Prepoznavanje govora u stvarnom vremenu pomoću FPGA

Labak, Matija

Master's thesis / Diplomski rad

2016

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **Josip Juraj Strossmayer University of Osijek, Faculty of Electrical Engineering, Computer Science and Information Technology Osijek / Sveučilište Josipa Jurja Strossmayera u Osijeku, Fakultet elektrotehnike, računarstva i informacijskih tehnologija Osijek**

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:200:200954

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2024-03-28**

Repository / Repozitorij:

Faculty of Electrical Engineering, Computer Science and Information Technology Osijek

zir.nsk.hr

Image not found or type unknown

SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU

ELEKTROTEHNIČKI FAKULTET OSIJEK

Sveučilišni studij

Matija Labak

PREPOZNAVANJE GOVORA U STVARNOM VREMENU POMOĆU FPGA

Diplomski rad

Osijek, 2016.

SADRŽAJ

1. UVOD 1
2. PREPOZNAVANJE GOVORA 3
2.1. Povijest i razvoj 3
2.2. Algoritmi i postupci sustava za prepoznavanje govora 7
2.3. Algoritmi za stvaranje značajki govornog signala 9
2.3.1. Linearno-prediktivna analiza – LPC 10
2.4. Algoritmi odabira jezičnih jedinica 11
2.4.1. Dinamičko savijanje vremena 11
2.4.2. Umjetne neuronske mreže 12
2.4.3. Skriveni Markovljevi modeli 13
2.4.4. Dubinske neuronske mreže 14
3. FPGA RAZVOJNI SUSTAV 15
3.1. Altium NanoBoard 15
3.2. Resursi dostupni FPGA dizajnu 18
3.2.1. Mikroprocesorski sustav TSK3000A 19
3.2.2. Analogni audio sustav 20
3.2.3. Audio CODEC 21
3.2.4. Nezavisni SRAM 22
3.3. Altium Designer 24
3.3.1. FPGA projekt 24
3.3.2. Ugrađeni projekt 27
3.3.3. Programiranje razvojne ploče 29
4. IMPLEMENTACIJA 32
4.1. Glasovi hrvatskog jezika 32
4.2. Generiranje osnovnih značajki govora 34
4.2.1. Kratkotrajna energija signala 34

4.2.2. Broj prelaska kroz nulu 35
4.2.3. Period impulsa 36
4.3. Model linearno-prediktivnog kodiranja za prepoznavanje govora 38
4.3.1. Pred-naglašavanje 40
4.3.2. Jednadžbe LPC analize 41
4.3.3. Autokorelacijska metoda 43
4.3.4. Levinson-Durbinova rekurzija 44
4.3.5. Algoritam strmog spusta 46
4.3.6. Mjerenje perioda impulsa iz signala greške 47
4.3.7. Normalizirana križna korelacija 48
4.3.8. Relativna udaljenost 48
4.4. Sustav za stvaranje usrednjenih značajki glasova 49
4.4.1. Sustav za stvaranje značajki glasovnog signala 49
4.4.2. Sustav za stvaranje srednjih vrijednosti značajki 52
4.5. Sustav za prepoznavanje glasova hrvatskog jezika 55
4.5.1. Algoritam sustava za prepoznavanje glasova 55
5. EKSPERIMENTALNI REZULTATI 59
5.1. Test 1 59
5.2. Test 2 61
5.2.1. Rezultati prepoznavanja pojedinih glasova 61
5.2.2. Izmjerene udaljenosti pojedinih glasova 62
5.3. Test 3 64
5.4. Test 4 65
5.5. Zaključak eksperimenta 67
6. ZAKLJUČAK 69
LITERATURA 71
SAŽETAK 76

ABSTRACT 76
ŽIVOTOPIS 77
PRILOZI 78

1. UVOD

Govor je osnovno sredstvo komunikacije među ljudima. Kao takav se nameće kao potencijalno sredstvo komuniciranja između ljudi i strojeva, što se tehnološkim napretkom sve više ostvaruje. Prepoznavanje govora je kompleksno znanstveno područje koje uključuje mnogo znanstvenih disciplina, kao što su lingvistika, akustika, elektrotehnika te računarstvo. Povezujući znanja o govoru i njegovoj obradi iz ovih disciplina ostvareni su sustavi za automatsko prepoznavanje govora. U strogo tehničkom smislu, sustav za automatsko prepoznavanje govora je sustav koji kao ulaz prima govor u određenom obliku, a izlaz je interpretacija ulaznog govora. To može značiti da se ulazni govor interpretira kao tekst, naredba ili nekakva druga vrsta interakcije koja je moguća između čovjeka i stroja. Postoji komercijalna potreba za sustavima koji mogu prepoznati govor u stvarnom vremenu te tako omogućiti bolju interakciju čovjeka sa strojem, što i danas pred inženjere stavlja određene izazove.

Zadatak ovog diplomskog rada je s Altium razvojnim sustavom temeljenim na FPGA razviti i testirati mogućnost prepoznavanja govora u stvarnom vremenu. Korišten razvojni sustav je Altium NanoBoard 3000 temeljen na Xilinx Spartan 3AN FPGA integriranom sklopu koji se razvija na programskoj platformi Altium Designer. Kako bi bilo moguće ispitati zadanu mogućnost potrebno je dobro poznavanje razvojnog sustava kao i problematike sustava za automatsko prepoznavanje govora.

Drugo poglavlje naslovljeno „Prepoznavanje govora“ sadrži kratak pregled povijesti razvoja sustava za automatsko prepoznavanje govora. Navedena su važnija istraživanja koja su dovela do novih metoda za pristup problematici, kao i uspješne komercijalne primjene. Opisana je struktura osnovnog sustava za prepoznavanje govora, kao i algoritmi i postupci koji se u navedenim sustavima koriste.

Poglavlje tri opisuje razvojni sustav koji je korišten za razvoj sustava opisanog u implementaciji i eksperimentu, Altium NanoBoard 3000. U poglavlju su opisani resursi razvojne ploče čija jezgra čini FPGA integrirani sklop, kao i program koji se koristi za programiranje ploče. Detaljno su obrađeni resursi koji se koriste u implementaciji i eksperimentu.

U implementaciji su objašnjeni matematički postupci i algoritmi koji se koriste kako bi se implementirao jednostavan sustav za prepoznavanje glasova. Obrađeni su postupci koji se koriste za dobivanje usrednjenih značajki govora koje se koriste za usporedbu u sustavu za prepoznavanje glasova. Isti postupci se koriste u sustavu za prepoznavanje glasova koji je opisan u poglavlju „Eksperimentalni rezultati“.

U poglavlju „Eksperimentalni rezultati“ je opisan sustav implementiran na razvojnoj ploči koji služi za prepoznavanje glasova koji se pojavljuju u hrvatskom jeziku. Sustav na temelju ulaznog govornog signala koji sadrži jedan glas odlučuje o tome koji glas je izrečen na temelju usrednjenih značajki govora. Usrednjene značajke govora su dobivene postupkom opisanom u poglavlju „Implementacija“. Provedeno je mjerenje te su obrazloženi rezultati mjerenja i preciznost razvijenog sustava.

2. PREPOZNAVANJE GOVORA

Prepoznavanje govora te razvijanje sustava za automatsko prepoznavanje je multidisciplinaran problem kao što je to navedeno u uvodu. U ovom poglavlju je dan kratak pregled povijesti i razvoj sustava za prepoznavanje govora radi stjecanja dojma i upoznavanja s problematikom s kojom su se istraživači u ovom području morali nositi. Pošto je u implementaciji razvijen sustav za prepoznavanje glasova, u nastavku je opisan princip rada tipičnog sustava za prepoznavanje govora na čijemu principu radi većina modernih sustava za prepoznavanje govora. Nadalje su podrobnije opisani algoritmi koji se koriste u sustavima za prepoznavanje govora. Priroda govornih signala omogućuje stvaranje značajki signala određenim algoritmima koji su opisani u nastavku. Algoritmi koji se koriste za analizu značajki te konačan odabir prepoznatih glasovnih jedinica čine ključan dio sustava za prepoznavanje govora, stoga su im posvećena posebna potpoglavlja.

2.1. Povijest i razvoj

Razvoj sustava za prepoznavanja govora seže nedaleko u povijest. Intenzivniji razvoj započinje tek u drugoj polovici dvadesetog stoljeća što je uzrokovano razvojem tehnologije. U ovom kratkom pregledu povijesti razvoja sustava za prepoznavanje govora spomenuti su važniji događaji i primjeri sustava za prepoznavanje govora. Spomenuti su određeni primjeri znanstvenih istraživanja kao i komercijalnih primjena. Detaljan pregled povijesti istraživanja razvoja sustava za automatsko prepoznavanje govora je dostupan u [1].

Prvim uspješnim sustavom za prepoznavanje govora se smatra „Audrey“ razvijen 1952. godine u Bell Laboratories. „Audrey“ je bio potpuno analogan sustav koji je mogao prepoznati izgovor znamenaka sa stankom između svake pojedine znamenke. Preciznost sustava je bila i do 99% ako je sustav bio prilagođen korisniku. Međutim, „Audrey“ nije imao komercijalnu primjenu zbog svoje neekonomičnosti. [2]

Deset godina nakon „Audrey“ sustava, IBM predstavlja „Shoebox“ sustav. „Shoebox“ sustav je prepoznavao 16 izgovorenih riječi engleskog jezika te znamenke od 0 do 9, a osim toga mogao je obavljati određene matematičke operacije. Prepoznavanje govora se odvijalo analognim audio filterima, a prepoznata riječ je bila prikazana pojedinim svjetlećim signalima. [2]

Kasnih 60-ih godina prošlog stoljeća Atal i Ikatura neovisno oblikuju osnovne koncepte linearno-prediktivnog kodiranja (engl. linear predictive coding, skraćeno LPC) koji se svodi na pojednostavljenu procjenu vokalnog trakta iz valnog oblika govornog signala [1]. Do sredine

70-ih, Ikatura, Rabiner, Levinson te drugi predlažu primjenu temeljne tehnike prepoznavanja obrazaca bazirane na LPC. Detaljan opis LPC analize se nalazi u poglavlju implementacija.

Tijekom 70-ih godina prošlog stoljeća dolazi do većih istraživanja na području prepoznavanja govora prvenstveno zbog ulaganja U.S. Department of Defense (ministarstvo obrane Sjedinjenih američkih država) [1]. DARPA Speech Understanding Research program je bio jedan od najvećih programa istraživanja prepoznavanja govora u povijesti koji je na kraju iznjedrio sustav „Harpy". „Harpy" je imao vokabular od 1011 riječi, a značajan je jer je uveo takozvani „beam search" – efikasniji algoritam pretraživanja grafova. Daljnji pokušaji implementacije sustava za prepoznavanje govora će se uglavnom svoditi na napredak u tehnikama pretraživanja.

Sedamdesetih godina u IBM-u se počinje istraživati primjena skrivenih Markovljevih modela (engl. hidden Markov models, skraćeno HMM) u svrhu prepoznavanja govora [3]. Sustavi za prepoznavanje govora temeljeni na skrivenim Markovljevim modelima u upotrebi su i danas.

80-ih godina se pojavljuju i komercijalna rješenja za prepoznavanje govora [2]. Kurzweil-ov program za prebacivanje govora u tekst 1985. je imao vokabular od 1000 riječi, dok je IBM-ov program za istu namjenu imao vokabular od 5000 riječi. Korisnik je prilikom korištenja ovakve programske podrške morao riječi izgovarati odvojeno, s kratkom pauzom između riječi.

Krajem 80-ih se počinje istraživati novi koncept u svrhu prepoznavanja govora – umjetne neuronske mreže (engl. artificial neural networks, skraćeno ANN) [1]. Tada su pokazale dobre rezultate za prepoznavanje jednostavnih jezičnih jedinica, kao što su izolirani fonemi i izolirane riječi. Međutim, prepoznavanje govora iziskuje rukovanje s vremenskim varijacijama za što neuronske mreže nisu pogodne. Stoga se većina istraživanja fokusira na primjenu Bayesove teorije odlučivanja na ovo područje, a ponajviše skrivenih Markovljevih modela.

Uspješnost statističkih metoda prepoznavanja govora je ponovo probudila interes DARPA-e za na ovom području, krajem osamdesetih i početkom devedesetih [2]. To je dovelo do razvoja novih sustava za prepoznavanje govora, kao što je CMU-ov sustav „Sphinx" koji je integrirao statističke metode skrivenih Markovljevih modela s algoritmom pretraživanja grafova prethodnog sustava „Harpy". „Sphinx" sustav je bilo moguće istrenirati za prepoznavanje specifičnog konteksta riječi sa složenom gramatikom i širokim vokabularom.

Prema [4], 1990. godine tvrtka „Dragon" je pustila u prodaju program za prepoznavanje govora namijenjen osobnim računalima s nazivom „Dragon Dictate", a sedam godina kasnije „Dragon

NaturallySpeaking“ koji je mogao prepoznati 100 riječi u minuti. Program je zahtijevao treniranje prije korištenja koje je trajalo 45 minuta.

1996. se pojavio prvi glasovni portal „VAL“ koji je razvila tvrtka „BellSouth“ [4]. „VAL“ je bio interaktivan sustav na pozivanje koji je davao informacije na osnovu toga što je korisnik rekao na telefonu.

2000-tih godina se pojavljuje širok raspon programskih podrški za prepoznavanje govora. Programske podrške prepoznavanja govora za osobna računala u prosjeku su imale postotak prepoznavanja oko 80%. Tadašnji sustavi su dobre rezultate prepoznavanja pokazivali samo za relativno ograničene vokabulare. Windows Vista i Mac OS X su dolazili sa sustavom za prepoznavanje koji je omogućavao upravljanje glasovnim komandama. Međutim, većina korisnika nije koristila ove značajke, što zbog nepouzdanosti, što zbog složenosti korištenja.

Google-ova „Voice Search“ aplikacija za iPhone je ponovo aktualizirala prepoznavanje govora za komercijalnu upotrebu. Mobilni uređaji su idealni za primjenu sustava za prepoznavanje govora. Male veličine mobilnih uređaja onemogućuju složenije ulazne jedinice, stoga se govorne naredbe nameću kao idealno rješenje. Google dodaje podršku za prepoznavanje govora prilagođenu korisniku u „Voice Search“ aplikaciji za Android sustav 2010. godine, što omogućava stvaranje statistike korisničkih naredbi kako bi buduće prepoznavanje naredbi bilo preciznije. „Google Chrome“ internetski pretraživač dobiva podršku „Voice Search“ 2011. godine. [4]

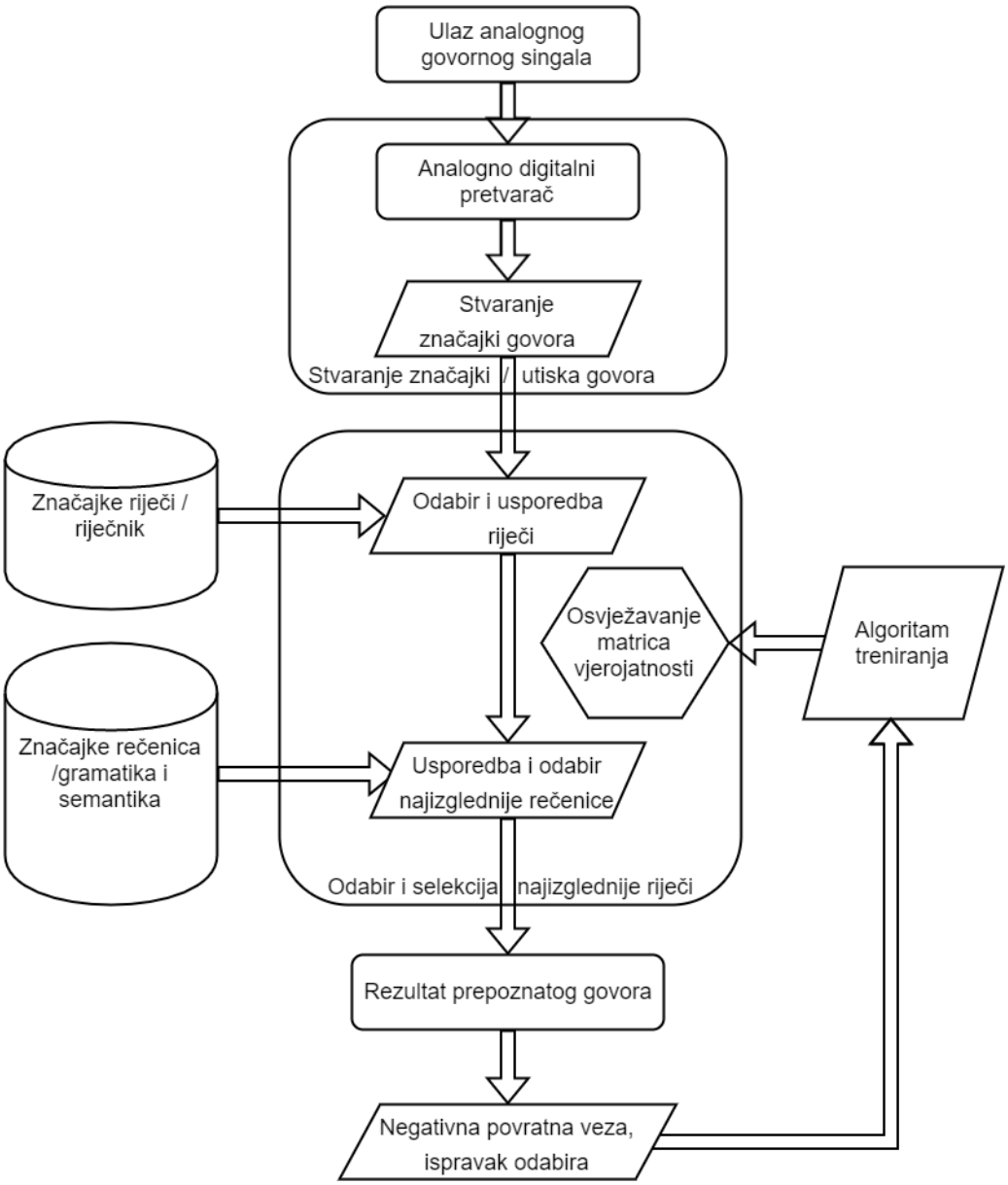
Apple-ovi mobilni uređaji danas dolaze s aplikacijom „Siri“. Siri i Google-ov „Voice Search“ su temeljeni na „Cloud“ uslugama. Siri aplikacija također prati korisničku povijest te na temelju korisničkih podataka generira adekvatni odgovor, dodajući k tome novu dimenziju interaktivnosti. Budući sustavi za prepoznavanje govora će težiti većoj interaktivnosti, poput Siri aplikacije. [4]

Danas na tržištu aplikacija za mobilne uređaje postoji mnoštvo aplikacija koji se koriste prepoznavanje govora za interakciju s korisnikom. Sve više se u automobile ugrađuju sustavi za prepoznavanje govora koji omogućuju kontrolu određenih sustava u automobilu koristeći glasovne komande. Tržište aplikacija za osobna računala svakako nije izostavljeno, kao što je spomenuto ranije, a noviji trend čine „Cloud“ rješenja. Trendovi programske podrške se kreću u smjeru prirodnije interakcije s čovjekom. Iz navedenog razvoja sustava za prepoznavanje govora se može zaključiti kako su se naprednije tehnike počele koristiti kako se povećavala snaga dostupnih računalnih sustava.

Većina današnjih sustava za prepoznavanje govora se koriste naprednijim statističkim tehnikama i unaprijeđenim skrivenim Markovljevim modelima kao osnovnim algoritmom za prepoznavanje izgovorenih struktura. Ovi sustavi mogu ostvariti veliku preciznost uz veliki raspon primjena, ali još uvijek ne mogu u potpunosti nadmašiti čovjeka, prvenstveno zbog toga što su sustavi dosegli krajnja ograničenja svojih modela. Međutim, istražuju se novi pristupi, kao što su dubinske neuronske mreže ili dubinsko strojno učenje (engl. deep neural networks, deep machine learning). Ovi modeli trebaju pružiti bolji pristup u sustavima za prepoznavanje govora, ali pri tome zahtijevaju veće računalne resurse. Sustavi temeljeni na dubinskom strojnom učenju se danas koriste u „Cloud“ rješenjima.

2.2. Algoritmi i postupci sustava za prepoznavanje govora

Gotovo svaki sustav za automatsko prepoznavanje govora ima istu osnovnu strukturu, prikazanu slikom 2.1 [5].



Slika 2.1: Blok dijagram tipičnog sustava za automatsko prepoznavanje govora.

Na početku sustava za automatsko prepoznavanje govora se nalazi sustav za dohvaćanje zvučnog signala. Sustav za prepoznavanje govora u realnom vremenu najčešće na samom početku sadrži mikrofon koji je spojen na sustav za obradu zvuka, odnosno na sklopovlje čija je jezgra analogno-digitalni pretvornik. Mogući su i sustavi s direktnim digitalnim ulazom. Zvuk, pa tako i govor je analogan signal. Zvuk se dohvaća i pretvara u digitalni signal te se nadalje u potpunosti obrađuje digitalno u modernim sustavima [5]. U povijesti su postojali pokušaji analogne obrade zvuka u

svrhu prepoznavanja govora što je obrađeno u prethodnom potpoglavlju, ali su napušteni u korist fleksibilnije digitalne obrade. [2]

Nakon što je stvoren digitalizirani govorni signal, sustav iz signala stvara govorne značajke signala, kao što je prikazano slikom 2.1. Signali govora su vremenski sporo promjenjivi, kvazi-stacionarni signali, kao što je navedeno u [6]. Ako govorni signal promatramo na relativno kratkom vremenskom odsječku, od 5 do 100 milisekundi, signal se doima stacionarnim. Promjena značajki signala između vremenskih odsječaka će značiti da je došlo do promjene izgovorenog glasa. Značajke govornog signala su određen skup vrijednosti koji su dobiveni određenim metodama i algoritmima nad određenim vremenskim isječkom digitaliziranog govornog signala. Značajke govornog signala predstavljaju statističke vrijednosti koje sadrže informaciju o trenutnom stanju signala. Omogućuju stvaranje i sažimanje informacije ključne za raspoznavanje osnovnih glasovnih jedinica – fonema. Osim fonema, mogu poslužiti i za raspoznavanje većih jezičnih jedinica, ovisno o primijenjenom algoritmu. U sljedećem potpoglavlju su opisani algoritmi za stvaranje značajki signala.

Nakon što su iz ulaznog govornog signala stvorene značajke dobiven je niz značajki ili utisak govornog signala. Na temelju utiska govornog signala potrebno je odabrati koja riječ je rečena iz baze značajki riječi – rječnika [5]. Značajkama snimljenog govornog signala se mjere udaljenosti od referentnih značajki koje se nalaze u bazi. Ovakav pristup se naziva prepoznavanje uzoraka iz predloška [7]. Udaljenosti značajki govora se mjere specifičnim metodama za mjerenje udaljenosti namijenjene određenoj vrsti značajki. Na temelju udaljenosti stvorenih značajki signala se dodjeljuju statistički podatci izgovorenom signalu. Riječima u rječniku se dodjeljuju statističke vrijednosti vjerojatnosti da su prepoznate. Za tu zadaću su zaduženi statistički algoritmi odlučivanja kao što su: dinamičko savijanje vremena (engl. dynamic zime warping), skriveni Markovljevi modeli, umjetne neuronske mreže te drugi koji će biti opisani u jednom od narednih potpoglavlja [8]. Ako se sustav bazira na prepoznavanju izoliranih riječi, prepoznata riječ će se odabrati na temelju najveće vrijednosti vjerojatnosti kojoj joj je dodijelio model.

Ako pretpostavimo da je sustav za prepoznavanje govora dizajniran tako da prepoznaje rečenice, iz govornog signala potrebno je, osim pojedinih riječi, prepoznati rečenicu. Sustav je, prema slici 2.1, prethodno dodijelio određene statističke podatke prethodno obrađenim nizovima riječi. Na redu je postupak koji ovisno o gramatičkim pravilima, semantičkim pravilima i karakteristikama rečenica iz baze podataka o jezičnoj strukturi odabire i mijenja prethodno prepoznate riječi. Za

opisani postupak se također koriste iste ili slične statističke metode i algoritmi odlučivanja kao što su korišteni prilikom odlučivanja o prepoznatoj riječi.

Sustav na kraju postupka na izlazu prikazuje prepoznatu riječ ili rečenicu, kao što je prikazano slikom 2.1. Većina modernih sustava sadrži negativnu povratnu vezu s bazama podataka u kojima su zapisani statistički podatci o značajkama glasova, riječi i rečenica. Podatci se osvježavaju trening algoritmima u svrhu preciznijeg prepoznavanja u budućnosti.

2.3. Algoritmi za stvaranje značajki govornog signala

Govor je sporo vremensko promjenjivi signal [6], odnosno kvazi-stacionaran. Na kratkim vremenskim odsječcima, trajanja 5 do 100 milisekundi, značajke govornog signala su poprilično stacionarne te se razmatraju kao takve. Promjena značajki signala znači da je došlo do promjene izgovorenog glasa. Informacija u govornom signalu je zapravo predstavljena u kratkoročnim promjenama amplitude spektra valnog oblika signala. Značajke signala se upravo izvlače iz amplituda spektara kratkotrajnih odsječaka (okvira) signala. Značajke signala omogućuju sažimanje informacije govornog signala što omogućuje kompresiju signala ili olakšavaju usporedbu značajki signala s drugim signalima, što je korisno za primjene prepoznavanja govora.

Ključna poteškoća prepoznavanja govora je velika varijabilnost govornog signala uzrokovana: različitim govornicima, različitim brzinama izgovora, izgovorenom sadržaju i kontekstu, emocijama govornika i akustičkim uvjetima [6]. Stvaranje značajki govora se provodi prvenstveno zbog smanjenja varijabilnosti. Značajke isječaka govornog signala obično su poprilično slične za različite uvjete glasovnog signala. Određena riječ izgovorena kao šapat, bezvučno, može imati slične značajke kao glas izgovoren na glas, zvučno. Postupak stvaranja značajki govora ima vrlo važnu ulogu u sustavu za prepoznavanje govora. Prema [6], postoje brojne metode za stvaranje značajki govora:

- linearno-prediktivna analiza (engl. linear predictive analysis, linear predictive coding, skraćeno LPC), odnosno linearno-prediktivno kodiranje,
- linearno-prediktivni cepstralni koeficijenti (engl. linear predictive cepstral coefficients, skraćeno LPCC),
- perceptualni linearno-prediktivni koeficijenti (engl. perceptual linear predictive coefficients, skraćeno PLPC),
- mel-frekvencijski cepstralni koeficijenti (engl. mel freqency cepstral coefficients, skraćeno MFCC),

- cepstralna analiza mel skale (engl. mel scale cepstrum analysis, skraćeno MEL),
- relativno filtriranje spektra domenskih logaritamskih koeficijenata (engl. relative spectra filtering of log domain coefficients, skraćeno RASTA ili RASTA-PLP),
- derivacija prvog reda (engl. first order derivative, poznatije kao DELTA),
- spektralna analiza snage (engl. power spectrum analysis, češće korišten naziv je fast Fourier tranform, skraćeno FFT).

Razlog stvaranja značajki govora iz kratkih vremenskih odsječaka je zbog sličnosti obrade zvuka u uhu čovjeka. Pužnica u uhu čovjeka provodi kvazi-frekvencijsku analizu [6]. Analiza u pužnici se odvija prema nelinearnoj frekvencijskoj skali, koja se naziva Bel skala ili mel skala. Skala počinje linearno do frekvencije od 1000 Hz, a nakon toga je logaritamska. Stoga je prilikom izvlačenja karakteristika uobičajeno provoditi sažimanje frekvencijske osi nakon spektralnog proračuna.

2.3.1. Linearno-prediktivna analiza – LPC

Linearno-prediktivna analiza (linearno prediktivno kodiranje, engl. linear predictive coding, skraćeno LPC) je jedna od najboljih metoda za analiziranje govora. Koristi se za kvalitetno kodiranje i sažimanje govornog signala pri niskom omjeru bitova [6]. Osnovna ideja iza linearno prediktivnog kodiranja je da specifični trenutni uzorak govornog signala može biti prikazan kao linearna kombinacija prošlih uzoraka govornog signala. Linearno-prediktivna analiza se temelji na ljudskoj glasovnoj tvorbi. Koristi uobičajene modele izvora i filtra u kojemu su grkljan, vokalni trakt i zračenje usana integrirani u jedan svepolni filtar koji simulira akustiku vokalnog trakta. Glavni princip linearno-prediktivne analize je minimizirati sumu kvadriranih razlika između originalnog signala i procijenjenog signala tijekom konačnog trajanja signala, to jest na jednom vremenskom odsječku signala (okviru). Princip se koristi za dobivanje jedinstvenog skupa predikcijskih koeficijenata. Predikcijski koeficijenti se obično procjenjuju za svaki vremenski okvir, koji je obično traje između 10 i 20 milisekundi. Drugi važan parametar je dobitak (engl. gain, oznaka G). Transfer funkcija vremenski promjenjivog digitalnog filtra je dana formulom:

$$H(z) = \frac{G}{\sum_{k=1}^{p} a_k \cdot z^{-k}}, \tag{2-1}$$

gdje k ide od 1 do p, gdje je p red LPC-a. Za proračun koeficijenata a_k se obično koristi Levinson-Durbinova rekurzija. LPC analiza svakog vremenskog okvira obično predviđa i procjenu da li je glas zvučan ili bezvučan. Za procjenu perioda titranja se koristi algoritam za procjenu perioda titranja odnosno frekvencije. Potrebno je naglasiti da će period titranja, dobitak i

koeficijenti varirati kako se mijenjaju vremenski okviri signala. U stvarnosti se koeficijenti uobičajeno ne koriste za prepoznavanje, pošto pokazuju vrlo visoku promjenjivost [6]. Uobičajeno se LPC koeficijenti prebacuju u kepstralne koeficijente koji su pogodniji za mjerenje udaljenosti značajki glasovnih signala. U [9] je opisan postupak kojim se LPC koeficijenti pretvaraju u LPC kepstralne koeficijente (skraćeno LPCC). Prednost LPCC koeficijenata nad LPC koeficijentima je mogućnost računanja srednje kepstralne razlike (engl. cepstral mean substraction, skraćeno CMS) čime se poništavaju učinci kanala.

Prema [6], vrste LPC metoda su:

- LPC glasovne pobude (engl. voice-excitation LPC),
- LPC preostale pobude (engl. residual excitation LPC),
- LPC pobude tona (engl. pitch excitation LPC),
- LPC višestruke pobude (engl. multiple excitation LPC, skraćeno MPLPC),
- LPC pobude regularnog pulsa (engl. regular pulse excited LPC, skraćeno RPLP),
- LPC kodirane pobude (engl. coded excited LPC, skraćeno CELP).

2.4. Algoritmi odabira jezičnih jedinica

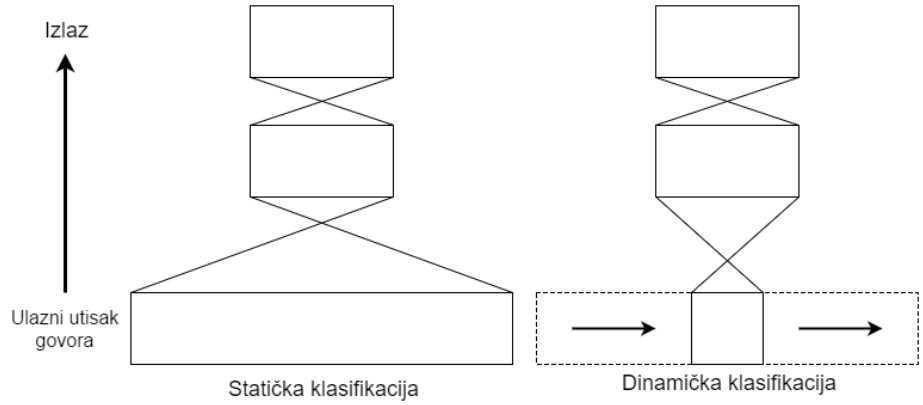
2.4.1. Dinamičko savijanje vremena

Algoritam dinamičkog savijanja vremena (engl. dynamic time warping) se provodi tako da se dvije sekvence značajki raspoređene u vremenu opetovano raširuju i skupljaju na vremenskoj osi sve dok nije postignuto podudaranje između sekvenci značajka [10]. Najčešće se koristi za izračun udaljenosti između dvije vremenske serije koje variraju u vremenu. Pogodnost algoritma dinamičkog savijanja vremena za prepoznavanje govora se nalazi u mogućnosti prepoznavanja govora pri različitim brzinama izgovora. U svrhu uspješnog uspoređivanja vremenskih sekvenci značajaka govora, vremenska komponenta se savija nelinearno. Algoritam dinamičkog savijanja vremena je u svojoj biti optimalan algoritam koji traži sličnosti između dva signala, odnosno slične obrasce. Kada se vrši savijanje vremenske osi, različite vremenske sekvence signala se rastežu i sakupljaju kako bi pogodili predložak dostupan u bazi podataka. Za slučaj da je govornik dio sekvence izgovorio brzo, taj dio sekvence se razvlači po vremenskoj osi. U slučaju da je govornik sekvencu izgovorio sporo, signali se sakupljaju kako bi pogodili predložak. Nedostatak ove metode je to što vrlo mali pomaci u točkama usporedbe signala mogu voditi to netočnog rezultata. Osim toga, nemoguće je sekvence pohraniti u obliku grafova kao kod skrivenih Markovljevih

modela te tako omogućiti složenije algoritme pretraživanja s manjim bazama za pohranu značajki govora.

2.4.2. Umjetne neuronske mreže

Problem prepoznavanja govora se svodi na problem prepoznavanja uzoraka, a pošto su neuronske mreže odličan alat za prepoznavanje uzoraka, brojni znanstvenici su ih primijenili u sustavima za prepoznavanja govora. Prema [11], neuronske mreže su se najprije koristile za vrlo jednostavne zadaće u sustavima za prepoznavanje govora, a to su: procjena zvučnih i bezvučnih vremenskih okvira govora, procjena da li su vremenski okviri govora nazalni, frikativni ili praskavi. Pošto su neuronske mreže uspješno rješavale navedene zadatke, nedugo zatim su uspješno primijenjene na problem klasifikacije fonema. Nakon klasifikacije fonema, primijenjene su i na prepoznavanje izoliranih riječi, ali u početku s ograničenim uspjehom. Prema [11], postoje dva osnovna načina primjene neuronskih mreža na klasifikaciju govora, a to su dinamički i statički što je prikazano slikom 2.2.

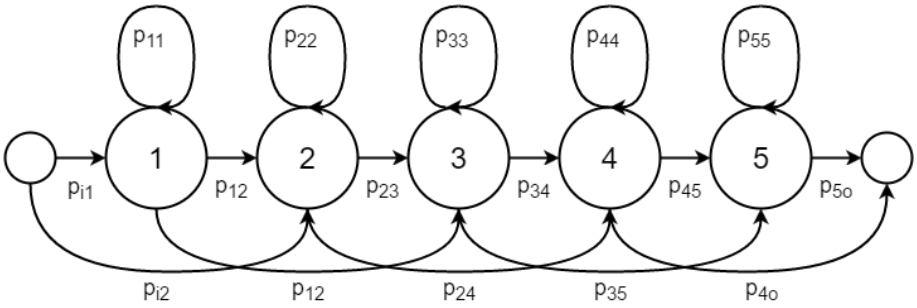


Slika 2.2: Statička i dinamička klasifikacija govornog signala kod neuronskih mreža.

U statičkoj klasifikaciji neuronskih mreža ulazni govorni signal se promatra kao jedna neodvojena cjelina te neuronska mreža donosi odluku u odnosu na karakteristike takve cjeline. Pri korištenju dinamičke klasifikacije neuronska mreža vidi samo mali okvir karakteristika govora, koji se s vremenom pomiče po cijeloj karakteristici signala govora te mreža donosi lokalne odluke na malim segmentima koje se kasnije trebaju integrirati u globalne odluke. Statička klasifikacija radi dobro na razini prepoznavanja fonema, ali na razini prepoznavanja riječi i fonema dinamička klasifikacija bolje obavlja zadanu zadaću. Obje klasifikacije mogu koristiti povratne veze, iako se povratne veze češće nalaze kod dinamičke klasifikacije.

2.4.3. Skriveni Markovljevi modeli

Skriveni Markovljevi modeli su temeljeni na dobro poznatim Markovljevim lancima iz teorije vjerojatnosti koji mogu modelirati nizove događaja ili stanja u vremenu [12]. Pružaju učinkovite algoritme za procjenu stanja i parametara, a provode automatsko dinamičko savijanje sekvenci koje su lokalno sabijene ili razvučene. Osim za modeliranje akustičkih sekvenci, koriste se i u druge svrhe. Iz topologije grafova Markovljevih lanaca može se uočiti Markovljevo svojstvo: sljedeće stanje u kojem će se nalaziti model ovisi samo o trenutnom stanju u kojem se model nalazi, bez obzira na način kojim se došlo u trenutno stanje te u kojim je prethodnim stanjima model bio. Na slici 2.3 je prikazan jedan Markovljev model koji može poslužiti za modeliranje riječi ili neke manje jezične jedinice poput fonema.

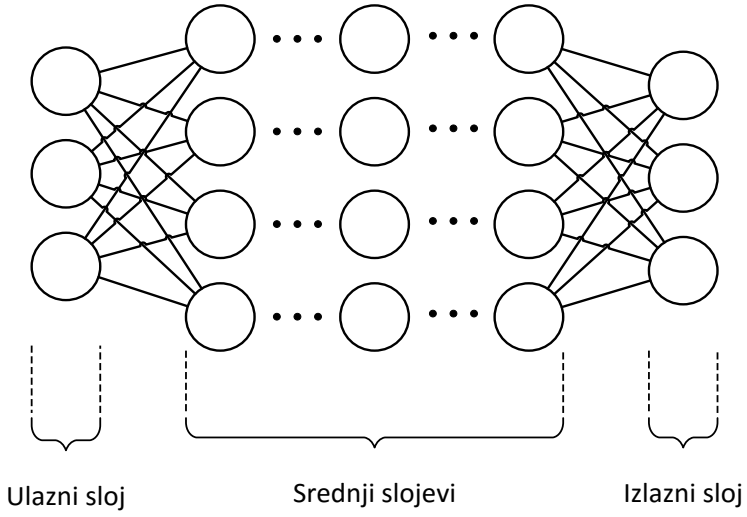


Slika 2.3: Markovljev model za modeliranje riječi ili fonema

Sustavi za prepoznavanje govora temeljeni na skrivenim Markovljevim modelima trebaju skup podataka za treniranje za svaku jedinicu govora za koju su namijenjeni prepoznati. Ako je sustav namijenjen prepoznavanju izoliranih riječi, potreban je skup podataka svake riječi u planiranom vokabularu sustava za prepoznavanje govora. Zbog zahtjeva treniranja za svaku jedinicu u vokabularu, primjena skrivenih Markovljevih modela može biti nepogodna za sustave sa zahtjevom prepoznavanja velikog vokabulara. Međutim, za sustave u kojima postoji ograničen broj jedinica u vokabularu, kao na primjer mali skup naredbi, skriveni Markovljevi modeli mogu polučiti odlične rezultate. Najčešće se koriste topologije modela od lijeva prema desno u kojima broj stanja ovisi o broju fonema koji se pojavljuju u određenoj riječi. Jedan fonem – jedno stanje je često pravilo kod ovakvih sustava. Ako se koriste jezične jedinice koje su manje od riječi, podatci mogu biti podijeljeni između različitih riječi. Za takav slučaj, nije potrebno trenirati sustav za svaku riječ koju sadrži u vokabularu, nego za svaku jedinicu s kojom su riječi vokabulara definirane. Vokabular ovakvog sustav je vrlo lako proširiv, ponekad bez potrebe za dodatnim treniranjem. Postoji određeni broj jezičnih jedinica manjih od riječi, a tipični modeli uključuju modele slogova, modele fonema ili modele akustički definiranih jedinica poznati kao fenoni [12].

2.4.4. Dubinske neuronske mreže

Noviji sustavi za prepoznavanje govora su temeljeni na modelima dubinskih neuronskih mreža (engl. deep neural networks), a često korišten termin je i dubinsko strojno učenje (engl. deep machine learning). Dubinske neuronske mreže su umjetne neuronske mreže koje sadrže više skrivenih slojeva između ulaznih i izlaznih slojeva, za razliku od plitkih umjetnih neuronskih mreža [13] [14]. Primjer dubinske neuronske mreže je prikazan slikom 2.4. Prema [15], dubinske neuronske mreže imaju više od tri sloja između izlaznog i ulaznog sloja. Služe modeliraju kompleksnih nelinearnih odnosa.



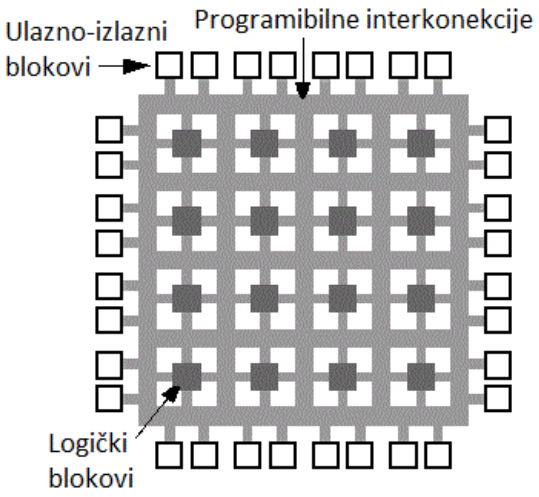
Slika 2.4: Prikaz dubinske neuronske mreže

Postoji veliki raspon varijanti dubinskih neuronskih mreža od kojih je većina potekla od neke prijašnje dobro poznate varijante. Raspon istraživanja u području dubinskih neuronskih mreža se neprestano povećava te se neprestano razvijaju novi modeli. Dubinske neuronske mreže su zamišljene kao mreže s kontrolom unaprijed (engl. feedforward), međutim za stvaranje jezičnih modela primjenjuju se povratne neuronske mreže (engl. recurrent neural netowrks), specifično model duge kratkotrajne memorije (engl. long short term memory, skraćeno LSTM) [16] [17]. U sustavima za prepoznavanja govora uspješno su primijenjene konvolucijske neuronske mreže, pokazujući bolje rezultate od ostalih modela [18].

Dubinske neuronske mreže su dobar model za sustave s velikim zahtjevima. Uspješno su primijenjene u sustavima gdje su postavljeni zahtjevi za velikim brojem različitih govornika, prepoznavanje više od jednog jezika te obuhvat kompletnog vokabulara jezika. Zahtijevaju velike računalne resurse za implementaciju, što ih čini idealnima za sustave temeljene na oblačnom (engl. cloud) računarstvu. [15]

3. FPGA RAZVOJNI SUSTAV

FPGA (engl. field programmable gate array, nizovi polja programibilnih vrata) su poluvodički uređaji koji su bazirani na matrici podesivih logičkih blokova (engl. configurable logic blocks, skraćeno CLB) spojenih preko programibilnih veza [19]. Slika 3.1 prikazuje strukturu FPGA integriranog kruga [20]. Prikazani su podesivi logički blokovi premreženi međusobnim programibilnim interkonekcijama koji su na kraju spojeni na ulazno izlazne blokove.



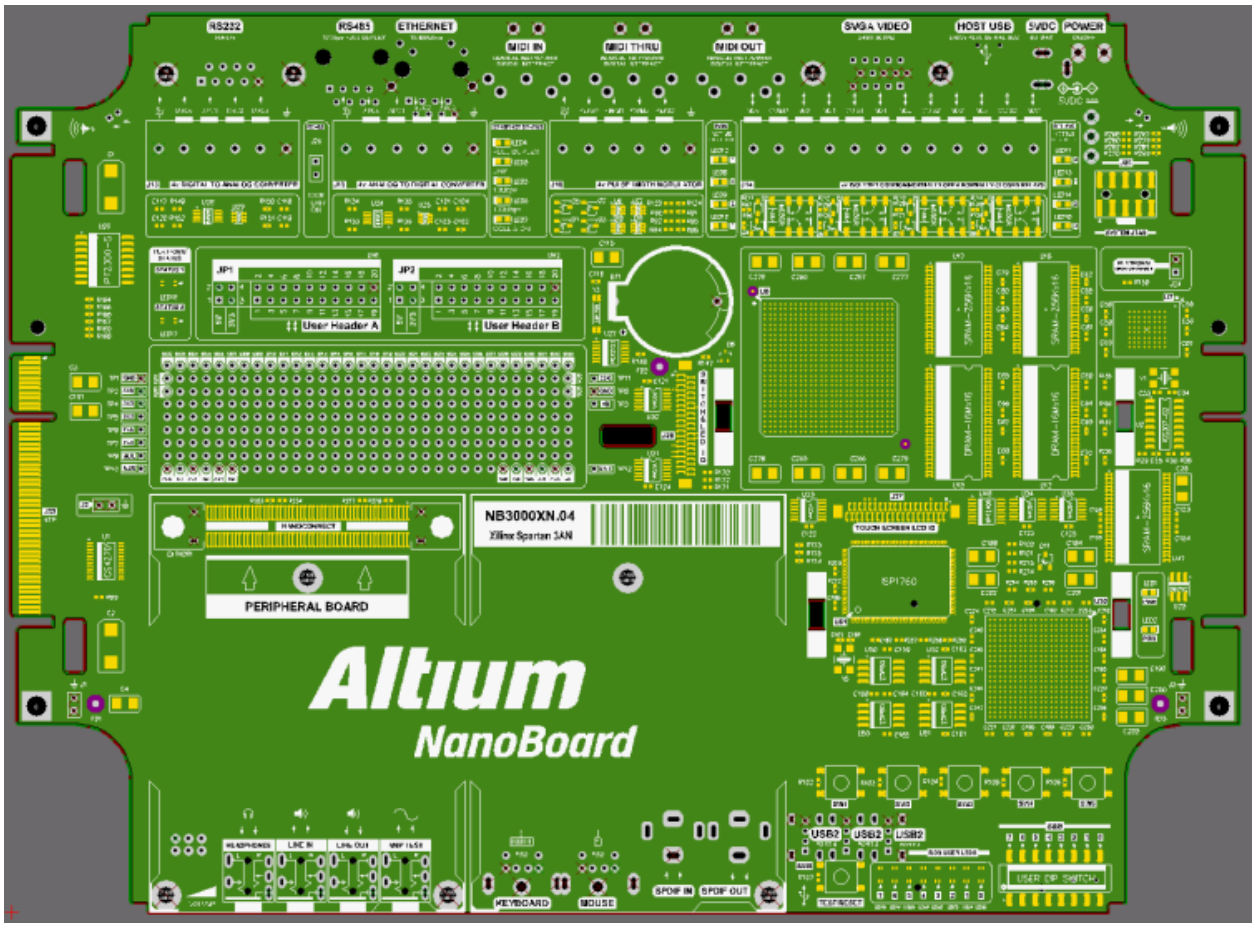
Slika 3.1: Struktura FPGA integriranog kruga

FPGA uređaji mogu biti reprogramirani da obavljaju željenu funkcionalnost nakon proizvodnje. To svojstvo razlikuje FPGA uređaje od ACIS-a (engl. application specific integrated circuits, prevedeno: aplikacijsko specifični integrirani krugovi) koji su posebno napravljeni kako bi obavljali predodređene zadatke. Iako postoje FPGA uređaji koji se mogu programirati samo jednom (skraćeno OTP, engl. one time programmable) dominantna vrsta su FPGA uređaji temeljeni na SRAM tehnologiji koji zadržavaju svojstvo reprogramibilnosti. Zbog navedenog svojstva FPGA uređaji se najčešće koriste za razvoje računalnih i logičkih sustava.

3.1. Altium NanoBoard

Sustav za prepoznavanje govora implementiran u poglavlju 5 je razvijan na razvojnoj ploči Altium NanoBoard NB3000XN.04 iz serije NB3000. Ploču razvija i proizvodi australsko poduzeće naziva Altium Limited. Prema [21], svaka od Altium-ovih ploča iz serije 3000 je 242·176 milimetara isprintana matična ploča sa šest slojeva, od toga četiri sloja za signal i dva ravna sloja, koja se napaja vanjskim napajanjem od 5V. Jedan od ravnih slojeva se koristi kao sloj za uzemljenje, dok se drugi uglavnom koristi za napajanje od 5V ili 3.3V. Na prednjem i stražnjem dijelu ploče su postavljene elektroničke komponente.

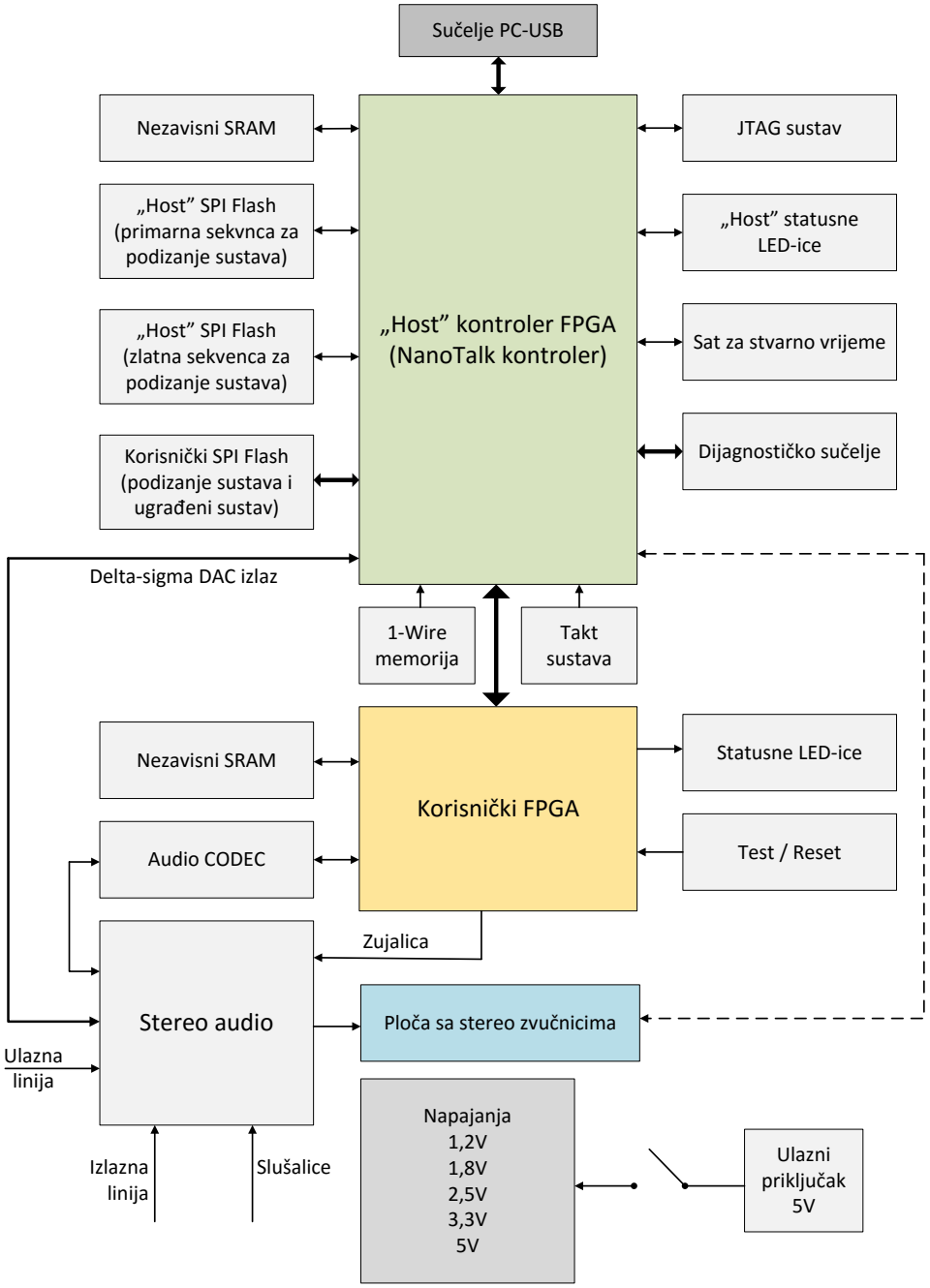
Svaka varijanta NanoBoarda 3000 serija, pa tako i varijanta NB3000XN.04, ima isti raspored komponenata na ploči. Ploče se razlikuju po fizičkim resursima koji se koriste, a to su: Host (NanoTalk) kontroler i korisnički FPGA integrirani sklop [21]. Slika 3.2 prikazuje fizički nacrt ploča NanoBoard serije 3000.



Slika 3.2: Fizički nacrt ploče NB3000XN.04

Prema [21], razvojni sustavi sadrže brojne ulazno-izlazne resurse, koji se dijele na resurse dostupne ugrađenom korisničkom FPGA te na resurse nedostupne korisničkom FPGA koji su dostupni samo matičnoj ploči. Dostupni su dodatni resursi koji dolaze odvojeno na perifernim pločama, a povezuju se preko namijenjenog utora na prednjoj strani ploče [21]. Detaljan opis resursa ploče dostupan je na [22] i [23].

Na web stranici u [24] se nalazi funkcionalni pregled serije NanoBoard- a 3000. Slika 3.3 prikazuje pojednostavljeni blok dijagram visoke razine NanoBoard-a 3000, čiju jezgru predstavlja Host Controller FPGA (NanoTalk Controller). Naznačeni su resursi matične ploče koji su korišteni u implementaciji, a posebno su istaknuti resursi koji su posvećeni Host mikroupravljaču te koji su dostupni korisničkom FPGA (User FPGA).



Slika 3.3: NanoBoard 3000 blok-dijagram

3.2. Resursi dostupni FPGA dizajnu

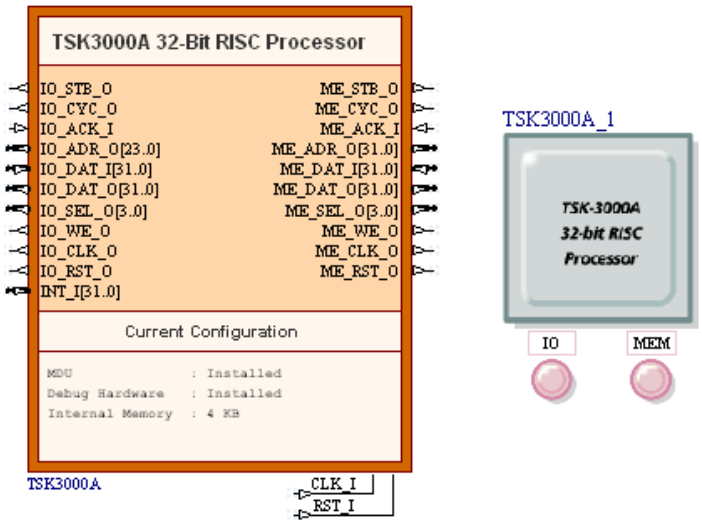
NanoBoard 3000 sadrži širok raspon resursa individualno spojenih na integrirani korisnički FPGA. Resursi omogućuju stvaranje različitih ugrađenih sustava na ploči (engl. embedded systems). Primjeri različitih ugrađenih sustava dolaze s programom Altium Designer, a na web stranici [25] se nalaze primjeri ugrađenih sustava namijenjenih NanoBoard-u koje su razvili različiti korisnici ploče. Na stranici [21] se nalaze hiperlinkovi na stranice koje sadrže sve resurse na matičnoj ploči kojima se može pristupiti te koji se mogu koristiti dizajnu koji je ciljan za korisnički FPGA uređaj (User FPGA). Popis svih resursa koji su dostupni korisničkom FPGA

- generatori takta sustava,
- serijska SPI Flash memorija,
- SRAM zajedničke sabirnice,
- SDRAM zajedničke sabirnice,
- Flash memorija zajedničke sabirnice,
- nezavisni SRAM,
- RS-232 serijsko sučelje,
- RS-485 serijsko sučelje,
- Ethernet priključak,
- PS/2 priključci za tipkovnicu i miš,
- korisnički USB priključak,
- USB hub,
- ADC sučelje,
- DAC sučelje,
- upravljači PWM jedinice napajanja,
- releji,
- audio CODEC,
- MIDI sučelje,
- SPDIF sučelje,
- Video izlaz,
- TFT LCD panel s zaslonom osjetljivim na dodir,
- korisničke DIP sklopke,
- korisničke RGB LED-ice,
- korisnički ulazno-izlazni pinovi,
- korisničko područje za prototipiranje,
- Test-Reset tipka,
- korisničke tipke opće namjene,
- čitač SD kartica (korisnički FPGA),
- IR prijemnik,
- zvučni signal.

Osim navedenih, postoje i resursi sustava, koji nisu dostupni korisniku i ne mogu se koristiti u FPGA dizajnu. Resurse sustava koristi razvojna ploča te programska podrška za razvojnu ploču. Na [21] prikazan je popis resursa sustava s linkovima koji podrobnije objašnjavaju svaki pojedini resurs. Popis resursa sustava:

- napajanje,
- Host FPGA(NanoTalk Controller),
- korisnički FPGA(User FPGA),
- konfiguracijska Flash memorija Host kontrolera,
- SRAM Host kontrolera,
- status LED-ice Host-a,
- korisničko FPGA napajanje i program LED-ice,
- konektor perifernih ploča,
- sučelje NanoBoard-računalo (USB port),
- audio sustav,
- SPI sat za stvarno vrijeme,
- čitač SD kartica (Host FPGA),
- ID memorija ploče,
- port za programiranje JTAG sustava,
- dijagnostičko sučelje.

3.2.1. Mikroprocesorski sustav TSK3000A



Slika 3.4: TSK3000A procesor

TSK3000A je 32 bitni, Wishbone kompatibilan, RISC procesor (engl. reduced instruction set computer), interno baziran na Harvard arhitekturi s jednostavnim pristupom vanjskoj memoriji [26]. Na lijevoj strani slike 3.4 je prikazan shematski prikaz, dok na se na desnoj strani nalazi Wishbone model [26]. Većina instrukcija je 32-bitne širine te se izvode u jednom ciklusu takta. Osim što podržava brzi pristup registrima, TSK3000A podržava stvaranje korisničko definirane količine blok RAM-a s trenutnim pristupom i dvostrukim portom. Procesor ima 32 prekida, od kojih je svaki individualno prilagodljiv na naponsku razinu, na rastući brid ili pak padajući brid.

3.2.2. Analogni audio sustav

NanoBoard 3000 sadrži analogni audio pod-sustav koji čine analogni mikser, pojačalo snage, 3.5 milimetarski ulazi i izlazi za reprodukciju zvučnog signala te zvučnici. Jezgru analognog audio sustava čini audio mikser koji služi za miješanje određenih audio izvora zajedno, pri tome se podrazumijeva da postoji poseban stupanj za svaki odvojeni kanal: lijevi i desni. Sljedeći audio izvori sudjeluju u miješanju:

- LineOut iz Host-a,
- LineOut iz periferne ploče,
- LineOut iz Audio CODEC-a,
- zvučni signal, zujalica (engl. buzzer),
- Audio Test signal.

Za pojačanje audio signala na ploči se koristi stereo audio pojačalo naziva PT2300 proizvođača Princeton Technology. Ulaz u pojačalo čini stereo audio signal koji je doveden s analognog miješala. Stereo pojačalo u izlaznom stupnju uređaja je ostvareno spajanjem dva para operacijskih pojačala u most konfiguraciju, što rezultira stereo diferencijalnim pojačanjem.

NanoBoard 3000 je opremljen 3.5 milimetarskim stereo audio priključkom za unos signala iz vanjske jedinice. Na ploči je označen s „LINE IN“. Prema [27], signali koji dolaze preko ovog ulaza, LIN_R i LIN_L, spojeni su preko perifernog konektora ploče na CS4270 Audio CODEC uređaj. „LINE IN“ audio ulaz nije spojen na analogni audio podsustav.

Negativni signali diferencijalnog stereo izlaza iz audio pojačala su spojena na dodatan 3.5 milimetarski audio konektor s nazivom „HEADPHONES“ koji služi priključivanju slušalica na ploču. Izlazni signali audio pojačala su dostupni na 3.5 milimetarskom stereo audio priključku, označeni s „LINE OUT“ na ploči. Ovo omogućava spajanje zvučnika s vlastitim napajanjem ili nekog drugog uređaja koji ima stereo audio ulaz.

Podešavanje snage izlaznog signala pojačala je ostvareno preko dvostrukog okretnog potenciometra (2×5Ω). Okretanjem potenciometra VR1 koji upravlja s istosmjernim naponom kontrole pojačanja u smjeru suprotnom od kazaljke na satu će eventualno u potpunosti prigušiti izlaz iz pojačala.

Diferencijalni stereo izlazi iz audio pojačala su spojeni na dvo-pinski konektor KK tipa na donjoj strani ploče. Izlazi su namijenjeni spajanju odvojene ploče sa zvučnicima za stereozvuk. Ploča sadrži dva zvučnika impedancije 4Ω, a spojena je na matičnu ploču preko dva konektora.

3.2.3. Audio CODEC

Kako bi bilo moguće povezati digitalne dijelove razvojne ploče s analognim audio sustavom potreban je audio CODEC (engl. coder-decoder). NanoBoard 3000 je opremljen CS4270 CODEC uređajem proizvođača Cirrus Logic. CS4270 je 24-bitni stereo audio CODEC s postavljenom frekvencijom uzorkovanja signala od 192 kHz. CODEC upravlja s analognim i digitalnim audio sustavom ploče.

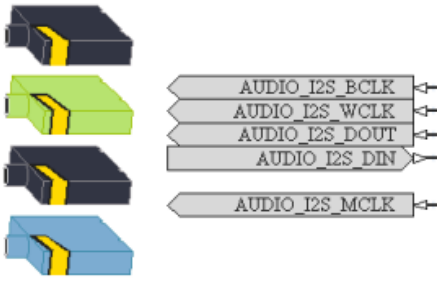
Analogni ulaz uređaja CS4270 čini par stereo-linijskih ulaza (LineIn_L, LineIn_R) koji su dobiveni s ulaza ploče naziva „LINE IN“. Analogni izlaz čine dva signala LineOut_L i LineOut_R. Oba izlaza se puštaju kroz jednostavne RC niskopropusne filtre iznosa granične frekvencije od otprilike 37 kHz. Izlazni signali se nakon toga odvode na analogno miješalo opisano u prethodnom potpoglavlju.

Svaki analogni izlazni signal se dohvaća s izlaza 24-bitnog stereo audio DAC-a (engl. digital to analog converter – digitalno-analogni pretvornik), koji ima mogućnost raditi na frekvencijama od 4 do 216 kHz. Frekvencija uzorkovanja se određuje prema postavkama *Speed* moda, programibilna preko unutarnjeg registra. Ista birana brzina se primjenjuje na DAC i ADC (engl. analog to digital converter – analogno-digitalan pretvornik). Pružena je digitalna kontrola jačine koja je pod kontrolom SPI dostupnih kontrolnih registara (engl. serial peripheral interface – serijsko periferno sučelje). Dobitak svakog kanala može varirati između 0 dB i -127 dB., s koracima od 0,5 dB. Kontrolni bitovi mogu biti postavljeni tako da u potpunosti priguše jedan od kanala neovisno o drugome.

Prijenos uzoraka digitalnog zvuka između CS4270 i procesora u FPGA dizajnu se odvija preko četvrerolinijske I2S sabirnice (engl. integrated interchip sound – ugrađen zvuk između integriranih krugova). Globalni signal takta i signal takta bita (označeni s AUDIO_I2S_BCLK i AUDIO_I2S_WCLK) se preuzimaju s FPGA dizajna, posebno preko posrednog I2S kontrolera. Povezani su istim redoslijedom na ulaze CODEC-a SCLK i LRCK. Dodatni signal takta s FPGA dizajna (AUDIO_I2S_MCLK) se dovodi na MCLK ulaz CODEC-a kako bi se omogućila upravljačka frekvencija za delta-sigma modulator te digitalne filtre. Iznos frekvencije signala AUDIO_I2S_MCLK je postavljen na 256 puta iznosa frekvencije signala AUDIO_I2S_WCLK.

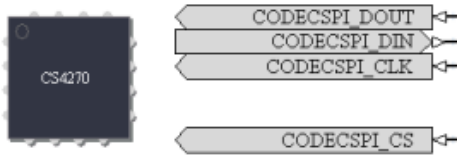
Za ostvarivanje komunikacije sa CS4270 preko I2S sabirnice te razmjene digitalnih audio podataka potrebno je postaviti AUDIO_CODEC komponentu iz „FPGA NB3000 Port-Plugin.IntLib“ biblioteke prikazanu slikom 3.5. Komponenta sadrži opisane ulazne signale

takta, AUDIO_I2S_BCLK, AUDIO_I2S_WCLK i AUDIO_I2S_MCLK, kao i digitalni audio ulaz i izlaz, AUDIO_I2S_DIN i AUDIO_I2S_DOUT.



Slika 3.5: AUDIO_CODEC komponenta iz „FPGA NB3000 Port-Plugin.IntLib“ biblioteke

Komponenta koja omogućuje komunikaciju s uređajem preko SPI sabirnice procesoru u FPGA dizajnu, a samim time i kontrolu uređaja CS4270, također se nalazi u „FPGA NB3000 Port-Plugin.IntLib“ biblioteci. Komponenta se naziva AUDIO_CODEC_CTRL te je prikazana slikom 3.6. Komponenta sadrži tri ulazna signala CODECSPI_DOUT, CODECSPI_CLK i CODECSPI_CS, te jedan izlazni signal CODECSPI_DIN.



Slika 3.6: AUDIO_CODEC_CTRL komponenta iz „FPGA NB3000 Port-Plugin.IntLib“ biblioteke

3.2.4. Nezavisni SRAM

Na NanoBoardu 3000 se nalazi nezavisni statički RAM (engl. random access memory, prijevod: memorija sa slučajnim pristupom) koji je korišten u implementaciji [P1] i [P2]. Nezavisni statički RAM je jedan od memorijskih resursa koji je dostupan korisničkom FPGA uređaju. Naziv „nezavisni“ se koristi za razlikovanje od SRAM-a zajedničke sabirnice. Nezavisni SRAM dolazi u obliku dva 4 megabitna, CMOS SRAM uređaja visoke brzine. Svaki uređaj je organiziran kao polje od 256 000 puta 16 bitova. Uređaji koriste 3.3 V napajanje matične ploče. Uređajima se pristupa odvojeno, što stvara dva odvojena memorijska područja kapaciteta od 512 kilobajta, zajedno čineći 1 megabajt.

Komunikacija s uređajima nezavisnog SRAM-a se u sučelju dizajna omogućuje dodavanjem komponenata u biblioteci „FPGA NB3000 Port-Plugin.IntLib“ naziva SRAM0 i SRAM1. Komponente su prikazane slikom 3.7.



Slika 3.7: SRAM0 i SRAM1 komponente iz „FPGA NB3000 Port-Plugin.IntLib“

3.3. Altium Designer

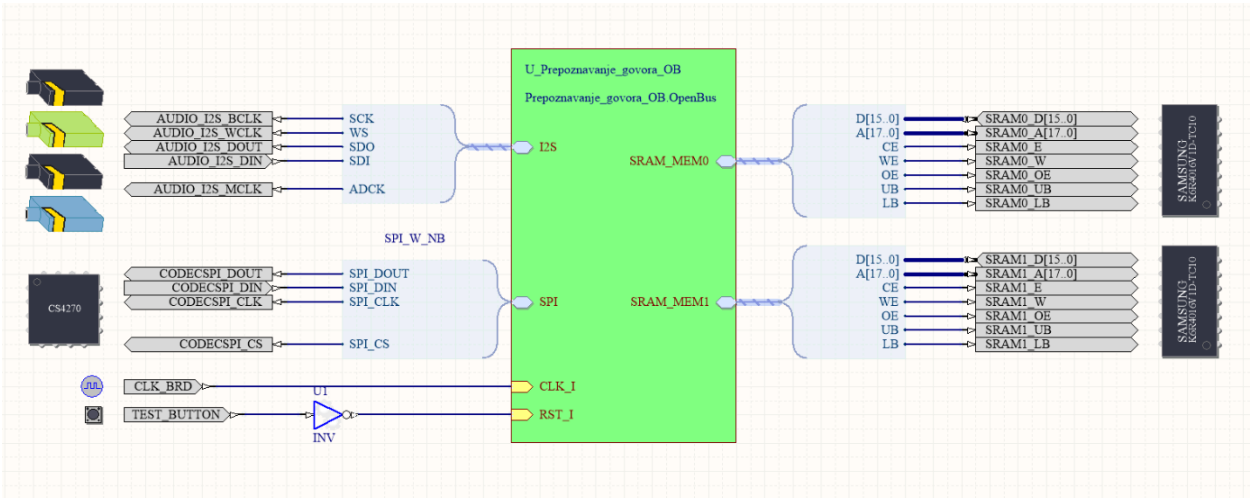
Altium Designer je program namijenjen automatskom elektroničkom dizajnu tiskanih električnih ploča, FPGA dizajnu i dizajnu ugrađenih računalnih sustava (engl. embedded systems), održavanju podržanih biblioteka te menadžmentu automatske distribucije razvijenog dizajna. Održava ga i proizvodi australska kompanija Altium Limited, kompanija koja proizvodi Altium NanoBoard korišten za razvijanje sustava [P1] i [P2]. Posljednja stabilna verzija Altium Designer-a je 16.1.9 puštena 3. lipnja 2016. godine. Altium Designer je namijenjen razvijanju projekata za Altium NanoBoard ploče. U poglavljima 4 i 5 za implementaciju je korištena verzija 16.0.5.

U nastavku je opisana struktura FPGA projekta i ugrađenog projekta sustava u prilozima [P1] i [P2] koji su namijenjeni pokretanju na razvojnoj ploči. FPGA projekt ima istu strukturu u prilozima [P1] i [P2], te je navedeni FPGA projekt opisan u narednom potpoglavlju. Ugrađeni projekti se razlikuju za [P1] i [P2] po kodu, no osnovna struktura ugrađenog projekta je ista. Funkcionalnost koda [P1] i [P2] koja se nalazi u ugrađenom projektu je opisana u četvrtom i petom poglavlju. Objašnjeno je programiranje NanoBoard 3000 ploče u programu Altium Designer.

3.3.1. FPGA projekt

FPGA projekt je dokument stvoren Altium Designerom koji obuhvaća cjelokupni dizajn koji je namijenjen za emulaciju na određenoj razvojnoj ploči. Na temelju FPGA projekta se prilikom prevođenja dizajna stvaraju ostali dokumenti unutar toka procesa dizajna koji u konačnici služe postavljanu dizajna na razvojnu ploču. Sustavi koji se nalaze u prilozima [P1] i [P2] su realizirani kao FPGA projekti u kojima su sadržani dodatni dokumenti.

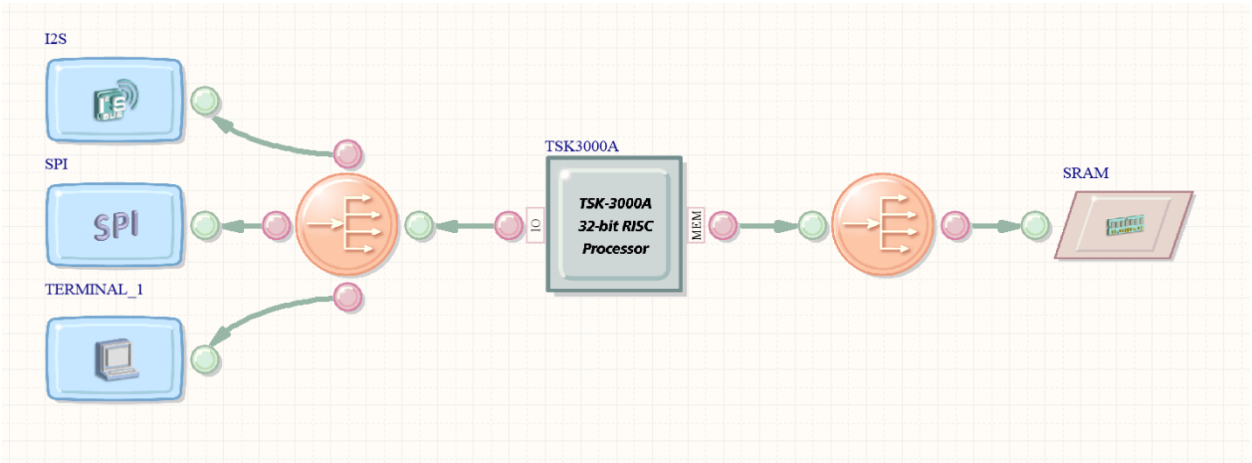
Dokument koji sadrži shemu dizajna koji se u konačnici emulira na ploči je shematski dokument (engl. schematic). Shematski dokument je sadržan unutar FPGA projekta. Na slici se nalazi 3.8 shema shematskih dokumenata projekata [P1] i [P2].



Slika 3.8: Shematic dokument FPGA projekta

Shematski dokument sadrži komponente koji su specifični za ploču za koju je ciljan FPGA projekt. Resursi specifični za korištenu ploču se nalaze u biblioteci „FPGA NB3000 Port-Plugin.IntLib“. Iz navedene biblioteke u projektima [P1] i [P2] su korištene komponente: „AUDIO_CODEC“, „AUDIO_CODEC_CONTROL“, „CLOCK_BOARD“, „TEST_BUTTON“, „SRAM0“ i „SRAM1“. Većina navedenih komponenti je spojena na sabirničke ulaze shematskog simbola uvezenog iz OpenBus dokumenta preko namijenjenih „Harness“ (engl. to hraness – upregnuti, obuzdati) konektora. Korišteni Harness konektori su: „I2S_W_Both“, „SPI_W_NB“ i „WB_MEM_CTRL_SRAM16_A18“. Komponenta „CLOCK_BOARD“ predstavlja signal takta ploče koji je potreban za ispravno funkcioniranje komponenata unutar OpenBus komponente. Između spoja komponente „TEST_BUTTON“ i ulaznog signala „RST_I“ na OpenBus komponenti postavljena je „INV“ komponenta čija je funkcionalnost logička negacija odnosno invertiranje. Komponenta „INV“ se nalazi unutar biblioteke „FPGA Generic.IntLib“. „TEST_BUTTON“ omogućava resetiranje procesora unutar OpenBus komponente.

Na slici 3.8 uvezena „OpenBus“ (prijevod: otvorena sabirnica) komponenta je prikazana pravokutnikom zelene boje. OpenBus dokument sadrži intuitivnu shemu FPGA dizajna koja se kreira dodavanjem jednostavnih komponenti koji se spajaju preko *Wishbone* komponenti i *master-slave* sustava sabirnica. Slika 3.9 prikazuje shemu OpenBus dokumenta FPGA projekta koja se nalazi u [P1] i [P2].

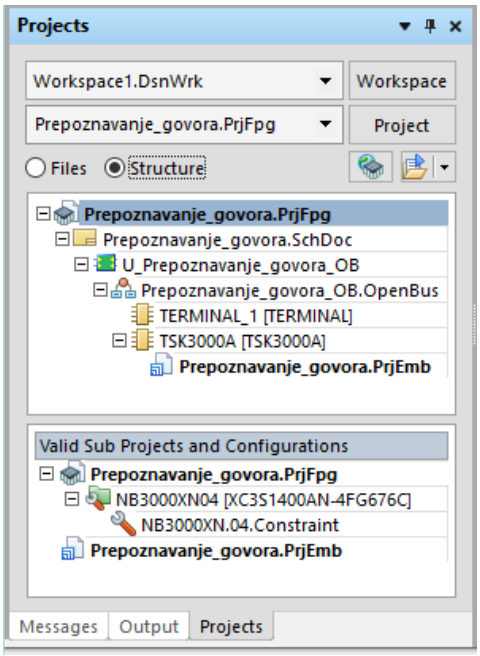


Slika 3.9: OpenBus shema FPGA projekta

Procesor TSK3000A je konfiguriran tako da sadrži 32 kilobajta vlastite memorije. SRAM je podešen kao asinkroni RAM te sadržava jedan MB memorije koja koristi način pristupa „2 puta 16 bitna širina“. I2S je podešen za primanje i slanje, a koristi HW spremnik s 4 kilobajtnom memorijom. Osim toga I2S komponenti je omogućeno korištenje prekida procesora „INT_1“. I2S služi sa slanje i primanje audio uzoraka s audio CODEC-a. SPI uređaju je onemogućen mod „Pin“, a koristi 32 bitnu veličinu prijenosa podataka. SPI služi podešavanju audio CODEC-a. Uređaj „TERMINAL_1“ omogućuje komunikaciju preko prozora terminala na računalu preko JTAG sučelja. Ulazno-izlazni uređaji su spojeni na namijenjenu sabirnicu procesora preko Wishbone komponente. SRAM je spojen na memorijsku sabirnicu procesora, također preko Wishbone komponente.

Struktura FPGA projekta [P1] i [P2] je prikazana na slici 3.10. U strukturi FPGA projekta se na najvišoj razini nalazi shematski dokument (ekstenzija *.SchDoc*). Unutar shematskog dokumenta se nalazi OpenBus dokument unutar kojega su prikazani uređaji „TERMINAL_1“ i „TSK3000A“. Na „TSK3000A“ je povezan ugrađeni projekt (ekstenzija *.PrjEmb*).

Na slici 3.10 je prikazana konfiguracija pripadnog FPGA projekta naziva „NB3000XN04“. Konfiguracije koriste dokumente ekstenzije *.Constraint* (engl. constraint, prijevod: ograničenje) koje sadrže specifičnosti i ograničenja određenih FPGA integriranih krugova, razvojnih ploča i resursa na razvojnim pločama. Navedena konfiguracija sadrži „NB3000XN.04.Contsraint“ dokument koji sadrži specifičnosti i ograničenja korištene razvojne ploče.



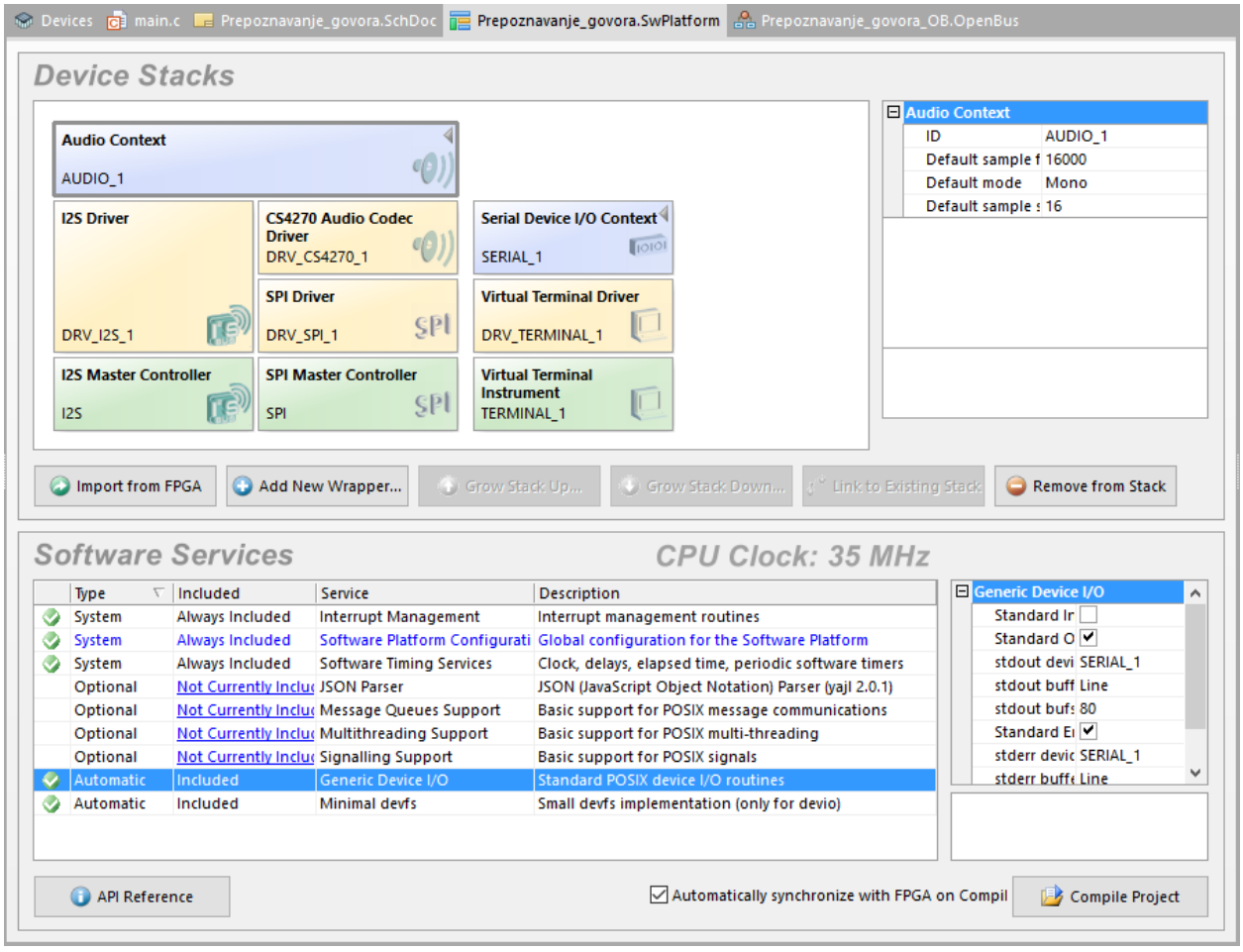
Slika 3.10: Prozor „Projects“ u kojem je prikazana struktura FPGA projekta

3.3.2. Ugrađeni projekt

Prema [28], ugrađeni projekt (engl. embedded project) je skup dokumenata dizajna koji su potrebni kako bi se proizvela programska podrška koja se može ugraditi zajedno s izvršnim procesorom u elektroničkom proizvodu. Srž ugrađenog projekta čini programski kod, izražen u asembleru ili C-u. Sav izvorni kod sadržan u ugrađenom projektu se prilikom kompajliranja prevodi u asemblerski jezik. Asembler na posljetku prevodi asemblerski kod u strojni jezik koji se naziva još i objektnim kodom. Objektne datoteke se povezuju zajedno te se mapiraju na određen memorijski prostor, čime nastaje jedna datoteka koja je spremna za ciljani sustav.

Osnova ugrađenog projekta je dokument SwPlatform (skraćeno od Software Platform, prijevod: platforma programske podrške). Software Platform je okvir programske podrške koja olakšava pisanje programa za pristup vanjskim jedinicama određenog hardverskog dizajna [29]. Osim toga, Software Platform olakšava pisanje protokola te pruža dodatne mogućnosti koje je moguće koristiti u programskoj podršci, kao što je višenitnost. Sofware Platform je u biti skup programskih modula isporučenih u obliku programskog koda. Programski moduli se dodaju u ugrađeni projekt radi upravljanja s raznim rutinama niskih razina koje su potrebne za kontrolu ili pristup vanjskim jedinicama. Svaki modul pruža pristup aplikaciji pružajući određenu funkcionalnost (na primjer, funkcija `audio_record()` iz *audio.c* omogućava snimanje uzoraka zvuka audio ulaza u niz). Altium Designer sadrži Software Platform Builder – grafičko korisničko sučelje koje služi konfiguriranju i dodavanju programskih modula u ugrađeni projekt.

Ugrađeni projekt sustava [P1] i [P2] je povezan na procesor TSK3000A u FPGA projektu, kao što je prikazano slikom 3.10. Navedeni ugrađen projekt sadrži SwPlatform dokument koji je prikazan slikom 3.11 u prozoru Software Platform Buildera u Altium Designeru. U projektima [P1] i [P2] je korištena verzija *Summer '10 Software Platform*. U prozoru Software Platform Buildera se nalazi dugme „Import from FPGA“ koje služi uvozu najnižih programskih modula prepoznatih vanjskih jedinica u FPGA projektu. Najniži programski moduli su prikazani pravokutnicima zelene boje. Nad njima su postavljeni programski moduli srednje razine prikazani pravokutnicima žute boje. Na vrhu su postavljeni programski moduli najviše razine prikazani pravokutnicima plave boje. U projektima [P1] i [P2] postoje dva stoga programskih modula koji na vrhovima imaju module „Audio Context“ i „Serial Device I/O Context“. „Audio Context“ modul ispod sebe sadrži dva stoga modula koji služe upravljanju I2S i SPI uređaja. Na slici 3.11 je prikazan izgled konačne programske platforme u prozoru Altium Designer-a.



Slika 3.11: Prozor Software Platform Buildera u kojemu je prikazana struktura programske platforme

„Audio Context“ programski modul u konačnici omogućuje puštanje i snimanje zvuka u glavnom C programu u projektima [P1] i [P2]. Frekvencija uzorkovanja je podešena na 16 kHz, a širina

uzorka zvuka iznosi 16 bita. Koristi se isključivo mono kanal. Postavke su dostatne za potrebe analize govornih signala.

„Serial Device I/O Context“ programski modul omogućuje ispis teksta u prozor terminala u programu Altium Designer-a. Terminal za komunikaciju s računalom koristi JTAG sučelje koje je na računalo fizički spojeno USB kablom. Ispis u prozor terminala se u C programskom kodu postiže korištenjem funckije `printf()` iz biblioteke *stdio.h*.

Ugrađeni projekt sadrži datoteke s C kodom. Kako bi u C kodu bilo moguće koristiti sve dodane programske module iz Software Platform datoteke u programski kod se uključuje zaglavlje *swplatform.h*. Umjesto uključivanja *swplatform.h* moguće je uključiti pojedina zaglavlja koja su potrebna korisniku u kodu. U projektima [P1] i [P2] koriste se funkcije snimanja i reproduciranja uzoraka zvuka te ispis u terminal, stoga su u *main.c* uključena zaglavlja *audio.h* i *stdio.h* umjesto *swplatform.h*.

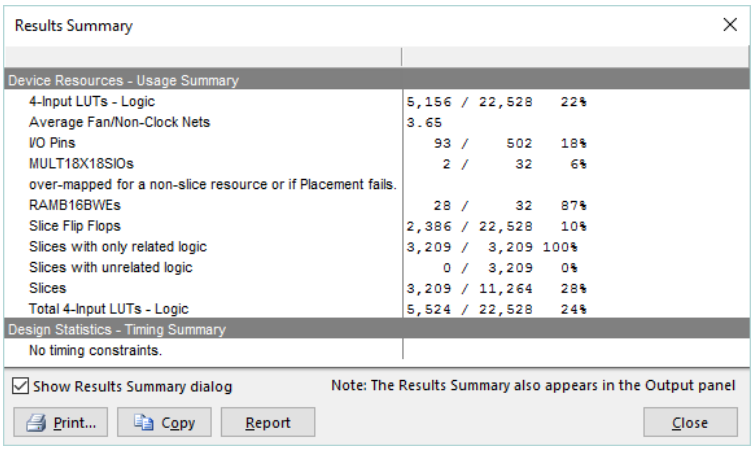
Ugrađeni projekt [P1] i [P2] dodatno je modificiran. U opcijama ugrađenog projekta izmijenjene su lokacijske opcije (engl. Locate Options). U lokacijskim opcijama ugrađenog projekta se nalazi prikaz lokacija programskog koda u memoriji. Odabrana je opcija „Force read-only items to be located in RAM“ (prijevod: prisili postavljanje stvari namijenjene samo čitanju u RAM) koja omogućuje smještaj programskog koda u memoriju SRAM-a. Razlog odabira opcije je veličina koda programa koja premašuje veličinu memorije procesora. Podešene su i opcije kompajlera (engl. Compiler Options). Pod stavkom povezivač (engl. Linker) podešene su veličine „Stack“ i „Heap“. Veličina „Stack“ je podešena na 5 kilobajta, dok je veličina „Heap“ podešena na 900 kilobajta.

3.3.3. Programiranje razvojne ploče

Altium NanoBoard ploča spaja se na računalo USB kablom. Komunikacija između ploče i računala, odnosno Altium Designera se odvija JTAG protokolom. Ploča se programira u kartici „Devices“ glavnog prozora Altium Designera prilikom „Live“ moda. U prozoru je omogućen izbor FPGA projekta te pripadne konfiguracije projekta.

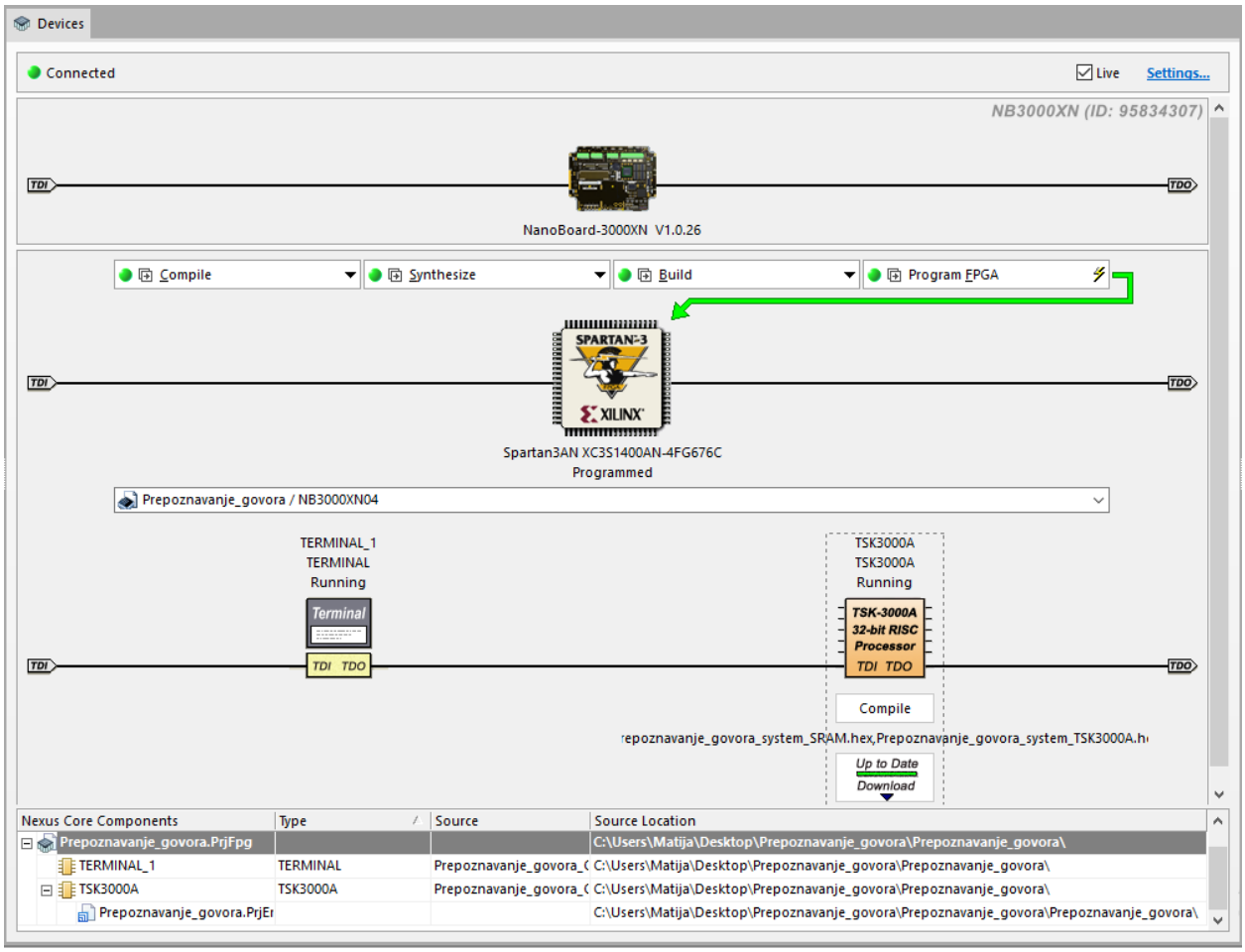
Korištena ploča temeljena je na Xilinx Spartan 3AN FPGA integriranom krugu, stoga Altium Designer mora prepoznati odgovarajući Xilinx-ov „Place and Route“ programski alat (engl. place and route – postavi i usmjeri). U projektu je korišten Xilinx ISE Pack 14.7. Razlog zašto je potrebno koristiti posebnu programsku podršku je to što je FPGA integrirani krug zaštićen patentnim pravom te je samo proizvođaču poznat dizajn FPGA integriranog kruga.

Programiranje ploče određenim projektom se odvija u četiri faze koji čine dio toka procesa dizajna (engl. proces flow): kompajliranje (engl. Compile), povezivanje (engl. Synthesize), izgradnja (engl. Build), te konačno programiranje FPGA (engl. Program FPGA). Faze se izvršavaju navedenim redoslijedom. Ako postoje izmjene datoteka usred struje procesa, potrebno je proći kroz sve procese koje se nalaze u nastavku struje procesa. Nakon uspješno izvršene prve tri faze pojavljuje se prozor sažetka rezultata prikazan slikom 3.12.



Slika 3.12: Prozor sažetka rezultata za projekte u prilogu

Faza programiranja služi za postavljanje dizajna na fizički FPGA uređaj na razvojnu ploču. Dostupna je isključivo u modu „Live“ nakon što su uspješno završene prethodne tri faze te nakon što je stvorena FPGA programska datoteka. Na slici 3.13 je prikazan glavni prozor Altium Designera s otvorenom „Devices“ karticom u kojemu je vidljiv isprogramiran projekt dostupan u prilogu. Prikazane su navedene faze struje procesa, izbornik s konfiguracijama, fizički uređaji (ploča i FPGA), virtualni instrumenti (kontrole procesora i terminal).



Slika 3.13: Prozor *Devices* u modu uživo s uspješno isprogramiranom pločom

4. IMPLEMENTACIJA

U ovom poglavlju je opisan sustav za prepoznavanje glasova hrvatskog jezika razvijen na Altium NanoBoard-u 3000 ploči pomoću programa Altium Designer. Sustav za prepoznavanje glasova je implementiran kao ugrađeni projekt u Altium Designeru shematskog projekta opisanom u poglavlju tri. Projekt se nalazi u prilogu [P1].

Pošto su za projekt [P1] potrebne usrednjene značajke glasova, razvijen je sustav za stvaranje usrednjenih značajki pojedinih glasova. Sustav za stvaranje usrednjenih značajki pojedinih glasova se sastoji od dva dijela priložena u [P2] i [P3]. Prilog [P2] se sastoji od ugrađenog projekta u Altium Designeru namijenjenog za shematski projekt opisan u poglavlju tri, za isti shematski projekta kao i [P1]. Pokretanjem [P2] u terminal projekta se ispisuju značajke snimljenog glasa s audio ulaza, koje se pohranjuju u tekstualnu datoteku s ekstenzijom *.log*. Tekstualna datoteka s pohranjenim značajkama se koristi kao ulazna za projekt u prilogu [P3]. Projekt [P3] je komandno-linijski program za Microsoft Windows platformu napravljen u Visual Studio Express 2015 okruženju u C++ programskom jeziku. Program [P3] iz ulazne tekstualne datoteke iščitava značajke i stvara usrednjene značajke koje se potom koriste u [P1].

U nastavku ovog poglavlja je dan opis algoritama i postupaka koje se koriste u prilozima. Opisani su algoritmi za stvaranje značajki govornog signala kao i algoritmi za mjerenje udaljenosti značajki govornog signala.

4.1. Glasovi hrvatskog jezika

U hrvatskom jeziku postoji 32 glasa koji imaju ulogu fonema. Osim 30 fonema koja su predstavljeni slovima hrvatske abecede, postoje fonemi /ie/ i /ŗ/. Fonem /ie/ predstavlja ijekavski refleks slavenskog glasa jat, u hrvatskom jeziku se zapisuje kao „ije“. Fonološka transkripcija /ŗ/ označava slogotvorno „r“ (primjeri riječi: „vrt“, „krv“, „trn“) koji za razliku od suglasnika „r“ (primjeri riječi: „riječ“, „razlika“, „krava“) ima veću zvonkost, duže trajanje i veći broj treptanja. Osim fonema hrvatski jezik sadrži alofone – ostvarenja fonema u različitim uvjetima. Uvježbani slušatelj bi trebao s lakoćom razlikovati alofone (/ŋ́/, /ʒ́/, /ɣ́/, /ś/, /ź/, /F/) u odnosu na njihove izvorne foneme. Hrvatski jezik prepoznaje i nefonemski glas naziva šva koji se u fonološkoj transkripciji zapisuje znakom /ə/, a čuje se prilikom izgovora samoglasnika samostalno. [30]

U [P1] je osmišljen sustav za prepoznavanje glasova hrvatskog jezika. Predviđeno je prepoznavanje 30 glasova, koji su predstavljeni fonemima. Fonem /ie/ je izbačen jer je dvoglas, a slogotvorno „r“ je izjednačeno sa suglasnikom „r“. Prepoznavanje alofona nije implementirano.

Sustav prepoznaje 30 glasova hrvatskog jezika koji se mogu prikazati slovima hrvatske abecede. Pošto se ispis u [P1] obavlja po ASCII standardu, određeni hrvatski znakovi su zamijenjeni određenom kombinacijom ASCII znakova. Tablica 4.1 prikazuje foneme hrvatskog jezika, koja slova ih predstavljaju, službenu fonološku transkripciju i način na koji se ispisuju u [P1].

	Fonološka transkripcija	Slovo	Ispis		Fonološka transkripcija	Slovo	Ispis
1.	/a/	A, a	A	**17.**	/ʎ/	Lj, lj	LJ
2.	/b/	B, b	B	**18.**	/m/	M, m	M
3.	/ʦ/	C, c	C	**19.**	/n/	N, n	N
4.	/ʧ/	Č, č	CX	**20.**	/ɲ/	Nj, nj	NJ
5.	/ʨ/	Ć, ć	CY	**21.**	/ɔ/	O, o	O
6.	/d/	D, d	D	**22.**	/p/	P, p	P
7.	/ʤ/	Dž, dž	DX	**23.**	/r/	R, r	R
8.	/ʥ/	Đ, đ	DY	**24.**	/s/	S, s	S
9.	/ɛ/	E, e	E	**25.**	/ʃ/	Š, š	SX
10.	/f/	F, f	F	**26.**	/t/	T, t	T
11.	/g/	G, g	G	**27.**	/u/	U, u	U
12.	/h/	H, h	H	**28.**	/ʊ/	V, v	V
13.	/i/	I, i	I	**29.**	/z/	Z, z	Z
14.	/j/	J, j	J	**30.**	/ʒ/	Ž, ž	ZX
15.	/k/	K, k	K	**31.**	/ie/	ije	-
16.	/l/	L, l	L	**32.**	/ɽ/	R, r	R

Tablica 4.1 Prikaz fonema u hrvatskom jeziku

Glasovi se dijele prema akustičkim i tvorbenim svojstvima. Za LPC analizu najvažnije akustičko svojstvo glasa je zvučnost, pošto je dokazano da LPC analiza pokazuje bolje rezultate za zvučne glasove. Prilikom izgovora zvučnih glasova koriste se glasnice, dok prilikom izgovora bezvučnih glasova zračna struja slobodno prolazi kroz grkljan. Svi samoglasnici su zvučni, uključujući fonem /ie/ i samoglasničko „r“ (/ɽ/). Prema načinu tvorbe suglasnici su podijeljeni na sonante i šumnike. Svi sonantni su zvučni, a čine ih: „j, „l“, „lj“, „m“, „n“, „nj“, „r“ i „v“. U tablici 4.2 je dan odgovarajući popis zvučnih i bezvučnih parova šumnika. [31]

Zvučni	/b/	/d/	/g/	/z/	/ʒ/ (ž)	/ʤ/ (dž)	/ʥ/ (đ)	-	-	-
Bezvučni	/p/	/t/	/k/	/s/	/ʃ/ (š)	/ʧ/ (č)	/ʨ/ (ć)	/f/	/ʦ/ (c)	/h/

Tablica 4.2 Parovi zvučnih i bezvučnih šumnika

4.2. Generiranje osnovnih značajki govora

Govor sadrži određene značajke koje se mogu vrlo jednostavnim metodama generirati iz okvira govornog signala trajanja 10 do 20 milisekundi. Takve značajke se koriste za jednostavne primjene u sustavima za automatsko prepoznavanje govora. U implementaciji su, osim linearno-prediktivnih koeficijenata, korištene još tri jednostavne značajke govornog signala:

- Kratkotrajna energija signalan,
- Kratkotrajni broj prelaska kroz nulu,
- Period impulsa.

U nastavku će podrobnije biti opisane navedene značajke, kako se stvaraju te kako pomažu pri prepoznavanju govora.

4.2.1. Kratkotrajna energija signala

Kratkotrajna energija signala (engl. short time energy, skraćeno STE) je značajka govora koja se računa na jednom okviru govornog signala. Ako se STE računa na cijelom području govornog signala, signal se obično podijeli na kratkotrajne vremenske okvire trajanja od 10 do 20 milisekundi. Pošto je podrazumijevana digitalna obrada, signal je predstavljen uzorcima u diskretnom vremenskom vremenskim odsječcima, tako da se STE računa na određenom broju uzoraka signala. Podjela govornog signala na kratkotrajne vremenske okvire se uobičajeno postiže množenjem signala s odgovarajućim vremenskim prozorom kojemu su definirane vrijednosti za vremensko područje u kojemu se želi saznati značajka signala, dok za ostale vrijednosti iznos funkcije vremenskog prozora je nula. Za uokvirivanje govornog signala u svrhu računanja STE se uobičajeno koristi pravokutan prozor, definiran funkcijom:

$$w[n] = \begin{cases} 1, & n \in [0, N-1] \\ 0, & inače. \end{cases} \tag{4-1}$$

Kratkotrajna energija signala je zbroj kvadrata uzoraka signala u jednom vremenskom odsječku. Ako $s[n]$ predstavlja uzorke govornog signala, a $w[n]$ je funkcija pravokutnog prozora definirana jednadžbom (4-1), kratkotrajna energija signala vremenskog okvira m, E_m je definirana sljedećom jednadžbom:

$$E_m = \sum_n \big[s[n]w[m-n]\big]^2. \tag{4-2}$$

Ako definiramo kvadriranu funkciju vremenskog prozora kao:

$$h[n] = [w[n]]^2, \tag{4-3}$$

onda možemo jednadžbu (4-2) napisati u obliku:

$$E_m = \sum_n [s[n]]^2 h[m-n]. \tag{4-4}$$

U projektima [P1] i [P2] funkcija za računanje kratkotrajne energije signala je implementirana u datoteci *feature_extraction.c* s nazivom `float short_time_energy(int16_t* input_sample, int size)`. Povratna vrijednost funkcije je kratkotrajna energija signala, a ulazni parametri su niz uzoraka i veličina prozora.

4.2.2. Broj prelaska kroz nulu

Kratkotrajni broj prelaska kroz nulu (engl. short time zero crossing count, skraćeno ZCC) je također jedna od značajki govornog signala koja se računa na vremenskim okvirima govornog signala trajanja od 10 do 20 milisekundi. Ako pretpostavimo da se u jednom vremenskom okviru govornog signala nalazi N uzoraka govornog signala, broj prelazaka kroz nulu računamo prema formuli [32]:

$$ZCC_i = \sum_{k=1}^{N-1} 0{,}5 \cdot |sign(s[k]) - sign(s[k-1])| \,. \tag{4-5}$$

U prethodnoj jednadžbi funkcija $sign(x)$ je funkcija predznaka broja (engl. sign – predznak) koja definirana tako da vraća vrijednost -1 za ulaznu negativnu vrijednost, $+1$ za ulaznu pozitivnu vrijednost te 0 ako je ulazna vrijednost nula. Broj prelazaka kroz nulu broji koliko puta govorni signal u danom vremenskom okviru prođe kroz vremensku os. Broj prelazaka kroz nulu odražava frekvencijski sadržaj vremenskog okvira govornog signala. Visok broj prelazaka kroz nulu podrazumijeva visoku frekvenciju.

Prije računanja broja prelazaka kroz nulu iz signala je potrebno odstraniti istosmjernu komponentu signala. Istosmjerna komponenta govornog signala se može odstraniti na cijelom govornom signalu odjednom, a može se odstraniti na jednom vremenskom okviru signala. Ako pretpostavimo da je signal koji sadrži istosmjernu komponentu $s_{DC}[n]$ definiran na području $n \in [0, N-1]$ s N uzoraka, onda se signal $s[n]$ koji ne sadrži istosmjernu komponentu dobiva tako što se od svakog uzorka signala $s_{DC}[n]$ oduzme srednja vrijednost istog signala na području njegove definicije. Postupak je definiran jednadžbom (4-6).

$$s[n] = s_{DC}[n] - \sum_{k=0}^{N-1} \frac{s_{DC}[k]}{N} \tag{4-6}$$

Funkcija za odstranjivanje istosmjerne komponente signala je `void remove_dc(int16_t* input_sample, int size, int16_t* output_sample)`, a nalazi se u datoteci *feature_extraction.c* projekata [P1] i [P2]. Prvi parametar funkcije čini ulazni niz, drugi veličinu ulaznog i izlaznog niza, a treći izlazni niz koji nakon pozivanja funkcije predstavlja uzorke signala bez istosmjerne komponente.

U datoteci *feature_extraction.c* projekata [P1] i [P2] se nalazi funkcija `int zero_crossing_count(int16_t* input_sample, int size)` koja računa broj prelazaka kroz nulu kao povratnu vrijednost. Prvi ulazni parametar je niz uzoraka za koje se podrazumijeva da je prethodno uklonjena istosmjerna komponenta, a drugi je veličina niza uzoraka.

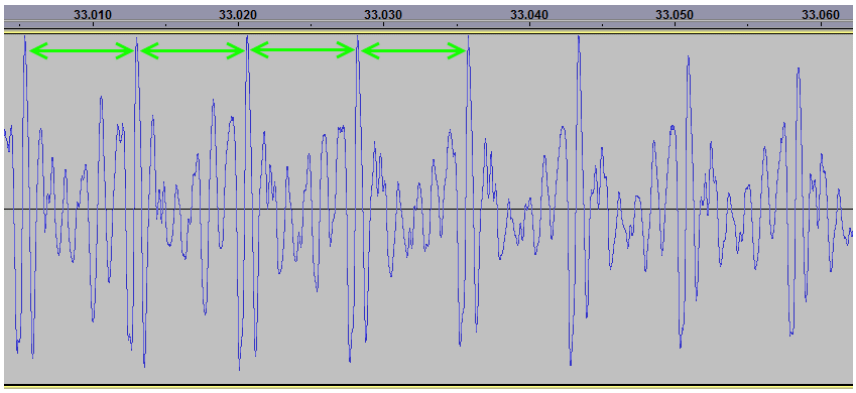
4.2.3. Period impulsa

Period impulsa (engl. pitch period, skraćeno PP) je parametar glasa koji je inverzan temeljnoj frekvenciji vibracije glasnica (4-7) [32].

$$pp = \frac{1}{f_g} \tag{4-7}$$

Pošto se glasnice ne koriste pri izgovoru bezvučnih glasova, period impulsa se definira isključivo za vremenske okvire govornog signala koji sadrže zvučni govor. Prema [32], postoje brojni načini za određivanje perioda impulsa, a temelje se na dva pristupa: određivanje iz vremenske domene i određivanje iz frekvencijske domene. Dvije najčešće metode za određivanje perioda impulsa koje koriste pristup određivanja iz vremenske domene su: autokorelacijska metoda kratkotrajnog vremenskog okvira (engl. short time autocorrelation function) i metoda prosječne razlike veličina (engl. average magnitude difference function, skraćeno AMDF). Pošto je u implementaciji korištena autokorelacijska metoda kratkotrajnog vremenskog okvira, isključivo će njena implementacija biti opisana u nastavku.

Uobičajena temeljna frekvencija glasnica je manja od 600 do 700 Hz. Stoga se prilikom određivanja perioda impulsa govorni signal propušta kroz nisko propusni filter koji filtrira komponente veće od 600 do 700 Hz. Glasovni signal je kvazi-periodičan za zvučne glasove, kao što je pisano u prethodnim poglavljima. Svaka od tehnika za određivanje perioda impulsa pokušava odrediti trajanje vremenskog perioda unutar jednog vremenskog okvira tako da na određeni način pronađe uzorke govornog signala koje predstavljaju impulse glasnica te izmjeriti vremensko trajanje između njih. Na slici 4.1 je prikazan valni oblik signala glasa „A“. Trajanje nekoliko perioda impulsa je označeno strelicama zelene boje.



Slika 4.1: Periodi impulsa na valnom obliku signala glasa „A“ označeni zelenim strelicama

Korelacija je vrlo česta tehnika koja se koristi u DSP procesorima kako bi se odredila vremenska razlika između dva signala, gdje je jedan od signala gotovo savršena preslika sa zakašnjenjem drugog signala. Autokorelacija je primjena iste tehnike kako bi se otkrio nepoznat period kvazi-periodičnog signala kao što je signal govora. Autokorelacijska funkcija s kašnjenjem od k uzoraka glasi:

$$\Phi[k] = \frac{1}{N}\sum_{n=0}^{N-1} s[n]s[n-k]\,. \tag{4-8}$$

Iz definicije (4-6) je vidljivo da je vrijednost $\Phi[k=0]$ jednaka prosječnoj energiji signala $s[n]$ koju signal ima na vremenskom okviru dužine N uzoraka. Kada bi signal $s[n]$ bio u potpunosti periodičan s periodom od P uzoraka tada bi vrijedilo $s[n+P] = s[n]$. Osim toga, vrijedilo bi i $\Phi[k=P] = \Phi[k=0]$ što je jednako prosječnoj energiji. Pošto govorni signali nisu periodični, nego kvazi-periodični, autokorelacijska funkcija za vrijednosti bliske PP poprima relativno velike vrijednosti. U slučajevima kada se vrijednosti k kreću između 0 i P, izraz $s[n]s[n-k]$ u formuli autokrelacijske funkcije (4-6) poprima različite pozitivne i negativne vrijednosti koje se unutar sumacije poništavaju, tako da autokorelacijska funkcija Φ za ulazne k između 0 i P ima postiže relativno male vrijednosti. Ako se raspolaže s okvirom zvučnog govornog signala s ukupno N uzoraka, graf funkcije $\Phi[k]$ kao funkcija k bio imao izrazite šiljke na vrijednostima gdje k iznosi $0, P, 2P, ...$, gdje $P = PP$. Graf funkcije $\Phi[k]$ bio poprimao male poprilično vrijednosti između šiljaka. Period impulsa P za okvir govornog signala se dobiva tako da se izmjeri udaljenost između šiljaka grafa autokorelacijske funkcije $\Phi[k]$.

Osim direktno iz govornog signala period impulsa je moguće mjeriti iz signala greške koji nastaje kao rezultat LPC analize. U projektima [P1] i [P2] je korištena procjena period impulsa iz signala greške LPC analize. U potpoglavlju LPC analize je objašnjeno zašto se period impulsa mjeri iz signala greške LPC analize.

4.3. Model linearno-prediktivnog kodiranja za prepoznavanje govora

U [33] je navedeno da je teorija linearno-prediktivnog kodiranja za primjene na govornim signalima relativno prihvaćena već duže vrijeme. LPC analiza je jedna od ponajboljih metoda za analizu govornog signala. LPC analiza je jedna od tehnika koja je postala dominantna za procjenu osnovnih parametara govornog signala, kao što su formanti, spektar, funkcije vokalnog trakta te pohranjivanje govornog signala pri visokom omjeru kompresije za pohranu ili za prijenos. Relativna jednostavnost primjene linearno-prediktivnog kodiranja nameće se kao idealno rješenje za implementaciju na NanoBoard-u na kojem su računalni resursi relativno ograničeni. U ovom potpoglavlju će biti opisan detaljan pregled svih matematičkih postupaka koji se tiču linearno-prediktivnog kodiranja.

U [33] autori navode četiri razloga za široku rasprostranjenost linearno-prediktivnog kodiranja.

1. LPC dobro modelira govorni signal. Kako je spomenuto u drugom poglavlju, kvazi-periodična priroda govornog signala (pogotovo zvučnih glasova) u LPC modelu s svepolnim filtrom (engl. all-pole filter) predstavlja dobru aproksimaciju ovojnice spektra govornog signala na kratkom uzorku. Prilikom bezvučnih i prijelaznih područja govornog signala LPC je manje učinkovit nego što je to slučaj za zvučna područja govornog signala, ali svejedno pruža koristan model za potrebe prepoznavnja govora.
2. Način na koji se LPC analiza primjenjuje na govornom signalu vodi do relativne odvojenosti izvora vokalnog trakta. Posljedično se dobiva štura reprezentacija značajki vokalnog trakta koje su direktno povezane sa stvaranjem glasa.
3. LPC je model koji se može analitički pratiti. Metoda LPC-a je matematički precizna i jednostavna za programsku implementaciju kao i za hardversku implementaciju.
4. LPC dobro funkcionira za primjene prepoznavanja govora što dokazuje to što se LPC koristi i danas u određenim sustavima za prepoznavanje govora.

Osnovna ideja LPC modela je da se dani uzorak signala govora u trenutku n, $s[n]$ može aproksimirati kao linearna kombinacija prošlih p uzoraka, tako da vrijedi:

$$s[n] \approx a_1 s[n-1] + a_2 s[n-2] + \cdots + a_p s[n-p], \tag{4-9}$$

pri čemu se pretpostavlja da su koeficijenti a_1, a_2, …, a_p konstantni na jednom okviru govornog signala. Približna jednakost se pretvara u potpunu jednakost ako se doda signal uzbude $Gu[n]$:

$$s[n] = \sum_{i=1}^{p} a_i \, s[n-i] + Gu[n] \,, \tag{4-10}$$

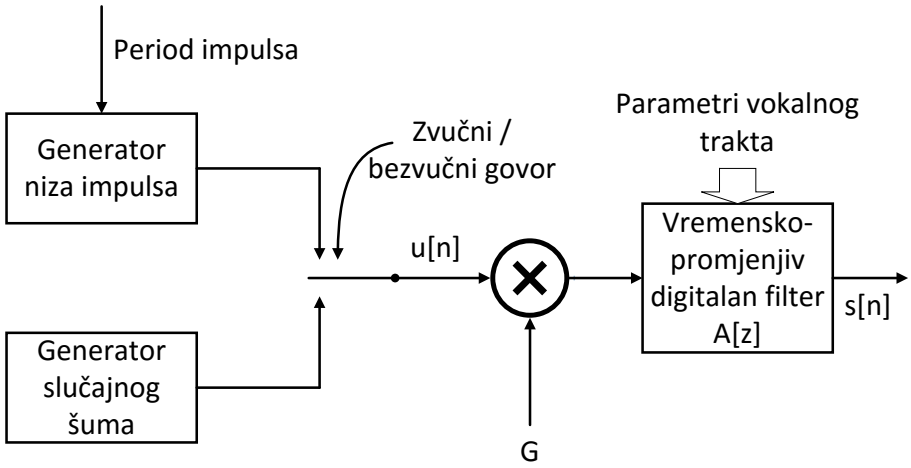
gdje je $u[n]$ normalizirana uzbuda, a G je dobitak uzbude. U [33] se navodi da se Z transformacijom jednadžbe (4-10) dobiva jednadžba:

$$S[z] = \sum_{i=1}^{p} a_i \, z^{-i} S[z] + GU[z] \,, \tag{4-11}$$

iz koje se izvodi prijenosna funkcija:

$$H[z] = \frac{S[z]}{GU[z]} = \frac{1}{1 - \sum_{i=1}^{p} a_i z^{-i}} = \frac{1}{A[z]} \,. \tag{4-12}$$

Temeljeno na prijašnjim saznanjima, poznato je da je funkcija ulaznog signala uzbude kvazi-periodični niz impulsa ako je u pitanju zvučni glasovni signal. Ako je u pitanju bezvučni glas ulazni signal uzbude čini generator slučajnog šuma. Razliku između zvučnih i bezvučnih glasova u stvaranju čini korištenje glasnica. U zvučnim glasovima se koriste glasnice, dok se pri stvaranju bezvučnih glasova ne koriste glasnice. Dobitak uzbude G se procjenjuje iz govornog signala, stoga se skalirani izvor koristi kao ulaz u digitalni filter $H[z]$, pod kontrolom parametara značajki govora koji se u danom trenutku proizvodi u usnoj šupljini. Stoga su parametri ovog modela zvučna-bezvučna klasifikacija, period titranja za zvučne glasove, dobitak uzbude te koeficijenti digitalnog filtra $\{a_k\}$. Svi navedeni parametri polako variraju u vremenu. Model stvaranja govornog signala baziran na LPC-u je prikazan slikom 4.2.



Slika 4.2: Model stvaranja govornog signala baziran na LPC

4.3.1. Pred-naglašavanje

Transfer funkcija vokalnog trakta glasi:

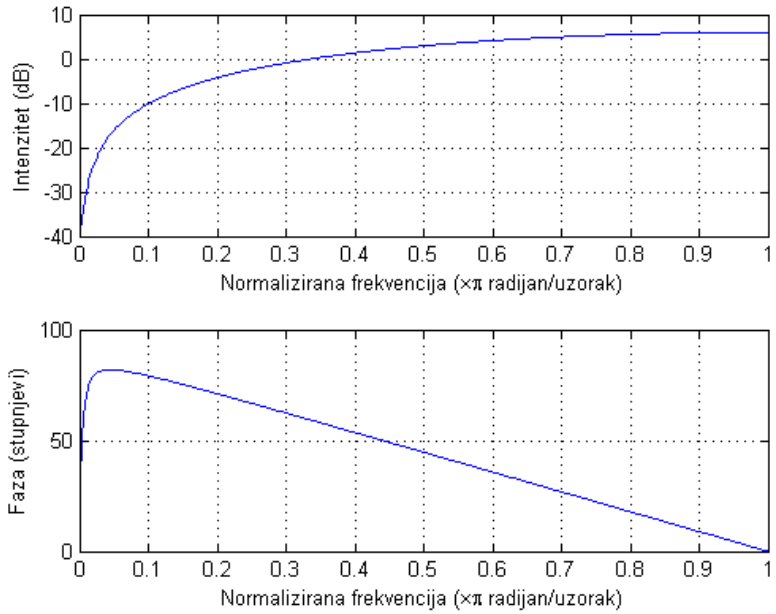
$$\frac{S[z]}{E[z]} = A_v \frac{1}{(1-z^{-1})^2} \frac{1}{1+\sum_{k=1}^{P} a_k z^{-k}} (1-z^{-1}). \tag{4-13}$$

Postoji trend od -6dB/oktavi s porastom frekvencije. Poželjna je kompenzacija ovog učinka pred-obradom govornog signala. Pred-obrada govornog signala koja ima učinak blokiranja učinka glotisa (dio grkljanske šupljine) poznata je kao pred-naglašavanje (engl. pre-emphasis filter). Pred-naglašavanje se postiže visokopropusnim filtriranjem koristeći jednadžbu diferencija:

$$y[n] = s[n] - as[n-1]. \tag{4-14}$$

Vrijednosti koeficijenta a iznose između 0.9 i 1 [32]. U implementaciji je korišten koeficijent $a = 0{,}99$ čija je amplitudna i fazna karakteristika nacrtana u Matlabu na slici 4.3. Prijenosna funkcija filtra pred-naglašavanja opisana jednadžbom diferencija (4-14) glasi:

$$H[z] = 1 - az^{-1}. \tag{4-15}$$



Slika 4.3: Amplitudna i fazna karakteristika filtra pred-naglašavanja s koeficijentom $a = 0{,}99$

Pred-naglašavanje je implementiranu u [P1] i [P2] u datoteci *feature_extraction.c* funkcijom `void filter_pre_emphasis(int16_t* input_sample, int16_t* output_sample, int sample_size)`. Prvi parametar je ulazni niz, drugi parametar je izlazni filtriran niz, a treći parametar je veličina oba niza.

4.3.2. Jednadžbe LPC analize

Bazirano na modelu prikazanom slikom 4.2, točna relacija između $s[n]$ i $u[n]$ je:

$$s[n] = \sum_{k=1}^{p} a_k \, s[n-k] + Gu[n] \, . \tag{4-16}$$

Razmatramo linearnu kombinaciju prethodnih uzoraka govornog signala procjenom $\tilde{s}[n]$, definiranu kao:

$$\tilde{s}[n] = \sum_{k=1}^{p} a_k \, s[n-k] \, . \tag{4-17}$$

Definira se greška predikcije jednadžbom:

$$e[n] = s[n] - \tilde{s}[n] = s[n] - \sum_{k=1}^{p} a_k \, s[n-k] \, , \tag{4-18}$$

a Z transformacijom dobivamo prijenosnu funkciju greške:

$$A[z] = \frac{E[z]}{S[z]} = 1 - \sum_{k=1}^{p} a_k z^{-k} \, . \tag{4-19}$$

Vidljivo je da ako je govorni signal stvoren linearnim sustavom predstavljenim modelom 4.2 koga opisuju jednadžbe (4-10) i (4-16), onda će signal greške predikcije biti isto što i član $Gu[n]$ koji predstavlja pojačani signal uzbude u spomenutim jednadžbama.

Osnovni problem linearno-predikcijske analize je odrediti skup predikcijskih koeficijenata $\{a_k\}$ izravno iz govornog signala takvih da spektralna svojstva digitalnog filtra na slici 4.2 odgovaraju valnom obliku trenutnog okvira govornog signala. Pošto spektralne karakteristike govora variraju s vremenom, predikcijski koeficijenti se procjenjuju na kratkom vremenskom odsječku govornog signala. Stoga se problem pronalaženja skupa predikcijskih koeficijenata svodi na traženje najmanje srednje kvadratne greške na kratkom vremenskom okviru govornog signala. Inače se ovakav tip analize provodi na svakom uzastopnom vremenskom okviru govornog signala trajanja 10 do 20 milisekundi, a uobičajeno se koeficijenti stvaraju za preklapajuće okvire, takve da se između svakog uzastopnog okvira generira skup koeficijenata za preklapajući okvir koji uzima zadnju polovicu prethodnog okvira i prvu polovicu sljedećeg okvira.

Za postavljanje jednadžbi koje trebaju rješenje kako bi odredili skup predikcijskih koeficijenata, definiraju se kratkotrajni okviri govornog signala te signala greške u trenutku n jednadžbama:

$$s_n[m] = s[n+m]\,, \tag{4-20}$$

$$e_n[m] = e[n+m]. \tag{4-21}$$

Nastoji se smanjiti signal srednje kvadratne greške u trenutku n:

$$E_n = \sum_m e_n^2[m]. \tag{4-22}$$

Jednadžba (4-22) prelazi u oblik:

$$E_n = \sum_m \left[s_n[m] - \sum_{k=1}^{p} a_k s_n[m-k] \right]^2, \tag{4-23}$$

ako se $e_n[m]$ izrazi pomoću $s_n[m]$. U svrhu dobivanja skupa predikcijskih koeficijenata, rješava se jednadžba (4-23) diferenciranjem E_n po svakom a_k te izjednačavajući izraz s nulom:

$$\frac{\partial E_n}{\partial a_k} = 0, \qquad k = 1, 2, 3, \ldots, p\ , \tag{4-24}$$

čime se dobiva:

$$\sum_m s_n[m-i]s_n[m] \ = \sum_{k=1}^{p} \hat{a}_k \sum_m s_n[m-i]s_n[m-k]\,. \tag{4-25}$$

Drugi član umnoška desne strane jednadžbe (4-25) je kovarijanca signala $s_n[m]$ na kratkom odsječku te je možemo zapisati jednadžbom:

$$\Phi_n[i,k] = \sum_m s_n[m-i]s_n[m-k]\,. \tag{4-26}$$

Jednadžba (4-25) zapisana kraće pomoću jednadžbe (4-26) glasi:

$$\Phi_n[i,0] = \sum_{k=1}^{p} \hat{a}_k \Phi_n[i,k]\,. \tag{4-27}$$

Jednadžba (4-27) definira skup p jednadžbi s p nepoznanica. Već je pokazano da se minimum srednje kvadratne greške $\hat{E}_n$ može iskazati kao:

$$\hat{E}_n = \sum_m {s_n}^2[m] - \sum_{k=1}^{p} \hat{a}_k \sum_m s_n[m-i]s_n[m-k]\,, \tag{4-28}$$

$$\hat{E}_n = \Phi_n[0,0] - \sum_{k=1}^{p} \hat{a}_k \Phi_n[0,k]. \tag{4-29}$$

Jednadžba (4-29) označava da se minimalna srednja kvadratna greška sastoji od nepromjenjivog člana $\Phi_n[0,0]$ i od člana koji ovisi o predikcijskim koeficijentima.

Za određivanje optimalnih predikcijskih koeficijenata $\hat{a}_k \Phi_n$ potrebno je proračunati $\Phi_n[i,k]$ gdje je $i \in [1,p]$, a $k \in [0,p]$ čime se dobiva p simultanih jednadžbi. Metoda rješavanja jednadžbi kao i metoda izračuna Φ_n je u praksi funkcija jake konveksnosti dometa m korištena u određivanju područja govora za analizu i područja na kojem se računa srednja kvadratna greška. U [33] su opisane dvije metode za određivanje područja govornog signala: autokorelacijska metoda te metoda kovarijance. Pošto je u implementaciji korištena autokorelacijska metoda, isključivo će ona biti opisana.

4.3.3. Autokorelacijska metoda

U [33] se navodi kako postoji jednostavan i izravan način definiranja granica m u sumaciji je pretpostaviti da je uzorak govornog signala $s_n[m]$ jednak nuli izvan intervala $m \in [0, N-1]$. Ova tvrdnja je jednaka tvrdnji da se govorni signal $s[m+n]$ množi prozorom konačne veličine $w[m]$ koji je jednak nuli u intervalu izvan $m \in [0, N-1]$. Zbog toga, uzorak govora za minimizaciju se može izraziti kao:

$$s_n[m] = \begin{cases} s[n+m] \cdot w[m], & m \in [0, N-1] \\ 0, & inače. \end{cases} \tag{4-30}$$

U praksi se najčešće koristi Hammingov prozor za uokvirivanje signala. Takav oblik prozora omogućuje smanjivanje utjecaja impulsa na početku i kraju okvira govornog signala, koji se pojavljuje kod zvučnih govornih signala. Najvažniji razlog takvog načina uokvirivanja je smanjivanje srednje kvadratne greške na početku i kraju okvira signala. [33]

Stvaranje Hammingovog prozora je ostvareno funkcijom `void hamming_window(float* w, int size)` u datoteci *feature_extraction.c* u projektima [P1] i [P2]. Prvi parametar funkcije je niz u kojem će biti spremljeni iznosi uzoraka prozora nakon izvršenja funkcije, a drugi parametar je veličina prozora.

Temeljeno na uokvirenom signalu prema jednadžbi (4-30), srednja kvadratna greška se računa:

$$E_n = \sum_{m=0}^{N-1+p} e_n^2[m]. \tag{4-31}$$

$\Phi_n[0,0]$ se može izraziti kao:

$$\Phi_n[\mathrm{i},\mathrm{k}] = \sum_{m=0}^{N-1+p} s_n[m-i]s_n[m-k], \quad \begin{matrix} i \in \langle 1,p \rangle \\ k \in \langle 0,p \rangle \end{matrix}, \tag{4-32}$$

ili kao:

$$\Phi_n[\mathrm{i},\mathrm{k}] = \sum_{m=0}^{N-1-(i-k)} s_n[m]s_n[m+i-k], \qquad \begin{matrix} i \in \langle 1,p \rangle \\ k \in \langle 0,p \rangle \end{matrix}. \tag{4-33}$$

Očigledno je da je jednadžba (4-33) funkcija varijable $(i - k)$, a ne dvije odvojene i i k. Stoga se kovarijacijska funkcija $\Phi_n[\mathrm{i},\mathrm{k}]$ svodi na jednostavnu autokorelacijsku funkciju:

$$\Phi_n[\mathrm{i},\mathrm{k}] = r_n[i-k] = \sum_{m=0}^{N-1-(i-k)} s_n[m]s_n[m+i-k]. \tag{4-34}$$

Autokorelacijska funkcija je parna što znači da vrijedi: $r_n[i-k] = r_n[k]$. Stoga se LPC jednadžbe mogu izraziti kao:

$$\sum_{p=0}^{k} r_n[|i-k|]\hat{a}_k = r_n[i], \qquad i \in \langle 1,p \rangle . \tag{4-35}$$

Jednadžbu (4-35) možemo izraziti u matričnom obliku:

$$\begin{bmatrix} r_n[0] & r_n[1] & r_n[2] & & r_n[p-1] \\ r_n[1] & r_n[0] & r_n[1] & \cdots & r_n[p-2] \\ r_n[2] & r_n[1] & r_n[0] & & r_n[p-3] \\ & \vdots & & \ddots & \vdots \\ r_n[p-1] & r_n[p-2] & r_n[p-3] & \cdots & r_n[0] \end{bmatrix} \begin{bmatrix} \hat{a}_1 \\ \hat{a}_2 \\ \hat{a}_3 \\ \vdots \\ \hat{a}_p \end{bmatrix} = \begin{bmatrix} r_n[1] \\ r_n[2] \\ r_n[3] \\ \vdots \\ r_n[\mathrm{p}] \end{bmatrix}. \tag{4-36}$$

Kako je navedeno u [33], matrica autokorelacijskih vrijednosti dimenzija $p \times p$ je Toeplitzova matrica. Topelitzova matrica je simetrična s jednakim vrijednostima na dijagonalama. Pošto su koeficijenti sustava (4-36) definirani Toeplitzovom matricom, sustav je moguće učinkovito riješiti pomoću nekoliko poznatih metoda. Metoda koje je korištena u implementaciji se naziva Durbinova metoda ili Levinson-Durbinova rekurzija.

U projektima [P1] i [P2] autokorelacijska funkcija je implementirana u datoteci *feature_extraction.c* kao `void auto_corellation(int16_t* input_sample, int size, float* ac_out, int order)`. Prvi parametar funkcije je ulazni niz uzoraka, drugi parametar je veličina ulaznih uzoraka. Treći parametar je izlazni niz u koji će biti zapisani iznosi autokorelacije. Četvrti parametar je red autokorelacije.

4.3.4. Levinson-Durbinova rekurzija

Durbinov algoritam ili Levinson-Durbinova rekurzija je numerički postupak koji služi za rješavanje linearnog sustava jednadžbi u kojima su koeficijenti nezavisnih varijabli smješteni u Toeplitzovu matricu, kao što je to slučaj za (4-36). Durbinov algoritam se sastoji od sljedećih jednadžbi koje se rješavaju prema redu pojavljivanja:

$$E^{(0)} = r[0] \tag{4-37}$$

$$a_i^{(i)} = \frac{r[i] - \sum_{j=1}^{i-1} a_{i-1}^{(j)} r[i-j]}{E^{(i-1)}}, \quad i \in [1, p] \tag{4-38}$$

$$a_j^{(i)} = a_j^{(i-1)} - \ a_i^{(i)} a_{i-j}^{(i-1)}, \quad j \in [1, i\rangle \tag{4-39}$$

$$E^{(i)} = \frac{(\ 1\ - (a_i^{(i)})^2\ )}{E^{(i-1)}} \tag{4-40}$$

$E^{(i)}$ predstavlja pomoćnu varijablu koja pojednostavljuje proračun jednadžbe (4-38). Gornji broj u zagradi kod varijabli predstavlja broj iteracije, što vidimo po tome što se u zagradi nalazi funkcija varijable i, dok donji broj predstavlja redni broj određenog koeficijenta. Na primjer: $a_j^{(i)}$ predstavlja j-ti koeficijent nezavisne varijable a iz i-te iteracije; $a_1^{(1)}$ predstavlja 1-ti koeficijent nezavisne varijable a iz 1-te iteracije; $a_2^{(12)}$ predstavlja 2-ti koeficijent nezavisne varijable a iz 12-te iteracije; $E^{(6)}$ predstavlja pomoćnu varijablu kojoj je vrijednost izračunata u šestoj iteraciji. Provođenje Durbinovog algoritma se odvije prema sljedećim koracima [34]:

Postavi $\boldsymbol{E^{(0)}}$, (4-37).
Za svaki $\boldsymbol{i}$, od **1** do $\boldsymbol{p}$:
  Izračunaj $\boldsymbol{a_i^{(i)}}$ prema (4-38).
  Za svaki $\boldsymbol{j}$, od **1** do $\boldsymbol{i}$, izračunaj:
    Izračunaj $\boldsymbol{a_j^{(i)}}$ prema (4-39).
  Izračunaj $\boldsymbol{E^{(i)}}$ prema (4-40).

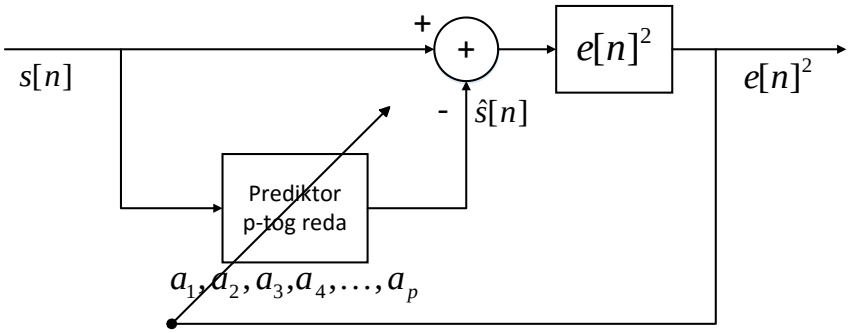
Algoritam 4.1: Algoritam Levinson-Durbinove rekurzije

Primjer izračuna LPC koeficijenata prema Levinson-Durbinovoj rekurziji se nalazi u prilogu [S1].

Algoritam Levinson-Durbinove rekurzije je implementiran funkcijom `void lpc_durbin(float* r, float** lpc, int lpc_size, float* e)` u datoteci `feature_extraction.c` u projektima [P1] i [P2]. Prvi parametar funkcije je niz u kojemu su pohranjeni rezultati potrebne autokorelacije. Drugi parametar je niz nizova LPC koeficijenata koji će biti izračunati nakon izvođenja funkcije. Treći parametar je red LPC-a. Četvrti parametar je niz pomoćne varijable označene s $E^{(i)}$ u ovom potpoglavlju.

4.3.5. Algoritam strmog spusta

Algoritam strmog spusta (engl. Steepest Descent) je algoritam s kojim se također računaju LPC koeficijenti. Na slici 4.4 je prikazan blok-dijagram linearnog prediktora.



Slika 4.4: Blok-dijagram linearnog prediktora

Predikcijski koeficijenti se prilagođuju kontinuirano tijekom analize radi smanjivanja kvadratne predikcijske greške $e[n]^2$ na najmanju moguću razinu. Jednadžba blok dijagrama prikazana slikom 4.4 je:

$$e[n]^2 = \left[s[n] - \sum_{k=1}^{p} a_k s[n-k] \right]^2, \tag{4-41}$$

gdje se a_k koeficijenti mijenjaju ovisno o iznosu $e[n]^2$. Koeficijenti se ažuriraju prema algoritmu strmog spusta. Prediktorski koeficijenti se ažuriraju za svaki novi uzorak prema slijedećoj jednadžbi:

$$a_k[n+1] = a_k[n] - c\frac{\partial[e[n]^2]}{\partial a_k}, \qquad c \in \langle 0{,}1 \rangle, \tag{4-42}$$

gdje c predstavlja stopu učenja. Stopa učenja korištena u implementacija u prilozima [P1] i [P2] iznosi $c = 0{,}3$. Derivacijom kvadratne greške u jednadžbi (4-42) dobiva se izraz:

$$\frac{\partial[e[n]^2]}{\partial a_k} = 2\, e[n]\frac{\partial e[n]}{\partial a_k} = -2\, e[n]\, s[n-k]. \tag{4-43}$$

Uvrštavanjem izraza (4-43) u (4-42) dobivaju se predikcijski koeficijenti:

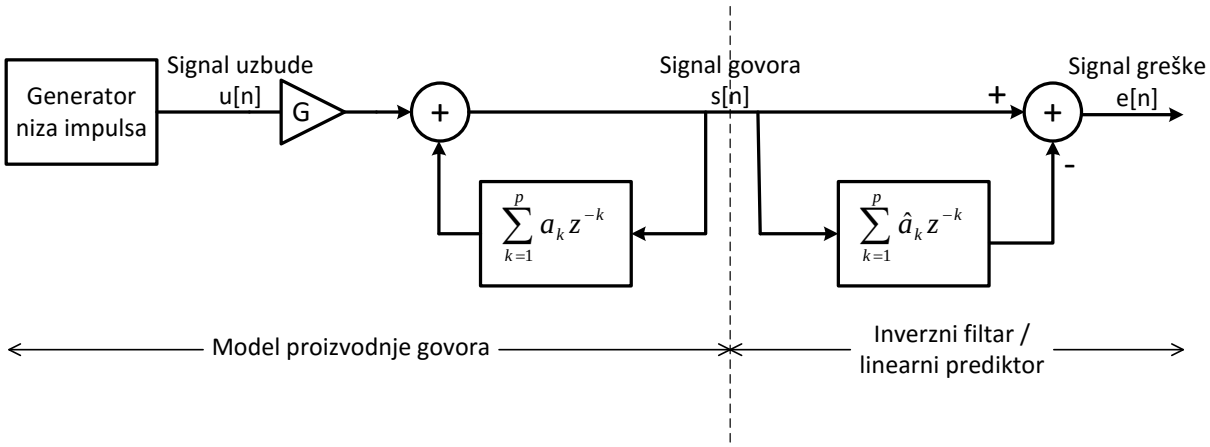
$$a_k[n+1] = a_k + c\, e[n]\, s[n-k], \qquad k \in [1, p]. \tag{4-44}$$

Algoritam strmog spusta, isto kao i Levinson-Durbinova rekurzija, služi za određivanje LPC koeficijenata. U implementaciji u prilozima [P1] i [P2] se koriste oba algoritma, na točno koji način je objašnjeno u nastavku poglavlja.

Algoritam strmog spusta je implementiran funkcijom `void steepest_descent(float** lpc_in, float* lpc_out, int lpc_size, int16_t* input_sample, int16_t* err, int frame_size, float c_sd)` u datoteci *feature_extraction.c*. Prvi ulazni parametar funkcije čine izračunati LPC koeficijenti. Drugi parametar funkcije čine izlazni LPC koeficijenti. Treći parametar je niz koji predstavlja signal greške. Četvrti parametar funkcije je veličina vremenskog okvira signala. Posljednji parametar čini stopu učenja algoritma strmog spusta.

4.3.6. Mjerenje perioda impulsa iz signala greške

Period impulsa je moguće procijeniti iz signala greške koji nastaje kao rezultat LPC analize. Prema LPC modelu govora, prilikom stvaranja zvučnih glasova izvorni signal modela čini generator impulsnog niza. Period impulsa je mjera koja procjenjuje udaljenosti između impulsa generatora impulsnog niza. Prema slici 4.5, LPC analiza stvara inverzan filtar u odnosu na filtar koji se nalazi u modelu proizvodnje govora.



Slika 4.5: Model proizvodnje govora zvučnog govora i inverzan filtar dobiven LPC analizom

Filtriranjem govornog signala inverznim filtrom nastalim LPC analizom efektivno se uklanjaju efekti filtra u modelu proizvodnje govora, što čini signal greške $e[n]$ gotovo jednakim signalu $Gu[n]$. Ovo svojstvo omogućuje procjenu perioda impulsa iz signala greške. Procjena perioda impulsa iz signala greške je računski manje zahtjevna nego procjena iz signala govora jer nije potrebno dodatno filtrirati signal govora, a sam signal greške nastaje kao nusprodukt LPC analize.

Pošto je za stvaranje signala greške potrebno filtrirati ulazni signal FIR (engl. finite impulse respone – konačni impulsni odziv) filtrom, u projektima [P1] i [P2] je implementirana funkcija filtriranja FIR filtrom. Funkcija `void filter_throught_lowpass_fir(int16_t*`

`input_sample, int16_t* output_sample, int sample_size, float* b, int b_size)` se nalazi u dateteci *feature_extraction.c*. Prvi parametar čini ulazni niz uzoraka signala, nakon čega slijedi izlazni niz uzoraka signala. Treći parametar je broj uzoraka ulaznog i izlaznog niza. Slijedi niz koeficijenata FIR filtra s njihovim brojem.

Procjena perioda impulsa je u projektima [P1] i [P2] implementirana funkcijom `float average_pitch_period(int16_t* input_sample, int size)`, u datoteci *feature_extraction.c*. Parametre funkcije čine niz i broj uzoraka nad kojima se proračunava period impulsa.

4.3.7. Normalizirana križna korelacija

U implementaciji je potrebno usporediti LPC koeficijente snimljenog signala s koeficijentima usrednjenih značajki snimljenog signala. Zbog jednostavnosti primjene i zahtjeva za računanjem u realnom vremenu za mjeru udaljenosti dva skupa LPC koeficijenata je korištena normalizirana križna korelacija (engl. normalised cross-corelation).

Zadana su dva skupa LPC koeficijenata, s ukupno p elemenata: a_i i b_i, $i \in \langle 1, p \rangle$. Normalizirana križna korelacija za ta dva skupa glasi:

$$a \star b = \frac{\sum_{i=1}^{p}(a_i - \bar{a})\left(b_i - \bar{b}\right)}{\sqrt{\sum_{i=1}^{p}(a_i - \bar{a})^2 \sum_{i=1}^{p}\left(b_i - \bar{b}\right)^2}}, \tag{4-45}$$

gdje je $\bar{a}$ i $\bar{b}$ srednja vrijednost skupova a_i i b_i. Normalizirana križna korelacija poprima vrijednosti između -1 i 1, što ovisi u sličnosti skupova a_i i b_i. Križna korelacija za dva potpuno identična skupa poprima vrijednost 1. Što su skupovi manje sličnih vrijednosti rezultat križne korelacije je manji.

Normalizirana križna korelacija implementirana je u *voice_features.c* datoteci ugrađenog projekta u prilogu [P1] u funkciji `float cross_corelation(float* a, float* b, int size)`. Prva dva ulazna parametra čine nizovi koje je potrebno usporediti. Posljednji parametar čini veličinu nizova koje je potrebno usporediti. Povratna vrijednost funkcije je iznos izračunate križne korelacije.

4.3.8. Relativna udaljenost

U prilogu [P1] broj prelazaka kroz nulu, period impulsa kao i LPC koeficijenti su korišteni za konačno odlučivanje o izgovorenom glasu. Kako bi se saznale udaljenosti značajki broja

prelazaka kroz nulu i perioda impulsa, korištena je mjera relativne udaljenosti. Relativna udaljenost D_x značajke x je računata po slijedećem pravilu:

$$D_x = \begin{cases} 1 - \frac{|x - \bar{x}|}{\bar{x}}, & ako\ vrijedi\ 1 - \frac{|x - \bar{x}|}{\bar{x}} > 0 \\ 0, & inače, \end{cases} \quad (4\text{-}46)$$

gdje je $\bar{x}$ usrednjena vrijednost značajke od koje se računa udaljenost. Ovako postavljena mjera udaljenosti definira raspon vrijednosti između 1 i 0. Uvjetom se onemogućuju negativne vrijednosti što mjeru čini pogodnom za osmišljavanje jednostavne mjere za skupno mjerenje svih značajki govora u [P1].

Relativna udaljenost je implementirana u *voice_features.c* datoteci ugrađenog projekta [P1] kao funkcije `float relative_distance_int(int a, int a_avg)` i `float relative_distance_float(float a, float a_avg)`. Ulazni parametri nacedenih funkcija su vrijednosti značajki za koje je potrebno usporediti udaljenost. Funkcija vraća vrijednost koja relativne udaljenosti prema (4-46).

4.4. Sustav za stvaranje usrednjenih značajki glasova

Sustav za stvaranje usrednjenih značajki glasova se sastoji od dva dijela: sustava za stvaranje značajki glasovnog signala priloženog u projektu [P2] i sustava za stvaranje srednjih vrijednosti značajki iz tekstualne datoteke u projektu [P3]. Svrha sustava za stvaranje usrednjenih značajki glasova je stvaranje usrednjenih značajki glasova koje služe kao parametri za mjerenje i usporedbu u konačnoj implementaciji sustava za prepoznavanje glasova.

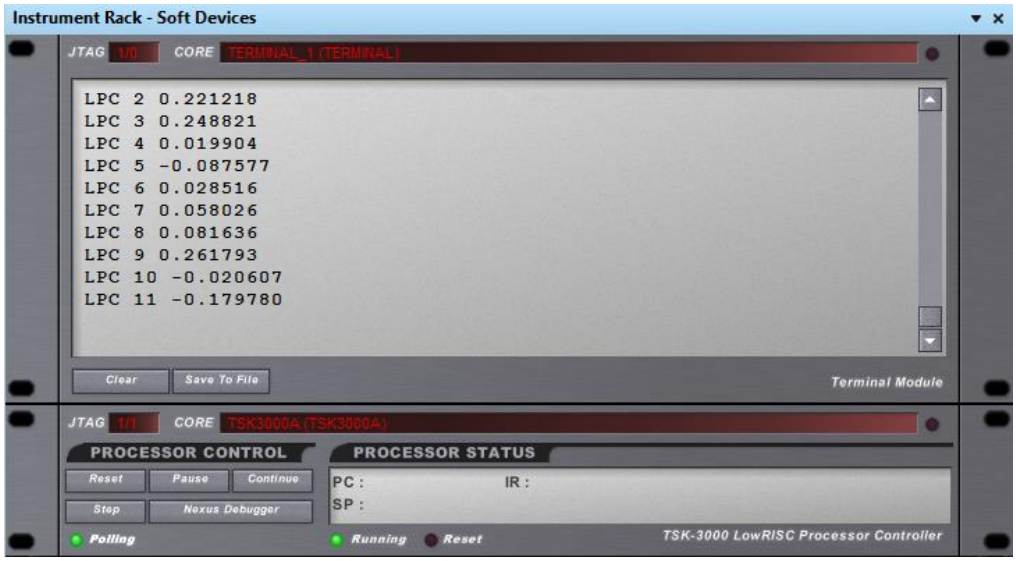
4.4.1. Sustav za stvaranje značajki glasovnog signala

Sustav za stvaranje značajki glasovnog signala je ugrađeni projekt za shematski projekt opisan u poglavlju 3 [P2]. Svrha sustava je ispis značajki signala jednog glasa hrvatskog jezika u tekstualnu datoteku. Ulaz u sustav je analogni električni signal koji je dobiven reprodukcijom snimka glasa na osobnom računalu. Osobno računalo reproducira zvuk snimljenog glasa na 3.5 milimetarskom audio izlazu koji je kablom spojen na 3.5 milimetarski audio ulaz na ploči.

Snimci glasova hrvatskog jezika koji su korišteni za reprodukciju za nalaze u prilogu [P4] te ih nazivamo izvornim snimcima. Izvorni snimci su snimljeni ugrađenim mikrofonom koji se nalazi na prijenosnom računalu ASUS X750LB programom Audacity. Jedan izvorni snimak snimljen je u nastojanju da bude reprezentativan snimak glasa hrvatskog jezika. Svi izvorni snimci potječu od iste muške osobe 20-tih godina.

U hrvatskom jeziku postoje fonemi koji se izgovaraju kontinuirano, bez zastoja zračne struje i fonemi koji se ne mogu izgovoriti kontinuirano. Kontinuirani glasovi su prilikom snimanja izvornih snimaka izgovoreni u potpunosti kontinuirano. Glasovi koji se ne mogu izgovoriti kontinuirano su snimljeni nastojeći ih govoriti neprestano ponavljajući bez pauza ili pak kontinuirano. Ako je nekontinuirani glas izrečen i snimljen kontinuirano, prilikom izgovora je nastojano zadržati vokalni trakt u obliku kojemu se taj glas najduže nalazi. Uz ime snimka je navedeno da li je izrečen kontinuirano ili isprekidano.

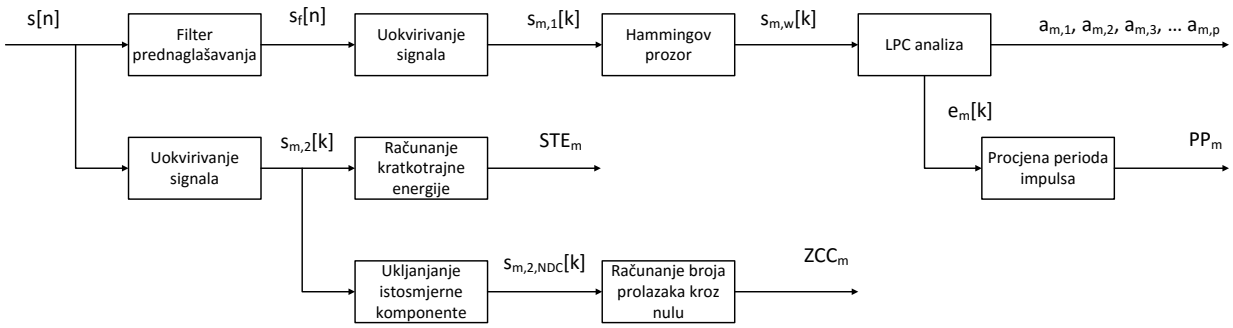
Projekt [P2] koristi programski implementiran terminal koji koristi JTAG komunikaciju na ploči. U terminal se ispisuje koristeći $printf()$ funkciju u C programskom kodu. Kontrole procesora i terminal usred ispisivanja značajki su prikazani slikom 4.6.



Slika 4.6: Prozor terminala za komunikaciju

Za implementaciju u [P1] je izabrana LPC analiza. Zbog toga je potrebno stvorit usrednjene značajke govora koje se koriste u LPC analizi. Značajke potrebne za LPC analizu su kratkotrajna energija signala, broj prelazaka kroz nulu, period impulsa i LPC koeficijenti. Stvaranje značajki signala je opisano u ovom poglavlju.

Princip rada stvaranja značajki govornog signala projekta [P2] prikazan je slikom 4.7.



Slika 4.7: Princip stvaranja značajki govornog signala

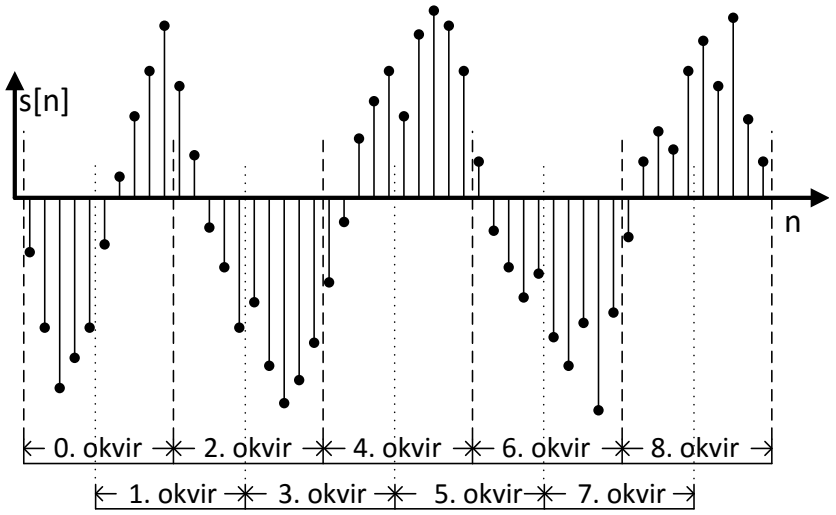
Na slici 4.7 je prikazan ulaz u sustav $s[n]$ koji se odmah na početku račva na dva dijela, a predstavlja digitalni ulazni audio signal. Signal $s[n]$ ulazi u gornji dio sustava iz kojega se na kraju stvaraju LPC koeficijenti i period impulsa za svaki okvir ulaznog signala. Ulazni signal $s[n]$ najprije ulazi u filter prednaglašavanja koji je opisan ranije u ovom poglavlju. Izlaz iz filtera je označen s $s_f[n]$. Nakon toga se provodi uokvirivanje signala, što će biti objašnjeno u nasatvku. Nakon uokvirivanja signala nastaje skup signala $s_{m,1}[k]$, gdje oznaka m označava redni broj uokvirenog signala. Na svakom uokvirenom signalu se provodi množenje s funkcijom Hammingovog prozora, tako da na izlazu imamo skup signala $s_{m,2,w}[k]$ koji su pušteni kroz Hammingov prozor. Spomenuti signali ulaze u blok za LPC analizu. Iz bloka za LPC analizu postoje dva izlaza. Prvi izlaz čine LPC koeficijenti dobiveni postupkom LPC analize koji je opisan kasnije u ovom poglavlju. LPC koeficijenti su računati za svaki m-ti okvir signala te su označeni s $a_{m,1}$, $a_{m,2}$, $a_{m,3}$, … $a_{m,p}$. U implementaciji [P1], pa tako i u [P2], računa se $p = 12$ LPC koeficijenata. Drugi izlaz LPC analize čini signal greške $e_{m,w}$ na temelju kojega se procjenjuje period impulsa PP_m za svaki okvir signala m.

Blok za LPC analizu najprije proračunava LPC koeficijente Levinson-Durbinovom rekurzijom. Dobiveni koeficijenti su nakon toga korišteni kao početni koeficijenti za algoritam strmog spusta za proračun na istom vremenskom okviru signala. Postupak je proveden na ovaj način kako bi se što više smanjio izlazni signal greške.

Drugi dio u kojega ulazi signal $s[n]$ u konačnici služi za računanje kratkotrajne energije signala te za računanje broja prolazaka kroz nulu za svaki okvir ulaznog signala.

Analiza značajki signala opisana u ovom poglavlju se provodi na kratkim vremenskim okvirima govornog signala trajanja između 10 i 20 milisekundi. U implementaciji [P1], pa stoga i u sustavu za stvaranje značajki [P2], ulazni analogni signal je uzorkovan frekvencijom od 16 000 kHz.

Trajanje jednog okvira govornog signala iznosi 12.5 milisekundi, što znači da jedan okvir sadržava 200 uzoraka signala. Okviri se signala se stvaraju za svakih 200 uzoraka signala uzastopno, a osim uzastopnih okvira, stvaraju se i okviri koji su nastali preklapanjem prednje polovice prethodnog okvira i stražnje polovice prethodnog okvira. Uokvirivanje signala se koristi kod sustava za stvaranje značajki govornog signala što možemo vidjeti na slici 4.7. Princip stvaranja okvira je prikazan slikom 4.8.



Slika 4.8: Prikaz uokvirivanja signala, između dva susjedna okvira dolazi jedan koji preklapa oba okvira

4.4.2. Sustav za stvaranje srednjih vrijednosti značajki

Nakon što su stvorene značajke signala sustavom [P2] one se spremaju u *.log* datoteku. Datoteka značajki se obrađuje tako da bude formatirana na način opisan u formatu 4.1.

```
0 frame                  n frame                  298 frame
LPC 1 1.207103           LPC 1 0.959143           LPC 1 0.826362
LPC 2 -0.483756          …                        …
LPC 3 0.286573           …                        …
LPC 4 -0.317703          …                        …
LPC 5 0.075318           …                        …
LPC 6 0.164228           …                        …
LPC 7 0.042887           …                        …
LPC 8 -0.089326          …                        …
LPC 9 0.091076           …                        …
LPC 10 -0.523877         …                        …
LPC 11 0.875556          …                        …
LPC 12 -0.484336         LPC 12 -0.219250         LPC 12 0.081051
STE 4933866000.000000    STE 6204501000.000000    STE 1836468000.000000
ZCC 15                   ZCC 11                   ZCC 2
PP 3.010638              PP 3.551724              PP 4.530120
```

Format datoteke 4.1: Format datoteke koji je potreban za ulaz u [P3]

Datoteka se sastoji od m skupova značajki okvira signala ispisanih jedan iza drugoga. U sustavu za stvaranje značajki glasovnog signala [P2] postavljeno je da se stvara $m = 299$ skupova značajki, od kojih svaki pripada jednom okviru snimljenog signala. Broj okvira odnosno skupova značajki glasovnog signala je podesiv u programskom kodu projekta [P1] i projekta [P2]. Svaki skup značajki započinje najprije rednim brojem značajke te riječju „frame“ (engl. okvir) odvojenu razmakom, na primjer: „0 frame“. U sljedećih $p = 12$ redova su ispisani LPC koeficijenti jednog okvira snimljenog glasovnog signala u formatu: „LPC p vrijednost„. Nakon ispisa LPC koeficijenta slijedi ispis ostalih značajki okvira, a to su kratkotrajna energija signala, broj prelazaka kroz nulu te period impulsa. Navedene karakteristike se ispisuju navedenim redoslijedom u formatu: „STE vrijednost“, novi red, „ZCC vrijednost“, novi red, „PP vrijednost“. Nakon navedenog ispisa slijedi ispis značajki slijedećeg okvira, sve do posljednjeg okvira. Datoteka navedenog formata se pohranjuje s ekstenzijom *.txt*.

Za svrhu stvaranja usrednjenih značajki glasovnog signala napravljen je projekt u Visual Studio Express razvojnom sučelju u C++ programskom jeziku. Projekt je napravljen kao komandno-linijski program. Pokretanjem programa otvara se prozor u kojemu se od korisnika pita putanja datoteke koja je formatirana na način opisan formatom 4.1. Nakon toga program traži cijeli broj kojim se određuje broj značajke kojemu ulazna datoteka pripada. Broj značajke služi kako bi se stvorio C kod koji se jednostavno može kopirati u C datoteku s kodom u projekt [P1]. Na primjer, broj značajke glasa „a“ je nula pošto je „a“ prvo slovo u abecedi. Broj značajke glasa „m“ 18 pošto je „m“ devetnaesto slovo u abecedi. Razlog tome što su svi brojevi značajki manji za jedan od mjesta u abecedi glasa i slova kojeg predstavljaju indeksi u C programskom jeziku koji počinju od nule, a ne od jedan.

U projektu [P1] funkcija za definiranje značajki pojedinih glasova je `void set_features_for_recognition(struct features* f)` deklarirana u datoteci C zaglavlja *voice_features.h,* a definirana u datoteci *voice_features.c*. Ulazni parametar predstavlja niz struktura tipa `features` u kojemu se pohranjuju skupovi usrednjenih značajki signala.

Srednja vrijednost $\overline{a_p}$ svakog od $p = 12$ LPC koeficijenata se računa tako da se izračuna zbroj svakog pojedinog koeficijenta s određenim indeksom, te se dobiveni zbroj podijeli s brojem koeficijenata koji su zbrojeni:

$$\overline{a_p} = \frac{\sum_{m=1}^{M} a_{m,p}}{M}. \tag{4-47}$$

Broj okvira koji je korišten prilikom stvaranja značajki $M = 299$. Na isti način se računaju i ostale značajke, usrednjena kratkotrajna energija signala, usrednjeni broj prelazaka kroz nulu te usrednjeni period impulsa:

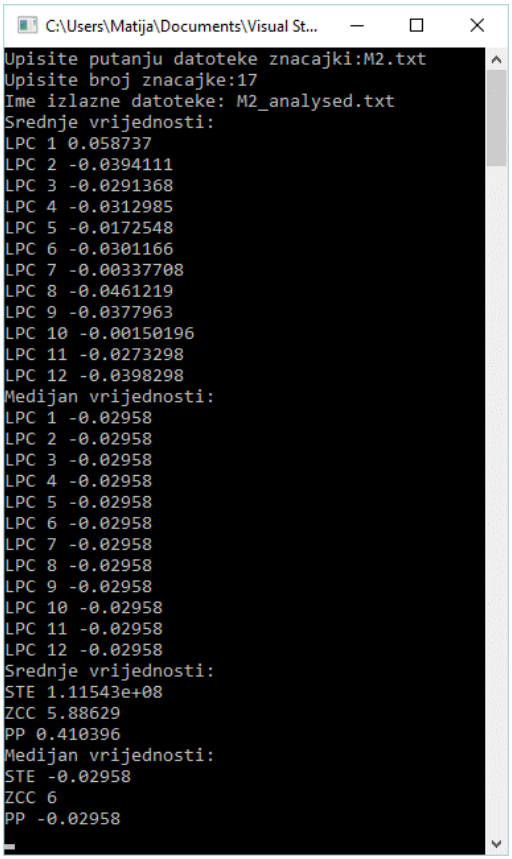
$$\overline{STE} = \frac{\sum_{m=1}^{M} STE_m}{M}, \tag{4-48}$$

$$\overline{ZCC} = \frac{\sum_{m=1}^{M} ZCC_m}{M}, \tag{4-49}$$

$$\overline{PP} = \frac{\sum_{m=1}^{M} PP_m}{M}. \tag{4-50}$$

Osim izračuna srednjih vrijednosti, implementiran je izračun medijana za svaku značajku. Međutim, u implementaciji [P1] su korištene isključivo usrednjene značajke.

Projekt za stvaranje usrednjenih značajki se nalazi se nalazi u prilogu [P3]. Na slici 4.9 se nalazi prozor programa s ispisom značajki za glas „m“. Ispis u prozoru se razlikuje od ispisa u datoteku.



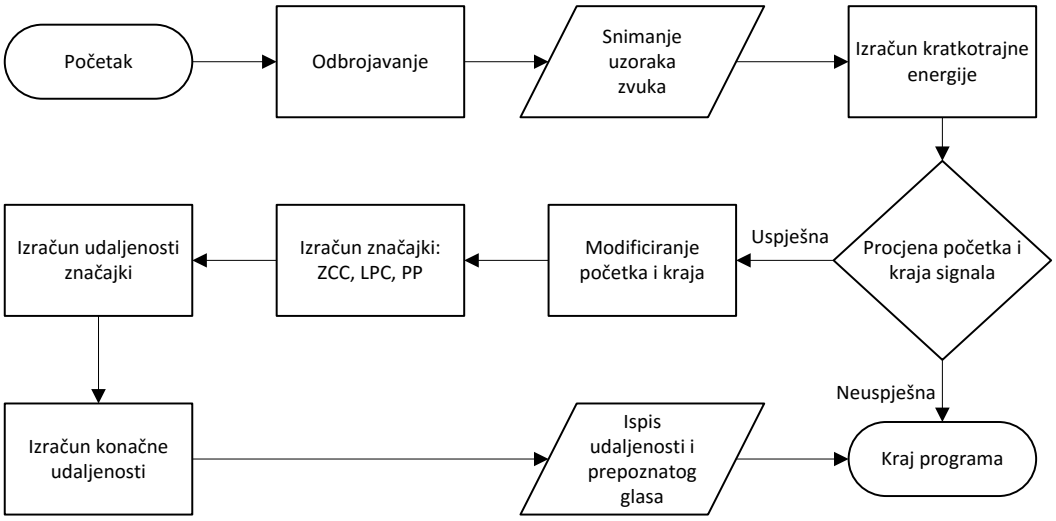
Slika 4.9: Prozor projekta za stvaranje usrednjenih značajki glasova

4.5. Sustav za prepoznavanje glasova hrvatskog jezika

Sustav za prepoznavanje glasova hrvatskog jezika je implementiran kao ugrađeni projekt shematskog projekta koji je stvoren u poglavlju tri za Altium NanoBoard 3000 ploču. Projekt se nalazi u [P1]. Ulaz u sustav čini analogni audio signal koji se dovodi preko 3.5 milimetarskog audio ulaza ploče. Izlaz sustava čini prozor terminala u Altium Designer-u. Sustav služi za prepoznavanje glasova hrvatskog jezika. Glasovi hrvatskog jezika su reproducirani na prijenosnom računalu ASUS X750LB. Snimci glasova hrvatskog jezika su snimljeni programom Audacity te predstavljaju glasove jedne muške osobe starosti 24 godine.

4.5.1. Algoritam sustava za prepoznavanje glasova

Sustav za prepoznavanje glasova započinje snimanjem određenog glasa, a završava ispisom prepoznatog glasa i proračunatim udaljenostima od svih glasova u sustavu. Sustav za prepoznavanje glasova koristi iste metode za stvaranje značajki govornog signala koje koristi sustav za stvaranje značajki govora. Osnovni parametri sustava za prepoznavanje glasova su isti kao i u sustavu za stvaranje značajki glasovnog signala. Algoritam sustava za prepoznavanje glasova je prikazan slikom 4.10.



Slika 4.10: Blok dijagram sustava za prepoznavanje glasova

Algoritam za prepoznavanje glasova počinje odbrojavanjem prema slici 4.102.1. Pošto se signal glasova koje treba prepoznati reproducira s računala, odbrojavanje omogućuje odgovarajuću pripremu korisnika računala za puštanje audio zapisa. Tijekom odbrojavanja, provodi se snimanje zvuka, ali se on na kraju ne sprema u memoriju. Snimanje zvuka tijekom odbrojavanja omogućuje otklanjanje istosmjerne komponente koja se očitava u uzorcima signala, ako se uzorci signala uzimaju odmah nakon inicijalizacija audio jedinice.

Nakon odbrojavanja snimaju se uzorci signala s audio ulaza. Koristi se mono kanal pri frekvenciji uzorkovanja od 16 kHz. Širina jednog uzorka je 16 bita od čega je jedan bit predznak. Za pohranu uzoraka u C kodu se koristi tip `int16_t`. Odjednom se snima 80 000 uzoraka, što odgovara trajanju od 5 sekundi. Prilikom snimanja uzoraka, zvuk se reproducira na audio izlazu, odnosno na zvučnicima razvojne ploče kako bi korisnik čuo kakav signal je snimljen.

Nakon snimanja zvuka, proračunava se kratkotrajna energija signala na svakom okviru i među okviru snimljenog signala, na isti način kao i u sustavu za stvaranje značajki govora. Na temelju izračunate kratkotrajne energije signala provodi se algoritam za detekciju početka i kraja signala. Algoritam traži indekse okvira signala u kojima snimljeni glas počinje i u kojima snimljeni glas završava. Indeks okvira u kojemu snimljeni glas započinje je prvi okvir signala u kojemu iznos kratkotrajne energije signala prelazi predodređenu kritičnu razinu. Indeks okvira u kojemu snimljeni glas završava je prvi indeks poslije određenog indeksa početka u kojemu iznos kratkotrajne energije signala pada ispod predodređene kritične razine. Nakon dodjele indeksa provjerava se da li su indeksi signala pravilno određeni. Indeksi mogu biti nepravilno određeni u slučajevima kada nije pušten nikakav signal prilikom snimanja, kada postoji snimljen glas na posljednjem okviru signala, kada je razlika između indeksa kraja i početka premalog iznosa te kada je pušteni signal pretih. U nabrojanim slučajevima će se ispisati poruka da signal nije dobro detektiran i program će završiti s izvođenjem.

Za svaki i, od 0 do N: //Za svaki okvir snimljenog signala.
  Izračunaj STE_i prema (4-4)
  Ako $STE_i > STE_{min}$ **i ako** $i_{start} = 0$ **i ako** $i_{end} = 0$:
    $i_{start} = i$
  Inače ako $STE_i < STE_{max}$ **i ako** $i_{start} \neq 0$ **i ako** $i_{end} = 0$:
    $i_{end} = i$
Ako $i_{start} = 0$ **i** $i_{end} = 0$ **i** $i_{end} - i_{start} < i_{min}$:
  Ispiši "Neuspješna detekcija."
  Izađi

Algoritam 4.2: Algoritam za traženje početnog i završnog okvira signala

Nakon uspješne detekcije ulaznog signala, uhvaćeni signal se reproducira na audio izlazu radi omogućavanja slušne provjere korisniku. Reproducira se zvuk koji je detektiran algoritmom za prepoznavanje početka i kraja signala.

Slijedi proračun modificiranih indeksa početka i kraja signala. Modificirani indeks početka signala je indeks koji ima višu vrijednost za određeni koeficijent (postotak) od prvotno proračunatog početnog indeksa signala u odnosu na razliku između početnog i krajnjeg indeksa. Isto tako, modificirani završni indeks signala je indeks signala koji ima vrijednost umanjenu za određeni

koeficijent od prvotno proračunatog indeksa u odnosu na razliku između početnog i krajnjeg indeksa. Proračun modificiranih indeksa je potreban kako se usporedba značajki signala ne bi provodila na cijeloj vremenskoj osi snimljenog signala, nego samo na određenom postotku signala koji se nalazi u središtu. Usporedbom značajki koje se nalaze u središtu signala izbjegava se utjecaj značajki na rubovima signala koje mogu odudarati od značajki signala u sredini te time više pridonositi krivoj procjeni prepoznatog glasa. Modificirani indeksi početka i kraja se računaju prema sljedećim jednadžbama:

$$i_{mod_start} = (i_{end} - i_{start}) \cdot K_{start} + i_{start}, \tag{4-51}$$

$$i_{mod_end} = (i_{end} - i_{start}) \cdot K_{end} - i_{end}, \tag{4-52}$$

gdje je K_{start} koeficijent modificiranog početka iznosa $0{,}2$, a K_{end} koeficijent modificiranog kraja iznosa $0{,}2$.

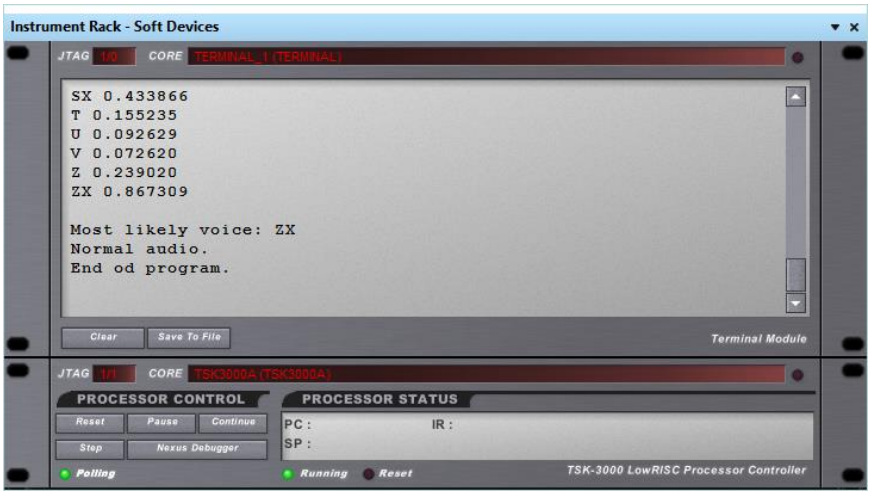
Proračun značajki signala se ne razlikuje od proračuna značajki koji je opisan u sustavu za stvaranje značajki signala, prikazan slikom 4.7. U sustavu za prepoznavanje glasova LPC koeficijenti, broj prelazaka kroz nulu i period impulsa računaju se isključivo za okvire govornog signala između modificiranog početnog indeksa i modificiranog završnog indeksa. Stvaranjem svake pojedine značajke provodi se istovremeno stvaranje udaljenosti značajke sa svakom usrednjenom značajkom u sustavu. Jednom izračunatim LPC koeficijentima za određeni okvir se za svaki postojeći glas stvara udaljenost koristeći normaliziranu križnu korelaciju. Isto tako broju prelazaka kroz nulu i periodu impulsa se za svaki postojeći glas računaju udaljenosti koristeći funkciju relativne udaljenosti. U projektu [P1] se koristi ukupno 30 glasova, stoga za svaki okvir postoji ukupno 30 vrijednosti usporedbi za svaku značajku, odnosno 90 vrijednosti usporedbi pošto se uspoređuju tri značajke.

Pošto postoje udaljenosti za svaku predviđenu značajku za svaki okvir koji je bio namijenjen ispitivanju, potrebno je konačno izračunati ukupnu udaljenost za svaki glas u sustavu. Za svaki glas koji postoji u sustavu dodijeljene su težine pojedinih značajki. Težine su vrijednosti koje određuju s kolikim postotkom udaljenost određene značajke određenog glasa sudjeluje u određivanju konačne udaljenosti glasa snimljenog signala. Pošto se usporedba provodi za udaljenosti LPC koeficijenata, udaljenosti broja prelazaka kroz nulu i udaljenosti perioda impulsa, u sustavu su dodijeljene težine za svaku od nabrojanih značajki. To znači da za svaki glas u sustavu postoje tri težine koje ukupno zbrojene čine broj jedan ili 100% izraženo u postotcima. Težine za pojedine glasove su dodijeljene ovisno o tome da li je glas zvučan ili bezvučan. Za zvučne glasove težine su dodijeljene na slijedeći način: $LPC_w = 0.8$, $PP_w = 0.1$ i $ZCC_w = 0.1$. Za bezvučne

glasove težine glase: $LPC_w = 0.8$, $PP_w = 0.0$ i $ZCC_w = 0.2$. Najveća težina je uvijek dodijeljena LPC koeficijentima. Za bezvučne glasove, procjena perioda impulsa nema smisla tako da ona nije korištena u izračunu, odnosno težina perioda impulsa za bezvučne glasove iznosi 0. Konačna udaljenost pojedinog glasa računa se prema sljedećoj formuli:

$$D_g = \frac{\sum_{i=1}^{N_m}(LPC_{w,g} \cdot LPC_D + PP_{w,g} \cdot PP_D + ZCC_{w,g} \cdot ZCC_D)}{N_m}, \quad (4\text{-}53)$$

gdje je $N_m = i_{mod_end} - i_{mod_end}$ – broj okvira nad kojima se računaju značajke glasova. U sustavu postoji 30 glasova, te se konačna udaljenost računa za svaki glas. Na kraju se ispisuje glas koji ima najveću udaljenost kao i popis konačnih udaljenosti od svih glasova. Na slici 4.11 je prikazan ispis konačnih udaljenosti glasova kao i prepoznat glas.



Slika 4.11: Prozor terminala u kojemu su ispisane udaljenosti pojedinih glasova i prepoznat glas

5. EKSPERIMENTALNI REZULTATI

U ovom poglavlju se nalaze rezultati provedenog testiranja na sustavu [P1]. Izvor rezultata testa 1 i testa 2 koja se nalaze u ovom poglavlju je Excel dokument dan u prilogu [P5]. Navedena tablica sadrži testiranje svakog snimka glasa u koji se nalazi u sustavu. Testni snimci se nalaze u prilogu [P4]. Provedena su dva testa za mjerenje točnosti sustava [P1]: test 1 i test 2, i dva testa za mjerenje vremena izvođenja proračuna nad snimljenim glasom u sustavu [P1]: test 3 i test 4.

5.1. Test 1

U ovom potpoglavlju se nalaze rezultati eksperimenta u kojemu su testirani izvorni glasovi. Izvorni glasovi su snimci glasova koji su korišteni za stvaranje značajki govora sustavom u prilogu [P2 i [P3]. Tablicom 5.1 su prikazani eksperimentalni rezultati jednog testiranja za svaki izvorni snimak glasa. U prvom stupcu tablice se nalazi naziv snimka koji je korišten u testiranju. U drugom stupcu se nalazi oznaka glasa koji je prepoznat prilikom provođenja eksperimenta. U trećem stupcu je prikazana udaljenost koja je dobivena prilikom postupka prepoznavanja od prepoznatog glasa. Četvrti stupac sadrži udaljenosti za slučaj da nije prepoznat glas koji je predviđen.

Prilikom prepoznavanja izvornih glasova, sustav za prepoznavanje nije uspješno prepoznao samo dva glasa: glas „M“ je prepoznao kao glas „V“, a glas „N“ je prepoznao kao glas „Lj“, što vidimo u tablici 5.1. Glas „M“ je po akustičkim svojstvima sličan glasu „V“, a oba glasa prilikom izgovora su slični muklom glasu šva „ə“, što vrijedi i za glasove „N“ i „Lj“. Razlika udaljenosti između glasa „M“ i „V“ iznosi 0,037533, a razlika udaljenosti između glasa „N“ i „Lj“ iznosi 0,063989. Obje razlike udaljenosti su relativno malih iznosa. Prema rezultatima testiranja izvornih glasova točnost sustava je 93,3%.

Naziv snimka	Prepoznati glas	Udaljenosti od prepoznatog glasa	Udaljenost od točnog glasa
A_kontinuirano (izvorni)	A	0,869583	-
B_isprekidano (izvorni)	B	0,693007	-
C_kontinuirano (izvorni)	C	0,933246	-
Č_isprekidano (izvorni)	Č	0,778218	-
Ć_kontinuirano (izvorni)	Ć	0,973981	-
D_kontinuirano (izvorni)	D	0,679232	-
Dž_kontinuirano (izvorni)	Dž	0,796052	-
Đ_kontinuirano (izvorni)	Đ	0,796052	-
E_kontinuirano (izvorni)	E	0,900051	-
F_kontinuirano (izvorni)	F	0,902883	-
G_isprekidano (izvorni)	G	0,755293	-
H_kontinuirano (izvorni)	H	0,823417	-
I_kontinuirano (izvorni)	I	0,928972	-
J_kontinuirano (izvorni)	J	0,858679	-
K_isprekidano (izvorni)	K	0,615025	-
L_kontinuirano (izvorni)	L	0,815665	-
Lj_kontinuirano (izvorni)	Lj	0,722247	-
M_kontinuirano (izvorni)	V	0,711404	0,673871
N_kontinuirano (izvorni)	Lj	0,707564	0,643575
Nj_kontinuirano (izvorni)	Nj	0,741074	-
O_kontinuirano (izvorni)	O	0,849029	-
P_isprekidano (izvorni)	P	0,506867	-
R_kontinuirano (izvorni)	R	0,799522	-
S_kontinuirano (izvorni)	S	0,955247	-
Š_kontinuirano (izvorni)	Š	0,940720	-
T_isprekidano (izvorni)	T	0,811534	-
U_kontinuirano (izvorni)	U	0,832391	-
V_kontinuirano (izvorni)	V	0,807615	-
Z_kontinuirano (izvorni)	Z	0,889286	-
Ž_kontinuirano (izvorni)	Ž	0,883799	-

Tablica 5.1 Prikaz testiranja za izvorne zvukove

Prema tablici 5.1, ukupni prosjek udaljenosti ispravno prepoznatih glasova iznosi 0,8059. Izračunate su prosječne udaljenosti za pojedine skupine glasova prikazane tablicom 5.2.

Skupina glasova	Prosjek udaljenosti skupine glasova
Samoglasnici	0,8726
Sonanti	0,7578
Zvučni šumnici	0,8981
Bezvučni šumnici	0,7300

Tablica 5.2 Prikaz prosječnih udaljenosti za pojedine skupine glasova

5.2. Test 2

U prethodnom potpoglavlju je opisan eksperiment koji je proveden na izvornim snimcima. U ovom potpoglavlju se nalazi pregled rezultata svih snimaka kojih se nalaze u prilogu [P5]. Kompletni rezultat mjerenja se nalazi u prilogu u Excel dokumentu [P5].

U prilogu [P4] se nalazi pet snimaka za svaki glas u sustavu. Prvi snimak određenog glasa čini izvorni snimak – snimak po kojemu je napravljena usrednjeni skup značajki određenog glasa. Mjerenja prvog snimka su istovjetna mjerenjima u prethodnom potpoglavlju – testu 1. Drugi snimak određenog glasa čini snimak koji je izgovoren kontinuirano ili isprekidano bez razmaka, po istom principu kojemu su snimani izvorni snimci. Preostala tri snimka su izgovorena kao glas izgovoren jednom tijekom snimanja. U tablici u prilogu [P5] uz naziv snimka nalazi način na koji je snimak prilikom snimanja izgovoren.

5.2.1. Rezultati prepoznavanja pojedinih glasova

Tablica 5.3 prikazuje postotak prepoznavanja testa 2. Sustav je ukupno 44% snimaka glasova prepoznao ispravno.

Glas	**A**	**B**	**C**	**Č**	**Ć**	**D**	**Dž**	**Đ**	**E**	**F**
Prepoznato	80%	40%	100%	20%	100%	20%	20%	20%	40%	40%
Glas	**G**	**H**	**I**	**J**	**K**	**L**	**Lj**	**M**	**N**	**Nj**
Prepoznato	80%	40%	80%	40%	80%	20%	20%	0%	20%	20%
Glas	**O**	**P**	**R**	**S**	**Š**	**T**	**U**	**V**	**Z**	**Ž**
Prepoznato	40%	20%	40%	20%	100%	60%	40%	40%	40%	40%

Tablica 5.3 Postotak točno prepoznatih testova za pojedini glas

Ako se rezultati iz [P5] pogledaju podrobnije može se uočiti određena logika u loše prepoznatim glasovima. Prva stvar koja se može uočiti da je većina loše prepoznatih zvučnih glasova je prepoznata kao nekakav drugi zvučni glas. Većina loše prepoznatih bezvučnih glasova je prepoznata kao drugi bezvučni glas.

Samoglasnik „E“ je dvaput zamijenjen s glasom „J“ i jednom s glasom „I“. Navedeni glasovi su bliski po akustičnim svojstvima te je ovakav rezultat logičan. Samoglasnik „O“ je dvaput zamijenjen s glasom „Nj“, te jednom s glasom „V“. Iako „Nj“ nije toliko slično po akustičkim svojstvima s glasom „O“, sličnost postoji u LPC analizi. Glas „Nj“ je sličan glasu „O“ prema LPC analizi pošto je umjesto glasa „Nj“ četiri puta prepoznat glas „O“. Ova sličnost se djelomice može pripisati sličnosti glasu šva. Glas „U“ je jednom prepoznat kao „O“, jednom kao „R“ te jednom kao „Nj“. Glasovi „O“ i „U“ su slični po akustičkim svojstvima.

Samoglasnici koji pripadaju sonantima prilikom neispravnog prepoznavanja su najčešće prepoznati kao drugi zvučni glasovi uključujući i samoglasnike. Najčešći razlog lošeg rezultata ovih glasova je njihova zvučnost koja ih približuje glasu šva. Raštrkane rezultate pokazuju rezultati testiranja glasova: „L“, „Lj“, „M“, „N“. Prilikom prepoznavanje ovih glasova najčešće je prepoznat drugi sonant, a ponekad samoglasnik ili zvučni šumnik. Ostali sonanti pokazuju točnost od 40%, a prilikom neispravnog prepoznavanja su zamijenjeni akustički bliskim glasovima.

Među glasovima zvučnih šumnika, „D“, „Dž“ i „Đ“ imaju točnost prepoznavanja 20%. Navedeni glasovi su podjednako krivo prepoznati kao drugi zvučni šumnici, samoglasnici i sonanti. Glasovi „Z“ je dvaput prepoznat kao glas „A“, što vrijedi i za slučaj glasa „Ž“. Razlog tome je sličnost s glasom „šva“. Glas „B“ je jednom prepoznat kao glas „D“, „G“ i „A“, pri čemu prva dva glasa pripadaju istoj skupini zvučnih šumnika. Glas „G“ je samo jednom prepoznat kao glas „Nj“.

Skupina bešumnih glasova pokazuje relativno dobre rezultate. Glasovi „C“, „Ć“, „Š“ su prepoznati u svim slučajevima točno. Glas „Č“ je četiri puta prepoznat kao glas „Š“, što nije začuđujuće pošto glas „Š“ možemo smatrati kontinuiranim izgovorom glasa „Č“. Glas „S“ je zbog istog razloga prepoznat tri puta kao glas „C“, a jednom je prepoznat kao glas „K“. Glas „T“ je u neispravnim slučajevima prepoznavanja prepoznat kao glas „C“. Glas „K“ je samo u jednom slučaju prepoznat kao glas „Ć“. Glas „F“ je dvaput prepoznat kao glas „T“, a jednom kao glas „Č“. Svi dosad navedeni primjeri neispravnog prepoznavanja ulaze u istu skupinu glasova prema akustičkim svojstvima. Glas „H“ je tri puta neispravno prepoznat kao jedan od zvučnih glasova, a glas „P“ dva puta. Razlozi zbog čega su posljednja dva glasa loše prepoznata je mala amplituda glasa u odnosu na ostale, što otežava stvaranje LPC značajki.

5.2.2. Izmjerene udaljenosti pojedinih glasova

Prosječna udaljenost od točnog glasa u testu 2 iznosi 0,56562, što je manji iznos od udaljenosti u testu 1. Manje udaljenosti ukazuju na varijabilnost značajki govornog signala koje su izračunate za različite varijante izrečenog glasa. U tablici 5.4 se nalaze prosječne udaljenosti od ispravnog glasa koje su izmjerene prilikom testa 2.

Glas	A	B	C	Č	Ć
Udaljenost	0,6342	0,5554	0,7977	0,7434	0,8202
Glas	D	Dž	Đ	E	F
Udaljenost	0,4365	0,3821	0,3937	0,6179	0,5703
Glas	G	H	I	J	K
Udaljenost	0,5223	0,5385	0,7350	0,5833	0,5101
Glas	L	Lj	M	N	Nj
Udaljenost	0,4991	0,4595	0,5318	0,5302	0,4780
Glas	O	P	R	S	Š
Udaljenost	0,6303	0,1582	0,5890	0,5611	0,8073
Glas	T	U	V	Z	Ž
Udaljenost	0,6266	0,6689	0,6224	0,3822	0,5835

Tablica 5.4: Prosječne udaljenosti od točnog glasa za svaki pojedini glas

Prosječna apsolutna razlika između udaljenosti točnog glasa i prepoznatog glasa za slučaj da glas nije dobro prepoznat iznosi: 0,11259. Ovaj broj je u odnosu na udaljenosti iz tablice 5.4 relativno malen. Ako bi na ovakvoj mjeri udaljenosti željeli konstruirati sustav za prepoznavanje izoliranih riječi, prilikom pretraživanja pojedinih fonema na određenim mjestima u nizu značajki snimljenog signala bilo bi moguće pronaći dovoljno blisku značajku. U tablici 5.5 su prikazane prosječne apsolutne razlike udaljenosti između prepoznatih glasova i točnih glasova iz [P5]. Tablica 5.5 može poslužiti za fino podešavanje određenih parametara sustava.

Glas	A	B	C	Č	Ć
A. r. udaljenosti	0,1847	0,0325	-	0,0337	-
Glas	D	Dž	Đ	E	F
A. r. udaljenosti	0,1339	0,2113	0,2135	0,0704	0,0667
Glas	G	H	I	J	K
A. r. udaljenosti	0,0775	0,1472	0,0482	0,0516	0,2141
Glas	L	Lj	M	N	Nj
A. r. udaljenosti	0,1638	0,1181	0,0904	0,0653	0,1100
Glas	O	P	R	S	Š
A. r. udaljenosti	0,0519	0,2568	0,0344	0,1072	-
Glas	T	U	V	Z	Ž
A. r. udaljenosti	0,0166	0,0478	0,0337	0,2787	0,1079

Tablica 5.5: Prosječna apsolutna razlika udaljenosti između prepoznatog glasa i točnog glasa za pojedini glas

5.3. Test 3

Radi ispitivanja mogućnosti prepoznavanja govora u realnom vremenu provedeno je vremensko testiranje sustava za prepoznavanje glasova hrvatskog jezika [P1]. U sustavu [P1] su na određenim mjestima naredbe koje mjere proteklo vrijeme od početka pokretanja sustava. Naredba `uint64_t clock_ms()` iz zaglavlja *timing.h* vraća vrijeme od početka izvođenja programa izraženo u milisekundama. Vrijeme od početka izvođenja je mjereno dva puta prilikom izvođenja programa: nakon snimljenih uzoraka zvučnog signala i nakon provedenog proračuna udaljenosti za sve glasove. Razlika ta dva vremena čini ukupno vrijeme koje je potrebno sustavu da prepozna glas.

U tablici 5.6 se nalazi prikaz vremenskog testiranja sustava [P1].

	Trajanje glasa (ms)	Broj okvira	Modificiran broj okvira	Vrijeme proračuna (ms)
1.	269	43	26	5329
2.	319	51	31	6104
3.	394	63	38	7210
4.	400	64	39	7260
5.	513	82	50	9120
6.	538	86	52	9564
7.	575	92	56	10064
8.	656	105	63	11077
9.	669	107	64	11574
10.	731	117	70	12545
11.	994	159	95	16659
12.	1019	163	98	17354
13.	1056	169	102	17995
14.	1069	171	103	18006
15.	1088	174	105	18414
16.	1131	181	109	19063
17.	1138	182	110	19280
18.	1156	185	111	19458
19.	1163	186	112	19587
20.	1206	193	116	20052
21.	1425	228	137	23525
Prosjek	834	133	80	14250

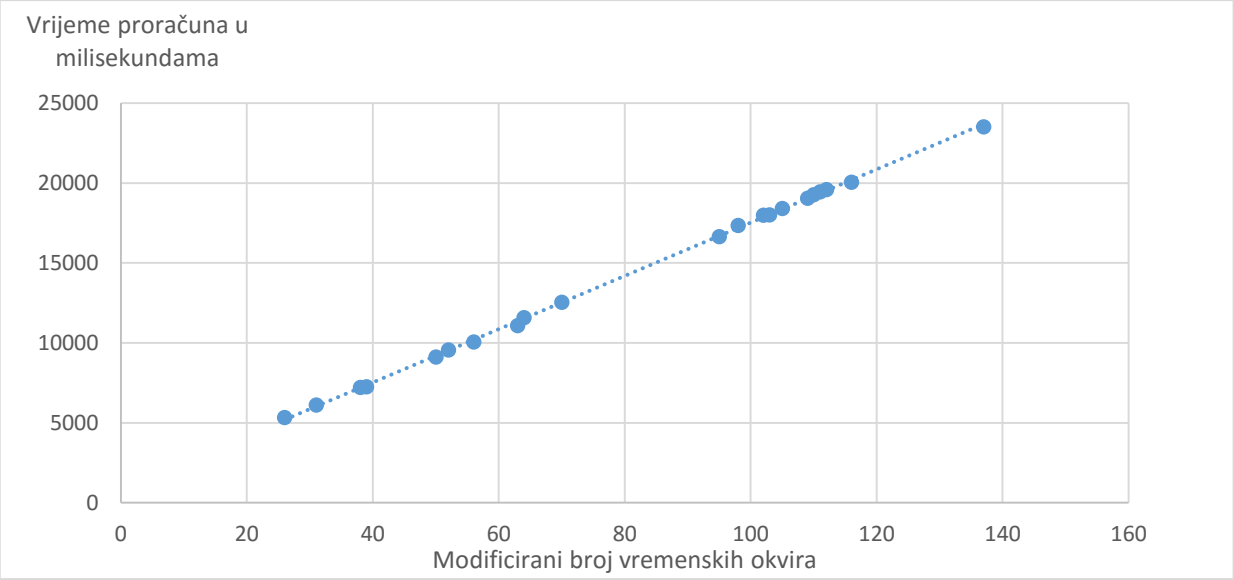
Tablica 5.6: Vremenski test glasova sustava [P1]

Svaki redak tablice predstavlja testiranje neodređenog snimka iz [P5]. Drugi stupac prikazuje procijenjeno vrijeme trajanja testiranog snimka glasa u milisekundama. Treći stupac prikazuje broj okvira koji je određen prilikom procjene početka i kraja glasovnog signala. Četvrti stupac prikazuje modificirani broj okvira na kojima se vrši proračun značajki. Peti stupac prikazuje

vrijeme u milisekundama koje je potrebno da se proračun izvrši, odnosno razliku vremena između dvije točke u toku programa.

U posljednjem redu tablice 5.6 prikazane su prosječne vrijednosti pojedinih stupaca tablice. Prosječno vrijeme koje je potrebno kako bi sustav izračunao najizgledniji glas je 14 250 milisekundi, odnosno 14,25 sekundi.

Slika 5.1 prikazuje graf koji pokazuje koliko je vremena je potrebno kako bi se dobio proračuna najizglednijeg glasa u sustav [P1] u odnosu na broj okvira nad kojima se proračunavaju značajke signala. Na grafu je vidljiv linearan trend – što je veći broj okvira iz kojih se proračunavaju značajke signala, to je veće vrijeme proračuna najizglednijeg glasa. Graf se temelji na tablici 5.6.



Slika 5.1: Vrijeme proračuna najizglednijeg glasa u odnosu na broj vremenskih okvira

5.4. Test 4

Radi saznanja mogućnosti prepoznavanja govora sa zahtjevom proračuna u realnom vremenu provedeno je vremensko testiranje sustava [P1] s snimcima riječi, za razliku od prethodnih testova gdje su testirani glasovi. Programom Audacity snimljene su određene riječi različitih dužina koje se nalaze u prilogu [P6]. Sustav [P1] je podešen tako da ne modificira procijenjeni početak i kraj signala. Vrijeme se u programu mjeri na dva mjesta: nakon snimljenih uzoraka zvučnog signala, te nakon proračuna konačnih udaljenosti, na isti način kao u testu 3. Razlika izmjerenih vremena predstavlja vrijeme koje je potrebno za konačni proračun na temelju kojih su napravljeni zaključci.

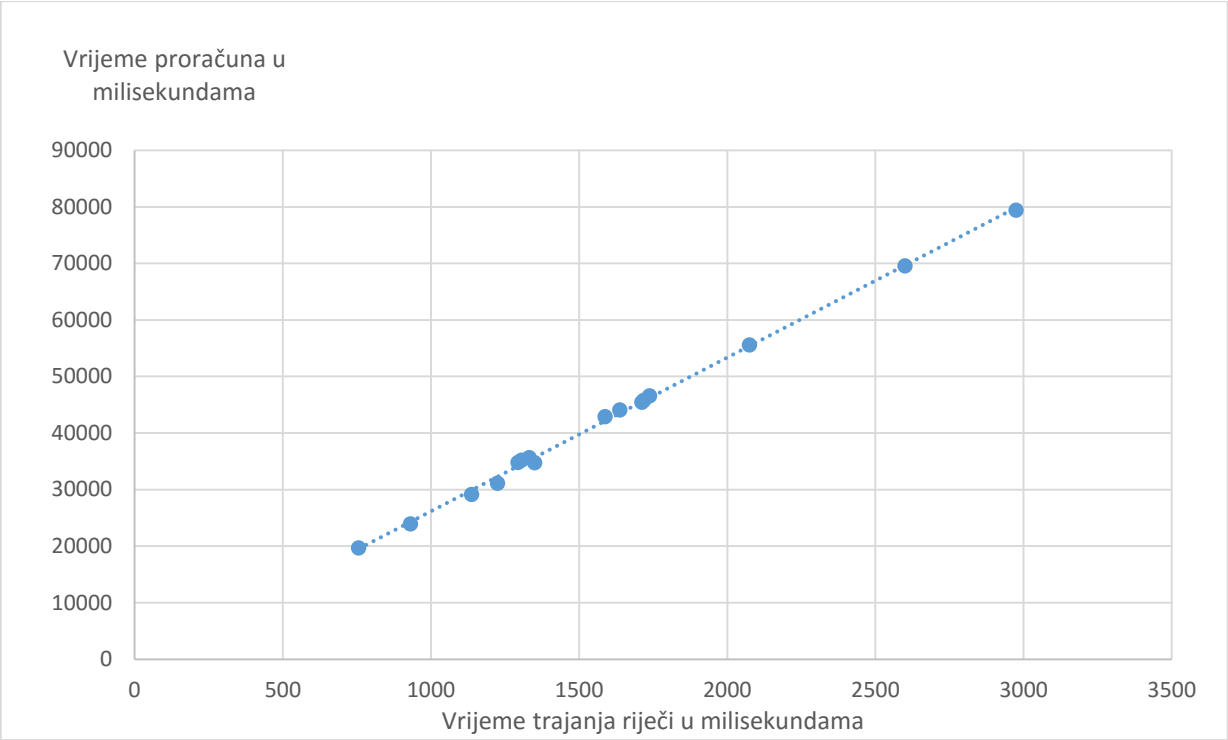
Tablica 5.7 predstavlja test vremena proveden na snimkama riječi u prilogu [P6]. Prvi stupac sadrži riječi koje su korištene prilikom testiranja. Drugi stupac sadrži procijenjeno vremensko trajanje riječi izraženo u milisekundama. Treći stupac sadrži broj okvira nad kojima se provodi proračun značajki signala. Četvrti stupac predstavlja vrijeme izraženo u milisekundama koje je potrebno za konačan proračun, odnosno razliku vremena u određenim točkama toka programa.

Riječ	Trajanje riječi (ms)	Broj okvira	Vrijeme (ms)
"Ne"	756	121	19685
"Da"	931	149	23909
"Tri"	1138	182	29159
"Dva"	1225	196	31119
"Sedam"	1350	216	34760
"Četiri"	1294	207	34821
"Jedan"	1306	209	35165
"Nula"	1331	213	35648
"Programski"	1588	254	42877
"Ugrađeni"	1638	262	44088
"Početak"	1713	274	45444
"Rekurzija"	1719	275	45796
"Koeficijent"	1738	278	46583
"Analizirati"	2075	332	55547
"Linearno-prediktivna"	2600	416	69535
"Otorinolarignologija"	2975	476	79424
Prosjek	1586	254	42098

Tablica 5.7: Mjerenje vremena proračuna za slučaj riječi iz [P6]

Riječi u tablici 5.7 su poredane prema procijenjenoj duljini trajanja. U posljednjem redu je izračunat prosjek svih stupaca tablice. Prosjek vremenskog trajanja riječi iznosi 1 586 milisekundi, odnosno 1,6 sekundi. Za jednu riječ prosječnog vremenskog trajanja je potreban proračun od 42 098 milisekundi, odnosno 42 sekunde.

Podatci tablice 5.7 su prikazani grafom na slici 5.2. Graf prikazuje ovisnost vremenskog trajanja signala riječi u odnosu na vrijeme trajanja proračuna značajki snimljenog signala riječi. Oba trajanja su izražena u milisekundama. Na grafu je naznačen linearan trend.



Slika 5.2: Vrijeme proračuna najizglednijeg glasa u odnosu na broj vremenskih okvira

5.5. Zaključak eksperimenta

Na temelju rezultata testa 1 može se zaključiti kako je dizajn sustava za prepoznavanje govora moguć s predloženom implementacijom. Sustav [P1] pokazuje točnost prepoznavanja u 93% slučajeva za snimke glasova po kojima su snimljene karakteristike. To je dovoljna točnost za implementaciju algoritma više hijerarhije koji bi omogućio prepoznavanje, primjerice, izolirane riječi.

Rezultati testa 2 ukazuju na točnost sustava od 44%. Slabiji rezultati testa 2 ukazuju na određene probleme metodologije sustava za prepoznavanje glasova [P1]. Sustav nije predviđen kako bi modelirao fonemsku koartikulaciju – promjenu značajki govornog signala u vremenu. Za uspješno prepoznavanje fonema potreban je određeni algoritam koji može uspješno modelirati fonem. Modeliranje jezičnih jedinica je opisano u poglavlju 2. Moguće je fino podesiti sustav, mijenjajući određene značajke te stvarati usrednjene značajke na reprezentativnijim snimcima glasova. Međutim, najveći problem ipak ostaje u nedostatnom modeliranja fonemske koartikulacije.

Test 3 i 4 služe za procjenu vremenskog trajanja proračuna koji je potreban sustavu [P1] da bi odabrao najizgledniji glas. Iz provedenih testova se zaključuje da je nemoguće ostvariti prepoznavanje govora u realnom vremenu za trenutni sustav, pošto je za prosječnu riječ proračun značajki u prosjeku traje 42 sekunde. Međutim, zahtjev za proračunom u realnom vremenu je vrlo

vjerojatno ostvariv. Postoji mogućnost implementacije određenih funkcionalnosti u kodu stvaranjem izdvojenih komponenti u FPGA projektu. Izdvojene komponente bi izvodile određene proračune u mnogo kraćem vremenu u odnosu na klasično softversko računanje. U [P1] bi bilo poželjno napraviti komponente koje omogućuju stvaranje značajki govornog signala. Druga opcija koja se pruža je korištenje više procesora u paraleli, što iziskuje poznavanje rada s višeprocesorskim sustavima. Sustav [P1] intenzivno koristi operacije s tipovima s pomičnim zarezom, te se nameće i mogućnost prebacivanja operacija tipova s pomičnim zarezom na zasebnu hardversku jedinicu, pošto TSK3000A nije optimiziran za efikasan rad s ovakvim tipom podataka.

6. ZAKLJUČAK

Cilj ovog diplomskog rada je bio istražiti mogućnost prepoznavanja govora u stvarnom vremenu na Altium razvojnom sustavu. Kako se mogućnost pokazala izvedivom, razvijen je jednostavan sustav za prepoznavanje glasova.

Opisana je povijest razvitka sustava za prepoznavanje govora, s navedenim primjerima istraživanja i uspješnih komercijalnih sustava. Sustavi za prepoznavanje govora sve više uspješno zamjenjuju druge metode unosa podataka u računalne sustave, što je glavni razlog razvoja ovakvih sustava. Obrađeni su osnovni algoritmi koji se koriste za prepoznavanje govora, koji su svrstani u dvije kategorije: algoritme za stvaranje značajki signala govora i algoritme odabira jezičnih jedinica. Današnji sustavi dolaze do granica modela na kojima su temeljeni stoga se današnji razvoj temelji na istraživanjima u području umjetne inteligencije.

U nastavku je obrađen FPGA razvojni sustav korišten za implementaciju sustava za prepoznavanje glasova. Korištena je razvojna ploča Altium NanoBoard 3000 pomoću programa Altium Designer. Razvojna ploča je korištena za stvaranje projekta koji emulira ugrađeni računalni sustav s osnovnim elementima koji su potrebni za implementaciju sustava za prepoznavanje glasova. Sustav se temelji na TSK3000A procesoru kojemu je dostupan 1 MB memorije, audio ulaz, te mogućnost ispisa teksta u prozor terminala na računalu.

Implementiran je sustav za prepoznavanje glasova hrvatskog jezika čiji algoritam se bazira na linearno-prediktivnoj analizi. Opisano je stvaranje osnovnih značajki govornog signala koje se koriste uz linearno-prediktivnu analizu: kratkotrajna energija signala, broj prelazaka kroz nulu i period impulsa. Linearno-prediktivna analiza se temelji na predviđanju parametara modela vokalnog trakta. LPC koeficijenti su parametri vokalnog trakta na kojima se temelji linearno-prediktivna analiza. U implementaciji je korištena Levinson-Durbinova rekurzija i algoritam strmog spusta za proračun LPC koeficijenata.

Stvoren je sustav za stvaranje usrednjenih značajki govora koji se sastoji od dva dijela: sustava za stvaranje značajki i sustava za računanje srednjih vrijednosti značajki. Sustav za stvaranje značajki je projekt namijenjen razvojnoj ploči. Služi za ispis značajki snimljenog signala. Nakon ispisa značajki, računaju se srednje vrijednosti značajki preko komandno-linijskog programa. U konačnici su izračunate usrednjene značajke ubačene u sustav za prepoznavanje glasova koji je implementiran na razvojnoj ploči.

Sustav za prepoznavanje glasova je namijenjen prepoznavanju glasova hrvatskog jezika. Sustav je testiran na izvornim snimcima glasova koji pokazuje dobre rezultate, s 93,3% točno prepoznatih glasova. Sustav je testiran na još dodatna četiri snimka osim izvornih glasova. Ukupna točnost sustava nakon provedenog eksperimenta je 44%. Provedeno je vremensko mjerenje sustava kao bi se ispitao zahtjev sustava za prepoznavanjem u realnom vremenu. Sustav u trenutnom stanju ne može ispuniti vremenski zahtjev pošto mu je za stvaranje značajki signala riječi prosječne veličine potrebno 42 sekunde. Uz određene nadogradnje sustav može ispuniti vremenski zahtjev.

LITERATURA

[1] L. R. Rabiner i B. H. Juang, »Automatic Speech Recognition - A Brief History of the Technology Development,« 8. Listopad 2004. [Mrežno]. Available: http://www.ece.ucsb.edu/Faculty/Rabiner/ece259/Reprints/354_LALI-ASRHistory-final-10-8.pdf. [Pokušaj pristupa 20. Lipanj 2016].

[2] M. Pinola, »Speech Recognition Through the Decades: How We Ended Up With Siri - Page 1,« PCWorld, 2. Studeni 2011. [Mrežno]. Available: http://www.pcworld.com/article/243060/speech_recognition_through_the_decades_how_we_ended_up_with_siri.html. [Pokušaj pristupa 20. Lipanj 2016].

[3] L. R. Rabiner, »First-Hand : The Hidden Markov Model,« ETHW, 12. Siječanj 2015. [Mrežno]. Available: http://ethw.org/First-Hand:The_Hidden_Markov_Model. [Pokušaj pristupa 20. Lipanj 2016].

[4] M. Pinola, »Speech Recognition Through the Decades: How We Ended Up With Siri - Page 2,« PCWorld, 2. Studeni 2011. [Mrežno]. Available: http://www.pcworld.com/article/243060/speech_recognition_through_the_decades_how_we_ended_up_with_siri.html?page=2. [Pokušaj pristupa 20. Lipanj 2016].

[5] C. Petersen, »A Guide to Speech Recognition Algorithms (Part 1),« 8. Prosinac 2015. [Mrežno]. Available: https://www.youtube.com/watch?v=i9Gn2QYrYpo. [Pokušaj pristupa 20. Lipanj 2016].

[6] U. Shrawankar i V. Thakare, »Techniques for Feature Extraction in Speech Recognition System: A Comparative Study,« [Mrežno]. Available: https://arxiv.org/ftp/arxiv/papers/1305/1305.1145.pdf. [Pokušaj pristupa 19. Lipanj 2016].

[7] I. Rendulić, Mjere udaljenosti u obradi govornog signala, Zagreb: Sveučilište u Zagrebu, Fakultet elektrotehnike i računarstva, 2011.

[8] D. Brunčić, Istraživanje programske podrške za diktiranje teksta, Osijek: Sveučilište Josipa Jurja Strossmayera u Osijeku, Elektrotehnički fakultet Osijek, 2014.

[9] J. Lyons, »Tutorial - Cepstrum and LPCCs,« 2012. [Mrežno]. Available: http://practicalcryptography.com/miscellaneous/machine-learning/tutorial-cepstrum-and-lpccs/. [Pokušaj pristupa 9. Srpanj 2016].

[10] A. H. Mansour, G. Z. A. Salh i K. A. Mohammed, »Voice Recognition using Dynamic Time Warping and Mel-Frequency Cepstral Coefficients Algorithms,« Travanj 2015. [Mrežno]. Available: http://research.ijcaonline.org/volume116/number2/pxc3902362.pdf. [Pokušaj pristupa 19. Lipanj 2016].

[11] J. Tebelskis, »Speech Recognition using Neural Networks,« Svibanj 1995. [Mrežno]. Available: http://robotics.bstu.by/mwiki/images/0/0f/(Brain_Study)_Speech_Recognition_using_Neural_Networks.pdf. [Pokušaj pristupa 20. Lipanj 2016].

[12] I. P. Wiggers i L. M. Rothkrantz, »Automatic Speech Recognition using Hidden Markov Models,« Rujan 2003. [Mrežno]. Available: http://www.kbs.twi.tudelft.nl/docs/syllabi/speech.pdf. [Pokušaj pristupa 20. Lipanj 2016].

[13] Y. Bengio, »Learning Deep Architectures for AI,« 2009. [Mrežno]. Available: http://sanghv.com/download/soft/machine%20learning,%20artificial%20intelligence,%20mathematics%20ebooks/ML/learning%20deep%20architectures%20for%20AI%20%282009%29.pdf. [Pokušaj pristupa 10. Srpanj 2016].

[14] J. Schmidhuber, »Deep learning in neural networks: An overview,« Siječanj 2015. [Mrežno]. Available: http://www.sciencedirect.com/science/article/pii/S0893608014002135. [Pokušaj pristupa 10. Srpanj 2016].

[15] K. Petersen, »A Guide to Speech Recognition Algorithms (Part 2),« 8. Prosinac 2015. [Mrežno]. Available: https://www.youtube.com/watch?v=49XO1KgfBAQ. [Pokušaj pristupa 10. Lipanj 2016].

[16] S. Hochreiter i J. Schmidhuber, »Long Short-term Memory,« 1997. [Mrežno]. Available: http://deeplearning.cs.cmu.edu/pdfs/Hochreiter97_lstm.pdf. [Pokušaj pristupa 10. Lipanj 2016].

[17] F. A. Gers, N. N. Schraudolph i J. Schmidhuber, »Learning Precise Timing with LSTM Recurrent Networks,« 2002. [Mrežno]. Available: http://www.jmlr.org/papers/volume3/gers02a/gers02a.pdf. [Pokušaj pristupa 10. Lipanj 2016].

[18] T. N. Sainath, A.-r. Mohamed, B. Kingsbury i B. Ramabhadran, »Deep Convolutional Neural Networks for LVCRS,« IBM T. J. Watson Research Center Yorktown Heights, NY 10598, U.S.A.; Department of Computer Science, University of Toronto, Canada, [Mrežno]. Available: http://www.cs.toronto.edu/~asamir/papers/icassp13_cnn.pdf. [Pokušaj pristupa 10. Srpanj 2016].

[19] Xilinx Inc., San Jose, California, U.S., »What is an FPGA?,« Xilinx Inc., 2016. [Mrežno]. Available: http://www.xilinx.com/training/fpga/fpga-field-programmable-gate-array.htm. [Pokušaj pristupa 21. Lipanj 2016].

[20] »About FPGAs,« [Mrežno]. Available: http://home.mit.bme.hu/~szedo/FPGA/fpgahw.htm. [Pokušaj pristupa 10. Srpanj 2016].

[21] Altium Limited, Sydney, Australia, »NanoBoard 3000 - Motherboard Resources,« Altium Limited, 15. Travanj 2014. [Mrežno]. Available: http://techdocs.altium.com/display/HWARE/NanoBoard+3000+-+Motherboard+Resources. [Pokušaj pristupa 20. Lipanj 2016].

[22] Altium Limited, Sydney, Australia, »NB3000XN Reference Design,« [Mrežno]. Available: http://downloads.altium.com/nanoboard/NanoBoard-NB3000XN.zip. [Pokušaj pristupa 21. Lipanj 2016].

[23] Altium Limited, Sydney, Australia, »NB3000XN Schematics,« 2010. [Mrežno]. Available: http://techdocs.altium.com/sites/default/files/wiki_attachments/208010/NanoBoard+3000XN+Schematics+%28Xilinx+variant%29.pdf. [Pokušaj pristupa 21. Lipanj 2016].

[24] Altium Limited, »Functional Overview of the NanoBoard 3000,« Altium Limited, 5. Ožujak 2014. [Mrežno]. Available: http://techdocs.altium.com/display/HWARE/Functional+Overview+of+the+NanoBoard+3000. [Pokušaj pristupa 29. Lipanj 2016].

[25] Altium Limited, »NanoBoard Examples,« Altium Limited, 2016. [Mrežno]. Available: https://designcontent.live.altium.com/NanoBoardExampleDetail/#NanoBoardExamples/. [Pokušaj pristupa 3. Srpanj 2016].

[26] Altium Limited, Sydney, Australia, »TSK3000A,« Altium Limited, 6. Studeni 2013. [Mrežno]. Available: http://techdocs.altium.com/display/FPGA/TSK3000A. [Pokušaj pristupa 21. Lipanj 2016].

[27] Altium Limited, Sydney, Australia, »NanoBoard 3000 - Audio System,« Altium Limited, 6. Studeni 2013. [Mrežno]. Available: http://techdocs.altium.com/display/HWARE/NanoBoard+3000+-+Audio+System. [Pokušaj pristupa 21. Lipanj 2016].

[28] Altium Limited, »Types of Projects in Altium Designer,« Altium Limited, 6. Studeni 2013. [Mrežno]. Available: http://techdocs.altium.com/display/ADOH/Types+of+Projects+in+Altium+Designer. [Pokušaj pristupa 7. Srpanj 2016].

[29] Altium Limited, »Introduction to the Software Platform,« Altium Limited, 6. Studeni 2013. [Mrežno]. Available: http://techdocs.altium.com/display/AEE/Introduction+to+the+Software+Platform. [Pokušaj pristupa 8. Srpanj 2016].

[30] B. Dropuljić i D. Petrinović, »Razvoj akustičkog modela hrvatskog jezika pomoću alata HTK,« *Automatika : časopis za automatiku, mjerenje, elektroniku, računarstvo i komunikacije,* svez. 51., br. 1., pp. 79-78, 2010.

[31] V. Volenec, Fonemika hrvatskog standardnog jezika, Zagreb: Filizofski fakultet Sveučilšta u Zagrebu, Odsjek za koratiskiku, 2012./2013..

[32] E. Ambikairajah, »Speech and Audio Processing 2: Speech Analysis - Professor E. Ambikairajah,« EET UNSW, 2. Travanj 2010. [Mrežno]. Available: https://www.youtube.com/watch?v=Y_mSQ7tTlvQ. [Pokušaj pristupa 1. Lipanj 2016].

[33] L. Rabiner i B.-H. Juang, Fundamentals of Speech Recognition, New Jersey: PTR Prentice-Hall, Inc., 1993.

[34] »Durbin Algorithm,« [Mrežno]. Available: http://ivoronline.com/Science/Mathematics/Durbin%20Algorithm/Durbin%20Algorithm.pdf. [Pokušaj pristupa 27. Lipanj 2016].

[35] Altium Limited, »NanoBoard 3000 - Audio CODEC,« Altium Limited, 5. Ožujak 2014. [Mrežno]. Available: http://techdocs.altium.com/display/HWARE/NanoBoard+3000+-+Audio+CODEC. [Pokušaj pristupa 29. Lipanj 2016].

[36] E. Ambikairajah, »Speech and Audio Processing 1: Introduction to Speech Processing - Professor E. Ambikairajah,« EET UNSW, 2. Travanj 2010. [Mrežno]. Available: https://www.youtube.com/watch?v=Xjzm7S__kBU. [Pokušaj pristupa 1. Srpanj 2016].

[37] E. Ambikairajah, »Speech and Audio Processing 3: Linear Predictive Coding (LPC) - Professor E. Ambikairajah,« EET UNSW, 2. Travanj 2010. [Mrežno]. Available: https://www.youtube.com/watch?v=lWH-Oh5KnNY. [Pokušaj pristupa 1. Srpanj 2016].

[38] T. Maksimović, »Usporedba hrvatske i engleske fonetike i fonologije,« 15. Studeni 2011. [Mrežno]. Available: hrcak.srce.hr/file/121501. [Pokušaj pristupa 20. Lipanj 2016].

[39] Altium Limited, Sydney, Australia, »TSK3000A Instruction Set,« Altium Limited, 16. Studeni 2013. [Mrežno]. Available: http://techdocs.altium.com/display/FPGA/TSK3000A+Instruction+Set. [Pokušaj pristupa 21. Lipanj 2016].

SAŽETAK

U ovom radu je dan pregled povijesti razvitka sustava za prepoznavanje govora. Opisani su osnovni algoritmi koje sustavi za prepoznavanje govora koriste. Objašnjeno je što su to značajke govornog signala te kako se stvaraju. Sustavi za prepoznavanje govora koriste algoritme odabira jezičnih jedinica. Dinamičko savijanje vremena je algoritam u kojem se govorni signal uspoređuje s drugim govornim signalom dinamičkim savijanjem značajki signala po vremenskoj osi. Umjetne neuronske mreže su alat koji se primjenjuje na problem pretraživanja uzoraka, pa tako i na problem prepoznavanja govora. Skriveni Markovljevi modeli uspješno modeliraju razne jezične jedinice te su jedan od najčešćih algoritama koje se koriste u sustavima za prepoznavanje govora. Dubinske neuronske mreže se koriste u sustavima s velikim zahtjevima. Za implementaciju je korišten Altium NanoBoard 3000 sustav koji je razvijan u programu Altium Designer. Razvijen je ugrađeni računalni sustav koji se temelji na TSK3000A procesoru, a uključuje audio jedinicu koja služi za unos zvuka s audio ulaza ploče. Sustav koji je implementiran se temelji na linearno-prediktivnoj analizi. Linearno-preditkivnom analizom nastaju LPC koeficijenti koji u konačnici služe za usporedbu s usrednjenim značajkama u sustavu za prepoznavanju glasova hrvatskog jezika. Provedeno je testiranje napravljenog sustava za prepoznavanje glasova hrvatskog jezika. Točnost sustava za cjelokupno proveden eksperiment iznosi 44%. Za snimke po kojima su napravljene referente usrednjene značajke sustava točnost iznosi 93%. Mjerenje vremena proračuna značajki je potvrdilo da sustav u trenutnom stanju ne može ispuniti vremenski zahtjev sustava za proračunom u stvarnom vremenu.

Ključne riječi: automatsko prepoznavanje govora, Altium NanoBoard, FPGA, značajke signala govora, linerano-preditkivna analiza, glasovi hrvatskog jezika

ABSTRACT

This thesis contains historical development overview of voice recognition systems. Basic algorithms that are used in voice recognition systems are described. Voice signal features are explained as much as process of their creation. Speech recognition systems are using algorithms for selection of language units. Dynamic time warping is algorithm in wich one speech singal is compared to another speech signal by dynamically warping characteristics of signal in time domain. Artificial neural networks are tool that is used for solving pattern comparison problems, wich means that they are also used for speech recognition purposes. Hidden Markov models are succesfully used for modeling different kinds of language units so they are one of the most frequently used algorithms in speech recognition systems. Deep neural networks are used in

systems with large requirements. Implementation is designed for Altium NanoBoard 3000 system in Altium Designer software. Embedded computer system that is based on TSK3000A processor has been developed. System also utilizes audio unit that is used for sound recording from audio input located on the board. Implemented system is based on linear predictive analysis. Results of linear predictive analysis are LPC coefficients that are used for comparisson with mean features in the system for recognition of Croatian phones. System for recogniton of Croatian phones has been tested. Precision of the system is 44% for every recording of individual phone. For recordings that are used for generation of mean features precision is 93%. It is confirmed by measuring feature extraction time that current system is inadequate for requirement of real time calculation.

Keywords: automatic speech recognition, Altium NanoBoard, FPGA, voice signal features, linear predictive analysis, Croatian phones

ŽIVOTOPIS

Matija Labak je rođen 15. studenog 1991. u Đakovu. Živi u Josipovcu Punitovačkom, u blizini Đakova. 2010. godine završava III. gimnaziju u Osijeku i upisuje Preddiplomski sveučilišni studij elektrotehnike na Elektrotehničkom fakultetu u Osijeku. 2013. godine završava Preddiplomski studij elektrotehnike i upisuje Diplomski studij elektrotehnike, smjer; komunikacije i informatika. Sudjeluje u 24-satnim natjecanjima u programiranju IEEE Xtreme (redom IEEE Xtreme 5.0, 6.0, 7.0, 8.0, 9.0) od 2011. do 2015. godine unutar tročlane ekipe pod nazivom SyntaxErrorETFOS. Godine 2015. postaje stipendistom tvrtke RT-RK. Stanuje u Osijeku. Od stranih jezika zna dobro engleski.

_____

potpis

PRILOZI

Prilog P1: Sustav za prepoznavanje glasova hrvatskog jezika – FPGA projekt s pripadnim ugrađenim projektom za program Altium Designer namijenjen pokretanju na ploči Altium NanoBoard 3000, nalazi se na CD-u

Prilog P2: Sustav za stvaranje i ispis značajki glasovnog signala značajki jezika – FPGA projekt s pripadnim ugrađenim projektom za program Altium Designer namijenjen pokretanju na ploči Altium NanoBoard 3000, nalazi se na CD-u

Prilog P3: Komandno-linijski program za računanje usrednjenih značajki, nalazi se na CD-u

Prilog P4: Snimljeni glasovi, nalaze se na CD-u

Prilog P5: Excel tablica koja sadrži test 1 i test 2, nalazi se na CD-u

Prilog P6: Snimljene riječi korištene u testu 4, nalaze se na CD-u

Prilog S1: Primjer izračuna LPC koeficijenata Levinson-Durbinovom metodom

Pretpostavimo da je $p = 3$ te da je računato rješenje jednadžbe po navedenom postupku. Izračunate vrijednosti bi se dobile sljedećim redoslijedom:

- Prvo je postavljena vrijednost $E^{(0)} = r[0]$ prema (4-37).
- Nakon toga je dobivena vrijednost $a_1^{(1)}$ prema (4-38). Za račun su potrebne vrijednosti $r[1]$ i $E^{(0)}$. Ovo je ujedno rješenje jednadžbe kada bi vrijedilo $p = 1$. Prednost Durbinovog postupka upravo u tome što postoji mogućnost prekida usred bilo koje iteracije, tako da rješenje vrijedi za sustav dimenzija rednog broja iteracije u kojem je prekinut.
- Korak opisan jednadžbom (4-39) je preskočen u prvoj iteraciji jer nije zadovoljen uvjet za izvršenje ovog koraka.
- Izračunata je vrijednost $E^{(1)}$ prema (4-40). Ovime završava prva iteracija.
- Druga iteracija je započeta ($i = 2$) te je izračunata vrijednost $a_2^{(2)}$ prema (4-38). Za račun su potrebne vrijednosti $r[2]$, $E^{(1)}$, $a_1^{(1)}$ i $r[1]$.
- Računa se $a_1^{(2)}$ prema (4-39). Za proračun su potrebne vrijednosti $a_1^{(1)} i\, a_2^{(2)}$ izračunate u prethodnim koracima.
- $E^{(2)}$ se dobiva prema (4-40) pomoću vrijednosti $a_2^{(2)}$ i $E^{(1)}$. Završetak druge iteracije.

- Treća iteracija započinje ($i = 3$) izračunom vrijednosti $a_3^{(3)}$ prema (4-38). Za račun su potrebne vrijednosti $r[3]$, $r[2]$, $r[1]$, $E^{(2)}$, $a_2^{(1)}$ i $a_2^{(2)}$.
- Računa se $a_1^{(3)}$ i $a_2^{(3)}$prema (4-39). Za proračun $a_1^{(3)}$su potrebne vrijednosti $a_1^{(2)}$, $a_3^{(3)}$ i $a_2^{(2)}$, a za $a_2^{(3)}$ su potrebne vrijednosti $a_1^{(2)}$, $a_3^{(3)}$ i $a_1^{(2)}$.
- $E^{(3)}$ se dobiva prema (4-40) pomoću vrijednosti $a_3^{(3)}$ i $E^{(2)}$. Ovime završava treća iteracija te ukupan proračun za slučaj $p = 3$.

Prethodnim postupkom su dobivena sljedeća rješenja:

1. iteracija: $a_1^{(1)}$
2. iteracija: $a_2^{(2)}$ i $a_1^{(2)}$
3. iteracija: $a_3^{(3)}$, $a_2^{(3)}$ i $a_1^{(3)}$

Konačna rješenja jednadžbe se uzimaju iz posljednje iteracije, ona glase:

$a_3 = a_3^{(3)}$,

$a_2 = a_2^{(3)}$,

$a_1 = a_1^{(3)}$.