

ALMA MATER STUDIORUM · UNIVERSITÀ DI BOLOGNA

SCUOLA DI SCIENZE

Corso di Laurea in Informatica Magistrale

Analisi delle funzionalità respiratorie

**Monitoraggio della respirazione attraverso uno stetoscopio
elettronico**

Tesi di Laurea in Sistemi Mobili

Relatore:
Chiar.mo Prof.
Vittorio Ghini

Presentata da:
Federico Viscomi

Sessione III
Anno Accademico 2011/2012

Parole chiave: respirazione, pattern recognition, apprendimento automatico, monitoraggio, segnali biomedici.

Indice

| | | |
|----------|--|-----------|
| 1 | Introduzione | 6 |
| 1.1 | Scopo | 6 |
| 1.2 | Motivazioni | 6 |
| 1.3 | Contenuto della tesi | 7 |
| I | Prerequisiti | 8 |
| 2 | Respirazione e apparato respiratorio | 9 |
| 2.1 | Anatomia dell'apparato respiratorio | 9 |
| 2.2 | Respirazione | 9 |
| 2.2.1 | Meccanica della respirazione | 10 |
| 2.3 | Diagnosi delle malattie dell'apparato respiratorio | 11 |
| 2.3.1 | Spirometro con pneumotacografo | 11 |
| 2.3.2 | Polisonnografia | 11 |
| 2.3.3 | Auscultazione | 11 |
| 2.3.4 | Stetoscopio | 11 |
| 2.3.5 | Invasività clinica | 12 |
| 2.4 | Patologia dell'apparato respiratorio | 12 |
| 2.4.1 | Apnee del sonno | 12 |
| 2.4.2 | Arresto respiratorio | 14 |
| 2.5 | Base funzionale dei suoni respiratori | 14 |
| 2.6 | Analisi acustica dei suoni respiratori | 14 |
| 2.7 | Schemi di respirazione | 16 |
| 3 | Acustica e audio digitale | 18 |
| 3.1 | L'acustica e il suono | 18 |
| 3.2 | Onde sonore | 18 |
| 3.3 | Suono analogico e digitale | 19 |
| 3.3.1 | Discretizzazione | 20 |

| | | |
|-----------|--|-----------|
| 3.4 | Segnale | 21 |
| 3.5 | Analisi nel dominio del tempo e nel dominio della frequenza | 23 |
| 3.6 | Analisi armonica e trasformata di Fourier | 24 |
| 3.6.1 | Serie di Fourier | 24 |
| 3.6.2 | Trasformata di Fourier | 24 |
| 3.6.3 | Trasformata discreta di Fourier | 25 |
| 3.7 | Wavelet | 25 |
| 3.8 | Trasformata wavelet continua | 26 |
| 4 | Filtri digitali | 27 |
| 4.1 | Denoising | 27 |
| 4.2 | Classificazione dei filtri digitali | 27 |
| 4.2.1 | Categorie di filtri | 28 |
| 4.3 | Filtrare i suoni respiratori | 29 |
| 5 | Tecniche risolutive | 32 |
| 5.1 | Beat detection | 32 |
| 5.1.1 | Preprocessing | 33 |
| 5.1.2 | Reduction | 33 |
| 5.1.3 | Postprocessing | 34 |
| 5.1.4 | Thresholding | 34 |
| 5.1.5 | Peak picking | 35 |
| 5.2 | Clustering | 35 |
| 5.3 | Reti neurali | 37 |
| 5.4 | Dimensione frattale | 38 |
| 5.5 | Catene di triple Markoviane | 38 |
| 5.5.1 | Nozioni di base di probabilità | 38 |
| 5.5.2 | Processo di Markov | 39 |
| 5.5.3 | Catena di Markov | 40 |
| 5.5.4 | Modelli nascosti di Markov | 40 |
| 5.5.5 | Catena di coppie markoviane | 40 |
| 5.5.6 | Catena di triple markoviane | 41 |
| II | Stato dell'arte e analisi delle metodologie | 42 |
| 6 | Stato dell'arte | 43 |
| 6.1 | Algoritmi di riconoscimento delle fasi e dei cicli respiratori | 43 |

| | | |
|------------|--|-----------|
| 6.1.1 | Automated respiratory phase and onset detection using only chest sound signal | 44 |
| 6.1.2 | Breath Analysis of Respiratory Flow using Tracheal Sounds | 45 |
| 6.1.3 | Acoustical respiratory signal analysis and phase detection | 47 |
| 6.1.4 | Computerized acoustical respiratory phase detection without air-flow measurement | 47 |
| 6.1.5 | Respiratory onset detection using variance fractal dimension | 48 |
| 6.1.6 | Automated respiratory phase detection by acoustical means | 49 |
| 6.1.7 | A Software Toolkit for Acoustic Respiratory Analysis | 52 |
| 6.2 | Classificazione degli algoritmi | 54 |
| 6.3 | Conclusioni | 56 |
| 7 | Analisi delle Metodologie | 57 |
| 7.1 | Metodo elementare | 58 |
| 7.2 | Beat detection | 58 |
| 7.2.1 | Scelta dell'algoritmo | 58 |
| 7.3 | Pattern recognition e apprendimento automatico | 59 |
| 7.3.1 | Reti neurali | 60 |
| 7.4 | Clustering | 61 |
| III | Sistema software | 62 |
| 8 | Requisiti | 63 |
| 8.1 | Utenti del sistema software | 63 |
| 8.2 | Requisiti funzionali | 64 |
| 8.3 | Requisiti funzionali opzionali | 64 |
| 8.4 | Requisiti non funzionali | 64 |
| 8.5 | Requisiti non funzionali opzionali | 68 |
| 8.6 | Scelta della soglia di allarme | 69 |
| 9 | Architettura | 70 |
| 9.1 | Componenti | 70 |
| 9.1.1 | Meccanismi di persistenza | 70 |
| 9.1.2 | Interfaccia con lo stetoscopio | 71 |
| 9.1.3 | Interfaccia utente | 75 |
| 10 | Implementazione | 77 |
| 10.1 | Algoritmo di riconoscimento della respirazione | 77 |

| | | |
|-----------|--|-----------|
| 10.2 | Algoritmo di beat detection | 78 |
| 10.3 | Riconoscimento delle apnee a rischio | 79 |
| 10.4 | Complessità computazionale dell'algoritmo | 81 |
| 11 | Test | 82 |
| 11.1 | Scelta dei casi di test | 82 |
| 11.2 | Valutazione dell'output | 84 |
| 11.3 | Casi di test e risultati | 87 |
| 11.3.1 | Casi di test sui file delle repository | 87 |
| 11.3.2 | Caso di test di tolleranza al rumore bianco | 88 |
| 11.3.3 | Caso di test di localizzazione di una apnea troppo lunga | 89 |
| 11.4 | Descrizione dei file di respiro reperiti da [14] | 89 |
| IV | Conclusioni | 90 |
| 12 | Conclusioni | 91 |
| 12.1 | Commenti ai risultati ottenuti | 91 |
| 12.2 | Sviluppi futuri | 91 |
| 13 | Bibliografia | 93 |

Capitolo 1

Introduzione

1.1 Scopo

Lo scopo di questa tesi è di implementare un prototipo di un meccanismo di monitoraggio della respirazione usando uno stetoscopio elettronico. Il punto di arrivo è un software che prende in input il segnale di uno stetoscopio elettronico e capisce se il soggetto sta respirando o no. Il viaggio per arrivare a questa meta passa anche da una esplorazione dello stato dell'arte, una tappa di per se importante.

1.2 Motivazioni

La sindrome da apnea del sonno è una patologia molto diffusa. Il sonno di un soggetto affetto da tale patologia è disturbato da apnee e da episodi di respirazione insufficiente. Forme medie e gravi di sindrome da apnea del sonno sono un fattore di rischio per, e una concausa di: pressione alta, malattie cardiache, diabete, depressione. Inoltre tale patologia contribuisce a creare un senso perenne di sonnolenza e spossatezza. Si stima che più della metà dei soggetti affetti da tale patologia non ne siano al corrente[63]. Questo a causa dei meccanismi di diagnosi più diffusi che sono costosi o scomodi e richiedono al paziente di trascorrere la notte in un centro specializzato. Inoltre i centri specializzati sono pochi e possono essere molto costosi. Sono in fase di sviluppo ma non hanno per il momento una diffusione capillare altri strumenti di diagnosi più pratici. Alcuni studi hanno analizzato i dati di alcuni soggetti che sono morti a causa di eventi cardiovascolari acuti e che erano affetti da forme medie o gravi di sindrome da apnea del sonno. La conclusione è stata che la maggior parte di tali soggetti è morta durante il sonno. In questa tesi sviluppiamo un prototipo di uno strumento non invasivo di monitoraggio del respiro e di diagnosi della sindrome da apnea del sonno: attraverso uno stetoscopio elettronico,

il sistema deve vigilare su un soggetto e registrare la frequenza e la durata delle apnee e, cosa più importante, deve svegliare il soggetto nel caso in cui l'apnea duri troppo.

1.3 Contenuto della tesi

La tesi è divisa in tre parti:

1. La prima parte contiene:

- I requisiti necessari alla comprensione del problema.
- I requisiti necessari alla comprensione delle tecniche di soluzione del problema. Le tecniche di soluzione sono:
 - Quelle di cui si parla nello stato dell'arte.
 - Quelle usate nel sistema software implementato.
 - Quelle di cui è stata valutata la fattibilità con esito positivo.

Questa parte parla di tutti gli argomenti necessari, con un livello di sintesi proporzionale all'importanza dell'argomento.

2. La seconda parte contiene:

- Un riassunto dello stato dell'arte e di come si può adattare il materiale presente nello stato dell'arte per risolvere il problema.
- Una analisi e una discussione delle metodologie in atto e in potenza per la risoluzione del problema.

3. La terza parte contiene la descrizione di una implementazione di un prototipo di un sistema che risolve il problema.

Parte I

Prerequisiti

Capitolo 2

Respirazione e apparato respiratorio

2.1 Anatomia dell'apparato respiratorio

L'*apparato respiratorio* (o anche sistema respiratorio) è un sistema biologico atto alla respirazione. Negli esseri umani l'anatomia funzionale dell'apparato respiratorio include [23]:

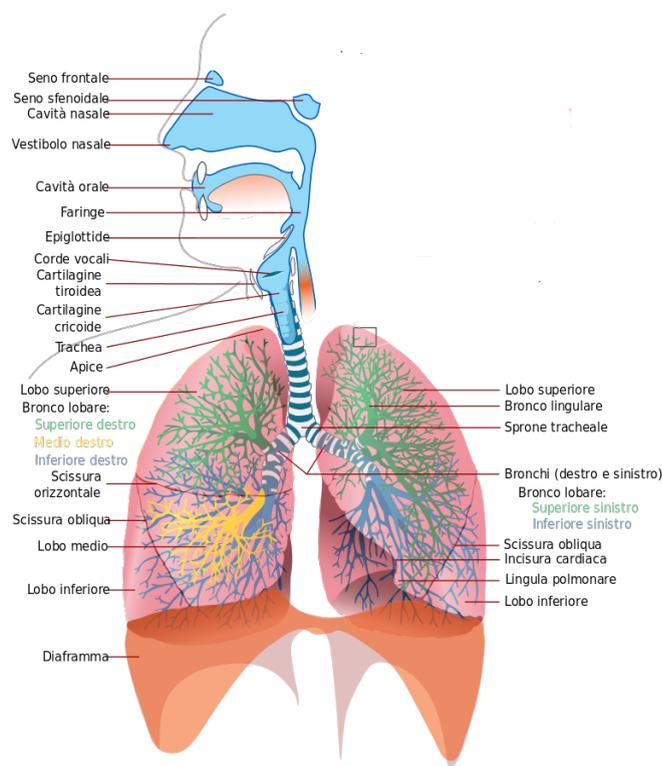
Vie aeree. Le vie aeree sono cavità in cui le sostanze gassose, vengono trasportate da o verso i polmoni. Le seguenti parti del corpo sono vie aeree: naso esterno, cavità orale, faringe, laringe, trachea. La figura 2.1 [50] è una rappresentazione più esaustiva dell'apparato respiratorio.

Polmoni. I polmoni sono l'organo essenziale per la respirazione. La loro principale funzione è quella di trasportare l'ossigeno dell'ambiente circostante al sangue e di espellere l'anidride carbonica. I polmoni contengono delle piccole sacche d'aria chiamate alveoli. Attraverso i capillari degli alveoli avviene lo scambio per diffusione di ossigeno ed anidride carbonica tra il sangue dell'organismo e l'aria contenuta negli alveoli. Il polmone destro è diviso in tre lobi: superiore, medio ed inferiore; mentre quello polmone sinistro è diviso in due lobi: uno superiore ed uno inferiore.

Muscoli respiratori. I muscoli respiratori sono il diaframma, i muscoli intercostali, i muscoli addominali, lo sternocleidomastoideo e i muscoli scaleni. Questi muscoli causano l'espansione polmonare.

2.2 Respirazione

La *respirazione* è il processo attraverso il quale l'organismo scambia aria tra i polmoni e l'ambiente circostante. La respirazione permette di acquisire ossigeno nel sangue che viene poi usato nel metabolismo e permette di eliminare i residui gassosi del metabolismo



come l'anidride carbonica. Durante la respirazione si scambia anche vapore acqueo. Il termine usato per indicare il respiro in condizioni normali è *eupnea*.

2.2.1 Meccanica della respirazione

La respirazione è ciclica e un ciclo di respirazione ha atto in tre fasi consecutive [52, 51]:

Inspirazione. Durante questa fase l'aria viene introdotta nei polmoni. L'inspirazione avviene grazie alla contrazione dei muscoli intercostali e del diaframma, tale contrazione provoca un aumento di volume polmonare e una diminuzione della pressione intrapleurica: ne consegue un'aspirazione dell'aria nei polmoni.

Espirazione. Durante questa fase l'aria viene espulsa dai polmoni. È determinata dal rilascio della forza elastica del parenchima polmonare. Il volume toracico diminuisce, i polmoni vengono compressi e l'aria espulsa.

Pausa. Tra una espirazione e l'inspirazione successiva ci può essere una pausa di durata variabile.

2.3 Diagnosi delle malattie dell'apparato respiratorio

2.3.1 Spirometro con pneumotacografo

Lo spirometro è uno strumento utilizzato per misurare i volumi d'aria polmonari. È composto da un sensore collegato a un boccaglio, attraverso il quale il paziente respira e da una parte che misura i movimenti di aria provocati dal soggetto. Lo spirometro con pneumotacografo è un tipo di spirometro dotato di una lamina con la funzione di restringimento e che causa una differenza di pressione. Questa differenza pressoria viene misurata da un manometro e poi riconvertita in un segnale proporzionale al flusso generato [30, 29].

2.3.2 Polissonografia

La *polissonografia* è una tecnica diagnostica che consiste nella registrazione simultanea di più parametri fisiologici durante la notte. Normalmente nel corso del test vengono registrati: il flusso d'aria della respirazione, il livello di ossigeno nel sangue, la posizione del corpo, l'attività cerebrale (attraverso un elettroencefalografo), il movimento degli occhi (attraverso un elettrooculografo) e l'attività cardiaca (attraverso un elettrocardiografo) [31].

2.3.3 Auscultazione

L'*auscultazione* è un sistema di diagnosi che consiste nell'ascoltare i suoni interni del corpo. Nella accezione che ci interessa, il termine auscultazione ha il significato di ascoltare attraverso uno stetoscopio i suoni prodotti dall'apparato respiratorio. Lo stetoscopio può essere posizionato in varie parti del torace e della schiena [38].

2.3.4 Stetoscopio

Lo stetoscopio è uno strumento medico utile all'auscultazione delle parti interne del corpo e soprattutto del torace. Ci sono due tipi principali di stetoscopio:

Stetoscopio acustico. Lo stetoscopio acustico funziona tramite la trasmissione di suoni provenienti dal corpo attraverso dei canali contenenti aria fino alle orecchie dell'ascoltatore. Uno dei problemi degli stetoscopi acustici è il basso volume del suono trasmesso. Ne consegue una difficoltà nell'eseguire una diagnosi precisa.

Stetoscopio elettronico. L'innovazione tecnologica nel campo degli stetoscopi consente oggi la rilevazione di una più ampia gamma di suoni ed una maggior qualità d'ascolto,

con la possibilità di ottenere registrazioni e riproduzioni di grande fedeltà dei suoni. A differenza degli stetoscopi acustici, che sono tutti basati sullo stesso metodo di funzionamento, gli stetoscopi elettronici variano molto tra un modello e l'altro. I più semplici funzionano grazie ad un microfono applicato al petto del paziente, ma di contro, risentono del suono dell'ambiente esterno che spesso causa interferenze. Alcuni stetoscopi elettronici forniscono direttamente un audio in output e si possono usare insieme ad un dispositivo esterno di registrazione. Altri possono trasmettere il segnale via wireless o bluetooth. Un dispositivo (ad esempio un computer) riceve questo segnale e può fare vari tipi di analisi. Altri modelli, più complessi, trasformano le onde sonore in impulsi elettrici, così da poter essere amplificate per un migliore ascolto. Recentemente, alcuni stetoscopi elettronici sono stati dotati di filtri, allo scopo di eliminare le interferenze sonore esterne e in più sono in grado di poter selezionare il range di frequenza in modo da poter ascoltare separatamente o meno i suoni cardiaci da quelli respiratori. [77]

2.3.5 Invasività clinica

Un parametro molto importante per valutare un sistema diagnostico è l'invasività clinica o semplicemente invasività. Questa si riferisce alla possibilità che l'esame finisca per compromettere ulteriormente lo stato di salute del soggetto. Siamo in presenza di un esame invasivo ad esempio, nel caso in cui l'esame possa portare agenti contaminanti (virus, batteri, tossine, sporcizia) all'interno del diretto interessato e quindi causare una infezione che aggravi le condizioni del paziente. L'invasività di un meccanismo diagnostico esce allo scoperto anche nei casi in cui un piccolo errore nella procedura procura danni al paziente [25]. Tra i sistemi diagnostici descritti in questa sezione il metodo meno invasivo è l'auscultazione.

2.4 Patologia dell'apparato respiratorio

2.4.1 Apnee del sonno

La *sindrome da apnea del sonno* è un disordine del sonno caratterizzato da ripetute apnee o ipopnee durante il sonno [26]. Una *apnea* è una pausa di durata anormale nella respirazione che supera i dieci secondi [17] e può durare anche alcuni minuti [16]. Una *ipopnea* è un evento caratterizzato da respirazione insufficiente, più precisamente una ipopnea si verifica quando il flusso d'aria si riduce di almeno il 30% per almeno dieci secondi e la desaturazione di ossigeno nel sangue è di almeno il 4% [17]. Ci sono tre forme di apnea del sonno:

Apnea centrale. La respirazione è interrotta per via di un mancato movimento dei muscoli respiratori, quindi il volume dei polmoni rimane invariato.

Apnea ostruttiva. La respirazione è interrotta a causa di un blocco fisico nelle vie aeree nonostante persistano movimenti respiratori [15].

Apnea mista. La respirazione è soggetta a entrambi i tipi di apnea appena descritti.

Le apnee del sonno si verificano sia nei bambini che negli adulti. I soggetti affetti da apnea del sonno possono manifestare i seguenti sintomi: eccessiva sonnolenza durante il giorno, tempi di reazione lenti, problemi alla vista, indebolimento delle funzioni del fegato e altro. Inoltre gravi forme di apnee del sonno ostruttive aumentano in modo significativo il rischio di eventi cardiovascolari fatali [17, 45, 46].

L'*indice di apnea-ipopnea (apnea-hypopnea index AHI)* è definito come il numero di eventi di apnea e di ipopnea in rapporto alla durata del sonno. L'AHI è un indicatore della gravità della sindrome di apnea del sonno e i suoi valori sono categorizzati tipicamente in: leggera da 5 a 15 episodi all'ora, moderata da 15 a 30 episodi all'ora o severa oltre i 30 episodi all'ora.

Lo studio [26] conclude che c'è una forte associazione tra la sindrome da apnea del sonno e l'ictus. In particolare dimostra che un indice da moderato a severo di AHI è associato ad un alto rischio di ictus e ipotizza che la sindrome da apnea del sonno contribuisca allo sviluppo di un ictus. Purtroppo resta ancora da scoprire se esiste un pattern respiratorio specifico che precede immediatamente un ictus.

Lo studio [18] prende in esame i polisonnogrammi e i certificati di morte di alcune persone che sono decedute a causa di una malattia cardiaca improvvisa. Le persone sono state divise in due gruppi: le persone del primo gruppo soffrivano di sindrome da apnea notturna mentre le persone del secondo no. Si è riscontrato che la maggior parte delle persone appartenenti al primo gruppo sono morte durante il sonno al contrario di quelle del secondo. Lo studio conclude che la gravità della sindrome da apnea del sonno è direttamente proporzionale al rischio di morte improvvisa per malattie cardiache durante il sonno.

Episodi acuti di apnea o ipopnea possono indurre: ipossiemia, aumento degli impulsi nel sistema nervoso simpatico, aumento brusco nella pressione sanguigna, aumento dello stress delle pareti cardiache, aritmie cardiache, ipercoagulabilità, stress ossidativo vascolare, infiammazioni sistemiche e altro. Questo potrebbe spiegare i dati osservati dallo studio. Resta un problema aperto quello di stabilire se nel primo gruppo di persone, la morte è immediatamente preceduta da un evento di apnea o ipopnea grave.

2.4.2 Arresto respiratorio

Un arresto respiratorio è l'interruzione della respirazione normale. È molto probabile che si verifichino traumi cerebrali se l'arresto respiratorio dura più di tre minuti e la morte è quasi certa se questo dura più di cinque minuti.

2.5 Base funzionale dei suoni respiratori

Secondo [13] i suoni normali che si possono sentire sul petto di un soggetto nella fase di inspirazione, vengono generati soprattutto nella parte lobare delle vie respiratorie. I suoni respiratori sono generati da turbolenze dell'aria nelle vie respiratorie. Le caratteristiche dei suoni sono molto variabili, si possono notare differenze da persona a persona che dipendono dal peso, dall'età, dallo stato di salute e altri fattori. I suoni respiratori variano anche rispetto alla densità del gas respirato la quale diminuisce con l'aumentare dell'altitudine rispetto al livello del mare. In generale però l'intensità dei suoni respiratori è proporzionale al quadrato del flusso d'aria. Lo studio [8] nota che i suoni inspiratori vengono prodotti in modo predominante nella zone periferiche dei polmoni, mentre i suoni espiratori vengono prodotti nelle zone più centrali.

2.6 Analisi acustica dei suoni respiratori

L'articolo [2] analizza l'energia spettrale dei suoni respiratori normali ascoltati sul petto di un soggetto. Conclude che questi seguono uno schema caratteristico, in particolare che la potenza decresce in modo esponenziale con la frequenza nell'intervallo dai 75 ai 2000Hz, dopo quest'ultimo valore la potenza è trascurabile. Secondo [3] il suono registrato sul petto di soggetti sani ha una intensità massima attorno ai 250Hz e decresce rapidamente fino ad arrivare ad un livello di intensità trascurabile attorno a una frequenza di circa 1000Hz. Nel caso dei soggetti malati invece i picchi di frequenze sono più alti.

Gli articoli [5, 6, 7, 1] classificano i suoni respiratori nel modo seguente (riassunto nella figura 2.1):

Suoni respiratori normali I suoni respiratori normali o fisiologici sono i suoni prodotti dal sistema respiratorio di un soggetto sano. Questi suoni sono più intensi in fase inspiratoria e decrescono in fase espiratoria (nonostante quest'ultima sia più lunga di quella inspiratoria) e ciò è dovuto al fatto che l'inspirazione avviene mediante contrazione muscolare, generando un flusso d'aria più veloce, rispetto all'espirazione, che è un fenomeno passivo. I suoni normali hanno una frequenza nell'intervallo dai 100 ai 1000Hz. A causa

dei suoni prodotti dal movimento dei muscoli del torace e dal diaframma e a causa di quelli prodotti dall'apparato cardiocircolatorio, i suoni dall'apparato respiratorio di solito non vengono studiati a frequenze minori di $60Hz$. I suoni respiratori normali si dividono in:

Murmori vescicolari Si generano per l'azione di filtro degli alveoli sull'aria in arrivo.

Suoni bronchiali Si generano nelle zone di passaggio dall'albero bronchiale agli alveoli, quindi per mescolanza di rumore bronchiale ed alveolare.

Soffio bronchiale Si ascolta in corrispondenza dei grossi bronchi in presenza di addensamento polmonare.

Suoni respiratori anormali I suoni respiratori anormali o patologici, sono suoni accidentali che non fanno parte del normale ciclo di respirazione. Ci sono due tipi di suoni anormali:

Continui Tra i suoni anormali continui si distinguono:

Ronchi o gemiti Questi suoni sono spesso gravi e sembrano dei rantoli, la loro frequenza dominante è $200Hz$, hanno quindi una bassa tonalità. Sono segno di broncocostrizione e si generano per il passaggio dell'aria in vie aeree ristrette per la presenza di muco o broncospasmo.

Rantolo secco (wheezes) Questi suoni hanno una frequenza dominante attorno ai $100Hz$ e hanno una durata maggiore di $100ms$. Sono un segnale caratteristico di una malattia ostruttiva polmonare.

Sibili Questi suoni sono dei *wheezes* molto più forti e sono la conseguenza di una ostruzione dinamica nella laringe o nella trachea. L'energia di questi suoni è concentrata in massima parte su una frequenza che si aggira attorno ai $1000Hz$.

Discontinui Tra i suoni anormali discontinui si notano:

Rantolo (Crackles) Questi suoni hanno una durata minore di $10ms$ e cadono in un vasto spettro di frequenze, tra 200 e $2000Hz$. Di solito sono indice di malattie cardiorespiratorie.

Fine crackles Sono i rantoli fini, anche detti rantoli crepitanti e hanno una tonalità alta. Sono dovuti alla ritardata apertura degli alveoli. Talvolta ad essi si sovrappongono gli sfregamenti pleurici, il cui reperto è molto simile.

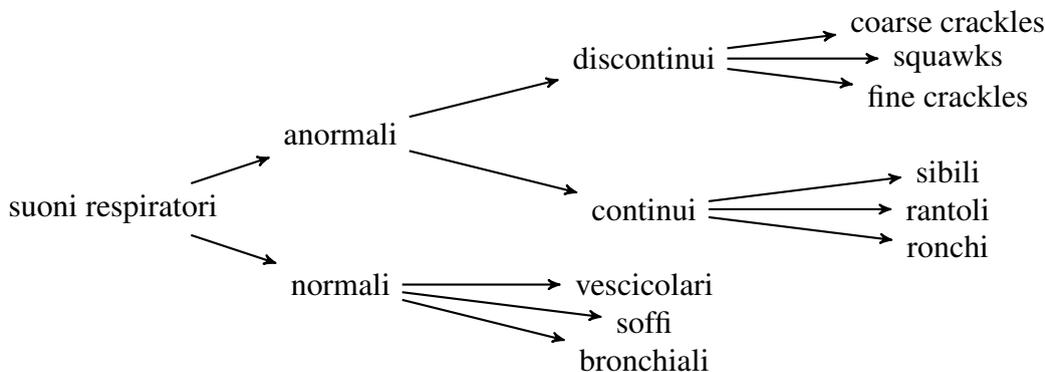


Figura 2.1: Classificazione dei suoni respiratori

Coarse crackles Sono i rantoli subcrepitanti, di una tonalità bassa, il cui reperto si modifica con la tosse. Sono segno di bronchite e si generano per il passaggio dell'aria attraverso il muco.

Squawks Questi suoni sono dei sibili di breve durata.

I risultati ottenuti da [8] ci dicono che in media, i suoni inspiratori hanno una intensità di circa $10db$ maggiore rispetto ai suoni espiratori a parità di flusso. Inoltre i suoni registrati da microfoni in prossimità della trachea sono più intensi di quelli registrati da microfoni sul petto.

I rantoli di solito vengono registrati in prossimità della trachea, dove gli effetti del filtraggio del corpo sono minimi. D'altro canto, i suoni discontinui accidentali generati a causa di patologie polmonari vengono riconosciuti meglio sulla parte basale bassa dei polmoni [9].

Rumore Ci sono altri suoni fisiologici che complicano l'auscultazione dei suoni respiratori. Se l'auscultazione avviene sul torace, questi sono i suoni cardiovascolari e i suoni gastrointestinali. Se si auscultano i suoni tracheali allora la fonte di disturbo sono i suoni dovuti alla deglutizione della saliva.

2.7 Schemi di respirazione

Lo studio [43] analizza gli schemi o pattern di respirazione normali. I soggetti coinvolti nello studio sono 65 e hanno età dai 20 agli 81 anni. Vari parametri sono stati presi in considerazione tra i quali: la frequenza di respirazione e il tempo di inspirazione. Lo studio conclude che la media delle frequenze di respirazione è 16.6 ± 2.8 respiri al minuto mentre

la durata media dei tempi di inspirazione è di 1.62 ± 0.31 secondi. Una frequenza di respirazione media di 16.6 respiri al minuto implica una durata media di 3.6s a respiro. Esistono vari schemi di respirazione anormali alcuni dei quali sono elencati nella figura 2.2. Descriviamo più in dettaglio gli schemi che compaiono nella figura dall'alto verso il basso:

Apneosi. L'apneosi è uno schema di respirazione caratterizzato da profondi annaspamenti durante l'inspirazione, dopo una inspirazione avviene una lunga pausa seguita da un rilascio insufficiente di aria. Notiamo che durante l'apneosi la pausa avviene dopo l'inspirazione e non dopo l'espirazione come nella sindrome da apnea ostruttiva.

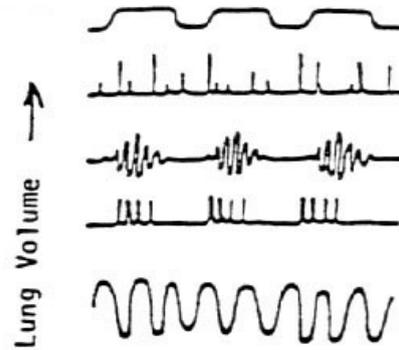


Figura 2.2: Elenco di pattern respiratori [44]

Respiro agonico. Il respiro agonico è uno schema anormale di respirazione caratterizzato da boccheggiamento e da una riduzione estrema della frequenza degli atti respiratori fino al loro totale arresto. L'uso corretto del termine si deve restringere all'ultimo respiro prima della morte.

Respiro di Cheyne Stokes. Il respiro di Cheyne Stokes è uno schema di respirazione anormale nel quale il respiro si fa da prima progressivamente più frequente e profondo, poi la frequenza respiratoria diminuisce gradualmente fino a portare ad una apnea. Lo schema si ripete e ogni ciclo di solito dura dai 30 secondi ai 2 minuti.

Respiro di Kussmaul. Il respiro di Kussmaul è una forma di iperventilazione compensatoria. È caratterizzato da atti respiratosi molto lenti ed in particolare da una inspirazione profonda e rumorosa a cui segue una breve apnea, poi una espirazione breve e gemente e infine una pausa post-espiratoria decisamente prolungata.

Capitolo 3

Acustica e audio digitale

3.1 L'acustica e il suono

L'acustica è quella branca della fisica che studia il suono, le sue cause, la sua propagazione e la sua ricezione. La percezione sonora è normalmente legata alle vibrazioni del timpano nell'orecchio. Queste vibrazioni sono provocate da piccole variazioni di pressione nell'aria. La variazione di pressione dell'aria è quindi l'equivalente fisico del suono [72]. La vibrazione provoca una successione di compressioni e rarefazioni nel mezzo dell'ambiente circostante e tale disturbo comincia a propagarsi lontano dalla sorgente in tutte le direzioni (un caso visibile è quello delle onde sull'acqua). L'effetto uditivo, consiste nella percezione da parte di un apposito dispositivo (orecchio di esseri viventi o microfoni artificiali) delle piccole e rapidissime vibrazioni emesse appunto da una sorgente sonora.

3.2 Onde sonore

La natura fisica del suono è di tipo ondulatorio, ovvero descrive un movimento che può essere rappresentato tramite un'onda. Si tratta di onde meccaniche che trasportano energia lontano dalla sorgente sonora. Alcune principali grandezze delle onde sono [76]:

Periodo. Il periodo, o durata dell'oscillazione rappresenta il tempo in cui l'onda compie un'oscillazione e torna alla condizione iniziale.

Frequenza. La frequenza di un'onda indica il numero di cicli completi che compie in un secondo:

$$f = \frac{1}{t}$$

dove f è la frequenza e t è il periodo. In generale la frequenza di un evento si misura in Hz . Ricordiamo che un Hz è il numero di eventi che accadono in un secondo.

Ampiezza. L'ampiezza o intensità invece è il valore massimo raggiunto dall'oscillazione stessa durante un periodo ed è determinata dalla quantità di energia impiegata. Viene espressa tramite i decibel.

Forma. La forma dell'oscillazione è determinata dal numero delle componenti parziali e dal loro rapporto di frequenza, ampiezza e fase. Essa rappresenta l'aspetto dell'onda in base all'ampiezza e al tempo servendosi di coordinate cartesiane.

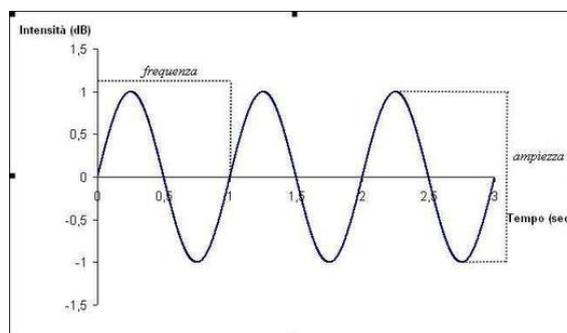


Figura 3.1: Rappresentazione delle principali grandezze dell'onda sonora [74]

3.3 Suono analogico e digitale

Il suono è un flusso informativo di natura temporale che scorre all'interno di apparecchiature elettroniche. Questa informazione può essere rappresentata in due forme: analogico e digitale. Una rappresentazione analogica è una rappresentazione o trasformazione di una grandezza fisica tramite una sua analogia: in altre parole, si intende un sistema in cui una quantità fisica continuamente variabile viene rappresentata da un'altra (ad esempio, la tensione di un segnale elettrico) nel modo più fedele possibile. La curva continua nel tempo delle variazioni di ampiezza viene rappresentata da una curva continua nel tempo delle variazioni di tensione elettrica.

Una rappresentazione digitale non cerca di imitare la curva continua di ampiezza con una curva analoga ad essa, ma piuttosto assegna dei numeri che rappresentano di volta in volta il valore dell'ampiezza in istanti successivi di tempo. Sarà la successione di numeri a rappresentare l'andamento della curva. La rappresentazione digitale non è continua, ma discreta; cioè esistono degli eventi ben definiti che sono i valori dell'ampiezza in precisi istanti di tempo. I vantaggi della rappresentazione digitale e cioè di un codice simbolico sono molti: dalle operazioni di copia del segnale, a quelle di manipolazione. Ma un aspetto totalmente nuovo è la possibilità di correzione degli errori introdotti dai supporti per la memorizzazione. Per errore si intende che alcuni numeri che rappresentano il segnale vengono letti in maniera differente da come era stati memorizzati o trasmessi.

3.3.1 Discretizzazione

Per sottoporre un segnale analogico ad elaborazioni digitali è necessario prima convertirlo in una sequenza di dati binari, ottenuti tramite due operazioni di discretizzazione: una discretizzazione nel dominio del tempo o campionamento che riduce gli infiniti valori di un segnale analogico in sequenze di campioni discreti e una discretizzazione nel dominio delle ampiezze o quantizzazione (del livello) che permette di rappresentare ciascun campione con un numero finito di bit. La necessità di elaborare il segnale in forma digitale porta a limitare il numero delle informazioni e quindi a misurare le grandezze solo a valori discreti del tempo, attribuendo loro un certo numero discreto di valori, variabili fra un massimo e un minimo. I segnali acquisiti diventano quindi serie di valori corrispondenti agli istanti per cui si è effettuato il campionamento.

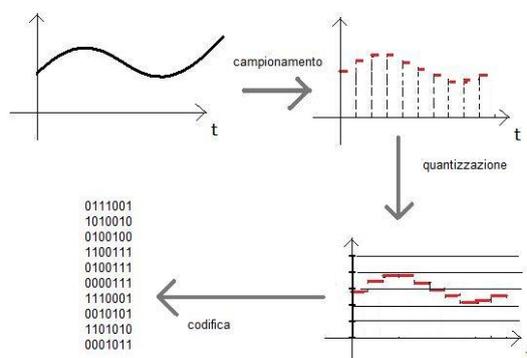


Figura 3.2: Discretizzazione di un suono da analogico a digitale

Campionamento

Campionare l'audio significa creare una sequenza di campioni, ovvero i valori di un segnale audio in diversi momenti temporali quindi ogni campione memorizzato rappresenta un'ampiezza ad un dato istante. Viene chiamata frequenza di campionamento il numero di volte che viene campionato il segnale in un secondo. [73].

Secondo il teorema di campionamento di Nyquist e Shannon in una conversione da analogico a digitale la minima frequenza di campionamento necessaria per evitare ambiguità e perdita di informazione nella ricostruzione del segnale analogico originario, con larghezza di banda finita e nota, è pari al doppio della frequenza massima del suono che stiamo convertendo. Se non viene rispettato questo teorema, cioè si ha un sottocampionamento del segnale analogico nel dominio del tempo allora nel dominio delle frequenze si ha la produzione di frequenze non proprie del segnale originario (*alias*) producendo cioè una distorsione del segnale originario divenuto ora non più fedele.

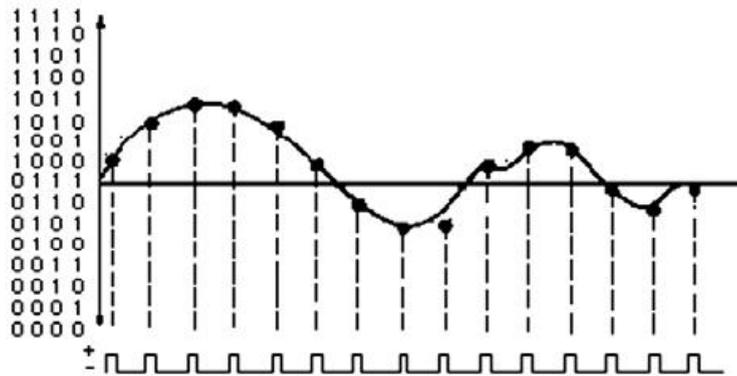


Figura 3.3: Campionamento e quantizzazione di un segnale continuo.

Quantizzazione

Dopo il campionamento, la conversione analogico-digitale viene completata con la quantizzazione, che consiste nell'associare ad ogni campione un valore discreto.

Per ottenere ciò i valori possibili della grandezza in questione vengono innanzitutto limitati tra un massimo ed un minimo intorno a dei valori discreti preventivamente definiti, definendo così le relative regioni di decisione e la dinamica del quantizzatore stesso: in tal modo il valore analogico della grandezza originaria, in corrispondenza del valore campionato in ascissa, verrà ricondotto al più prossimo dei valori discreti preventivamente definiti tramite il processo di decisione.

Con la quantizzazione vengono però introdotti degli errori detti *errori di quantizzazione* pari alla differenza tra il valore quantizzato e il suo valore reale nel campo continuo. L'errore massimo possibile che potrà essere introdotto volta per volta sarà quindi pari alla metà dell'intervallo discreto discriminabile o regione di decisione. L'insieme di questi errori conduce al rumore di quantizzazione.

3.4 Segnale

In generale un segnale è una funzione di una o più variabili che contiene informazioni relative ad un fenomeno fisico. In questo caso ci interessano i segnali sonori che sono delle funzioni dell'ampiezza rispetto al tempo. I segnali possono essere classificati secondo le seguenti proprietà:

Continuità nel tempo Un segnale può essere a tempo continuo o a tempo discreto a seconda che il dominio della funzione sia non numerabile o numerabile.

Continuità nell'ampiezza Un segnale può essere ad ampiezza continua oppure ad ampiezza discreta (o quantizzato) a seconda che l'immagine della funzione sia non numerabile o numerabile.

Periodicità Un segnale può essere periodico se esiste una quantità T nel dominio del tempo tale che per ogni tempo t vale $s(T + t) = s(t)$. Se non vale questa proprietà allora il segnale è aperiodico. Un segnale si dice quasi periodico se è composto dalla somma di segnali periodici con diverse frequenze che tra di loro stanno in rapporti non razionali.

Determinatezza Un segnale si dice determinato se è perfettamente noto e rappresentabile con una funzione che ne specifica l'andamento in ogni istante invece viene chiamato aleatorio se non è completamente noto a priori, ma può assumere un qualunque andamento entro una classe di funzioni specificata da alcune proprietà statistiche.

Stazionarietà Un segnale stocastico, si dice stazionario se le sue proprietà statistiche non cambiano nel tempo, altrimenti si dice non stazionario.

Energia Dato un segnale $s(t)$ definiamo l'energia del segnale come:

$$E = \int_{-\infty}^{\infty} |s(t)|^2 dt$$

La definizione acquista significato fisico quando il segnale è reale, in tal caso, infatti supponendo che $s(t)$ rappresenti la tensione applicata o la corrente immessa ad una resistenza di 1 Ohm , questa è l'energia da essa dissipata. Definiamo la potenza media di un segnale continuo $s(t)$ il limite:

$$P = \frac{1}{2T} \lim_{T \rightarrow \infty} \int_{-T}^T |s(t)|^2 dt$$

Il segnale $s(t)$ è detto ad energia finita se E è finito e diverso da zero. Il segnale è detto a potenza media finita se P è finito e diverso da zero. Si noti che se un segnale ha energia finita la sua potenza media è nulla e se un segnale ha potenza media finita la sua energia è infinita, pertanto le due classi sono disgiunte. Una classe importante di segnali a potenza media finita è costituita dai segnali periodici; in tal caso l'energia è finita e la potenza media coincide con quella calcolata in un periodo.

3.5 Analisi nel dominio del tempo e nel dominio della frequenza

I principali metodi di analisi del segnale possono essere riassunti nei concetti di analisi nel dominio del tempo e analisi nel dominio della frequenza. È importante osservare che questi due modi di affrontare un problema sono tra loro intercambiabili, nel senso che, sotto opportune condizioni, nessuna informazione viene persa nel passare da un dominio all'altro. Il vantaggio che deriva dall'introduzione dei due domini è la possibilità di cambiare la prospettiva con la quale si osserva un dato fenomeno. In questo modo un problema che appare di difficile soluzione in un dominio può risultare molto più semplice nell'altro [71].

Analisi nel dominio del tempo Questa forma di rappresentazione è quella che ci è maggiormente familiare, in essa appaiono le variazioni subite dal segnale al trascorrere del tempo. Lo strumento più frequentemente usato e che opera notoriamente nel dominio del tempo è l'oscilloscopio.

Analisi nel dominio della frequenza Invece di analizzare le variazioni del segnale al passare del tempo, mostra come e quanto un segnale si suddivide o è distribuito nelle varie bande di frequenza, definite all'interno di un dato range. Viene chiamata analisi spettrale e lo strumento matematico che consente di trasferire lo studio dei segnali e dei sistemi dal dominio del tempo al dominio della frequenza è la trasformata di Fourier.

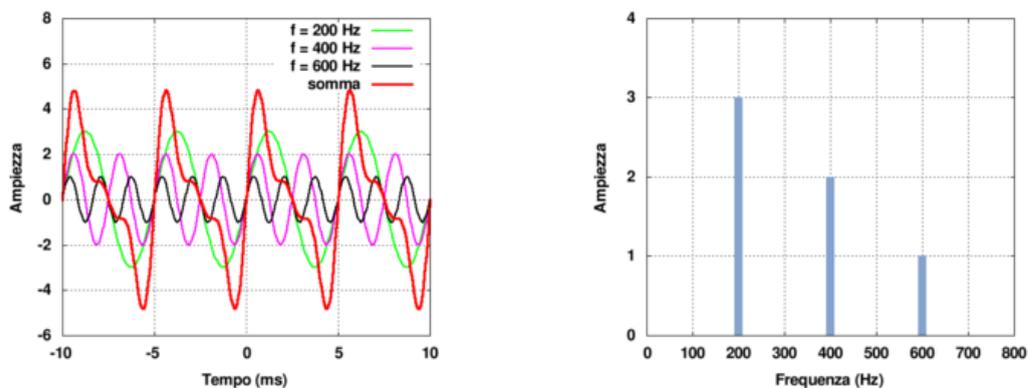


Figura 3.4: Esempi di grafici nel dominio del tempo e nel dominio delle frequenze [75]

3.6 Analisi armonica e trasformata di Fourier

L'analisi armonica è la branca della matematica che studia la rappresentazione delle funzioni o dei segnali come sovrapposizione di onde fondamentali. Indaga e generalizza la nozione di serie di Fourier e trasformata di Fourier.

L'interpretazione dello sviluppo in serie di Fourier è che un segnale periodico di potenza finita si può sviluppare come combinazione lineare di funzioni periodiche semplici la cui frequenza è un numero intero. Il tutto si generalizza facilmente al caso in cui il periodo sia T e quindi la frequenza fondamentale abbia il valore $1/T$.

L'importanza della formula è che tutta l'informazione di una funzione periodica continua può essere espressa con un'infinità numerabile (quindi discreta) di valori complessi.

3.6.1 Serie di Fourier

La serie di Fourier è una rappresentazione di una funzione mediante una combinazione lineare di funzioni sinusoidali fondamentali. Un segnale $f(t)$ è una funzione e quindi si può decomporre usando una serie di Fourier. La decomposizione di un segnale attraverso una serie di Fourier ci dà una descrizione delle frequenze che lo compongono [66, 68].

Sia $f(x)$ una funzione definita sui numeri reali e periodica di periodo 2π . Diremo che $f(x)$ è sviluppabile in serie di Fourier se esistono dei coefficienti a_0, a_k, b_k con k numero naturale tali che

$$f(x) = a_0 + \sum_{k \in \mathbb{N}} (a_k \cos(kx) + b_k \sin(kx))$$

La formula precedente è lo sviluppo in serie di Fourier della funzione f e i coefficienti di Fourier si possono calcolare nel seguente modo:

$$a_0 = \frac{1}{2\pi} \int_{-\pi}^{\pi} f(x) dx \quad a_k = \frac{1}{\pi} \int_{-\pi}^{\pi} f(x) \cos(kx) dx \quad b_k = \frac{1}{\pi} \int_{-\pi}^{\pi} f(x) \sin(kx) dx$$

[67]

3.6.2 Trasformata di Fourier

La trasformata di Fourier permette di scomporre e successivamente ricombinare, un segnale generico in una somma infinita di sinusoidi con frequenze, ampiezze e fasi diverse. L'insieme di valori in funzione della frequenza è detto spettro di ampiezza e spettro di fase.

Se il segnale in oggetto è un segnale periodico, la sua trasformata di Fourier è un insieme di valori discreti, che in tal caso prende il nome di spettro discreto o spettro a pettine. Mentre nel caso in cui il segnale sia non periodico lo spettro è continuo, e tanto

più è esteso lungo l'asse delle frequenze quanto più è limitato nel dominio originario della variabile indipendente, e viceversa. Se il segnale ha un valore medio diverso da zero la serie restituisce anche una componente costante che lo rappresenta.

Sia $f \in L^1(\mathbb{R}^n)$ una funzione integrabile, la trasformata continua di Fourier, detta anche semplicemente trasformata di Fourier, è definita nel modo seguente:

$$\mathbb{F}(f) = t \mapsto \frac{1}{(2\pi)^{\frac{n}{2}}} \int_{\mathbb{R}^n} e^{-i \cdot t \cdot x} dx$$

[66][68]

3.6.3 Trasformata discreta di Fourier

La trasformata di Fourier opera su funzioni continue, sia nel dominio dei tempi che delle frequenze. Al contrario, la trasformata discreta di Fourier opera su funzioni a dominio discreto. Sia S_n un insieme di sequenze periodiche di periodo n . Un generico elemento di S_n si può scrivere come $y = \{y_j\}_{j \in \mathbb{N}}$ e si può pensare come un segnale periodico discreto nel quale y_j è il valore del segnale al tempo j . Definiamo la trasformata discreta di Fourier di y come la sequenza

$$(\mathbb{F}_n(y))_k = z_k$$

dove

$$z_k = \sum_{j=0}^{n-1} y_j \bar{w}^{j \cdot k} \quad w = \exp\left(\frac{s\pi i}{n}\right)$$

[66]

3.7 Wavelet

Il processo di convertire un segnale dal dominio del tempo a dominio delle frequenze viene fatto tramite la trasformata di Fourier, però Fourier non fornisce informazioni sufficienti su segnali non stazionari in quanto determina soltanto le componenti di frequenza di un segnale, ma non la loro posizione nel tempo. La trasformata Wavelet consente un'analisi migliore per questo tipo di segnali.

Una wavelet è una forma d'onda oscillante di lunghezza finita, con un'ampiezza che parte da zero e aumenta e decrementa tornando a zero. Al contrario delle sinusoidi usate nella trasformata di Fourier, le wavelet sono più concentrate nel tempo. Di solito forniscono un'analisi del segnale che è localizzato sia nel tempo che nella frequenza mentre la trasformata di Fourier è localizzata solo nella frequenza. Un esempio di wavelet è rappresentato dalla figura seguente:

Una funzione $\psi(t)$ è una wavelet se soddisfa i seguenti criteri:

- Una wavelet deve avere energia finita.

$$E = \int_{-\infty}^{\infty} |\psi(t)|^2 dt < \infty$$

- Se Ψ è la trasformata di Fourier della wavelet $\psi(t)$ allora $\Psi(0) = 0$. Cioè la wavelet non ha componenti nella frequenza zero.
- La trasformata di Fourier di una wavelet complessa deve essere reale e deve essere nulla nelle frequenze negative.

3.8 Trasformata wavelet continua

Sia $\psi(t)$ una wavelet, definiamo la trasformata wavelet continua di una funzione $x(t)$ come

$$X(a, b) = \frac{1}{\sqrt{a}} \int_{-\infty}^{\infty} \psi\left(\frac{t-b}{a}\right) x(t) dt$$

La scala o parametro di dilatazione a corrisponde all'informazione di frequenza e il parametro di traslazione b si riferisce alla locazione della funzione wavelet mentre viene fatta spostare lungo il segnale quindi corrisponde all'informazione temporale nella trasformata. Esiste anche la trasformata wavelet discreta che opera su segnali a tempo discreto e che restituisce coefficienti discreti nel dominio delle wavelet. I parametri a e b assumono valori in una griglia discreta

$$a = 2^{-j} \quad b = k \cdot 2^{-j} \quad j, k \in \mathbb{Z}$$

[69][70]

Capitolo 4

Filtri digitali

4.1 Denoising

Uno degli ostacoli principali dell'analisi computerizzata dei suoni polmonari è la presenza di rumore nei segnali. In questo caso per rumore si intendono quei suoni provenienti da strumenti come ventilatore, aria condizionata, e altri rumori ambientali che possono contaminare i segnali sonori del polmone. La natura rumorosa dei suoni polmonari è un fattore di impedimento che vieta l'identificazione di funzioni utili per la diagnostica. Quindi il denoising di segnali sonori polmonari è d'obbligo per l'utilizzo efficace della diagnosi e in questo capitolo approfondiremo le varie tecniche per l'eliminazione dei rumori.

4.2 Classificazione dei filtri digitali

Un filtro digitale è una funzione in grado di modificare il contenuto armonico di un segnale elettrico complesso. Queste modifiche si esprimono in termini di fasci di frequenze che vengono evidenziate o soppresse a seconda delle caratteristiche del tipo di filtro usato. Una importante penalizzazione per i filtri digitali è che sono limitati nelle frequenze: un filtro digitale deve sempre e comunque rispettare il teorema di Nyquist, ovvero il teorema del campionamento, che impone al filtro un limite massimo sulle frequenze che può elaborare, altrimenti si ottengono segnali disturbati da aliasing.

Si distinguono due classi di filtri: lineari e non lineari.

Filtri lineari La funzione dà in uscita un valore che è una combinazione lineare dei valori compresi nella finestra del segnale appena analizzato, cioè l'output del filtro può mettersi in relazione con i valori presi in ingresso. Si possono definire filtri lineari che puliscono il segnale dal rumore oppure che esaltano le discontinuità.

Filtri non lineari Per questo tipo di filtro non è possibile definire un operatore lineare; solitamente sono operatori di rango, cioè operatori che agiscono sui valori presi in ingresso dopo averli ordinati quindi l'output dipende contemporaneamente da più ingressi.

La differenza sostanziale tra i due tipi di filtri è che, mentre per i primi si può applicare la trasformata di Fourier con tutte le sue proprietà, nei secondi questa operazione non è possibile.

4.2.1 Categorie di filtri

I filtri si dividono in quattro principali categorie:

Filtro passa alto

Il filtro passa alto permette il passaggio di una banda di frequenze al di sopra di una determinata frequenza limite (*frequenza di taglio*). Il taglio del filtro indica la zona della banda di frequenze che divide le due sezioni: frequenze passanti (alte), frequenze tagliate (basse).

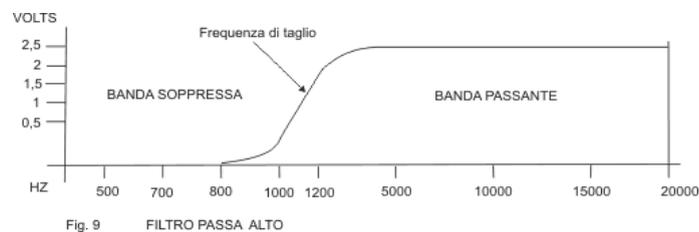


Figura 4.1: Esempio di un filtro passa alto [76]

Filtro passa basso (low pass).

Il filtro passa basso è esattamente l'opposto del filtro passa alto cioè permette il passaggio di una banda di frequenze al di sotto di una determinata frequenza limite.

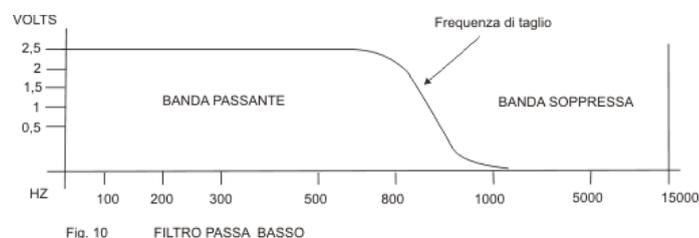


Figura 4.2: Esempio di un filtro passa basso [76]

Filtro passa banda (band pass).

Il passa banda deriva dall'accoppiamento di un passa alto con un passa basso, quindi lascia passare solo una banda ristretta di frequenze che rientrano tra le frequenze di taglio dei due filtri accoppiati. Comunque bisogna tener conto che al di sopra e al di sotto di tali frequenze le altre componenti del segnale vengono eliminate, non in maniera netta, ma in modo graduale dipendente dalla pendenza della curva di selettività del filtro che varia a seconda del livello qualitativo del filtro stesso.

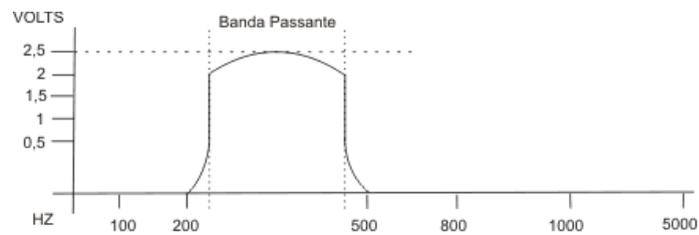


Figura 4.3: Esempio di un filtro passa banda [76]

Filtro soppressore di banda (band reject).

Il filtro a soppressione di banda è esattamente il contrario del filtro passa banda, in quanto pur essendo costituito dall'unione di un passa alto con un passa basso, respinge solo una ristretta banda di frequenze lasciando passare tutte le altre.

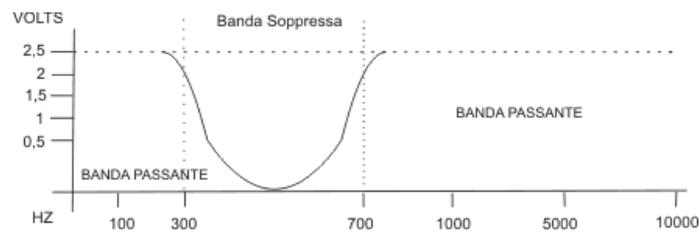


Figura 4.4: Esempio di un filtro soppressore di banda [76]

4.3 Filtrare i suoni respiratori

Entrando più nel dettaglio, durante l'analisi della letteratura attuale abbiamo riscontrato diversi tipi di filtri per l'eliminazione del rumore (denoising) e per eliminare il battito cardiaco. L'eliminazione dei suoni emessi dal cuore è un problema cruciale per il riconoscimento del respiro. Nello scenario previsto abbiamo un file audio del respiro

preso grazie a uno stetoscopio elettronico; molti modelli di stetoscopi già concentrano funzioni per filtrare il battito cardiaco, rilasciando quindi un suono a cui bisogna applicare solo un filtraggio per il denoising, però nell'eventualità in cui uno stetoscopio non sia fornito di queste funzioni bisogna trovare una tecnica di filtraggio ottimale che la maggior parte delle volte consiste nell'applicare più di un filtro al suono che stiamo analizzando. Si può subito notare attraverso l'analisi della letteratura approfondita nel capitolo 6, che il filtro usato maggiormente è quello passa banda, che permette di eliminare le frequenze che non rientrano nel range di frequenza dei suoni respiratori. Per una migliore comprensione dell'analisi dei sistemi dei capitoli successivi si andranno a illustrare alcune tecniche da loro usate.

Downsampling

Un filtro di downsampling o di sottocampionamento riduce la frequenza di campionamento di un segnale. Di solito lo scopo di questo filtro è di ridurre la complessità computazionale dei successivi trattamenti del segnale. Il fattore di sottocampionamento è un numero intero o razionale maggiore di uno. Questo fattore moltiplica il tempo di campionamento o equivalentemente divide la frequenza di campionamento. Una implementazione di un filtro di sottocampionamento deve fare attenzione a non violare le condizioni del teorema sul campionamento di Shannon-Nyquist, altrimenti ci sarà la presenza di aliasing nel segnale di output. Questa tecnica viene usata sia da [27] che da [41].

filtro mediano

Il filtro mediano è un filtro non lineare ed è usato per la rimozione dei picchi di rumore dal segnale ed influenzano di solito solo una piccola percentuale dei campioni ma in modo notevole. L'idea principale di questo filtro è quella di rimpiazzare i campioni del segnale con la mediana dei campioni in un certo intervallo di campioni. Di solito questo intervallo si chiama finestra, la quale si muove su tutto il segnale campione per campione.

funzione finestra

Nell'elaborazione numerica dei segnali una funzione finestra è una funzione che vale zero al di fuori di un certo intervallo. Quando un'altra funzione è moltiplicata per una funzione finestra, anche il prodotto assume valori nulli al di fuori dell'intervallo. Una definizione più generale di funzione finestra non richiede l'annullarsi al di fuori di un intervallo, ma che il prodotto per la funzione di finestra sia una funzione a quadrato sommabile, ovvero che la funzione finestra si annulli in maniera sufficientemente rapida. Le funzioni finestra sono importanti nel progetto dei filtri nell'analisi spettrale.

Siano N l'ampiezza in numero di campioni di una finestra (tipicamente una potenza di 2) ed n un numero intero, che assume valori da 0 ad $N - 1$. Ci sono varie funzioni finestre, ad esempio:

- la finestra rettangolare descritta dall'equazione

$$w(n) = 1$$

- la finestra di Hamming descritta dall'equazione

$$w(n) = 0.54 - 0.46 \cos\left(\frac{2\pi n}{N-1}\right)$$

è stata progettata per minimizzare il livello del lobo laterale, mentre mantiene approssimativamente la stessa larghezza del lobo principale.

- la finestra di Hann o Hanning descritta dall'equazione:

$$w(n) = 0.5 \cos\left(1 - \frac{2\pi n}{N-1}\right)$$

- la finestra di Blackmann descritta dall'equazione:

$$w(n) = \frac{1-0.16}{2} - \frac{1}{2} \cos\left(1 - \frac{2\pi n}{N-1}\right) + \frac{0.16}{2} \cos\left(\frac{4\pi n}{N-1}\right)$$

Capitolo 5

Tecniche risolutive

Questo capitolo introduce le premesse teoriche necessarie per la comprensione di:

- Algoritmi usati nello stato dell'arte.
- Algoritmi usati nel sistema implementato.
- Algoritmi dei quali è stata valutata l'adeguatezza alla soluzione del problema.

L'organizzazione dei contenuti in sezioni è illustrata nella figura 5.1

5.1 Beat detection

Ad un livello intuitivo siamo tutti in grado di capire cos'è il ritmo e siamo anche in grado di capire qual'è il ritmo di una canzone quando l'ascoltiamo.

Il ritmo è definito da una successione di accenti, intendendo con accento il maggior rilievo (variazione di intensità, altezza o timbro) che alcuni suoni hanno rispetto ad altri

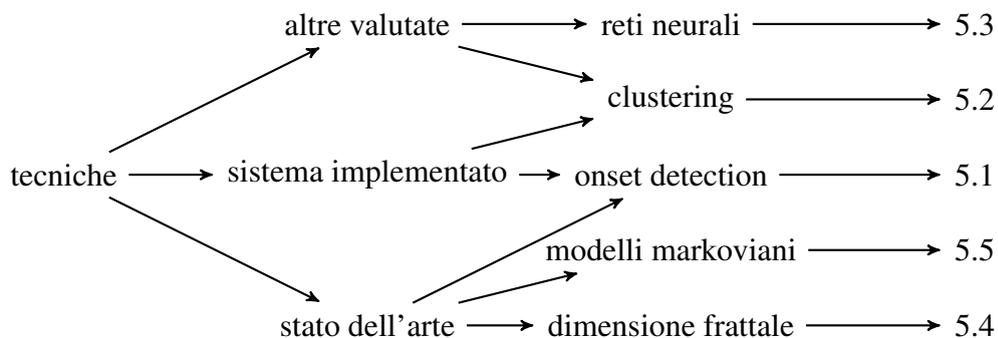


Figura 5.1: Organizzazione dei contenuti nelle sezioni del capitolo

nell'ambito di un brano o una frase musicale. La sequenza degli accenti di un brano musicale tende normalmente a ripetersi a intervalli regolari ed è questa ripetizione che viene chiamata ritmo del brano: la piu' breve sequenza non periodica(quella che viene ripetuta) viene anche chiamata cellula ritmica. Possiamo definire un onset come un accento importante alla caratterizzazione del ritmo. Per trovare il ritmo dobbiamo prima passare dagli onset.

La maggior parte degli algoritmi di onset detection seguono uno schema assimilabile a quello illustrato nella figura 5.2 e descritto nelle sezioni seguenti.

5.1.1 Preprocessing

Il concetto di preprocessing implica la trasformazione del segnale originale con lo scopo di accentuare o attenuare alcuni aspetti del segnale in base alla rilevanza che hanno nel dominio del problema. Questa fase è opzionale e ci sono vari modi di affrontarla ad esempio la separazione nelle frequenze. In questo caso l'informazione viene appunto separata in bande di frequenza.

5.1.2 Reduction

La riduzione è il processo di trasformare il segnale audio in un segnale con una frequenza di campionamento molto piu' ridotta e che manifesta in modo piu' evidente la posizione degli onset. Questo è il cuore degli algoritmi di onset detection. Chiamiamo l'output di questa fase anche detection function. Classifichiamo i metodi di riduzione in due categorie: metodi basati su caratteristiche esplicite del segnale e metodi basati su modelli probabilistici.

Metodi basati su caratteristiche esplicite del segnale. Le caratteristiche esplicite di un segnale possono rientrare nelle categorie: temporali, spettrali, spettrali di fase e spettrotemporali.

Caratteristiche temporali. Nel dominio del tempo del segnale si nota spesso che una occorrenza di un onset di solito è accompagnata da un aumento dell'ampiezza del segnale. Alcuni metodi di onset de-

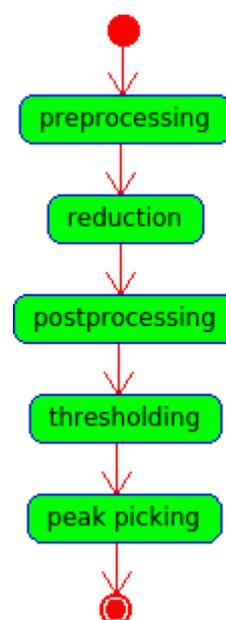


Figura 5.2: Diagramma di attività di un generico algoritmo di beat detection

tection si avvantaggiano di questa proprietà e creano una detection function che segue l'envelope del segnale. Questi metodi danno risultati soddisfacenti nei casi in cui c'è un onset molto forte rispetto al sottofondo.

Metodi basati su modelli probabilistici. I metodi statici per l'onset detection sono basati sull'assunto che il segnale può essere descritto da qualche modello di probabilità. Quindi si può costruire un sistema che fa inferenza probabilistica riguardo i tempi probabili di cambiamenti improvvisi nel segnale in base alle osservazioni disponibili. Il successo di questo approccio dipende dalla somiglianza tra la distribuzione di probabilità descritta dal modello e la distribuzione reale dei dati e si può quantificare usando modelli di selezione Bayesiani. Ci sono vari metodi di questo tipo.

5.1.3 Postprocessing

Questa fase è opzionale e serve per facilitare le fasi successive attraverso l'aumento dell'uniformità e della consistenza di alcuni aspetti della detection function che vengono trasformati in massimi locali isolati e facili da trovare. In questa fase possiamo trovare ad esempio: algoritmi di riduzione del rumore e algoritmi di selezione di parametri utili al calcolo del thresholding nelle fasi successive.

5.1.4 Thresholding

Ci possono essere dei picchi nella detection function che non sono correlati ad un onset di interesse. Quindi è necessario definire un threshold che separi in modo efficace i picchi che corrispondono ad un evento di interesse e quelli che non vi corrispondono. In questa fase un onset che corrisponde ad un evento di interesse è definito come un picco nel quale la detection function supera una certa soglia. Ci sono due approcci principali alla scelta della soglia:

Thresholding fisso. La soglia in questione non varia nel corso della vita dell'algoritmo.

Thresholding adattativo. La soglia in questione varia ed è di solito una certa funzione della detection function. Ad esempio:

Linear smoothing. La soglia è una combinazione lineare degli elementi all'interno della finestra corrente della detection function.

Non linear smoothing. La soglia è una combinazione non lineare degli elementi all'interno della finestra corrente della detection function. Ad esempio una combinazione dei quadrati.

Percentiles smoothing. I metodi precedenti hanno lo svantaggio che quando ci sono dei picchi relativamente grandi seguiti da picchi piccoli, questi ultimi vengono nascosti. Per evitare questo problema si possono usare metodi basati sui percentili ad esempio la mediana.

5.1.5 Peak picking

Nella fase di scelta dei picchi si devono solamente identificare i massimi locali al di sopra del threshold.

5.2 Clustering

Il clustering è il processo di trovare informazioni strutturali all'interno di un insieme di dati. I meccanismi di clustering separano e organizzano dati non etichettati in vari gruppi o cluster, ognuno dei quali contiene dati che sono simili tra loro. Un buon algoritmo di clustering produce cluster nel quale la somiglianza intracluster, cioè tra gli elementi di un cluster, è alta mentre la somiglianza intercluster, cioè tra elementi di cluster diversi, è bassa. Per una discussione approfondita sul clustering si rimanda a [59].

Se vogliamo essere più formali possiamo dire che un m -clustering di un insieme di dati X rispetto ad una distanza o misura di dissimilarità $d : P(X) \rightarrow \mathbb{R}$ e ad una soglia t è C_1, \dots, C_n tale che:

partizione C_1, \dots, C_n è una partizione di X cioè: ogni C_i è un sottoinsieme di X ; i C_i sono a due a due disgiunti; l'unione dei C_i dà X

intersimilarità per ogni i : $d(C_i) < t$

Di seguito descriveremo alcuni tipi di algoritmi di clustering. Ci sono due tipi principali di algoritmi di clustering a seconda dei parametri di input. Una categoria di algoritmi prende in input il numero di cluster, oltre che gli elementi di input e la funzione di dissimilarità, e restituisce un clustering dei dati. Una seconda categoria di algoritmi prende in input un valore di soglia e restituisce un clustering.

Clustering sequenziale. Questi algoritmi leggono gli elementi di input uno per volta in sequenza. In questo tipo di algoritmi non conosciamo a priori il numero di cluster presenti nell'input. Invece prendono in input una soglia θ e il massimo numero di cluster permessi q . L'idea alla base di questi algoritmi è che quando viene letto un elemento, questo viene o assegnato ad un cluster esistente o ad un nuovo cluster a seconda della distanza tra

```

input : A sequence of data  $x_1, \dots, x_n$ 
output: A clustering of the data  $C_1, \dots, C_m$ 

1  $m \leftarrow 1$ 
2  $C_m \leftarrow \{x_1\}$ 
3 for  $i \leftarrow 2$  to  $n$  do
4   find  $k$  such that the distance between  $x_i$  and  $C_k$  is minimum
5   if  $\text{distance}(x_i, C_k) > \text{threshold}$  then
6     increments  $m$ 
7      $C_m \leftarrow \{x_i\}$ 
8   else
9      $C_m \leftarrow C_m \cup \{x_i\}$ 

```

Algorithm 1: Sequential clustering algorithm

l'elemento e i cluster esistenti. Uno schema generale per implementare un algoritmo di clustering sequenziale è dati in figura 1.

Un algoritmo di clustering sequenziale si dice *a cluster sequenziali* se produce cluster sequenziali e cioè è tale che ad ogni iterazione: o si aggiunge l'elemento corrente nello stesso cluster dell'iterazione precedente, o si aggiunge l'elemento in un nuovo cluster.

Se non si conosce in anticipo neanche il valore di soglia allora si può stimare il numero di cluster che ci sono nell'input usando l'algoritmo 2.

```

input : A sequence of data  $x_1, \dots, x_n$ 
output: An estimation of the number of clusters

1 for  $\text{threshold} \leftarrow a$  to  $b$  step  $c$  do
2   run clustering algorithm with current threshold
3   store the number of clusters obtained
4 return the most frequent number of clusters

```

Algorithm 2: Algoritmo di stima del numero dei cluster. Usato nella sezione 7.4

Clustering gerarchico. Il clustering gerarchico è un approccio di clustering che mira a costruire una gerarchia di cluster. Le strategie per il clustering gerarchico sono tipicamente di due tipi:

Agglomerativo questo tipo di algoritmo produce una sequenza di clustering che hanno numero di cluster decrescente.

Divisivo questo tipo di algoritmo produce una sequenza di clustering che hanno numero di cluster crescente. Ad ogni iterazione un algoritmo di questo tipo prende i cluster risultati dall'iterazione precedente e li divide.

Algoritmi di clustering basati su una funzione di ottimizzazione del costo. In questi algoritmi il numero di cluster è fisso. Si cerca di assegnare gli elementi di input ai cluster in modo da ottimizzare il valore di una funzione di costo.

5.3 Reti neurali

Le *reti neurali artificiali* sono modelli matematici e computazionali della corteccia cerebrale. Gli elementi di base di una rete neurale artificiale sono:

neuroni artificiali o unita'

collegamenti I neuroni artificiali sono connessi attraverso *collegamenti*. Ogni collegamento ha un *peso* numerico, in particolare il collegamento tra l'unita' i e l'unita' j ha peso $w_{i,j}$.

Una certa unita' i può inviare un *segnale* o *attivazione* a_i su tutti i suoi collegamenti. Il peso di un collegamento determina la forza e il segno del segnale. Assumiamo l'esistenza di una certa unita' di input che manda il segnale a_0 . Ogni unita' si comporta nel modo seguente:

1. calcola la somma pesata dei suoi input

$$in_j = \sum_{i=0}^n w_{i,j} a_i$$

2. calcola il proprio output in base al valore calcolato al punto precedente e ad una *funzione di attivazione* g :

$$a_j = g(in_j)$$

La funzione di attivazione di solito è una funzione del tipo:

$$g(x) = \begin{cases} 0 & \text{if } x < 0 \\ 1 & \text{if } x > 0 \end{cases} \quad \text{o} \quad g(x) = \frac{1}{1+e^{-x}}$$

Ci sono due tipi di reti neurali:

feed forward I collegamenti tra unita' sono unidirezionali. In questo caso la rete neurale è un grafo diretto aciclico. Una rete neurale di tipo feed forward rappresenta una funzione dei suoi input nel senso che non ha uno stato interno o memoria. Una rete feed forward è di solito divisa in *layers* o livelli. Ogni livello è un insieme di unita', ogni unita' riceve input solo da unita' del livello immediatamente precedente. Una rete con un solo livello si dice anche *single-layer* o *monolivello*.

recurrent network Ci possono essere dei cicli nei collegamenti. Una rete di questo tipo rappresenta un sistema dinamico che può anche oscillare o avere un comportamento caotico. Inoltre la risposta di una recurrent network dipende dallo stato iniziale della rete e dall'input quindi questo tipo di rete può supportare una memoria a breve termine.

5.4 Dimensione frattale

In geometria frattale la dimensione frattale è una quantità statistica che da una indicazione di quanto completo appare un frattale per riempire lo spazio. La dimensione frattale inoltre è una misura della complessità di un insieme di dati. Viene usata per analizzare segnali in un vasto numero di ricerche scientifiche. Una proprietà della dimensione frattale è che è indipendente dal contenuto di energia nel segnale ma è dipendente dalle frequenze. Ci sono molti modi di definire matematicamente la dimensione frattale ad esempio quello illustrato in [35]:

$$D_{\sigma} = \lim_{\delta t \rightarrow 0} \frac{\log(\text{Var}(\delta S, \delta t))}{2 \cdot \log(\delta t)}$$

In questa formula: δt è il valore assoluto di una variazione temporale $|t_2 - t_1|$ mentre δS è la variazione del segnale nel tempo $S(t_2) - S(t_1)$.

5.5 Catene di triple Markoviane

5.5.1 Nozioni di base di probabilità

Evento. Spazio Campionario. Dato un esperimento casuale, definiamo un evento elementare ω come uno dei possibili esiti dell'esperimento stesso. Definiamo un evento come un qualsiasi insieme di eventi elementari. Definiamo lo spazio campionario come l'insieme di tutti gli eventi elementari. Uno spazio campionario si indica di solito con il nome Ω .

σ -algebra Dato un insieme Ω , si definisce σ -algebra su Ω una famiglia F di sottoinsiemi di Ω tale che:

- Ω appartiene ad F .
- L'insieme F è chiuso rispetto a complementare.
- L'insieme F è chiuso rispetto ad unione numerabile.

Spazio misurabile. Insiemi misurabili. Uno spazio misurabile è una coppia (Ω, F) tale che F è una σ -algebra su Ω . Gli elementi di F sono detti insiemi misurabili in Ω .

Topologia. Spazio topologico. Dato un insieme X , una topologia su X è un insieme T di sottoinsiemi di X tale che:

- L'insieme vuoto e X appartengono a T .
- L'unione di una quantità arbitraria di insiemi appartenenti a T appartiene a T .
- L'intersezione di un numero finito di insiemi appartenenti a T appartiene a T .

Uno spazio topologico è una coppia (X, T) , dove X è un insieme e T è una topologia su X . In uno spazio topologico gli insiemi che costituiscono T si dicono aperti in X .

Funzione misurabile. Sia (X, F) uno spazio misurabile e (Y, T) uno spazio topologico. Una funzione $f : X \rightarrow Y$ viene detta misurabile se la controimmagine di ogni elemento di T è in F .

Variabile casuale o aleatoria. Sia dato uno spazio campionario Ω su cui è definita una misura di probabilità ν , una variabile casuale è una funzione misurabile dallo spazio campionario a uno spazio misurabile.

Processo stocastico. Si definisce processo stocastico una famiglia di variabili aleatorie

$$\{X(t) \mid t \in T \subseteq \mathbf{R}^+\}$$

dipendenti dal tempo, definite su un unico spazio campione Ω finito e che assumono valori in un insieme definito spazio degli stati del processo. Un processo stocastico è quindi un insieme di funzioni che evolvono nel tempo, ognuna delle quali è associata ad un determinato elemento dello spazio campione, così che il risultato di un esperimento casuale corrisponde di fatto all'estrazione di una di queste funzioni. Un processo stocastico discreto è un processo stocastico nel quale T è un sottoinsieme dei numeri naturali.

5.5.2 Processo di Markov

Un processo stocastico markoviano o processo di Markov è un processo stocastico nel quale la probabilità di transizione che determina il passaggio ad uno stato di sistema dipende unicamente dallo stato di sistema immediatamente precedente (proprietà di Markov) e

non dal come si è giunti a tale stato. Formalmente questo può essere scritto come

$$P(X_{n+1} \leq x_{n+1} |_{i=1}^n X_i \leq x_i) = P(X_{n+1} \leq x_{n+1} | X_n \leq x_n)$$

Questa è detta proprietà di Markov, o condizione di assenza di memoria.

5.5.3 Catena di Markov

Una catena di Markov è un processo di Markov con spazio degli stati discreto, quindi si tratta di un processo stocastico che assume valori in uno spazio discreto e che gode della proprietà di Markov. L'insieme di spazio degli stati può essere finito o infinito (numerabile). Nel primo caso si parla di catena di Markov a stati finiti. Una catena di Markov può essere tempo-continua o tempo-discreta, in base all'insieme di appartenenza della variabile tempo (continuo o discreto). Formalmente, una catena di Markov è una sequenza di variabili aleatorie X_1, X_2, \dots , che soddisfa la condizione di Markov.

5.5.4 Modelli nascosti di Markov

In un modello nascosto di Markov abbiamo due processi stocastici: il processo di interesse inosservabile $X = (X_s)_{s \in \mathcal{S}}$ e il processo osservabile $Y = (Y_s)_{s \in \mathcal{S}}$. Un modello nascosto di Markov è (X, Y, T, E, π) tale che:

$T : X \times X \rightarrow [0; 1]$ è la probabilità di transizione. In particolare

$$p(X_{t+1} = x_i | X_t = x_j) = T(x_i, x_j)$$

è la probabilità che il processo nascosto passi dallo stato x_i allo stato x_j .

$E : X \times Y \rightarrow [0; 1]$ è la probabilità condizionata di X rispetto ad Y . Cioè se osserviamo un certo valore y per il processo osservabile Y allora E ci dice qual'è la probabilità che il processo nascosto assuma certi valori.

$\pi : X \rightarrow [0; 1]$ è la probabilità iniziale che il processo nascosto si trovi in un certo stato.

5.5.5 Catena di coppie markoviane

Supponiamo che $X = (X_1, \dots, X_n)$ e $Y = (Y_1, \dots, Y_1)$ siano due processi stocastici, nei quali ogni X_i ha valori in un insieme finito $\Omega = \{\omega_1, \dots, \omega_n\}$ e ogni Y_i ha valori reali. Sia $Z = (Z_1, \dots, Z_n)$ il processo stocastico delle coppie di X, Y , cioè $Z_i = (X_i, Y_i)$. I processi

X e Y sono una catena di coppie markoviane se Z è un processo di Markov cioè quando

$$p(z) = p(z_1)p(z_2|z_1) \cdots p(z_n|z_{n-1})$$

5.5.6 Catena di triple markoviane

Le catene di triple markoviane sono una estensione delle catene di coppie di Markov nelle quali la distribuzione di probabilità di (X, Y) è una distribuzione di probabilità marginale di una catena di Markov (X, U, Y) . Anche nelle catene di triple markoviane cerchiamo X a partire da Y , il processo S serve solo come aiuto al calcolo. Sia $S = N$ l'insieme dei numeri naturali senza lo zero e siano $X = (X_n)_{n \in N}$, $Y = (Y_n)_{n \in N}$, $U = (U_n)_{n \in N}$ tre processi, nei quali le variabili X_i assumono valori in uno spazio campionario $\Omega = \{\omega_1, \dots, \omega_k\}$, le variabili Y_i sono a valori reali e le variabili U_i assumono valori nell'insieme $\Lambda = \{\lambda_1, \dots, \lambda_m\}$. Con lo scopo di semplificare la notazione, poniamo $T_n = (X_n, U_n, Y_n)$, $Z_n = (X_n, Y_n)$ e $V_n = (X_n, U_n)$ e chiamiamo T, Z e V i processi corrispondenti. Diremo che Z è una catena di triple di Markov se esiste un processo U nel quale T è una catena di Markov. In questo caso allora (V, Y) è una catena di Coppie di Markov. Possiamo scrivere la distribuzione di probabilità di (V_i, Y) come

$$p(v_i, y) = \alpha^i(v_i)\beta^i(v_i)$$

con $\alpha^i(v_i)$ e $\beta^i(v_i)$ definite rispettivamente come

$$\alpha^i(v_i) = p(y_1, \dots, y_{i-1}, y_i, v_i) \quad \beta^i(v_i) = p(y_{i+1}, \dots, y_n, v_i, y_i)$$

Parte II

Stato dell'arte e analisi delle metodologie

Capitolo 6

Stato dell'arte

Nell'esame della letteratura sull'argomento analisi automatica dei suoni respiratori, si possono riconoscere le seguenti funzionalità principali:

Classificazione Classificare i suoni respiratori secondo quanto detto in 2.6.

Riconoscimento Localizzare gli inizi (in seguito detti anche onset) delle fasi respiratorie.

Diagnosi Stabilire la presenza o l'assenza di alcune patologie dell'apparato respiratorio.

Flusso Stimare il flusso respiratorio in valori relativi al flusso nullo e al flusso massimo.

Suoni cardiaci Eliminazione dei suoni cardiocircolatori dai suoni respiratori.

Esiste anche una certa letteratura sull'argomento più generale: analisi della respirazione attraverso segnali biomedici. In questi studi si usano segnali provenienti da vari sensori, ad esempio: elettrocardiografo, fotoplethysmografo e sensori di conducibilità applicati ad una fascia elastica sull'addome.

Nessuno degli articoli esaminati propone di suonare una allarme se la durata della pausa respiratoria è troppo lunga.

Contenuto del capitolo Questo capitolo si concentra solo sui sistemi che riconoscono le fasi respiratorie perché questi possono essere adattati al problema della localizzazione di apnee troppo lunghe.

6.1 Algoritmi di riconoscimento delle fasi e dei cicli respiratori

Ricordiamo che per fase respiratoria intendiamo un evento di inspirazione, di espirazione oppure la pausa che intercorre tra due fasi respiratorie. Con il termine ciclo respiratorio

intendiamo una fase di inspirazione seguita da una fase di espirazione seguita da una pausa. Si parla di ciclo respiratorio perché naturalmente è una ripetizione continua delle tre fasi.

6.1.1 *Automated respiratory phase and onset detection using only chest sound signal [27]*

In questo articolo viene trattato un algoritmo per la localizzazione delle fasi di respirazione attraverso i suoni polmonari.

input I dati considerati in input sono tre registrazioni fatte su due soggetti sani (un uomo e una donna) di anni 25 e 30. In una prima fase i soggetti hanno cercato di mantenere una frequenza di respiro breve, successivamente ai soggetti è stato chiesto di aumentare la frequenza. La frequenza di campionamento delle registrazioni è di $10240Hz$. In questo studio si sono usati solo i suoni registrati sul petto dei soggetti sul quarto interspazio con lo scopo di avere una distinzione ottima tra le varie fasi della respirazione.

algoritmo Il segnale di input passa attraverso le seguenti fasi in pipeline:

1. Riduzione della sequenza di campionamento. Gli spettri di potenza dei suoni del cuore e dei polmoni normalmente hanno componenti di frequenza inferiori a $500Hz$. Pertanto, la frequenza di campionamento è stata abbassata a $1000Hz$ poiché una larghezza di banda di $500Hz$ è sufficiente per catturare il suono polmonare e la relativa fase respiratoria.
2. Filtro passabanda per ridurre in parte i suoni del cuore e il rumore e per isolare i suoni polmonari.
3. Filtro non lineare per eliminare completamente alcuni picchi rimasti nel segnale per via dei suoni del cuore. A causa della sovrapposizione degli spettri di polmone e toni cardiaci, il filtro passabanda non elimina completamente i toni cardiaci. Si sceglie quindi di usare un filtro non lineare per far sì che i suoni inerenti al cuore vengano eliminati invece di essere levigati, come succederebbe con un normale filtro lineare.
4. Ulteriore filtro passabanda impostato sulla frequenza tipica delle fasi respiratorie. L'output appena ottenuto viene sottoposto a un ulteriore filtro passabanda impostato sulla frequenza tipica delle fasi respiratorie.

5. Zero crossing usato per stimare gli inizi delle fasi respiratorie. I cambiamenti di fase respiratoria avvengono nei passaggi attraverso lo zero: dove il segno del segnale cambia da positivo a negativo inizia una fase di espirazione mentre un cambiamento di segno da negativo a positivo corrisponde ad un inizio di inspirazione.

conclusioni La media dell'errore di questo algoritmo non supera i $100ms$ di ritardo nel riconoscimento degli onset, ed è un intervallo di tempo assolutamente accettabile ai fini della diagnosi in tempo reale della sindrome da apnea del sonno. Tuttavia l'algoritmo deve essere modificato per trattare variabili pattern respiratori e nel caso in cui i soggetti dormano, alcuni parametri devono essere modificati. Lo studio non riporta nessun dato sulla complessità computazionale e sull'implementazione dell'algoritmo.

6.1.2 *Breath Analysis of Respiratory Flow using Tracheal Sounds* [28]

In questo studio gli autori studiano le differenze che ci sono tra la fase inspiratoria e la fase espiratoria in due quantità relative ad un segnale tracheale filtrato con un filtro passa banda. Queste due quantità sono la media e la varianza logaritmica dell'energia. Lo studio usa uno spirometro per misurare il flusso. Il flusso viene diviso in base al valore assoluto in: basso, medio, alto e molto alto. Questo algoritmo quindi non ricava il flusso a partire dal suono ma è utile per comprendere e sviluppare algoritmi di riconoscimento delle fasi respiratorie.

input I dati presi in input sono delle registrazioni di suoni tracheali registrati su nove soggetti sani e non fumatori i quali non hanno mai avuto gravi malattie respiratorie. Inoltre lo studio aveva a disposizione anche il flusso d'aria registrato attraverso uno spirometro con pneumotacografo

algoritmo L'algoritmo ha due flussi di esecuzione indipendenti, il primo è il seguente:

1. Filtro passa alto con frequenza di taglio di $70Hz$ per rimuovere il rumore a bassa frequenza.
2. Nelle fasi seguenti l'algoritmo considera solo le porzioni del suono registrate quando il segnale del flusso era al di sotto del 20% del flusso medio o al di sopra del 20% di esso. Perché in queste condizioni il suono tracheale si può considerare stazionario.
3. Lo spettro di potenza dei suoni della trachea è stato calcolato in una finestra di $50ms$ (512 campioni) con il 75% di sovrapposizione tra finestre successive. Per ogni fase

respiratoria durante la quale c'era un diverso flusso d'aria, è stata calcolata la media della potenza dei suoni tracheali in decibel entro sei predefinite bande di frequenza: da 70 a 300Hz, da 300 a 450Hz, da 450 a 600Hz, da 600 a 800Hz, da 800 a 1000Hz e da 1000 a 1200Hz.

4. Dato che le intensità dei suoni respiratori variano da soggetto a soggetto, per ogni soggetto i valori calcolati in precedenza sono stati normalizzati rispetto al valore massimo.
5. Si è poi calcolata la media dei valori normalizzati tra soggetti diversi per ogni fase respiratoria, inoltre sono state calcolate la media e l'errore standard per diversi tassi di flusso e intervalli di frequenza.

mentre il secondo flusso di esecuzione è:

1. Il segnale dei suoni della trachea sono stati filtrati attraverso un filtro passa alto nelle stesse frequenze menzionate in precedenza.
2. Il segnale filtrato è stato in seguito segmentato in finestre di dimensione 50ms (512 campioni) con il 75% di sovrapposizione tra finestre successive usando una finestra di Hanning.
3. Si calcola il logaritmo della varianza dei segmenti precedenti.
4. In ciascuna finestra il valore precedente viene normalizzato rispetto al valore massimo per ridurre le interferenze dei suoni del cuore
5. In seguito viene calcolata la media all'interno delle diverse bande di frequenza e diversi valori del flusso d'aria.

conclusioni Da una analisi dello spettrogramma dei suoni tracheali, si può vedere che l'intensità del suono tracheale aumenta con l'aumentare del valore assoluto del flusso. Inoltre anche la varianza logaritmica normalizzata e la media di potenza normalizzata seguono i cambiamenti nel valore assoluto del flusso. Nello spettrogramma, nella varianza logaritmica normalizzata e nella media normalizzata della potenza sono evidenti le transizioni di fase respiratoria. Quando il flusso era medio o alto, si ha una maggiore differenza di media dell'energia normalizzata tra la fase inspiratoria ed espiratoria nella banda di frequenze dai 300 ai 450Hz. Inoltre questa banda di frequenza ottiene la seconda maggior differenza di media dell'energia normalizzata tra la fase inspiratoria e la fase espiratoria quando il flusso è basso o molto alto. Quindi questo intervallo di frequenza è stato scelto come ottimale per esaminare i cambiamenti nella media della potenza rispettivamente alle fasi respiratorie.

6.1.3 *Acoustical respiratory signal analysis and phase detection* [33]

Questo articolo propone un approccio di modellazione statistica per il riconoscimento delle fasi respiratorie.

input L'input usato per i test è preso da due segnali respiratori di due soggetti nell'intervallo di frequenza dai 250 ai 312.5Hz. Il primo dura 16s e contiene circa tre cicli di respirazione, il secondo dura 8 secondi e contiene anch'esso tre cicli.

algoritmo Prima di tutto l'analisi del suono viene fatta nel dominio dei pacchetti wavelet per incrementare l'accuratezza della determinazione dei suoni. Il sistema di riconoscimento è implementato attraverso una segmentazione del segnale in tre parti alle quali viene associata una etichetta che può essere: inspirazione, espirazione e transizione. Viene adottata una rete Bayesiana, usando una versione con vincoli di una catena di Markov a triple.

conclusioni I risultati sono buoni ma è presente qualche errore di classificazione. Questo a causa del rumore ambientale nella prima registrazione e a causa della forte variazione di intensità tra le fasi espiratorie nella seconda registrazione.

6.1.4 *Computerized acoustical respiratory phase detection without air-flow measurement* [38]

Questo studio sviluppa un metodo per riconoscere le fasi respiratorie a partire dai suoni prodotti dall'apparato respiratorio. Viene implementato anche un programma MATLAB.

input Sono stati studiati 21 soggetti di età dai 4 ai 51 anni. I soggetti sono stati divisi in gruppi di età: 7 bambini e 4 bambine con una età media di 10 anni e 4 uomini e 6 donne con una età media di 32 anni. Tutti i soggetti godevano di buona salute e non avevano infezioni delle vie respiratorie nelle quattro settimane precedenti alla registrazione. Sei accelerometri sono stati usati per registrare i suoni respiratori. I dispositivi di registrazione erano attaccati sulla trachea e su varie parti del petto. Viene anche registrato in modo simultaneo, il flusso d'aria attraverso uno pneumotacografo con un trasduttore di differenza di pressione. L'algoritmo prende in input solo i suoni della registrazione mentre il flusso d'aria registrato dallo pneumotacografo serve esclusivamente per valutare la qualità dei risultati dell'algoritmo.

algoritmo

1. Il segnale è stato amplificato, filtrato attraverso un filtro passa banda con banda dai 50 ai 200Hz e digitalizzato ad una frequenza di campionamento pari a 10240Hz.
2. Una finestra di Hanning è stata usata per segmentare il segnale sonoro in finestre di 2048 campioni con il 50% di sovrapposizione tra segmenti successivi.
3. Lo spettro di potenza di ogni segmento è stato calcolato con una trasformata veloce di Fourier. Le bande di frequenza usate per calcolare la potenza media di ogni segmento erano: dai 150 ai 300Hz, dai 300 ai 450Hz, dai 450 ai 600Hz e dai 600 ai 1200Hz. L'algoritmo in modalità di esecuzione semiautomatica offre all'utente la possibilità di visualizzare i dati e di identificare eventuali suoni accidentali da rimuovere.
4. Per ogni segmento di 200ms viene calcolata la media della potenza del segnale registrato sulla parete del petto nella posizione con la quale si ottiene la maggiore differenza di suono tra inspirazione ed espirazione. Questo è risultato in un segnale nel quale i picchi corrispondono al massimo flusso d'aria durante l'inspirazione.
5. È stata usata una finestra mobile per riconoscere i picchi inspiratori per il segnale completo. Per venire in contro alla variabilità nella frequenza respiratoria, la lunghezza della finestra è stata scelta in modo tale da approssimare la durata di un ciclo di respirazione e la finestra è stata spostata in incrementi che approssimano metà di un ciclo di respirazione. Dato che il suono polmonare è più forte durante l'inspirazione che durante l'espirazione, l'algoritmo usa i picchi del suono polmonare per determinare i picchi di inspirazione. Invece i suoni tracheali sono forti sia durante l'inspirazione che durante l'espirazione e quindi li usa per determinare gli onset delle fasi respiratorie.
6. Infine l'algoritmo classifica come inspirazione quella regione temporale che è compresa tra due onset tracheali e che contiene un picco polmonare. Tutto il resto viene classificato come espirazione.

conclusioni Il software ha ottenuto una accuratezza massima nella stima delle fasi respiratorie senza l'uso dei dati sulla misurazione diretta del flusso d'aria.

6.1.5 *Respiratory onset detection using variance fractal dimension* [35]

Anche [35] si occupa del problema di sviluppare un metodo acustico non invasivo per riconoscere le fasi respiratorie senza una misurazione diretta del flusso d'aria e lo fa usando

la dimensione frattale. Questa è una misura della complessità di un segnale. Una proprietà della dimensione frattale è che è indipendente dalla potenza del segnale. Questo sistema si basa sull'assunto che durante la transizione di fase respiratoria, il segnale sonoro ha un comportamento caotico a causa del cambiamento di momento nel flusso d'aria mentre questo cambia di direzione. Quindi, ipotizza che la varianza della dimensione frattale del suono respiratorio abbia un picco negli onset dei cicli respiratori.

input Lo stesso input utilizzato dallo studio [38], il quale è già stato descritto in precedenza.

algoritmo Per trovare gli onset dei cicli respiratori, viene calcolata la varianza della dimensione frattale usando un segmento di 128 campioni pari a $12.5ms$ con il 50% di sovrapposizione tra segmenti adiacenti. In seguito viene usata una finestra di lunghezza pari alla durata approssimativa della metà di un respiro cioè $0.7s$ con lo scopo di riconoscere i picchi nella varianza della dimensione frattale.

conclusioni Da un confronto tra il riconoscimento degli onset con il reale flusso d'aria i risultati mostrano che l'intervallo di errore va dai $31ms$ ai $49ms$. L'aspetto positivo di questo algoritmo è che fa una analisi esclusivamente nel dominio del tempo del segnale.

6.1.6 Automated respiratory phase detection by acoustical means [37]

Lo studio [37] si concentra sull'automazione del processo di riconoscimento per via acustica delle fasi della respirazione senza l'ausilio di misure del flusso. Viene implementato un programma in C++ dotato di una interfaccia grafica in grado di lavorare in modo semiautomatico o completamente automatizzato.

input L'input è costituito da un insieme di 17 suoni registrati sulla trachea e sul torace (sul secondo interspazio sinistro medioclavicolare e sul terzo interspazio destro medioclavicolare) di 11 soggetti sani con età dai 4 ai 35 anni. Inoltre erano disponibili anche i dati del flusso d'aria misurati attraverso uno pneumotacografo.

algoritmo

1. Il segnale del suono della respirazione viene segmentato in segmenti di lunghezza $100ms$, ognuno con una sovrapposizione del 50% tra segmenti adiacenti.
2. Per ogni segmento viene calcolato lo spettro di potenza usando una trasformata veloce di Fourier.

3. I segnali del torace vengono filtrati lasciando una banda di frequenza dai $150Hz$ ai $300Hz$ mentre i segnali della trachea vengono filtrati lasciando una banda di frequenza dai $150Hz$ ai $600Hz$
4. Viene presa la potenza media dei segnali.
5. Per riconoscere i picchi di inspirazione dai segnali del petto, inizialmente, viene calcolata la pendenza per ogni campione, prendendo la differenza tra punti adiacenti. I campioni che hanno una differenza positiva con il campione precedente e una differenza negativa con il campione successivo sono i possibili picchi.
6. Per affinare la ricerca dei picchi inspiratori, viene usata una piccola finestra mobile. Dato che il valore del flusso può variare molto da respiro a respiro, la finestra è stata scelta in modo da coprire un ciclo completo di respirazione nel segnale (approssimativamente $2s$ quindi 20 campioni). Una finestra di questo tipo è applicata ai picchi determinati al punto precedente. Vengono determinati i punti massimi della finestra che hanno almeno una deviazione standard al di sopra della media. Ci si aspetta che i picchi di inspirazione siano molto più alti dei picchi di espirazione per lo spettro di potenza dei suoni registrati sul petto. Poiché i possibili picchi possono consistere sia in picchi di inspirazione che in picchi di di espirazione, bisogna usare una soglia per eliminare i picchi di espirazione. L'algoritmo di riconoscimento dei picchi è progettato per essere usato sia in modalità automatica che in modalità semiautomatica. Nella modalità semiautomatica, gli utenti possono aggiungere i picchi mancanti o eliminare i picchi riconosciuti dopo aver dato un'occhiata alla potenza media dello spettro dei segnali.
7. Per trovare gli onset del respiro dai segnali tracheali, prima, si calcola la pendenza ad ogni campione.
8. In questo caso, i campioni che hanno una differenza negativa rispetto ai campioni precedenti e una differenza positiva rispetto ai campioni successivi vengono selezionati come potenziali onset. Dato che in media la durata delle fasi respiratorie è approssimativamente $1s$, la distanza tra un picco e l'onset più vicino è di circa $500ms$ o 5 campioni. Per stare sicuri, viene considerata troppo vicina una differenza di 2 campioni ($200ms$), dunque i potenziali onset riconosciuti che sono considerati troppo vicini ad uno dei picchi precedenti vengono esclusi dalla selezione.
9. Per calcolare il valore medio e la deviazione standard di ogni campione, si usa una finestra mobile di lunga 10 campioni (approssimativamente metà della durata di un

respiro). I potenziali onset con deviazione standard almeno 0.5 più piccola della media locale vengono scelti per una successiva analisi.

10. Nel processo di riconoscimento dei picchi tramite i risultati ottenuti viene stimata la distanza media tra due picchi inspiratori. La lunghezza media di una fase respiratoria è calcolata come la metà della media della distanza tra i picchi. Questa informazione è usata per raggruppare i potenziali onset in cluster. Si classifica come onset del respiro solo il punto minimo all'interno di ogni cluster.
11. Per massimizzare l'accuratezza del programma di riconoscimento della fase respiratoria viene fatta un'ulteriore ottimizzazione aggiungendo un algoritmo di auto-correzione. Questo algoritmo confronta gli onset trovati con i picchi stimati per assicurarsi che esistano solo due onset tra due picchi inspiratori. Gli onset che non vengono riconosciuti durante il processo di selezione precedente vengono aggiunti mentre gli onset extra che esistono tra due picchi vengono rimossi. Con i picchi stimati e gli onset si possono prevedere le fasi respiratorie o la direzione del flusso d'aria. Gli intervalli tra due onset con un picco sono inspiratori mentre gli intervalli senza picchi tra gli onset sono espiratori.

conclusioni Per calcolare le prestazioni complessive del sistema sono stati valutati tre parametri:

- Percentuale delle fasi respiratorie riconosciute in modo corretto dall'algoritmo di riconoscimento dei picchi.
- Percentuale degli onset riconosciuti in modo corretto dall'algoritmo di riconoscimento degli onset.
- Media delle differenze in millisecondi tra gli onset riconosciuti e quelli trovati direttamente dalla misurazione del flusso.

Nella modalità semiautomatica i picchi di inspirazione vengono riconosciuti con una accuratezza del 100%. In modalità completamente automatica l'accuratezza media è del 93% con una deviazione standard del 7%. Confrontato con il flusso misurato direttamente, in media, c'è solo un ritardo di 118ms di riconoscimento tra gli onset rilevati e quelli reali. Dato che è inevitabile un minimo flusso critico all'interno della trachea per generare un suono udibile dallo stetoscopio, ci si aspetta una differenza tra gli onset reali dei cicli della respirazione ottenuti attraverso uno pneumotacografo e gli onset stimati dal sistema. Studi precedenti mostrano che questa differenza è di circa 40ms[38]. Questo sistema stima gli onset delle fasi respiratorie e la direzione del flusso d'aria ma non tenta di stimare anche il valore assoluto del flusso.

6.1.7 A Software Toolkit for Acoustic Respiratory Analysis [41]

L'obiettivo principale di questo studio è la realizzazione di un software per il riconoscimento delle fasi respiratorie e per la classificazione dei suoni respiratori.

input Usa i suoni tracheali di cinque soggetti sani, registrati con uno stetoscopio elettronico ad una frequenza di campionamento di $22050Hz$.

algoritmo

1. La fase di estrazione dei suoni polmonari rimuove la componente del segnale le cui frequenze si trovano al di fuori della banda di frequenza che va dai 100 ai $2500Hz$, dato che i suoni respiratori tracheali di solito non cadono al di fuori di questa banda. Per farlo viene usato un filtro passa banda appropriato. Questo filtro usa una finestra di Blackman di dimensione pari a un dodicesimo della frequenza di campionamento del segnale di input. Questo filtro è importante soprattutto per rimuovere i suoni ad alta frequenza che sono di origine cardiocircolatoria.
2. Una fase successiva prende il valore assoluto del segnale
3. Il segnale attraversa una fase di sottocampionamento di un fattore 60 applicando una finestra mobile di 60 campioni che prende un campione ogni 60 . Lo scopo della riduzione della frequenza di campionamento è quello di ridurre il carico computazionale. Lo scopo della fase di trattamento del segnale è quello di trovare la forma generale del segnale. Il sottocampionamento non degrada la forma di questa a patto che la nuova frequenza di campionamento sia almeno il doppio della frequenza di respirazione che si può evincere tipicamente dai suoni respiratori tracheali. Se assumiamo che la durata tipica di una fase respiratoria è di un secondo allora l'envelope del segnale sarà di $1Hz$. Questo significa che la nuova frequenza di campionamento deve essere di almeno $2Hz$. In questo caso la nuova frequenza di campionamento è di circa $367Hz$.
4. Il segnale attraversa una fase di riduzione del rumore che usa un filtro a mediana per ridurre il valore assoluto dei picchi di durata più breve. Qui il segnale viene prima partizionato in segmenti contigui di $0.5s$. Per ogni segmento viene calcolato il valore mediano. Per ogni campione se il valore del campione supera una soglia di cinque volte il valore mediano del segmento allora il campione è azzerato perché potrebbe essere rumore.
5. Il processo precedente viene ripetuto un'altra volta.

6. Il segnale viene sottocampionato di nuovo di un fattore due.
7. Il segnale passa attraverso un filtro passa basso con una finestra di Hamming di $50ms$. Lo scopo di questo filtro è di ridurre i suoni che hanno una varianza alta e una frequenza alta.
8. Applicare un filtro passa basso che ha una impulse response di lunghezza fissa e arbitraria su una segnale non stabile può portare a risultati indesiderati. Poiché la durata delle fasi respiratorie spazia su un grande intervallo di valori, un filtro che va bene in un caso può non andare bene per un altro caso. Questo problema viene affrontato usando un filtro con una impulse response di lunghezza che dipende dalle caratteristiche del segnale di input. La lunghezza w della finestra usata dal filtro dovrebbe approssimare la media della durata di una fase respiratoria nel segnale di input. Per determinare la lunghezza del filtro, si parte con una fase di detrend, nella quale la media globale del segnale di input viene sottratta da ogni campione in modo tale che la media dell'output sia zero.
9. Per approssimare la media della durata di una fase respiratoria ci si basa sull'assunto che questa sia proporzionale alla ampiezza dei picchi. La fase di calcolo della durata media di una fase respiratoria, approssima questa quantità trovando il picco più ampio che interseca la linea di ampiezza zero. Supponiamo che il segmento originato da questa intersezione abbia lunghezza L . Allora la dimensione w della finestra è $w = \text{ceil}(L/2) + 2$. Questa scelta è giustificata solo empiricamente.
10. Solo a questo punto possiamo applicare il filtro passa basso con una finestra di Hamming di lunghezza w .
11. Usiamo informazioni nel dominio del tempo per classificare il suono di input in respiratorio o non respiratorio. Questa fase usa il segnale della fase precedente e la dimensione w della finestra che approssima la durata media di una fase respiratoria.
12. Il segnale viene diviso in segmenti contigui di lunghezza $3w$
13. Viene calcolata l'energia totale E di un segmento sommando i valori
14. L'energia E viene confrontata ad una soglia. La soglia è uguale all'energia di base nel segmento. Definiamo l'ampiezza di base come la media delle ampiezze nell'area del segnale nella quale non c'è respirazione e definiamo l'energia di base $E_{baseline}$ come la lunghezza del segmento moltiplicata per l'ampiezza di base. Viene usato un valore fisso determinato empiricamente per l'ampiezza di base. Se l'energia E supera l'energia di base $E_{baseline}$ allora classifichiamo il segmento come

respiro altrimenti lo classifichiamo come non respiro. Viene prodotta l'etichetta 1 se c'è respiro altrimenti viene prodotta l'etichetta 0.

15. La fase precedente produce una sequenza di etichette. Etichette uguali e contigue vengono raggruppate.
16. Un respiro è un gruppo di etichette 1 seguito da un gruppo di etichette 0.

conclusioni Il sistema ha una accuratezza del 97%

6.2 Classificazione degli algoritmi

In questa sezione ci occupiamo della classificazione degli algoritmi visti nello stato dell'arte. La prima distinzione che si può fare è quella riguardante la metodologia usata nella soluzione del problema. In particolare si nota che gli articoli [27] [38] [37] [41] usano tutti una tecnica di beat detection per analizzare il respiro e ricavare gli onset della respirazioni, applicano dei filtri passabanda per l'eliminazione del rumore e del battito cardiaco e usando un threshold adattivo per rilevare gli onset.

In particolare abbiamo che i sistemi [38] [37] sono molto simili tra loro perché entrambi dopo aver applicato dei filtri passabanda suddividono l'onda in segmenti e ne calcolano la potenza media infine tramite una finestra mobile con una durata pari a un ciclo respiratorio riconoscono gli onset della respirazione. Le differenze tra questi due sistemi sono per l'appunto minime come per esempio la scelta della dimensione dei segmenti di analisi: il primo utilizza un segmento di 200ms, mentre il secondo di 100ms, più altri piccoli dettagli.

L'articolo [27] invece usa una tecnica di riconoscimento differente: oltre a un filtraggio passabanda usa anche un filtro non lineare con lo scopo di eliminare meglio i suoni cardiaci. Per la stima degli onset invece viene usata l'analisi dello zero-crossing in quanto le rispettive fasi di inspirazione ed espirazione avvengono ai passaggi per lo zero.

L'ultimo articolo che sfrutta il beat detection [41] dopo particolari filtri e downsampling usa come l'articolo precedente la tecnica di zero-crossing, ma la utilizza in modo differente: viene usata per determinare il ciclo di fase respiratoria, supponendo che in un ciclo di respirazione sono presenti un numero prefissato di passaggi per lo zero dell'onda. Questa tecnica viene usata quindi per determinare la finestra di analisi alla quale viene poi calcolata l'energia totale e confrontata con una soglia fissa determinata empiricamente; se la l'energia totale del segmento supera la soglia fissa il segmento viene classificato come respiro.



Figura 6.1: Schema riassuntivo delle tecniche usate dagli articoli nello stato dell'arte

Oltre alla tecnica di beat detection abbiamo analizzato un sistema di natura probabilistica che sfrutta le catene di Markov. L'articolo [33] modella il problema attraverso un modello nascosto di Markov nel quale la variabile nascosta può assumere tre possibili valori: inspirazione, espirazione e pausa. I risultati di questo algoritmo sono molto influenzabili dal rumore esterno e da rumori inaspettati emessi dal soggetto, in conclusione non risulta un algoritmo ottimale a meno che il suono prima di essere analizzato venga sottoposto a un miglior filtraggio.

Infine abbiamo l'articolo [35] che risolve il problema della localizzazione delle fasi respiratorie usando la dimensione frattale. Come i sistemi basati sul beat detection suddivide il suono in segmenti e sfrutta una finestra di lunghezza pari a mezzo respiro. Viene scelta questa tecnica in quanto si ipotizza che la varianza della dimensione frattale del suono respiratorio abbia un picco negli onset delle fasi respiratore. Un'altra differenza di questo sistema consiste nel fatto che l'analisi è esclusivamente nel dominio del tempo, a differenza di tutti gli algoritmi di beat detection.

Implementazioni [35] presenta un algoritmo risolutivo e i risultati di alcuni casi di test ma non fa nessun riferimento all'implementazione. [38] sostiene di aver sviluppato un programma MATLAB per riconoscere le fasi respiratorie usando i suoni respiratori e ne descrive il funzionamento e i risultati di alcuni test. Tuttavia non rende il software disponibile e non entra nel dettaglio della complessità e dei tempi di esecuzione. [33] e [27] descrivono un algoritmo e mostrano i risultati dei test di questo algoritmo quindi assumiamo che lo abbiano anche implementato anche se non ne parla affatto. [41] descrive in modo molto dettagliato un software MATLAB che riconosce le fasi respiratorie, fa anche una classificazione dei suoni respiratori ed è dotato di una interfaccia grafica. Non rende tuttavia tale software disponibile. [37] implementa un software in C++ per il riconoscimento delle fasi respiratorie dai suoni tracheali dotato di una interfaccia grafica però non lo rende disponibile.

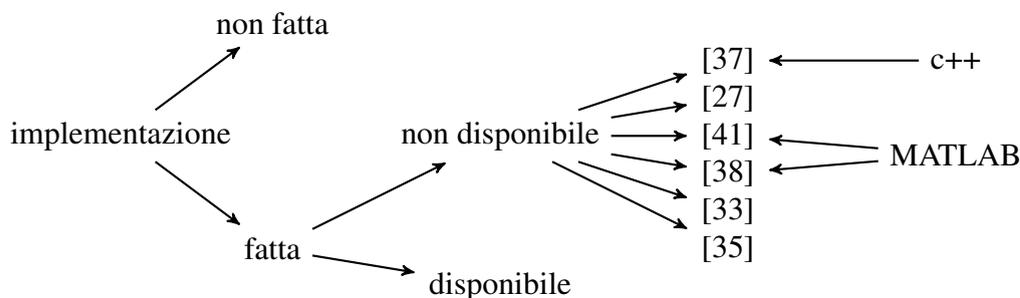


Figura 6.2: Schema riassuntivo delle implementazioni nello stato dell'arte

6.3 Conclusioni

Questi sistemi di analisi risultano tutti molto utili per l'analisi delle fasi respiratorie. Nessuno di questi però contiene un'analisi nel caso in cui il respiro sia assente. Solo [41] classifica i segmenti come respiro o non respiro, però la sua analisi non viene fatta online e cioè per determinare le soglie per la classificazione ha bisogno di analizzare l'intero file audio contenente il suono respiratorio. Un'importante problematica che non viene affrontata in questi articoli sono i rumori provenienti dal soggetto come per esempio la deglutizione ed eventuali suoni dovuti a una respirazione anormale (crackles, sibili, rantoli ecc..) che in alcuni casi possono rappresentare un disturbo durante l'analisi. In altre parole gli studi trovati nello stato dell'arte partono dall'ipotesi che i suoni di input siano presenti e siano normali, quindi non sono usabili direttamente in uno scenario reale nel quale il soggetto potrebbe smettere di respirare e potrebbe avere varie patologie (bronchite, pleurite, roncopia).

Capitolo 7

Analisi delle Metodologie

In generale un sistema che riconosce la respirazione a partire da alcuni segnali fisiologici continui, deve essere sensibile al cambiamento di alcune caratteristiche del segnale che sono omomorfe alla presenza, al volume, al flusso o alla frequenza della respirazione. Queste caratteristiche sono in generale dipendenti dal contesto quindi ci aspettiamo che un buon algoritmo faccia leva su delle quantità statistiche del segnale o su una qualche forma di apprendimento automatico. Ci aspettiamo anche che tali caratteristiche rispettino un qualche principio di località questo perché le proprietà della respirazione cambiano molto nel lungo termine.

La figura 7.1 mostra uno schema generale nel quale rientrano tutti i possibili sistemi software di riconoscimento della respirazione attraverso dei sensori.

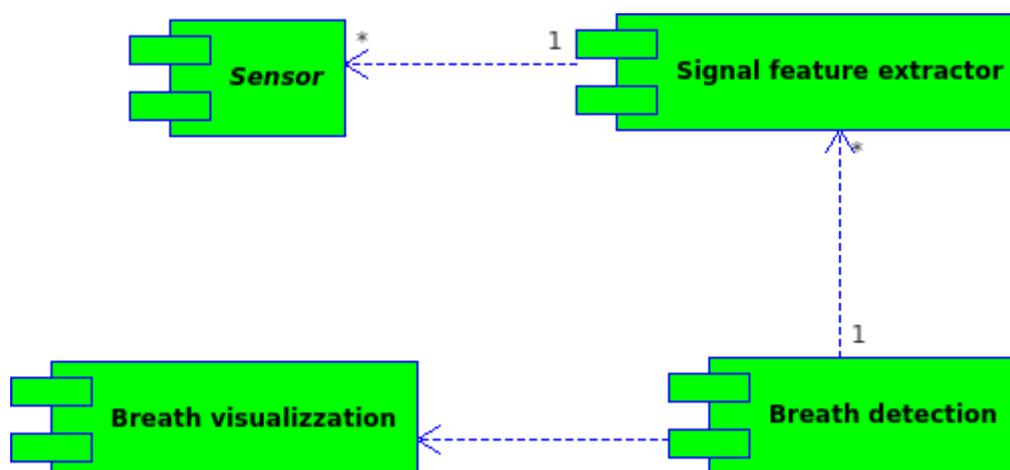


Figura 7.1: Schema generale di un software di riconoscimento della respirazione

7.1 Metodo elementare

Siamo tentati dal dire che per risolvere il problema è sufficiente l'algoritmo 3. Anche nelle ipotesi che non ci sia alcun rumore ambientale, non abbiamo la certezza che tale algoritmo funzioni per via di altri rumori fisiologici che potrebbero essere rilevati dallo stetoscopio ad esempio suoni cardiovascolari, suoni gastrointestinali e muscolari. L'algoritmo tuttavia è sottospecificato in quanto ci sono vari gradi di libertà: la scelta della soglia e la scelta della dimensione della finestra. Dai dati disponibili nella letteratura riguardo i suoni registrati da uno stetoscopio non siamo in grado di dire se possiamo scegliere i parametri dell'algoritmo in modo da farlo funzionare correttamente. Riteniamo quindi che almeno un primo livello di trattamento del segnale sia necessario per risolvere questo problema. Inoltre una implementazione così banale è probabilmente possibile realizzarla ad un livello più basso ad esempio direttamente nel firmware dello stetoscopio.

```
input : A block of sound samples
1 foreach window w in the input block do
2   instantSoundEnergy ← w.getSoundEnergy()
3   if instantSoundEnergy > threshold then
4     | is a breath;
5   else
6     | is not a breath;
```

Algorithm 3: Naive breath detection

7.2 Beat detection

La soluzione è nata dall'osservazione che il respiro ha un certo ritmo e quindi si può trattare il respiro come se fosse musica. La ricerca sul beat detection ha portato ad una serie di algoritmi e metodi per il trattamento di suoni musicali. Da una analisi di questi si capisce che possono essere applicati con opportune modifiche anche ai suoni prodotti dal respiro.

7.2.1 Scelta dell'algoritmo

Secondo [20] ci sono varie proprietà da considerare per scegliere un algoritmo di onset detection. Ad esempio:

- La complessità dell'algoritmo.

- Le caratteristiche della piattaforma sulla quale ci si aspetta che l'algoritmo venga usato.
- La presenza o meno di vincoli sul tempo di esecuzione dell'algoritmo.
- Il dominio dell'input:
 - Se il segnale ha dei beat molto marcati e presenta relativamente poche voci(ad esempio la musica tecno) allora è adeguato un metodo nel dominio del tempo.
 - Se il segnale da analizzare è complesso, ad esempio nel caso della musica sinfonica nella quale c'è una base di strumenti che fanno da accompagnamento e quindi dettano il ritmo della musica e altri gruppi di strumenti più legati alla melodia, allora conviene usare un metodo basato su informazione di fase nel dominio delle frequenze, in quanto in questo caso i beat sono legati molto al timbro degli strumenti.
- Se è necessaria una localizzazione precisa nel tempo e nelle frequenze allora si possono usare metodi basati sulle wavelet.
- Se la complessità computazionale non è un problema ed è presente un insieme adatto di segnali di allenamento allora si possono usare metodi basati su apprendimento automatico e informazioni statistiche(reti neurali, support vector machine, modelli nascosti di Markov).

Ricordiamo che gli algoritmi di onset detection sono progettati per funzionare su brani musicali. Questi possono essere un insieme molto complesso di voci musicali. Nel nostro caso le ipotesi sull'input sono più semplici perché possiamo assimilare il suono registrato dallo stetoscopio sul petto ad un brano musicale composto da due voci: i suoni respiratori e i suoni cardiovascolari. Anche nel caso dei suoni tracheali abbiamo due voci: i suoni respiratori e i suoni della deglutizione. Quindi concludiamo che serve un algoritmo che sfrutta sia una analisi nel dominio delle frequenze che una analisi nel dominio del tempo.

7.3 Pattern recognition e apprendimento automatico

Un'altra possibile soluzione attinge al campo del riconoscimento vocale. In particolare alcune tecniche di riconoscimento vocale usano dei classificatori che hanno come mattoni di base i fonemi. Il suono della respirazione in un certo senso può essere pensato come un linguaggio parlato nel quale ci sono solo due tipi di fonemi: l'inspirazione e l'espiazione. Sia nel linguaggio parlato che nella respirazione è anche importante il riconoscimento del silenzio.

7.3.1 Reti neurali

Una soluzione di questo tipo si può implementare attraverso reti neurali. In generale si può procedere nel modo seguente:

1. Prima di tutto bisogna disporre di un database di registrazioni di suoni respiratori.
2. I suoni possono attraversare una fase di filtraggio nella quale si rimuove il rumore, si riduce eventualmente la frequenza di campionamento e si cerca di eliminare i suoni cardiovascolari.
3. Questi suoni si segmentano in segmenti di due tipi:
 - Suoni che contengono respirazione, cioè suoni che contengono il suono prodotto dall'espiazione o dall'inspirazione
 - Suoni che contengono pause respiratorie. Per pause respiratorie in questo caso si intende una pausa in generale e quindi non necessariamente una apnea.
4. Per ogni segmento si calcola una sequenza di S proprietà statistiche interessanti ad esempio: la media dei valori assoluti dei coefficienti di Fourier in determinate bande di frequenza; la potenza media dei coefficienti wavelet in determinate bande di frequenza; la deviazione standard dei coefficienti in determinate bande di frequenza; il rapporto tra la media dei valori assoluti in bande di frequenza adiacenti.
5. Si calcola inoltre la quantità l che è la media della durata dei segmenti che contengono suoni respiratori.
6. A questo punto si usano i valori calcolati in precedenza per allenare una rete neurale. Una rete neurale è un meccanismo per approssimare una funzione quindi per allenare una rete neurale bisogna fornire ad essa delle coppie (*input*, *output*). In questo caso possiamo usare due neuroni di output e un numero di neuroni di input pari ad n , dove n è il numero di proprietà dei segmenti di suono calcolate in precedenza. Possiamo costruire un insieme di allenamento nel modo seguente. Se S è una sequenza relativa ad un segmento di respiro allora inseriamo nell'insieme $(S, (1, 0))$ mentre se S è una sequenza relativa ad un segmento di pausa inseriamo nell'insieme $(S, (0, 1))$.
7. Dopo che la rete è allenata possiamo usarla per classificare un segnale di input. Come prima cosa si divide il segnale in segmenti di dimensione pari ad l
8. Si da il segmento in pasto alla rete neurale; questa produce l'output: (s, p) .
9. Se s è maggiore di p allora il segmento è un suono; altrimenti è una pausa.



Figura 7.2: Forma d'onda di un segnale che contiene respiro.

7.4 Clustering

Guardando il grafico della forma d'onda del segnale della respirazione, in assenza di forte rumore, nella giusta scala temporale ed entro certi limiti di quantità temporale del segnale, siamo in grado di dividere il grafico in parti regolari e cicliche e di classificarle in: espirazione, inspirazione, pausa. Questo approccio metodologico risulta particolarmente chiaro osservando il grafico in figura 7.4.

Tale processo intuitivo si può inquadrare nel problema più generale del clustering. Possiamo procedere nel modo seguente:

1. Dividere l'input in blocchi di lunghezza ad esempio 20s o in ogni caso un valore abbastanza grande da essere sicuri che ci sia un ciclo respiratorio completo.
2. Trovare un clustering adatto ai dati di input.
3. Un respiro si può definire come un cluster di inspirazione seguito da un cluster di espirazione seguito da una pausa. Se il clustering contiene una sequenza di respiri allora il soggetto sta respirando altrimenti no.

Rimane il problema di scegliere l'algoritmo di clustering adatto alla nostra situazione. Nel nostro caso siamo di fronte ad un problema di clustering nel quale lo spazio del segnale di input è bidimensionale. Una dimensione è il tempo discreto e l'altra dimensione è il valore dei campioni. Gli algoritmi di clustering impongono di conoscere a priori il numero di cluster oppure un valore di soglia. Una possibilità è quella di usare l'algoritmo 2 per determinare il più probabile numero di cluster. In seguito confrontiamo questo valore con il numero di fasi che ci dovrebbero essere in media nel blocco di input analizzato. Questo dovrebbe andare bene almeno per il primo blocco. Per i blocchi successivi possiamo tenere traccia del numero di cluster e confrontare il valore corrente con i valori passati.

Parte III

Sistema software

Capitolo 8

Requisiti

8.1 Utenti del sistema software

Il software può rientrare in due categorie:

- La prima categoria si chiama *auscultazione assistita dal calcolatore*, alla quale afferiscono sistemi di supporto alle decisioni cliniche, progettati per aiutare il medico nella diagnosi attraverso i suoni del corpo. In questo caso gli utenti del sistema fanno parte del personale medico-sanitario di una struttura clinica.
- La seconda categoria è il monitoraggio mobile di segnali biomedici. In questo caso gli utenti del sistema possono essere soggetti privi di conoscenze mediche o infermieristiche.

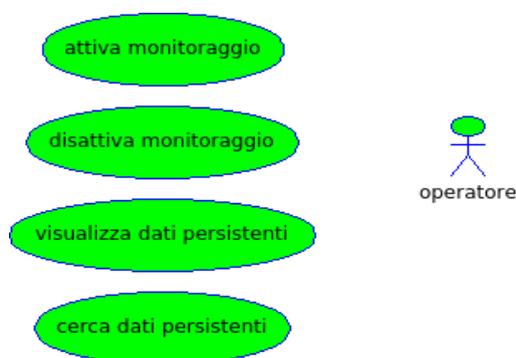


Figura 8.1: Diagramma dei casi d'uso



Figura 8.2: Diagramma dei casi d'uso opzionali

8.2 Requisiti funzionali

Il diagramma dei casi d'uso è illustrato in figura 8.1: I casi d'uso: attiva monitoraggio, visualizza dati persistenti e cerca dati persistenti sono descritti nelle tabelle 8.1, 8.2 e 8.3.

8.3 Requisiti funzionali opzionali

Il sistema può essere arricchito con i requisiti illustrati in figura 8.2. I casi d'uso: visualizza flusso e invia dati sono descritti nella tabelle 8.4 e 8.5.

8.4 Requisiti non funzionali

Sistemi real time

[47] e [48] danno le seguenti definizioni. Diciamo che un sistema è in *tempo reale* o *real time* se la correttezza dell'output dipende anche dal tempo impiegato per calcolarlo. I sistemi real time hanno delle scadenze entro le quali devono dare una risposta. Possiamo classificare i sistemi real time in base alle conseguenze subite dalla mancanza di una risposta entro la scadenza prevista:

hard Non rispettare una scadenza è un errore fatale.

firm In questo caso si possono tollerare frequenti mancanze nel rispetto delle scadenze, ma queste mancanze possono degradare la qualità del servizio.

soft Per i sistemi real time soft, l'obiettivo diventa quello di rispettare un certo sottoinsieme di scadenze con lo scopo di ottimizzare alcuni criteri che dipendono dall'applicazione.

Caratteristiche real time del sistema

In un sistema real time di elaborazione di segnali digitali, il segnale in input può essere virtualmente illimitato nel tempo. In realtà dei valori che massimizzano la durata del

| Caso d'uso | Attiva monitoraggio |
|-----------------------------------|---|
| Attori | C'è un solo attore attivo: un soggetto che intende monitorare il proprio respiro oppure un membro del personale medico sanitario che intende monitorare il respiro di un paziente. Il soggetto del quale si misura il respiro è un attore passivo in questo caso d'uso perché in generale esso può evitare di interagire in modo attivo col sistema stesso. |
| Precondizioni | <ul style="list-style-type: none"> • Lo stetoscopio elettronico è stato installato correttamente sul torace del soggetto o sulla trachea del soggetto. • I dispositivi di interfaccia tra lo stetoscopio e il sistema sono configurate correttamente. |
| Sequenza principale degli eventi | <ol style="list-style-type: none"> 1. L'attore attiva il sistema di monitoraggio attraverso l'interfaccia utente. In questo passo l'attore specifica se intende memorizzare o meno i dati e in che forma. L'attore può specificare la soglia di allarme per l'apnea. 2. Il sistema stabilisce una connessione con lo stetoscopio elettronico. Se questo passo fallisce allora comincia la sequenza alternativa degli eventi. 3. Il sistema passa in fase di monitoraggio. 4. Se la frequenza di respirazione scende al di sotto di una certa soglia critica, il sistema lo segnala attraverso una parte dedicata dell'interfaccia utente, ad esempio un segnale acustico. |
| Sequenza alternativa degli eventi | <ol style="list-style-type: none"> 1. Il sistema segnala un errore appropriato attraverso l'interfaccia utente. |
| Postcondizioni | Il sistema di monitoraggio è attivo. |

Tabella 8.1: Caso d'uso: attiva monitoraggio

| Caso d'uso | Visualizza dati persistenti |
|----------------------------------|---|
| Attori | C'è un solo attore ed è l'operatore del sistema. In questo caso d'uso un utente può consultare i dati relativi ai monitoraggio passati. |
| Precondizioni | L'utente conosce le chiavi di accesso ai dati: nome del soggetto e data. |
| Sequenza principale degli eventi | <ol style="list-style-type: none"> 1. L'attore inserisce nome del soggetto e/o data nella parte dell'interfaccia utente dedicata alla visualizzazione dei dati persistenti. 2. Il sistema visualizza i dati relativi alle chiavi inserite, se queste sono corrette. Altrimenti visualizza un messaggio di errore. |
| Postcondizioni | - |

Tabella 8.2: Caso d'uso: visualizza dati persistenti

| Caso d'uso | Cerca dati persistenti |
|----------------------------------|---|
| Attori | Utente del sistema. In questo caso d'uso un utente può cercare i dati relativi ai monitoraggio passati. Le chiavi di ricerca sono il nome del soggetto e la data. |
| Precondizioni | - |
| Sequenza principale degli eventi | <ol style="list-style-type: none"> 1. L'attore inserisce nome del soggetto e/o data nella parte dell'interfaccia utente dedicata alla ricerca dei dati persistenti. 2. Il sistema visualizza i risultati della ricerca. |
| Postcondizioni | - |

Tabella 8.3: Caso d'uso: cerca dati persistenti

| Caso d'uso | Visualizza flusso |
|----------------------------------|---|
| Attori | Utente del sistema. |
| Precondizioni | - |
| Sequenza principale degli eventi | In questo caso d'uso il sistema riconosce la localizzazione nel tempo degli eventi inspiratori ed espiratori e visualizza un grafico che ha il tempo sulle ascisse e il valore stimato del flusso sulle ordinate. Nello standard SI un flusso d'aria si misura in litri al secondo. Non è necessario stimare il valore oggettivo del flusso ma basta usare valori significativi in modo relativo al flusso massimo e al flusso nullo. |
| Postcondizioni | - |

Tabella 8.4: Caso d'uso: visualizza flusso

| Caso d'uso | Invia dati. |
|----------------------------------|---|
| Attori | Utente del sistema. |
| Precondizioni | L'utente conosce le chiavi di accesso ai dati e l'URL del personale medico sanitario. |
| Sequenza principale degli eventi | <ol style="list-style-type: none"> 1. L'utente inserisce le chiavi di accesso ai dati e l'URL del personale medico sanitario nella parte dell'interfaccia utente dedicata a questo caso d'uso. 2. Il sistema invia i dati e mostra il risultato dell'operazione (successo o fallimento con messaggio d'errore significativo). |
| Postcondizioni | - |

Tabella 8.5: Caso d'uso: invia dati

segnale si potrebbero trovare ma sono abbastanza grandi da costringerci ad usare una particolare definizione di scadenze temporali. Il ritardo nell'elaborazione deve essere limitato anche se il processo continua per un tempo illimitato. Quindi consideriamo la media del tempo di elaborazione del segnale per campione di segnale in un intervallo di tempo abbastanza piccolo rispetto ai vincoli real time, ad esempio un secondo. Questa media non deve essere maggiore del periodo di campionamento. Questo criterio vale sia che il segnale venga esaminato in blocchi sia che il segnale venga esaminato campione per campione[49]. In altre parole un sistema real time di elaborazione di un segnale virtualmente illimitato deve avere un tempo di esecuzione per secondo di segnale, minore di un secondo.

La velocità di esecuzione un algoritmo è una grandezza data dal rapporto tra la dimensione dell'input e il tempo di esecuzione su di esso. In questo caso la dimensione dell'input è la durata del segnale e non ci interessa la velocità calcolata su tutto il segnale di input ma quella calcolata ad intervalli regolari di dimensione piccola rispetto ai vincoli real time. Quindi si può definire la velocità v del processo come la quantità di campioni che ci sono in un secondo di segnale, fratto il tempo impiegato per l'elaborazione di un secondo di segnale. In un secondo di segnale ci sono un numero di campioni pari alla frequenza di campionamento del segnale $f_c Hz$. Il sistema può calcolare tale velocità ogni secondo e quindi produrre una sequenza di velocità v_1, v_2, \dots . Una condizione necessaria affinché il sistema si trovi sempre (ogni secondo) in uno stato valido è la seguente:

$$\forall n. v_n \geq f_c$$

Il sistema in fase di monitoraggio deve segnalare la presenza di apnee troppo lunghe. In particolare consideriamo troppo lunga una apnea di 30 secondi. Partiamo dall'assunto che le sole scadenze siano quelle relative all'evento apnea troppo lunga e che non rispettare una scadenza significa non dare l'allarme in tempo prima che il soggetto rischi gravi problemi cardiorespiratori. Allora ci troviamo in presenza di un sistema real time hard. Tuttavia i vincoli di tempo per le scadenze sono blandi relativamente ai tempi di esecuzione che si possono prevedere su un calcolatore moderno.

8.5 Requisiti non funzionali opzionali

Il software deve funzionare bene anche in presenza di rumore.

8.6 Scelta della soglia di allarme

La soglia oltre la quale una apnea è considerata pericolosa deve essere configurabile. Il valore soglia deve essere stabilito da personale medico qualificato. Si può intuire che una soglia troppo bassa potrebbe degradare la qualità del sonno del soggetto in un modo patogenico o quantomeno in un modo tale da rendere inutile il monitoraggio. Al contrario una soglia troppo alta espone il soggetto ad un rischio troppo elevato. Ci si può aspettare che la soglia di allarme non sia oggettiva ma debba essere personalizzata e che vari con l'età e alcuni parametri fisiologici del soggetto.

Capitolo 9

Architettura

9.1 Componenti

Il sistema di monitoraggio ha le componenti illustrate nella figura 9.1. Discutiamo di ognuna di esse più in dettaglio nel seguito del capitolo

9.1.1 Meccanismi di persistenza

Il sistema implementa dei meccanismi di persistenza. Un'interfaccia minimale di una componente che implementa un meccanismo di persistenza deve offrire le operazioni seguenti:

- Aggiungere i dati di nuovo soggetto.
- Aggiungere una entry di dati di un monitoraggio relativa ad un certo soggetto.
- Cercare una entry di dati di un monitoraggio a partire dai dati del soggetto e/o dalla data del monitoraggio.
- Reperire una entry di dati di un monitoraggio a partire da una chiave cercata in precedenza.

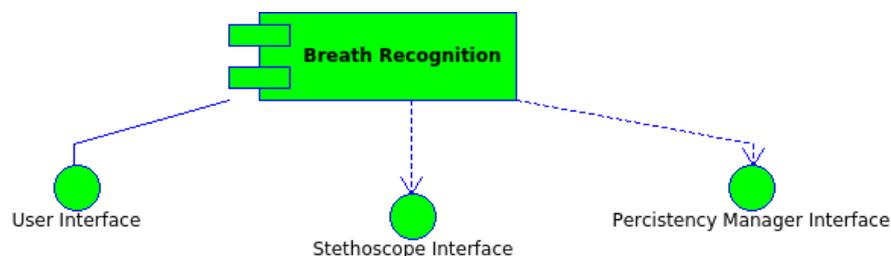


Figura 9.1: Diagramma delle componenti del sistema

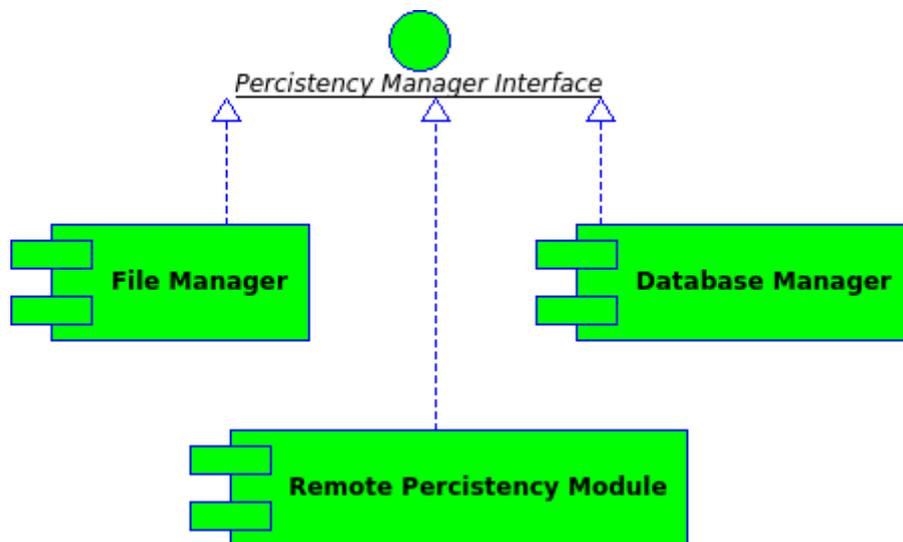


Figura 9.2: Possibili componenti che implementano l'interfaccia con un meccanismo di persistenza

La figura 9.2 elenca alcuni possibili modi di implementare questo meccanismo di persistenza.

I dati persistenti sono: il nome del soggetto, la data, l'indice di apnea-ipopnea e una sequenza di etichette nell'insieme: respiro o pausa respiratoria. Ogni etichetta corrisponde a 3s di segnale. Il sistema permette di salvare i dati di un monitoraggio sulla memoria di massa del computer sotto forma di file. Vengono usati i meccanismi di serializzazione offerti da Java. La serializzazione di Java permette di memorizzare e leggere uno stream di oggetti Java[56]. Per ottimizzare questa operazione, i dati vengono memorizzati prima in un buffer, quando il buffer si riempie vengono scritti sulla memoria di massa. Anche se questa ottimizzazione può sembrare inutile in quanto tutti i file system moderni sfruttano un meccanismo implicito di buffering, la dimensione di quest'ultimo potrebbe avere un valore predefinito troppo piccolo.

9.1.2 Interfaccia con lo stetoscopio

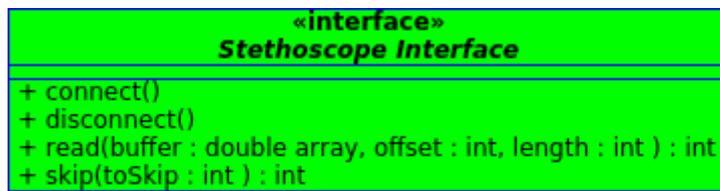


Figura 9.3: Operazioni dell'interfaccia con lo stetoscopio

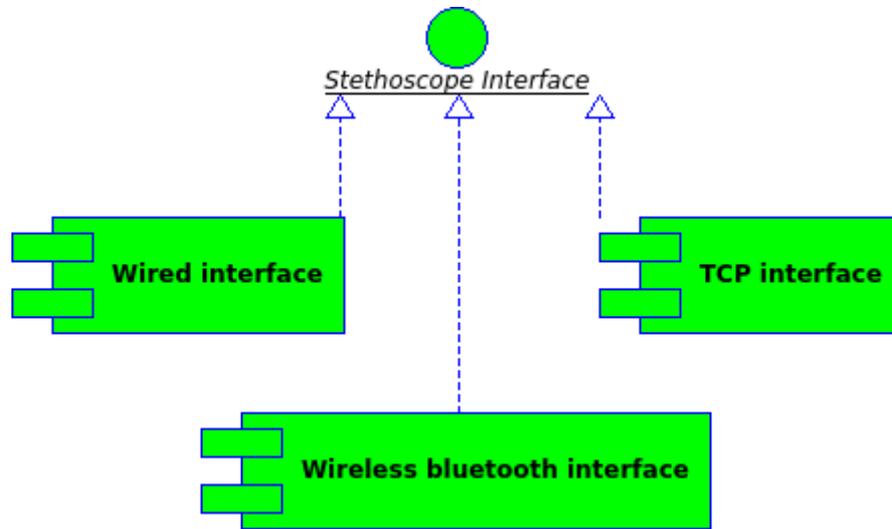


Figura 9.4: Possibili componenti che implementano l'interfaccia con lo stetoscopio

In generale i sistemi di monitoraggio continuo di parametri fisiologici hanno bisogno di interfacciarsi con i sensori di interesse. Questo sistema non è una eccezione. Riteniamo che una interfaccia minima con lo stetoscopio elettronico sia quella illustrata nella figura 9.3. Le operazioni illustrate sono:

connect Inizializza la connessione con il sensore ed allocare eventuali risorse necessarie.

disconnect Terminare la connessione con il sensore e deallocare eventuali risorse.

read Legge *length* campioni di input e li memorizza nel buffer a partire dall'offset specificato.

skip Tralascia un certo numero di campioni. Può essere utile per recuperare in parte un eventuale ritardo globale.

Il sistema deve includere almeno una componente che implementa l'interfaccia tra il dispositivo di monitoraggio e il sensore. Alcune opzioni sono illustrate in figura 9.4.

Ci sono appunto vari modi di implementare una connessione tra lo stetoscopio e il dispositivo di monitoraggio. Una scelta che dobbiamo fare riguarda il mezzo di trasmissione del segnale:

via cavo In questa modalità spesso è necessaria la presenza di personale medico specializzato. Tra gli svantaggi di questa impostazione notiamo: ridotta mobilità del sistema, probabile necessità di alimentazione elettrica dalla rete, rischio di gestione inadeguata di contatti nei cavi.

senza cavo Una connessione senza fili presenta alcuni vantaggi ad esempio permette al paziente di muoversi più liberamente e i costi di installazione si possono ridurre.

Riteniamo che per questo sistema sia più adeguata una connessione senza fili.

Un'altra scelta riguarda il protocollo di comunicazione, il quale però potrebbe dipendere anche dalla scelta del mezzo di trasmissione.

Siamo in presenza di un problema la cui modellazione porta naturalmente a pensare ad un pattern architetturale di tipo client server. Un server è un dispositivo fisico o virtuale che possiede una risorsa da condividere[55], in questo caso il server è lo stetoscopio e la risorsa da condividere è il suono che esso registra. Un cliente è un dispositivo fisico o virtuale che richiede una certa risorsa ad un server[55], in questo caso il client è il dispositivo di analisi del suono.

bluetooth In base alle considerazioni fatte da [53], la tecnologia wireless bluetooth si adatta bene al nostro sistema. Il bluetooth permette di stabilire semplici connessioni ad hoc tra dispositivi che hanno a disposizione poca energia elettrica e che sono posti a piccola distanza tra di loro, dove per piccola distanza approssimativamente intendiamo che i dispositivi si trovano nella stessa stanza o anche entro certi limiti in una stessa struttura ospedaliera. I valori precisi di consumi, distanze e velocità di trasmissione variano da dispositivo a dispositivo. Lo stetoscopio elettronico invia continuamente dati al sistema di riconoscimento attraverso il canale bluetooth. I dati inviati dipendono dal particolare stetoscopio ma possiamo aspettarci che questi siano sotto forma di pacchetti di un segnale audio digitale. Sugeriamo quindi di implementare il sistema nel modo illustrato nella figura 9.5

velocità dell'interfaccia Supponiamo che il sistema abbia una velocità media di esecuzione di v_s campioni di segnale al secondo, e supponiamo che v sia maggiore della frequenza di campionamento e che quindi il sistema riesca ad analizzare un segnale di un secondo in un tempo minore di un secondo. Questa velocità si intende calcolata senza contare il tempo di trasmissione dallo stetoscopio al sistema.

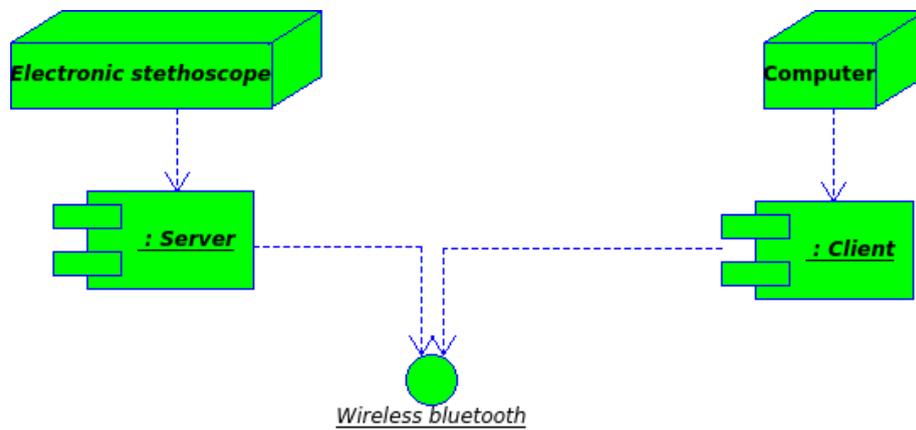


Figura 9.5: Architettura del sistema

Supponiamo che la velocità di trasmissione dell'interfaccia tra sistema e stetoscopio sia v_i bit al secondo e siano f_c la frequenza di campionamento del segnale e $size$ la dimensione in bit di un campione di segnale. Allora deve valere

$$\frac{f_c}{v_s} + \frac{f_c}{\left\lfloor \frac{v_i}{size} \right\rfloor} \leq 1s$$

affinché il sistema non accumuli ritardo.

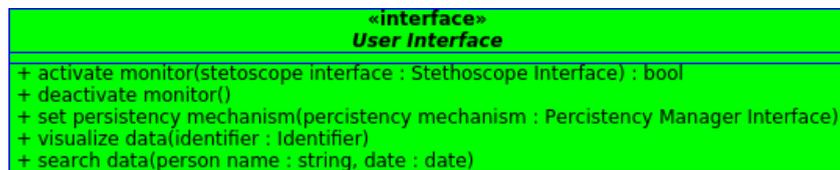


Figura 9.6: Operazioni dell'interfaccia utente

9.1.3 Interfaccia utente

Una interfaccia utente per questo sistema dovrebbe fornire almeno le operazioni elencate nella figura 9.6.

Sono state implementate due interfacce minimali:

console Un interprete di comandi da console che ha questo aspetto:

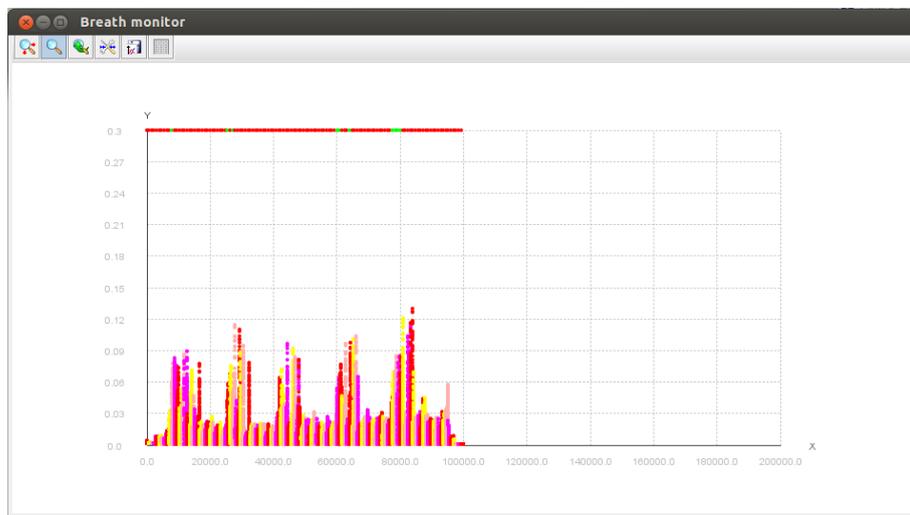
```

Breath Monitor Beta version
by Federico Viscomi
type help for a command list

$ help

exit                exit application
help                print this help
list                list working directory content
monitor -f filename start breath recognition on given file name
stop                stop current monitoring if any
$
  
```

GUI Una interfaccia grafica basata su Swing [78] che offre le stesse funzioni di quella grafica e in più usa la libreria open source JMathPlot [79] per disegnare il grafico nel dominio del tempo del segnale. Questo grafico è utile solo nella versione iniziale del sistema per motivi di debug. Mentre in una versione successiva del sistema si può rimpiazzare questo grafico con quello del flusso d'aria. La finestra principale è illustrata in figura 9.1.3.



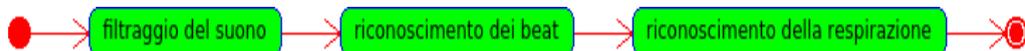
Capitolo 10

Implementazione

Tecnologie usate Il sistema è stato implementato in Java usando Eclipse come ambiente di sviluppo.

10.1 Algoritmo di riconoscimento della respirazione

Le fasi di cui si compone l'algoritmo di riconoscimento della respirazione ad un alto livello di astrazione sono illustrate nel diagramma seguente:



Filtraggio del segnale audio Il segnale attraversa la successione di fasi a cascata rappresentate nella figura 10.1. Ogni filtro è implementato in modo simile a quanto specificato dall'interfaccia *InputStream* di Java. Siamo davanti ad un tipico caso di design di tipo *pipeline* in quanto l'output di un filtro è l'input del filtro successivo (eccetto che per l'ultimo filtro). Il segnale audio, anche nel caso in cui venga letto da un file, è trattato come uno *stream* di dati. Più in dettaglio le fasi di filtraggio sono le seguenti:

Buffering/Windowing Questa fase è necessaria in quanto alcuni dei filtri successivi lavorano su blocchi di input e non sul singolo campione. Inoltre la presenza del buffer può diminuire il tempo totale di elaborazione. In questo caso l'input è letto da un file quindi non ci sono problemi di overflow.

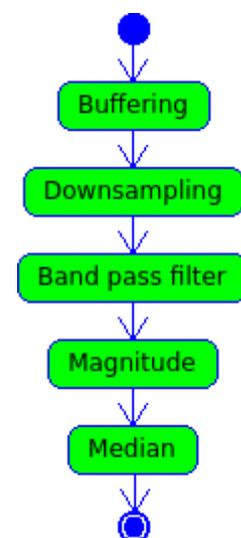


Figura 10.1: Diagramma di attività della fase di filtraggio

Una condizione sufficiente affinché il software rispetti i propri requisiti real time è che la velocità di elaborazione sia sempre maggiore di: un secondo di segnale fratto un secondo di tempo di elaborazione.

Downsampling La sequenza di campionamento viene ridotta con lo scopo di aumentare l'efficienza delle fasi successive dell'algoritmo. Gli spettri di potenza dei suoni respiratori e dei suoni cardiaci hanno frequenze al di sotto dei 500Hz . Quindi si può abbassare la frequenza di campionamento a 1000Hz in quanto una larghezza di banda di 500Hz è adeguata a catturare i suoni respiratori[27].

Bandpassfiltering Questo filtro lascia passare solo i suoni che si trovano nella banda di frequenza dai 100 ai 1500Hz , il risultato è un suono nel quale sono più facilmente distinguibili i suoni normali della respirazione. Inoltre questo filtro elimina anche alcuni suoni respiratori anormali e parte dei suoni cardiocircolatori. Questo filtro è implementato grazie alle librerie JSTK reperite all'indirizzo [54]. JSTK sta per Java speech toolkit e fornisce tra le altre cose una libreria di tecniche usate per il riconoscimento vocale. La libreria è rilasciata secondo la licenza GPLv3.

Magnitudefiltering Questo filtro semplicemente prende il valore assoluto del segnale.

Medianfiltering Questo è un classico filtro a mediana con finestra rettangolare di dimensione 10ms e serve per smorzare i suoni accidentali che hanno una intensità relativamente alta rispetto al suono respiratorio e una durata relativamente bassa rispetto alla durata delle fasi respiratorie.

10.2 Algoritmo di beat detection

L'algoritmo di beat detection scelto è un basato su caratteristiche esplicite del segnale, prese sia dal dominio del tempo che dal dominio delle frequenze. Il thresholding è adattativo. Scegliamo la dimensione di un blocco in modo tale che contenga $4s$ di campioni dell'input. Mentre scegliamo una dimensione della finestra pari a 10ms . Sia $w[1\dots n]$ una finestra all'interno di un blocco di campioni di suono filtrato, definiamo l'energia del suono della finestra w come:

$$\sum_{i=1}^n w[i]^2$$

definiamo invece l'energia media locale come

$$\sum_{i=1}^m sb[i]^2$$

sb è un buffer che memorizza gli ultimi valori della media dell'energia locale. Il buffer ha la dimensione adatta ad ospitare i valori relativi ad un blocco. Un modo semplice per il beat detection è la ricerca di picchi nell'energia del suono. In questo modello ci proponiamo di riconoscere le variazioni di energia attraverso il calcolo dell'energia istantanea in una finestra di segnale delle dimensioni di 10ms e il confronto di questa con la media dell'energia di un blocco di al più 4s del segnale. Non calcoliamo la media su tutto l'input disponibile perché ci possono essere dei cambiamenti notevoli nel suono della respirazione su un periodo di tempo lungo. Fatte queste premesse possiamo dire che riconosciamo un beat solo quando l'energia istantanea è superiore all'energia media.

```
input : A block of filtered sound samples
output: A sequence of boolean

1 soundEnergyHistoryBuffer.clear()
2 foreach window w in the input block do
3   instantSoundEnergy ← w.getInstantSoundEnergy()
4   averageLocalEnergy ←
   soundEnergyHistoryBuffer.getAverageLocalEnergy()
5   soundEnergyHistoryBuffer.write(instantSoundEnergy)
6   if instantSoundEnergy > averageLocalEnergy then
7     | beatSequence.add(true)
8   else
9     | beatSequence.add(false)
10  return beatSequence
```

Algorithm 4: beat detection

10.3 Riconoscimento delle apnee a rischio

La fase di riconoscimento del respiro prende in input le sequenze di booleani date in output dalla fase precedente. Ricordiamo che un elemento della sequenza ha valore di verità vero se corrisponde ad un beat altrimenti vale falso. Questa fase si divide nelle due fasi seguenti.

Clustering La prima fase del riconoscimento è un *clustering* dei dati della fase precedente. In questo contesto un cluster è una sequenza di booleani di lunghezza non nota a

priori. Un cluster non deve avere una lunghezza superiore ad una certa soglia ad esempio la lunghezza che corrisponde ad un segnale di 1s.

Usiamo il seguente algoritmo di clustering sequenziale e con cluster sequenziali di lunghezza non nota a priori:

```

input : A sequence of boolean  $x_1, \dots, x_n$ 
output: A clustering of the data  $C_1, \dots, C_m$ 

1  $m \leftarrow 1$ 
2  $C_m.data \leftarrow \{x_1\}$ 
3  $C_m.type \leftarrow x_1$  for  $i \leftarrow 2$  to  $n$  do
4   if  $(distance(x_i, C_m) > threshold) \vee (C_m > maxSize) \vee i == n$  then
5     increments  $m$ 
6      $C_m.data \leftarrow \{x_i\}$ 
7      $C_m.type \leftarrow x_i$ 
8   else
9      $C_m.data \leftarrow C_m.data \cup \{x_i\}$ 

```

Algorithm 5: Beat clustering

La distanza tra un elemento x_i e un cluster C_m è

$$distance(x_i, C_m) = \begin{cases} \frac{beat\ count}{cluster\ size(C_m)} & se\ x_i = false \\ 1 - \frac{beat\ count}{cluster\ size(C_m)} & se\ x_i = true \end{cases}$$

La soglia è di 0.7 ed è stata scelta in modo empirico.

La fase di cluster serve per correggere eventuali falsi beat della fase precedente, ad esempio un beat che non era causato dal respiro e che non è stato eliminato dalla fase di filtraggio oppure serve per aggiungere beat mancanti ad esempio nel caso di una piccola apnea localizzata immediatamente dopo la fase inspiratoria e prima della fase espiratoria.

Riconoscimento delle apnee a rischio L'algoritmo considera come pausa respiratoria una sequenza non vuota di cluster C_1, \dots, C_n tale che $C_i.type = false$ per ogni i . Appena la durata totale della pausa respiratoria supera una soglia critica, il sistema suona una allarme e si mette in attesa che l'allarme venga spenta attraverso un comando apposito dell'interfaccia utente.

10.4 Complessità computazionale dell'algoritmo

Calcolare in modo preciso la complessità asintotica dell'algoritmo è difficile perchè è difficile determinare la complessità del filtro passa banda della libreria JSTK, tuttavia escludendo questa libreria, il resto dell'algoritmo è lineare nel numero dei campioni. La complessità è relativa alla macchina virtuale Java.

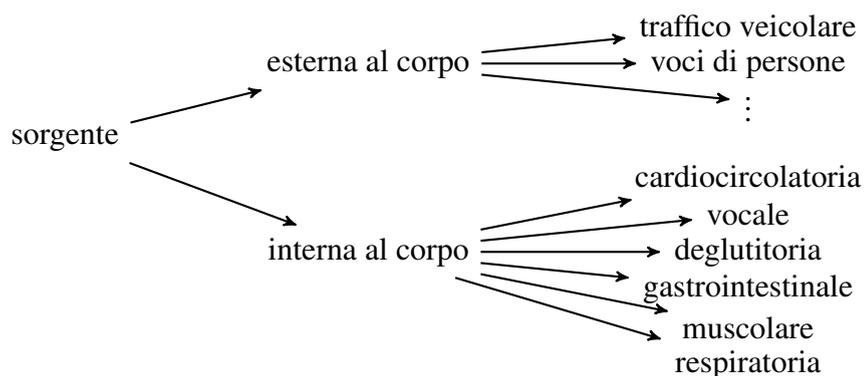
Capitolo 11

Test

11.1 Scelta dei casi di test

Per la scelta dei casi di test usiamo un approccio di tipo black-box e quindi esaminiamo da prima lo spazio dell'input e poi i possibili scenari di uso. Il tipo dell'input è l'insieme di tutti i possibili segnali audio di durata arbitraria. Lo spazio dell'input è un sottoinsieme del tipo dell'input nel quale rientrano tutti i segnali audio che possono essere ascoltati da uno stetoscopio elettronico posizionato sul petto di un soggetto.

È possibile individuare alcune classi di suoni nello spazio dell'input in base alle sorgenti:



Alcuni casi di test ci vengono dati dai possibili valori che può avere l'input in uno scenario di uso reale del software. Ad esempio alcuni casi di test possono avere come input:

- Un file audio abbastanza lungo da simulare un monitoraggio del sonno reale. Lo scopo di un caso d'uso con questo input è la valutazione della velocità a lungo termine dell'algoritmo.

- Dei suoni respiratori sovrapposti a rumore di vari tipo ed intensità. Lo scopo di un caso d'uso con questo input è la valutazione della tolleranza al rumore.
- Suoni respiratori senza rumore. Lo scopo di un caso d'uso con questo input è la valutazione del funzionamento del software in uno scenario ideale.
- Un file audio contenente solo rumore. Questo caso di test serve per capire se il software può rilevare la presenza di respiro in suoni che non contengono alcun respiro. In uno scenario di uso reale corretto questo caso non si verifica ma è comunque interessante.
- Un file con una frequenza di campionamento molto elevata. Questo caso di test rientra nella categoria stress test. Ci aspettiamo che il sistema si comporti bene se ha un input file con una frequenza di campionamento molto elevata grazie al filtro di sottocampionamento.
- Un file contenente suoni respiratori sovrapposti a forti suoni cardiaci.
- Un file respiratorio contenente una apnea più lunga della soglia massima.

Per creare dei casi che valutano la resistenza al rumore si può procedere nel modo illustrato nella figura 11.1 e cioè:

1. Scegliere una file contenente una sorgente di rumore e scegliere una intensità della sorgente di rumore.
2. Filtrare il file di rumore in base ad un certo modello acustico del corpo. Cioè cercare di prevedere cosa lo stetoscopio sente se è presente la sorgente di rumore scelta. Questo modello acustico è necessariamente un modello approssimato. In una prima fase elementare di modellazione possiamo usare un semplice filtro attenuatore e supporre che il rumore sia di tipo additivo.
3. Scegliere un file contenente un respiro.
4. Fare un mix dei file.

Arriva in nostro aiuto il software open source Audacity [80] il quale offre una vasta gamma di funzioni di trattamento dell'audio digitale. Quelle utili per i casi di test sono:

- Aggiungere rumore di vari tipi (bianco, rosa e marrone) e con varia intensità.

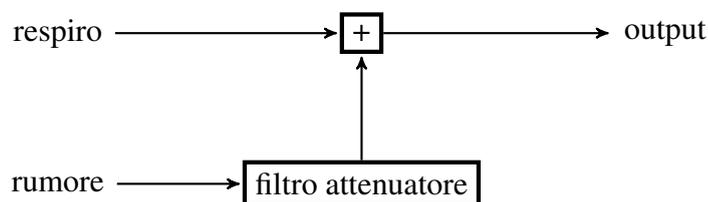


Figura 11.1: Processo di aggiunta del rumore

- Aggiungere segmenti di silenzio di lunghezza a scelta nei file audio e quindi simulare la presenza di una apnea lunga.
- Concatenare file audio. Questa funzione è utile perché i file audio che abbiamo a disposizione reperiti da [14] sono di lunghezza troppo breve (minore di 15s).
- Applicare filtri di vario tipo tra i quali un filtro attenuatore.

In generale possiamo creare un caso di test attraverso la concatenazione di segmenti di file audio. Ogni segmento contiene una pausa respiratoria sovrapposta ad un rumore oppure un respiro sovrapposto ad un rumore. La tabella seguente riassume le proprietà di un segmento di base:

| contenuto | tipo o sorgente | intensità |
|--------------------|--|-----------|
| respiro | [normale anormale misto, bronchiale vescicolare, continuo discontinuo, ronchi wheeze stridor crackles squawks] | [0-10] |
| rumori | [bianco rosa marrone, interno esterno, gastrointestinale deglutitorio vocale, ecc..] | [0-10] |
| pausa respiratoria | - | - |

Tabella 11.1: Proprietà di un segmento di base.

Non abbiamo a disposizione alcuno stetoscopio però usiamo le registrazioni reperite da [14] e partiamo da queste per creare alcuni casi di test. Purtroppo le fonti non riportano i dettagli di come sono stati registrati i suoni.

11.2 Valutazione dell'output

Valutare la localizzazione delle fasi respiratorie. Se si vuole valutare un algoritmo che localizza nel tempo le fasi respiratorie allora pensiamo ad un suono respiratorio come

ad una sequenza di cicli respiratori e ad un ciclo respiratorio come una fase di inspirazione seguita da una fase di espirazione seguita da una pausa.

Quindi lo spazio di input è classificabile in sequenze crescenti di numeri razionali che indicano la posizione temporale dell'inizio di ciascuna fase. L'output dell'algoritmo sarà una sequenza di numeri razionali che indicano la localizzazione temporale dell'inizio di ciascuna fase. Definiamo la differenza tra l'output e il descrittore della classe di cui fa parte l'input come la sequenza dei valori assoluti delle differenze delle singole componenti. La qualità dell'algoritmo si può misurare in termini di qualche proprietà statistica di questa differenza, ad esempio la media.

Valutare la localizzazione delle apnee. Se si vuole valutare un algoritmo che riconosce la presenza di una apnea allora possiamo classificare lo spazio dell'input in sequenze di coppie di numeri razionali tali che ogni coppia contiene il tempo di inizio e il tempo di fine di una apnea. In tal caso anche l'output sarà una sequenza di coppie di numeri razionali. Supponiamo che in input ci sia una apnea che comincia al tempo s_{in} e finisce al tempo f_{in} . Si possono verificare vari casi:

- Esiste un solo intervallo temporale di output (s_{out}, f_{out}) che interseca l'intervallo (s_{in}, f_{in}) . Allora definiamo l'errore di riconoscimento dell'evento (s_{in}, f_{in}) come

$$|s_{out} - s_{in}| + |f_{out} - f_{in}|$$

e diciamo che l'evento (s_{in}, f_{in}) è un *vero positivo* cioè un evento riconosciuto in modo corretto.

- Se non ci sono intervalli temporali di output che intersecano (s_{in}, f_{in}) allora diciamo che questo evento è un *falso positivo*. I falsi positivi sono gli errori più gravi e che quindi incidono maggiormente nella valutazione di un algoritmo.
- Se ci sono più intervalli temporali di output che intersecano (s_{in}, f_{in}) allora la scelta di come valutare la qualità dell'algoritmo è arbitraria e può essere ad esempio

$$|\min(s_{out}) - s_{in}| + |\max(f_{out}) - f_{in}|$$

Rimane il caso in cui l'algoritmo produce un output (s_{out}, f_{out}) ma non abbiamo nessun intervallo di input che lo interseca. In tal caso parliamo di *falso negativo* ed è un errore meno grave di un falso positivo.

Valutare la localizzazione delle apnee a rischio. Se si vuole valutare un algoritmo che riconosce la presenza di una apnea troppo lunga allora possiamo classificare lo spazio

dell'input come nel caso precedente ma l'analisi che ne risulta è diversa. Supponiamo che in input ci sia una apnea troppo lunga (cioè maggiore di una certa soglia S di sicurezza) che comincia al tempo s_{in} e finisce al tempo f_{in} . Si possono verificare vari casi:

- C'è almeno un intervallo temporale di output (s_{out}, f_{out}) tale che
 - (s_{out}, f_{out}) interseca (s_{in}, f_{in})
 - $s_{out} + S \leq s_{in} + S + T$ dove T è una soglia di tolleranza dell'errore.
 - $f_{out} - s_{out} > S$

In tal caso diciamo che l'evento (s_{in}, f_{in}) è un *vero positivo* cioè un evento riconosciuto in modo corretto.

- Se non ci sono intervalli temporali di output con le proprietà elencate al punto precedente allora diciamo che l'evento (s_{in}, f_{in}) è un *falso positivo*.

Rimane il caso in cui l'algoritmo produce in output un intervallo (s_{out}, f_{out}) di lunghezza maggiore della soglia di sicurezza ma non abbiamo nessun intervallo di input che lo interseca e che ha durata maggiore della soglia di sicurezza. In tal caso abbiamo un *falso negativo*. Si può anche definire cosa intendiamo per *vero negativo* e cioè una mancanza di apnea troppo lunga che l'algoritmo non classifica come apnea troppo lunga. Tuttavia questa definizione non si applica in modo elegante alle premesse fatte in questo paragrafo. La tabella seguente ricapitola i vari casi:

| | | |
|-----------------------------------|-----------------------|-----------------------|
| <i>evento: apnea troppo lunga</i> | evento presente | evento assente |
| evento riconosciuto | vero positivo | falso negativo |
| evento non riconosciuto | falso positivo | vero negativo |

Si può capire facilmente quale significato abbia la classificazione precedente, se si immagina uno scenario di uso del software.

Vero positivo. Il soggetto ha una apnea nel sonno troppo lunga e il sistema suona l'allarme. Il soggetto si sveglia, spegne l'allarme e torna a dormire sano e salvo (si spera).

Vero negativo. Il soggetto non ha una apnea nel sonno troppo lunga e il sistema non suona l'allarme. Questo caso è auspicabile. Maggiore è la frequenza di questi casi, maggiore è la qualità del sonno del soggetto.

Falso positivo. Il soggetto ha una apnea nel sonno troppo lunga e il sistema non suona l'allarme. Questo caso è rischioso per la salute del paziente.

Falso negativo. Il soggetto non ha una apnea nel sonno troppo lunga e il sistema suona l'allarme. Il soggetto si sveglia, spegne l'allarme e torna a dormire. Non ha modo di capire se si è verificato un vero positivo o un falso negativo (quindi non impreca contro

gli sviluppatori del software). I falsi negativi degradano la qualità del sonno del soggetto ma non sono un rischio grave per la salute quanto i falsi positivi. Tuttavia se il degrado nella qualità del sonno è eccessivo potrebbe causare danni psicofisici al soggetto che eventualmente superano quelli causati dalla sindrome da apnea del sonno. Quindi è anche importante che il sistema non generi molti falsi negativi.

11.3 Casi di test e risultati

I test fatti sono di tipo *oracolo* nel senso che l'output dell'algoritmo in ogni caso di test viene confrontato con il risultato che l'algoritmo dovrebbe fornire. I test sono stati fatti su un laptop *Hp Pavilion g6* con le seguenti caratteristiche:

| | |
|-----------------------|--|
| Microprocessore | Intel Core <i>i3</i> – 2330M da 2,2GHz |
| Cache microprocessore | 3MB di cache L3 |
| Memoria | DDR3 da 6GB |

Tabella 11.2: Caratteristiche del calcolatore usato per i test.

11.3.1 Casi di test sui file delle repository

I file usati in questo caso di test sono descritti in una sezione successiva 11.4. In seguito all'ascolto e alla visualizzazione della forma d'onda dei file, si nota che nessuno dei file contiene una apnea troppo lunga. La tabella 11.3 riporta nell'ordine: il nome del file, il tempo di esecuzione totale dell'algoritmo e un errore approssimato nella localizzazione delle apnee. Il tempo di esecuzione dell'algoritmo è espresso come media dei tempi di esecuzione per secondo di segnale.

| File (.wav) | Tempo | Errore |
|---------------------|-------|---|
| Coarse crackles | 2ms | 0.4s nei veri positivi più un falso negativo |
| Normal vesicular | 14ms | 0.2s solo veri positivi |
| Pleural friction | 3ms | Il suono è troppo rumoroso e il sistema riconosce solo due pause respiratorie in modo corretto su quattro |
| Inspiratory stridor | 3ms | Il sistema riconosce l'inspirazione con una margine di errore di 0.4s ma confonde l'espiazione con una pausa perché questa ha una intensità molto bassa |

Tabella 11.3: Esito dei test sui file delle repository

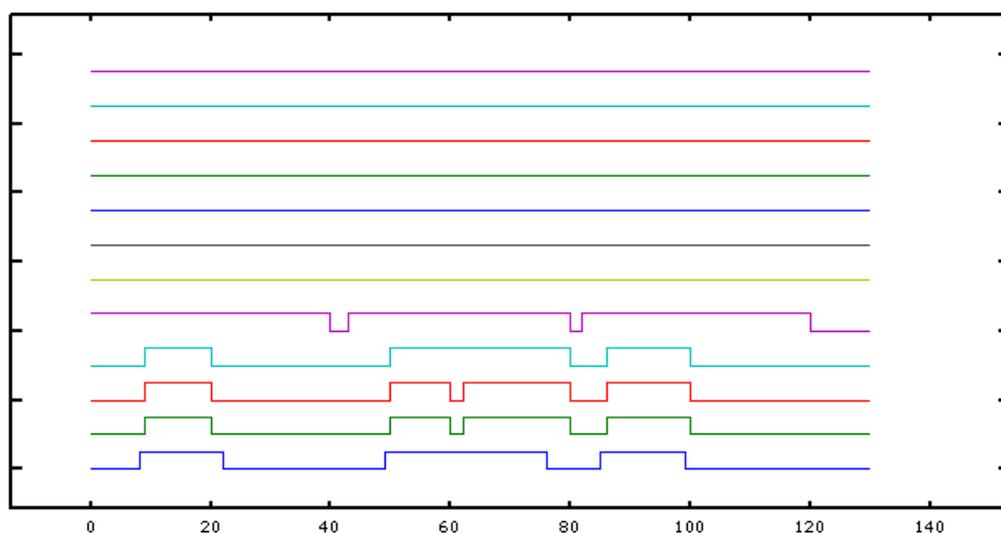


Figura 11.2: Risultati del test di tolleranza al rumore bianco.

11.3.2 Caso di test di tolleranza al rumore bianco

Il protagonista di questo caso di test è il file *Normal vesicular.wav* che contiene un suono respiratorio normale disturbato da un rumore leggero. A questo file aggiungiamo con Audacity del rumore bianco di intensità crescente e valutiamo le prestazioni del sistema. Il file ha le seguenti caratteristiche:

- L'intensità massima è circa $0.2dB$.
- L'intensità media delle fasi inspiratorie è circa $0.08dB$.
- L'intensità media delle fasi di pause respiratorie è circa $0.02dB$.

L'intensità del rumore aggiunto va da $0dB$ a $0.2dB$ con un incremento di $0.02dB$, quindi eseguiamo 11 test. In nessun caso era presente una apnea troppo lunga e in nessun caso l'algoritmo ha rilevato la presenza di una apnea troppo lunga quindi dal punto di vista del riconoscimento di apnee troppo lunghe, l'algoritmo funziona in modo corretto. È comunque interessante valutare l'output dell'algoritmo con un maggior livello di dettaglio. La figura 11.2 illustra una rappresentazione dell'output dell'algoritmo sui file di test e contiene un grafico per ogni file di input.

I valori sulle ascisse segnano il tempo in decimi di secondo. I grafici contenuti nella figura dal basso verso l'alto escluso il primo sono relativi a file che hanno una quantità di rumore crescente e mostrano quali parti dei rispettivi file vengono riconosciuti come respiro e quali parti vengono riconosciuti come apnea. Invece il primo grafico in basso rappresenta il file originale in termini di fasi di respiro e fasi di pausa, stimate da un ascolto del file. I valori di questo grafico sono approssimativi e non è possibile ottenere valori più

| Nome file | Durata | Frequenza | Classificazione dei suoni respiratori |
|-------------------------|--------|-----------|--|
| Coarse crackles.wav | 12s | 25/60Hz | Normali e presenza di crackles |
| Normal vesicular.wav | 13s | 13.8/60Hz | Normali |
| Inspiratory stridor.wav | 14s | 21/60Hz | Normali e presenza di stridor durante la fase inspiratoria |
| Pleural friction.wav | 20s | 18/60Hz | Normali e presenza di frizione pleurica. Il suono è molto rumoroso e non è difficile distinguere le fasi respiratorie. |

Tabella 11.4: Descrizione dei file delle repository.

precisi se non si misura il flusso d'aria in modo diretto. Notiamo che gli ultimi 7 grafici dal basso sono semplicemente dei segmenti di retta, questo perchè l'algoritmo riconosce l'intero file come respirazione cioè non riconosce alcuna pausa. Mentre nei primi 5 grafici dal basso il segmento di retta può essere in basso ad indicare una pausa oppure in alto ad indicare la presenza di una inspirazione o di una espirazione.

11.3.3 Caso di test di localizzazione di una apnea troppo lunga

In questo caso di test prendiamo il file normal.wav reperito da [14], che contiene un respiro normale di un soggetto sano, lo concateniamo varie volte e aggiungiamo con Audacity una pausa respiratoria molto lunga. La pausa comincia al tempo 35s e termina al tempo 1h : 17s. La lunghezza totale del file è 1h : 41s. Il sistema riconosce in modo corretto la pausa dal tempo 35s al tempo 1h : 15s. L'allarme suona al tempo 63s cioè 30s dopo l'inizio della pausa, e questo è esattamente quello che ci aspettiamo. Quindi possiamo dire che questo caso di test si è concluso con successo.

11.4 Descrizione dei file di respiro reperiti da [14]

La tabella 11.4 seguente riporta una descrizione di alcuni dei file reperiti da [14]. Le colonne della tabella riportano nell'ordine: nome del file, durata del file, classificazione dei suoni respiratori contenuti nel file e infine frequenza di respirazione espressa in cicli di respirazione al secondo. La frequenza di campionamento di tutti i file (espressa in numero di campioni al secondo) è di 8000Hz e lo schema di respirazione è normale.

Parte IV

Conclusioni

Capitolo 12

Conclusioni

12.1 Commenti ai risultati ottenuti

I risultati ottenuti sono incoraggianti e costituiscono solo un punto di partenza verso un sistema usabile in uno scenario reale.

12.2 Sviluppi futuri

L'applicazione può essere modificata per:

- Migliorare il riconoscimento delle fasi respiratorie.
- Aggiungere la possibilità di classificare i suoni respiratori secondo la classificazione 2.6.
- Estendere l'applicazione con la funzione di riconoscere gli schemi di respiro secondo quanto specificato nella sezione 2.7

Si possono sviluppare alcuni pezzi mancanti dell'applicazione ad esempio:

- Implementare una interfaccia con uno stetoscopio elettronico e fare dei test su soggetti affetti da sindrome di apnea del sonno.
- Implementare un algoritmo di stima del flusso respiratorio.

Un altro sviluppo futuro consiste nello studiare la portabilità dell'applicazione su un dispositivo mobile, sia nel caso in cui si usa il microfono in dotazione del dispositivo che nel caso in cui il dispositivo riceva i dati da uno stetoscopio elettronico. Per questo scopo una tecnologia da valutare è J2ME.

Ci sono vari modi di procedere utili alla ricerca nell'ambito dell'analisi dei suoni respiratori, ad esempio:

- Creare un database di registrazioni di suoni respiratori.
- Creare un database di casi di test completo secondo quanto indicato nel capitolo 11.
- Creare un modello acustico approssimato del torace.
- Implementare dei meccanismi di tolleranza al rumore esterno. Ad esempio usare due microfoni: uno che registra il rumore ambientale e uno che registra i suoni respiratori e usare il modello acustico per estrarre il rumore ambientale dai suoni respiratori.
- Fare una analisi approfondita dello stato dell'arte della separazione dei suoni cardiovascolari dai suoni respiratori, partendo ad esempio dagli articoli [81, 82, 83, 84, 85, 86].

Capitolo 13

Bibliografia

- [1] G. Charbonneau, E. Ademovic, B.M.G. Cheetham, L.P. Malmberg, J. Vanderschoot, A.R.A. Sovijärvi. *Basic techniques for respiratory sound analysis*.
- [2] N. Gavriely, Y. Palti, G. Alroy. *Spectral characteristics of normal breath sounds*. Journal of applied physiology 1981 Feb;50(2):307-14.
- [3] S.K. Chowdhury, A.K.Majumder. *Digital spectrum analysis of respiratory sounds*. IEEE Transactions on Biomedical Engineering. Date of Publication: Nov. 1981. Volume: BME-28, Issue: 11 Page(s):784-788
- [4] Simon Müller. *Classification of Respiratory Sounds*. Institut für Stochastik und Anwendungen, Universitaet Stuttgart. www.isa.uni-stuttgart.de/LstStoch/Mueller/rbk_17_12_08.pdf. data di accesso 28/01/2013.
- [5] <http://atlantemedicina.wordpress.com/2008/11/04/auscultazione-del-torace/>
data di accesso 28/01/2013.
- [6] Zbigniew Korona, Mieczyslaw M. Kokar. *Lung sound recognition using model theory based feature selection and fusion*. Applied Signal Processing (October 1998), 5(3), pg. 152-169.
- [7] C. Kandaswamy. *Neural classification of lung sounds using wavelet coefficients*. Computer in biology and medicine, Volume 34, Issue 6, September 2004, Pages 523 - 537.
- [8] Kompis, M. *Distribution of inspiratory and expiratory respiratory sound intensity on the surface of the human thorax*. Engineering in Medicine and Biology Society, 1997. Proceedings of the 19th Annual International Conference of the IEEE.

- [9] Rosqvist T, Paajanen E, Kallio K, Rajala HM, Katila T, Piirilä P, Malmberg P, Sovijärvi A. *Toolkit for lung sound analysis*. Department of Technical Physics, Helsinki University of Technology, Finland.
- [10] M. Bahoura, C. Pelletier. *Respiratory sounds classification using cepstral analysis and gaussian mixture models*. Département de Mathématiques, d'Informatique et de Génie (DMIG). Université du Québec à Rimouski, Rimouski, Qc, Canada, G5L 3A1
- [11] P.A. Mastorocostas, D.N. Varsamis, C.A. Mastorocostas, C.S. Hilas. *A Dynamic Fuzzy Model for Processing Lung Sounds*. Department of Informatics and Communications Technological Educational Institute of Serres 62124, Serres, Greece.
- [12] Feng Jin, Student Member, IEEE and Farook Sattar, Member, IEEE. *A new automaed approach for identification of respiratory sounds*. School of Electrical and Electronic Engineering, Nanyang Technological University, Nanyang Avenue, Singapore 639798.
- [13] Hans Pasterkamp, Steve S. Kraman, George R. Wodicka. *Respiratory sounds: advances beyond the stethoscope*.
- [14] Sound repositories:
- http://solutions.3mitalia.it/wps/portal/3M/it_IT/Littmann/stethoscope/education/heart-lung-sounds/
 - <http://tracheostomy.com/resources/videos/index.htm>
 - <http://faemse.org/downloads.html>
 - <http://www.meddean.luc.edu/lumen/MedEd/medicine/pulmonar/pd/auditory.htm>
- [15] Antonietta Bisulli. *Sindrome delle apnee ostruttive nel sonno(osas): effetti cognitivi del trattamento con pressione continua positiva(CPAP)*. Dottorato di ricerca. Università di Bologna.
- [16] NHLBI: Health Information for the Public. *Sleep Apnea: What Is Sleep Apnea?*. U.S. Department of Health and Human Services.
- [17] H. Klar Yaggi, M.D., M.P.H., John Concato, M.D., M.P.H., Walter N. Kernan, M.D., Judith H. Lichtman, Ph.D., M.P.H., Lawrence M. Brass, M.D., and Vahid Mohsenin, M.D. *Obstructive Sleep Apnea as a Risk Factor for Stroke and Death*. New England Journal of Medicine 2005; 353:2034-2041N.

- [18] Apoor S. Gami, M.D., Daniel E. Howard, B.S., Eric J. Olson, M.D., and Virend K. Somers, M.D., Ph.D. *Day Night Pattern of Sudden Death in Obstructive Sleep Apnea*. New England Journal of Medicine 2005; 352:1206-1214.
- [19] Christopher Wren, Sam Richmond, Liam Donaldson. *Presentation of congenital heart disease in infancy: implications for routine examination*. Archive of Disease in Childhood; Fetal and Neonatal Ed. 1999 January; 80(1): F49–F53.
- [20] Jussi Pekonen. *Onset detection methods for musical sounds*. Helsinki university of technology.
- [21] Juan Pablo Bello, Laurent Daudet, Samer Abdallah, Chris Duxbury, Mike Davies, and Mark B. Sandler. *A Tutorial on Onset Detection in Music Signals*.
- [22] Ismo Kauppinen. *Methods for detecting impulsive noise in speech and audio signals*. University of Turku, Department of Physics, FIN-ZOO14 Turku, Finland.
- [23] <http://pacs.unica.it/biblio/fisiopatologia/fisiopatologia1.pdf>
data di accesso: 28/01/2013.
- [24] Roberto Marani, Gennaro Gelao, and Anna Gina Perri. *A New System for Continuous Monitoring of Breathing and Kinetic Activity*. Electrical and Electronic Department, Polytechnic of Bari, Via E. Orabona, Bari 70125, Italy.
- [25] <http://it.wikipedia.org/wiki/Invasivit%C3%A0>
data di accesso: 28/01/2013.
- [26] Arzt M, Young T, Finn L, Skatrud JB, Bradley TD. *Association of sleep disordered breathing and the occurrence of stroke*. Toronto General Hospital, University Health Network, 9N 943, 200 Elizabeth Street, Toronto, ON M5G 2C4, Canada.
- [27] Yildirim I, Ansari R, Moussavi Z. *Automated respiratory phase and onset detection using only chest sound signal*. University of Illinois at Chicago, IL 60607, USA.
- [28] Saiful Huq, Azadeh Yadollahi, Zahra Moussavi. *Breath Analysis of Respiratory Flow using Tracheal Sounds*. Department of Electrical and Computer Engineering, University of Manitoba, Winnipeg, MB, Canada.
- [29] <http://www.treccani.it/vocabolario/pneumotacografo/>
data di accesso: 29/01/2013.
- [30] http://it.wikipedia.org/wiki/Spirometro#Spirometro_con_pneumotacografo
data di accesso: 29/01/2013.

- [31] <http://www.nlm.nih.gov/medlineplus/ency/article/003932.htm>
data di accesso: 04/02/2013.
- [32] Tiago H. Falk and Wai Yip Chan. *Modulation filtering for heart and lung sound separation from breath sound recordings*. Department of Electrical and Computer Engineering, Queen's University, Canada.
- [33] *Acoustical respiratory signal analysis and phase detection*. S. Le Cam, Ch. Collet, F. Salzenstein. Université Strasbourg 1.
- [34] A.R.A. Sovijärvi, J. Vanderschoot, J.E. Earis. *Standardization of computerized respiratory sound analysis*.
- [35] Yee Leng Yap, Zahra Moussavi, *Respiratory onset detection using variance fractal dimension*, Department of Electrical Engineering, University of Manitoba.
- [36] January Gnitecki, Zahra Moussavi. *The fractality of lung sounds: A comparison of three waveform fractal dimension algorithms*. Faculty of Engineering, Department of Electrical and Computer Engineering, University of Manitoba, Winnipeg, MB Canada R3T 5V6.
- [37] Joo S. Chuah , Zahra K. Moussavi. *Automated respiratory phase detection by acoustical means*. Department of Electrical and Computer Engineering, University of Manitoba, Winnipeg, Manitoba, R3T 2N2, Canada.
- [38] Z. Moussavi, M. Leopando, H. Pasterkamp, G. Rempel. *Computerized acoustical respiratory phase detection without airflow measurement*. Medical and Biological Engineering and Computing 2000, Volume 38, Issue 2, pp 198-203.
- [39] H. A. Mansy, T. J. Royston, R. A. Balk, R.H. Sandier. *Pneumothorax detection using computerised analysis of breath sounds*. Medical and Biological Engineering and Computing 2002, Volume 40, Issue 5, pp 526-532.
- [40] Laura Mason. *Signal Processing Methods for Non-Invasive Respiration Monitoring*. Trinity College Michaelmas 2002.
- [41] Gina Ann Yi. *A Software Toolkit for Acoustic Respiratory Analysis*. Massachusetts institute of technology.
- [42] Stephen Webley Hainsworth. *Techniques for the Automated Analysis of Musical Audio*. Magdalene College December 2003.

- [43] M J Tobin; T S Chadha; G Jenouri; S J Birch; H B Gazeroglu; M A Sackner. *Breathing patterns. 1. Normal subjects..* Chest Journal. 1983;84(2):202-205.
- [44] <http://www.meddean.luc.edu/lumen/MedEd/medicine/pulmonar/physio/pf11.htm>
data di accesso: 01/02/2013.
- [45] Susan Redline¹, Gayane Yenokyan², Daniel J. Gottlieb^{3,4}, Eyal Shahar⁵, George T. O'Connor³, Helaine E. Resnick^{6,7}, Marie Diener-West², Mark H. Sanders⁸, Philip A. Wolf³, Estella M. Geraghty⁹, Tauqeer Ali⁹, Michael Lebowitz¹¹, and Naresh M. Punjabi. *Obstructive Sleep Apnea Hypopnea and Incident Stroke.*
- [46] Roberto Munoz, Joaquín Duran-Cantolla, Eduardo Martínez-Vila, Jaime Gallego, Ramón Rubio, Felipe Aizpuru and Germán De La Torre. *Severe Sleep Apnea and Risk of Ischemic Stroke in the Elderly.*
- [47] www.cis.upenn.edu/~lee/06cse480/lec-real-time-scheduling.pdf
data di accesso: 02/02/2013.
- [48] <http://en.wikipedia.org/wiki/Real-time>
data di accesso: 02/02/2013.
- [49] S.M. Kuo, B.H. Lee, and W. Tian. *Real-Time Digital Signal Processing: Implementations and Applications.* Wiley, 2006. ISBN 0-470-01495-4.
- [50] http://it.wikipedia.org/wiki/File:Respiratory_system_complete_it.svg
data di accesso: 04/02/2013.
- [51] <http://it.wikipedia.org/wiki/Respirazione>
data di accesso: 04/02/2013.
- [52] Ronald B. George, Richard A. Matthay, Michael A. Matthay, Richard W. Light. *Chest Medicine: Essentials of Pulmonary and Critical Care Medicine. Edition 5.* Lippincott Williams and Wilkins.
- [53] Žilbert Tafa, Radovan Stojanović. *Bluetooth based approach to monitoring biomedical signals.* Proceedings of the 5th WSEAS International Conference on Telecommunications and Informatics, Istanbul, Turkey, May 27-29, 2006 (pp415-420).
- [54] <http://code.google.com/p/jstk/>
data di accesso: 04/02/2013.

- [55] http://en.wikipedia.org/wiki/Client-server_model
data di accesso: 04/02/2013.
- [56] <http://docs.oracle.com/javase/1.5.0/docs/guide/serialization/spec/serial-arch.html#6428>
data di accesso: 04/02/2013.
- [57] Glenn Fung. *A Comprehensive Overview of Basic Clustering Algorithms*. June 22, 2001.
- [58] S.S. Kraman. *Lung Sounds: An Introduction to the Interpretation of the Auscultatory Finding*. Northbrook, IL: Amer. College of Chest Physicians, 1993, audio tape.
- [59] Sergios Theodoridis, Konstantinos Koutroumbas. *Pattern recognition*. Elsevier Academic Press.
- [60] C. Richard Johnson, Jr, William A. Sethares, Andrew G. Klein. *Software Receiver Design: Build Your Own Digital Communication System in Five Easy Steps*. Cambridge University Press. p. 417.
- [61] http://en.wikipedia.org/wiki/File:Signal_envelopes.png
data di accesso: 04/02/2013.
- [62] Belle A. Sheno. *Introduction to digital signal processing and filter design*. John Wiley and Sons.
- [63] *A Neural Network System for Detection of Obstructive Sleep Apnea Through SpO2 Signal Features*
- [64] *Chains de Markov Triplet*. Wojciech Pieczynski.
- [65] *Multisensor triplet Markov chains and theory of evidence*. Wojciech Pieczynski.
- [66] Albert Boggess, Francis J. Narcowich. *A first course in wavelets with Fourier analysis*.
- [67] Paolo Marcellini, Carlo Sbordone. *Analisi matematica uno*.
- [68] <http://it.wikipedia.org/wiki/Fourier-Transformation>.
- [69] Rami Cohen. *Signal Denoising Using Wavelets*. Department of Electrical Engineering Technion, Israel Institute of Technology.
- [70] R.J.E. Merry. *Wavelet Theory and Applications. A literature study*.

- [71] Analisi dei segnali campionati. www.diee.unica.it/misure/Dispense/Misure_Elettroniche_dm270/Analisi_di_segnaali_campionati.pdf. data di accesso: 04/02/2013.
- [72] Carlo Drioli, Nicola Orio, *Elementi di Acustica e Psicoacustica*
- [73] Corso di elaborazione audio. <http://www.mediasystemnet.it/videocorsi.html>. data di accesso: 04/02/2013.
- [74] <http://editoria.wiki-site.com/index.php/DIGITALIZZAZIONE>. data di accesso: 04/02/2013.
- [75] http://fisicaondemusica.unimore.it/Teorema_di_Fourier.html. data di accesso: 04/02/2013.
- [76] <http://www.suonoelettronico.com>. data di accesso: 04/02/2013.
- [77] *Gli stetoscopi elettronici*. Giorgio Carlo Monti, Massimo Fusello. http://www.simg.it/Documenti/Rivista/2002/08-10_2002/12.pdf. data di accesso: 04/02/2013.
- [78] <http://docs.oracle.com/javase/6/docs/technotes/guides/swing/>. data di accesso: 04/02/2013.
- [79] <http://code.google.com/p/jmathplot/>. data di accesso: 04/02/2013.
- [80] <http://audacity.sourceforge.net/> data di accesso: 04/02/2013.
- [81] *Blind source extraction of heart sound signals from lung sound recordings exploiting periodicity of the heart sound*. T. Tsalaile, S. M. Naqvi, K. Nazarpour, S. Sanei and J. A. Chambers.
- [82] *Heart sounds separation from lung sounds using independent component analysis*. M. T. Pourazad, Z. Moussavi, F. Farahmand, R. K. Ward. Proceedings of the 2005 IEEE Engineering in Medicine and Biology 27th Annual Conference Shanghai, China, September 1-4, 2005.
- [83] *Modulation filtering for heart and lung sound separation from breath sound recordings*. Tiago H. Falk and Wai-Yip Chan. International Journal of Computer and Electrical Engineering, Vol. 2, No. 3, June, 2010 1793-8163.
- [84] *Separating Heart Sound from Lung Sound Using LabVIEW* T E Ayoob Khan, P Vijayakumar.

- [85] *Separating Heart Sounds from Lung Sounds* January Gnitecki, Zahara M. K. Moussavi.
- [86] *Separation of heart sound signal from lung sound signal by adaptive line enhancement* Thato Tsalaile, Saeid Sanei. 15th European Signal Processing Conference (EUSIPCO 2007), Poznan, Poland, September 3-7, 2007, copyright by EURASIP.