



**UNIVERSITÀ DEGLI STUDI DI MILANO**  
**FACOLTÀ DI SCIENZE MATEMATICHE,**  
**FISICHE E NATURALI**

*Corso di Laurea in Scienze e Tecnologie della comunicazione musicale*

**TRASCRIZIONE AUTOMATICA DELLA MELODIA DA  
BRANI MONOTIMBRICI PER CHITARRA**

RELATORE

Prof. Luca Andrea LUDOVICO

CORRELATORE

Dott. Adriano BARATE'

TESI DI LAUREA DI

Roberto LIMONTA

Matr. 738913

Anno Accademico 2010/2011



*Sono tante le persone che mi hanno accompagnato e aiutato lungo questo percorso. Ringrazio i miei genitori e la mia famiglia per avermi dato la possibilità di iniziare un cammino e di averlo portato a termine sostenendomi in ogni scelta e in ogni momento, soprattutto nelle sere prima degli esami. Un ringraziamento speciale va a mia sorella, una persona molto importante nella mia vita, capace di aiutarmi a risolvere tutto con estrema facilità, facendomi ritrovare la strada giusta nei momenti più difficili. La ringrazio per avermi sempre stimolato a fare il meglio.*

*Gli amici. Fortunatamente sono tanti e, grazie a loro, ho trascorso tanti bei momenti insieme. Ricordo tutte le sere passate a ridere e a scherzare fino a tarda notte senza accorgercene, le partite di calcetto interminabili e i viaggi in macchina passati a cantare come pazzi. In particolare vorrei ringraziare coloro che mi sono stati più vicini: Fabio, Fede, Kri, Bea e Buddha.*

*Ringrazio tutta BmRadio, Berny, Diego e Lory per avermi dato la possibilità di fare lo stage e per tutti i pomeriggi nei quali, invece di studiare, abbiamo fatto dirette e registrazioni.*

*Molto importanti sono stati tutti i miei compagni di studio. Ringrazio Kekko, Banzi, Ste Zack, Jack, Ste, Andre, Sam e tutti gli altri con i quali ho passato intere giornate a studiare, seguire lezioni e sostenerci l'uno con l'altro senza, però, privarci di momenti di divertimento e di risate.*

*Un ringraziamento lo dedico a tutti i professori dell'università per gli insegnamenti che mi hanno trasmesso.*

*Ringrazio Dario che, con le sue nozioni, mi ha dato un grande aiuto per realizzare il progetto.*

*Infine, vorrei ringraziare la mia ragazza Maria che ho conosciuto proprio all'inizio di questo percorso e che, per tutto questo tempo, mi è sempre stata vicina nel migliore dei modi.*

## Sommario

0. INTRODUZIONE .....	6
1. ELEMENTI DI ACUSTICA .....	8
1.1 Il suono .....	8
1.1.1 Il tono puro.....	9
1.1.2 Il tono complesso .....	10
1.2 Intervalli frequenziali.....	11
2. ANALISI DEL SUONO DI CHITARRA .....	13
2.1 Caratteristiche dello strumento .....	13
2.2 La forma d'onda.....	14
2.3 Lo spettro .....	15
3. ALGORITMI .....	18
3.1 Algoritmi per il timbro.....	18
3.1.1 Riconoscimento vocale .....	18
3.1.2 Riconoscimento dei modelli timbrici.....	19
3.2 Algoritmo per la dinamica .....	20
4. PROTOTIPO .....	21
4.1 Strumenti utilizzati.....	22
4.2 Il progetto.....	22
4.2.1 Specifiche di ingresso .....	23
4.2.2 Analisi temporale .....	24
4.2.3 Analisi spettrale .....	28
4.2.4 Spartito.....	31
5. TEST SET .....	34
5.1 Nota singola .....	34
5.2 Due note .....	35
5.3 Semitoni .....	36
5.4 Corda a vuoto.....	37
5.5 Nota/pausa/nota .....	38
5.6 Scala cromatica .....	39
5.7 Scala maggiore.....	41
5.8 Bicordo.....	42
5.9 Accordo.....	43
5.10 Utilizzo delle dita per pizzicare le corde .....	44
5.11 Arpeggio .....	45
5.12 Intervallo composto .....	46
5.13 Armonico naturale .....	47
5.14 Bending .....	48
5.15 Slide .....	49
5.16 Vibrato .....	50
5.17 Terzina .....	51
5.18 Hammer-on .....	52

5.19 Pull-off .....	53
5.20 Semplice melodia.....	55
6. CONSIDERAZIONI E CONCLUSIONI .....	58
6.1 Risultati .....	58
6.2 Problematiche riscontrate e risolte.....	59
6.3 Implementazioni future.....	60
7. BIBLIOGRAFIA E SITOGRAFIA .....	62
7.1 Bibliografia .....	62
7.2 Sitografia.....	63

## 0. INTRODUZIONE

Ogni segnale ha la caratteristica di contenere delle informazioni. Queste possono essere estratte, elaborate e ritrasmesse. In particolare, l'estrazione dell'informazione è un passaggio molto importante in quanto prevede l'utilizzo di diversi algoritmi per ottenere il risultato richiesto.

In campo musicale esistono algoritmi che sono in grado di fornire un'informazione precisa, come, ad esempio, l'altezza di un segnale, mentre altri che ci forniscono risposte molto più complesse.

Un algoritmo è un metodo utilizzato per ottenere un certo tipo di risultato tramite un numero di passi finiti. Partendo proprio da questa semplice definizione si andrà a creare e a sviluppare il seguente elaborato, ricercando e studiando i concetti base di analisi di un segnale audio.

Nel primo capitolo del seguente elaborato si presenteranno gli elementi di acustica grazie allo studio dei segnali semplici e composti, mettendo in evidenza le caratteristiche più importanti di ognuno di questi analizzando i parametri fondamentali del segnale: altezza, ampiezza e timbro. In conclusione al capitolo verrà data una definizione di intervallo frequenziale spiegando brevemente la scala temperata e le sue caratteristiche principali.

Nel secondo capitolo si entrerà nello specifico: verrà preso in considerazione un segnale particolare, ovvero un suono di chitarra. Verrà fatta una descrizione del suo spettro analizzando gli elementi che lo compongono ed, infine, verranno inseriti dei cenni sull'acustica dello strumento.

Il terzo capitolo sarà dedicato alla descrizione di alcuni algoritmi noti in letteratura, in particolare, riguardanti l'altezza e il timbro di un suono molto utili per lo sviluppo pratico e teorico del prototipo creato.

Il quarto capitolo è il cuore dell'elaborato. Verrà presentato il progetto di trascrizione automatica della melodia da brani monotimbrici per chitarra. Verranno presentati parti del codice sorgente ed analizzati descrivendo passo dopo passo i calcoli effettuati dal prototipo elencando i punti forti ed i limiti.

Il quinto capitolo prevede un sezione dedicata al “test set”, dove verranno analizzati dei campioni registrati indicandone così i risultati ottenuti, quindi la qualità del prototipo in esame.

Nel sesto capitolo verranno fatte alcune considerazione relative al lavoro svolto sottolineando così la buona riuscita o meno del prototipo. Inoltre, verrà dedicata una sezione per eventuali implementazioni future da poter applicare.

L'ultimo capitolo è composto da una sitografia e una bibliografia dei testi utilizzati per lo sviluppo del presente elaborato, riportando i vari materiali utilizzati per lo studio di algoritmi, elementi di acustica, del linguaggio di programmazione Matlab e teoria musicale.

# 1. ELEMENTI DI ACUSTICA

## 1.1 Il suono

Il suono è un fenomeno generato dalla variazione dello stato di quiete di un gas, solitamente aria. L'onda acustica è generata da una sorgente messa in vibrazione creando così questa variazione di pressione che può essere divisa in due parti: compressione e rarefazione. Ogni onda acustica è caratterizzata dal periodo ovvero il tempo che un'onda ci impiega a completarsi passando quindi dalla compressione alla rarefazione. L'unità di misura del suono è la frequenza, cioè il numero di cicli di compressione e rarefazione nell'unità di tempo:

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

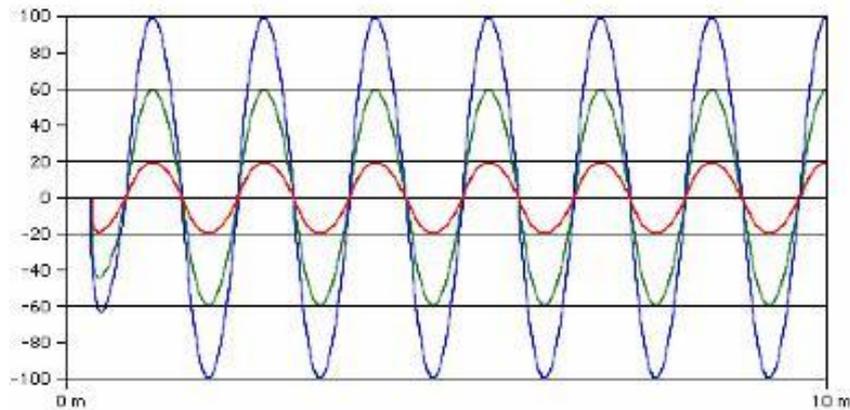
L'unità di misura della frequenza è l'Hz; l'orecchio umano è in grado di percepire suoni che partono dai 20Hz fino a raggiungere i 20kHz.

Un'altra caratteristica molto importante del suono è l'ampiezza. Questa grandezza permette di rilevare se un suono è più forte o più debole di un altro suono.

Abbiamo infine il timbro che delinea la tipologia dello strumento dal quale il suono è stato emesso.

La seguente tabella riassume brevemente la relazione tra i parametri percettivi e quelli fisici:

<b>Parametro Percettivo</b>	<b>Parametro Fisico</b>	<b>Risultato</b>
Altezza	Frequenza	Suono grave/acuto
Intensità	Ampiezza	Suono forte/debole
Timbro	Forma d'onda	Natura del suono



**Figura 1** Grafico contenente tre sinusoidi con uguale frequenza, ma ampiezze diverse tra loro

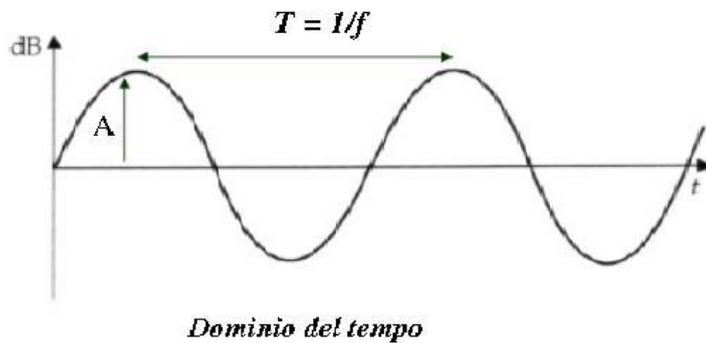
### 1.1.1 Il tono puro

Tra gli infiniti suoni che si possono produrre in natura, il più semplice di tutti, ma anche il più importante, è il tono puro. Questo suono è un'onda acustica semplice che nel corso degli anni è stata soggetta a diversi studi per esaminare il comportamento e le caratteristiche dei suoni.

Il tono puro è un segnale periodico la cui forma d'onda è quella di una semplice senoide ed è caratterizzato dalla seguente espressione:

$$A(t) = A \sin(2\pi ft + \varphi)$$

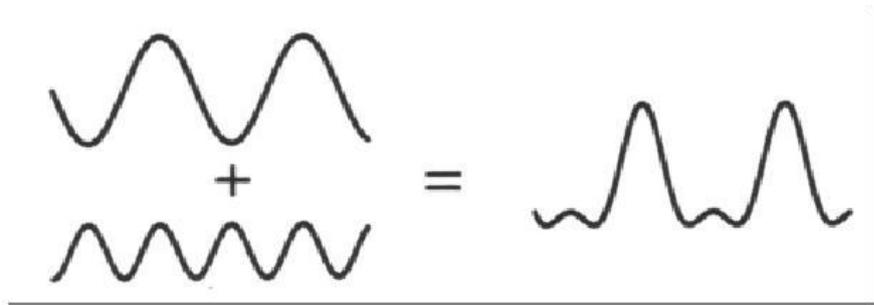
Come si può notare già dalla formula, questo tipo di segnale è caratterizzato da una sola ampiezza ( $A$ ), una sola frequenza ( $f$ ) e una sola fase ( $\varphi$ ) che permettono così di rilevare il tono puro.



**Figura 2** Comportamento di una sinusoide nel dominio del tempo

### 1.1.2 Il tono complesso

Il tono complesso si ottiene dalla sovrapposizione di due o più toni puri con frequenze, ampiezze a fasi diverse tra loro. Ciò che caratterizza un suono complesso dagli altri è il timbro, ovvero la struttura frequenziale del suono stesso che ci permette di riconoscere la sorgente dalla quale è stato generato. Un suono di chitarra avrà un timbro nettamente diverso da quello di una tromba proprio perché la composizione spettrale è differente. Ciò che ci permette di riconoscere il timbro di un suono è l'attacco ovvero l'inizio della nota, questo perché la maggior parte delle informazioni sono contenute proprio in quella zona del suono. D'altro canto se non ci fosse l'attacco della nota sarebbe quasi impossibile decifrare il timbro del suono preso in esame; in natura, infatti, la maggior parte dei suoni che ci circonda sono suoni complessi con caratteristiche spettrali ben distinte.



**Figura 3** Somma di due sinusoide che formano un suono complesso

L'analisi di Fourier consente di scomporre dei segnali non periodici nelle componenti più elementari ottenendo, così, l'analisi frequenziale del segnale chiamato spettro frequenziale. La trasformata di Fourier rappresenta un segnale come una somma lineare di sinusoidi che prenderanno il nome di componenti frequenziali del segnale. La frequenza più bassa è detta frequenza fondamentale del segnale (armonica fondamentale) mentre le altre armoniche, con frequenze multiple alla fondamentale, sono chiamate armoniche secondarie.

Proprio basandoci sul principio sopra enunciato andremo successivamente ad analizzare, in modo più dettagliato, il suono e lo spettro di una chitarra classica.

## 1.2 Intervalli frequenziali

Come già descritto precedentemente, l'orecchio umano è in grado di percepire suoni con frequenza compresa tra i 20Hz e i 20kHz mentre tutte le altre frequenze vengono denominate infrasuoni, se al di sotto dei 20Hz, o ultrasuoni, se al di sopra dei 20kHz.

Nel paragrafo dedicato al tono complesso è stato spiegato come un suono sia composto da diverse componenti armoniche, ma solo una frequenza è chiamata fondamentale, la quale darà il nome alla nota. Proprio partendo da questo concetto verranno analizzati gli intervalli frequenziali, quindi la relazione tra una nota e l'altra.

Gli intervalli frequenziali utilizzati per la maggior parte delle applicazioni audio sono di due tipi:

- **ottava**, intervallo frequenziale che separa due frequenze, di cui una è il doppio dell'altra (es. 220Hz / 440Hz).
- **Decade**, intervallo frequenziale che separa due frequenze, di cui una è dieci volte l'altra (es. 20Hz/ 200Hz).

Nella costruzione degli strumenti musicali vengono utilizzati altri tipi di intervalli frequenziali. Il più utilizzato è l'intervallo minimo appartenente alla scala musicale temperata: il **semitono**. La scala temperata ha quindi la caratteristica di essere composta da una ottava divisa in 12 parti uguali tra loro e ogni semitono corrisponde alla frequenza del semitono precedente moltiplicato per  $2^{1/2}$ , in modo tale che la prima nota dell'ottava successiva avrà una frequenza doppia rispetto alla prima nota dell'ottava

precedente; quindi il rapporto di frequenze che identifica il semitono temperato è pari a  $f = \sqrt[12]{2}$ .

La seguente tabella riporta gli intervalli frequenziali della scala temperata:

Unisono	1:1	$1$
Semitono	1,059463:1	$f$
Tono	1,122462:1	$f^2$
Terza minore	1,189201:1	$f^3$
Terza maggiore	1,259921:1	$f^4$
Quarta giusta	1,334840:1	$f^5$
Quarta aumentata	1,414214:1	$f^6$
Quinta giusta	1,498307:1	$f^7$
Sesta minore	1,587401:1	$f^8$
Sesta maggiore	1,681793:1	$f^9$
Settima minore	1,781797:1	$f^{10}$
Settima maggiore	1,887749:1	$f^{11}$
Ottava	2:1	$2$

## 2. ANALISI DEL SUONO DI CHITARRA

Il seguente capitolo ha il compito di analizzare il suono di una chitarra. Il presente elaborato prende in considerazione diversi suoni di chitarra (acustica) che verranno analizzati sia dal punto di vista della forma d'onda che per quanto riguarda quello spettrale.

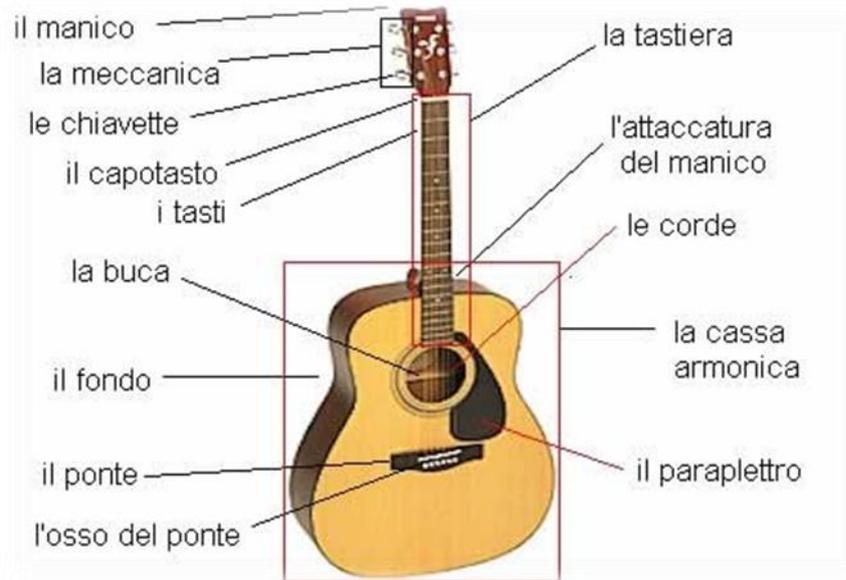
### 2.1 Caratteristiche dello strumento

La corda della chitarra è un classico esempio di corda elastica fissata agli estremi che, dopo una eccitazione forzata, inizia a vibrare. Seguendo proprio la definizione di **suono**, una volta pizzicata la corda, questa inizia a muovere le molecole d'aria circostanti creando così delle zone di compressione e rarefazione generando una specifica frequenza e quindi una nota. Quando viene pizzicata la corda, questa inizia oscillare verso l'alto e verso il basso e, grazie alla cassa di risonanza della chitarra, queste vibrazioni saranno amplificate così da poter essere udibili dal nostro orecchio. Diverso invece è il discorso che riguarda una chitarra elettrica, in quanto il suono viene prodotto dall'interazione elettromagnetica tra corde metalliche e pick-up posti nelle vicinanze del ponticello.

Una frequenza generata da una corda di chitarra pizzicata ha le seguenti caratteristiche:

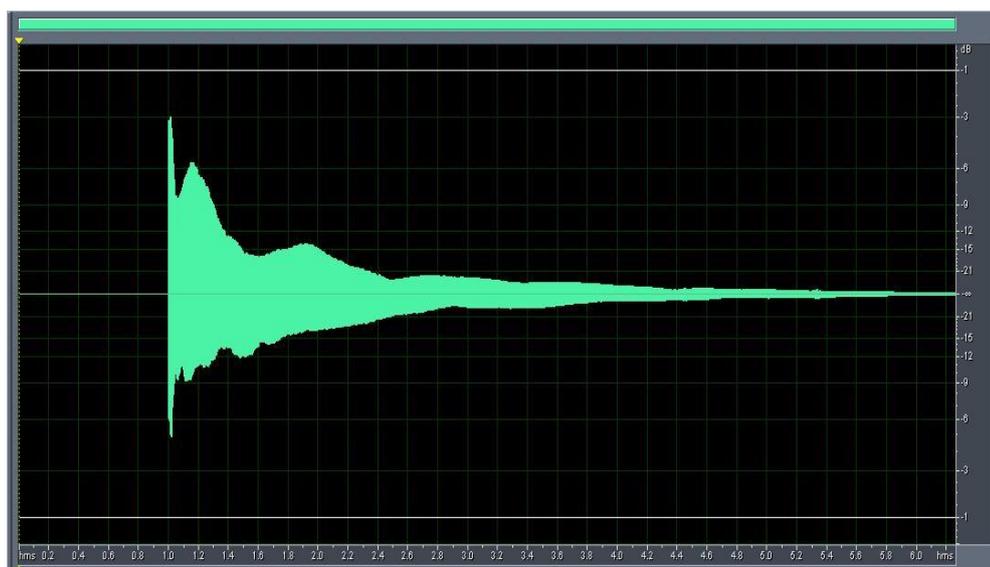
- è inversamente proporzionale alla lunghezza della corda: più lunga è una corda, minore è il numero delle vibrazioni al minuto secondo e meno acuto è il suono prodotto;
- è inversamente proporzionale al diametro: più grossa è una corda, minore è il numero delle vibrazioni e meno acuto il suono prodotto;
- è direttamente proporzionale al quadrato della tensione: più si tende una corda, maggiore è il numero di vibrazioni e più acuto è il suono prodotto;
- è inversamente proporzionale al quadrato della densità: più la corda è densa, minore è il numero delle vibrazioni e meno acuto è il suono prodotto.

Da queste caratteristiche sopra enunciate possiamo quindi dedurre che per generare suoni acuti occorrono corde sottili, corte e ben tese; per ottenere suoni gravi occorrono corde spesse, lunghe e leggermente tese.



**Figura 4** Elementi principali che compongono la chitarra

## 2.2 La forma d'onda



**Figura 5** Suono emesso dalla vibrazione di una corda di chitarra

La figura sopra riporta la forma d'onda (un LA 220 Hz) di una chitarra. Si può notare che l'attacco della nota è molto deciso, ciò è dovuto dal contatto del plettro (o del dito) con la corda. Successivamente avviene uno smorzamento ondulato del suono causato dalla vibrazione della corda che pian piano tende ad estinguersi.

Bisogna però precisare che nella chitarra una stessa nota può avere caratteristiche leggermente diverse in base alla posizione nella quale viene suonata: per esempio, un LA suonato con la quinta corda a vuoto avrà un suono leggermente differente rispetto a un LA suonato sul quinto tasto della sesta corda nonostante si stia parlando della stessa nota, alla stessa frequenza. Altra caratteristica di questo strumento è il modo in cui vengono suonate le corde: solitamente si utilizza il plettro oppure le dita. Questi due modi di suonare creano un suono diverso tra loro modificando, anche se in modo leggero, il timbro. Per concludere anche la zona in cui viene pizzicata la corda contribuisce a variare il timbro del suono, per esempio se suonata al ponticello, al centro oppure verso il manico.

### 2.3 Lo spettro

Di fondamentale importanza nello sviluppo di questo elaborato è lo studio dello spettro del suono di una chitarra.

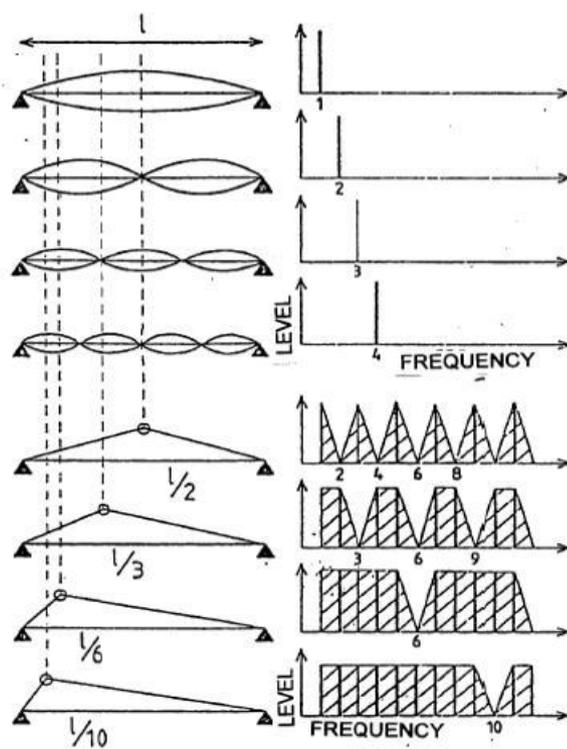
La chitarra è uno strumento che ha una banda di frequenza che parte da frequenze basse per raggiungere le frequenze medie. La seguente tabella riporta le frequenze delle sei corde della chitarra

<b>Nota</b>	<b>Frequenza (Hz)</b>
Mi <sub>2</sub>	82
La <sub>2</sub>	110
Re <sub>3</sub>	147
Sol <sub>3</sub>	196
Si <sub>3</sub>	247
Mi <sub>4</sub>	330

Il  $Mi_4$  non è l'ultima ottava disponibile dello strumento, ma sulla prima corda si può arrivare a raggiungere almeno un'altra ottava muovendosi lungo la corda del mi cantino (con la chitarra elettrica si può ottenere una maggiore estensione).

Analizzeremo ora nello specifico lo spettro del suono di chitarra. Iniziamo dal fatto che la corda è un corpo che vibra, fissato ai suoi estremi. Questo permette quindi di avere una lunghezza ben definita della corda. Perciò premendola con il dito sul manico ne variamo lunghezza e, di conseguenza, anche la frequenza che otteniamo; per esempio, se diminuiamo la lunghezza della corda della metà, si ottiene un suono di una ottava maggiore rispetto alla corda suonata a vuoto.

Lo spettro, inoltre, varia anche in base al punto nel quale viene sollecitata la corda. Nella seguente immagine riportiamo degli esempi di spettri di una chitarra.

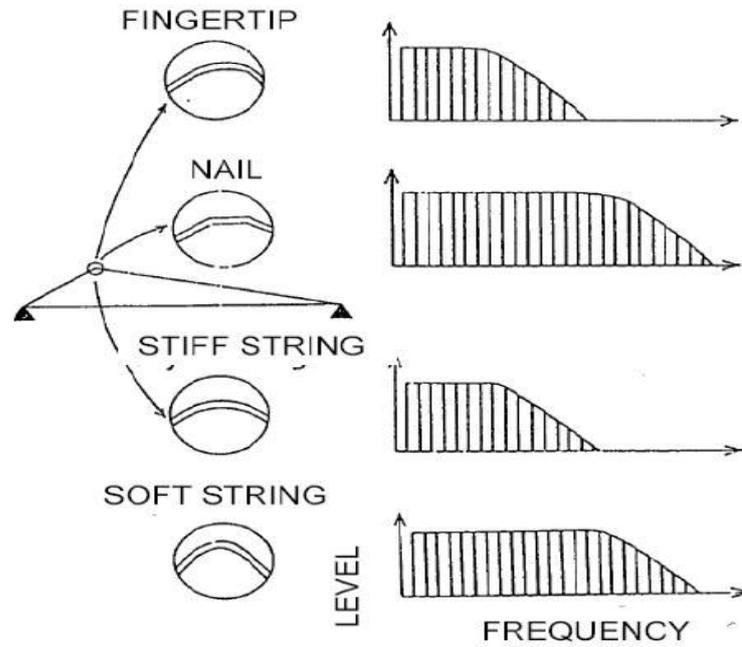


**Figura 6** Esempi di diversi spettri di suono di chitarra al variare del punto di sollecitazione della corda

Si può notare come varia lo spettro in base al punto di contatto.

In questa immagine, invece, si può notare come varia lo spettro al variare del modo in cui la corda viene pizzicata: nel primo caso con le dita, nel secondo con l'unghia.

Negli altri due casi viene presa in considerazione la tensione della corda: una molto tirata e l'altra più molle.



**Figura 7** Esempi di spettro del suono di chitarra al variare della tensione della corda e del modo in cui viene pizzicata.

## 3. ALGORITMI

Un algoritmo è, per definizione, un metodo per ottenere un certo risultato con un numero di passi finiti. In ingresso avremo dei dati che descrivono il problema e, di conseguenza, in uscita dei dati come risultato del problema.

Nel seguente paragrafo verranno riproposti degli algoritmi, alcuni fondamentali per la realizzazione del prototipo proposto mentre altri utili per gli studi di quest'ultimo. Tengo a precisare che i seguenti algoritmi sono già noti in letteratura e, in questo caso, si fa riferimento, tra gli altri, all'elaborato “ MAPPATURA DELLE INFORMAZIONI E SEGMENTAZIONE DI SEGNALI AUDIO” di G. Presti nel quale vengono considerati alcuni algoritmi utili per il segnale audio.

### 3.1 Algoritmi per il timbro

Diversi sono gli algoritmi per il timbro, in questo paragrafo verrà analizzato il riconoscimento vocale per poi estendere questo algoritmo ad un campo molto più ampio ovvero al riconoscimento dei modelli timbrici.

#### 3.1.1 Riconoscimento vocale

L'apparato fonatorio può essere paragonato ad uno strumento a corde con una cassa di risonanza. Nello specifico ci sono le corde vocali che emettono la nota mentre l'apparato risonatore è responsabile di un filtraggio, quindi l'interpretazione della nota come vocale. I picchi di risonanza determinati dalle cavità orali sono chiamati formanti.

Una specifica vocale è identificata dal rapporto tra seconda e prima formante:  $F2 / F1$  mentre tutte le altre formanti rappresentano semplicemente delle “sfumature” linguistiche.

Nella seguente tabella vengono rappresentati gli intervalli in cui possono cadere  $F1$  e  $F2$  delle vocali italiane:

Vocali	Voce maschile		Voce Femminile	
	F1 (Hz)	F2 (Hz)	F1 (Hz)	F2 (Hz)
i	200 – 400	1900 – 2500	200 – 300	2500 – 3100
é	300 – 500	1800 – 2400	300 – 500	2400 – 3000
è	500 – 600	1700 – 2000	500 – 700	2100 – 2900
a	600 – 900	1000 – 1500	800 – 1200	1200 – 1600
ò	500 – 600	750 – 1000	550 – 750	900 – 1500
o	400 – 500	700 – 800	400 – 500	750 – 1100
u	300 – 400	550 – 750	200 – 400	600 – 1100

Il procedimento con cui dato un segnale in ingresso, si risale al testo è divisi in alcuni passi:

1. Una volta ottenuto il segnale vocale tramite la FFT passiamo dal dominio temporale al dominio frequenziale. La FFT avviene su un segmento audio della durata di centesimi di secondo, ricavando così un grafico con le ampiezze di ogni frequenza che compone il suono.
2. Si riorganizzano i dati ottenuti tramite l'applicazione delle regole e del dizionario fonetico della lingua.
3. Riconoscimento dei singoli fonemi. Il riconoscitore vocale ha un database composto da molti grafici e, ognuno di questi, contiene molti dei suoni che la voce umana può produrre.
4. Composizione dei fonemi in parole.

### 3.1.2 Riconoscimento dei modelli timbrici

Le informazioni descritte nel capitolo precedente possono essere utilizzate per tutti gli strumenti che possiedono una cassa di risonanza. Una volta ottenuta la frequenza fondamentale del suono è possibile calcolare il rapporto tra questa e gli altri armonici. In questo modo è possibile identificare la famiglia a cui appartiene lo strumento o meglio ancora lo strumento stesso. Secondo la classificazione Hornbostel-Sachs gli strumenti musicali si possono dividere in ben cinque famiglie diverse:

1. Aerofoni: emettono un suono per mezzo di una colonna d'aria che vibra all'interno dello strumento.

2. Cordofoni: il suono è prodotto dalla vibrazione di una corda che può essere sfregata da un arco, pizzicata o percossa da un martelletto.
3. Membranofoni: il suono è prodotto dalla vibrazione di membrane.
4. Idiofoni: il suono è prodotto dalla vibrazione del corpo dello strumento stesso.
5. Elettrofoni: il suono è generato per via di un circuito elettrico.

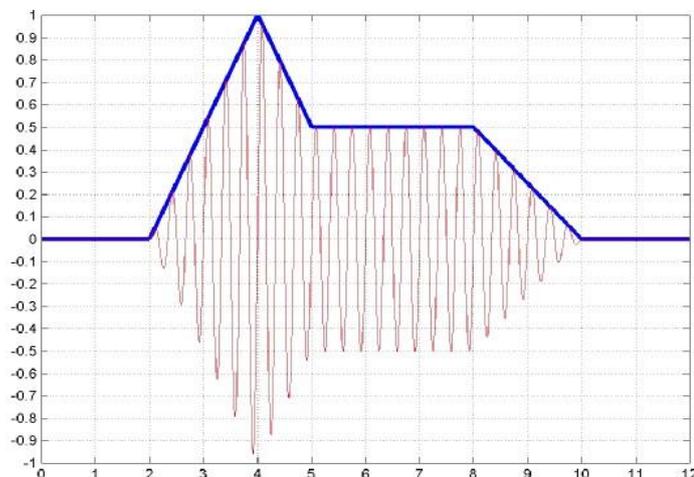
### 3.2 Algoritmo per la dinamica

La dinamica di un segnale indica l'escursione tra un valore minimo e uno massimo di un segnale. Questo è molto importante per studiare l'andamento del segnale stesso (nel nostro caso di un suono) per capire come esso si comporta nel tempo.

Di seguito verrà riportato un semplice algoritmo di studio della dinamica di un segnale con una breve spiegazione sulle varie fasi di un suono.

Una volta suonato una nota, questa può essere descritta in quattro fasi ben distinte:

1. Attacco: l'ampiezza raggiunge, a partire da zero, il suo valore massimo.
2. Decadimento: dopo l'attacco, parte dell'energia iniziale viene persa e l'ampiezza diminuisce.
3. Sostegno: l'ampiezza mantiene un livello quasi costante per un certo tempo.
4. Rilascio: l'ampiezza decresce fino ad estinguersi completamente.



**Figura 8** grafico di un segnale con il suo inviluppo

Dalla figura si può notare che il segnale (in rosso) è circoscritto da una curva che descrive l'andamento dell'ampiezza (in blu). Questa curva prende il nome di involuppo del segnale. L'involuppo è molto importante per studiare l'andamento del segnale che varia da strumento a strumento.

A partire dall'involuppo del segnale è possibile calcolare la derivata del segnale permettendo così di ricavare il fronte di salita o discesa di un segnale chiamato transiente. Il transiente, quindi, è una variazione di ampiezza molto rapida della nota ed è utile per discriminare gli istanti in cui il segnale ha subito una brusca mutazione dinamica.

## 4. PROTOTIPO

Finalmente entriamo nel cuore dell'elaborato. In questo capitolo verrà analizzato e spiegato il prototipo che ha il compito di trascrivere melodie di chitarra; inoltre verranno riportati alcuni pezzi di codice e i risultati ottenuti.

### 4.1 Strumenti utilizzati

Prima di iniziare con la vera e propria spiegazione, è giusto dedicare qualche riga agli strumenti utilizzati per la realizzazione del prototipo.

Per quanto riguarda la parte software è stato utilizzato il linguaggio di programmazione Matlab. Questo linguaggio è molto utile per l'analisi di segnali di ogni tipo anche se, nel caso specifico, sono stati utilizzati solo segnali audio a partire da una sinusoide fino ad arrivare a un segnale più complesso, cioè, quello del suono di una chitarra. Facilmente si possono ottenere dei grafici del segnale (o di una elaborazione del segnale) per poter anche visivamente analizzarne il comportamento.

Altri software utilizzati, per registrare le tracce audio, sono Cool Edit Pro e Reaper. Entrambi sono programmi di editing audio molto utili per creare dei suoni sintetici come semplici toni puri e toni composti fin ad arrivare a registrare delle piccole tracce di chitarra acustica. La loro utilità è stata notevole perché, una volta acquisito il segnale era possibile modificarlo, manipolarlo ed esportarlo secondo le specifiche più adatte.

Per quanto riguarda il microfono è stato utilizzato un BEHRINGER B-2 PRO posizionato a pochi centimetri dalla buca della chitarra in modo tale da acquisire per bene il suono che verrà comunque normalizzato a 0 dB successivamente. La chitarra utilizzata per le registrazioni è una chitarra acustica FLORENCIA.

### 4.2 Il progetto

Il prototipo creato ha come scopo quello di fornire una trascrizione a basso livello di una melodia di chitarra. Per trascrizione a basso livello si intende che il risultato finale ottenuto sarà una matrice contenente la frequenza della nota, il nome della nota, il secondo di inizio e il secondo di fine.

## 4.2.1 Specifiche di ingresso

Il prototipo assume in ingresso solo i file con determinate caratteristiche. Innanzitutto il file deve essere di tipo audio ed espressamente di tipo .wav, con frequenza di campionamento a 44100Hz a 16 bit. Questi tre parametri sono fondamentali.

```
[x, fs, bit]=wavread('nomefile.wav');  
sound(x, fs, bit);
```

La prima riga di codice ha il compito di leggere un file tramite il comando `wavread` e salva delle informazioni in tre matrici: `x`, `fs`, `bit`. Le ultime due rappresentano semplicemente la frequenza di campionamento e i bit di quantizzazione mentre la prima trasforma il file musicale in una successioni di valori.

Il comando `sound` permette di far suonare il file acquisito. Di seguito, riporto il grafico di un file .wav dato in ingresso a Matlab.

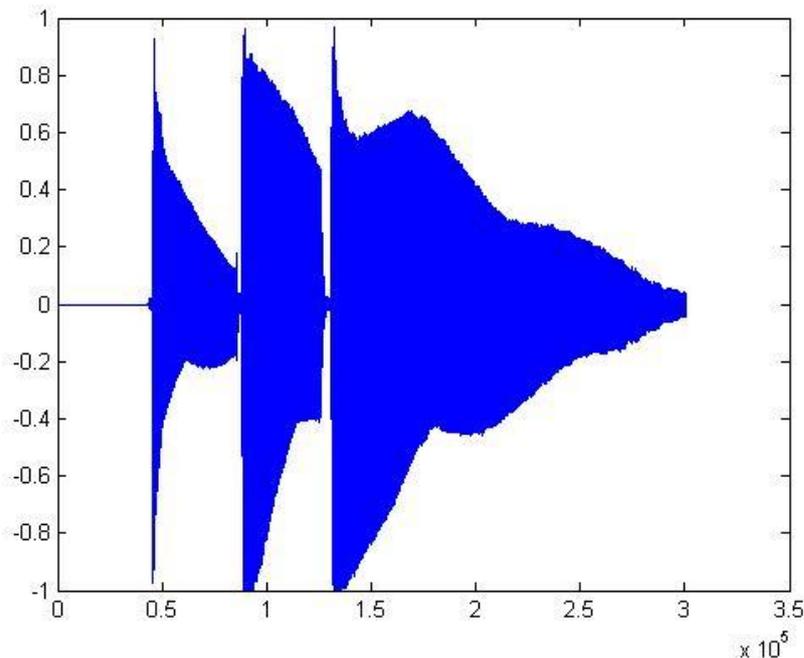


Figura 9 grafico di una melodia di chitarra in ingresso a Matlab

Per evitare degli errori riscontrati durante le varie prove è stato deciso di inserire una matrice 0 prima della matrice del segnale, ovvero `x`.

```
x=[zeros(fs,1); x];
```

In questo modo, prima dei valori della melodia vera e propria sono stati messi dei valori nulli, questo per evitare che il primo evento non fosse tenuto in considerazione da Matlab.

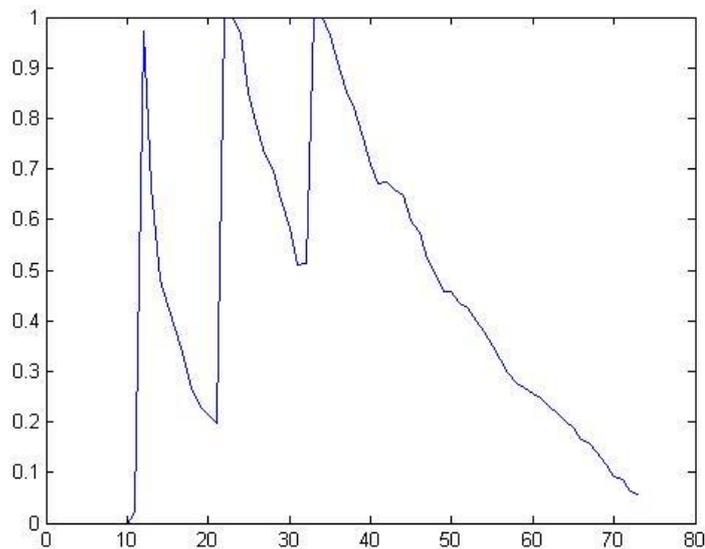
## 4.2.2 Analisi temporale

Una volta chiarite le specifiche di ingresso, possiamo analizzare le varie parti di codice che si occupano di elaborare il segnale ed estrarne delle informazioni molto utili. Partiamo inizialmente dall'analisi temporale, utile per determinare la fase di inizio e di fine di ogni evento con tutte le sue concause.

### **Inviluppo e derivata della melodia.**

L'analisi temporale inizia con l'individuazione dell'inviluppo della melodia in ingresso. Come già descritto nel capitolo 4.1.1, l'inviluppo di un segnale è fondamentale per descrivere la sua ampiezza.

Seguendo ciò che viene descritto nell'elaborato di G.Presti otteniamo l'inviluppo di una melodia in ingresso:



**Figura 10** grafico dell'inviluppo di una melodia in ingresso a Matlab

Il file audio preso in considerazione è una melodia composta da tre note. Si può notare come i tre eventi siano ben distinti l'uno dall'altro, composti da un picco iniziale seguito da una decadimento della nota.

Di seguito viene riportato il codice utile per ricavare l'involuppo del segnale:

```
E = abs(x);  
for i =1:floor(lsig/m)  
a(i) = max(E((i-1)*m+1 : i*m));  
end
```

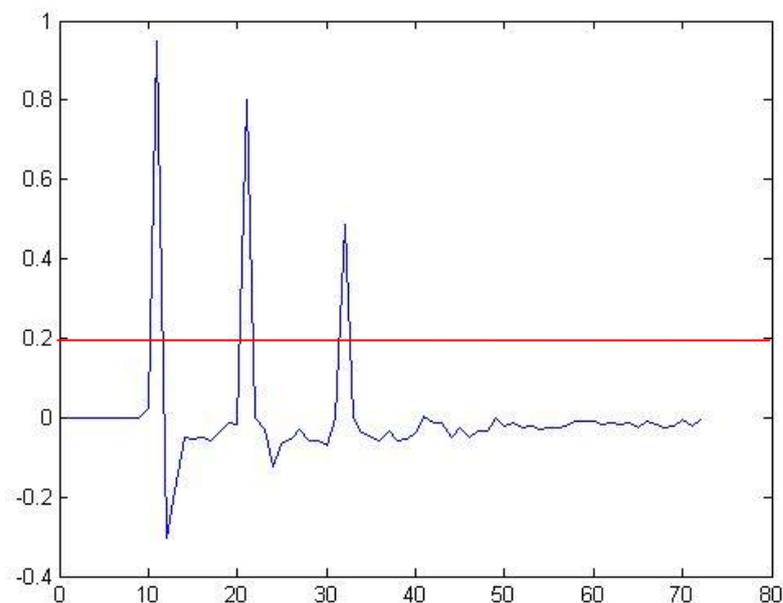
In primo luogo, viene calcolato il valore assoluto del segnale con il semplice comando `abs(x)`.

Il ciclo `for` permette, invece, di calcolare il valore massimo all'interno di ogni finestra. Si è posto il valore della finestra (`m`) a 4096 campioni. Facendo un rapporto tra la lunghezza del segnale (`lsig`) ed il valore della finestra otteniamo il numero di finestre che varia in base alla melodia data in ingresso. Il risultato che otteniamo è l'involuppo rappresentato precedentemente.

Come ultimo passo calcoliamo la derivata dell'involuppo:

```
d = diff(a);
```

Ciò che otteniamo è la seguente figura:



**Figura 11** grafico della derivata dell'involuppo della melodia

In questo modo ora è possibile determinare il picco dell'involuppo per identificare così l'inizio di ogni nota:

```
[v,idx]=find(d > 0.2);
```

Il valore di 0,2 (si veda la linea rossa nel grafico sopra) è un valore stimato durante le varie prove e ciò assicura che vengano presi in considerazione tutti gli attacchi delle note senza, però, interferire con il resto della nota stessa. In questo modo abbiamo ottenuto l'inizio di ogni evento.

### **Calcolo della fine degli eventi.**

Prima di mostrare il codice è opportuno segnalare la creazione di una matrice che conterrà gli indici di inizio e di fine di ogni evento:

```
tempi = [idx; zeros(1,length(idx))];
```

Nella prima riga avremo gli indici di ogni inizio evento mentre nella seconda tutti quelli della fine di ogni evento.

Possiamo ora analizzare la porzione di codice per la rilevazione della fine di ogni evento:

```
for j= 1: size(tempi,2)
    i = tempi(1,j)+1;
    if j == size(tempi,2)
        while i< length(a)
            if a(i)<= 0.01
                break
            end
            i=i+1;
        end
        tempi(2,j)=i-1;
    else
        while i<tempi(1,j+1)
            if a(i)<= 0.01
                break
            end
            i=i+1;
        end
        tempi(2,j)=i-1;
    end
end
```

Il codice sorgente è abbastanza semplice ed è suddiviso in strutture e costrutti ben definiti. Bisogna innanzitutto evidenziare che i casi gestiti saranno in particolare due: l'ultimo evento della melodia e tutti gli altri eventi presenti.

Iniziamo ad analizzare il blocco di codice utile per ricavare l'ultima nota della melodia:

```
while i < length(a)
    if a(i) <= 0.01
        break
    end
    i=i+1;
end
tempi(2,j)=i-1;
```

Il ciclo while funziona alla condizione che il contatore sia minore della lunghezza dell'involuppo del segnale. Se questa condizione viene verificata si può procedere ad analizzare l'involuppo all'istante  $i$ , se incontra un valore minore o uguale a 0.01 il ciclo si interrompe e a quell'evento viene attribuito il valore subito precedente allo 0.01 che sarà l'indice di fine evento dell'ultima nota. Il valore di 0.01 si è dimostrato migliore, rispetto allo 0, per permettere al prototipo di svolgere dei buoni calcoli anche in presenza di un leggero rumore di fondo causato dai diversi componenti elettrici durante la fase di registrazione.

In modo molto simile funziona il blocco di codice che permette di trovare la fine dell'evento di tutte le altre note:

```
while i < tempi(1,j+1)
    if a(i) <= 0.01
        break
    end
    i=i+1;
end
tempi(2,j)=i-1;
```

La differenza sostanziale sta nella porzione di segnale da analizzare, ovvero dal primo evento al penultimo evento. Per quanto riguarda la restante parte, il codice funziona esattamente come quello descritto precedentemente.

I due blocchi appena descritti sono all'interno di un altro ciclo:

```
for j= 1: size(tempi,2)
```

Questo ciclo for semplicemente indica che l'analisi viene effettuata solo se il contatore j è compreso tra il valore 1 e quello massimo della matrice tempi precedentemente descritta.

La sezione dedicata all'analisi temporale si può ritenere conclusa e ciò che abbiamo ottenuto è la seguente matrice contenente gli indici di inizio e di fine di ogni evento:

Init.	11	21	32
Fin.	20	31	72

Dalla seguente matrice si può notare che anche l'accuratezza di calcolo è buona. La melodia in esame è composta da tre note una di seguito all'altra ed è importante sottolineare come al concludersi un evento venga subito rilevato quello successivo.

### 4.2.3 Analisi spettrale

Una volta conclusa la sezione dedicata all'analisi temporale ci spostiamo in quella dedicata all'analisi frequenziale. In questo punto del programma il prototipo effettuerà una scansione delle varie frequenze di ogni nota e ne ricaverà la frequenza fondamentale.

Qui di seguito il codice relativo:

```
for cont= 1:size(tempi,2)
    lim_min = tempi(1,cont)*m;
    lim_max = tempi(2,cont)*m;
    sig = x(lim_min : lim_max);

    X = fft(sig);
    [Ymax, MaxIndex] = max(abs(X));
    ind1=MaxIndex(1);
    f =(0:length(sig)-1)/length(sig);
    phiStim=f(ind1);

    fStim(cont)=fs*phiStim;
    tStim(cont)=(lim_max-lim_min)/fs;
    tInit(cont) = lim_min/fs -1;
    tFin(cont) = lim_max/fs -1;
end
```

In sostanza, possiamo dividere questa analisi in 3 blocchi che andremo ad analizzare opportunamente.

Prima, però, è necessario sottolineare che anche questa analisi è contenuta nel seguente ciclo:

```
for cont= 1:size(tempi,2)
```

Come in quella temporale anche in questo caso si farà una analisi contenuta all'interno della matrice tempi.

### **Ritaglio della nota**

Prima di effettuare qualsiasi calcolo per rilevare la frequenza reale della nota, bisogna individuare prima la nota stessa. Essendo già in possesso degli indici di inizio e fine evento possiamo convertirli in campioni e ritagliare la porzione del segnale dato in ingresso:

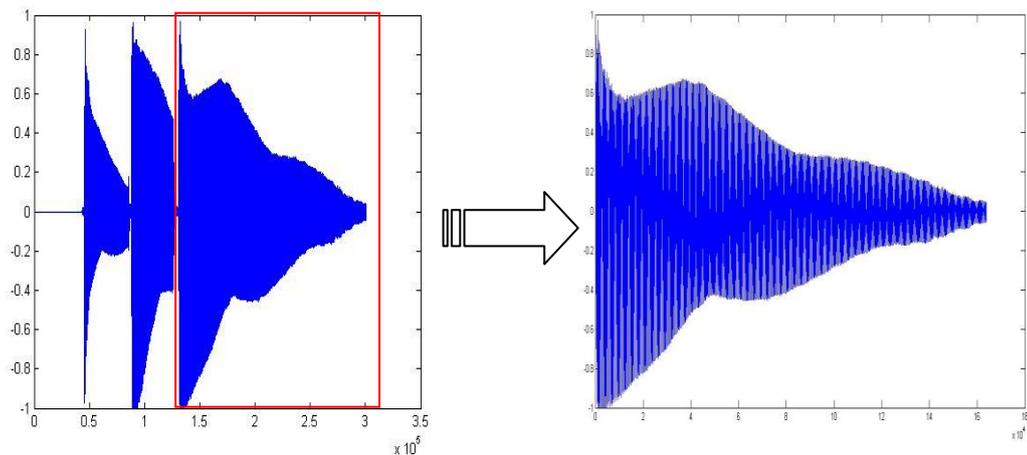
```
lim_min = tempi(1,cont)*m;  
lim_max = tempi(2,cont)*m;  
sig = x(lim_min : lim_max);
```

Semplicemente `lim_min` prende l'indice di inizio evento e lo moltiplica per la finestra `m` sempre del valore di 4096, in questo modo, ottengo il campione in quella posizione.

Lo stesso procedimento lo si effettua per `lim_max` considerando, però, l'indice di fine evento. Il contatore ovviamente ha il banale compito di spostarsi di posizione all'interno della matrice eventi ad ogni iterazione del ciclo.

Come ultimo passo si ritaglia la porzione di segnale che corrisponde quindi alla nota, questo lo possiamo ottenere semplicemente prendendo il range di campioni che va da `lim_min` a `lim_max`.

Dalla figura è possibile capire come avviene il ritaglio della singola nota.



### Individuazione della fondamentale

Il blocco di codice che verrà ora analizzato ha il compito di ricavare, per il momento, l'indice della frequenza fondamentale di ogni nota. Come precedentemente spiegato, il seguente prototipo utilizza solamente melodie di chitarra come input, quindi lo studio di questo strumento è fondamentale. Nel capitolo dedicato alla chitarra è stato dimostrato come la frequenza fondamentale è anche la frequenza con energia maggiore. Partendo da questa definizione è stato sviluppato il seguente blocco di codice:

```
X = fft(sig);
[Ymax, MaxIndex] = max(abs(X));
ind1=MaxIndex(1);
f=(0:length(sig)-1)/length(sig);
phiStim=f(ind1);
```

La prima riga di codice è il cuore di questo blocco: la trasformata di Fourier di ogni singola nota. In questo modo otteniamo una sommatoria di valori che descrivono le caratteristiche del segnale in ingresso.

Le restanti righe di codice si occupano di estrapolare l'indice della fondamentale e successivamente “entrare” nel vettore delle frequenze. Tutto questo genera phiStim utile per calcolare il valore reale della frequenza fondamentale delle note.

Nell'ultima parte dedicata all'analisi in frequenza si inizia a delineare delle informazioni utili per quello che sarà il risultato finale. Fin ora abbiamo ottenuto tanti valori e tante matrici, ma pochi risultati veramente significativi. Questo blocco ha il compito di riorganizzare tutto in funzione del tempo e delle frequenze:

```
fStim(cont)=fs*phiStim;
```

In questo modo otteniamo un vettore contenente le frequenze fondamentali di ogni nota:

Frequenze	162,69	145,35	129,2
-----------	--------	--------	-------

Per trovare, invece, la durata della nota `tStim`, l'inizio in secondi della nota `tInit` e la fine `tFin` utilizziamo rispettivamente le seguenti righe di codice:

```
tStim(cont) = (lim_max - lim_min) / fs;  
tInit(cont) = lim_min / fs - 1;  
tFin(cont) = lim_max / fs - 1;
```

Otterremo, anche in questo caso, tre vettori distinti contenente ognuno i propri valori. A questo punto anche l'analisi frequenziale è conclusa. Nel prossimo paragrafo ci si occuperà di riordinare i risultati dando così un senso vero e proprio al lavoro svolto dal prototipo.

#### 4.2.4 Spartito

Con questo paragrafo giungiamo alla conclusione del prototipo. Avendo ottenuto tutti i dati utili bisogna organizzarli in una unica matrice così che li renda veramente utili.

La creazione di questa matrice, chiamata *spartito*, che inizialmente è una matrice nulla del tipo `spartito={}`, avviene all'interno del seguente ciclo `for`:

```
for i = 1:length(fStim)
```

In questo modo, si prende in considerazione la matrice `fStim` e vengono considerati tutti i valori delle frequenze che compongono la melodia.

La gestione del vettore avviene in un modo molto semplice ovvero attribuendo ad ogni colonna tutte le informazioni necessarie di ogni nota, in modo tale che per ogni evento avremo una tabella contenente la frequenza della nota, il tempo di inizio, quello di fine e il nome della nota.

Qui di seguito il blocco di codice con il risultato che fornisce:

```
spartito{1,id} = fStim(i);  
spartito{2,id} = tInit(i);  
spartito{3,id} = tFin(i);
```

Frequenza	162,69	145,35	129,2
Tempo inizio	0.09	1,95	2,97
Tempo fine	1,86	2,88	6,69

In questo modo, però, manca quella riga contenente il nome della nota. Questo non è casuale, ma richiede una spiegazione a parte.

### Nome della nota

Per prima cosa è necessario dare un senso anche ai valori delle frequenze ottenute, ovvero convertirle in nome nota. Diversi sono i modi per ottenere questo tipo di risultato e quello proposto nel seguente elaborato può risultare il più ridondante, ma sicuramente anche quello più immediato per la gestione dei valori. Per evitare le continue ripetizioni del blocco di codice, verranno presentate solo alcune righe:

```

if (fStim(i)<84 && fStim(i)>81 ||fStim(i)<167 && fStim(i)>161
||fStim(i)<337 && fStim(i)>327 ||fStim(i)<680 && fStim(i)>650)
    spartito{4,id} = 'Mi';

    elseif (fStim(i)<89 && fStim(i)>85 ||fStim(i)<179 &&
fStim(i)>171 ||fStim(i)<357 && fStim(i)>340 ||fStim(i)<725 &&
fStim(i)>688)

        spartito{4,id}= 'Fa';

        elseif (fStim(i)<94 && fStim(i)>91 ||fStim(i)<188 &&
fStim(i)>182 ||fStim(i)<380 && fStim(i)>360 ||fStim(i)<766 &&
fStim(i)>730)

            spartito{4,id}= 'Fa#';
. . .
end

```

Nel seguente costrutto if/elseif sono presenti una serie di operazioni logiche and (&&) ed or ( || ) che hanno il compito di confrontare il valore corrente della frequenza con quelli presenti all'interno di ogni caso. Se il valore rientra in uno dei casi descritti, gli verrà attribuito il nome della nota.

I valori di comparazione sono relativi alle frequenze che partono dalla ottava II fino all'ottava V, range di frequenze che copre più o meno quello di una chitarra acustica (ventesimo tasto della prima corda). Per ragioni di sicurezza si è dovuto aggiungere, per ogni nota, un piccolo range di frequenze in quanto le frequenze ottenute

dall'elaborazione del segnale in ingresso non hanno un valore identico a quello teorico di una nota; per esempio per un LA 440Hz, si lavora all'interno di un range che va da 430 a 455 Hz assicurandosi così un buon risultato. I risultati ottenuti verranno aggiunti alla matrice spartito inserendo una nuova riga a quelle già esistenti.

### Gestione della pausa

Fin ora si è sempre parlato di frequenze e di note, ma come ben sappiamo in una melodia ci possono essere diverse pause. Nel prototipo in esame è stata scritta una parte di codice, che cerca di ovviare a questo problema.

Analizziamo il seguente codice:

```
if tInit(i+1)-tFin(i) > 0.1
    id = id + 1;
    spartito{1,id} = '0';
    spartito{2,id} = tFin(i) + 1/fs;
    spartito{3,id} = tInit(i+1) - 1/fs;
    spartito{4,id} = 'PAUSA';
end
```

Come per la conversione delle frequenze in nome nota, si utilizza un costrutto if. La condizione inserita nel costrutto risulta essere molto semplice, ma piuttosto efficace: si effettua una semplice differenza tra l'inizio dell'istante successivo e la fine dell'istante corrente. Se questo valore è superiore a una certa soglia (nel caso preso in esame 0,1) possiamo considerarlo una pausa tra le due note.

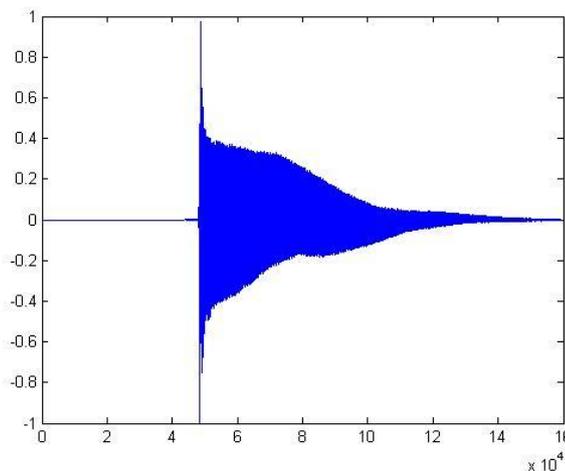
Una volta verificata questa condizione è necessario inserire anche questa informazione all'interno della matrice spartito attribuendo alla riga delle frequenze il valore 0 mentre in quella dedicata al nome della nota si inserisce " Pausa".Le le righe appartenenti al tempo di inizio e di fine dell'evento vengono trattate come nel caso di un evento corrispondente a una nota.

## 5. TEST SET

Una volta descritto, in modo dettagliato, il funzionamento dei diversi blocchi di codice del prototipo è necessario effettuare diverse prove per esaminare la buona riuscita del progetto. Il test set permette, quindi, di valutare i punti forti e deboli del programma, prendendo diversi file audio, come campioni in esame, e descriverne i risultati ottenuti. All'interno di questo capitolo saranno presi in considerazione dei campioni che daranno un'ottima risposta in output e altri che daranno dei risultati poco significativi.

### 5.1 Nota singola

Come in ogni test, si parte dal caso più semplice, ovvero, quella di analizzare una singola nota. La nota presa in esame è un FA a 175Hz.



**Figura 12** Grafico della nota presa in esame

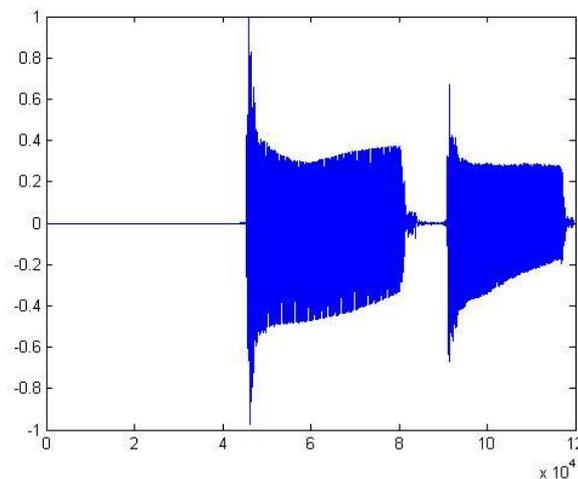
Di seguito viene mostrata una tabella di comparazione tra valori ottenuti e reali.

	<b>Valori ottenuti</b>	<b>Valori reali</b>
<b>Frequenza (Hz)</b>	174,058	175
<b>Tempo Inizio (s)</b>	0,0216	0
<b>Tempo Fine (s)</b>	2,251	2,468

Comparando i valori reali con quelli ottenuti si può dedurre che il risultato ottenuto è soddisfacente. La frequenza stimata è prossima a quella reale. Il risultato, ovviamente, non può essere uguale a quello reale perché, durante la registrazione, la presenza di rumore, causato dal microfono e altri componenti elettrici, si somma al segnale utile. Durante la fase di calcolo, il prototipo, non può distinguere questi due segnali, ma fornisce, comunque, un ottimo risultato. Nella riga relativa al tempo di inizio, il valore ottenuto è 0,0216. Ciò avviene perché, durante la fase di calcolo, il prototipo effettua diverse operazioni inserendo così un leggero ritardo che, comunque, è prossimo allo 0. L'ultimo risultato ottenuto è il tempo finale e può essere considerato un buon risultato. Rispetto al valore reale, il valore ottenuto termina leggermente prima, ma ciò è dovuto dalla soglia di fine evento impostata, descritta nel paragrafo 4.2.2, per risolvere le problematiche causate dal leggero rumore di fondo.

## 5.2 Due note

Nel secondo test effettuato, si analizzano due note consecutive a distanza di un tono nello specifico un DO a 131 Hz e un RE a 147 Hz. Questa prova è fondamentale perché ha il compito di separare i due eventi e calcolarne i valori.



**Figura 13** Grafico delle due note analizzate

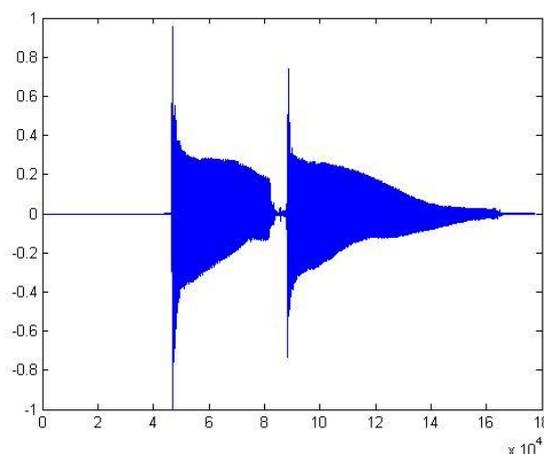
Di seguito è riportata la tabella di comparazione tra i risultati ottenuti e quelli reali.

	<b>DO ottenuto</b>	<b>RE ottenuto</b>	<b>DO reale</b>	<b>RE reale</b>
<b>f (Hz)</b>	130,272	147,137	131	147
<b>Tempo Inizio (s)</b>	0,021	1,043	0	1.047
<b>Tempo Fine (s)</b>	0,950	1,601	0.992	1.714

Nonostante la maggior complessità del segnale in ingresso, rispetto al test precedente, si ottengono dei risultati veramente positivi. Per quanto riguarda le frequenze, in entrambi i casi, la differenza tra valori ottenuti e reali è pressoché minima. Risultati confortanti sono anche quelli relativi al tempo di inizio e di fine di entrambi gli eventi. La difficoltà, in questo test, consisteva proprio nell'individuare e calcolare il valore di inizio e fine di ogni evento. Il prototipo ha risposto in modo eccellente a questa prova.

### 5.3 Semitoni

Dopo aver testato il comportamento del prototipo, avendo dato in ingresso due note con distanza pari a un tono, è opportuno eseguire un test molto simile, ma con un file audio contenente due note con distanza pari a un semitono. Le note musicali presenti nel file audio sono un MI a 165 Hz e un FA a 175 Hz.



**Figura 14** Grafico dei due semitoni analizzati

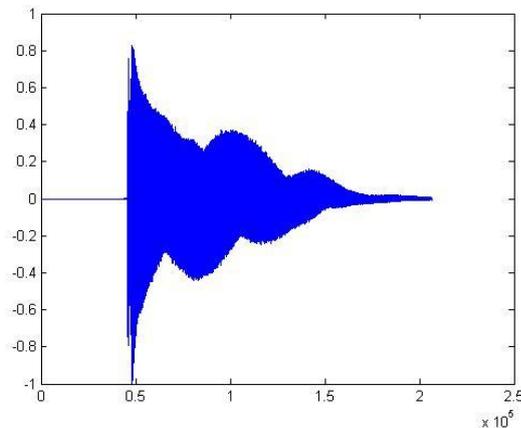
La seguente tabella mostra i risultati calcolati dal prototipo comparati con i risultati reali.

	<b>MI ottenuto</b>	<b>FA ottenuto</b>	<b>MI reale</b>	<b>FA reale</b>
<b>f (Hz)</b>	163,88	173,87	165	175
<b>Tempo Inizio (s)</b>	0,0216	0,9504	0	0,99
<b>Tempo Fine (s)</b>	0,857	2,808	0,93	3

La difficoltà di calcolo di questo test consiste nel rilevare la frequenza di due note molto vicine tra loro. Dalla tabella è possibile notare che la differenza tra la prima nota e la seconda è solamente di 10Hz. Nonostante ciò, il prototipo è stato in grado di estrarre la frequenza fondamentale di entrambe le note che risulta essere molto vicina a quella reale. Buoni possono essere considerati i valori ricavati per il tempo di inizio dell'evento e quello di fine anche se, nel caso della trascrizione, la prima nota termina leggermente in anticipo causando, così, un inizio anticipato delle seconda.

## 5.4 Corda a vuoto

Tipico degli strumenti a corde, in particolare della chitarra, è il suono di una corda a vuoto. Rispetto alle note ottenute premendo i tasti dello strumento, la corda vuota ha la caratteristica di avere un contenuto armonico ricco. Il test che si andrà a descrivere ha il compito di stabilire se il prototipo è in grado di estrarre la fondamentale nonostante le molte informazioni che la nota contiene. La nota presa in esame è la quarta corda della chitarra, ovvero un RE con frequenza pari a 147 Hz.



**Figura 15** Grafico del suono di una corda a vuoto

Il grafico mostra come la nota presa in esame sia ricca di informazioni. Questo, lo si può notare dall'andamento ondulato del segnale e dal lento decadimento della nota che sono elementi caratteristici di una corda suonata a vuoto.

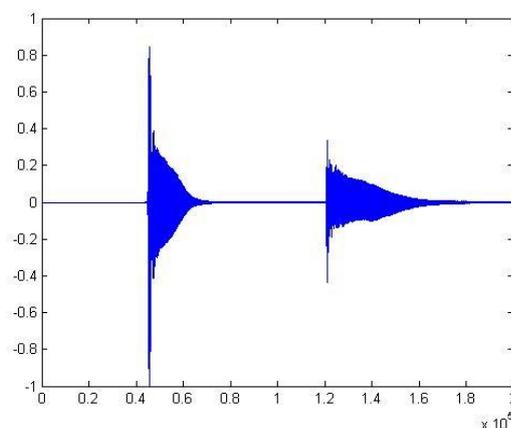
Di seguito viene mostrata la tabella contenente i risultati ottenuti e quelli reali.

	Valori ottenuti	Valori reali
<b>Frequenza (Hz)</b>	145,631	147
<b>Tempo Inizio (s)</b>	0,0216	0
<b>Tempo Fine (s)</b>	3,551	3,685

Anche in questo caso si può affermare che il prototipo in esame fornisce dei risultati positivi. In questo test il valore ottenuto della frequenza è di circa 1,5 Hz inferiore rispetto a quello reale. Il prototipo deve analizzare un contenuto ricco di informazioni per l'estrazione della fondamentale, questo perché l'attacco della nota è ricco di armoniche. Per quanto riguarda il calcolo del tempo di inizio e di fine della nota si può notare come i risultati ottenuti siano prossimi a quelli reali. Il test appena descritto può essere considerato positivo.

### 5.5 Nota/pausa/nota

Spesso, all'interno di una melodia, è facile trovare delle pause. Il prototipo è in grado di rilevare la pausa non tanto come evento, ma come un intervallo di tempo, tra la fine di un evento e l'inizio di quello successivo, nel quale la presenza di segnale utile è nulla.



**Figura 16** Grafico nota-pausa-nota

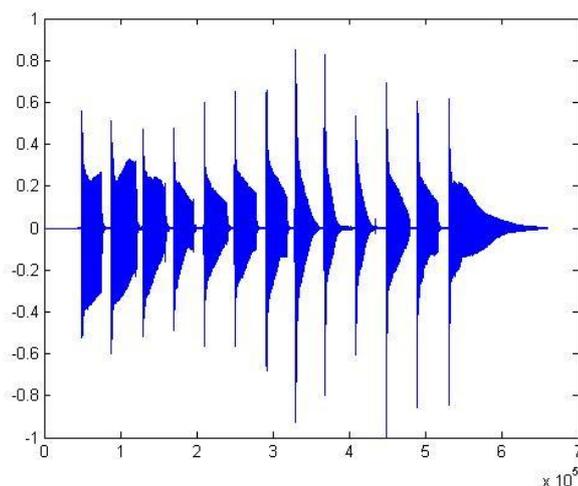
Questo test ha il compito di mostrare come il prototipo gestisce la presenza di una pausa tra due note e che tipo di risultato fornisce in output. Dal grafico è semplice distinguere le due note che, nel caso in esame, sono un LA a 220Hz e un SI a 247Hz. Nella seguente tabella verranno riportati i risultati ottenuti e spiegata, in particolare, la colonna dedicata alla pausa.

	<b>LA</b>	<b>PAUSA</b>	<b>SI</b>
<b>Frequenza (Hz)</b>	220,706	0	246,799
<b>Tempo Inizio (s)</b>	0,021	0,578	1,693
<b>Tempo Fine (s)</b>	0,578	1,693	2,901

Dalla tabella è possibile notare come la gestione della pausa sia molto accurata. In giallo sono evidenziati il secondo di fine della nota LA e quello di inizio della pausa. È sorprendente come i due valori siano identici. Questo indica come il prototipo reagisca davanti all'assenza di segnale e ad un intervallo di tempo superiore alla soglia impostata definendolo come pausa tra le due note. Infatti, è possibile notare (evidenziato in azzurro) come la fine della pausa corrisponde all'inizio della nuova nota. La rilevazione e la gestione della pausa può considerarsi un ottimo risultato.

## 5.6 Scala cromatica

Dopo aver analizzato dei casi semplici, ma fondamentali per la riuscita del prototipo, è molto importante studiare i risultati che si ottengono quando in input viene dato un file audio contenente diverse note musicali. Il test eseguito è una trascrizione della scala cromatica partendo da un DO a 131 Hz fino a raggiungere, per semitono, un DO a 262 Hz per un totale di 13 note.



**Figura 17** Grafico di una scala cromatica

Osservando il grafico è possibile notare come i tredici eventi siano ben distinti. Va precisato che, durante la fase di registrazione, l'esecutore ha suonato la scala cromatica in modo da scandire bene ogni singola nota, evitando di sovrapporre due note tra loro.

La seguente tabella mostra i risultati ottenuti.

	<b>Do</b>	<b>Do#</b>	<b>Re</b>	<b>Re#</b>	<b>Mi</b>	<b>Fa</b>
<b>Frequenza (Hz)</b>	130,3	137,5	147,1	154,7	164,1	173,4
<b>Tempo Inizio (s)</b>	0,021	0,95	1,87	2,81	3,73	4,57
<b>Tempo Fine (s)</b>	0,857	1,78	2,71	3,55	4,47	5,40

	<b>Fa#</b>	<b>Sol</b>	<b>Sol#</b>	<b>La</b>	<b>La#</b>	<b>Si</b>	<b>Do</b>
<b>Frequenza (Hz)</b>	184,2	195,1	207,6	219,9	233,2	246,2	261,2
<b>Tempo Inizio (s)</b>	5,50	6,43	7,26	8,19	9,12	10,1	10,9
<b>Tempo Fine (s)</b>	6,33	7,17	7,91	8,84	9,95	10,79	13,39

Dalla tabella si può notare come il prototipo sia riuscito a separare tutti gli eventi della scala cromatica. Le difficoltà a cui andava incontro sono sostanzialmente di due tipi: la prima riguarda la presenza di molte note in pochi secondi; la seconda, invece, riguarda la distanza di semitono tra una nota e l'altra, non sempre facile da rilevare. Il prototipo, però, ha gestito il test in maniera più che sufficiente riconoscendo tutti gli eventi del

caso, attribuendo, ad ognuno di questi, un valore di frequenza molto simile a quello reale. Per quanto riguarda il calcolo dell'inizio e della fine di ogni nota si ottengono, ancora una volta, valori vicini a quelli reali. La durata dell'intero file ha un valore reale di 13,94 secondi mentre quella calcolata dal prototipo è di 13,39 secondi. La prova effettuata ha riscontrato degli ottimi risultati.

## 5.7 Scala maggiore

Un altro caso molto importante che richiede di essere analizzato è una delle più semplici, ma anche una delle più utilizzate, scale musicali: la scala maggiore. Il file audio preso in esame contiene una scala di Do maggiore con un range di frequenze che va da 131 a 262 Hz.

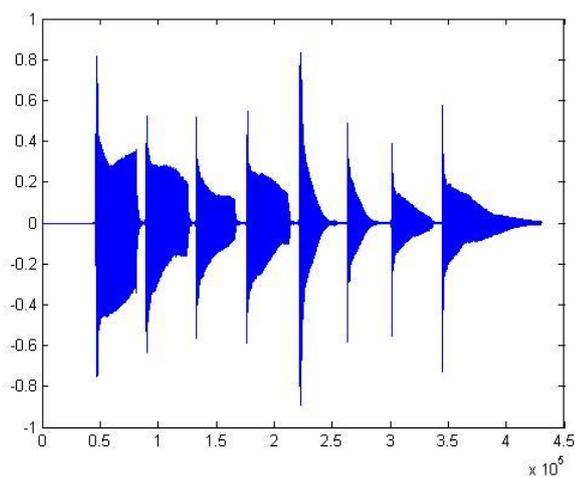


Figura 18 Grafico di una scala maggiore

L'approccio a questa analisi è molto simile a quello effettuato nel precedente test. Facilmente sono riconoscibili gli eventi che costituiscono il file audio esaminato, caratterizzato dalla sequenza di note tipica della scala maggiore ascendente.

I risultati ottenuti sono contenuti all'interno della tabella.

	<b>Do</b>	<b>Re</b>	<b>Mi</b>	<b>Fa</b>	<b>Sol</b>	<b>La</b>	<b>Si</b>	<b>Do</b>
<b>f (Hz)</b>	130,3	146,4	164,7	173,3	195,1	220,7	246,5	261,6
<b>t Inizio</b>	0,021	0,95	1,972	2,99	4,015	4,944	5,78	6,801
<b>t Fine</b>	0,85	1,87	2,91	3,92	4,75	5,501	6,709	8,65

La trascrizione della scala di Do maggiore è avvenuta in modo soddisfacente. Le otto note presenti nel file audio sono state ben riconosciute dal prototipo e separate opportunamente, attribuendo i giusti valori di frequenza e di inizio e fine evento prossimi a quelli reali. La durata reale del file audio è di 8,75 secondi mentre quella calcolata dal programma è di 8,65 secondi. Anche in questo caso il test può considerarsi valido.

## 5.8 Bicordo

Il test che verrà descritto in questo paragrafo è molto particolare. Fino a questo punto dell'elaborato sono stati trattati solamente file audio contenenti note singole in successione e mai casi di due o più note eseguite contemporaneamente. Nonostante il prototipo non è stato progettato per questi ultime situazioni è curioso esaminare il risultati che vengono dati in output. Il bicordo preso in analisi è composto da un LA a 220 Hz e un RE a 294 Hz.

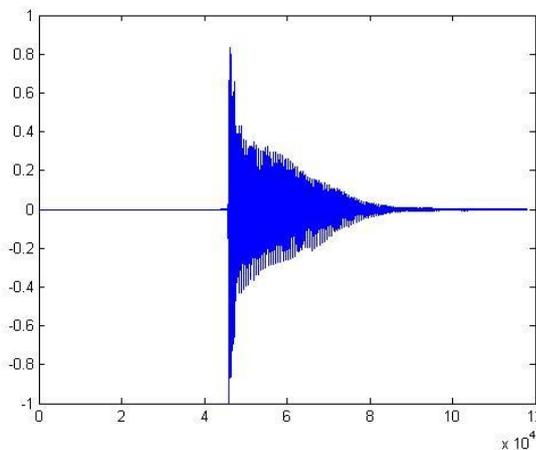


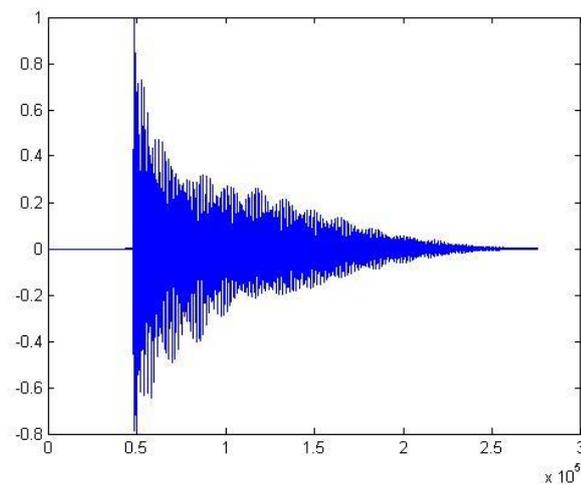
Figura 19 Grafico del bicordo preso in analisi

Caratteristica del bicordo, rispetto ad una nota singola, è la grande quantità di componenti armoniche, delle due note, che si sommano tra di loro. Questo rende molto complicata l'individuazione delle note. Il prototipo vede il bicordo come un singolo evento e, nonostante la sua complessità, si riesce a delineare una durata abbastanza simile a quella reale. Per quanto riguarda, invece, il risultato ottenuto nella colonna

della frequenza è curioso come, durante la fase di calcolo, venga rilevata una delle due note che costituiscono il bicordo. In questo caso, la nota rilevata è quella più grave ovvero il LA e la frequenza che viene calcolata è pari a 220,71 Hz, valore quasi identico a quello reale. Il test appena descritto non può considerarsi sufficiente, ma rilascia sicuramente dei risultati curiosi che hanno un significato importante legato al bicordo in ingresso.

## 5.9 Accordo

Dopo aver analizzato il risultato ottenuto dall'analisi di un bicordo, si analizza il comportamento del prototipo quando in ingresso riceve un accordo. Sulla base del test precedente è sicuramente curioso scoprire ciò che viene calcolato e dato in uscita. L'accordo in questione è un SOL maggiore in posizione aperta composto, quindi, dalle seguenti note: sol, si, re, sol, si, sol.



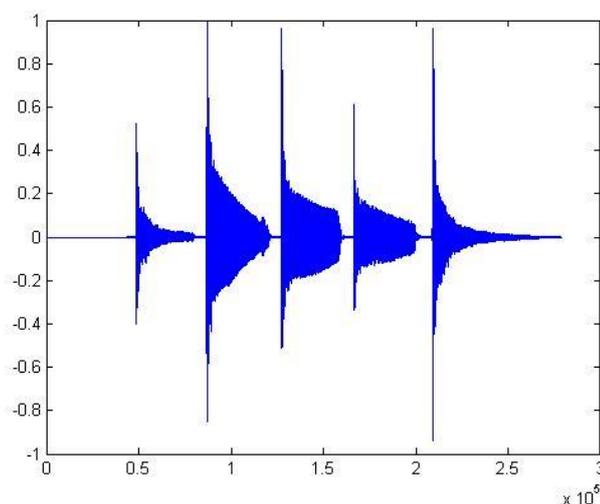
**Figura 20** Grafico dell'accordo analizzato

Esaminando il grafico è molto semplice notare come il segnale sia ricco di informazioni. In primo luogo bisogna sottolineare che in questo test le note suonate contemporaneamente sono ben sei, di cui alcune sono corde a vuoto. Per l'analisi temporale otteniamo un risultato abbastanza significativo, in quanto la durata che il prototipo calcola è di 4,479 secondi mentre quella reale è di 5,2 secondi. Nessun valore, invece, viene espresso nella colonna dedicata alle frequenze proprio perché il prototipo

non è in grado di rilevare la nota fondamentale dell'accordo, essendo costituito da diverse note suonate contemporaneamente.

## 5.10 Utilizzo delle dita per pizzicare le corde

Se tutte le prove registrate sono suonate, dall'esecutore, con l'utilizzo del plettro, per pizzicare le corde, è molto importante dedicare almeno un test con una melodia suonata con le dita. Questa tecnica è molto utilizzata dai chitarristi, non solo classici, ma anche moderni, per creare alcuni effetti timbrici leggermente diversi da quelli ottenuti utilizzando il plettro. Come descritto nei primi capitoli il timbro è caratterizzato dalla composizione armonica dello strumento.



**Figura 21** Grafico della melodia suonata con le dita

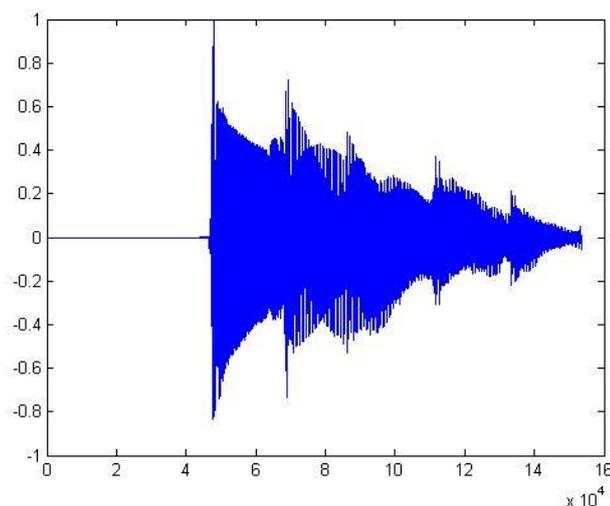
Il file audio è costituito da 5 note in successione. Il prototipo, però, non riscontra molte difficoltà nella rilevazione della frequenza di ogni nota e della durata di ognuna di queste. La seguente tabella mostra i risultati ottenuti.

	<b>Re</b>	<b>La</b>	<b>Si</b>	<b>Do</b>	<b>Re</b>
<b>f (Hz)</b>	294,27	220,11	247,62	261,62	294,28
<b>t Inizio (s)</b>	0,021	0,95	1,87	2,71	3,73
<b>t Fine (s)</b>	0,85	1,78	2,62	3,64	4,85

Nonostante il timbro sia leggermente diverso, rispetto agli altri file audio suonati con il plettro, il programma è in grado di ricavare la fondamentale, ciò significa che anche con l'utilizzo delle dita, per pizzicare le corde, le componenti frequenziali utili per ricavare la frequenza della nota sono presenti. La gestione temporale, invece, avviene come in tutti gli altri casi analizzati ottenendo degli ottimi risultati.

## 5.11 Arpeggio

Altro caso piuttosto interessante da studiare è il risultato ottenuto dall'analisi di un arpeggio. L'arpeggio è una tecnica musicale applicata ad un accordo, in cui le note musicali sono suonate in successione e mantenute fino alla fine dell'arpeggio. Il file audio esaminato è composto da un arpeggio dell'accordo di RE maggiore.



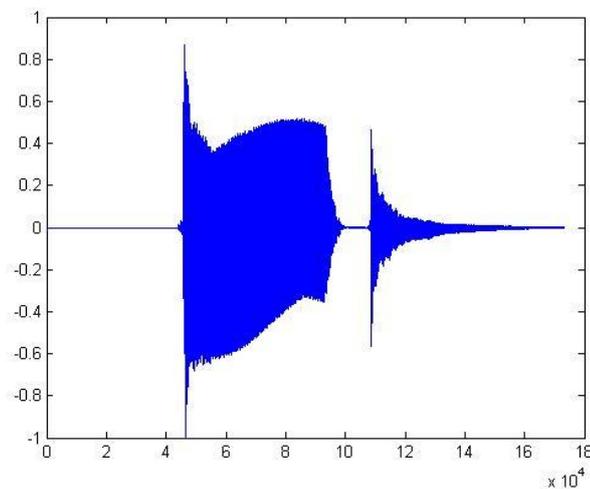
**Figura 22** Grafico dell'arpeggio esaminato

Il grafico che si ottiene dall'analisi di un arpeggio è molto simile a quello di un accordo. Infatti, accordo e arpeggio sono legati tra di loro, in quanto quest'ultimo è un abbellimento dell'accordo. Le informazioni contenute in un arpeggio sono elevate e, quindi, per il prototipo risulta molto difficile distinguere le varie note presenti nel file audio. Altro problema che si riscontra è l'analisi temporale di ogni singola nota dell'arpeggio proprio perché i diversi eventi non sono ben scanditi rendendo, così, impossibile il calcolo dell'inizio e di fine nota. Unica nota positiva, di questo test, è che

il programma, in qualche modo, riesce a rilevare la prima nota suonata, ovvero un RE con frequenza pari a 147 Hz.

## 5.12 Intervallo composto

In questo paragrafo verrà analizzato un intervallo composto. L'intervallo composto è un intervallo che supera l'estensione di un'ottava. Il file audio in esame è costituito da un DO a 131 Hz e un RE con frequenza pari a 294 Hz, ottenendo un intervallo di 9<sup>a</sup> maggiore. È molto curioso, quindi, studiare i risultati che il prototipo calcola quando vengono date in input due note con una netta differenza di frequenza.



**Figura 23** Grafico di un intervallo composto

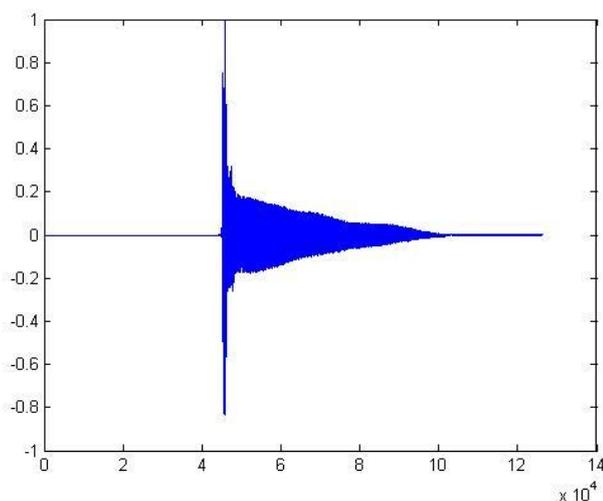
In questa prova il programma è in grado, senza problemi, di calcolare le frequenze fondamentali di entrambe le note e di rilevare il tempo di inizio e di fine di entrambi gli eventi. La seguente tabella riporta i risultati ottenuti comparandoli a quelli reali.

	<b>DO ottenuto</b>	<b>RE ottenuto</b>	<b>DO reale</b>	<b>RE reale</b>
<b>f (Hz)</b>	130,73	294,28	131	294
<b>Tempo Inizio</b>	0,0217	1,415	0	1,423
<b>Tempo Fine</b>	1,322	2,529	1,319	2,929

Si può notare come i valori delle frequenze siano ottimi e vicini a quelli reali. Anche nel caso dell'analisi temporale il prototipo ha risposto positivamente al test.

### 5.13 Armonico naturale

Gli armonici naturali sono una caratteristica degli strumenti a corde come la chitarra. Vengono utilizzati dai chitarristi per produrre suoni particolare per creare abbellimenti. L'armonico analizzato, in questo paragrafo, è uno dei più utilizzati e si ottiene sfiorando con il polpastrello, della mano sinistra, la corda all'altezza del XII tasto mentre con la mano destra si pizzica la corda. Nel file audio la corda pizzicata sarà la quinta, ovvero il LA.



**Figura 24** Grafico dell'armonico naturale analizzato

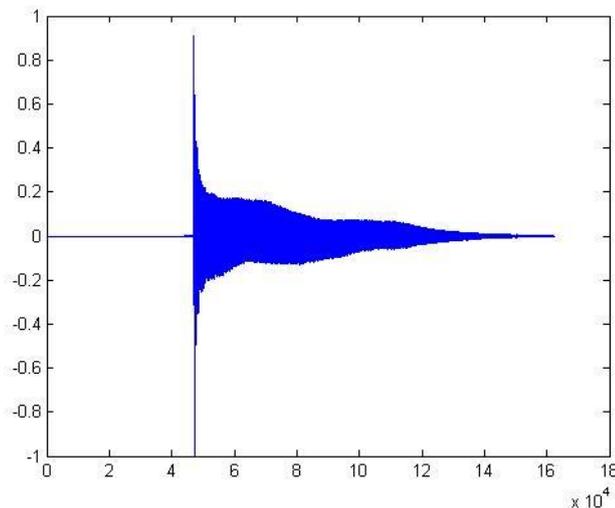
Dal grafico si può notare come l'attacco dell'armonico sia in evidenza e ricco di energia mentre il decadimento è molto veloce con poca energia. Questo, però, è sufficiente per calcolare la frequenza fondamentale mentre rende l'analisi temporale leggermente imprecisa. Di seguito la tabella con i risultati.

	<b>Armonico LA ottenuto</b>	<b>Armonico LA reale</b>
<b>Frequenza (Hz)</b>	219,942	220
<b>Tempo Inizio (s)</b>	0,0217	0
<b>Tempo Fine (s)</b>	1,322	1,862

Analizzando questa tabella si può concludere che il test effettuato è più che sufficiente in quanto risponde in modo adeguato alle richieste iniziali.

## 5.14 Bending

Il bending è una tecnica molto utilizzata dai chitarristi. La tecnica del bending consiste nel suonare una nota, che non sia una corda a vuoto, e successivamente spingere la corda verso l'alto (o verso il basso) in direzione perpendicolare al manico, provocando una maggiore tensione e, quindi, l'aumento dell'altezza della nota. L'intervallo che solitamente si ottiene tra la nota di partenza e quella di arrivo è compreso tra il semitono e i due toni. La prova effettuata in questo paragrafo riguarda un bending di semitono, composto da un DO a 262 Hz che raggiunge il DO# a 277 Hz per poi ritornare al DO iniziale.

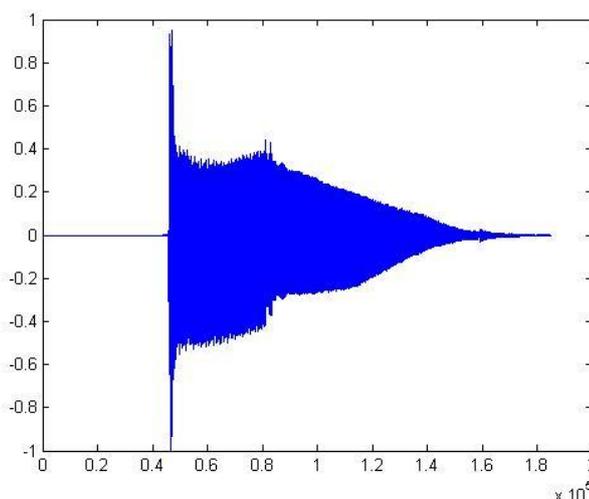


**Figura 25** Grafico del bending analizzato

Dal grafico si può notare come il segnale sia povero di informazioni, in quanto, oltre all'attacco iniziale, il decadimento della nota risulta avere poca energia. Infatti, il risultato che il prototipo dà in uscita è semplicemente quello della prima nota suonata, ovvero il Do, mentre del bending non vi è traccia. Il prototipo, quindi, individua solamente un evento all'interno del file audio proprio perché non riesce a rilevare altri attacchi oltre la prima nota. La durata del segnale calcolata, invece, ha un valore molto vicino a quello reale. Il test non può ritenersi sufficiente, in quanto il prototipo non è in grado di rilevare tutti gli eventi del caso in analisi.

## 5.15 Slide

Lo slide è un movimento di scivolamento di un dito, della mano sinistra, lungo una corda su uno o più tasti consecutivi. L'effetto che si ottiene sulla chitarra è un veloce passaggio cromatico dalla nota di partenza fino alla nota di arrivo. Nel file audio preso in esame troviamo la nota di partenza che è un DO a 131 Hz fino a raggiungere, tramite slide, un RE a 147 Hz, passando, quindi, per due semitoni.



**Figura 26** Grafico di due note suonate con l'effetto slide

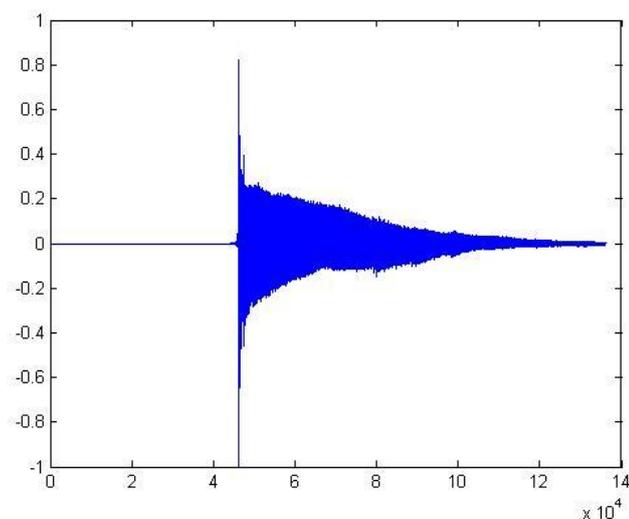
Già dal grafico si può notare come l'attacco della seconda nota manchi del tutto. Infatti, il risultato che il prototipo fornisce è quello di una sola nota, ovvero la nota iniziale. Mancando la parte fondamentale per ricavare la durata e la frequenza delle seconda nota, il prototipo prende in considerazione il file audio come un unico evento, trascrivendolo nel seguente modo:

<b>Nome nota</b>	Do
<b>Frequenza (Hz)</b>	130,99
<b>Tempo Inizio</b>	0,0216
<b>Tempo fine</b>	2,808

L'effetto slide è un caso molto difficile da studiare e il prototipo non risponde adeguatamente alle richieste del test.

## 5.16 Vibrato

Il vibrato è un ulteriore effetto molto utilizzato all'interno di esecuzioni musicali. La sua tecnica prevede di far oscillare il dito, della mano sinistra, sulla corda ripetutamente, ottenendo un effetto di vibrazione periodico del suono. Nell'analisi del file audio la nota alla quale viene applicato l'effetto vibrato è un Do a 262 Hz.



**Figura 27** Grafico della nota analizzata con effetto vibrato

Si può facilmente notare come l'attacco della nota sia ben definito mentre il decadimento è abbastanza lento, quindi utile per calcolare in modo dettagliato la fine della nota. Durante la registrazione, al segnale utile è stato aggiunto del leggero rumore causato dallo sfregamento della corda sul manico proprio per ottenere l'effetto vibrato. Nonostante ciò, il prototipo ha risposto egregiamente calcolando i seguenti valori:

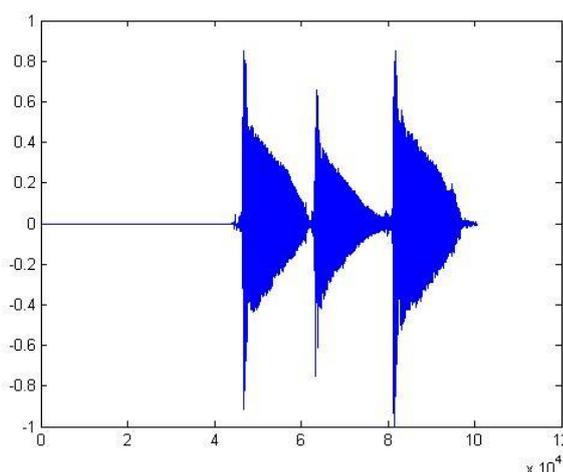
	<b>Valori ottenuti</b>	<b>Valori reali</b>
<b>Frequenza (Hz)</b>	262,49	262
<b>Tempo Inizio (s)</b>	0,021	0
<b>Tempo Fine (s)</b>	1,972	2,08

Comparando i valori ottenuti con quelli reali si nota come la fase di calcolo del prototipo sia stata molto accurata sia nella rilevazione della frequenza fondamentale, sia

nell'analisi temporale individuando egregiamente il valore di inizio e di fine della nota. Il risultati ottenuti da questo testi possono essere considerati ottimi.

## 5.17 Terzina

La terzina è un gruppo irregolare formato da tre note, di uguale durata, che occupano lo spazio ritmico di due note dello stesso valore. Uno degli esempi più comuni è la terzina di ottavo, ovvero l'esecuzione di tre note all'interno di un quarto. Essendo un gruppo irregolare che spesso viene riscontrato è giusto dedicare un paragrafo sull'analisi di questo caso. La terzina in analisi è costituita da tre ottavi della stessa frequenza ovvero un LA a 220 Hz mentre la velocità di esecuzione è piuttosto lenta, circa 60 bpm, proprio per permettere al prototipo di individuare i tre eventi.



**Figura 28** Grafico della terzina presa in analisi

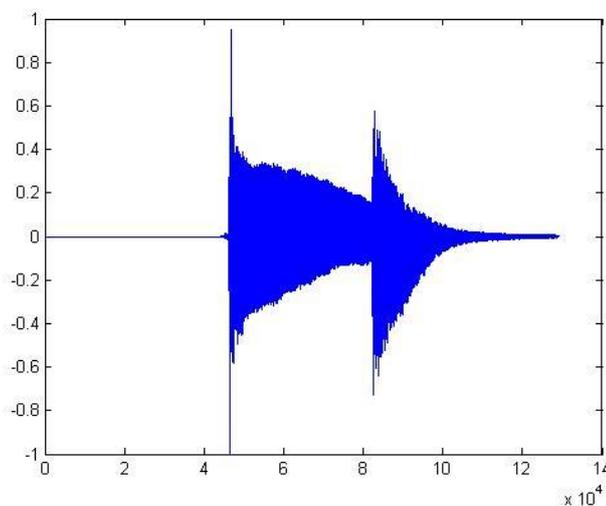
Il grafico mostra i tre eventi in successione ben definiti. Il prototipo, ovviamente, non tiene in considerazione che il file audio è costituito da un gruppo irregolare di note, ma le analizza come tre eventi ben distinti con un inizio e una fine precisa. Di seguito la tabella con i risultati ottenuti.

	<b>La</b>	<b>La</b>	<b>La</b>
<b>Frequenza (Hz)</b>	222,49	222,49	220,70
<b>Tempo inizio</b>	0,0216	0,393	0,764
<b>Tempo fine</b>	0,3003	0,671	1,136

Il calcolo effettuato dal prototipo può considerarsi sufficiente, in quanto le frequenze ottenute sono molto vicine al valore reale, così per l'analisi temporale anche se la durata di ogni nota è leggermente inferiore rispetto a quella reale, questo è dovuto dal fatto che i tre eventi sono in successione ed eseguiti, comunque, molto più velocemente rispetto a due ottavi all'interno dello stesso spazio ritmico. In conclusione si può affermare che anche questa prova è più che positiva.

### 5.18 Hammer-on

L'hammer-on, chiamato anche legatura ascendente, è una tecnica molto utilizzata dai chitarristi. Questa tecnica permette di passare da una nota più grave a una più acuta senza pizzicare, con la mano destra, la corda, ma premendo con il dito, della mano sinistra, la nota che si desidera suonare. La prima nota emessa, però, verrà pizzicata con la mano destra. Si ottiene, quindi, un effetto di legatura tra le due note. Il file audio in esame comprende due note: la prima è un Fa a 175 Hz e un Sol a 196 Hz.



**Figura 29** Grafico di due note suonate con la tecnica hammer-on

Il grafico realizzato dal prototipo mostra come le due note siano legate tra di loro, in quanto i due eventi non sono staccati. Infatti, la seconda non inizia subito dopo il termine della prima, ma leggermente in anticipo creando così l'effetto di legatura ascendente. Nonostante ciò, il prototipo è in grado di calcolare e distinguere i due eventi, ma è giusto sottolineare che, per ottenere dei risultati positivi, l'esecuzione delle

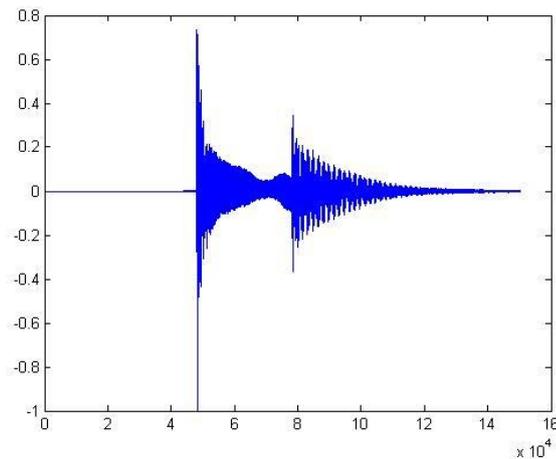
due note, con questa tecnica, deve essere accurata, proprio perché, in fase di registrazione, l'energia della seconda nota sarà minore rispetto la prima. Di seguito la tabella con l'analisi ottenuta.

	<b>Fa</b>	<b>Sol</b>
<b>Frequenza (Hz)</b>	173,606	195,947
<b>Tempo Inizio (s)</b>	0,0216	0,857
<b>Tempo Fine (s)</b>	0,764	1,786

La complessità di calcolo in questo test è piuttosto complicata, ma i risultati ottenuti possono considerarsi soddisfacenti, in quanto le frequenze ottenute hanno valori riconducibili a quelli reale. Per quanto riguarda il calcolo della durata delle singole note è giusto precisare che il termine della prima nota è leggermente anticipato rispetto a quello reale che ha un valore pari a 0,857 secondi. Per l'analisi temporale della seconda nota è sorprendente come il tempo di inizio calcolato sia identico a quello reale mentre per il tempo di fine evento otteniamo un valore leggermente inferiore a quello reale.

### **5.19 Pull-off**

Il pull off, chiamato anche legatura discendente, è una tecnica chitarristica molto utilizzata. Questa tecnica permette di passare da una nota più acuta a una nota più grave, della stessa corda, senza l'utilizzo della mano destra che pizzica la corda. Il pull off consiste nel suonare la prima nota premendo già la nota successiva, più bassa, e poi "strappare" col dito la nota che abbiamo suonato. La prima nota emessa, però, verrà pizzicata con la mano destra. Si ottiene, quindi, un effetto di legatura tra le due note. Il file audio analizzato è costituito da due note, rispettivamente un SI con frequenza pari a 247 Hz e un LA a 220Hz.



**Figura 30** Grafico delle note analizzate suonate con la tecnica pull off

Dal grafico è possibile distinguere i due eventi, ma è giusto sottolineare come il secondo evento abbia una energia inferiore rispetto al primo. Questo è una caratteristica della tecnica utilizzata proprio perché la seconda nota non viene pizzicata. L'executore, durante la fase di registrazione, è stato attento a dar la giusta energia alla seconda nota, per permettere al prototipo di rilevarla come un nuovo evento. Di seguito la tabella con i risultati ottenuti.

	<b>Si</b>	<b>La</b>
<b>Frequenza (Hz)</b>	247,62	220,29
<b>Tempo Inizio (s)</b>	0,0216	0,7647
<b>Tempo Fine (s)</b>	0,6718	1,9721

Nonostante il file audio da analizzare fosse piuttosto articolato, il prototipo ha risposto positivamente, calcolando le frequenze fondamentali con valori prossimi a quelli reali. Per l'analisi temporale otteniamo, anche in questo caso, ottimi risultati in quanto il tempo di fine reale del primo evento è pari 0,687 secondi mentre quello calcolato, ed espresso in tabella, è di 0,6718 secondi. Per il calcolo temporale del secondo evento otteniamo buoni risultati anche se leggermente inferiori, causati dalla minore presenza di segnale. I valori reali di inizio e fine del secondo evento sono rispettivamente 0,691 e 02,327 secondi. Nonostante ciò, si può concludere affermando che il test eseguito su questo file audio è superato.

## 5.20 Semplice melodia

Una volta effettuati diversi test, analizzando i punti forti e deboli del prototipo, è giunto il momento di dare input una vera e propria melodia. Fin ora, si sono effettuate prove contenenti poche note, utili, però, per visualizzare il comportamento e i risultati calcolati dal programma. In questo test viene data in input un semplice melodia che racchiude, però, diverse altezze delle note e diverse durate e si analizza la trascrizione che il prototipo fornisce in uscita. La melodia in esame è composta dal tema principale di “jingle bells” per un totale di otto battute e ben ventidue note.



Figura 31 Spartito della melodia analizzata

Dalla partitura si può notare la semplicità della melodia, ma, allo stesso tempo, la ricchezza di durate diverse delle note e di intervalli che vanno dal semitono a quello di terza minore. La velocità di esecuzione, invece, è tipica dell'andante in quanto la melodia è intorno agli 85 bpm.

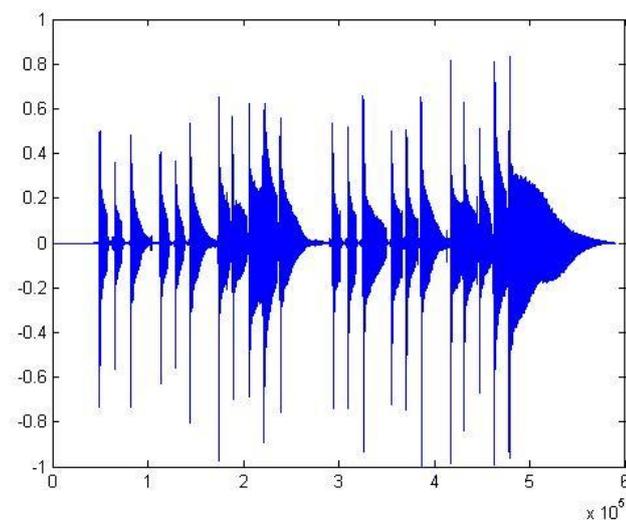


Figura 32 Grafico della melodia analizzata

Dal grafico è possibile notare come gli eventi siano ben evidenziati e distinti tra di loro. Va precisato che durante la fase di registrazione, l'esecutore si è impegnato a suddividere ogni singolo evento evitando di sovrapporre le note tra di loro, senza modificare, però, la natura della melodia. Nel file audio registrato è possibile ascoltare qualche leggero rumore causato dal plettro che pizzica la corda, ma, come vedremo, questo inconveniente non intaccherà la trascrizione finale. Di seguito è riportata la tabella con la trascrizione automatica del tema musicale di "jingle bells".

<b>nota</b>	<b>La</b>	<b>Do</b>	<b>Fa</b>	<b>Sol</b>	<b>La</b>						
<b>f</b> <b>(Hz)</b>	220,6	218,9	219,9	218,9	218,9	220,7	218,9	261,9	175,8	197,3	219,6
<b>t in</b> <b>(s)</b>	0,114	0,393	0,764	1,507	1,879	2,250	2,901	3,272	3,643	4,015	4,387
<b>Tfin</b> <b>(s)</b>	0,301	0,671	1,414	1,786	2,157	2,808	3,179	3,551	3,922	4,294	5,315



<b>nota</b>	<b>La#</b>	<b>La#</b>	<b>La#</b>	<b>La#</b>	<b>La</b>	<b>La</b>	<b>Do</b>	<b>La#</b>	<b>La</b>	<b>Sol</b>	<b>Fa</b>
<b>f</b> <b>(Hz)</b>	233,2	233,2	233,2	233,2	218,9	220,7	261,9	233,2	220,6	197,3	173,9
<b>t in</b> <b>(s)</b>	5,594	5,965	6,337	6,987	7,359	7,730	8,380	8,752	9,123	9,402	9,774
<b>Tfin</b> <b>(s)</b>	5,873	6,244	6,894	7,266	7,637	8,287	8,659	9,031	9,309	9,681	12,096

La tabella mostra i numerosi calcoli effettuati dal prototipo. È sorprendente come il programma sia stato in grado di gestire un buon numero di eventi separandoli uno a uno e riorganizzarli nelle varie colonne. L'analisi frequenziale è stata eseguita egregiamente in quanto, oltre ai numerosi eventi presenti nella melodia, vi era anche la componente di rumore causata dal plettro. Nonostante ciò, il prototipo non ha considerato questo rumore come segnale utile. Per quanto riguarda, invece, l'analisi temporale si ottengono buoni risultati, in quanti i valori di durata delle singole note sono piuttosto vicini a quelli reali. La durata reale dell'intero file audio è di 12,355 secondi mentre quella calcolata è di 12,096 confermando, ulteriormente, la bontà dell'analisi temporale.

Rispetto alla partitura della figura 32 è possibile notare che le note con alterazioni, in questo caso *SIb*, vengano trattate dal prototipo come *LA#*. Questo non è un errore del prototipo, ma è una scelta stabilita, in fase di programmazione, in quanto due note con la stessa frequenza possono avere due nomi diversi.

In conclusione si può affermare che la trascrizione della melodia ha ottenuto risultati più che soddisfacenti.

## 6. CONSIDERAZIONI E CONCLUSIONI

Il seguente capitolo, quello conclusivo, ha il compito di dare un giudizio ai risultati ottenuti durante le prove per realizzazione dell'elaborato evidenziando i vantaggi e gli svantaggi del prototipo.

Successivamente sarà necessario soffermarsi sulle eventuali implementazioni da attuare per modificare e migliorare il prototipo stesso.

### 6.1 Risultati

Dopo aver analizzato, nel capitolo 5, in modo dettagliato il funzionamento del prototipo con ben venti campioni testati, è necessario esprimere un giudizio complessivo sui vantaggi e gli svantaggi del programma.

La trascrizione che otteniamo può essere definita una trascrizione a “basso livello”, non per l'accuratezza del risultato, ma più per l'utilizzo finale. Ciò che otteniamo è una semplice tabella nella quale sono inseriti tutti i valori utili per il riconoscimento della nota e per la sua durata. Sicuramente, è molto importante notare che gli eventi sono ben organizzati, considerato che ognuno di questi ha una colonna dedicata posizionata secondo l'ordine cronologico delle note della melodia data in input.

Concludendo, i risultati ottenuti, nelle le prove effettuate, sono molto soddisfacenti sia per quanto riguarda l'analisi temporale sia quella frequenziale. Nella prima si è potuto stimare una durata, per ogni nota, prossima a quella reale e anche nel caso di melodie contenenti della pause, i risultati ottenuti sono molto buoni.

Per quanto riguarda l'analisi frequenziale, l'accuratezza di calcolo è stata ottima permettendo di gestire le diverse casistiche riscontrate. Bisogna comunque sottolineare che un suono di chitarra è un suono complesso ricco di armoniche secondarie che rendono, sicuramente, molto difficile la rilevazione della frequenza fondamentale e, quindi, la nota. Inoltre, durante la fase di registrazione abbiamo potuto verificare che un leggero rumore si somma al segnale utile, ma, fino ad una certa soglia, il prototipo è in grado di ignorarlo. Dettò ciò, nel complesso, il lavoro svolto ha raggiunto gli obiettivi prefissati.

## 6.2 Problematiche riscontrate e risolte

Analizzando, invece, tutto ciò che riguarda le operazioni eseguite dal prototipo bisogna soffermarsi su diverse casistiche. La maggior parte dei problemi riscontrati sono dovuti, sicuramente, dal file audio dato in ingresso. Come già esposto all'inizio del capitolo 4, il file audio in ingresso deve assolutamente seguire i parametri di campionamento e quantizzazione descritti, in caso contrario otterremo un errore già nelle prime righe di codice del prototipo.

L'unica operazione effettuata di editing audio sui file audio è una semplice normalizzazione a 0 dB del segnale.

Si è preferito registrare le melodie di chitarra utilizzando un microfono posto davanti la buca dello strumento piuttosto che utilizzare una chitarra acustica elettrificata, passando, quindi, da un amplificatore, il quale aggiungerebbe, alla registrazione, ulteriore rumore di fondo causato dai componenti elettrici che lo compongono.

A questo punto è molto importante trattare di alcuni inconvenienti riscontrati durante la fase di registrazione che non riguardano la parte tecnica, ma del modo di suonare le melodie. Nelle prime prove effettuate, capitava spesso che alcuni eventi venissero completamente ignorati durante la fase di calcolo e, di conseguenza, non apparivano nello spartito finale. Dopo una accurata analisi si è giunti alla conclusione che, durante la fase di registrazione, quando una nota veniva suonata mentre quella precedente non era ancora terminata del tutto, l'attacco veniva alterato e quindi ignorato dal programma. Questo è causato, nella maggior parte dei casi, dal fatto che il decadimento di un suono di chitarra è piuttosto lento permettendo così alle sue componenti frequenziali di sommarsi con quelle appartenenti all'attacco della nota successiva.

In conclusione si è cercato di registrare delle melodie prestando attenzione anche sul modo di suonare cercando di evitare il problema descritto precedentemente.

I casi sopra descritti riguardano tutto ciò che è stato riscontrato nella fase di registrazione e che è stato risolto in modo opportuno.

Per quanto riguarda la compilazione del programma non sono stati riscontrati molti problemi. Inizialmente è stato creato un semplice schema a blocchi, dopodiché si è provveduto a realizzare tutte le parti che lo costituiscono. Un inconveniente riscontrato durante la realizzazione del prototipo è stato causato dal primo evento della melodia che non sempre veniva considerato. Questo problema è stato risolto inserendo un vettore di

valori nulli prima del vettore contenente i valori della melodia, in modo tale che, nella fase di calcolo, il programma avesse la possibilità di individuare tutti gli eventi.

### **6.3 Implementazioni future**

L'elaborato realizzato, nonostante gli ottimi risultati ottenuti, è solamente l'inizio di uno studio che può essere raffinato ed implementato. L'estrazione di informazioni da un file audio non è un lavoro semplice, ma ha molte casistiche da affrontare e risolvere. All'interno di questo progetto sono state prese in considerazione le problematiche maggiori e sono state risolte grazie alle conoscenze sviluppate.

A partire dai risultati ottenuti, sarebbe molto interessante effettuare ulteriori studi per migliorare il progetto. Come primo caso di studio sarebbe molto importante raffinare la trascrizione passando da una tabella contenente dei valori a una vera e propria partitura scritta. Le casistiche da prendere in considerazione sono diverse e molte; ad esempio bisognerebbe, per ogni melodia presa in considerazione, calcolarne i bpm e, quindi, ottenere il valore di ogni nota (quarto, ottavo, sedicesimo ecc.). Questo tipo di problema non è stato preso in considerazione nell'elaborato in quanto i valori delle note e pause fanno riferimento al tempo assoluto. Altro caso, piuttosto complicato, da prendere in considerazione per una trascrizione su pentagramma è legato ad un discorso melodico e armonico, ovvero analizzare le melodie e ricavarne il centro tonale, secondo le regole di armonia, ottenendo così le alterazioni in chiave con gli opportuni # e b. In questo modo il posizionamento della nota sul pentagramma sarà corretto dato che, in armonia moderna, due note con la stessa frequenza possono essere chiamate con lo stesso nome (es. Re# e Mib), ma solo in base al centro tonale si può dedurre quale delle due sia corretta. Molto complessa, sicuramente, è la trascrizione dei gruppi irregolari, come le terzine, proprio perché non rispettano la suddivisione naturale del brano musicale.

Un altro campo di studio molto interessante è quello legato a melodie composte da due o più note sovrapposte fino a raggiungere la polifonia, ovvero uno stile che combina due o più voci indipendenti tra loro. Diverse volte, all'interno di questo elaborato, è stato sottolineato come le melodie utilizzate per l'analisi fossero composte da singole note in successione, questo per rendere meno complicata l'estrazione di ogni singola nota.

In conclusione, si può certamente affermare che le implementazioni future, da attuare a questo prototipo, sono molte e interessanti, ricche di sfaccettature e di campi di studio

che non riguardano solamente l'elaborazione e l'estrazione di informazioni da un segnale, ma che, invece, tengono in considerazione l'aspetto melodico/armonico il quale richiede un'analisi approfondita e articolata.

## 7. BIBLIOGRAFIA E SITOGRAFIA

### 7.1 Bibliografia

- [1] G. Presti, Mappatura delle informazioni e segmentazione di segnali audio, Università degli Studi di Milano, 2008.
- [2] M. Malcangi, Informatica applicata al suono per la comunicazione musicale - Musical digital audio, Santarcangelo di Romagna: Maggioli editore, 2008.
- [3] S. Cingolani e R. Spagnolo, Acustica musicale e architettonica, Novara: De Agostini Scuola, 2008.
- [4] G. Naldi e L. Pareschi, Matlab Concetti e progetti, Milano: Apogeo, 2002.
- [5] M. Bonvini, La fisica della chitarra, Dipartimento di Fisica (DIFI), Università di Genova, 2007.
- [6] A. Bertoni e G. Grossi, Elaborazione Numerica dei Segnali, Università degli Studi di Milano - Dipartimento di Scienze dell'Informazione, 2003.
- [7] M. Malcangi, Elaborazione numerica del segnale - Algoritmi e Metodologie per il DSP: Teoria e pratica, Milano: Libreria Clup, 2005.
- [8] W. Piston, Armonia, Torino: E.D.T. Edizioni di Torino, 1989.
- [9] A. Mancuso, Scale musicali, Università degli studi di Milano, 2009.
- [10] A. Mancuso, Introduzione e cenni di acustica musicale, Università degli studi di Milano, 2009.
- [11] P. Arena e A. Basile, Mathworks Matlab, Università degli studi di Milano, 2002.
- [12] D. Manca, Tutorial di Matlab, Politecnico di Milano.
- [13] A. Falletto, Algoritmi e tecnologie per il riconoscimento vocale - stato dell'arte e sviluppi futuri, Torino, 2007.
- [14] G. Agostini, «Sistemi informatici per il riconoscimento automatico di timbriche musicali in segnali monofonici mediante tecniche di statistica multivariata,» Politecnico di Milano, 1999, pp. 20-28.
- [15] S. Battiato, Audio digitale, Università di Catania, 2007.
- [16] M. Sozzi, Fisica musicale, Università di Pisa, 2006.

## 7.2 Sitografia

- [1] Gruppo di Ricerca sull'Insegnamento/Apprendimento della Fisica, «La Chitarra,» Marzo 2007. [Online]. Available: <http://www.uop-perg.unipa.it/promonda/pagine/chitarra.htm>.
- [2] Gruppo di Ricerca sull'Insegnamento/Apprendimento della Fisica, «Il suono - caratteristiche del suono,» Febbraio 2007. [Online]. Available: [http://www.uop-perg.unipa.it/promonda/pagine/caratteristiche\\_del\\_suono.htm](http://www.uop-perg.unipa.it/promonda/pagine/caratteristiche_del_suono.htm).
- [3] M. Graziani, «Dispense di Acustica per Musicisti,» Ottobre 2009. [Online]. Available: [http://www.maurograziani.org/text\\_pages/acoustic/acustica/](http://www.maurograziani.org/text_pages/acoustic/acustica/).
- [4] McGill University, «Audio Processing in Matlab,» [Online]. Available: <http://www.music.mcgill.ca/~gary/307/week1/matlab.html>.
- [5] The MathWorks, «Product Documentation,» [Online]. Available: <http://www.mathworks.it/>.
- [6] E. W. Kamen e B. S. Heck, «Fundamentals of signals and system,» 2000. [Online]. Available: <http://users.ece.gatech.edu/~bonnie/book/book.html>.
- [7] Matlab Tutorials, «Matlab Tutorials - The Language Of Technical Computing,» [Online]. Available: <http://www.matlabtutorials.com/>.
- [8] Wikibooks, «MATLAB Programming/Error Messages,» [Online]. Available: [http://en.wikibooks.org/wiki/MATLAB\\_Programming/Error\\_Messages](http://en.wikibooks.org/wiki/MATLAB_Programming/Error_Messages).
- [9] Polimi, 2002. [Online]. Available: <http://risorse.dei.polimi.it/dsp/courses/fens/files/stimfreq.m>.
- [10] «Anatomia della chitarra acustica,» [Online]. Available: <http://piero.irradium.net/ANATOMIA%20DELLA%20CHITARRA%20ACUSTICA.htm>.
- [11] P. Mazzucchelli, «Fondamenti di segnali e trasmissione,» [Online]. Available: <http://risorse.dei.polimi.it/dsp/courses/fst/files/lab3.pdf>.