete [] pulse;
    delete [] out;

  }
  catch (exception& e) {
    cout << "Error: " << e.what () << endl;
  }
  catch (...) {
    cout << "Fatal error: unknown exception" << endl;
  }
  return 0;
}

// EOF
```

Riferimenti bibliografici

[1] R. Moore. *Elements of computer music*. Prentice Hall - Englewood cliffs, New Jersey 07632, 1990.
[2] M. Puckette. *The Theory and Technique of Electronic Music*. World Scientific Publishing Co. Pte. Ltd., 2007.
[3] J. Smith. *Introduction to digital filters with audio applications*. WWW edition - http://ccrma.stanford.edu/ jos/filters/, 2009.
[4] Unknown. *INTRODUCTION TO DIGITAL FILTERS*. Found on the internet; look for digfilt.pdf.

Università degli studi di Bologna, Via Zamboni 38 - 40126 Bologna, Italia