

ALMA MATER STUDIORUM - UNIVERSITÀ DI
BOLOGNA
SEDE DI CESENA

Scuola di Ingegneria ed Architettura
Corso di Laurea in Ingegneria dei Sistemi Elettronici per lo
Sviluppo Sostenibile

SPERIMENTAZIONE DI
ALGORITMI DI
LOCALIZZAZIONE PER
SORGENTI ACUSTICHE

Elaborato in
Laboratorio di Elettronica e Telecomunicazioni

Tesi di Laurea di:
ELIA FAVARELLI

Relatore:
Prof. Ing.
ANDREA GIORGETTI

SESSIONE III
ANNO ACCADEMICO 2014–2015

PAROLE CHIAVE

Localizzazione Audio

Array di microfoni

MATLAB

Simulazione

DigilentTM

Indice

Introduzione	1
1 Sistema di Acquisizione	3
1.1 Dispositivi e soluzioni impiegati	3
1.2 Principio di funzionamento	4
1.3 Condizioni ottimali di misura	5
2 Sensori	7
2.1 Capsule microfoniche	7
2.2 Filtro RC	9
2.3 Sensore definitivo	10
3 <i>Digilent Analog Discovery</i>	11
3.1 Waveform	13
3.1.1 Scope	14
3.1.2 WaveGen	15
3.1.3 Voltage	16
3.2 Interfacciamento MATLAB	17
3.2.1 Limiti sulle prestazioni	18
4 Funzioni MATLAB	19
4.1 offsetAD1/offsetAD2	19
4.2 acqAD1/acqAD2	20
4.3 estimationangle	21
4.4 pos	22
4.5 test_offline	24
5 GUI MATLAB	27
5.1 Interfaccia grafica	27
5.2 Funzioni GUI	29
5.2.1 estimate_OpeningFcn	29

5.2.2	Calibrate_Callback	30
5.2.3	Generate_Landscape_Callback	30
5.2.4	Start_Callback	31
5.3	Funzioni aggiuntive	34
5.3.1	Offline_Callback	34
5.3.2	Clear_Callback	34
5.3.3	estimate_error_Callback	35
6	Conclusioni	37
	Elenco Figure	39
	Bibliografia	41
	Ringraziamenti	43

Introduzione

L'obiettivo di questo elaborato è quello di realizzare un sistema di acquisizione *real-time* e *low cost*, in grado di stimare la posizione di una sorgente sonora.

Per acquisizione *real-time* si intende un sistema in grado di seguire gli spostamenti della sorgente audio nell'ambiente, limitando il più possibile i tempi di latenza fra un'acquisizione e l'altra.

A tal fine si sfrutteranno due coppie di microfoni, che concorreranno in maniera sequenziale alla stima dell'angolo di arrivo del suono (e quindi della retta su cui giace il punto di provenienza del suono). Combinando i dati raccolti dai quattro microfoni sarà dunque possibile risalire alla posizione di provenienza della sorgente sonora.

Il software (implementato in MATLAB) verrà ampiamente discusso nei seguenti capitoli, il suo scopo sarà quello di gestire l'acquisizione dati, garantire il funzionamento *real-time* e fornire una stima di posizione della sorgente sonora, mostrando il relativo errore commesso durante la misura, nel caso in cui si conosca a priori la posizione della sorgente.

Capitolo 1

Sistema di Acquisizione

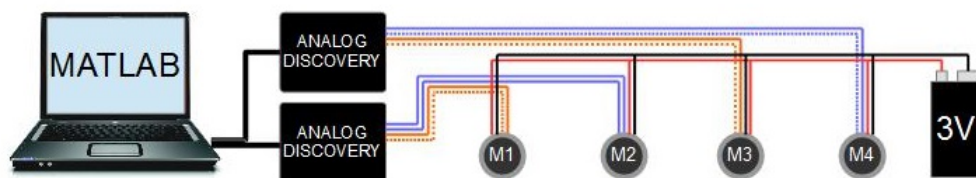


Figura 1.1: Sistema di acquisizione

Come si nota dalla *figura 1.1*, il sistema è suddiviso in 3 parti:

- Programma MATLAB che gestisce le schede di acquisizione dati.
- Schede *Analog Discovery* che acquisiscono dati dai 2 canali di input analogici e li trasmettono al PC tramite porta USB.
- Microfoni e alimentazione che forniscono i segnali analogici necessari per effettuare la stima della posizione della sorgente sonora.

1.1 Dispositivi e soluzioni impiegati

Per realizzare il sistema di acquisizione audio sono stati impiegati i seguenti dispositivi:

- Pacco batterie da 3V, necessario per l'alimentare dei 4 microfoni.

- 4 capsule microfoniche che fungono da sensori sonori, è bene verificarne il rapporto segnale-rumore e la banda di funzionamento (presenti sui datasheet); per avere un'esperienza di misura migliore è inoltre fortemente consigliato prevedere un circuito di amplificazione del segnale generato in uscita dai microfoni, altrimenti il segnale misurato risulterà dell'ordine delle decine di mV (in questo elaborato tale soluzione non verrà trattata, saranno quindi necessari suoni di elevata intensità per far sì che i microfoni li rilevino).
- Componenti discreti, essenziali per far sì che i microfoni forniscano un segnale apprezzabile ai morsetti, è necessario costruire un piccolo circuito RC che funga da filtro e che operi un adattamento di impedenza (si discuterà di tali aspetti nel capitolo successivo).
- PC con software MATLAB (N.B. è necessario avere in dotazione almeno la versione 2013 del software, altrimenti il pacchetto di funzioni *digilent* non saranno supportati).
- Scheda di interfacciamento al PC, necessaria per trasferire il segnale analogico acquisito dai microfoni al PC; per svolgere questa funzione sono state impiegate 2 *Analog Discovery* (per un totale di 4 canali) di cui si tratterà in maniera approfondita nel capitolo 4.

1.2 Principio di funzionamento

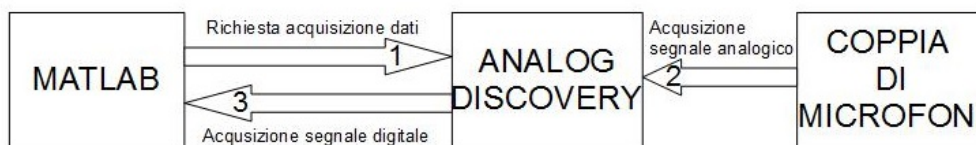


Figura 1.2: Diagramma a Blocchi

Per capire come funziona il sistema di acquisizione si analizzi il diagramma a blocchi in *figura 1.2*.

Il programma MATLAB avvia, su richiesta dell'utente, delle sessioni *digilent* con cui è possibile contattare le periferiche *Analog Discovery*, le quali risponderanno con un'acquisizione dati dai canali analogici.

È importante sottolineare che i due canali di acquisizione analogica, presenti nelle schede, sono fra loro sincronizzati, condizione fondamentale per stimare correttamente l'angolo di arrivo del suono.

Il software ordina quindi un'acquisizione prima ad una poi all'altra scheda, (l'ordine di esecuzione è del tutto arbitrario e non influisce sulla stima) ciascuna delle due *Analog Discovery* risponde quindi con 2 vettori dati che contengono le misure dei canali analogici collegati ai rispettivi 2 microfoni. A questo punto il resto dell'elaborazione viene eseguita dal software, al termine della quale viene restituito il valore della posizione della sorgente sonora nell'istante in cui il segnale è stato acquisito.

1.3 Condizioni ottimali di misura

Per assicurare una buona stima è importante in primo luogo trovarsi in uno spazio sufficientemente ampio (minimo una decina di metri) per minimizzare i riverberi dell'ambiente e utilizzare un PC portatile non collegato al caricabatterie (altrimenti il *ripple* residuo dell'alimentazione si propaga fino ai dispositivi di misura alimentati tramite la porta USB). Altrettanto importante è effettuare le misure in un ambiente poco rumoroso e con sorgenti sonore molto intense per evitare che il rumore di fondo prevalga sulla sorgente audio.

Un aspetto meno intuitivo (valutato sperimentalmente) è invece rappresentato dalla convenienza di utilizzare come sorgente un suono il più possibile *impulsato*, per il semplice motivo che la funzione di *cross correlazione*, con cui si stima l'angolo, fornisce una stima migliore con questo tipo di segnali.

Vi sono anche due aspetti rilevanti sul posizionamento reciproco dei microfoni:

- Distanza relativa, è bene che i microfoni si trovino distanti fra loro almeno 1 m, affinché il segnale sonoro arrivi ben ritardato su un microfono rispetto all'altro. Distanze inferiori verranno indagate in futuro.
- Punti ciechi, utilizzando 2 coppie di microfoni e quindi l'intersezione di 2 sole rette, in qualsiasi modo si dispongano, vi sarà sempre una retta di punti nello spazio per cui le 2 rette saranno sovrapposte fra loro (parallele) e il punto risulterà dunque impossibile da stimare; vi saranno anche posizioni nelle vicinanze di tali punti, per cui piccole variazioni di angolo determineranno grandi errori nella stima della posizione, che risulterà quindi stimata con molta meno accuratezza (*figura 1.3*).

Per risolvere il problema dei punti ciechi è sufficiente utilizzare una terza coppia di microfoni.

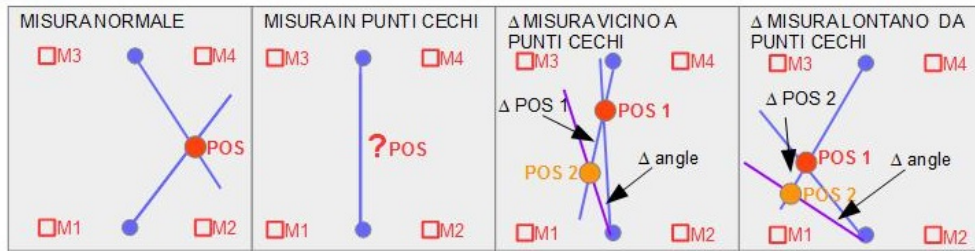


Figura 1.3: Punti critici di misura

Capitolo 2

Sensori

Come già accennato, i sensori audio sono composti da una capsula microfonica capacitiva e da un filtro RC (*figura 2.1*) di cui ora si tratterà nel dettaglio.

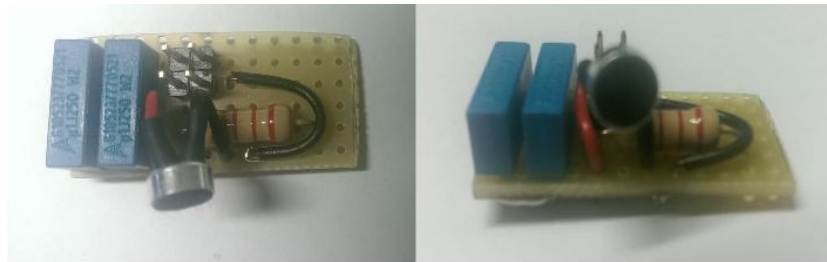


Figura 2.1: Sensore audio

2.1 Capsule microfoniche

Il microfono a condensatore sfrutta l'effetto capacitivo. All'interno troviamo una membrana metallica, le cui lamine vengono sollecitate dalle vibrazioni trasmesse nell'aria. Per funzionare il microfono necessita di una fonte di alimentazione [1].

I datasheet [2] forniscono informazioni dettagliate sulle capsule:

- **Directivity:** possono essere omnidirezionali oppure unidirezionali. Nel primo caso la risposta del microfono è il più possibile uniforme in tutte le direzioni, nel secondo invece il microfono risulta molto sensibile al suono solo se orientato in direzione della sorgente acustica, in caso contrario il segnale captato risulterà di gran lunga inferiore.

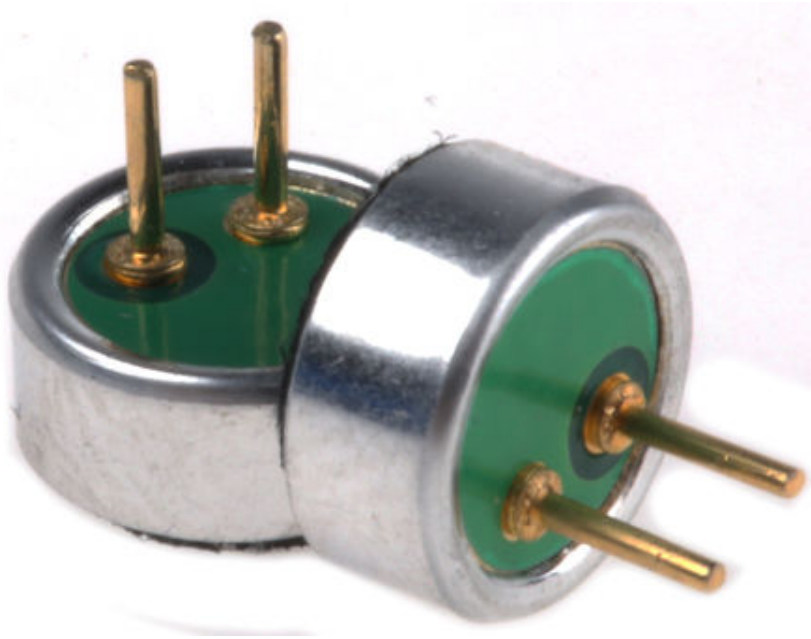


Figura 2.2: Capsule microfoniche [3]

- Standard Operating Voltage: livello di tensione che garantisce il corretto funzionamento della capsula.
- Maximum Operating Voltage: massimo livello di tensione applicabile alla capsula senza danneggiarla.
- Maximum Current Consumption: massima corrente assorbita dalla capsula (utile per valutare il consumo di potenza e la corrente che deve erogare il generatore).
- Output Impedence: impedenza d'uscita, necessaria per l'adattamento di impedenza e per dimensionare il filtro.
- Frequency: range di frequenze udibili dal microfono.
- Sensitivity: sensibilità della capsula microfonica (espressa in dB).
- Signal to Noise Ratio: rapporto segnale-rumore (espressa in dB).
- Operating Temperature: range di temperature a cui può funzionare il dispositivo.

Per implementare il sensore audio è stata utilizzata la *capsula microfonica a condensatore RS*:

- Tensione operativa: 2V
- Sensibilità: -42dB
- Frequenza: 50Hz - 16kHz (rappresenta quasi tutta la banda udibile dall'essere umano)
- Omnidirezionale: caratteristica fondamentale per l'applicazione in esame
- Impedenza d'uscita: 2,2 KOhm

2.2 Filtro RC

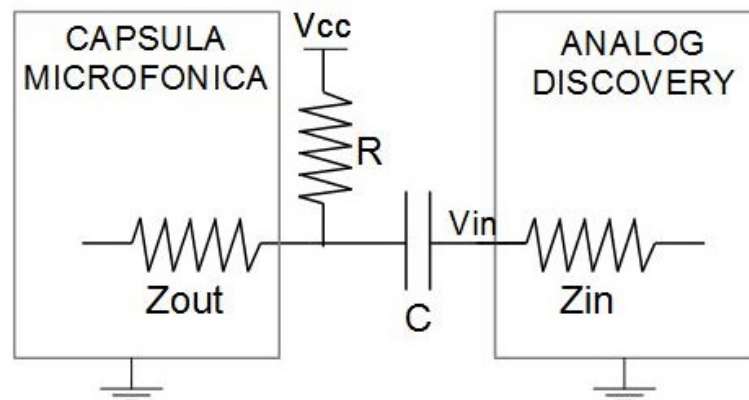


Figura 2.3: Schema Filtro

Per alimentare la capsula microfonica e acquisire il segnale analogico ai morsetti della stessa, è necessario il circuito RC mostrato in *figura 2.3*.

Per poter dimensionare i 2 parametri (R e C) viene fornito il valore di impedenza d'uscita della capsula microfonica ($Z_{out}=2,2\text{ KOhm}$) e quello dell'impedenza di ingresso del canale di acquisizione analogico della scheda *digilent* (1 MOhm).

Per prima cosa si sceglie $R=Z_{out}$, così da garantire la massima escursione del segnale che verrà poi letto dall'*Analog Discovery*. È necessario poi inserire un condensatore per separare il segnale frequenziale dall'alimentazione DC. Per dimensionare il condensatore bisogna tenere conto che complessivamente

la rete RC si comporta come un filtro passa alto, diventa quindi fondamentale il valore di C che determinerà la frequenza di taglio del filtro secondo la seguente espressione:

$$f_T = \frac{1}{2\pi RC} \quad (2.1)$$

Imponendo $f_T = 50Hz$ per assicurare tutta la banda di funzionamento del microfono e allo stesso tempo per limitare l'effetto dei disturbi a bassa frequenza, si ricava il valore di C pari a $1,45 \mu F$, approssimato a $2 \mu F$ e implementato mediante il parallelo di 2 condensatori da $1 \mu F$ ciascuno.

2.3 Sensore definitivo

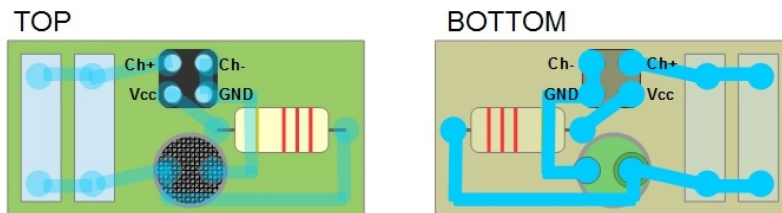


Figura 2.4: Footprint Sensore

Per realizzare il sensore definitivo è quindi sufficiente collegare i componenti su di un supporto adeguato (nel caso in esame è stata utilizzata una millefori per ospitare tutti i componenti).

L'alimentazione viene portata al microfono tramite il resistore (nella capsula acquistata il terminale positivo è segnalato attraverso un pallino verde scuro attorno al reoforo corrispondente), l'altro terminale del microfono è posto a massa; il segnale viene preso fra i 2 condensatori (collegati in parallelo) e massa (figura 2.4).

Per garantire un buon collegamento con l'*Analog Discover* sono stati utilizzati dei pin *I/O general purpose*.

Capitolo 3

Digilent Analog Discovery



Figura 3.1: Analog Discovery [4]

L'*Analog Discovery* [5] è un dispositivo multifunzione, di dimensione ridotta e a basso costo, in grado di generare ed acquisire segnali digitali e analogici (*figura 3.1*). Il dispositivo presenta le seguenti periferiche e i rispettivi PIN:

- 2 canali di acquisizione dati analogici (1+, 1-, 2+, 2-)
- 4 PIN di massa (rappresentati con una freccia rivolta verso il basso)
- 1 generatore di tensione DC (V+, V-)

- 2 generatori di forme d'onda (W1, W2)
- 2 PIN per la gestione del trigger (T1, T2)
- 16 PIN di I/O digitali (enumerati da 0 a 15)

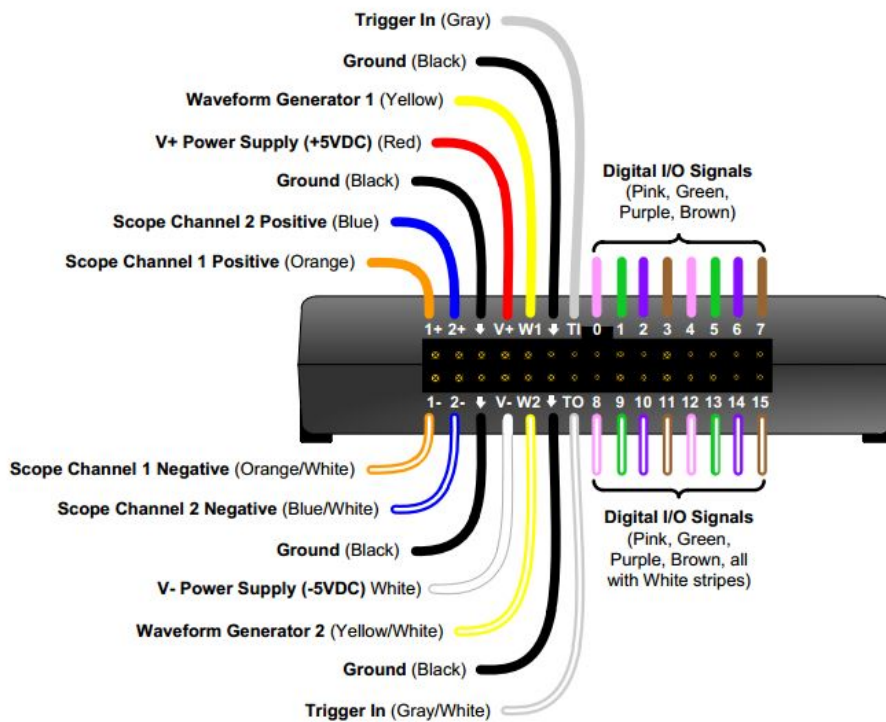


Figura 3.2: PIN I/O Analog Discovery [6]

Tramite il software *Waveform* sviluppato dalla *Digilent* e disponibile gratuitamente sul sito della casa stessa, è possibile interfacciare con semplicità il dispositivo a un qualunque PC. È supportato in oltre l'interfacciamento con MATLAB attraverso l'apposito *Support Package* scaricabile gratuitamente dal sito *MathWorks*.

Per iniziare a lavorare con l'*Analog Discovery* è quindi sufficiente seguire i seguenti passi:

- Dotarsi di una scheda di acquisizione *Analog Discovery*
- Installare il software *Waveform*
- Installare il *Support Package*

3.1 Waveform



Figura 3.3: Waveform Digilent schermata iniziale

All'avvio il programma controlla automaticamente la presenza di dispositivi *Analog Discovery* collegati al PC e viene richiesto a quale di essi interfacciarsi (N.B. *non è possibile lavorare con 2 dispositivi diversi all'interno dello stesso workspace, è però possibile lanciare 2 volte il programma associandolo a 2 dispositivi differenti*). Compare poi la schermata mostrata in *figura 3.3* dove è possibile selezionare una fra le seguenti opzioni:

- Scope: Oscilloscopio a 2 canali
- WaveGen: Generatore di forme d'onda
- Voltage: Generatore di tensione DC

3.1.1 Scope

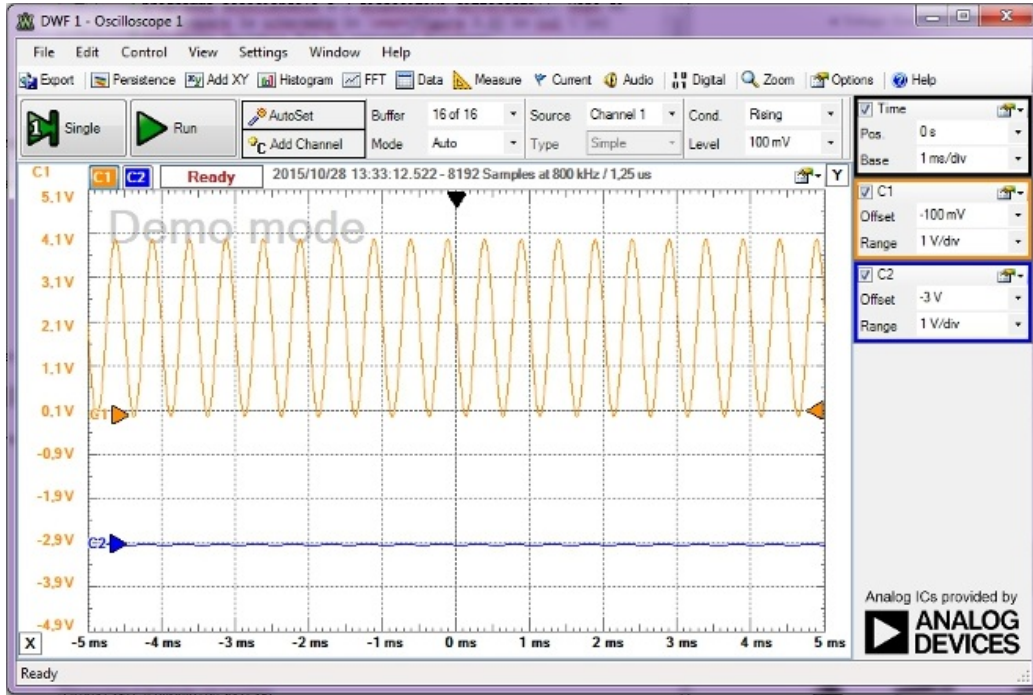


Figura 3.4: Waveform Scope

La funzione Scope consente di effettuare misure analogiche da entrambi i canali di una scheda. Analizzando la schermata in *figura 3.4* si possono notare le seguenti funzionalità:

- Run: lancia l'acquisizione da entrambi i canali della scheda *diligent* collegata (da interrompere manualmente premendo nuovamente il tasto).
- Single: effettua un'unica misurazione a partire da un evento di trigger (da impostare prima di effettuare tale misura), se l'evento di trigger non si verifica dopo un certo lasso di tempo la misura viene comunque lanciata automaticamente.
- Export: consente di esportare il file relativo all'ultima misurazione in formato *txt* o *csv*.
- Nella parte centrale sono presenti dei menù a tendina con cui è possibile impostare le specifiche dell'evento di trigger.

- Nella parte destra della schermata è possibile impostare range e offset dei 2 canali di acquisizione analogica e si può modificare la base dei tempi; modificando la risoluzione temporale si modifica anche la frequenza di campionamento con cui viene effettuata la misura (N.B. *Siccome il numero di campioni è fissato pari a 8192, aumentando la risoluzione temporale della finestra se ne ridurrà indirettamente anche l'ampiezza*).

3.1.2 WaveGen

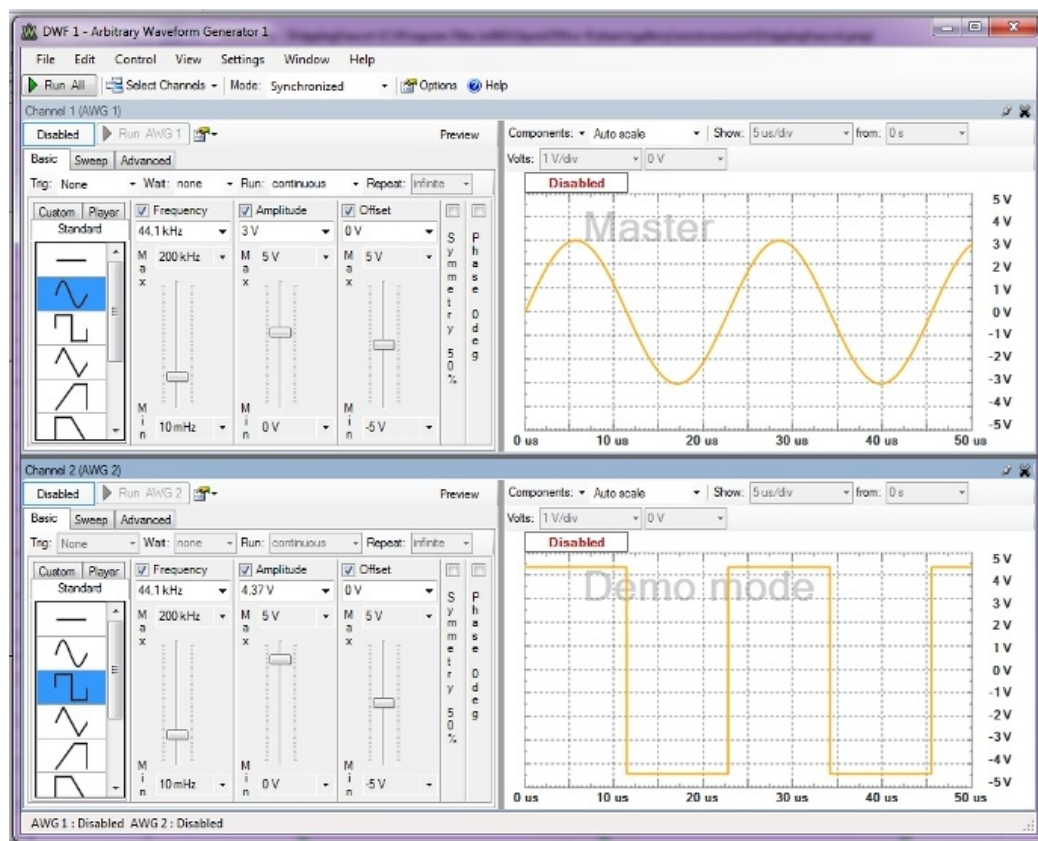


Figura 3.5: Waveform WaveGen

La funzione *WaveGen* consente di generare segnali analogici in totale libertà. Analizzando la schermata in *figura 3.5* si possono notare le seguenti funzionalità:

- Run All: genera su entrambi i canali AWG dell'*Analog Discovery* i segnali impostati (da arrestare manualmente premendo nuovamente il tasto).
- Run AWG 1/2: genera sul canale AWG selezionato la forma d'onda impostata (da arrestare manualmente premendo nuovamente il tasto).
- Disabled/Enabled: indica se il canale AWG è abilitato oppure no; se il canale è abilitato, ma non è stato attivato il comando Run, in uscita dai pin AWG verrà portato il valor medio della funzione che si intende generare, in caso contrario il valore sarà zero.
- In alto a sinistra si trovano i comandi Basic, Sweep e Advanced, selezionando le varie opzioni è possibile avere funzionalità sempre più complesse per generare forme d'onda particolari (è possibile generare anche modulazioni AM, FM).
- Nel lato sinistro sono presenti i comandi Standard, Custom e Player. Il comando Standard consente di generare forme d'onda base modificandone duty cycle, ampiezza, sfasamento, ecc.; il comando Custom consente di editare forme d'onda in maniera più raffinata fino a decidere i valori dei singoli campioni; infine il comando player consente di caricare file in formato *wav* e riprodurre le forme d'onda ad esso associate.

3.1.3 Voltage

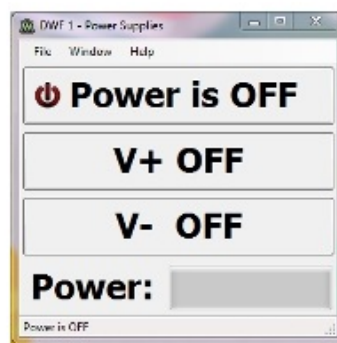


Figura 3.6: Waveform Voltage

Come si nota dalla *figura 3.6* l'interfaccia è molto semplice. Sono presenti tre tasti, il primo attiva il generatore di tensione, gli altri 2 abilitano le porte V+ e V- dell'*Analog Discovery*. È possibile generare tensioni con valori -5 V e +5 V rispetto ai PIN di massa ma non è consentita la regolazione di tali livelli. Nella parte bassa della finestra vi è un comodo indicatore che mostra la potenza erogata dai terminali V+ e V-.

3.2 Interfacciamento MATLAB

Sul sito MathWorks sono presenti vari esempi su come interfacciare l'*Analog Discovery* con MATLAB sia per generare segnali, sia per acquisirli [7]. In questo elaborato verrà trattata solamente la parte di acquisizione dati. La procedura per avviare una sessione di misura con l'*Analog Discovery* attraverso MATLAB è la seguente:

- Collegare il dispositivo al PC (N.B. è importante effettuare il collegamento prima di aprire MATLAB altrimenti il dispositivo non verrà riconosciuto dal programma).
- Creare una sessione digilent mediante cui si controllerà il dispositivo.

```
s = daq.createSession('digilent');
```

- Aggiungere un canale di input analogico con cui si effettuerà la misura.

```
ch = addAnalogInputChannel(s,'AD1', 1, 'Voltage');
```

- Impostare frequenza di campionamento, durata dell'acquisizione e range di misura (sono disponibili i range [-2.5 2.5] e [-25 25]).

```
s.Rate = 300e3;  
s.Channels.Range = [-2.5 2.5];  
s.DurationInSeconds = 0.5;
```

- Lanciare l'acquisizione, che restituirà un vettore di ampiezze e un asse temporale (*triggerTime* contiene le informazioni inerenti a data e ora in cui è stata effettuata la misura).

```
[data, timestamps, triggerTime] = s.startForeground;
```

3.2.1 Limiti sulle prestazioni

Purtroppo i dispositivi Analog Discovery non supportano i comandi *background* di MATLAB ovvero è impossibile eseguire altre istruzioni mentre il dispositivo è impegnato ad effettuare una misurazione o a generare un segnale. Un'ulteriore limitazione consiste nell'impossibilità di avviare sessioni di misura multiple, ovvero prima di avviare una nuova misurazione è necessario terminare la precedente (è tuttavia possibile acquisire dati contemporaneamente dai 2 canali di una singola scheda). Informazioni più dettagliate sui limiti di questi dispositivi sono presenti sul sito *MathWorks* nella sezione dedicata agli *Analog Discovery* sotto la voce *Vendor Limitation* [8].

Capitolo 4

Funzioni MATLAB

In questo capitolo verranno discusse tutte le funzioni utilizzate nel software che gestisce l'acquisizione dati e la stima della posizione, ad una breve descrizione sulle finalità, seguirà la spiegazione delle singole istruzioni. Molte funzioni risultano duplicate per gestire 2 Analog Discovery contemporaneamente, in tali casi verranno citate entrambe nel titolo del capitolo ma ne verrà esposta solamente una.

4.1 `offsetAD1/offsetAD2`

Questa funzione si occupa di effettuare una stima dell'offset dei 2 canali di misura di una scheda *digilent* (è necessario quindi che la sorgente acustica non sia presente durante l'esecuzione di tale funzione) e restituisce 2 valori (*offset1* ed *offset2*) che verranno poi utilizzati per eliminare l'offset durante le misure vere e proprie.

Per prima cosa, come già analizzato nel capitolo precedente, si crea una sessione di acquisizione dati con la scheda *digilent* impostando: frequenza di campionamento 44.1 KHz, durata 2 s e il range inferiore disponibile (le misure risultano molto più accurate scegliendo il range adeguato).

```
s = daq.createSession('digilent');
ch_in = addAnalogInputChannel(s,'AD1', [1 2], 'Voltage');
s.Rate = 44.1e3;
s.DurationInSeconds = 2;
s.Channels(1).Range = [-2.5 2.5];
s.Channels(2).Range = [-2.5 2.5];
```

Dopo di che si avvia l'acquisizione tramite il comando:

```
[data] = s.startForeground;
```

In questo caso l'acquisizione avviene da entrambi i canali, i dati raccolti vengono salvati in una matrice (*data*) anzichè in un vettore, per estrapolare quindi dalla matrice le informazioni di interesse, vengono inizializzati 2 vettori (*data1* e *data2*) per poi essere riempiti (attraverso un ciclo for) ciascuno con una colonna di elementi della matrice *data*.

```
data1=zeros(length(data),1);  
data2=zeros(length(data),1);  
for i=1:length(data)  
data1(i)=data(i,1);  
data2(i)=data(i,2);  
end
```

A questo punto per estrapolare il valor medio della misura effettuata è sufficiente utilizzare il comando *mean* sui due vettori.

```
offset1 = mean(data1);  
offset2 = mean(data2);  
end
```

4.2 acqAD1/acqAD2

La funzione *acqAD1* è quella che si occupa dell'acquisizione vera e propria. Effettua una misura in contemporanea su entrambi i canali di acquisizione dell'*Analog Discovery* e restituisce 2 vettori dati (*data1* e *data2*) e un terzo vettore che contiene l'asse dei tempi (*timestamps*), necessita in ingresso degli argomenti *offset1* e *offset2* calcolati con la precedente funzione.

Al solito si procede inizializzando una sessione di acquisizione dati, questa volta di durata 1 secondo (il tempo di acquisizione è stato scelto in modo da avere un giusto compromesso fra qualità della stima e tempo che intercorre fra una misura e la successiva).

```
s = daq.createSession('digilent');  
ch_in = addAnalogInputChannel(s,'AD1', [1 2], 'Voltage');  
s.Rate = 44.1e3;  
s.DurationInSeconds = 1;  
s.Channels(1).Range = [-2.5 2.5];  
s.Channels(2).Range = [-2.5 2.5];
```

Anche in questo caso, avviando l'acquisizione, i dati vengono salvati nella matrice *data*, quindi vengono inizializzati 2 vettori per rendere più pratica la successiva elaborazione.

```
[data, timestamps] = s.startForeground;
data1=zeros(length(data),1);
data2=zeros(length(data),1);
```

Infine attraverso un ciclo for i dati contenuti nella matrice vengono inseriti nei 2 vettori e allo stesso tempo viene sottratto (elemento per elemento) il valore di offset del canale, ricavato precedentemente mediante la funzione *offsetAD1*

```
for i=1:length(data)
data1(i)=data(i,1)-offset1;
data2(i)=data(i,2)-offset2;
end

end
```

4.3 estimationangle

Questa funzione è stata fornita a priori dal relatore e consente di stimare gli angoli di arrivo del suono a fronte delle misurazione effettuate con i microfoni; necessita quindi in ingresso dei dati acquisiti precedentemente (x_1, x_2, x_3, x_4) e delle distanze fra i microfoni collegati allo stesso dispositivo (L_1, L_2).

Per prima cosa viene effettuata l'*autocorrelazione* fra tutti gli elementi dei 2 vettori dati contenenti le acquisizioni effettuate con 2 microfoni collegati alla stessa scheda *digilent*.

```
lungcorr=length(x1)+length(x2)-1;
N=pow2(nextpow2(lungcorr));
z=real(iff(fft(x1,N).*fft(x2(end:-1:1),N)));
```

Vengono poi definiti i parametri che verranno utilizzati nelle prossime istruzioni: la frequenza di campionamento (fs), la velocità del suono (v) e il reciproco della frequenza di campionamento, ovvero il tempo che intercorre fra l'acquisizione di un campione e l'altro (dt).

```
v=340.29;
fs=44.1e3;
dt=1/fs;
```

A questo punto stimando il picco della funzione di *cross correlazione* è possibile ricavare il tempo che intercorre fra l'istante in cui il suono giunge ad un microfono e l'altro, quindi tramite calcoli trigonometrici si ricava l'angolo di arrivo del suono in radianti, poi convertito in gradi.

```
[~,locmax]=max(abs(z));
delay=(locmax-length(x1))*dt;
thetarad=asin(v*delay/L1);
thetadeg=thetarad/pi*180;
theta1 = thetadeg;
```

Infine si ripetono gli stessi passaggi anche per l'altra coppia di microfoni così da ottenere la stima di entrambi gli angoli.

```
lungcorr2=length(x3)+length(x4)-1;
N2=pow2(nextpow2(lungcorr2));
z2=real(ifft(fft(x3,N2).*fft(x4(end:-1:1),N2)));
[~,locmax2]=max(abs(z2));
delay2=(locmax2-length(x3))*dt;
thetarad2=asin(v*delay2/L2);
thetadeg2=thetarad2/pi*180;
theta2 = thetadeg2;
end
```

4.4 pos

La funzione *pos* si occupa di calcolare la posizione da cui proviene il suono, a partire dai 2 angoli forniti dalla funzione *estimationangle*; necessita quindi in ingresso dei due angoli stimati in precedenza (*theta1*, *theta2*) e delle coordinate x-y dei singoli microfoni (*x1*, *x2*, *x3*, *x4*, *y1*, *y2*, *y3*, *y4*). La funzione restituisce in uscita le coordinate x-y della posizione stimata (*x*, *y*).

Viene scelto come punto di coordinate (0,0) il punto estremo in basso a sinistra del landscape (ambiente) in cui si stà effettuando la misura (N.B. *tutte le coordinate delle posizioni dei microfoni andranno rapportate a tale riferimento*); dopo aver posizionato i microfoni, vengono calcolati i coefficienti angolari delle 2 rette passanti per i 2 microfoni appartenenti a ciascuna coppia (*m12*, *m34*).

```
m12 = (y2-y1)/(x2-x1);
m34 = (y4-y3)/(x4-x3);
```

Si ricavano poi gli angoli corrispondenti a tali coefficienti angolari (in radianti) e si convertono anche quelli forniti in ingresso della funzione *estimationangle*, da gradi a radianti.

```
thetarif12 = atan(m12); %In radianti
thetarif34 = atan(m34); %In radianti
theta1rad = theta1*pi/180;
theta2rad = theta2*pi/180;
```

A questo punto è possibile calcolare l'angolo di interesse per entrambe le coppie di microfoni, ovvero quello compreso fra la retta orizzontale del sistema di riferimento e le rette ricavate dalla stima degli angoli di arrivo del suono; è quindi possibile calcolare anche i coefficienti angolari delle 2 rette con tali inclinazioni.

```
angle1 = pi/2+thetarif12-theta1rad;
angle2 = pi/2+thetarif34-theta2rad;
m1 = tan(angle1);
m2 = tan(angle2);
```

Si ricavano quindi le coordinate dei punti medi delle 2 coppie di microfoni (punti per cui sicuramente passeranno le 2 rette necessarie per la stima della posizione).

```
pmx12 = (x2+x1)/2;
pmx34 = (x4+x3)/2;
pmy12 = (y2+y1)/2;
pmy34 = (y4+y3)/2;
```

Ora avendo per entrambe le rette il coefficiente angolare e un punto, è possibile scrivere un sistema in 2 equazioni (le equazioni delle due rette con coefficienti angolari fissati e passanti per il punto medio della relativa coppia di microfoni) e 2 incognite (le coordinate (x,y) in cui le rette si intersecano) che risolvendolo, restituisce il punto stimato di provenienza del suono.

```
x = (pmy34-pmy12+(pmx12*m1)-(pmx34*m2))/(m1-m2);
y = pmy12+(m1*(x-pmx12));
```

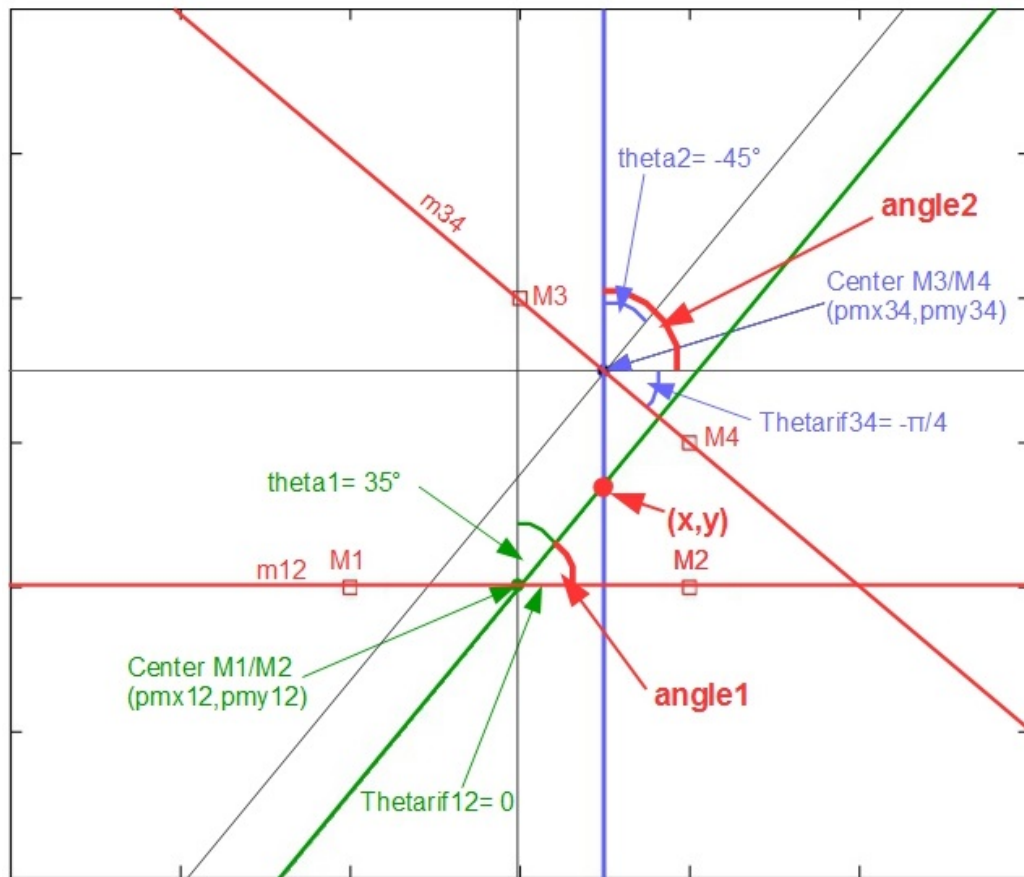


Figura 4.1: Chiamata della funzione $pos(35,-45,2,4,3,4,2,2,4,3)$

4.5 test_offline

La funzione *test_offline* consente di effettuare la stima della posizione di provenienza del suono a partire da 4 file *wav* contenenti le registrazioni di 4 microfoni effettuate a priori.

I 4 file *wav* vengono caricati in 4 vettori e si ricavano con la stessa istruzione le informazioni inerenti alla frequenza di campionamento (fs) e al numero di bit ($nbits$).

```
[dataAD11]=wavread('outputAD11.wav');
[dataAD12]=wavread('outputAD12.wav');
[dataAD21]=wavread('outputAD21.wav');
[dataAD22,fs,nbits]=wavread('outputAD22.wav');
```

Per stimare la posizione a partire dai 4 vettori così definiti è sufficiente chiamare le funzioni già descritte in precedenza

```
[theta1,theta2] = estimationangle(dataAD11, dataAD12,...  
dataAD21, dataAD22, LAD1, LAD2);  
[x, y] = pos(theta1, theta2, x1, x2, x3, x4, y1, y2, y3, y4);
```


Capitolo 5

GUI MATLAB

La GUI di MATLAB [9] è uno strumento per creare graficamente finestre interattive. Con il comando *guide* si accede al tool grafico che consente di creare interfacce oppure modificare modelli creati in precedenza o predefiniti. Una volta creata la finestra graficamente (cui è associato un file *.fig*), viene creato automaticamente un secondo file MATLAB (*.m*), che contiene delle funzioni di Callback per ogni item generato (per item si intendono finestre, pulsanti, slider, ecc.). Attraverso tali funzioni è possibile definire quali istruzioni dovrà eseguire il programma nel momento in cui l'item, a cui è associata tale funzione, verrà interpellato. Quando le istruzioni associate all'item terminano, il programma si mette in attesa della chiamata di un'altra Callback.

5.1 Interfaccia grafica

L'interfaccia grafica creata per l'applicazione in esame, è mostrata nella *figura 5.1*. La finestra presenta:

- 6 bottoni interattivi le cui funzioni verranno discuse nella prossima sezione
- Il pannello *Microphone* in cui inserire le coordinate delle posizioni dei 4 microfoni
- Il pannello *Landscape* in cui inserire le dimensioni dello spazio ove si effettua la misura e il numero di acquisizioni
- Il pannello *Results* che contiene le coordinate della posizione stimata

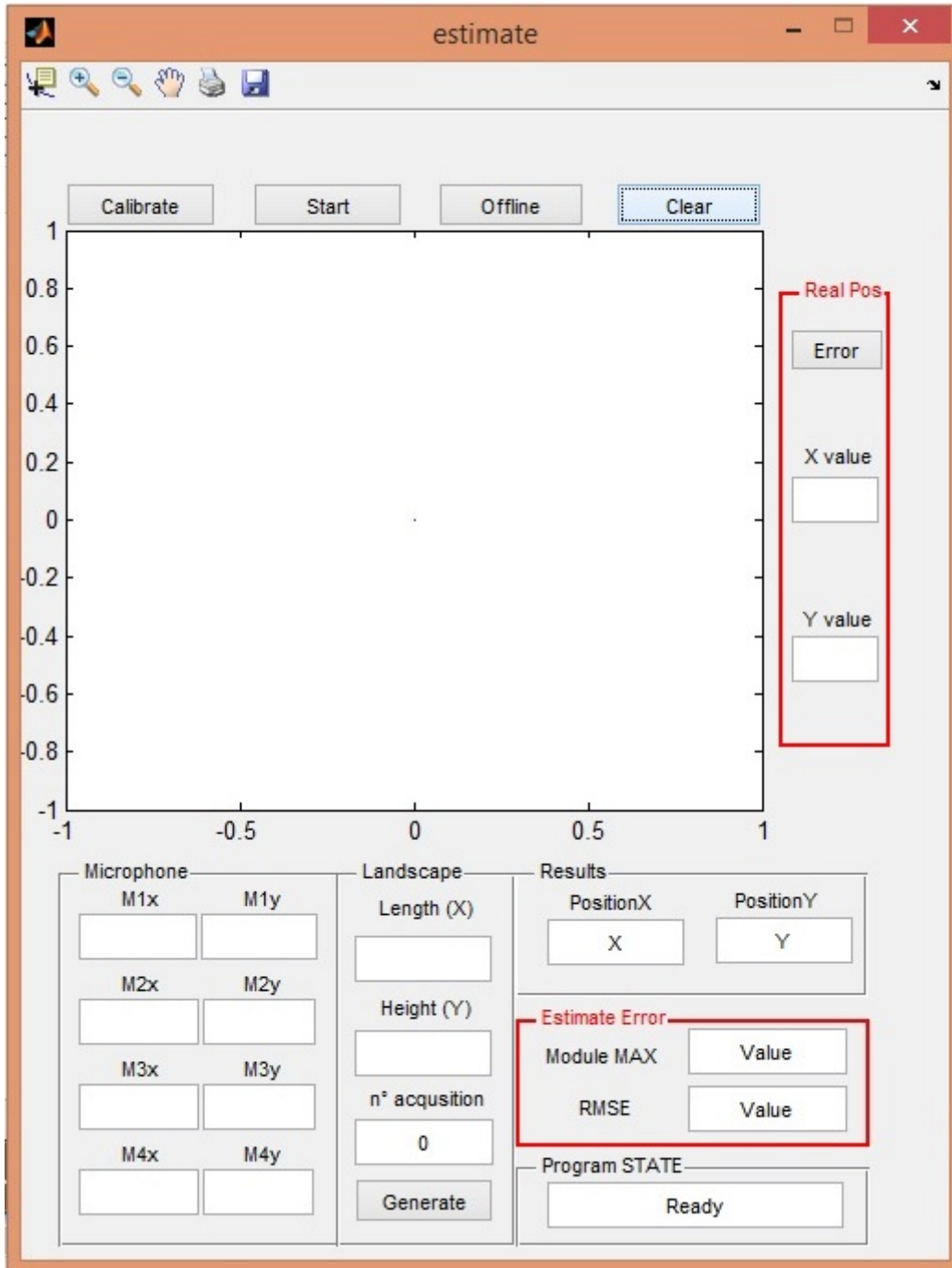


Figura 5.1: interfaccia grafica

- Il pannello *Estimate Error* contenente l'errore massimo commesso e il *root mean square error* (RMSE) stimato durante una campagna di misure a posizione nota
- Il pannello *Real Pos* in cui inserire la posizione da cui proviene il suono (nota a priori)
- La barra *program STATE* che contiene lo stato del programma (*Ready* o *Busy*)

5.2 Funzioni GUI

Qui di seguito verranno discusse e commentate tutte le funzioni associate alle varie Callback, generate automaticamente dopo aver creato l'interfaccia grafica (N.B. *le linee di comando non commentate sono quelle generate da MATLAB e vengono descritte automaticamente all'interno del programma*).

5.2.1 estimate_OpeningFcn

L'*OpeningFcn* contiene tutte le istruzioni da eseguire automaticamente all'apertura della finestra. Per poter mantenere in memoria variabili che dovranno passare da una Callback all'altra, è necessario inizializzare in questa sezione tali variabili (altrimenti al momento della chiusura della Callback che le ha generate verranno cancellate).

Si inizializzano quindi le variabili di offset che, dopo esser state calcolate mediante la Callback *Calibrate*, dovranno rimanere in memoria per poter essere utilizzate da tutte le altre Callback.

```
function estimate\_OpeningFcn(hObject, eventdata, handles,...  
varargin)
```

```
handles.offsetAD11=0;  
handles.offsetAD12=0;  
handles.offsetAD21=0;  
handles.offsetAD22=0;
```

```
handles.output = hObject;  
guidata(hObject, handles);
```

5.2.2 Calibrate_Callback

Questa Callback è associata al tasto *Calibrate* e si occupa di calcolare i valori di offset dei vari canali di acquisizione attraverso una misura da effettuare in totale silenzio.

Per prima cosa si segnala che il programma è occupato ad effettuare una misura scrivendo nella barra *Program State* lo stato *Busy*.

```
function Calibrate_Callback(hObject, eventdata, handles)
set(handles.editProgram_STATE, 'string', 'Busy');
```

Si calcola poi l'offset con le funzioni *offsetAD1* e *offsetAD2* descritte nel capitolo precedente e si aggiornano le variabili *handles* per poter utilizzare i valori ricavati nelle prossime Callback.

```
[offsetAD11, offsetAD12] = offsetAD1;
[offsetAD21, offsetAD22] = offsetAD2;
handles.offsetAD11=offsetAD11;
handles.offsetAD12=offsetAD12;
handles.offsetAD21=offsetAD21;
handles.offsetAD22=offsetAD22;
```

Infine si segnala che il programma ha terminato l'esecuzione delle istruzioni ed è pronto a processarne di nuove.

```
set(handles.editProgram_STATE, 'string', 'Ready');
```

5.2.3 Generate_Landscape_Callback

Generate_Landscape si occupa di mostrare nel grafico lo scenario impostato attraverso gli specchietti di testo editabile, una volta inserite quindi le coordinate dei 4 microfoni e le dimensioni dell'ambiente, in cui avverranno le misure, questi verranno visualizzati nel grafico al centro della finestra.

Per prima cosa vengono letti i valori inseriti nei riquadri di testo editabile.

```
function Generate_Landscape_Callback(hObject, eventdata, handles)

M1x = str2double(get(handles.editM1x, 'string'));
M2x = str2double(get(handles.editM2x, 'string'));
M3x = str2double(get(handles.editM3x, 'string'));
```

```

M4x = str2double(get(handles.editM4x,'string'));
M1y = str2double(get(handles.editM1y,'string'));
M2y = str2double(get(handles.editM2y,'string'));
M3y = str2double(get(handles.editM3y,'string'));
M4y = str2double(get(handles.editM4y,'string'));
LENGTH = str2double(get(handles.editlength,'string'));
HEIGHT = str2double(get(handles.editheight,'string'));

```

Per mantenere le proporzioni fra altezza e larghezza del landscape, viene controllato quale dei 2 valori è maggiore; successivamente si impostano i limiti degli assi su tale valore, aumentato di un margine pari a 1/10 del valore stesso, per lasciare un piccolo bordo estetico ai lati del landscape.

```

if LENGTH>HEIGHT+HEIGHT/10;
    MAX=LENGTH+LENGTH/10;
    min=-LENGTH/10;
else
    MAX=HEIGHT+HEIGHT/10;
    min=-HEIGHT/10;
end

```

Non rimane che plottare tutti gli elementi, ovvero i 4 microfoni, i nomi vicino ad ognuno di essi e i 4 segmenti che delimitano il landscape (N.B. siccome la funzione *plot* risulta molto estesa, viene ripetutamente mandata a capo con i 3 punti).

```

plot([0,0],[0,HEIGHT], 'b',[0,LENGTH],[0,0],...
'b', [0,LENGTH],[HEIGHT,HEIGHT], 'b', [LENGTH,LENGTH],[0,HEIGHT],...
'b', M1x, M1y, 'rs', M2x, M2y, 'rs', M3x, M3y,...
'rs', M4x, M4y, 'rs'), axis([min MAX min MAX])

xlabel('X[m]')
ylabel('Y[m]')
text(M1x,M1y,' M1','Color','red','FontSize',8)
text(M2x,M2y,' M2','Color','red','FontSize',8)
text(M3x,M3y,' M3','Color','red','FontSize',8)
text(M4x,M4y,' M4','Color','red','FontSize',8)

```

5.2.4 Start_Callback

La funzione *Start_Callback* è quella che si occupa di effettuare le stime vere e proprie della posizione della sorgente acustica. Dopo aver fornito la posizione

dei microfoni, le dimensioni del landscape e il numero di acquisizioni che si intendono eseguire, premendo il tasto *Start* si avvia un numero di stime real time, pari al numero indicato nella casella *n acquisizioni*.

Al solito si leggono i valori contenuti nelle caselle di testo editabile e si impostano i limiti degli assi.

```
function Start_Callback(hObject, eventdata, handles)

M1x = str2double(get(handles.editM1x,'string'));
M2x = str2double(get(handles.editM2x,'string'));
M3x = str2double(get(handles.editM3x,'string'));
M4x = str2double(get(handles.editM4x,'string'));
M1y = str2double(get(handles.editM1y,'string'));
M2y = str2double(get(handles.editM2y,'string'));
M3y = str2double(get(handles.editM3y,'string'));
M4y = str2double(get(handles.editM4y,'string'));
LENGTH = str2double(get(handles.editlength,'string'));
HEIGHT = str2double(get(handles.editheight,'string'));
n = str2double(get(handles.editnacq,'string'));
if LENGTH>HEIGHT
    MAX=LENGTH+LENGTH/10;
    min=-LENGTH/10;
else
    MAX=HEIGHT+HEIGHT/10;
    min=-HEIGHT/10;
end
```

Si calcola poi la distanza cartesiana fra i microfoni di ogni coppia, da fornire successivamente in ingresso alla funzione di stima dell'angolo di arrivo.

```
LAD1=sqrt((M1x-M2x)^2+(M1y-M2y)^2);
LAD2=sqrt((M3x-M4x)^2+(M3y-M4y)^2);
```

A questo punto si segnala che il programma è occupato ad effettuare le misurazioni e tramite un ciclo *while* si effettua il numero di stime richieste. All'interno del ciclo *while* oltre alle funzioni di acquisizione, stima dell'angolo di arrivo, stima della posizione e *plotting* dei valori ottenuti, è presente anche un controllo sulla posizione stimata, che verrà plottata solo se rientra nei limiti del landscape, altrimenti verrà visualizzato sul grafico il messaggio *SIGNAL LOST*. Prima di uscire dal ciclo *while* vengono aggiornate le caselle

contenenti le coordinate della posizione stimata e viene decrementato l'indice di conteggio del numero di misure mancanti al termine del processo, cosicché possa poi essere riportato nella casella di testo editabile *n acquisizioni*. Infine si segnala nuovamente che il programma è pronto per effettuare una nuova misura (*Ready*)

```

set(handles.editProgram_STATE,'string','Busy');
while n>0
    [dataAD11, dataAD12] = acqAD1(handles.offsetAD11,...
    handles.offsetAD12);
    [dataAD21, dataAD22] = acqAD2(handles.offsetAD21,...
    handles.offsetAD22);
    LAD1=sqrt((M1x-M2x)^2+(M1y-M2y)^2);
    LAD2=sqrt((M3x-M4x)^2+(M3y-M4y)^2);
    [theta1,theta2] = estimationangle(dataAD11, dataAD12,...
    dataAD21, dataAD22, LAD1, LAD2);
    [x, y] = pos(theta1, theta2, M1x, M2x, M3x, M4x,...
    M1y, M2y, M3y, M4y);

    if x<HEIGHT && x>0 && y<LENGTH && y>0
        plot(x, y, 'bo', [0,0],[0,HEIGHT],...
        'b', [0,LENGTH],[0,0], 'b', [0,LENGTH],[HEIGHT,HEIGHT],...
        'b', [LENGTH,LENGTH],[0,HEIGHT], 'b', M1x, M1y, 'rs',...
        M2x, M2y, 'rs', M3x, M3y, 'rs', M4x, M4y, 'rs'),...
        axis([min MAX min MAX])
        xlabel('X[m]')
        ylabel('Y[m]')
        text(M1x,M1y,' M1','Color','red','FontSize',8)
        text(M2x,M2y,' M2','Color','red','FontSize',8)
        text(M3x,M3y,' M3','Color','red','FontSize',8)
        text(M4x,M4y,' M4','Color','red','FontSize',8)
    else
        plot([0,0],[0,HEIGHT], 'b', [0,LENGTH],[0,0],...
        'b', [0,LENGTH],[HEIGHT,HEIGHT],...
        'b', [LENGTH,LENGTH],[0,HEIGHT],...
        'b', M1x, M1y, 'rs', M2x, M2y, 'rs',M3x, M3y, 'rs',...
        M4x, M4y, 'rs'), axis([min MAX min MAX])
        xlabel('X[m]')
        ylabel('Y[m]')
        text(M1x,M1y,' M1','Color','red','FontSize',8)
        text(M2x,M2y,' M2','Color','red','FontSize',8)

```

```
    text(M3x,M3y,' M3','Color','red','FontSize',8)
    text(M4x,M4y,' M4','Color','red','FontSize',8)
    text(0,min/2,'SIGNAL LOST','Color','red','FontSize',10)
end

set(handles.editpositionx,'string',x);
set(handles.editpositiony,'string',y);
n=n-1;
set(handles.editnacq,'string',n);
end

set(handles.editProgram_STATE,'string','Ready');
```

5.3 Funzioni aggiuntive

Benchè le funzioni analizzate fino ad ora siano sufficienti per effettuare la stima della posizione, sono state aggiunte altre 3 funzioni per rendere più agevoli le misurazioni e per fornire alcune informazioni aggiuntive sulla qualità delle stime effettuate. Per evitare ridondanza, di queste funzioni saranno enunciate solo le differenze con quelle precedentemente definite e verranno analizzate le relative funzionalità.

5.3.1 Offline_Callback

La funzione è del tutto analoga alla *Start_Callback*, con l'unica differenza che la stima viene effettuata una sola volta (non serve quindi il ciclo *while*) e che i dati vengono caricati da 4 file *wav* (con la funzione vista nel precedente capitolo) anzichè essere direttamente acquisiti. Questa funzione è pensata per effettuare una campagna di misure *offline* e in un secondo momento effettuarne la stima.

5.3.2 Clear_Callback

Il tasto *Clear*, cui è associata la Callback in analisi, consente di riportare la finestra allo stato in cui si trovava al momento dell'apertura, ovvero grafico vuoto e caselle editabili nella configurazione di *default*.

5.3.3 estimate_error_Callback

A questa Callback sono stati riservati i due pannelli rossi presenti nella GUI. La funzione serve ad ottenere un feedback sulla misura effettuata. Conoscendo a priori la posizione di origine della sorgente acustica, la funzione *estimate_error_Callback*, cui è associato il tasto *Error*, oltre ad effettuare il numero di stime indicato dall'utente al termine della sessione plotta nuovamente tutte le misure effettuate evidenziando la posizione effettiva della sorgente acustica (inserita a priori dall'utente mediante gli appositi box editabili) e riporta nel pannello *Estimate Error* le seguenti informazioni:

- *Module MAX*, ovvero l'errore massimo commesso durante la sessione di misure effettuata
- *Root Mean Square Error (RMSE)*, che rappresenta la media degli errori commessi durante la sessione di misure secondo la seguente espressione:

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (x_i - x)^2 + (y_i - y)^2} \quad (5.1)$$

n = numero di misure effettuate

x_i = coordinata x stimata dalla misura i -esima

y_i = coordinata y stimata dalla misura i -esima

x = coordinata x della posizione reale

y = coordinata y della posizione reale

Le misure che finiscono al di fuori dal landscape non vengono considerate nella stima del *root mean square error*, al termine della serie di misure viene riportato nel box *n acquisizioni* il numero di stime finite all'interno del landscape.

Capitolo 6

Conclusioni

La *figura 6.1* si riferisce ad una serie di misure effettuate in uno scantinato dalle dimensioni ridotte e con un alto tasso di riverbero. Come si può notare dall'immagine la stima non è ottimale, per migliorare il risultato delle stime si rimanda il lettore al capitolo “*1.3 condizioni ottimali di misura*”.

Per cercare di migliorare le prestazioni del sistema, si può lavorare ai seguenti aspetti:

- Dotare i microfoni di un sistema di amplificazione del segnale acquisito, in modo da aumentare notevolmente la qualità delle acquisizioni e la sensibilità dei microfoni.
- Utilizzare delle schede di acquisizione audio alternative e metterle a confronto con gli *Analog Discovery*.
- Utilizzare algoritmi alternativi per stimare la posizione dell'angolo, in sostituzione alla funzione di *cross-correlazione*, impiegando più di 2 microfoni.
- Utilizzare più di 2 coppie di microfoni, così da ottenere una stima più accurata.

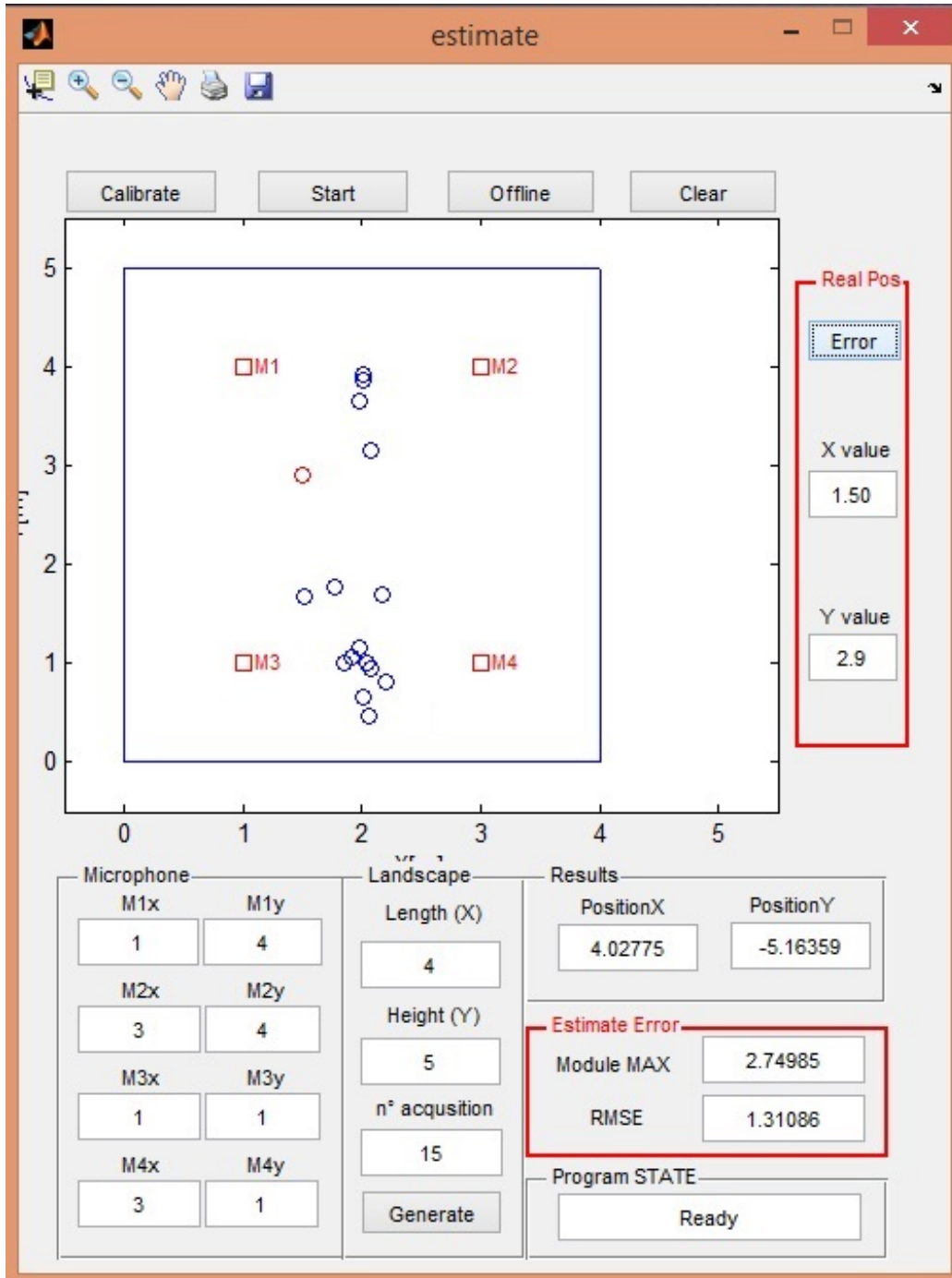


Figura 6.1: Stima dell'errore

Elenco delle figure

1.1	Sistema di acquisizione	3
1.2	Diagramma a Blocchi	4
1.3	Punti critici di misura	6
2.1	Sensore audio	7
2.2	Capsule microfoniche [3]	8
2.3	Schema Filtro	9
2.4	Footprint Sensore	10
3.1	Analog Discovery [4]	11
3.2	PIN I/O Analog Discovery [6]	12
3.3	Waveform Digilent schermata iniziale	13
3.4	Waveform Scope	14
3.5	Waveform WaveGen	15
3.6	Waveform Voltage	16
4.1	Chiamata della funzione $pos(35,-45,2,4,3,4,2,2,4,3)$	24
5.1	interfaccia grafica	28
6.1	Stima dell'errore	38

Bibliografia

[1] V3recording

<http://www.v3recording.com/microfoni-tipologie-caratteristiche/>

[2] Capsule microfoniche RS

<http://docs-europe.electrocomponents.com/webdocs/141f/0900766b8141f86c.pdf>

[3] RS Microphone

<http://it.rs-online.com/largeimages/F7800728-01.jpg>

[4] Kamai

http://kamami.com/published/publicdata/BTC10/attachments/SC/products_pictures/digilent_analog_discovery_cont.jpg

[5] Digilent Analog Discovery

<http://www.digilentinc.com/Products/Detail.cfm?NavPath=2,842,1018&Prod=ANALOG-DISCOVERY&CFID=17548965&CFTOKEN=2fee1c74d03c337d-04D02043-5056-0201-023658403CA5B92F>

[6] Coolcapangineer

<https://coolcapengineer.files.wordpress.com/2013/06/pinoutanalog.jpg>
grossmembranmikrofon/

[7] Getting started acquiring data with Analog Discovery

<http://it.mathworks.com/help/daq/examples/getting-started-acquiring-data-with-digilent-analog-discovery.html>

- [8] Vendor limitation

http://it.mathworks.com/help/daq/digilent-analog-discovery-devices_bt0oih4-1.html

- [9] GUI MATLAB

<http://it.mathworks.com/discovery/matlab-gui.html>

- [10] Calandrino, Chiani: *Lezioni di comunicazioni elettriche* (2013)

- [11] Mathworks - Guida a MATLAB

http://it.mathworks.com/academia/student_center/tutorials/launchpad.html

- [12] Scheda tecnica Analog Discovery

https://www.digilentinc.com/Data/Products/ANALOG-DISCOVERY/Discovery_TRM_RevB_1.pdf

- [13] Mathworks - Guida Analog Discovery

<http://it.mathworks.com/hardware-support/digilent-analog-discovery.html>

Ringraziamenti

Un sentito ringraziamento a tutti coloro che mi hanno aiutato nella realizzazione di questo elaborato.

Ringrazio il professore relatore Andrea Giorgetti, per la pazienza, la disponibilità e per gli aiuti e i consigli fornitimi.

Ringrazio i miei amici e parenti per il loro supporto e la loro presenza che non è mai mancata.

Ringrazio infine la mia famiglia, mia madre, mio padre e mia sorella per la forza che mi infondono e senza i quali nulla sarei.

...illumina l'oscurità

