

**UNIVERZA V LJUBLJANI
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO**

Andraž Božiček

**UPORABA ALGORITMOV ZA REDUKCIJO BAZE MREŽE PRI
POLINOMSKI APROKSIMACIJI S KOEFICIENTI KONČNE
DOLŽINE**

MAGISTRSKO DELO

Mentor: prof. dr. Dušan Kodek

Ljubljana, 2011

Št.: 139-MAG-RI/2011
Datum: 27. 05. 2011



Andraž BOŽIČEK, univ. dipl. inž. rač. in inf.

Ljubljana

Fakulteta za računalništvo in informatiko Univerze v Ljubljani izdaja naslednjo magistrsko nalogo

Naslov naloge: **Uporaba algoritmov za redukcijo baze mreže pri polinomske aproksimaciji s koeficienti končne dolžine**

Application of lattice basis reduction algorithms for polynomial approximation with finite wordlength coefficients

Tematika naloge:

Na problem načrtovanja optimalnih KEO digitalnih filtrov lahko gledamo kot na problem polinomske aproksimacije. Obstajajo zelo učinkoviti aproksimacijski minimax algoritmi, ki pa kot rezultat dajejo koeficiente "neskončne" natančnosti. Pri praktični uporabi filtrov je zeleno, da imajo filtri koeficiente, ki so končni oz. čim krajše dolžine. Eden od pristopov učinkovite polinomske aproksimacije je tudi uporaba dognanj iz teorije mrež (angl. lattice theory) oz. metod za redukcijo baze mreže (angl. lattice basis reduction). Preučite algoritme s tega področja, ki se lahko uporabljajo za polinomske aproksimacije v L_2 in L_∞ normi. Te algoritme implementirajte za načrtovanje optimalnih KEO filtrov s koeficienti končne dolžine in primerjajte rezultate z že obstoječimi metodami za načrtovanje.

Mentor:


prof. dr. Dušan Kodek



Dekan:


prof. dr. Nikolaj Zimic

Št.: 139-MAG-RI/2011

Datum: 27. 05. 2011



Andraž BOŽIČEK, univ. dipl. inž. rač. in inf.

Ljubljana

Fakulteta za računalništvo in informatiko Univerze v Ljubljani izdaja naslednjo magistrsko nalogo

Naslov naloge: **Uporaba algoritmov za redukcijo baze mreže pri polinomski aproksimaciji s koeficienti končne dolžine**

Application of lattice basis reduction algorithms for polynomial approximation with finite wordlength coefficients

Tematika naloge:

Na problem načrtovanja optimalnih KEO digitalnih filtrov lahko gledamo kot na problem polinomske aproksimacije. Obstajajo zelo učinkoviti aproksimacijski minimax algoritmi, ki pa kot rezultat dajejo koeficiente "neskončne" natančnosti. Pri praktični uporabi filtrov je željeno, da imajo filtri koeficiente, ki so končni oz. čim krajše dolžine. Eden od pristopov učinkovite polinomske aproksimacije je tudi uporaba dognanj iz teorije mrež (angl. lattice theory) oz. metod za redukcijo baze mreže (angl. lattice basis reduction). Preučite algoritme s tega področja, ki se lahko uporabljajo za polinomsko aproksimacijo v L_2 in L_∞ normi. Te algoritme implementirajte za načrtovanje optimalnih KEO filtrov s koeficienti končne dolžine in primerjajte rezultate z že obstoječimi metodami za načrtovanje.

Mentor:

prof. dr. Dušan Kodek



Dekan:

prof. dr. Nikolaj Zimic

IZJAVA O AVTORSTVU

magistrskega dela

Spodaj podpisani/-a **Andraž Božiček**,

z vpisno številko **63020327**,

sem avtor/-ica magistrskega dela z naslovom

**Uporaba algoritmov za redukcijo baze mreže pri polinomski aproksimaciji
s koeficienti končne dolžine**

S svojim podpisom zagotavljam, da:

- sem magistrsko delo izdelal/-a samostojno pod vodstvom mentorja (naziv, ime in priimek)

prof. dr. Dušan Kodek

in somentorstvom (naziv, ime in priimek)

-
- so elektronska oblika magistrskega dela, naslova (slov., angl.), povzetka (slov., angl.) ter ključne besede (slov., angl.) identični s tiskano obliko magistrskega dela
 - in soglašam z javno objavo elektronske oblike magistrskega dela v zbirki »Dela FRI«.

V Ljubljani, dne _____ Podpis avtorja/-ice: _____

Zahvala

Rad bi se zahvalil svojemu mentorju, prof. dr. Dušanu Kodeku, za strokovno svetovanje, potrpežljivost in spodbudo pri nastajanju magistrskega dela. Hvala tudi vsem ostalim, ki ste mi vsa ta leta stali ob strani.

Kazalo

Povzetek	1
Abstract.....	3
1. Uvod	5
2. Redukcija baze mreže.....	7
2.1. Osnovne matematične definicije.....	7
2.2. Algoritem LLL.....	9
2.3. Problem najbližjega vektorja in Babaiev algoritem.....	14
3. Načrtovanje optimalnih filtrov KEO	16
3.1. Podajanje zelenih lastnosti filtra	16
3.2. Filtri KEO z linearno fazo	18
3.3. Optimalni filtri KEO z linearno fazo	21
3.4. Načrtovanje optimalnih filtrov KEO z linearno fazo.....	22
3.4.1. Načrtovanje filtrov s koeficienti "neskončne" natančnosti.....	22
3.4.2. Obstoječe metode za načrtovanje filtrov s koeficienti končne dolžine	23
3.4.3. Polinomska aproksimacija in uporaba Babaievega algoritma in hevristike pri načrtovanju filtrov s koeficienti končne dolžine	24
3.4.4. Načrtovanje filtrov s koeficienti končne dolžine kot reševanje sistema diofantskih enačb.....	27
4. Metode za načrtovanje filtrov KEO tipa 1 s koeficienti končne dolžine.....	30
4.1. Metoda zaokroževanja	30
4.2. Metoda z uporabo celoštevilskega linearnega programiranja.....	31
4.3. Metoda z uporabo Babaievega algoritma in hevristike	32
4.4. Metoda z reševanjem sistema diofantskih enačb	35
5. Preizkus implementiranih metod načrtovanja	36
5.1. Rezultati	36

6.	Sklepne ugotovitve.....	45
7.	Priloge	47
7.1.	Seznam slik.....	47
7.2.	Seznam tabel.....	47
7.3.	Seznam grafov	47
7.4.	Izseki izvorne kode v Scilabu.....	48
7.4.1.	Algoritem LLL.....	48
7.4.2.	Babaiev algoritem	49
7.4.3.	Hevristika	50
7.4.4.	Metoda za izboljšanje LLL reducirane baze	50
8.	Seznam uporabljenih virov	52

Seznam uporabljenih kratic in simbolov

LLL – Lenstra, Lenstra, Lovasz

DSP – digitalno procesiranje signalov (angl. Digital Signal Processing)

CVP – problem najbližjega vektorja (angl. Closest Vector Problem)

GPS – globalni navigacijski sistem (angl. Global Positioning System)

MILP – celoštevilsko linearno programiranje (angl. Mixed Integer Linear Programming)

FIR – končni enotin odziv (angl. Finite Impulse Response)

KEO – končni enotin odziv

NEO – neskončni enotin odziv

Povzetek

Redukcija baze mreže je koncept za reševanje raznovrstnih problemov. Algoritme za redukcijo baze mreže srečamo danes na mnogih področjih, vključno s povsem praktičnimi aplikacijami, kot so na primer kriptografija, navigacija GPS in brezžične komunikacije.

V našem magistrskem delu smo z algoritmi za redukcijo baze mreže poskušali rešiti problem načrtovanja optimalnih digitalnih filtrov s končnim enotnim odzivom (filtrov KEO) z linearno fazo in s koeficienti končne dolžine. Problem smo najprej formulirali kot problem polinomske aproksimacije funkcije, ki predstavlja želeni frekvenčni odziv danega filtra. Kakovost aproksimacije se meri s čim manjšo maksimalno absolutno napako oz. odstopanjem od želenega frekvenčnega odziva. Pri načrtovanju smo uporabili Babaiev algoritem po metodi najbližje ravnine, ki rešuje problem najbližjega vektorja. Babaiev algoritem kot vhod dobi reducirano bazo mreže z uporabo algoritma LLL. Oba omenjena algoritma dajeta rezultate glede na L_2 normo. Optimalni koeficienti filtra so sicer rezultat polinomske aproksimacije v L_∞ normi, vendar za reševanje našega problema v L_∞ normi obstajajo le algoritmi eksponentne kompleksnosti. Zaradi tega smo uporabili algoritme, ki v polinomskem času rešijo problem v L_2 normi in se nato poskušali približati rezultatu v L_∞ normi z uporabo hevristike. Poleg polinomske aproksimacije smo problem načrtovanja filtrov formulirali še kot sistem diofantskih enačb s spodnjo in zgornjo mejo in ga poskusili rešiti z uporabo algoritma LLL.

Obe omenjeni metodi načrtovanja smo preizkusili na več nizih nizkoprepustnih in pasovnozopornih filtrov z različnimi lastnostmi frekvenčne domene. Reševanje sistema diofantskih enačb žal ni dalo rešitve, zato smo naredili le primerjavo Babaievega algoritma in hevristike z dvema obstoječima metodama, ki ne uporabljata algoritmov za redukcijo baze mreže. Prva metoda je preprosta in temelji na zaokroževanju neomejenih koeficientov filtra, izračunanih z uporabo Remezovega algoritma. Ta metoda daje suboptimalne rezultate. Druga metoda temelji na celoštevilskem linearnem programiranju, ki da vselej optimalne rezultate, vendar je počasna in včasih ne dobimo rezultatov v doglednem času.

Metoda načrtovanja filtrov z uporabo Babaievega algoritma in hevristike se je na testiranjih izkazala kot hitra in za določene nize filtrov boljša kot metoda z zaokroževanjem. V določenih primerih smo dobili optimalen rezultat, vendar žal ne vedno. Pri načrtovanju filtrov, ki imajo različno uteženo napako v posameznem pasu, se metoda z uporabo Babaievega algoritma in hevristike ni izkazala za dobro, kar je ena od večjih pomanjkljivosti

te metode. Ravno zaradi te pomanjkljivosti bi se v prihodnosti veljalo osredotočiti na uporabo algoritmov za redukcijo baze mreže pri načrtovanju optimalnih filtrov KEO z uporabo celoštevilskega linearnega programiranja. Algoritem LLL je v praksi že bil uporabljen pri določanju spodnje meje aproksimacijske napake, ki bistveno pohitri branch-and-bound algoritem.

Ključne besede: redukcija baze mreže, načrtovanje filtrov KEO, koeficienti končne dolžine, algoritem LLL, problem reševanja najbližjega vektorja

Abstract

Lattice basis reduction is a concept for solving diverse problems. Lattice basis reduction algorithms have been successfully used in many areas, including practical applications such as cryptography, GPS navigation and wireless communications.

In this master's thesis we have used lattice basis reduction algorithms to solve the problem of designing optimal linear phase finite impulse response filters (FIR filters) with finite wordlength coefficients. The problem was first formulated as the problem of minimax polynomial approximation where the function we try to approximate is represented by the desired frequency response of the filter. The quality of the approximation is measured by minimum of the maximum absolute error or deviation from the desired frequency response. We have used Babai's nearest plane algorithm in the filter design. Babai's algorithm solves the closest vector problem and it uses the basis reduced by the LLL algorithm as an input. Both LLL and Babai's algorithm give result relative to the L_2 norm. Optimal filter coefficients are given by the polynomial approximation in L_∞ norm, but only algorithms with exponential complexity are available for solving this problem in L_∞ norm. We have used algorithms which solve the problem in L_2 norm and then added heuristics which improves the results relative to L_∞ norm. We also tried a method where the problem was formulated as a system of diophantine equations with lower and upper bound which was then attempted to solve by using the LLL algorithm.

Both mentioned methods of filter design were tested on different sets of lowpass and bandstop filters with different frequency-domain specifications. Unfortunately, we didn't get any result from solving the system of diophantine equations. So we have only compared Babai's algorithm and heuristics with two existing design methods which do not use lattice basis reduction algorithms. The first method is simple and is based on rounding infinite precision coefficients obtained by Remez algorithm. This method gives suboptimal results. The second method is based on integer programming, which always gives optimal results but is slow and sometimes the problem is not solved in a reasonable amount of time.

Design method with Babai's algorithm and heuristics has been proved in tests to be fast and for almost all sets of filters better than rounding method. In certain cases we have obtained the same results as with integer programming, but unfortunately not always. The set of filters where this method was not proved to be good was the set of filters that have different weight of the error in each band. That is also the major drawback of using Babai's algorithm

and heuristics for designing optimal FIR filter with finite wordlength coefficients. In the future the research should be focused on using the lattice basis reduction algorithms with integer programming. The LLL algorithm has been already used to derive a lower bound for approximation error which increases the speed of the branch-and-bound algorithm.

Keywords: lattice basis reduction, FIR filter design, finite wordlength coefficients, LLL algorithm, closest vector problem

1. Uvod

Načrtovanje optimalnih digitalnih filtrov s končnim enotnim odzivom (v nadaljevanju filtrov KEO) lahko zastavimo kot Čebišev aproksimacijski problem [16]. Gre za kriterij pri načrtovanju, da je utežena aproksimacijska napaka med želenim in dejanskim frekvenčnim odzivom filtra enakomerno porazdeljena v prepustnem in zapornem pasu. Kriterij minimizira maksimalno absolutno napako. Obstajajo dobri minimax aproksimacijski algoritmi (primer takšnega algoritma je Remezov algoritem), ki dajejo optimalne koeficiente polinomov v L_∞ normi. Rezultat standardnih aproksimacijskih algoritmov so tako koeficienti, ki so neomejeni oz. pravimo tudi, da so "neskončne" natančnosti. Pri praktični uporabi želimo imeti (zaradi narave aplikacije ali uporabe določene strojne opreme) koeficiente, ki so krajši oz. končne dolžine (na primer dolžine 8 bitov namesto 32 bitov). Takšen primer je recimo aplikacija, ki uporablja cenejše procesorje DSP¹ s celoštevilsko aritmetiko.

Obstaja več pristopov, s katerimi dobimo koeficiente končne dolžine. Želja pri tem je, da so ti koeficienti optimalni. Najpreprostejša ideja, da bi koeficiente filtra izračunane z Remezovim algoritmom zaokrožili na željeno število bitov, se izkaže za slabo. Z uporabo tako zaokroženih koeficientov dobimo slab frekvenčni odziv [9]. Tako zaokroženi koeficienti tudi niso več optimalni. V [9] je prav tako predstavljena uporaba celoštevilskega linearnega programiranja (v nadaljevanju MILP²) kot načina za načrtovanje filtrov KEO z optimalnimi koeficienti končne dolžine. Slabost postopka z uporabo MILP je počasnost, zato rezultatov večkrat ne dobimo v doglednem času. Pristop, ki bistveno pohitri izračun omejenih koeficientov je predstavljen v [11]. S poznavanjem spodnje meje aproksimacijske napake zmanjšamo število podproblemov, ki jih moramo izračunati in na ta način se tudi pohitri izračun optimalnih koeficientov končne dolžine.

Problema reševanja polinomske aproksimacije so se lotili tudi s pristopi, ki temeljijo na postopku redukcije baze mreže oz. temeljujoč na algoritmu LLL³ [4]. Algoritem LLL sodi na področje teorije mrež (angl. lattice theory) in nosi ime po svojih treh avtorjih. Formalen opis algoritma najdemo v [12], medtem ko je implementacija algoritma skupaj s psevdokodo bolj nazorno pojasnjena v [18]. Cilj algoritma LLL je zmanjšati bazo mreže, tako da je mreža enakovredno določena s krajšimi in skoraj ortogonalnimi vektorji. Algoritem se uporablja na

¹ digitalno procesiranje signalov (angl. Digital Signal Processing)

² celoštevilsko linearno programiranje (angl. Mixed Integer Linear Programming)

³ algoritem Lenstra-Lenstra-Lovasz

mного področjih. Praktična uporaba algoritma zajema področje brezžičnih komunikacij, sistemov GPS⁴ in predvsem kriptografije.

Kot že omenjeno, sega algoritem LLL tudi na področje aproksimacije. Na problem aproksimacije lahko gledamo kot na problem najkrajšega vektorja [6]. Ta problem glede na L_2 normo rešuje Babaiev algoritem, ki kot vhod dobi bazo reducirano z uporabo algoritma LLL. Formalen opis algoritma je v [3]. S pomočjo prej omenjenega algoritma in hevrističnih metod so v [4] učinkovito aproksimirali funkcije v L_∞ normi.

V [2] so se lotili reševanja sistema diofantskih enačb z uporabo algoritma LLL. Tudi načrtovanje optimalnih filtrov KEO z linearno fazo in koeficienti končne dolžine lahko zastavimo kot reševanje sistema diofantskih enačb s spodnjo in zgornjo mejo.

V magistrskem delu sta v poglavju 2 predstavljena algoritem LLL za redukcijo baze mreže in Babaiev algoritem za reševanje problema najkrajšega vektorja. Predstavljena je tudi psevdokoda, ki je bila uporabljena pri implementaciji. V poglavju 3 smo opisali načrtovanje optimalnih filtrov KEO z Remezovim algoritmom, ki daje koeficiente "neskončne" natančnosti in prav tako metode, ki se uporabljajo za načrtovanje filtrov končne dolžine. To sta metodi z zaokroževanjem in z uporabo MILP. Polega tega sta opisani tudi metodi, ki uporabljata algoritme za redukcijo baze mreže. V naslednjem, 4. poglavju, je predstavljena implementacija vseh metod pri načrtovanju optimalnih filtrov KEO tipa 1. Rezultati testiranja in primerjave metod so predstavljeni v poglavju 5. V 6. poglavju so strnjene sklepne ugotovitve in razmislek glede nadaljnjih raziskav. Zadnje, 7. poglavje, je namenjeno kazalu slik, tabel in grafov ter izsekom izvirne kode.

⁴ globalni navigacijski sistem (angl. Global Positioning System)

2. Redukcija baze mreže

2.1. Osnovne matematične definicije

Osnovne matematične definicije so delno povzete po [8].

Definicija 2.1. \mathbb{R}^n označuje n -dimenzionalni realni vektorski prostor s skalarnim produktom $\langle \cdot, \cdot \rangle: \mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R}$. Za vse vektorje $\vec{u}, \vec{v}, \vec{w} \in \mathbb{R}^n$ in $\lambda \in \mathbb{R}$ ima skalarni produkt naslednje lastnosti:

- $\langle \vec{u} + \vec{w}, \vec{v} \rangle = \langle \vec{u}, \vec{v} \rangle + \langle \vec{w}, \vec{v} \rangle$,
- $\langle \lambda \vec{u}, \vec{v} \rangle = \lambda \langle \vec{u}, \vec{v} \rangle$,
- $\langle \vec{u}, \vec{v} + \vec{w} \rangle = \langle \vec{u}, \vec{v} \rangle + \langle \vec{u}, \vec{w} \rangle$,
- $\langle \vec{u}, \lambda \vec{v} \rangle = \lambda \langle \vec{u}, \vec{v} \rangle$,
- $\langle \vec{u}, \vec{v} \rangle = \langle \vec{v}, \vec{u} \rangle$,
- $\langle \vec{u}, \vec{u} \rangle \geq 0$, za $\vec{u} \neq 0$.

Definicija 2.2. Preslikavi $\|\cdot\|: \mathbb{R}^n \rightarrow \mathbb{R}$ pravimo norma, če za vse $\vec{u}, \vec{v} \in \mathbb{R}^n$ in $\lambda \in \mathbb{R}$ velja:

- $\|\lambda \vec{v}\| = |\lambda| \cdot \|\vec{v}\|$,
- $\|\vec{u} + \vec{v}\| \leq \|\vec{u}\| + \|\vec{v}\|$,
- $\|\vec{u}\| \geq 0$, za $u \neq 0$.

Definicija 2.3. Za vektor $\vec{u} = (u_1, u_1, \dots, u_n)^T$ lahko definiramo naslednje norme:

- norma L_1 ali manhattanska razdalja: $\|\vec{u}\|_1 = \sum_{i=1}^n |u_i|$,
- norma L_2 ali Evklidska razdalja: $\|\vec{u}\|_2 = \sqrt{\langle \vec{u}, \vec{u} \rangle} = \sqrt{\sum_{i=1}^n u_i^2}$,
- norma L_∞ ali Čebiševa razdalja: $\|\vec{u}\|_\infty = \max_{i=1,2,\dots,n} |u_i|$,
- norma L_p : $\|\vec{u}\|_p = \sqrt[p]{\langle \vec{u}, \vec{u} \rangle} = \sqrt[p]{(\sum_{i=1}^n |u_i|^p)}$.

Med normami veljajo naslednja razmerja:

- $\|\vec{u}\|_2 \leq \|\vec{u}\|_1 \leq \sqrt{n} \|\vec{u}\|_2$,
- $\|\vec{u}\|_\infty \leq \|\vec{u}\|_2 \leq n \|\vec{u}\|_\infty$.

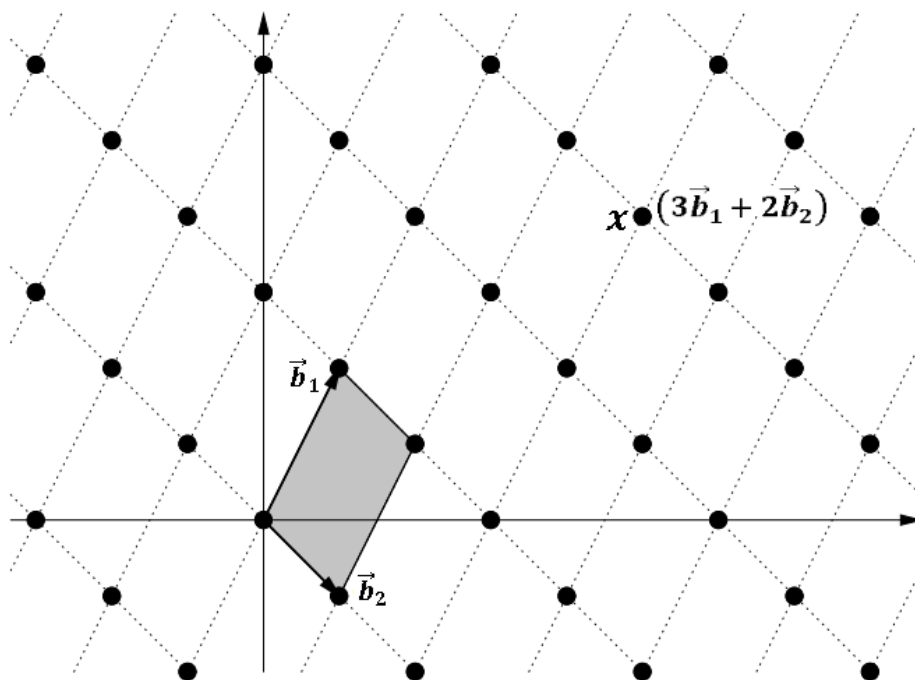
Niz l linearno neodvisnih vektorjev $\vec{b}_1, \vec{b}_2, \dots, \vec{b}_l \in \mathbb{R}^n$, $l \leq n$, predstavimo z matriko B . Stolpci v matriki B predstavljajo vektorje $\vec{b}_1, \vec{b}_2, \dots, \vec{b}_l$.

Definicija 2.4. Mreža⁵ L vpeta z vektorji $\vec{b}_1, \vec{b}_2, \dots, \vec{b}_l$ je definirana kot linearna kombinacija vektorjev $\vec{b}_1, \vec{b}_2, \dots, \vec{b}_l$

$$L = \{x: x = \sum_{j=1}^l \alpha_j \vec{b}_j, \alpha_j \in \mathbb{Z}, 1 \leq j \leq l\}. \quad (1)$$

Niz vektorjev $\vec{b}_1, \vec{b}_2, \dots, \vec{b}_l$ imenujemo baza mreže.

Na sliki 1 je prikazana dvodimenzionalna mreža, ki jo vpenjata vektorja \vec{b}_1 in \vec{b}_2 .



Slika 1: Prikaz dvodimenzionalne mreže. Mrežo določata vektorja \vec{b}_1 in \vec{b}_2 . Ena od kombinacij vektorjev, ki nas iz točke 0 pripelje do točke x , je $(3\vec{b}_1 + 2\vec{b}_2)$.

Definicija 2.5. Celoštevilski nesingularni matriki U z $\det(U) = \pm 1$ pravimo unimodularna matrika.

Naslednjim operacijam nad matriko pravimo elementarne stolpčne operacije:

- menjava dveh stolpcev,
- množenje stolpca z -1 ,
- prištetje stolpca pomnoženega s celim številom nekemu drugemu stolpcu matrike.

⁵ angl. lattice

Celoštevilna matrika U je unimodularna, če in samo če lahko matriko U izpeljemo iz enotske matrike z uporabo elementarnih stolpčnih operacij.

Mreža lahko ima več različnih baz.

Če sta B in B' bazi iste mreže L , potem velja $B' = BU$, za neko unimodularno matriko U velikosti $l \times l$.

2.2. Algoritem LLL

Algoritem LLL nosi ime po svojih treh avtorjih. Leta 1982 so Arjen Lenstra, Hendrik Lenstra and László Lovász v članku [12] predstavili algoritem, ki v polinomskem času razstavi polinome z racionalnimi koeficienti na produkt nerazcepnih faktorjev. Kasneje se je izkazalo, da je uporabnost algoritma mnogo večja. Algoritem LLL je uporaben v teoriji mrež za redukcijo baze mreže. Aplikacije (kot je tudi omenjeno v [19]) najdemo na raznovrstnih področjih kot so kriptografija in kriptanaliza, geometrija števil, diofantske aproksimacije in nenazadnje tudi na področju satelitskega pozicioniranja (GPS) in brezžičnih komunikacij.

Opis algoritma LLL je povzet po [1]. Algoritem LLL z uporabo niza elementarnih stolpčnih operacij na prvotni bazi B za določeno mrežo L tvori t.i. reducirano bazo B' z vektorji $\vec{b}'_1, \dots, \vec{b}'_l$. Pojem reducirana baza tako v nadaljevanju označuje reducirano bazo pridobljeno z algoritmom LLL. Vektorji $\vec{b}'_1, \dots, \vec{b}'_l$ so kratki in skoraj ortogonalni, vektor \vec{b}'_1 pa je aproksimacija najkrajšega vektorja v mreži. Na podani bazi B pridobimo ortogonalne vektorje z uporabo Gram-Schmidtovega postopka ortogonalizacije. Gram-Schmidtovi vektorji ne pripadajo nujno bazi, vendar napenjajo enak vektorski prostor $\vec{b}'_1, \dots, \vec{b}'_l$. Uporabimo jih lahko kot referenčne vektorje za algoritem LLL.

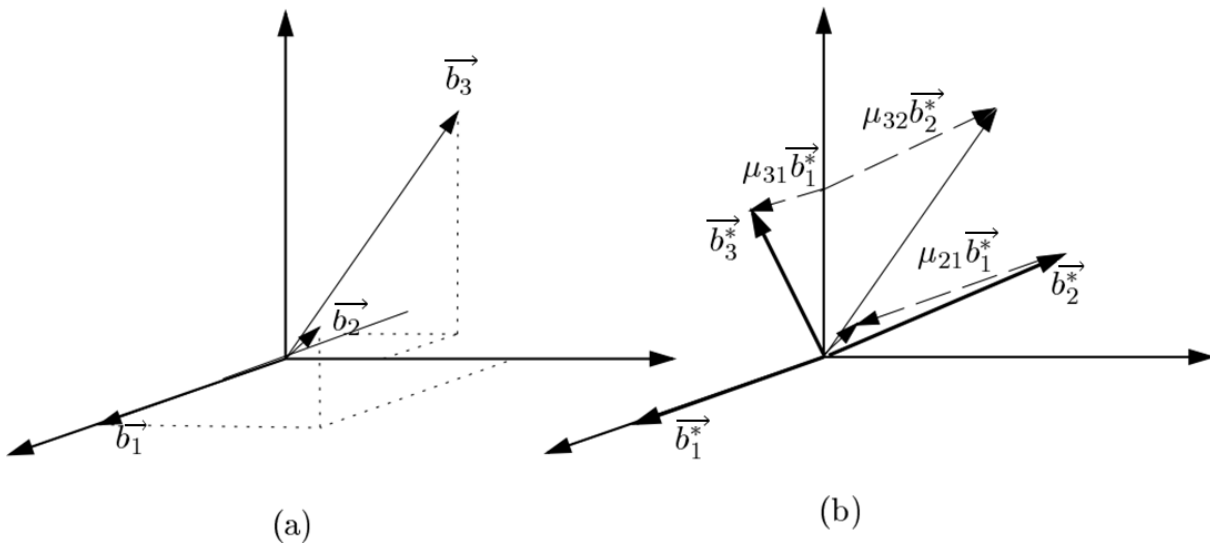
Definicija 2.6. Gram-Schmidtov postopek neodvisnim vektorjem $\vec{b}_j, 1 \leq j < l$, priredi ortogonalne vektorje $\vec{b}_j^*, 1 \leq j < l$. Vektorje $\vec{b}_j^*, 1 \leq j < l$, in realna števila $\mu_{j,k}, 1 \leq k < j < l$, pridobimo s pomočjo rekurzije na naslednji način:

$$\vec{b}_1^* = \vec{b}_1, \tag{2}$$

$$\vec{b}_j^* = \vec{b}_j - \sum_{k=1}^{j-1} \mu_{jk} \vec{b}_k^* \text{ za } 2 \leq j \leq n, \tag{3}$$

$$\mu_{j,k} = \frac{\langle \vec{b}_j^T, \vec{b}_k^* \rangle}{\langle \vec{b}_k^*, \vec{b}_k^* \rangle} \text{ za } 1 \leq k < j < l. \quad (4)$$

Vektor \vec{b}_j^* je projekcija vektorja \vec{b}_j na ortogonalni komplement $\sum_{k=1}^{j-1} \mathbb{R}\vec{b}_k$. Katerikoli par Gram-Schmidtovih vektorjev \vec{b}_i^*, \vec{b}_k^* je medsebojno ortogonalen. Množitelj μ_{jk} podaja dolžino komponente vektorja \vec{b}_j v smeri \vec{b}_k^* . Množitelj μ_{jk} je enak 0, če in samo če je \vec{b}_j ortogonalen \vec{b}_k^* . Praktičen primer postopka Gram-Schmidtove ortogonalizacije za bazo s tremi vektorji je prikazan na sliki 2.



Slika 2: Primer Gram-Schmidtovega postopka ortogonalizacije. Na sliki (a) je so predstavljeni prvotni vektorji $\vec{b}_1, \vec{b}_2, \vec{b}_3$, slika (b) pa predstavlja vektorje \vec{b}_1^*, \vec{b}_2^* in \vec{b}_3^* po končanem postopku Gram-Schmidtove ortogonalizacije.

Definicija 2.7. Baza vektorjev $\vec{b}_1, \vec{b}_2, \dots, \vec{b}_l$ je LLL reducirana, če velja naslednje:

$$|\mu_{j,k}| \leq \frac{1}{2} \text{ za } 1 \leq k < j \leq l, \quad (5)$$

$$\|\vec{b}_j^* + \mu_{j,j-1}\vec{b}_{j-1}^*\|_2^2 \geq \sigma \|\vec{b}_{j-1}^*\|_2^2 \text{ za } i < j < l. \quad (6)$$

Pogoj (5) velja, če je komponenta vektorja \vec{b}_j v smeri vektorja \vec{b}_k^* relativno kratka glede na ta vektor ali če sta vektorja \vec{b}_j^* in \vec{b}_k^* skoraj ortogonalna. V primeru, da ta pogoj ne velja, potem algoritem LLL nadomesti \vec{b}_j^* z $\vec{b}_j^* - [\mu_{j,k}]\vec{b}_k^*$. V tem primeru definiramo $[a]$ kot $[a] = \left\lfloor a - \frac{1}{2} \right\rfloor$. $[a]$ predstavlja torej najbližje celo število k številu a . Temu koraku v algoritmu pravimo redukcija velikosti in zagotavlja, da so rezultat relativno kratki bazni vektorji. Če je izpolnjen le pogoj (5), se lahko zgodi, da vektorji niso medsebojno ortogonalni. Prav tako se lahko zgodi, da je prvi vektor zelo dolg, ostali pa relativno kratki glede na prvi vektor. Temu se izognemo z upoštevanjem pogoja (6).

Pogoj (6) lahko z upoštevanjem, da sta vektorja \vec{b}_{j-1}^* in \vec{b}_j^* ortogonalna, prepisemo v

$$\|\vec{b}_j^*\|_2^2 \geq (\sigma - \mu_{j,j-1}^2) \|\vec{b}_{j-1}^*\|_2^2 \geq (\sigma - \frac{1}{4}) \|\vec{b}_{j-1}^*\|_2^2. \quad (7)$$

Če je pogoj izpolnjen, lahko rečemo, da vektor \vec{b}_j^* ni mnogo krajši kot \vec{b}_{j-1}^* . V primeru, da je pogoj (6) izpolnjen, algoritem LLL vektorja \vec{b}_j in \vec{b}_{j-1} med seboj zamenja.

Konstanta σ v neenačbi (6) je lahko poljubno realno število, ki zadostuje pogoju $\frac{1}{4} < \sigma < 1$. V praksi se največkrat uporablja $\sigma = \frac{3}{4}$, ta vrednost za σ je bila uporabljena tudi v [I2]. Isto vrednost za σ smo uporabili tudi mi.

Celoten opis algoritma lahko strnemo v spodnjo psevdokodo [I8]. V glavni zanki programa kličemo proceduri A in B. Procedura A izvede redukcijo velikosti vektorja, procedura B pa izvede zamenjavo dveh vektorjev.

Algoritem LLL

Opis: Za bazo B , ki jo dobi na vhodu, izračuna reducirano bazo ($\sigma = \frac{3}{4}$).

Vhod: Matrika B , katere stolpci predstavljajo bazo mreže L .

Izhod: Matrika B , katere stolpci predstavljajo reducirano bazo mreže L .

Izračunaj $\mu_{i,j}$ in \vec{b}_i^* z uporabo Gram-Schmidtovega postopka.

Za $i = 1$ do n , naredi $\mathfrak{B}_i = \langle \vec{b}_i^*, \vec{b}_i^* \rangle$.

$k = 2$.

Ponavljaj:

Izvrši proceduro A za $l = k - 1$.

Če $\mathfrak{B}_k < (\frac{3}{4} - \mu_{k,k-1}^2) \mathfrak{B}_{k-1}$ potem

Izvrši proceduro B.

Če $k < 2$, potem $k = k - 1$.

Sicer

Za $l = k - 2$ do 1 izvrši proceduro A.

$k = k + 1$.

Konec.

Dokler ($k > n$).

Procedura A

Opis: Poskrbi, da $\mu_{k,l}$ zadosti pogoju $|\mu_{k,l}| \leq \frac{1}{2}$.

Vhod: Celi števili k, l ; kvadratni matriki B, U dimenzij $n \times n$.

Izhod: vrne iste spremenljivke kot jih dobimo na vhodu.

Če $(|\mu_{k,l}| > \frac{1}{2})$, potem:

$$r = [\mu_{k,l}].$$

$$\vec{b}_k = \vec{b}_k - r\vec{b}_l.$$

Za $j = 1$ do $l - 1$, naredi $\mu_{k,j} = \mu_{k,j} - r\mu_{l,j}$.

$$\mu_{k,l} = \mu_{k,l} - r.$$

Konec.

Po izhodu iz procedure A imamo izpolnjen pogoj (5).

Procedura B

Opis: Zamenjava \vec{b}_k in \vec{b}_{k-1} .

Vhod: Celo število k ; kvadratni matriki B, U velikosti $n \times n$; vektor \mathfrak{B} .

Izhod: vrne iste spremenljivke kot jih dobimo na vhodu.

Postavi: $u = \mu_{k,k-1}$.

$$\mathfrak{B} = \mathfrak{B}_k + u^2\mathfrak{B}_{k-1}.$$

$$\mu_{k,k-1} = u\mathfrak{B}_{k-1}/\mathcal{B}.$$

$$\mathfrak{B}_k = \mathfrak{B}_{k-1}\mathfrak{B}_k/\mathcal{B}.$$

$$\mathfrak{B}_{k-1} = \mathcal{B}.$$

Zamenjaj vektorja \vec{b}_{k-1} in \vec{b}_k .

Za $j = 1$ do $k - 2$ zamenjaj elementa $\mu_{k-1,j}$ in $\mu_{k,j}$.

Za $i = k + 1$ do n :

$$t = \mu_{i,k-1}.$$

$$\mu_{i,k-1} = \mu_{k,k-1}\mu_{i,k-1} + \mu_{i,k} - \mu_{i,k}\mu_{k,k-1}u.$$

$$\mu_{i,k} = t - u\mu_{i,k}.$$

Konec.

Po izhodu iz procedure B imamo izpolnjen pogoj (6).

Trditev 2.1. : Algoritem LLL se konča v polinomskem času.

Dokaz. V vsaki iteraciji algoritma LLL uporabi proceduri A in B. Obe proceduri se izvršita v polinomskem času. Pokazati je potrebno, da je število ponavljanj obeh procedur polinomsko.

Definiramo naslednjo potenčno funkcijo

$$D(\vec{b}_1, \dots, \vec{b}_n) = \prod_{i=1}^n \|\vec{b}_i^*\|_2^{n-i}. \quad (8)$$

Očitno je, da procedura A ne spremeni vrednosti D , ker ne spreminja Gram-Schmidtove baze vektorjev. V vsaki iteraciji procedure B se zmanjša D za konstanten faktor. V tej proceduri zamenjamo vektorja \vec{b}_k in \vec{b}_{k-1} , če pogoj (6) ni izpolnjen, oz. če velja $\|\vec{b}_{k-1}\|_2^2 \geq \left(\frac{1}{\sigma-1/4}\right) \|\vec{b}_k^*\|_2^2$. To v primeru, da je $\sigma = \frac{3}{4}$ pomeni, da vsaka menjava vektorjev zmanjša D za vsaj faktor 2.

Potrebno je še pokazati, da ima D zgornjo in spodnjo mejo. Če upoštevamo, da je $\|\vec{b}_i^*\|_2 \leq \|\vec{b}_i\|_2$, potem je zgornja meja za D enaka $(\max_i \|\vec{b}_i\|_2)^{\frac{n(n-1)}{2}}$. D lahko zapišemo kot $\prod_{i=1}^n |\det(\Lambda_i)|$, kjer Λ_i predstavlja mrežo vpeto z vektorji $\vec{b}_1, \dots, \vec{b}_i$. Če predvidimo, da so bazni vektorji mreže cela števila, je vrednost D vsaj 1.

Trditev 2.2. : Če je $\vec{b}_1, \dots, \vec{b}_n$ reducirana baza, potem $\|\vec{b}_1\|_2 \leq 2^{n-\frac{1}{2}} \lambda(L)$, kjer je L dolžina najkrajšega vektorja.

Dokaz. Velja ($\sigma = \frac{3}{4}$)

$$\|\vec{b}_{i+1}^*\|_2^2 \geq \left(\frac{3}{4} - \mu_{i+1,i}^2\right) \|\vec{b}_i^*\|_2^2 \geq \frac{1}{2} \|\vec{b}_i^*\|_2^2. \quad (9)$$

Z indukcijo po i , dobimo

$$\|\vec{b}_i^*\|_2^2 \geq \frac{1}{2^{i-1}} \|\vec{b}_1^*\|_2^2 = \frac{1}{2^{i-1}} \|\vec{b}_1\|_2^2. \quad (10)$$

Velja: $\forall b \in L, \|\vec{b}\|_2 > \min_i \|\vec{b}_i^*\|_2$. Zato $\lambda(L) \geq \min_i \|\vec{b}_i^*\|_2$, kar skupaj z neenačbo (10) da zeleno neenačbo

$$\|\vec{b}_1\|_2^2 \leq \min_i \{2^{i-1} \|\vec{b}_i^*\|_2^2\} \leq 2^{n-1} \min_i \{\|\vec{b}_i^*\|_2^2\} \leq 2^{n-1} \lambda(L)^2. \quad (11)$$

2.3. Problem najbližjega vektorja in Babaiev algoritem

Problem najbližjega vektorja (v nadaljevanju CVP⁶) je računski problem. Podana je matrika B , ki določa bazo mreže L , in vektor $\vec{w} \in \mathbb{R}^n$. Želimo izračunati vektor $\vec{v} \in L$, tako da je $\|\vec{w} - \vec{v}\|_2$ minimalno.

Kot je navedeno v [4], je leta 1981 van Emde Boas dokazal, da je CVP NP-poln problem. Natančno rešitev CVP lahko dobimo z uporabo Kannanovega algoritma, vendar je zahtevnost algoritma super-eksponentna. V praksi se zato največkrat uporablja Babaiev algoritem po metodi najbližje ravnine (angl. Babai nearest plane method) [3], ki ima polinomsko zahtevnost, vendar aproksimira CVP z eksponentnim faktorjem. Žal trenutno še ne obstaja algoritem polinomske zahtevnosti, ki bi CVP aproksimiral s polinomskim aproksimacijskim faktorjem.

Psevdokoda Babaievega algoritma je prikazana spodaj in je povzeta po [7].

Babaiev algoritem

Vhod: LLL reducirana baza B z vektorji $\vec{b}_1, \dots, \vec{b}_n$; vektorji $\vec{b}_1^*, \dots, \vec{b}_n^*$ izračunani z Gram-Schmidtovim postopkom; vektor \vec{v}

Rezultat: aproksimacija najbližjega vektorja vektorju \vec{v}

$\vec{t} = \vec{v}$;

Za ($j = n$; $j \geq 1$; $j - -$) naredi:

$$c = \left\lfloor \frac{\langle \vec{t}, \vec{b}_j^* \rangle}{\langle \vec{b}_j^*, \vec{b}_j^* \rangle} \right\rfloor.$$

$$\vec{t} = \vec{t} - c\vec{b}_j.$$

Konec.

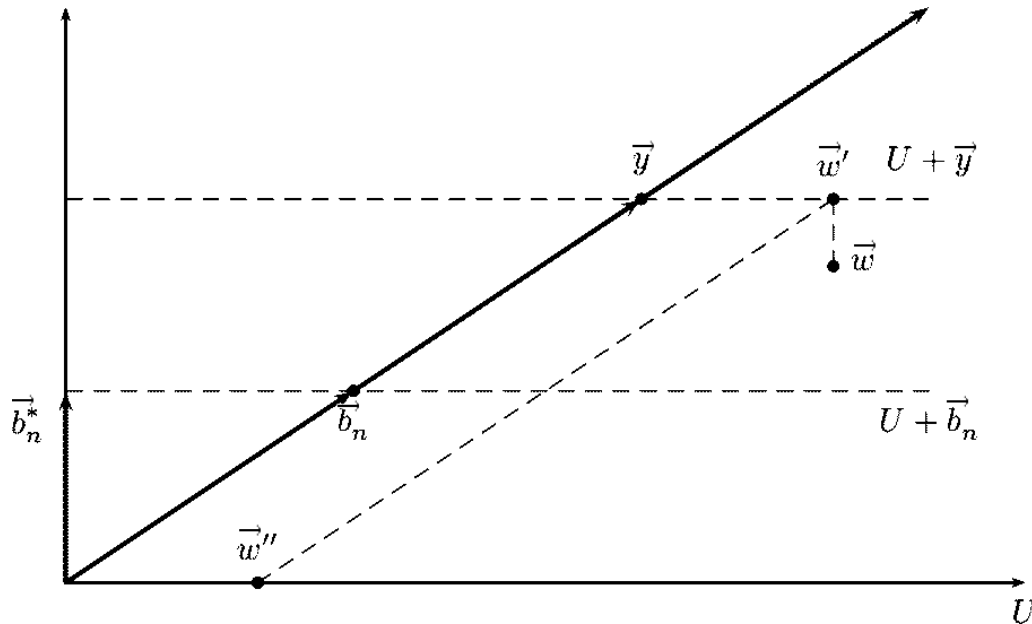
Vrni $\vec{v} - \vec{t}$.

Čeprav Babaiev algoritem deluje tudi z bazo, ki ni reducirana, se v praksi izkaže, da so rezultati boljši, če je le-ta reducirana [7]. Glavni korak algoritma lahko grobo opišemo kot iskanje celoštevilске kombinacije baznih vektorjev $\vec{b}_1, \dots, \vec{b}_n$, ki so čim bližje iskanemu vektorju \vec{v} .

Za lažjo predstavo je na sliki 3 prikazano delovanje Babaievega algoritma. U je definiran kot podprostor določen z vpetjem vektorjev $\vec{b}_1, \dots, \vec{b}_{n-1}$, $L' = L \cap U$ je podmreža

⁶ problem najbližjega vektorja (angl. Closest Vector Problem)

določena z vektorji $\vec{b}_1, \dots, \vec{b}_{n-1}$. Ideja metode najbližje ravnine je najti vektor $\vec{y} \in L$, tako da je razdalja med vektorjem \vec{w} in ravnino $U + \vec{y}$ minimalna. Določimo \vec{w}' kot pravokotno projekcijo vektorja \vec{w} na ravnino $U + \vec{y}$ in $(\vec{w}'' = \vec{w}' - \vec{y}) \in U$. CVP rešujemo naprej induktivno na vektorjih \vec{w}'' in L' , da dobimo $\vec{y}' \in L'$. V vsakem koraku se nam dimenzija zmanjša za 1.



Slika 3: Prikaz Babaijevega algoritma za reševanje CVP. Os x predstavlja podprostor U (dimenzije $n-1$), y os je pravokotna glede na U .

Trditev 2.3. Če je baza $\vec{b}_1, \dots, \vec{b}_n$ LLL reducirana s faktorjem $\sigma = \frac{3}{4}$, potem je izhod Babaijevega algoritma za $\vec{w} \in \mathbb{R}^n$ vektor \vec{v} , tako da za vse $\vec{u} \in L$ velja

$$\|\vec{v} - \vec{w}\|_2 < 2^{\frac{n}{2}} \|\vec{u} - \vec{w}\|_2. \quad (12)$$

Zgornja meja za faktor aproksimacije kot ga podaja trditev 1.3, ne daje dobrih pričakovanj pri uporabi Babaijevega algoritma. V praksi se izkaže (kot je navedeno tudi v [4]), da je omenjena meja zastavljena precej pesimistično, saj je glede na L_2 normo vektor \vec{v} kot rezultat algoritma bližje danemu vektorju \vec{w} .

3. Načrtovanje optimalnih filtrov KEO

Na področju digitalnega procesiranja signalov je filter naprava ali proces, ki signalu odstrani neželene komponente ali značilke. Največkrat se s pomočjo filtrov iz signala odstranijo določene frekvence ali šum [5].

Filte lahko razdelimo na mnogo načinov. Ena od delitev je delitev na filtre s končnim enotnim odzivom (v nadaljevanju KEO) in filtre z neskončnim enotnim odzivom (v nadaljevanju NEO). Filtri NEO so sorodni analognim filtrom in kot ime pove odziv na enotin impulz zamre šele po neskončnem času. Vsak izhodni vzorec filtra NEO je odvisen od prejšnjih vhodnih in prejšnjih izhodnih vzorcev. Izhodni vzorec filtra KEO je odvisen le od prejšnjih vhodnih vzorcev. Filtri KEO so v praksi posebno uporabni pri problemih, kjer je zahteva po linearni fazi.

Teorijo glede filtrov KEO smo povzeli po [16].

3.1. Podajanje zelenih lastnosti filtra

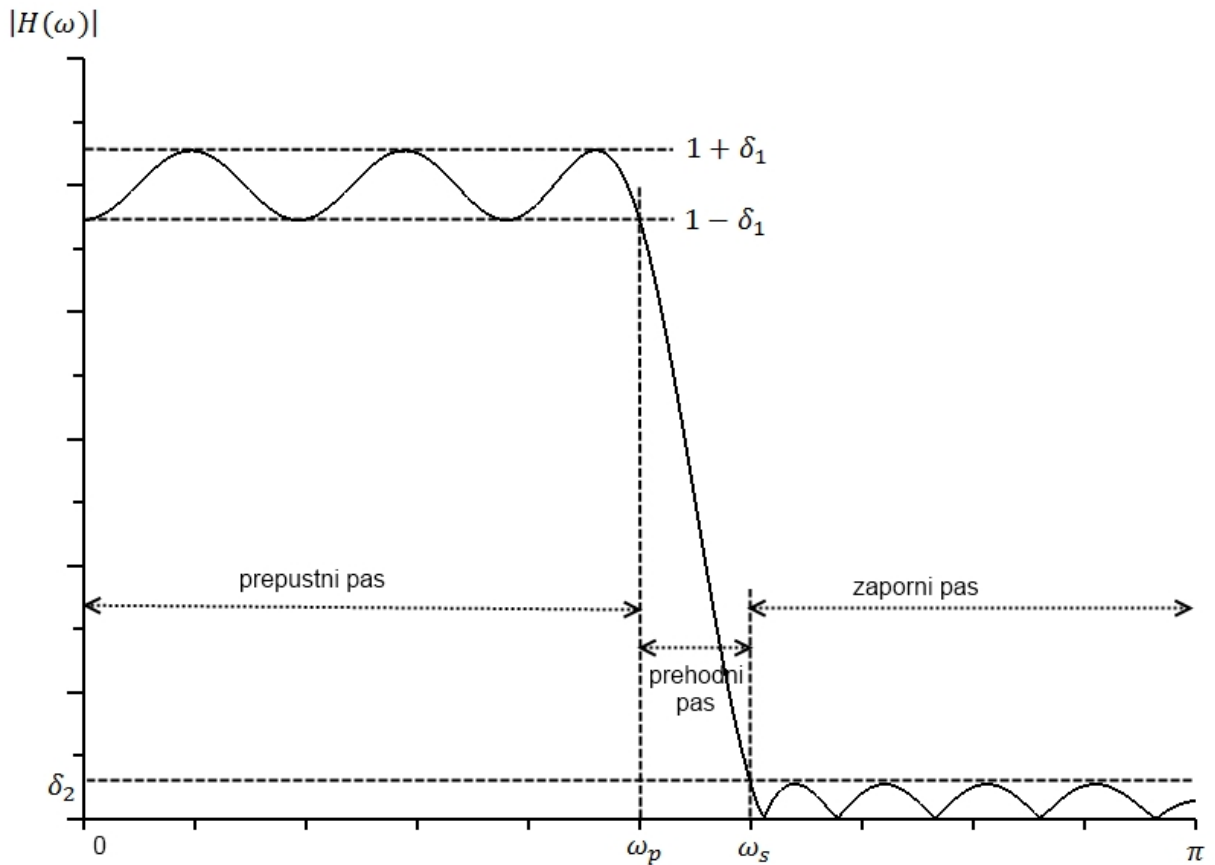
Želene lastnosti filtra so podane v frekvenčni domeni v smislu zelenega amplitudnega in faznega odziva filtra. Amplitudni odziv filtra si lahko pogledamo za primer nizkoprepustnega filtra. Če bi želeli imeti idealni nizkoprepustni filter, bi amplitudni odziv filtra opisali z enačbo:

$$|H_{dr}(\omega)| = \begin{cases} 1, & 0 \leq \omega \leq \omega_p \\ 0, & \omega_s \leq \omega \leq \pi \end{cases} \quad (13)$$

Tako definirani idealni nizkoprepustni filter bi bil nekavzalen in ga ne bi bilo mogoče realizirati. V praksi moramo zeleni amplitudni odziv definirati malo bolj ohlapno. Enačbo (13) preoblikujemo, tako da dobimo dejanski amplitudni odziv filtra:

$$\begin{aligned} 1 - \delta_1 &\leq |H_r(\omega)| \leq 1 + \delta_1, & 0 \leq \omega \leq \omega_p, \\ |H_r(\omega)| &\leq \delta_2, & \omega_s \leq \omega \leq \pi. \end{aligned} \quad (14)$$

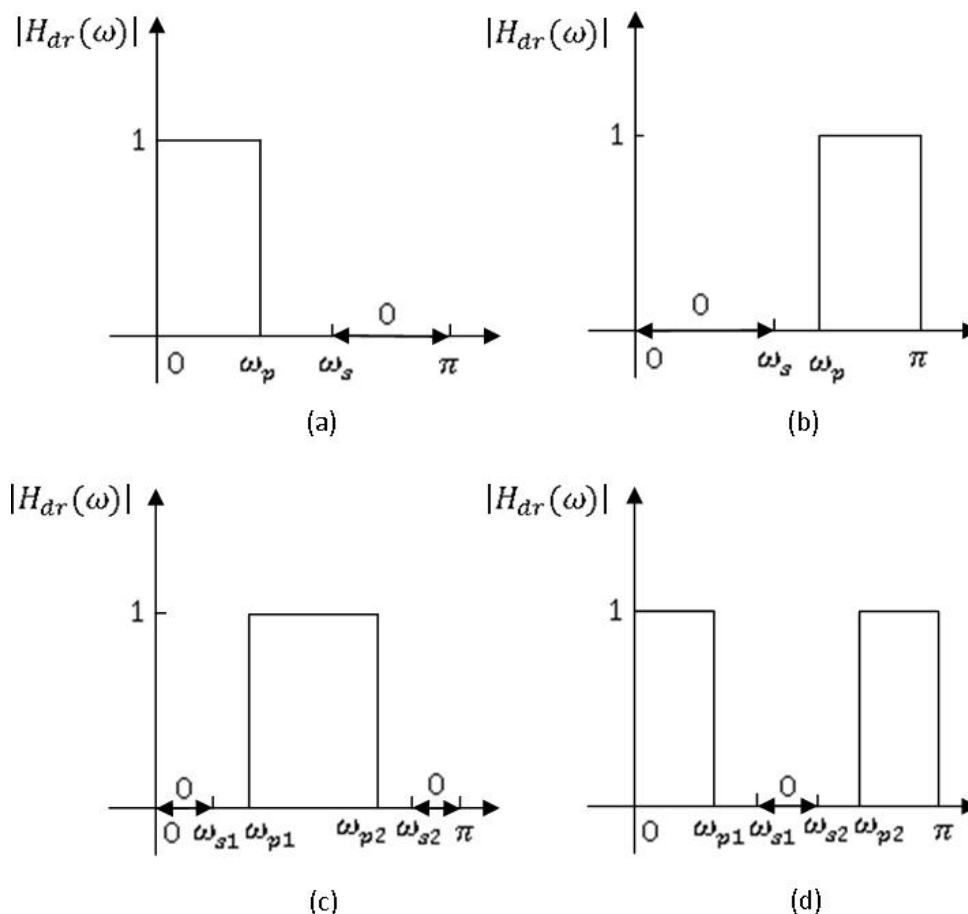
Na sliki 4 je prikazan dejanski amplitudni odziv za primer nizkoprepustnega filtra.



Slika 4: Amplitudni odziv za primer nizkoprepustnega filtra.

Prepustni pas sega od frekvenc 0 do ω_p , zaporni pas od frekvenc ω_s do π . Med prepustnim in zapornim pasom je prehodni pas. Napaka v prepustnem pasu je označena z δ_1 in napaka v zapornem pasu z δ_2 . Ponavadi za predstavitev grafa frekvenčnega odziva filtra uporabimo logaritmično merilo, tako da je amplitudni odziv prikazan v decibelih (dB). Napaka v prepustnem pasu je tako $20\log_{10}(1 + \delta_1)$ dB in napaka v zapornem pasu $-20\log_{10}(\delta_2)$ dB [14].

Poleg nizkoprepustnih filtrov se v praksi najpogosteje uporabljajo še viskoprepustni, pasovnoprepustni in pasovnozaporni filtri. Na sliki 5 je prikazan idealni amplitudni odziv za vse štiri najbolj pogosto uporabljene filtre.



Slika 5: Prikaz želenega (idealnega) amplitudnega odziva za štiri najbolj pogoste vrste filtrov: (a) visokoprepustni filter, (b) nizkoprepustni filter, (c) pasovnoprepustni filter, (d) pasovnozaporni filter.

3.2. Filtri KEO z linearno fazo

Večina filtrov KEO je filtrov z linearno fazo. Če želimo zasnovati filter z linearno fazo, običajno uporabimo filter KEO. Lastnost linearne faze se nanaša na pogoj, da je fazni odziv filtra linearna funkcija frekvence. Rezultat tega je, da je zakasnitev skozi filter enaka za vse frekvence. Takšen filter ne povzroča faznega ali zakasnitvenega popačenja, kar je pomembna prednost filtrov KEO pred filtri NEO. Ker je pri filtrih KEO z linearno fazo faza vnaprej določena, s frekvenčnim odzivom filtra mislimo na amplitudni odziv filtra.

Filtri KEO so običajno zasnovani, da imajo linearno fazo (čeprav to ni nujno). Pri filtrih KEO z linearno fazo so koeficienti vedno simetrični okoli nekega osrednjega koeficienta. Tako je npr. prvi koeficient filtra enak zadnjemu itd. Pri filtrih KEO z linearno fazo imamo 4 različne tipe filtrov glede na simetrijo odziva na enotin impulz in glede na to, ali imamo sodo ali liho število koeficientov filtra. Predstavitev vseh 4 tipov:

- **tip 1, M liho število, simetričen odziv na enotin impulz $h(n) = h(M - 1 - n)$**

Realni frekvenčni odziv $H_r(\omega)$ v tem primeru izračunamo kot

$$H_r(\omega) = h\left(\frac{M-1}{2}\right) + 2 \sum_{n=0}^{\frac{M-3}{2}} h(n) \cos \omega\left(\frac{M-1}{2} - n\right). \quad (15)$$

Če je $k = \frac{M-1}{2} - n$, lahko definiramo nov niz parametrov filtra $a(k)$ kot

$$a(k) = \begin{cases} h\left(\frac{M-1}{2}\right), & k = 0 \\ 2h\left(\frac{M-1}{2} - k\right), & k = 1, 2, \dots, \frac{M-1}{2} \end{cases} \quad (16)$$

in dobimo bolj pregledno obliko za računanje frekvenčnega odziva

$$H_r(\omega) = \sum_{k=0}^{\frac{M-1}{2}} a(k) \cos \omega k. \quad (17)$$

- **tip 2, M sodo število, simetričen odziv na enotin impulz $h(n) = h(M - 1 - n)$**

Realni frekvenčni odziv $H_r(\omega)$ v tem primeru izračunamo kot

$$H_r(\omega) = 2 \sum_{n=0}^{\frac{M}{2}-1} h(n) \cos \omega\left(\frac{M-1}{2} - n\right). \quad (18)$$

Indeks n nadomestimo s k , tako da velja $k = \frac{m}{2} - n$ in definiramo nov niz parametrov filtra $b(k)$

$$b(k) = 2h\left(\frac{M}{2} - k\right), \quad k = 1, 2, \dots, \frac{M}{2}. \quad (19)$$

Iz formule (18) lahko izpeljemo

$$H_r(\omega) = \sum_{k=1}^{\frac{M}{2}} b(k) \cos \omega\left(k - \frac{1}{2}\right). \quad (20)$$

Če definiramo niz parametrov $\tilde{b}(k)$, da veljajo naslednje zveze

$$\begin{aligned} \tilde{b}(0) &= \frac{1}{2}b(1), \\ \tilde{b}(k) &= 2b(k) - \tilde{b}(k-1), \quad k = 1, 2, 3, \dots, \frac{M}{2} - 2, \end{aligned} \quad (21)$$

$$\tilde{b}\left(\frac{M}{2} - 1\right) = 2b\left(\frac{M}{2}\right),$$

dobimo novo formulo za realni frekvenčni odziv

$$H_r(\omega) = \cos \frac{\omega}{2} \sum_{k=0}^{\frac{M}{2}-1} \tilde{b}(k) \cos \omega k. \quad (22)$$

- **tip 3, M liho število, antisimetričen odziv na enotin impulz $h(n) = -h(M - 1 - n)$**

Realni frekvenčni odziv $H_r(\omega)$ v tem primeru izračunamo kot

$$H_r(\omega) = 2 \sum_{n=0}^{\frac{M-3}{2}} h(n) \sin \omega \left(\frac{M-1}{2} - n \right). \quad (23)$$

Če nadomestimo indeks n s $k = \frac{M-1}{2} - n$ in definiramo nov niz koeficientov $c(k)$

$$c(k) = 2h \left(\frac{M-1}{2} - k \right), \quad k = 1, 2, \dots, \frac{M-1}{2}, \quad (24)$$

lahko zapišemo formulo (23) v obliki

$$H_r(\omega) = \sum_{k=1}^{\frac{M-1}{2}} c(k) \sin \omega k. \quad (25)$$

Formulo (25) lahko naprej preoblikujemo v

$$H_r(\omega) = \sin \omega \sum_{k=0}^{\frac{M-3}{2}} \tilde{c}(k) \cos \omega k, \quad (26)$$

pri čemer moramo definirati še koeficiente $\tilde{c}(k)$

$$\begin{aligned} \tilde{c} \left(\frac{M-3}{2} \right) &= c \left(\frac{M-1}{2} \right), \\ \tilde{c} \left(\frac{M-5}{2} \right) &= 2c \left(\frac{M-3}{2} \right), \\ &\vdots \\ \tilde{c}(k-1) - \tilde{c}(k+1) &= 2c(k), \quad 2 \leq k \leq \frac{M-5}{2}, \\ \tilde{c}(0) + \frac{1}{2}\tilde{c}(2) &= c(1). \end{aligned} \quad (27)$$

- **tip 4, M sodo število, antisimetričen odziv na enotin impulz $h(n) = -h(M-1-n)$**

Realni frekvenčni odziv $H_r(\omega)$ v tem primeru izračunamo kot

$$H_r(\omega) = 2 \sum_{n=0}^{\frac{M}{2}-1} h(n) \sin \omega \left(\frac{M-1}{2} - n \right). \quad (28)$$

Če nadomestimo indeks n s $k = \frac{M}{2} - n$ in definiramo nov niz koeficientov $d(k)$

$$d(k) = 2h \left(\frac{M}{2} - k \right), \quad k = 1, 2, \dots, \frac{M}{2}, \quad (29)$$

lahko zapišemo formulo (28) v obliki

$$H_r(\omega) = \sum_{k=1}^{\frac{M}{2}} d(k) \sin \omega \left(k - \frac{1}{2} \right). \quad (30)$$

Formulo (30) lahko naprej preoblikujemo v

$$H_r(\omega) = \sin \frac{\omega}{2} \sum_{k=0}^{\frac{M}{2}-1} \tilde{d}(k) \cos \omega k, \quad (31)$$

pri čemer moramo definirati še koeficiente $\tilde{d}(k)$

$$\tilde{d} \left(\frac{M}{2} - 1 \right) = 2d \left(\frac{M}{2} \right),$$

$$\tilde{d}(k-1) - \tilde{d}(k) = 2d(k), 2 \leq k \leq \frac{M}{2} - 1, \quad (32)$$

$$\tilde{d}(0) - \frac{1}{2}\tilde{d}(1) = d(1).$$

Iz enačb (17), (22), (26) in (31) je razvidna posplošena oblika izračun $H_r(\omega)$

$$H_r(\omega) = Q(\omega)P(\omega). \quad (33)$$

V tabeli 1 so združene vse formule za izračun $H_r(\omega)$ za posamezen tip filtra KEO z linearno fazo.

$H_r(\omega) = Q(\omega)P(\omega)$			
Tip filtra	Opis	$Q(\omega)$	$P(\omega)$
Tip 1	M lih, simetrija $h(n) = h(M-1-n)$	1	$\sum_{k=0}^{\frac{M-1}{2}} a(k) \cos \omega k$
Tip 2	M sod, simetrija $h(n) = h(M-1-n)$	$\cos \frac{\omega}{2}$	$\sum_{k=0}^{\frac{M}{2}-1} \tilde{b}(k) \cos \omega k$
Tip 3	M lih, antisimetrija $h(n) = -h(M-1-n)$	$\sin \omega$	$\sum_{k=0}^{\frac{M-3}{2}} \tilde{c}(k) \cos \omega k$
Tip 4	M sod, antisimetrija $h(n) = -h(M-1-n)$	$\sin \frac{\omega}{2}$	$\sum_{k=0}^{\frac{M}{2}-1} \tilde{d}(k) \cos \omega k$

Tabela 1: Tabela s formulami za izračun realnega frekvenčnega odziva $H_r(\omega)$ za filtre KEO z linearno fazo.

Oblika zapisa za $P(\omega)$, ki združuje vse tipe filtrov, bi bila

$$P(\omega) = \sum_{k=0}^L \alpha(k) \cos \omega k, \quad (34)$$

kjer $\alpha(k)$ predstavlja koeficiente za določen tip filtra. $L = \frac{M-1}{2}$ za filtre tipa 1, $L = \frac{M}{2} - 1$ za filtre tipa 2 in 4 ter $L = \frac{M-3}{2}$ za filtre tipa 3.

3.3. Optimalni filtri KEO z linearno fazo

Načrtovanje optimalnih filtrov KEO je povezano s Čebiševim aproksimacijskim problemom. Optimalni kriterij načrtovanja je minimizacija maksimalne absolutne napake filtra, tako da je utežena absolutna aproksimacijska napaka med želenim frekvenčnim odzivom $H_{dr}(\omega)$ in dejanskim frekvenčnim odzivom filtra $H_r(\omega)$ čim bolj enakomerno porazdeljena v prepustnem in zapornem pasu. Poleg enakomerne porazdeljenosti napake si tudi želimo

izbrati relativne velikosti napak v posameznem pasu. Te velikosti izbiramo s pomočjo utežnostne funkcije $W(\omega)$, ki jo največkrat definiramo kot

$$W(\omega) = \begin{cases} \frac{\delta_1}{\delta_2}, & \omega \text{ v prepustnem pasu} \\ 1, & \omega \text{ v zapornem pasu} \end{cases}. \quad (35)$$

Uteženo aproksimacijsko napako lahko definiramo kot

$$\begin{aligned} E(\omega) &= W(\omega)[H_{dr}(\omega) - H_r(\omega)] = W(\omega)[H_{dr}(\omega) - Q(\omega)P(\omega)] \\ &= W(\omega)Q(\omega) \left[\frac{H_{dr}(\omega)}{Q(\omega)} - P(\omega) \right] = \widehat{W}(\omega)[\widehat{H}_{dr}(\omega) - P(\omega)], \end{aligned} \quad (36)$$

kjer sta modificirana utežnostna funkcija in modificiran želeni frekvenčni odziv definirana kot

$$\widehat{W}(\omega) = W(\omega)Q(\omega) \text{ in} \quad (37)$$

$$\widehat{H}_{dr}(\omega) = \frac{H_{dr}(\omega)}{Q(\omega)}. \quad (38)$$

Čebišev aproksimacijski problem je iskanje koeficientov polinoma $P(\omega)$, ki minimizirajo maksimalno vrednost funkcije napake $E(\omega)$. Matematično lahko problem formuliramo

$$\min_{P(\omega)} [\max_{\omega \in S} |E(\omega)|] = \min_{P(\omega)} [\max_{\omega \in S} |\widehat{W}(\omega)[\widehat{H}_{dr}(\omega) - P(\omega)]|], \quad (39)$$

ali

$$\begin{aligned} \min_{P(\omega)} [\max_{\omega \in S} |E(\omega)|] &= \\ \min_{P(\omega)} [\max_{\omega \in S} |\widehat{W}(\omega)[\widehat{H}_{dr}(\omega) - \sum_{k=0}^L \alpha(k) \cos \omega k]|], \end{aligned} \quad (40)$$

če v (39) vključimo še enačbo (34). Množica S je v enačbah predstavljena kot disjunktna unija vseh frekvenčnih pasov (npr. $S = [0, \omega_p] \cup [\omega_s, \pi]$ v primeru nizkoprepustnega filtra). Prehodni pasovi se pri optimizaciji ne upoštevajo.

3.4. Načrtovanje optimalnih filtrov KEO z linearno fazo

3.4.1. Načrtovanje filtrov s koeficienti "neskončne" natančnosti

Reševanje optimizacijskega problema, kot je opisan z enačbo (39), lahko opravimo z Remezovim algoritmom po Parks-McClellanovi metodi [15]. S tem algoritmom dobimo tipično koeficiente filtra v obliki 32-bitnih števil v plavajoči vejici. 32-bitna dolžina koeficientov sicer ne predstavlja neskončne natančnosti, se pa ta izraz pogosto uporablja v primerjavi s krajšimi koeficienti (dolžine 8 ali 10 bitov). Krajši koeficienti filtrov se uporabljajo pri procesorjih DSP s končno številsko aritmetiko.

Remezov algoritem temelji na Čebiševem alternacijskem izreku. Izrek pravi, da je aproksimacija optimalna, kadar funkcija napake $|E(\omega)|$ v (40) doseže svojo maksimalno

vrednosti v vsaj $L + 2$ primerih (to je eden več kot je koeficientov). Če so frekvence, kjer je maksimum dosežen, urejene na naslednji način

$$\omega_1 < \omega_2 < \dots < \omega_{M+2}, \quad (41)$$

potem maksimalna napaka alternira

$$E(\omega_{i+1}) = -E(\omega_i), \quad i = 1, 2, \dots, L + 1. \quad (42)$$

Remezov algoritem po Parks-McClellanovi metodi lahko zapišemo v naslednjih korakih:

1. Inicializacija: izberemo niz $L+2$ ekstremalnih frekvenc.
2. Izračunamo najboljšo Čebiševo aproksimacijo za dani ekstremalni niz frekvenc. Tako dobimo δ .
3. Interpolacija: Izračunamo funkcijo napake $E(\omega)$ za celoten niz frekvenc S .
4. V nizu frekvenc S poiščemo lokalne maksimume $|E(\omega)|$.
5. Če $\max_{\omega \in S} |E(\omega)| > \delta$, potem osvežimo ekstremalni niz frekvenc z izborom novih frekvenc, kjer ima $|E(\omega)|$ lokalne maksimume. Funkcija napake mora alternirati in frekvence morajo biti urejene kot v (41) in (42). Nadaljujemo na koraku 2.
6. Če $\max_{\omega \in S} |E(\omega)| \leq \delta$, smo končali.

3.4.2. Obstoječe metode za načrtovanje filtrov s koeficienti končne dolžine

Kot je bilo že uvodoma omenjeno, želimo imeti zaradi uporabe cenejših in hitrejših procesorjev DSP s fiksno aritmetiko tudi filtre, ki imajo krajše koeficiente. Ti so ponavadi dolžine 8 ali 10 bitov. Eden od načinov, kako priti do krajših koeficientov, je z zaokroževanjem koeficientov "neskončne" natančnosti, ki jih pridobimo z uporabo Remezovega algoritma. Izkaže se, da ta rešitev ni najboljša in da takšni filtri niso več optimalni. Maksimalna napaka filtra z zaokroženimi koeficienti je lahko do 30 dB večja kot pri optimalnem filtru s koeficienti končne dolžine [9].

Filtre KEO z linearno fazo in koeficienti končne dolžine lahko dobimo z uporabo MILP. Koeficienti filtra $h(k)$ so b -bitna cela števila iz niza I_b in sicer

$$I_b = \{-2^{b-1}, \dots, -1, 0, 1, \dots, 2^{b-1}\}. \quad (43)$$

Če imamo b -biten koeficient, je 1 bit namenjen predznaku koeficienta, preostali $b - 1$ biti pa predstavljajo velikost.

Problem iskanja koeficientov končne dolžine definiramo (podobno kot v [9,17]) kot optimizacijski problem (izhajamo iz (39))

$$\left\{ \begin{array}{l} \text{minimiziraj } \delta, \\ \text{tako da veljajo omejitve:} \\ \frac{P(\omega_l)}{2^{b-1}} - \frac{2^{b-1}\delta}{\widehat{W}(\omega_l)} \leq \widehat{H}_{dr}(\omega_l), \\ -\frac{P(\omega_l)}{2^{b-1}} - \frac{2^{b-1}\delta}{\widehat{W}(\omega_l)} \leq -\widehat{H}_{dr}(\omega_l), \\ -2^{b-1} \leq \alpha(k) \leq 2^{b-1}, \\ \alpha(k) \in \mathbb{Z}, 0 \leq k \leq L. \end{array} \right. \quad (44)$$

Spremenljivko ω_l diskretiziramo na množici S , ki predstavlja unijo vseh frekvenčnih pasov, brez upoštevanja prehodnih pasov. Reševanje optimizacijskega problema (44) je časovno potraten postopek. MILP uporablja branch-and-bound algoritem, ki sistematično preišče celoten prostor možnih rešitev in vrne najboljšo. Za namen načrtovanja optimalnih filtrov KEO z linearno fazo in koeficienti končne dolžine je bila razvita spodnja meja aproksimacijske napake, ki bistveno izboljša hitrost branch-and-bound algoritma [11]. V [10] je uporabljen algoritem LLL pri izračunu spodnje meje aproksimacijske napake, kar še dodatno pohitri branch-and-bound algoritem.

3.4.3. Polinomska aproksimacija in uporaba Babaievega algoritma in heuristike pri načrtovanju filtrov s koeficienti končne dolžine

Če želimo v računalniku predstaviti poljubno netrivialno matematično funkcijo, jo ponavadi predstavimo z njeno aproksimacijo. Problem aproksimacije lahko definiramo kot iskanje funkcije $g(x)$, ki pripada danemu razredu funkcij, in ki je čim bližje funkciji $f(x)$, ki jo želimo aproksimirati. Polinomi so funkcije z lepimi lastnostmi, saj je računanje vrednosti enostavno. Aproksimacijska funkcija $g(x)$ zato največkrat pripada razredu polinomov.

Tipičen aproksimacijski problem je torej iskanje polinoma $g(x)$ stopnje $\leq n$, ki je čim bližje $f(x)$. Razdaljo med funkcijama definiramo z uporabo norme. Kakovost aproksimacije predstavlja norma ostanka $\|f - g\|$. Različne norme pripeljejo do različnih aproksimacijskih funkcij. Najpogosteje uporabljeni sta L_2 in L_∞ norma. Definicija obeh norm za primer vektorjev je podana v poglavju 1.1. in jo lahko analogno definiramo tudi za funkcije.

Če predpostavimo, da je podana funkcija $f(x)$ zvezna na intervalu $[a, b]$, lahko definiramo aproksimacijski problem:

- v L_2 normi kot iskanje g , ki minimizira

$$\|f - g\|_2 = \sqrt{\int_a^b |f(x) - g(x)|^2 dx}, \quad (45)$$

- v L_∞ normi kot iskanje g , ki minimizira

$$\|f - g\|_\infty = \max_{a \leq x \leq b} |f(x) - g(x)|. \quad (46)$$

Kot aproksimacijski polinom lahko nastopa tudi trigonometrični polinom. Kosinusni polinomi nastopajo tudi pri načrtovanju filtrov KEO z linearno fazo. Glede na (36) in (40) vidimo, da je cilj načrtovanja filtrov z uporabo polinomske aproksimacije ta, da sta želeni frekvenčni odziv \widehat{H}_{dr} in polinom $P(\omega)$ čim bližje. Če zapišemo $P(\omega)$ v obliki vektorja \vec{v} in \widehat{H}_{dr} v obliki vektorja \vec{y} na spodnji način

$$\underbrace{\begin{pmatrix} \frac{\alpha_0}{2^{b-1}} + \frac{\alpha_1 \cos \omega_1}{2^{b-1}} + \dots + \frac{\alpha_{L-1} \cos(L-1)\omega_1}{2^{b-1}} + \frac{\alpha_L \cos L\omega_1}{2^{b-1}} \\ \frac{\alpha_0}{2^{b-1}} + \frac{\alpha_1 \cos \omega_2}{2^{b-1}} + \dots + \frac{\alpha_{L-1} \cos(L-1)\omega_2}{2^{b-1}} + \frac{\alpha_L \cos L\omega_2}{2^{b-1}} \\ \vdots \\ \frac{\alpha_0}{2^{b-1}} + \frac{\alpha_1 \cos \omega_l}{2^{b-1}} + \dots + \frac{\alpha_{L-1} \cos(L-1)\omega_l}{2^{b-1}} + \frac{\alpha_L \cos L\omega_l}{2^{b-1}} \end{pmatrix}}_{\vec{v}} \text{ in } \underbrace{\begin{pmatrix} \widehat{H}_{dr}(\omega_1) \\ \widehat{H}_{dr}(\omega_2) \\ \vdots \\ \widehat{H}_{dr}(\omega_l) \end{pmatrix}}_{\vec{y}}, \quad (47)$$

potem želimo, da sta omenjena vektorja čim bližje glede na L_∞ normo. Frekvence $\omega_1, \omega_2, \dots, \omega_l$, ki nastopajo v obeh vektorjih, predstavljajo enakomerno diskretizirane frekvence na intervalu od 0 do π . Upoštevamo tudi prehodni pas.

Če polinom $P(\omega)$ razbijemo na posamezne vektorje, dobimo

$$\alpha_0 \underbrace{\begin{pmatrix} \frac{1}{2^{b-1}} \\ 1 \\ \vdots \\ 1 \\ \frac{1}{2^{b-1}} \end{pmatrix}}_{\vec{v}_0} + \alpha_1 \underbrace{\begin{pmatrix} \frac{\cos \omega_1}{2^{b-1}} \\ \cos \omega_2 \\ \vdots \\ \cos \omega_l \\ \frac{\cos \omega_l}{2^{b-1}} \end{pmatrix}}_{\vec{v}_1} + \dots + \alpha_{L-1} \underbrace{\begin{pmatrix} \frac{\cos(L-1)\omega_1}{2^{b-1}} \\ \cos(L-1)\omega_2 \\ \vdots \\ \cos(L-1)\omega_l \\ \frac{\cos(L-1)\omega_l}{2^{b-1}} \end{pmatrix}}_{\vec{v}_{L-1}} + \alpha_L \underbrace{\begin{pmatrix} \frac{\cos L\omega_1}{2^{b-1}} \\ \cos L\omega_2 \\ \vdots \\ \cos L\omega_l \\ \frac{\cos L\omega_l}{2^{b-1}} \end{pmatrix}}_{\vec{v}_L}. \quad (48)$$

Najti moramo torej celoštevilске koeficiente $\alpha_0, \dots, \alpha_L$, tako da minimiziramo

$$\|\vec{v} - \vec{y}\|_\infty = \|\alpha_0 \vec{v}_0 + \alpha_1 \vec{v}_1 + \dots + \alpha_L \vec{v}_L - \vec{y}\|_\infty. \quad (49)$$

Na problem (49) lahko gledamo kot na iskanje najbližjega vektorja v L_∞ normi. Kot je že bilo omenjeno v poglavju 2.3., obstaja Kannanov algoritem, ki rešuje ta problem, vendar je njegova zahtevnost super-eksponentna. V praksi je bolje uporabiti Babaiev algoritem, ki v polinomskega času reši enak problem za L_2 normo.

Babaiev algoritem kot rezultat vrne vektor \vec{v} . V [4] je opisana hevrstika, ki je bila uporabljena tudi v našem magistrskem delu. Hevrstika predvideva, da bi moral biti rezultat v L_∞ normi blizu rezultatu v L_2 normi. Z uporabo hevrstike se preišče okolica vektorja \vec{v} , tako da se omenjenemu vektorju prišteva in odšteva vektorje, ki predstavljajo polinom $P(\omega)$ in so LLL reducirani. Na ta način dobimo kandidate za rezultat, ki je v L_∞ normi boljši od predhodnega rezultata. Izračunamo L_∞ normo med kandidatom in želenim frekvenčnim

odzivom in če je L_∞ norma manjša kot pri predhodnem rezultatu, potem obdržimo nov rezultat. Postopek ponavljamo, dokler je rezultat še možno izboljšati. Hevristika je natančneje opisana s spodnjo psevdokodo.

Hevristika

Opis: Preiščemo soseščino vektorja \vec{v} , da bi se želenemu frekvenčnemu odzivu (vektor \vec{y}) približali v L_∞ normi.

Vhod: vektor \vec{v} , vektor \vec{y} in LLL reducirana baza B' .

Vektor \vec{v} predstavlja frekvenčni odziv filtra in je v L_2 normi najbližje želenemu frekvenčnemu odzivu filtra. Vektor \vec{y} predstavlja želeni frekvenčni odziv. LLL reducirana baza B' vsebuje vektorje $\vec{v}'_0, \vec{v}'_1, \dots, \vec{v}'_L$.

Izhod: vektor \vec{v} .

1. Množico Z napolni z $2L + 2$ vektorji:

$$\vec{z}_0 = \vec{v} + \vec{v}'_0, \vec{z}_1 = \vec{v} - \vec{v}'_0, \vec{z}_2 = \vec{v} + \vec{v}'_1, \vec{z}_3 = \vec{v} - \vec{v}'_1, \dots, \vec{z}_{2L} = \vec{v} + \vec{v}'_L, \vec{z}_{2L+1} = \vec{v} - \vec{v}'_L.$$

2. Kot referenčno normo izračunaj L_∞ normo med vektorjem \vec{v} in vektorjem \vec{y} , ki predstavlja želeni frekvenčni odziv

$$ref = \|\vec{v} - \vec{y}\|_\infty.$$

3. Za $i = 0$ do $2L + 1$ izračunaj

$$\delta_i = \|\vec{z}_i - \vec{y}\|_\infty.$$

4. Če obstaja tak $\min_{i=1\dots 2L+1} \delta_i$, da

$$\delta_i < ref,$$

potem novi $\vec{v} = \vec{z}_i$ in nadaljuj na koraku 2.

Sicer vrni \vec{v} .

Opisana metoda načrtovanja filtrov KEO z linearno fazo in koeficienti končne dolžine z uporabo algoritmov za redukcijo baze mreže ima tudi slabost. Pri načrtovanju namreč ni mogoče upoštevati utežnostne funkcije $\widehat{W}(\omega)$.

3.4.4. Načrtovanje filtrov s koeficienti končne dolžine kot reševanje sistema diofantskih enačb

V [2] je uporabljen algoritem LLL za reševanje sistema diofantskih enačb s spodnjo in zgornjo mejo spremenljivk. Problem, ki se rešuje z uporabo algoritma LLL, je definiran kot

$$\text{Ali obstaja vektor } \vec{x} \in \mathbb{Z}^n, \text{ tako da } A\vec{x} = \vec{d}, 0 \leq \vec{x} \leq \vec{u}? \quad (50)$$

Pri tem je predpostavljeno, da je A celoštevilaska matrika velikosti $m \times n$, kjer je $m \leq n$, \vec{d} je celoštevilski vektor dolžine m in \vec{u} celoštevilski vektor dolžine n . Kot je v članku omenjeno, gre za NP-poln problem.

Problem (50) je mogoče nekoliko prirediti in ga uporabiti za reševanje problema načrtovanja filtrov KEO z linearno fazo in koeficienti končne dolžine:

$$\text{Ali obstaja vektor } \vec{\alpha} \in \mathbb{Z}^{L+1}, \text{ tako da } A\vec{\alpha} = \vec{d}, -2^{b-1} \leq \vec{\alpha} \leq 2^{b-1}? \quad (51)$$

Vektor $\vec{\alpha}$ predstavlja koeficiente $\alpha_0, \alpha_1, \dots, \alpha_L$. Matrika A je velikosti $n \times (L + 1)$, kjer je $L + 1$ število koeficientov filtra in n število frekvenc, kjer definiramo želeni frekvenčni odziv filtra. Veljati mora $n < L + 1$. Vektor \vec{d} predstavlja želeni frekvenčni odziv filtra v izbranih n frekvencah, b pa predstavlja dolžino koeficientov filtra. Matrika A je predstavljena kot

$$A = \begin{pmatrix} 2^{b-1} & 2^{b-1} \cos \omega_1 & \dots & 2^{b-1} \cos(L-1)\omega_1 & 2^{b-1} \cos L\omega_1 \\ 2^{b-1} & 2^{b-1} \cos \omega_2 & \dots & 2^{b-1} \cos(L-1)\omega_2 & 2^{b-1} \cos L\omega_2 \\ \vdots & \vdots & \dots & \vdots & \vdots \\ 2^{b-1} & 2^{b-1} \cos \omega_n & \dots & 2^{b-1} \cos(L-1)\omega_n & 2^{b-1} \cos L\omega_n \end{pmatrix}. \quad (52)$$

Frekvence $\omega_1, \omega_2, \dots, \omega_n$ lahko predstavljajo frekvence, kjer frekvenčni odziv filtra izračunan z uporabo Remezovega algoritma doseže svoj maksimum. Izbranih frekvenc mora biti manj kot je število koeficientov. Vektor \vec{d} lahko predstavimo kot:

$$\vec{d} = \begin{pmatrix} \widehat{H}_{dr}(\omega_1) \\ \widehat{H}_{dr}(\omega_2) \\ \vdots \\ \widehat{H}_{dr}(\omega_n) \end{pmatrix}. \quad (53)$$

Kot je opisano v [2], z B definiramo bazo mreže na naslednji način:

$$B = \begin{pmatrix} I^{(L+1)} & 0^{((L+1) \times 1)} \\ O^{(1 \times (L+1))} & N_1 \\ N_2 A & -N_2 \vec{d} \end{pmatrix}. \quad (54)$$

$I^{(L+1)}$ predstavlja $(L + 1)$ -dimenzionalno identično matriko, $0^{(p \times q)}$ predstavlja matriko velikosti $p \times q$, ki vsebuje le ničle. Števili N_1 in N_2 določimo na naslednji način

$$N_1 > N_{01}, \text{ kjer je } N_{01}^2 > (L + 1)2^{n(L+1)(2b-2)} \text{ in} \quad (55)$$

$$N_2 > N_{01}N_1.$$

Z uporabo algoritma LLL rešimo problem (51) brez omejitve spremenljivk. Pri tem dobimo reducirano bazo B' . Iz reducirane baze izluščimo prvih $L + 1 - n$ stolpcev in dobimo matriko \hat{B}' . Rešitev problema (49) brez omejitve spremenljivk obstaja, če so vsi stolpci \hat{B}' razen zadnjega oblike

$$\begin{pmatrix} \vec{\alpha}_i \\ 0 \\ \mathbf{0}_{(n \times 1)} \end{pmatrix} \quad (56)$$

in zadnji stolpec oblike

$$\begin{pmatrix} \vec{\alpha}_d \\ N_1 \\ \mathbf{0}_{(n \times 1)} \end{pmatrix}. \quad (57)$$

Za rešitev sistema diofantskih enačb vzamemo vektor $\vec{\alpha}_d$, čeprav tudi vektorji $\vec{\alpha}_i, i = 1, \dots, L - n$, rešijo enačbo $A\vec{\alpha} = \vec{d}$. Za rešitev $\vec{\alpha}_d$ je potrebno preveriti, ali elementi vektorja oz. koeficienti filtra ustrezajo spodnji in zgornji meji. Če ustrezajo, potem imamo rešitev, sicer poskusimo z linearnimi kombinaciji preostalih vektorjev $\vec{\alpha}_i$ spremeniti vektor $\vec{\alpha}_d$. Celoten algoritem je opisan spodaj.

Algoritem za reševanje sistema diofantskih enačb s spodnjo in zgornjo mejo za primer načrtovanja filtrov KEO z linearno fazo in koeficienti končne dolžine

Opis: Algoritem izračuna rešitev sistema diofantskih enačb $A\vec{\alpha} = \vec{d}, -2^{b-1} \leq \vec{\alpha} \leq 2^{b-1}$

Vhod: matrika A , vektor \vec{d} in vektor \vec{u}

Izhod: vektor $\vec{\alpha}_d$

1. Pripravi bazo B , ki je oblike kot je opisano v (52).
2. S pomočjo algoritma LLL izračunaj LLL reducirano bazo B' .
3. Izlušči bazo \hat{B}' in preveri, če so stolpci oblike (54) in (55).
4. Če niso, zaključí. V tem primeru rešitve nismo našli.
Sicer nadaljuj.
5. Iz zadnjega stolpca \hat{B}' vzami vektor $\vec{\alpha}_d$.
6. Če velja $-2^{b-1} \leq \vec{\alpha}_d \leq 2^{b-1}$, potem kot rešitev vrni $\vec{\alpha}_d$.

Sicer poskušaj s prištevanjem in odštevanjem preostalih stolpcev $\vec{\alpha}_i$ k $\vec{\alpha}_d$ zadostiti pogoju $-2^{b-1} \leq \vec{\alpha}_d \leq 2^{b-1}$ na naslednji način:

- a. T je število, ki pove, koliko koeficientov presega spodnjo in zgornjo mejo za koeficiente.
 - b. Vektorju $\vec{\alpha}_d$ sistematično prištevaj ali odštevaj enega od preostalih vektorjev matrike \hat{B}' .
 - c. Če se število T zmanjša, nadaljuj na koraku 6.
 - d. Sicer rešitev v okviru spodnje in zgornje meje ne obstaja.
-

Prav tako kot pri metodi načrtovanja z uporabo Babaijevega algoritma in hevrstike tudi tukaj pri načrtovanju ni mogoče upoštevati utežnostne funkcije $\hat{W}(\omega)$.

4. Metode za načrtovanje filtrov KEO tipa 1 s koeficienti končne dolžine

Metode za načrtovanje filtrov KEO z linearno fazo in s koeficienti končne dolžine so bile razvite za filtre tipa 1. Z majhnimi spremembami je mogoče metode uporabiti tudi za druge 3 tipe filtrov. Kot je prikazano v tabeli 1, imajo filtri tipa 1 liho število koeficientov in simetričen odziv na enotin impulz. Frekvenčni odziv lahko zapišemo s formulo

$$H_r(\omega) = h\left(\frac{M-1}{2}\right) + 2 \sum_{n=0}^{\frac{M-3}{2}} h(n) \cos \omega\left(\frac{M-1}{2} - n\right), \quad (58)$$

kjer M predstavlja število koeficientov filtra. Število bitov za predstavitev posameznega koeficienta smo označili z b .

Metode so bile implementirane v okolju Scilab⁷. Scilab je prostodostopen in odprtokoden programski paket namenjen predvsem numeričnemu računanju. Posebnosti implementacije posameznih metod sledijo v nadaljevanju poglavja 4.

4.1. Metoda zaokroževanja

Metoda zaokroževanja temelji na zaokroževanju koeficientov "neskončne" natančnosti izračunanih z Remezovim algoritmom. V Scilab so že vključene metode za delo z digitalnim procesiranjem signalov in tako tudi metode za načrtovanje digitalnih filtrov. Najprej je bila uporabljena funkcija *eqfir*, ki služi za načrtovanje filtrov KEO z linearno fazo in koeficienti "neskončne" natančnosti in je definirana kot:

`[hn]=eqfir(nf,bedge,des,wate)`, kjer so vhodni argumenti funkcije definirani kot:

- *nf* definira število koeficientov filtra,
- *bedge* je matrika, s katero določimo frekvenčne pasove,
- *des* je vektor, ki določa želeno vrednost amplitude v posameznem pasu,
- *wate* je vektor, ki določa utežnostno funkcijo v posameznem pasu.

Izhod funkcije *hn* predstavlja izračunane koeficiente.

Rezultat te funkcije, koeficiente filtra, z uporabo funkcije *round* zaokrožimo na želeno število bitov.

⁷ Dostopno na: <http://www.scilab.org>

4.2. Metoda z uporabo celoštevilskega linearnega programiranja

Za načrtovanje z uporabo metode celoštevilskega linearnega programiranja oz. MILP je bilo potrebno v okolje Scilab vključiti modul `lp_solve`⁸. Modul `lp_solve` je prostodostopen in je namenjen reševanju optimizacijskih problemov s pomočjo simpleksnega algoritma in branch-and-bound algoritma za primer celoštevilskih spremenljivk. Funkcija `lp_solve` je definirana kot

`[obj, x] = lp_solve(f, a, b, e, vlb, vub, xint)`, ki rešuje splošno definiran optimizacijski problem:

$$\begin{aligned} \max \quad & v = f'x, \\ ax & \langle \rangle b, \\ vlb & \leq x \leq vub, x \in \mathbb{Z}. \end{aligned}$$

Argumenti funkcije `lp_solve` so definirani kot:

- f je vektor dolžine n , ki predstavlja koeficiente linearne kriterijske funkcije,
- a je matrika $m \times n$, ki predstavlja linearne omejitve,
- b je vektor, ki predstavlja desno stran neenačb oz. omejitev,
- e je vektor dolžine m , ki določa operator neenakosti:
 - $e(i) = -1$ (manjše kot),
 - $e(i) = 0$ (enako),
 - $e(i) = 1$ (večje kot),
- vlb je vektor, ki določa spodnje meje rezultata,
- vub je vektor, ki določa zgornje meje rezultata,
- $xint$ je vektor, s katerim določimo, katere spremenljivke morajo biti celoštevilске.

Izhod funkcije je določen z:

- obj , ki določa optimalno vrednost kriterijske funkcije,
- x , ki predstavlja optimalno vrednost spremenljivk.

Naša naloga je bila reševanje problema, ki je za filter KEO z linearno fazo in koeficienti končne dolžine definiran v splošnem kot (44). Metoda `lp_solve` rešuje le maksimum kriterijske funkcije. Za načrtovanje filtrov KEO tipa 1 je bilo zato potrebno optimizacijski problem za metodo `lp_solve` definirati kot

⁸ Dostopno na: <http://lpsolve.sourceforge.net>

$$\left\{ \begin{array}{l} \text{maksimiraj } -\delta, \\ \text{tako da veljajo omejitve:} \\ \frac{2h_0 \cos(\frac{M-1}{2}\omega)}{2^{b-1}} + \frac{2h_1 \cos(\frac{M-3}{2}\omega)}{2^{b-1}} + \dots + \frac{2h_{\frac{M-3}{2}} \cos \omega}{2^{b-1}} + \frac{h_{\frac{M-1}{2}}}{2^{b-1}} - \frac{2^{b-1}\delta}{W(\omega)} \leq H_{dr}(\omega), \\ -\frac{2h_0 \cos(\frac{M-1}{2}\omega)}{2^{b-1}} - \frac{2h_1 \cos(\frac{M-3}{2}\omega)}{2^{b-1}} - \dots - \frac{2h_{\frac{M-3}{2}} \cos \omega}{2^{b-1}} - \frac{h_{\frac{M-1}{2}}}{2^{b-1}} - \frac{2^{b-1}\delta}{W(\omega)} \leq -H_{dr}(\omega), \\ -2^{b-1} \leq h(k) \leq 2^{b-1}, \\ h(k) \in \mathbb{Z}, 0 \leq k \leq \frac{M-1}{2}. \end{array} \right. \quad (59)$$

4.3. Metoda z uporabo Babaievega algoritma in hevrstike

Okolje Scilab ne vključuje algoritmov za redukcijo baze mreže. V Scilab je bilo zato potrebno implementirati algoritem LLL in Babaiev algoritem s hevrstiko. Pseudokoda algoritma LLL je podana v poglavju 2.2., Babaievega algoritma v poglavju 2.3. in hevrstike v poglavju 3.4.3.

Ker so bili načrtovani filtri KEO tipa 1, je bilo potrebno definirati vektorja \vec{v} in \vec{y} , za katera smo želeli, da sta čimbližje skupaj. Vektor \vec{v} definira dejanski frekvenčni odziv filtra in ga lahko zapišemo kot

$$\underbrace{\begin{pmatrix} \frac{2h_0 \cos(\frac{M-1}{2}\omega_1)}{2^{b-1}} + \frac{2h_1 \cos(\frac{M-3}{2}\omega_1)}{2^{b-1}} + \dots + \frac{2h_{\frac{M-3}{2}} \cos \omega_1}{2^{b-1}} + \frac{h_{\frac{M-1}{2}}}{2^{b-1}} \\ \frac{2h_0 \cos(\frac{M-1}{2}\omega_2)}{2^{b-1}} + \frac{2h_1 \cos(\frac{M-3}{2}\omega_2)}{2^{b-1}} + \dots + \frac{2h_{\frac{M-3}{2}} \cos \omega_2}{2^{b-1}} + \frac{h_{\frac{M-1}{2}}}{2^{b-1}} \\ \vdots \\ \frac{2h_0 \cos(\frac{M-1}{2}\omega_l)}{2^{b-1}} + \frac{2h_1 \cos(\frac{M-3}{2}\omega_l)}{2^{b-1}} + \dots + \frac{2h_{\frac{M-3}{2}} \cos \omega_l}{2^{b-1}} + \frac{h_{\frac{M-1}{2}}}{2^{b-1}} \end{pmatrix}}_{\vec{v}}, \quad (60)$$

vektor \vec{y} pa želeni frekvenčni odziv filtra in ga lahko zapišemo kot

$$\underbrace{\begin{pmatrix} H_{dr}(\omega_1) \\ H_{dr}(\omega_2) \\ \vdots \\ H_{dr}(\omega_l) \end{pmatrix}}_{\vec{y}}. \quad (61)$$

Nadaljni razcep vektorja \vec{v} na manjše vektorje lahko zapišemo

$$h_0 \underbrace{\begin{pmatrix} \frac{2 \cos(\frac{M-1}{2}\omega_1)}{2^{b-1}} \\ \frac{2 \cos(\frac{M-1}{2}\omega_2)}{2^{b-1}} \\ \vdots \\ \frac{2 \cos(\frac{M-1}{2}\omega_l)}{2^{b-1}} \end{pmatrix}}_{\vec{v}_0} + h_1 \underbrace{\begin{pmatrix} \frac{2 \cos(\frac{M-3}{2}\omega_1)}{2^{b-1}} \\ \frac{2 \cos(\frac{M-3}{2}\omega_2)}{2^{b-1}} \\ \vdots \\ \frac{2 \cos(\frac{M-3}{2}\omega_l)}{2^{b-1}} \end{pmatrix}}_{\vec{v}_1} + \dots + \frac{h_{M-3}}{2} \underbrace{\begin{pmatrix} \frac{2 \cos \omega_1}{2^{b-1}} \\ \frac{2 \cos \omega_2}{2^{b-1}} \\ \vdots \\ \frac{2 \cos \omega_l}{2^{b-1}} \end{pmatrix}}_{\vec{v}_{\frac{M-3}{2}}} + \frac{h_{M-1}}{2} \underbrace{\begin{pmatrix} \frac{1}{2^{b-1}} \\ \frac{1}{2^{b-1}} \\ \vdots \\ \frac{1}{2^{b-1}} \end{pmatrix}}_{\vec{v}_{\frac{M-1}{2}}}. \quad (62)$$

Frekvenčni prostor je diskretiziran na celotnem intervalu od $[0, \pi]$. Želeni frekvenčni odziv v prepustnem pasu je 1, v zapornem 0, v prehodnem pasu pa je aproksimiran rezultat, ki ga da Remezov algoritem. Testiranja so namreč pokazala, da je rezultat v tem primeru precej boljši, kot če je prehodni pas nedefiniran. Določitev prehodnega pasu se je izkazala kot pomembna pri končni kakovosti aproksimacije.

S pomočjo algoritma LLL je bila reducirana baza mreže, ki jo lahko označimo z B' in vsebuje vektorje $\vec{v}_0, \vec{v}_1, \dots, \vec{v}_{\frac{M-3}{2}}, \vec{v}_{\frac{M-1}{2}}$. Baza B' je bila reducirana glede na L_2 normo.

Potrebno je omeniti, da smo rezultat redukcije baze poskušali izboljšati še z metodo za približanje LLL reducirane baze k L_∞ normi. Če je B prvotna baza mreže in B' LLL reducirana baza, potem velja $B' = BU$ (kot je bilo omenjeno v 2.1.). U je unimodularna matrika s celoštevilskimi koeficienti. Ideja je bila preveriti, ali preoblikovanje matrike U privede do izboljšanja baze B' glede na L_∞ normo. Izboljšanje baze B' bi pomenilo, da bi bili vektorji v bazi krajši glede na L_∞ normo kot v prvotno reducirani bazi. Posameznemu stolpcu U tako prištejemo ali odštejemo enega od drugih stolpcev matrike U (pomnoženim s faktorjem 1). Nato preverimo, ali to izboljša bazo B' v L_∞ normi. Pseudokoda metode je objavljena spodaj.

Metoda za približanje LLL reducirane baze k L_∞ normi

Opis: Z metodo želimo modificirati LLL reducirano bazo, ki jo da algoritem LLL, in je reducirana glede na L_2 normo in tako narediti vektorje baze krajše glede L_∞ normo

Vhod: baza B , LLL reducirana baza B'

Izhod: reducirana baza B''

Izračunaj unimodularno matriko U ($B' = BU$).

Za $i = 1$ do n (n je število vektorjev v bazi)

Za $j = 1$ do n

Če $i \neq j$ potem

1. i -temu stolpcu U prištej j -ti stolpec U .
 2. Izračunaj reducirano bazo B'' z novim U .
 3. Preveri L_∞ normo i -tega vektorja B'' .
 4. Če je norma manjša, potem ohrani U , sicer vrni U v prejšnje stanje.
- Korake 1, 2, 3 in 4 ponovi za odštevanje j -tega stolpca.

Vrni B'' .

Konec.

Metoda za približanje LLL reducirane baze k L_∞ normi je bila preizkušena na 18 filtrih nizov A, C in E. Lastnosti filtrov so opisane v tabeli 2 poglavja 5. Pri načrtovanju filtrov smo celoten frekvenčni prostor diskretizirali na 512 frekvenc. V nobenem od primerov ni prišlo do izboljšanja lastnosti filtra, se je pa podaljšal čas izvajanja. Tako je npr. pri načrtovanju nizkoprepustnega filtra s 45 koeficienti in dolžino koeficientov 10 bitov, metoda podaljšala čas izvajanja za skoraj šestkrat. Namesto 210 ms je načrtovanje trajalo 1,2 s. Metode za izboljšanje baze pri končnih načrtovanjih filtrov zato nismo uporabili.

Pri testiranju metode za izboljšanje baze je bilo opaziti, da je metoda izboljšala LLL reducirano bazo, če je bilo število frekvenc ω , kjer je določen želeni frekvenčni odziv, majhno. Preverili smo, ali bi do izboljšanje baze glede na L_∞ normo prišlo, če bi bil želeni frekvenčni odziv določen le v točkah, kjer frekvenčni odziv filtra izračunan z Remezovim algoritmom doseže ekstremalne vrednosti. Ugotovljeno je bilo, da bi v tem primeru pri vseh filtrih (to je pri vseh 18 filtrih iz nizov A, C in E) prišlo do izboljšanja baze. Izboljšanje baze je bilo upoštevano, če je bil vsaj eden izmed baznih vektorjev po izvedbi metode krajši glede na L_∞ normo kot pred izvedbo metode. V 12 primerih je bil krajši tudi vektor \vec{b}'_1 (kot aproksimacija najkrajšega vektorja v bazi mreže).

Pri končnem testiranju metode za načrtovanje filtrov z uporabo Babaievega algoritma in hevrstike je kot vhodni podatek za Babaiev algoritem služila baza reducirana v L_2 normi. Kot je že bilo omenjeno, metode za približanje baze k L_∞ normi nismo uporabili. Rezultat Babaievega algoritma so bili celoštevilski koeficienti, ki so določali najbližji vektor želenemu frekvenčnemu odzivu.

4.4. Metoda z reševanjem sistema diofantskih enačb

Za reševanje sistema diofantskih enačb kot metode načrtovanja optimalnih filtrov KEO tipa 1 z linearno fazo in koeficienti končne dolžine je bil v okolju Scilab implementiran algoritem opisan v 3.4.4.

Za algoritem je bilo potrebno pripraviti vhodne podatke. Baza B (54) je vsebovala matriko A , ki smo jo definirali kot

$$A = \begin{pmatrix} 2^{b-1} \cos\left(\frac{M-1}{2}\right) \omega_1 & 2^{b-1} \cos\left(\frac{M-3}{2}\right) \omega_1 & \cdots & 2^{b-1} \cos \omega_1 & 2^{b-1} \\ 2^{b-1} \cos\left(\frac{M-1}{2}\right) \omega_2 & 2^{b-1} \cos\left(\frac{M-3}{2}\right) \omega_2 & \cdots & 2^{b-1} \cos \omega_2 & 2^{b-1} \\ \vdots & \vdots & & \vdots & \vdots \\ 2^{b-1} \cos\left(\frac{M-1}{2}\right) \omega_n & 2^{b-1} \cos\left(\frac{M-3}{2}\right) \omega_n & \cdots & 2^{b-1} \cos \omega_n & 2^{b-1} \end{pmatrix}. \quad (63)$$

Vektor \vec{d} je bil definiran enako kot v (53). Zaradi pogoja algoritma, da mora biti število diofantskih enačb manjše kot je število koeficientov filtra, smo izbrali samo $M - 2$ frekvenci, kjer smo določili zeleni frekvenčni odziv filtra. Te frekvence so bile enake frekvencam, kjer frekvenčni odziv filtra izračunan z Remezovim algoritmom doseže ekstremalne vrednosti.

Na testnih primerih se je izkazalo, da na način kot ga podaja algoritem v 3.4.4. ni mogoče načrtovati filtrov. Stolpci LLL reducirane baze \hat{B}' namreč niso bili v nobenem primeru zahtevane oblike (56) in (57). Razlog, zakaj rešitve za sistem diofantskih enačb ni bilo mogoče dobiti, lahko iščemo v tem, da gre pri načrtovanju filtrov za aproksimacijo zelenega frekvenčnega poteka in ne za natančno prileganje dejanskega in zelenega frekvenčnega odziva. Če bi želeli, da ima sistem diofantskih enačb rešitev, bi morali uganiti napako aproksimacije oz. preizkusiti, ali obstaja rešitev sistema enačb za določen razpon možnih napak aproksimacije.

5. Preizkus implementiranih metod načrtovanja

Vse štiri metode načrtovanja so bile preizkušene na nizkoprepustnem filtru s 45 koeficienti in dolžino koeficientov 8 bitov. Preizkus je bil opravljen še na 5 nizih filtrov z različnimi značilnostmi, ki določajo frekvenčno domeno filtra. Te značilnosti so podane v tabeli 2. Niz filtrov je označen s črko od A do E. Filtri nizov A, B in E so nizkoprepustni, filtri nizov C in D pa pasovnozaporni. Filtri nizov B in D imajo utež za napako v zapornem pasu filtra določeno s faktorjem 10, v prepustnem pasu pa s faktorjem 1. Vsi ostali nizi filtrov, imajo v vseh pasovih enakomerno uteženo napako. Za vsak niz filtrov je bilo načrtovanih 6 filtrov z različnim številom in z različnimi dolžinami koeficientov.

Niz	Pas 1	Pas 2	Pas 3
A	$\omega_p = [0; 0,40\pi]$ prepustni $W(\omega_p) = 1$	$\omega_s = [0,50\pi; \pi]$ zaporni $W(\omega_s) = 1$	
B	$\omega_p = [0; 0,40\pi]$ prepustni $W(\omega_p) = 1$	$\omega_s = [0,50\pi; \pi]$ zaporni $W(\omega_s) = 10$	
C	$\omega_{p1} = [0; 0,24\pi]$ prepustni $W(\omega_{p1}) = 1$	$\omega_s = [0,40\pi; 0,68\pi]$ zaporni $W(\omega_s) = 1$	$\omega_{p2} = [0,84\pi; \pi]$ prepustni $W(\omega_{p2}) = 1$
D	$\omega_{p1} = [0; 0,24\pi]$ prepustni $W(\omega_{p1}) = 1$	$\omega_s = [0,40\pi; 0,68\pi]$ zaporni $W(\omega_s) = 10$	$\omega_{p2} = [0,84\pi; \pi]$ prepustni $W(\omega_{p2}) = 1$
E	$\omega_p = [0,02\pi; 0,42\pi]$ prepustni $W(\omega_p) = 1$	$\omega_s = [0,52\pi; 0,98\pi]$ zaporni $W(\omega_s) = 1$	

Tabela 2: Lastnosti filtrov, ki pripadajo posameznemu nizu filtrov.

5.1. Rezultati

Nizkoprepustni filter KEO z linearno fazo s 45 koeficienti in s koeficienti dolžine 8 bitov je bil zasnovan z uporabo vseh prej opisanih metod. Kot je že bilo omenjeno, z metodo načrtovanja z reševanjem diofantskih enačb nismo uspeli zasnovati nobenega filtra, zato rezultati te metode tudi niso prikazani.

Omenjeni nizkoprepustni filter pripada nizu filtrov A, tako da zanj veljajo lastnosti, ki so za ta niz filtrov opisane v tabeli 2. Rezultat načrtovanja, to so vrednosti posameznih koeficientov, so prikazane v tabeli 3. Celoštevilski koeficienti v tabeli 3 so pomnoženi z 2^7 . Pri uporabi algoritmov za redukcijo baze je prikazan rezultat po uporabi Babaievega algoritma (in predtem algoritma LLL) in nato še po uporabi hevrstike na dobljenem rezultatu. Opazimo, da je bila s hevrstiko spremenjena vrednost 1 koeficientu (h(3) oz. h(42)). Če med metodami primerjamo le vrednosti koeficientov, vidimo, da velikih razlik med rezultati ni. V vrednostih se med seboj razlikujejo največ štirje koeficienti.

	zaokroževanje	algoritmi za redukcijo baze		MILP
		Babaiev alg.	Babaiev alg. s hevrstiko	
h(0)=h(44)	0	0	0	0
h(1)=h(43)	-1	0	0	0
h(2)=h(42)	0	0	0	0
h(3)=h(41)	1	1	0	1
h(4)=h(40)	0	0	0	0
h(5)=h(39)	-1	-1	-1	-1
h(6)=h(38)	-1	-1	-1	-1
h(7)=h(37)	1	1	1	1
h(8)=h(36)	1	1	1	1
h(9)=h(35)	-1	-1	-1	-1
h(10)=h(34)	-2	-2	-2	-2
h(11)=h(33)	0	0	0	0
h(12)=h(32)	3	3	3	3
h(13)=h(31)	1	1	1	0
h(14)=h(30)	-4	-4	-4	-4
h(15)=h(29)	-2	-2	-2	-2
h(16)=h(28)	5	5	5	5
h(17)=h(27)	5	5	5	5
h(18)=h(26)	-6	-6	-6	-6
h(19)=h(25)	-12	-12	-12	-12
h(20)=h(24)	6	6	6	7
h(21)=h(23)	40	40	40	40
h(22)	58	58	58	57

Tabela 3: Izračunani koeficienti za filter dolžine 45 (prepustni pas $\omega_p = [0; 0, 4\pi]$, zaporni pas $\omega_s = [0, 50\pi; \pi]$, utež funkcije napake v prepustnem pasu $W(\omega_p) = 1$ in zapornem pasu $W(\omega_s) = 1$). Tabela prikazuje koeficiente izračunane na 3 načine: z zaokroževanjem koeficientov, ki so izračunani z Remezovim algoritmom (1. stolpec), s pomočjo Babaievega algoritma in Babaievega algoritma s hevrstiko (2. in 3. stolpec) in s pomočjo MILP (4. stolpec).

Koeficienti prikazani v tabeli so pomnoženi z 2^7 .

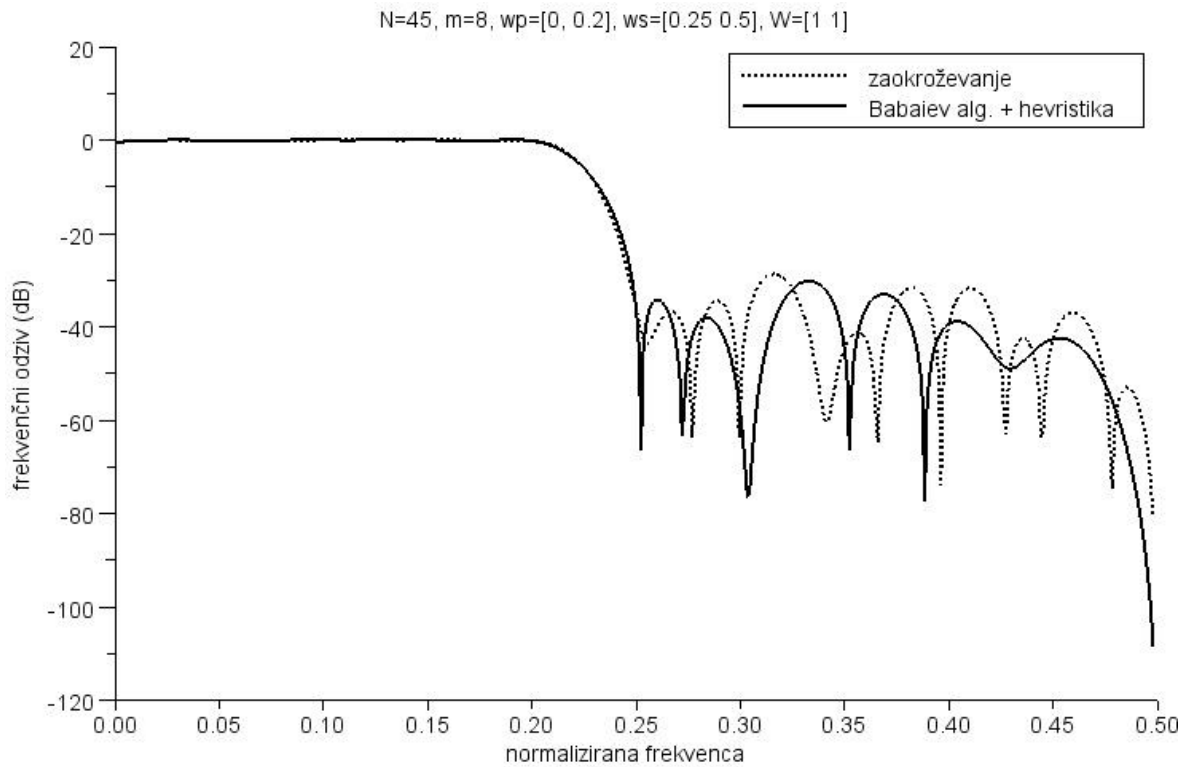
V tabeli 4 so predstavljene napake, ki jih dosežemo s posamezno metodo načrtovanja. Če gledamo razliko med želenim frekvenčnim odzivom in frekvenčnim odzivom, ki ga dobimo z izračunanimi koeficienti, vidimo, da je rezultat, ki ga da Babaiev algoritem boljši kot metoda z zaokroževanjem. Filter izračunan z uporabo Babaievega algoritma in hevristike je od optimalnega filtra slabši 1,05 dB v zapornem pasu in 0,03 dB slabši v prepustnem pasu. Hevristika je uspela zmanjšati maksimalno napako filtra. Filter izračunan z metodo zaokroževanja je od optimalnega filtra slabši 2,53 dB v zapornem in 0,02 dB slabši v prepustnem pasu. Za primerjavo so v tabeli tudi podatki za koeficiente, ki so izračunani z uporabo Remezovega algoritma in niso končne dolžine.

	zaokroževanje	algoritmi za redukcijo baze		MILP	Remez
		Babaiev alg.	Babaiev alg. s hevristiko		
max. napaka	0,037059	0,032884	0,032668	0,028847	0,009638
napaka prepustni pas	0,031250 (0,267279 dB)	0,029542 (0,252882 dB)	0,032668 (0,279217 dB)	0,028847 (0,247016 dB)	0,007136 (0,061765 dB)
napaka zaporni pas	0,037059 (28,622046 dB)	0,032884 (29,660432 dB)	0,031262 (30,099566 dB)	0,027691 (31,153227 dB)	0,009638 (40,319905 dB)
čas izvajanja	0,064 s	0,192 s	0,210 s	44,431 s	0,057 s

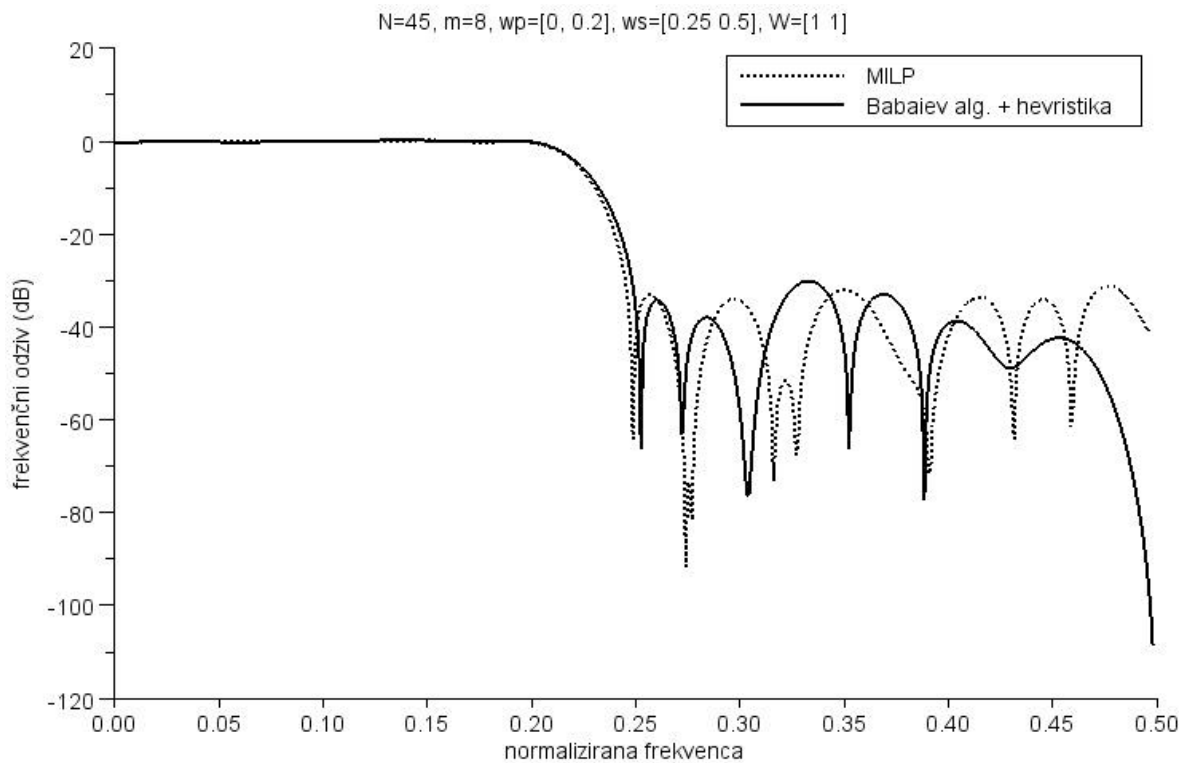
Tabela 4: Prikaz napake za filter dolžine 45 (prepustni pas $\omega_p = [0; 0, 4\pi]$, zaporni pas $\omega_s = [0, 50\pi; \pi]$, utež funkcije napake v prepustnem pasu $W(\omega_p) = 1$ in zapornem pasu $W(\omega_s) = 1$). Koeficienti filtrov so dolžine 8 bitov.

Omeniti je potrebno tudi čas izvajanja posamezne metode. Metoda z uporabo MILP ima v najslabšem primeru eksponentno časovno kompleksnost. Čas načrtovanja s to metodo je bil dobrih 44 s, čas načrtovanja z uporabo Babaievega algoritma in hevristike pa približno 200 ms, pri čemer je hevristika trajala 10% tega časa.

Graf 1 in graf 2 prikazujeta frekvenčne odzive filtrov načrtovanih z eno od omenjenih metod. Uporabljeno je logaritemsko merilo in x os grafa, ki prikazuje frekvence, je normalizirana. Na grafu 1 sta skupaj prikazana frekvenčna odziva filtrov načrtovanih z metodo zaokroževanja in Babaievega algoritma s hevristiko. V zapornem pasu vidimo, da so valovi filtra načrtovanega z metodo zaokroževanja malenkost višji. Podobno je opazno na grafu 2, kjer je primerjan frekvenčni odziv filtra načrtovanega z MILP in filtra načrtovanega z Babaievim algoritmom in hevristiko. Tudi tukaj opazimo, da je eden od valov filtra, ki je načrtovan s slednjo metodo, nekoliko višji.



Graf 1: Primerjava frekvenčnega odziva dveh filtrov. Prvi filter ima koeficiente izračunane s pomočjo Babaievega algoritma in hevrstike, drugi filter ima zaokrožene koeficiente izračunane s pomočjo Remezovega algoritma. Dolžina koeficientov je 8 bitov.



Graf 2: Primerjava frekvenčnega odziva dveh filtrov. Prvi filter ima koeficiente izračunane s pomočjo Babaievega algoritma in hevrstike, drugi filter ima koeficiente izračunane z MILP. Dolžina koeficientov je 8 bitov.

Metode načrtovanja je bila preizkušena še na vseh nizih filtrov, torej skupno je bilo zasnovanih 30 filtrov. V tabeli 5 so predstavljeni rezultati načrtovanja za 6 filtrov niza A. Zanimala nas je predvsem napaka aproksimacije želenega frekvenčnega odziva. Če gledamo le maksimalno napako, potem je bila metoda načrtovanja z uporabo Babaievega algoritma in hevrstike baze mreže v 5 primerih boljša in v enem primeru enaka kot metoda načrtovanja z zaokroževanjem. V dveh primerih smo dobili enak rezultat kot z metodo MILP (torej optimalen rezultat).

		niz filtrov A			
		$(\omega_p = [0; 0,40\pi], \omega_s = [0,50\pi; \pi], W(\omega_p) = 1, W(\omega_s) = 1)$			
		zaokroževanje	Babaiev alg. in hevrstika	MILP	Remez
M=25, b=7	max. napaka napaka prepustni pas napaka zaporni pas čas izvajanja	0,078125 0,071233 (0,597680 dB) 0,078125 (22,144199 dB)	0,065237 0,053737 (0,454648 dB) 0,065237 (23,710069 dB) 0,158 s (6,962% časa hevrstika)	0,065237 0,053737 (0,454648 dB) 0,065237 (23,710069 dB) 2,374 s	0,045058 0,039771 (0,338754 dB) 0,045058 (26,924614 dB)
M=35, b=8	max. napaka napaka prepustni pas napaka zaporni pas čas izvajanja	0,032668 0,032668 (0,279217 dB) 0,031262 (30,099566 dB)	0,032668 0,032668 (0,279217 dB) 0,031262 (30,099566 dB) 0,141 s (7,801% časa hevrstika)	0,029966 0,029966 (0,256458 dB) 0,027432 (31,234851 dB) 18,247 s	0,019706 0,015978 (0,137690 dB) 0,019706 (34,107958 dB)
M=45, b=8	max. napaka napaka prepustni pas napaka zaporni pas čas izvajanja	0,037059 0,031250 (0,267279 dB) 0,037059 (28,622046 dB)	0,032668 0,032668 (0,279217 dB) 0,031262 (30,099566 dB) 0,192 s (9,375% časa hevrstika)	0,028847 0,028847 (0,247016 dB) 0,027691 (31,153227 dB) 44,431 s	0,009638 0,007136 (0,061765 dB) 0,009638 (40,319905 dB)
M=45, b=10	max. napaka napaka prepustni pas napaka zaporni pas čas izvajanja	0,013872 0,011812 (0,102001 dB) 0,013872 (37,157368 dB)	0,010182 0,010182 (0,087997 dB) 0,010156 (39,865142 dB) 0,196 s (10,714% časa hevrstika)	0,010182 0,010182 (0,087997 dB) 0,010156 (39,865142 dB) 265,654 s	0,009638 0,007136 (0,061765 dB) 0,009638 (40,319905 dB)
M=55, b=10	max. napaka napaka prepustni pas napaka zaporni pas čas izvajanja	0,00979 0,009790 (0,084623 dB) 0,008976 (40,938399 dB)	0,009144 0,008129 (0,070321 dB) 0,009144 (40,776879 dB) 0,240 s (9,583% časa hevrstika)	0,008296 0,008296 (0,0717609 dB) 0,008257 (41,663554 dB) 1130,051 s	0,004472 0,002945 (0,025547 dB) 0,004472 (46,989442 dB)
M=65, b=11	max. napaka napaka prepustni pas napaka zaporni pas čas izvajanja	0,008789 0,008789 (0,076007 dB) 0,003411 (49,342723 dB)	0,004883 0,004883 (0,042308 dB) 0,004462 (47,009493 dB) 0,371 s (9,434% časa hevrstika)	0,003938 0,003816 (0,0330823 dB) 0,003938 (48,094486 dB) 5140,320 s	0,001965 0,001128 (0,009793 dB) 0,001965 (54,131326 dB)

Tabela 5: Rezultati načrtovanja za niz filtrov A

Filtri niza B so imeli enake lastnosti kot filtri niza A, le da je bila napaka po pasovih različno utežena. Napaka v prestopnem pasu je imela utež 1, v zapornem pa 10. Kot je že bilo omenjeno, metoda načrtovanja z uporabo Babaievega algoritma in hevrstike pri aproksimaciji ne more upoštevati utežnostne funkcije. Zato smo kot želeni frekvenčni odziv vzeli frekvenčni odziv izračunan z Remezovim algoritmom. Rezultati kažejo, da v nobenem primeru ne dobimo dobrih rezultatov. Rezultati so primerljivi z metodo načrtovanja z zaokroževanjem.

		niz filtrov B			
		$(\omega_p = [0; 0,40\pi], \omega_s = [0,50\pi; \pi], W(\omega_p) = 1, W(\omega_s) = 10)$			
		zaokroževanje	Babaiev alg. in hevrstika	MILP	Remez
M=25, b=8	napaka prepustni pas napaka zaporni pas čas izvajanja	0,140625 (1,142858 dB) 0,032914 (29,652412 dB)	0,143501 (1,164729 dB) 0,039063 (28,164799 dB) 0,115 s (7,826% časa hevrstika)	0,154246 (1,245968 dB) 0,015234 (36,343721 dB) 2,100 s	0,123032 (1,007840 dB) 0,014649 (36,683908 dB)
M=35, b=9	napaka prepustni pas napaka zaporni pas čas izvajanja	0,074219 (0,621855 dB) 0,015902 (35,970695 dB)	0,066406 (0,558454 dB) 0,022931 (32,791497 dB) 0,151 s (10,596% časa hevrstika)	0,073276 (0,614228 dB) 0,007313 (42,718089 dB) 41,073 s	0,052807 (0,446976 dB) 0,006920 (43,198273 dB)
M=45, b=9	napaka prepustni pas napaka zaporni pas čas izvajanja	0,038010 (0,324030 dB) 0,011719 (38,622374 dB)	0,034366 (0,293486 dB) 0,015908 (35,967865 dB) 0,181 s (9,392 časa hevrstika)	0,052641 (0,445606 dB) 0,005681 (44,911504 dB) 246,493 s	0,021128 (0,181608 dB) 0,003122 (50,111271 dB)
M=45, b=11	napaka prepustni pas napaka zaporni pas čas izvajanja	0,024179 (0,207514 dB) 0,006177 (44,184338 dB)	0,025522 (0,218903 dB) 0,006282 (44,037975 dB) 0,213 s (5,634% časa hevrstika)	0,026655 (0,228491 dB) 0,002673 (51,460021 dB) 370,673 s	0,021128 (0,181608 dB) 0,003122 (50,111271 dB)
M=55, b=11	napaka prepustni pas napaka zaporni pas čas izvajanja	0,010579 (0,091405 dB) 0,006234 (44,104361 dB)	0,011861 (0,102419 dB) 0,004939 (46,127145 dB) 0,221 s (9,955% časa hevrstika)	0,016660 (0,143515 dB) 0,001643 (55,687249 dB) 1477,151 s	0,007932 (0,068627 dB) 0,001351 (57,383789 dB)
M=65, b=12	napaka prepustni pas napaka zaporni pas čas izvajanja	0,005313 (0,046023 dB) 0,002421 (52,319283 dB)	0,005351 (0,046356 dB) 0,002632 (51,595627 dB) 0,287 s (9,756 časa hevrstika)	*	0,003542 (0,030708 dB) 0,000684 (63,296886 dB)

Tabela 6: Rezultati načrtovanja za niz filtrov B

*metoda z uporabo MILP po 72 urah ni vrnila rezultata

Metode načrtovanja smo preizkusili tudi z načrtovanjem pasovnozaporovnih filtrov. Rezultati so prikazani v tabeli 7. Napaka je bila po pasovih enakomerno utežena. Metoda načrtovanja z uporabo Babaievega algoritma in hevristike je bila v vseh primerih boljša kot metoda z zaokroževanjem. Maksimalna napaka je bila namreč manjša kot pri metodi z zaokroževanjem. V enem primeru (za filter s 25 koeficienti in dolžino besede 7 bitov) smo dobili enak rezultat kot pri metodi z MILP.

		niz filtrov C $(\omega_{p1} = [0; 0,24\pi], \omega_s = [0,40\pi; 0,68\pi], \omega_{p2} = [0; 0,40\pi],$ $W(\omega_{p1}) = 1, W(\omega_s) = 1, W(\omega_{p2}) = 1)$			
		zaokroževanje	Babaiev alg. in hevristika	MILP	Remez
M=25, b=7	max. napaka napaka prepustni pas 1 napaka zaporni pas napaka prepustni pas 2 čas izvajanja	0,041507 0,020126 (0,173077 dB) 0,039566 (28,053590 dB) 0,041507 (0,353243 dB)	0,036676 0,035751 (0,305108 dB) 0,036676 (28,712324 dB) 0,025882 (0,221947 dB) 0,103 s (8,738% časa hevristika)	0,036676 0,035751 (0,305108 dB) 0,036676 (28,712324 dB) 0,025882 (0,221947 dB) 1,310 s	0,014417 0,012851 (0,110915 dB) 0,014417 (36,822330 dB) 0,013856 (0,119523 dB)
M=35, b=8	max. napaka napaka prepustni pas 1 napaka zaporni pas napaka prepustni pas 2 čas izvajanja	0,046875 0,046875 (0,397897 dB) 0,031312 (30,085720 dB) 0,021800 (0,187317 dB)	0,03125 0,031250 (0,267279 dB) 0,027220 (31,302189 dB) 0,031250 (0,267279 dB) 0,144 s (8,333% časa hevristika)	0,016767 0,015625 (0,134668 dB) 0,016767 (35,510893 dB) 0,015625 (0,134668 dB) 12,230 s	0,003069 0,002633 (0,022840 dB) 0,003069 (50,258656 dB) 0,003021 (0,026197 dB)
M=45, b=8	max. napaka napaka prepustni pas 1 napaka zaporni pas napaka prepustni pas 2 čas izvajanja	0,030466 0,028970 (0,248058 dB) 0,030466 (30,323676 dB) 0,015625 (0,134668 dB)	0,029409 0,017706 (0,152448 dB) 0,029409 (30,630265 dB) 0,025045 (0,214861 dB) 0,231 s (6,926% časa hevristika)	0,016085 0,014451 (0,124622 dB) 0,011794 (38,566678 dB) 0,016085 (0,138601 dB) 303,998 s	0,000958 0,000672 (0,005835 dB) 0,000958 (60,369476 dB) 0,000830 (0,007210 dB)
M=45, b=10	max. napaka napaka prepustni pas 1 napaka zaporni pas napaka prepustni pas 2 čas izvajanja	0,00993 0,007813 (0,067595 dB) 0,005372 (45,396831 dB) 0,009930 (0,085825 dB)	0,006843 0,006843 (0,059238 dB) 0,006312 (43,996101 dB) 0,005859 (0,050745 dB) 0,193 s (10,881% časa hevristika)	0,004761 0,004170 (0,036145 dB) 0,004761 (46,446036db) 0,003906 (0,033861 dB) 979,619 s	0,000958 0,000672 (0,005835 dB) 0,000958 (60,369476 dB) 0,000830 (0,007210 dB)
M=55, b=10	max. napaka napaka prepustni pas 1 napaka zaporni pas napaka prepustni pas 2 čas izvajanja	0,010839 0,003741 (0,032435 dB) 0,010839 (39,299963 dB) 0,009766 (0,084412 dB)	0,007636 0,004872 (0,042211 dB) 0,007636 (42,343242 dB) 0,007033 (0,060878 dB) 0,222 s (9,009% časa hevristika)	0,004365 0,004365 (0,037831 dB) 0,003788 (48,431801 dB) 0,003941 (0,034164 dB) 19626,482 s	0,000294 0,000182 (0,001580 dB) 0,000294 (70,618690 dB) 0,000244 (0,002118 dB)
M=65, b=11	max. napaka napaka prepustni pas 1 napaka zaporni pas napaka prepustni pas 2 čas izvajanja	0,004883 0,004352 (0,037723 dB) 0,003526 (49,055157 dB) 0,004883 (0,042308 dB)	0,003438 0,003438 (0,029809 dB) 0,003135 (50,075243 dB) 0,002930 (0,025410 dB) 0,265 s (9,811% časa hevristika)	*	0,000094 0,000051 (0,000439 dB) 0,000094 (80,578461 dB) 0,000073 (0,000638 dB)

Tabela 7: Rezultati načrtovanja za niz filtrov C

*metoda z uporabo MILP po 72 urah ni vrnila rezultata

Niz filtrov D je imel za razliko od niza filtrov C le različno uteženo funkcijo napake. V zapornem pasu je bila napaka utežena s faktorjem 10, v obeh prepustnih pasovih pa z 1. Ponovno se je izkazalo (kot je razvidno iz rezultatov v tabeli 8), da metoda načrtovanja z Babaievim algoritmom in hevrstiko v primerih, ko imamo filtre, kjer napako po pasovih različno utežimo, ni dobra. Po kakovosti aproksimacije je primerljiva z metodo z zaokroževanjem.

niz filtrov D					
$(\omega_{p1} = [0; 0,24\pi], \omega_s = [0,40\pi; 0,68\pi], \omega_{p2} = [0; 0,40\pi],$					
$W(\omega_{p1}) = 1, W(\omega_s) = 10, W(\omega_{p2}) = 1)$					
		zaokroževanje	Babaiev alg. in hevrstika	MILP	Remez
M=25, b=8	napaka prepustni pas 1 napaka zaporni pas napaka prepustni pas 2 čas izvajanja	0,062500 (0,526579 dB) 0,014408 (36,827757 dB) 0,062500 (0,526579 dB)	0,062500 (0,526579 dB) 0,014408 (36,827757 dB) 0,062500 (0,526579 dB) 0,092 s (5,435% časa hevrstika)	0,078125 (0,653382 dB) 0,007966 (41,975194 dB) 0,078125 (0,653382 dB) 5,257 s	0,048097 (0,408028 dB) 0,005610 (45,020218 dB) 0,050151 (0,425036 dB)
M=35, b=9	napaka prepustni pas 1 napaka zaporni pas napaka prepustni pas 2 čas izvajanja	0,015987 (0,137765 dB) 0,012202 (38,271575 dB) 0,027344 (0,234316 dB)	0,023438 (0,201227 dB) 0,011957 (38,447836 dB) 0,023438 (0,201227 dB) 0,131 s (9,924% časa hevrstika)	0,031250 (0,267279 dB) 0,003302 (49,624459 dB) 0,031250 (0,267279 dB) 91,463 s	0,010438 (0,090190 dB) 0,001355 (57,362088 dB) 0,011482 (0,099160 dB)
M=45, b=9	napaka prepustni pas 1 napaka zaporni pas napaka prepustni pas 2 čas izvajanja	0,017680 (0,152227 dB) 0,010906 (39,247064 dB) 0,018901 (0,162638 dB)	0,014144 (0,121992 dB) 0,011689 (38,644406 dB) 0,011097 (0,095854 dB) 0,166 s (9,639% časa hevrstika)	0,022836 (0,196120 dB) 0,002552 (51,862387 dB) 0,025071 (0,215079 dB) 1184,371 s	0,002238 (0,019415 dB) 0,000335 (69,495145 dB) 0,002672 (0,023175 dB)
M=45, b=11	napaka prepustni pas 1 napaka zaporni pas napaka prepustni pas 2 čas izvajanja	0,004244 (0,036785 dB) 0,003222 (49,836784 dB) 0,003906 (0,033863 dB)	0,003761 (0,032603 dB) 0,002954 (50,591109 dB) 0,004551 (0,039436 dB) 0,172 s (11,628% časa hevrstika)	0,006396 (0,055378 dB) 0,000745 (62,556875 dB) 0,005306 (0,045966 dB) 3159,022 s	0,002238 (0,019415 dB) 0,000335 (69,495145 dB) 0,002672 (0,023175 dB)
M=55, b=11	napaka prepustni pas 1 napaka zaporni pas napaka prepustni pas 2 čas izvajanja	0,002167 (0,018804 dB) 0,003710 (48,612609 dB) 0,004379 (0,037955 dB)	0,002167 (0,018804 dB) 0,003710 (48,612609 dB) 0,004379 (0,037955 dB) 0,225 s (6,667% časa hevrstika)	0,005180 (0,044877 dB) 0,000618 (64,18023 dB) 0,005859 (0,050742 dB) 96780,802 s	0,000686 (0,005956 dB) 0,000116 (78,686009 dB) 0,000879 (0,007631 dB)
M=65, b=12	napaka prepustni pas 1 napaka zaporni pas napaka prepustni pas 2 čas izvajanja	0,002650 (0,022991 dB) 0,002013 (53,921818 dB) 0,001836 (0,015929 dB)	0,002275 (0,019736 dB) 0,002204 (53,136520 dB) 0,001851 (0,016059 dB) 0,317 s (8,517% časa hevrstika)	*	0,000147 (0,001278 dB) 0,000029 (90,722240 dB) 0,000208 (0,001803 dB)

Tabela 8: Rezultati načrtovanja za niz filtrov D

*metoda z uporabo MILP po 72 urah ni vrnila rezultata

Niz filtrov E predstavlja 6 nizkoprepustnih filtrov z enakomerno uteženo napako. Filtri so podobno definirani kot filtri niza A, le da sta prepustni in zaporni pas nekoliko ožja. Rezultati načrtovanja so prikazani v tabeli 9. Metoda načrtovanja z uporabo Babaievega algoritma in hevristike se je izkazala za dobro. V 5 primerih smo dobili boljše rezultate, v enem primeru pa enak rezultat kot pri metodi z zaokroževanjem. Optimalnega rezultata nismo dobili v nobenem primeru, rezultat za filter (N=35, m=8) se je optimalnemu rezultatu praktično povsem približal.

		niz filtrov E			
		$(\omega_p = [0,02\pi; 0,42\pi], \omega_s = [0,52\pi; 0,98\pi], W(\omega_p) = 1, W(\omega_s) = 1)$			
		zaokroževanje	Babaiev alg. in hevristika	MILP	Remez
M=25, b=7	max. napaka napaka prepustni pas napaka zaporni pas čas izvajanja	0,077633 0,077633 (0,649416 dB) 0,053909 (25,366695 dB)	0,062141 0,056496 (0,477359 dB) 0,062141 (24,132447 dB) 0,099 s (9,091 časa hevristika)	0,06071 0,057537 (0,485911 dB) 0,060710 (24,334795 dB) 2,558 s	0,047752 0,042683 (0,363045 dB) 0,047752 (26,420172 dB)
M=35, b=8	max. napaka napaka prepustni pas napaka zaporni pas čas izvajanja	0,046934 0,037344 (0,318453 dB) 0,046934 (26,570201 dB)	0,033004 0,033004 (0,282036 dB) 0,023939 (32,417859 dB) 0,148 s (12,838 časa hevristika)	0,032888 0,032888 (0,281065 dB) 0,032028 (29,889404 dB) 31,449 s	0,023334 0,019738 (0,169770 dB) 0,023340 (32,637809 dB)
M=45, b=8	max. napaka napaka prepustni pas napaka zaporni pas čas izvajanja	0,035784 0,030685 (0,262519 dB) 0,035784 (28,926267 dB)	0,035784 0,030685 (0,262519 dB) 0,035784 (28,926267 dB) 0,177 s (6,780 časa hevristika)	0,028987 0,027058 (0,231899 dB) 0,028987 (30,755935 dB) 179,119 s	0,010087 0,007526 (0,065127 dB) 0,010087 (39,924897 dB)
M=45, b=10	max. napaka napaka prepustni pas napaka zaporni pas čas izvajanja	0,015263 0,014410 (0,124275 dB) 0,015263 (36,327077 dB)	0,011185 0,011185 (0,096610 dB) 0,010983 (39,185925 dB) 0,183 s (9,836 časa hevristika)	0,010104 0,010104 (0,087322 dB) 0,009345 (40,588414 dB) 208,619 s	0,010087 0,007526 (0,065127 dB) 0,010087 (39,924897 dB)
M=55, b=10	max. napaka napaka prepustni pas napaka zaporni pas čas izvajanja	0,011802 0,009135 (0,078990 dB) 0,011802 (38,561246 dB)	0,009119 0,009119 (0,078847 dB) 0,008865 (41,046321 dB) 0,239 s (11,297 časa hevristika)	0,008199 0,008199 (0,070925 dB) 0,007929 (42,015632 dB) 497,376 s	0,004894 0,003675 (0,031862 dB) 0,004894 (46,206233 dB)
M=65, b=11	max. napaka napaka prepustni pas napaka zaporni pas čas izvajanja	0,006887 0,006887 (0,059618 dB) 0,004248 (47,436447 dB)	0,00541 0,005196 (0,045012 dB) 0,005410 (45,335582 dB) 0,283 s (9,894 časa hevristika)	0,004047 0,004047 (0,035081 dB) 0,004014 (47,928453 dB) 9250,299 s	0,002588 0,001824 (0,015830 dB) 0,002588 (51,740417 dB)

Tabela 9: Rezultati načrtovanja za niz filtrov E

6. Sklepne ugotovitve

Optimalni filtri KEO z linearno fazo in koeficienti končne dolžine so bili zasnovani z uporabo algoritmov za redukcijo baze. Pri prvi metodi sta bila za reševanje polinomske aproksimacije uporabljena Babaiev algoritem, ki dobi kot vhod LLL reducirano bazo, in hevrastika, ki poskuša rezultat Babaievega algoritma približati L_∞ normi. Pri drugi metodi je bil uporabljen algoritem LLL za reševanje sistema diofantskih enačb s spodnjo in zgornjo mejo.

Testiranja so pokazala, da je uporaba algoritmov za reducirano bazo mreže pri polinomski aproksimaciji s koeficienti končne dolžine smiselna. Metoda z reševanjem sistema diofantskih enačb sicer ni dala rešitev. Uspešna je bila metoda, ki rešuje problem načrtovanja filtrov z uporabo Babaievega algoritma in hevrastike. To metodo smo primerjali z metodo zaokroževanja koeficientov, ki jih da Remezov algoritem in so suboptimalni, in z metodo z uporabo MILP, ki da vedno optimalne koeficiente.

Izkazalo se je, da je metoda, ki pri načrtovanju uporablja Babaiev algoritem in hevrastiko, zelo hitra. Vendar ima načrtovanje optimalnih filtrov KEO določene posebnosti, ki jih pri običajni polinomski aproksimaciji ponavadi ni potrebno upoštevati. Pri zasnovi filtrov želimo utežiti napako aproksimacije. Največkrat želimo, da je napaka v zapornem pasu filtra bolj utežena, oz. da so valovi filtra v zapornem pasu manjši kot v prepustnem. Velika slabost metode, ki bazira na polinomski aproksimaciji z uporabo algoritmov za redukcijo baze je, da utežnostne funkcije pri zasnovi filtrov ne moremo upoštevati. Pri testiranju je bil zato v teh primerih aproksimiran frekvenčni odziv filtra, ki je rezultat Remezovega algoritma. To se ni izkazalo za dobro. Pri filtrih, ki imajo enakomerno uteženo napako aproksimacije po vseh frekvenčnih pasovih, se je omenjena metoda načrtovanja izkazala za dobro. V nekaterih primerih smo dobili optimalne rezultate, v praktično vseh primerih pa boljše rezultate kot pri metodi z zaokroževanjem. Problem te metode je tudi ta, da pri rezultatu ne vemo, ali je optimalen. Optimalnost rezultata je pokazala le primerjava z metodo, ki uporablja MILP.

Metodi načrtovanja filtrov z uporabo algoritmov za redukcijo baze mreže puščata odprte možnosti za nadaljnje raziskave. Metoda polinomske aproksimacije z Babaievim algoritmom uporablja hevrastiko, ki poskuša rezultat v L_2 normi izboljšati in se s preiskovanjem okolice približati optimalnemu rezultatu v L_∞ normi. Izboljšava hevrastike bi v določenih primerih izboljšala rezultat. Bistvena težava, ki jo ta metoda načrtovanja prinaša in bi jo bilo potrebno za resnejšo uporabo metode rešiti, je nezmožnost upoštevavanja utežnostne funkcije.

Algoritmi za redukcijo baze mreže so sicer že bili uspešno uporabljeni pri načrtovanju optimalnih filtrov KEO z linearno fazo in koeficienti končne dolžine z metodo MILP [10], ki je sicer najbolj razširjena za načrtovanje tovrstnih filtrov. Algoritem LLL je služil pri izračunu spodnje meje aproksimacijske napake, ki bistveno pohitri branch-and-bound algoritem. V tej smeri bi bilo smiselno preizkusiti še algoritme za redukcijo baze mreže v L_∞ normi (eden izmed teh algoritmov je opisan v [13]). V literaturi zaenkrat še ni zaslediti množične uporabe algoritmov za redukcijo baze v L_∞ normi na povsem praktičnih aplikacijah. Razlog je najverjetneje eksponentna zahtevnost tovrstnih algoritmov.

7. Priloge

7.1. *Seznam slik*

Slika 1: Prikaz dvodimenzionalne mreže	8
Slika 2: Primer Gram-Schmidtovega postopka ortogonalizacije	10
Slika 3: Prikaz Babaievega algoritma za reševanje CVP	15
Slika 4: Amplitudni odziv za primer nizkoprepustnega filtra	17
Slika 5: Prikaz želenega (idealnega) amplitudnega odziva za štiri najbolj pogoste vrste filtrov	18

7.2. *Seznam tabel*

Tabela 1: Tabela s formulami za izračun realnega frekvenčnega odziva	21
Tabela 2: Lastnosti filtrov, ki pripadajo posameznemu nizu filtrov.	36
Tabela 3: Izračunani koeficienti za filter dolžine 45.	37
Tabela 4: Prikaz napake za filter dolžine 45.	38
Tabela 5: Rezultati načrtovanja za niz filtrov A.	40
Tabela 6: Rezultati načrtovanja za niz filtrov B	41
Tabela 7: Rezultati načrtovanja za niz filtrov C	42
Tabela 8: Rezultati načrtovanja za niz filtrov D.	43
Tabela 9: Rezultati načrtovanja za niz filtrov E	44

7.3. *Seznam grafov*

Graf 1: Primerjava frekvenčnega odziva dveh filtrov (metoda zaokroževanja in Babaiev algoritem).	39
Graf 2: Primerjava frekvenčnega odziva dveh filtrov (metoda MILP in Babaiev algoritem).	39

7.4. Izseki izvorne kode v Scilabu

7.4.1. Algoritem LLL

```

function [B,indeks] = lll(B)

    [row,col] = size(B);

    for i=1:col
        indeks(i) = i;
    end

    B_ = zeros(row,1);
    b_ = zeros(row,col);

    b_(1:row,1) = B(1:row,1);
    B_(1) = sum( b_(1:row,1) .* b_(1:row,1));

    mi = zeros(row,col);

    for i=2:col
        b_(1:row,i) = B(1:row,i);
        for j=1:i-1
            mi(i,j) = sum( B(1:row,i) .* b_(1:row,j) ) / B_(j);
            b_(1:row,i) = b_(1:row,i) - (mi(i,j) .* b_(1:row,j));
        end
        B_(i) = sum( b_(1:row,i) .* b_(1:row,i));
    end

    k=2;
    n=col;
    while (k<=n)
        //Procedura A
        if abs(round(mi(k,k-1)*100)/100) > 1/2 then
            r = round(mi(k,k-1));
            B(1:row,k) = B(1:row,k) - r .* B(1:row,k-1);
            for j=1:1:k-2
                mi(k,j) = mi(k,j) - r*mi(k-1,j);
            end
            mi(k,k-1) = mi(k,k-1) - r;
        end

        if ( B_(k) < ( (3/4 - mi(k,k-1)^2) * B_(k-1) ) ) then
            // Procedura B
            u=mi(k,k-1);
            B1 = B_(k) + u^2*B_(k-1);
            mi(k,k-1) = u*B_(k-1)/B1;
            B_(k) = B_(k-1)*B_(k)/B1;
            B_(k-1) = B1;

            //zamenjava vektorjev
            ind_temp1 = indeks(k-1);
            ind_temp2 = indeks(k);
            indeks(k-1) = ind_temp2;
            indeks(k) = ind_temp1;

            btemp = B(1:row, k-1);
            B(1:row, k-1) = B(1:row, k);
            B(1:row, k) = btemp;

            //zamenjava elementov
            for j=1:k-2
                mitemp = mi(k,j);
                mi(k,j) = mi(k-1,j);
                mi(k-1,j) = mitemp;
            end
        end
    end
end

```

```

end

for i = k+1:n
    t = mi(i,k-1);
    mi(i,k-1) = mi(k,k-1)*mi(i,k-1) + mi(i,k) - mi(i,k)*mi(k,k-1)*u;
    mi(i,k) = t - u*mi(i,k);
end

if (k>2) then
    k=k-1;
end

else
    for l=k-2:-1:1
        // Procedura A
        if abs(mi(k,l)) > 1/2 then
            abs(mi(k,l));
            r = round(mi(k,l));
            B(1:row,k) = B(1:row,k) - r .* B(1:row,l);
            for j=1:l-1
                mi(k,j) = mi(k,j) - r*mi(l,j);
            end
            mi(k,l) = mi(k,l) - r;
        end
    end
    k=k+1;
end
end

endfunction

```

7.4.2. Babaiev algoritem

```

function [w, c] = babai(B, v)

[ row, col ] = size(B)

//Gram-Schmidt

B_ = zeros(row,1);
b_ = zeros(row,col);

b_(1:row,1) = B(1:row,1);
B_(1) = sum( b_(1:row,1) .* b_(1:row,1) );

mi = zeros(row,col);

for i=2:col
    b_(1:row,i) = B(1:row,i);
    for j=1:i-1
        mi(i,j) = sum( B(1:row,i) .* b_(1:row,j) ) / B_(j);
        b_(1:row,i) = b_(1:row,i) - (mi(i,j) .* b_(1:row,j));
    end
    B_(i) = sum( b_(1:row,i) .* b_(1:row,i) );
end

//end Gram-Schmidt

n = col;

t = v;

for j=n:-1:1
    c(j) = round(sum( t .* b_(1:row,j) ) / sum( b_(1:row,j) .* b_(1:row,j) ));
    t = t - c(j) * B(1:row,j);
end

```

```

end
w = v - t;
endfunction

```

7.4.3. Hevristika

```

flag = %T;
while flag == %T,
    for i=1:2*n
        st_vektor = round(i/2);
        w_(1:ind_wp2-ind_wp1+1, i) = w(ind_wp1:ind_wp2) + (-
1)^i*B(ind_wp1:ind_wp2, st_vektor);
        w_(ind_wp2-ind_wp1+2:ind_wp2-ind_wp1+ind_ws2-ind_ws1+2, i) =
w(ind_ws1:ind_ws2) + (-1)^i*B(ind_ws1:ind_ws2, st_vektor);
        norme(i) = norm(w_(:,i)-D_, 'inf');
    end

    w_ref(1:ind_wp2-ind_wp1+1) = w(ind_wp1:ind_wp2);
    w_ref(ind_wp2-ind_wp1+2:ind_wp2-ind_wp1+ind_ws2-ind_ws1+2) =
w(ind_ws1:ind_ws2);

    [minimum, ind_min] = min(norme);
    if ( minimum < norm(w_ref-D_, 'inf') ) then
        st_vektor = round(ind_min/2);
        w = w + (-1)^ind_min*B(1:grid, st_vektor);
        result(st_vektor) = result(st_vektor) + (-1)^ind_min;
    else
        flag = %F;
    end
end

```

7.4.4. Metoda za izboljšanje LLL reducirane baze

```

function [B, sprememba] = lll_inf(A,B)

sprememba = 0;

C=A\B;
[row,col] = size(C);

for i=1:col
    for j=1:col
        if (i ~= j) then
            C_stari = C;
            B_stari = B;
            C(:, i) = C(:, i) + C(:, j);
            B = A * C;

            stara_norma = norm(B_stari(:, i), 'inf');
            nova_norma = norm(B(:, i), 'inf');

            if( nova_norma < stara_norma ) then

                sprememba = 1;
            else
                B = B_stari;
                C = C_stari;
            end
        end
    end
end

```

```
C_stari = C;
B_stari = B;
C(:, i) = C(:, i) - C(:, j);

B = A * C

    stara_norma = norm(B_stari(:, i), 'inf');
    nova_norma = norm(B(:, i), 'inf');

if( nova_norma < stara_norma ) then
    spremenba = 1;
else
    B = B_stari;
    C = C_stari;
end

    end
end
endfunction
```

8. Seznam uporabljenih virov

- [1] K. Aardal, "Lattice basis reduction and integer programming," Utrecht University: Information and Computing Sciences, Technical Report UU-CS 1997-37, 1999.
- [2] K. Aardal, C. Hurkens, and A. K. Lenstra, "Solving a System of Diophantine Equations with Lower and Upper Bounds on the Variables," *Mathematics of Operations Research*, vol. 25, pp. 427-442, avgust 2000.
- [3] L. Babai, "On Lovász' Lattice Reduction and the Nearest Lattice Point Problem," in *Proceedings of the 2nd Symposium of Theoretical Aspects of Computer Science (STACS '85)*, London, 1985, pp. 13-20.
- [4] N. Brisebarre and S. Chevillard, "Efficient polynomial L-approximations," *ARITH '07 Proceedings of the 18th IEEE Symposium on Computer Arithmetic*, pp. 169-176, junij 2007.
- [5] (2011, avgust) Filter (signal processing). [Online].
[http://en.wikipedia.org/wiki/Filter_\(signal_processing\)](http://en.wikipedia.org/wiki/Filter_(signal_processing))
- [6] S. Galbraith, "Algorithms for the Closest and Shortest Vector Problem," in *Mathematics of Public Key Cryptography.*, 2011, ch. 19.
- [7] H. Guillaum, "LLL: A Tool for Effective Diophantine Approximation," in *The LLL Algorithm Survey and Applications*, Brigitte Vallée and Phong Q. Nguyen, Eds. Berlin: Springer Verlag, 2010, pp. 215-265.
- [8] V. S. Ivanova, "Lattice Basis Reduction in Infinity Norm," Technische Universitaet Darmstadt Department of Mathematics, Bachelor Thesis.
- [9] D. M. Kodek, "Design of optimal finite wordlength FIR digital filters using integer programming techniques," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 38, no. 3, pp. 304- 308, junij 1980.
- [10] D. M. Kodek, "LLL Algorithm And the Optimal Finite Wordlength FIR Design," sprejeto v objavo v *IEEE Transactions on Signal Processing*, 2011.
- [11] D. M. Kodek, "Performance Limit of Finite Wordlength FIR Digital Filters," *IEEE Transactions on Signal Processing*, vol. 53, no. 7, pp. 2462- 2469, julij 2005.
- [12] A. K. Lenstra, H. W. Lenstra, and L. Lovász, "Factoring polynomials with rational coefficients," *Mathematische Annalen*, vol. 264, no. 4, pp. 515-634, december 1982.
- [13] L. Lovász and E. H. Scarf, "The Generalized Basis Reduction Algorithm," *Mathematics*

- of Operations Research*, vol. 17, no. 3, pp. 751-764, avgust 1992.
- [14] R. G. Lyons, "Finite Impulse Response Filters," in *Understanding Digital Signal Processing, Second Edition*.: Prentice Hall PTR, 2004, ch. 5, pp. 157-204.
- [15] J. H. McClellan and T. W. Parks, "A personal history of the Parks-McClellan algorithm," *IEEE Signal Processing Magazine*, vol. 2, p. 22, marec 2005.
- [16] J. G. Proakis and D. G. Manolakis, "Design of Digital Filters," in *Digital Signal Processing (Third Edition)*.: Prentice Hall, 1996, ch. 8, pp. 637-652.
- [17] I. Selesnick. Linear-Phase FIR Filter Design by Linear Programming. [Online].
<http://cnx.org/content/m11115/latest/>
- [18] N. P. Smart, *The algorithmic resolution of diophantine equations*.: Cambridge University Press, 1998, pp. 59-76.
- [19] D. Wuebben, D. Seethaler, J. Jalden, and G. Matz, "A Survey of Lattice Reduction Techniques with Applications to Wireless Communications," *IEEE Signal Processing Magazine*, vol. 28, no. 3, pp. 70-91, maj 2011.