

Ciele cvičenia

V rámci cvičenia sa budeme zaoberať číslicovými filtermi s konečnou impulzovou odozvou (FIR – Finite Impulse Response). Opíšeme základné štruktúry týchto filtrov: priamu formu I. a II., kaskádne a paralelné zapojenie. Vysvetlíme použitie bilinéarnej transformácie pri návrhu filtrov a naučíme sa pracovať so špecializovaným nástrojom SPTOOL[®] programu MATLAB[®], určeným pre rýchly a jednoduchý návrh číslicových filtrov. Na záver cvičenia, s využitím nástroja SPTOOL navrhne FIR filter podľa zadania, implementujeme ho na vývojovej doske TMS320C6713 DSK a zmeriame jeho parametre.

1. Úvod

Ako už vieme, správanie číslicových filtrov opisujeme diferenčnými rovnicami. V prípade filtrov FIR má táto rovnica nasledujúci všeobecný rekurzívny tvar

$$\begin{aligned} y(n) &= \sum_{k=0}^N b_k x(n-k) = \\ &= b_0 x(n) + b_1 x(n-1) + \dots + b_N x(n-N) \end{aligned} \quad (10.1)$$

Vidíme, že postupnosť výstupných vzoriek $y(n)$ závisí len od postupnosti vstupných vzoriek $x(n)$, $x(n-1)$, $x(n-2)$, ..., $x(n-N)$, FIR filtre nemajú spätnú väzbu.

Ak budeme predpokladať, že všetky počiatočné podmienky v rovnici (10.1) sú rovné nule, potom po z-transformácii tejto rovnice dostávame

$$Y(z) = b_0 X(z) + b_1 z^{-1} X(z) + \dots + b_N z^{-N} X(z) \quad (10.2)$$

Nech v rovnici (10.2) N , potom prenosová funkcia $H(z)$ bude

$$H(z) = \frac{Y(z)}{X(z)} = \frac{b_0 + b_1 z^{-1} + \dots + b_N z^{-N}}{1} = \frac{N(z)}{D(z)} \quad (10.3)$$

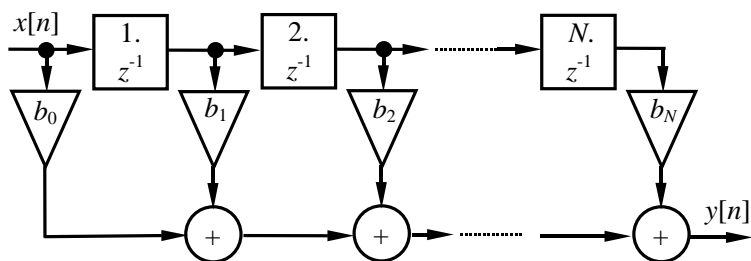
kde $N(z)$ a $D(z)$ reprezentujú polynómy čitateľa a menovateľa prenosovej funkcie. Takáto prenosová funkcia obsahuje len nulové body, okrem N -násobného pólu v $z = 0$. Takýto FIR filter má vlastnosť vždy stabilného filtra.

2. Štruktúra FIR filtra

FIR filter je možné zostrojiť pomocou funkčných blokov do priamej tranzverzálnej sústavy. Priame formy je možné transponovať a kombinovať v paralelnej alebo kaskádovej štruktúre tak, aby sme získali filtre vyšších rádov. Tieto štruktúry a ich kombinácie teraz postupne opíšeme.

Priama forma

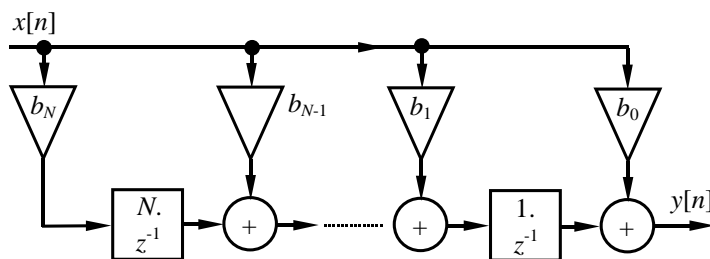
Ak sa rozhodneme realizovať FIR filter presne podľa výrazu pre jeho prenos (10.3), bude jednoznačne daný štruktúrou podľa obr.1. Pre filter N -tého rádu bude mať štruktúra $2N$ oneskorovacích členov, reprezentovaných pri z-transformácii výrazom z^{-1} .



Obr.1. Priamy tvar realizácie FIR sústavy.

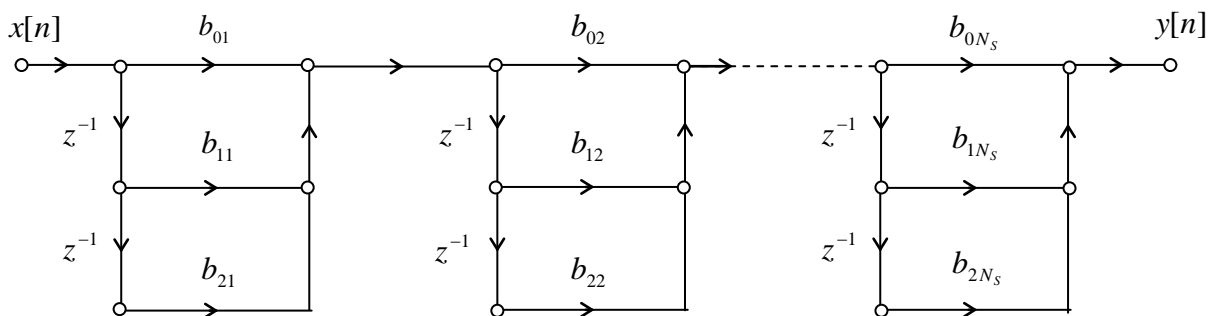
Transponovaná forma.

Predchádzajúcu štruktúru FIR filtra je možné použitím metódy transpozície zmeniť na transponovaný priamy model FIR sústavy, ktorú vidíme na obr.2.



Obr.2 Transponovaný priamy tvar realizácie FIR sústavy.

Existujú samozrejme, ďalšie štruktúry vhodné pre aplikácie v oblasti adaptívnych filtrov, rozoznávania hlasu, atď. Tieto sú však obvykle výpočtovo náročnejšie než priama forma, resp. transponovaná priama forma a vyžadujú viac operácií násobenia. Ich výhodou je však nižšia citlivosť na efekt kvantovania. Kvantizačná chyba súvisiaca s koeficientami FIR filtra závisí od veľkosti zmeny pozície núl jeho prenosovej funkcie v komplexnej rovine. Pre jednoduchšie programové riešenie a elimináciu chýb sa štruktúry rozkladajú do jednotlivých sekcií 2. rádu zapojené v kaskáde (za sebou). Dosiadne sa tejto skutočnosti rozkladom prenosovej funkcie $H(z)$ (obr. 3).



Obr. 3 Orientovaný graf kaskádnej realizácie FIR sústavy

Zadanie

1. Preštudujte si teoretický úvod k problematike.
2. V prostredí CCS otvorte projekt *fir.pjt*, upravte súbor *fir.c* tak, aby vkladacia direktíva *#include* vložila do programu súbor s koeficientami pre pásmovú zádrž *bs2700.cof*. Na prilinkovanie súboru s koeficientmi použite "Scan All Dependencies" v hlavnom menu.
3. Projekt uložte a vykonajte kompiláciu. Výsledný objektový súbor *fir.out* uložte do pamäte procesora a spustite program. Pomocou osciloskopu a funkčného generátora odmerajte jeho charakteristiky.
4. Pomocou importovanej funkcie v programovom prostredí CCS graficky zobrazte časovú a frekvenčnú charakteristiku filtra pásmovej zádrže (View → Graph → Time/Frequency). Porovnajte zobrazené charakteristiky filtra z odmeranými charakteristikami filtra z úlohy č.3.
5. Odskúšajte napísaný program aj pre ostatné typy filtrov s koeficientami uloženými v príslušnom *.cof súbore.

K úlohe 2)

Preštudujte výpis súboru *fir.c*:

```
//Fir.c FIR filter. Include coefficient file with length N

#include "bs2700.cof"           //coefficient file
#include "dsk6713_aic23.h"     //codec-dsk support file
Uint32 fs=DSK6713_AIC23_FREQ_8KHZ; //set sampling rate
int yn = 0;                   //initialize filter's output
short dly[N];                 //delay samples

interrupt void c_int11()      //ISR
{
    short i;

    dly[0]=input_sample();    //input newest sample
    yn = 0;                   //initialize filter's output
    for (i = 0; i < N; i++)
        yn += (h[i] * dly[i]); //y(n) += h(i)* x(n-i)
    for (i = N-1; i > 0; i--) //starting @ end of buffer
        dly[i] = dly[i-1];   //update delays with data move
    output_sample(yn >> 15); //scale output filter sample
    return;
}

void main()
{
    comm_intr();              //init DSK, codec, McBSP
    while(1);                 //infinite loop
}
```