

Nyolcadfokú digitális, kaszkád IIR szűrő tervezése



LÁSZLÓ
HAJNALKA

ÖSSZEFOGLALÁS

A dolgozat egy Weaver realizációjú transzmultiplexer berendezés csatornaszűrőjének megvalósításához kapcsolódva, egy nyolcadfokú digitális IIR szűrő tervezéséről, számítógépes szimulációról számol be. A dolgozat kitér a tervezés menetére, a számítógépes szimuláció az együtthatók módosulásának hatását szemlélteti. A szűrő TMS32010 jelfeldolgozó processzorral való megvalósításakor a szűrő a tényleges digitális környezetbe kerül.

Bevezetés

Digitális szűrők tervezésekor a legelterjedtebben az [1]-ben is részletesen leírt tervezési módszert szokták alkalmazni. Egy másik, a [2] irodalomban felléphető szűrő tervezés ettől a klasszikusnak nevezhető módszertől lépéseiben eltérő, de végeredménye ugyancsak egy, bilineáris transzformációval nyert $H(z)$ digitális transzferfüggvény.

1. Nyolcadfokú digitális szűrő tervezése

1.1 A szűrő tervezése

A C 08 A típusú analóg szűrőnek megfelelő nyolcadfokú digitális IIR szűrő specifikációja:

- az áteresztő tartomány = 1,6 kHz-ig tart és itt a csillapítás $A_{\max} = 0,27$ dB;
- a zárósáv = 2 kHz-nél kezdődik, ebben a tartományban a csillapítás $A_{\min} = 85,35$ dB;
- a mintavételezési frekvencia: 56 kHz.

A digitális szűrő tervezése most a következőképpen történik:

1. Keresünk a kiválasztott szűrő specifikációnak megfelelő $H_A(s)$ analóg átviteli függvényt. A [3] szűrőtáblázatban lévő adatokból (reflexió tényező, zérus — pólus gyökök értékei, A_{\min} zárósávi csillapítás, konstans értéke, zárósávi normalizált frekvencia stb.) a $H_A(s)$ analóg transzferfüggvény pontos kifejtése nem igényel bonyolult számítást.

2. Kiszámítjuk a digitális cutoff (levágási) frekvenciát a mintavételezési frekvenciából (56 kHz) és az áteresztő tartományi f_p frekvenciából (1,6 kHz):

$$\omega_c = 2 \cdot \Pi \cdot f_p \cdot T,$$

ahol T a mintavételezési frekvencia reciproka.

Esetünkben $\omega_c = 0,1795195$ rad/s.

3. A bilineáris transzformációhoz szükséges előtorzítás megadja az analóg cutoff (levágási) frekvenciát:

$$\Omega_c = \operatorname{tg}(\omega_c/2), \text{ esetünkben } \Omega_c = 0,900016.$$

Beérkezett: 1987. IX. 2. (H)

A $2/T$ -vel való szorzást az irodalomban nem vették figyelembe, valószínűleg ezt a szorzási faktort 1-nek veszik.

4. Ezzel az értékkel skálázva a $H_A(s)$ -t egy $H'_A(s)$ átviteli függvényt kapunk, amelyből a bilineáris transzformáció segítségével kiszámíthatjuk a $H(z)$ digitális transzferfüggvényt.

5. A $H'_A(s)$ -ből $s = (z-1)/(z+1)$ bilineáris transzformációt alkalmazva adódik $H(z)$:

$$H(z) = 6,676175 \cdot 10^{-5} \cdot (z^2 - 1,934500z + 1) / (z^2 \cdot 1,954962z + 0,9876002) \cdot (z^2 - 1,916631z + 1) / (z^2 - 1,933389z + 0,9594879) \cdot (z^2 - 1,838748z + 1) / (z^2 - 1,91088z + 0,9256088) \cdot (z^2 - 1,053993z + 1) / (z^2 - 1,893201z + 0,8979977).$$

Ha a két szűrőtervezési módszert összehasonlítjuk, az utolsó lépés kivételével az egyes lépések meglehetősen eltérnek egymástól, főképpen a sorrendjüket illetően. Míg a klasszikus módszernél a digitális frekvenciából kiindulva lehet eljutni az analóg transzferfüggvényig, az utóbb leírt módszernél egyenesen a $H_A(s)$ függvény a kiindulópont.

1.2 A digitális szűrőt analizáló programok

1.2.1 A $H_A(s)$ átviteli karakterisztikát analizáló program

Ez a program 3 fő részből áll:

- az adatbevitel ($H_A(s)$ együtthatói),
- a frekvenciától függő csillapítás értékek kiszámítása ($a = 20 \times \lg |H_A(s)|$ [dB]),
- az eredmények grafikonos formában való kiíratása.

A program futtatási eredménye az 1. ábrán látható.

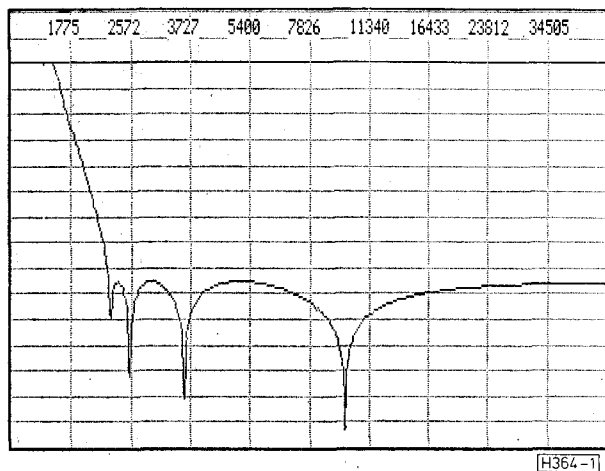
1.2.2 Az időtartományi viselkedést leíró program

A program felépítése:

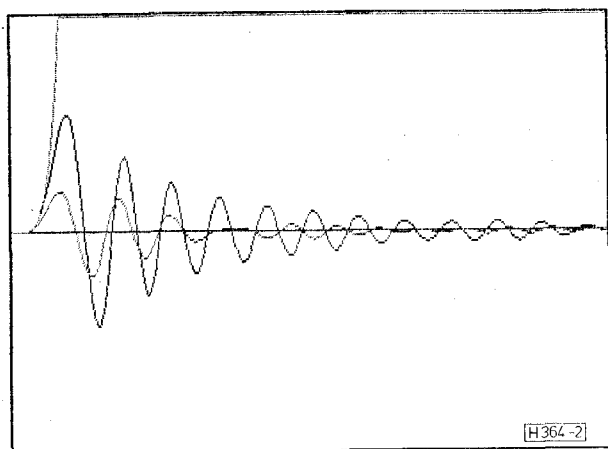
1. — szinusz jel minták generálása a mintavételezési frekvencia figyelembevételével.

2. — az $x(n)$ szinusz minták bevétele a szűrő első másodfokú alaptagjának bementén és rekurzív képlet alapján a 4 alaptag kimeneti mintáinak kiszámítása.

3. — A nyolcadfokú szűrő kimenetén lévő $z(n)$ diszkrét jelemták kirajzoltatása különböző, a felhasználó által definiálható frekvenciák esetén.



1. ábra. A szűrő analóg karakterisztikája (10 dB/osztás)



2. ábra. A nyolcadfokú digitális IIR-szűrő időtartománybeli viselkedése.

- erős folytonos vonal: a szűrő kimeneti jele,
- halvány vékony vonal: a 8 bites kerekített kimeneti jel
- halvány pont-pont vonal: a 8 bites csonkított kimeneti jel

A program futtatásakor azonnal látható, hogy egy szűrő teljesíti-e az előírt specifikációt, ugyanis az áteresztő tartományban a szinuszminták alakúan jönnek ki a szűrő kimenetén, míg az áteresztő tartomány felett és a zárószámban a szűrő csillapítja a szinuszmintákat. Ezenkívül megfigyelhető a tranzienst viselkedés, ami azt jelenti, hogy kb. csak 10 mintavételezési ciklus után következik be az állandósult állapot.

A futtatási eredmények a 2. ábrán láthatók.

A programokat egy M 68000 processzorral működő SINCLEAR QL gépen írtam és futtattam.

1.3 A kerekítés és csonkítás hatása

A szinuszminták feldolgozása során a szorzások után kapott eredmények több bites formában állnak rendelkezésre, mint amennyire a további számolásokhoz szükségünk van. A felesleges bitektől csonkítással vagy kerekítéssel tudunk meg-

szabadulni. A legkisebb helyértékű bitek elhagyásának az az ára, hogy a szűrők mintái, együttműködő, ki- és bemenő adatai csak bizonyos pontossággal fognak megfelelni a névleges értéknek.

A kerekítés és csonkítási műveletek hatása a szűrő kimeneti jelére a 2. ábrán látható.

Eleinte csak a beérkező szinuszmintákat kerekítettem ill. csonkoltam, ekkor a relatív hibák értékei voltak, 10^{-2} , 10^{-6} nagyságrendben. Aztán kerekítettem és csonkoltam a szűrő együttműködőit is, mivel ez a realizáláshoz közelebb áll, így viszont a szűrő kimenetén a csonkolt jelek az eredeti (nem csonkolt) jelekhez képest fázistorzulást szenvednek (ahogy a 2. ábrán is látható pl. 8-bites kerekítés ill. csonkítás esetén).

2. Realizálás TMS32010 jelleldolgozó processzorral

Egy adott fókuszszámú, digitális IIR-szűrőt realizálhatunk TMS 32010 processzorral egy TMS forrásnyelvű programon keresztül, ez utóbbit az EVM editorjában szerkesztve. Ezután a programot az EVM assemblerjével lefordítjuk, és az assembly fordítással nyert gépi kódoknak megfelelően a TMS32010 processzor elvégzi a szűrést.

Ennél sokkal kényelmesebb, megoldást nyújt egy olyan program, amelyhez a felhasználó csupán annyit kell tudjon, hogy hány másodfokú alaptagból épül fel a realizálandó kaszkád struktúrájú digitális szűrő és mekkora az alaptagonkénti szűrőegység pontos decimális értéke.

Egy ilyen program íródott a SINCLEAR QL gép Super Basic nyelvén és fő részét a szűrést végrehajtó TMS forrásnyelvű program gépi kódjainak generálása képezi. Az így kapott gépi kódokat elmentjük egy file-ban a QL magnóján, és a QL és TMS EVM közötti illesztő program betöltése után az LPM utasítást használva, a file-ban lévő gépi kódok betöltődnek a processzor program memóriájába. Az EVM egy újabb, hasznos szolgáltatásával, a REVERSE ASSEMBLER-rel (RASM) a gépi kódokból visszafejthetjük a TMS forrásnyelvű (source-) programot leellenőrizhetjük, hogy a Basic nyelvű programunk valóban a helyes gépi kódokat generálta-e? A program memória megfelelő területéről, ahol lehelyeztük a szűrő kimeneti jelmintáit, a hexadecimális formátumú kódok megtekintésével is meggyőződhetünk, hogy a processzor realizálta-e a szűrőt. Az eredményt oszcilloszkóp és a Fejlesztő Rendszerhez tartozó Analóg kártya, vagy kirajzoló program segítségével tehetjük szemléletessé.

IRODALOM

- [1] Dr. Simonyi Ernő: Digitális szűrők. (Műszaki Könyvkiadó, Budapest, 1984. 451—452. oldal)
- [2] Mark Howard Eitzel: Hardware design of a digital single-sideband generator. (University of Illinois at Urbana Champaign, 1978, pp. 24—25.)
- [3] Christian, Erich and Egon Eisenmann: Filter Design Tables and Graphs. (John Wiley and Sons. Inc., 1966. p. 201)

László Hajnalka
BME Villamosmérnöki Kar