ALMA MATER STUDIORUM – UNIVERSITA' DI BOLOGNA
SCUOLA DI INGEGNERIA E ARCHITETTURA

Corso di Laurea Magistrale in
INGEGNERIA BIOMEDICA

Algoritmi di Spike Detection per applicazioni neuroprotesiche: sviluppo di modelli, implementazione e valutazione delle performance

Tesi di Laurea Magistrale in
Bioingegneria della Riabilitazione

RELATORE:
Prof. Chiari Lorenzo

PRESENTATA DA:
Scandellari Carolina

CORRELATORI:
Dott.ssa Chiappalone Michela

Dott. Averna Alberto

Dott. Buccelli Stefano

SESSIONE III
ANNO ACCADEMICO 2018-2019

Alle mie amiche,

Realizziamoci.

Indice

Introduzione

Un report dell'Organizzazione Mondiale della Sanità (OMS, World Health Organization - WHO) del 2006 (WHO, 2006) ha mostrato come i disordini neurologici costituiscano il 6,3% delle cause di malattia in tutto il mondo con l'incidenza maggiore in Europa (11,2%). I disordini legati al sistema nervoso rappresentano una delle più invalidanti condizioni cliniche insieme a malattie come HIV, ischemie cardiache ecc. Tra i disordini neurologici, quello con maggiore incidenza è l'ictus, 55%, seguito da demenza e Alzheimer, 12%, emicrania, 7.9%, ed epilessia, 7% (Panuccio et al., 2018) Il numero di persone colpite dalle prime due patologie è destinato ad aumentare a causa dell'invecchiamento generale della popolazione, mentre le altre condizioni cliniche colpiscono persone di ogni età e ne limitano non poco le attività quotidiane. Il ripristino delle funzioni cognitive, sensoriali e motorie in queste condizioni è dunque diventato una delle priorità della sanità globale.

Oltre ai tassi d'incidenza di queste malattie, bisogna ricordare che alcuni pazienti possono essere resistenti ai farmaci, come nel caso dell'epilessia. Per altre patologie invece i farmaci attualmente presenti sul mercato non sono in grado di curare direttamente la malattia, ma tendono ad attenuarne i sintomi, causando talvolta effetti indesiderati anche molto gravi. Questo accade ad esempio per le cosiddette sindromi da disconnessione (nelle quali possiamo far rientrare anche l'ictus). Una sindrome da disconnessione indica una patologia in cui, a causa di una lesione, aree cerebrali prima comunicanti non sono più in grado di scambiare informazioni (dirette o indirette) l'una con l'altra. Gli scambi diretti di informazioni prevedono una connessione anatomica (*anatomical connectivity*) tra i neuroni, gli scambi indiretti invece si riferiscono alla correlazione tra neuroni differenti anche in assenza di

un legame (*functional connectivity*) o alla diretta influenza che una popolazione di neuroni esercita su un'altra (*effective connectivity*) (Poli et al., 2015). E'pertanto prioritario che la ricerca si concentri nel trovare soluzioni alternative alla terapia farmacologica che siano efficaci per trattare tali patologie.

Una concreta opportunità è fornita dall'approccio Neuroingegneristico, che sfrutta la tecnologia per proporre soluzioni innovative per il trattamento dei disordini del sistema nervoso. Tra le varie soluzioni, sono presenti le neuroprotesi, ovvero dispositivi in grado di sostituire un'area danneggiata del cervello o di ricollegare in modo artificiale due aree disconnesse per ripristinarne le funzionalità, bypassando la lesione che ha causato il danno. Diversi tipi di neuroprotesi cerebrali sono stati presentati in letteratura, ad esempio una protesi ippocampale per il ripristino della memoria (Berger et al., 2011; Hampson et al., 2018) oppure la neuroprotesi del cervelletto per il ripristino della memoria motoria (Herreros et al., 2014). Tra questi, il dispositivo sviluppato presso la University of Kansas (Guggenmos et al., 2013) si è dimostrato essere efficace in esperimenti effettuati su modello animale con lesione focale in area motoria. Grazie all'efficacia della neuroprotesi, gli animali impiantati presentavano un recupero motorio in seguito alla lesione, risultato che fa ben sperare in una possibile riabilitazione da questo tipo di danno cerebrale. Il funzionamento di questo dispositivo è basato sull'impianto di due matrici di micro-elettrodi, posizionate in due regioni cerebrali disconnesse a causa della lesione. Una matrice è deputata alla registrazione del segnale neuronale in tempo reale e l'altra alla stimolazione elettrica della corteccia cerebrale. Quando il primo dispositivo registra un evento nell'area cerebrale di registrazione somministra una scarica di corrente nell'altra area disconnessa dalla prima attraverso un elettrodo, andando così a creare un ponte artificiale in grado di riconnettere le due aree scollegate dalla lesione. Risulta evidente che, in questo tipo di dispositivi, sia importantissimo effettuare una identificazione corretta degli eventi neuronali, escludendo ogni forma artefattuale e l'eventuale rumore ambientale.

Il mio lavoro di tesi si inserisce nell'ambito della collaborazione tra il Rehab Technologies Lab (IIT, Genova, Italy) e la University of Kansas (KUMC, Kansas City, KS, USA), in relazione al progetto per lo sviluppo di neuroprotesi innovative per il recupero motorio a seguito di danni cerebrali. L'obiettivo primario consiste nello sviluppo e nella validazione di diversi algoritmi di detezione degli eventi elettrofisiologici significativi (i.e. spike) al fine di migliorare il funzionamento delle neuroprotesi. Nello specifico, il mio lavoro di Tesi si concentra sulla *Spike Detection.* Uno dei problemi fondamentali della Spike Detection è la mancanza di un *ground truth*, ovvero di una conoscenza a priori della precisa localizzazione degli spikes nel tracciato. Per sopperire a questa mancanza, si può pensare di sviluppare modelli computazionali il più realistici possibili, di cui si conoscano tutte le proprietà, su cui poi andare a testare gli algoritmi di Spike Detection per valutarne le prestazioni.

Nel contesto descritto sopra, si inseriscono gli **obiettivi** di questa Tesi, che sono riportati di seguito:

1. **Fornire** un ground truth tramite lo sviluppo di un modello computazionale di segnale neurale.
2. **Studiare** e adattare un set di algoritmi di Spike Detection già presenti in letteratura.
3. **Modificare** un algoritmo ad alte prestazioni sviluppato all'interno di IIT in passato (ma impiegato per applicazioni diverse) per adattarlo alle caratteristiche dei segnali in esame.
4. **Confrontare** le prestazioni di tutti gli algoritmi di Spike Detection.

Questo lavoro di Tesi riporta i risultati ottenuti durante la mia esperienza di Tirocinio presso il Rehab Technologies Lab.

La Tesi è organizzata nei seguenti capitoli: il Capitolo 1 illustra alcune delle basi delle Neuroscienze quali i potenziali d'azione e le registrazioni elettrofisiologiche, insieme alle applicazioni della

Neuroingegneria; il Capitolo 2 spiega la modellazione del segnale neuronale e riporta le problematiche in relazione alla discriminazione delle singole unità neuronali; il Capitolo 3 riporta una overview dei più comuni metodi di Spike Detection utilizzati allo stato dell'arte e spiega il funzionamento degli algoritmi scelti in questo lavoro; il Capitolo 4 illustra il modello di segnale generato; il Capitolo 5 spiega le modifiche apportate all'algoritmo sviluppato da IIT; il Capitolo 6 riporta i risultati ottenuti e i confronti tra questi. Un Capitolo di Conclusioni discute i risultati del lavoro svolto e le prospettive future dello stesso.

Capitolo 1: Neuroscienze e Neuroingegneria

Le Neuroscienze si occupano dello studio del sistema nervoso a diversi livelli di complessità. Si tratta di un campo di ricerca multidisciplinare che comprende aspetti inerenti alla biologia, fisiologia, anatomia, biologia molecolare, biologia dello sviluppo, citologia, modelli matematici e psicologia per capire le proprietà dei neuroni e delle reti neurali.

La Neuroingegneria è un'area di ricerca interdisciplinare che coniuga i metodi delle Neuroscienze e dell'Ingegneria per analizzare funzioni neurologiche e creare soluzioni a problemi associati a limitazioni e disfunzioni cerebrali (Durand, 2006). La Neuroingegneria sfrutta quindi aspetti sperimentali, teorici, computazionali e clinici, per risolvere i problemi delle Neuroscienze a livello molecolare, cellulare e di sistema.

Questo capitolo illustra alcune delle basi di Neurofisiologia e di Neuroingegneria utili a contestualizzare il mondo della spike detection.

1.1 Basi di Neurofisiologia

Il Sistema Nervoso è costituito dal complesso delle strutture che permettono all'organismo di ricevere stimoli dall'ambiente, di elaborarli e di reagire velocemente e in modo appropriato ad essi. La sua unità di base è il neurone, che comunica le informazioni attraverso i potenziali d'azione.

1.1.1 Neuroni e Potenziali d'Azione

Il potenziale d'azione, detto anche *spike*, è il segnale utilizzato dal sistema nervoso per ricevere, analizzare e trasmettere informazioni. Questi segnali vengono prodotti dai neuroni, unità funzionali del sistema nervoso.

I neuroni vengono classificati in moltissimi modi, specialmente quelli del cervello, ma la loro struttura principale non cambia, anche se esistono diverse forme (Fig.1.1), ed è composta da un polo ricevitore (dendriti), un polo di trasmissione (corpo cellulare) e da uno stelo conduttore (assone). L'assone può essere ricoperto da una guaina lipidica di mielina, interrotta ad intervalli regolari in zone chiamate nodi di Ranvier, che velocizza la trasmissione del segnale. La lunghezza degli assoni è molto variabile, a seconda della funzione che deve svolgere il neurone.

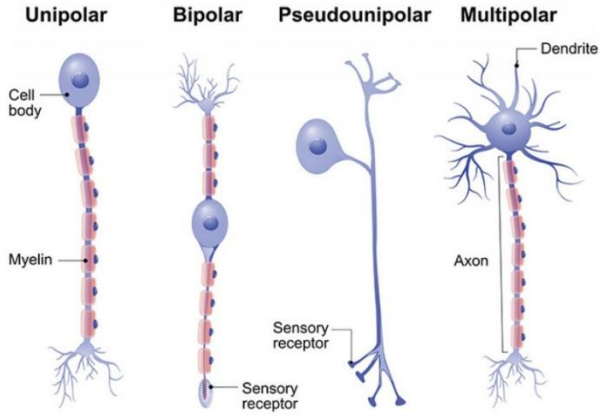


Fig.1.1 Diverse tipologie di neuroni. In ordine da sinistra a destra: unipolari, bipolari, pseudo-unipolari, multipolari. Cellule unipolari: caratteristiche del sistema nervoso degli invertebrati, i segmenti dell'assone possono fungere anche come superficie recettiva oltre che come zona di trasmissione. Cellule bipolari: il dendrite porta informazioni alla cellula e l'assone trasporta le informazioni provenienti dalla cellula. Cellule pseudo-unipolari: sottoclassi di cellule bipolari, si tratta delle cellule dei gangli delle radici dorsali del midollo spinale che portano informazioni di natura sensitiva al sistema nervoso centrale. Cellule multipolari: possiedono molti dendriti e costituiscono il tipo di neurone più comune del sistema nervoso centrale dei mammiferi. Immagine tratta da https://www.alzheimer-riese.it/images/stories/ArticlePics/different-kinds-of-neurons.jpg

Il potenziale d'azione (Fig.1.2) origina nella zona a cavallo tra il corpo cellulare e l'assone, detta cono d'emergenza. Le ramificazioni finali dell'assone costituiscono l'elemento pre-sinaptico e stabiliscono le connessioni sinaptiche con i dendriti o con il corpo cellulare di altri neuroni, elementi post-sinaptici. Le ramificazioni di ogni assone possono fare sinapsi con altri neuroni, fino a 1000 (Kandel et al., 1981). Il potenziale d'azione si trasmette da un neurone all'altro proprio tramite la sinapsi, cioè i contatti funzionali (e non fisici, le sinapsi avvengono infatti in una fessura che separa la terminazione pre-sinaptica dalla cellula post-sinaptica) tra i neuroni. Queste possono essere elettriche, quando consentono un flusso diretto e passivo di corrente elettrica da un neurone all'altro, o chimiche, quando la comunicazione tra i neuroni è veicolata da neurotrasmettitori.

Il flusso di informazione segue le regole della polarizzazione dinamica formulata da Cajal, che afferma che i messaggi nervosi in ogni neurone seguono una direzione costante e prevedibile: dai dendriti alla zona d'innesco (cono d'emergenza) dove insorge il potenziale d'azione, che si propaga unidirezionalmente lungo l'assone, verso le ramificazioni di questo dove viene liberato il neurotrasmettitore. La propagazione lungo l'assone è caratterizzata da una rigenerazione continua del potenziale d'azione che avviene grazie al fenomeno della conduzione, e che gli permette di viaggiare immutato anche per grandi distanze.

I potenziali d'azione presentano tre caratteristiche principali:

1. Forma stereotipata: la forma del potenziale d'azione è caratteristica per ogni neurone.
2. Presenza di un potenziale di soglia: si tratta di un evento *tutto o del nulla*, cioè: se il segnale che arriva a livello del cono d'emergenza non supera un certo voltaggio (definito potenziale di soglia) allora lo spike non avviene affatto. Se il potenziale di soglia viene superato, l'intensità dell'ingresso non ha più influenza e lo spike si genera univocamente.

3. Presenza dei periodi refrattari: il periodo refrattario è quel periodo di tempo, della durata di circa 1-2 ms, in cui anche in presenza di uno stimolo sopra-soglia non viene generato uno spike. I periodi refrattari sono di due tipi: assoluto e relativo, nel primo caso nessuno stimolo in ingresso sarà mai in grado di generare uno spike e il neurone si comporta come se fosse una cellula non eccitabile. Al termine del periodo refrattario assoluto, è presente il periodo refrattario relativo, in cui uno stimolo di intensità abbastanza forte da superare il nuovo livello di soglia (più elevato del precedente), può generare uno spike.

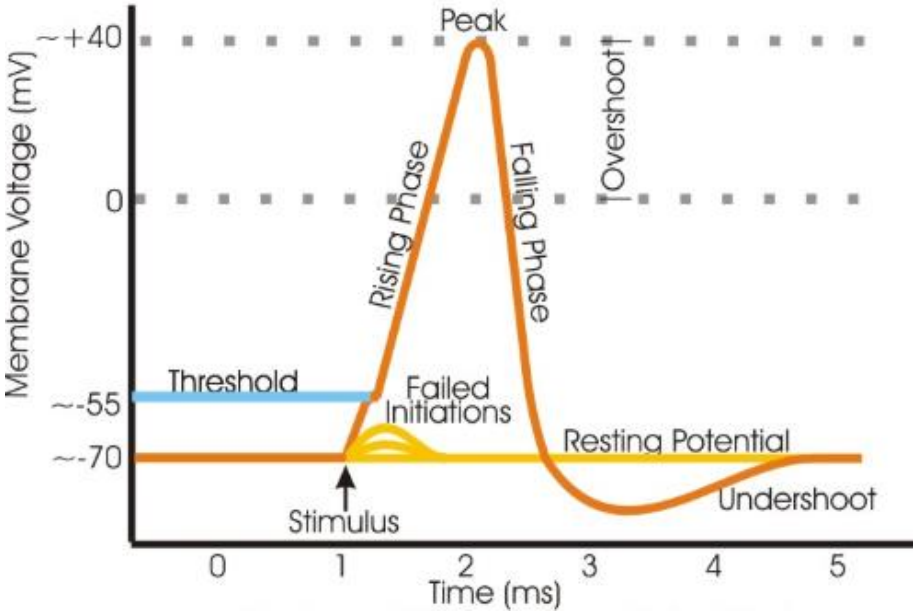


Fig.1.2 Rappresentazione schematica del potenziale d'azione. In arancione è rappresentato l'andamento del potenziale di membrana che parte dal potenziale di riposo (-70 mV). Se il segnale in ingresso riesce a farne aumentare il valore fino al raggiungimento della soglia, si scatena l'evento, altrimenti, in giallo, non si scatena nulla. L'aumento del potenziale di membrana oltre a 0 mV viene chiamato overshoot, il superamento in negativo del potenziale di riposo è definito undershoot. Immagine tratta da https://it.wikipedia.org/wiki/Potenziale_d%27azione

Per spiegare come avviene il potenziale d'azione (Fig.1.2) bisogna prima introdurre il potenziale di membrana, ovvero il potenziale elettrico che si determina ai due lati della membrana cellulare a causa di una diseguale distribuzione di ioni. Gli ioni

utilizzano i canali ionici presenti sulla membrana del neurone, per entrare/uscire dalla cellula. I canali ionici sono infatti dotati di cancelli che si aprono o chiudono in risposta a stimoli quali la presenza di neurotrasmettitori o una variazione nella differenza di potenziale. Il potenziale di membrana a riposo per un neurone è pari circa a -70 mV, valore per il quale sono chiusi sia i canali del sodio che del potassio (si tratta di canali voltaggio-dipendenti). Uno stimolo che arriva nella zona d'innesco modifica il valore del potenziale di membrana, depolarizzandola. Se la depolarizzazione fa arrivare il potenziale di membrana fino al valore di soglia (circa -50 mV) allora si aprono i canali del sodio che permettendo l'entrata di ioni Sodio, intensificano la depolarizzazione con un meccanismo a feed-back positivo (Fig.1.3) fino ad arrivare al valore di 30 mV. A questo voltaggio, si chiudono i canali del sodio e si aprono quelli del potassio che facendo uscire ioni potassio dalla cellula, favoriscono una ripolarizzazzione della membrana, facendone scendere il potenziale fino a -90 mV (iper-polarizzazione). A questo voltaggio, si chiudono anche i canali del potassio e il potenziale di membrana torna al valore di riposo.

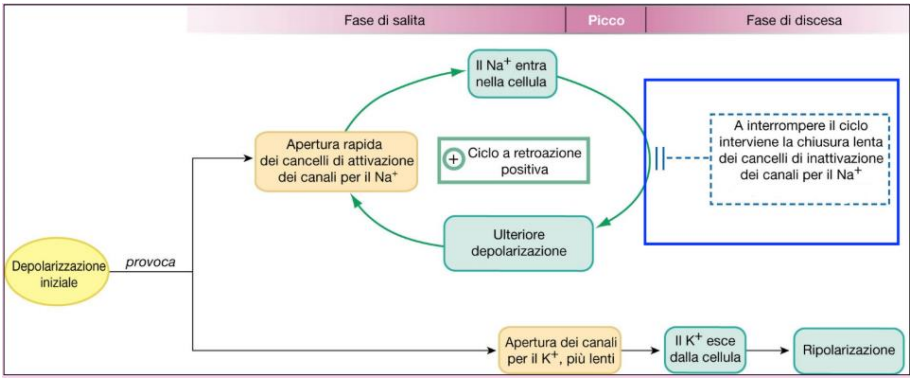


Fig.1.3 Meccanismo di funzionamento del potenziale d'azione. Una depolarizzazione iniziale provoca una repentina apertura dei cancelli del sodio che entrano in un circolo a feed-back positivo interrotto dai meccanismi di inattivazione degli stessi cancelli. Più lentamente si aprono i canali del potassio che ripolarizzano la cellula. Immagine modificata dalle slide di Fisiologia, prof.ssa Benfenati, Università di Bologna.

Questi complicati processi biochimici possono essere investigati tramite tecniche elettrofisiologiche. L'elettrofisiologia è infatti quel ramo delle neuroscienze che studia l'attività elettrica dei

neuroni e in particolare i processi molecolari e cellulari che stanno alla base dei loro segnali. Lo scopo è decodificare i «messaggi» neuronali per rispondere a domande a livello sistemico (come ad esempio il ruolo di un singolo neurone in una rete) ma anche per studiare specifici canali ionici, potenziale di membrana, e le molecole che danno ad ogni tipologia di neurone le sue caratteristiche (Carter & Shieh, 2015). Questi segnali vengono registrati grazie alla differenza di potenziale che si genera tra i due elettrodi che si utilizzano per le misurazioni, l'elettrodo di riferimento e quello di registrazione. A seconda del tipo di registrazione che viene effettuata, se intera o esterna alla cellula, i segnali visualizzati sono di tipo diverso (Fig.1.4)

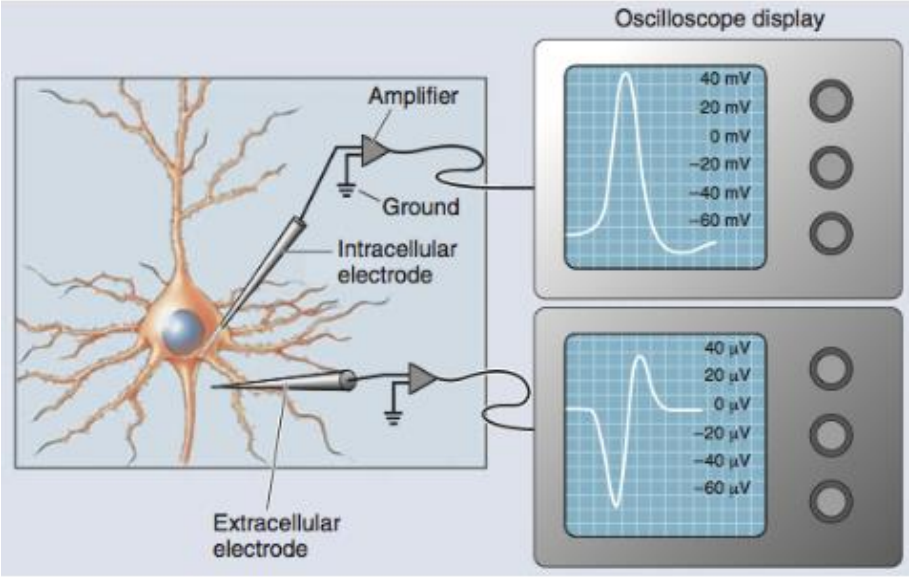


Fig.1.4 Schematizzazione di diverse visualizzazioni di uno spike. Nel caso di registrazione intracellulare tramite elettrodo intracellulare, lo spike viene visualizzato con una forma d'onda che va circa da -70 mV a 40 mV. Nel caso di registrazione extra-cellulare (con elettrodo extracellulare, la forma d'onda dello spike registrato varia e presenta un'ampiezza di tre ordini di grandezza inferiore. Figura tratta da Neuroscience in Action, https://canvas.brown.edu/courses/851434/pages/measuring-brain-signals?module_item_id=6410921

1.1.2 Registrazioni intra-cellulari: Voltage Clamp e Current Clamp

Il *Voltage Clamp* e il *Current Clamp* sono metodi di registrazione intra-cellulare, cioè con l'elettrodo che penetra la cellula. Questi due tipi di registrazione possono essere effettuati anche con un'altra configurazione, detta *Patch-Clamp* (Fig.1.5), che prevede che l'unità di registrazione, in questo caso una micro-pipetta in vetro, sia aderente alla membrana. Il Patch-Clamp crea meno danni alla cellula rispetto alle comuni misure intracellulari e permette dunque di eseguire migliori misurazioni.

Voltage Clamp

Questa tecnica è stata utilizzata da Hodgkin e Huxley per studiare i periodi refrattari e le costanti di tempo di attivazione e inattivazione dei canali ionici. Gli studiosi hanno determinato sperimentalmente per punti le funzioni che governano l'apertura e la chiusura dei canali effettuando degli esperimenti a blocco di voltaggio (voltage clamp) sull'assone gigante del calamaro (Hodgkin & Huxley, 1952). Questi importanti esperimenti hanno permesso di misurare le correnti di sodio e di potassio, da cui è possibile risalire alle conduttanze del sodio, g_{Na}, e del potassio, g_k.

Gli esperimenti in voltage clamp vengono tuttora utilizzati per misurare i cambiamenti di corrente delle membrane cellulari e funzionano misurando la differenza di potenziale rilevata da due elettrodi collegati ad un voltmetro e posti rispettivamente all'interno ed all'esterno dell'assone. Tale tensione, Vm, viene quindi confrontata con un valore di riferimento desiderato (*command signal*): ogni volta che si verifica una differenza fra Vm e il valore desiderato, viene iniettata tramite un terzo elettrodo, una corrente che permette di riportare Vm al valore desiderato. Attraverso questo sistema in retroazione, con guadagno molto alto, il potenziale di membrana viene tenuto praticamente costante al valore voluto, anche al variare delle conduttanze del sodio e del potassio. Infine, un amperometro permette

la misura della corrente iniettata, che coincide con la corrente che attraversa i canali del sodio e del potassio (infatti, essendo Vm costante, la corrente iniettata deve necessariamente uguagliare la corrente delle specie ioniche).

Current Clamp

Il Current Clamp è il caso duale del Voltage Clamp, ovvero viene imposto il valore di corrente e registrato il potenziale di membrana. La cellula viene stimolata tramite un treno di impulsi rettangolari di corrente, forniti da un elettrodo posto intra-cellularmente. Conoscendo la resistenza degli elettrodi intra ed extra-cellulari, è possibile valutare la resistenza di membrana e, quindi il suo voltaggio, attraverso la legge di Ohm.

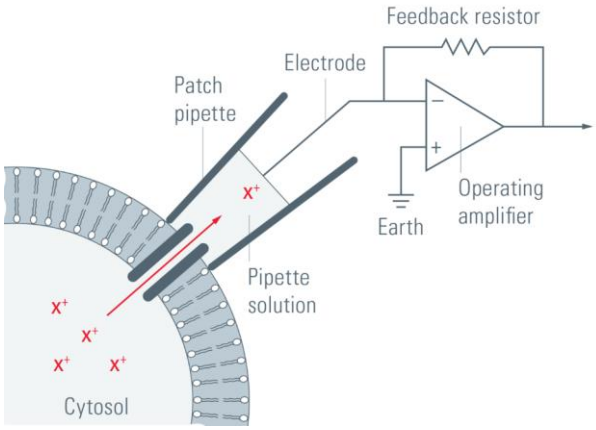


Fig.1.5 Set-up sperimentale di un esperimento in current-clamp con configurazione patch-clamp. La micro-pipetta è in grado di isolare elettricamente una porzione della membrana cellulare, contiene inoltre una soluzione che contiene l'elettrodo collegato ad un aamplificatore. Figura presa da https://www.leica-microsystems.com/science-lab/the-patch-clamp-technique

1.1.3 Registrazioni extra-cellulari

L'elettrofisiologia extra-cellulare, come indica il termine, prevede che la registrazione avvenga all'esterno della cellula. Si tratta di una delle tecniche più importanti e utilizzate per studiare le funzioni cerebrali (Buzsáki et al., 2012). Si tratta di registrazioni ad alta risoluzione spaziale e temporale che permettono di ottenere

informazioni riguardanti lo spiking e l'attività sinaptica dei neuroni nella regione registrata. Queste misure vengono effettuate utilizzando MEAs, *Micro-Electrode Array*s oppure tramite MEMS, *Micro Electro-Mechanical Systems.*

Tra i MEMS si ricordano due dispositivi principali: la sonda Michingan e l'array Utah (Fig.1.6, sinistra). I primi sono dotati di una maggiore densità di sensori e sono caratterizzati dal fatto che la parte sensibile degli elettrodi si sviluppa lungo tutta la loro lunghezza e non solo in punta. Gli array Utah sono matrici tri-dimensionali di 100 elettrodi in silicio disposti in configurazione 10X10 e con parte sensibile solo sulla punta. Questi dispositivi sono intra-corticali, ovvero penetrano la corteccia cerebrale, e dunque presentano problematiche legate alle condizioni a lungo termine degli impianti, che possono causare una forte risposta immunitaria.

I MEAs sono dispositivi che contengono anche centinaia di micro-elettrodi attraverso i quali i segnali elettrici vengono registrati o somministrati. A seconda del tipo di segnale che viene registrato, in-vitro o in-vivo, si utilizzano MEAs diversi (Fig.1.6).

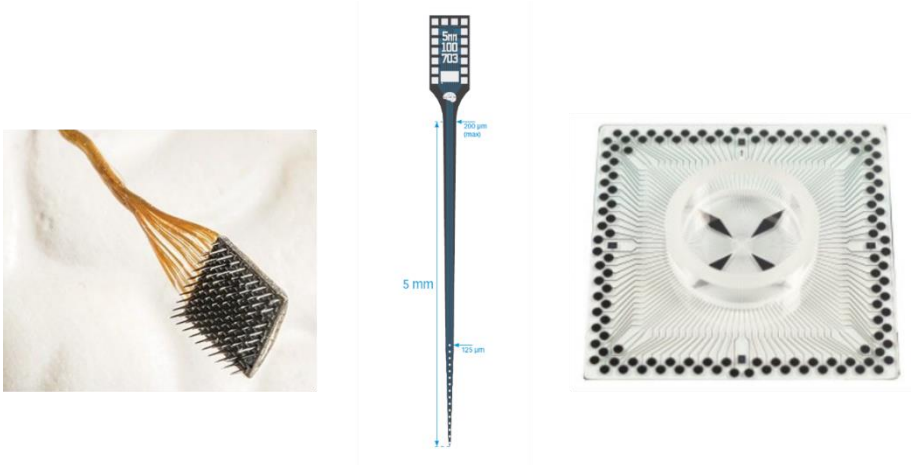


Fig.1.6 Diversi tipi di dispositivi di registrazione. A sinistra, array Utah con i 100 elettrodi in silicio. Al centro, MEAs per registrazioni in-vivo con elettrodi in punta ad alta densità della NeuroNexus. A destra, MEAs per registrazioni in-vitro da culture cellulari.

Nelle applicazioni in-vitro (Fig.1.6, destra), questi array sono utilizzati per culture cellulari o fette di tessuto cerebrale. Il numero di

elettrodi presenti varia a seconda del tipo di applicazione: ci sono modelli che prevedono l'utilizzo di 60 elettrodi, altri 120 oppure 256. Negli ultimi dieci anni, sono comparsi MEAs per applicazioni in-vitro accoppiati con una circuiteria CMOS, il che ha permesso di registrare da più di 4000 elettrodi contemporaneamente, riuscendo a discriminare anche le singole unità neuronali (Stringer et al., 2018). Attraverso questi dispositivi è possibile inoltre stimolare elettricamente le cellule e osservare la propagazione della corrente con risoluzione spaziale e temporale elevata.

L'acquisizione dei segnali neuronali extracellulari (Fig.1.7), prevede un'amplificazione del segnale e un filtraggio passa-banda. Per i neuroni che sono vicini alla punta dell'elettrodo, nel raggio di 50-100 µm (Buzsáki et al., 2012), il *Signal to Noise Ratio* (SNR) è abbastanza buono da permettere la discriminazione di ogni neurone (o unità). Per neuroni più distanti (fino a 150 µm), l'elettrodo è ancora in grado di percepire gli spikes, ma il loro segnale è maggiormente coperto dal rumore. Neuroni ancora più lontani non danno un contributo dal punto di vista degli spikes ma solo come "rumore di sottofondo". Queste registrazioni sono dette Multi Unit Activity (MUA) in quanto comprendono gli spikes provenienti da più neuroni, a differenza da registrazioni provenienti da un solo neurone, Single Unit Activity (SUA). Per discriminare le singole unità dei MUA si utilizza una tecnica chiamata *Spike Sorting*. Il capitolo 2 presenta ulteriori dettagli sulla procedura di analisi del segnale extra-cellulare.

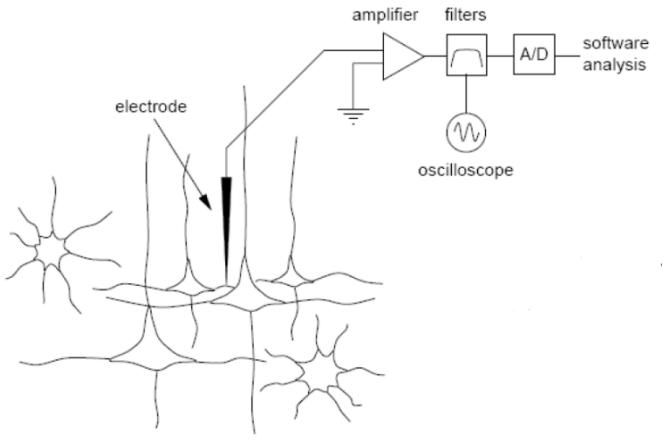


Fig.1.7 Schema di acquisizione del segnale neurale extra-cellulare. Il segnale acquisito viene amplificato, filtrato, digitalizzato e poi analizzato.

1.1.4 Stimolazioni intra-cellulari ed extra-cellulari

Ad oggi sappiamo che l'applicazione di una corrente elettrica ad una cellula eccitabile, quale un neurone, produce uno spike. La stimolazione elettrica esterna delle cellule nervose è stata effettuata per la prima volta da Luigi Galvani nel 1791 (Galvani, 1791), ma la vera comprensione del fenomeno non è stata possibile fino alla scoperta della membrana cellulare del primo '900. Grazie poi agli esperimenti di Hodgkin e Huxley (Hodgkin & Huxley, 1952) degli anni '50 si è scoperto che, durante uno spike, la corrente causa una depolarizzazione da un lato della membrana cellulare e una iper-polarizzazione dal lato opposto della stessa. Se questa depolarizzazione, causata da una corrente esterna, raggiunge il valore di soglia, allora la cellula che è stata eccitata genera uno spike. Come nel caso delle registrazioni, anche le stimolazioni possono essere interne alla cellula o esterne ad essa.

Stimolazioni intra-cellullari

Uno dei metodi più semplici per effettuare una stimolazione intra-cellullare è l'utilizzo di due micro-elettrodi che penetrano la cellula (Patterson & Kesner, 1981). Uno di questi due sarà l'elettrodo di stimolazione e sarà quindi collegato ad una sorgente di corrente, l'altro, quello di registrazione, sarà invece collegato ad un amplificatore ad alta impedenza (Fig.1.8).

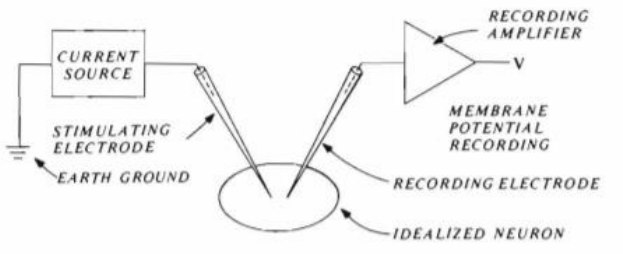


Fig1.8. Schema di apparato per stimolazione intra-cellulare con due micro-elettrodi. A sinistra è presente l'elettrodo di stimolazione collegato a una sorgente di corrente,

a destra invece è presente l'elettrodo per la registrazione collegato ad un amplificatore.Tratto da Capitolo 2 di (Patterson & Kesner, 1981)

Stimolazione e registrazione possono anche essere eseguite tramite l'utilizzo di un solo elettrodo. In questa configurazione, il singolo elettrodo è collegato a turno ad un amplificatore per effettuare la registrazione, e ad un generatore di corrente costante per effettuare la stimolazione. Il generatore di corrente è solitamente formato da un generatore di voltaggio in serie a un resistore di corrente (Araki & Otani, 1955).

Queste strategie hanno permesso lo sviluppo di più recenti tecniche di stimolazione di singoli neuroni in-vivo, come quelle effettuate da Doron e Brecht (Doron & Brecht, 2015). La stimolazione intra-cellulare fornisce un ottimo controllo sulla generazione degli spike, sia dal punto di vista delle forme d'onda che del tempo di sparo, ma sono molto difficili da realizzare su modelli in-vivo. Il loro gruppo ha sviluppato quindi un nuovo approccio chiamato *nanostimolazione* basato sull'iniezione di correnti juxtacellulari, cioè tramite elettrodi posti molto vicino alla cellula ma extracellularmente.

Stimolazioni extra-cellulari

La stimolazione extra-cellulare prevede che lo stimolo elettrico sia somministrato tramite degli elettrodi esterni ai neuroni e non internamente a questi. Si tratta di una stimolazione non invasiva indotta dall'applicazione di rampe di voltaggio da parte dell'elettrodo. L'eccitazione neuronale è dovuta all'aumento del potenziale dell'ambiente extra-cellulare che depolarizza la membrana fino al raggiungimento della soglia che genera lo spike.

1.2 Basi di Neuroingegneria

Uno degli obiettivi che si pone la Neuroingegneria è proporre soluzioni riabilitative per danni e patologie del sistema nervoso. Come riportato in letteratura (Durand, 2006) gli scopi e le principali applicazioni di questa disciplina sono:

- Brain-Machine Interface (BMI) o Brain-Machine-Brain Interface (BMBI) o Brain-Computer Interface (BCI)
- Neuromodulazione e neuroprotesi
- Neuro-riabilitazione e neuro-terapia
- Sistemi neuro-meccanici
- Neuro-robotica
- Neuro-informatica
- Rigenerazione del tessuto neurale
- Neuroscienze teoriche e computazionali

Questo tipo di dispositivi si basa su tecnologie ingegneristiche (mutuate dalla teoria dei controlli) che sfruttano il trattamento dei segnali elettrofisiologici secondo diverse modalità. Nel seguito quindi verranno presentate le nozioni di base che vengono sfruttate in questo campo come le strategie di controllo in open o closed-loop, le tecniche di stimolazioni invasive e non ed infine alcune delle principali applicazioni della Neuroingegneria.

1.2.1 Open-loop e Closed-loop

I dispositivi sviluppati per interfacciarsi con il cervello possono essere implementati con diverse configurazioni, tipicamente in open-loop o closed-loop. Queste configurazioni si differenziano per la presenza nella seconda di un controllo in feed-back (Fig.1.9). Entrambe le configurazioni sono costituite da due unità: il dispositivo (Device, D), e il tessuto neuronale o cerebrale (Brain, B), caratterizzati ciascuno dalla propria funziona ingresso/uscita (Input/Output, I/O). Nel caso del dispositivo la funzione sarà chiamata I_D/O_D (Input Device/Output Device), per la controparte biologica invece I_B/O_B (Input Brain/Output Brain).

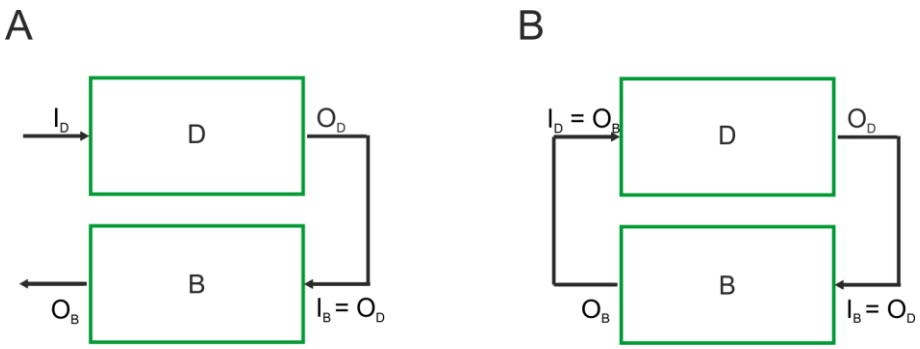


Fig.1.9 Rappresentazione schematica delle architetture in open-loop e closed-loop. A) Open-loop: l'output del dispositivo (O_D) diventa l'input del cervello (I_B). B) Closed-loop: O_D diventa I_B, il cervello produce un'uscita (O_B), che diventa l'ingresso del dispositivo (I_D). (Averna, 2019)

Molti dispositivi per applicazioni neuroprotesiche hanno una configurazione in open-loop, che non risponde a esigenze o cambiamenti dell'ambiente in cui opera, infatti O_B non ha effetti sul dispositivo che agisce sul sistema nervoso stesso (e quindi su B). Questa caratteristica penalizza la configurazione perché non ha il controllo sullo stimolo che fornisce, in quanto lo stimolo viene erogato indipendentemente dall'ambiente circostante. Nelle configurazioni in closed-loop, invece, il segnale di feed-back O_B può essere utilizzato per controllare e regolare il comportamento dell'intero sistema. Questa caratteristica risulta positiva perché permette al dispositivo di agire solo quando è necessario, risparmiando ad esempio in termini di potenza erogata. Per gli esempi riguardanti open e closed-loop, si facciariferimento al paragrafo Applicazioni della Neuroingegneria.

1.2.2 Tecniche di stimolazione invasiva

A livello di applicazione pre-clinica o clinica, le stimolazioni invasive sono dette tali in quanto l'elettrodo erogante la corrente di stimolo penetra il tessuto cerebrale a livello della corteccia, come nel caso dell*'Intra-cortical Micro-stimulation* (ICMS), o nelle zone profonde/sottocorticali (e.g. gangli della base) come nel caso della *Deep Brain Stimulation* (DBS). Di seguito è riportata una breve descrizione per entrambe le modalità.

Intra-cortical Micro-stimulation

La ICMS prevede l'utilizzo di elettrodi ad ago inseriti nel tessuto cerebrale in grado di somministrare correnti elettriche. Questa tecnica è stata utilizzata nei primi anni ottanta per ottenere una mappatura della corteccia motoria (Donoghue & Wise, 1982). Successivamente, questa tecnica ha trovato largo impiego nell'ambito delle BMIs, come strumento per fornire feed-back sensoriale artificiale (Flesher et al., 2016). Recentemente Guggenmos (Guggenmos et al., 2013) ha utilizzato la ICMS per sviluppare un dispositivo in grado di riconnettere due aree cerebrali disconnesse in seguito a una lesione.

Deep-Brain Stimulation

I modulatori cerebrali sono dispositivi che regolano i pattern di attività cerebrale tramite l'applicazione di corrente elettrica o di campi magnetici. La DBS è la più diffusa tecnica di neuromodulazione e viene tipicamente utilizzata su pazienti farmaco-resistenti affetti da disordini motori quali Parkinson, distonia e tremori (Antonini et al., 2018). Attualmente la DBS è utilizzata anche per trattare patologie di altro tipo, come epilessia (Laxpati et al., 2014; Li & Cook, 2018; Kwon et al., 2020), disordini ossessivi-compulsivi, dolori cronici e depressione (Berlim et al., 2014; Coenen et al., 2018; Dandekar et al., 2018).

La DBS prevede una procedura chirurgica per l'impianto di un dispositivo in grado di inviare impulsi elettrici nelle aree limitrofe ai siti di impianto. Nonostante si tratti di una procedura in uso da ormai 60 anni, il suo effetto sulle aree cerebrali non è ancora del tutto chiaro. Inoltre, le caratteristiche del segnale elettrico di stimolazione, come ampiezza e frequenza, devono essere impostate a dovere per avere un risultato positivo, e variano da paziente a paziente. Il protocollo ad oggi più diffuso per la DBS è in *open-loop*, ciò significa che il dispositivo non "ascolta" quanto accade nell'ambiente circostante ma si limita a stimolarlo indipendentemente. Recenti studi (Arlotti et al., 2016; Arlotti et al., 2018; Mohammed et al., 2018) si stanno focalizzando sullo sviluppo di un dispositivo DBS operante in closed-loop, detto adaptive

DBS (aDBS) ovvero in grado di adattarsi e rispondere di conseguenza allo stato dell'ambiente circostante.

1.2.3 Tecniche di stimolazione non invasiva

In questa categoria ricadono vari tipi di stimolazione transcranica quali la *Transcranial Magnetic Stimulation* (TMS), e la sua versione *repetitive TMS* (rTMS), la *Transcranical Direct Current Stimulation* (tDCS), la *Transcranical Alternate Current Stimulation* (tACS) e la *Transcranical Focused Ultrasound Stimulation* (tFUS). Questi metodi sono non invasivi, ossia non prevedono impianti di elettrodi nel tessuto cerebrale. La TMS è stata introdotta nel 1985 da Barker (Barker et al., 1985), si tratta di una tecnica in grado di eccitare focalmente popolazioni di neuroni e quindi adatta all'osservazione dell'eccitabilità e connettività delle popolazioni neurali, oppure per interagire con attività neurali spontanee o relative a un task richiesto (Siebner et al., 2009). La tDCS (Nitsche & Paulus, 2000) è stata introdotta quindici anni dopo la TMS, questa tecnica prevede l'applicazione di corrente con bassa ampiezza al cervello attraverso due elettrodi in due aree distinte dello scalpo: la corrente modula l'eccitabilità cellulare dei neuroni a riposo andandone a polarizzare la membrana. Questa tecnica ha una resoluzione temporale molto bassa e molti studi sono stati condotti per capirne la vera efficacia (Brückner & Kammer, 2017). La tACS (Antal et al., 2008) prevede lo stesso principio di funzionamento della tDCS ma con l'utilizzo di una corrente alternata somministrata sottoforma di onda sinusoidale. Si pensa che la natura alternata della stimolazione possa andare a modificare i ritmi endogeni del cervello, andando a sincronizzare o desincronizzare popolazioni neurali (Reato et al., 2013). Viene utilizzata quindi nello studio dei ritmi delle frequenze cerebrali (Santarnecchi et al., 2013), con non pochi dibatitti sul suo effetto su uomo (Noury & Siegel, 2017). La tFUS (Tufail et al., 2010), infine, è stata introdotta nel 2010 da in uno studio in cui si dimostrava che la tFUS aveva una risoluzione spaziale maggiore della TMS (la tecnica più focale illustrata sopra).

Questa tecnica sta venendo investigata e sembra essere molto promettente (Tyler et al., 2018).

A livello clinico, queste tecniche vengono impiegate per trattare diverse condizioni neuropsichiatriche come schizofrenia, dipendenze e depressione. La TMS viene studiata anche come protocollo riabilitativo in seguito ad ictus (Smith & Stinear, 2016; Wang et al., 2017; Dionísio et al., 2018).

1.2.4 Applicazioni della Neuroingegneria

I lavori pionieristici degli anni ’70 ad opera di Fetz (Fetz, 1969) e poi alla fine degli anni ’90 da Schwartz (Schwartz, 2004) e Nicolelis (Nicolelis, 2001), hanno fatto sì che ci fosse un incredibile sviluppo dei dispositivi di riparazione del cervello come modulatori cerebrali, interfacce cervello-macchina (*Brain-Machine Interfaces*, BMIs) e protesi cerebrali.

I dispositivi neuroingegneristici per la riparazione del tessuto neuronale possono essere suddivisi in tre macro-categorie, dipendenti dall’obiettivo terapeutico implementato. E’importante sottolineare che la suddivisione non è spesso netta e i confini tra queste categorie sono labili in quanto uno stesso dispositivo può avere più obiettivi terapeutici che possiamo suddividere nelle seguenti categorie:

Rectify (Correzione): A questa categoria appartengono tutti i sistemi che, attraverso la neuromodulazione, sono in grado di bloccare l’attività patologica (ad esempio i tremori nei pazienti con disordini motori oppure gli attacchi epilettici nei pazienti che soffrono di epilessia).

Replace (Sostituzione): In questa categoria ricadono tutti i dispositivi che vanno a sostituire o superare un circuito cerebrale o una parte del corpo non funzionante.

Retrain (Rieducazione): A questa categoria appartengono tutti i sistemi che guidano il cervello a promuovere correttamente i fenomeni plastici, aiutando il cervello a riparare se stesso.

Una caratteristica comune a tutti questi dispositivi è il loro approccio *elettroceutico*, così definito per sottolineare la differenza con il tradizionale approccio *farmaceutico*. L'elettroceutica nasce infatti per sopperire ai bisogni terapeutici di alcuni pazienti farmaco-resistenti (Reardon, 2014).

Vediamo ora alcuni esempi per ogni categoria.

Rectify

Un'applicazione che rientra nella categoria della correzione è legata al trattamento dei disordini motori e dell'epilessia.

Come anticipato, **i disordini motori** vengono trattati con DBS. L'architettura più utilizzata è in open-loop, il dispositivo quindi invia in continuazione scariche elettriche ad alta frequenza tramite dei parametri preimpostati dagli operatori in maniera personalizzata per ciascun paziente. Lo svantaggio principale di questa tecnica consiste nella somministrazione di un carico elevato di corrente che talvolta può non essere necessaria, ma che può causare effetti collaterali, come la dischinesia (Habets et al., 2018). Una possibile soluzione è l'utilizzo di protocolli in closed-loop, utilizzando *adaptive* DBS (aDBS), in cui la stimolazione dipende dal segnale neurale registrato in tempo reale. Questi dispositivi sono attualmente oggetto di numerosi studi, volti ad investigarne le potenzialità come quello effettuato dal gruppo di Habets sull'utilizzo dall'aDBS sul Parkinson (Habets et al., 2018),(Mohammed et al., 2018) e sull'utilizzo dell'aDBS nella routine quotidiana (Arlotti et al., 2016).

L'epilessia è una malattia cronica di cui soffrono circa 50 milioni di persone in tutto il mondo in ogni fascia di età (WHO, 2019). Al momento la terapia più utilizzata è l'assunzione di farmaci anticonvulsivi, ma il 30% dei pazienti non risponde alle terapie farmacologiche. Una soluzione elettroceutica per questo problema è

fornita dal dispositivo *Responsive NeuroStimulation System* (RNS)(Heck et al., 2014) dell'azienda Neuropace, approvato dalla Food and Drug Association nel 2013. Questo dispositivo è uno stimolatore che opera in closed-loop: è dotato di algoritmi che analizzano il segnale registrato in tempo reale e sono in grado di riconoscere quando è in atto un attacco, innescando di conseguenza la produzione di una scarica elettrica.

Replace

In questa categoria rientrano impianti cocleari, protesi visive, neuroprotesi per arti mancanti e neuroprotesi cerebrali/spinali.

Gli impianti cocleari sono dispositivi impiantabili che restituiscono il senso dell'udito in persone con danni severi all'orecchio interno. Si tratta di una tecnologia ormai consolidata e dunque della neuroprotesi più utilizzata al mondo. Il dispositivo è formato da un microfono esterno che detetta i suoni, un processore che elabora il segnale proveniente dal microfono, un trasmettitore collegato a un ricevitore/stimolatore che riceve i segnali e li converte in impulsi elettrici somministrati da una matrice di elettrodi in diverse regioni del nervo uditivo. Questo dispositivo non restituisce l'udito come lo intendiamo comunemente a causa del ridotto numero di elettrodi che codificano i suoni (ne possono essere impiantati solo 16-22) contro le 30.000 fibre del nervo uditivo, ma può fornire alla persona che li utilizza un'utile rappresentazione dei suoni dell'ambiente. Una review sugli impianti cocleari è presente in (Wouters et al., 2015)

Le neuroprotesi visive trovano applicazione nelle patologie ereditarie quali la retinite pigmentosa oppure in caso di patologie relative all'età come la degenerazione maculare. In queste patologie non sono i neuroni della retina ad essere compromessi, bensì i fotorecettori. Il numero di pazienti che utilizza questo tipo di protesi è estremamente inferiore rispetto agli utilizzatori delle protesi cocleari, visto che si tratta di dispositivi di più recente sviluppo. Inoltre, bisogna considerare che l'ambiente biologico dell'occhio è molto più difficile da trattare, in quanto molto ricco di fluidi. Ci sono diverse strategie per

la creazione di queste protesi e nessuna sembra dare vantaggi significativi sulle altre. Queste protesi si dividono in subretiniche ed epiretiniche a seconda di dove sono posizionati gli elettrodi: i dispositivi epiretinici si collocano tra lo strato di cellule gangliari dell'occhio e l'umor vitreo, mentre le protesi subretiniche nella parte posteriore della retina in sostituzione ai fotorecettori. La protesi epiretinica più utilizzata è Argus II (Bloch et al., 2019), è composta da occhiali con fotocamera integrata e collegata a un'unità portatile di analisi delle immagini che processa e comunica le informazioni tramite una bobina esterna, che a sua volta comunica con la sua controparte interna, posta sulla sclera. Una volta ricevuto il segnale tramite radio-frequenza, questo viene ri-trasformato in segnale elettrico e trasmesso ad un dispositivo composto da un array di 60 micro-elettrodi posto sulla retina, che riceve quindi un'immagine corrispondente ai micro-elettrodi attivi. Una review di protesi visive e trial clinici in cui sono utilizzate è prrsente in (Mirochnik & Pezaris, 2019)

Le neuroprotesi d'arto, sia superiori che inferiori, hanno come obiettivo il ripristino del movimento. In questo campo rientrano le BMIs che hanno l'obiettivo di ripristinare la funzione motoria in seguito ad amputazione di un arto, condizioni neurologiche disabilitanti e danni cerebrali. Le BMIs generalmente registrano segnale cerebrale dale aree motorie di interesse in moto da decodificare l'intenzione di movimento, questa informazione è poi trasmessa al dispositivo controllato (ad esempio un arto robotico) che si muove di conseguenza. Alcuni tipi di BMIs operano in closed loop e dunque informazioni sullo stato del dispositivo esterno vengono convertite in un segnale elettrico inviato ad aree sensoriali della corteccia cerebrale. Si tratta di sistemi utilizzati in ricerca e ancora lontani dalla pratica clinica, dove si utilizza il segnale elettromiografico per stabilire il movimento dell'arto robotico. Un interessante caso di neuroprotesi sensoriale riguarda gli studi condotti dal gruppo di Micera per convertire sensazioni tattili registrate da una protesi di arto superiore, in stimolazioni elettriche inviate al nervo del braccio (Micera, 2016; Cutrone & Micera, 2019)

Le neuroprotesi cerebrali sono attualmente oggetto di studio. Nello specifico, sono state sviluppate neuroprotesi per sostituire la funzionalità dell'ippocampo (i.e. neuroprotesi ippocampali) per il ripristino della memoria a seguito di malattie come Alzheimer e demenza senile, oppure per perdita di memoria dovuta a danni cerebrali o farmaci. Si tratta di studi pionieristici e ancora in fase di approfondimento, per cui si è ancora lontani dall'ottenere un effettivo ripristino della memoria (Berger et al., 2011; Hampson et al., 2013; Hampson et al., 2018)

A questa categoria appartiene anche il progetto Europeo BrainBow, che si poneva l'obiettivo di create una neuroprotesi cerebrale innovativa. La neuroprotesi è stata testata

su culture cellulari in vitro coltivate su MEAs. Questo progetto, coordinato dall'Istituto Italiano di Tecnologia (IIT), coinvolgeva l'Università di Genova, l'Università di Tel Aviv e il Centre National de la Recherche Scientifique (CNRS, Bordeaux). Gli esperimenti prevedevano l'utilizzo di un laser per effettuare una lesione tra due popolazioni neurali in-vitro, andandone così a modificarne le connessioni intrapopolazione. E' stato quindi sviluppato un prototipo di neuroprotesi in grado registrare il segnale dalle popolazioni neuronali, analizzarlo e somministrare impulsi di corrente in tempo reale. L'obiettivo consisteva nel riconnettere, tramite la neuroprotesi artificiale, le due popolazioni neurali. fornendo così un proof-of-principle per possibili applicazioni cliniche nel caso di ictus o danni cerebrali focali (Buccelli et al., 2019).

Retrain

In questo gruppo rientrano le protesi cortico-corticali e le protesi spinali. Si tratta di sistemi artificiali collegati direttamente al cervello per sostituire un'area danneggiata o per ricollegare due aree disconnesse e ripristinarne le funzionalità. Questi dispisitivi rientrano nello stadio di sperimentazione preclinica.

Le protesi cortico-corticali vengono utilizzate nei casi di danni traumatici al cervello (*Traumatic Brain Injury*, TBI), o al midollo spinale, e di ictus. Queste protesi si basano sul concetto di plasticità Hebbiana per riconnettere parti cerebrali disconnesse in seguito a lesione, rafforzando specifiche connessioni tra i neuroni. La plasticità Hebbiana (Hebb, 1950) è riassunta nell'espressione di Carla Shatz *"neurons that fire together wire together"*, ovvero neuroni che hanno spike negli stessi istanti temporali tendono a formare una connessione. Questo fenomeno fa sì che due neuroni legati da plasticità Hebbiana vengano eccitati anche se solo uno dei due riceve uno stimolo. Questo fenomeno è stato sfruttato in un esempio di protesi cortico-corticale nella corteccia motoria presentato dal gruppo di Fetz (Jackson et al., 2006). In questo lavoro è stato dimostrato su modello animale che una riorganizzazione stabile della produzione motoria può essere ottenuta attraverso una connessione artificiale tra due siti prima non comunicanti. In altre parole, gli studiosi sono riusciti a sincronizzare due aree cerebrali la cui attività non era prima sincronizzata attraverso l'uso di un dispositivo in grado di effettuare Spike Detection in un'area e ICMS nell'altra area. Ogni volta che veniva registrato uno spike nel sito di registrazione, il dispositivo mandava uno stimolo all'altro sito, forzandolo a sparare. Grazie alla plasticità, dopo giorni di condizionamento, l'area sottoposta a ICMS sparava in concomitanza con l'area su cui veniva fatta la registrazione, anche in assenza del dispositivo. Questo effetto è durato poco più di una settimana dopo l'utilizzo del dispositivo ed interessava solamente le aree direttamente coinvolte dalla neuroprotesi. Numerosi altri studi sono seguiti e sono tuttora in corso per valutare come modellare la plasticità a lungo termine. Lavori analoghi sono stati presentati da Zanos (Zanos et al., 2018) e da Zrenner (Zrenner et al., 2018), quest'ultimo ha effettuato uno studio per verificare una possibile applicazione della plasticità come aiuto alla riabilitazione post-ictus su uomo.

Le prime protesi cerebrali con protocollo closed-loop per utilizzo in caso di lesione cerebrale focale sono state presentate dal gruppo di Nudo (Guggenmos et al., 2013) (Fig.1.10). Gli autori hanno

realizzato un protocollo closed-loop definito *Activity-Dependent Stimulation* (ADS), perchè utilizza l'attività neurale registrata in un gruppo di cellule per triggerare lo stimolo ad un altro gruppo di neuroni, posti in un sito diverso. Questa tecnica ha avuto successo nello stabilire una connessione tra la corteccia somatosensoriale e la corteccia premotoria di un ratto che aveva subito una lesione nell'area motoria, che in condizioni normali fa da ponte fra le due aree. Nello specifico, il dispositivo prevedeva la registrazione e successiva digitalizzazione, dell'attività neurale extra-cellulare attraverso un elettrodo impiantato nell'area premotoria. Il sistema era in grado poi di individuare gli spikes presenti nel segnale e conseguentemente somministrare, solo negli istanti corrispondenti agli spikes, scariche di corrente attraverso un micro-elettrodo impiantato nell'area sensoriale. Si tratta quindi di una *Brain-Machine-Brain Interface* (BMBI), che utilizza ICMS in closed-loop. In questa applicazione, risulta fondamentale identificare correttamente gli istanti dei veri spikes in tempo reale. Questo è solo uno degli esempi in cui un'efficace ed affidabile Spike Detection, oggetto di questa Tesi, risulta fondamentale.

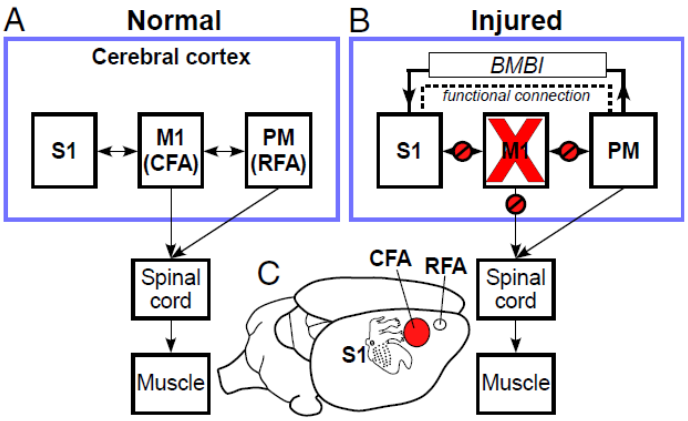


Fig.1.10 Schema dell'approccio utilizzato da Gugenmoss per lo sviluppo del suo dispositivo. A) Connessione in condizioni normali della corteccia somato-sensoriale primairoa (S1), della corteccia motoria primaria (M1) (in particolar modo della caudal forelimb area, CFA) e della corteccia premotoria (PM, in particolar modo la rostral forelimb area RFA) di un ratto. Queste tre aree comunicano tra di loro, M1 e PM comunicano anche con il midollo spinale che provvede poi all'attivazione muscolare. B) Effetti sulle connessioni dopo una lesione a M1. Le connessioni cortico-

corticali tra S1 e PM mediate da M1 non sono più presenti, la comunicazione è ristabilita dalla BMBI, in questo modo la comunicazione al midollo spinale viene preservata anche se solo da parte di PM (Guggenmos et al., 2013).

Neuroprotesi spinali. L'Organizzazione Mondiale della Sanità (WHO, 2013) afferma che ogni anno tra le 250.000 e le 500.000 persone vengono colpite da un trauma al midollo spinale (*Spinal Cord Injury*, SCI). Le protesi spinali sfruttano anch'esse la plasticità neurale ma agiscono direttamente sul midollo spinale per ripristinare la funzione motoria. Il gruppo di ricerca di Courtine (Courtine & Sofroniew, 2019) è da anni impegnato in questo campo per il ripristino della locomozione, utilizzando una strategia che comprende un approccio sia biologico che neuro-tecnologico. Dal punto di vista biologico, si cerca di favorire la ricrescita endogena degli assoni a livello della lesione. L'approccio neuro-tecnologico è realizzato tramite *Targeted Spinal Cord Stimulation*, che consiste nell'utilizzare uno stimolatore impiantato nella spina dorsale lumbosacrale in grado di essere triggerato in tempo reale ogni volta che il soggetto intenda compiere un passo. Questa ricerca ha fatto passi da gigante negli ultimi anni e si è recentemente arrivati alle prime sperimentazioni cliniche su uomo (Bachmann et al., 2013; Bouton et al., 2016; Flesher et al., 2016; Wagner et al., 2018).

Come emerge dalla descrizione di tutti questi dispositivi, è importante avere a disposizione strumenti tecnologici e algoritmici in grado di poter rilevare e processare in tempo reale il segnale neuronale. La Spike Detection è un punto cruciale del processamento del segnale ed è quindi fondamentale avere a disposizione tecniche in grado di rilevare tutti i veri spikes soltanto, ovvero ridurre il più possibile il numero di falsi negativi e falsi positivi.

Capitolo 2: Analisi di segnali neuronali

In questo capitolo è presente una spiegazione del workflow, allo stato dell'arte, dell'analisi dei segnali neurali extra-cellulari.

2.1 Filtraggio

Il segnale neuronale è formato da due componenti (Fig.2.1), in bassa (1-300 Hz) e in alta frequenza (300-3000 Hz). A seconda del tipo di analisi o di impiego che si vuole fare del segnale, ci si concentra su una delle due tipologie che sono i *Local Field Potentials* (LFPs, i.e. segnali in bassa frequenza) e la *Multi Unit Activity* (MUA, i.e. segnali in alta frequenza) che verranno illustrate nei paragrafi successivi.

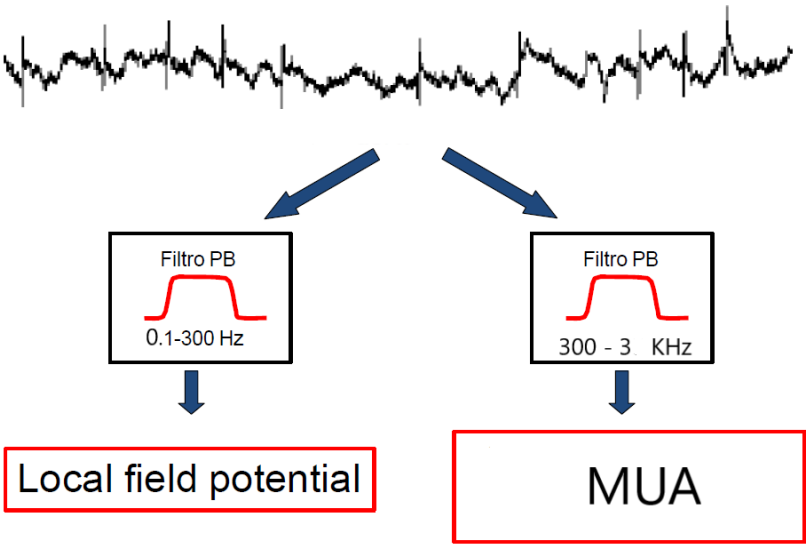


Fig.2.1 Schema esplicativo dei due filtraggi a cui è sottoposto il segnale extracellulare. Il filtro passa-banda 0.1-300 Hz isola i Local Fiedl Potential, il filtro passa-banda 300-3000 Hz isola invece gli spikes di più neuroni (MUA). Figura modificata da slide Prof.Chiari, Universià di Bologna

2.1.1 Local Field Potential

Le registrazioni extra-cellulari sfruttano array di elettrodi dove ogni elettrodo (detto anche canale) registra il segnale nell'ambiente a lui circostante. Ogni neurone contribuisce con il suo stesso voltaggio non ad un solo elettrodo, ma a tutti quelli che gli si trovano vicino. Ogni membrana eccitabile e ogni tipo di corrente trans-membranale contribuisce al campo elettrico extra-cellullare che viene registrato dagli elettrodi (Buzsáki et al., 2012). Il campo formato è quindi una super-imposizione di tutti i processi ionici che avvengono nell'ambiente neurale e viene chiamato *Local Field Potential (LFPs)*. Le caratteristiche degli LFPs come la forma d'onda, l'ampiezza e la frequenza, dipendono dall'inverso della distanza dell'elettrodo dall'evento che viene registrato. Anche il timing con cui avvengono gli eventi contribuisce a cambiare la forma degli LFPs.

Il filtraggio tra 0.1-300 Hz viene effettuato per ottenere il contenuto in frequenza degli LFPs che contiene il rumore di sottofondo e che riflette le dinamiche presenti nel tessuto neurale attorno all'elettrodo (circa 1 mm di diametro attorno a questo (Buzsáki et al., 2012)).

2.1.2 Multi Unit Activity

Il segnale filtrato passa-banda (300-3000 Hz) contiene al suo interno i segnali provenienti da più unità neurali. Questo tipo di registrazione viene definita Multi Unit Activity (MUA, Fig.2.2) ed è relativo alla presenza degli spikes i quali hanno una frequenza più elevata. Grazie a questo filtraggio possono essere visualizzati gli spike sovra-imposti di neuroni vicini. Per ottenere l'attività dei singoli neuroni (*Single Unit Activity*, SUA, Fig.2.2) è necessario effettuare un procedimento chiamato Spike Sorting che verrà illustrato nel paragrafo 2.3.

Analisi del segnale Multi Unit

Arrivati a questo punto è necessario distinguere tra segnale analogico e segnale digitalizzato. Il potenziale d'azione è il segnale biologico, e quindi analogico, lo spike è la sua rappresentazione digitale. I potenziali d'azione solitamente non si presentano come un evento singolo ma sotto forma di sequenza detta treno. Un treno di spike è una sequenza di eventi che possiedono le proprietà di essere indistinguibili e istantanei, cioè gli spike di un singolo treno non si differenziano l'uno dall'altro se non per gli istanti in cui avvengono. I treni di spike possono essere visti come treni di delta di Dirac, dove ad ogni impulso corrisponde un evento. Matematicamente la formalizzazione di un singolo treno (Single Train, ST), quindi una SUA, sarà (Eq.1):

$$ST = \sum_{s=1}^{N} \delta(t - t_s) \qquad \textit{Eq. 1}$$

Dove N è il numero totale degli eventi. Mentre l'attività di più neuroni registrata simultaneamente avrà questa descrizione matematica (Eq.2):

$$ST_j = \sum_{s=1}^{N_j} \delta(t - t_s) \qquad \textit{Eq. 2}$$

Dove j va da 1 a M e M è il numero totale di neuroni presenti.

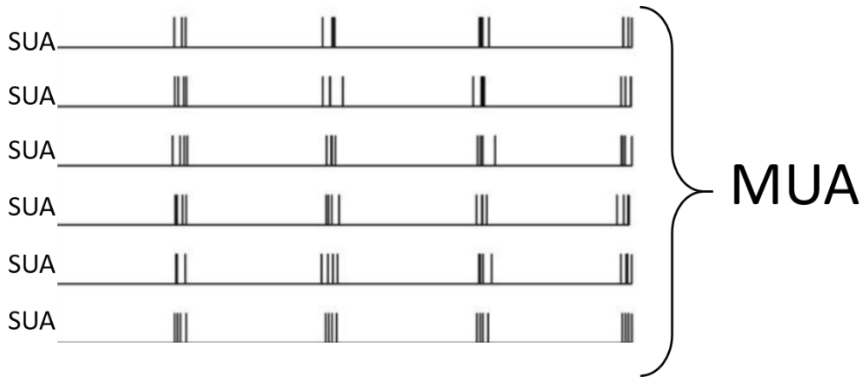


Fig.2.2 Rappresentazione grafica di diversi SUA come spike train che sovraimposti formano il MUA.

L’analisi di una MUA prevede quindi diversi step, riassunti in Fig. 2.3, che sono:

1. Filtraggio del segnale con un filtro passa-banda tra 300-3000 Hz
2. Spike Detection, cioè il riconoscimento da parte di algoritmi di spike nel segnale registrato e poi filtrato
3. Spike Sorting, discriminazione delle forme d’onda con relativa generazione di classi
4. Ottenimento del Single Unit Spike Train

.

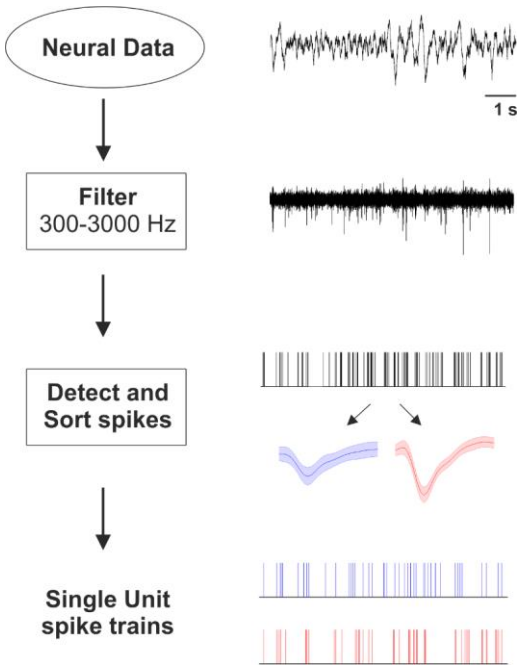


Fig.2.3 Schema del workflow dell’analisi di Multi Unit Activity. Il segnale registrato dagli elettrodi viene filtrato con un filtro passa-banda tra 300 e 3000 Hz per ottenere gli spike. Questi vengono individuati e visualizzati come treno di impulsi di Dirac dove ogni evento corrisponde a una δ. Subiscono la procedura di sorting che li divide in classi, sempre sottoforma di treni di δ per ottenere le Single Unit Activity. Figura tratta da (Averna, 2019)

Vedremo ora più nel dettaglio i passaggi relativi all’analisi della MUA che portano poi all’ottenimento della SUA.

2.2 Spike Detection

La Spike Detection è la procedura che si effettua con algoritmi automatizzati in grado di riconoscere nel segnale gli eventi significativi ricercati, ovvero gli spikes. Questo argomento verrà trattato nel capitolo 3, dove verranno illustrati gli algoritmi di Spike Detection utilizzati allo stato dell'arte.

2.3 Spike Sorting

Lo Spike Sorting è da considerarsi un problema di clustering statistico che porta alla classificazione delle forme d'onda presenti nel segnale in analisi. Come già affermato, il segnale viene registrato da array di elettrodi proviene da Multi Unit (MU), per ottenere le caratteristiche di ogni unità presente (Single Unit, SU), è necessario discriminare le forme d'onda presenti, ricordiamo infatti che ogni neurone possiede la sua forma d'onda caratteristica. Il sorting è un procedimento che permette di capire a quali classi (e quindi a quali unita) appartengono le forme d'onda.

Ottenere le attività dei SU è basilare per comprendere i meccanismi di funzionamento del cervello. L'importanza dello Spike Sorting è evidente in quelle applicazioni in cui è cruciale andare ad analizzare caratteristiche quali la frequenza di scarica e i suoi cambiamenti, le relazioni con altri neuroni e LFPs, ecc. In questi casi l'attività di un singolo neurone può mostrare processi già ben assestati come l'inibizione laterale e la competizione tra unità vicine. Un altro tipo di applicazione in cui è fondamentale andare ad etichettare l'attività dei singoli neuroni è nello studio dei pattern di connettività dei neuroni vicini o nell'organizzazione topografica di un'area cerebrale. L'algoritmo di spike sorting che ho utilizzato in questa Tesi è Waveclus, di cui parlerò nel paragrafo successivo.

2.3.1 Waveclus

L'algoritmo utilizzato per effettuare il sorting è Waveclus (Quiroga et al., 2004), un sistema di spike sorting non supervisionato. La base di questo metodo è l'utilizzo della trasformata Wavelet (WT),

una decomposizione del segnale in tempo-frequenza con un'ottima risoluzione nel dominio dello spazio e del tempo, e del clustering super-paramagnetico (SPC). I passi che effettua l'algoritmo sono:

1. Spike detection
2. Selezione delle feature degli spike
3. Clustering delle spike features

Il primo step è stato in realtà saltato, in quanto all'algoritmo sono già state fornite le posizioni degli spike tramite la Visual Inspection. Nel secondo step, un piccolo set di coefficienti della WT è scelto da ogni spike come input per l'algoritmo di clustering. Ogni coefficiente della WT infatti caratterizza la forma dello spike in differenti scale e tempi, l'obiettivo è scegliere pochi coefficienti per andare a discriminare al meglio le diverse classi degli spike. Scegliere il giusto numero di coefficienti è importante perché se ne vengono scelti troppi, si creeranno troppe classi simili tra loro, se ne vengono scelti troppi pochi allora si creeranno delle macro-classi poco selettive con all'interno membri potenzialmente molto diversi. Questa selezione è effettuata automaticamente tramite un test per la normalità di Lilliefors, il quale utilizza la media e la varianza di una popolazione per andare a controllare che la massima differenza tra la funzione distribuzione empirica e la funzione di distribuzione cumulativa di una distribuzione normale sia statisticamente significativa.

Infine, la SPC classifica gli spike secondo il set di coefficienti della WT selezionato, basandosi sul metodo del K-nearest neighbors e un parametro chiamato temperatura. È presente un'analogia con i modelli *spin glass (*Quiroga et al., 2004*)*, in cui ad alte temperature gli spin cambiano verso randomicamente (fase paramagnetica). A basse temperature, tutti gli spin sono orientati invece nella stessa maniera (fase ferromagnetica). In un certo range di temperature, solo gruppi di spin vicini sono in grado di cambiare direzione simultaneamente (fase super-paramagnetica). Questo accade anche in questo problema di clustering: per basse temperature, tutti i punti cambiano il loro stato e vengono visti come un unico cluster, per alte temperature, invece, i

punti cambieranno indipendentemente l'uno dall'altro. Per le temperature corrispondenti alla fase super-paramagnetica, solo i punti che appartengono allo stesso cluster cambieranno insieme. Il procedimento è raffigurato in Fig.2.4.

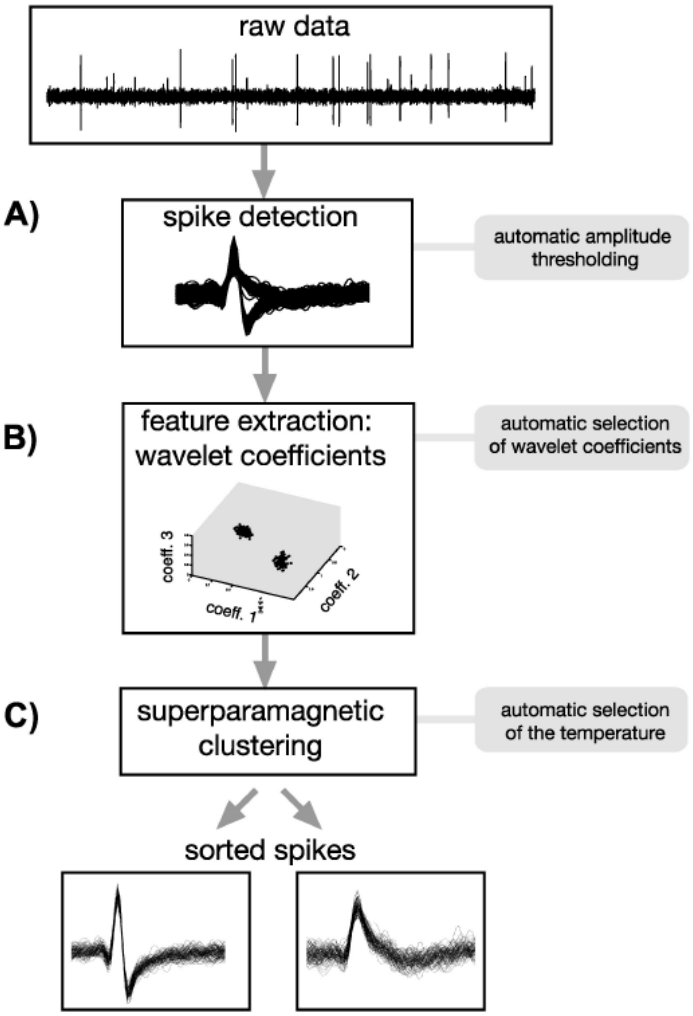


Fig.2.4 Schema del funzionamento di Waveclus. I segnali vengono sottoposti a una sogliatura automatica dell'ampiezza (nel mio caso questo procedimento è stato saltato in quanto la SD era già avvenuta) per effettuare la Spike Detection. Dopodiché viene fatta la selezione automatica dei coefficienti della Wavelet che entrano nel processo del super-paramagnetic clustering che effettua in automatico la scelta della temperatura migliore. In uscita si hanno le classi di spikes presenti nel segnale.Figura tratta da (Quiroga et al., 2004)

2.4 Single Unit Spike Train

Come illustrato nel paragrafo 2.1.2, gli spike train possono essere visti come treni di delta di Dirac. Un treno di delta può essere considerato come un processo puntuale, cioè una sequenza stocastica di eventi temporali dove l'evento è lo spike (Kass, 2018). Questa stocasticità influenza la variabilità degli *Inter Spike Intervals* (ISI), ovvero dagli intervalli tra uno spike e il successivo. Le ISI possono essere modellate attraverso delle distribuzioni, come la distribuzione Esponenziale, Gamma e Gaussiana Inversa. Le ISI influenzano a loro volta il calcolo della frequenza di scarica (*Firing Rate*, FR). Il calcolo della frequenza di scarica media (*Mean Firing Rate*, MFR), ovvero su tutto il tracciato registrato, viene effettuato semplicemente dividendo il numero totale degli spikes per la lunghezza del segnale in questione. Inoltre, il firing delle SU può essere di diverse tipologie.

2.4.1 Distribuzioni degli Inter Spike Intervals

Il modello matematico più semplice per descrivere la variabilità delle ISI è il processo di Poisson omogeneo (Kass et al., 2018) (Eq.3):

$$f(x) = \lambda e^{-\lambda x} \qquad Eq.\ 3$$

Dove λ rappresenta la frequenza di scarica del neurone. Questo processo però non è sufficiente per descrivere l'andamento delle ISI a causa della presenza del periodo refrattario assoluto e del periodo refrattario relativo. La probabilità che avvenga uno spike durante il periodo refrattario assoluto è tendente a zero, ma cresce gradualmente durante il periodo refrattario relativo. Da questa prospettiva è come se i neuroni avessero una memoria, il processo di Poisson però è per definizione privo di memoria, servono quindi altri tipi di processi per andare a mimare questi comportamenti.

Il secondo processo utilizzato è il processo Gamma. Questo processo si verifica quando un modello integra-e-spara è guidato da un processo di Poisson in ingresso (Gerstein & Mandelbrot, 1964). Questo processo è un processo di rinnovo caratterizzato dall' equazione (Eq.4):

$$f(x) = \frac{\lambda^{\alpha}}{\Gamma(\alpha)} x^{\alpha-1} e^{-\lambda x} \qquad Eq.\ 4$$

Dove α è il parametro di forma e λ è definito come (Eq.5)

$$\lambda = \alpha R \qquad Eq.\ 5$$

In cui R è la frequenza di scarica. Il processo di Poisson è un caso speciale del processo gamma in cui il parametro di forma è uguale a 1. Valori di α<1 fanno diventare la distribuzione iper-esponenziale, rendendo più probabili intervalli piccoli, valori invece di α>1 fanno avvicinare la distribuzione a una distribuzione normale (Pipa, 2013).

L'ultimo tipo di distribuzione utilizzata è la Gaussiana Inversa. Allo stesso modo, considerando un processo di Poisson eccitatorio e inibitorio come entrata in un neurone di tipo integra-e-spara, le ISI risultano modellabili anche tramite una Gaussiana Inversa (Tuckwell, 1988). La formulazione di questo processo è riportata in Eq.6:

$$f(x) = \left(\frac{\lambda}{2\pi x^3}\right)^{\frac{1}{2}} exp\left\{-\frac{\lambda(x-\mu)^2}{2\mu^2 x}\right\} \qquad Eq.\ 6$$

Dove μ è pari al reciproco della frequenza di scarica e λ è uguale a (Eq.7):

$$\lambda = \frac{\mu}{\mu^2 - 1} \qquad Eq.\ 7$$

2.4.2 Tipologie di firing

Le principali tipologie di firing che un neurone è in grado di generare (Fig.2.5) sono Tonic Spiking, Phasic Spiking, Tonic Bursing, Phasic Bursting, Mixed mode e Spike frequency adaptation. In realtà le tipologie di firing dei neuroni sono più di una ventina (Izhikevich, 2004), ma ne ho riportate solo alcune per compattezza.

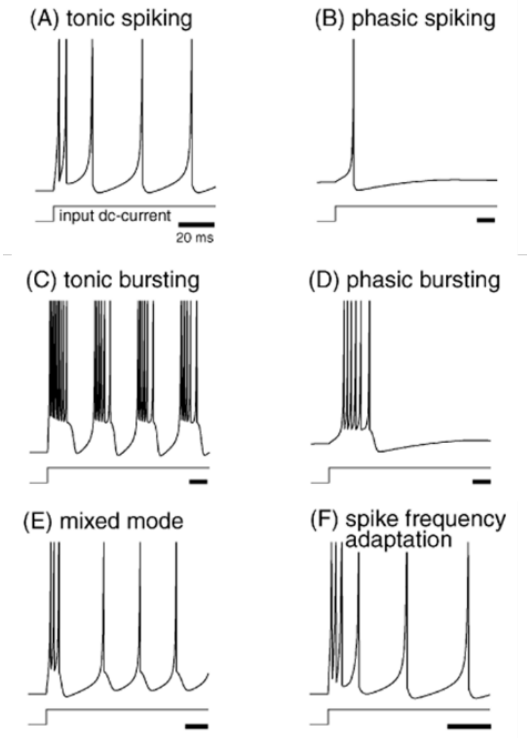


Fig.2.5 Rappresentazione grafica di alcuni tipi di segnale in uscita dai neuroni. Sotto ad ogni figura è rappresentato lo stimolo, che è un impulso di corrente, e la relativa durata. Figura tratta da.(Izhikevich, 2004)

Il *Tonic Spiking* (Fig.2.5A), rappresenta la caratteristica che hanno i neuroni di essere eccitabili: se soggetti ad uno stimolo, i neuroni continuano a "sparare" segnali in uscita. Il *Phasic Spiking* (Fig.2.5B), invece, mostra come un neurone sia in grado di sparare un solo segnale in uscita, nonostante lo stimolo persista nel tempo, e poi rimanga quiesciente. Alcuni neuroni sono in grado di generare una scarica di spikes detta *burst* quando stimolati (Fig.2.5C), la cui frequenza inter-burst può arrivare fino a 50 Hz e che sono coinvolti nelle oscillazioni gamma cerebrali. Questo fenomeno è detto *Tonic Bursting*. Analogamente, è possibile anche una versione di *Phasic Bursting (*Fig. 2.5D*)*, in cui alcuni neuroni utilizzano una sola scarica di burst per comunicare l'inizio di una trasmissione. Altri neuroni, invece, esibiscono un comportamento misto, presentando un phasic burst iniziale che poi mutano in tonic spiking (Fig.2.5E). Infine, tra quelle qui riportate, è possibile la *Spike Frequency Adaptation* (Fig.2.5F), che è presente nel tipo più comune di neuroni eccitatori dei mammiferi ovvero i *regular spiking* (RS). Questi neuroni hanno un comportamento

del tipo tonic spiking che però diminuisce in frequenza se lo stimolo rimane lo stesso, mostrando quindi un adattamento allo stimolo.

2.5 Problemi relativi all'analisi del segnale

Uno dei principali problemi relativi all'analisi del segnale MUA è la mancanza di un ground truth con cui confrontarsi per valutare la bontà di un algoritmo. In linea di principio, un ground truth è ottenibile tramite registrazioni intra-cellulari effettuate contemporaneamente a quelle extra-cellulari (Zucca et al., 2017) di uno o al massimo due neuroni alla volta. Allo stato dell'arte, in assenza di registrazioni locali (effettuate, ad esempio, tramite current patch-clamp) il gold standard è la Visual Inspection (VI), ovvero la valutazione visiva del segnale registrato da parte di esperti del settore. Questi esperti sono in grado di andare a discriminare gli eventi significativi, ovvero gli spikes, dal rumore di fondo. Il problema è che queste valutazioni sono soggettive, piuttosto lunghe ed impegnative e difficilmente applicabili quando il numero di canali di registrazione cresce di pari passo con la tecnologia disponibile. Il numero di elettrodi di registrazione, e quindi di canali, è aumentato enormemente grazie all'integrazione della circuiteria CMOS sugli (paragrafo 1.1.2). A titolo di esempio con la sonda Neuropixel (Jun et al., 2017) si possono registrare 384 canali contemporaneamente, chiaramente non gestibili a mano da un operatore.

La strategia adottata per superare il problema è quindi creare dei modelli matematici che possano riprodurre l'andamento dei segnali neurali, per quanto riguarda frequenza degli spike e forme d'onda. In questo modo si cerca di rendere il segnale sintetico il più fisiologico possibile e si ha un ground truth grazie al quale è possibile valutare le performance di diversi algoritmi. È stata proprio questa la procedura adottata nell'ambito di questa Tesi e verrà spiegata più approfonditamente nel Capitolo 4.

Capitolo 3: Algoritmi di Spike Detection

La Spike Detection (SD) è la procedura in cui si vanno a ricercare eventi significativi, gli spikes, all'interno di un segnale neurale.

3.1 Overview dei più comuni metodi di Spike Detection

Gli algoritmi di Spike Detection possono essere generalmente divisi in tre categorie (Lieb et al., 2017) a seconda delle proprietà degli spike che gli algoritmi sfruttano per identificarli:

1. **Sogliatura semplice**: sono gli algoritmi più facili da implementare e si basano sull'ampiezza dello spike. Si suppone infatti che l'ampiezza dello spike abbia un valore picco-picco maggiore del rumore e che quindi qualunque punto superi una soglia, in grado di separare le due categorie spike-rumore, sia uno spike
2. **Correlazione di Template**: si basano sulla forma dello spike. Le forme d'onda degli spike (template) vengono utilizzate per effettuare misure di similarità con segmenti del segnale. Quando la similarità è maggiore di una determinata soglia, viene identificato lo spike. La difficoltà di questi metodi sta nell'andare proprio a scegliere i template, necessitando o di una conoscenza a priori delle forme d'onda presenti o di approcci generici per la scelta di template adatti. In questa Tesi non è stato considerato alcun algoritmo appartenente a questa categoria.

3. **Energia transitoria**: si basano sul comportamento transitorio dello spike, come rapidi cambi di ampiezza. Questi transitori introducono infatti dei pattern nelle frequenze con delle caratteristiche diverse da quelli dovuti ai rumori.

Esistono anche algoritmi più recenti che sfruttano invece l'entropia o la dimensione frattale e che non ricadono in nessuna di queste categorie (Lieb et al., 2017).

Gli algoritmi che ho valutato in questa Tesi fanno parte delle categorie 1 e 3 e sono, in ordine di semplicità computazionale:

1. Hard Threshold (HT)
2. Hard Threshold Local Maxima (HTLM)
3. Adaptive Threshold Local Maxima (ATLM)
4. Precise Timing Spike Detection (PTSD)
5. Modified Precision Timing Spike Detection (mPTSD)
6. Time Frequency based Convolution algorithm (TIFCO)
7. Stationary Wavelet based Teager Energy Operator (SWTTEO)
8. Smoothed Non-linear Energy Operator (SNEO)

Gli algoritmi HT, HTLM e ATLM sono stati implementati da me e sono i più semplici dal punto di vista computazionale e fanno parte della prima categoria. L'algoritmo PTSD (Maccione et al., 2009) è stato scritto all'interno di IIT diversi anni fa, mPTSD è una sua versione modificata da me e di cui parlerò nel Capitolo 5, e anch'essi fanno parte della prima categoria. Gli algoritmi TIFCO, SWTTEO e SNEO (Yang & Mason, 2014; Malik et al., 2016) sono stati invece trovati in letteratura e ritenuti adatti al confronto in quanto, negli articoli in cui vengono utilizzati, vengono impiegati dataset simili a quello sviluppato in questo lavoro. Questi ultimi tre algoritmi fanno parte invece della terza categoria di SD. Di seguito, una descrizione dettagliata di ognuno degli otto algoritmi utilizzati.

3.2 Hard Threshold (HT)

In questo algoritmo, uno spike viene detettato se il segnale supera una certa soglia, in figura 3.1 si può vedere una semplice rappresentazione del meccanismo di funzionamento. La soglia è pari a un parametro (*MultCoeff*) moltiplicato per la deviazione standard del rumore presente nel segnale. È presente un periodo refrattario (RP, *refractory period*) impostato a 1 ms per tutti gli algoritmi (equivalente a 24 campioni per una frequenza di campionamento pari a 24 kHz). La scelta di questo valore è stata fatta per rispettare la fisiologia del segnale neuronale e la naturale dinamica neuronale(Maccione et al., 2009). Per come è stato scritto l'algoritmo, se il segnale resta sottosoglia anche dopo il RP, il nuovo spike non viene detettato. Infatti, per essere detettato, il segnale deve superare la soglia, risalire in valore al di sopra (in modulo) di questa e poi, trascorso almeno un RP, per far sì che venga riconosciuto un nuovo spike, la soglia deve essere nuovamente superata.

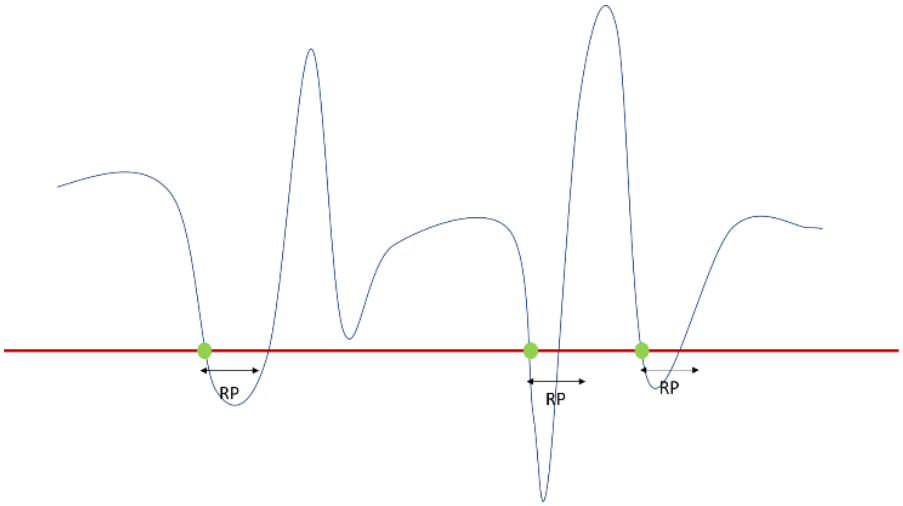


Fig.3.1 Esempio schematico di come avviene la detection in HT. La linea rossa rappresenta la soglia, in verde i campioni che vengono riconosciuti come spike. Come si può vedere, anche se il segnale rimane sottosoglia, quei campioni non sono detettati come spike, ma solo il primo campione che supera la soglia è identificato come spike. In nero è segnato il periodo refrattario (RP).

3.3 Hard Threshold Local Maxima (HTLM)

HTLM ha un funzionamento molto simile a quello dell'algoritmo precedente: lo spike viene detettato solo se il segnale supera una soglia, sempre calcolata come moltiplicazione di un coefficiente (*MultCoeff*) per la deviazione standard del rumore presente (o stimato) nel segnale. La modifica consta nell'utilizzo della funzione built-in di Matlab **findpeaks.m** in grado di trovare i massimi locali. Questa necessita in ingresso il segnale, il valore *'MinPeakHeight'*, cioè il valore minimo che il picco deve superare affinché sia ritenuto spike (e quindi la soglia), e il valore *'MinPeakDistance'*, cioè il RP impostato anche qui pari a 1 ms (24 campioni). In figura 3.2 è mostrato come avviene il processo di detection per HTML.

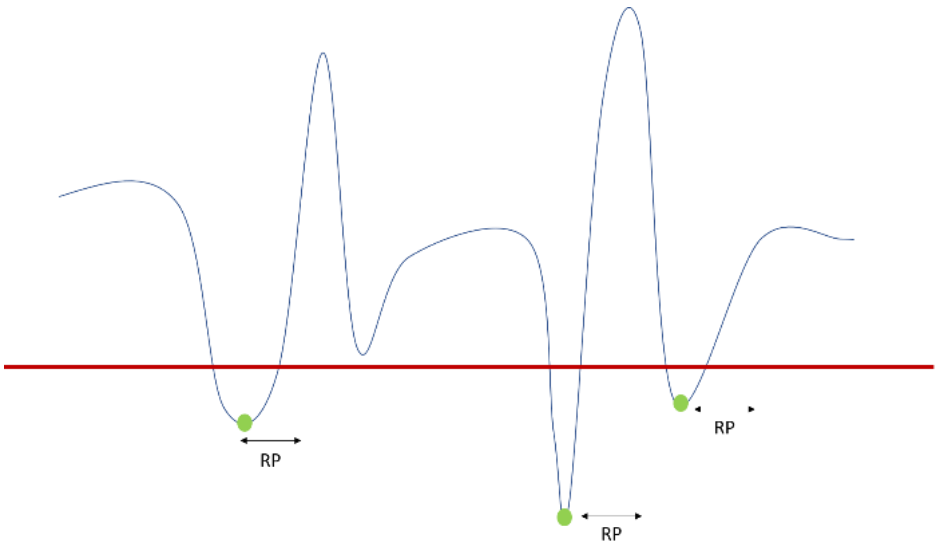


Fig.3.2 Esempio schematico di come avviene la detection in HTLM. In rosso è rappresentata la soglia, in verde i campioni che vengono riconosciuti come spike. Come si può vedere solo il campione relativo al picco che supera la soglia è identificato come spike. In nero è segnato il periodo refrattario (RP).

3.4 Adaptive Threshold Local Maxima (ATLM)

Questo algoritmo prevede sempre l'utilizzo di una soglia, calcolata come moltiplicazione tra un coefficiente (*MultCoeff*) e la deviazione standard del segnale (non solo del rumore come avveniva negli altri due algoritmi), e della funzione **findpeaks.m**. Quello che

cambia rispetto agli altri due algoritmi, inoltre, è che la deviazione standard non viene calcolata su tutto il segnale ma su delle finestre dello stesso. I parametri che utilizza sono quindi due: *multCoeff* e *TimeWindow*, che è l'ampiezza della finestra di tempo del segnale su cui viene calcolata la deviazione standard. La soglia varierà quindi per ogni finestra di segnale, se in quella finestra il segnale è più ampio allora la soglia sarà più alta e viceversa, da qui il termine adaptive, cioè adattivo. Questo permette di andare a discriminare meglio spike e rumore in quanto non è detto che la distribuzione degli spike all'interno del segnale sia omogenea, possono esserci finestre prive di spike che contengono solo rumore e che grazie a questo algoritmo vengono scartate. In figura 3.3 è rappresentato lo schema di funzionamento di ATLM.

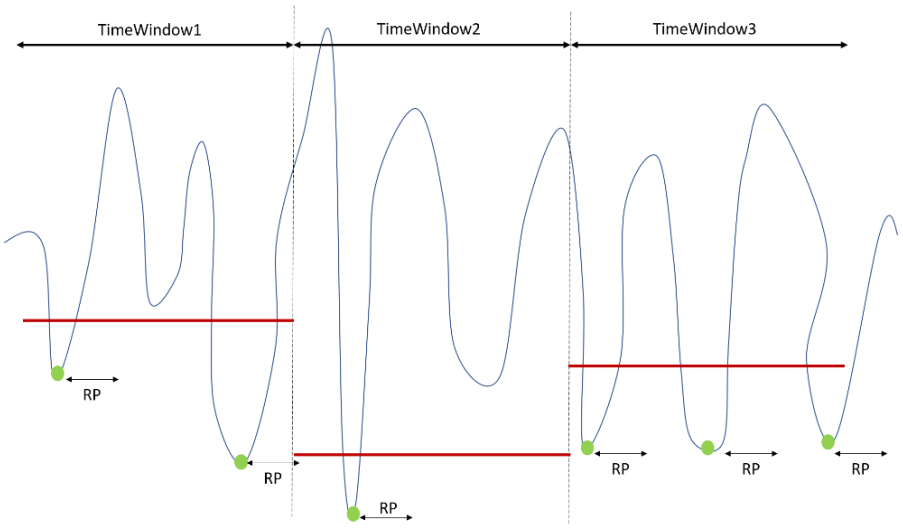


Fig.3.3 Esempio schematico di come avviene la detection in ATLM. In rosso sono raffigurate le soglie per ogni TimeWindow, finestra temporale che divide il segnale. In verde i campioni che vengono riconosciuti come spike, cioè quelli corrispondenti al picco che, per ogni finestra, supera la soglia calcolata come deviazione standard del segnale nella finestra per un coefficiente moltiplicativo. In nero è segnato il periodo refrattario (RP).

3.5 Precise Timing Spike Detection (PTSD)

Il funzionamento di PTSD si discosta dai precedenti algoritmi sotto diversi punti di vista: primo tra tutti la presenza di una soglia differenziale invece che unipolare, inoltre, oltre a RP, è presente anche un altro intervallo di tempo chiamato *Peak Lifetime Period* (PLP), ovvero la durata massima che può avere uno spike. Procedendo con ordine, l'algoritmo ricerca un massimo/minimo relativo (1° RMM, *Relative Maximum/Minimum*), se trova un massimo allora andrà a ricercare nel segnale il più vicino minimo (2° RMM) entro la finestra PLP, se trova invece un minimo procederà alla stessa maniera ma cercando un massimo. Se la differenza tra 1° RMM e 2° RMM, ovvero l'ampiezza picco-picco dello spike, supera la soglia differenziale (DT, *Differential Threshold*), allora viene identificato lo spike (Fig. 3.4a). Nel caso il 2° RMM cadesse alla fine della finestra PLP, viene utilizzata un'altra finestra temporale chiamata *overshoot* per andare a trovare il corretto valore del picco (Fig. 3.4b). La DT è sempre calcolata come deviazione standard del rumore moltiplicata per un coefficiente moltiplicativo (*MultCoeff*).

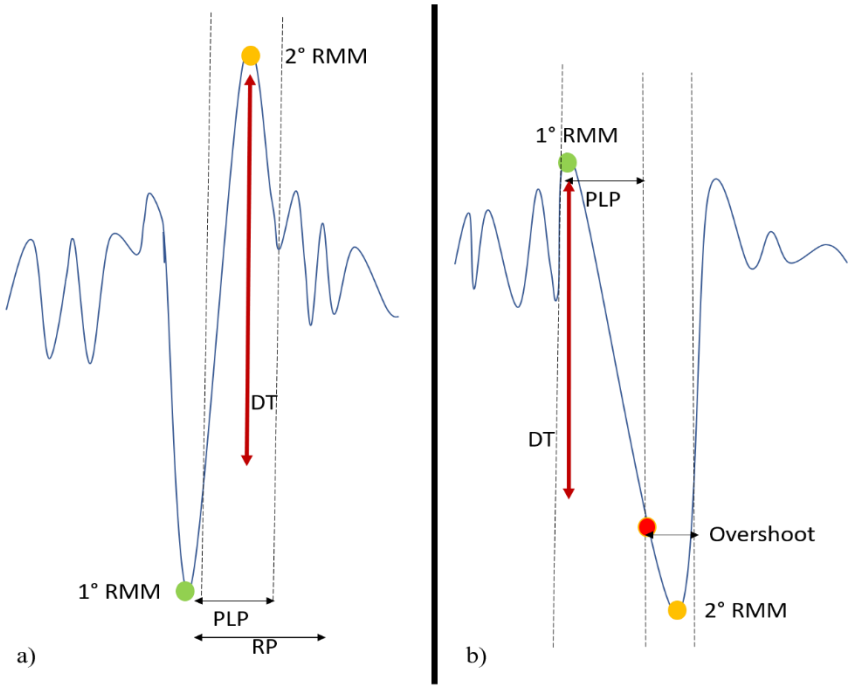


Fig3.4 Schema del funzionamento di PTSD. a) Il 1° RMM viene trovato (verde) ed è un minimo, l'algoritmo ricerca allora il massimo più vicino all'interno del PLP,

trovando così il 2° RMM (giallo). La loro differenza supera la DT (rosso), quindi lo spike viene associato al 1° RMM (verde). È mostrato anche RP, maggiore di PLP. b) Viene trovato il 1° RMM come massimo (verde), il minimo più vicino (rosso) coincide con la fine del PLP. In questo caso parte la finestra dell'overshoot che permette di andare a trovare il minimo corretto (2° RMM, giallo), a cui viene assegnato lo spike.

3.6 Modified Precision Timing Spike Detection (mPTSD)

Si tratta della versione modificata di PTSD, uno degli obiettivi di questa Tesi. Di questo algoritmo si parlerà in un capitolo a parte.

3.7 Time Frequency based Convolution algorithm (TIFCO)

Primo degli algoritmi che si basano sull'analisi dell'energia del segnale. Sviluppato da Lieb (Lieb et al., 2017) questo algoritmo ha dei presupposti più complessi e meno immediati degli algoritmi precedenti, e si basa sulla rappresentazione tempo-frequenza degli spike, infatti viene utilizzata una versione discreta della *short-time Fourier Transform* (STFT) nota come trasformata Gabor.

La trasformata Gabor è definita come segue (Eq. 8)

$$c(m,l) = \sum_{n=0}^{L-1} \frac{1}{f(n)g(n-al)e^{-2\pi inm/M}} \qquad \textit{Eq. 8}$$

Dove f(n) è il segnale, g(n) è una finestra discreta traslata di a campioni, l'esponenziale è l'operatore di modulazione in cui M è il numero dei bin della frequenza e L è la lunghezza del segnale. Quindi la trasformata Gabor è il prodotto del segnale con una versione traslata e modulata di una finestra g(n). La scelta della finestra g(n) è dovuta al principio di indeterminazione che governa la risoluzione simultanea del tempo e della frequenza. Ad esempio, una finestra rettangolare, che possiede un'ottima localizzazione nel tempo, nel dominio della frequenza avrà una localizzazione molto scarsa. Una finestra gaussiana è il compromesso migliore per entrambi i domini ma non è robusta, cosa che è invece la finestra di Hanning, che si dimostra essere la scelta

migliore per l'implementazione della trasformata Gabor. La scelta del parametro *m* nell'esponenziale permette di andare a specificare solo un certo intervallo di frequenze. Scegliendo infatti *m* nell'intervallo intero [0, M-1], la frequenza andrà da 0 a 2 volte la frequenza di Nyqvist. Ad esempio, se *m* appartiene all'intervallo [(M-1)/8, (M-1)/4] allora verranno considerate solo le frequenze che partono da un quarto della frequenza di Nyqvist a metà della stessa. Questa caratteristica trova applicazione nella SD in quanto riduce i contributi in frequenza indesiderati, infatti gli spike esibiscono un certo comportamento tempo-frequenza, diverso dal rumore flicker (Lieb et al., 2017), nel range 500-3500 Hz.

Nel primo step di questo algoritmo vengono ignorati tutti i contributi al di sotto dei 500 Hz e al di sopra dei 3500 Hz andando a scegliere una giusta parametrizzazione della trasformata Gabor (e quindi un *m* adatto). Successivamente, viene applicato un filtro a media mobile per rafforzare la struttura dello spike, cioè a seconda della scelta del kernel del filtro viene imposta la persistenza dell'intervallo di tempo e di frequenze desiderato.

La descrizione matematica del kernel è la seguente: sia K(m', l') il kernel di una convoluzione bi-dimensionale, ad esempio di un filtro a media mobile di dimensione M'x N'. Allora, i coefficienti tempo-frequenza modificati dal kernel sono così definiti (Eq.9):

$$v(m,l) = \sum_{m'} \sum_{l'} |c(m-m', l-l')|^2 \, K(m', l') \qquad \textit{Eq. 9}$$

Nello step seguente, viene calcolata una variabile decisionale y(l) andando a sommare cumulativamente le frequenze (Eq.10):

$$y(l) = \sum_{m}^{M} v(m,l) \qquad \textit{Eq. 10}$$

Ogni volta che uno spike è presente all'istante di tempo l, y(l) tende a un valore maggiore rispetto a quando è presente solo rumore. La variabile decisionale subisce quindi una sogliatura per estrarre l'esatta localizzazione degli spike. Nel diagramma a blocchi in fig. 3.5 è presente una spiegazione schematica del funzionamento dell'algoritmo.

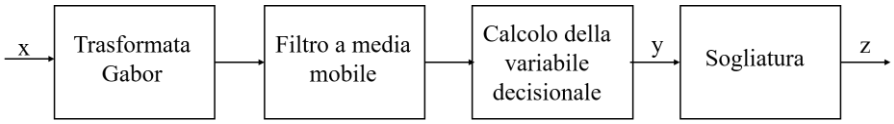


Fig.3.5 Diagramma a blocchi del funzionamento di TIFCO. Il segnale x viene trasformato utilizzando la trasformata Gabor per andare ad escludere delle componenti tempo-frequenza appartenenti al rumore. L'uscita subisce quindi un filtraggio a media mobile, il cui kernel modifica i coefficienti tempo-frequenza. Questi coefficienti sommati cumulativamente, andranno a definire la variabile decisionale y che ha un valore maggiore quando sono presenti gli spike rispetto a quando è presente solo rumore. Effettuando quindi la sogliatura della variabile y, si possono ottenere gli istanti di tempo in cui è presente uno spike.

3.8 Stationary Wavelet based Teager Energy Operator (SWTTEO)

Si tratta del secondo algoritmo sviluppato da Lieb (Lieb et al., 2017). Anche questo si basa sull'analisi dell'energia del segnale, in particolare utilizza la trasformata wavelet stazionaria (SWT, *Stationary Wavelet Transform*) e il *Teager Energy Operator* (TEO).

Prima di parlare della SWT bisogna introdurre la trasformata wavelet discreta (*DWT, Discrete Wavelet Transform*), che si basa sulla discretizzazione logaritmica, e non lineare come nel caso della trasformata Gabor, dell'asse delle frequenze. La DWT consiste sostanzialmente in un primo filtraggio passa-alto e in un susseguente sotto-campionamento di fattore 2, e allo stesso tempo in un filtraggio passa-basso con uguale sotto-campionamento. I coefficienti risultanti sono chiamati coefficiente di dettaglio (c_D) e coefficiente di approssimazione (c_A), in Fig.3.6 viene schematizzato il funzionamento.

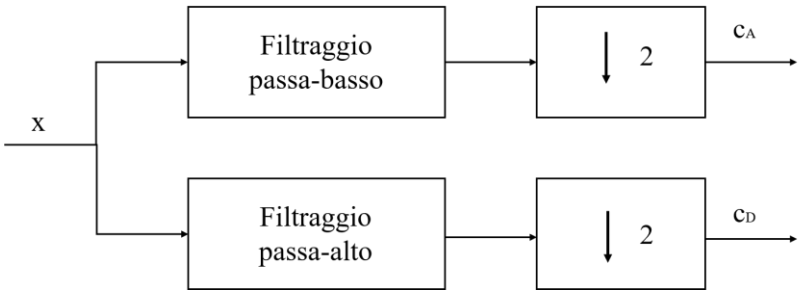


Fig.3.6 Diagramma a blocchi del funzionamento della DWT. Il segnale x in ingresso subisce un filtraggio passa-basso, un seguente sotto-campionamento di fattore due e

restituisce in uscita i coefficienti di approssimazione. Allo stesso tempo il segnale entra in un filtro passa-alto, subisce lo stesso sotto-campionamento e restituisce i coefficienti di dettaglio.

La formulazione di $c_{A/D}$ è la seguente (Eq.11)

$$c_{D-A}(l) = \sum_{n=0}^{L-1} f(n)\varphi_{D-A}(2n-l) \qquad \textit{Eq. 11}$$

Dove $\varphi_{D\text{-}A}$ indica se si sta utilizzando il filtro per ottenere il coefficiente c_D o quello per il coefficiente c_A. Questa procedura può essere iterata k volte andando a sostituire l'input con il coefficiente di approssimazione dell'iterazione precedente, si parla in questo caso di decomposizione wavelet di livello k.

Il problema della DWT però è la mancanza di tempo-invarianza a causa del sotto-campionamento, infatti una versione traslata dello stesso segnale in ingresso non porterebbe ad una versione traslata dei coefficienti della wavelet. La SWT sostituisce il sotto-campionamento con un sovra-campionamento di fattore 2 tramite zero-padding, andando ad eliminare il problema.

Dopo aver utilizzato la SWT, l'algoritmo sfrutta il TEO, operatore sviluppato da Teager e Keiser (Kaiser, 1990; Kaiser, 1993). La formulazione del TEO nel caso continuo è la seguente (Eq.12):

$$\psi\big(x(t)\big) = \dot{x}^2(t) - x(t)\ddot{x}(t) \qquad \textit{Eq. 12}$$

Il termine operatore energetico è motivato dall'analogia di questa formulazione con l'oscillatore di massa armonico, dove $x(t)=A\cos(\omega t + \varphi)$ e l'energia è $E=\frac{1}{2} m \omega^2 A^2$. È verificata infatti l'uguaglianza $\psi(x(t)) = \omega^2 A^2$, che significa che $\psi(x(t))$ è proporzionale all'energia. Nel caso discreto la formulazione del TEO diventa (Eq.13):

$$\psi\big(x(n)\big) = x^2(n) - x(n-1)x(n+1) \qquad \textit{Eq. 13}$$

Successivamente all'applicazione del TEO, l'uscita viene fatta convolvere con una finestra di Hamming della stessa lunghezza dello spike per effettuare lo smoothing. Tenendo conto che ci sono diversi livelli di trasformata, questo procedimento viene effettuato per ogni

livello, dopodiché le uscite della convoluzione vengono sommate e solo dopo il risultato viene sogliato. Il numero di livelli della decomposizione wavelet, il k precedente, dipende dal contenuto in frequenza dello spike. Per il range di frequenze 5-35 KHz, che più o meno è lo stesso utilizzato come riferimento per questa Tesi 3-30 KHz, è stato dimostrato che una decomposizione di livello 2 risulta essere la più adatta (Lieb et al., 2017). In figura 3.7 si può apprezzare il funzionamento schematizzato dell'algoritmo.

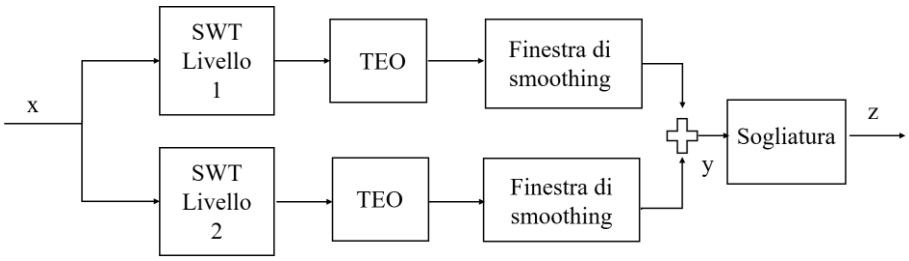


Fig.3.7 Schema di funzionamento dell'algoritmo SWTTEO. Il segnale x in ingresso subisce la decomposizione tramite trasformata wavelet in 2 livelli. La SWT effettua due filtraggi: passa-alto che restituisce il coefficiente c_D e passa-basso che restituisce il coefficiente c_A. Nel caso si volessero effettuare altre iterazioni basterebbe sostituire il segnale x in ingresso con il c_A calcolato e così via. L'uscita della SWT entra nel blocco del TEO la cui uscita viene fatta convolvere con una finestra di smoothing. Le uscite dei due livelli vengono quindi sommate (y) e il risultato sottoposto a sogliatura per discriminare gli istanti degli spike.

3.9 Smoothed Non-Linear Energy Operator (SNEO)

SNEO è la versione filtrata con una finestra di Bartlett del *Non-Linear Energy Operator* (NEO), la sigla SNEO sta infatti per *Smoothed Non-Linear Energy Operator*. L'operatore NEO è definito anche come TEO (Eq.7), detto anche TKEO (*Teager-Keiser Energy Operator*), questo algoritmo utilizza quindi lo stesso operatore energetico di SWTTEO. La differenza con l'algoritmo del paragrafo precedente è la mancanza dell'utilizzo SWT.

La formulazione dell'operatore SNEO è la seguente (Eq.14):

$$\psi\big(x(n)\big) = \psi(x(n)) \otimes w(n) \qquad \textit{Eq. 14}$$

Dove ψ(x(n)) è l'operatore energetico che subisce una convoluzione con la finestra w(n). Dopo la convoluzione è prevista sempre la sogliatura per andare a detettare gli istanti degli spike. In figura 3.8 è rappresentato il funzionamento schematico dell'algoritmo, e in figura 3.9 uno schema grafico del funzionamento degli algoritmi che utilizzano un operatore energetico con sogliatura.

Fig.3.8 Diagramma a blocchi del funzionamento di SNEO. Il segnale x entra nel blocco dell'operatore energetico non lineare, l'uscita viene convoluta con una finestra di smoothing e il risultato y viene sogliato per ottenere gli istanti degli spike.

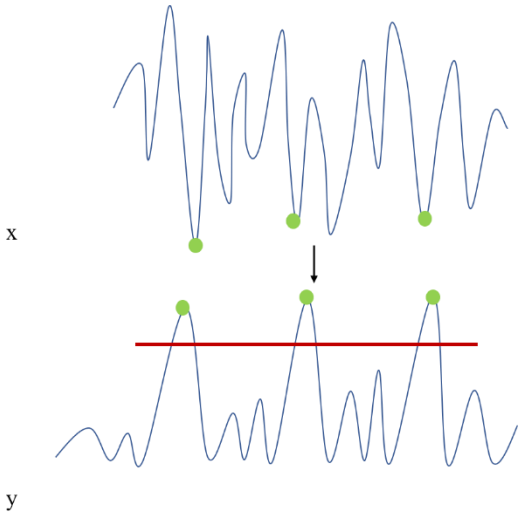


Fig.3.9 Rappresentazione grafica di come agisce SNEO. Sopra, il segnale in ingesso x, dove in verde sono segnati gli istanti degli spike, subisce l'operatore SNEO la cui uscita è riportata in basso (y). Viene fatta una sogliatura semplice del segnale y, dove la soglia è la linea rossa. In verde sono segnati gli istanti ritenuti spike dall'algoritmo.

Capitolo 4: Sviluppo di un modello computazionale di groundtruth

In questo capitolo verrà mostrato il procedimento di creazione del modello di segnale extra-cellulare in vivo utilizzato in questa Tesi.

4.1 Generazione di Processi Puntuali

I dati che si vogliono andare a simulare sono registrazioni extra-cellulari in-vivo. La procedura per ottenere un modello matematico di spike train da impiegare come ground truth ha seguito i seguenti step:

1. Ottenimento delle classi di template dalle forme d'onda estratte da registrazioni in-vivo su ratti (Fig. 4.1a). Tramite Visual Inspection sono stati trovati gli istanti corrispondenti agli spikes, le relative forme d'onda hanno subito il processo di sorting per poter discriminare le singole classi di template che corrispondono alle unità che insieme formavano il MU registrato.
2. Creazioni di tre tipi di distribuzioni di ISI: esponenziale, gamma e gaussiana inversa (Fig.4.1b). Da queste distribuzioni è stato creato lo spike train come sequenza di δ di Dirac. Andando a sovrapporre alle posizioni degli spikes, trovate come somma cumulativa delle ISI, le classi di template ottenute nello step

precedente, si ottengono segnali provenienti da singole cellule neuronali privi di rumore, Single Unit (SU).

3. Sommando tre unità si ottiene il segnale proveniente da Multi Unit, a questo verrà poi sommato il rumore creato con Simulink nei vari livelli di SNR desiderati.

Il modello creato è quindi un modello ibrido in quanto coniuga sia le ISI create sinteticamente sia le forme d'onda fisiologiche registrate da animali. Esistono infatti anche altri tipi di modelli che utilizzano forme d'onda generate, anch'esse sinteticamente ma questo tipo di modello risulta essere alquanto complesso e costoso dal punto di vista computazionale (Hagen et al., 2015).

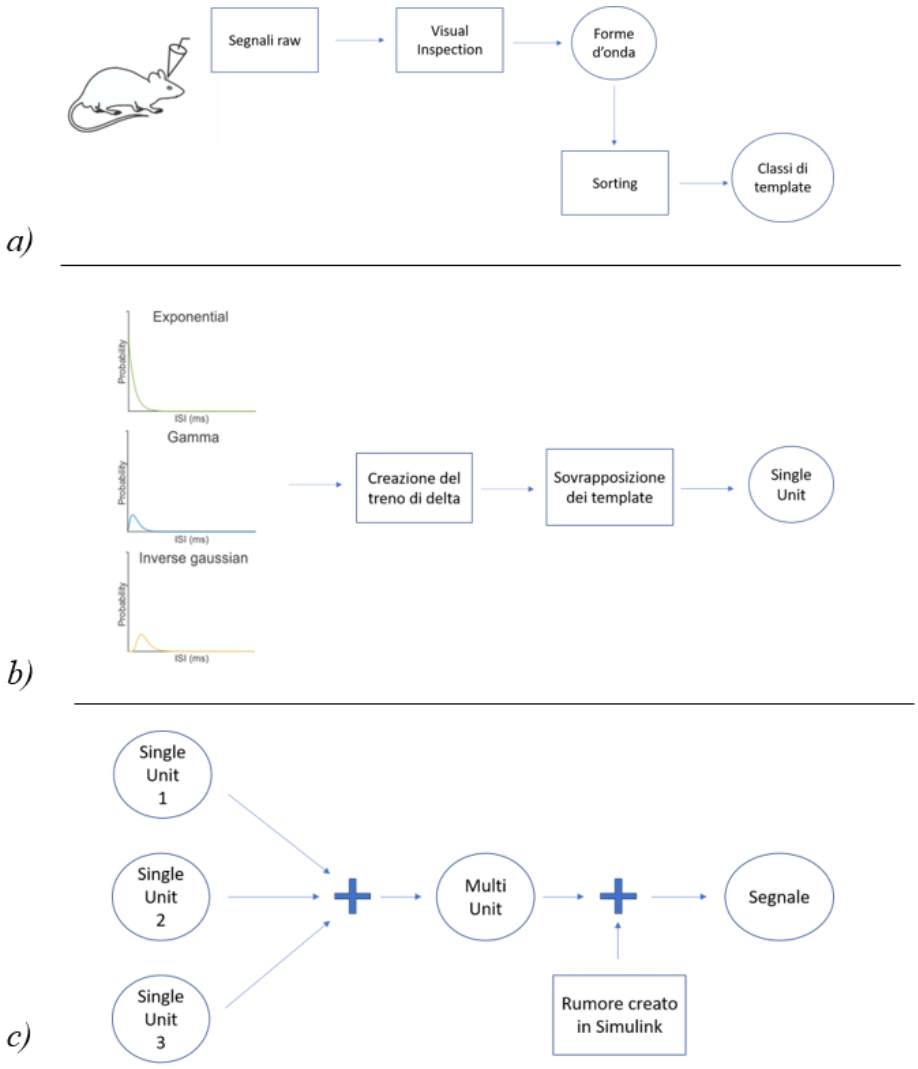


Fig.4.1 a) Estrazione dei template delle forme d'onda: i segnali raw vengono acquisiti dall'animale, l'esperto effettua la VI andando a evidenziare gli spike presenti nel segnale. Le forme d'onda attorno agli eventi segnalati vengono estratte, a questo

punto viene effettuato il sorting che le divide nelle rispettive classi. b) Generazione della SU: i 3 tipi di distribuzioni creati (esponenziale, gamma e gaussiana inversa) rappresentano le ISI, che sommate cumulativamente, rappresentano la posizione degli spike. Ad ogni posizione viene fatta corrispondere una δ di Dirac formando così un treno di δ che poi andrà ad indicare dove sovrapporre la forma d'onda del template per generare i SU. c) Generazione del segnale sintetico: le tre SU estratte vengono sommate e formano il MU privo di rumore a cui viene sommato a sua volta il rumore creato in Simulink per ogni livello di SNR desiderato, in output si ha il segnale completo al variare del rapporto segnale rumore.

4.1.1 Implementazione in Matlab

In Matlab è presente la funzione built-in **makedist.m**, in cui è sufficiente specificare il tipo di distribuzione ('Exponential','Gamma','InverseGaussian') e le coppie parametro-valore desiderate ('parameter','parameterValue'). E'stata effettuata anche una procedura di troncatura tramite la funzione **truncate.m** per andare a escludere valori limite dalla distribuzione. In questo caso, la distribuzione rappresenta le ISI dei neuroni e pertanto si è deciso di escludere valori irrealistici per via dei periodi refrattari, al di sotto di 1 ms.

Per la distribuzione esponenziale l'unico parametro variabile è λ ed è stato fatto variare 100 volte, all'interno di un intervallo di FR fisiologico tra 1 e 22 spikes/s (Averna et al., 2019), per avere un ampio range di ISI. Il range di valori in cui è stato fatto variare λ è riportato in Tab.4.1. La definizione della distribuzione esponenziale in Matlab però prevede l'utilizzo del valor medio μ all'esponente, definito come il reciproco di λ, bisogna quindi tenere conto di questa variazione quando si va a definire il parametro. Per quanto riguarda le altre due distribuzioni invece, per avere sempre 100 variazioni, considerando che possiedono due parametri da far variare, sono stati fatti variare entrambi 10 volte per ottenere così 100 variazioni in totale. I valori si trovano in Tab.4.1. Nel grafico 4.2 sono invece mostrati gli istogrammi e le funzioni di densità di probabilità delle distribuzioni.

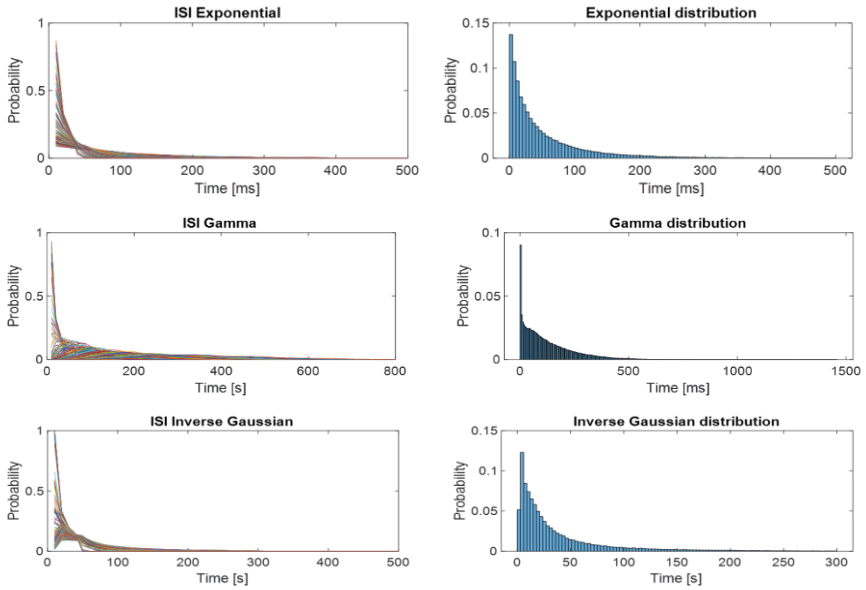


Fig.4.2 Sulla sinistra le funzioni densità di probabilità e sulla destra gli istogrammi delle tre distribuzioni, in alto esponenziale, al centro Gamma, in basso Gaussiana Inversa.

.

Distribuzione	**Funzione Densità di Probabilità**	**Range di variazione dei parametri**
Esponenziale	$f(\mathrm{x}) = \lambda e^{-\lambda \mathrm{x}}$	$\lambda = 1$ - 22
Gamma	$f(x) = \frac{\lambda^{\alpha}}{\Gamma(\alpha)} \mathrm{x}^{\alpha-1} \mathrm{e}^{-\lambda \mathrm{x}}$	$\alpha = 0.1$-9 $\lambda = 0.1$- 0.5
Gaussiana Inversa	$f(x) = \left(\frac{\lambda}{2\pi \mathrm{x}^3}\right)^{1/2} \exp\left\{-\frac{\lambda(x-\mu)^2}{2\mu^2 x}\right\}$	$\mu = 0.045$ - 1 $\lambda = \frac{\mu}{\mu^2 - 1}$

Tab.4.1 Range di variazione dei parametri per ogni tipo di distribuzione utilizzata e relativa funzione di densità di probabilità. La distribuzione esponenziale possiede solo un parametro variabile, λ, mentre le altre due distribuzioni possiedono due parametri variabili: la distribuzione gamma α e λ, la gaussiana inversa μ e λ, dove quest'ultima è funzione di μ.

4.2 Spike Sorting

Partendo quindi dai segnali registrati da ratti, sui quali è stata effettuata la VI, è stato effettuato con un algoritmo comunemente utilizzato, ovvero *Waveclus (*Quiroga et al., 2004*)*, che effettua un clustering di tipo super-paramagnetico. Le forme d'onda estratte hanno una durata di 32 campioni, che alla frequenza di 24414 Hz corrispondono a circa 1.3 ms. In figura 4.3 si possono vedere le registrazioni in vivo utilizzate in cui sono stati evidenziati gli eventi ritenuti spike da arte degli operatori, mentre in Tab.4.2 sono riportate le forme d'onda estratte dalle registrazioni. In Tab.4.3 sono presenti le caratteristiche delle registrazioni, in Tab.4.4 invece sono riportate le caratteristiche delle classi ottenute.

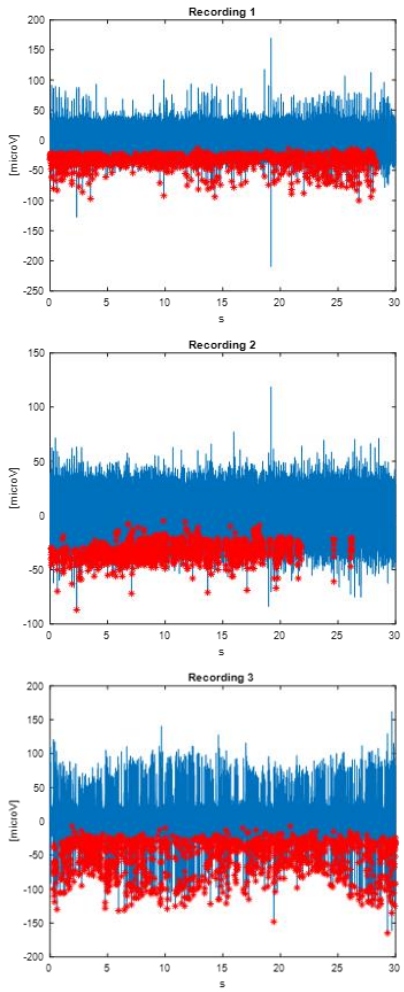
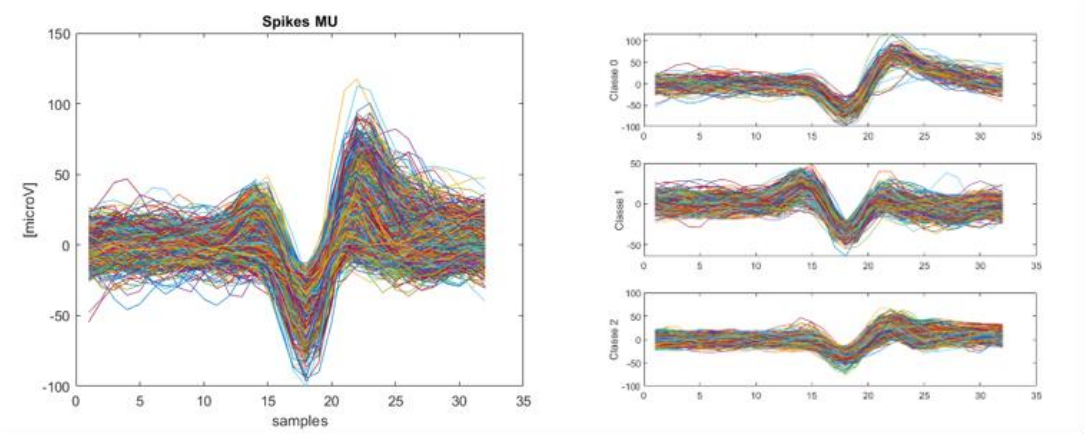
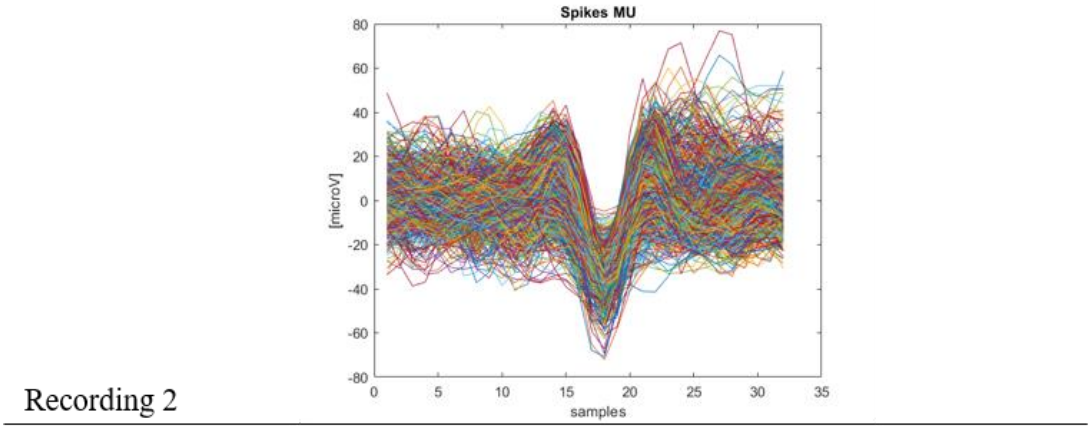


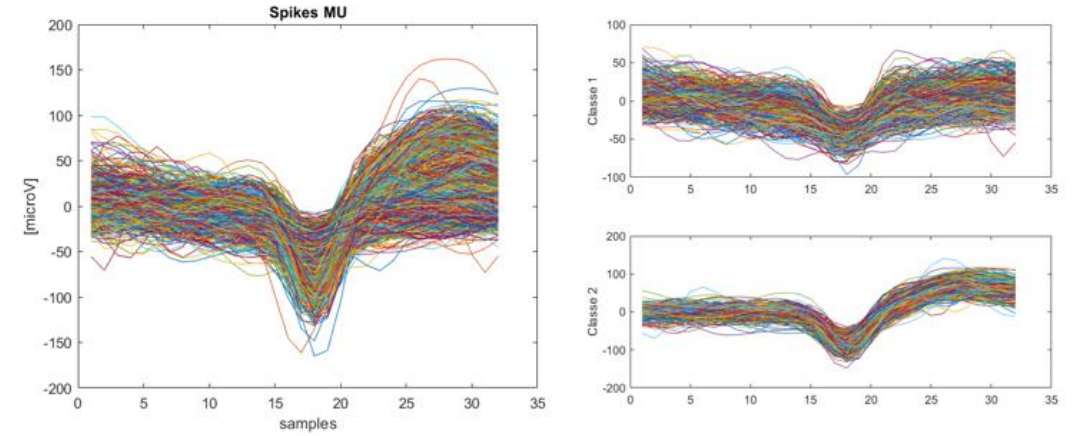
Fig.4.3 Le tre registrazioni da cui sono state estratte le forme d'onda che sono state utilizzate nel modello su cui è stata effettuata la VI. Sull'asse x è presente il Tempo in secondi, sull'asse y il Voltaggio in µV.



Recording 1



Recording 2



Recording 3

Fig.4.4 Forme d'onda trovate tramite visual inspection. Ogni forma d'onda estratta ha la durata di 32 campioni (1.3 ms). Per il recording 1 e 3 sulla colonna di sinistra sono presenti tutte le forme d'onda estratte, sulla destra invece le classi trovate grazie allo spike sorting. Nel recording 2 lo spike sorting ha evidenziato una sola famiglia di curve

Recording	Durata [s]	Durata visual inspection [s]	Numero di spikes	MFR [spikes/s]	Classi ottenute
1	30	28.3	1179	42	3
2	30	21.8	1330	61	1
3	30	29.9	1393	47	2

Tab.4.2 Registrazioni con relativa durata complessiva, durata delle VI, numero di spikes trovati dall'operatore, mean firing rate (numero di spikes/durata VI), e classi di forme d'onda ottenute.

	Recording 1			Rec 2	Recording 3	
	Classe 0	Classe 1	Classe 2	Classe 1	Classe 1	Classe 2
Numero di spikes	144	504	531	1228	925	468
MFR (spikes/s)	5.5	17.7	18.2	58	30.8	15.6

Tab.4.3 Numero di spikes e MFR (numero di spikes/durata della VI) per ogni classe di ogni registrazione (recording). Recording 1 possiede 3 classi, recording 2 solo 1, e recording 3 ne possiede 2.

4.3 Da Single Unit a Multi Unit

Estratte quindi le forma d'onda tramite il sorting, ogni classe ottenuta è stata sovrapposta alle matrici di processi puntali costruite precedentemente. Avendo quindi a disposizione 3 tipi di distribuzioni, quindi 3 matrici di 100 righe (una per ogni variazione del parametro) e di lunghezza pari a 60 s (che in campioni corrispondono a 60 x fs, dove fs è la frequenza di campionamento pari a 24 kHz), e 6 classi di forme d'onda, sono state create 18 matrici (Fig.4.4) in totale dove, per ogni matrice, ogni riga corrisponde a un SU (Fig.4.5) con frequenza di firing diversa. La forma d'onda viene scelta randomicamente all'interno di

ciascuna classe associata alla matrice, mantenendo quindi una certa variabilità anche all'interno della single unit.

Un segnale Multi Unit (MU) è pari alla somma dei segnali delle Single Unit (SU), solitamente dalle 2 alle 4 unità. Il segnale MU che ho creato prevede l'utilizzo di 3 SU con template diversi. Il procedimento con cui è stato creato è il seguente: da ciascuna delle 18 matrici descritte sopra viene scelta randomicamente una riga, e quindi una SU. Le 18 righe scelte vengono sovrapposte in una matrice da cui vengono estratte, sempre randomicamente, 3 righe, che rappresentano le SU che sommate formeranno il MU. In figura 4.5 si può avere un riscontro visivo del procedimento.

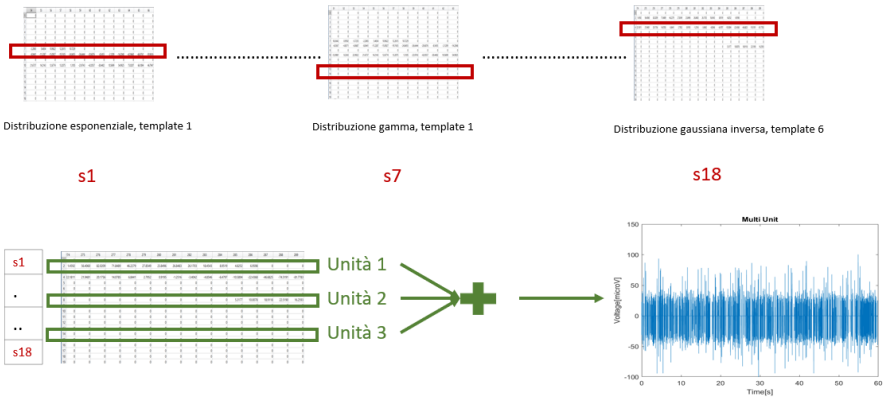


Fig.4.5 Spiegazione visiva della creazione del MU. 18 segnali (s1… s18 in rosso) vengono estratti randomicamente dalle rispettive matrici di distribuzioni (esponenziale, gamma e gaussiana inversa) con le forme d'onda sovrapposte. Essendo le classi di forme d'onda 6, il totale è 18 matrici. Sovrapponendo i 18 segnali si ottiene un'ulteriore matrice di 18 righe, da cui vengono estratte nuovamente, sempre in maniera random, 3 righe (in verde). Queste vengono sommate e in questo modo si ottiene il MU privo di rumore.

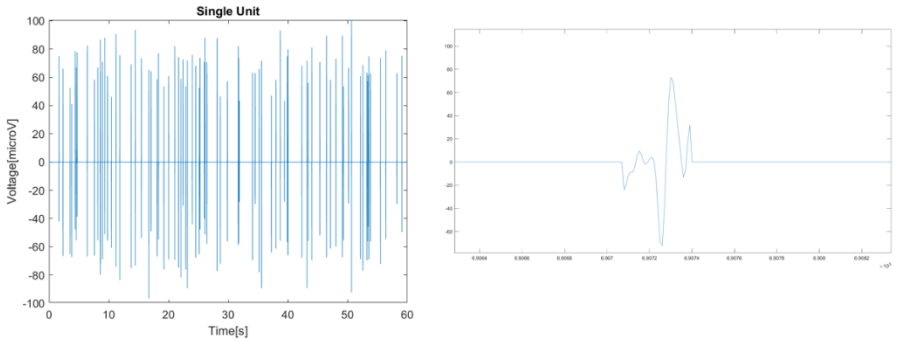


Fig.4.6 Esempio di segnale SU privo di rumore. A sinistra è presente l'intero segnale della durata di 60 s, a destra è stato isolato un singolo spike per mostrarne la forma d'onda.

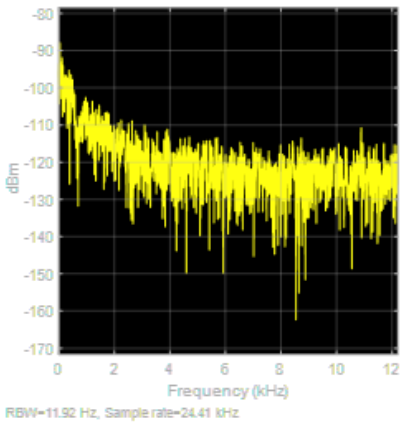
4.4 Generazione del rumore

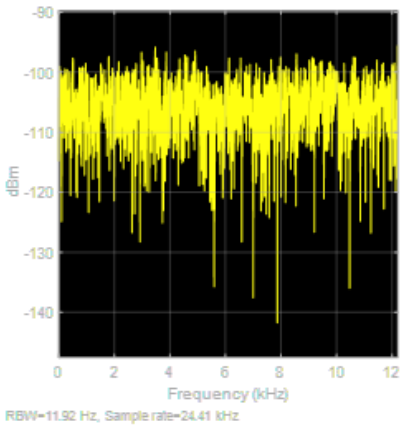
Il rumore tipicamente presente nelle registrazioni è il rumore il rumore rosa o flicker, bianco o gaussiano e la 50 Hz (Maccione et al., 2009). Il rumore flicker presenta componenti a bassa frequenza con una potenza maggiore rispetto a quelle con frequenze elevate. L'andamento del suo spettro infatti (Fig.4.7, alto), è del tipo 1/f Il rumore bianco ha ampiezza costante su tutto lo spettro delle frequenze (Fig.4.7, centro). L'onda sinusoidale a 50 Hz viene introdotta a causa dell'interfacciamento elettronico dei cavi che, attaccandosi alla corrente, introducono questo disturbo.

Il rumore è stato generato utilizzando Simulink (Fig.4.8). Le simulazioni hanno una durata pari a quella del segnale che si vuole simulare, ovvero 60 secondi. I blocchi che uniti generano l'output creano i tre tipi di rumori desiderati, ovvero: bianco, flicker e un'onda sinusoidale a 50 Hz. Il rumore bianco è stato generato semplicemente utilizzando il blocco corrispettivo in Simulink, andando a settare valor medio nullo e varianza 0.12 μV^2 (Maccione et al., 2009) il suo output viene scalato per un parametro pari a $1.5x10^4$, che crea il rumore che genera SNR più basso, che viene poi riscalato per andare a creare gli altri SNR. Il rumore flicker è stato costruito seguendo le indicazioni di Maccione (Maccione et al., 2009), andando ad implementare la funzione di trasferimento discreta (Eq.15):

$$H(z) = \frac{a}{(1-b \cdot z^{-1})^{1/2}} \qquad \textit{Eq. 15}$$

Dove a e b modellano rispettivamente l'ampiezza e la pendenza dello spettro del rumore. Come valori di questi parametri ho utilizzato quelli presenti nel lavoro sopra citato ovvero a=1.3x10-6 e b=1.4. A monte del blocco che implementa la funzione di trasferimento discreta è presente un generatore di numeri randomici. Il rumore bianco I tre tipi di rumori vengono quindi sommati per restituire il rumore totale in uscita.





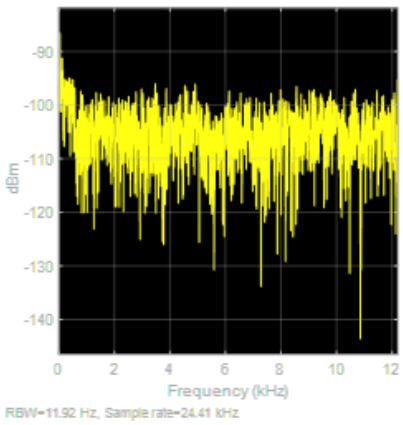


Fig.4.7 Spettri di potenza dei rumori. In alto, lo spettro del rumore flicker, si può apprezzare l'andamento dello spettro 1/f. Al centro lo spettro del rumore bianco, costante, in basso lo spettro del rumore totale che è una combinazione dei due.

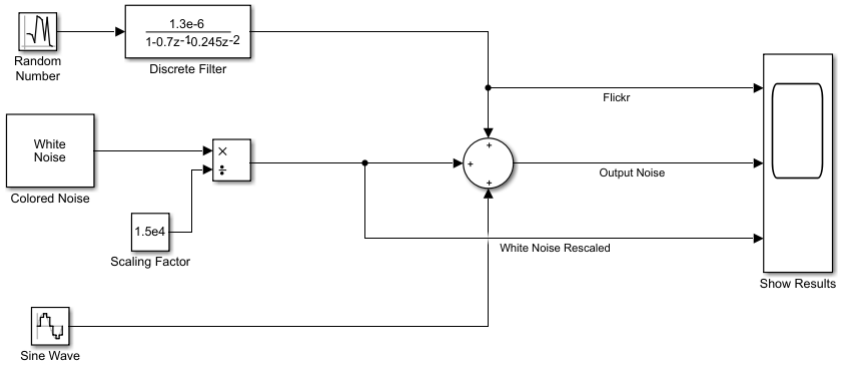


Fig.4.8 Schema della creazione del rumore in Simulink. In alto a sinistra è presente il blocco generatore di numeri random che entra nel blocco del filtro discreto per creare il rumore flicker. Al di sotto è presente il blocco del rumore bianco diviso per il suo fattore di scala $1.5x10^4$. In basso è presente il blocchetto della forma d'onda sinusoidale a 50 Hz. Tutti gli output dei blocchi entrano in un sommatore che restituisce poi in uscita il rumore totale.

4.5 Segnali sintetici generati

4.5.1 Rumore

Utilizzando come definizione di SNR quella di Matlab, calcolandolo con il comando **snr***(Segnale,Rumore)*, che calcola la potenza dello spettro del termine Segnale e la divide per la potenza del Rumore, sono stati creati 10 livelli di SNR andando a moltiplicare il primo rumore generato, come precedentemente spiegato, per dei parametri, e successivamente andando a effettuare un filtraggio tramite filtro passa banda, con un filtro di Butterworth, tra 300 e 3000 Hz. Le coppie parametro-SNR sono riportate in tabella 4.4.

Livello di Rumore	Parametro Moltiplicativo	SNR ottenuto
1	0.1	0.86
2	0.14	0.57
3	0.18	0.43
4	0.23	0.35
5	0.27	0.29
6	0.32	0.25
7	0.36	0.22
8	0.41	0.2
9	0.45	0.18
10	0.5	0.16

Tab.4.4 Coppie parametro-SNR. Il parametro moltiplicativo va a scalare il rumore in uscita da Simulink. L'SNR viene successivamente calcolato con il comando buil-in di Matlab snr. Il livello di rumore sta ad indicare l'ampiezza del rumore via via crescente.

4.5.2 Single Unit

Il single unit creato è stato creato a partire dalla distribuzione gaussiana inversa, presenta un MFR di 6,6 spikes/s con un totale 395 spikes in 60 s di segnale simulato. Il template utilizzato è Recording 2.

4.5.3 Multi Unit

Il Multi Unit che è stato creato andando a sommare 3 SU con le caratteristiche riportare in Tab.4.5 Il numero totale di spikes presenti è 1407, la MFR 23,4 spikes/s.

La rappresentazione grafica del segnale ottenuto al variare del rumore con gli istanti degli spikes sovrapposti è riportata in figura 4.8.

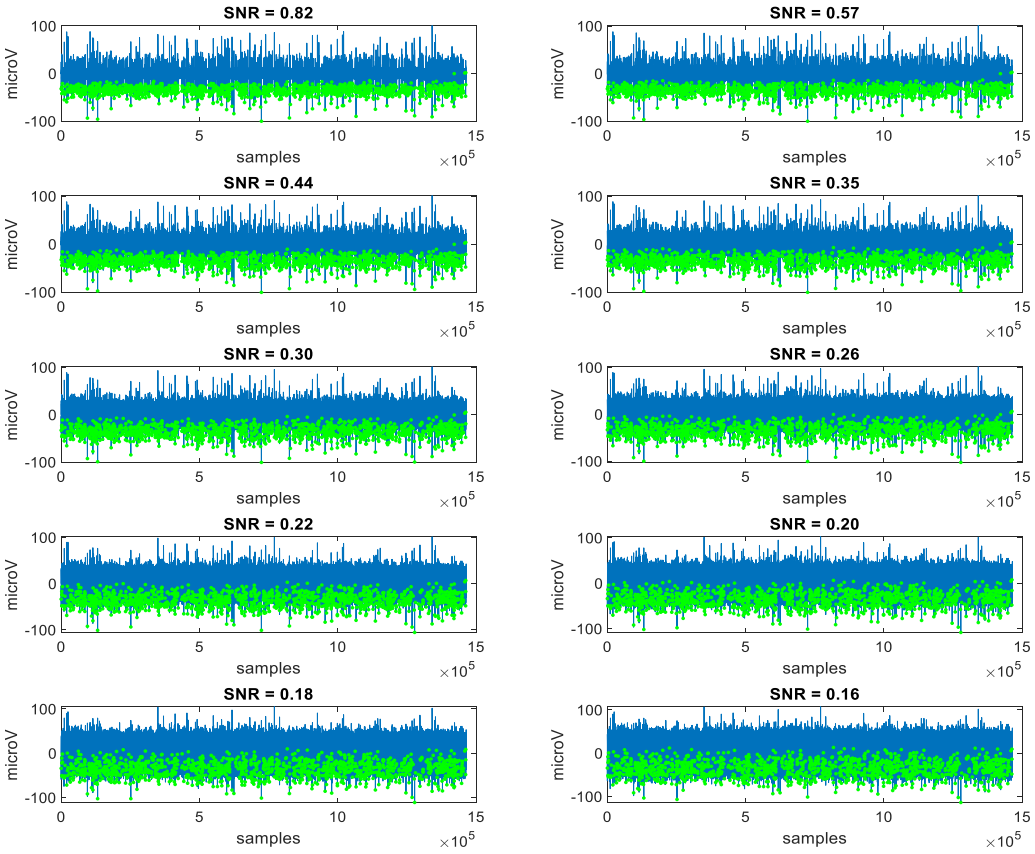


Fig.4.8 Multi Unit a seconda del livello del rumore presente. L'SNR diminuisce andando dall'alto verso il basso, con conseguente aumento del rumore nella stessa direzione. In verde sono segnate le posizioni degli spike. Queste non cambiano a livello di sample con l'aumentare del rumore ma cambiano a livello di ampiezza in quanto si sommano con il rumore presente che diventa sempre più forte.

Unità	Distribuzione	Template	Numero di spikes	MFR (spikes/s)
1	Gaussiana Inversa	Recording 3 Classe 1	121	2
2	Gamma	Recording 1 Classe 1	1218	20,3
3	Gaussiana Inversa	Recording 1 Classe 0	68	1,1

Tab.4.5 Unità con relativa distribuzione, template, numero di spikes e MFR che sommate hanno formato il MU.

Capitolo 5: mPTSD e indici di performance

Uno degli obiettivi di questa Tesi consiste nella modifica all'algoritmo Precise Timing Spike Detection (PTSD) (Maccione et al., 2009) le cui caratteristiche sono state descritte nel Capitolo 3. L'algoritmo PTSD era stato inizialmente pensato per analizzare segnali in-vitro (Bologna et al., 2010), ma viste le sue ottime performance si è pensato di impiegarlo anche per segnali acquisiti in vivo (Averna, 2019). Uno dei principali problemi di PTSD è la condizione di uscita di identificazione dello spike: l'algoritmo ricerca infatti dei punti di massimo/minimo relativo all'interno di un intervallo, ma questi punti non necessariamente coincidono con un picco. Questa caratteristica peggiora le performance generali dell'algoritmo ed è stata il punto di partenza per andarlo a modificare.

In questo capitolo vengono anche illustrati gli indici statistici per valutare le performance degli algoritmi e un confronto di alcuni di questi per mostrare che le migliorie apportate a PTSD ne hanno realmente incrementato le performance.

5.1 Modified Precise Timing Spike Detection (mPTSD)

In PTSD, la condizione per determinare la presenza o meno del 2° RMM (Fig.3.4), ovvero un massimo/minimo locale dopo che è già stato trovato un altro minimo/massimo locale, è che debba esserci un campione all'interno del PLP, o dell'overshoot, che sia minore, nel caso si stia cercando un minimo locale, o maggiore, nel caso si stia cercando

un massimo locale, di tutti i suoi campioni precedenti. In mPTSD, oltre a questa condizione, nel caso si stia cercando il minimo, il campione deve essere anche minore del campione successivo, mentre, se si sta cercando un massimo, il campione deve essere anche maggiore di quello successivo. Queste condizioni fanno in modo che venga proprio ricercato un picco massimo o minimo nel segnale dopo che è già stato identificato un altro picco minimo o massimo. La ricerca del secondo picco avviene sempre all'interno del periodo di tempo chiamato PLP, e al massimo dell'overshoot se non viene trovato il picco entro la fine del PLP. La sogliatura unipolare viene sempre effettuata sul picco negativo, e lo spike assegnato all'istante di tempo del picco negativo corrispondente. Se il secondo picco non viene trovato, non viene assegnato lo spike anche se è presente un minimo/massimo relativo nell'intervallo di ricerca. È sempre presente il periodo refrattario RP pari a 24 campioni. Uno schema di funzionamento di mPTSD è visibile in Fig.5.1.

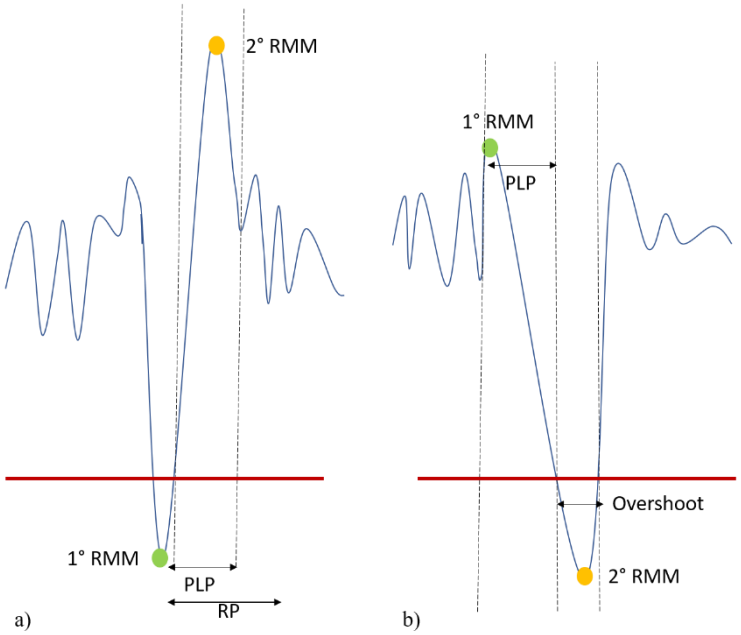


Fig.5.1 Schema di funzionamento di mPTSD. a) Il 1° RMM viene trovato (verde) ed è un minimo, l'algoritmo ricerca allora il massimo più vicino all'interno del PLP, trovando così il 2° RMM (giallo). Il picco negativo (verde) supera la soglia (rosso), lo spike viene quindi associato al 1° RMM (verde). È mostrato anche RP, maggiore di PLP. b) Viene trovato il 1° RMM come massimo (verde), il minimo più vicino non viene trovato entro la fine del PLP. In questo caso parte la finestra dell'overshoot che

permette di andare a trovare il minimo corretto (2° RMM, giallo), a cui viene assegnato lo spike in quanto campione minimo del picco negativo che supera la soglia (rosso).

Se dalla figura precedente, il funzionamento può sembrare pressoché identico a PTSD, che si ricorda è riportato in Fig.3.4. In figura 5.2 vengono mostrati degli esempi in cui mPTSD agisce correttamente mentre PTSD fallisce. Utilizzando come colore di riferimento il verde per PTSD e il rosso per mPTSD, si può vedere come vadano ad agire le migliorie apportate. La 5.2a) mostra un esempio in cui la finestra di tempo PLP insieme all'overshoot non va a coprire la durata totale dello spike, e che quindi, visto che viene ricercato semplicemente un minimo rispetto al campione precedente, lo spike non venga associato al picco, cosa invece che fa mPTSD. Nella Fig.5.2b) si può vedere ancora una volta come il semplice controllo sulla condizione di entrata non basti: l'algoritmo riconosce infatti un massimo locale, e poi semplicemente associa lo spike al minimo più vicino che trova.

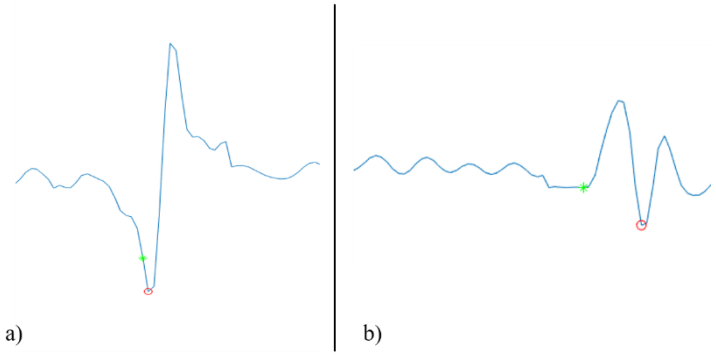


Fig.5.2 Diversi comportamenti per PTSD (verde) e mPTSD (rosso) nel riconoscimento degli spike. a) PTSD riconosce un massimo e comincia la ricerca del minimo più vicino entro il PLP+overshoot, campione segnato in verde e gli associa lo spike sbagliando così l'identificazione. b) PTSD trova un massimo e con un procedimento analogo a quello appena descritto, associa allo spike il minimo che trova, che però non fa nemmeno parte dello spike. Sia in a) che b) mPTS (rosso) trova correttamente lo spike.

Per quanto riguarda la soglia, il mantenimento della soglia differenziale ha mostrato un incremento nelle performance quasi nullo. Le performance migliorano decisamente, insieme alla modifica sulla

condizione di uscita, con l'utilizzo di una soglia unipolare. Ricordando che la soglia è definita come un parametro moltiplicativo (*MultCoeff*) che moltiplica la deviazione standard del rumore, il miglioramento dovuto all'utilizzo della soglia unipolare èprobabilmente imputabile al fatto che un parametro moltiplicativo per una soglia unipolare sia più facile da settare che per una soglia differenziale, migliorando di conseguenza le performance dell'algoritmo.

5.2 Indici di valutazione delle performance

Gli istanti di tempo in cui gli algoritmi segnalano essere presente gli spikes, vengono confrontati con gli istanti di tempo degli spikes dati dal modello per andare vedere quanti e quali di questi istanti, detti anche *timestamps* (TS), siano effettivamente spikes. Possono verificarsi quattro casi a cui corrispondono quattro etichette differenti:

Vero Positivo: l'algoritmo riconosce un'istante come spike e questo è presente anche nel modello, identificazione corretta *True Positive* (TP). L'obiettivo degli algoritmi è di massimizzarli per farli coincidere con i TS effettivamente presenti nel modello.

Falso Positivo: l'algoritmo riconosce come spike un TS che non è presente nel modello, identificazione scorretta *False Positive* (FP). In questo caso, l'algoritmo deve cercare di contenerne il numero il più possibile.

Vero Negativo: l'algoritmo non riconosce il TS come spike e questo non è presente nemmeno nel modello, identificazione corretta *True Negative* (TN). La definizione di Negativo però non è immediata in questo contesto e non è nemmeno univoca per chi lavora in questo settore, si è scelto di rappresentare i negativi come delle finestre di tempo lunghe quanto uno periodo refrattario, 1 ms o 24 campioni alla frequenza di 24 kHz, in cui non sono presenti eventi, Eq.16.

$$Negative = \frac{lunghezza\ del\ segnale - NREF \cdot wlen}{wlen} \qquad Eq.\ 16$$

Dove NREF è il numero di spikes presenti nel modello e wlen è la finestra di tempo sopra citata pari a 24 campioni. Ovviamente i Negative sono un numero molto maggiore dei Positive.

Falso Negativo: istanti di tempo che l'algoritmo non riconosce come spike ma che in realtà sono presenti nel modello, identificazione scorretta *False Negative* (FN). Anche questo numero è da minimizzare.

Le performance degli algoritmi possono essere valutate attraverso degli indici statistici che si basano sul numero dei TP, FP, TN, FN. In questa Tesi ne sono stati utilizzati dodici indici che andrò ora ad analizzare.

Sensitivity: detto anche *True Positive Rate* (TPR), si riferisce all'abilità dell'algoritmo di detettare correttamente gli eventi effettivamente presenti nel segnale. In altre parole, è la proporzione di TS identificati come TP tra tutti gli eventi presenti nel modello, matematicamente (Eq.17):

$$Sensitivity = \frac{TP}{TP+FN} \qquad Eq.\ 17$$

Ricordando che nel caso ideale FN tende a 0, la Sensitivity ideale tende a 1. Il suo complementare è la *False Negative Rate*.

Specificity: detto anche *True Negative Rate* (TNR), si riferisce all'abilità dell'algoritmo di rifiutare correttamente gli istanti privi di eventi. Detto diversamente, è la proporzione di TS identificati come TN su tutti i N (Eq.18):

$$Specificity = \frac{TN}{N} \qquad Eq.\ 18$$

Nel caso ideale, TN tende ad N, quindi la specificity sarà pari a 1. Il suo complementare è la *False Positive Rate*.

Precision: o *Positive Predictive Value* (PPV), è l'abilità dell'algoritmo di detettare correttamente gli eventi, quindi un rapporto tra tutto ciò che identifica come Positive. Matematicamente (Eq.19):

$$Precision = \frac{TP}{TP+FP} \qquad Eq.\ 19$$

Idealmente FP tende a 0, quindi la precision tenderà a 1. Il suo complementare è la *False Discovery Rate*.

Negative Predictive Value (NPV): è l'abilità dell'algoritmo di detettare correttamente tutti quelli che sono non-eventi, ovvero le finestre prive di spike, quindi un rapporto tra tutto ciò che identifica come Negative. Matematicamente (Eq.20):

$$NPV = \frac{TN}{TN+FN} \qquad \textit{Eq. 20}$$

Idealmente FN tende a 0, quindi NPV tenderà a 1. Il suo complementare è la *False Omission Rate*.

False Negative Rate (FNR): o Miss Rate (MR), è il complementare della TPR, quindi i suoi termini saranno gli opposti di quelli della TPR (dove prima c'era True ora ci sarà False e dove c'era Positive ci sarà Negative). Matematicamente è (Eq.21):

$$FNR = \frac{FN}{FN+TP} \qquad \textit{Eq. 21}$$

Idealmente FN tende a 0, quindi indipendentemente da TP, nel caso ideale FNR tende a 0.

False Positive Rate (FPR): o *Fall-Out* (FO), è il complementare della specificity, si riferisce all'abilità dell'algoritmo di identificare gli eventi che siano corretti o meno. Matematicamente (Eq.22):

$$FPR = \frac{FP}{FP+TP} \qquad \textit{Eq. 22}$$

Nel caso ideale, FP tende ad 0, quindi indipendentemente da TP, la FPR sarà pari a 0.

False Discovery Rate (FDR): è il complementare della precision, Matematicamente (Eq.23)

$$FDR = \frac{FP}{FP+TP} \qquad \textit{Eq. 23}$$

Idealmente FP tende a 0, quindi la FDR tenderà a 0 nel caso ideale indipendentemente da TP.

False Omission Rate (FOR): è il complementare di NPV, è la misura di quanto l'algoritmo sbaglia nel detettare i non-eventi. Matematicamente (Eq.24):

$$FOR = \frac{FN}{FN+TN} \qquad Eq.\ 24$$

Idealmente FN tende a 0, quindi FOR nel caso ideale tende a 0. I TN, però, sono un numero nettamente maggiore, quindi questo indice avrà sempre un valore molto basso.

Efficiency: o *Critical Success Index* (CSI), è un buon indice riassuntivo delle prestazioni generali dell'algoritmo. È definito in questo modo (Eq.25):

$$Efficiency = \frac{TP}{NREF+FN} \qquad Eq.\ 25$$

L'obiettivo degli algoritmi è di massimizzare questo indice, che idealmente tende a 1.

Accuracy: altro indice delle prestazioni in generale. La sua definizione è la seguente (Eq.26):

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \qquad Eq.\ 26$$

Considerando che, FP e FN nel caso ideale tendono a 0, l'accuracy ideale tende a 1.

F1 score: è la media armonica della precision e della sensitivity. Matematicamente (Eq.27):

$$F1\ score = \frac{2 \cdot TP}{2 \cdot TP+FP+FN} \qquad Eq.\ 27$$

Visto che, FP e FN idealmente tendono a 0, nel caso ideale questo indice tende a 1.

Matthews Correlation Coefficient: indice solitamente utilizzato nel Machine Learning come misura di qualità di un classificatore dicotomico. È equivalente all'indice di correlazione di Pearson, ma il termine MCC è maggiormente usato nel campo della bioinformatica. Dà una buona valutazione delle performance dell'algoritmo anche

quando le classi sono di grandezza molto diversa, proprio come nel caso analizzato dove gli eventi positivi (cioè gli spikes) sono molto minori degli eventi negativi (cioè le finestre prive di spikes). La sua formulazione matematica è la seguente (Eq.28):

$$MCC = \frac{TP \cdot TN - FP \cdot FN}{\sqrt{(TP+FP)\cdot(TP+FN)\cdot(TN+FP)\cdot(TN+FN)}} \qquad \textit{Eq. 28}$$

Questo indice è compreso tra -1 e +1, dove -1 è il caso peggiore e +1 il migliore.

In tabella 5.1 sono riportati gli andamenti degli indici all'aumentare delle quattro etichette TP, FP, TN, FN. Ovviamente, se questi diminuissero gli indici si comporterebbero al contrario.

	TPR	TNR	PPV	FNR	NPV	FPR	FDR	FOR	CSI	ACC	F1	MCC
↑TP	↑	—	↑	↓	—	—	↓	—	↑	↑	↑	↑
↑FP	—	—	↓	—	—	↑	↑	—	—	↓	↓	↓
↑FN	↓	—	—	↑	↓	—	—	↑	↓	↓	↓	↓
↑TN	—	↑	—	—	↑	—	—	↓	—	↑	—	↑
Ideale	1	1	1	0	1	0	0	0	1	1	1	1

Tab.5.1 Andamento degli indici all'aumentare ti TP, FP, FN, TN. Le frecce verso l'alto indicano l'aumento dell'indice, quelle verso il basso la sua diminuzione. L'ultima riga rappresenta i valori ideali che ogni parametro dovrebbe raggiungere. Le righe orizzontali stanno a indicare che quel parametro non è coinvolto nel calcolo dell'indice. Logicamente, l'aumento di TP è un parametro positivo in tutti gli indici in cui è presente, mentre l'aumento di FP è dannoso allo stesso modo. Stesso discorso per FN e TN: l'aumento del primo incide negativamente, l'aumento del secondo positivamente.

5.3 Confronto tra PTSD e mPTSD

Essendo lo sviluppo di una miglioria all'algoritmo PTSD uno degli obiettivi di questa Tesi, vengono riportati per completezza i confronti effettuati tra PTSD e mPTSD. Sono mostrati i grafici di confronto dei sei indici più informativi, ovvero: Efficiency (Fig.5.2) Sensitivity (Fig.5.3), Precision (Fig.5.4), Specificity (Fig.5.5), F1 Score (Fig.5.6) e MCC (Fig.5.7). Questi vengono mostrati per tre livelli di SNR (basso 0.16, medio 0.29, alto 0.86) per tutti i 20 MultCoeff utilizzati. Infine, è presente un confronto grafico dell'indice utilizzato per raccogliere tutti i risultati, e che viene illustrato nel capitolo 6, Final Score. Questo punteggio, che al massimo può raggiungere il valore 12, indica le performance totali dell'algoritmo ed è utile a capire quale dei due sia il migliore. Dall'analisi di questo indice, l'algoritmo migliore risulta essere mPTSD, il quale raggiunge il valore di FS maggiore sia per i livelli di SNR alti che bassi (Fig.5.8).

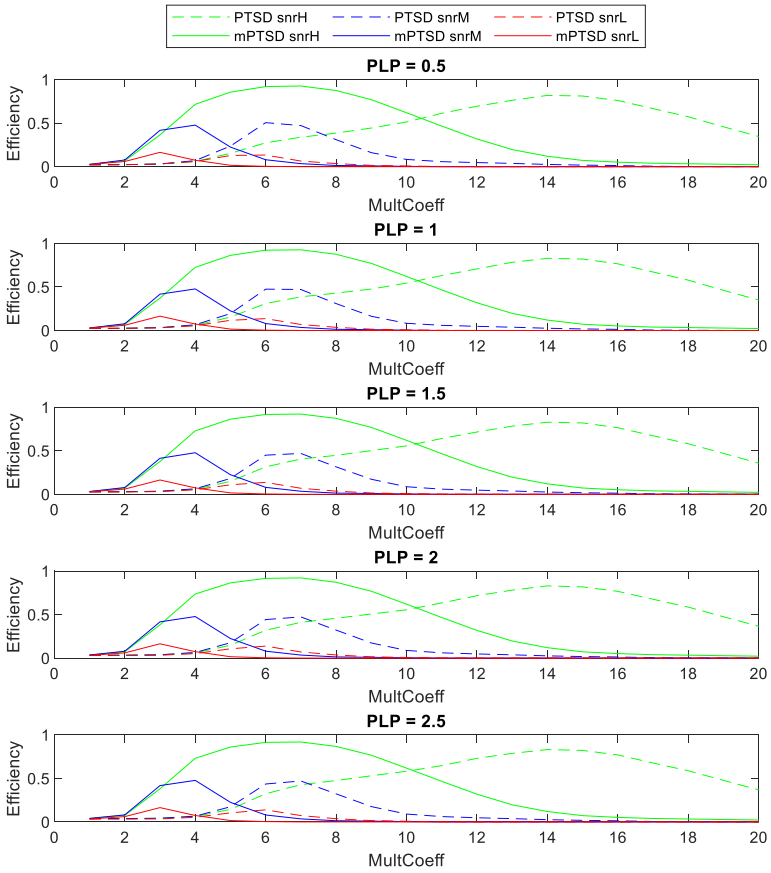


Fig.5.2. Efficiency: il PLP non influisce sull'effeciency, i grafici sono infatti praticamente identici. mPTSD e PTSD sono comparabili, anche se mPTSD performa leggermente meglio nel caso di SNR minore e maggiore.

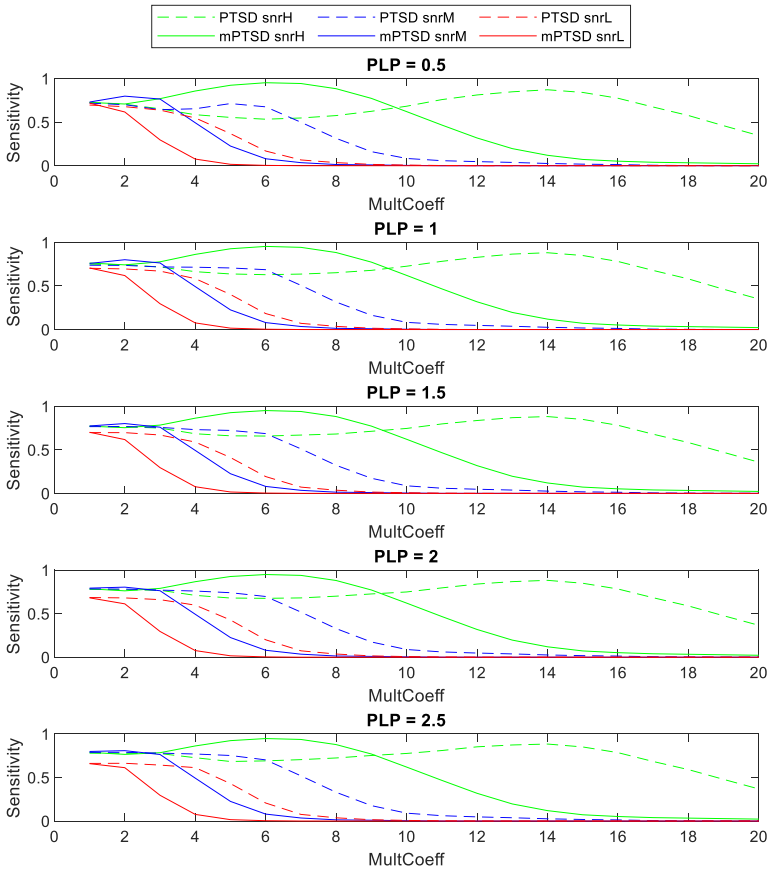


Fig.5.3 Sensititvity: Il PLP in questo caso influisce maggiormente sulle performance, si può vedere che nel PLP più piccolo il valore di Sensitivity nel caso di SNR peggiore (snrL, rosso) è maggiore, e diminuisce man mano aumenta il PLP. Le performance di PTSD sembrano una versione traslata di quelle di mPTSD, dovuto probabilmente alla diversa definizione di soglia nei due algoritmi che per questo rispondo meglio a MultCoeff diversi. La pendenza delle curve è comparabile e anche in questo caso le performance di mPTSD e PTSD non sono particolarmente differenti.

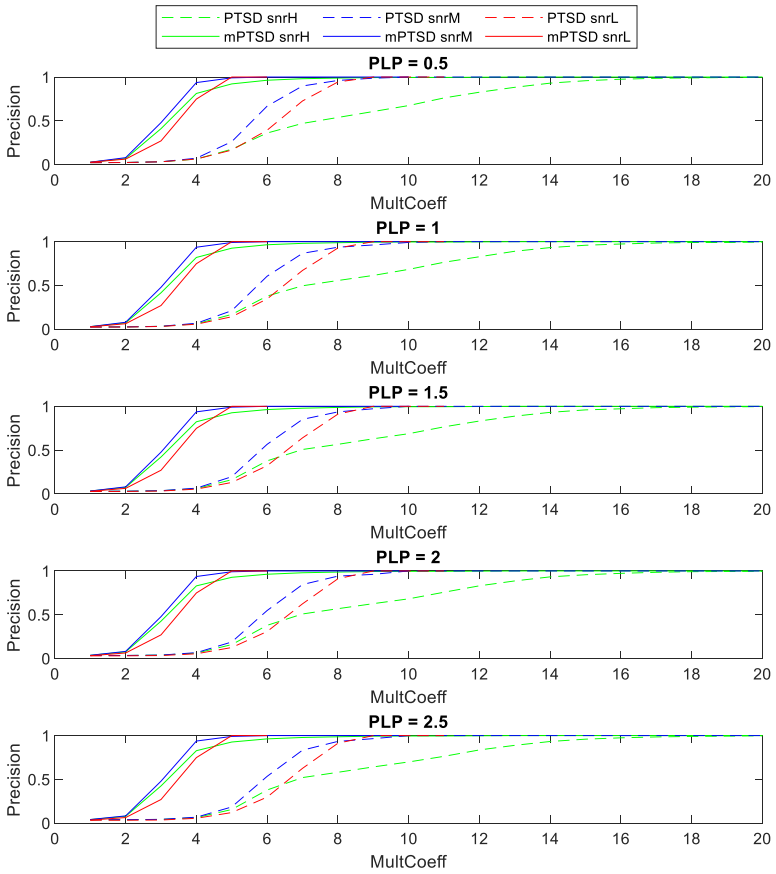


Fig.5.4 Precision: In questo indice, mPTSD performa meglio in quanto le salite delle rette ad ogni PLP e livello di SNR sono più ripide, questo significa che viene raggiunto più velocemente il valore ideale.

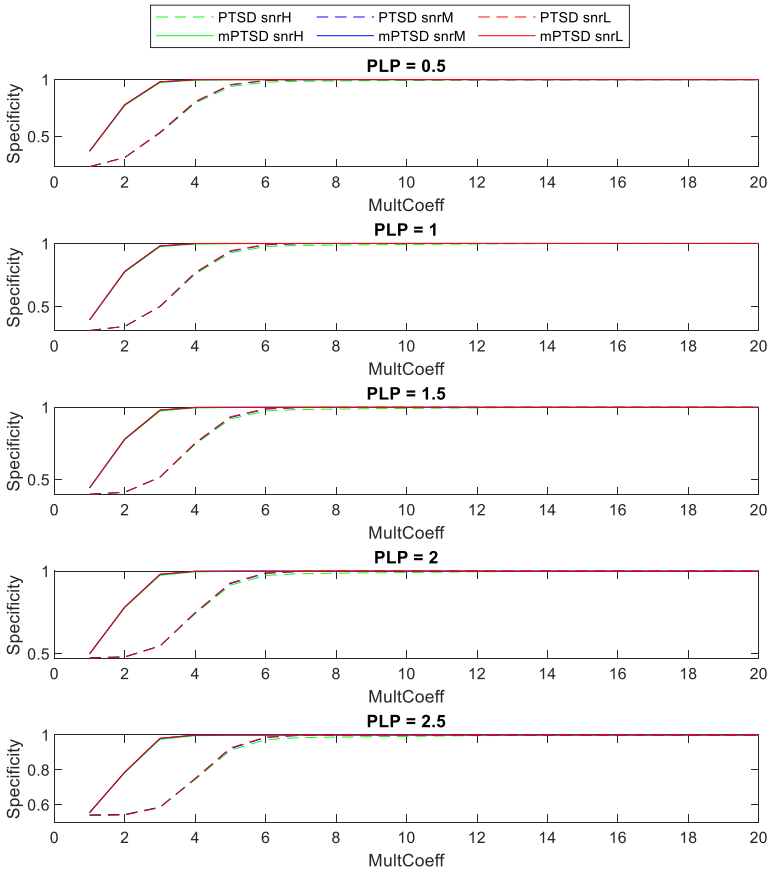


Fig.5.5 Specificity: Questo indice non varia né al variare del PLP né dell'SNR, tanto che i grafici si presentano quasi uguali l'uno all'altro. mPTSD presenta una rampa in salita più ripida, significa che raggiunge prima il valore ideale e quindi può essere definito migliore dal punto di vista della Specificity.

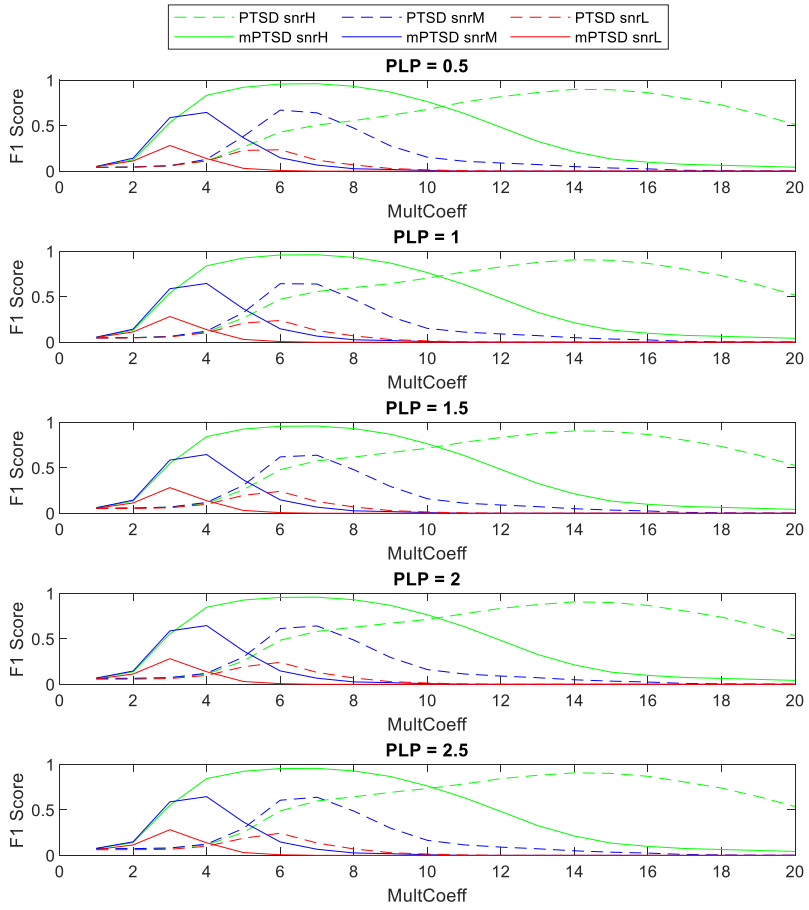


Fig.5.6 F1 Score: Questo indice ha un'andamento molto simile a quello di Efficiency, il PLP infatti non influisce sui suoi valori e in grafici si presentano praticamente identici l'uno all'altro. mPTSD e PTSD hanno performance molto simili, mPTSD si comporta leggermente meglio nei livelli di SNR minore e maggiore, ma i due algoritmi restano sempre comparabili.

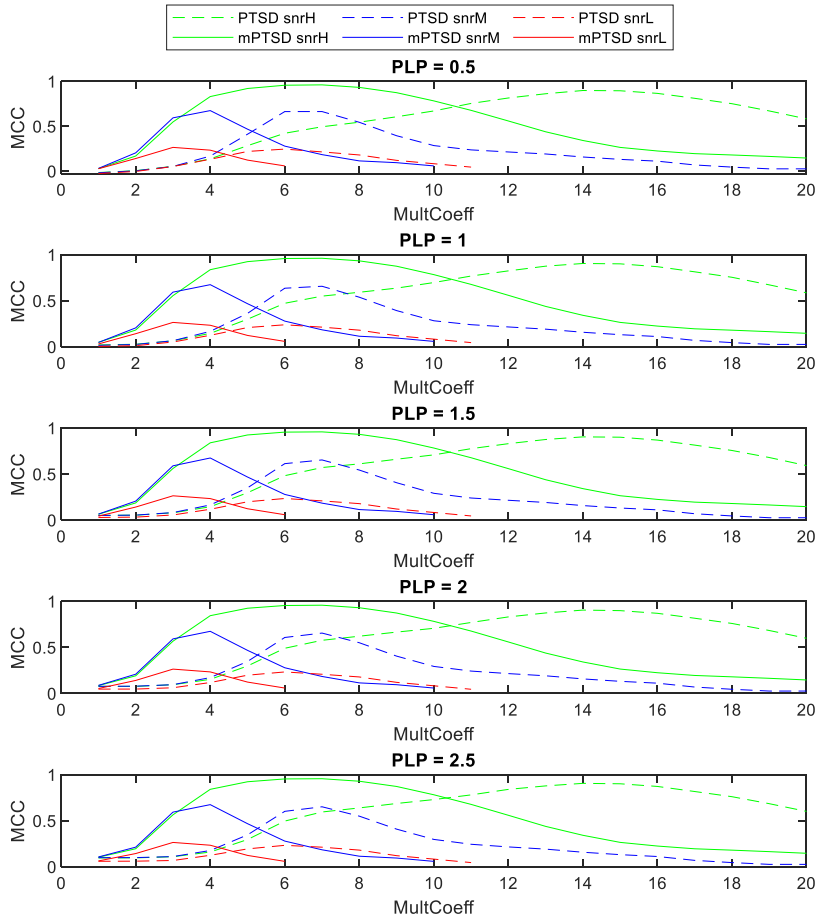


Fig.5.7MCC: Anche in questo caso l'aspetto grafico è simile a quello di Efficiency e F1 Score. La differenza sta nel fatto che le curve relative a mPTSD si interrompono e non arrivano a coprire tutti coefficienti. Questo significa che sono presenti dei NaN all'interno della formulazione di MCC, dovuto al fatto che per una soglia unipolare utilizzare un MultCoeff troppo elevato comporta non trovare alcuno spike. PTSD copre invece tutti i valori perché possiede una soglia differenziale e quindi può abbracciare un range di coefficienti più ampio. I valori più alti sono comunque raggiunti da mPTSD.

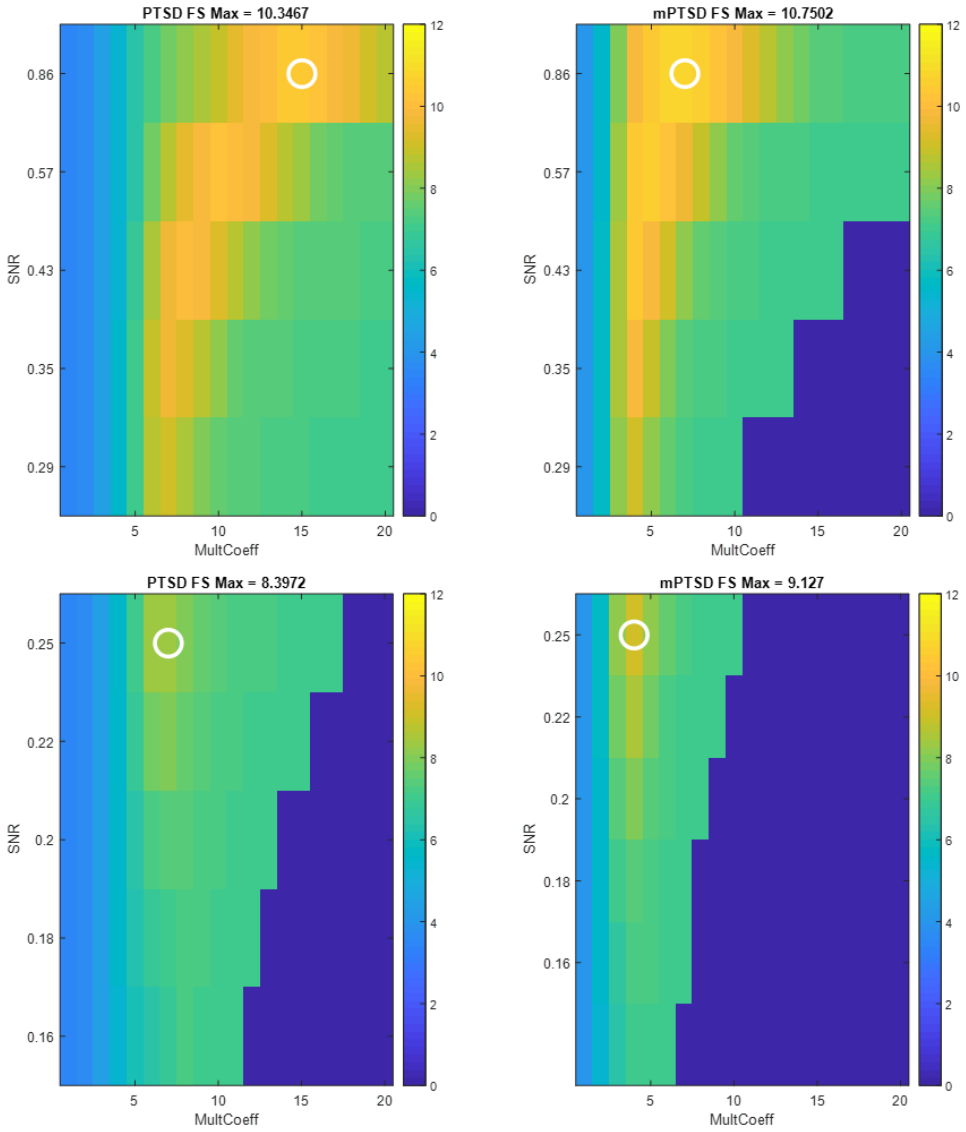


Fig. 5.8 Final Score: Questo è il valore che riassume tutti gli indici utilizzati. Nel calcolarlo sono stati inseriti anche gli altri indici di cui non sono presenti i grafici in quanto non sono altro che l'opposto di quelli mostrati o, come nel caso dell'Accuracy, poco informativi in questo contesto. In questo grafico sulla sinistra troviamo PTSD e sulla destra mPTSD. La riga in alto contiene i livelli di SNR più elevati, e quindi i casi più facili, e la riga in basso i livelli i SNR più bassi, casi difficili ma più interessanti dal punto di vista prestazionale. Visto che il PLP non incideva particolarmente, come visto nei grafici precedenti, è stato fissato al primo valore per entrambi, 0.5 ms. Il Finale Score (FS) più elevato (cerchio bianco) ottenuto da ***PTSD*** *nel caso migliore è* ***10.34****, contro il* ***10.75*** *di* ***mPTSD.*** *Nel caso peggiore continua a essere inferiore* ***PTSD*** *con* ***8.39****, mentre* ***mPTSD*** *arriva a* ***9.12****.*

Capitolo 6: Risultati e confronto tra gli algoritmi

In questo capitolo sono presentati i risultati ottenuti, sia con il segnale SU che con il MU, per tutti gli algoritmi proposti. Prima però è necessario fare una breve introduzione sui metodi di valutazione e sulla procedura di matching degli spikes.

6.1 Spikes Matching

Per confrontare gli algoritmi, viene effettuato un *matching* tra gli istanti di tempo che vengono riconosciuti come spike e quelli presenti nel modello. Viene creata una finestra di tolleranza di 20 campioni, 10 prima e 10 dopo il timestamp, attorno ad ogni istante in cui l'algoritmo ha detettato uno spike: se in questa finestra cade un TS presente nel modello allora al TS trovato viene assegnata l'etichetta di Vero Positivo. Dieci campioni alla frequenza di 24414 Hz equivalgono a 0.4 ms, la finestra totale in cui si effettua la ricerca è quindi pari a 0.8 ms. La differenza tra TS trovato e il TS presente nel modello è chiamata *jitter*, più questo valore è piccolo e più l'algoritmo è preciso nell'identificazione degli spikes corretti. Algoritmi meno precisi, come HT e HTLM, presentano valori di jitter più elevati. Purtroppo, data la numerisità dei grafici, questo dato è stato calcolato ma non riportato in questo contesto.

Per la valutazione degli algoritmi di SD non esistono dei criteri univoci. In questa Tesi sono stati utilizzati 12 indici perché ognuno di

questi esplora un aspetto diverso delle performance degli algoritmi: ad esempio, un algoritmo può avere una sensitivity pari a 1 (e quindi molto alta), ma essere in grado di identificare un numero molto ridotto di spikes, peccando quindi in efficiency.

6.2 Formulazione del punteggio finale: Final Score

Per raggruppare insieme tutte le diverse informazioni, si è deciso di sommare gli indici per ottenere un valore chiamato *Final Score* (FS). Questo valore può raggiungere un massimo di 12, e verrà utilizzato per confrontare gli algoritmi condensando tutti gli indici di prima in un solo parametro. Per gli indici che idealmente tendono a 0, è stato calcolato il complementare e poi sommato agli altri. Dall'analisi di FS si riesce inoltre ad avere un'idea della robustezza dell'algoritmo: ovvero dei cambiamenti delle performance di questo al variare dei parametri. Un algoritmo robusto non cambia infatti le sue performance se i parametri subiscono piccole variazioni. Questa è un'analisi molto utile perché la SD, soprattutto in applicazioni real-time, prevede di fissare una soglia "alla cieca" ovvero quella che l'operatore ritiene più adatta a priori. Quindi algoritmi che non variano troppo le performance variando i parametri sono i migliori da utilizzare.

6.3 HT

L'algoritmo HT prevede la variazione di un solo parametro (hMultCoeff, MC) che va a moltiplicare la soglia. MC è stato fatto variare 10 volte nell'intervallo 1-10, e l'analisi è stata effettuata per tutti i livelli di rumore creati.

I dodici indici sono stati calcolati dopo il matching degli spikes trovati dall'algoritmo. I grafici in Fig.6.1 e Fig.6.2 ne riportano l'andamento al variare di MultCoeff per 3 livelli di SNR: 0.16, 0.29 e 0.86.

6.3.1 Single Unit

Dai grafici in figura 6.1 specialmente dai grafici di efficiency, MCC e F1Score, si nota che le prestazioni generali non sono delle migliori: solitamente si fa riferimento al livello di SNR minore perché è il caso più realistico, e, in questo caso, i massimi valori raggiunti si attestano sempre sotto 0.2 per gli indici sopra citati, valore piuttosto basso. Si può anche già notare che l'algoritmo non è particolarmente robusto in quanto possiede dei picchi di massimo in corrispondenza di alcuni parametri che scendono subito al parametro successivo. Vediamo ora, per i tre livelli di SNR con il parametro che massimizza gli indici di performance generale (MC = 4), i valori di ciascun indice (tab. 6.2).

SU	TPR	TNR	PPV	FNR	NPV	FPR	FDR	FOR	CSI	ACC	F1	MCC
SNR 0.16	0.03	0.99	0.27	0.96	0.99	0.0005	0.72	0.006	0.03	0.99	0.05	0.09
SNR 0.29	0.38	0.99	0.79	0.61	0.99	0.0006	0.20	0.004	0.34	0.99	0.51	0.55
SNR 0.86	0.77	0.99	0.72	0.22	0.99	0.0019	0.27	0.001	0.59	0.99	0.74	0.74

Tab. 6.2. Valori raggiunti per ciascun indice da HT con MC=4 per tre livelli di SNR nel SU. I valori degli indici tendono a migliorare all'aumentare dell'SNR ma così non è per TNR, NPV e ACC (accuracy) che mantengono lo stesso valore. Si ricorda che CSI (efficiency) F1 e MCC sono gli indici di prestazione generale e rappresentano la chiave di lettura delle performance dell'algoritmo. HT al suo meglio, nel caso migliore, raggiunge un ***CSI 0.59, F1 0.74*** *e* ***MCC 0.74****. Nel caso peggiore, rispettivamente raggiunge* ***0.03, 0.05*** *e* ***0.09****.*

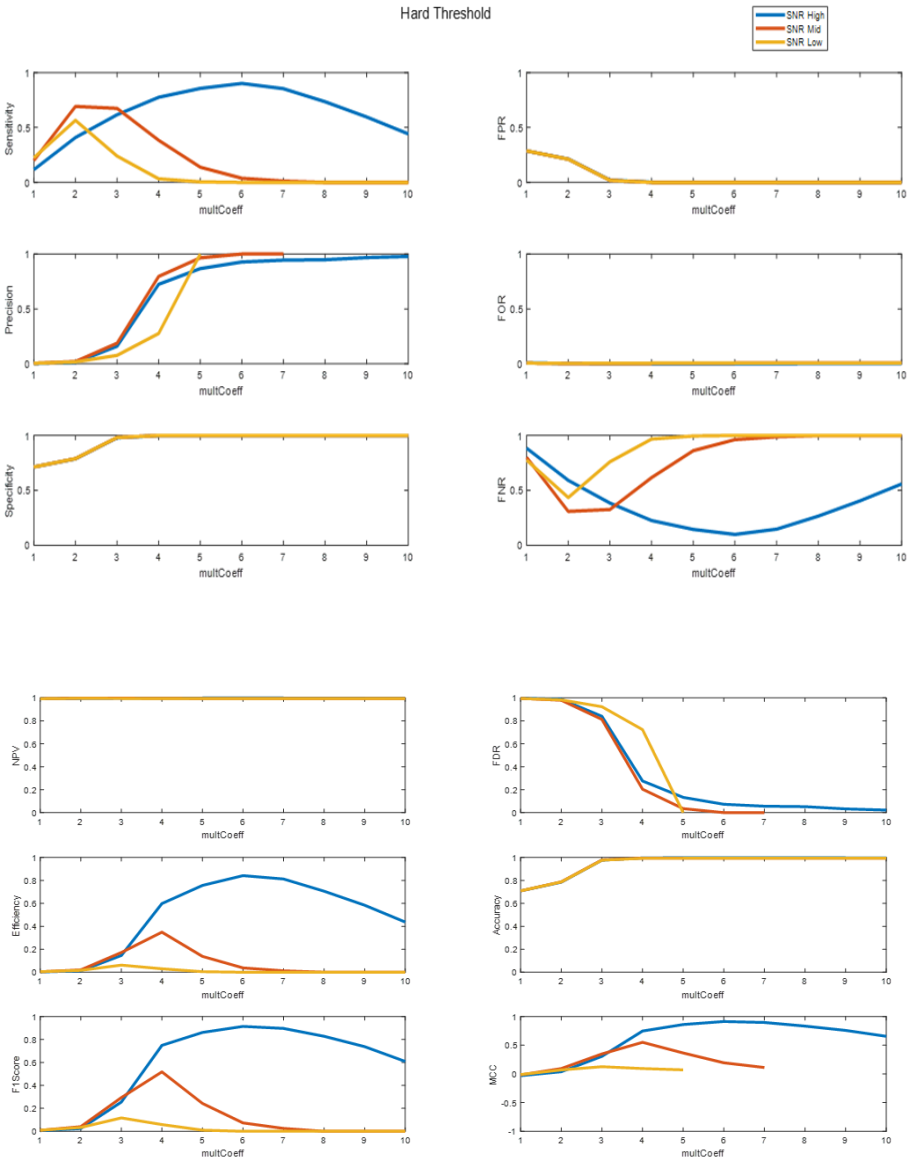


Fig.6.1 Rappresentazione grafica per tutti gli indici di HT su SU. In blu sono rappresentati i risultati relativi al livello di SNR maggiore (SNR High), in rosso al livello intermedio

6.3.2 Multi Unit

Le considerazioni sul MU sono analoghe a quelle del SU, si fa notare solo un aumento delle prestazioni in generale, forse dovuto al fatto che siano presenti più spikes e che quindi le probabilità di detettare uno spike siano maggiori. La tabella 6.3 è analoga alla tabella 6.2

MU	TPR	TNR	PPV	FNR	NPV	FPR	FDR	FOR	CSI	ACC	F1	MCC
SNR 0.16	0.07	0.99	0.74	0.92	0.97	0.0006	0.25	0.02	0.07	0.97	0.13	0.23
SNR 0.29	0.49	0.99	0.93	0.50	0.98	0.0007	0.06	0.01	0.47	0.98	0.64	0.67
SNR 0.86	0.78	0.99	0.80	0.21	0.99	0.0044	0.19	0.0051	0.65	0.99	0.79	0.79

Tab.6.3. Valori raggiunti per ciascun indice da HT con MC=4 per tre livelli di SNR nel MU. I valori degli indici tendono a migliorare all'aumentare dell'SNR anche in questo caso, e ugualmente TNR, NPV e ACC (accuracy) mantengono lo stesso valore. HT al suo meglio, nel caso migliore, raggiunge un ***CSI 0.65, F1 0.79 e MCC 0.79****. Nel caso peggiore, rispettivamente raggiunge* ***0.07, 0.13 e 0.23****. In entrambi i casi le performance sul MU sono migliori che quelle del SU.*

6.3.3 Tempo computazionale

Il tempo computazionale è stato calcolato facendo uso della funzione tic toc di Matlab, e varia anch'esso al variare della soglia e del livello di SNR. In Fig.6.3 è riportato l'andamento del tempo per il SU e per il MU. Il tempo computazionale varia anche a seconda che si tratti di un SU o di MU, e raggiunge valori maggiori proprio con il secondo. Il primo valore di entrambi è maggiore perché si tratta del livello di SNR più alto combinato alla soglia più bassa: verranno identificati più TS come spike, aumentando di conseguenza il tempo computazionale. Il massimo raggiunto nel SU è 0.085 s, mentre nel MU 0.11 s. Si tratta comunque di un algoritmo piuttosto veloce.

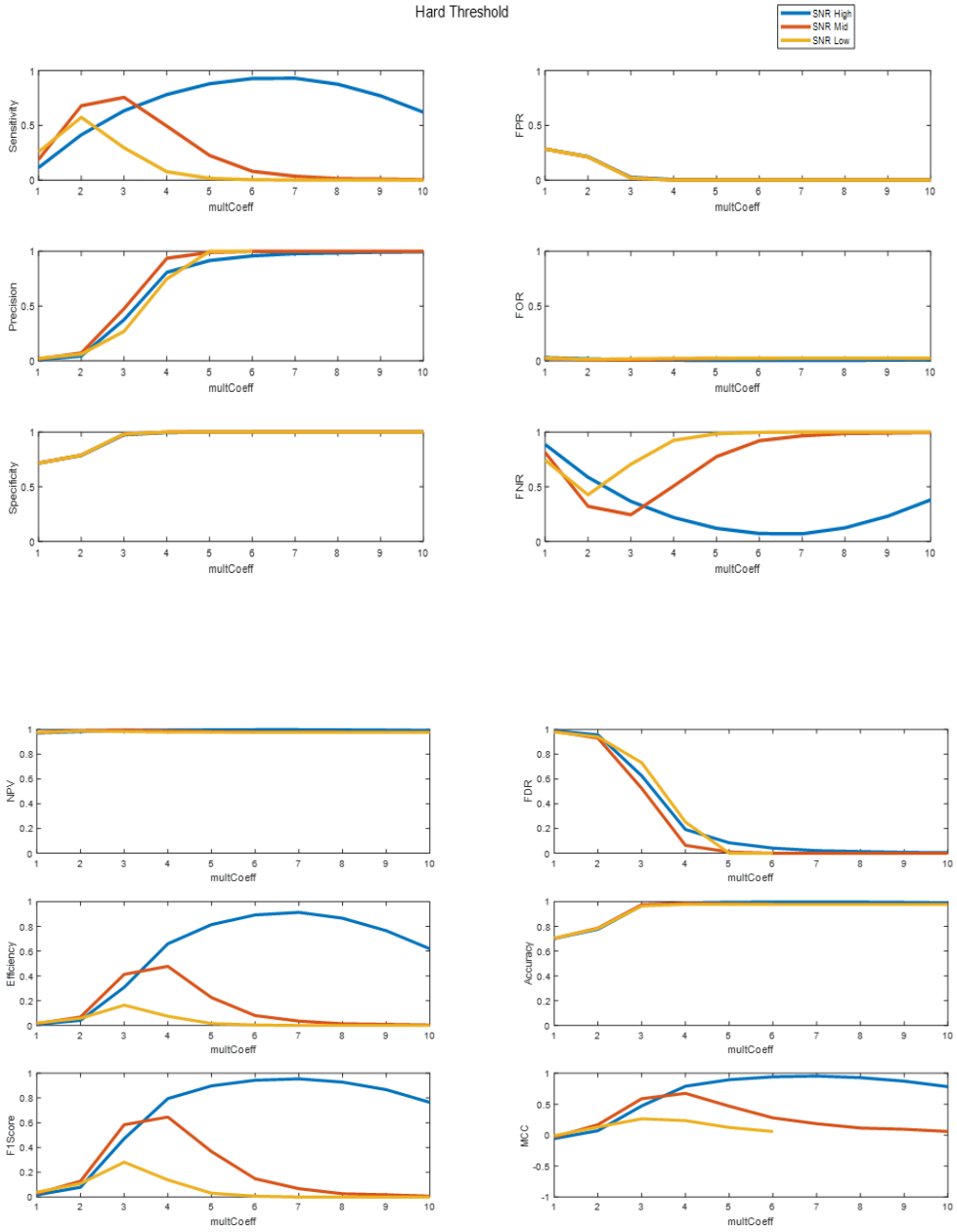


Fig.6.2 Rappresentazione grafica per tutti gli indici di HT su MU. In blu sono rappresentati i risultati relativi al livello di SNR maggiore (SNR High), in rosso al livello intermedio (SNR Mid) e in giallo al livello più basso (SNR Low).

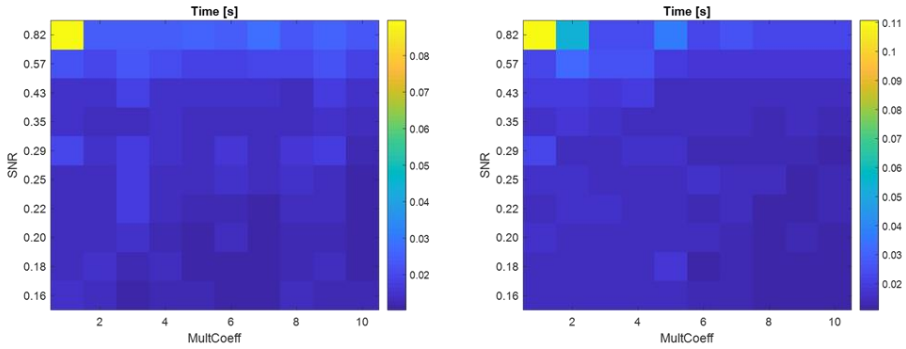


Fig.6.3 Tempi computazionali di HT. Sinistra, tempo per il SU: il tempo massimo è raggiunto per il livello di SNR più alto e il MC più basso. Destra, tempo per MU: analogamente, il tempo massimo è raggiunto con l'SNR più alto e MC più basso. Il primo valore più in gialloè probabilmente dovuto ad un problema di inizializzazione: non è da considerarsi quindi valido.

6.4 HTLM

Anche l'algoritmo HTLM prevede la variazione di un solo parametro (MultCoeff) che va a moltiplicare la soglia. MultCoeff è stato fatto variare 10 volte nell'intervallo 1-10, e l'analisi è stata effettuata per tutti i livelli di rumore creati.

I dodici indici sono stati sempre calcolati dopo il matching degli spikes trovati dall'algoritmo. I grafici in Fig. 6.4 e Fig. 6.5 ne riportano l'andamento al variare di MultCoeff per gli stessi 3 livelli di SNR di sopra per SU e per MU.

6.4.1 Single Unit

Dai grafici in figura N., si può vedere che le prestazioni non sono particolarmente diverse da HT, e stesso discorso può essere fatto per la robustezza. Anche in questo algoritmo, il parametro migliore sembra essere MultCoeff = 4. In tabella 6.4 sono riportati, analogamente al caso precedente, i valori per ogni indice ai tre livelli di SNR per MultCoeff = 4.

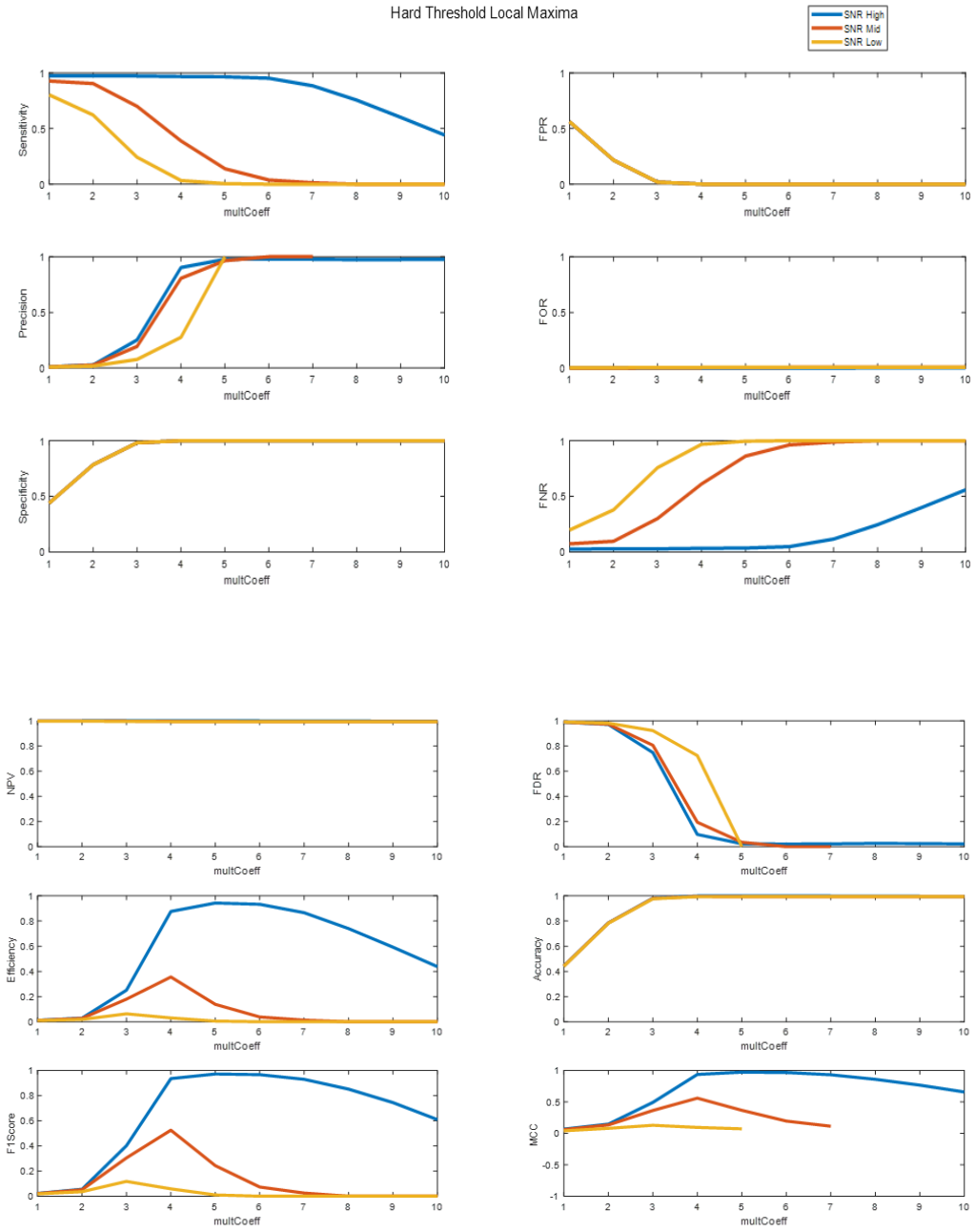


Fig.6.4 Rappresentazione grafica per tutti gli indici di HTLM su SU. In blu sono rappresentati i risultati relativi al livello di SNR maggiore (SNR High), in rosso al livello intermedio (SNR Mid) e in giallo al livello più basso (SNR Low).

SU	TPR	TNR	PPV	FNR	NPV	FPR	FDR	FOR	CSI	ACC	F1	MCC
SNR 0.16	0.03	0.99	0.27	0.96	0.99	0.0005	0.72	0.006	0.03	0.99	0.05	0.09
SNR 0.29	0.38	0.99	0.80	0.61	0.99	0.0006	0.19	0.004	0.35	0.99	0.52	0.55
SNR 0.86	0.96	0.99	0.90	0.03	0.99	0.0006	0.09	0.0002	0.87	0.99	0.93	0.93

Tab.6.4 Valori raggiunti per ciascun indice da HTLM con MC=4 per tre livelli di SNR nel SU. I valori degli indici tendono a migliorare all'aumentare dell'SNR ma così non è per TNR, NPV e ACC (accuracy) che mantengono lo stesso valore. Si ricorda che CSI (efficiency) F1 e MCC sono gli indici di prestazione generale e rappresentano la chiave di lettura delle performance dell'algoritmo. HTLM al suo meglio, nel caso migliore, raggiunge un ***CSI 0.87, F1 0.93 e MCC 0.93****. Nel caso peggiore, rispettivamente raggiunge* ***0.03, 0.05 e 0.09****. Nel caso peggiore raggiunge gli stessi valori di HT, mentre nel caso migliore gli è superiore.*

6.4.2 Multi Unit

Le considerazioni sul MU sono analoghe a quelle fatte precedentemente per HT, anche in questo caso si nota un aumento delle prestazioni in generale rispetto al SU. Questa tabella (Tab.6.5) è pressoché identica a Tab.6.4 relativa al MU di HT.

MU	TPR	TNR	PPV	FNR	NPV	FPR	FDR	FOR	CSI	ACC	F1	MCC
SNR 0.16	0.07	0.99	0.74	0.92	0.97	0.0006	0.25	0.02	0.07	0.97	0.13	0.23
SNR 0.29	0.49	0.99	0.93	0.50	0.98	0.0007	0.06	0.01	0.48	0.98	0.64	0.67
SNR 0.86	0.78	0.99	0.80	0.21	0.99	0.0044	0.19	0.0051	0.65	0.99	0.79	0.79

Tab.6.5 Valori raggiunti per ciascun indice da HTLM con MC=4 per tre livelli di SNR nel MU. I valori degli indici tendono a migliorare all'aumentare dell'SNR anche in questo caso, e ugualmente TNR, NPV e ACC mantengono lo stesso valore. HTLM al suo meglio, nel caso migliore, raggiunge un ***CSI 0.65, F1 0.79 e MCC 0.79****. Nel caso peggiore, rispettivamente raggiunge* ***0.07, 0.13 e 0.23****. In entrambi i casi si tratta degli stessi valori di HT*

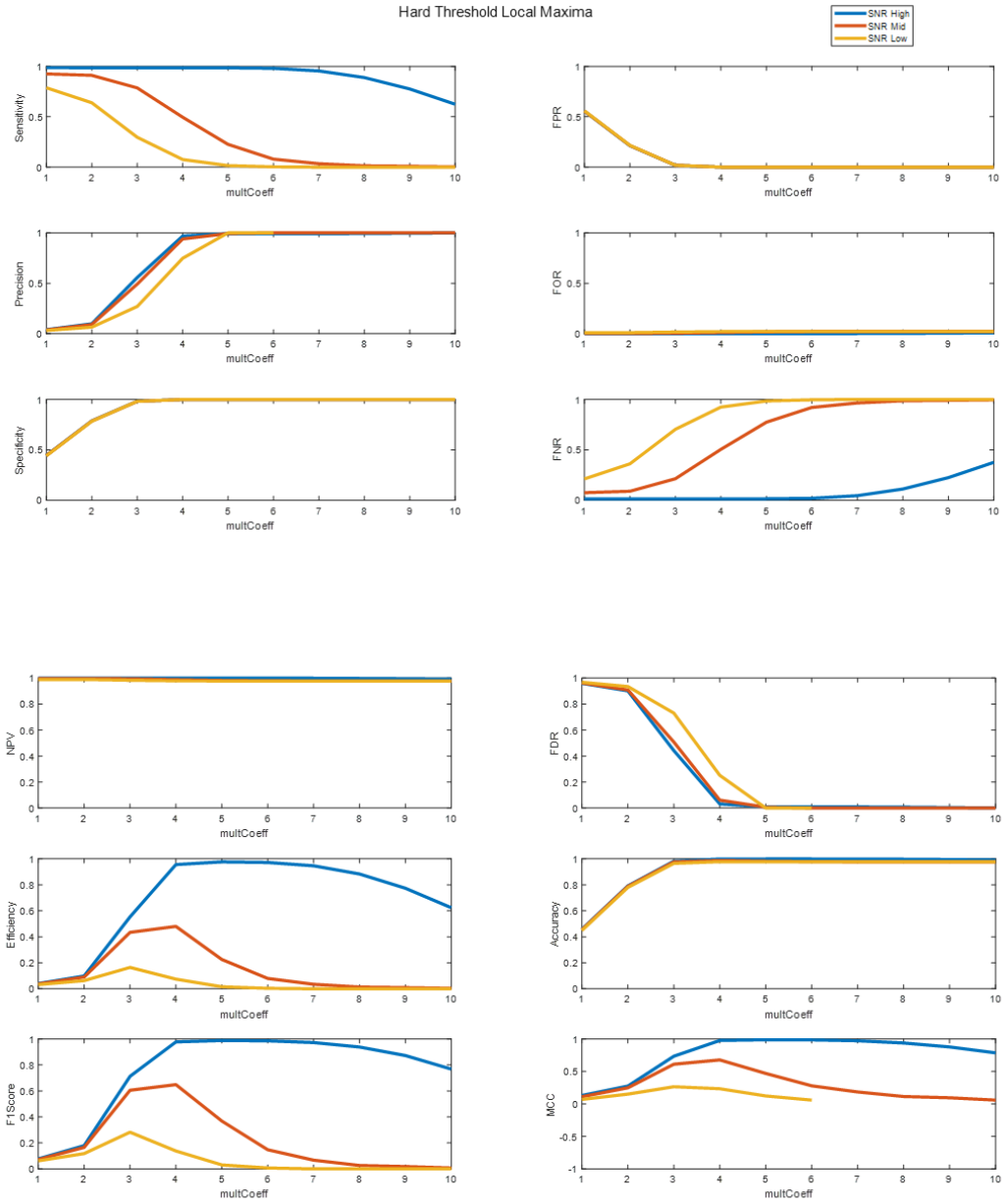


Fig.6.5 Rappresentazione grafica per tutti gli indici di HTLM su MU. In blu sono rappresentati i risultati relativi al livello di SNR maggiore (SNR High), in rosso al livello intermedio (SNR Mid) e in giallo al livello più basso (SNR Low).

6.4.3 Tempo computazionale

È stato calcolato il tempo computazionale (Fig.6.6). SU e MU sono comparabili tra loro, il tempo più lungo è raggiunto da SU con 7.5 s. Questo algoritmo è due ordini di grandezza più lento di HT, non è quindi particolarmente veloce

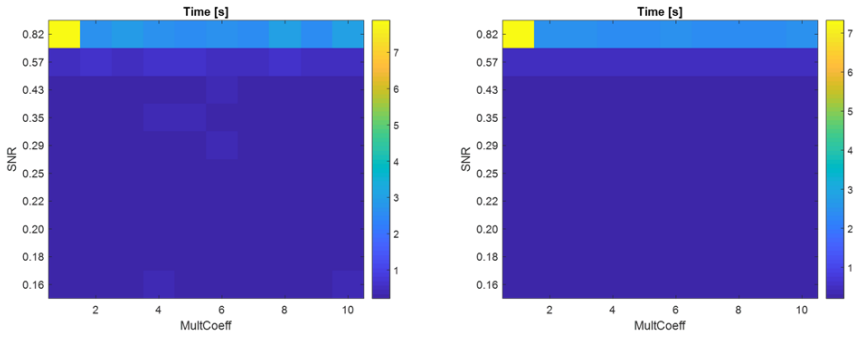
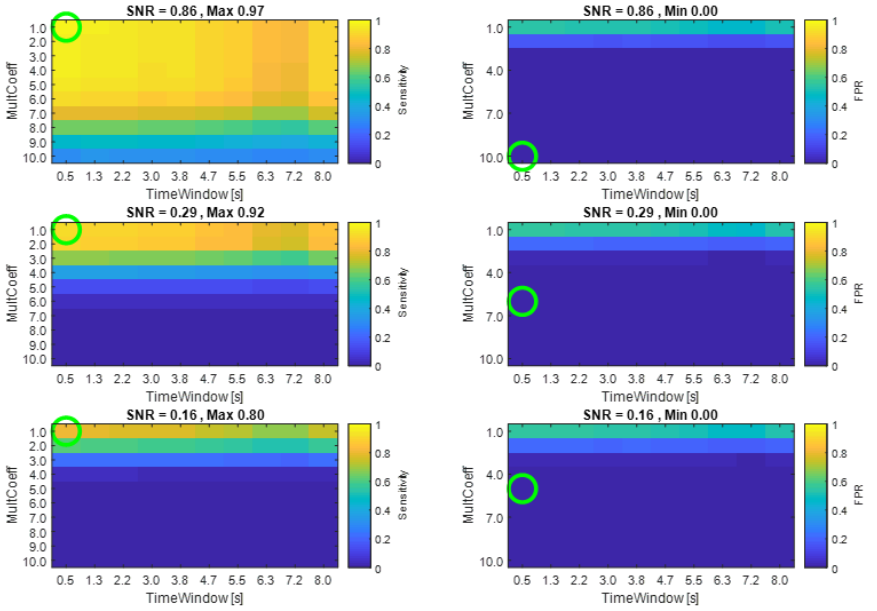


Fig.6.6 Tempi computazionali di HTLM. Sinistra, tempo per il SU: il tempo massimo è raggiunto per il livello di SNR più alto e il MC più basso. Destra, tempo per MU: analogamente, il tempo massimo è raggiunto con l'SNR più alto e MC più basso. Anche qui il primo valore alto iniziale è probabilmente dovuto a un problema di inizializzazione iniziale, da non considerarsi quindi valido.

6.5 ATLM

In questo algoritmo i parametri che variano sono due: MultCoeff, che come al solito moltiplica la soglia, e TimeWindow (TW) che definisce l'ampiezza della finestra su cui viene calcolata la deviazione standard del segnale. MC varia dieci volte tra 1 e 10, TW varia 10 volte nell'intervallo 0.5 - 8 s, utilizzando il comando *linspace* di Matlab, assumendo i seguenti valori: 0.5, 1.3, 2.1, 3, 3.8, 4.6, 5.5, 6.3, 7.1 e 8. Ovviamente questi valori vengono poi trasformati in campioni. Essendo presenti due parametri, i grafici avranno un aspetto diverso da quello visto precedentemente. Sono riportati sempre gli stessi tre livelli di SNR dei precedenti algoritmi, tutti i MC e tutte le TW.

a)



b)

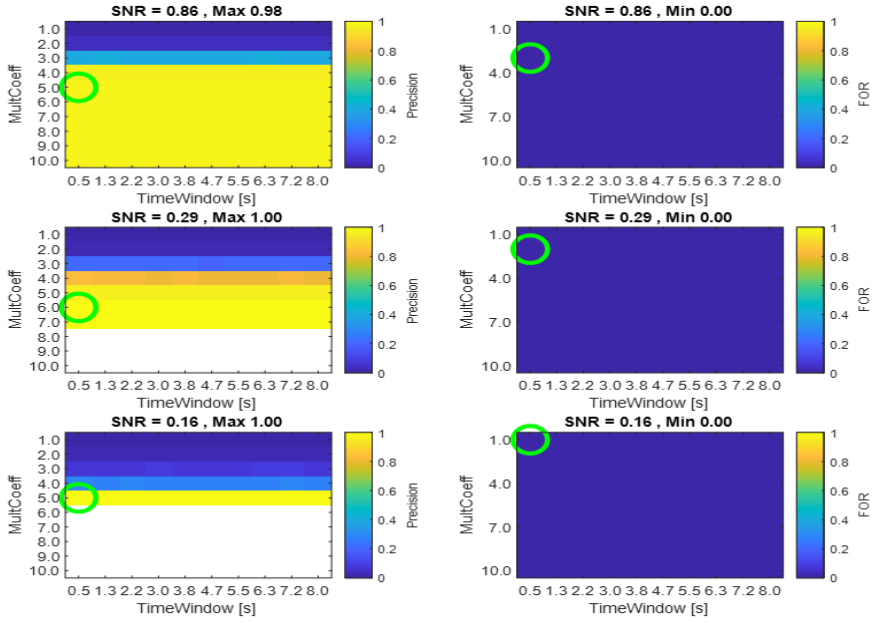
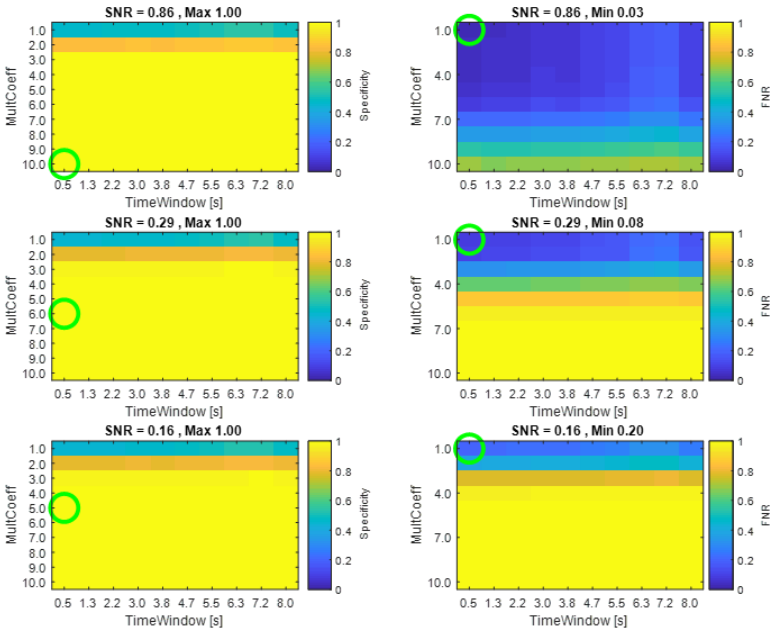


Fig.6.7. Sensitivity, FPR, Precision e FOR per ATLM SU per tre livelli di SNR. a). Sensitivity (sinistra) e FPR (destra). I cerchi in verdi indicano dove si trova la massima sensitivity e la minima FPR (in quanto è da minimizzare). Sopra ad ogni grafico è indicato il massimo per la sensitivity e il minimo per FPR raggiunti. La prima si attesta sempre attorno a valori piuttosto elevati, diminuendo all'aumentare dell'SNR. La FPR invece ha valore 0 per la maggior parte dei parametri b) Precision (sinistra) e FOR (destra). I cerchi in verdi indicano dove si trova la massima precision e la minima FOR (in quanto è da minimizzare). Sopra ad ogni grafico è indicato il massimo per la precision e il minimo per FOR raggiunti. Il bianco nei grafici di sinistra indica la presenza dei NaN, significa che l'algoritmo dopo aver raggiunto il

valore massimo, non trova più spike. La formulazione del FOR con i TN al denominatore fa si che questo indice tenda sempre a 0

.

a)



b)

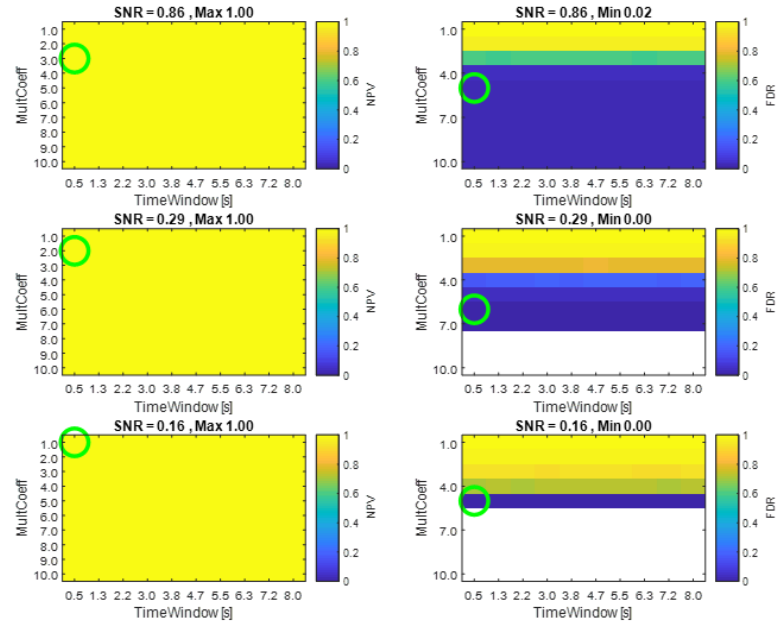
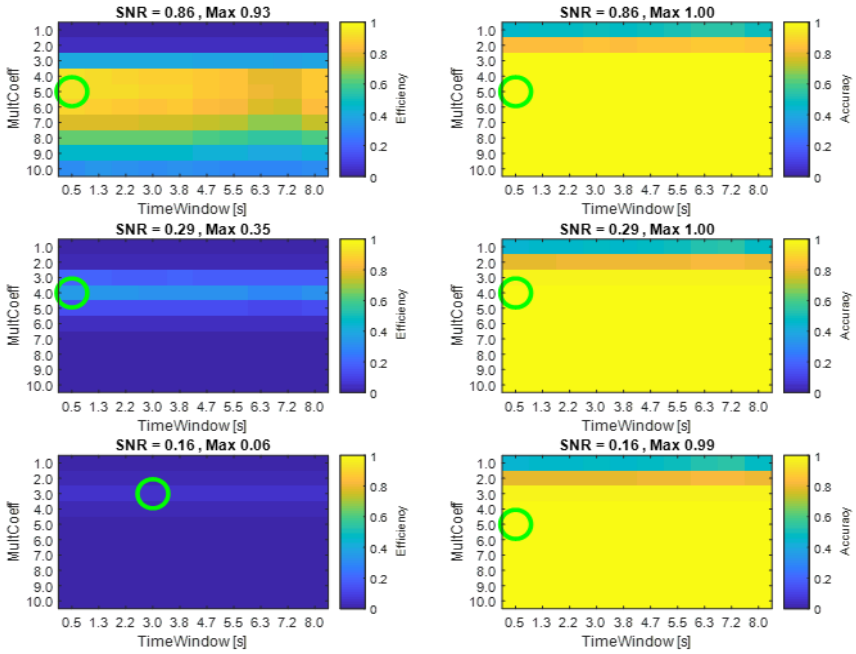


Fig.6.8. Specificity, FNR, NPV e FDR per ATLM SU per tre livelli di SNR. a) Specificity (sinistra) e FNR (destra). I cerchi in verdi indicano dove si trova la massima specificity e la minima FNR (in quanto è da minimizzare). Sopra ad ogni grafico è indicato il massimo per la specificity e il minimo per FNR raggiunti. La prima si attesta al valore ideale per la maggior parte dei parametri. È interessante notare come dove c'è il massimo della specificity (caso migliore), c'è anche il massimo di FNR (caso peggiore) e viceversa. b) NPV (sinistra) e FDR (destra). I

cerchi in verdi indicano dove si trova la massima NPV e la minima FDR (in quanto è da minimizzare). Sopra ad ogni grafico è indicato il massimo per la NPV e il minimo per FDR raggiunti. Il bianco nei grafici di destra indica la presenza dei NaN, significa che l'algoritmo, dopo aver raggiunto il valore minimo di FDR, non trova più spike. La formulazione di NPV con i TN molto maggiori dei FN, fa sì che questo valore tenda sempre al valore ideale 1

a)



b)

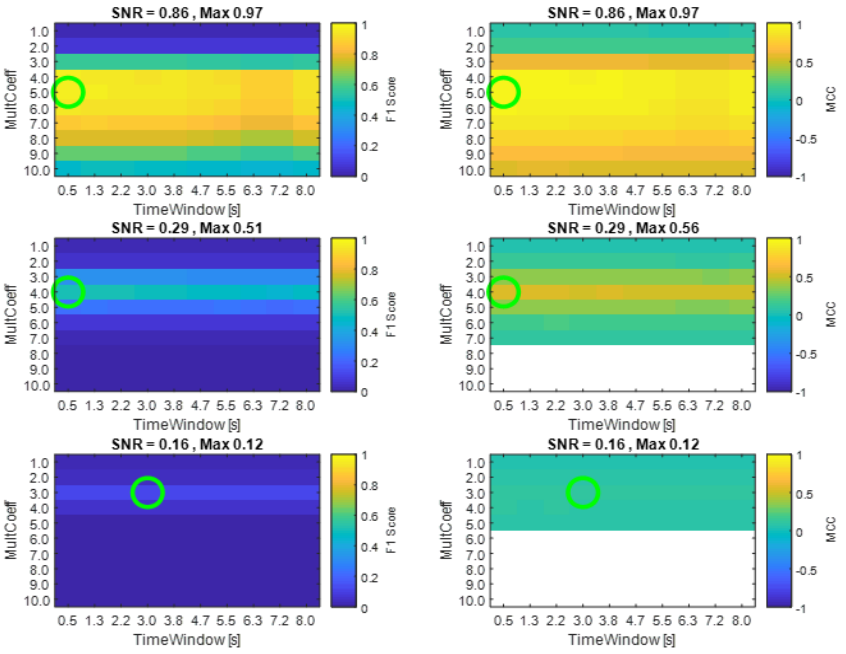


Fig.6.9. Efficiency, Accuracy, F1S e MCC per ATLM SU per tre livelli di SNR. a) Efficiency (sinistra) e accuracy (destra). I cerchi in verdi indicano dove si trovano le massime efficiency e accuracy. Sopra ad ogni grafico ne sono indicati i valori

raggiunti. Entrambe diminuiscono all'aumentare dell'SNR, accuracy si attesta al valore ideale per la maggior parte dei parametri. b) F1S (sinistra) e MCC (destra). I cerchi in verdi indicano dove si trovano i massimi valori F1S e MCC, riportati sopra ai grafici. Il bianco indica la presenza dei NaN, significa che qualche termine al denominatore di MCC si annulla, mandando il calcolo di MCC a NaN. Si ricorda inoltre che MCC va da -1 a 1, per questo l'asse dei valori è diverso.

6.5.1 Single Unit

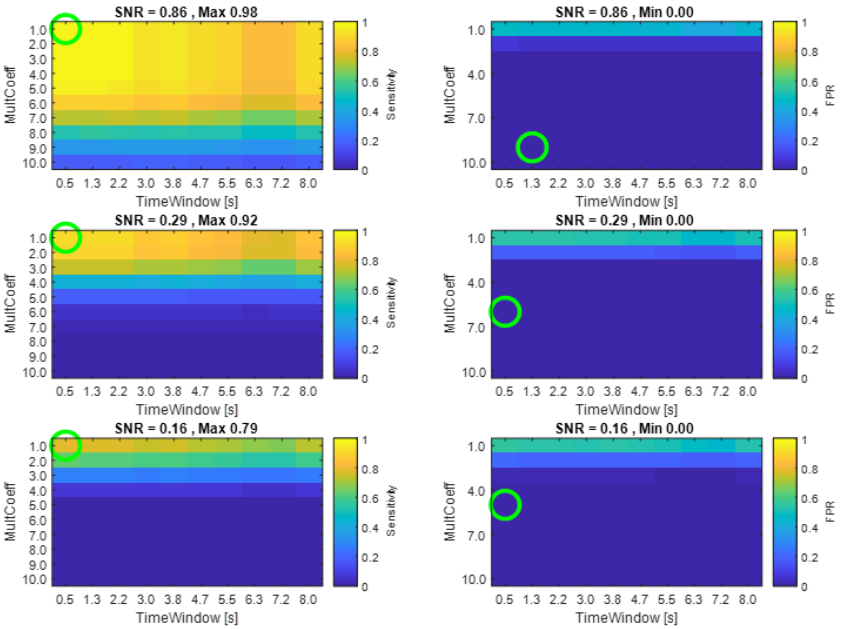
In Fig.6.7. si possono vedere sensitivity, FPR, precision e FOR, in Fig.6.8 specificity, FNR, NPV e FDR, in Fig6.9. accuracy, efficiency, F1S e MCC.

La TimeWindow (TW) migliore per le analisi sembra essere la prima, 0.5 s, il MultCoeff (MC) migliore varia maggiormente ma per lo più è compreso nell'intervallo 3-5. Per continuità, riporterò in tabella 6.6 i valori come nei precedenti due algoritmi fissando TW a 0.5 e MC a 4.

SU	TPR	TNR	PPV	FNR	NPV	FPR	FDR	FOR	CSI	ACC	F1	MCC
SNR 0.16	0.03	0.99	0.29	0.96	0.99	0.0005	0.70	0.006	0.03	0.99	0.05	0.09
SNR 0.29	0.37	0.99	0.84	0.62	0.99	0.0004	0.16	0.004	0.34	0.99	0.51	0.55
SNR 0.86	0.96	0.99	0.95	0.03	0.99	0.0002	0.04	0.0002	0.92	0.99	0.96	0.96

Tab.6.6 Valori raggiunti per ciascun indice da ATLM con MC=4 e TW=0.5 per tre livelli di SNR nel SU. I valori degli indici tendono a migliorare all'aumentare dell'SNR ma così non è per TNR, NPV e ACC (accuracy) che mantengono lo stesso valore. Si ricorda che CSI (efficiency) F1 e MCC sono gli indici di prestazione generale e rappresentano la chiave di lettura delle performance dell'algoritmo. HTLM al suo meglio, nel caso migliore, raggiunge un ***CSI 0.92, F1 0.96 e MCC 0.96****. Nel caso peggiore, rispettivamente raggiunge* ***0.03, 0.05 e 0.09****. Nel caso peggiore raggiunge gli stessi valori di HTLM, mentre nel caso migliore gli è superiore.*

a)



b)

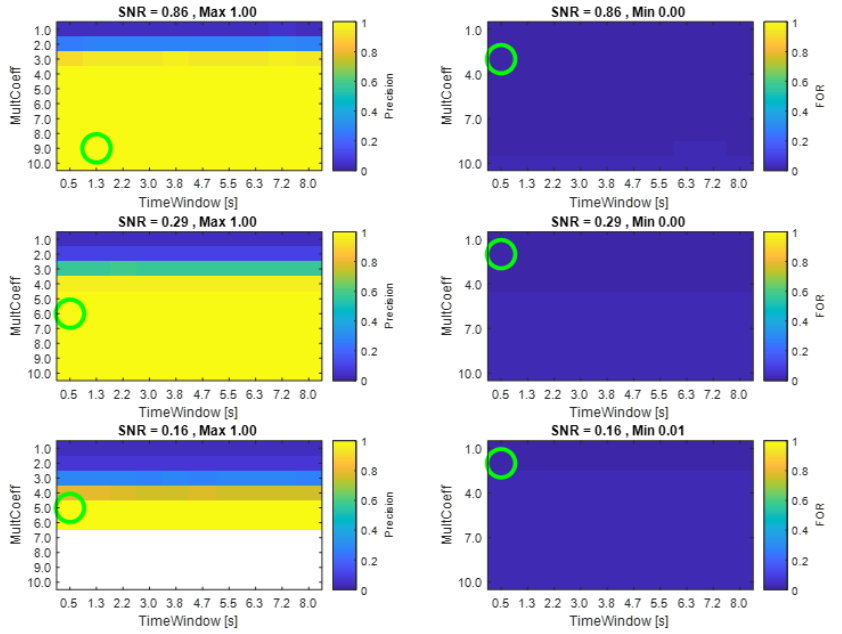
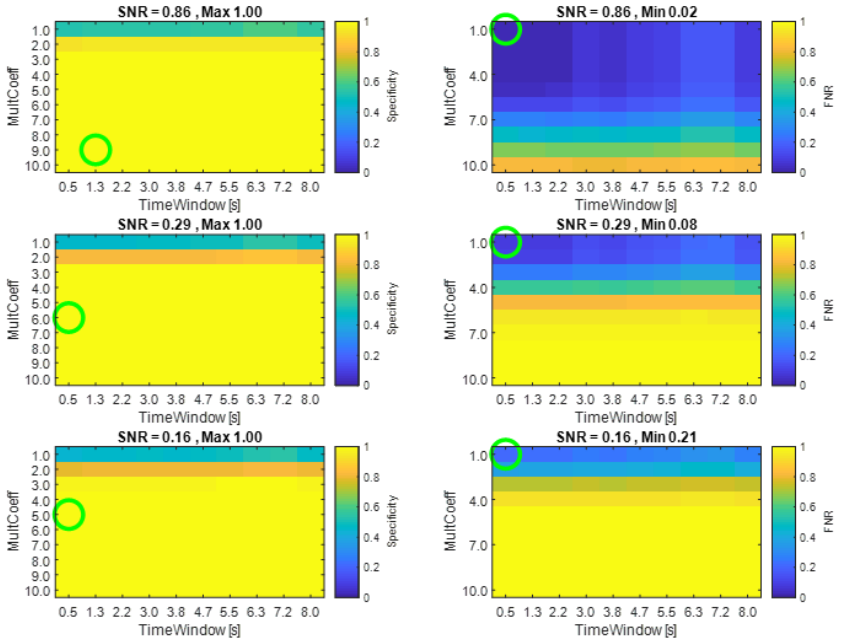


Fig. 6.10. Sensitivity, FPR, Precision e FOR per ATLM MU per tre livelli di SNR. a). Sensitivity (sinistra) e FPR (destra). I cerchi in verdi indicano dove si trova la massima sensitivity e la minima FPR (in quanto è da minimizzare). Sopra ad ogni grafico è indicato il massimo per la sensitivity e il minimo per FPR raggiunti. La prima si attesta sempre attorno a valori piuttosto elevati, diminuendo all'aumentare dell'SNR. La FPR invece ha valore 0 per la maggior parte dei parametri b) Precision (sinistra) e FOR (destra). I cerchi in verdi indicano dove si trova la massima precision e la minima FOR (in quanto è da minimizzare). Sopra ad ogni grafico è indicato il massimo per la precision e il minimo per FOR raggiunti. Il bianco nei grafici di sinistra indica la presenza dei NaN, significa che l'algoritmo dopo aver raggiunto il

valore massimo, non trova più spike. La formulazione del FOR con i TN al denominatore fa si che questo indice tenda sempre a 0.

a)



b)

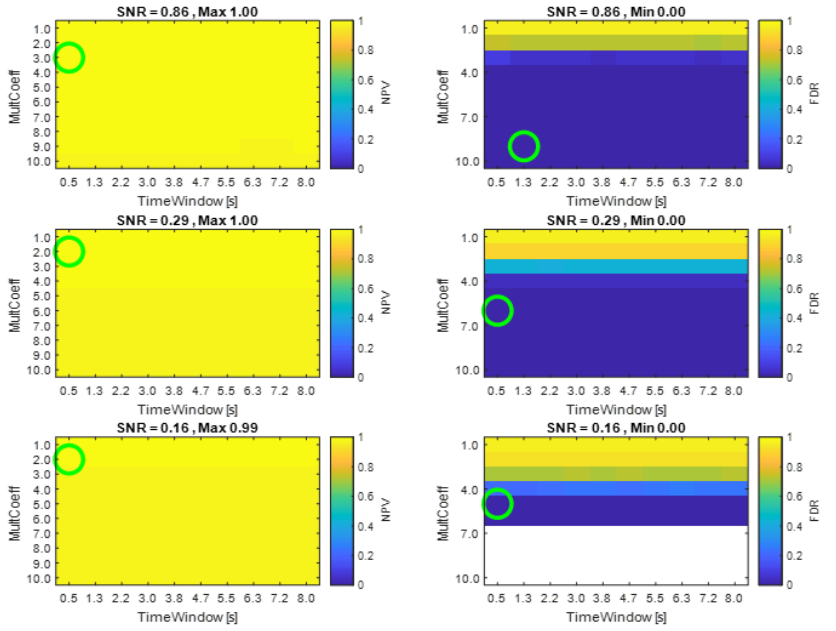
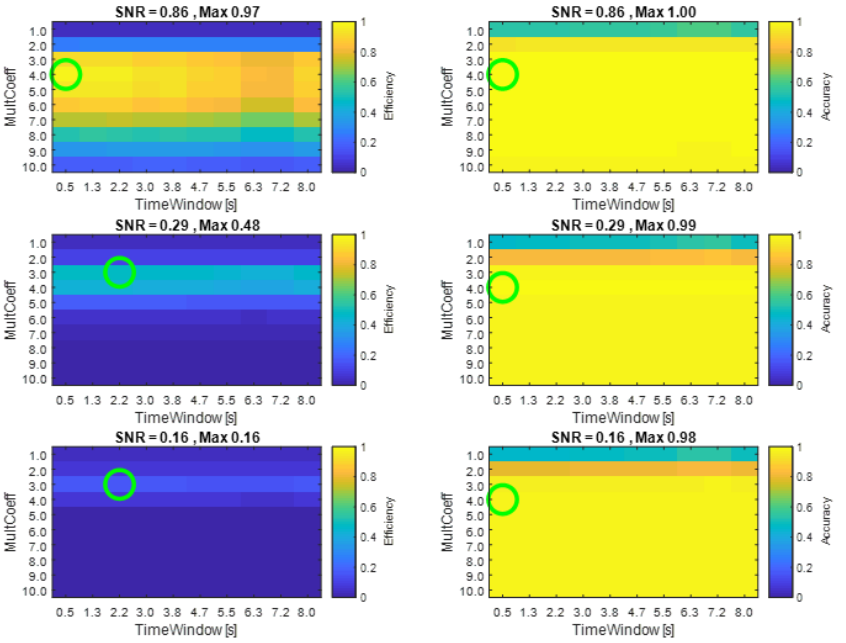


Fig.6.11 Specificity, FNR, NPV e FDR per ATLM MU per tre livelli di SNR. a) Specificity (sinistra) e FNR (destra). I cerchi in verdi indicano dove si trova la massima specificity e la minima FNR (in quanto è da minimizzare). Sopra ad ogni grafico è indicato il massimo per la specificity e il minimo per FNR raggiunti. La prima si attesta al valore ideale per la maggior parte dei parametri. È interessante notare come dove c'è il massimo della specificity (caso migliore), c'è anche il massimo di FNR (caso peggiore) e viceversa. b) NPV (sinistra) e FDR (destra). I cerchi in verdi indicano dove si trova la massima NPV e la minima FDR (in quanto è

da minimizzare). Sopra ad ogni grafico è indicato il massimo per la NPV e il minimo per FDR raggiunti. Il bianco nei grafici di destra indica la presenza dei NaN, significa che l'algoritmo, dopo aver raggiunto il valore minimo di FDR, non trova più spike. La formulazione di NPV, con i TN molto maggiori dei FN, fa sì che questo valore tenda sempre al valore ideale 1.

.

a)



b)

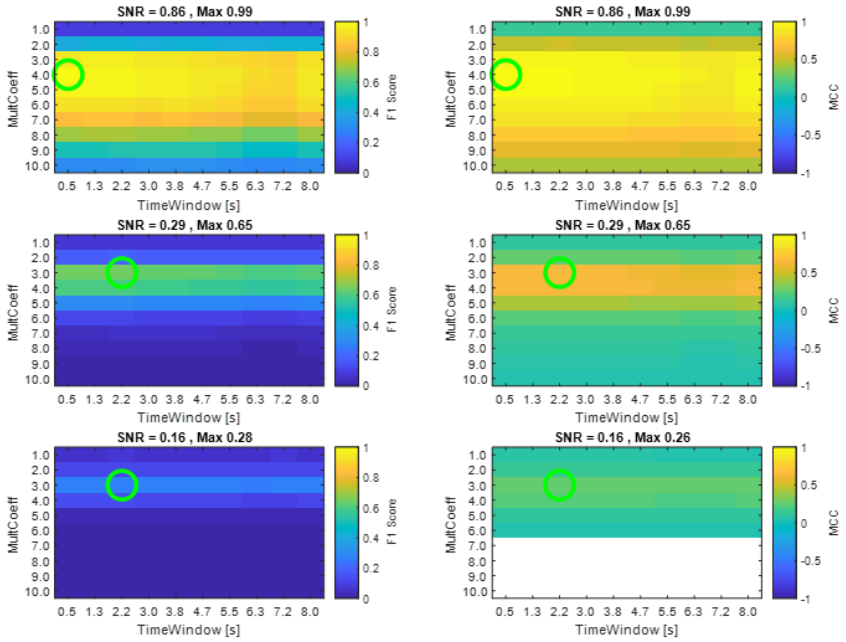


Fig.6.12. Efficiency, Accuracy, F1S e MCC per ATLM MU per tre livelli di SNR. a) Efficiency (sinistra) e accuracy (destra). I cerchi in verdi indicano dove si trovano le massime efficiency e accuracy. Sopra ad ogni grafico ne sono indicati i valori raggiunti. Entrambe diminuiscono all'aumentare dell'SNR, accuracy si attesta al

valore ideale per la maggior parte dei parametri. b) F1S (sinistra) e MCC (destra). I cerchi in verdi indicano dove si trovano i massimi valori F1S e MCC, riportati sopra ai grafici. Il bianco indica la presenza dei NaN, significa che qualche termine al denominatore di MCC si annulla, mandando il calcolo di MCC a NaN. Si ricorda inoltre che MCC va da -1 a 1, per questo l'asse dei valori è diverso.

6.5.2 Multi Unit

In Fig.6.10 si possono vedere sensitivity, FPR, precision e FOR, in Fig.6.11 specificity, FNR, NPV e FDR, in Fig.6.12 accuracy, efficiency, F1S e MCC.

La TW migliore continua ad essere la prima, 0.5 s, MC migliore varia maggiormente, si è scelto 4 come prima per un confronto più immediato, i valori sono mostrati in Tab.6.7.

MU	TPR	TNR	PPV	FNR	NPV	FPR	FDR	FOR	CSI	ACC	F1	MCC
SNR 0.16	0.07	0.99	0.78	0.92	0.97	0.0004	0.21	0.02	0.07	0.97	0.13	0.23
SNR 0.29	0.45	0.99	0.96	0.56	0.98	0.0003	0.03	0.0132	0.42	0.98	0.60	0.64
SNR 0.86	0.98	0.99	0.99	0.018	0.99	0.0002	0.009	0.0004	0.97	0.99	0.98	0.98

Tab.6.7. Valori raggiunti per ciascun indice da ATLM con MC=4 e TW=0.5 per tre livelli di SNR nel MU. I valori degli indici tendono a migliorare all'aumentare dell'SNR anche in questo caso, e ugualmente TNR, NPV e ACC mantengono lo stesso valore. ATLM al suo meglio, nel caso migliore, raggiunge un CSI 0.97, F1 0.98 e MCC 0.98. Nel caso peggiore, rispettivamente raggiunge 0.07, 0.13 e 0.23. Nel caso peggiore sono gli stessi valori raggiunti da HTLM, nel caso migliore lo supera.

6.5.3 Tempo computazionale

È stato calcolato il tempo computazionale (Fig.6.13) sempre utilizzando il parametro TW=0.5. SU e MU sono dello stesso ordine di grandezza, il tempo più lungo è raggiunto da MU che supera i 7 s. I tempi di questo algoritmo sono comparabile con HTLM, quindi non particolarmente veloce.

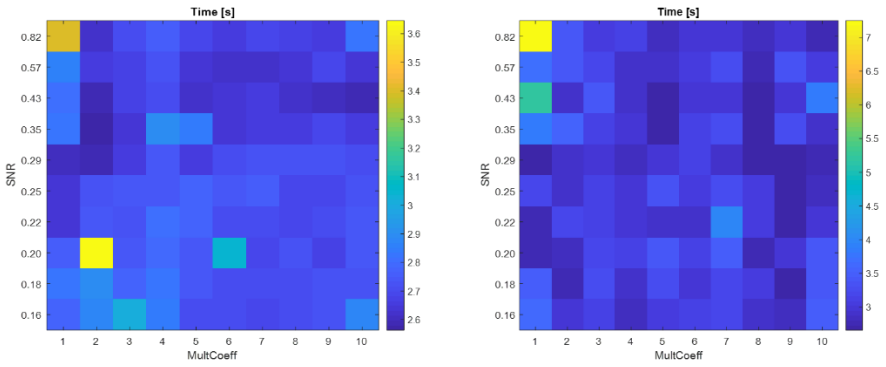
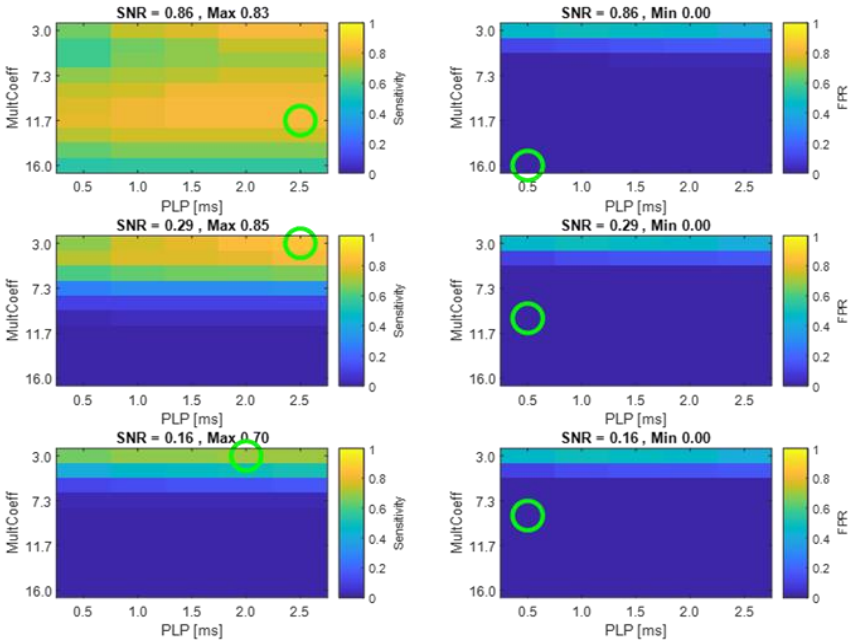


Fig.6.13 Tempi computazionali di ATLM con TW=0.5. Sinistra, tempo per il SU: il tempo massimo è raggiunto per il livello di SNR 0.20 più alto e il MC=2. Destra, tempo per MU, il tempo massimo è raggiunto con l'SNR più alto e MC più basso.

6.6 PTSD

Anche in questo algoritmo i parametri che variano sono due: MultCoeff, che come al solito moltiplica la soglia, e PLP che definisce l'ampiezza della finestra in cui viene ricercato il 2° RMM. MC varia dieci volte tra 3 e 16, sempre utilizzando il comando *linspace*, i valori cambiano in questo caso perché PTSD utilizza una soglia differenziale e non unipolare come nei casi precedenti. I valori che assume sono: 3, 4.4, 5.8, 7.3, 8.7, 10.2, 11.6, 13.1, 14.5, 16. PLP varia cinque volte nell'intervallo 0.5 – 2.5 ms, assumendo i seguenti valori: 0.5, 1, 1.5, 2, 2.5. Ovviamente questi valori vengono poi trasformati in campioni, sono solo cinque e non dieci a causa della grossa quantità di memoria che necessita questo algoritmo. Essendo presenti due parametri, i grafici avranno lo stesso aspetto di quelli di ATLM. Sono riportati sempre gli stessi tre livelli di SNR, tutti i MC e tutti i PLP.

a)



b)

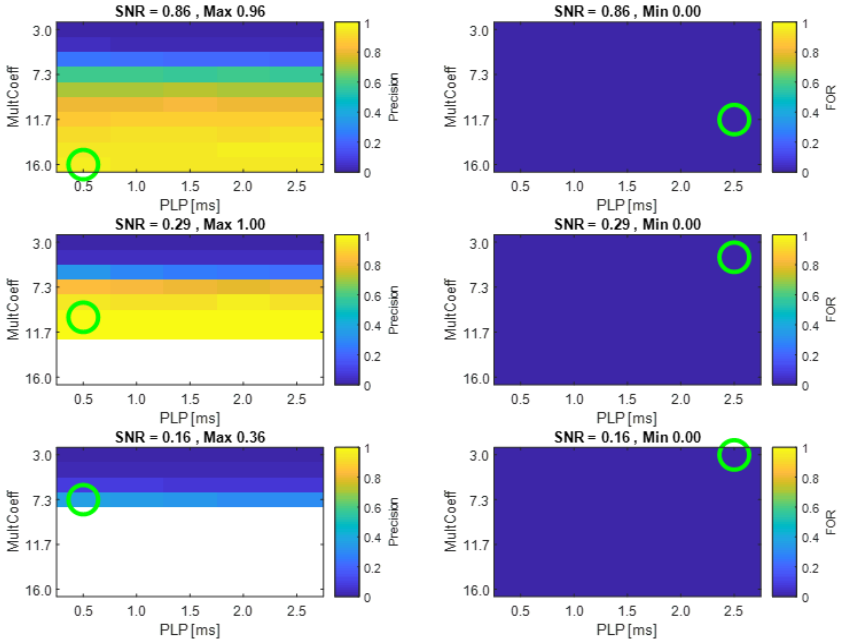
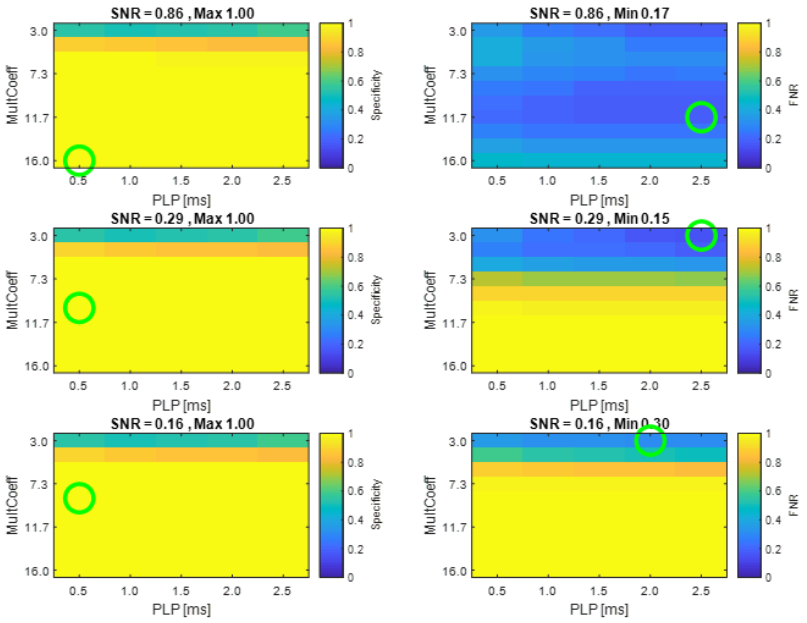


Fig.6.14 Sensitivity, FPR, Precision e FOR per PTSD SU per tre livelli di SNR. a). Sensitivity (sinistra) e FPR (destra). I cerchi in verdi indicano dove si trova la massima sensitivity e la minima FPR (in quanto è da minimizzare). Sopra ad ogni grafico è indicato il massimo per la sensitivity e il minimo per FPR raggiunti. La prima si attesta sempre attorno a valori piuttosto elevati, diminuendo all'aumentare dell'SNR. La FPR invece ha valore 0 per la maggior parte dei parametri b) Precision (sinistra) e FOR (destra). I cerchi in verdi indicano dove si trova la massima precision e la minima FOR (in quanto è da minimizzare). Sopra ad ogni grafico è indicato il massimo per la precision e il minimo per FOR raggiunti. Il bianco nei grafici di sinistra indica la presenza dei NaN, significa che l'algoritmo dopo aver raggiunto il

valore massimo, non trova più spike. La formulazione del FOR con i TN al denominatore fa si che questo indice tenda sempre a 0.

a)



b)

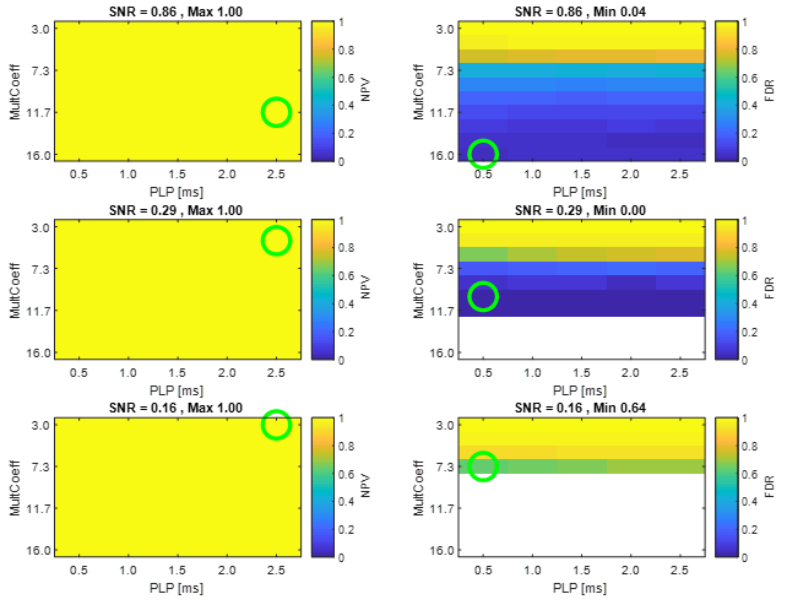
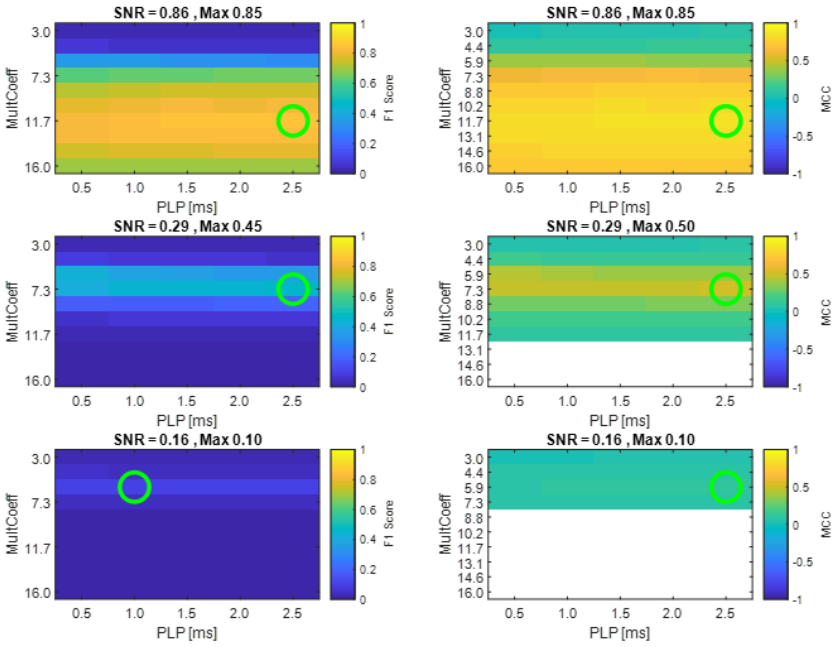


Fig.15 Specificity, FNR, NPV e FDR per PTSD SU per tre livelli di SNR. a) Specificity (sinistra) e FNR (destra). I cerchi in verdi indicano dove si trova la massima specificity e la minima FNR (in quanto è da minimizzare). Sopra ad ogni grafico è indicato il massimo per la specificity e il minimo per FNR raggiunti. La prima si attesta al valore ideale per la maggior parte dei parametri. È interessante notare come dove c'è il massimo della specificity (caso migliore), c'è anche il massimo di FNR (caso peggiore) e viceversa. b) NPV (sinistra) e FDR (destra). I cerchi in verdi indicano dove si trova la massima NPV e la minima FDR (in quanto è da minimizzare). Sopra ad ogni grafico è indicato il massimo per la NPV e il minimo per

FDR raggiunti. Il bianco nei grafici di destra indica la presenza dei NaN, significa che l'algoritmo, dopo aver raggiunto il valore minimo di FDR, non trova più spike. La formulazione di NPV con i TN molto maggiori dei FN, fa sì che questo valore tenda sempre al valore ideale 1.

a)



b)

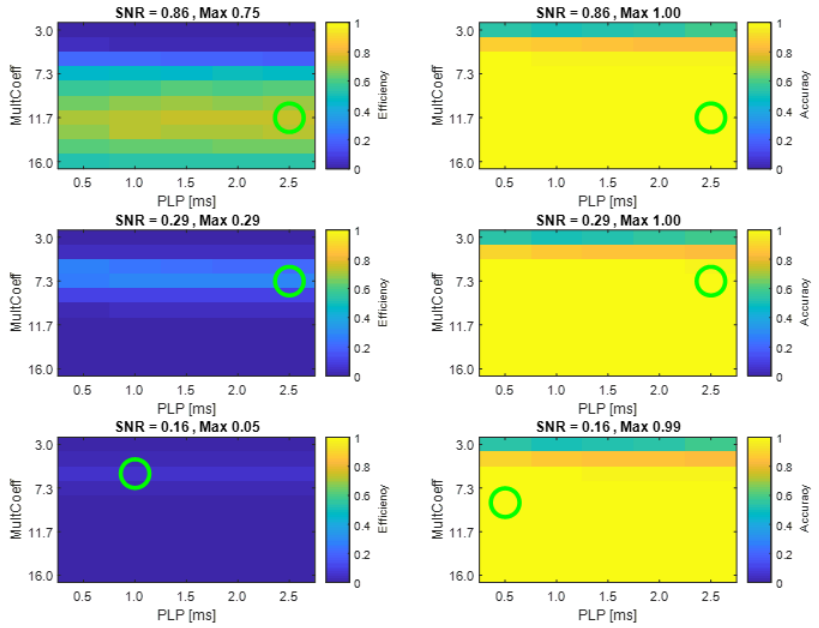


Fig.16 Efficiency, Accuracy, F1S e MCC per PTSD SU per tre livelli di SNR. a) Efficiency (sinistra) e accuracy (destra). I cerchi in verdi indicano dove si trovano le massime efficiency e accuracy. Sopra ad ogni grafico ne sono indicati i valori

raggiunti. Entrambe diminuiscono all'aumentare dell'SNR, accuracy si attesta al valore ideale per la maggior parte dei parametri. b) F1S (sinistra) e MCC (destra). I cerchi in verdi indicano dove si trovano i massimi valori F1S e MCC, riportati sopra ai grafici. Il bianco indica la presenza dei NaN, significa che qualche termine al denominatore di MCC si annulla, mandando il calcolo di MCC a NaN. Si ricorda inoltre che MCC va da -1 a 1, per questo l'asse dei valori è diverso.

6.6.1 Single Unit

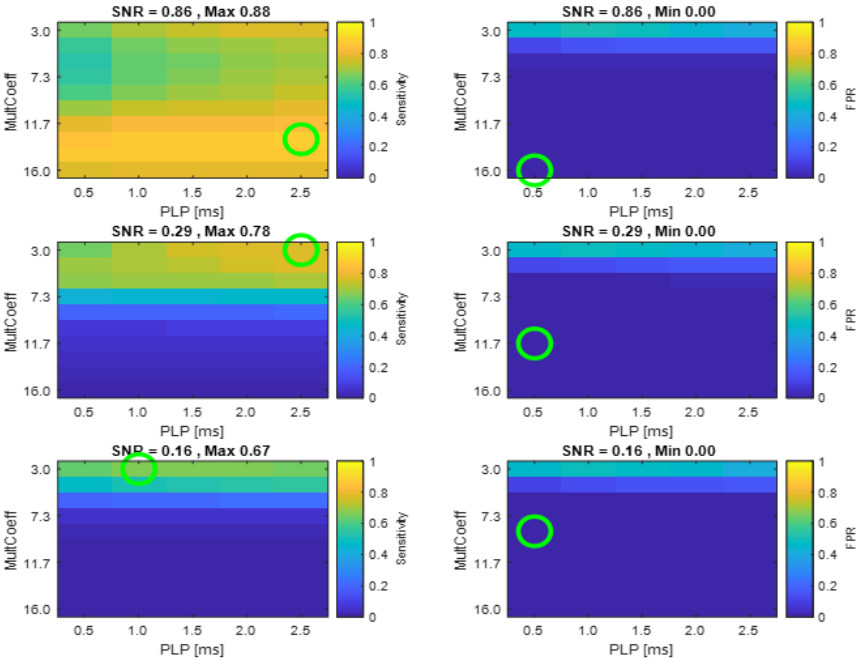
In Fig.6.14 si possono vedere sensitivity, FPR, precision e FOR, in Fig.6.15 specificity, FNR, NPV e FDR, in Fig6.15 accuracy, efficiency, F1S e MCC.

Non è facile stabilire il PLP migliore per le analisi in quanto è molto variabile. Stesso discorso me MC. Dovendo scegliere per creare una tabella (Tab.6.8) come nei paragrafi precedenti, si è optato per PLP= 2.5 e MC= 7.3.

SU	TPR	TNR	PPV	FNR	NPV	FPR	FDR	FOR	CSI	ACC	F1	MCC
SNR 0.16	0.02	0.99	0.30	0.97	0.99	0.0003	0.70	0.0063	0.02	0.99	0.04	0.08
SNR 0.29	0.3	0.99	0.79	0.68	0.99	0.0005	0.20	0.0044	0.29	0.99	0.45	0.50
SNR 0.86	0.75	0.99	0.56	0.25	0.99	0.0038	0.43	0.0016	0.47	0.99	0.64	0.64

Tab.6.8 Valori raggiunti per ciascun indice da PTSD con MC=7.3 e PLP=2.5 per tre livelli di SNR nel SU. I valori degli indici tendono a migliorare all'aumentare dell'SNR ma così non è per TNR, NPV e ACC (accuracy) che mantengono lo stesso valore. Si ricorda che CSI (efficiency) F1 e MCC sono gli indici di prestazione generale e rappresentano la chiave di lettura delle performance dell'algoritmo. HTLM al suo meglio, nel caso migliore, raggiunge un ***CSI 0.47, F1 0.64 e MCC 0.64****. Nel caso peggiore, rispettivamente raggiunge* ***0.02, 0.04 e 0.08.***

a)



b)

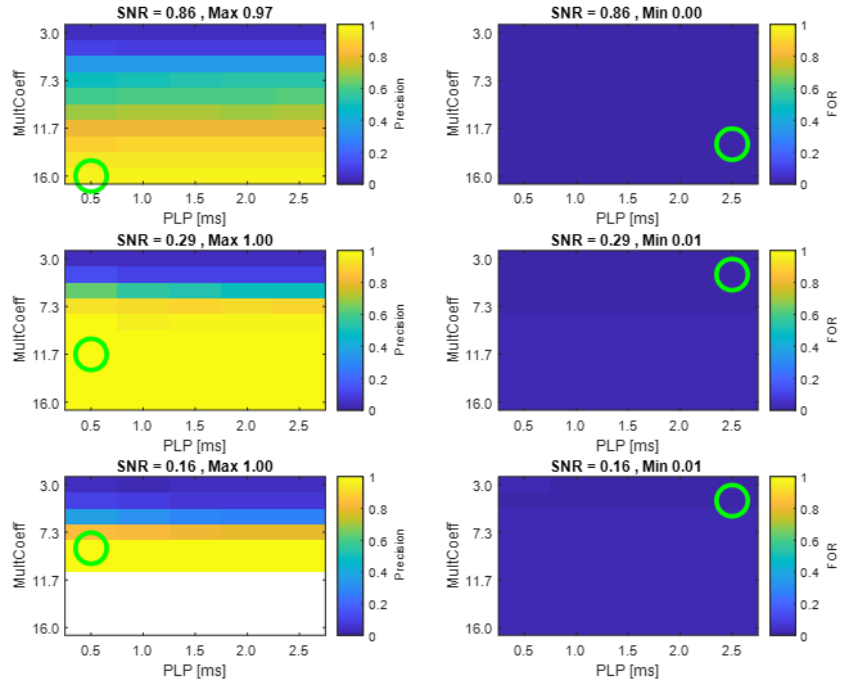
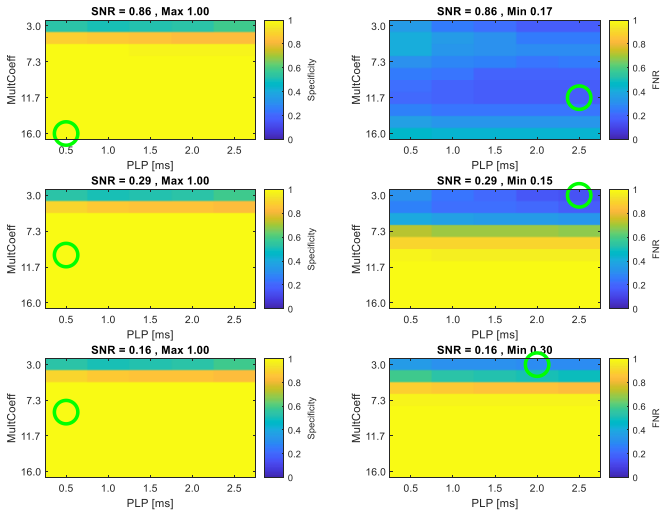


Fig.6.14. Sensitivity, FPR, Precsion e FOR per PTSD SU per tre livelli di SNR. a). Sensitivity (sinistra) e FPR (destra). I cerchi in verdi indicano dove si trova la massima sensitivity e la minima FPR (in quanto è da minimizzare). Sopra ad ogni grafico è indicato il massimo per la sensitivity e il minimo per FPR raggiunti. La prima si attesta sempre attorno a valori piuttosto elevati, diminuendo all'aumentare dell'SNR. La FPR invece ha valore 0 per la maggior parte dei parametr) Precision (sinistra) e FOR (destra). I cerchi in verdi indicano dove si trova la massima precision e la minima FOR (in quanto è da minimizzare). Sopra ad ogni grafico è indicato il massimo per la precision e il minimo per FOR raggiunti. Il bianco nei grafici di

sinistra indica la presenza dei NaN, significa che l'algoritmo dopo aver raggiunto il valore massimo di precision, non trova più spike. La formulazione del FOR con i TN al denominatore fa si che questo indice tenda sempre a 0

a)



b)

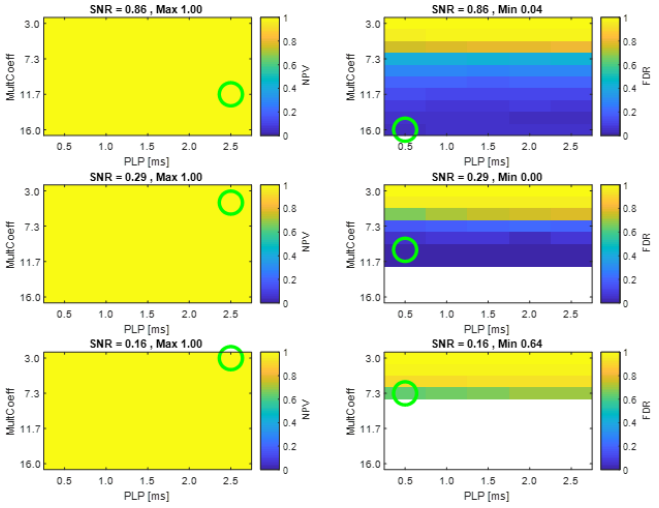
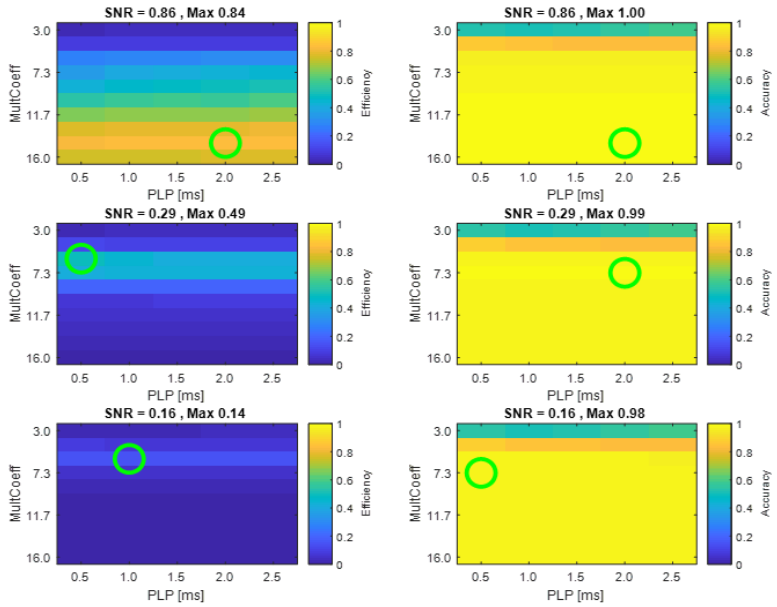


Fig.6.15 Specificity, FNR, NPV e FDR per PTSD SU per tre livelli di SNR. a) Specificity (sinistra) e FNR (destra). I cerchi in verdi indicano dove si trova la massima specificity e la minima FNR (in quanto è da minimizzare). Sopra ad ogni grafico è indicato il massimo per la specificity e il minimo per FNR raggiunti. La prima si attesta al valore ideale per la maggior parte dei parametri. È interessante notare come dove c'è il massimo della specificity (caso migliore), c'è anche il massimo di FNR (caso peggiore) e viceversa. b) NPV (sinistra) e FDR (destra). I

cerchi in verde indicano dove si trova la massima NPV e la minima FDR (in quanto è da minimizzare). Sopra ad ogni grafico è indicato il massimo per la NPV e il minimo per FDR raggiunti. Il bianco nei grafici di destra indica la presenza dei NaN, significa che l'algoritmo, dopo aver raggiunto il valore minimo di FDR, non trova più spike. La formulazione di NPV con i TN molto maggiori dei FN, fa sì che questo valore tenda sempre al valore ideale 1.

a)



b)

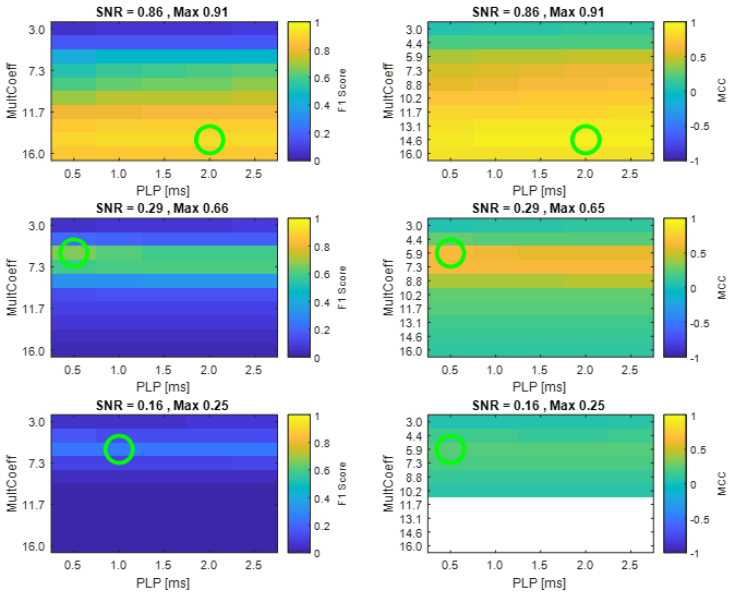


Fig.6.16. Efficiency, Accuracy, F1S e MCC per PTSD SU per tre livelli di SNR. a) Efficiency (sinistra) e accuracy (destra). I cerchi in verdi indicano dove si trovano le massime efficiency e accuracy. Sopra ad ogni grafico ne sono indicati i valori

raggiunti. Entrambe diminuiscono all'aumentare dell'SNR, accuracy si attesta al valore ideale per la maggior parte dei parametri. b) F1S (sinistra) e MCC (destra). I cerchi in verdi indicano dove si trovano i massimi valori F1S e MCC, riportati sopra ai grafici. Il bianco indica la presenza dei NaN, significa che qualche termine al denominatore di MCC si annulla, mandando il calcolo di MCC a NaN. Si ricorda inoltre che MCC va da -1 a 1, per questo l'asse dei valori è diverso.

6.6.2 Multi Unit

In Fig.6.17 si possono vedere sensitivity, FPR, precision e FOR, in Fig.6.18 specificity, FNR, NPV e FDR, in Fig.6.19 accuracy, efficiency, F1S e MCC.

Si è scelto di mantenere PLP = 2.5 e MC = 7.3 per il confronto immediato tra Tab.6.9 e Tab.6.8

MU	TPR	TNR	PPV	FNR	NPV	FPR	FDR	FOR	CSI	ACC	F1	MCC
SNR 0.16	0.05	0.99	0.79	0.94	0.97	0.0003	0.20	0.0217	0.05	0.97	0.10	0.21
SNR 0.29	0.45	0.99	0.89	0.54	0.98	0.0013	0.10	0.0126	0.43	0.98	0.60	0.63
SNR 0.86	0.7	0.98	0.54	0.28	0.99	0.014	0.45	0.0068	0.44	0.97	0.61	0.61

Tab.6.9 Valori raggiunti per ciascun indice da PTSD con MC=7.3 e PLP=2.5 per tre livelli di SNR nel MU. I valori degli indici tendono a migliorare all'aumentare dell'SNR anche in questo caso, e ugualmente TNR, NPV e ACC mantengono lo stesso valore. PTSD al suo meglio, nel caso migliore, raggiunge un ***CSI 0.44, F1 0.61*** *e MCC 0.61. Nel caso peggiore, rispettivamente raggiunge* ***0.05, 0.1 e 0.21.***

6.6.3 Tempo computazionale

Il tempo computazionale (Fig.6.20) è stato calcolato utilizzando PLP = 2.5. SU e MU sono dello stesso ordine di grandezza, il tempo più lungo è raggiunto con i livelli di SNR più bassi del MU che supera i 5.5 s. Questo algoritmo è più veloce degli ultimi visti ma meno veloce di HT.

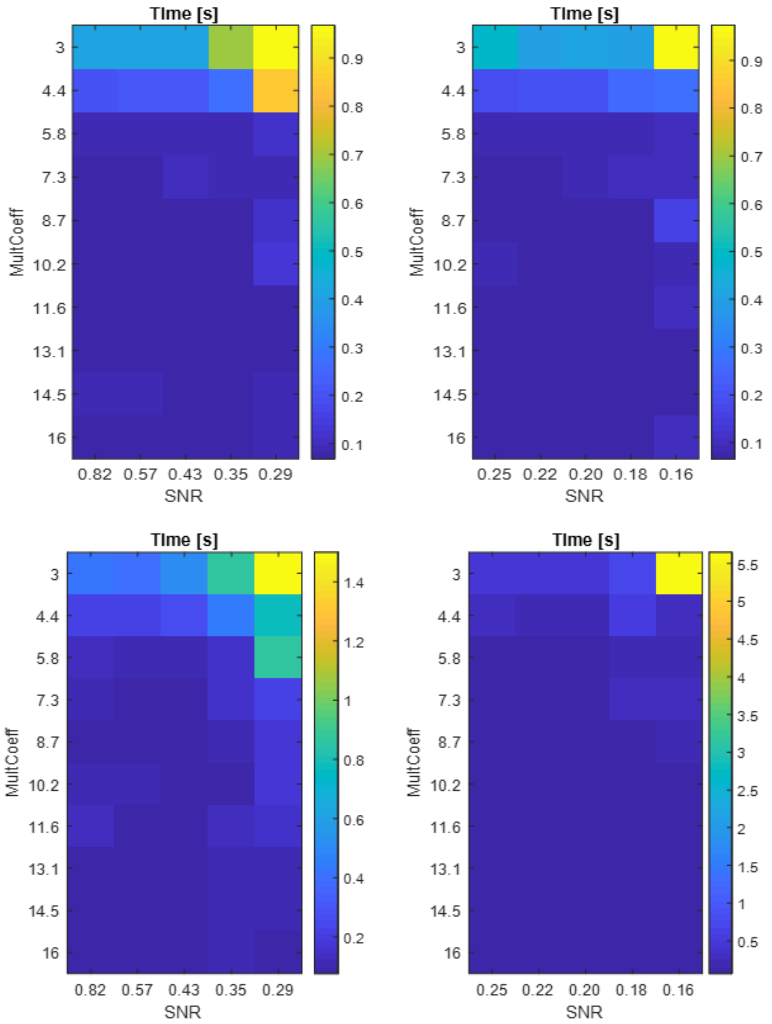


Fig.6.20 Il grafico del tempo computazionale è diviso in quattro e non in due a causa della grande quantità di memoria di cui necessita PTSD. Per risolvere il problema si è deciso di dividere in due parte i livelli di SNR, quindi da 0.82 a 0.29 e da 0.25 a 0.16. In alto è rappresentato il SU, in basso il MU. A sinistra ci sono gli SNR più bassi e a destra i più alti.

6.7 mPTSD

In questo algoritmo variano gli stessi parametri che variano in PTSD con l'unica differenza che la soglia è unipolare quindi MC è stato fatto variare nuovamente dieci volte tra 1 e 10. Essendo presenti due parametri, i grafici avranno lo stesso aspetto di quelli di ATLM e PTSD. Sono riportati sempre gli stessi tre livelli di SNR, tutti i MC e tutti i PLP.

6.7.1 Single Unit

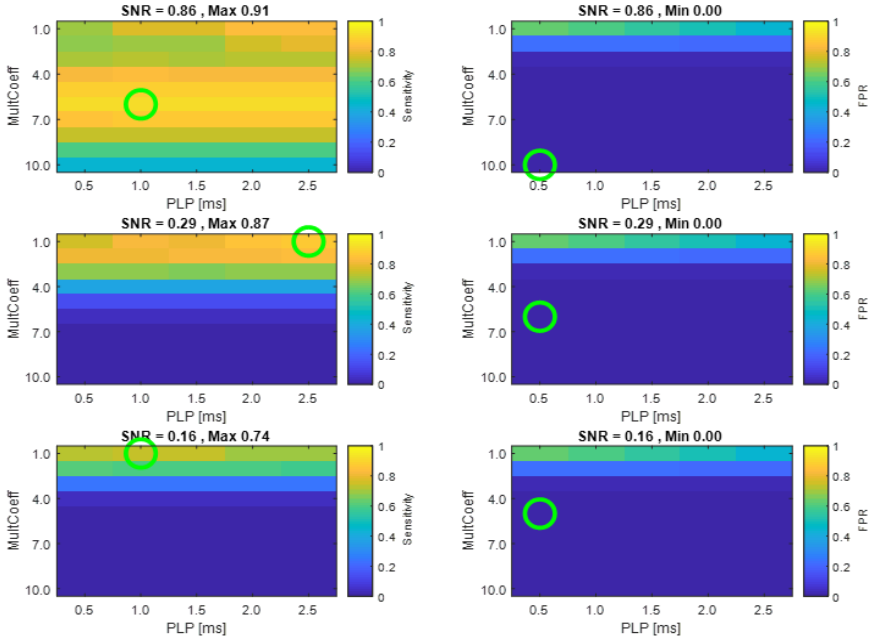
In Fig.6.21 si possono vedere sensitivity, FPR, precision e FOR, in Fig.6.22 specificity, FNR, NPV e FDR, in Fig.6.23 accuracy, efficiency, F1S e MCC.

Il PLP influisce poco sulla variazione degli indici, lo stesso non si può dire per MC. Dovendo scegliere per creare le tabelle come sopra, i valori degli indici con PLP = 1 e MC = 4 sono mostrati in Tab.6.10.

SU	TPR	TNR	PPV	FNR	NPV	FPR	FDR	FOR	CSI	ACC	F1	MCC
SNR 0.16	0.03	0.99	0.27	0.96	0.99	0.0005	0.72	0.0063	0.03	0.99	0.05	0.09
SNR 0.29	0.38	0.99	0.80	0.61	0.99	0.0004	0.19	0.004	0.35	0.99	0.52	0.55
SNR 0.86	0.81	0.99	0.73	0.18	0.99	0.0019	0.26	0.0012	0.63	0.99	0.77	0.77

Tab.6.10 Valori raggiunti per ciascun indice da mPTSD con MC=4 e PLP=1 per tre livelli di SNR nel SU. I valori degli indici tendono a migliorare all'aumentare dell'SNR ma così non è per TNR, NPV e ACC (accuracy) che mantengono lo stesso valore. Si ricorda che CSI (efficiency) F1 e MCC sono gli indici di prestazione generale e rappresentano la chiave di lettura delle performance dell'algoritmo. mPTSD al suo meglio, nel caso migliore, raggiunge un ***CSI 0.63, F1 0.77 e MCC 0.77.*** *Nel caso peggiore, rispettivamente raggiunge* ***0.03, 0.05 e 0.09****. Nel caso migliore, vengono raggiunti valori superiori a PTSD, nel caso peggiore leggermente più alti ma comparabili.*

a)



b)

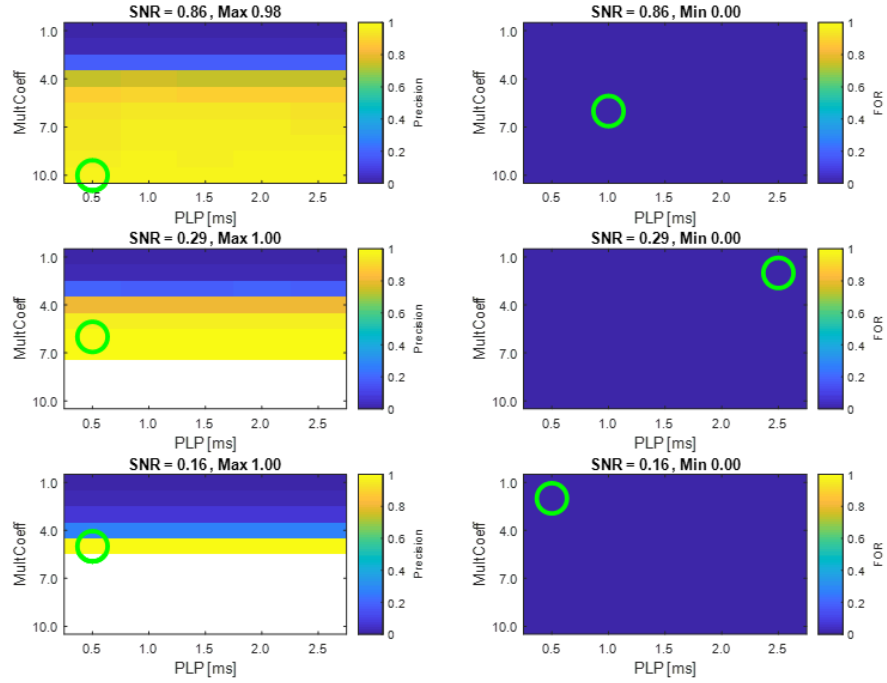
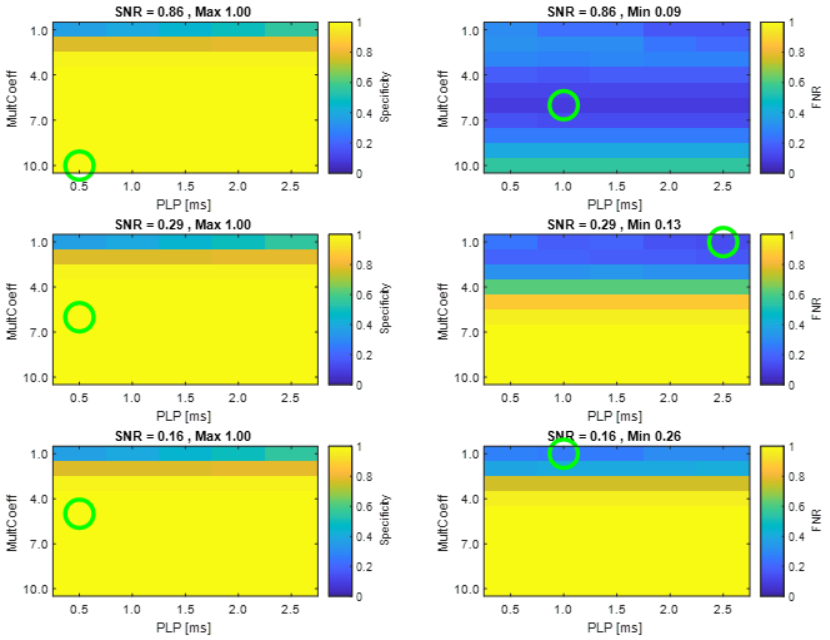


Fig.6.21 Sensitivity, FPR, Precsion e FOR per mPTSD SU per tre livelli di SNR. a). Sensitivity (sinistra) e FPR (destra). I cerchi in verdi indicano dove si trova la massima sensitivity e la minima FPR (in quanto è da minimizzare). Sopra ad ogni grafico è indicato il massimo per la sensitivity e il minimo per FPR raggiunti. La prima si attesta sempre attorno a valori piuttosto elevati, diminuendo all'aumentare dell'SNR. La FPR invece ha valore 0 per la maggior parte dei parametri b) Precision (sinistra) e FOR (destra). I cerchi in verdi indicano dove si trova la massima precision e la minima FOR (in quanto è da minimizzare). Sopra ad ogni grafico è indicato il massimo per la precision e il minimo per FOR raggiunti. Il bianco nei grafici di

sinistra indica la presenza dei NaN, significa che l'algoritmo dopo aver raggiunto il valore massimo, non trova più spike. La formulazione del FOR con i TN al denominatore fa si che questo indice tenda sempre a 0.

a)



b)

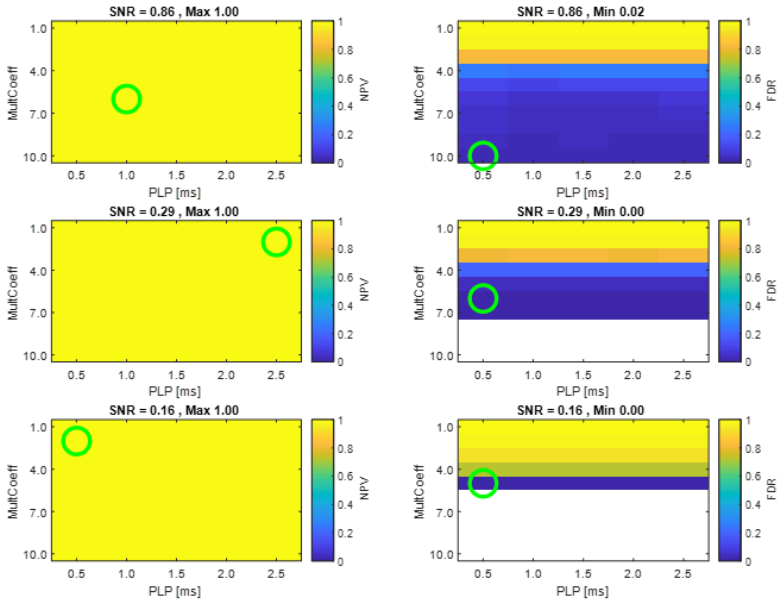


Fig.6.22 Specificity, FNR, NPV e FDR per mPTSD SU per tre livelli di SNR. a) Specificity (sinistra) e FNR (destra). I cerchi in verdi indicano dove si trova la massima specificity e la minima FNR (in quanto è da minimizzare). Sopra ad ogni grafico è indicato il massimo per la specificity e il minimo per FNR raggiunti. La prima si attesta al valore ideale per la maggior parte dei parametri. È interessante notare come dove c'è il massimo della specificity (caso migliore), c'è anche il

massimo di FNR (caso peggiore) e viceversa. b) NPV (sinistra) e FDR (destra). I cerchi in verdi indicano dove si trova la massima NPV e la minima FDR (in quanto è da minimizzare). Sopra ad ogni grafico è indicato il massimo per la NPV e il minimo per FDR raggiunti. Il bianco nei grafici di destra indica la presenza dei NaN, significa che l'algoritmo, dopo aver raggiunto il valore minimo di FDR, non trova più spike. La formulazione di NPV con i TN molto maggiori dei FN, fa sì che questo valore tenda sempre al valore ideale 1.

.

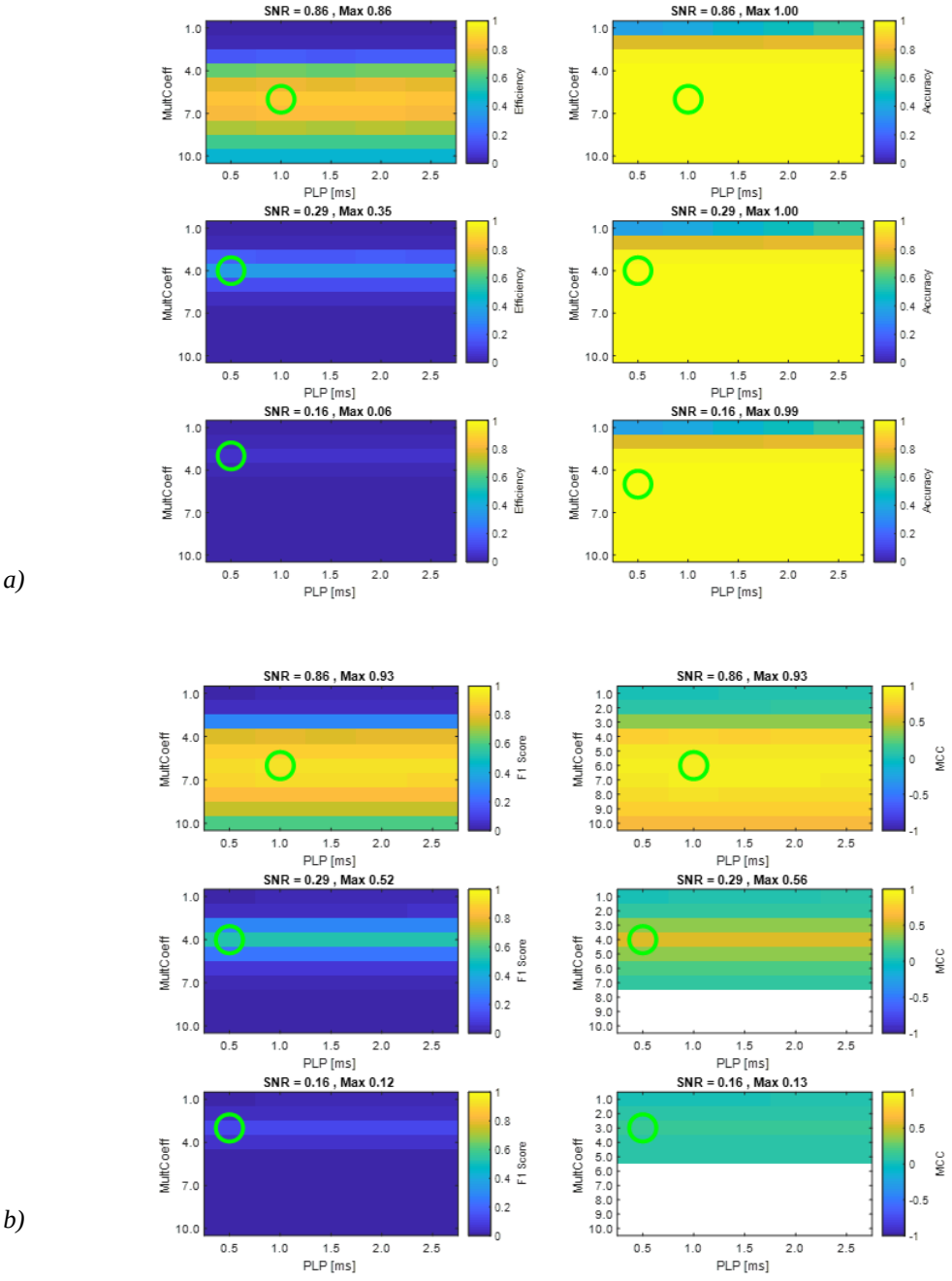


Fig.6.23 Efficiency, Accuracy, F1S e MCC per mPTSD SU per tre livelli di SNR. a) Efficiency (sinistra) e accuracy (destra). I cerchi in verdi indicano dove si trovano le

massime efficiency e accuracy. Sopra ad ogni grafico ne sono indicati i valori raggiunti. Entrambe diminuiscono all'aumentare dell'SNR, accuracy si attesta al valore ideale per la maggior parte dei parametri. b) F1S (sinistra) e MCC (destra). I cerchi in verdi indicano dove si trovano i massimi valori F1S e MCC, riportati sopra ai grafici. Il bianco indica la presenza dei NaN, significa che qualche termine al denominatore di MCC si annulla, mandando il calcolo di MCC a NaN. Si ricorda inoltre che MCC va da -1 a 1, per questo l'asse dei valori è diverso.

6.7.2 Multi Unit

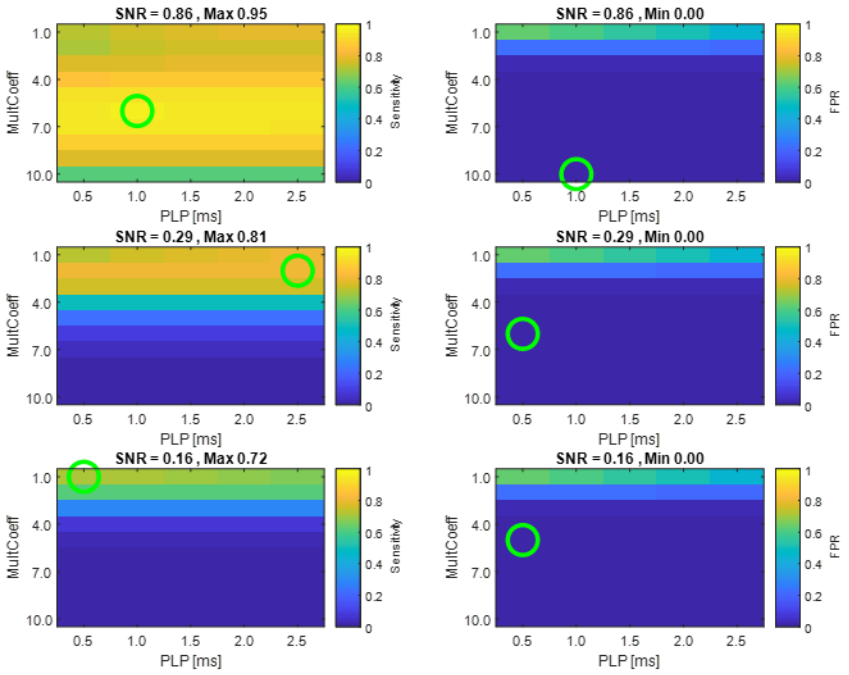
In Fig.6.24 si possono vedere sensitivity, FPR, precision e FOR, in Fig.6.25 specificity, FNR, NPV e FDR, in Fig.6.26 accuracy, efficiency, F1S e MCC.

La TW migliore continua ad essere la prima, 0.5 s, MC migliore varia maggiormente, si è scelto 4 come prima per un confronto più immediato (Tab.6.11).

MU	TPR	TNR	PPV	FNR	NPV	FPR	FDR	FOR	CSI	ACC	F1	MCC
SNR 0.16	0.07	0.99	0.74	0.92	0.97	0.0006	0.25	0.02	0.07	0.97	0.13	0.23
SNR 0.29	0.49	0.99	0.93	0.50	0.98	0.0007	0.063	0.0118	0.47	0.98	0.64	0.67
SNR 0.86	0.85	0.99	0.81	0.14	0.99	0.0047	0.18	0.0034	0.71	0.99	0.83	0.83

Tab.6.11 Valori raggiunti per ciascun indice di mPTSD con MC=4 e PLP=1 per tre livelli di SNR nel MU. I valori degli indici tendono a migliorare all'aumentare dell'SNR anche in questo caso, e ugualmente TNR, NPV e ACC mantengono lo stesso valore. mPTSD al suo meglio, nel caso migliore, raggiunge un ***CSI 0.71, F1 0.83*** *e MCC 0.83. Nel caso peggiore, rispettivamente raggiunge* ***0.07, 0.13 e 0.23****. Nel caso migliore, vengono raggiunti valori superiori a PTSD, nel caso peggiore leggermente più alti ma comparabili.*

a)



b)

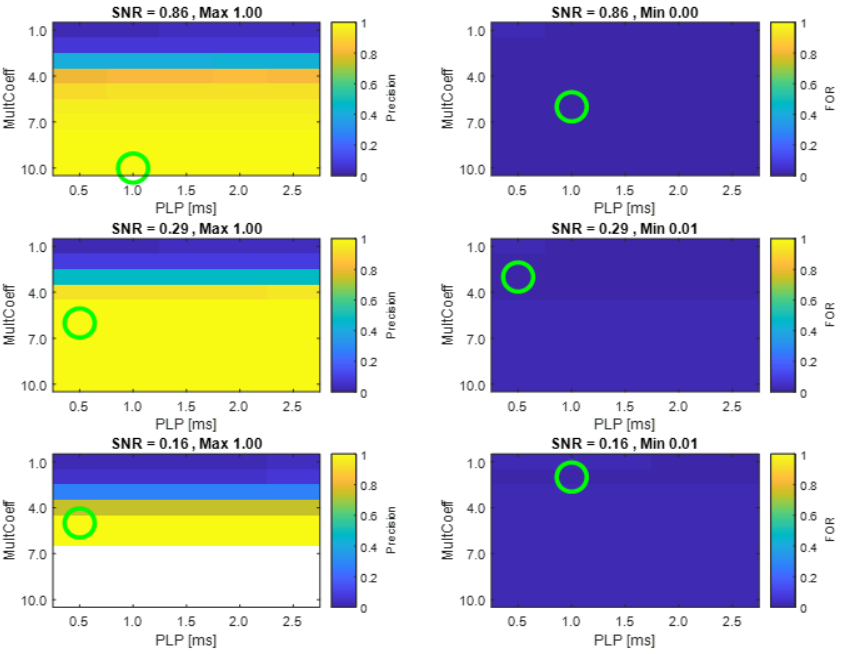
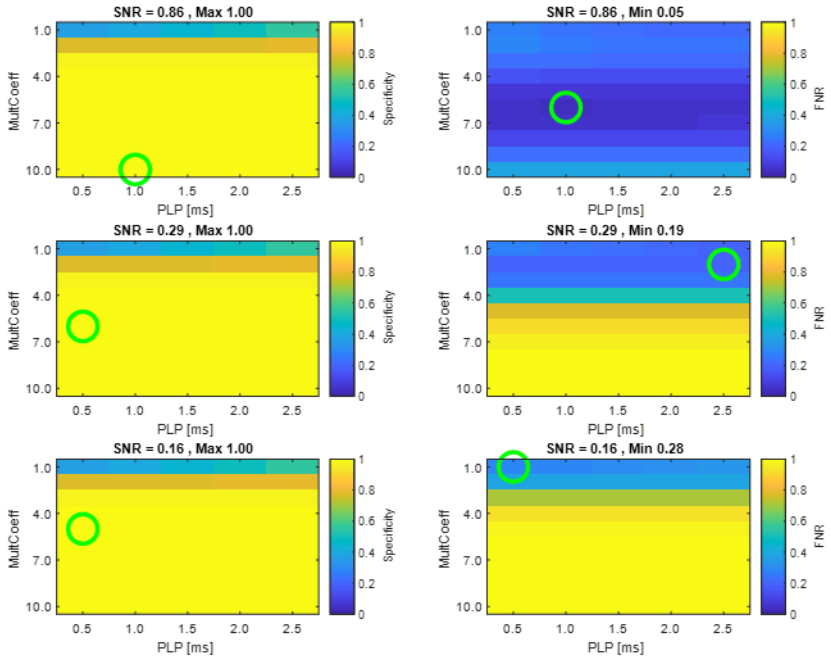


Fig.6.24 Sensitivity, FPR, Precsion e FOR per mPTSD MU per tre livelli di SNR. a). Sensitivity (sinistra) e FPR (destra). I cerchi in verdi indicano dove si trova la massima sensitivity e la minima FPR (in quanto è da minimizzare). Sopra ad ogni grafico è indicato il massimo per la sensitivity e il minimo per FPR raggiunti. La prima si attesta sempre attorno a valori piuttosto elevati, diminuendo all'aumentare dell'SNR. La FPR invece ha valore 0 per la maggior parte dei parametri b) Precision (sinistra) e FOR (destra). I cerchi in verdi indicano dove si trova la massima precision e la minima FOR (in quanto è da minimizzare). Sopra ad ogni grafico è indicato il massimo per la precision e il minimo per FOR raggiunti. Il bianco nei grafici di

sinistra indica la presenza dei NaN, significa che l'algoritmo dopo aver raggiunto il valore massimo, non trova più spike. La formulazione del FOR con i TN al denominatore fa si che questo indice tenda sempre a 0

a)



b)

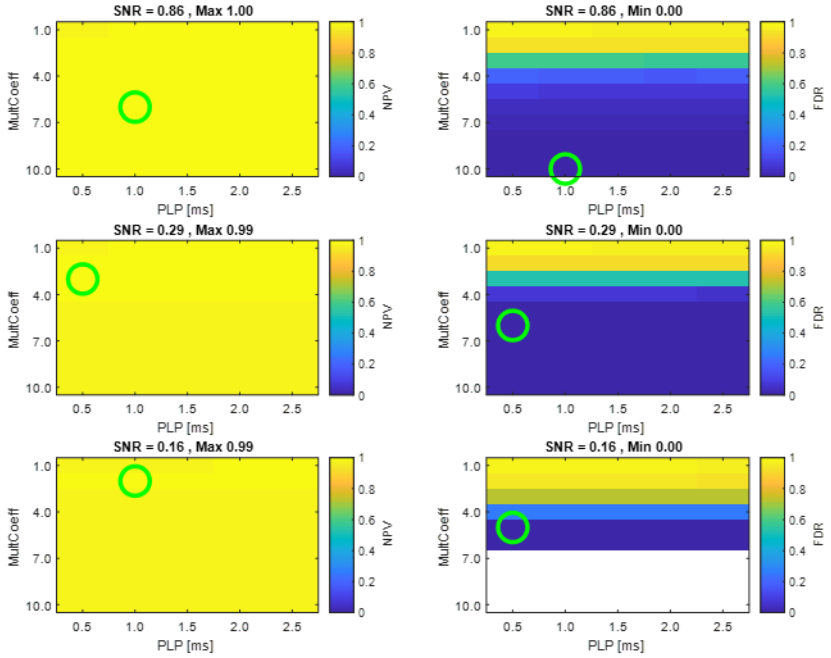
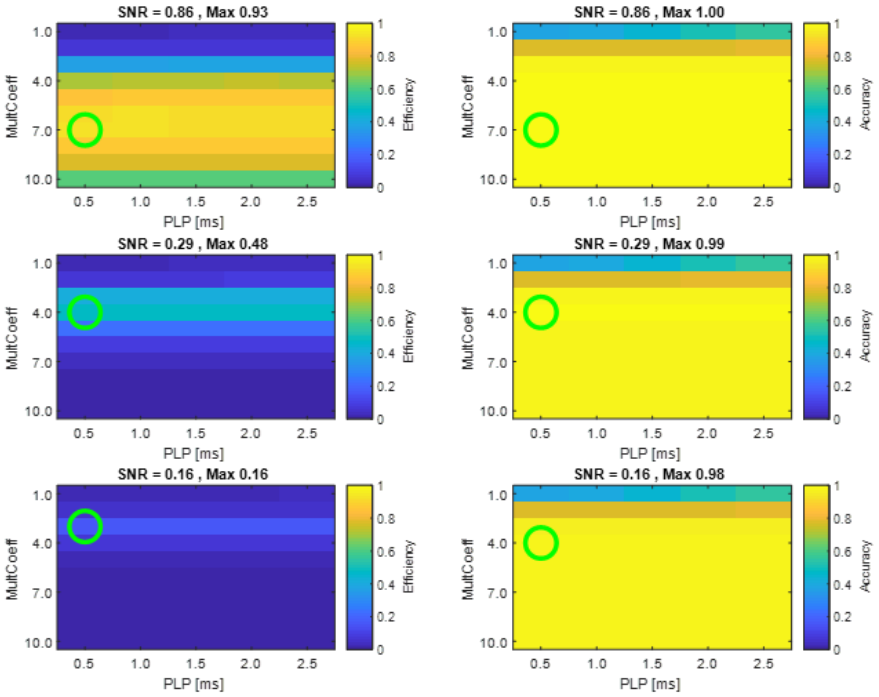


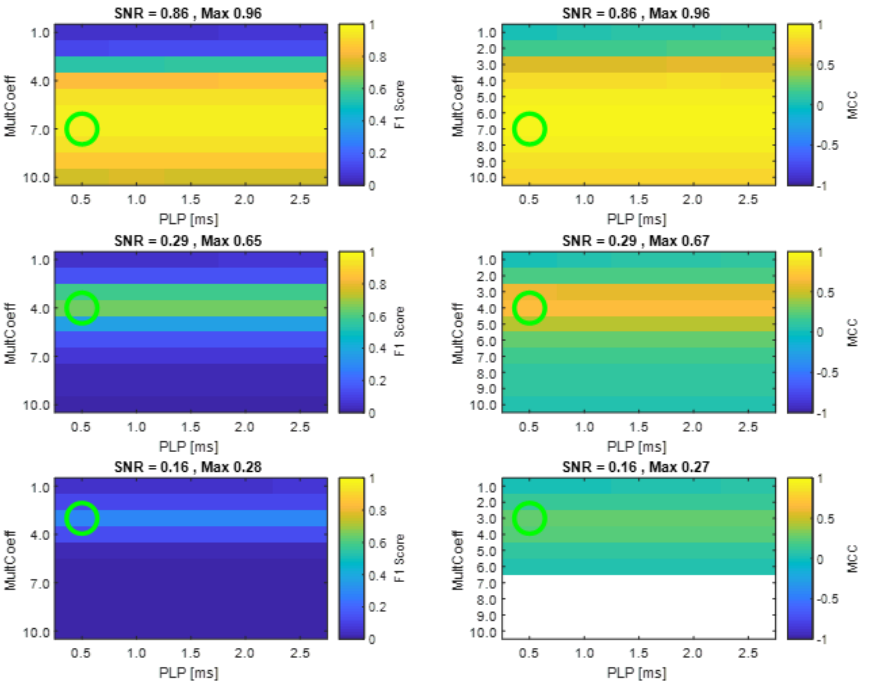
Fig.6.25 Specificity, FNR, NPV e FDR per mPTSD MU per tre livelli di SNR. a) Specificity (sinistra) e FNR (destra). I cerchi in verdi indicano dove si trova la massima specificity e la minima FNR (in quanto è da minimizzare). Sopra ad ogni grafico è indicato il massimo per la specificity e il minimo per FNR raggiunti. La prima si attesta al valore ideale per la maggior parte dei parametri. È interessante notare come dove c'è il massimo della specificity (caso migliore), c'è anche il

massimo di FNR (caso peggiore) e viceversa. b) NPV (sinistra) e FDR (destra). I cerchi in verdi indicano dove si trova la massima NPV e la minima FDR (in quanto è da minimizzare). Sopra ad ogni grafico è indicato il massimo per la NPV e il minimo per FDR raggiunti. Il bianco nei grafici di destra indica la presenza dei NaN, significa che l'algoritmo, dopo aver raggiunto il valore minimo di FDR, non trova più spike. La formulazione di NPV, con i TN molto maggiori dei FN, fa sì che tenda a 1.

a)



b)



.

Fig.6.26 Efficiency, Accuracy, F1S e MCC per mPTSD MU per tre livelli di SNR. a) Efficiency (sinistra) e accuracy (destra). I cerchi in verdi indicano dove si trovano le massime efficiency e accuracy. Sopra ad ogni grafico ne sono indicati i valori raggiunti. Entrambe diminuiscono all'aumentare dell'SNR, accuracy si attesta al valore ideale per la maggior parte dei parametri. b) F1S (sinistra) e MCC (destra). I cerchi in verdi indicano dove si trovano i massimi valori F1S e MCC, riportati sopra ai grafici. Il bianco indica la presenza dei NaN, significa che qualche termine al denominatore di MCC si annulla, mandando il calcolo di MCC a NaN. Si ricorda inoltre che MCC va da -1 a 1, per questo l'asse dei valori è diverso.

Tempo computazionale

È stato calcolato il tempo computazionale (Fig.6.27) sempre utilizzando come PLP il valore 1. SU e MU anche se MU raggiunge tempi doppi rispetto a SU. L'algoritmo è più veloce degli algoritmi precedenti ma meno di HT.

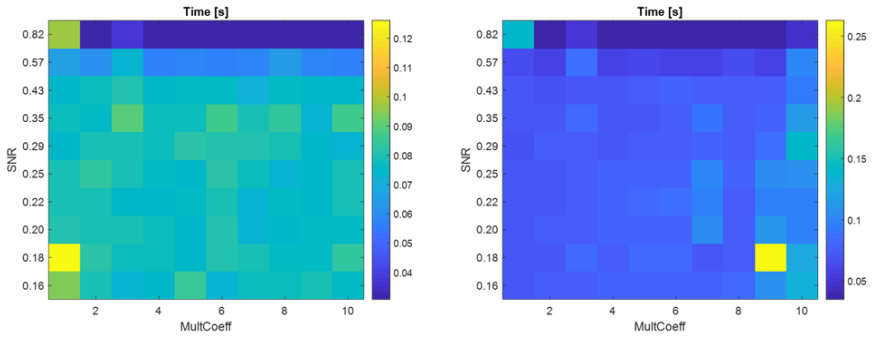


Fig.6.27 Tempi computazionali di mPTSD con PLP=1. Sinistra, tempo per il SU: il tempo massimo è raggiunto per il livello di SNR 0.18 più alto e il MC=1. Destra, tempo per MU, il tempo massimo è raggiunto con il livello di SNR 0.18 più alto e MC= 9.

6.8 TIFCO

TIFCO prevede la variazione di un solo parametro, MC, che è stato fatto variare dieci volte nell'intervallo 1-10, e l'analisi è stata effettuata per tutti i livelli di rumore creati.

I dodici indici sono stati calcolati dopo il matching degli spikes trovati dall'algoritmo. I grafici in Fig.6.28 e Fig.6.29 ne riportano l'andamento al variare di MultCoeff per 3 livelli di SNR: 0.16, 0.29 e 0.86.

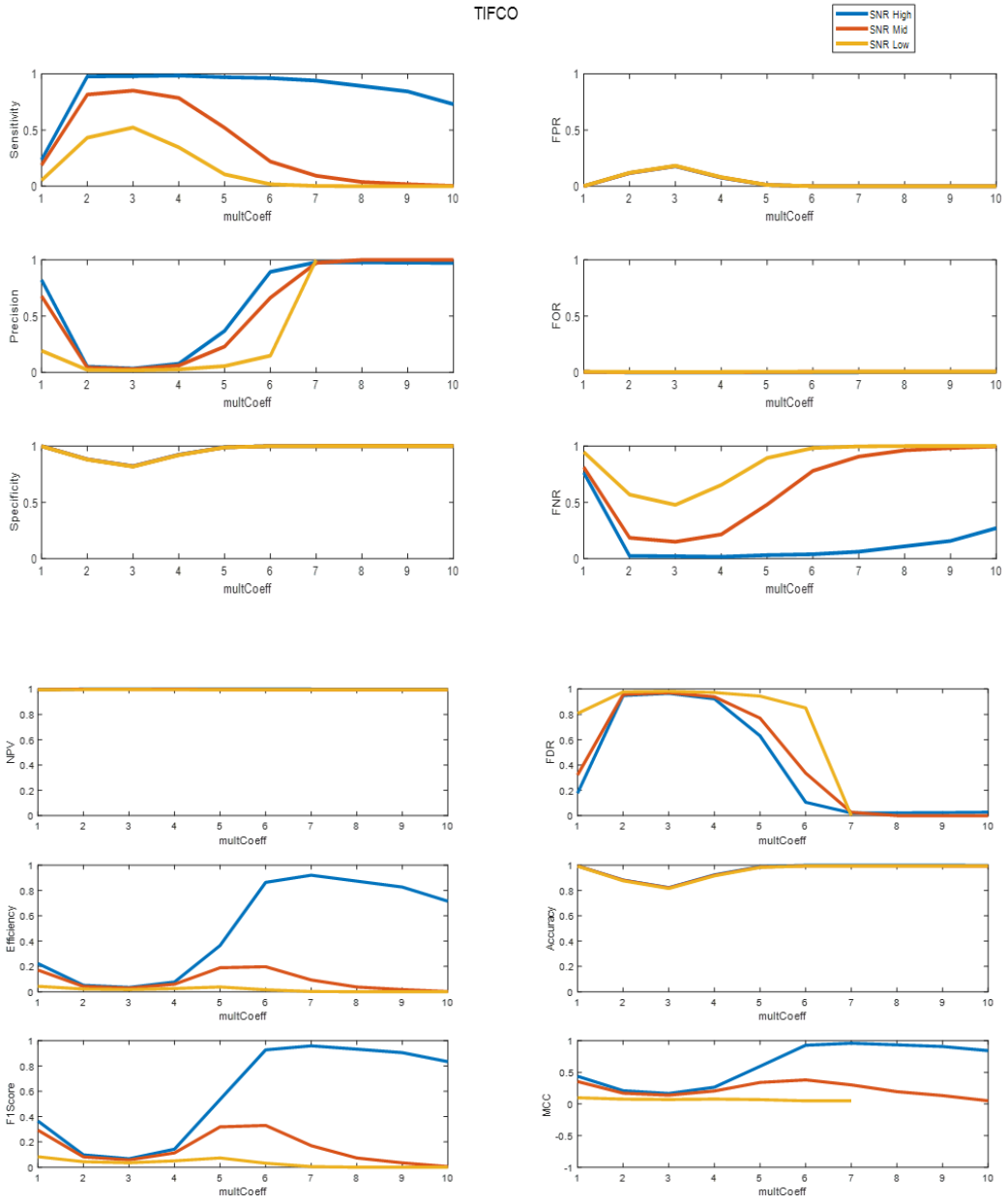


Fig.6.28 Rappresentazione grafica per tutti gli indici di TIFCO su SU. In blu sono rappresentati i risultati relativi al livello di SNR maggiore (SNR High), in rosso al livello intermedio (SNR Mid) e in giallo al livello più basso (SNR Low).

6.8.1 Single Unit

Dai grafici in Fig.6.29 si osserva che le prestazioni generali non sono eccellenti, facendo sempre riferimento al caso peggiore, i massimi valori raggiunti si attestano sempre sotto 0.2 negli indici di prestazione generale, valore piuttosto basso. La robustezza è sicuramente migliorata rispetto ad HT e HTLM. Vediamo ora, per i tre livelli di SNR con il parametro che massimizza gli indici di performance generale (MC = 6), i valori di ciascun indice (Tab.6.12)

SU	TPR	TNR	PPV	FNR	NPV	FPR	FDR	FOR	CSI	ACC	F1	MCC
SNR 0.16	0.01	0.99	0.14	0.98	0.99	0.0006	0.85	0.0064	0.016	0.99	0.03	0.04
SNR 0.29	0.21	0.99	0.66	0.78	0.99	0.0007	0.33	0.0051	0.19	0.99	0.33	0.37
SNR 0.86	0.96	0.99	0.89	0.03	0.99	0.0007	0.10	0.0002	0.86	0.99	0.92	0.92

Tab.6.12 Valori raggiunti per ciascun indice da TIFCO con MC=6 per tre livelli di SNR nel SU. I valori degli indici tendono a migliorare all'aumentare dell'SNR ma così non è per TNR, NPV e ACC (accuracy) che mantengono lo stesso valore. Si ricorda che CSI (efficiency) F1 e MCC sono gli indici di prestazione generale e rappresentano la chiave di lettura delle performance dell'algoritmo. TIFCO al suo meglio, nel caso migliore, raggiunge un ***CSI 0.86, F1 0.92 e MCC 0.92.*** *Nel caso peggiore, rispettivamente raggiunge* ***0.016, 0.03 e 0.04.***

6.8.2 Multi Unit

Le considerazioni sul MU sono analoghe a quelle del SU, si fa notare solo un aumento delle prestazioni in generale. La Tab.6.13 è analoga a Tab.6.12.

MU	TPR	TNR	PPV	FNR	NPV	FPR	FDR	FOR	CSI	ACC	F1	MCC
SNR 0.16	0.05	0.99	0.70	0.94	0.97	0.0005	0.29	0.02	0.05	0.97	0.096	0.18
SNR 0.29	0.29	0.99	0.92	0.70	0.98	0.0005	0.078	0.0164	0.28	0.98	0.44	0.51
SNR 0.86	0.84	0.99	0.98	0.15	0.999	0.0003	0.015	0.0037	0.83	0.99	0.90	0.90

Tab.6.13 Valori raggiunti per ciascun indice da TIFCO con MC=6 per tre livelli di SNR nel MU. I valori degli indici tendono a migliorare all'aumentare dell'SNR anche in questo caso, e ugualmente TNR, NPV e ACC mantengono lo stesso valore. HTLM al suo meglio, nel caso migliore, raggiunge un ***CSI 0.83, F1 0.90*** *e* ***MCC 0.90****. Nel caso peggiore, rispettivamente raggiunge* ***0.05, 0.096 e 0.18.***

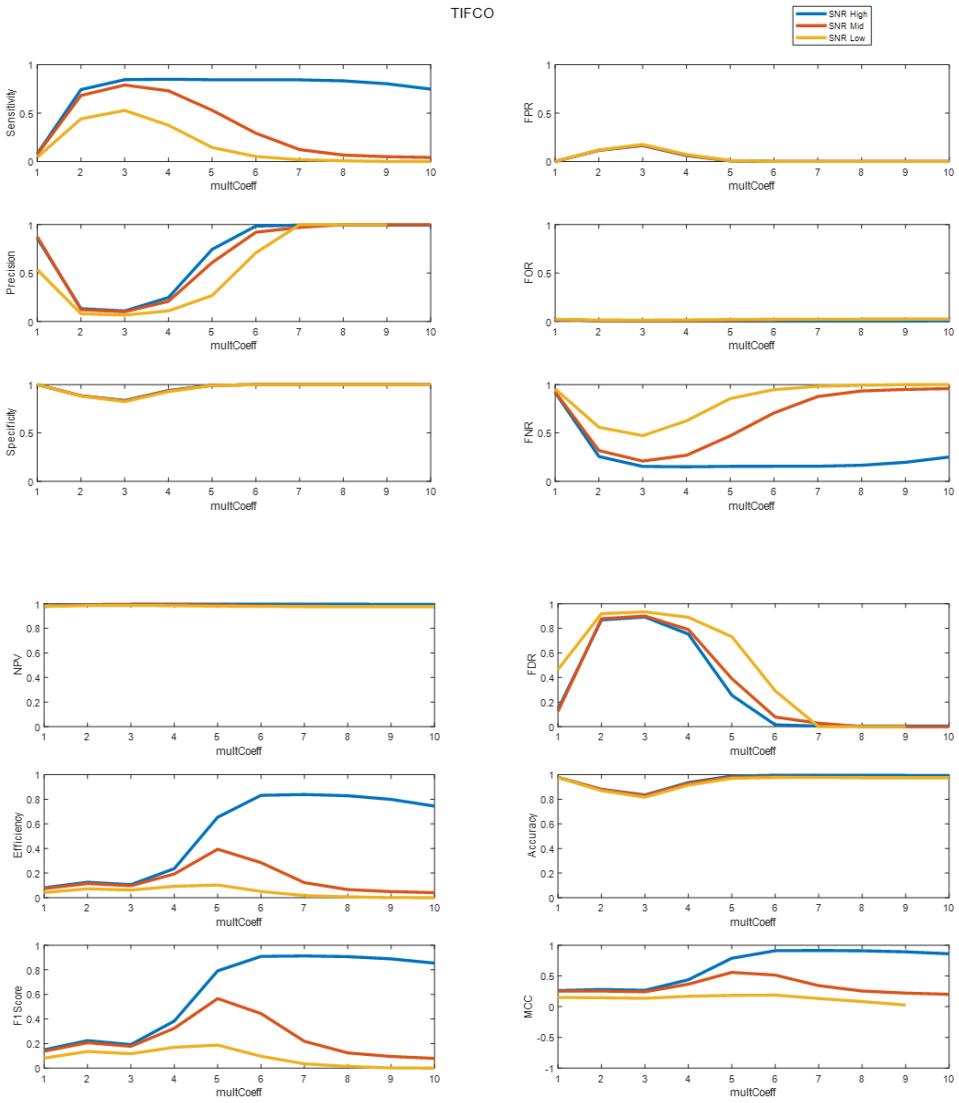


Fig.6.29 Rappresentazione grafica per tutti gli indici di TIFCO su MU. In blu sono rappresentati i risultati relativi al livello di SNR maggiore (SNR High), in rosso al livello intermedio (SNR Mid) e in giallo al livello più basso (SNR Low).

6.8.3 Tempo computazionale

L'analisi del tempo computazionale (Fig.6.30) mostra come questo algoritmo sia il più lento visto fino ad ora, con tempi che raggiungono anche i due minuti nel SU e un minuto e mezzo nel MU.

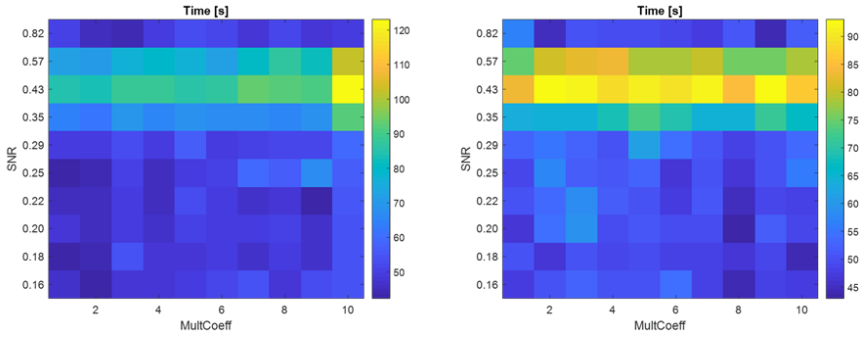


Fig.6.30 Tempi computazionali di TIFCO. Sinistra, tempo per il SU: il tempo massimo è raggiunto per il terzo livello di SNR e il MC maggiore, si nota che l'intero livello di SNR crea dei problemi dal punto di vista dei tempi computazionali. Destra, tempo per MU: analogamente, il tempo massimo è raggiunto con gli stessi valori ma complessivamente i tempi sono minori.

6.9 SWTTEO

SWTTEO prevede la variazione di un solo parametro, MC, che è stato fatto variare dieci volte nell'intervallo 110-10.000 (tramite il comando linspace), e l'analisi è stata effettuata per tutti i livelli di rumore creati. I valori di MC in questo caso sono: 110, 1208, 2307, 3406, 4505, 5604, 6703, 7802, 8901, 10.000. Da notare che in questo algoritmo i valori di MC variano anche se si tratta di una soglia unipolare.

I dodici indici sono stati calcolati dopo il matching degli spikes trovati dall'algoritmo. I grafici in Fig.6.31 e Fig.6.32 ne riportano l'andamento al variare di MultCoeff per gli stessi tre livelli di SNR delle altre analisi per SU e per MU.

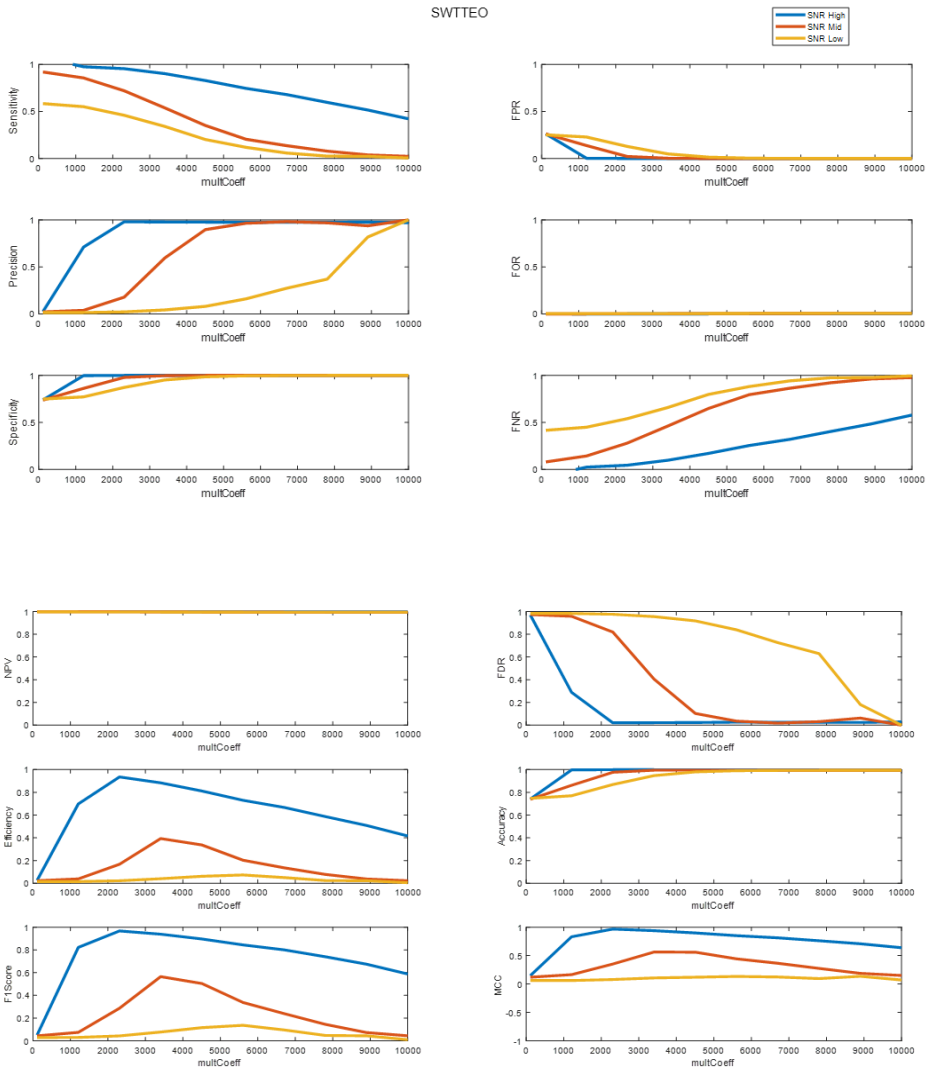


Fig.6.31 Rappresentazione grafica per tutti gli indici di SWTTEO su SU. In blu sono rappresentati i risultati relativi al livello di SNR maggiore (SNR High), in rosso al livello intermedio (SNR Mid) e in giallo al livello più basso (SNR Low).

6.9.1 Single Unit

Si tratta sicuramente dell'algoritmo più robusto visto fino ad ora, non sono infatti presenti bruschi picchi di massimo ma plateau e discese non ripide. Tutti gli indici hanno valori comparabili o più elevati rispetto agli algoritmi visti in precedenza, vediamone ora in Tab.6.14 i valori avendo fissato MC = 5604.

SU	TPR	TNR	PPV	FNR	NPV	FPR	FDR	FOR	CSI	ACC	F1	MCC
SNR 0.16	0.11	0.99	0.16	0.88	0.99	0.004	0.83	0.0057	0.07	0.99	0.13	0.13
SNR 0.29	0.20	1	0.96	0.79	0.99	0.00004	0.03	0.0052	0.20	0.99	0.33	0.44
SNR 0.86	0.74	0.99	0.97	0.25	0.99	0.0001	0.02	0.0017	0.73	0.99	0.84	0.85

Tab.6.14 Valori raggiunti per ciascun indice da SWTTEO con MC=5604 per tre livelli di SNR nel SU. I valori degli indici tendono a migliorare all'aumentare dell'SNR ma così non è per TNR, NPV e ACC (accuracy) che mantengono lo stesso valore, TNR raggiunge il valore ideale nel livello di SNR 0.29. Si ricorda che CSI (efficiency) F1 e MCC sono gli indici di prestazione generale e rappresentano la chiave di lettura delle performance dell'algoritmo. SWTTEO al suo meglio, nel caso migliore, raggiunge un ***CSI 0.73, F1 0.84 e MCC 0.85****. Nel caso peggiore, rispettivamente raggiunge* ***0.07, 0.13 e 0.13****.*

6.9.2 Multi Unit

Come anche negli altri casi, le considerazioni sul MU sono analoghe a quelle del SU, tranne che per un aumento delle prestazioni in generale. La Tab.6.15 è analoga a Tab.6.14.

MU	TPR	TNR	PPV	FNR	NPV	FPR	FDR	FOR	CSI	ACC	F1	MCC
SNR 0.16	0.18	0.99	0.53	0.81	0.98	0.0038	0.46	0.019	0.15	0.97	0.27	0.30
SNR 0.29	0.32	0.99	0.98	0.67	0.98	0.00013	0.017	0.015	0.32	0.98	0.48	0.56
SNR 0.86	0.81	0.99	0.99	0.18	0.99	0.00008	0.004	0.0044	0.81	0.99	0.89	0.89

Tab.6.15 Valori raggiunti per ciascun indice da SWTTEO con MC=5604 per tre livelli di SNR nel MU. I valori degli indici tendono a migliorare all'aumentare dell'SNR anche in questo caso, e ugualmente TNR, NPV e ACC mantengono lo stesso valore. SWTTEO al suo meglio, nel caso migliore, raggiunge un ***CSI 0.81, F1 0.89*** *e* ***MCC 0.89****. Nel caso peggiore, rispettivamente raggiunge* ***0.15, 0.27 e 0.30****. Sono i valori più alti raggiunti fino ad ora.*

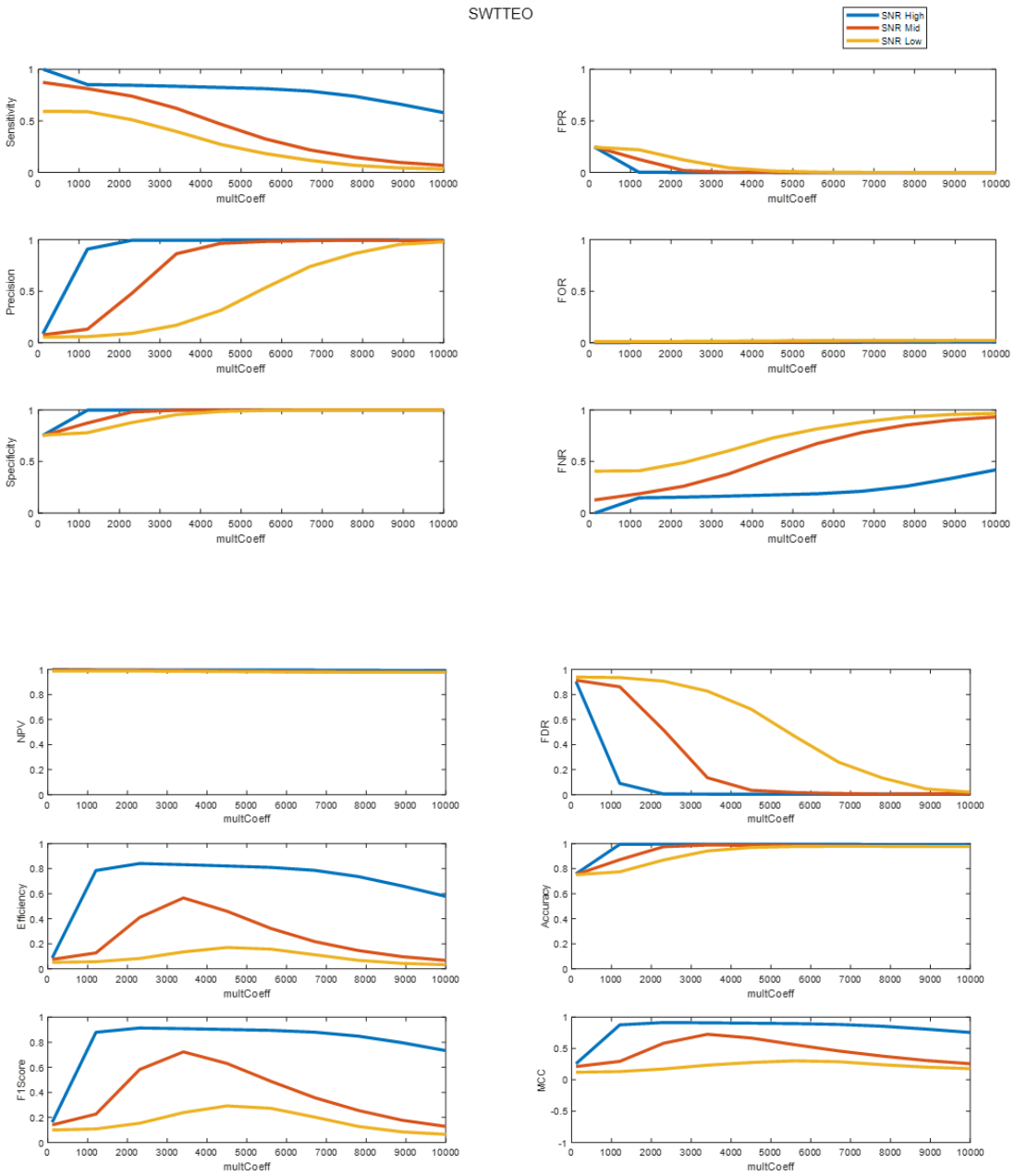


Fig.6.32 Rappresentazione grafica per tutti gli indici di SWTTEO su MU. In blu sono rappresentati i risultati relativi al livello di SNR maggiore (SNR High), in rosso al livello intermedio (SNR Mid) e in giallo al livello più basso (SNR Low).

6.9.3 Tempo computazionale

Come anche nei casi precedenti è stato calcolato il tempo computazionale (Fig.6.33). Questa analisi mostra che SWTTEO è un algoritmo lento, con alcune prestazioni che rientrano nell'ordine del minuto, in ogni caso più veloce di TIFCO.

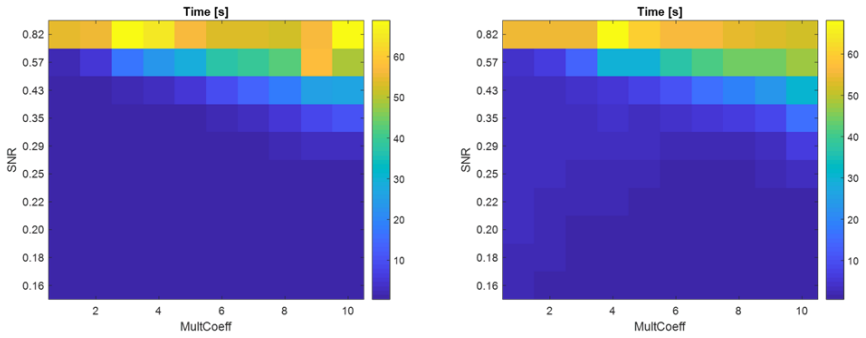


Fig.6.33 Tempi computazionali di SWTTEO. Sinistra, tempo per il SU. Destra, tempo per il MU. In entrambi i casi il tempo aumenta all'aumentare di MC e del livello di SNR.

6.10 SNEO

L'algoritmo SNEO prevede l'utilizzo di due parametri: MC, che modifica la soglia, e SmoothFactor (SF), che è la lunghezza in campioni della finestra di smoothing che utilizza questo algoritmo. MC varia sempre 10 volte nell'intervallo 1-10, mentre SF assume i seguenti valori: 1, 11, 21, 31, 41, 51, 61, 71, 81, 91.

6.10.1 Single Unit

In Fig.6.34 si possono vedere sensitivity, FPR, precision e FOR, in Fig.6.35 specificity, FNR, NPV e FDR, in Fig.6.36 accuracy, efficiency, F1S e MCC.

Lo SF influisce molto sulla variazione degli indici, così come MC. Il miglior compromesso che massimizza quasi tutti gli indici è SF=1 e MC=7. I valori degli indici con questi valori di parametri sono riportati in Tab.6.16.

SU	TPR	TNR	PPV	FNR	NPV	FPR	FDR	FOR	CSI	ACC	F1	MCC
SNR 0.16	0.11	0.99	0.16	0.88	0.99	0.004	0.83	0.0057	0.07	0.99	0.13	0.13
SNR 0.29	0.20	1	0.96	0.79	0.99	0.00004	0.03	0.0052	0.20	0.99	0.33	0.44
SNR 0.86	0.74	0.99	0.97	0.25	0.99	0.0001	0.02	0.0017	0.73	0.99	0.84	0.85

Tab.6.16 Valori raggiunti per ciascun indice da SNEO con MC=7 e SF=1 per tre livelli di SNR nel SU. I valori degli indici tendono a migliorare all'aumentare dell'SNR ma così non è per TNR, NPV e ACC (accuracy) che mantengono lo stesso valore. Si ricorda che CSI (efficiency) F1 e MCC sono gli indici di prestazione generale e rappresentano la chiave di lettura delle performance dell'algoritmo. mPTSD al suo meglio, nel caso migliore, raggiunge un ***CSI 0.63, F1 0.77 e MCC 0.77****. Nel caso peggiore, rispettivamente raggiunge* ***0.03, 0.05 e 0.09.***

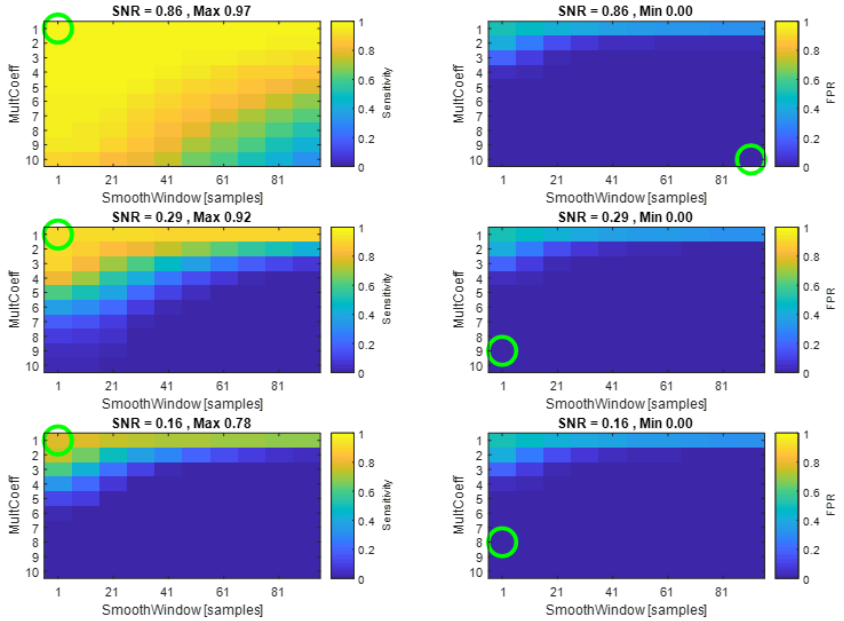
6.10.2 Multi Unit

In Fig.6.38 si possono vedere sensitivity, FPR, precision e FOR, in Fig.6.39 specificity, FNR, NPV e FDR, in Fig.6.40 accuracy, efficiency, F1S e MCC Si mantengono i parametri di sopra per creare la Tab.6.17.

MU	TPR	TNR	PPV	FNR	NPV	FPR	FDR	FOR	CSI	ACC	F1	MCC
SNR 0.16	0.07	0.99	0.74	0.92	0.97	0.0006	0.25	0.02	0.07	0.97	0.13	0.23
SNR 0.29	0.49	0.99	0.93	0.50	0.98	0.0007	0.063	0.0118	0.47	0.98	0.64	0.67
SNR 0.86	0.85	0.99	0.81	0.14	0.99	0.0047	0.18	0.0034	0.71	0.99	0.83	0.83

Tab.6.17. Valori raggiunti per ciascun indice da SNEO con MC=7 e SF=1 per tre livelli di SNR nel MU. I valori degli indici tendono a migliorare all'aumentare dell'SNR anche in questo caso, e ugualmente TNR, NPV e ACC mantengono lo stesso valore. ATLM al suo meglio, nel caso migliore, raggiunge un ***CSI 0.97, F1 0.98 e MCC 0.98.*** *Nel caso peggiore, rispettivamente raggiunge* ***0.07, 0.13 e 0.23****.*

a)



b)

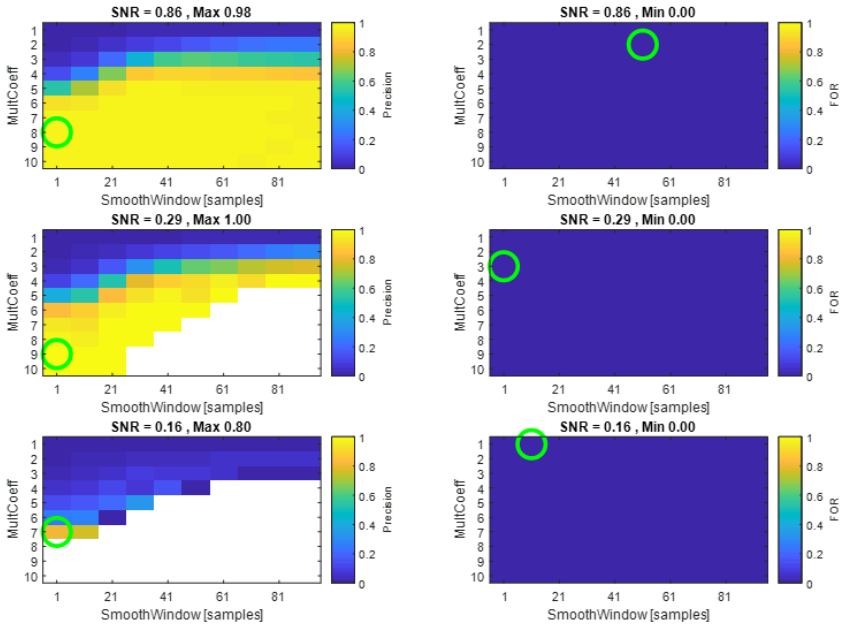
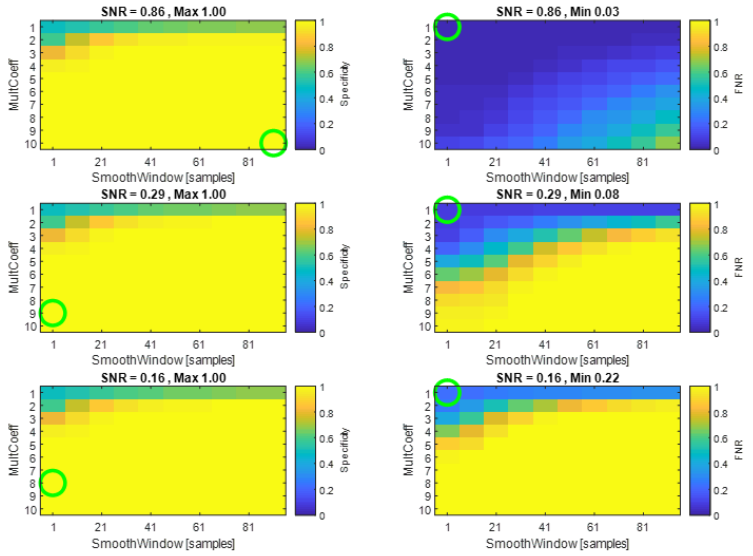


Fig.6.34 Sensitivity, FPR, Precsion e FOR per SNEO SU per tre livelli di SNR. a). Sensitivity (sinistra) e FPR (destra). I cerchi in verdi indicano dove si trova la massima sensitivity e la minima FPR (in quanto è da minimizzare). Sopra ad ogni grafico è indicato il massimo per la sensitivity e il minimo per FPR raggiunti. La prima si attesta sempre attorno a valori piuttosto elevati, diminuendo all'aumentare dell'SNR. La FPR invece ha valore 0 per la maggior parte dei parametri b) Precision (sinistra) e FOR (destra). I cerchi in verdi indicano dove si trova la massima precision e la minima FOR (in quanto è da minimizzare). Sopra ad ogni grafico è indicato il massimo per la precision e il minimo per FOR raggiunti. Il bianco nei grafici di sinistra indica la presenza dei NaN, significa che l'algoritmo dopo aver raggiunto il

valore massimo, non trova più spike. La formulazione del FOR con i TN al denominatore fa si che questo indice tenda sempre a 0.

a)



b)

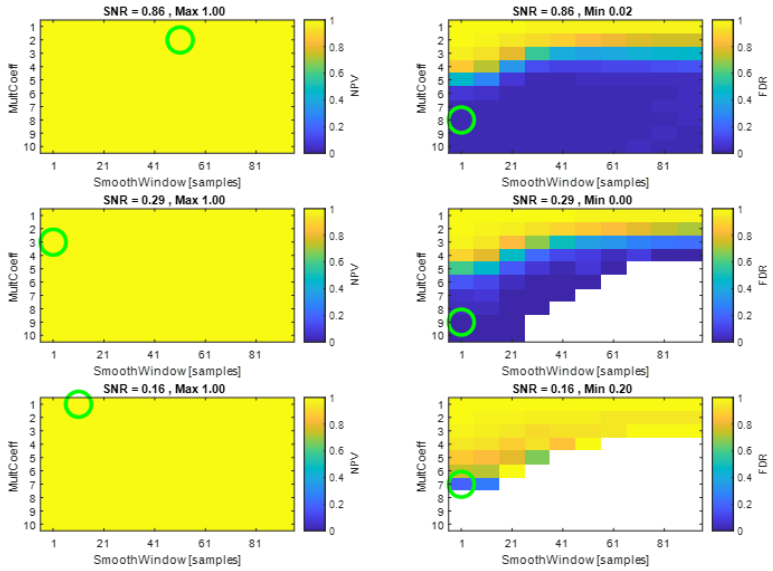
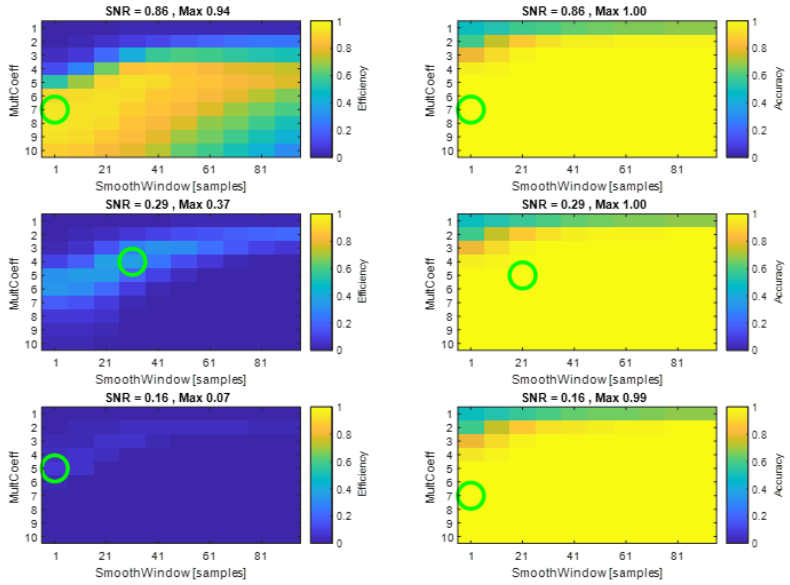


Fig.6.35. Specificity, FNR, NPV e FDR per SNEO SU per tre livelli di SNR. a) Specificity (sinistra) e FNR (destra). I cerchi in verdi indicano dove si trova la massima specificity e la minima FNR (in quanto è da minimizzare). Sopra ad ogni grafico è indicato il massimo per la specificity e il minimo per FNR raggiunti. La prima si attesta al valore ideale per la maggior parte dei parametri. È interessante notare come dove c'è il massimo della specificity (caso migliore), c'è anche il massimo di FNR (caso peggiore) e viceversa. b) NPV (sinistra) e FDR (destra). I cerchi in verdi indicano dove si trova la massima NPV e la minima FDR (in quanto è

da minimizzare). Sopra ad ogni grafico è indicato il massimo per la NPV e il minimo per FDR raggiunti. Il bianco nei grafici di destra indica la presenza dei NaN, significa che l'algoritmo, dopo aver raggiunto il valore minimo di FDR, non trova più spike. La formulazione di NPV con i TN molto maggiori dei FN, fa sì che questo valore tenda sempre al valore ideale 1.

.

a)



b)

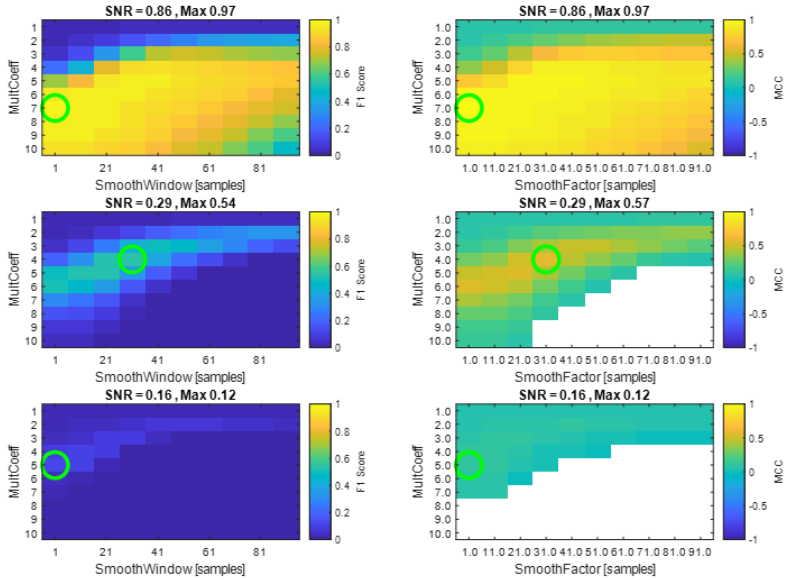


Fig.6.36. Efficiency, Accuracy, F1S e MCC per ATLM SU per tre livelli di SNR. a) Efficiency (sinistra) e accuracy (destra). I cerchi in verdi indicano dove si trovano le massime efficiency e accuracy. Sopra ad ogni grafico ne sono indicati i valori raggiunti. Entrambe diminuiscono all'aumentare dell'SNR, accuracy si attesta al valore ideale per la maggior parte dei parametri. b) F1S (sinistra) e MCC (destra). I

cerchi in verdi indicano dove si trovano i massimi valori F1S e MCC, riportati sopra ai grafici. Il bianco indica la presenza dei NaN, significa che qualche termine al denominatore di MCC si annulla, mandando il calcolo di MCC a NaN. Si ricorda inoltre che MCC va da -1 a 1, per questo l'asse dei valori è diverso.

6.10.3 Tempo computazionale

È stato calcolato il tempo computazionale (Fig.6.37) sempre utilizzando come PLP il valore 1. SU e MU anche se MU raggiunge tempi doppi rispetto a SU. L'algoritmo è più veloce degli algoritmi precedenti ma meno di HT.

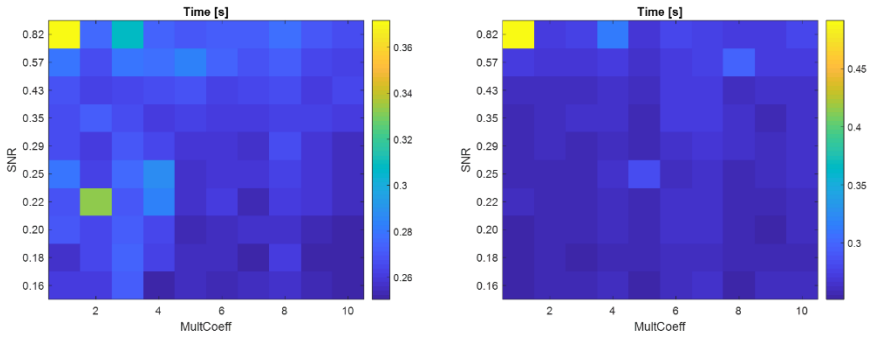
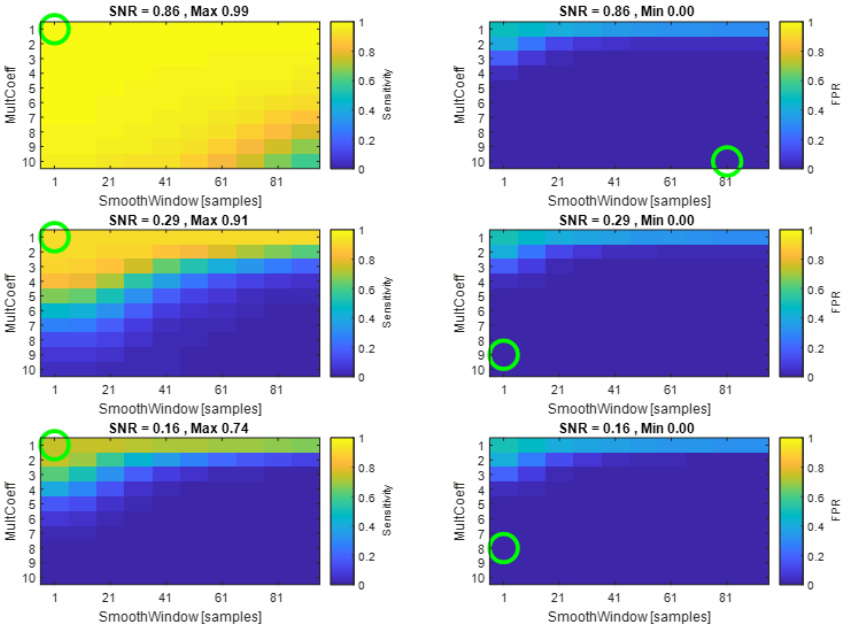


Fig.6.37 Tempi computazionali di SNEO con SmoothFactor=1. Sinistra, tempo per il SU: il tempo massimo è raggiunto per il livello di SNR 0.18 più alto e il MC=1. Destra, tempo per MU, il tempo massimo è raggiunto con il livello di SNR 0.18 più alto e MC= 9.

a)



b)

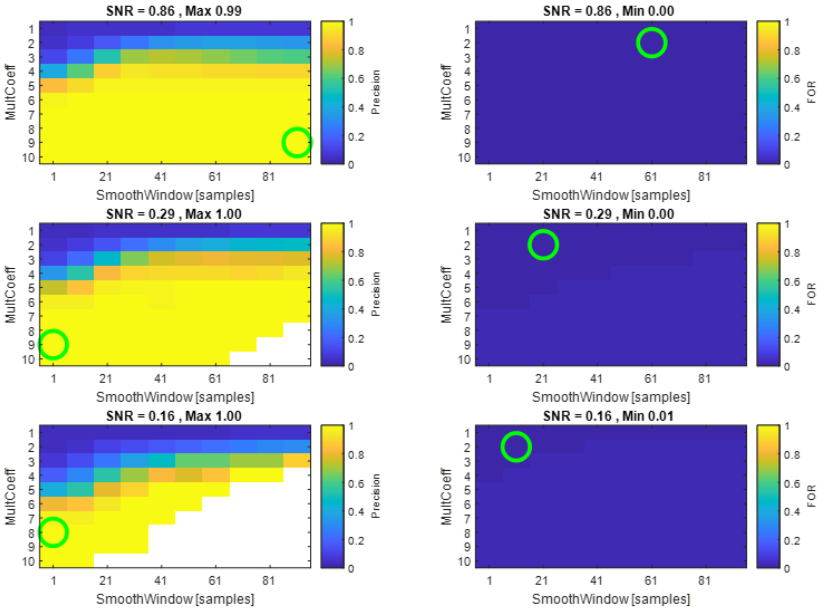
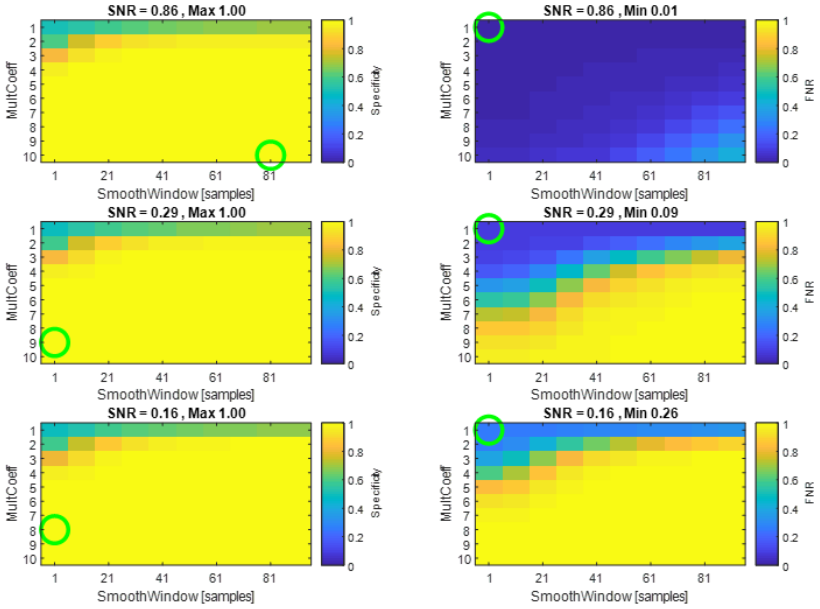


Fig..6.38 Sensitivity, FPR, Precsion e FOR per SNEO MU per tre livelli di SNR. a). Sensitivity (sinistra) e FPR (destra). I cerchi in verdi indicano dove si trova la massima sensitivity e la minima FPR (in quanto è da minimizzare). Sopra ad ogni grafico è indicato il massimo per la sensitivity e il minimo per FPR raggiunti. La prima si attesta sempre attorno a valori piuttosto elevati, diminuendo all'aumentare dell'SNR. La FPR invece ha valore 0 per la maggior parte dei parametri b) Precision (sinistra) e FOR (destra). I cerchi in verdi indicano dove si trova la massima precision e la minima FOR (in quanto è da minimizzare). Sopra ad ogni grafico è indicato il massimo per la precision e il minimo per FOR raggiunti. Il bianco nei grafici di sinistra indica la presenza dei NaN, significa che l'algoritmo dopo aver raggiunto il

valore massimo, non trova più spike. La formulazione del FOR con i TN al denominatore fa si che questo indice tenda sempre a 0

a)



b)

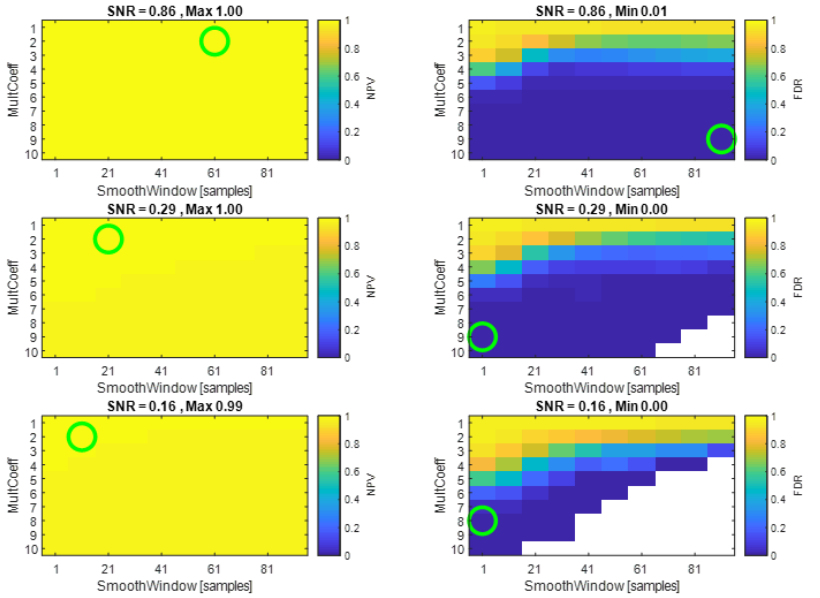
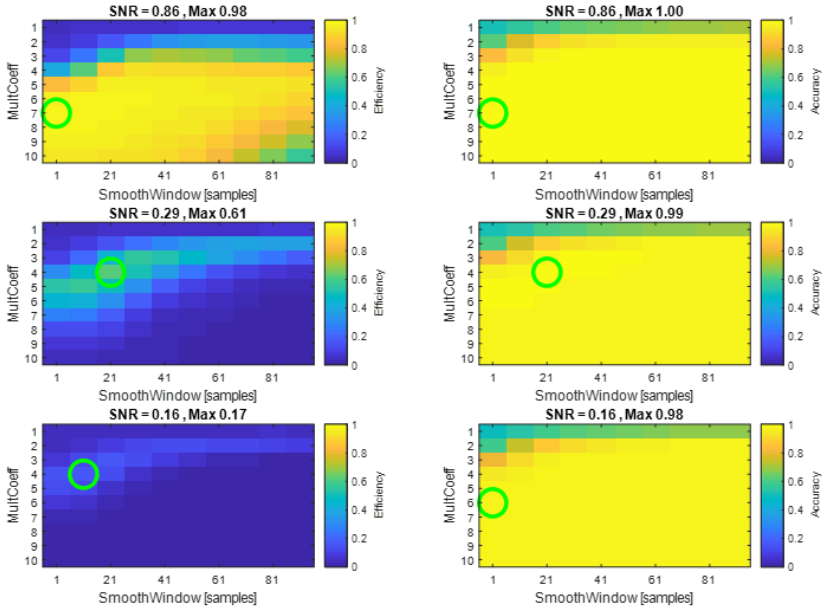


Fig.6.39 Specificity, FNR, NPV e FDR per SNEO MU per tre livelli di SNR. a) Specificity (sinistra) e FNR (destra). I cerchi in verdi indicano dove si trova la massima specificity e la minima FNR (in quanto è da minimizzare). Sopra ad ogni grafico è indicato il massimo per la specificity e il minimo per FNR raggiunti. La prima si attesta al valore ideale per la maggior parte dei parametri. È interessante notare come dove c'è il massimo della specificity (caso migliore), c'è anche il massimo di FNR (caso peggiore) e viceversa. b) NPV (sinistra) e FDR (destra). I cerchi in verdi indicano dove si trova la massima NPV e la minima FDR (in quanto è

da minimizzare). Sopra ad ogni grafico è indicato il massimo per la NPV e il minimo per FDR raggiunti. Il bianco nei grafici di destra indica la presenza dei NaN, significa che l'algoritmo, dopo aver raggiunto il valore minimo di FDR, non trova più spike. La formulazione di NPV, con i TN molto maggiori dei FN, fa sì che tenda a 1.

a)



b)

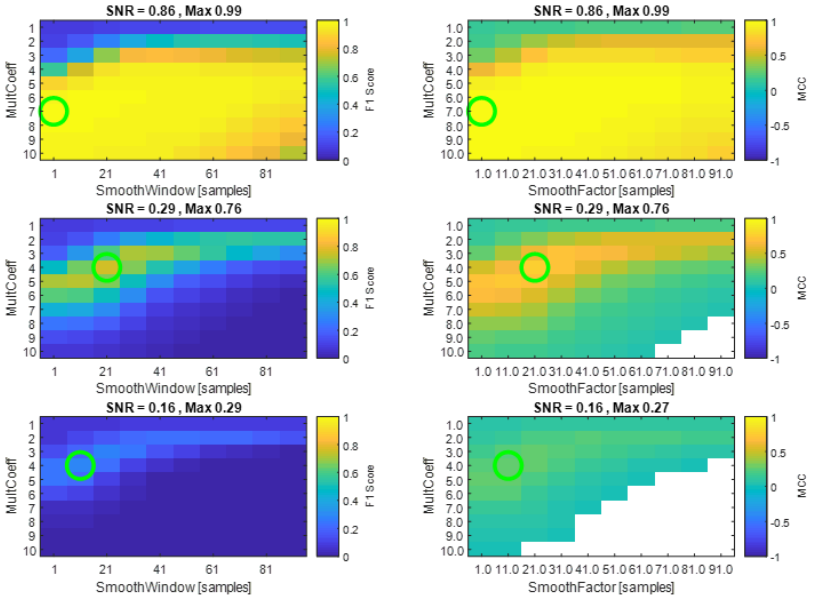


Fig.6.40 Efficiency, Accuracy, F1S e MCC per SNEO MU per tre livelli di SNR. a) Efficiency (sinistra) e accuracy (destra). I cerchi in verdi indicano dove si trovano le massime efficiency e accuracy. Sopra ad ogni grafico ne sono indicati i valori

raggiunti. Entrambe diminuiscono all'aumentare dell'SNR, accuracy si attesta al valore ideale per la maggior parte dei parametri. b) F1S (sinistra) e MCC (destra). I cerchi in verdi indicano dove si trovano i massimi valori F1S e MCC, riportati sopra ai grafici. Il bianco indica la presenza dei NaN, significa che qualche termine al denominatore di MCC si annulla, mandando il calcolo di MCC a NaN. Si ricorda inoltre che MCC va da -1 a 1, per questo l'asse dei valori è diverso.

6.11 Confronto tra gli algoritmi

Per condensare però le informazioni e dare una stima di quale sia il miglior algoritmo per l'analisi di questo tipo di segnale, è necessario mostrare dei confronti immediati tra gli indici di prestazione. Non tutti gli indici sono però informativi, dalle analisi infatti FOR, NPV, FNR, accuracy e FPR, mantengono quasi sempre gli stessi valori per tutti i parametri in tutti gli algoritmi e verranno quindi trascurati.

Per mostrare il massimo delle prestazioni che possono raggiungere gli algoritmi nei vari indici, nel caso di quegli algoritmi che possiedono due parametri, si è deciso di scegliere la finestra che li massimizza, mantenendone il valore nei tre livelli di SNR. I parametri quindi saranno diversi tra gli algoritmi. Visto che le performance del MU sono semplicemente una versione scalata di quelle del SU, verranno solo riportati i risultati relativi al MU in quanto sono questi i segnali che vengono analizzati nella realtà.

L'ultimo confronto che viene fatto è relativo al Final Score di cui si è parlato all'inizio del capitolo, anche in questo caso vale lo stesso discorso del MU e del SU e quindi verranno riportati solo i risultati relativi al MU.

6.11.1 Sensitivity

Facendo riferimento solo al caso di SNR peggiore, la sensitivity migliore è raggiunta da SNEO perché, sebbene il valore massimo sia toccato da HTLM e ATLM, è più robusto degli altri mantenendo un valore elevato (Fig.6.41).

6.11.2 Precision

Facendo riferimento solo al caso di SNR peggiore, la precison migliore è raggiunta da SNEO perché è la curva che si trova al di sopra delle curve che arrivano a fine grafico per più tempo (Fig.6.42).

6.11.3 Specificity

Facendo riferimento solo al caso di SNR peggiore, la specificity migliore è raggiunta da SWTTEO e TIFCO (Fig.6.43).

6.11.4 Efficiency

Si ricorda che efficiency è uno degli indici di prestazioni in generale. Facendo riferimento solo al caso di SNR peggiore, l'efficiency migliore è raggiunta da mPTSD, il più robusto con il valore più alto è però e SWTTEO (Fig.6.44).

6.11.5 F1 Score

Anche F1 Score è uno degli indici di prestazioni in generale. Facendo riferimento solo al caso di SNR peggiore, F1 Score migliore è raggiunto da mPTSD, il più robusto con il valore più alto è però e SWTTEO (Fig.6.45).

6.11.6 MCC

MCC è uno l'ultimo degli indici di prestazioni in generale. Facendo riferimento solo al caso di SNR peggiore, MCC migliore è raggiunta da SWTTEO che è anche il più robusto (Fig.6.46).

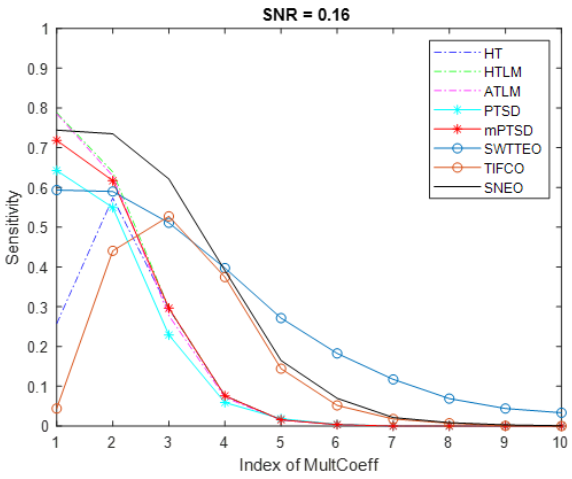


Fig.6.41 a) Andamenti delle sensitivity nei casi migliori del secondo parametro per chi lo possiede, per ogni algoritmo per il livello di SNR peggiore. L'algoritmo migliore in questo caso è SNEO perché, sebbene i valori più alti siano raggiunti da HTLM e ATML, questi scendono molto rapidamente, mentre SNEO cala più dolcemente, suggerendo una maggiore robustezza. I peggiori sono TIFCO, PTSD e HT che si interrompe già al secondo coefficiente indicando una scarsissima robustezza.

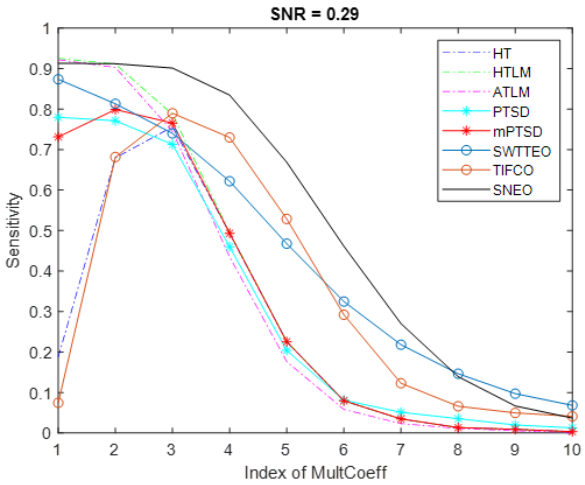


Fig.6.41 b) Andamenti delle sensitivity nei casi migliori del secondo parametro per chi lo possiede, per ogni algoritmo per il livello di SNR intermedio. L'algoritmo migliore in questo caso è sempre SNEO perché, raggiunge i valori più elevati e cala più dolcemente, suggerendo una maggiore robustezza. Il peggiore è HT, che si interrompe al quarto coefficiente.

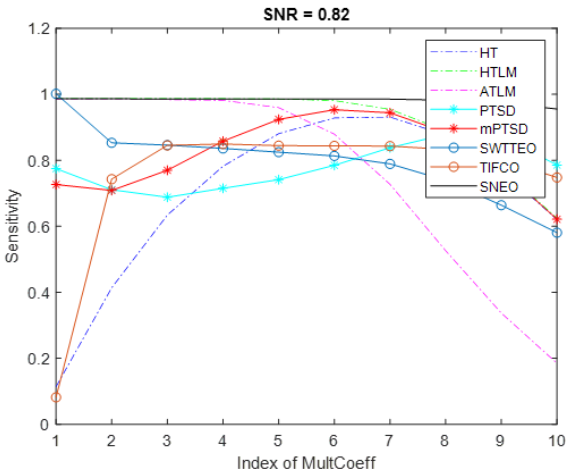


Fig.6.41 c) Andamenti delle sensitivity, nei casi migliori del secondo parametro per chi lo possiede, per ogni algoritmo per il livello di SNR più alto. Il migliore è SNEO che si mantiene al valore ideale per tutti i parametri. I peggiori sono HT e ATLM che mostrano la loro scarsissima robustezza.

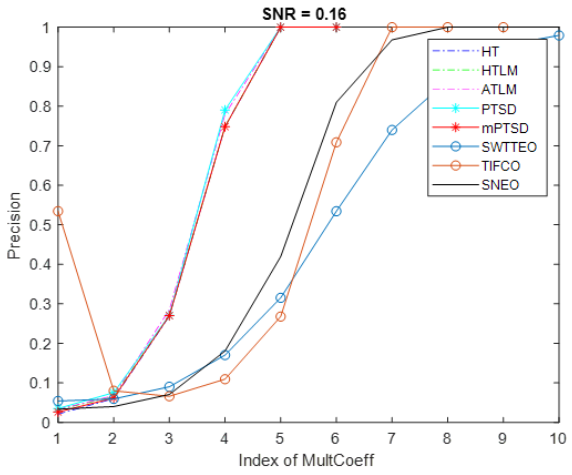


Fig.6.42 a) Andamenti delle precision, nei casi migliori del secondo parametro per chi lo possiede, per ogni algoritmo per il livello di SNR peggiore. Tutti gli algoritmi raggiungono il valore ideale. I primi a raggiungerlo sono HT, HTLM, ATLM, PTSD e mPTSD, ma questi si interrompono perché l'algoritmo non è più in grado di trovare spike con quei parametri, non rendendoli particolarmente robusti. Il peggiore è SWTTEO.

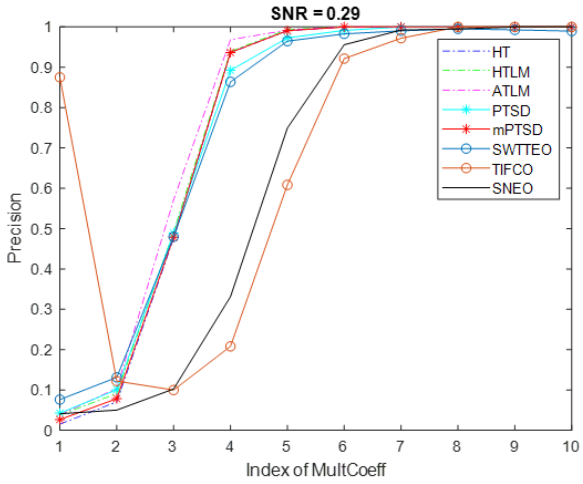


Fig.6.42 b) Andamenti delle precision, nei casi migliori del secondo parametro per chi lo possiede, per ogni algoritmo per il livello di SNR peggiore. Per il caso di SNR 0.29 la situazione è analoga al grafico precedente, tranne per il fatto che gli algoritmi sopracitati arrivano a fine corsa. Il peggiore è SNEO.

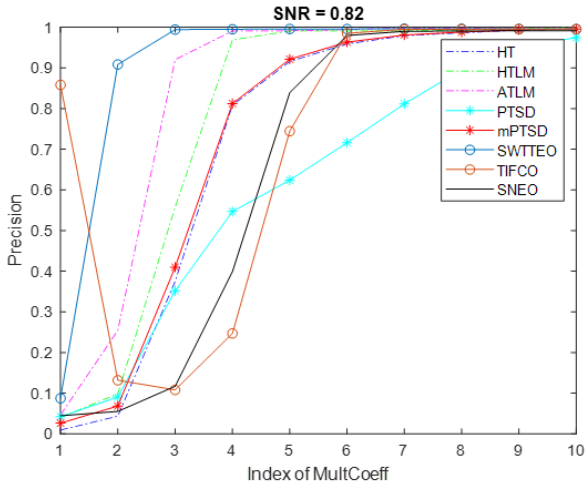


Fig.6.42 c) Andamenti delle precision, nei casi migliori del secondo parametro per chi lo possiede, per ogni algoritmo per il livello di SNR migliore. Il migliore algoritmo è SWTTEO perché è il primo a raggiungere il valore ideale e a mantenerlo, il peggiore PTSD,

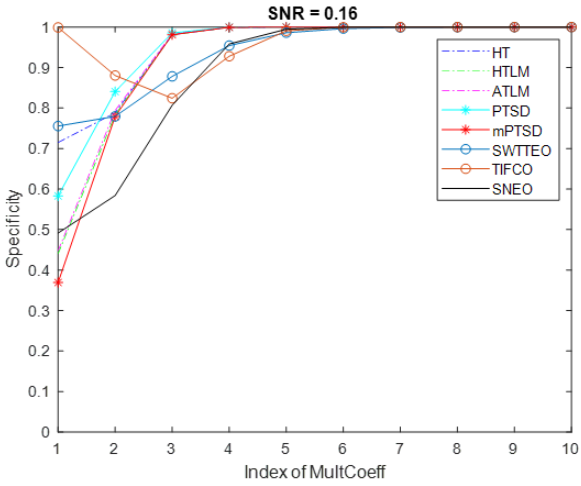


Fig.6.43 a) Andamenti delle specificity, nei casi migliori del secondo parametro per chi lo possiede, per ogni algoritmo per il livello di SNR peggiore.Tutti gli algoritmi raggiungono il valore ideale. I primi a raggiungerlo sono HT, HTLM, ATLM, PTSD e mPTSD. Allo stesso tempo, mPTSD è quello che parte dal valore più basso. In questo caso il migliore è TIFCO, che si fa notare, è l'unico che parte dal valore ideale, peggiora e risale. Il peggiore è SNEO.

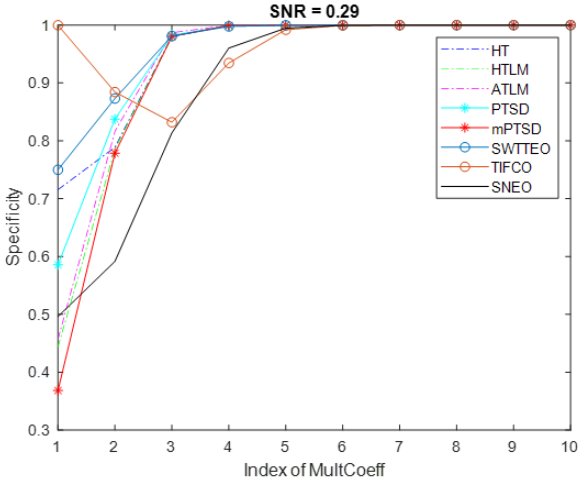


Fig.6.43 b) Andamenti delle specificity, nei casi migliori del secondo parametro per chi lo possiede, per ogni algoritmo per il livello di SNR intermedio. La situazione è analoga al grafico sopra tranne che il migliore è SWTTEO. Il peggiore resta SNEO.

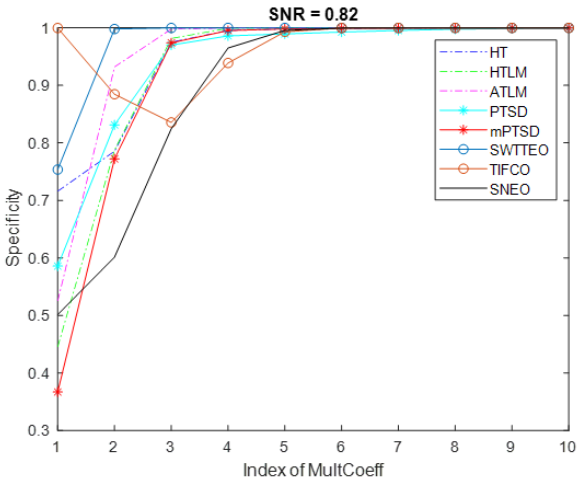


Fig.6.43 c) Andamenti delle specificity, nei casi migliori del secondo parametro per chi lo possiede, per ogni algoritmo per il livello di SNR migliore. L'algoritmo migliore è SWTTEO e il peggiore è SNEO.

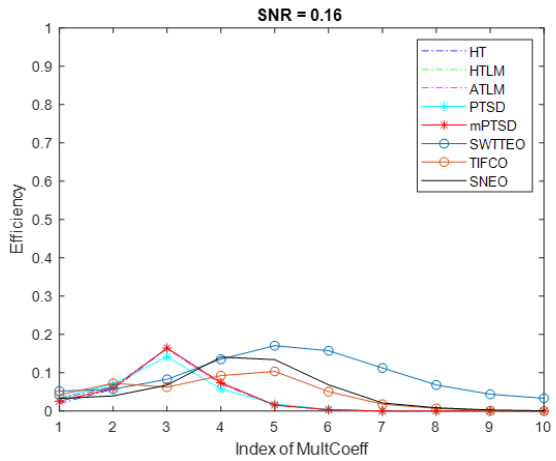


Fig.6.44 a) Andamenti delle efficiency, nei casi migliori del secondo parametro per chi lo possiede, per ogni algoritmo per il livello di SNR peggiore. Il valore massimo è raggiunto da mPTSD, il più robusto è però SWTTEO.

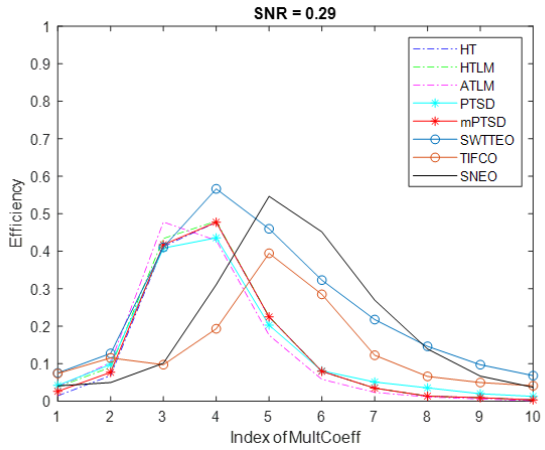


Fig.6.44 b) Andamenti delle efficiency, nei casi migliori del secondo parametro per chi lo possiede, per ogni algoritmo per il livello di SNR intermedio. Il migliore è SWTTEO seguito da SNEO, anche se nessuno si dimostra particolarmente robusto

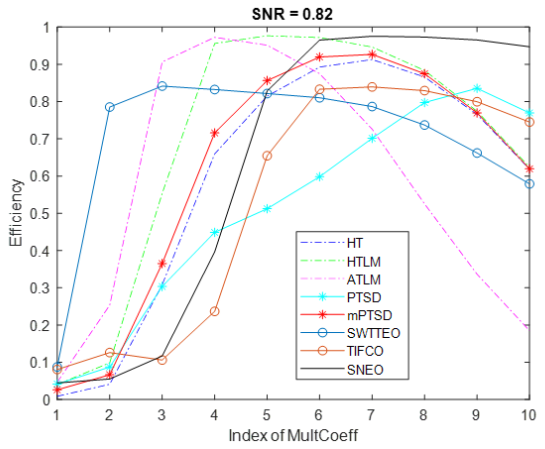


Fig.6.44 c) Andamenti delle efficiency, nei casi migliori del secondo parametro per chi lo possiede, per ogni algoritmo per il livello di SNR migliore. Le performance migliorano nettamente per tutti, il peggiore è PTSD.

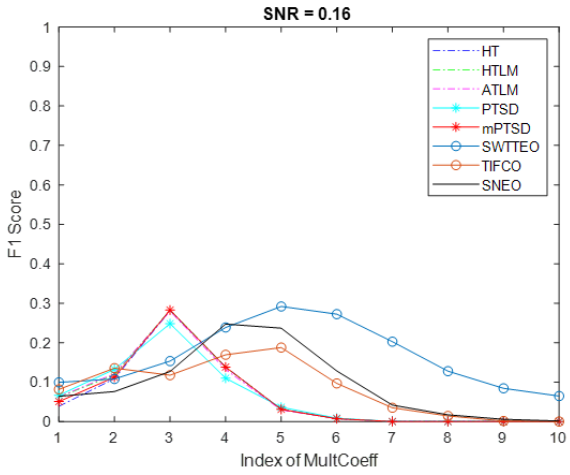


Fig.6.45 a) Andamenti di F1 Score, nei casi migliori del secondo parametro per chi lo possiede, per ogni algoritmo per il livello di SNR peggiore. Il grafico è analogo a quello dell'efficiency ma con valori più elevati. Il valore massimo è raggiunto da mPTSD, il più robusto è però SWTTEO.

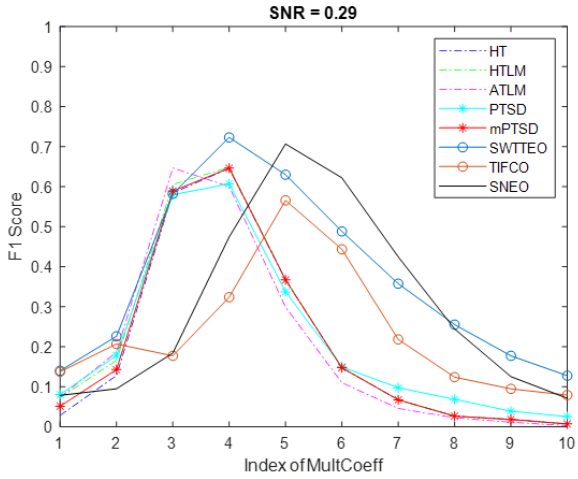


Fig.6.45 b) Andamenti di F1 Score, nei casi migliori del secondo parametro per chi lo possiede, per ogni algoritmo per il livello di SNR intermedio. Il migliore è SWTTEO seguito da SNEO, anche se nessuno si dimostra particolarmente robusto sempre come in efficiency.

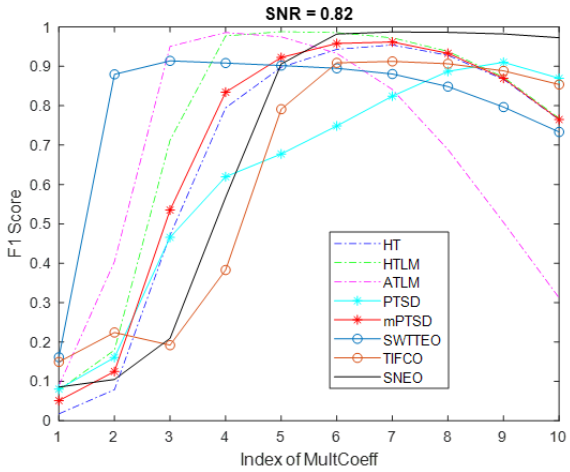


Fig.6.45 c) Andamenti di F1 Score, nei casi migliori del secondo parametro per chi lo possiede, per ogni algoritmo per il livello di SNR migliore. Le performance migliorano nettamente per tutti, il peggiore è PTSD insieme a TIFCO.

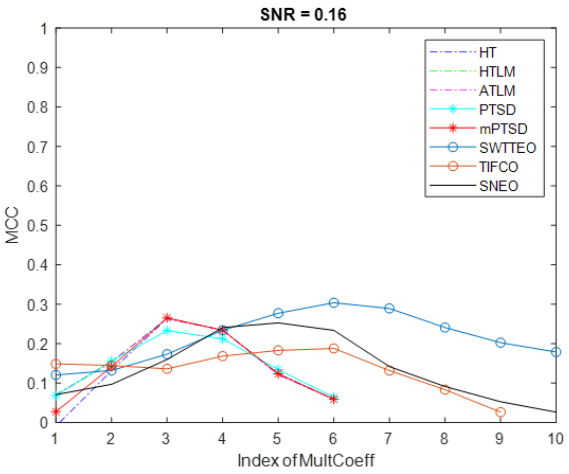


Fig.6.46 a) Andamenti di MCC nei casi migliori del secondo parametro per chi lo possiede, per ogni algoritmo per il livello di SNR peggiore. Il valore massimo è raggiunto da SWTTEO che è anche il più robusto. Da notare che che HT, HTLM ATLM, PTSD e mPTSD si interompono bruscamente, indicando l'annullamento di uno dei termini al denominatore di MCC.

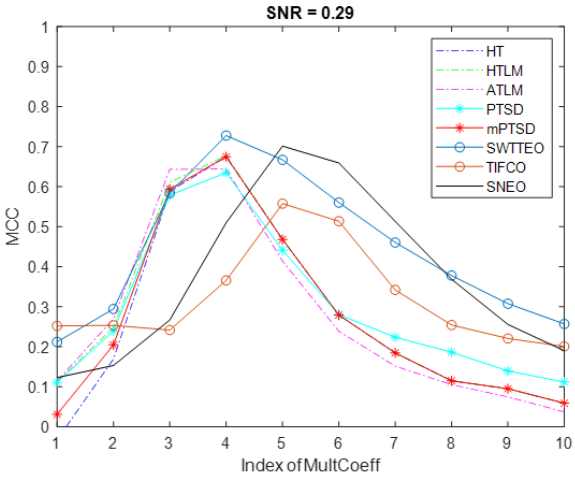


Fig.6.46 b) Andamenti di MCC nei casi migliori del secondo parametro per chi lo possiede, per ogni algoritmo per il livello di SNR intermedio. Il migliore è SWTTEO seguito da SNEO e mPTSD, la robustezza sembra migliorare rispetto ai due indici precedenti

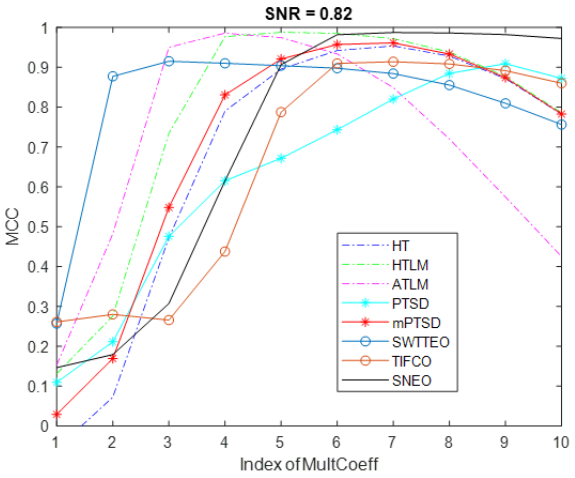


Fig.6.46 c) Andamenti di MCC nei casi migliori del secondo parametro per chi lo possiede, per ogni algoritmo per il livello di SNR migliore. Le performance migliorano nettamente per tutti, il peggiore è PTSD insieme a TIFCO.

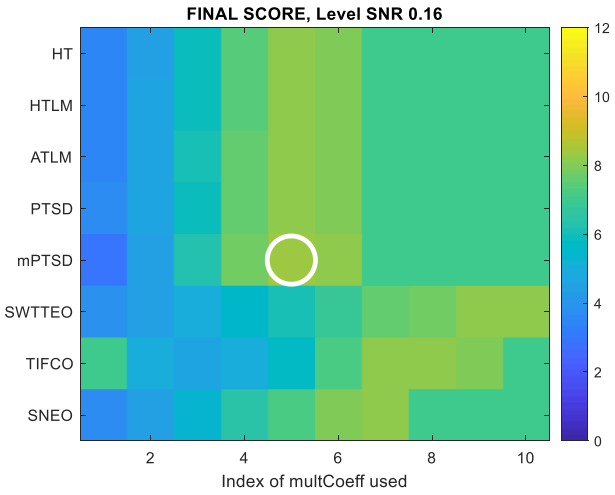


Fig6.47 a) Final Score per il livello di SNR più basso. L'algoritmo che raggiunge il valore più alto è l'algoritmo mPTSD, con un punteggio di ***8.39/12****. Essendo il livello di SNR più basso è normale che i valori di FS siano piuttosto bassi. HT, HTLM, ATLM e PTSD raggiungono un punteggio simile a mPTSD, ma i valori prima e dopo il punteggio massimo sono più bassi rispetto a mPTSD indicando anche minore robustezza. SWTTEO è il più robusto, non presentando ma la presenza dei NaN (che sono stati trasformati in 0) indicata dal colore verde acqua, ma non raggiunge valori abbastanza elevati.*

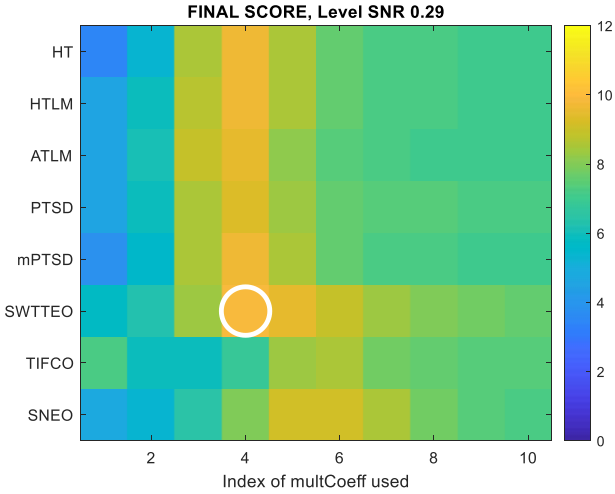


Fig.6.47 b) Final Score per il livello di SNR intermedio. Il migliore è l'algoritmo SWTTEO con un punteggio di ***9,82/12.*** *SWTTEO è come al solito il più robusto, ed è seguito da mPTSD. TIFCO è il peggiore, seguito da SNEO. Bisogna considerare che la differenza con gli altri algoritmi in termini di punteggio è di pochi decimali, quindi non si può parlare di migliore in assoluto.*

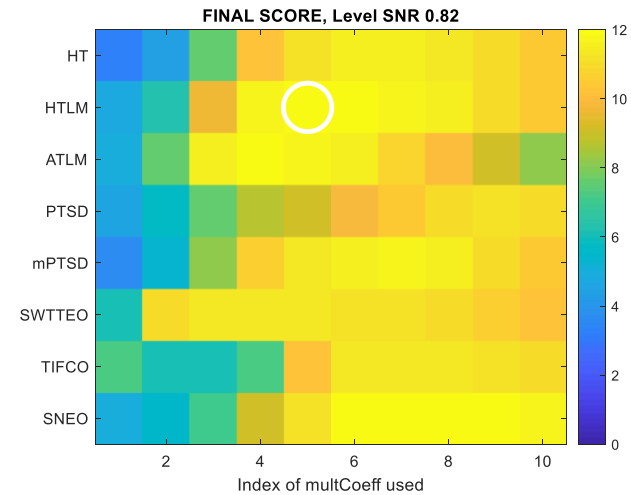


Fig.6.47 c) Final Score per il livello di SNR maggiore. Questo è il livello di SNR meno informativo perché è il più "facile", infatti le performance aumentano per tutti gli algoritmi. L'algoritmo che raggiunge un FS più elevato è HTML con ***11.89/12*** *ma tutti gli algoritmi tranne PTSD, che è il peggiore, raggiungono valori vicini a questo, inferiori di pochi decimali. Migliorata la robustezza in tutti.*

6.11.7 Final Score

Come indice riassuntivo di tutti gli indici analizzati, quindi anche quelli non riportati nei grafici di confronto. Nel caso di SNR peggiore il migliore risulta essere mPTSD, per l'SNR intermedio il vincitore è SWTTEO e, infine, nel caso migliore vince HTLM. In Fig. 6.47 sono mostrati gli andamenti del FS al variare degli *Index of MultCoeff*, in quanto gli MC non sono uguali per tutti gli algoritmi ma sono sempre 10, per i tre livelli di SNR solitamente analizzati (a,b,c). I cerchi bianchi segnalano il raggiungimento del FS più elevato, indicando l'algoritmo e il parametro con cui lo raggiunge, Il colore giallo corrisponde a un punteggio maggiore e il blu al minore.

I vincitori sono gli algoritmi che raggiungono il punteggio più alto nei vari livelli di SNR:

- Per il livello più basso, nonché il caso più interessante, il vincitore è mPTSD con un punteggio di 8.39/12.
- Per il livello intermedio il vincitore è SWTTEO con un punteggio di 9.82/12.
- Per il livello più alto, il vincitore è HTLM con un punteggio di 11.89/12 ma con gli altri algoritmi che raggiungono valori molto vicini a questo.

Non c'è un algoritmo vincitore assoluto sugli altri perché l'algoritmo vincitore cambia a seconda del livello di SNR, anche in quelli qui non riportati, e poi perché tutti raggiungono valori abbastanza comparabili l'uno all'altro.

Conclusioni

Le attività descritte nell'ambito di questa Tesi si inseriscono nel settore multidisciplinare della Neuroingegneria. La Neuroingegneria è un campo in continua crescita che mira a trovare soluzioni tecnologiche alternative per i disordini del sistema nervoso. Parte dell'interesse per queste tecnologie è dovuta alla possibilità di andare a sostituire le terapie farmacologiche con l'impiego di stimolazioni elettriche focali somministrate da dispositivi innovativi in grado di comunicare direttamente con il cervello. Tali dispositivi vengono defeniti neuroprotesi cerebrali, un esempio dei quali è stato sviluppato pressola Kansas University (Guggenmos et al., 2013). Questa neuroprotesi è stata testata su modello animale e ha dimostrato la sua efficacia in caso di lesione all'area motoria, favorendo il recupero della funzionalità della zampa anteriore (notevolmente compromessa a seguito del danno cerebrale). Il principio di funzionamento della neuroprotesi si basa sul riconoscimento di un segnale di interesse (spike) nella regione sensoriale del cervello dell'animale e la conseguente stimolazione nella regione premotoria al fine di creare un 'ponte artificiale' tra le due regioni, bypassando così quella lesionata. Determinarne la tempistica esatta degli spikes e discriminarli dal rumore ambientale è quindi un passo fondamentale nello sviluppo o nel miglioramento di un dispositivo di questo tipo. Ed è con questa importante motivazione e finalità che ho affrontato questo lavoro di Tesi, focalizzato sulla Spike Detection.

La Spike Detection consiste nel riconoscimento degli spikes presenti nel tracciato elettrofisiologico. Per effettuarla si utilizzano algoritmi che sfruttano diverse strategie per riuscire a riconoscere gli eventi. La difficoltà dello sviluppo di queste protesi sta anche nella natura del segnale in questione in quanto non se ne conoscono tutte le proprietà: si tratta di un segnale di natura stocastica, facilmente soggetto

a rumore ambientale e di cui non si conoscono gli istanti degli spikes a priori. Il primo passo, quindi, per andare a fare delle considerazioni suglialgoritmi che analizzano questi segnali, è creare un modello di segnale di cui si conoscano tutte le proprietà.

Gli obiettivi che questo lavoro si Tesi si è posto sono quindi stati i seguenti:

1. **Fornire** un ground truth tramite lo sviluppo di un modello computazionale di segnale neurale.
2. **Studiare** e adattare un set di algoritmi di Spike Detection già presenti in letteratura.
3. **Modificare** un algoritmo ad alte prestazioni sviluppato all'interno di IIT in passato (ma impiegato per applicazioni diverse) per adattarlo alle caratteristiche dei segnali in esame.
4. **Confrontare** le prestazioni di tutti gli algoritmi di Spike Detection.

Il workflow che ho seguito parte dalla creazione del segnale sintetico andando a generare un set di distribuzioni utilizzate in letteratura, ovvero la distribuzione esponenziale, gamma e gaussiana inversa (Gerstein & Mandelbrot, 1964; Tuckwell, 1988; Pipa et al., 2013; Kass et al., 2018) che riproducono gli Inter Spike Intervals (ISI), ovvero le distanze che intercorrono tra gli spikes, e la cui somma cumulativa ne restituisce la posizione. Si tratta quindi di un processo puntale che dà origine a un treno di delta di Dirac dove ogni impulso corrisponde a una posizione dello spike. Per rendere il segnale creato il più fisiologico possibile, si è scelto di sovrappore forme d'onda presenti in segnali registrati in-vivo agli istanti degli spikes. Nello specifico, le forme d'onda sono state estratte da segnali extra-cellulari registrate da ratti. Da queste registrazioni si ottengono le Multi Unit Activity (MUA), ovvero un segnale provenienie da popolazioni eterogenee di neuroni. Per ottenere le forma d'onda, bisogna prima discriminare le

singole unità che compongo il MUA, ovvero le Single Unit Activity (SUA). Questo è fattibile grazie alla proprietà che hanno i neuroni registrati tramite elettrodi extracellulari di generare ognuno una sua onda caratteristica. Utilizzando il procedimento di Spike Sorting che effettua un clustering delle forme d'onda, si possono ottenere le famiglie di curve provenieneti da una singola unità. Sommando 3 SU ho ottenuto il MUA, a cui ho successivamento aggiunto il rumore creato in ambiente Simulink in diversi livelli di SNR, che dal caso migliore al peggiore risultano essere: 0.86, 0.57, 0.43, 0.35, 0.29, 0.25, 0.22, 0.2, 0.18, 0.16.

Il modello creato riproduce fedelmente un segnale biologico perché ne rispetta le caratteristiche quali le distribuzioni delle ISI, la frequenza di scarica (Averna et al., 2018) e le forme d'onda, provenienti da segnali in-vivo. Anche altri studi, come quello di Swindale (Swindale & Spacek, 2015), utilizzano un approccio modellistico simile a quello da me adottato.

Ai segnali sono stati applicati algoritmi di Spike Detection allo stato dell'arte e ne sono poi state valutate le performance. È stato anche modificato un algoritmo ad alte prestazioni sviluppato diversi anni fa da IIT, il Precise Timing Spike Detection (PTSD) (Maccione et al., 2009), per cercare di migliorarne le prestazioni quando applicato a dati in vivo. Gli algoritmi che ho utilizzato per il confronto sono stati in parte implementati da me e in parte adattati da algoritmi trovati in letteratura e sono stati testati su segnali SUA e MUA

Le strategie per l'identificazione degli eventi spike utilizzate dagli algoritmi di SD si basano sostanzialmente su tre caratteristiche: (1) ampiezza del segnale; (2) correlazione di template di forme d'onda; (3) energia. Della prima classe fanno parte gli algoritmi: Hard Threshold (HT), Hard Threshold Local Maxima (HTLM), Adaptive Threshold Local Maxima (ATLM), Precise Timing Spike Detection (PTSD) (Maccione et al., 2009) e modified Precise Timing Spike Detection (mPTSD). Della terza classe fanno parte: Time-Frequency based Convolution algorithm (TIFCO) (Lieb et al., 2017), Stationary

Wavelet based Teager Energy Operator (SWTTEO) (Lieb et al., 2017) e Smoothed Non-Linear Energy Operator (SNEO)(Malik et al., 2016). Non ho preso in considerazione algoritmi appartenenti alla seconda classe.

Per valutare le performance degli algoritmi sono stati utilizzati 12 indici statistici che si basano sul conteggio dei True Positive, False Positive, True Negative e False Negative. Per cercare di riassumere tutti gli indici, che mostrano ognuno un aspetto diverso delle performance dell'algoritmo, è stato calcolato il Final Score (FS), ovvero sono stati sommati i punteggi dei vari indici (il reciproco nel caso in cui gli indici dovessero essere minimizzati) per un massimo punteggio di 12.

Le modifiche apportate all'algoritmo PTSD, ovvero la ricerca di picco minimo/massimo dopo aver già trovato un altro picco massimo/minimo e la sostituzione della soglia differenziale con una soglia secca, hanno portato ad un miglioramento nelle prestazioni di questo algoritmo: ad esempio il FS calcolato per PTSD nel caso di SNR intermedio è pari a 8.4, mentre mPTSD raggiunge 9.1.

Nel caso di SNR più alto (e quindi il caso più semplice) HTLM otteneva il punteggio più alto, anche se tutti gli altri algoritmi raggiungevano valori molto simili, inferiori di qualche decimale, risultando quindi di prestazioni comparabili. Nel caso di SNR intermedio, l'algoritmo più performante si è dimostrato essere SWTTEO con un FS di 9.8, mentre nel caso di SNR più basso (che è anche il caso più interessante in quanto il segnale extra-cellulare è un segnale molto rumoroso e quindi si avvicina di più alle condizioni reali) l'algoritmo migliore si è mostrato essere mPTSD con un punteggio di 8.4.

Oltre alle considerazioni prestazionali degli algoritmi bisogna fare anche delle considerazioni su robustezza e tempo computazionale. La robustezza è sicuramente una caratteristica da ricercare negli algoritmi, in quanto il valore della soglia viene definito in modo abbastnza casuale o basato su considerazioni empiriche dall'operatore,

ma al tempo stesso è importante raggiungere prestazioni elevate. Ad esempio, nel caso di SNR basso il migliore algortimo risulta essere mPTSD che non presenta però una forte robustezza.

Le problematiche legate alla scelta di un algortimo di Spike Detection sono molteplici e dipendono da diversi fattori quali il livello di rumore presente nella registrazione e il tipo di elettrodo utilizzato.

Questo mio lavoro di Tesi rappresenta un piccolo ma importante passo che va nella direzione di sviluppare algoritmi sempre più performanti per future applicazioni di neuroingegneria e, specificatamente, nell'ambito delle neuroprotesi, un campo in rapidissima espansione e con altissimo potenziale di innovazione.

Bibliografia

Antal, A., Boros, K., Poreisz, C., Chaieb, L., Terney, D. & Paulus, W. (2008) Comparatively weak after-effects of transcranial alternating current stimulation (tACS) on cortical excitability in humans. *Brain Stimul*, **1**, 97-105.

Antonini, A., Moro, E., Godeiro, C. & Reichmann, H. (2018) Medical and surgical management of advanced Parkinson's disease. *Mov. Disord.*, **33**, 900-908.

Araki, T. & Otani, T. (1955) Response of single motoneurons to direct stimulation in toad's spinal cord. *J. Neurophysiol.*, **18**, 472-485.

Arlotti, M., Marceglia, S., Foffani, G., Volkmann, J., Lozano, A.M., Moro, E., Cogiamanian, F., Prenassi, M., Bocci, T., Cortese, F., Rampini, P., Barbieri, S. & Priori, A. (2018) Eight-hours adaptive deep brain stimulation in patients with Parkinson disease. *Neurology*, **90**, e971-e976.

Arlotti, M., Rossi, L., Rosa, M., Marceglia, S. & Priori, A. (2016) An external portable device for adaptive deep brain stimulation (aDBS) clinical research in advanced Parkinson's Disease. *Med Eng Phys*, **38**, 498-505.

Averna, A. (2019) Evaluating the impact of intracortical microstimulation on distant cortical brain regions for neuroprosthetic applications. University of Genova.

Averna, A., Guggenmos, D., Pasquale, V., Semprini, M., Nudo, R. & Chiappalone, M. (2018) Neuroengineering Tools For Studying The Effect Of Intracortical Microstimulation In Rodent Models. 2018 40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC). IEEE, City.

Averna, A., Pasquale, V., Murphy, M., Rogantin, M.P., Acker, G.V., Nudo, R.J., Chiappalone, M. & Guggenmos, D. (2019) Differential effects of open- and closed-loop intracortical microstimulation on firing patterns of neurons in distant cortical areas. *Neuroscience*.

Bachmann, L.C., Matis, A., Lindau, N.T., Felder, P., Gullo, M. & Schwab, M.E. (2013) Deep brain stimulation of the midbrain locomotor region improves paretic hindlimb function after spinal cord injury in rats. *Sci Transl Med*, **5**, 208ra146.

Barker, A.T., Jalinous, R. & Freeston, I.L. (1985) Non-invasive magnetic stimulation of human motor cortex. *Lancet*, **1**, 1106-1107.

Berger, T.W., Hampson, R.E., Song, D., Goonawardena, A., Marmarelis, V.Z. & Deadwyler, S.A. (2011) A cortical neural prosthesis for restoring and enhancing memory. *J Neural Eng*, **8**, 046017.

Berlim, M.T., McGirr, A., Van den Eynde, F., Fleck, M.P.A. & Giacobbe, P. (2014) Effectiveness and acceptability of deep brain stimulation (DBS) of the subgenual cingulate cortex for treatment-resistant depression: a systematic review and exploratory meta-analysis. *J Affect Disord*, **159**, 31-38.

Bloch, E., Da Cruz, L. & Hsu-Lin Luo, Y. (2019) Advances in retinal prosthesis systems. *Therapeutic Advances in Ophthalmology*.

Bologna, L.L., Pasquale, V., Garofalo, M., Gandolfo, M., Baljon, P.L., Maccione, A., Martinoia, S. & Chiappalone, M. (2010) Investigating neuronal activity by SPYCODE multi-channel data analyzer. *Neural Networks*, **23**, 685-697.

Bouton, C.E., Shaikhouni, A., Annetta, N.V., Bockbrader, M.A., Friedenberg, D.A., Nielson, D.M., Sharma, G., Sederberg, P.B., Glenn, B.C., Mysiw, W.J., Morgan, A.G., Deogaonkar, M. & Rezai, A.R. (2016) Restoring cortical control of functional movement in a human with quadriplegia. *Nature*, **533**, 247-250.

Brückner, S. & Kammer, T. (2017) Both anodal and cathodal transcranial direct current stimulation improves semantic processing. *Neuroscience*, **343**, 269-275.

Buccelli, S., Bornat, Y., Colombi, I., Ambroise, M., Martines, L., Pasquale, V., Bisio, M., Tessadori, J., Nowak, P., Grassia, F., Averna, A., Tedesco, M., Bonifazi, P., Difato, F., Massobrio, P., Levi, T. & Chiappalone, M. (2019) A Neuromorphic Prosthesis to Restore Communication in Neuronal Networks. *iScience*, **19**, 402-414.

Buzsáki, G., Anastassiou, C.A. & Koch, C. (2012) The origin of extracellular fields and currents--EEG, ECoG, LFP and spikes. *Nat. Rev. Neurosci.*, **13**, 407-420.

Carter, M. & Shieh, J. (2015) *Guide to Research Techniques in Neuroscience - 2nd Edition*. Academic Press.

Coenen, V.A., Sajonz, B., Reisert, M., Bostroem, J., Bewernick, B., Urbach, H., Jenkner, C., Reinacher, P.C., Schlaepfer, T.E. & Mädler, B. (2018) Tractography-assisted deep brain stimulation of the superolateral branch of the medial forebrain bundle (slMFB DBS) in major depression. *Neuroimage Clin*, **20**, 580-593.

Courtine, G. & Sofroniew, M.V. (2019) Spinal cord repair: advances in biology and technology. *Nat. Med.*, **25**, 898-908.

Cutrone, A. & Micera, S. (2019) Implantable Neural Interfaces and Wearable Tactile Systems for Bidirectional Neuroprosthetics Systems. *Adv Healthc Mater*, **8**, e1801345.

Dandekar, M.P., Fenoy, A.J., Carvalho, A.F., Soares, J.C. & Quevedo, J. (2018) Deep brain stimulation for treatment-resistant depression: an integrative review of preclinical and clinical findings and translational implications. *Mol. Psychiatry*, **23**, 1094-1112.

Dionísio, A., Duarte, I.C., Patrício, M. & Castelo-Branco, M. (2018) The Use of Repetitive Transcranial Magnetic Stimulation for Stroke Rehabilitation: A Systematic Review. *J Stroke Cerebrovasc Dis*, **27**, 1-31.

Donoghue, J.P. & Wise, S.P. (1982) The motor cortex of the rat: cytoarchitecture and microstimulation mapping. *J. Comp. Neurol.*, **212**, 76-88.

Doron, G. & Brecht, M. (2015) What single-cell stimulation has told us about neural coding. *Philos Trans R Soc Lond B Biol Sci*, **370**.

Durand, M.D. (2006) What is Neural Engineering? *J Neural Eng*.

Fetz, E.E. (1969) Operant conditioning of cortical unit activity. *Science*, **163**, 955-958.

Flesher, S.N., Collinger, J.L., Foldes, S.T., Weiss, J.M., Downey, J.E., Tyler-Kabara, E.C., Bensmaia, S.J., Schwartz, A.B., Boninger, M.L. & Gaunt, R.A. (2016) Intracortical microstimulation of human somatosensory cortex. *Sci Transl Med*, **8**, 361ra141.

Galvani, A. (1791) De viribus electricitatis in motu musculari commentarius.

Gerstein, G.L. & Mandelbrot, B. (1964) RANDOM WALK MODELS FOR THE SPIKE ACTIVITY OF A SINGLE NEURON. *Biophys. J.*, **4**, 41-68.

Guggenmos, D., Azin, M., Barbay, S., Mahnken, J.D., Dunham, C., Mohseni, P. & Nudo, R.J. (2013) Restoration of function after brain damage using a neural prosthesis. *Proceedings of the National Academy of Sciences of the United States of America*, **110**.

Habets, J.G.V., Heijmans, M., Kuijf, M.L., Janssen, M.L.F., Temel, Y. & Kubben, P.L. (2018) An update on adaptive deep brain stimulation in Parkinson's disease. *Mov. Disord.*, **33**, 1834-1843.

Hagen, E., Ness, T.V., Khosrowshahi, A., Sorensen, C., Fyhn, M., Hafting, T., Franke, F. & Einevoll, G.T. (2015) ViSAPy: a Python tool for biophysics-based generation of virtual spiking activity for evaluation of spike-sorting algorithms. *J Neurosci Methods*, **245**, 182-204.

Hampson, R.E., Song, D., Opris, I., Santos, L.M., Shin, D.C., Gerhardt, G.A., Marmarelis, V.Z., Berger, T.W. & Deadwyler, S.A. (2013) Facilitation of memory encoding in primate hippocampus by a neuroprosthesis that promotes task-specific neural firing. *J Neural Eng*, **10**, 066013.

Hampson, R.E., Song, D., Robinson, B.S., Fetterhoff, D., Dakos, A.S., Roeder, B.M., She, X., Wicks, R.T., Witcher, M.R., Couture, D.E., Laxton, A.W., Munger-Clary, H., Popli, G., Sollman, M.J., Whitlow, C.T., Marmarelis, V.Z., Berger, T.W. & Deadwyler, S.A. (2018) Developing a hippocampal neural prosthetic to facilitate human memory encoding and recall. *Journal of Neural Engineering*, **15**, 036014.

Hebb, D.O. (1950) The organization of behavior: A neuropsychological theory. . *Science Education*, **34**, 336-337.

Heck, C.N., King-Stephens, D., Massey, A.D., Nair, D.R., Jobst, B.C., Barkley, G.L., Salanova, V., Cole, A.J., Smith, M.C., Gwinn, R.P., Skidmore, C., Van Ness, P.C., Bergey, G.K., Park, Y.D., Miller, I., Geller, E., Rutecki, P.A., Zimmerman, R., Spencer, D.C., Goldman, A., Edwards, J.C., Leiphart, J.W., Wharen, R.E., Fessler, J., Fountain, N.B., Worrell, G.A., Gross, R.E., Eisenschenk, S., Duckrow, R.B., Hirsch, L.J., Bazil, C., O'Donovan, C.A., Sun, F.T., Courtney, T.A., Seale, C.G. & Morrell, M.J. (2014) Two-year seizure reduction in adults with medically intractable partial onset epilepsy treated with responsive neurostimulation: final results of the RNS System Pivotal trial. *Epilepsia*, **55**, 432-441.

Herreros, I., Giovannucci, A., Taub, A.H., Hogri, R., Magal, A., Bamford, S., Prueckl, R. & Verschure, P.F.M.J. (2014) A Cerebellar Neuroprosthetic System: Computational Architecture and in vivo Test. *Front Bioeng Biotechnol*, **2**.

Hodgkin, A.L. & Huxley, A.F. (1952) A quantitative description of membrane current and its application to conduction and excitation in nerve. *The Journal of Physiology*, **117**, 500-544.

Izhikevich, E.M. (2004) Which model to use for cortical spiking neurons? *IEEE Trans Neural Netw*, **15**, 1063-1070.

Jackson, A., Mavoori, J. & Fetz, E.E. (2006) Long-term motor cortex plasticity induced by an electronic neural implant. *Nature*, **444**, 56-60.

Jun, J.J., Steinmetz, N.A., Siegle, J.H., Denman, D.J., Bauza, M., Barbarits, B., Lee, A.K., Anastassiou, C.A., Andrei, A., Aydın, Ç., Barbic, M., Blanche, T.J., Bonin, V., Couto, J., Dutta, B., Gratiy, S.L., Gutnisky, D.A., Häusser, M., Karsh, B., Ledochowitsch, P., Lopez, C.M., Mitelut, C., Musa, S., Okun, M., Pachitariu, M., Putzeys, J., Rich, P.D., Rossant, C., Sun, W.-l., Svoboda, K., Carandini, M., Harris, K.D., Koch, C., O'Keefe, J. & Harris, T.D. (2017) Fully Integrated Silicon Probes for High-Density Recording of Neural Activity. *Nature*, **551**, 232-236.

Kaiser, J.F. (1990) On a simple algorithm to calculate the 'energy' of a signal. International Conference on Acoustics, Speech, and Signal Processing. City. p. 381-384 vol.381.

Kaiser, J.F. (1993) Some useful properties of Teager's energy operators. *Proceedings of the ICASSP'93*. IEEE Computer Society, City. p. 149-152.

Kandel, E., Jessell, T. & Schwartz, J.H. (1981) *Principi di Neuroscienze*

Kass, e.a. (2018) Computational Neuroscience: Mathematical and Statistical Perspectives. *Annual Review of Statistics and Its Application*.

Kass, R.E., Amari, S.-I., Arai, K., Brown, E.N., Diekman, C.O., Diesmann, M., Doiron, B., Eden, U.T., Fairhall, A.L., Fiddyment, G.M., Fukai, T., Grün, S., Harrison, M.T., Helias, M., Nakahara, H., Teramae, J.-n., Thomas, P.J., Reimers, M., Rodu, J., Rotstein, H.G., Shea-Brown, E., Shimazaki, H., Shinomoto, S., Yu, B.M. & Kramer, M.A. (2018) Computational Neuroscience: Mathematical and Statistical Perspectives. *Annual Review of Statistics and Its Application*, **5**, 183-214.

Kwon, C.-S., Jetté, N. & Ghatan, S. (2020) Perspectives on the current developments with neuromodulation for the treatment of epilepsy. *Expert Rev Neurother*, **20**, 189-194.

Laxpati, N.G., Kasoff, W.S. & Gross, R.E. (2014) Deep brain stimulation for the treatment of epilepsy: circuits, targets, and trials. *Neurotherapeutics*, **11**, 508-526.

Li, M.C.H. & Cook, M.J. (2018) Deep brain stimulation for drug-resistant epilepsy. *Epilepsia*, **59**, 273-290.

Lieb, F., Stark, H.G. & Thielemann, C. (2017) A stationary wavelet transform and a time-frequency based spike detection algorithm for extracellular recorded data. *J Neural Eng*, **14**, 036013.

Maccione, A., Gandolfo, M., Massobrio, P., Novellino, A., Martinoia, S. & Chiappalone, M. (2009) A novel algorithm for precise identification of spikes in extracellularly recorded neuronal signals. *Journal of Neuroscience Methods*, **177**, 241-249.

Malik, M.H., Saeed, M. & Kamboh, A.M. (2016) Automatic threshold optimization in nonlinear energy operator based spike detection. *Conference proceedings: ... Annual International Conference of the IEEE Engineering in Medicine and Biology Society. IEEE Engineering in Medicine and Biology Society. Annual Conference*, **2016**, 774-777.

Micera, S. (2016) Staying in Touch: Toward the Restoration of Sensory Feedback in Hand Prostheses Using Peripheral Neural Stimulation. *IEEE Pulse*, **7**, 16-19.

Mirochnik, R.M. & Pezaris, J.S. (2019) Contemporary approaches to visual prostheses. *Military Medical Research*, **6**, 19.

Mohammed, A., Bayford, R. & Demosthenous, A. (2018) Toward adaptive deep brain stimulation in Parkinson's disease: a review. *Neurodegener Dis Manag*, **8**, 115-136.

Nicolelis, M.A. (2001) Actions from thoughts. *Nature*, **409**, 403-407.

Nitsche, M.A. & Paulus, W. (2000) Excitability changes induced in the human motor cortex by weak transcranial direct current stimulation. *The Journal of Physiology*, **527**, 633-639.

Noury, N. & Siegel, M. (2017) Phase properties of transcranial electrical stimulation artifacts in electrophysiological recordings. *Neuroimage*, **158**, 406-416.

Panuccio, G., Semprini, M., Natale, L., Buccelli, S., Colombi, I. & Chiappalone, M. (2018) Progress in Neuroengineering for brain repair: New challenges and open issues. *Brain and Neuroscience Advances*, **2**, 2398212818776475.

Patterson, M.M. & Kesner, R.P. (1981) *Electrical Stimulation Research Techniques*. Elsevier.

Pipa, G., Grün, S. & van Vreeswijk, C. (2013) Impact of spike train autostructure on probability distribution of joint spike events. *Neural Comput*, **25**, 1123-1163.

Pipa, G., Van Vreeskwijk (2013) Impact of Spike Train Autostructure on Probability Distribution of Joint Spike Events. *Neural Comput*.

Poli, D., Pastore, V.P. & Massobrio, P. (2015) Functional connectivity in in vitro neuronal assemblies. *Front. Neural Circuits*, **9**.

Quiroga, R.Q., Nadasdy, Z. & Ben-Shaul, Y. (2004) Unsupervised spike detection and sorting with wavelets and superparamagnetic clustering. *Neural Computation*, **16**.

Reardon, S. (2014) Electroceuticals spark interest. *Nature*, **511**, 18.

Reato, D., Rahman, A., Bikson, M. & Parra, L.C. (2013) Effects of weak transcranial alternating current stimulation on brain activity—a review of known mechanisms from animal studies. *Front. Hum. Neurosci.*, **7**.

Santarnecchi, E., Polizzotto, N.R., Godone, M., Giovannelli, F., Feurra, M., Matzen, L., Rossi, A. & Rossi, S. (2013) Frequency-dependent enhancement of fluid intelligence induced by transcranial oscillatory potentials. *Curr. Biol.*, **23**, 1449-1453.

Schwartz, A.B. (2004) Cortical neural prosthetics. *Annu. Rev. Neurosci.*, **27**, 487-507.

Siebner, H.R., Hartwigsen, G., Kassuba, T. & Rothwell, J.C. (2009) How does transcranial magnetic stimulation modify neuronal activity in the brain? Implications for studies of cognition. *Cortex*, **45**, 1035-1042.

Smith, M.-C. & Stinear, C.M. (2016) Transcranial magnetic stimulation (TMS) in stroke: Ready for clinical practice? *J Clin Neurosci*, **31**, 10-14.

Stringer, C., Pachitariu, M., Steinmetz, N., Bai Reddy, C., Carandini, M. & Kenneth, D.H. (2018) Spontaneous behaviors drive multidimensional, brain-wide activity *bioRxiv*.

Swindale, N.V. & Spacek, M.A. (2015) Spike detection methods for polytrodes and high density microelectrode arrays. *J Comput Neurosci*, **38**, 249-261.

Tuckwell, H.C. (1988) *Introduction to Theoretical Neurobiology*. Cambridge University Press.

Tufail, Y., Matyushov, A., Baldwin, N., Tauchmann, M.L., Georges, J., Yoshihiro, A., Tillery, S.I.H. & Tyler, W.J. (2010) Transcranial pulsed ultrasound stimulates intact brain circuits. *Neuron*, **66**, 681-694.

Tyler, W.J., Lani, S.W. & Hwang, G.M. (2018) Ultrasonic modulation of neural circuit activity. *Curr Opin Neurobiol*, **50**, 222-231.

Wagner, F.B., Mignardot, J.-B., Le Goff-Mignardot, C.G., Demesmaeker, R., Komi, S., Capogrosso, M., Rowald, A., Seáñez, I., Caban, M., Pirondini, E., Vat, M., McCracken, L.A., Heimgartner, R., Fodor, I., Watrin, A., Seguin, P., Paoles, E., Van Den Keybus, K., Eberle, G., Schurch, B., Pralong, E., Becce, F., Prior, J., Buse, N., Buschman, R., Neufeld, E., Kuster, N., Carda, S., von Zitzewitz, J., Delattre, V., Denison, T., Lambert, H., Minassian, K., Bloch, J. & Courtine, G. (2018) Targeted neurotechnology restores walking in humans with spinal cord injury. *Nature*, **563**, 65-71.

Wang, Z., Song, W.-Q. & Wang, L. (2017) Application of noninvasive brain stimulation for post-stroke dysphagia rehabilitation. *Kaohsiung J. Med. Sci.*, **33**, 55-61.

WHO (2006) Stroke: a global response is needed.

WHO (2013) Spinal cord injury.

WHO (2019) Epilepsy.

Wouters, J., McDermott, H.J. & Francart, T. (2015) Sound Coding in Cochlear Implants: From electric pulses to hearing. *IEEE Signal Processing Magazine*.

Yang, Y. & Mason, A.J. (Year) Optimization of nonlinear energy operator based spike detection circuit for high density neural recordings. 2014 IEEE International Symposium on Circuits and Systems (ISCAS). City. p. 1396-1399.

Zanos, S., Rembado, I., Chen, D. & Fetz, E.E. (2018) Phase-Locked Stimulation during Cortical Beta Oscillations Produces Bidirectional Synaptic Plasticity in Awake Monkeys. *Curr. Biol.*, **28**, 2515-2526.e2514.

Zrenner, C., Desideri, D., Belardinelli, P. & Ziemann, U. (2018) Real-time EEG-defined excitability states determine efficacy of TMS-induced plasticity in human motor cortex. *Brain Stimul*, **11**, 374-389.

Zucca, S., D'Urso, G., Pasquale, V., Vecchia, D., Pica, G., Bovetti, S., Moretti, C., Varani, S., Molano-Mazon, M., Chiappalone, M., Panzeri, S. & Fellin, T. (2017) An inhibitory gate for state transition in cortex. *eLife*.