UNIVERSITÀ DEGLI STUDI DI PADOVA

UNIVERSITÀ DEGLI STUDI DI PADOVA

DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE

CORSO DI LAUREA IN
INGEGNERIA ELETTRONICA

SOUNDRISE 2.0: Sviluppo di un modello di riconoscimento timbrico per un sistema di assistenza web dedicato a persone con disabilità uditive

Relatore: PROF. SERGIO CANAZZA TARGON
Correlatore: DOTT. ALESSANDRO FIORDELMONDO

Laureando: RICCARDO FILA

Anno Accademico 2022-2023
Data di laurea 29/09/2023

Sommario

Soundrise è un'applicazione destinata a persone con disabilità uditive che fornisce una rappresentazione visiva delle caratteristiche della voce, analizzando in tempo reale il segnale audio proveniente dal microfono. Tramite la sua interfaccia accessibile e pensata appositamente per i bambini, Soundrise si propone come strumento di educazione al linguaggio per utenti con difficoltà nell'udito. In particolare vengono analizzate quattro proprietà del suono: intensità, altezza, durata e timbro; ciascuna delle quali determina una specifica connotazione grafica di un sole animato visibile sullo schermo. La prima versione del programma è stata sviluppata da Stefano Giusto e Marco Randon, tesisti di laurea magistrale, nel 2012. Ad oggi, dopo molti anni di mancato mantenimento del codice e il consistente progresso tecnologico, si è reso opportuno lo sviluppo di una nuova versione. Quest'ultima, denominata Soundrise 2.0, è in realtà la ricostruzione ex novo del progetto originale e si pone come obiettivi principali il funzionamento multipiattaforma (sfruttando le potenzialità e la diffusione dei linguaggi web) e un rinnovamento nell'aspetto che renda l'interfaccia più accattivante. Il progetto grafico è stato curato dal collega laureando Gabriele Turetta, mentre la programmazione del sistema di analisi del suono è stata eseguita dal sottoscritto e dal collega Andrea Zanetti. Questo elaborato si incentra sull'analisi di una delle quattro proprietà della voce trattate da Soundrise, ovvero il timbro. È oggetto della presente tesi una descrizione delle principali caratteristiche dello spettro di una produzione vocale che permettono di distinguere i fonemi delle vocali pronunciate. Vengono inoltre esposte le fasi dello sviluppo di un sistema di riconoscimento automatico dei fonemi, a partire dalla presentazione delle tecnologie utilizzate.

Ringraziamenti

Un sentito ringraziamento al prof. Sergio Canazza e al dott. Alessandro Fiordelmondo per avermi proposto questo progetto speciale, al quale non ho saputo dire di no.

Ringrazio di tutto cuore i miei amici, compagni di merenda e briscola in cinque nella magica cornice dell'Ex-Fiat, che mi hanno sostenuto standomi vicino con affetto e simpatia.

Un ringraziamento particolare va a Nicola Andreatta per il suggerimento dell'algoritmo di fattorizzazione, e a tutti coloro che hanno dedicato il loro tempo ad ascoltarmi e consigliarmi.

Grazie alla mia famiglia, ai miei genitori, alle nonne e ai nonni sempre presenti nella mia vita.

Indice

Elenco delle figure

Capitolo 1

L'educazione acustica

In questa tesi ci si propone di sviluppare un sistema automatico di riconoscimento del timbro della voce, che verrà poi integrato nel progetto Soundrise 2.0.

I soggetti a cui Soundrise si rivolge principalmente sono le persone sordomute e i deboli di udito. Il programma è pensato in modo particolare per i bambini affetti da questa disabilità, in quanto è nella loro età che si incentrano gli sforzi educativi, che saranno senz'altro di maggior rilievo visto il loro bisogno di attenzioni speciali. Lo scopo di Soundrise è quello di fornire un valido strumento che possa aiutare, anche in minima parte, a donare l'uso dell'udito e della parola a queste persone.

1.1 L'allenamento acustico

In quella che è definita da Montessori l'«età più ricettiva» ha una notevole importanza l'educazione ai sensi: far scoprire ai bambini il maggior numero possibile di stimoli e di novità li aiuta ad affinare i sensi e permette all'educatore di scoprire eventuali difetti percettivi. [1, p. 94]

Anche per i bambini colpiti da sordità assume un significato rilevante il «bagno acustico» (o «bagno sonoro»), un allenamento uditivo che offre loro un'immersione costante in una vastità di suoni differenti. Tutto ciò viene fatto approfittando dei residui di udito lasciati dalla disabilità (che può presentarsi con diversi gradi di perdita), e aiutandosi con degli amplificatori elettroacustici. [2, p. 109]

Il bagno acustico dev'essere intrapreso il prima possibile nella vita del bambino, ed è il primo livello dell'educazione all'uso della parola. Dopo aver imparato ad ascoltare, il bambino inizia a distinguere i suoni apprendendo le caratteristiche

di timbro, altezza, intensità e durata. [3, p. 480] In seguito darà un significato a ciascuna impressione acustica e, gradualmente, arriverà a distinguere le parole.

I quattro parametri sonori appena descritti sono quelli su cui opera Soundrise, e per i quali esso si propone come strumento che aiuti il bambino nel riconoscimento.

Risulta utile al metodo educativo la tipicità dei programmi di rispondere immediatamente alle richieste dell'utente. Se il bambino riceve un riscontro (che può essere rappresentato da uno stimolo di vario genere) sulla correttezza della risposta data, avrà modo di ricordarla come giusta o sbagliata e migliorerà il suo grado di apprendimento. [3, p. 505]

1.2 Il linguaggio simbolico

Soundrise deve il suo funzionamento a un linguaggio visivo simbolico. Per ogni caratteristica della voce mostra sullo schermo una specifica rappresentazione grafica, che può essere percepita da un bambino affetto da sordità. Una volta scelto il parametro che si vuole convertire in immagine, un sole animato sullo schermo cambierà d'aspetto in tempo reale in base ai dati sonori acquisiti dal microfono del dispositivo.

Degli studi hanno dimostrato che i simboli visivi normalmente utilizzati per l'insegnamento hanno un significato differente per i bambini affetti da sordità. [3, p. 507] Occorre quindi adoperare degli stimoli semplici che possano essere più facilmente recepiti come importanti durante l'attività. Quelli usati da Soundrise per le quattro caratteristiche della voce sono: [4, p. 32]

- Altezza: altezza del sole sull'orizzonte
- Intensità: dimensione del sole
- Durata: occhi aperti e bocca sorridente in presenza di suono
- Timbro: colore del sole
  - A: Rosso
  - E: Verde
  - I: Blu
  - O: Arancione
  - U: Grigio

Il colore giallo viene applicato al sole in assenza di una vocale riconosciuta. La scelta dei colori è stata fatta sulla base di uno studio riguardante l'associazione di colori e grafemi.[1] [5, p. 8]

Soundrise 2.0 risponde a tutte le caratteristiche vocali della prima versione. La grafica, completamente rinnovata, è stata sviluppata dal collega Gabriele Turetta.[2] La programmazione del sistema di riconoscimento di altezza, intensità e durata è stata curata dal collega Andrea Zanetti. Il riconoscimento del timbro invece viene trattato in questa tesi.

[1] J. Simner, J. Ward, M. Lanz, A. Jansari, K. Noonan, L. Glover e D. A. Oakley. Non-random associations of graphemes to colours in synaesthetic and non-synaesthetic populations. *Cognitive neuropsychology*, 22(8):1069-85, 2005.

[2] G. Turetta. *Soundrise 2.0: sviluppo di un'interfaccia grafica interattiva in Three.js per supportare persone con disabilità uditive.* Tesi di laurea, Università di Padova, 2023.

Capitolo 2

Modello di produzione del parlato

Per comprendere adeguatamente il metodo di riconoscimento timbrico sviluppato in questa ricerca, è opportuno descrivere l'apparato fonatorio dell'essere umano e una sua astrazione matematica su cui il suddetto metodo si appoggia.

2.1 Vocali

Le vocali sono dei foni prodotti con il canale epilaringeo aperto per tutta la loro durata. [6, p. 87] Proprio per l'importante contributo della cavità orale nella produzione delle vocali, esse sono chiamate anche fonemi di risonanza. [3, p. 59]

La distinzione delle vocali è data dalla posizione della lingua: sollevandosi e abbassandosi rispetto al palato, la lingua varia il grado di apertura del diaframma glottopalatale che classifica le vocali da aperte a chiuse, mentre spostandosi avanti e indietro si ottengono le vocali anteriori e posteriori. [6, p. 87] Una rappresentazione della posizione della lingua nella cavità orale è data dalla figura 2.1, [7, p. 133] ad ogni posizione corrisponde una vocale cardinale diversa.

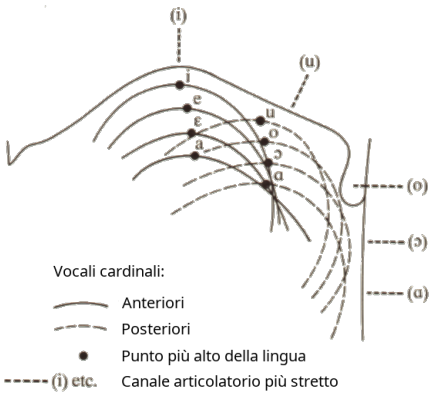
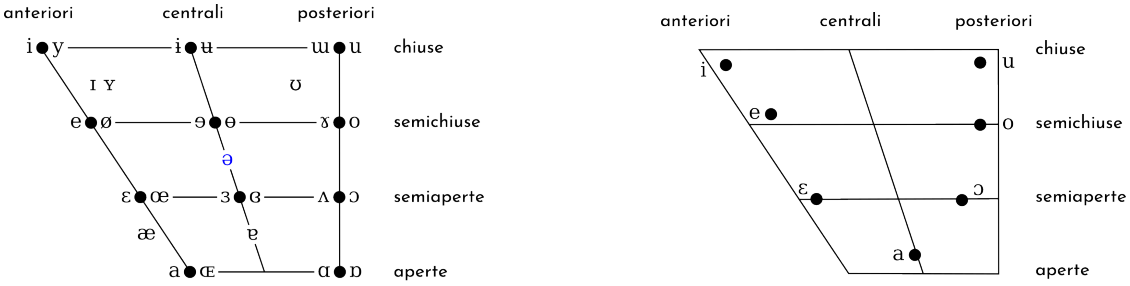


Figura 2.1: Le vocali cardinali prodotte in seguito allo spostamento della lingua.

Questa classificazione delle vocali, basata sullo spostamento della lingua e sull'arrotondamento labiale, [8, p. 40] è stata introdotta da Bell[1] e successivamente perfezionata da Jones nel modello delle vocali cardinali.[2] Una vocale si definisce cardinale quando, mentre viene prodotta, la lingua è alla massima distanza possibile dalla posizione che assume nella realizzazione della vocale centrale ə. [6, p. 89]

Collegando i punti più alti raggiunti dalla lingua durante la pronuncia delle vocali cardinali si ottiene lo schema trapezoidale riportato in figura 2.2[3][4], scelto per rappresentare tutte le vocali nell'alfabeto fonetico internazionale (più comunemente noto con l'acronimo inglese IPA, *International Phonetic Alphabet*), un sistema di trascrizione che associa ad ogni fonema un carattere. Lo schema riporta la classificazione definita da Bell. In riferimento ad esso, le vocali cardinali sono quelle situate sul perimetro del trapezio, fra cui compaiono le vocali usate in lingua italiana. [6, p. 113]



(a) Versione internazionale, in blu la vocale centrale ə.

(b) Versione italiana.

Figura 2.2: Lo schema trapezoidale delle vocali secondo lo standard IPA. Le vocali a destra dei punti sono arrotondate, quelle a sinistra non arrotondate.

2.1.1 Formanti

Una determinata conformazione della cavità orale fornisce delle specifiche risonanze alla voce, dette *formanti*. Nella voce umana si possono individuare diverse formanti, identificate da una frequenza e da una larghezza di banda espresse in

[1] A. M. Bell. *Visible Speech: the Science of Universal Alphabetics.* Simpkin-Marshall & Co, London, 1877.

[2] D. Jones. *An Outline of English Phonetics.* Heffer, Cambridge, 1957.

[3] *IPA vowel chart* (`https://commons.wikimedia.org/wiki/File:IPA_vowel_chart.svg`), ultimo accesso: 18 settembre 2023.

[4] *Italian vowel chart* (`https://commons.wikimedia.org/wiki/File:Italian_vowel_chart.svg`), ultimo accesso: 18 settembre 2023.

Hz, e indicate come F_1, F_2, F_3 e via dicendo (spesso, per semplicità, ci si riferisce a una formante alludendo solo alla sua frequenza).

Ogni risonanza corrisponde a un punto di maggiore intensità nello spettro del segnale in corrispondenza della relativa frequenza, la cui altezza di picco è data dal rapporto tra la frequenza e la larghezza di banda. [9]. Una specifica combinazione di questi picchi descrive un comportamento spettrale che corrisponde univocamente al fono prodotto da tali risonanze. A titolo di esempio, in figura 2.3 è riportato lo spettrogramma delle sette vocali italiane. Segue una tabella con le relative formanti usate come riferimento per il riconoscimento nel programma. [6, p. 158-163]

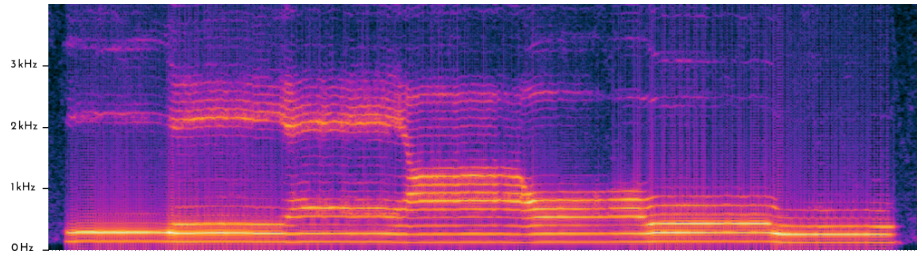


Figura 2.3: Lo spettrogramma delle vocali i, e, ɛ, a, ɔ, o, u. Le aree più chiare rappresentano una maggiore intensità dello spettro.

Vocale	i	e	ɛ	a	ɔ	o	u
F_1 (Hz)	280	360	560	800	520	360	280
F_2 (Hz)	2240	2040	1840	1280	920	800	720

Per riconoscere un fonema, quindi, occorre individuare le sue formanti. È stato ampiamente dimostrato dalla comunità scientifica che, nel caso delle vocali, sono sufficienti le prime due formanti F_1 e F_2 a distinguere un fono dall'altro. [6, p. 159] [10] [11] Per convincersi di questa asserzione basta osservare la figura 2.4 [12] che riporta in un grafico cartesiano le regioni di esistenza delle principali vocali in funzione delle frequenze formanti.

Com'è possibile evincere non c'è quasi nessuna sovrapposizione tra le regioni, quindi una volta note le frequenze delle prime due formanti diventa immediata l'associazione alla vocale corrispondente. Possiamo notare una similitudine riguardo la collocazione delle vocali sul grafico delle regioni di esistenza e sul diagramma vocalico dell'IPA. [6, p. 166]

L'obiettivo di questa tesi sta nel calcolo in tempo reale delle formanti della voce, da cui poi individuare la vocale pronunciata. Come si vedrà più avanti, questo richiede l'uso di una procedura nota come *Analisi predittiva lineare*

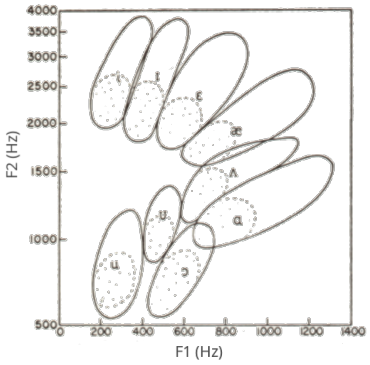


Figura 2.4: Le regioni di esistenza delle vocali, in funzione delle prime due formanti.

dalla quale si ricava una tupla ordinata di coefficienti reali corrispondenti alle caratteristiche spettrali di una breve porzione di segnale. Da questi coefficienti è possibile disegnare un grafico che approssima la FFT e individuare i punti di massimo — ovvero le frequenze formanti, per come sono state definite — o, in alternativa, procedere per via analitica e ricavare così i valori delle frequenze formanti. È stato scelto il secondo metodo per peculiarità di realizzazione e perché più preciso. [13, p. 450]

2.2 Il modello di Fant

Con l'assunzione di analizzare brevi intervalli di segnale vocale (dell'ordine di poche decine di millisecondi) in cui i parametri di eccitazione e risonanza sono pressoché costanti, [13, p. 98] è possibile ricavare un modello lineare e tempo invariante del comportamento della voce umana che, per sua natura, è un processo variabile nel tempo e non lineare. Uno dei modelli di maggior successo nell'ambito della fonetica articolatoria, nonché quello su cui si basa la teoria della codifica predittiva, [14, p. 1] è il modello lineare di produzione del parlato sviluppato da Gunnar Fant nel 1960.[5]

Esso è rappresentato in figura 2.5: l'ingresso, detto anche sorgente di eccitazione, può essere una sequenza di impulsi modulati dalla glottide, oppure un segnale di rumore, a seconda che si voglia pronunciare un fonema con o senza sonorità. Tale segnale di ingresso viene poi filtrato e articolato dai vari organi del tratto epilaringeo, ciascuno dei quali viene rappresentato da un sistema lineare con una sua funzione di trasferimento, e il segnale risultante è la voce udibile e comprensibile all'orecchio. Il sistema complessivo è un filtro, indicato con la fun-

[5] G. C. M. Fant. *Acoustic Theory of Speech Production.* Mouton and Co., 's-Gravenhage, The Netherlands, 1960.

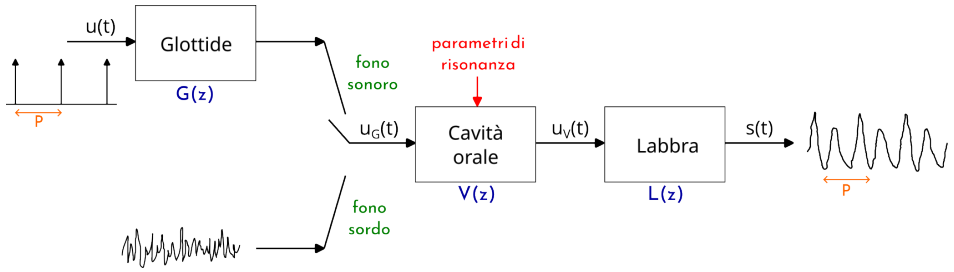


Figura 2.5: Il modello di produzione del parlato di Fant

zione di trasferimento $H(z)$, che trasforma un segnale di eccitazione in un fono, le cui caratteristiche risiedono dunque nei parametri del filtro illustrati di seguito.

2.2.1 Eccitazione

Analizziamo solo il processo di produzione dei foni sonori (come lo sono i vocoidi), che avviene quando la glottide, lo spazio che si crea fra le due pliche vocali situate nella laringe, [8, p. 33] è in posizione di sonorità. Le pliche vibrano a una determinata frequenza, che scandisce gli impulsi della sorgente di eccitazione e definisce l'altezza della voce. Nei segnali audio è identificata come *frequenza fondamentale* o *prima armonica* [6, p. 119] e indicata con F_0; corrisponde all'inverso del periodo di pitch.[6] La glottide determina la quantità d'aria coinvolta nella fonazione e quindi l'intensità del suono. La funzione di trasferimento corrispondente $G(z)$ è un filtro passa-basso a due poli, con frequenza di taglio attorno ai 100 Hz: [14, p. 5]

$$G(z) = \frac{1}{1 - e^{-cT} z^{-1}} \tag{2.1}$$

2.2.2 Risonanza

Il segnale $u_G(t)$ passa poi per la cavità orale, che funge da cassa di risonanza e fornisce le principali connotazioni al fono, già descritte prima come frequenze formanti. Ciascuna di esse corrisponde a una coppia di poli della funzione di trasferimento $V(z)$. Il modello non è preciso, in quanto i suoni nasali e fricativi richiedono una rappresentazione sia di risonanza che anti-risonanza (poli e zeri) [13, p. 99] [15, p. 158], ma per la presente ricerca incentrata sullo studio delle

[6] L'altezza è una delle caratteristiche della voce analizzate da Soundrise. Per una trattazione approfondita sull'argomento si rimanda alla tesi del collega Andrea Zanetti.

vocali una funzione di trasferimento con soli poli è sufficiente.

$$V(z) = \frac{1}{\prod_{i=1}^{K}[1 - 2\mathrm{e}^{-c_i T}\cos(b_i T)z^{-1} + \mathrm{e}^{-2c_i T}z^{-2}]} \tag{2.2}$$

Il denominatore di $V(z)$ prevede K formanti, le cui frequenze e larghezze di banda sono, rispettivamente, $F_i = b_i/2\pi$ e $B_i = c_i/2\pi$. [14, p. 7] Un modello più accurato prevederebbe l'uso di un infinito numero di formanti, l'uso del fattore di correzione aiuterebbe a sopperire a questa approssimazione, tuttavia secondo Rabiner in una rappresentazione digitale questo fattore può essere ignorato.[7] In figura 2.6 [16, p. 106] vediamo rappresentato il contributo di una formante (corrispondente a una coppia di poli complessi coniugati) nello spettro del segnale vocale.

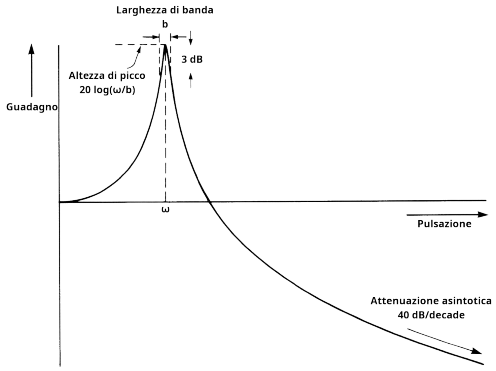


Figura 2.6: La risposta in frequenza di una formante. La somma di tutti i contributi dovuti a ciascuna formante fornisce una buona approssimazione dello spettro del segnale.

2.2.3 Articolazione finale

L'ultimo apparato del tratto epilaringeo corrisponde alle labbra, che forniscono al segnale $u_V(t)$ l'articolazione modellata da $L(z)$:

$$L(z) = 1 - z^{-1} \tag{2.3}$$

2.2.4 Modello complessivo

Unendo i diversi contributi, possiamo ottenere la funzione di trasferimento dell'intero filtro $H(z)$ che conta uno zero e $2K + 2$ poli:

[7] L. R. Rabiner. Digital-Formant Synthesizer for Speech-Synthesis. *Journal of the Acoustical Society of America*, 43:822-828, 1968.

$$\begin{aligned} H(z) &= G(z)V(z)L(z) \\ &= \frac{(1-z^{-1})}{(1-\mathrm{e}^{-cT}z^{-1})^2\left\{\prod_{i=1}^{K}[1-2\mathrm{e}^{-c_iT}\cos(b_iT)z^{-1}+\mathrm{e}^{-2c_iT}z^{-2}]\right\}} \end{aligned} \tag{2.4}$$

Considerando il termine al denominatore $(1-\mathrm{e}^{-cT}z^{-1})$ approssimato a $(1-z^{-1})$ si ha la cancellazione dello zero e di un polo del modello glottidale. Ne risulta una funzione di trasferimento con soli poli, corrispondenti agli zeri del denominatore che definiamo $A(z)$. Possiamo dunque scrivere la relazione tra il segnale di eccitazione e il segnale vocale come segue:

$$S(z) = H(z)U(z) = \frac{1}{A(z)}U(z) \tag{2.5}$$

Questa formulazione di $H(z)$ è particolarmente conveniente per l'analisi predittiva lineare. Rappresenta un filtro di ordine $2K+1$

$$A(z) = 1 - \sum_{i=1}^{q} a_i z^{-i} \qquad q = 2K \tag{2.6}$$

Il modello qui presentato può essere adoperato per due finalità diverse: la sintesi del parlato, utile a produrre dei segnali vocali artificiali impostando adeguatamente i parametri del sistema, e l'analisi del parlato — ovverosia il processo inverso a quello della sintesi — che consente la stima dei parametri relativi a un segnale vocale reale. Noto lo scopo di questa ricerca, i successivi riferimenti al modello di Fant saranno finalizzati all'ottenimento dei parametri (in particolare quelli relativi al tratto vocalico) che renderanno possibile a un elaboratore il riconoscimento automatico di un vocoide pronunciato.

Capitolo 3

Codifica predittiva lineare

La codifica predittiva lineare (in inglese *Linear predictive coding*, abbreviato LPC) è una tecnica di analisi e codifica del segnale vocale introdotta nella fine degli anni '60. [14, p. 18] La sua peculiarità sta nel poter estrarre delle informazioni relative al dominio della frequenza senza effettuare un'esplicita trasformazione del segnale in tale dominio.

Il suo funzionamento, come suggerisce il nome, si basa su un modello di predizione lineare applicato all'ampiezza dei campioni del segnale audio, o meglio, alla loro variazione nel tempo. Ad alte frequenze di campionamento possiamo supporre che i campioni audio acquisiti a breve distanza di tempo non presentino grandi differenze l'uno dall'altro. Possiamo quindi approssimare il valore di un campione $s(n)$ con una combinazione lineare di un finito numero di campioni precedenti

$$\begin{aligned} \tilde{s}(n) &= \alpha_1 s(n-1) + \alpha_2 s(n-2) + ... + \alpha_p s(n-p) \\ &= \sum_{k=1}^{p} \alpha_k s(n-k) \end{aligned} \tag{3.1}$$

commettendo un errore di predizione, definito come

$$e(n) = s(n) - \tilde{s}(n) = s(n) - \sum_{k=1}^{p} \alpha_k s(n-k) \tag{3.2}$$

In figura 3.1 [16, p. 148] viene illustrato in maniera intuitiva il concetto di errore di predizione. I coefficienti α che garantiscono la miglior approssimazione dei campioni (ovvero con il minor errore di predizione), nell'ambito di una ridotta porzione del segnale, sono detti coefficienti di predizione lineare (in inglese *Linear*

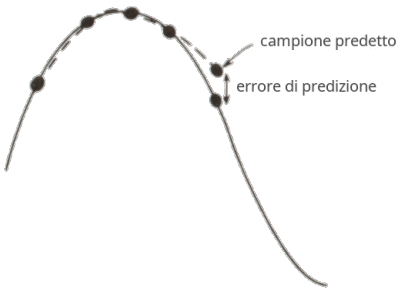


Figura 3.1: Esempio di segnale con campioni ottenuti dalla predizione lineare.

predictive coefficients) e codificano un segmento del segnale audio. [16, p. 124] Per il momento ci limitiamo a considerare l'ordine di predizione p come un grado di precisione, successivamente assumerà un diverso significato più attinente allo scopo della presente ricerca.

Questa tecnica di compressione risulta particolarmente efficace per il salvataggio e la trasmissione in tempo reale di un segnale vocale, visto il basso numero di bit richiesti dal segnale codificato. Come illustrato nella figura 3.2 (dove si è scelto un segnale di ingresso campionato a 10 kHz) a venire trasmesso è il segnale di errore, che per sua natura non rappresenta numeri di ordine elevato e occupa poca banda. I coefficienti LPC invece possono venire trasmessi ad intervalli di tempo più lunghi, questo perché nel parlato la pronuncia di un determinato fonema (le cui informazioni per la decodifica risiedono nei coefficienti α) solitamente dura dai 10 ai 20 ms, [13, p. 105] e sarebbe ridondante il calcolo e la trasmissione a frequenze più elevate.

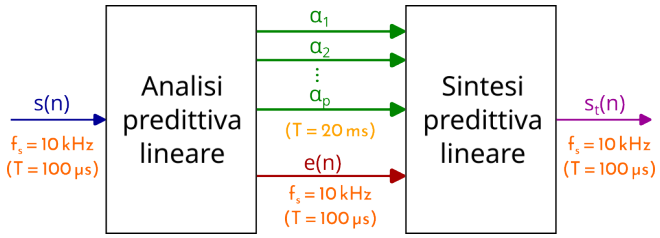


Figura 3.2: Schema basilare di una tipica applicazione della codifica predittiva lineare.

3.1 Relazione con il modello fisico

Come accennato in precedenza, la codifica predittiva lineare si appoggia al modello di produzione del parlato descritto nel capitolo 2, pertanto è opportuno portarci nel dominio della trasformata zeta. Applicando la trasformazione all'equazione 3.2 relativa all'errore di predizione, otteniamo

$$E(z) = S(z) - \sum_{k=1}^{p} \alpha_k z^{-k} S(z) \tag{3.3}$$

Possiamo raccogliere $S(z)$ ed esprimerlo in funzione dell'errore:

$$S(z) = \frac{1}{1 - \sum_{k=1}^{p} \alpha_k z^{-k}} E(z) \tag{3.4}$$

A questo punto non possiamo non notare un'importante analogia tra la funzione di trasferimento che lega $E(z)$ a $S(z)$ e la funzione di trasferimento finale del modello di Fant (il cui denominatore è espresso nell'equazione 2.6). A patto di definire l'ordine $p = q$, i coefficienti $\alpha_k = a_i$ e identificare l'eccitazione di ingresso $U(z)$ con l'errore di predizione $E(z)$, i due sistemi sono equivalenti.

Tale corrispondenza ci porta a concludere che i coefficienti α_k che verranno calcolati con la tecnica di analisi predittiva (accertandosi di scegliere un ordine di predizione p significante) saranno i parametri di risonanza della cavità orale.

Il modello di produzione del parlato verrà usato, tramite la tecnica della predizione lineare, per effettuare un'analisi del segnale vocale da cui si otterranno i coefficienti a_i relativi al fono pronunciato. In riferimento alla figura 2.5, lo schema — inizialmente pensato per una produzione sonora — verrà percorso al contrario nel metodo che andremo a illustrare: il segnale $s(t)$ sarà l'ingresso (il flusso audio proveniente dal microfono) e attraverso il procedimento ci si ricondurrà al segnale $u(t)$ (nel nostro caso l'errore di predizione). Alla fine del processo verranno ottenuti i parametri di risonanza da cui ricavare le frequenze formanti.

La codifica predittiva lineare risulta efficace perché, nel modellare lo spettro del parlato con un modello a soli poli, rispecchia i punti di forza dell'apparato uditivo umano che è più sensibile all'individuazione delle risonanze (i poli del modello) che delle anti-risonanze (gli zeri). [17, p. 79]

3.2 Determinazione dei coefficienti di predizione

In letteratura esistono diversi metodi di calcolo dei coefficienti LPC che variano principalmente per approcci distinti al problema, i quali portano a procedure risolutive utilizzanti differenti strumenti matematici. I principali metodi, denotati appunto dal procedimento matematico di cui fanno uso, sono tre:

- Metodo dell'autocorrelazione

- Metodo della covarianza
- Metodo del filtro reticolare

Da questi derivano altre formulazioni, che presentano poche differenze strutturali dai metodi elencati. [13, p. 397] In questa sezione verrà introdotto il problema da cui si diramano i vari metodi risolutivi.

Per rispettare le ipotesi di tempo invarianza e lavorare su un breve intervallo di tempo, dobbiamo definire un segmento del parlato $s_n(m)$ che proviene dal segnale completo: inizia dal campione n e indicizza il campione $n + m$.

$$s_n(m) = s(n + m) \tag{3.5}$$

In questa sezione viene definito solo il punto iniziale del segmento giacché i diversi metodi operano in maniera differente sul suo punto finale.

Un elemento chiave della computazione è la minimizzazione dell'errore quadratico medio MSE (sempre nell'ambito di un breve intervallo di segnale), questo per avere un basso errore di predizione e una codifica più efficiente del segnale $e(n)$ in caso di trasmissione. Definiamo l'errore quadratico medio come

$$MSE = \sum_m e_n(m)^2 = \sum_m \left[s_n(m) - \sum_{k=1}^{p} \alpha_k s_n(m-k) \right]^2 \tag{3.6}$$

e, per minimizzarlo, lo deriviamo in base ai coefficienti α_k e poniamo il risultato delle derivate parziali a 0. Otterremo una serie di p equazioni:

$$\frac{\partial MSE}{\partial \alpha_j} = 0 \qquad j = 1, 2, ..., p \tag{3.7}$$

$$-2 \sum_m s_n(m-j) \left[s_n(m) - \sum_{k=1}^{p} \alpha_k s_n(m-k) \right] = 0 \tag{3.8}$$

$$\sum_{k=1}^{p} \alpha_k \sum_m s_n(m-j) s_n(m-k) = \sum_m s_n(m) s_n(m-j) \tag{3.9}$$

$$\sum_{k=1}^{p} \alpha_k \phi_n(j,k) = \phi_n(j,0) \tag{3.10}$$

L'espressione 3.10 è stata ottenuta sostituendo la funzione di autocorrelazione ϕ definita come:

$$\phi_n(a,b) = \sum_m s_n(m-a) s_n(m-b) \tag{3.11}$$

Per ottenere il set di coefficienti che minimizza l'errore bisogna calcolare i valori dell'autocorrelazione $\phi_n(a, b)$ per $1 \leq a \leq p$ e $0 \leq b \leq p$, e successivamente risolvere il sistema lineare di p equazioni con altrettante incognite definite dall'equazione 3.10.

3.3 Metodo dell'autocorrelazione

In questa sezione verrà illustrato il procedimento noto come *metodo dell'autocorrelazione*. Essendo questa la soluzione scelta per l'implementazione, i seguenti passaggi verranno ripresi nel capitolo successivo che descrive l'algoritmo.

3.3.1 Operazioni preliminari

Occorre fare alcune assunzioni rispetto alla formulazione del problema enunciata nella precedente sezione. Il segmento del segnale preso in esame $s_n(m)$ dev'essere nullo all'esterno dell'intervallo $1 \leq m \leq N-1$, con N lunghezza del segmento. [13, p. 401]

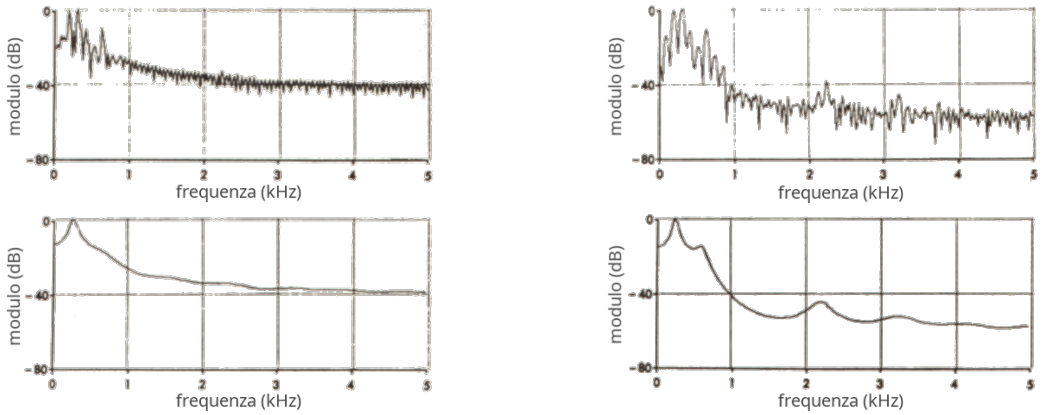
Questo equivarrebbe all'applicazione di una finestra rettangolare (una funzione uguale a 1 in un determinato intervallo e a 0 al di fuori di esso), tuttavia è stato dimostrato che una finestra come quella di Hamming (definita in 3.12) è più appropriata per una successiva analisi delle formanti. [18]

$$w(n) = \begin{cases} 0.54 - 0.46\cos(\frac{2\pi n}{N-1}), & 0 \leq n \leq N-1 \\ 0, & \text{altrimenti} \end{cases} \tag{3.12}$$

La finestra di Hamming, applicata direttamente al segnale nel dominio del tempo, attenua gradualmente l'ampiezza dei campioni man mano che ci si avvicina ai margini di essa. Un esempio di tale attenuazione può essere apprezzato nel primo grafico della figura 3.4. L'uso di questa finestra porta dei vantaggi anche all'analisi nel dominio della frequenza, infatti permette una miglior distinzione delle formanti come evidenziato in figura 3.3. [18]

La nuova definizione del segmento di segnale da prendere in esame è

$$s_n(m) = s(n+m)w(m) \tag{3.13}$$



(a) Uso della finestra rettangolare

(b) Uso della finestra di Hamming

Figura 3.3: Raffronto degli effetti sullo spettro del segnale (originale e approssimato) dopo l'applicazione di una finestra rettangolare e di una finestra di Hamming.

3.3.2 Formulazione

Riscriviamo anche la definizione dell'errore quadratico medio aggiungendo il limite superiore alla sommatoria. Partendo da un numero di campioni pari a N, l'errore di predizione di ordine p è definito anche per i successivi p campioni:

$$MSE = \sum_{m=0}^{(N-1)+p} e_n(m)^2 \tag{3.14}$$

Da qui risulta più chiaro l'effetto della finestra di Hamming: come da 3.2, l'errore è definito come una differenza di campioni consecutivi del segnale. In prossimità dei limiti dell'intervallo, se non viene applicata alcuna attenuazione, campioni del segnale originale e campioni moltiplicati per 0 andranno a produrre un errore elevato (dell'ordine dei campioni del segnale originale). Con una progressiva attenuazione del segnale in prossimità degli estremi dell'intervallo si evita un aumento in modulo dell'errore di predizione in tali punti.

Aggiungiamo il limite alla sommatoria anche alla definizione della funzione di autocorrelazione

$$\phi_n(a,b) = \sum_{m=0}^{(N-1)+p} s_n(m-a)s_n(m-b) \tag{3.15}$$

e, sapendo che il segmento di ingresso è nullo al di fuori dell'intervallo $[0, N-1]$,

possiamo esprimere ϕ_n come segue:

$$\phi_n(a,b) = \sum_{m=0}^{(N-1)-(a-b)} s_n(m)s_n(m+a-b) \tag{3.16}$$

Introduciamo la funzione di autocorrelazione a tempo breve:

$$R_n(k) = \sum_{m=0}^{(N-i)-k} s_n(m)s_n(m+k) \tag{3.17}$$

notando che è una funzione pari, può essere sostituita a ϕ_n con la seguente relazione:

$$\phi_n(a,b) = R_n(|a-b|) \tag{3.18}$$

e possiamo esprimere l'equazione risolutiva 3.10 come

$$\sum_{k=1}^{p} \alpha_k R_n(|j-k|) = R_n(j) \qquad j = 1,2,...,p \tag{3.19}$$

Anche in questo caso, una volta risolta l'equazione 3.17 calcolando R_n partendo dai campioni del segmento di segnale, si possono ottenere i coefficienti α dalle p equazioni definite dalla 3.19 ed espresse in forma matriciale come di seguito:

$$\begin{bmatrix} R_n(0) & R_n(1) & R_n(2) & ... & R_n(p-1) \\ R_n(1) & R_n(0) & R_n(1) & ... & R_n(p-2) \\ R_n(2) & R_n(1) & R_n(0) & ... & R_n(p-3) \\ ... & & & & ... \\ R_n(p-1) & R_n(p-2) & R_n(p-3) & ... & R_n(0) \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \\ ... \\ \alpha_p \end{bmatrix} = \begin{bmatrix} R_n(1) \\ R_n(2) \\ R_n(3) \\ ... \\ R_n(p) \end{bmatrix} \tag{3.20}$$

3.3.3 Metodo di Durbin

Osservando che quella presentata nell'equazione 3.20 è una matrice di Toeplitz (tutte le diagonali contengono lo stesso elemento), ci si può ricondurre al metodo ricorsivo di Durbin per risolvere efficientemente il sistema. Questa soluzione è stata proposta inizialmente da Levinson[1] e in seguito rielaborata da Robinson.[2] [18]

[1] N. Levinson. The Wiener RMS (root mean square) error criterion in filter design and prediction. *J. Math. Phys*, 25(4):261-278, 1947.

[2] E. A. Robinson. *Statistical Communication and Detection.* Hafner, New York, 1967.

L'algoritmo provvede a calcolare i coefficienti α facendo progressive approssimazioni ad ogni iterazione. A venire gradualmente approssimati sono i coefficienti e l'errore E: tali variabili sono denotati con un numero in apice che specifica l'indice dell'iterazione da considerare quando vi si riferisce. Dopo l'assegnazione di partenza $E^{(0)} = R_n(0)$ si iterano i seguenti passi per $1 \leq i \leq p$

1. $k_i = \frac{R_n(i) - \sum_{j=1}^{i-1} \alpha_j^{(i-1)} R_n(i-j)}{E^{(i-1)}}$
2. $\alpha_i^{(i)} = k_i$
3. $\alpha_j^{(i)} = \alpha_j^{(i-1)} - k_i \alpha_{i-j}^{(i-1)}$ per $1 \leq j \leq i-1$
4. $E^{(i)} = (1 - k_i^2) E^{(i-1)}$

Alla fine delle iterazioni, i coefficienti α cercati sono le relative stime finali: $\alpha_j = \alpha_j^{(p)}$

I coefficienti k_i sono detti di *correlazione parziale* (o PARCOR) e hanno un ruolo importante per assicurare la stabilità del sistema. Se vale

$$-1 \leq k_i \leq 1 \tag{3.21}$$

è automaticamente garantita la stabilità del sistema, ovvero tutte le radici complesse derivanti dalla fattorizzazione del polinomio dei coefficienti LPC sono situate, nel piano complesso, all'interno del cerchio con raggio unitario. [13, p. 419]

3.4 Scelta dei parametri

Il numero di coefficienti, indicato con p, è arbitrario e può essere definito in base al livello di precisione richiesto, tuttavia il valore ottimale dipende dalla frequenza di campionamento secondo la relazione $p = f_s + \gamma$, con f_s espressa in kHz e $\gamma = 4$ o 5. Questo per garantire un numero sufficiente di poli per rappresentare sia le formanti presenti nella banda di campionamento (una coppia di poli copre ragionevolmente un raggio di 700 Hz) [18] che altri contributi di articolazione provenienti dal modello fisico. [19, p. 346] Si fa notare in figura 3.4 [19, p. 347] la rilevanza di un valore di p sufficientemente alto per l'affioramento delle formanti.

Anche la lunghezza ottimale del segmento da analizzare (in termini di numero di campioni) dipende linearmente dalla frequenza di campionamento: $N = \delta f_s$

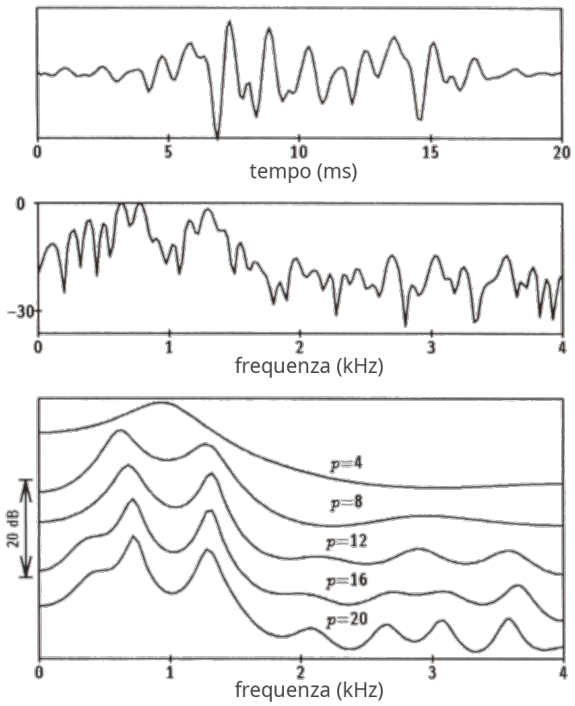


Figura 3.4: In alto il segnale audio da analizzare, al centro lo spettro del segnale, in basso l'approssimazione dello spettro calcolata dai coefficienti LPC con un diverso ordine di predizione.

con δ ideale nell'intervallo da 20 a 35. [18] In particolare, per il metodo dell'autocorrelazione, è richiesto che sia dell'ordine di qualche periodo di pitch per ottenere risultati affidabili, ma sufficientemente breve per ottimizzare il tempo di computazione. [13, p. 420]

Capitolo 4

Descrizione dell'algoritmo

In questo capitolo verrà descritto l'algoritmo di riconoscimento nella sua interezza. Il programma è stato scritto in linguaggio JavaScript, sfruttando la libreria Web Audio Api. In vista dell'inclusione nel progetto Soundrise il codice risiede in un file a sé stante ed è predisposto per essere incluso e utilizzato da un ambiente esterno.

Ad ogni invocazione della funzione `getVowel()` il programma calcola le prime due frequenze formanti della voce e le confronta con quelle caratteristiche delle vocali cardinali italiane, al fine di trovare un'eventuale corrispondenza.

4.1 Ambiente esterno

Iniziamo con la presentazione di un esempio di ambiente che andrà a includere e usare la funzione `getVowel()`. Il seguente codice JavaScript, quando viene invocata la funzione `start()`, effettua una richiesta per ottenere l'accesso al canale audio del microfono del dispositivo. Se la richiesta (inoltrata all'utente dal browser) va a buon fine, la funzione imposta il flusso come sorgente dell'istanza `AudioContext` salvata nella variabile `ctx`, e la connette a un analizzatore. Quest'ultimo elemento serve ad estrarre i campioni PCM del segnale su cui lavorerà l'algoritmo di riconoscimento.

```
1  let update = 100;
2
3  function start() {
4    let ctx = new AudioContext({latencyHint: "interactive"});
5    // Ottenimento dell'audio dal microfono
6    navigator.mediaDevices
7    .getUserMedia({audio: {
```

```
8       echoCancellation: false,
9       noiseSuppression: false,
10      autoGainControl: false,
11      highpassFilter: false
12    }})
13    .then((stream) => { // Flusso audio ottenuto
14      const source = ctx.createMediaStreamSource(stream);
15      const analyser = ctx.createAnalyser();
16      source.connect(analyser);
17      analyser.fftSize = 32768;
18      let signal = new Float32Array(analyser.fftSize);
19
20      setInterval(function () {
21        analyser.getFloatTimeDomainData(signal);
22        console.log(getVowel(ctx.sampleRate, signal));
23      }, update);
24    })
25    .catch((err) => {
26      console.log("Permesso negato: " + err);
27    });
28  }
```

Si fa notare l'assegnazione del valore $32768 = 2^{15}$ all'attributo `fftSize`, che definisce il numero di campioni restituiti dal metodo `getFloatTimeDomainData()`: tale cifra è il massimo valore assegnabile[1] ed è richiesto un alto valore di campioni dalla computazione, siccome è prevista un'operazione di sotto-campionamento.

Dopo l'inizializzazione, viene invocata automaticamente la funzione `getVowel()` a cui si passa il segnale di ingresso su cui lavorare. Dopo varie prove si è individuato il valore ottimale di 100 ms come durata dell'intervallo ogni cui effettuare la chiamata, tuttavia può essere cambiato in base a diverse esigenze di precisione o efficienza.

4.2 Dichiarazioni iniziali

Passiamo alla presentazione del file in cui sono situate le funzioni per il riconoscimento timbrico. In questa sezione leggiamo le dichiarazioni di alcune variabili utili (opportunamente descritte dai commenti nel codice) e della funzione `getVowel()` invocata dall'ambiente esterno.

[1] *AnalyserNode: fftSize property* (`https://developer.mozilla.org/en-US/docs/Web/API/AnalyserNode/fftSize`), ultimo accesso: 17 settembre 2023.

```
var p = 15; // Ordine di predizione
var N = 256; // Lunghezza della porzione di segnale da considerare
var fs; // Frequenza di campionamento originale
var fu = 10000; // Frequenza ottimale a cui sotto-campionare (
    espressa in Hz)
var signal = new Float32Array(32768);
// Formanti maschili per I, E (acuto), E (grave), A, O (grave), O (
    acuto), U
// I valori a margine (primo e ultimo) sono inesistenti: servono
    solo per agevolare il calcolo dei range nell'algoritmo
var form1 = [100,  280,  360,  560,  800,  520, 360, 280, 100];
var form2 = [2460, 2240, 2040, 1840, 1280, 920, 800, 720, 600];
var vocali = ['I', '&Eacute;', '&Egrave;', 'A', '&Ograve;', '&Oacute
    ;', 'U'];
var signal;

function getVowel(s, sampleRate) {
  signal = s;
  fs = sampleRate;
  let R = autocorrelation();
  let lpc = durbin(R);
  let roots = durand(lpc);
  let valid = formants(roots, fu);
  return compare(valid);
}
```

4.3 Sotto-campionamento e finestra

I primi passaggi da descrivere sono quelli che convertono il segnale di origine in quello su cui si andrà a operare. Tali passaggi sono un sotto-campionamento alla frequenza `f1` e l'applicazione della finestra di Hamming descritta nella sezione 3.3.1.

Per una scelta di ottimizzazione, entrambe queste funzioni non operano mai sull'intero segnale, ma eseguono il calcolo solo per i campioni richiesti all'atto della chiamata. In particolare la funzione `efficientUs()` avrebbe una complessità di $O(n)$: siccome il calcolo dell'autocorrelazione a tempo breve la invocherebbe più volte per la restituzione dello stesso campione (vedasi sezione successiva), una volta che è stato calcolato, esso viene salvato nell'array `usx` per essere restituito con complessità costante alle successive chiamate della funzione.

```
let usx;
```

```
2  function efficientUs(i) {
3    if (usx[i] == 0) {
4      let ratio = fs / fu;
5
6      usx[i] = 0;
7      let index = Math.floor(i * ratio);
8
9      for (let j = 0; j < ratio; j++) {
10       usx[i] += parseFloat(signal[index + j]) * ham(N, i);
11     }
12     usx[i] /= ratio;
13   }
14   return usx[i];
15 }
16
17 function ham(N, i) {
18   return 0.54 - 0.46 * Math.cos((2 * Math.PI * i) / (N - 1));
19 }
```

4.4 Autocorrelazione a tempo breve

La seguente funzione calcola il l'autocorrelazione a tempo breve come definita nell'equazione 3.17.

Notare che l'algoritmo usa il segnale sotto-campionato (i cui singoli campioni sono restituiti dalla funzione `effcientUs()` definita poc'anzi) invece di quello originale.

```
1  function autocorrelation() {
2    usx = new Float32Array(signal.length * fu / fs);
3    var R = new Float32Array(p + 1);
4
5    for (var k = 0; k <= p; k++) {
6      R[k] = 0;
7      for (var m = 0; m <= N - 1 - k; m++) {
8        R[k] += efficientUs(m) * efficientUs(m + k);
9      }
10   }
11   return R;
12 }
```

4.5 Coefficienti LPC

Una volta calcolati i valori di autocorrelazione si possono ricavare i coefficienti LPC tramite il metodo di Durbin definito nella sezione 3.3.3.

```
function durbin(R) {
  let lpc = [];
  let alpha = [];
  let k = [];
  let E = R[0];

  // Passi iterativi
  for (let i = 1; i <= p; i++) {
    k[i] = R[i];
    for (let j = 1; j <= i - 1; j++) {
      k[i] -= alpha[j][i - 1] * R[i - j];
    }
    k[i] /= E;
    alpha[i] = [];
    alpha[i][i] = k[i];
    for (let j = 1; j <= i - 1; j++) {
      alpha[j][i] = alpha[j][i - 1] - k[i] * alpha[i - j][i - 1];
    }
    E = (1 - k[i] * k[i]) * E;
  }

  // Stime finali dei coefficienti da restituire
  for (let i = 0; i < p; i++) {
    lpc[i + 1] = -alpha[i + 1][p];
  }
  lpc[0] = 1;
  return lpc;
}
```

4.6 Estrazione delle formanti

L'estrazione delle formanti richiede la fattorizzazione del polinomio $A(z)$ nelle sue radici complesse, che viene effettuata con il metodo di Durand-Kerner. Questo algoritmo, partendo dalle *deg* (il grado del polinomio da fattorizzare) radici complesse di 1 facilmente calcolabili, esegue un finito numero di iterazioni in

cui aggiorna i valori di queste radici per avvicinarle gradualmente a quelle del polinomio.[2]

```
function durand(cf) {
  const deg = cf.length - 1; // Grado del polinomio
  const n = 8; // Numero di iterazioni

  // Inizializzazione dalle radici dell'unita'
  var roots = [];
  for (let i = 0; i < deg; i++) {
    const theta = (2 * Math.PI * i) / deg;
    const root = { real: Math.cos(theta), imag: Math.sin(theta) };
    roots[i] = root;
  }

  for (let i = 0; i < n; i++) {
    var preroots = roots;
    for (let j = 0; j < deg; j++) {
      // Valutazione del polinomio con regola di Horner
      var p = { real: cf[0], imag: 0 };
      for (let k = 1; k <= deg; k++) {
        p = sumc(mulc(p, preroots[j]), { real: cf[k], imag: 0 });
      }

      var div = { real: 1, imag: 0 };
      for (let k = 0; k < deg; k++) {
        if (j != k) {
          div = mulc(div, subc(preroots[j], preroots[k]));
        }
      }
      roots[j] = subc(preroots[j], divc(p, div));
    }
  }

  return roots;
}

// Funzioni aritmetiche per i numeri complessi
function sumc(a, b) {
  return { real: a.real + b.real, imag: a.imag + b.imag };
}
```

[2] P. Fraigniaud. The Durand-Kerner polynomials roots-finding method in case of multiple roots. *BIT Numerical Mathematics*, 31:112–123, 1991. https://doi.org/10.1007/BF01952788

```
39
40 function subc(a, b) {
41   return { real: a.real - b.real, imag: a.imag - b.imag };
42 }
43
44 function mulc(a, b) {
45   return {
46     real: a.real * b.real - a.imag * b.imag,
47     imag: a.real * b.imag + a.imag * b.real
48   };
49 }
50
51 function divc(a, b) {
52   const denominator = b.real * b.real + b.imag * b.imag;
53   return {
54     real: (a.real * b.real + a.imag * b.imag) / denominator,
55     imag: (a.imag * b.real - a.real * b.imag) / denominator
56   };
57 }
```

Dalle radici ottenute vengono calcolate le frequenze formanti (selezionando solo i valori validi) dalla funzione `formants()`. Gli intervalli di valori accettati sono $[200 \div 1600]$ Hz per F_1 e $[700 \div 3000]$ Hz per F_2. [20]

```
1  function formants(roots, fs) {
2    let ff = [];
3    for (let i = 0; i < roots.length; i++) {
4      let f = fs * Math.atan2(roots[i].imag, roots[i].real) / (2 *
           Math.PI);
5      let b = -fs * Math.log(Math.sqrt((roots[i].real ** 2) + (roots[i
           ].imag ** 2))) / Math.PI;
6      if (f >= 0 && b >= 0 && b <= 600) {
7        ff.push({ 'freq': f, 'band': b });
8      }
9    }
10   ff.sort((a, b) => a.freq - b.freq);
11
12   // Estrazione delle formanti valide
13   let valid = [0];
14   let j = 0;
15   let minval = [200, 700];
16   let maxval = [1600, 3000];
17   for (let i = 0; i < ff.length; i++) {
18     if (ff[i].freq >= minval[j] && ff[i].freq <= maxval[j]) {
19       valid[j + 1] = ff[i];
```

```
      j++;
    }
    if (j > 2)
    break;
  }
  return valid;
}
```

4.7 Comparazione

Infine, la funzione `compare()` confronta le frequenze formanti trovate al fine di trovare una corrispondenza. Se il confronto ottiene un risultato viene restituito il carattere della vocale corrispondente, altrimenti `null`.

```
function compare(valid) {
  if (valid.length < 2)
  return null;

  let i;
  for (i = 1; i <= 7; i++) {
    let max = form2[1 - 1] - ((form2[i - 1] - form2[i]) / 2);
    let min = form2[i + 1] + ((form2[i] - form2[i + 1]) / 2);
    if (valid[2].freq > min && valid[2].freq < max) {
      let max1 = form1[i] + 200;
      let min1 = form1[i] - 200;
      if (!(valid[1].freq > min1 && valid[1].freq < max1)) {
        i = 8;
      }
      break;
    }
  }

  if (i != 8)
  return vocali[i - 1];
  else
  return null;
}
```

Capitolo 5

Conclusioni

Il programma qui presentato può essere ancora perfezionato per un riconoscimento più preciso delle vocali. In primo luogo potrebbero essere svolti degli altri test variando i parametri di ordine di predizione p, lunghezza del segmento N e frequenza di sotto-campionamento `fu`.

In vista di un allenamento più accurato sarebbe opportuno adattare i valori delle formanti a cui il programma fa riferimento a quelle che l'utente effettivamente produce con la sua conformazione anatomica. È noto infatti che la dimensione del canale epilaringeo varia da persona a persona, e di conseguenza le risonanze possono divergere dai valori standard.

Questo adattamento può essere effettuato mediante un addestramento della macchina, oppure in seguito all'identificazione delle frequenze formanti della vocale centrale. Nel secondo caso serve riconoscere anche la terza formante e si sfrutta la proprietà distintiva di ə secondo la quale tutte le formanti sono equidistanti l'una dall'altra, ovvero F_2, F_3, F_4, ecc..., sono tutti multipli dispari di F_1. [6, p. 124] Nota la distanza in Hz tra due qualsiasi formanti di ə, si può applicare una proporzione per adattare i valori standard delle formanti a quelle del tratto epilaringeo dell'utente.

Ulteriori possibili orizzonti di sviluppo sono la sperimentazione di altri metodi, come la ricerca dei picchi nel grafico approssimato dello spettro del segnale (costruito dai coefficienti LPC) anche se non molto preciso, oppure il metodo di calcolo dei coefficienti LPC tramite la struttura del filtro reticolare, per il quale non serve applicare una finestra al segnale di ingresso. [13, p. 417]

Il riconoscimento delle vocali è attuabile anche analizzando la distanza euclidea dei coefficienti cepstrali (derivati dalla trasformata di Fourier dello spettro in scala logaritmica) da pattern preventivamente registrati. [17, p. 90]

Bibliografia

[1] M. Montessori. *Educare alla libertà.* Oscar Mondadori, Milano, 2014.

[2] A. Elmi. *Il sordo parziale: Aggiornamenti diagnostici e terapeutici sulla problematica della metasordastria.* Tipografia editrice La garangola, Padova, 1970.

[3] L. Pizzamiglio. *I disturbi del linguaggio - Manuale di diagnosi e terapia.* Etas Libri, 1977.

[4] S. Giusto. *Soundrise: studio e progettazione di un'applicazione multimodale interattiva per la didattica basata sull'analisi di feature vocali.* Tesi di laurea magistrale, Università di Padova, 2012.

[5] M. Randon. *Soundrise: sviluppo e validazione di un'applicazione multimodale interattiva per la didattica basata sull'analisi di feature vocali.* Tesi di laurea magistrale, Università di Padova, 2012.

[6] A. Giannini e M. Pettorino. *La fonetica sperimentale.* Edizioni scientifiche italiane, Napoli, 1992.

[7] J. C. Catford. *A Practical Introduction to Phonetics.* Clarendon Press, Oxford, 1994.

[8] L. Canepari. *Avviamento alla fonetica.* Piccola biblioteca Einaudi, Torino, 2006.

[9] C. G. M. Fant. On the Predictability of Formant Levels and Spectrum Envelopes from Formant Frequencies. *For Roman Jakobson*, pages 109–120, 1956.

[10] H. K. Dunn. The Calculation of Vowel Resonances, and an Electrical Vocal Tract. *Journal of the Acoustical Society of America*, 22:740–753, 1950.

[11] S. E. Estes, H. R. Kerby, H. D. Maxey e R. M. Walker. Speech Synthesis from Stored Data. *IBM J, Res. Develop.*, 8:2–12, 1964.

[12] K. N. Stevens e A. S. House. Development of a Quantitative Description of Vowel Articulation. *Journal of the Acoustical Society of America*, 27:484–493, 1955.

[13] L. R. Rabiner e R. W. Schafer. *Digital Processing of Speech Signals*. Prentice-Hall signal processing series. Prentice-Hall, Englewood Cliffs, New Jersey, 1978.

[14] J. D. Markel e A. H. Gray. *Linear Prediction of Speech*. Communications and Cybernetics. Springer Verlag, Berlin, 1976.

[15] J. I. Makhoul e J. J. Wolf. *Linear Prediction and the Spectral Analysis of Speech*. Bolt Beranek and Newman, 50 Moulton Street, Cambridge, Massachusetts, 1972.

[16] I. H. Witten. *Principles of Computer Speech*. Computers and People series. Academic Press, London, 1982.

[17] G. Bristow. *Electronic Speech Synthesis: Techniques, Technology, and Applications*. Granada, London, 1984.

[18] J. D. Markel. Digital Inverse Fitering - A New Tool for Formant Trajectory Estimation. *IEEE Trans. Audio Electroacoust.*, AU-20:129–137, 1972.

[19] D. O'Shaughnessy. *Speech Communication: Human and Machine*. Addison-Wesley series in electrical engineering: Digital signal processing. Addison-Wesley, Reading, Massachusetts, 1987.

[20] M. Matsumura, H. Yamane e K. Fujī. Recognition of Vowels and a Semivowel Using Formant Locus Extracted by Cubic Spline. *Electronics and Communications in Japan (Part 3: Fundamental Electronic Science)*, 72(11):73–86, 1989.