Metod implementacije sistema za prepoznavanje govora na embedded uređajima

Branislav Milojković

***Sadržaj* — U radu se diskutuju algoritmi pogodni za implementaciju na embedded platformama, a koji se koriste pri rešavanju problema prepoznavanja govora. Biće omogućeno izvršiti prepoznavanje govora koje zavisi od govornika (speaker dependant), i to pretragom kroz rečnik unapred pripremljenih fraza. Opisana je teorijska osnova primenjenih algoritama, kao i način praktične implementacije svake faze sistema.**

***Ključne reči* — DTW, Embedded, LPC, MFCC, Prepoznavanje govora**

I. Uvod

OVAJ rad se bavi problematikom implementacije sistema za prepoznavanje govora na embedded uređajima. Želja je da se prikažu neophodni teorijski i praktični koncepti koji omogućavaju da ovakav sistem može da se u potpunosti dizajnira i implementira. Rad je podeljen na tri oblasti – metod dobijanja svojstava glasa, ilustracija mehanizma za prepoznavanje reči, i opis završenog proizvoda. Prva oblast se bavi konverzijom digitalnog signala u oblik koji je pogodna reprezentacija govora, koja je ujedno kompaktna i omogućava efektivnu pretragu na takav način da će reči koje poseduju slične glasove biti obeležene kao bliske jedna drugoj. Druga oblast se bavi algoritmima koji se koriste u samoj pretrazi. Svi opisani algoritmi su tako probrani da nisu suviše velike kompleksnosti, ni vremenski ni prostorno, tako da su pogodni za implementaciju na platformama sa ograničenim količinama resursa. Treća oblast opisuje način funkcionisanja završenog sistema sa korisnikove tačke gledišta.

B. Milojković, Računarski fakultet, Srbija (telefon: 381-63-664622; e-mail: bmilojkovic@raf.edu.rs).

II. DOBIJANJE SVOJSTAVA

A. Prikupljanje signala

Moderni računari su sposobni da se bave poslovima vezanim za prikupljanje signala gotovo čisto softverski. Na primer, mnoge PC zvučne kartice poseduju mogućnost direktnog upisa u memoriju, tako da nema potrebe za opterećivanjem procesora sa prekidnim I/O operacijama. Operativni sistemi su obično sposobni da obave sve neophodne AD/DA operacije u realnom vremenu.

Da bi se sistem prepoznavanja govora kvalitetnije implementirao, moguće je izvršiti preklapanje izvršavanja nekih od komponenti, kao što su digitalizacija govora, izvlačenje i transformacija svojstava, akustička pretraga šablona, i pretraga šablona zasnovana na modelima jezika. Mnogi operativni sistemi dolaze sa gotovim mehanizmima za preklapanje ovakvih procesa u okruženju koje je inače paralelno. Baferi moraju da se pravilno alociraju da bi se obezbedio sinhronizovan rad svake pojedinačne komponente. Veći baferi su generalno neophodni na sporijim računarima, zbog relativno spore obrade i uvođenja potencijalnog kašnjenja. Odgovarajuće veličine bafera mogu da se odrede eksperimentalnim putem, tako što se računar stavi u rad sa različitim nivoima opterećenja, i merenjem kašnjenja.

Konkretno, kada je u pitanju obrada glasovnog signala, veličine bafera se kreću u opsegu od 4 do 64 kB, sa 16-kHz frekvencijom odabiranja glasa i 16-bit A/D preciznošću. U praksi, 16-kHz frekvencija odabiranja je dovoljna da se uhvati frekventni opseg govora (8 kHz). Smanjeni opsezi, kao što su npr. kod telefonskih veza, generalno povećavaju nivo greške pri prepoznavanju govora. Tabela 1 pruža neke empirijski dobijene rezultate za relativne odnose povećanja nivoa greške pri različitim frekvencijama odabiranja. Ako uzmemo 8kHz kao početni uzorak, možemo da smanjimo nivo greške sa sličnim mehanizmom prepoznavanja za 10% tako što povećamo frekvenciju odabiranja na 11kHz. Ako dalje povećamo frekvenciju odabiranja na 16kHz, nivo greške može dodatno da se smanji za još 10%. Povećavanje frekvencije do konačnih 22kHz više nema uticaja pošto je većina osobina govora unutar opsega do 8kHz.

TABELA 1: SMANJENJE NIVOA GREŠKE PREMA POVEĆANJU FREKVENCIJE ODABIRANJA

Frekvencija odabiranja	***Relativno smanjenje nivoa greške***
8 kHz	Referentna tačka
11 kHz	+10%
16 kHz	+10%
22 kHz	+0%

B. Detekcija graničnih tačaka

Postoji više modela koji se mogu koristiti da bi se aktivirao proces prikupljanja govornog signala. Dva najčešće korišćena pristupa su *govor-na-pritisak* i *kontinualno slušanje*. Kod govora-na-pritisak postoji poseban taster u interfejsu uređaja koji aktivira prikupljanje govornog signala, što štiti sistem od šuma i eliminiše nepotrebno korišćenje procesora za detekciju događaja nevezanih za govor. Ovakav način rada nekada od korisnika zahteva i da drži taster pritisnut dok govori. Pritisak tastera naznačava sistemu da govor počinje, dok otpuštanje tastera naznačava da treba završiti sa prikupljanjem zvučnog signala. Očigledan nedostatak ovog pristupa je zahtev da aplikacija mora da se pokrene svaki put kada korisnik želi da izda komandu.

Kod modela kontinualnog slušanja, sistem konstantno prikuplja zvučni signal i automatski detektuje da li u signalu ima govora ili ne. U tom slučaju, sistem mora da poseduje tzv. detektor graničnih tačaka, koji se tipično zasniva na veoma efikasnim klasifikatorima šablona. Ovaj klasifikator se koristi da eliminše onaj deo signala koji je očigledna tišina, dok se precizno ograničavanje govora ostavlja modulu za prepoznavanje govora. U poređenju sa modelom govor-na-pritisak, kod kontinualnog slušanja se troši mnogo veća količina procesorskog vremena, i postoji rizik da dođe do loše klasifikacije.

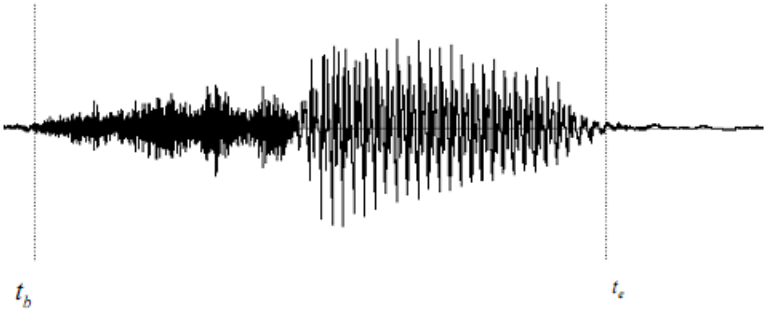
Detektor graničnih tačaka se tipično oslanja na neku gornju granicu energije, koja se menja u funkciji od vremena. Logaritmovana vrednost energije može da se dinamički generiše na osnovu nekog već prethodno pročitanog okvira. Moguće je postaviti i ograničenja na minimalne dužine reči, kako bi se eliminisali kratki pikovi koji tipično predstavljaju neki šum.

Nije od kritične važnosti da detektor graničnih tačaka omogući savršeno precizno ograničavanje reči. Ključni zahtev ovog dela sistema je nizak nivo odbacivanja (tj. bitno je da detektor graničnih tačaka ne klasifikuje pogrešno govor kao tišinu ili šum). Svaki pogrešno odbačen deo signala koji je zapravo govor će sigurno voditi pogrešnom tumačenju reči. Sa druge strane, potencijalno pogrešno prihvatanje (tj. klasifikacija perioda tišine ili šuma kao govora) nije toliko strašno, zato što modul za prepoznavanje govora može da izvrši kompenzaciju, uz primenu adekvatnih modela šuma, kao što su modeli pucketanja, klikova, udaraca usnama i pozadinskog šuma.

Eksplicitni detektori graničnih tačaka rade prihvatljivo dobro sa snimcima koji poseduju odnos signal/šum (SNR - signal to noise ratio) od 30dB ili veći, ali dosta lošije prolaze kod govora sa većim nivoom šuma. Kao što je napomenuto, modul za prepoznavanje govora može takođe da određuje

granične tačke reči primenom rečnika i primenom modela tišine / šuma. Ovaj pristup je tipično mnogo pouzdaniji od eksplicitne detekcije graničnih reči zasnovane na gornjoj granici energije, zato što prepoznavanje govora može istovremeno da detektuje granice reči, kao i same reči i klase šuma koji okružuje reči, ali je za ovaj proces noephodno više procesorskog vremena. Kompromisno rešenje je primeniti jednostavan adaptivni dvoklasni (jeste govor / nije govor) klasifikator za pronalaženje govornih aktivnosti (uz primenu dovoljno velikih bafera sa obe strane), i obavestiti modul za prepoznavanje govora da treba obaviti dodatno procesiranje.

Kada se dovoljan broj okvira za redom klasifikuje kao govor, modul za prepoznavanje govora se obaveštava o ovome, i on počinje da procesira signal. Kao što je prikazano na sl. 1, treba uključiti i izvestan broj okvira pre „početnog“ trenutka t_b, da bi modul za prepoznavanje govora mogao da izvrši potencijalno kompoenzovanje greške. Slično, kada se dovoljan broj okvira klasifikuje kao tišina nakon trenutka t_e, treba obezbediti izvestan broj dodatnih okvira modulu za prepoznavanje govora, kako bi i njih proverio.



Sl. 1. Granice treba produžiti kako bi modul za prepoznavanje govora mogao da detektuje greške pri odsecanju

C. Linearno prediktivno kodovanje

Teorija linearnog prediktivnog kodovanja (Linear Predictive Coding, LPC) i njena konkretna primena u raznim oblastima postoji već dugi niz godina [2]. U ovom odeljku ćemo ukratko opisati kako se LPC može primeniti kod sistema koji se bave prepoznavanjem govora. Počećemo navođenjem nekih od razloga zašto se baš LPC pokazao kao dobar pristup:

- LPC pruža dobar model govornog signala. Kod bezvučnih i tranzijentnih delova govora, LPC pristup je manje efektivan nego kod zvučnih, ali i pored toga pruža dovoljno prihvatljiv model.

- Način na koji se LPC primenjuje pri analizi govornog signala vodi dobrom razdvajanju različitih mogućih izvora unutar govornog aparata. Kao rezultat toga, moguće je napraviti analizu glasa na osnovu izvora zvuka, što znamo da je direktno povezano sa time koji zvuk je u pitanju.
- LPC je jasan analitički model. Metod LPC-a se može jasno predstaviti matematički i direktno se odatle preneti u implementaciju u softveru ili hardveru.
- LPC model daje dobre rezultate prepoznavanja govora u praksi.

Osnovna ideja iza LPC modela je ta da se dati odbirak govora u trenutku n, $s(n)$, može aproksimirati kao linearna kombinacija prethodnih p odbiraka, na sledeći način:

$$s(n) \approx a_1 s(n-1) + a_2 s(n-2) + \cdots + a_p s(n-p), \quad (1)$$

gde se za kofeficijente $a_1, a_2, \ldots a_p$ uzimaju konstantne vrednosti na nivou okvira koji se analizira. Ovu relaciju možemo da pretvorimo u pravu jednakost na sledeći način:

$$s(n) = \sum_{i=1}^{p} a_i s(n-i) + Gu(n), \quad (2)$$

Gde je $u(n)$ normalizovana pobuda, a G je pojačanje pobude.

Sa druge strane, prostu linearnu kombinaciju prethodnih p uzoraka nazivamo procenom $\tilde{s}(n)$, i definišemo je kao:

$$\tilde{s}(n) = \sum_{k=1}^{p} a_k s(n-k) \quad (3)$$

Sada možemo da formiramo i grešku predikcije kao:

$$e(n) = s(n) - \tilde{s}(n) = s(n) - \sum_{k=1}^{p} a_k s(n-k) \quad (4)$$

sa prenosnom funkcijom greške:

$$A(z) = \frac{E(z)}{S(z)} = 1 - \sum_{k=1}^{p} a_k z^{-k} \quad (5)$$

Jasno je da kada je $s(n)$ generisano od strane linearnog sistema kao što je opisan prethodno, greška predikcije $e(n)$ će biti baš $Gu(n)$, tj. skalirana pobuda.

Ključni problem LPC pristupa jeste odrediti skup koeficijenata $\{a_k\}$ direktno iz govornog signala, tako da spektralna svojstva dobijenog digitalnog filtera odgovaraju onima iz govornog zvučnog talasa unutar okvira koji se analizira. Pošto se spektralne karakteristike govora menjaju u vremenu, koeficijenti u zadatom trenutku n, moraju da se procenjuju na osnovu kratkog segmenta govora koji se dešava u relativnoj blizini trenutka n. Tako dolazimo do pristupa u kojem tražimo takve koeficijente da srednja kvadratna greška predikcije na kratkom delu govornog signala bude minimalna. (Obično se ova vrsta spektralne analize vrši na susednim okvirima govornog signala, sa širinom okvira od 10ms.)

Kako bismo postavili jednačine koje treba rešiti da bi se dobili koeficijenti, prvo definišemo kratkoročne segmente govora i greške u trenutku n kao:

$$s_n(m) = s(n+m) \tag{6}$$

$$e_n(m) = e(n+m) \tag{7}$$

i želimo da minimizujemo srednju kvadratnu grešku signala u trenutku n:

$$E_n = \sum_m e_n^2(m) \tag{8}$$

što možemo transformisati primenom definicije $e_n(m)$ preko $s_n(m)$, na sledeći način:

$$E_n = \sum_m \left[s_n(m) - \sum_{k=1}^{p} a_k s_n(m-k) \right]^2 \tag{9}$$

Da bismo rešili jednačinu (3.9) po a_k, parcijalno diferenciramo E_n po svakom a_k, i izjednačavamo sa nulom:

$$\frac{\partial E_n}{\partial a_k} = 0, \qquad k = 1,2,\dots,p \tag{10}$$

što nam daje:

$$\sum_m s_n(m-i)s_n(m) = \sum_{k=1}^{p} \hat{a}_k \sum_m s_n(m-i)s_n(m-k) \tag{11}$$

Prepoznavanjem da su nam svi izrazi oblika $\sum_m s_n(m-i)s_n(m-k)$ takođe izrazi kratkoročne kovarijanse od $s_n(m)$, to jest:

$$\phi_n(i,k) = \sum_m s_n(m-i)s_n(m-k) \tag{12}$$

možemo da izrazimo jendačinu (3.11) u kompaktnoj notaciji:

$$\phi_n(i,0) = \sum_{k=1}^{p} \hat{a}_k \phi_n(i,k) \tag{13}$$

Čime opisujemo skup od p jednačina i p nepoznatih. Jasno se izvodi da se srednja kvadratna greška $\hat{E}_n$ može dobiti sa:

$$\hat{E}_n = \sum_m s_n^2(m) - \sum_{k=1}^{p} \hat{a}_k \sum_m s_n(m)s_n(m-k) \tag{14}$$

$$= \phi_n(0,0) - \sum_{k=1}^{p} \hat{a}_k \phi_n(0,k) \tag{15}$$

D. Autokorelacioni metod za računanje LPC

U jednačini (3.12) nismo postavili nikakav konkretan opseg za sumu. Kod autokorelacionog metoda [8] [9], pretpostavljamo da $s_n[m]$ uzima vrednost 0 u intervalu $0 \le m < M$:

$$s_n[m] = s[n+m]w[m] \tag{16}$$

Gde je w neka prozorska funkcija, na primer Hamingov prozor, koja ima vrednost 0 van opsega $0 \le m < M$. Uz ovu pretpostavku, odgovarajuća greška predikcije $e_n[m]$ će biti ne-nulta na intervalu $0 \le m < M + p$, i, samim tim, totalna vrednost greške će biti:

$$E_n = \sum_{n=0}^{M+p-1} e_n^2[m] \tag{17}$$

Sa ovim opsegom definisanim, jednačina (3.12) se može izraziti kao:

$$\phi_n(i,k) = \sum_{m=0}^{M+p-1} x_n[m-i]x_n[m-k] = \sum_{n=0}^{M-1-(i-k)} x_n[m]x_n[m+i-k] \tag{18}$$

Ili, alternativno:

$$\phi_n(i,k) = R_n[i-k] \tag{19}$$

gde je $R_n[l]$ autokorelaciona sekvenca od $x_n[m]$:

$$R_n[l] = \sum_{m=0}^{M-1-l} x_n[m]x_n[m+l] \tag{20}$$

Kombinovanjem jednačina (3.20) i (3.13), dobijamo:

$$\sum_{k=1}^{p} \hat{a}_k \, R_n[|i-k|] = R_n[i] \tag{21}$$

Što odgovara sledećoj matričnoj jednačini:

$$\begin{pmatrix} R_n[0] & R_n[1] & R_n[2] & \dots & R_n[p-1] \\ R_n[1] & R_n[0] & R_n[1] & \dots & R_n[p-2] \\ R_n[2] & R_n[1] & R_n[0] & \dots & R_n[p-3] \\ \dots & \dots & \dots & \dots & \dots \\ R_n[p-1] & R_n[p-2] & R_n[p-3] & \dots & R_n[0] \end{pmatrix} \begin{pmatrix} a_1 \\ a_2 \\ a_3 \\ \dots \\ a_p \end{pmatrix} = \begin{pmatrix} R_n[1] \\ R_n[2] \\ R_n[3] \\ \dots \\ R_n[p] \end{pmatrix} \tag{22}$$

Matrica u jednačini (3.22) je simetrična i svi elementi na glavnoj dijagonali su identični. Takve matrice se nazivaju *Toeplitz* matrice. Durbinova rekurzija koristi ovu činjenicu da dobije veoma efikasan algoritam za računanje koeficijenata a:

- Inicijalizacija

$$E^0 = R[0] \tag{23}$$

- Iteracija. Za $i = 1, \dots, p$, izvršiti sledeću rekurziju:

$$j_i = \left(R[i] - \sum_{k=1}^{i-1} a_k^{i-1} R[i-k]\right)/E^{i-1} \tag{24}$$

$$a_i^i = j_i \tag{25}$$

$$a_k^i = a_k^{i-1} - j_i a_{k-1}^{i-1}, \quad 1 \le k < i \tag{26}$$

$$E^i = (1 - j_i^2)E^{i-1} \tag{27}$$

- Konačno rešenje:

$$a_k = a_k^p, \qquad 1 \le k \le p \tag{28}$$

Gde se koeficijenti j_i nazivaju *reflektivni koeficijenti (koeficijenti parcijalne korelacije)*, i ograničeni su na opseg od -1 do 1.

E. MFCC

Uvodimo pojam Mel-Frequency Cepstrum Coefficients (MFCC), kao vrstu određivanja osobina glasovnog signala koja je zasnovana na perceptivnom modelu. Kod ovakvih modela se oslanjamo na osobine ljudskog slušnog aparata.

1) Bilinearna transformacija

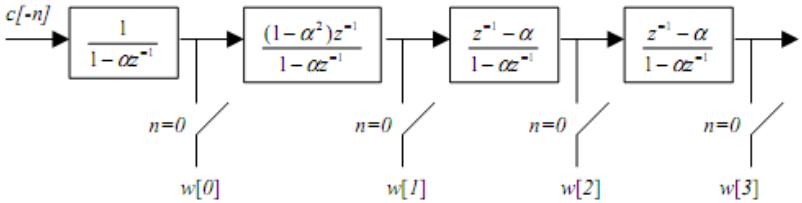
Transformacija:

$$s = \frac{z^{-1}-\alpha}{1-\alpha z^{-1}} \tag{29}$$

za $0 < \alpha < 1$ pripada klasi *bilinearnih* transformacija. U pitanju je mapiranje unutar kompleksne ravni koje mapira jediničnu krug na njega samog. Frekventna transformacija se dobija uvođenjem smena $z = e^{j\omega}$ i $s = e^{j\Omega}$.

$$\Omega = \omega + 2arctan\left[\frac{\alpha sin\,(\omega)}{1-\alpha cos\,(\omega)}\right] \tag{30}$$

Oppenheim [4] je pokazao da je prednost ove transformacije to što može da se iskoristi da pretvori vremensku sekvencu u linearnoj frekvenciji u drugu vremensku sekvencu u modifikovanoj frekvenciji, kao što je prikazano na sl. 2. Ova bilinearna transformacija je uspešno primenjivana na cepstralne i autokorelacione koeficijente.



Sl. 2. Implementacija cepstralnih koeficijenata kao funkcije koeficijenata linearno-frekventnog spektruma

Za konačan broj cepstralnih koeficijenata, bilinearna transformacija sa slike 13 daje beskonačan broj modifikovanih cepstralnih koeficijenata. Pošto se u praksi obično radi odsecanje, bilinearna transformacija je jednaka množenju matrica, gde je matrica funkcija faktora modifikacije α. Shikano [5] je pokazao da su ovi modifikovani cepstralni koeficijenti adekvatni za prepoznavanje govora.

2) Mel-Frequency Cepstrum

MFCC je način predstavljanja osobina glasovnog signala koji je definisan kao realni cepstrum kratkog okvira signala koji je izveden iz FFT tog signala. Razlika u odnosu na realni cepstrum je ta da se primenjuje nelinearna skala, koja služi da aproksimira način rada ljudskog slušnog aparata. Davis i Mermelstein [7] su pokazali da je MFCC predstavljanje pogodno za primene u prepoznavanju govora.

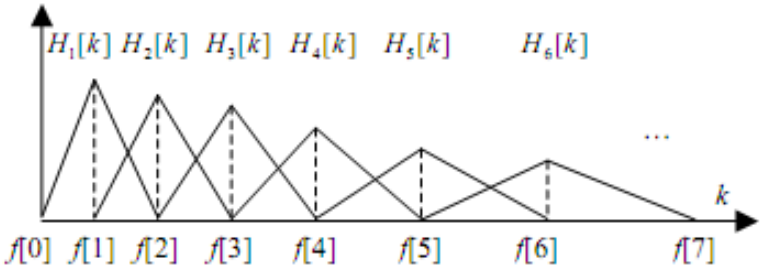
Ako je dat DFT ulaznog zvučnog signala:

$$X_a[k] = \sum_{n=0}^{N-1} x[n]e^{-j2\pi nk/2}, \quad 0 \le k \le N \tag{31}$$

definišemo banku filtera sa M filtera ($m = 0,1,2,\dots,M$), gde je filter m trougaoni filter zadat sa:

$$H_m[k] = \begin{cases} 0 & k < f[m-1] \\ \frac{2(k-f[m-1])}{(f[m+1]-f[m-1])(f[m]-f[m-1])} & f[m-1] \le k \le f[m] \\ \frac{2(f[m+1]-k)}{(f[m+1]-f[m-1])(f[m+1]-f[m])} & f[m] \le k \le f[m+1] \\ 0 & k > f[m+1] \end{cases} \quad (32)$$

Ovakvi filteri daju srednju vrednost spektruma oko svakog centra frekvencije, sa postepenim povećanjem propusnog opsega, kao što je prikazano na sl. 3.



Sl. 3 Trougaoni filteri koji se koriste pri računanju MFCC

Alternativno se filteri mogu definisati kao:

$$H'_m[k] = \begin{cases} 0 & k < f[m-1] \\ \frac{(k-f[m-1])}{(f[m]-f[m-1])} & f[m-1] \le k \le f[m] \\ \frac{(f[m+1]-k)}{(f[m+1]-f[m])} & f[m] \le k \le f[m+1] \\ 0 & k > f[m+1] \end{cases} \quad (33)$$

što zadovoljava $\sum_{m=0}^{M-1} H'_m[k] = 1$. Mel-cepstrum koji se dobije sa $H_m[k]$ ili sa $H'_m[k]$ će se razlikovati za konstantan vektor po svim ulaznim vrednositma, tako da je izbor nebitan kada se koristi u sistemu koji je treniran pomoću istih filtera.

Definišimo f_l i f_h kao najnižu i najvišu frekvenciju banke filtera u Hz, F_s je frekvencija odabiranja u Hz, M je broj filtera, i N je veličina FFT-a. Granične tačke $f[m]$ su uniformno raspoređene na mel-skali:

$$f[m] = \left(\frac{N}{F_s}\right) B^{-1}\left(B(f_l) + m\frac{B(f_h)-B(f_l)}{M+1}\right) \quad (34)$$

Gde je mel-skala B data sa:

$$B(f) = 1125\, ln\left(1 + \frac{f}{700}\right) \quad (35)$$

i njena inverzna funkcija sa:

$$B^{-1}(b) = 700\left(exp\left(\frac{b}{1125}\right) - 1\right) \quad (36)$$

Onda logaritamsku energiju izlaza svakog filtera računamo sa:

$$S[m] = ln[\sum_{k=0}^{N-1}|X_a[k]|^2 H_m[k]], \quad 0 \le m < M \quad (37)$$

Mel-frekventni cepstrum je onda diskretna kosinusna transformacija M izlaza filtera:

$$c[n] = \sum_{m=0}^{M-1} S[m] cos\left(\frac{\pi n\left(m+\frac{1}{2}\right)}{M}\right), \quad 0 \le n < M \quad (38)$$

gde M varira zavisno od implementacije u opsegu od 24 do 40. Za prepoznavanje govora se tipično koriste samo prvih 13 cepstrum koeficijenata.

F. Dobijanje Cepstrum koeficijenata iz LPC koeficijenata

Primena cepstrum koeficijenata preko LPC koeficijenata je posebno značajna, zato što predstavlja veoma dobar model ljudskog govornog aparata.

Ako nam je dat LPC filter:

$$H(z) = \frac{G}{1-\sum_{k=1}^{p} a_k z^{-k}} \quad (39)$$

i za njega nađemo logaritam:

$$\widehat{H}(z) = ln\ G - ln\ (1 - \sum_{l=1}^{p} a_l z^{-l}) = \sum_{k=-\infty}^{\infty} \hat{h}[k] z^{-k} \quad (40)$$

I potom nađemo izvod po z od obe strane:

$$\frac{-\sum_{n=1}^{p} n a_n z^{-n-1}}{1-\sum_{l=1}^{p} a_l z^{-l}} = -\sum_{k=-\infty}^{\infty} k\hat{h}[k] z^{-k-1} \quad (41)$$

Množenjem obe strane sa $-z(1 - \sum_{l=1}^{p} a_l z^{-l})$, dobijamo:

$$\sum_{n=1}^{p} n a_n z^{-n} = \sum_{n=-\infty}^{\infty} n\hat{h}[n] z^{-n} - \sum_{l=1}^{p}\sum_{k=-\infty}^{\infty} k\hat{h}[k] a_l z^{-k-l} \quad (42)$$

Što, kada zamenimo $l = n - k$, i izjednačimo po z^{-1}, dobijamo:

$$\begin{aligned} n a_n &= n\hat{h}[n] - \sum_{k=1}^{n-1} k\hat{h}[k] a_{n-k}, \quad 0 < n \le p \\ 0 &= n\hat{h}[n] - \sum_{k=n-p}^{n-1} k\hat{h}[k] a_{n-k}, \quad n > p \end{aligned} \quad (43)$$

tako da se kompleksan cepstrum može dobiti iz LPC koeficijenata pomoću sledeće rekurzije:

$$\hat{h}[n] = \begin{cases} 0 & n < 0 \\ ln\ G & n = 0 \\ a_n + \sum_{k=1}^{n-1}\left(\frac{k}{n}\right)\hat{h}[k] a_{n-k} & 0 < n \le p \\ \sum_{k=n-p}^{n-1}\left(\frac{k}{n}\right)\hat{h}[k] a_{n-k} & n > p \end{cases} \quad (44)$$

Napomenimo da, dok je broj LPC koeficijenata konačan, broj cepstralnih koeficijenata je beskonačan. Istraživanje u oblasti prepoznavanja govora je pokazalo da konačan broj od 12-20 (zavisno od frekvencije odabiranja) pruža zadovoljavajuću preciznost. Ova rekurzija ne bi smela da se koristi u

suprotnom pravcu, tj. da se pokuša dobijanje LPC koeficijenata od bilo kog skupa cepstralnih koeficijenata.

III. PREPOZNAVANJE REČI

Nakon što smo dobili svojstva koja koristimo da predstavimo zvučni signal na kompaktan i praktičan način, dolazi faza samog prepoznavanja reči. U ovoj fazi nam je cilj da iskoristimo dobijena svojstva kao ulaz za algoritam pretrage, čiji će krajnji cilj biti prepoznata reč, ujedno i konačan rezultat rada našeg sistema.

A. Deterministički pristup

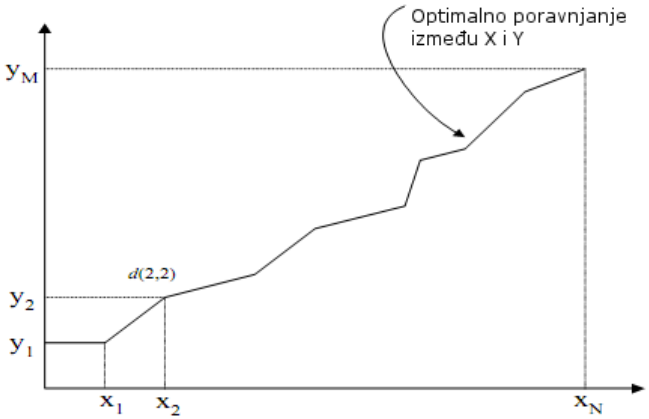
Za razliku od probabilističkih metoda (kao što je HMM), moguće je vršiti pretraživanje datog uzorka glasovnog signala na deterministički način. Ovakvi metodi se najčešće svode na pokušaje pronalaženja poklapanja između vektora svojstava koji su dobijeni u prethodnom koraku (LPC, MFCC, ...).

B. Pretraživanje po vektorima svojstava

Osnovni problem koji se javlja kod pokušaja poređenja šablona vektora svojstava, jeste pojava promene u vremenu trajanja govornog artifakta (bio to glas, slovo, reč, ili bilo koja glasovna celina koju posmatramo pri prepoznavanju...). Pri poređenju odvojenih delova jednog artifakta, treba uzeti u obzir da promena brzine izgovaranja dela artifakta ne treba da doprinosi (lingvističkoj) različitosti među artifaktima. Javlja se potreba da se izvrši normalizacija promene brzine izgovaranja kako bi se dobio smislen rezultat poređenja. Jedan od najčešće korišćenih algoritama koji omogućava baš takvu vrstu normalizacije jeste DTW.

1) Dinamičko programiranje i DTW

Koncept iz dinamičkog programiranja, koji se takođe naziva dinamička distorzija vremena (Dynamic Time Warp, DTW), se često koristi u svrhe pronalaženja relativne sličnosti između dva govorna šablona. Kod sistema koji su zasnovani na šablonima, svaki šablon se sastoji od sekvence govornih vektora. Ukupna mera rastojanja između šablona se računa kao udaljenost između vektora. DTW metod može da izvrši pomeraj govornih šablona $(x_1, x_2, \dots, x_N)$ i $(y_1, y_2, \dots, y_M)$ u vremenskoj dimenziji, kako bi se eliminisale nellinearne promene, kao što je prikazano na sl. 4.



Sl. 4. Direktno poređenje dva šablona X i Y

Ovaj problem se svodi na pronalaženje minimalne udaljenosti unutar rešetke koja je sačinjena od ova dva šablona. Svakom paru (i, j) dodeljuje se rastojanje $d(i, j)$, koje predstavlja udaljenost između vektora x_i i y_j. Da bi se pronašla optimalna putanja od početne tačke $(0,0)$ do krajnje (N, M) s leva na desno, treba da sračunamo optimalnu akumuliranu udaljenost $D(N, M)$. Jedno rešenje je da izračunamo sve moguće akumulirane udaljenosti od $(1,1)$ do (N, M) i pronađemo onu koja ima minimalnu vrednost. Pošto ima M mogućih koraka u svakom pomeraju s leva na desno, pronalaženje svih mogućih putanja od $(1,1)$ do (N, M) je proces eksponencijalne kompleksnosti. Principi dinamičkog programiranja mogu drastično da smanje količinu izračunavanja tako što izbegavaju one putanje koje sigurno ne mogu da dovedu do optimalnog rezultata. Pošto optimalna putanja u svakom koraku mora da bude zasnovana na prethodnom koraku, minimalno rastojanje $D(i, j)$ mora da zadovolji sledeću jednačinu:

$$D(i, j) = min[D(i - 1, k) + d(k, j)] \quad (45)$$

Prednost dinamičkog programiranja leži u činjenici da kada se jednom pronađe rešenje pod-problema, moguće je sačuvati to rešenje i ne računati ga ponovo u budućnosti. Ovaj važan princip se javlja na više mesta u sistemima prepoznavanja govora.

U praksi, da bi DTW dao smislene rezultate, neophodno je nametnuti neka ograničenja na samu funkciju distorzije. Ako ova ograničenja nisu postavljena, moguće je da se desi da signali koji predstavljaju potpuno različite reči dobiju veoma blisku udaljenost, jednostavno zato što DTW po

svojoj prirodi vrši odabir minimlanih udaljenosti. Tipična ograničenja koja se postavljaju uključuju:

- Ograničenja graničnih tačaka
- Uslovi monotonosti
- Oganičenja lokalnog kontinuiteta
- Ograničenja globalne putanje
- Težina nagiba

Ova ograničenja su detaljno opisana u [6].

IV. OPIS REALIZOVANOG SISTEMA

A. Ulaz/izlaz

Razvojno i izvršno okruženje za ovaj sistem je Mikroelektronika EasyMx PRO V7 razvojna ploča sa ARM Cortex STM407VG mikrokontrolerom. Korišćen je kompajler Mikro C PRO za ARM v. 4.0.0. Ulazno/izlazni interfejs je sledeći:

- Mikrofon je povezan na AUDIO IN port (VS1053 audio čip)
- Taster funkcija je povezana na PA0 dugme.
- Jumper funkcija je povezana na PA4 dugme.
- Žuti LED indikator (Busy) je PD0 LED.
- Crveni LED indikator (Error) je PD1 LED.

B. Korišćenje sistema

1) Pokretanje sistema

- Pri pokretanju sistema, korisnik treba da sačeka približno 8 sekundi, ili dok se Busy LED lampica ne ugasi kako bi audio čip mogao da se pripremi za rad.
- Nakon što sistem završi sa pripremom, on ulazi u režim učenja.
- Oba LED indikatora su isključena.

2) Režim učenja

- Kada je taster pritisnut, sistem je u stanju učenja.
- Korisnik treba da izgovori jasno i glasno jednu reč ili frazu dok je tater pritisnut. Mikrofon bi trebalo da se nalazi na udaljenosti od 10 do 20 cm od govornika. Pojačanje mikrofonskog signala je fiksirano na 64. Detektor glasovne aktivnosti se automatski aktivira na 66 dB. Ova vrednost može da se promeni kako bi se sistem prilagodio okruženju (visok nivo buke, udaljenost mikrofona, itd.) preko varijable threshold.

- Pauza između reči treba da bude približno jedan sekund. Dužina ovog perioda može da se promeni izmenom vrednosti BUFFER_SIZE. Ova izmena može da utiče na preciznost prepoznavanja.
- Dok sistem snima reči, Busy LED lampica je upaljena.
- Korisnik treba da sačeka da se lampica ugasi da bi nastavio sa procesom učenja (taster pritisnut), ili da produži na fazu prepoznavanja (taster otpušten).
- Ako ponestane prostora u fleš memoriji, Error LED lampica će trepereti sa periodom od 0.5 sekundi. Poslednja izgovorena reč neće biti uskladištena. Veličina fleš memorije je definisana preko konstante R_LIMIT.

3) Prepoznavanje

- Nakon otpuštanja tasera, sistem ulazi u stanje prepoznavanja.
- Pauza između reči trebalo bi da bude oko jedan sekund, ili makar dok je Busy lampica upaljena.
- Sistem će prepoznati reč ili frazu, i proizvesti nedekodovan indeks. Sistem potom poziva korisničku funkciju decode(), kojoj se prosleđuje dobijeni indeks kao argument. Ova funkcija može da izvrši dekodovanje indeksa na različite načine. Na primer, može da uključi ili isključi digitalne izlaze, ili može da pošalje podatke preko komunikacione linije. U trenutnoj implementaciji se indeks prikazuje na TFT ekranu.
- Ako korisnik pokuša da izvrši proces prepoznavanja, a nema reči koje bi mogle da budu prepoznate (baza podataka je prazna), sistem će prijaviti grešku paljenjem crvene Error LED lampice. Korisnik u tom slučaju treba da pritisne taster i snimi makar jednu reč.

4) Brisanje reči

Ako korisnik želi da obriše sve reči iz fleš memorije, treba da pritisne istovremeno PA0 i PA4 taatere. Ova procedura je uvedena kako bi se onemogućilo slučajno brisanje. Nakon što se tasteri otpuste, sistem ulazi u fazu pokretanja, a potom fazu prepoznavanja. U tom trenutku neće biti reči koje mogu biti prepoznate (baza podataka je prazna), tako da korisnik treba da pritisne taster i snimi makar jednu reč.

LITERATURA

[1] Xuedong Huang, Alex Acero, Hsiao-Wuen Hon, Spoken language processing: a guide to theory, algorithm and system development, Prentice Hall, 2001.
[2] J. D. Markel and A. H. Gray, Linear Prediction of Speech, Springer-Verlag, 1976.
[3] A.V. Oppenheim and D.H. Johnson, "Discrete Representation of Signals", The Proc. Of the IEEE, 1972, 60 (June), pp. 681-691.
[4] K. Shikano, K.-F. Lee, and R. Reddy, "Speaker Adaptation through Vector Quantization", IEEE Int. Conf. on Acoustics, Speech and Signal Processing, 1986, Tokyo, Japan pp, 2643-2646.
[5] L. Rabiner, Biing-Hwang Juang, Fundamentals of Speech Recognition, Prentice Hall, 1993.
[6] S. Davis and P. Mermelstein, "Comparison of Parametric Representations for Monosyllable Word Recognition in Contuinously Spoken Sentences", IEEE Trans. On Acoustics, Speech and Signal Processing, 1980, 28(4), pp. 357-366.
[7] J. Makhoul, "Spectral Analysis of Speech by Linear Prediction", IEEE Trans. on Acoustics, Speech and Signal Processing, 1973, 21(3), pp. 140-148.
[8] J.D. Merkel and A.H. Gray, "On Autocorrelation Equations as Applied to Speech Analysis", IEEE Trans. on Audio and Electroacoustics, 1973, AU-21 (April), pp. 69-79.

ABSTRACT

In this paper we discuss speech recognition related algorithms that are suitable for implementation on embedded devices. We will provide a system that can perform speaker dependant speech recognition, via a dictionary search. The dictionary is prefilled with phrases that need to be recognized. We will describe the theoretical foundation of the used algorithms, as well as provide a practical implementation description for all relevant system components.

METHOD FOR IMPLEMENTING A SPEECH RECOGNITION SYSTEM ON EMBEDDED PLATFORMS

Branislav Milojković