

Università degli Studi di Padova



Facoltà di Ingegneria

Corso di Laurea in Bioingegneria

**SVILUPPO DI UN SISTEMA SOFTWARE
PER LA RILEVAZIONE E ANALISI DI LO-
CAL FIELD POTENTIALS NEL MODELLO
ANIMALE**

Relatore: Ch.mo Prof. Giovanni Sparacino

Correlatori: Ing. Maria Rubega

Ing. Guglielmo Frigo

Laureando: Giacomo Lezziero

Anno Accademico 2014-2015

RINGRAZIAMENTI

Desidero ringraziare il professore Sparacino e il professore Vassanelli per avermi dato la possibilità di collaborare con il NeuroChip Laboratory.

Ringrazio la dottoranda Maria Rubega per essere stata sempre disponibile ed avermi aiutato nella realizzazione di questa tesi.

Ringrazio il dottorando Guglielmo Frigo e il laboratorio di misure elettroniche, per avermi permesso di accedere a tutti gli strumenti necessari per realizzare il software in LabVIEW.

Ringrazio anche Roberto Fontana e Claudia Cechetto, per avermi permesso di testare il software in esperimenti reali presso il NeuroChip Laboratory.

INDICE

1. LOCAL FIELD POTENTIAL NELLA RICERCA IN NEUROFISIOLOGIA	7
1.1 Definizione di Local Field Potential (LFP) e meccanismi di generazione	7
1.2 LFP nel modello animale	10
1.3 Uso degli LFP nella ricerca	12
1.4 Analisi dei segnali LFP	13
1.4.1 Problemi di interpretazione del segnale LFP	13
1.4.2 Analisi in frequenza del segnale LFP	14
1.4.3 Analisi tramite Current Source Density	17
1.5 Sistemi per l'acquisizione di LFP	18
1.5.1 Tipologia sensori	18
1.5.2 Microelettrodi (micropipette)	18
1.5.3 EOSFET	19
1.5.4 Software	19
1.6 Scopo della tesi	20
2. PROGETTAZIONE DEL SOFTWARE	21
2.1 Analisi dei requisiti	21
2.2 Scelte procedurali	22
2.3 L'ambiente LabVIEW	22
3. REALIZZAZIONE DEL SOFTWARE DI ACQUISIZIONE	25
3.1 L'interfaccia utente	25
3.1.1 Configurazione parametri	27
3.1.2 Visualizzazione dati acquisiti	27
3.2 Tipologia di acquisizione	30
3.3 Gestione buffer	31
3.4 Stimolazione	34
3.5 Architettura multicore	35
3.6 Elaborazione real-time	36
3.7 Salvataggio dati	38
3.8 Elaborazione post acquisizione	40
4. TEST DEL SOFTWARE	43
4.1 Setup sperimentale	43
4.2 Procedura acquisizione	44
4.2.1 Acquisizione continua	44
4.2.2 Acquisizione a finestre	45
4.3 Bilancio del test	47

5. CONCLUSIONI.....	49
5.1 Bilancio del lavoro svolto.....	49
5.2 Sviluppi futuri.....	50
BIBLIOGRAFIA E SITOGRAFIA	53
Libri e articoli	53

INDICE FIGURE, TABELLE E GRAFICI

Elenco delle tabelle

Elenco delle figure

Figura 1: Segnale LFP registrato dall'ippocampo del topo	7
Figura 2: Esempio di pattern di correnti entranti ed uscenti in una popolazione di neuroni.....	9
Figura 3: Anatomia dell'Ippocampo di ratto [8].....	11
Figura 4: Schema del modello SPEAR [8]	11
Figura 5: a Informazione trasportata alle varie frequenze.....	15
Figura 6: a Un esempio di un segnale contenente un sharp wave-ripple complex (SPW-R)	16
Figura 7: Interfaccia grafica. Configurazione parametri.....	25
Figura 8: Interfaccia grafica. Visualizzazione grafici	26
Figura 9: Interfaccia grafica: parametri aggiuntivi	28
Figura 10: Finestra per visualizzare ed eliminare le finestre acquisite	28
Figura 11: Gestione pulsante "Start".....	29
Figura 12: Codice per il controllo dei parametri dell'interfaccia grafica.....	29
Figura 13: Acquisizione dati da scheda e memorizzazione nella coda FIFO	30
Figura 14: Buffer lunghezza fissa	32
Figura 15: Buffer circolare.....	32
Figura 16: Schema funzionamento algoritmo "min-max decimation" [27].....	33
Figura 17: Implementazione min-max decimation.....	33
Figura 18: Codice creazione canale output	34
Figura 19: Struttura cicli while	35
Figura 20: Calcolo coefficienti filtro.....	36
Figura 21: Filtraggio dati acquisiti	36
Figura 22: Codice media mobile	37
Figura 23: Uso funzione "DAQmx Configure Logging"	38
Figura 24: Esempio salvataggio dati nel caso di acquisizione a finestre	38
Figura 25: Codice conversione da tdms a mat	39
Figura 26: Esempio di selezione di una porzione di segnale acquisito.....	40
Figura 27: Spettro della sotto-finestra di Figura 27	40
Figura 28: Codice per la gestione della sotto- finestra.....	41
Figura 29: Gestione dell'evento associato alla variabile globale.....	41
Figura 30: Uso subVI Figura 29	42
Figura 31: Codice per la lettura dinamica del file tdms	42
Figura 32: Esempio segnale con una finestra di 30 secondi	44
Figura 33: Esempio di spettro del segnale di Figura 33 nel range 5-200 Hz.....	45
Figura 34: Esempio di acquisizione a finestre con durata di 2 secondi	46
Figura 35: Esempio filtraggio segnale di Figura 25 nel range 1-100 Hz.....	46

Figura 36: Esempio di media di 20 finestre 46

1. LOCAL FIELD POTENTIAL NELLA RICERCA IN NEUROFISIOLOGIA

1.1 Definizione di Local Field Potential (LFP) e meccanismi di generazione

I Local Field Potential (LFP) sono potenziali extracellulari acquisibili nella corteccia cerebrale e caratterizzati da frequenze fino a 500 Hz (Figura 1). Nella ricerca in neurofisiologia l'analisi di questi segnali permette la valutazione dell'attività sinaptica di una popolazione di neuroni. Infatti, molti studi sia nel topo che nel macaco, hanno utilizzato questi potenziali per analizzare meccanismi della rete corticale coinvolti nei processi cognitivi [1], motori [2] e altri processi quali attenzione, memoria e percezione [3] [4] [5].

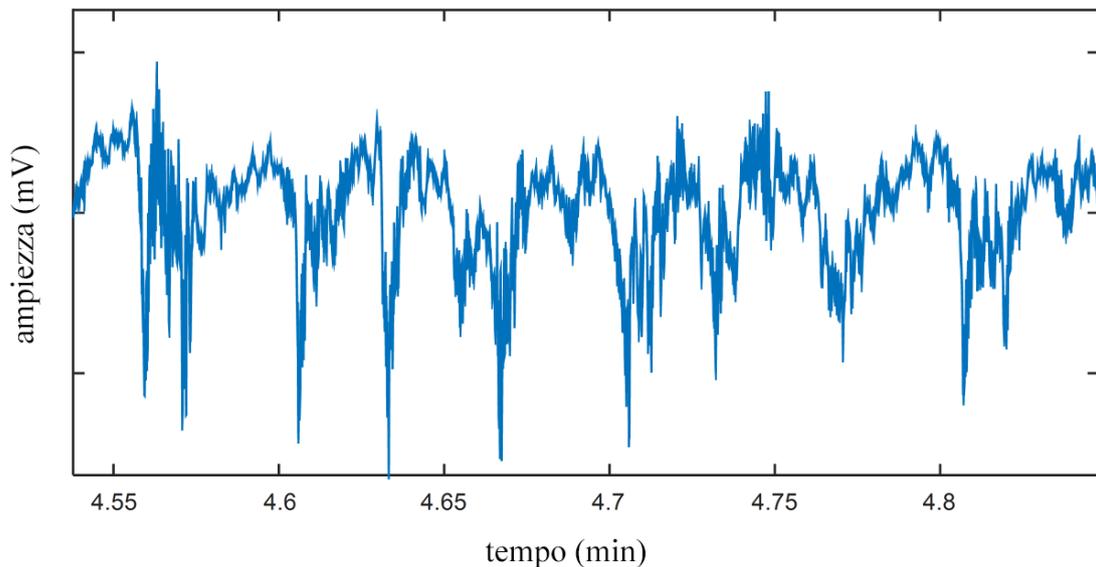


Figura 1: Segnale LFP registrato dall'ippocampo del topo

I potenziali extracellulari che danno origine al LFP nascono dalle correnti che attraversano la membrana cellulare. Il contributo delle correnti generate dai neuroni attivi si sovrappone per dare luogo ad un potenziale V_e , misurato rispetto ad un potenziale di riferimento, grazie a piccoli elettrodi nel cervello. Ogni membrana cellulare, sia essa localizzata a livello di soma o di dendrite, contribuisce a questo potenziale extracellulare. La forma e la frequenza del segnale LFP dipendono dal contributo di sorgenti multiple e dalle proprietà del tessuto cerebrale come illustrato di seguito. [5] [6]

Fattori quali la distanza dall'elettrodo, il numero di neuroni attivi e la loro forma, distribuzione spaziale e sincronia temporale, influenzano l'ampiezza del segnale LFP.

Un primo fattore che influenza l'ampiezza del segnale LFP è la distanza dall'elettrodo, infatti l'ampiezza di V_e diminuisce in modo inversamente proporzionale alla distanza da esso. L'ampiezza del segnale rilevato, oltre a dipendere dalla posizione dell'elettrodo, cresce con il numero di neuroni attivi, in quanto il flusso di correnti di ogni neurone si sovrappone. Sono, infatti, le correnti sinaptiche le componenti principali del segnale LFP. Dato che il flusso di correnti sinaptiche è un fenomeno relativamente lento, cresce la probabilità che il flusso di correnti si sovrapponga nel tempo. Si possono riconoscere due tipi di flussi nella membrana cellulare: quelli entranti dovuti alle correnti ioniche che entrano nella cellula e quelli uscenti dovuti alle correnti di verso opposto. Per mantenere il bilanciamento elettrico si genera un flusso di correnti dai flussi uscenti verso quelli entranti. Questa configurazione porta alla formazione di un dipolo elettrico, il quale contribuisce alla generazione del segnale LFP. Dalla Figura 2 si può vederne un esempio. Quando una popolazione di neuroni riceve uno stimolo sinaptico, si forma pattern di correnti entranti e uscenti (Figura 2a). Questo flusso di correnti si traduce in un LFP rilevato dall'elettrodo (Figura 2c). La Figura 2c mostra i colori utilizzati per rappresentare le correnti uscenti (source) ed entranti (sink).

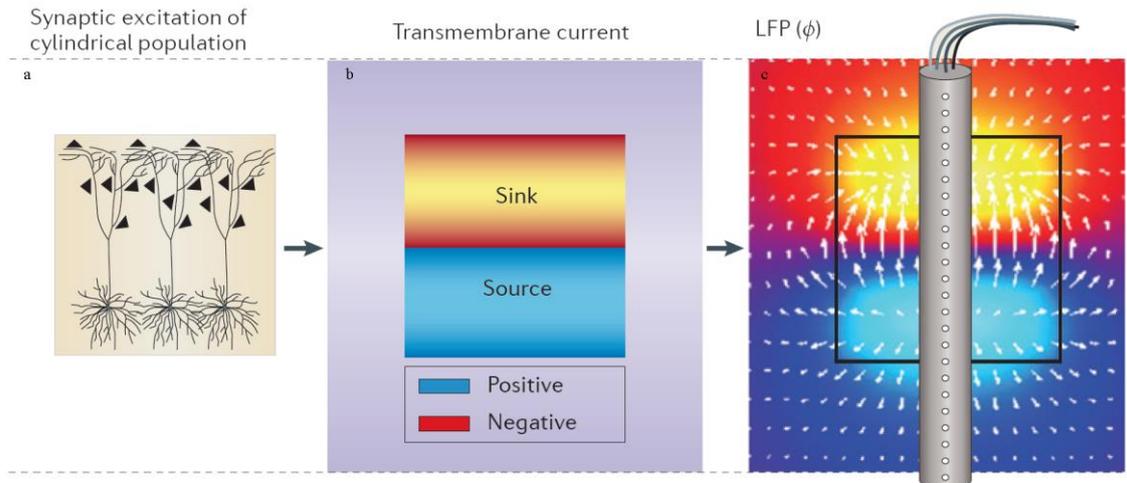


Figura 2: Esempio di pattern di correnti entranti ed uscenti in una popolazione di neuroni

Non tutti i neuroni contribuiscono al potenziale extracellulare in ugual misura: il loro contributo relativo è infatti influenzato dalla forma della cellula. In particolare, le cellule piramidali, le più numerose, presentano dei lunghi dendriti, i quali possono generare un dipolo elettrico particolarmente forte. Questi dipoli presentano una struttura ad anello aperto dovuta alla separazione spaziale tra sorgente e correnti di ritorno. In contrasto, i neuroni sfericamente simmetrici presentano una struttura ad anello chiuso. Assumendo una componente extracellulare omogenea, i due fattori più importanti che determinano l'intensità del campo elettrico di tali neuroni sono il loro allineamento spaziale e la loro sincronia temporale. Per esempio, i neuroni piramidali sono distesi parallelamente uno con l'altro. Questa conformazione è ideale per la sovrapposizione di dipoli attivi in sincronia, ed è questo il motivo per cui i segnali LFP hanno un'ampiezza elevata nella corteccia. Ma l'ampiezza finale del segnale LFP è anche influenzata dalla sincronia temporale dell'aggregato di neuroni e dalla conduttività del volume di tessuto cerebrale di interesse. Quest'ultimo determina l'intensità del LFP rilevato in un punto diverso rispetto a quello di generazione. Infatti, considerando la conduttività del tessuto cerebrale puramente ohmica si può osservare come V_e dipenda dall'intensità, dalla posizione e dalla conduttività del materiale.

Si è anche osservato che il segnale LFP varia l'intensità della sua ampiezza nella sinapsi apicale rispetto a quella registrata vicino al soma. La prima presenta infatti una traccia temporale meno ampia rispetto alla seconda. Si osserva quindi un fenomeno di filtraggio

temporale dovuto alla proprietà della struttura spaziale neuronale. Questo effetto, che attenua certe frequenze piuttosto che altre, si nota in tutti i segnali generati dalle correnti transmembrana. Nonostante siano le correnti transmembrana a dominare nel formare i segnali LFP, anche gli impulsi generati dalle sinapsi contribuiscono al segnale, come per esempio gli impulsi dovuti ai canali sodio o al calcio. Infatti, anche i potenziali d'azione generati da canali Na^+ nella membrana cellulare neuronale influenzano l'ampiezza del segnale LFP, in particolare l'ampiezza della componente ad alta frequenza. Questi potenziali generano delle grandi variazioni di V_e vicino al soma ma, a causa della loro breve durata (<2 ms), il contributo rimane confinato alle alte frequenze. Infine, va ricordato che la membrana presenta una proprietà di risonanza, infatti anche un debole input può produrre un'ampia risposta [5] [6].

1.2 LFP nel modello animale

Nei segnali LFP acquisiti da animali è importante definire gli intervalli di definizioni delle bande in frequenza, poiché sono diverse rispetto a quelle individuate nell'individuo umano. Per esempio, nel caso del ratto si possono distinguere due bande principali: banda theta da 6 a 10 Hz e banda gamma tra 40 e 100 Hz. Nel caso del topo si considera la banda theta da 4 a 12 Hz e la banda gamma da 30 a 80 Hz [7]. Dalle acquisizioni e dalle analisi dei segnali LFP acquisiti dall'ippocampo di ratti si osserva come le oscillazioni in banda theta siano consistenti durante la fase REM (Rapid Eye Movement) e durante le attività motorie. Le onde gamma sono legate alle oscillazioni più lente in banda theta, infatti, in mancanza di pattern in banda theta si osserva una diminuzione in potenza e regolarità delle oscillazioni in banda gamma. [7]

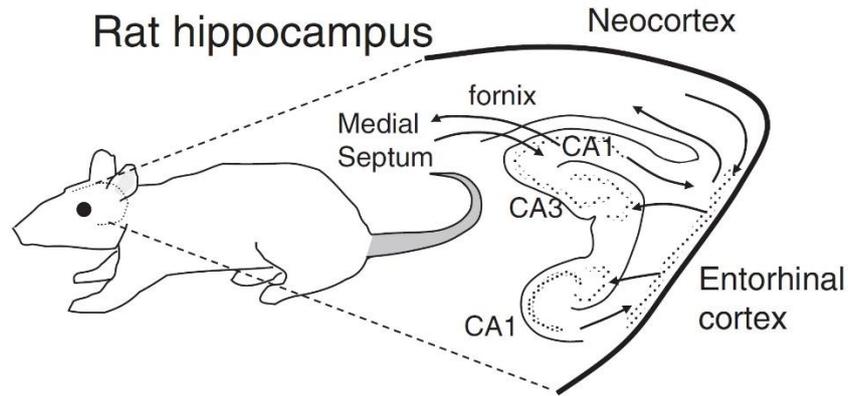


Figura 3: Anatomia dell'Ippocampo di ratto [8].

Si osserva anche una correlazione tra l'intensità delle onde theta e gamma. Ma i pattern nelle onde gamma si possono osservare anche in assenza di oscillazione in theta, suggerendo che sia presente un meccanismo simile per la generazione di onde theta e gamma. Inoltre, si è visto che esiste una correlazione tra l'area CA1 e CA3 in quanto, l'aumentare di attività sinaptica nell'area CA3 provoca un aumento dell'attività nell'area CA1 (Figura 3: Anatomia dell'Ippocampo di ratto.). In particolare, durante studi sulla memoria su ratti, si è visto come le onde gamma vengano generate in corrispondenza di una specifica fase delle onde theta. Un possibile modello per interpretare questo fenomeno è lo SPEAR model (Separate Phases of Encoding And Retrieval).

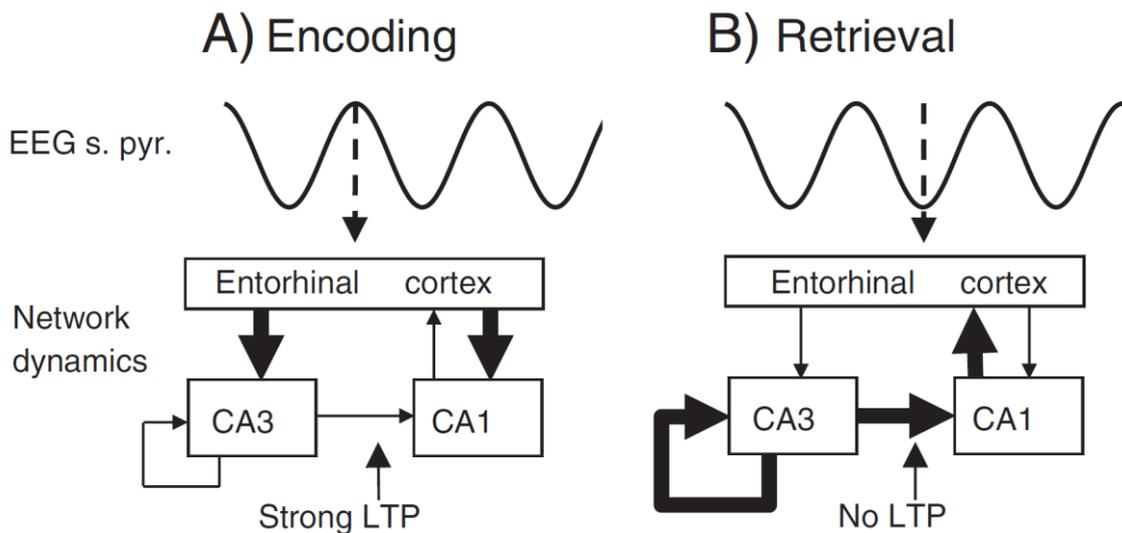


Figura 4: Schema del modello SPEAR [8]

La Figura 4 mostra lo schema del modello SPEAR. Il modello interpreta due fasi importanti, la prima valuta la memorizzazione delle informazioni da input esterni (encoding) e la seconda interpreta la fase di rimozione (decoding). Durante la fase di encoding si osserva un forte stimolo sinaptico dalla corteccia entorinale (frecce spesse), la trasmissione tra area CA3 a CA1 è debole e si verifica un potenziamento a lungo termine in modo da memorizzare le nuove informazioni. Nella fase di decoding si verifica una forte comunicazione tra le due aree ma un debole potenziamento a lungo termine al fine di impedire una nuova memorizzazione [8]. L'utilità di questi studi sulla memoria di ratti è emersa nello studio della malattia dell'Alzheimer. Si è osservato che in soggetti malati si verifica una alterazione nella relazione tra oscillazioni in onde theta e gamma. [9]

1.3 Uso degli LFP nella ricerca

Tra gli scopi della ricerca sugli LFP in neurofisiologia è comprendere come decifrare l'informazione trasmessa nel tessuto cerebrale dalle registrazioni dell'attività cellulare di decine o centinaia di neuroni contemporaneamente. Esistono due approcci per comprendere come il cervello decifri l'informazione trasmessa da una popolazione di neuroni. Il primo usa un algoritmo di decoding per predire uno stimolo da un pattern di risposte neuronali [10]. Il secondo utilizza i concetti dalla teoria dell'informazione per determinare quanta informazione è contenuta nello stimolo. [11]

I vantaggi del primo approccio (l'analisi di popolazione) sono molteplici [12]:

- l'informazione trasportata dalla popolazione di neuroni viene analizzata permettendo di determinare come i singoli membri hanno contribuito al segnale;
- il sistema di neuroni postsinaptici deve interpretare la risposta neuronale in un unico trial;
- le caratteristiche dello stimolo contenute nel treno di impulsi possono essere valutate applicando stimoli contenenti particolari caratteristiche;
- si può valutare come le diverse caratteristiche influenzino le performance di un algoritmo di decoding;

- l'informazione fornita da LFP o impulsi può essere combinata.

Si possono individuare quindi tre passi fondamentali. Prima di tutto si registra l'attività neuronale attraverso gli elettrodi. Poi, l'attività di ogni singolo neurone viene estratta dai dati. Infine, si interpretano i dati attraverso algoritmi di decoding e con la teoria dell'informazione. Gli algoritmi di decoding forniscono la predizione di quale stimolo o comportamento suscita una particolare risposta neuronale. Un esempio deriva dall'uso del decoding Bayesiano. Per validare i risultati alcune risposte sono usate come training set e il rimanente fornisce le performance dell'algoritmo.

Un altro potente metodo per studiare l'attività neuronale si basa sulla teoria di Shannon [12] [13]. Data la probabilità che un dato stimolo si presenti, si calcola l'entropia di Shannon, ossia la quantità di informazione o incertezza presente nel segnale in analisi. L'entropia vale zero solo se si presenta sempre lo stesso stimolo (massima certezza), altrimenti assume un valore positivo minore o uguale a 1. Se la risposta della popolazione dei neuroni contiene informazioni sullo stimolo, allora la sua osservazione ne riduce l'incertezza, minore è l'incertezza più il valore dell'entropia si avvicina al valore nullo.

Esistono dunque diversi metodi per l'analisi di popolazione da considerarsi complementari, infatti ognuno è in grado di interpretare un particolare aspetto del segnale acquisito. Molto spesso è quindi necessario affiancare le diverse metodologie per poter comprendere a fondo i meccanismi neurofisiologici. [12]

1.4 Analisi dei segnali LFP

1.4.1 Problemi di interpretazione del segnale LFP

Una problematica per l'interpretazione neurofisiologica dei LFP è l'estensione spaziale della zona che ha partecipato alla generazione del segnale, cruciale per determinarne il ruolo della popolazione di neuroni. Quando i neuroni ricevono un input sinaptico, il loro contributo si somma per produrre un LFP netto di popolazione. I neuroni più vicini all'elettrodo contribuiscono maggiormente al segnale rispetto a quelli più lontani. Tuttavia, per motivi geometrici la densità di neuroni aumenta linearmente all'aumentare della

distanza. Quindi il contributo relativo dipende da questi due effetti contrastanti. La situazione si differenzia ulteriormente se si considera che le sorgenti LFP possono essere correlate o incorrelate, mostrando rispettivamente interferenze costruttive o distruttive, influenzando quindi la distanza spaziale. Il livello di correlazione influenza il profilo dei segnali LFP: input sinaptici correlati generano un LFP che comprende tutta la popolazione di neuroni mentre input incorrelati danno luogo ad un LFP che è localizzato attorno all'input sinaptico. L'interpretazione di questi segnali risulta quindi complessa a causa della forte dipendenza con la correlazione degli input sinaptici e quindi con lo stimolo usato. [5] [6]

1.4.2 Analisi in frequenza del segnale LFP

Tradizionalmente, i segnali LFP vengono decomposti e analizzati nel dominio della frequenza attraverso l'analisi di Fourier [14]. Tuttavia, la definizione di bande di frequenza indipendenti è spesso arbitrario. Un metodo utilizzato è l'analisi della correlazione in ampiezza tra le varie bande. Se due bande sono indipendenti, ossia non correlate, allora probabilmente catturano due contributi differenti. Vengono utilizzati due tipi di correlazioni: la correlazione di segnale e di rumore. La correlazione di segnale riflette la differenza o similarità delle frequenze al variare delle condizioni esterne. La correlazione di rumore riflette sia la componente dovuta al rumore neuronale, cioè tutte quelle oscillazioni nel segnale LFP non associate ad una risposta dovuta ad uno stimolo, ma include anche l'effetto dei processi neuromodulatori dovuti alle alterazioni della trasmissione sinaptica tra neuroni. Questi due metodi non solo permettono di identificare il limite tra bande statisticamente indipendenti ma permettono di capire come le condizioni esterne si riflettano nell'attività di ogni singola componente del LFP. Molto spesso la separazione tra le varie bande non è netta. Un approccio consiste nell'utilizzo del metodo "Information maximizing band-separation method" [5] [15]. Questa procedura permette di identificare il confine che massimizza l'informazione che si ottiene separando lo spettro in varie bande.

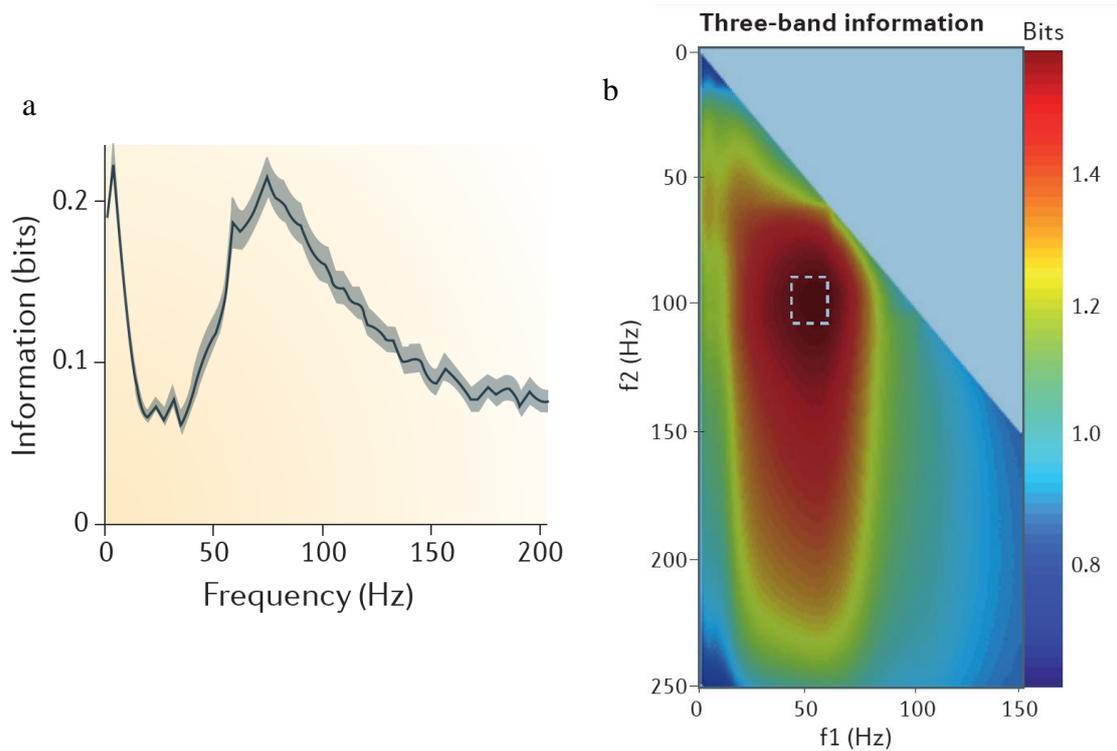
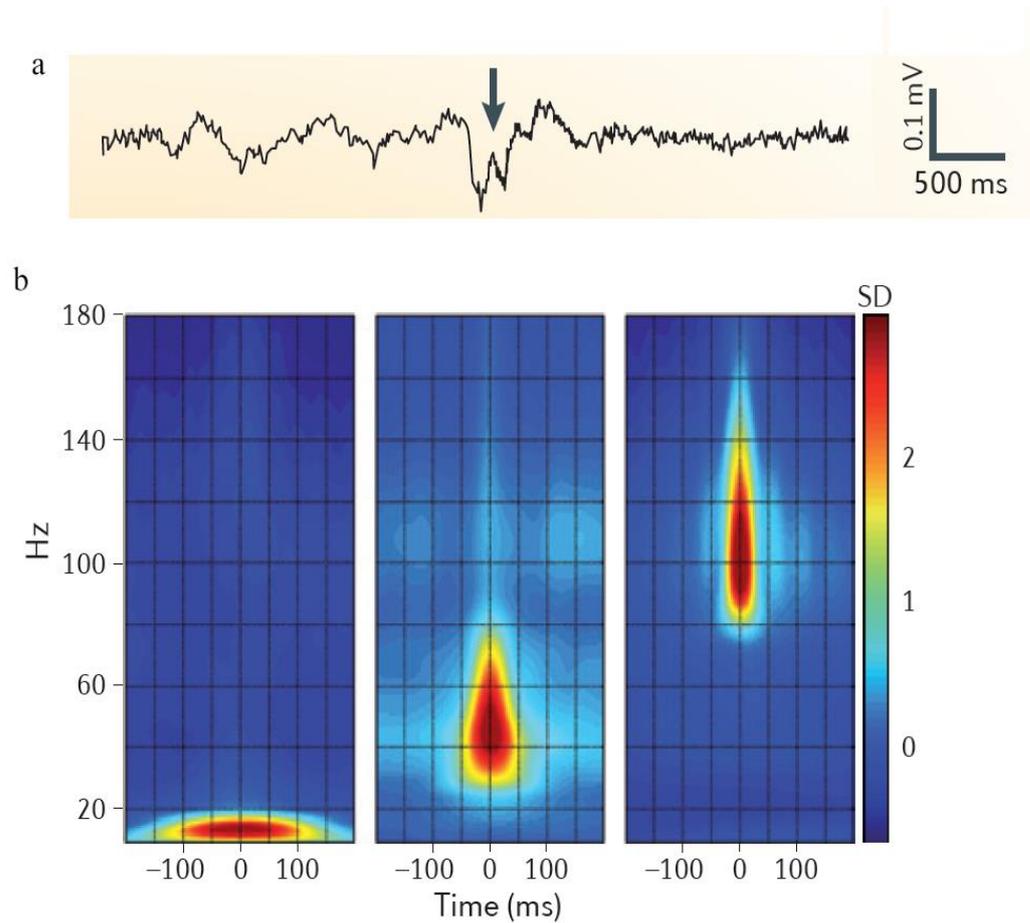


Figura 5: a / Informazione trasportata alle varie frequenze
 b / Separazione LFP in bande attraverso il metodo
 “Information maximizing band-separation method”

La Figura 5a mostra, in un esempio, la quantità di informazione trasportata dal segnale LFP alle varie frequenze. La Figura 5b presenta il risultato dell’applicazione del metodo “Information maximizing band-separation method”. Lo spettro di potenza del segnale LFP è analizzato nelle bande tra 0 Hz-f1, tra f1-f2 e tra f2-250 Hz e visualizzato in funzione di f1 e f2. Il rettangolo tratteggiato identifica i limiti entro i quali si massimizza l’informazione trasportata nelle varie bande.

Un limite dell’analisi di Fourier è che non permette di osservare la risposta transitoria ad un determinato stimolo. Per risolvere questo inconveniente si utilizza l’algoritmo “matching Pursuit” [5] [16] che permette di scomporre LFP in funzioni oscillatorie e transitorie. Per capire la funzione non è sufficiente analizzare le componenti nelle varie bande ma è necessario anche considerare il timing tra le diverse bande, ossia gli istanti di tempo nei quali si verifica una determinata condizione come, ad esempio, l’aumento di potenza nello spettro in una determinata banda. Un buon metodo è quello di clusterizzare le misure spettrali attorno ai valori di timing, in modo da raggruppare quelle componenti che presentano delle caratteristiche in comune. Riprendendo l’esempio precedente, si possono

ottenere diversi cluster contenenti quelle misure che presentano un aumento della potenza dello spettro attorno allo stesso istante di tempo.



*Figura 6: a / Un esempio di un segnale contenente un sharp wave-ripple complex (SPW-R)
b / Un esempio di applicazione del metodo di clusterizzazione [17]*

La Figura 6a mostra un esempio di segnale contenente un pattern oscillatorio denominato sharp wave-ripple complex e indicato dalla freccia. Una volta identificato l'evento di interesse si applica l'algoritmo di clusterizzazione in tre classi: nelle bande sigma, gamma e ripple (140-220 Hz). Il risultato, presente in Figura 6b, mostra la media degli spettri nelle tre classi attorno all'evento considerato.

Un modo per aumentare l'efficacia di questo metodo è applicare diversi stimoli che inducano l'attivazione di specifiche zone di interesse. [5] [6]

1.4.3 Analisi tramite Current Source Density

I segnali extracellulari forniscono informazioni sul comportamento di neuroni. Tuttavia, gli stessi effetti macroscopici possono essere generati da eventi microscopici differenti. È quindi necessario cercare di inferire le caratteristiche delle correnti di dipolo dal campo extracellulare. Questa procedura è definita problema inverso. Per localizzare le sorgenti di correnti si utilizza il concetto di CSD (Current Source Density). A causa del numero elevato di elementi che contribuiscono alla corrente transmembrana (dendriti, soma e assoni) non è possibile stimare le singole componenti ma si stima CSD ossia la corrente media transmembrana. L'analisi CSD tradizionale si basa sull'ipotesi che il tessuto neocorticale sia eterogeneo lungo la direzione verticale e piuttosto omogeneo lungo quella laterale. Di conseguenza la variazione laterale di LFP viene trascurata e la CSD può essere stimata con una doppia derivata del segnale extracellulare registrato (la prima derivata fornisce la corrente extracellulare e la seconda permette di approssimare la variazione di corrente locale). Nel caso in cui la sorgente violi l'assunzione di omogeneità laterale il metodo non è più applicabile e si utilizza l'analisi CSD inversa (iCSD).

La iCSD permette una più generica assunzione per la distribuzione geometrica delle sorgenti CSD. Per prima cosa stabilisce un modello forward che descrive come le correnti locali vengono tradotte in LFP. Successivamente inverte questo modello per permettere il calcolo diretto delle posizioni delle sorgenti. Questo approccio permette anche di includere informazioni a priori sulle sorgenti neuronali come, per esempio, dimensione e larghezza. La più elevata risoluzione spaziale e una più accurata interpretazione biofisiologica permettono di rilevare aspetti complementari a quelli ricavati dall'analisi del segnale LFP. [5] [6] [18]

1.5 Sistemi per l'acquisizione di LFP

1.5.1 Tipologia sensori

Negli ultimi anni l'uso di sistemi microelettromeccanici (MEMS) è aumentato sensibilmente. In particolare, recentemente si è visto aumentare l'uso di chip impiantabili per la registrazione in vivo come i Brain-Chip-Interface (BHCI), sistemi ibridi dove dispositivi micro elettronici stabiliscono un contatto fisico con i neuroni. Si possono distinguere tre livelli di interfaccia per i chip, identificati in base al tipo di scala: neuroni, tessuto e cervello. Attualmente, i neuroni sono interfacciati con micro elettrodi metallici o micro trasduttori oxide-isolated per la stimolazione o registrazione della loro attività. Questo primo livello implica che ci sia un contatto diretto tra neurone e superficie del sensore [19]. Il secondo livello, ottenuto ponendo un sottile strato di tessuto di alcuni centinaia di micrometri in contatto con il chip, prevede che ci sia un'interazione tra tessuto cerebrale e chip [20]. Infine, il terzo livello è rappresentato dai chip impiantati nel cervello o in altre parti del sistema nervoso [21].

Una soluzione diffusa è realizzare il chip attraverso l'uso di transistor MOSFET (Metal-Oxide-Semiconductor Field Effect Transistor). Si possono distinguere tre generazioni di chip. Le prime due sono formate da array lineari e vengono utilizzati per l'elettrocorticogramma. Di conseguenza non vengono impiantati ma utilizzati solo per rilevazioni nella porzione più esterna della corteccia. Il primo esemplare è formato da due array con 64 siti di registrazione mentre la sua evoluzione consiste in una matrice di sensori 128 x 128. Quest'ultimo presenta una dimensione di 5.4 mm x 6.5mm. L'ultima generazione consiste in un chip impiantabile formato da 4 siti di registrazione [22].

Nonostante la diffusione di chip spesso vengono utilizzati degli elettrodi costituiti da un singolo sito di registrazione.

1.5.2 Microelettrodi (micropipette)

Un primo tipo di sensore è composto da micropipette. Sono costruite attraverso una strumentazione apposita, in modo da poter avere un processo di produzione affidabile. Vengono testate in soluzione di Krebs in modo da verificare la resistenza intrinseca dei sensori. Questa vale circa 0.3 M Ω per micropipette usate per la stimolazione e circa 1 – 2

M Ω nel caso di acquisizione. Infine vengono rivestite da uno strato di Sylgard in modo da minimizzare l'effetto capacitivo tra pipetta e soluzione. Le dimensioni tipiche sono di 1.05x1.05x100 mm. [23]

1.5.3 EOSFET

Una seconda tipologia di sensori è costituita da chip EOSFET. La caratteristica principale consiste nel fornire struttura di contatto a matrice. Ogni chip è costituito da due parti: un ago costituito da un array di transistor e una piastra di contatto. La dimensione tipiche dell'ago è di circa 2mm di lunghezza e 300 μ m di larghezza. I chip vengono realizzati da un wafer di silicon-on-insulator (SOI) con un processo simile a quello utilizzato per i wafer di solo silicio. [23]

1.5.4 Software

Attualmente il software utilizzato permette di acquisire segnali di durata limitata e non consente una configurazione flessibile in modo da potersi adattare a diverse tipologie di esperimenti. Inoltre non presenta una gestione ottimale dell'acquisizione multi canale introdotta dai sensori di ultima generazione. Infatti l'evoluzione dei sensori permette di acquisire contemporaneamente segnali LFP da un numero sempre crescente di siti di registrazione. Questo consente di osservare il comportamento di una vasta popolazione di neuroni favorendone quindi una migliore comprensione e valutazione. Tuttavia, al crescere della complessità dei sensori, aumenta anche la quantità di dati da acquisire ed analizzare. Di conseguenza è necessario affiancare un adeguato sistema di acquisizione in grado di gestire in modo efficace la grande mole di dati offrendo allo stesso tempo una serie di strumenti in grado di favorire una valutazione in tempo reale dell'acquisizione in corso.

1.6 Scopo della tesi

Lo scopo di questa tesi, svolta in collaborazione con NeuroChip Laboratory, presso il Dipartimento di Scienze Biomediche dell'Università di Padova, è la realizzazione di un software di acquisizione per analisi di segnali elettrofisiologici come quelli presentati nel paragrafo 1. L'applicativo deve essere in grado di visualizzare il segnale acquisito in tempo reale affiancato da alcune elaborazioni come filtraggio e spettro in frequenza. Un importante requisito è di poter salvare i dati in formato nativo di LabVIEW, il file tdms e anche in formato mat per semplificare l'importazione dei dati in MATLAB per una post-elaborazione.

Nei prossimi capitoli sarà descritta la fase preliminare di progettazione del software (Capitolo 2) e successivamente l'implementazione del codice e la sua descrizione (Capitolo 3). Il sistema verrà poi testato in laboratorio ai fini della registrazione di LFP nella corteccia di ratti in modo da verificare la validità e l'efficacia del software creato.

2. PROGETTAZIONE DEL SOFTWARE

2.1 Analisi dei requisiti

La fase preliminare alla realizzazione del software è dedicata all'analisi dei requisiti necessari.

Il software proposto, realizzato in LabView, nasce per acquisire dati da una scheda DAQ (National Instruments NI PCI-6071E, risoluzione di 12-bit e frequenza di campionamento fino a 1.25MS/s). Generalmente, negli esperimenti neurofisiologici, il segnale è acquisito a frequenze tipiche dell'ordine di alcune decine di KHz ed è necessario che venga visualizzato in tempo reale. Poiché si desidera progettare un software che si adatti all'acquisizione di qualsiasi segnale elettrofisiologico e a diversi set up sperimentali, si sono proposti i seguenti requisiti fondamentali:

- Definire un'interfaccia grafica intuitiva per la configurazione dei parametri dell'esperimento
- Permettere due possibili modalità di acquisizione: continua o suddivisa a finestre di lunghezza scelta dall'utente
- Visualizzare in tempo reale la traccia acquisita e la traccia filtrata nelle bande di frequenza di interesse
- Generare un segnale di stimolazione per acquisire le risposte a stimolo
- Salvare i dati sia in formato tdms sia in formato mat

2.2 Scelte procedurali

Per quanto riguarda la realizzazione dell'interfaccia grafica, poiché le due modalità di acquisizione condividono i parametri di configurazione, è possibile realizzare un'unica interfaccia. Riguardo la possibilità di visualizzare in tempo reale i campioni acquisiti, dato che gli esperimenti tipici hanno una durata variabile tra alcuni minuti e diverse ore e la frequenza di campionamento è dell'ordine di alcune decine di KHz, la mole di dati da gestire può diventare elevata. Di conseguenza si è scelto di mantenere in memoria solo i dati attualmente visualizzati attraverso un buffer e di salvarli su disco fisso nel momento in cui questi non siano più disponibili all'utente. Un altro requisito necessario è la possibilità di introdurre una forma di stimolazione. Una caratteristica fondamentale è il sincronismo rispetto alla fase di acquisizione, per questo motivo si è scelto di sfruttare il sincronismo intrinseco delle schede di acquisizione in modo da ottenere i risultati migliori. Infine, si è inserita la possibilità di acquisire più canali contemporaneamente.

Lo sviluppo del codice sarà accompagnato da numerosi test realizzati attraverso strumenti di simulazione, in modo da rendere l'algoritmo efficiente ed essere così eseguito su diverse configurazioni hardware.

2.3 L'ambiente LabVIEW

LabVIEW (Laboratory Virtual Instrumentation Engineering Workbench) è l'ambiente di sviluppo integrato per il linguaggio di programmazione visuale di National Instruments. LabVIEW nasce nel 1983 con l'intento di fornire un valido strumento software per testare velocemente l'hardware prodotto dalla stessa National Instruments. La prima versione viene rilasciata nel 1986 per Macintosh e nel 1992 ne viene sviluppata una versione multiplatforma. Lo sviluppo di LabVIEW è sempre stato affiancato da una forte integrazione con gli apparati hardware, creando così un sistema affidabile per la creazione di applicativi per acquisizione e analisi di dati [24]. Per questo motivo si è scelto di realizzare il software in LabVIEW in modo da garantirne la massima affidabilità e compatibilità con i dispositivi di acquisizione.

Nell'ambiente di sviluppo, ogni programma o sottoprogramma è denominato VI (Virtual Instrument). Si compone di tre parti principali:

- Pannello frontale

- Schema a blocchi
- Riquadro connessioni

Il pannello frontale presenta l'interfaccia grafica di un VI. Sono presenti i controlli (input) e gli indicatori (output). Lo schema a blocchi presenta il codice in formato grafico. Gli stessi oggetti del pannello frontale compaiono come terminali. Possono essere inserite funzioni matematiche di base e anche altri VI denominati subVI. Le varie funzioni possono essere collegate tra di loro attraverso linee di flusso. Infine, il riquadro delle connessioni permette di definire gli input e gli output di un subVI ai quali collegare le linee di flusso.

3. REALIZZAZIONE DEL SOFTWARE DI ACQUISIZIONE

3.1 L'interfaccia utente

L'interfaccia utente è divisa in due parti. La prima è dedicata alla configurazione dei parametri tipici di un esperimento (Figura 7) mentre la seconda è destinata alla visualizzazione dei segnali acquisiti (Figura 8).

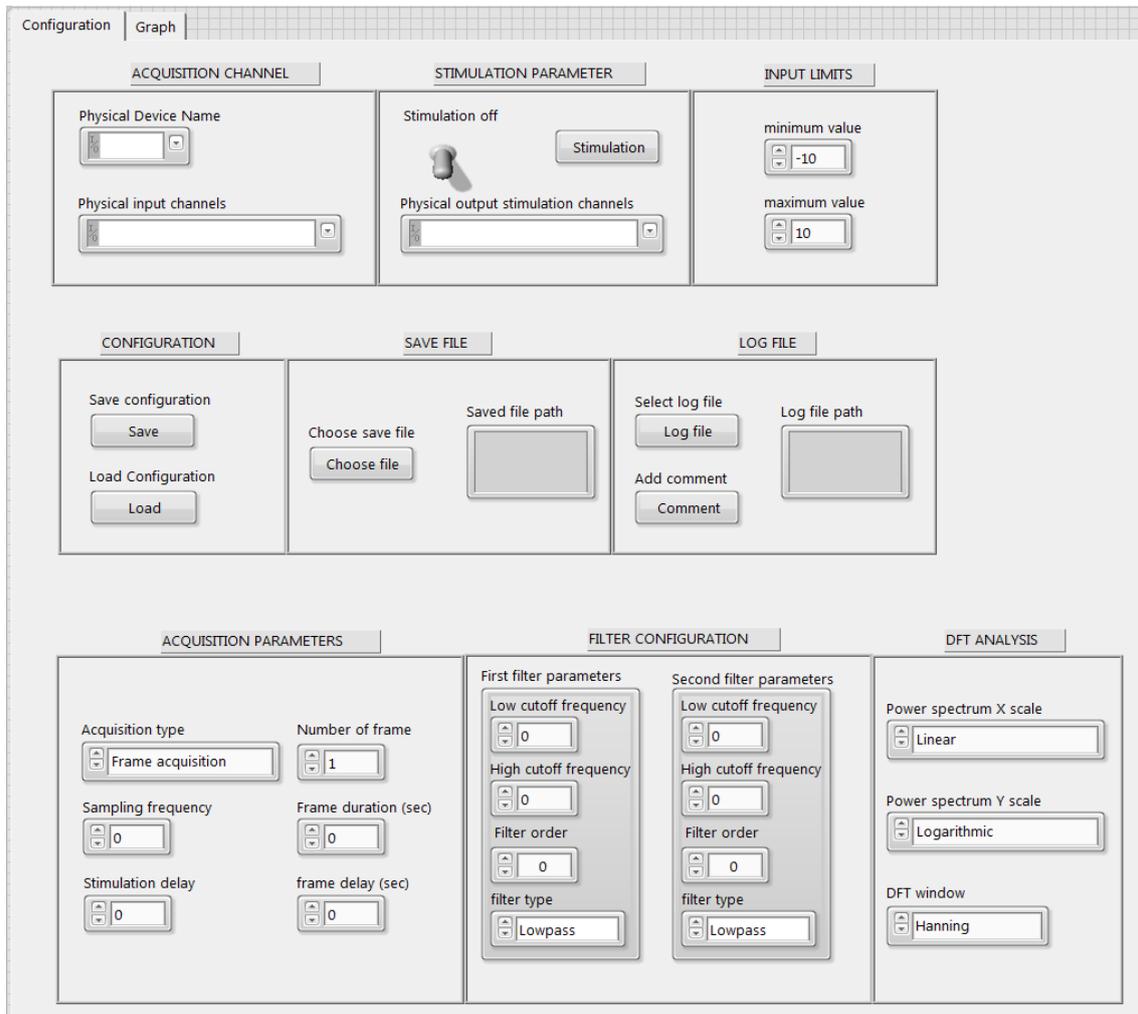


Figura 7: Interfaccia grafica. Configurazione parametri



Figura 8: Interfaccia grafica. Visualizzazione grafici

3.1.1 Configurazione parametri

La Figura 7 mostra i parametri configurabili da parte dell'utente. Si possono osservare tre porzioni principali ognuna dei quali è suddivisa in moduli. La prima parte è costituita dai moduli "Acquisition Channel", "Stimulation Parameter" e "Input limit". Il primo è dedicato alla configurazione della scheda di acquisizione. Si può definire la scheda da utilizzare e successivamente i canali di input. Il secondo permette di impostare un canale di output utilizzato per generare una eventuale stimolazione. Attraverso il terzo modulo è possibile configurare il range di ampiezza del segnale di input in modo da ottimizzare l'efficacia del quantizzatore. La seconda parte permette all'utente di definire tre tipi di file. Nel primo modulo, "Configuration", è possibile salvare un file di configurazione utile per caricare velocemente un template di esperimento. Nel secondo, "Save file", si definisce il file di salvataggio nel quale memorizzare i dati acquisiti. Infine, il modulo "Log file" definisce il file di log contiene i parametri dell'esperimento affiancato ad un commento libero dell'utente. L'ultima parte permette di definire i parametri di acquisizione ("Acquisition Parameters"), del filtraggio ("Filter configuration"), le scale degli assi dello spettro ed infine il tipo di finestra per il calcolo della DFT ("Dft analysis"). I parametri di acquisizione consistono nella frequenza di campionamento, nel tipo di acquisizione e nell'intervallo di stimolazione. In base alla tipologia di acquisizione scelta, il software automaticamente definisce gli ulteriori parametri specifici per il tipo di acquisizione. Nel caso di acquisizione continua l'utente deve inserire la durata dell'acquisizione e la durata della finestra utilizzata per visualizzare gli ultimi dati acquisiti. Nel caso di acquisizione separata in finestre, l'utente deve inserire il numero di finestre da acquisire, la loro durata, il ritardo tra finestre ed infine il ritardo nella stimolazione.

3.1.2 Visualizzazione dati acquisiti

La seconda parte dell'interfaccia riporta i grafici che visualizzano il segnale acquisito, il filtraggio e lo spettro. Nel caso di acquisizione a finestre è disponibile anche la visualizzazione della media delle finestre nel tempo e il suo spettro. Sono anche presenti alcuni parametri aggiuntivi (Figura 9).

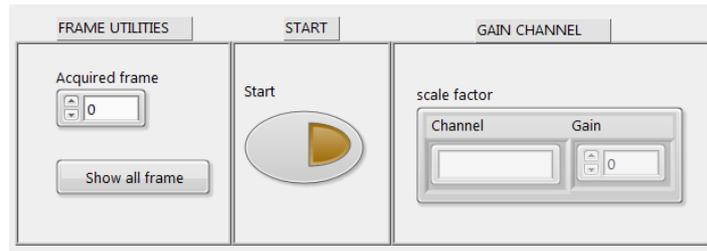


Figura 9: Interfaccia grafica: parametri aggiuntivi

È presente un pulsante per iniziare la procedura di acquisizione (modulo “Start”) ed infine è disponibile la possibilità di specificare un fattore di guadagno diverso per ogni canale di input (modulo “Gain channel”). Il modulo “Frame utilities” contiene un indicatore per il numero di finestre acquisite ed un pulsante, “Show all frame”, che permette di visualizzare tutti i segnali acquisiti al termine dell’esperimento (Figura 10). La stessa interfaccia permette di selezionare o eliminare le finestre acquisite che non rispettano determinate caratteristiche.

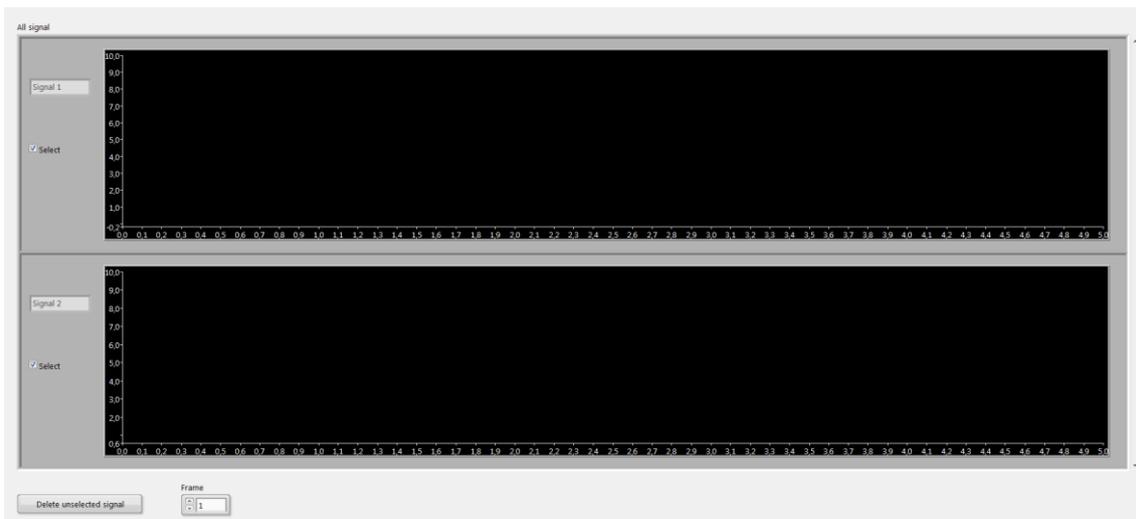


Figura 10: Finestra per visualizzare ed eliminare le finestre acquisite

La gestione dell’interfaccia grafica avviene attraverso l’uso di una struttura di gestione eventi che permette di definire, per ogni pulsante, uno stato nel quale inserire la porzione di codice relativa. In Figura 11 viene riportato, per semplicità, il solo stato di gestione del pulsante “Start”. Vengono configurati i canali di input e output, viene creato il file per il

salvataggio dei dati, la coda FIFO ed infine vengono controllati i parametri inseriti nell'interfaccia grafica. Nel caso di errore l'utente viene informato, attraverso un popup, dei parametri mancanti o errati (Figura 12). Infine, la dimensione dei grafici viene adattata automaticamente alla risoluzione dello schermo per ottimizzare la visualizzazione dei segnali acquisiti.

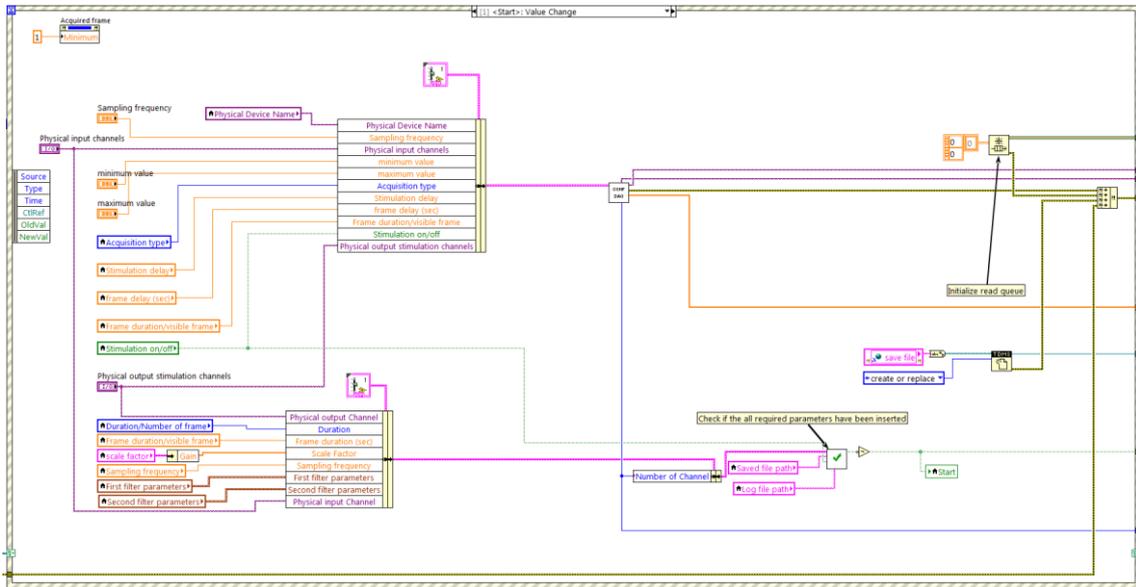


Figura 11: Gestione pulsante "Start"

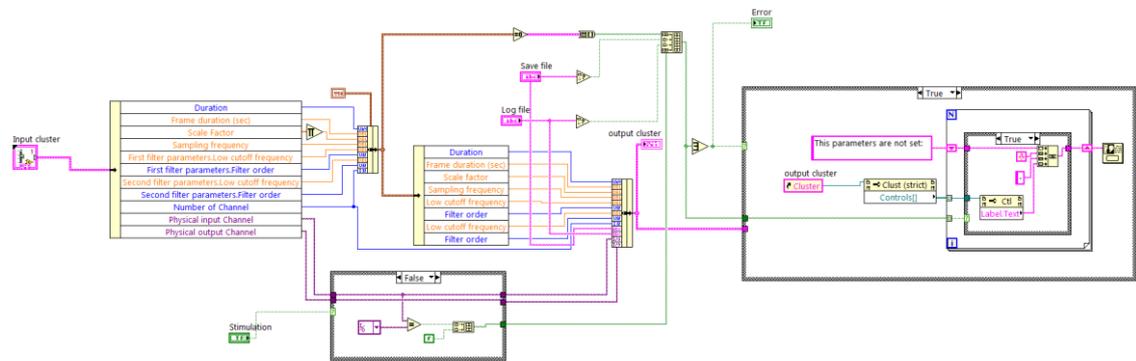


Figura 12: Codice per il controllo dei parametri dell'interfaccia grafica

3.2 Tipologia di acquisizione

Sono disponibili due tipologie di acquisizione:

- Acquisizione continua
- Acquisizione a finestre

Le due tipologie sono molto simili e differiscono solo per l'interpretazione dei dati acquisiti. L'algoritmo è stato quindi suddiviso in due parti. La prima, comune, che si occupa di trasferire i dati dalla scheda di acquisizione alla memoria RAM (Random Access Memory). La seconda parte, invece, è specifica per il tipo di acquisizione.

La prima parte acquisisce i dati dalla scheda di acquisizione ad intervalli regolari di 100 ms. Moltiplica questi dati per il fattore di guadagno specifico per ogni canale e infine li memorizza in memoria RAM attraverso l'uso di una coda FIFO (Figura 13).

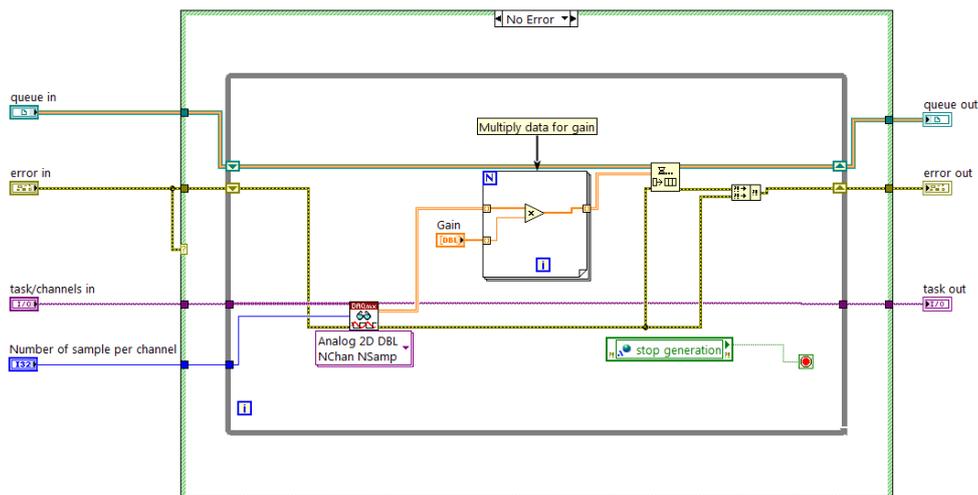


Figura 13: Acquisizione dati da scheda e memorizzazione nella coda FIFO

Nel caso di acquisizione continua i dati vengono prelevati dalla coda FIFO ad intervalli regolari di 250 ms ed inseriti in un buffer circolare. Quando il buffer viene riempito, i dati vengono salvati su disco. Le successive letture dalla coda FIFO sovrascrivono i dati nel buffer circolare.

Nel caso di acquisizione a finestre i dati vengono memorizzati in un buffer di lunghezza fissa ad intervalli di 250 ms. Al riempimento del buffer i dati vengono salvati su disco e

Figura 14: Buffer lunghezza fissa

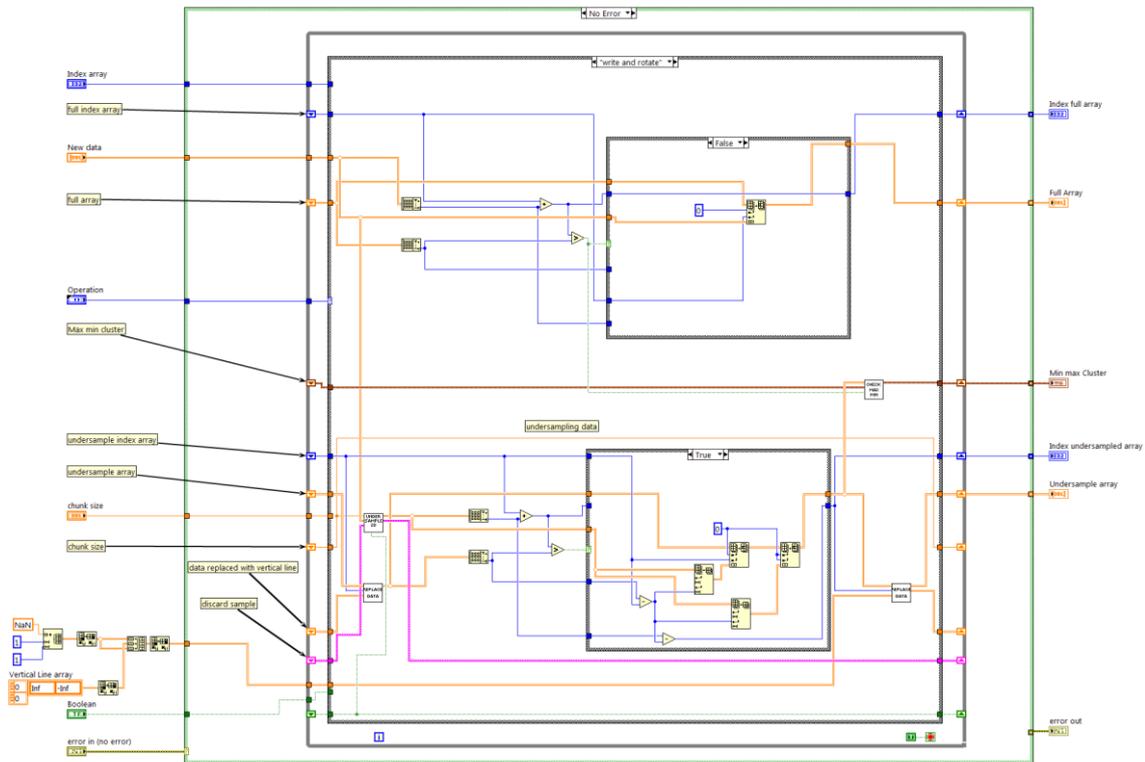


Figura 15: Buffer circolare

Entrambi gli stati implementano anche un algoritmo di decimazione, utile per visualizzare segnali di lunga durata. Considerando, per esempio, una frequenza di campionamento di 10KHz e una finestra di visualizzazione di 10 secondi, si dovrebbero visualizzare segnali contenenti 100000 punti. I monitor attualmente utilizzati, sono caratterizzati da un numero di pixel compreso tra 1000 e 2000, molto inferiore al numero di campioni da visualizzare. Di conseguenza si avrebbe un utilizzo non ottimale di risorse hardware e software. L'algoritmo adatta quindi il numero di campioni ai pixel disponibili alla visualizzazione. È stato utilizzato il metodo "min-max decimation" proposto dalla National Instruments. Il funzionamento schematico è riportato nella Figura 16, assieme alla griglia dei pixel. [26]

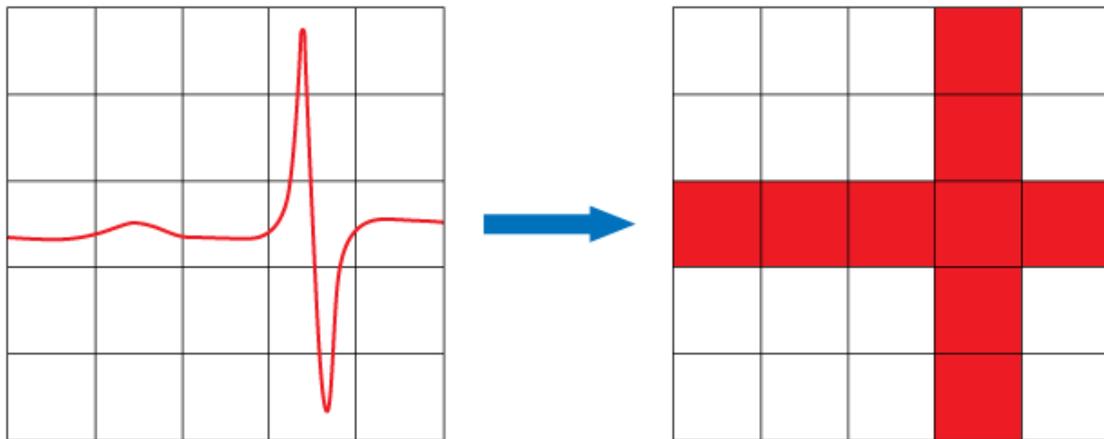


Figura 16: Schema funzionamento algoritmo “min-max decimation” [28]

L’algoritmo considera la larghezza del grafico espressa in n pixel. Per ridurre gli artefatti il segnale viene suddiviso in $2n$ frammenti. Il vettore decimato è creato memorizzando, per ogni intervallo, il massimo e il minimo secondo l’ordine di occorrenza. Questo permette di ridurre considerevolmente il numero di campioni da visualizzare mantenendo però le caratteristiche del segnale originale come, ad esempio, i picchi. Il codice viene riportato in Figura 17 [28]. Il vantaggio di questa gestione di memoria consiste nell’ottenere, attraverso l’uso di un unico VI, la memorizzazione del segnale originale e sottocampionato.

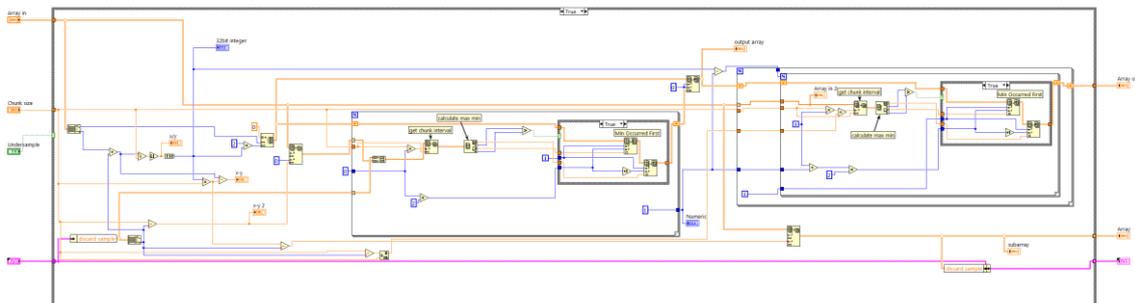


Figura 17: Implementazione min-max decimation

3.4 Stimolazione

Una caratteristica del software è la possibilità di introdurre un segnale di stimolazione. Il segnale, generato attraverso un output analogico, ha la forma di un rect con valori diversi da zero in corrispondenza della stimolazione e zero altrimenti. Per ottenere una sincronizzazione tra l'acquisizione e la stimolazione si è sfruttata la capacità di trigger intrinseca a livello hardware.

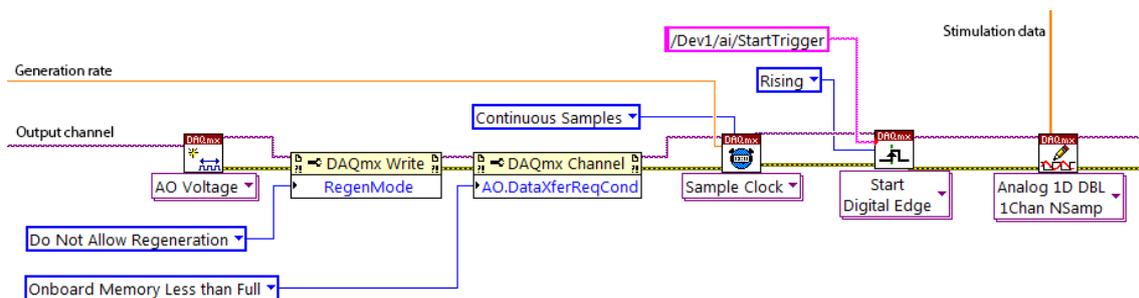


Figura 18: Codice creazione canale output

Analizzando il codice di Figura 18 si osserva la creazione del canale di output attraverso il VI “AO Voltage”. Viene disabilitata la rigenerazione automatica del segnale e viene definita la modalità di trasferimento dei campioni dal pc alla scheda di acquisizione. Il VI “Sample Clock” definisce la frequenza di campionamento del segnale e il VI “Start Digital Edge” riceve come input il trigger utilizzato per iniziare la stimolazione e, in questo caso, è riferito all'acquisizione. In questo modo si configura la scheda di acquisizione affinché gestisca automaticamente la sincronizzazione. Infine, viene definito il segnale di stimolazione.

Il vantaggio di questa implementazione emerge nel ritardo che si può ottenere tra l'acquisizione e la stimolazione. In questo modo si possono raggiungere sincronizzazione dell'ordine dei nanosecondi in quanto il trigger è legato al fronte di salita del clock interno della scheda. La sincronizzazione software non avrebbe permesso di raggiungere prestazioni migliori di alcuni millisecondi. [27]

3.5 Architettura multicore

La procedura di acquisizione, divisa in due parti, e la generazione della stimolazione sono implementati attraverso tre cicli while. In questo modo è possibile sfruttare l'architettura multicore delle CPU moderne ed eseguire così contemporaneamente più operazioni. Un ulteriore vantaggio di questa struttura è la separazione del codice dedicato all'acquisizione dei dati dalla scheda rispetto a quello dedicato all'elaborazione. In questo modo è possibile assicurare che la procedura di acquisizione venga eseguita nel più breve tempo possibile evitando così una possibile perdita di campioni.

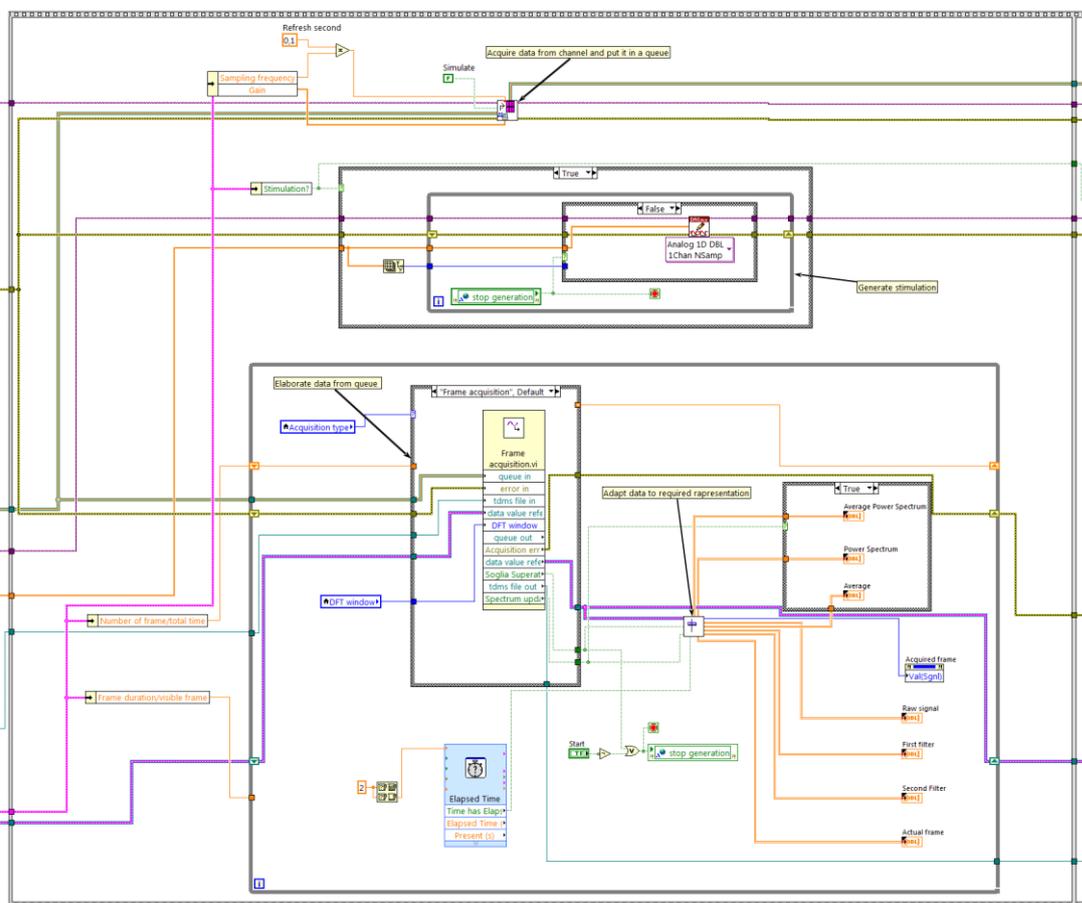


Figura 19: Struttura cicli while

Dalla Figura 19 si possono vedere i tre cicli while utilizzati. Il primo è dedicato alla lettura dei campioni dalla scheda, il secondo si occupa della generazione di un'eventuale stimolazione ed il terzo elabora i dati ad aggiorna i grafici dell'interfaccia grafica. [29]

3.6 Elaborazione real-time

Il software fornisce diverse elaborazioni in tempo reale per fornire all'utente un feedback dell'acquisizione in corso. Vengono riportati i grafici contenenti:

- Il segnale grezzo
- Il segnale filtrato
- Lo spettro

Nel caso di acquisizione a continua viene riportato:

- La media delle spettri

Nel caso di acquisizione a finestre vengono riportati:

- La finestra attuale
- La media delle finestre nel tempo
- Lo spettro della media delle finestre

L'aggiornamento dei grafici avviene secondo due intervalli di refresh. Lo spettro e la media delle finestre vengono calcolati al riempimento del buffer di memoria e dipende quindi dalla durata della finestra di visualizzazione.

Gli altri grafici vengono aggiornati ogni 250 ms. Al fine di minimizzare l'onere computazionale i parametri del filtro vengono calcolati prima di iniziare l'acquisizione (Figura 20). Successivamente i dati acquisiti vengono filtrati e lo stato del filtraggio viene memorizzato per una successiva iterazione (Figura 21).

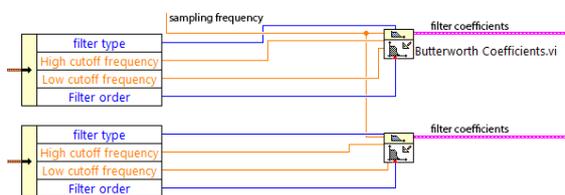


Figura 20: Calcolo coefficienti filtro

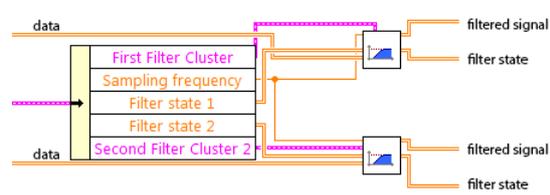


Figura 21: Filtraggio dati acquisiti

Il calcolo della media degli spettri è stata implementata attraverso l'uso di FGV (Figura 22). Viene utilizzata una matrice a tre dimensioni le quali rappresentano, rispettivamente, i canali di input, i campioni e gli spettri nel tempo. La terza dimensione viene gestita come

un buffer circolare attraverso l'uso di un indice interno alla FGV. Di conseguenza, all'inserimento di un nuovo vettore, vengono sovrascritti i dati più vecchi. [29]

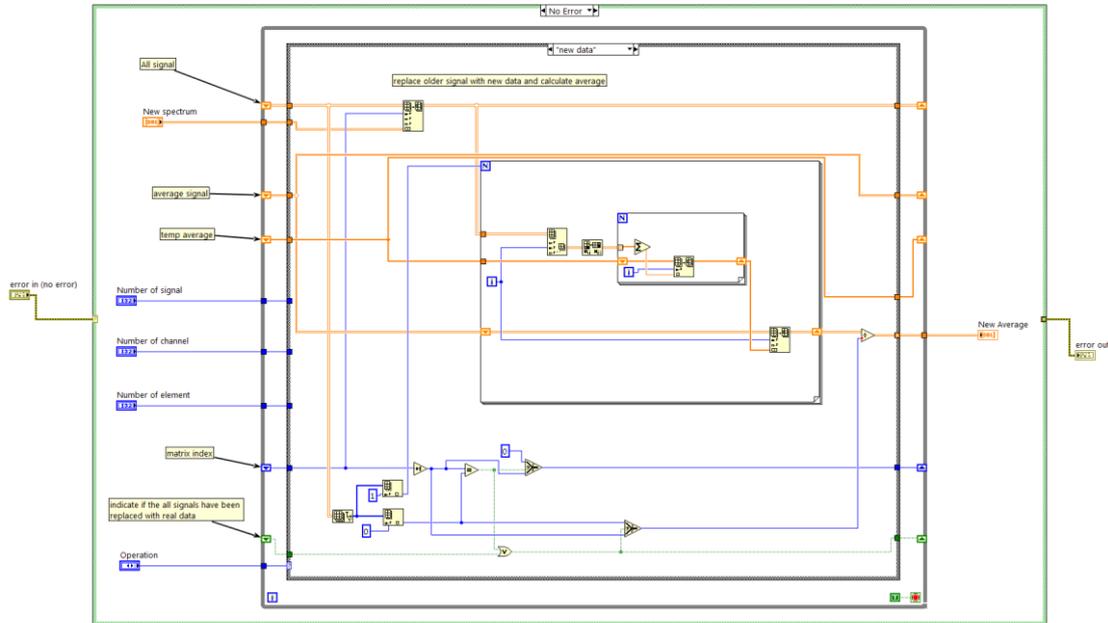


Figura 22: Codice media mobile

3.7 Salvataggio dati

Il salvataggio dei dati su disco fisso avviene attraverso due modalità. La prima utilizza una funzione integrata in LabVIEW e permette di memorizzare i dati in modo automatico attraverso l'uso del VI "DAQmx Configure Logging" (Figura 23). Questa implementazione sfrutta il meccanismo della DMA (Direct Memory Access) che permette di accedere alla memoria interna senza coinvolgere la CPU (Central Processing Unit).

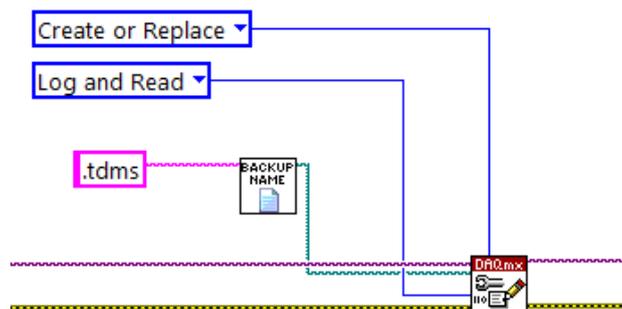


Figura 23: Uso funzione "DAQmx Configure Logging"

In questo modo è possibile mantenere una copia di backup in caso di crash improvvisi o errori durante l'acquisizione.

La seconda modalità utilizza il salvataggio dei dati su file tdms secondo la tipologia di acquisizione scelta. In questo caso il salvataggio avviene al riempimento del buffer di memoria (Figura 24).

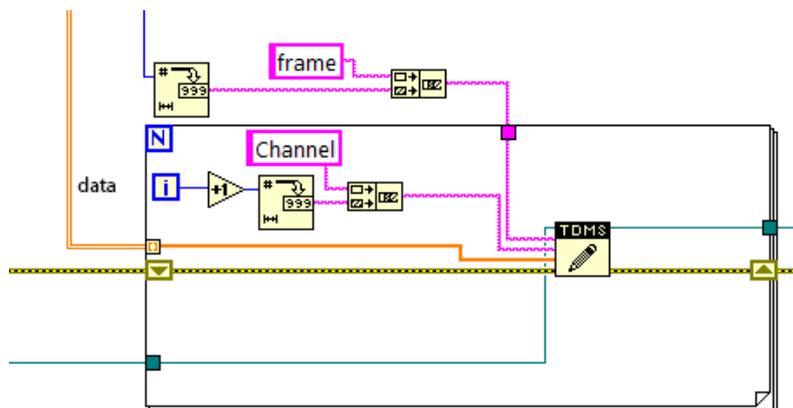


Figura 24: Esempio salvataggio dati nel caso di acquisizione a finestre

La scelta di usare il formato nativo di LabVIEW per la memorizzazione dei dati durante l'acquisizione permette di ottenere le massime prestazioni e la massima affidabilità durante la scrittura su disco.

Al termine dell'acquisizione viene generato anche un file in formato mat, il formato nativo di MATLAB, per favorire una post elaborazione in un secondo momento (Figura 25).

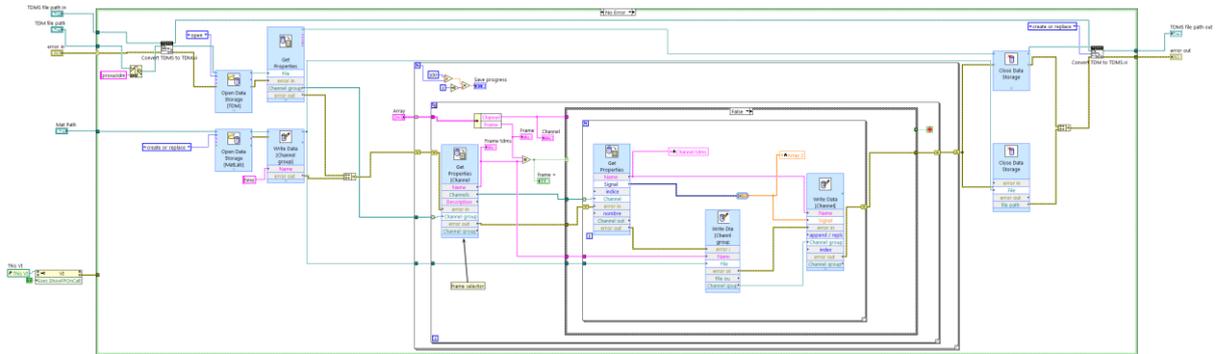


Figura 25: Codice conversione da tdms a mat

Il codice utilizza il file tdms salvato durante l'acquisizione e converte i dati in formato mat attraverso l'uso del DataPlugin contenuto nel modulo Diadem, un software per la gestione e l'analisi di dati della National Instruments. [26]

3.8 Elaborazione post acquisizione

Sono disponibili alcune elaborazioni al termine dell'acquisizione. La prima permette di eliminare le finestre acquisite che non soddisfano determinate caratteristiche. Il codice utilizzato è presente in Figura 25 e l'interfaccia è presente in Figura 10.

Un'ulteriore funzione introdotta permette di scegliere una delle finestre acquisite e di selezionarne una porzione per un'analisi più accurata. Un esempio è riportato in Figura 26.

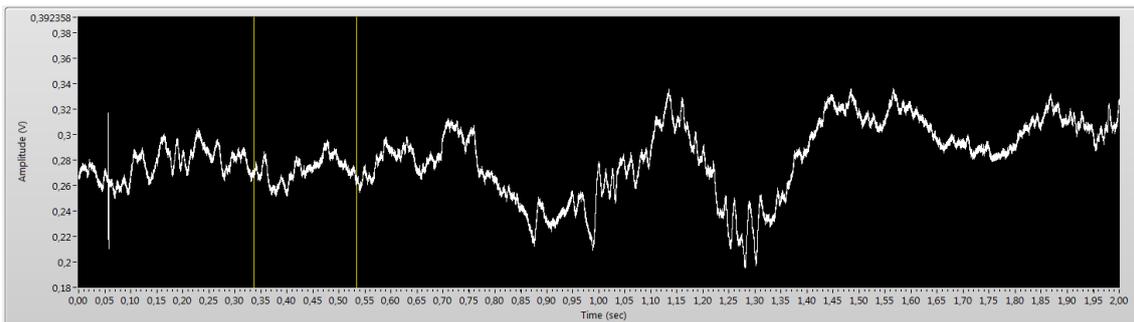


Figura 26: Esempio di selezione di una porzione di segnale acquisito

Al termine della selezione viene ricalcolato lo spettro della sotto-finestra (Figura 27).

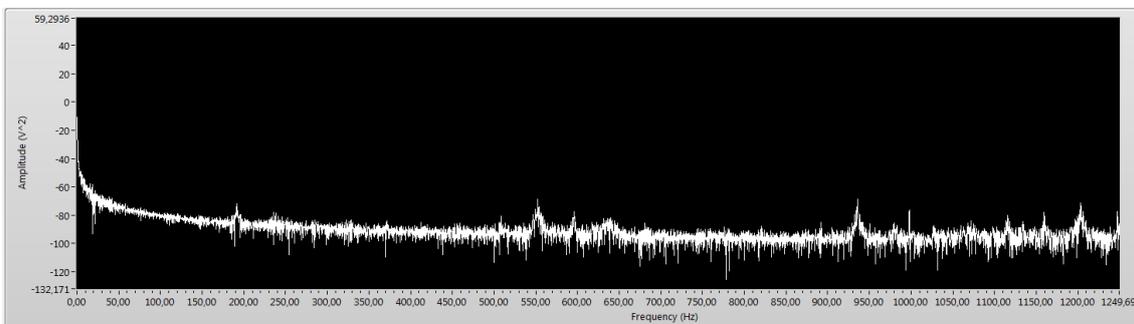


Figura 27: Spettro della sotto-finestra di Figura 26

Allo stesso modo può essere selezionata una porzione dei segnali filtrati.

In Figura 28 si osserva la porzione di codice dedicata alla gestione della sottofinestra. La possibilità di scegliere un porzione non è implementata in LabVIEW. Si è realizzata la completa gestione degli eventi del mouse in modo da rilevare l'area del grafico selezionata e convertirla nei corrispondenti istanti di tempo. Il codice è realizzato nel modo più generale possibile attraverso l'uso di una struttura ad eventi affiancata da una macchina

a stati per ogni evento del mouse, entrambe implementate all'interno di un subVI. Affinché quest'ultimo sia in grado di comunicare con il VI principale, viene generato un evento attraverso l'uso di una variabile globale. Il VI principale gestisce ed estrae dalla variabile globale le coordinate temporali e il grafico corrispondente (Figura 29). In questo modo è possibile adattare agevolmente questa implementazione per un numero arbitrario di grafici come mostrato in Figura 30. [26]

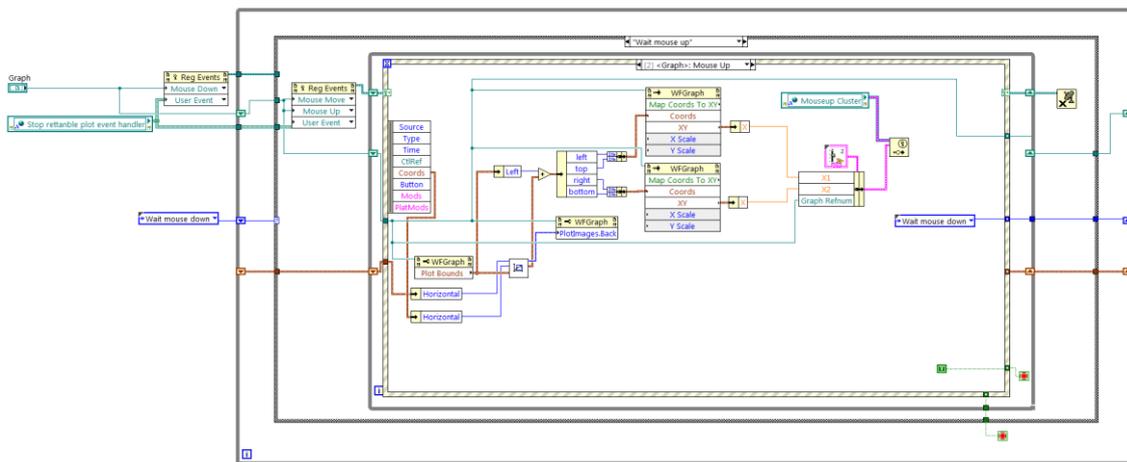


Figura 28: Codice per la gestione della sotto-finestra

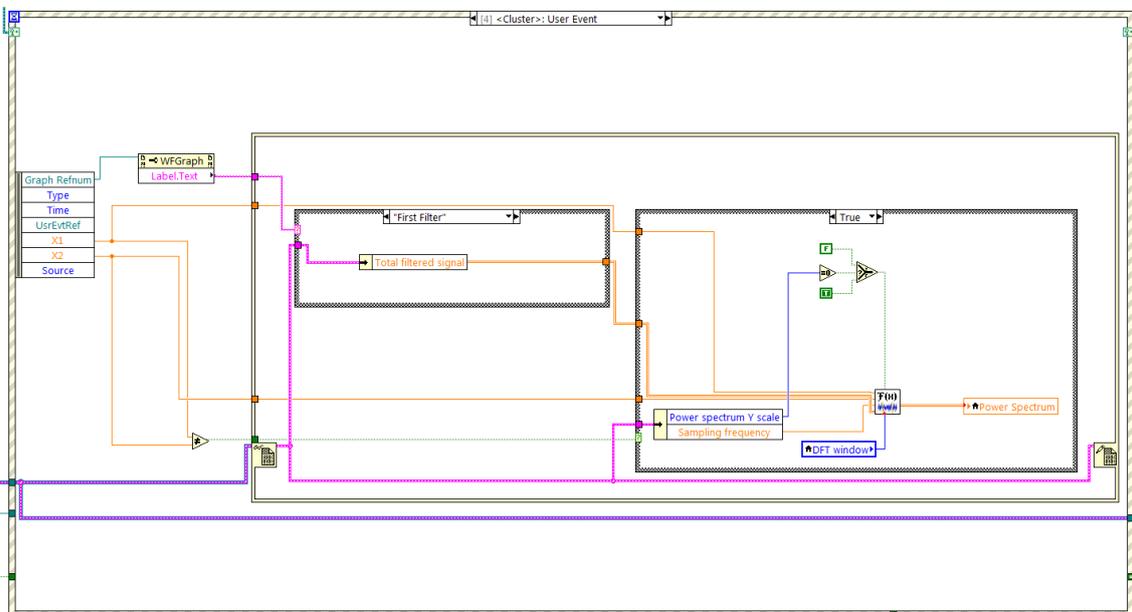


Figura 29: Gestione dell'evento associato alla variabile globale

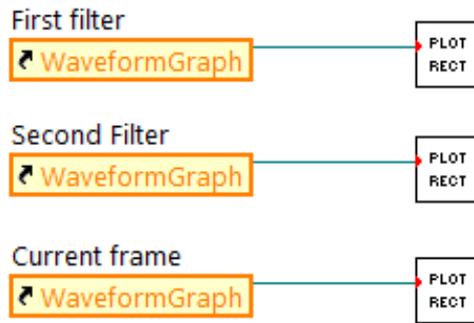


Figura 30: Uso subVI Figura 28

La gestione dell'acquisizione attraverso un buffer di memoria permette di incrementare le prestazioni e diminuire l'uso di risorse hardware. Questo approccio però non consente di avere i dati acquisiti disponibili per la post elaborazione. Di conseguenza si è implementata la lettura dinamica del file tdms salvato durante l'acquisizione in modo da estrarre solo la finestra di interesse. Infine, per la post elaborazione i segnali visualizzati non vengono decimati per mantenere la massima risoluzione anche nel caso di un eventuale zoom da parte dell'utente. Il codice utilizzato è presente in Figura 31.

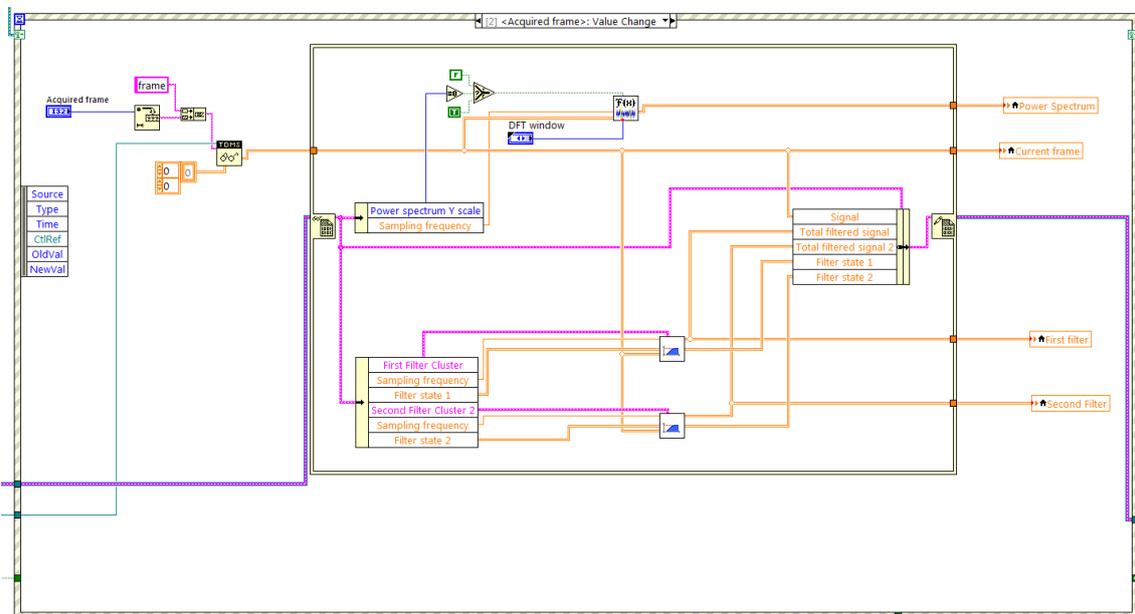


Figura 31: Codice per la lettura dinamica del file tdms

4. TEST DEL SOFTWARE

4.1 Setup sperimentale

Il software è testato in esperimenti reali presso il NeuroChip Laboratory (Dipartimento di Scienze Biomediche, Università degli Studi di Padova nel periodo 10/10/2014 – 30/01/2015). I test sono effettuati su un esemplare femmina di ratto. La procedura preliminare prevede la somministrazione di anestesia e successivamente una fase di monitoraggio in modo da verificare la mancanza di reazione a stimoli esterni. La temperatura corporea è mantenuta a 37.0 ± 0.5 °C. La testa del ratto è fissata attraverso uno strumento apposito ed, effettuando un'operazione chirurgica, è possibile inserire l'elettrodo all'interno dell'ippocampo. Il segnale, prima di essere acquisito, è amplificato di cento volte e filtrato attraverso un filtro passa-basso di Butterworth con frequenza di taglio di un 1 KHz. La scheda di acquisizione usata è la PCI-6071E I/O associata alla terminaliera BNC-2090.

4.2 Procedura acquisizione

Una prima fase di test coinvolge l'utilizzo di una soluzione elettrofisiologica assieme ad un elettrodo per validare il corretto funzionamento dell'applicativo. Vengono usate varie combinazioni di parametri di acquisizione in modo da esaminare le diverse possibili condizioni di esperimento. Dopo aver verificato il funzionamento, il software è testato su esperimenti reali.

4.2.1 Acquisizione continua

L'acquisizione continua è utilizzata per acquisire due canali contemporaneamente. Il primo è dedicato all'acquisizione del LFP mentre il secondo è destinato al ECG. Le caratteristiche tipiche di questo esperimento sono una frequenza di campionamento di 10KHz, acquisizioni di durata da 20 minuti ad alcune ore con una finestra di visualizzazione di alcune decine di secondi.

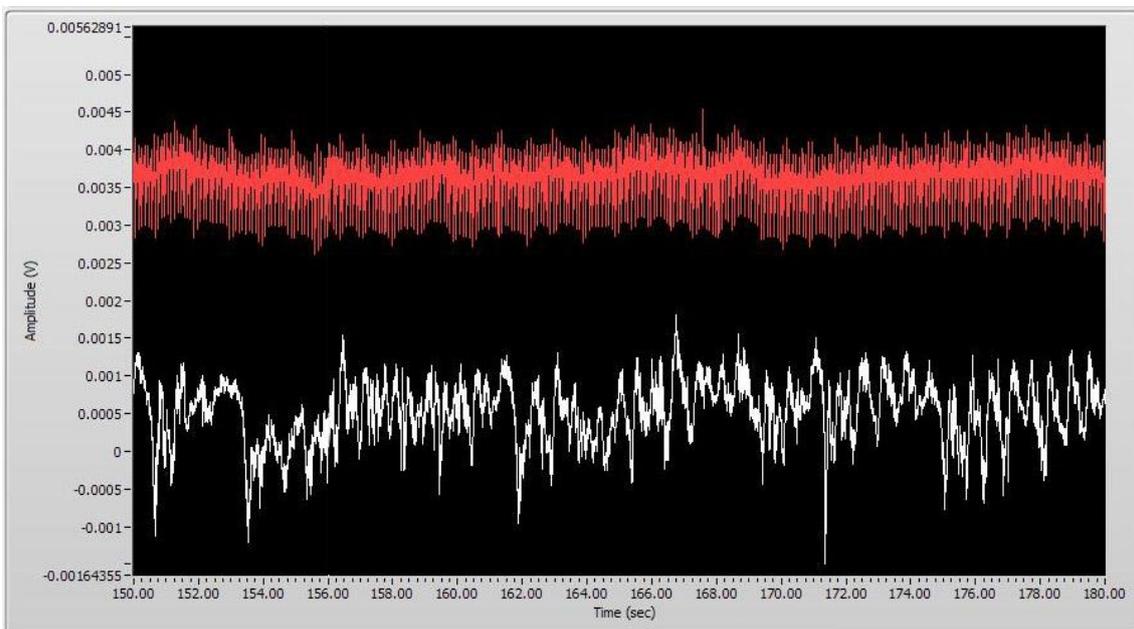


Figura 32: Esempio segnale con una finestra di 30 secondi

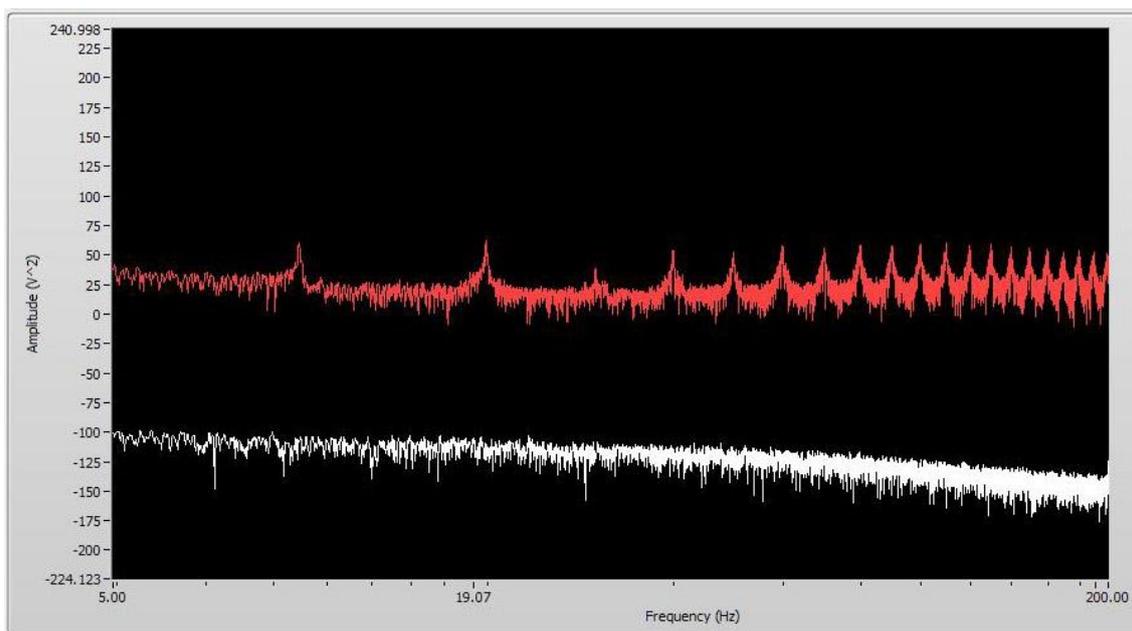


Figura 33: Esempio di spettro del segnale di Figura 32 nel range 5-200 Hz

La Figura 32 riporta un esempio di acquisizione. In rosso è rappresentato il segnale ECG mentre in bianco il segnale LFP. Entrambi sono visualizzati in una finestra di 30 secondi. Il software gestisce automaticamente lo spazio disponibile del grafico per adattarlo alla visualizzazione dei canali di input. I canali vengono riportati sullo stesso grafico in modo da facilitare la visualizzazione. In Figura 33 viene riportata la DFT.

4.2.2 Acquisizione a finestre

Nel caso di acquisizione a finestre, l'acquisizione ha utilizzato un solo canale di input affiancato da una stimolazione. I parametri tipici sono una frequenza di campionamento di 10kHz, acquisizione di 20 finestre della durata di 2 secondi. La stimolazione è generata attraverso un segnale campionato alla frequenza di 1 KHz con durata della stimolazione pari a 1 ms e ampiezza di 10 Volt. La Figura 34 riporta un esempio di acquisizione. Si può vedere l'artefatto da stimolazione a 0.06 secondi dall'inizio della finestra, sovrapposto al segnale LFP. In Figura 35 si può vedere il segnale filtrato nel range 1-100Hz. In Figura 36 è presente la media nel tempo delle varie finestre. Si può vedere come questa operazione riduca il rumore nella misura consentendo una valutazione migliore del segnale acquisito in tempo reale. Si può anche notare come l'artefatto da stimolazione si

mantenga fissato al 0.06 secondi dall'inizio della finestra, verificando così l'efficacia della sincronizzazione hardware.

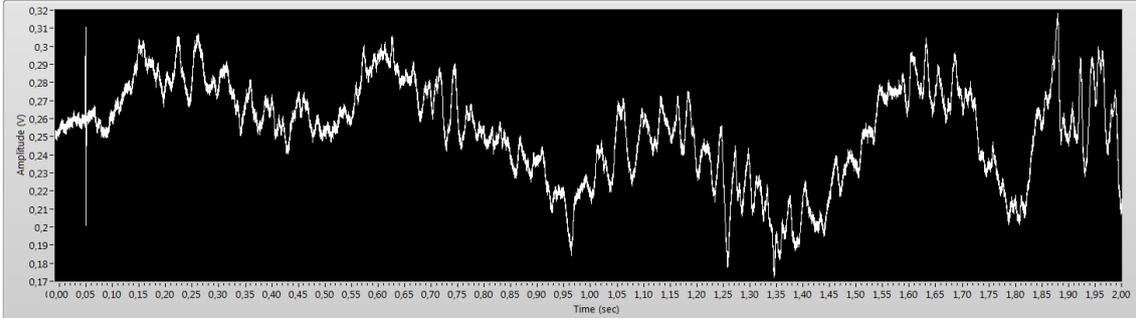


Figura 34: Esempio di acquisizione a finestre con durata di 2 secondi

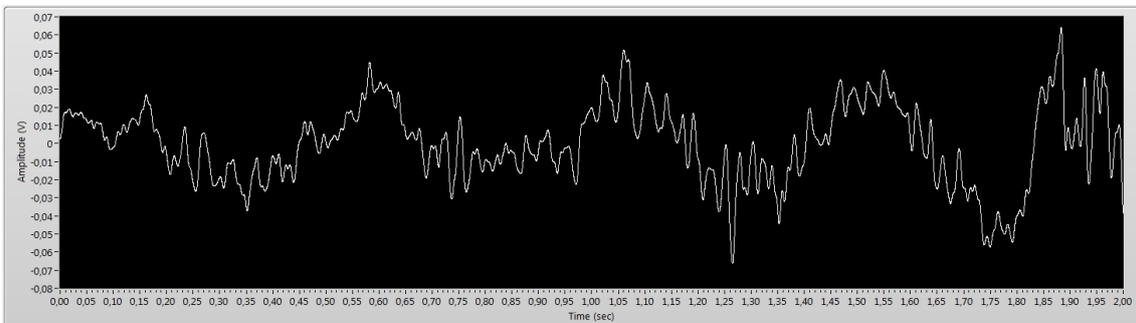


Figura 35: Esempio filtraggio segnale di Figura 24 nel range 1-100 Hz

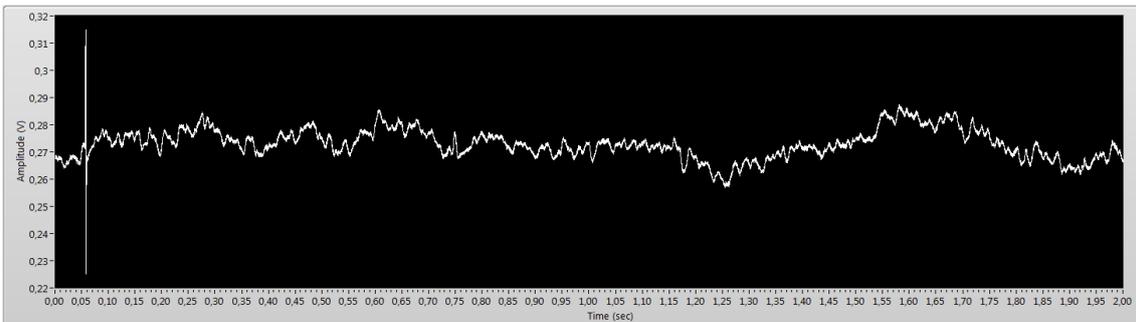


Figura 36: Esempio di media di 20 finestre

4.3 Bilancio del test

I test effettuati hanno permesso di verificare la capacità del software nel gestire dati provenienti da esperimenti reali. L'applicativo è stato utilizzato in diverse sessioni di test confermandone così anche l'affidabilità. L'acquisizione continua ha permesso di esaminare la capacità di gestire più canali contemporaneamente mentre l'acquisizione a finestre ha dimostrato l'efficacia della sincronizzazione hardware. Inoltre, è stata anche verificata la correttezza nella creazione dei file mat importando i dati in MatLab al termine della procedura di acquisizione. Durante le diverse prove si è anche monitorato l'utilizzo delle risorse hardware. Anche con acquisizioni di alcune ore, si è verificato che, su una CPU i7-2620M, il software non ha utilizzato più del 10% della CPU e circa 500 MB di RAM, confermando così l'efficacia delle ottimizzazioni effettuate a livello di codice.

5. CONCLUSIONI

5.1 Bilancio del lavoro svolto

In questi ultimi anni la complessità degli esperimenti neurofisiologici è aumentata sensibilmente e anche la quantità di dati da gestire durante l'acquisizione. Il seguente lavoro ha presentato un'implementazione di un software per l'acquisizione di segnali elettrofisiologici. I software attualmente disponibili non permettono una completa personalizzazione dei parametri dell'esperimento limitando così le tipologie di acquisizioni disponibili. La progettazione del software è avvenuta quindi valutando le caratteristiche richieste dagli esperimenti reali in laboratorio. L'integrazione di questi elementi caratteristici ha quindi permesso di realizzare un software flessibile e adattabile alle diverse tipologie di esperimenti. È stato realizzato un codice efficiente integrando le linee guide proposte dalla National Instruments. Il software è stato realizzato in modo da essere adattato alle diverse tipologie di acquisizione, come osservato nella sezione di test. Sono state definite le tipologie di acquisizione più importanti. La prima permette di acquisire campioni in modo continuo, utile nel caso di esperimenti di lunga durata. La seconda modalità effettua una acquisizione a finestre, utile con esperimenti con una stimolazione ripetuta ad intervalli regolari. L'interfaccia grafica permette di configurare molteplici parametri in modo da adattare l'acquisizione a diverse esigenze. È disponibile un'acquisizione multi-canale affiancata da un'eventuale stimolazione, entrambe configurabili separatamente. Sono presenti diverse analisi in tempo reale in modo da permettere all'utente di avere un primo feedback dell'acquisizione in corso. È possibile definire due filtri contemporaneamente ed è possibile visualizzare lo spettro del segnale. Inoltre, sono presenti alcune elaborazioni post esperimento in modo da poter verificare rapidamente la qualità dell'acquisizione appena effettuata e valutare così la successiva acquisizione. Per ultimo, il salvataggio dei dati in formato tdms e mat permette di utilizzare agevolmente MATLAB per una post elaborazione, mantenendo allo stesso tempo la massima compatibilità con

LabVIEW. Il software è stato testato principalmente per segnali LFP, ma può essere utilizzato anche per acquisire segnali di diversa natura e non necessariamente confinati in esperimenti neurofisiologici. La funzionalità del codice è stata testata in esperimenti reali dimostrandone così l'efficacia.

5.2 Sviluppi futuri

Il software presentato in questa tesi è stato realizzato in LabVIEW e, attraverso una progettazione modulare, è stato possibile racchiudere tutte le principali funzioni all'interno del proprio subVI. In questo modo è possibile intervenire facilmente sul codice scritto ed effettuare modifiche o aggiunta di funzionalità. Questo permette di aggiornare l'applicativo per un possibile sviluppo futuro come, per esempio, l'acquisizione da chip con numerosi canali di input. Inoltre, considerata la forte integrazione con l'hardware offerto dalla National Instruments, il software è compatibile con differenti dispositivi di acquisizione non utilizzati in questa tesi attualmente in commercio o di nuova generazione. Infine, grazie alla gestione del tutto generale dei dati acquisiti, è possibile sfruttare le funzionalità dell'applicativo anche in esperimenti neurofisiologici non trattati in questa tesi.

BIBLIOGRAFIA E SITOGRAFIA

Libri e articoli

- [1] Szymanski, Garcia-Lazaro and Schnupp, "Current source density profiles of stimulus-specific adaptation in rat auditory cortex," *Journal of Neurophysiology*, vol. 102, pp. 1483 - 1490, 2009.
- [2] H. Scherberger, M. R. Jarvis and R. A. Andersen, "Cortical Local Field Potential Encodes Movement Intentions in the Posterior Parietal Cortex," *Neuron*, vol. 46, pp. 347-354, 2005.
- [3] Pesaran, Pezaris, Sahani, Mitra and Andersen, "Temporal structure in neuronal activity during working memory in macaque parietal cortex," *Nature neuroscience*, pp. 805-801, 2002.
- [4] T. Womelsdorf, F. P., Mitra and D. P. P., "Gamma-band synchronization in visual cortex predicts," *Nature*, pp. 733-736, 2006.
- [5] Einevoll, Kayser, Logothetis and Panzeri, "Modelling and analysis of local field potentials for studying the function of cortical circuit," *Nature*, pp. 770-785, 2013.
- [6] Buzsaki, Anastassiou and Koch, "The origin of extracellular fields and current - EEG, ECoG, LFP and spikes," *Nature*, pp. 407-420, 2012.
- [7] Buzsaki, Buhl, Harris, Csicsvari, Czeh and Morozov, "Hippocampal network patterns of activity in the mouse," *Neuroscience*, pp. 201-211, 2003.
- [8] Hasselmo and Stern, "Theta rhythm and the encoding and retrieval of space and time," *NeuroImage*, pp. 656-666, 2014.
- [9] Goutagny, "Alterations in hippocampal network oscillations and theta-gamma coupling arise before A β overproduction in a mouse model of Alzheimer's disease," *European journal of neuroscience*, vol. 37, pp. 1896-1902, 2013.
- [10] L. F. Abbott, "Decoding neuronal firing and modelling," *Quarterly Reviews of Biophysics*, pp. 291-331, 1994.
- [11] A. Borst and F. E. Theunissen, "Information theory and neural coding," *nature neuroscience*, p. 947-957, 1999.

- [12] Quiroga and Panzeri, "Extraction information from neuronal population: information theory and decoding approaches," *Nature*, pp. 173-185, 2009.
- [13] G. Deco and D. Obradovic, "An Information-Theoretic Approach to neural Computing," *Springer*, 1997.
- [14] P. Mitra and H. Bokil, "Observed Brain Dynamics," *Journal of the Neurological Sciences*, 2009.
- [15] C. Magri, A. Mazzoni, N. Logothetis and S. Panzeri, "Optimal band separation of extracellular field potentials.," *Journal Neuroscience Methods*, pp. 66-89, 2012.
- [16] S. G. Mallat and Z. Zhang, "Matching pursuits with timefrequency," *IEEE Transactions on Signal Processing*, vol. 41, pp. 3397-3415, 1993.
- [17] N. K. Logothetis, O. Eschenko, Y. Murayama, T. S. M. Augath, H. C. Evrard, M. Besserve and A. Oeltermann, "Hippocampal–cortical interaction during periods of subcortical silence," *Nature*, p. 547–553, 2012.
- [18] C. Nicholson and J. A. Freeman, "Theory of current source-density analysis and determination of conductivity tensor for anuran cerebellum.," *Journal of Neurophysiology*, pp. 956-368, 1975.
- [19] H. A. S. J and S. ME, "Long-term, multisite, parallel, in-cell recording and stimulation by an array of extracellular microelectrodes.," *Journal of Neurophysiology*, p. 559–568, 2010.
- [20] M. Hutzler, A. Lambacher, B. Eversmann, M. Jenkner and R. Thewes, "High-resolution multitransistor array recording of electrical field potentials in cultured brain slices," *Journal of Neurophysiology*, p. 1638–1645, 2006.
- [21] S. Girardi, M. Maschietto, R. Zeitler, M. Mahmud and S. Vassanelli, "High resolution cortical imaging using electrolyte-(metal)-oxidesemiconductor field effect transistors," *IEEE EMBS conference on neural engineering*, pp. 269-272, 2010.
- [22] Vassanelli, Mahmud, Girardi and Maschietto, "On the way to large-scale and high-resolution brain-chip interfacing," *Springer*, 2012.
- [23] Cechetto, "Current source density analysis of high resolution local field potentials recorded by implanted neurochips in the rat's brain," *Tesi di Laurea*.
- [24] J. Jerome, "Virtual instrumentation using labview," PHI Learning Pvt, 2010, pp. 12-19.
- [25] N. Instrument, LabVIEW Core 1, user manual, 2012.
- [26] N. Instrument, LabVIEW Core 2, user manual, 2012.
- [27] N. Instrument, Data acquisition and signal condition, user manual, 2012.
- [28] I. National, "Memory Management for Large Data Sets," June 2010. [Online]. Available: http://zone.ni.com/reference/en-XX/help/371361G-01/lvconcepts/memory_management_for_large_data_sets/. [Accessed 2015].
- [29] N. Instrument, LabVIEW real-time 1, user manual, 2012.
- [30] A. e. a. Belitski, "Low-frequency local field potentials," *Journal of Neurophysiology*, p. 5696–5709, 2008.

