

UNIVERSITÀ DEGLI STUDI DI PADOVA  
FACOLTÀ DI INGEGNERIA  
DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE

Corso di Laurea Magistrale in Bioingegneria

**CGMLab: UNA GUI MATLAB PER  
L'ANALISI E L'ELABORAZIONE DI  
SEGNALI MISURATI DA SENSORI  
CONTINUI DELLA GLICEMIA**

**Relatore:** Prof. Giovanni Sparacino

**Correlatore:** Ing. Andrea Facchinetti

**Laureanda:** Denise De Zanet

Anno Accademico 2012-2013



*A mamma e papà,*

*che mi hanno cresciuta trasmettendomi il valore  
degli affetti e dei sacrifici, ponendo sempre  
la loro fiducia nella famiglia e nella vita*



# Indice

<b>Introduzione</b>	<b>1</b>
<b>1 Diabete e monitoraggio della glicemia mediante sensori continui</b>	<b>3</b>
1.1 Il diabete mellito . . . . .	3
1.1.1 I tipi di diabete . . . . .	4
1.1.2 Complicanze indotte dal diabete . . . . .	5
1.2 Self Monitoring of Blood Glucose (SMBG) . . . . .	6
1.3 Continuous Glucose Monitoring (CGM) . . . . .	8
1.3.1 Dispositivi minimamente invasivi . . . . .	8
1.3.2 Dispositivi non invasivi . . . . .	10
1.4 Conclusioni . . . . .	11
<b>2 Problemi del segnale CGM e margini di miglioramento</b>	<b>13</b>
2.1 Punti critici del segnale CGM . . . . .	13
2.1.1 Aspetti fisiologici legati al sito di misura . . . . .	13
2.1.2 Aspetti legati alla calibrazione . . . . .	14
2.1.3 Rumore di misura . . . . .	16
2.1.4 Predizione del livello glicemico futuro e generazione di allarmi . . . . .	18
2.2 Il concetto di Smart CGM (sCGM) Sensor . . . . .	18
2.3 Scopo della tesi . . . . .	20
<b>3 Progetto di Graphical User Interface (GUI) in MatLab</b>	<b>21</b>
3.1 Aspetti generali . . . . .	21
3.2 Fasi della progettazione . . . . .	22
3.3 Componenti grafici inseribili in una GUI . . . . .	23
3.4 Programmazione e gestione delle callbacks . . . . .	26
3.4.1 Due approcci realizzativi: GUIDE e costruzione programmata . . . . .	27
<b>4 CGMLab: una GUI MatLab per l'analisi e l'elaborazione di segnali continui di glicemia</b>	<b>31</b>
4.1 Progettazione del modulo per il pre-processing e l'inserimento dei dati . . . . .	33

4.1.1	Layout: <i>CGM Data Selection</i> (pannello 1) . . . . .	34
4.1.2	Layout: <i>Other Data Types</i> (pannello 2) . . . . .	35
4.1.3	Layout: <i>Dataset Available</i> (pannello 3) . . . . .	36
4.2	Dettagli implementativi del modulo per il pre-processing e l'inserimento dei dati . . . . .	37
4.2.1	Dati e callbacks: <i>Continuous Glucose Monitoring Data</i> (pannello 1) . . . . .	40
4.2.2	Dati e callbacks: <i>Other Data Types</i> (pannello 2) . . . . .	44
4.2.3	Dati e callbacks: <i>Dataset Available</i> (pannello 3) . . . . .	46
<b>5</b>	<b>Modulo di CGMLab per lo smoothing</b>	<b>51</b>
5.1	Filtraggio stocastico con filtro di Kalman . . . . .	51
5.1.1	Richiami teorici . . . . .	51
5.1.2	Descrizione statistica di $u(t)$ . . . . .	55
5.1.3	Stima delle matrici $Q$ ed $R$ e criteri di regolarizzazione . . . . .	56
5.2	Implementazione dello smoothing e variabilità del SNR . . . . .	58
5.3	Aspetti numerici: diagonalizzazione . . . . .	60
5.4	Progettazione del modulo per lo smoothing . . . . .	62
5.4.1	Layout: <i>Regularization Settings</i> e <i>Regularization Criteria</i> (pannello 1 e 2) . . . . .	63
5.4.2	Layout: <i>Smoothing Button</i> and <i>Estimated Parameters</i> (pannello 3) . . . . .	64
5.4.3	Layout: <i>Display Data and Results</i> (pannello 4) . . . . .	65
5.5	Dettagli implementativi del modulo per lo smoothing . . . . .	66
5.5.1	Dati e callbacks: <i>Regularization Settings</i> e <i>Regularization Criteria</i> (pannello 1 e 2) . . . . .	69
5.5.2	Dati e callbacks: <i>Smoothing Button</i> and <i>Estimated Parameters</i> (pannello 3) . . . . .	73
5.5.3	Dati e callbacks: <i>Display Data and Results</i> (pannello 4) . . . . .	74
<b>6</b>	<b>Modulo di CGMLab per la predizione</b>	<b>79</b>
6.1	Predizione della glicemia . . . . .	79
6.2	Algoritmi di predizione di letteratura . . . . .	80
6.2.1	Modelli Auto-Regressivi (AR) . . . . .	81
6.2.2	Modelli polinomiali . . . . .	83
6.2.3	Forgetting Factor ( $\mu$ ) e Prediction Horizon (PH) . . . . .	85
6.2.4	Predizione con filtro di Kalman . . . . .	87
6.3	Indici quantitativi per la valutazione di un predittore . . . . .	87
6.4	Progettazione del modulo per la predizione . . . . .	89
6.4.1	Layout: <i>Prediction Settings</i> (pannello 1) . . . . .	90
6.4.2	Layout: <i>Display Data and Results</i> (pannello 2) . . . . .	90
6.5	Dettagli implementativi del modulo per la predizione . . . . .	91
6.5.1	Dati e callbacks: <i>Prediction Settings</i> (pannello 1) . . . . .	99
6.5.2	Dati e callbacks: <i>Display Data and Results</i> (pannello 2) . . . . .	101

*INDICE*

vii

**Conclusioni**

**105**





# Introduzione

Il diabete è una malattia cronica caratterizzata da una persistente instabilità del livello glicemico del sangue, dovuta all'impossibilità da parte del pancreas di produrre insulina (diabete di tipo 1) o ad un malfunzionamento sia nella secrezione insulinica che nell'azione che essa svolge (diabete di tipo 2), con conseguenti possibili complicanze a lungo termine (dovute soprattutto a eventi di iperglicemia) e a breve termine (legate a eventi ipoglicemici). Negli ultimi anni si è registrato un incremento preoccupante di incidenza della patologia e l'Organizzazione Mondiale della Sanità (OMS) prevede che entro il 2030 i pazienti diabetici nel mondo supereranno i 360 milioni. La crescita della popolazione, il suo invecchiamento, e soprattutto sedentarietà, stress e cattive abitudini alimentari sembrano essere le cause principali del diabete, che rappresenta oggi un vero e proprio problema sociale ed economico, oltre che sanitario.

Attualmente la terapia più comune per il trattamento del diabete consiste in iniezioni sottocutanee periodiche di insulina, accompagnate da un auto-monitoraggio della glicemia (Self Monitoring of Blood Glucose, SMBG) che consiste in 3-4 misurazioni invasive pungi-dito al giorno che consentono di aggiustare e regolare il trattamento. Tuttavia con quest'approccio non si è in grado di identificare eventuali episodi d'ipoglicemia o iperglicemia che possono verificarsi tra una misurazione e l'altra.

Per superare questi limiti, dall'inizio del XXI secolo sono stati studiati e sviluppati dispositivi in grado di monitorare con (quasi) continuità la concentrazione di glucosio, ovvero con misurazioni ogni 1-5 minuti. Il monitoraggio effettuato tramite dispositivi di questo tipo prende il nome di Continuous Glucose Monitoring (CGM).

Come verrà descritto nel Capitolo 1, i dispositivi CGM possono essere considerati un efficace strumento per il miglioramento della terapia del diabete. Un monitoraggio quasi continuo, per lo più minimamente invasivo, consente di individuare un numero maggiore di episodi pericolosi rispetto al convenzionale SMBG. Le applicazioni basate sull'utilizzo dei dati CGM sono diverse. Per esempio, le serie temporali di dati CGM possono essere analizzate retrospettivamente per modificare la gestione della terapia del paziente. In tempo reale, invece, la grande quantità di informazioni ottenuta da questi dispositivi può essere sfruttata per individuare, eventualmente anche con anticipo, episodi critici di ipoglicemia o iperglicemia, generando allarmi. Tuttavia, i dispositivi

CGM non dispongono ancora dell'accuratezza e della affidabilità richiesta per poter sfruttare al meglio tutte queste potenzialità. In particolare, l'accuratezza e la precisione delle stime della concentrazione di glucosio e la tempestività nella generazione degli allarmi sono aspetti critici che devono essere migliorati.

Come discusso nel Capitolo 2, da diversi anni il Gruppo di Ricerca di Bioingegneria del Dipartimento di Ingegneria dell'Informazione (DEI) dell'Università di Padova sta sviluppando algoritmi per l'analisi e l'elaborazione dei dati CGM al fine di migliorare gli aspetti critici sopraelencati. Nello specifico, l'attività di ricerca si è focalizzata sullo sviluppo di quattro algoritmi, funzionanti in tempo reale e in cascata all'output fornito dal sensore CGM, atti a incrementare il rapporto segnale-disturbo sul dato CGM (smoothing), a migliorare l'accuratezza puntuale del valore misurato (calibration), a predire il valore futuro della concentrazione di glucosio (prediction) e, infine, a generare allarmi di ipo e iperglicemia (alert generation). Sinora, però, l'uso di questi algoritmi al di fuori del DEI è stato limitato dalla non disponibilità di software diversi da prototipi MatLab per solo uso interno. L'obiettivo di questo lavoro è quello di realizzare un'interfaccia grafica in MatLab che consenta l'automatica applicazione degli algoritmi suddetti in modo immediato. Una GUI (Graphical User Interface), infatti, oltre a costituire una piattaforma di lavoro per ricercatori esperti, che in questo modo non si vedono costretti a reimplementare ogni volta algoritmi complessi, costituisce l'unico canale di comunicazione tra queste figure professionali e altri utenti, siano essi medici del metabolismo o utenti medi non esperti di programmazione.

Nel Capitolo 4 viene presentata l'interfaccia realizzata, denominata CGM-Lab (Continuous Glucose Monitoring Laboratory), composta da due finestre principali, una per il caricamento e il pre-processing dei dati e una dedicata alla loro elaborazione. Il software consente di inserire in automatico serie di dati in formati e modalità differenti, attraverso un controllo robusto dei data missing, dei dati oltre soglia e della periodicità del segnale. Inoltre è possibile caricare e visualizzare contemporaneamente anche dati di calibrazione, campioni plasmatici e altri tipi di dati che l'utente può voler inserire.

Il Capitolo 5 e il Capitolo 6 presentano la seconda finestra grafica, dedicata alla scelta del modulo di processamento su cui lavorare, in particolare del modulo di smoothing o di quello di predizione del segnale CGM. Il filtraggio è svolto mediante filtro di Kalman, tenendo in considerazione la variabilità del rapporto segnale-rumore presente nei dati in esame e consentendo la scelta del criterio di stima parametrica da applicare. Per quanto riguarda, invece, la predizione, questa può essere orientata verso l'impiego di modelli polinomiali, autoregressivi o predittori di Kalman e dà la possibilità di confrontare fino a due profili predittivi contemporaneamente, senza nessuno sforzo di ricodifica degli algoritmi da parte dell'utente.

Infine si traggono alcune conclusioni e si discutono alcuni possibili sviluppi futuri di CGMLab.

# Capitolo 1

## Diabete e monitoraggio della glicemia mediante sensori continui

### 1.1 Il diabete mellito

Il diabete mellito è una patologia cronica del metabolismo (cioè del processo che l'organismo utilizza per ricavare dagli alimenti l'energia e le sostanze di cui ha bisogno) dovuta all'incapacità del pancreas di produrre sufficienti quantità di insulina o ad un malfunzionamento sia nella secrezione insulinica che nell'azione che essa svolge. Come conseguenza, la concentrazione di glucosio nel sangue aumenta al di sopra dei livelli normali (180-200 mg/dl), portando a complicanze dannose per l'organismo, sia a breve che a lungo termine. Secondo l'Organizzazione Mondiale della Sanità (OMS) sono circa 250 milioni le persone affette da diabete in tutto il mondo, di cui il 50% rappresenta casi non diagnosticati, e più dell'80% delle morti correlate a questa patologia avvengono in paesi a basso e medio reddito. L'OMS e la International Diabetes Federation (IDF) stimano inoltre che i numeri sono destinati a raddoppiare, fino a superare i 360 milioni di soggetti affetti da diabete entro il 2030 [1] [2]. Un aumento di queste proporzioni è dovuto in parte all'invecchiamento generale della popolazione, ma soprattutto alla diffusione di condizioni a rischio come sovrappeso e obesità, scorretta alimentazione e sedentarietà. Tenendo presente, inoltre, che tali dati rappresentano una sottostima del fenomeno, considerando i casi non diagnosticati, questa panoramica introduttiva evidenzia come il diabete stia assumendo proporzioni epidemiche, rappresentando un problema di interesse mondiale sia dal punto di vista economico che dal punto di vista sociale. La Figura 1.1 illustra la diffusione del diabete nel mondo, confermando quanto appena affermato. Durante la digestione, gli zuccheri presenti negli alimenti vengono trasformati in glucosio, che rappresenta la principale fonte di energia dell'organismo. Il glucosio entra nel sangue e, grazie all'azione insulinica, passa nelle cellule per fornire energia o essere ulteriormente trasformato in

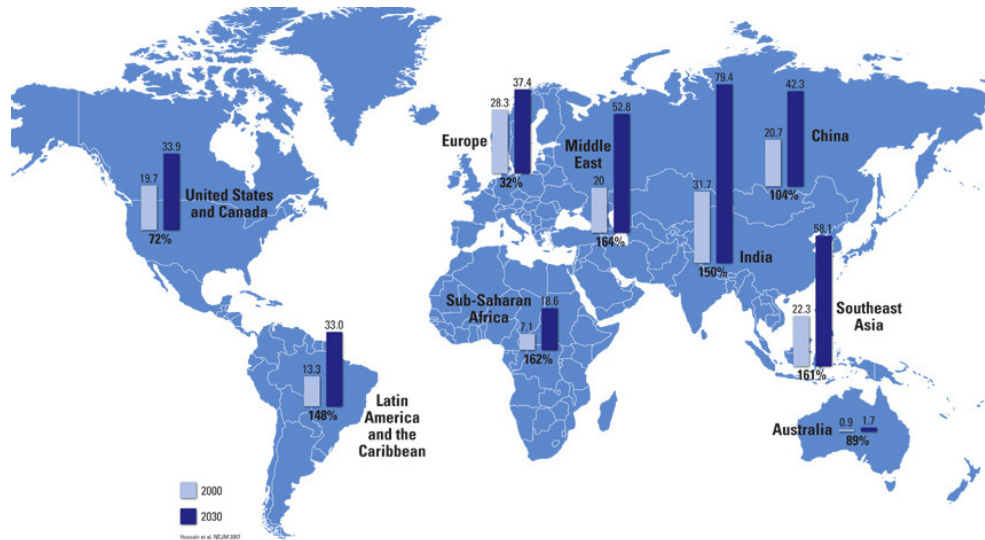


Figura 1.1: Panoramica mondiale sulla diffusione del diabete [3].

altre sostanze. L'insulina è un ormone prodotto dalle cellule  $\beta$  del pancreas, una ghiandola posta dietro allo stomaco. Nelle persone affette da diabete il pancreas produce una quantità insufficiente di insulina oppure le cellule non rispondono all'insulina prodotta dal pancreas, fenomeno noto con il nome di insulino-resistenza. L'organismo perde così la sua principale fonte di energia, perché il glucosio viene eliminato con l'urina. Vi sono due tipi principali di diabete: il diabete di tipo 1 e il diabete di tipo 2 [5] [6].

### 1.1.1 I tipi di diabete

Il diabete di tipo 1, conosciuto anche come diabete insulino-dipendente, si sviluppa comunemente nell'infanzia o nell'adolescenza e riguarda circa il 10% delle persone affette da diabete. In questo tipo di diabete, considerato una malattia autoimmune, il livello di insulina secreta è insufficiente perché il sistema immunitario, alterato, attacca e distrugge progressivamente le cellule  $\beta$  del pancreas. L'organismo dei pazienti affetti da diabete di tipo 1 ha bisogno perciò di ricevere dall'esterno l'insulina necessaria per un corretto funzionamento del metabolismo del glucosio, attraverso iniezioni sottocutanee periodiche. Non sono ancora conosciute le cause scatenanti del diabete insulino-dipendente, ma è stata dimostrata la presenza nel sangue di anticorpi diretti contro antigeni presenti a livello delle cellule pancreatiche che producono insulina. In generale si ritiene che siano coinvolti fattori genetici, ambientali e virali. I sintomi si manifestano in un breve arco di tempo, anche se la distruzione delle cellule  $\beta$  può essere iniziata molto prima. Essi comprendono aumentata produzione di urina (poliuria), sete intensa (polidipsia), fame, stanchezza profonda (astenia), perdita di peso. Se non diagnosticato e non trattato con insulina, il diabete di tipo 1 può portare al coma e alla morte.

Il diabete di tipo 2 è la forma più comune del diabete e comprende circa il 90% dei casi della patologia. In questo tipo di diabete, detto anche insulino-indipendente, le cellule pancreatiche sono intatte e producono insulina, ma spesso quest'ultima non è prodotta in quantità sufficiente oppure l'organismo non riesce ad utilizzarla in maniera efficiente (insulino-resistenza), comportando un accumulo di glucosio nella circolazione sanguigna. La sintomatologia del diabete di tipo 2 si sviluppa gradualmente e appare simile, anche se più sfumata, a quella del diabete insulino-dipendente, ma non necessita di dosi quotidiane di insulina. Il rischio di sviluppare la malattia aumenta con l'età ed è correlato a obesità, sedentarietà, fattori etnici e genetici.

### 1.1.2 Complicanze indotte dal diabete

Il diabete rappresenta una delle patologie a più alto rischio di morte e di disabilità. La pericolosità di questa patologia non deriva soltanto dalle disfunzioni metaboliche che essa comporta, ma anche e soprattutto dalle conseguenze che può avere, nel breve e nel lungo periodo, sull'intero organismo. Se non controllato e trattato in maniera adeguata, il diabete può determinare complicanze acute o a breve termine, associate soprattutto agli eventi di ipoglicemia, oppure croniche o a lungo termine, associate invece alle escursioni iperglicemiche [4].

Se il livello di glucosio nel sangue scende al di sotto della soglia critica di 70 mg/dl si parla di ipoglicemia. Nei pazienti diabetici gli episodi di ipoglicemia sono meno frequenti rispetto a quelli di iperglicemia, ma la carenza dell'unica fonte di energia per il sistema nervoso può portare alla perdita dei sensi, seguita in certi casi da coma chetoacidotico, gravi alterazioni ematiche e morte.

Se la concentrazione di glucosio nella circolazione supera invece i 180 mg/dl si parla di iperglicemia. Gli episodi di iperglicemia sono più frequenti di quelli di ipoglicemia e, in alcuni pazienti, lo stato iperglicemico si mantiene per gran parte della giornata. Le complicanze croniche colpiscono organi e tessuti, tra cui gli occhi, i reni, il cuore, i vasi sanguigni e i nervi periferici. La retinopatia diabetica è un danno a carico dei piccoli vasi sanguigni che irrorano la retina e può portare alla perdita della vista. I pazienti diabetici possono inoltre sviluppare con maggiore probabilità malattie oculari come glaucoma e cataratta. Si parla di nefropatia diabetica in corrispondenza di una riduzione progressiva della funzione di filtro del rene che, se non trattata, può condurre all'insufficienza renale, con conseguente necessità di dialisi o di trapianto del rene. Il rischio di malattie cardiovascolari è da 2 a 4 volte più alto nelle persone con diabete che nel resto della popolazione e le complicanze che colpiscono, in generale, il cuore e il sistema cardiovascolare costituiscono oltre il 50% delle morti per diabete. Una delle complicanze più frequenti è rappresentata dalla neuropatia diabetica e secondo l'OMS essa si manifesta a livelli diversi nel 50% dei diabetici [5]. Può comportare perdita di sensibilità, dolore e danni agli arti, disfunzioni del cuore, degli occhi, dello stomaco ed è una delle principali cause

di impotenza maschile. Inoltre le modificazioni della struttura dei vasi sanguigni e dei nervi possono causare ulcerazioni e problemi agli arti inferiori, in modo particolare ai piedi, sede di carichi e sforzi. Questo può talvolta rendere necessaria l'amputazione degli arti e statisticamente costituisce la prima causa di amputazione degli arti inferiori di origine non traumatica. Le complicanze croniche del diabete possono essere prevenute, o se ne può rallentare la progressione, attraverso uno stretto controllo di tutti i fattori di rischio correlati (emoglobina glicata, pressione sanguigna, lipidi nel sangue).

## 1.2 Self Monitoring of Blood Glucose (SMBG)

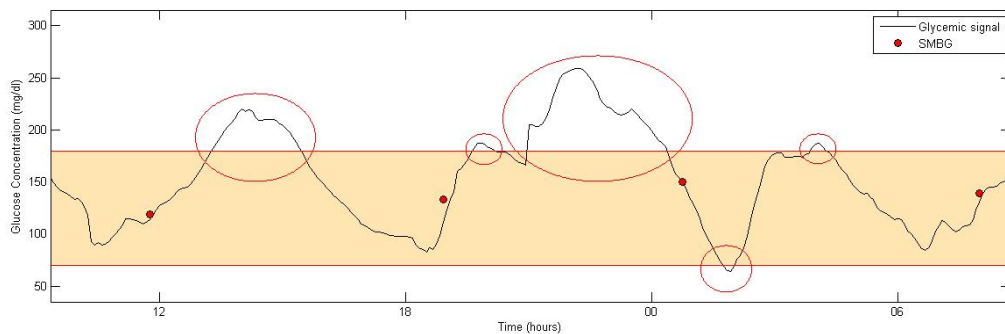
Il trattamento terapeutico per i pazienti diabetici si basa sostanzialmente su iniezioni periodiche di insulina, sul controllo del regime alimentare e su un esercizio fisico equilibrato e mirato. L'obiettivo di tale terapia è quello di mantenere la concentrazione del glucosio nel sangue il più possibile all'interno del range euglicemico di normalità, così da ridurre i rischi di ulteriori patologie.

Nella terapia convenzionale il dosaggio di insulina viene regolato in relazione a 3-4 misurazioni di glicemia effettuate giornalmente e autonomamente dal paziente. Questo tipo di controllo prende il nome di Self Monitoring of Blood Glucose (SMBG) e permette di ottenere misure dirette, discrete e accurate della concentrazione del glucosio nel sangue. I dispositivi utilizzati sono costituiti da un pungidito che permette il prelievo del sangue dal polpastrello e la misura del valore glicemico corrispondente, generalmente espressa in mg/dl. L'analisi SMBG risulta utile al fine di valutare l'efficacia e l'adeguatezza del protocollo terapeutico in atto. L'automonitoraggio consente, infatti, di avere conferma immediata di eventuali stati di iperglicemia o di ipoglicemia e di prendere le relative contromisure. Il paziente, attraverso misurazioni periodiche della propria glicemia, riesce a gestire autonomamente la malattia. In Figura 1.2 è mostrato un dispositivo per l'automonitoraggio della glicemia. Il FreeStyle InsuLinx<sup>®</sup> della Abbott Diabetes Care è dotato anche di un software di supporto per l'auto-gestione del diabete, il quale fornisce un riepilogo e una tabella dei livelli glicemici giornalieri e delle dosi di insulina effettuate, confronta le concentrazioni glicemiche dei diversi momenti della giornata e mostra le impostazioni correnti del misuratore e del calcolatore di bolo. Questi strumenti hanno permesso, fin dagli anni '80, un notevole passo avanti nella terapia del diabete mellito. Tuttavia, presentando importanti limitazioni, essi non possono essere considerati come la soluzione definitiva nel trattamento della patologia. Data l'invasività dei dispositivi SMBG sarebbe eccessivamente lesivo per il paziente misurare con più frequenza la concentrazione di glucosio nel sangue, ma 3-4 misure al giorno risultano, d'altra parte, insufficienti a seguire le variazioni glicemiche che si manifestano durante la giornata in relazione alla attività fisica, alla dieta e alla terapia insulinica. Ciò comporta l'impossibilità di pre-



**Figura 1.2:** Dispositivo per il SMBG: FreeStyle InsuLinx<sup>®</sup>, Abbott Diabetes Care [7].

venire ed evitare eventi critici, come ad esempio iperglicemie post-prandiali e ipoglicemie notturne, con il conseguente rischio di sviluppare complicanze a breve o a lungo termine. Un esempio di queste limitazioni è riportato in Figura 1.3. L'intervallo colorato tra le due soglie (fascia euglicemica) rappresenta il range entro cui dovrebbe spaziare il segnale glicemico ideale, non patologico. Come è evidente, le 4 concentrazioni acquisite (in rosso) non consentono di monitorare sufficientemente l'andamento della glicemia, la quale varia molto di più di ciò che si potrebbe pensare osservando le sole 4 misure considerate.



**Figura 1.3:** Confronto tra profilo glicemico (linea nera) e misurazioni SMBG (pallini rossi). La fascia colorata rappresenta il range di normalità.

Con il solo utilizzo di dispositivi SMBG, non è possibile, quindi, gestire e controllare in maniera sicura e completa la patologia diabetica. Appare evidente la necessità di monitorare la glicemia con una frequenza di acquisizione la più elevata possibile, compatibilmente con l'invasività degli strumenti a disposizione. Fino a qualche anno fa l'unica soluzione a questa necessità si concretizzava con un campionamento endovenoso ospedaliero più frequente, fino a 15 minuti di intervallo tra un campione e il successivo. Tuttavia questa procedura è fortemente invasiva e il suo utilizzo è consentito solo durante brevi ricoveri ospedalieri nell'ambito di esperimenti scientifici. Il superamento di questi ostacoli tecnologici ha costituito l'obiettivo che ha portato alla realizzazione, a partire dagli inizi del ventunesimo secolo, di nuovi dispositi-

vi in grado di monitorare in modo quasi continuo e minimamente invasivo la concentrazione di glucosio durante l'arco della giornata.

## 1.3 Continuous Glucose Monitoring (CGM)

Il Continuous Glucose Monitoring (CGM) può essere considerato oggi come un importante strumento in grado di migliorare il controllo glicemico nei pazienti in trattamento insulinico. Questa importante innovazione tecnologica è stata inizialmente introdotta nella pratica clinica come uno strumento di analisi retrospettiva a breve termine rivolto essenzialmente al personale medico coinvolto nella gestione e nel trattamento dei pazienti diabetici. L'evoluzione della tecnologia è progredita negli ultimi anni fino all'approvazione di sistemi di rilevamento del glucosio in grado di essere utilizzati in tempo reale, per prolungati periodi di tempo, in maniera prospettica. Analogamente, anche le pubblicazioni scientifiche degli ultimi anni sembrano rispecchiare l'interesse e l'importanza crescente su questo argomento.

I sistemi CGM sono pensati per ridurre al minimo l'invasività e sono in generale trasportabili, agevolando il paziente nel loro utilizzo quotidiano. Negli ultimi anni sono stati ideati e realizzati molti dispositivi per il monitoraggio continuo del glucosio e la maggior parte di questi è già stata approvata e commercializzata. Dal punto di vista tecnologico e operativo, i dispositivi CGM possono essere suddivisi in due categorie: sistemi minimamente invasivi e sistemi non invasivi.

### 1.3.1 Dispositivi minimamente invasivi

I dispositivi minimamente invasivi devono il loro nome al fatto che la misura della concentrazione di glucosio è attuata non direttamente nel sangue, bensì nel fluido interstiziale, cioè lo spazio tra capillari sanguigni e cellule, dove il sensore è impiantato e dove diffondono il glucosio e le altre sostanze nutritive. Questi sistemi CGM consistono in genere di tre componenti, la cui comunicazione è solitamente di tipo wireless:

- un sensore, solitamente impiantato nella sottocute dell'addome o del braccio, contenente un enzima che reagisce con il glucosio nel fluido interstiziale producendo un segnale proporzionale alla concentrazione del glucosio nell'interstizio (solitamente esprimibile in mV o mA);
- un trasmettitore, variabile in forma e dimensione, che misura in tempo reale il segnale corrispondente alla reazione chimica;
- un ricevitore, dotato di memoria e schermo, che riceve le informazioni dal trasmettitore e ne permette la visualizzazione.

Molti strumenti di questo tipo possono essere collegati ad un sistema di acquisizione e monitoraggio remoto, dove i dati vengono memorizzati per successive



analisi e possono essere utilizzati anche per la ricerca. Per ottenere un sistema del tutto automatizzato, oltre al monitoraggio del glucosio in continua, è possibile in alcuni casi combinare alla strumentazione descritta anche una Continuous Subcutaneous Insulin Infusion (CSII) pump, che agisce in catena chiusa a seconda dei valori di glicemia predetti. Si parla in questo caso di pancreas artificiale.

I principali dispositivi minimamente invasivi per il Continuous Glucose Monitoring presenti nel mercato, commercializzati da diverse compagnie e sviluppati con tecnologie differenti, sono:

- Guardian<sup>®</sup> REAL-Time CGM System (Medtronic MiniMed, Northridge, CA), che ha ricevuto l'approvazione della Food and Drug Administration (FDA) nel giugno del 2005; tutti i dispositivi Medtronic utilizzano lo stesso tipo di sensore, con registrazioni ogni 10 secondi e trasmissione al monitor del valore medio di queste ultime ogni 5 minuti, fino a 3 giorni di acquisizione continua [12];
- Paradigm<sup>®</sup> REAL-Time REVEL<sup>TM</sup> System (Medtronic MiniMed, Northridge, CA), che ha ricevuto l'approvazione FDA nel marzo 2007; appartiene ai sistemi CGM di seconda generazione [12];
- FreeStyle Navigator<sup>®</sup> CGM System (Abbott Diabetes Care, Alameda, CA), che ha ricevuto l'approvazione della FDA nel marzo del 2008; il dispositivo è dotato di un sensore elettrochimico che trasmette l'informazione relativa alla variazione di glucosio con una frequenza di un valore al minuto, fino ad un massimo di 5 giorni consecutivi [8];
- Dexcom SEVEN<sup>®</sup> Plus CGM System (Dexcom, San Diego, CA), che ha ricevuto l'approvazione FDA nel febbraio 2009 e il marchio CE nel novembre dello stesso anno; appartiene ai dispositivi CGM di terza generazione, la lettura del glucosio interstiziale viene trasmessa ogni 5 minuti al monitor e il sensore sottocutaneo può essere indossato fino a un massimo di 7 giorni [9];
- Dexcom G4 Platinum CGM System (Dexcom, San Diego, CA), che ha ricevuto l'approvazione FDA nell'ottobre del 2012; dotato di sensore ultrasottile, appartiene ai dispositivi CGM di quarta generazione e misura il glucosio a intervalli di 5 minuti durante il giorno, fino ad un massimo di 7 giorni consecutivi, con un'accuratezza fino al 20% maggiore rispetto al SEVEN<sup>®</sup> Plus [9];
- GlucoDay<sup>®</sup> System (Menarini Diagnostics, Florence, Italy), che ha ricevuto il marchio CE in Europa e che monitora il glucosio con una frequenza di acquisizione di un campione ogni 3 minuti; è il primo sistema che utilizza la microdialisi per monitorare la glicemia [11].

In Figura 1.4 e 1.5 si riportano le immagini di alcune delle tecnologie presentate.



**Figura 1.4:** *Paradigm<sup>®</sup> REAL-Time REVEL<sup>™</sup> System, Medtronic MiniMed, Northridge, CA [12].*



**Figura 1.5:** *Dexcom SEVEN<sup>®</sup> Plus CGM System e Dexcom G4 Platinum CGM System, Dexcom, San Diego, CA [9].*

La maggior parte dei dispositivi attualmente in commercio permette una più completa gestione del segnale glicemico grazie all'ausilio di un software accessorio, specifico per ciascuna tecnologia, che consente il trasferimento delle misure registrate al computer al fine di tenere memoria, aggiornare o analizzare statisticamente l'andamento nel tempo del livello di glucosio. Si citano, a scopo conoscitivo, il Dexcom Studio Data Management Software (Dexcom) e il CareLink Personal Software (Medtronic).

### 1.3.2 Dispositivi non invasivi

I dispositivi non invasivi rappresentano un progresso dal punto di vista sia tecnologico che clinico e un'alternativa per i pazienti diabetici, poiché non è richiesto un campionamento sanguigno. La maggior parte di questi sistemi è ancora oggi in fase di studio e di sviluppo, soprattutto per quanto riguarda la qualità della misura. A differenza dei dispositivi minimamente invasivi, questi utilizzano metodi di misura transcutanei per acquisire l'informazione, e solitamente si basano sullo studio di liquidi corporei che non siano il sangue e il fluido interstiziale, come ad esempio l'urina, la saliva e il fluido oculare. Le tecniche impiegate includono la spettroscopia a raggi infrarossi, la spettroscopia a fluorescenza, la spettroscopia di Raman, la spettroscopia fotoacustica, la

polarimetria ottica, tecniche di risonanza superficiale e altri metodi ottici [13] [14] [15].

## 1.4 Conclusioni

I dispositivi CGM devono il loro successo alla portabilità e alla minima invasività, nonché alla capacità di fornire un'informazione completa e dettagliata sul segnale glicemico, dando la possibilità al medico e al paziente sia di monitorarne in tempo reale il profilo temporale sia di analizzarne retrospettivamente l'andamento. Di conseguenza, ciò permette di adeguare l'assunzione di cibo e farmaci, l'attività fisica e i livelli insulinici di iniezione al fine di evitare il raggiungimento di stati di iperglicemia e ipoglicemia. Un ulteriore vantaggio dei sistemi CGM è dato dalla presenza, nella maggior parte di questi dispositivi, di una funzionalità in grado di generare allarmi acustici e visivi nel momento in cui la concentrazione di glucosio oltrepassa le soglie del range euglicemico.

Tuttavia, sebbene i progressi e il miglioramento apportati da questi strumenti nella gestione e nel controllo del diabete siano evidenti, i dispositivi CGM non sono ancora stati approvati per sostituire i dispositivi SMBG nella misura della glicemia in terapia convenzionale in quanto sono ancora presenti alcuni problemi legati all'accuratezza e precisione delle misure [24], discussi nel Capitolo 2 che segue.



# Capitolo 2

## Problemi del segnale CGM e margini di miglioramento

### 2.1 Punti critici del segnale CGM

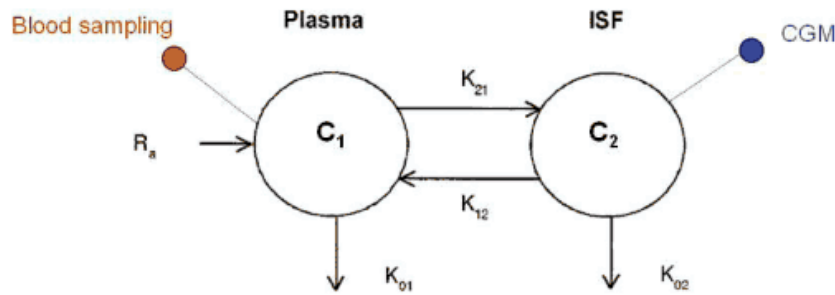
#### 2.1.1 Aspetti fisiologici legati al sito di misura

Per problemi di invasività del processo di misura, i sensori CGM non possono misurare direttamente la concentrazione di glucosio nel plasma. Il sensore, quindi, effettua delle acquisizioni dall'interstizio, ma ciò che si ottiene è la misura di una corrente elettrica, non della glicemia. Nella generazione del segnale CGM devono quindi essere presi in considerazione due aspetti: è necessario convertire la corrente in glicemia usando dei valori di riferimento SMBG e in tale processo di conversione bisogna tener conto della presenza di una dinamica dovuta alla provenienza delle misure da siti differenti. A causa di questo processo di misura e trasduzione, il segnale in uscita dal sensore CGM risulta essere una versione distorta del segnale di concentrazione di glucosio plasmatico. In particolare, si possono riscontrare una distorsione dovuta al processo di diffusione del glucosio e una dovuta ad un deterioramento della calibrazione.

Il processo di diffusione del glucosio tra sangue e interstizio può essere descritto tramite il modello bicompartimentale illustrato in Figura 2.1, in cui il compartimento sinistro e quello destro rappresentano rispettivamente le concentrazioni plasmatica e interstiziale di glucosio,  $R_a$  rappresenta la velocità di comparsa del glucosio e ogni  $K_{ij}$  rappresenta la costante di diffusione verso il compartimento  $i$ -esimo, dal compartimento  $j$ -esimo. La dinamica del glucosio interstiziale può essere descritta tramite l'equazione differenziale del primo ordine

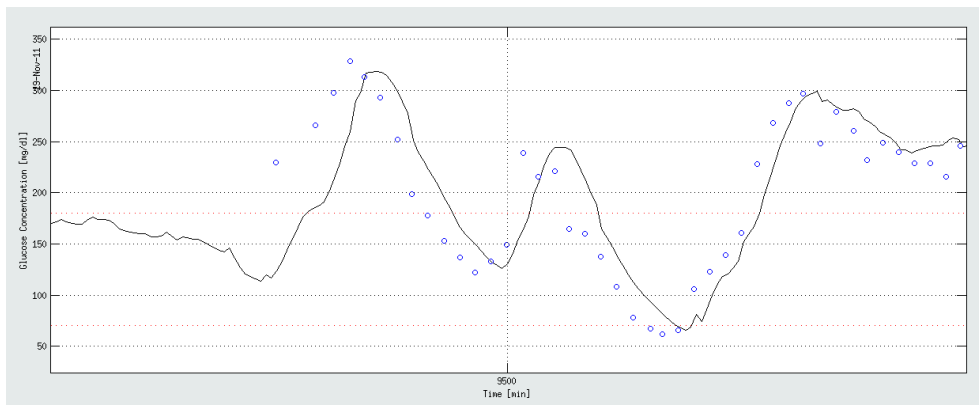
$$I\dot{G}(t) = -\frac{1}{\tau}IG(t) + \frac{g}{\tau}BG(t) \quad (2.1)$$

dove  $g$  rappresenta il guadagno statico del sistema che mette in relazione il glucosio plasmatico al glucosio interstiziale (che si può considerare uguale a 1, essendo le due concentrazioni uguali nel caso statico), mentre  $\tau$  è la costante di tempo, che può variare a seconda dell'individuo.



**Figura 2.1:** Modello bicompartimentale per la descrizione della dinamica plasma-interstizio [25].

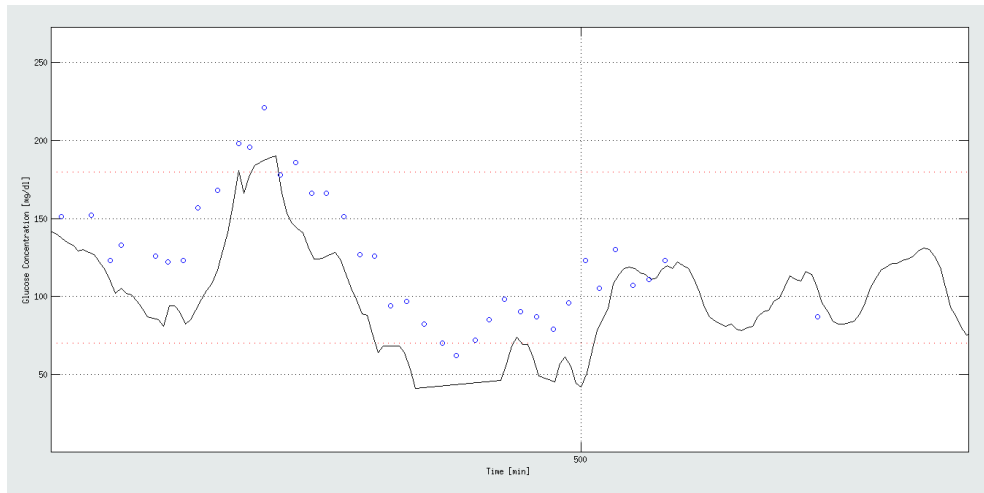
La distorsione introdotta dalla dinamica plasma-interstizio si manifesta in un ritardo temporale del segnale CGM rispetto ai valori plasmatici reali, specialmente in condizioni di rapide variazioni di glicemia come quelle successive ai pasti o dovute a ipoglicemie notturne. Quanto detto è visibilmente comprensibile osservando la Figura 2.2. Il sistema agisce quindi come un filtro passa-basso, introducendo una distorsione in fase e in ampiezza tra glucosio plasmatico e interstiziale. Gli studi effettuati su questo problema hanno permesso di valutare il gradiente di diffusione ritenendolo pressoché costante, mentre la costante di tempo legata al ritardo di diffusione risulta solitamente compresa tra i 5 e i 10 minuti e può variare nel tempo.



**Figura 2.2:** Scostamento del segnale CGM (linea continua) rispetto ai campioni SMBG (pallini). Si vede con chiarezza il ritardo dovuto alla cinetica BG-IG.

### 2.1.2 Aspetti legati alla calibrazione

Come anticipato, il segnale CGM può presentare anche una deriva verticale dovuta a problemi di calibrazione, come si osserva in Figura 2.3 confrontando le misure CGM con quelle ottenute dai dispositivi pungi-dito.

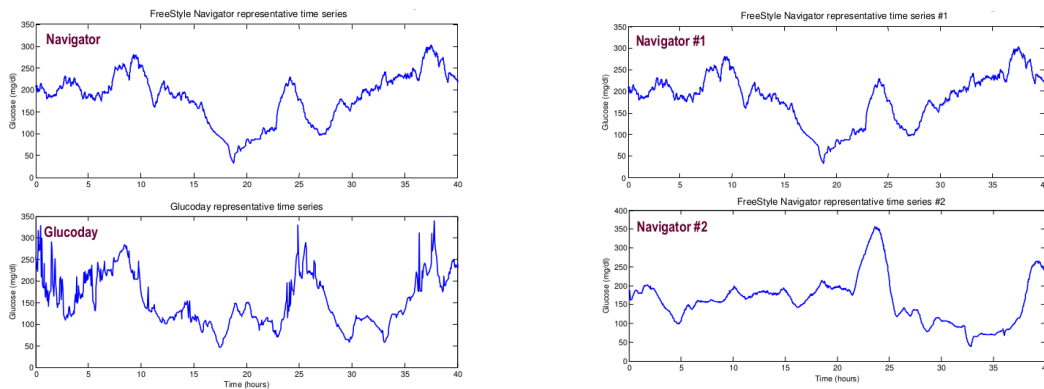


**Figura 2.3:** *Scostamento verticale del segnale CGM (linea continua) rispetto ai campioni SMBG (pallini). L'errore sistematico che si osserva è dovuto alla perdita di calibrazione.*

I dispositivi CGM misurano una corrente elettrica proporzionale alla concentrazione di glucosio interstiziale; tale corrente deve poi essere moltiplicata per un fattore di scala calcolato al momento dell'innesto del sensore utilizzando un paio di misurazioni di glucosio plasmatico. Il fattore di scala prende il nome di fattore di calibrazione e nel caso ideale dovrebbe essere considerabile pressoché costante. Successivamente all'inserimento del sensore nella sottocute è necessario che il segnale si mantenga adeguatamente preciso e stabile nel tempo, ma alcuni sensori evidenziano una modifica delle prestazioni che si accompagna ad una ridotta accuratezza del sistema rispetto alle variazioni reali della glicemia. Questo è il motivo per cui bisogna ricalibrare il sensore, operazione che prevede una modifica del fattore di calibrazione e che viene effettuata quando il segnale acquisito risulta relativamente stabile, cioè in corrispondenza degli intervalli in cui la differenza dovuta alla cinetica plasma-interstizio è minima. Poiché nel corso del suo utilizzo il sensore tende a degradarsi, talvolta anche a causa di interazioni indesiderate con i fluidi biologici, è necessario ripetere la calibrazione più volte durante il giorno, fino a che non si renda necessaria la definitiva sostituzione dello stesso, dopo un tempo variabile tra i 2 e i 7 giorni, a seconda della tipologia. Diversi studi pubblicati in letteratura ([16] [17] [18]) hanno tentato di chiarire e risolvere il problema della calibrazione, la maggior parte dei quali in maniera retrospettiva attraverso l'applicazione di modelli lineari regressivi, a partire dal maggior numero possibile di misurazioni plasmatiche a disposizione. Negli ultimi anni sono stati presentati dei metodi di calibrazione on-line implementati utilizzando il filtro di Kalman (KF) e il filtro di Kalman Esteso (EKF), al fine di migliorare il segnale CGM parallelamente alla possibilità di descrivere la variazione nel tempo della precisione del sensore ([19] [20] [21]).

### 2.1.3 Rumore di misura

Nell'intenzione di visualizzare una serie temporale che onori in maniera affidabile i valori reali e affinché tale serie sia utilizzabile per il controllo della glicemia, è indispensabile ridurre il più possibile il rumore, inevitabilmente presente in qualsiasi strumento di misura, da cui sono affetti i dati. Infatti, in aggiunta agli errori di calibrazione, il segnale CGM presenta componenti di rumore che tipicamente appartengono alle alte frequenze e che dipendono dalla tecnologia del sensore, in particolare dal sistema di acquisizione. Per una comprensione più immediata della variabilità del Signal-to-Noise Ratio (SNR) tra un sensore e l'altro, nella parte destra di Figura 2.4 sono presentati i profili glicemici rappresentativi di due tecnologie CGM differenti. In entrambi i segnali si percepisce visibilmente la presenza di un rumore di misura che si sovrappone al segnale glicemico vero e che può variare, come verrà descritto nel dettaglio in seguito, oltre che da sensore a sensore, anche da un soggetto ad un altro, come si evince dalla parte sinistra di Figura 2.4, e anche durante il giorno all'interno della stessa acquisizione.



**Figura 2.4:** *A destra: profili CGM rappresentativi di due differenti tecnologia (FreeStyle Navigator e GlucoDay). A sinistra: profili CGM di due soggetti diversi acquisiti con tecnologia FreeStyle Navigator [31].*

Essendo il rumore presente tipico delle alte frequenze, il filtraggio causale passa-basso rappresenta l'approccio maggiormente adottato per ridurlo. Tale rumore è considerato in prima approssimazione additivo e bianco (i cui campioni sono cioè completamente scorrelati tra loro) e consente di esprimere il segnale CGM attraverso l'equazione

$$y(t) = u(t) + v(t) \quad (2.2)$$

dove  $y(t)$  è il livello di glucosio misurato,  $u(t)$  rappresenta il valore vero, ma sconosciuto, relativo alla concentrazione di glucosio e  $v(t)$  è il rumore di misura che si sovrappone a  $u(t)$ . L'obiettivo del filtraggio è quello di ricavare  $u(t)$  a partire dalla misura rumorosa  $y(t)$ . Uno dei maggiori problemi di questo metodo è, però, dato dal fatto che nella maggior parte dei casi le bande



di  $u(t)$  e di  $v(t)$  nel dominio della frequenza risultano in parte sovrapposte e questo rende inapplicabile il filtraggio passa-basso senza una conseguente distorsione del segnale vero. In particolare, la distorsione di cui si parla si manifesta con un ritardo temporale di  $u(t)$ , fenomeno che porta a stabilire necessariamente un compromesso tra regolarità del segnale e ritardo temporale al fine di garantire il più possibile l'utilizzabilità clinica del profilo glicemico filtrato, in particolare in vista della possibilità di predire, senza falsi positivi e senza considerevoli ritardi, eventuali superamenti delle soglie di iperglicemia e soprattutto di ipoglicemia.

Dalla letteratura si evince quanto sia necessario poter filtrare in tempo reale il segnale CGM, sia per quanto concerne i dispositivi da commercializzare, sia nella ricerca universitaria. La maggior parte delle aziende produttrici di sistemi CGM non ha mai pubblicato informazioni dettagliate sugli algoritmi di filtraggio implementati nel software dei propri dispositivi. Nonostante ciò, le informazioni ottenute da studi e comparazioni consentono di affermare che la maggior parte dei sistemi CGM in commercio sembra implementare internamente una sorta di filtro a media mobile (MA) a parametri costanti [32]. Il filtro a media mobile è un tipo di filtro a risposta impulsiva finita (FIR), lineare e causale che trova applicazione in moltissimi ambiti riguardanti l'elaborazione dei segnali. Data una serie di misure CGM descritte dall'equazione (2.3), il filtro calcola la media pesata degli ultimi  $N$  campioni misurati e fornisce, quindi, una stima  $\hat{u}(t)$  del reale valore di concentrazione di glucosio data da

$$\hat{u}(t) = \sum_{k=0}^{N-1} b_k y(t-k) \quad (2.3)$$

dove  $b_k$  è il peso dato al campione misurato  $k$  tempi prima del campione corrente. Ogni  $b_k$  deve essere formulato in maniera tale da soddisfare la condizione complessiva espressa dall'equazione

$$\sum_{k=0}^{N-1} b_k = 1 \quad (2.4)$$

A seconda della combinazione dei pesi scelti è possibile realizzare differenti filtri di questo tipo che, però, presentano il limite di non poter riadattare nel tempo i parametri di modello e lo svantaggio di introdurre un ritardo che cresce al crescere del grado di regolarizzazione applicato.

Per superare queste limitazioni, recentemente è stato presentato un metodo di filtraggio del rumore sviluppato in un contesto Bayesiano ed implementato mediante un filtro di Kalman, che consente di stimare in tempo reale le variazioni inter-individuali e intra-individuali del SNR [26] [27]. Questo algoritmo è stato implementato nel software sviluppato in questa tesi e verrà presentato dettagliatamente nel Capitolo 5.

### 2.1.4 Predizione del livello glicemico futuro e generazione di allarmi

Un ulteriore miglioramento per rendere un segnale CGM più utile nel trattamento del diabete consiste nella possibilità di prevedere in anticipo il raggiungimento di soglie critiche di ipoglicemia ed iperglicemia. La conoscenza anticipata dell'avverarsi di tali eventi è possibile attraverso la predizione della glicemia futura e permette di generare allarmi se si prevede un attraversamento di soglia. Sono sufficienti 20, 30 minuti di anticipo, in modo tale da lasciare il tempo al paziente di agire per evitare l'evento. La predizione deve essere accurata e precisa, così da predire la maggior parte degli eventi e limitare il numero di falsi allarmi.

Negli ultimi anni la ricerca ha impiegato molte risorse per realizzare nella maniera migliore e più innovativa possibile questi obiettivi [31] [33]. Senza dubbio, per ridurre al minimo la propagazione di rumore ed errori, ogni metodo predittivo deve elaborare inizialmente il segnale regolarizzandone l'andamento. Detto altrimenti, perché ci siano dei risultati affidabili e incoraggianti riguardo la predizione futura della glicemia è necessario che vengano fatti degli ulteriori passi avanti per rendere maggiormente robusti gli algoritmi di regolarizzazione.

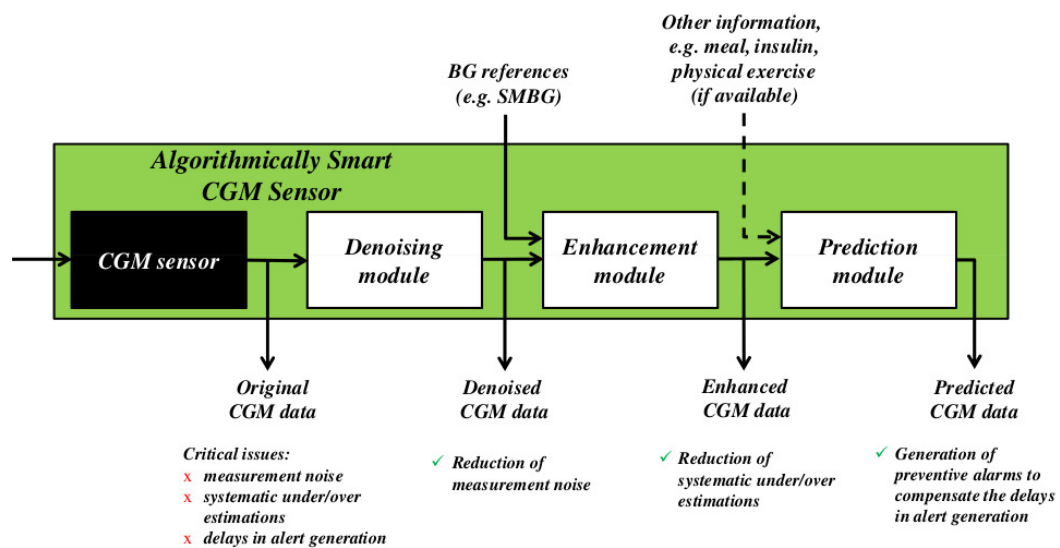
## 2.2 Il concetto di Smart CGM (sCGM) Sensor

I sistemi per il monitoraggio continuo del glucosio rappresentano un'innovazione tecnologica e clinica che in questi anni ha permesso ad aziende ed università di raggiungere importanti traguardi nel trattamento del diabete. Da un punto di vista pratico si è visto come, parallelamente ai grandi vantaggi che sistemi di questo tipo offrono, molto sono ancora oggi i problemi da indagare. Per quanto concerne la rilevazione di un segnale continuo, affidabile e preciso tramite strumenti minimamente invasivi, ciò su cui si punta è il miglioramento degli algoritmi di calibrazione, di smoothing, di predizione e di generazione di allarmi, al fine di poter ottenere un profilo temporale in tempo reale che tenga in considerazione la variabilità delle caratteristiche di ciascun segnale nel tempo. Accanto a questi aspetti riguardanti l'elaborazione dei dati, altri problemi aperti riguardano la portabilità dei dispositivi, la loro interazione con i tessuti umani e le caratteristiche di alimentazione. La ricerca sta cercando, in relazione a queste tematiche, di proporre al mercato sistemi CGM con sensori di dimensioni sempre più ridotte, confortevoli e facili da impiantare, biocompatibili, stabili nell'interazione con i tessuti biologici e durevoli sia nel funzionamento che nelle prestazioni.

In letteratura e nei siti ufficiali delle aziende produttrici non sono mai stati pubblicati i dettagli implementativi dei sistemi in commercio. Quanto conosciuto sui software di sistema deriva da studi di analisi, simulazione e comparazione dei risultati poiché ogni gruppo di lavoro, aziendale o universitario che sia, tende in generale a lavorare in maniera indipendente. Le realtà azien-

dali dei principali dispositivi CGM intraprendono percorsi autonomi rivolti al miglioramento dei propri software, mentre i gruppi di ricerca universitari utilizzano dati CGM e risultati delle aziende per rendere i propri algoritmi più efficienti, robusti, affidabili di quelli da cui provengono tali dati.

In [22] e [23] è stato proposto il concetto di Smart Continuous Glucose Monitoring Sensor (sCGM), dove con il termine sensore smart ci si riferisce ad un sistema composto da un sensore CGM commerciale e da algoritmi operanti in cascata, dedicati al miglioramento dell'accuratezza e della precisione del segnale in uscita dal sensore.



**Figura 2.5:** Schema dell'architettura dello Smart CGM Sensor [23].

Per un sensore di questo tipo, sia nel caso di controllo ad anello aperto (con generazione di allarmi) che di controllo ad anello chiuso (pancreas artificiale), gli obiettivi funzionali da implementare sono: la calibrazione, e ricalibrazione, per rendere il segnale CGM il più possibile affidabile e aderente al segnale vero, il filtraggio per regolarizzarne il profilo massimizzando il rapporto segnale-rumore, la predizione temporale e la generazione di allarmi iperglicemici e ipoglicemici. La Figura 2.5 mostra uno schema dell'architettura del sCGM: in questo caso l'informazione fornita in output dal sensore (qualsiasi sensore in commercio potrebbe potenzialmente essere inserito in questo sistema) viene elaborata da una serie di moduli software in cascata, in questo caso uno per il denoising, uno per la calibrazione e uno per la predizione. Fino ad ora i tre moduli algoritmici del sCGM sono stati testati, mediante codici MatLab prototipali, su dati ottenuti dai sistemi FreeStyle Navigator e GlucoDay e applicati a dati CGM ottenuti da SEVEN Plus, con lo scopo di rendere più robusto, affidabile ed efficiente il monitoraggio continuo del glucosio.

## 2.3 Scopo della tesi

Il lavoro che viene presentato è frutto dell'interesse nel realizzare un software che costituisca una sorta di piattaforma informatica in grado di elaborare il segnale CGM in maniera modulare, grazie all'implementazione cooperante dei moduli di calibrazione, di smoothing, di predizione e di generazione di allarmi. In relazione a quanto detto sull'importanza della ricerca a livello dei singoli gruppi di lavoro, è emersa la necessità di creare un software che potesse inglobare contemporaneamente i vari algoritmi di elaborazione delle serie CGM realizzati al Dipartimento di Ingegneria dell'Informazione (DEI) di Padova. Attualmente nell'ambito della calibrazione e della generazione di allarmi non sono stati ancora individuati in maniera completamente convincente ed univoca degli algoritmi di gestione dei dati, perciò per questa tesi si è scelto, come punto di partenza, di implementare i moduli di smoothing e di predizione temporale. Il programma è stato realizzato con il linguaggio di programmazione MatLabR2012b e prevede l'interazione con l'utente tramite Graphical User Interface (GUI). Il codice è oper-source, cosa che dà la possibilità di ampliare, migliorare e completare le caratteristiche del software.

Nel Capitolo 3 verranno illustrate le caratteristiche progettuali e funzionali di un'interfaccia grafica, mentre nel Capitolo 4 verrà presentato il software realizzato, denominato *Continuous Glucose Monitoring Laboratory* (CGMLab). Nei Capitoli successivi, 5 e 6, verranno poi discussi i moduli specifici per denoising e predizione.

# Capitolo 3

## Progetto di Graphical User Interface (GUI) in MatLab

### 3.1 Aspetti generali

Una GUI è un'interfaccia grafica realizzabile con linguaggi di programmazione, composta da una o più finestre, che appare sul monitor del computer, contenente componenti di controllo che consentono un utilizzo interattivo delle funzioni implementate. L'utente che manovra l'interfaccia non necessariamente deve conoscerne o comprenderne i dettagli implementativi sulla programmazione a basso livello, né deve realizzare script o funzioni per interagire con il software. Nel linguaggio di programmazione MatLab una GUI è realizzata attraverso un set di funzioni individuali o annidate. Il software è, quindi, creato testualmente in maniera sequenziale, ma l'esecuzione dei comandi è di tipo procedurale, event-driven. Gli eventi sono gestiti graficamente attraverso l'interazione con componenti grafici ai quali si aggancia, per la parte di calcolo ed elaborazione, la programmazione usuale. Considerando la flessibilità, l'immediatezza e l'utilità a cui si punta realizzando una GUI, la programmazione di questi oggetti richiede scelte di design che rendano l'interfaccia il più possibile immediata e semplice nell'utilizzo.

Al fine di fornire una panoramica generale sulla creazione e gestione di una interfaccia grafica utente in MatLab, verranno descritte le fasi principali di progettazione grafica di una GUI, verranno presentate le componenti grafiche inseribili e saranno sommariamente introdotti e commentati alcuni tra i principali passaggi di codifica dell'interfaccia. Questa introduzione teorica vuole fornire una base conoscitiva che possa coadiuvare sufficientemente e completare la comprensione di questo lavoro. Per i dettagli di quanto presentato si consiglia la lettura dei manuali di approfondimento [28] [29] [30].

## 3.2 Fasi della progettazione

Prima di realizzare concretamente un'interfaccia grafica utente è necessario progettarela, dal punto di vista grafico e funzionale. Le fasi di progettazione di una GUI sono sostanzialmente tre: analisi del problema, progetto e realizzazione di un prototipo cartaceo e codifica, a cui si sommano dei test di valutazione e validazione intermedi.

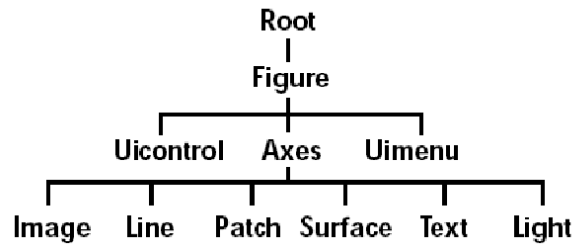
Prima di disegnare una finestra di lavoro è necessario tenere in considerazione chi la utilizzerà e a quale scopo. Non a caso, dal termine Graphical User Interface si evince che l'utente ricopre un ruolo chiave nella fase iniziale di analisi, in quanto il progettista deve comprendere ciò di cui l'utente ha bisogno, il modo in cui queste funzionalità devono apparire, e realizzare l'interfaccia adattandosi alle conoscenze di chi andrà ad interagirvi. Detto altrimenti, il successo di una GUI si misura sulla base di quanto questa sia in grado di andare incontro all'utente, sia esso un bambino o un giovane studente o un ricercatore esperto, attraverso una finestra interattiva quanto più possibile familiare, consistente e intuitiva.

Dopo aver analizzato le informazioni riguardanti l'utente a cui è destinato il software e averne compreso gli obiettivi, è possibile pensare alla progettazione grafica vera propria dell'interfaccia. In questa fase, che non prevede ancora alcuna codifica, si prendono in esame gli oggetti necessari a realizzare le funzioni richieste, i tipi di dati a disposizione e i tipi di risultati che si vogliono ottenere e visualizzare. Questo comporta la specifica degli input, del controllo che su di essi si deve effettuare, della loro visualizzazione ed elaborazione e degli output risultanti. Le funzioni e le operazioni sui dati devono essere organizzate in raggruppamenti grafici logici ed intuitivi e devono essere richiamate e identificate grazie a un nome breve e sintetico, ma rappresentativo. Dopo aver deciso cosa attuare tramite l'interfaccia e come gestire e mettere in relazione le funzioni e i dati, la cosa più utile da fare è disegnare su un foglio di carta un prototipo della GUI. In questo modo, prima di pensare ai dettagli implementativi riguardanti colori e font da impiegare, ci si concentra sull'aspetto e sul comportamento dell'interfaccia al fine di valutare preliminarmente se essa appare o meno come si desidera. Grazie a questo step intermedio tra progettazione e codifica, è possibile ottimizzare posizione, dimensioni e organizzazione di ciascun componente grafico, prevenendo riprogrammazioni temporalmente dispendiose qualora l'aspetto della GUI non dovesse essere soddisfacente.

Avendo realizzato un prototipo dell'interfaccia così come la si desidera costruire e avendo associato a ciascuna sua parte e componente una precisa funzionalità, eventualmente connessa ad altre, è possibile procedere alla codifica della GUI, cioè alla sua realizzazione computazionale, al fine di ottenere un'interfaccia che, oltre ad essere stata progettata secondo specifiche esigenze, funzioni anche correttamente. Un'interfaccia utente in MatLab può essere costruita, come verrà approfondito in seguito, per due vie: utilizzando una guida per la costruzione grafica o programmando completamente la GUI.

### 3.3 Componenti grafici inseribili in una GUI

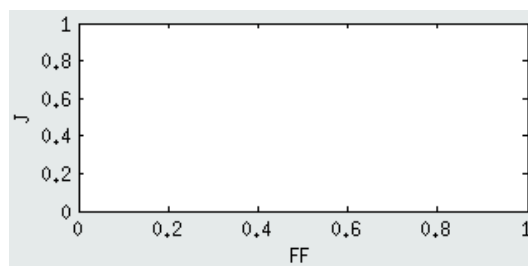
Gli oggetti che possono essere inseriti in un'interfaccia grafica utente sono di vario tipo, ma ciascuno di essi è univocamente identificabile. In Figura 3.1 sono illustrate, disposte gerarchicamente, tutte le categorie di oggetti che possono comporre una GUI.



**Figura 3.1:** *Categorie di oggetti di un'interfaccia grafica, disposte gerarchicamente [28].*

La maggior parte degli elementi di una interfaccia grafica utente, siano essi di controllo o puramente grafici, sono creati con lo scopo di perseguire un'azione, modificare delle impostazioni per azioni future o visualizzare risultati. Gli elementi di controllo inseriti in CGMLab includono push button, toggle button, radio button, slider, edit text control, static text control, pop-up menu e check box. Altri componenti sono axes, panel, table e button group. Di seguito se ne illustrano brevemente le caratteristiche principali, rimandando l'approfondimento ai manuali teorici [28] [29] [30].

a) Axes: componente che permette alla GUI di visualizzare grafici, come immagini e plot (Figura 3.2).



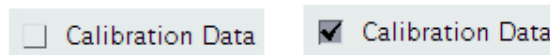
**Figura 3.2:** *Axes vuoto, con assi ed etichette inizializzati.*

b) Button Group: pannello utilizzato per gestire la selezione di più radio button e toggle button ad azione, di solito, mutuamente esclusiva (Figura 3.3).



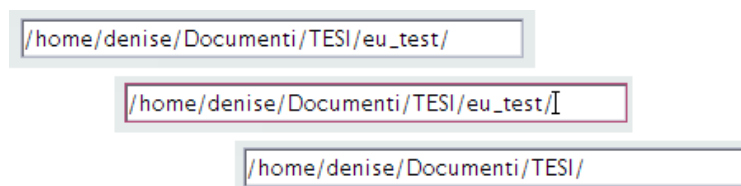
**Figura 3.3:** *Button Group con alcuni pulsanti disabilitati.*

c) Check Box: casella di spunta che genera un'azione se selezionata ed è utile per indicare lo stato di un'opzione o un attributo e nel caso in cui l'utente necessiti di esprimere un certo numero di scelte indipendenti e non mutuamente esclusive; presenta due stati possibili: on, in cui la casella quadrata contiene una spunta, e off, in corrispondenza del quale la casella di presenta vuota (Figura 3.4).



**Figura 3.4:** *Esempio di Check Box deselezionato (sinistra) e selezionato (destra).*

d) Editable Text: casella di testo che permette all'utente di inserire un contenuto o modificarlo, utilizzata se si vuole in input una stringa o una serie di numeri; è indicabile inizializzare la casella di testo con una stringa o un valore di default, così da non rendere l'utente necessariamente obbligato a inserirne uno (Figura 3.5).



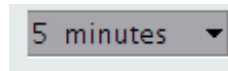
**Figura 3.5:** *Editable Text, in cui viene modificato il testo inserito.*

e) Panel: componente che permette la suddivisione grafica della figura, conferendo maggior modularità e gestibilità anche alla codifica relativa; ciascun pannello può avere un titolo e oggetti propri al suo interno.

f) Pop-Up Menu: menu a tendina che si apre verso il basso permettendo la scelta di una sola tra le opzioni in lista; l'opzione selezionata viene visualizzata nella casella che rappresenta il menu in forma chiusa; a differenza di un editable

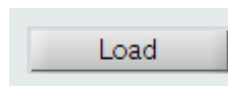


text, il pop-up menu è riconoscibile da un simbolo posto sull'estremo destro dell'oggetto in cui un triangolino rivolto in basso indica il verso di apertura del menu (Figura 3.6).



**Figura 3.6:** *Pop-Up Menu chiuso.*

g) Push Button: pulsante che genera lo stesso tipo di azione ogni volta che viene premuto; finché il pulsante è premuto esso appare nel suo stato di depressione, mentre riassume l'aspetto originario quando viene rilasciato; un push button invoca un evento in maniera immediata ed è uno tra i componenti maggiormente utilizzati nelle interfacce grafiche; poiché i push buttons rappresentano un'azione, spesso sono etichettati con un verbo o una parola che la richiami alla mente in maniera immediata e intuitiva (Figura 3.7).



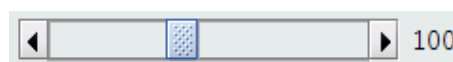
**Figura 3.7:** *Push Button per il caricamento dei dati.*

h) Radio Button: pulsante simile al check box che indica un'opzione da selezionare; è tipicamente mutuamente esclusivo all'interno di un gruppo di pulsanti dello stesso tipo e presenta due stati ad esso associati, di selezione e deselegione, in corrispondenza dei quali il simbolo circolare dell'oggetto contiene o meno un pallino, rispettivamente (Figura 3.8).



**Figura 3.8:** *Esempio di Radio Button deselegionato (sinistra) e selezionato (destra).*

i) Slider: componente che accetta input numerici, all'interno di un range prestabilito, per mezzo di una barra scorrevole e due frecce agli estremi; la posizione corrente della barra indica la localizzazione del valore corrispondente all'interno del range di valori (Figura 3.9).



**Figura 3.9:** *Barra scorrevole, posta a 100 all'interno del range 30-200.*

j) Static Text: componente che consente il controllo della visualizzazione di stringhe testuali all'interno dell'interfaccia ma non avvia nessun tipo di azione;

generalmente è utilizzato come etichetta di altri componenti di controllo e, a differenza dell'editable text, non può essere modificato dall'utente.

k) Table: oggetto tabella, formato da una serie di colonne e righe numerate ed etichettabili, che può contenere nelle proprie caselle stringhe di testo, numeri o indicatori di scelta; le dimensioni della tabella si adattano automaticamente alle dimensioni del contenuto oppure possono essere impostate nella codifica; è possibile rendere alcune colonne internamente modificabili (Figura 3.10).

	AR(30-0.8-1)	LIN(30-0.8-1)
RMSE	31.4456	33.4544
ESOD	0.7461	0.0230
gain time	15	5
J	49.4029	13.7151
FF Opt	NaN	NaN

**Figura 3.10:** Table con 5 righe e 2 colonne.

l) Toggle Button: componente binario, o bipolare, che genera un'azione se premuto e che presenta due stati, on e off; non presenta uno stato intermedio, perciò esso rimane nel suo stato selezionato o deselezionato finché l'utente non clicca nuovamente su di esso (Figura 3.11).



**Figura 3.11:** Esempio di Toggle Button deselezionato (sinistra) e selezionato (destra).

Ciascun oggetto grafico, sia esso componente di controllo o oggetto di visualizzazione grafica, possiede un set di proprietà che possono essere manipolate dal programmatore fino al raggiungimento dell'aspetto desiderato. Tali proprietà, numerose e dettagliate dal punto di vista gestionale, non vengono in questa sede presentate. Si ritiene superfluo e dispendioso un approfondimento ulteriore a riguardo, essendo questa la presentazione di una GUI e non un manuale teorico su come realizzarne una. Per i dettagli implementativi si consiglia la lettura di [28] [29] [30].

## 3.4 Programmazione e gestione delle callbacks

Il fulcro della creazione di un'interfaccia grafica utente è concentrato nella codifica, nella gestione del controllo e del calcolo. Al di là del design, della scelta dei componenti e dell'intuitività nel suo utilizzo, una GUI deve saper

svolgere le funzioni per cui è stata creata. Come accade per ciascuna interfaccia grafica, l'utente entra in contatto soltanto con gli aspetti visibili risultanti dal processo di costruzione, senza talvolta immaginare che la sola codifica di inizializzazione di un componente richiede svariate righe o svariate decine di righe di codice. Un'interfaccia utente in MatLab può essere realizzata, come accennato, con l'ausilio di GUIDE (Graphical User Interface Development Environment) o costruita completamente in maniera programmata.

### 3.4.1 Due approcci realizzativi: GUIDE e costruzione programmata

GUIDE è un tool grafico per la realizzazione di interfacce grafiche utente in MatLab che consente di ridurre notevolmente i tempi di sviluppo rispetto alla programmazione testuale. Questo approccio permette di costruire la parte grafica dell'interfaccia in maniera guidata e più immediata, a partire da una figura a cui si possono aggiungere i componenti scelti. Man mano che viene perfezionata la parte grafica, GUIDE associa a quanto creato un file di codice contenente le callbacks (chiamate a funzione). La parte grafica viene salvata in un FIG-file, mentre la codifica è contenuta in un M-file a cui il programmatore deve aggiungere le parti di controllo e di calcolo.

Un'alternativa a questo approccio è rappresentata dalla possibilità di programmare completamente la propria interfaccia grafica, attraverso un approccio più costruttivo e meno guidato. In questo secondo caso, infatti, la definizione di componenti e rispettive proprietà e interazioni, avviene tramite stesura di un file di codice. La figura viene così creata, non riaprendo il FIG-file automaticamente salvato come con GUIDE, bensì lanciando in esecuzione il file con il codice volontariamente programmato. La figura non necessita di essere salvata tra una sessione di lavoro ed un'altra proprio perché ogni volta che viene lanciato il programma essa viene completamente ricreata. In relazione alla complessità della GUI che si sta andando a creare è consigliabile indirizzarsi verso un approccio piuttosto che l'altro, come schematicamente indicato nella seguente Tabella 3.1.

GUI	Tecnica
Box di Dialogo	In MatLab sono disponibili delle funzioni che creano all'occorrenza delle finestre di dialogo (errori, avvertimenti, domande, valori in input, ecc.)
GUI con poche componenti	La procedura di completa programmazione rappresenta una possibilità. È facile in questo caso usare la programmazione per creare la GUI ed eseguire chiamate strutturate in modo semplice.

GUI mediamente complesse	GUIDE permette di raggiungere un buon compromesso tra facilità di costruzione e autonomia nel controllo dei componenti.
GUI complesse con numerose componenti e spesso in relazione con altre GUI	La procedura di completa programmazione rappresenta una necessità. GUIDE diventa in questo caso dispendioso e poco gestibile, mentre è meglio scrivere dei programmi che creino in maniera puntuale e riproducibile la GUI (considerando anche la possibilità di dover cambiare piattaforma hardware).

---

**Tabella 3.1**

Appare chiaro come il codice risultante dai due diversi approcci risulti differente. I file di GUI costruite in maniera completamente programmata sono generalmente più lunghi, perché necessitano di esprimere e definire ciascuna proprietà e caratteristica dei suoi componenti grafici, a differenza di un file creato con GUIDE che lascia da gestire solo il controllo e i calcoli, essendo tutto il resto definito nel FIG-file. La scelta del tipo di approccio da adottare dipende quindi in parte dall'esperienza di chi programma e in parte dal tipo di GUI che si intende realizzare. È possibile combinare i due approcci, creando l'interfaccia con GUIDE e modificandola poi programmaticamente, mentre la consequenzialità in senso contrario non è permessa.

Arrivati a questo punto è necessario fornire alcuni sommari concetti su come viene codificata un'interfaccia utente in MatLab. Dopo averne presentato i due approcci realizzativi, considerato il fatto che CGMLab è stata realizzata in maniera completamente programmata, è importante capire almeno come viene impostato questo tipo di codifica, chiamata anche tecnica di programmazione a basso livello. Tipicamente, un semplice codice GUI, contiene le seguenti sezioni:

- commenti che vengono visualizzati in risposta all'invocazione dell'help di MatLab;
- inizializzazione dei dati e dei parametri che compaiono nella GUI, compresi quelli necessari alla costruzione dei suoi componenti;
- costruzione della figura principale e degli oggetti grafici che vi compaiono (nel linguaggio MatLab una GUI viene codificata come una figura);
- eventuali inizializzazioni aggiuntive di cui i componenti necessitano per esistere;

- callbacks relative ai componenti inseriti.

Le callbacks, o chiamate a funzione, sono delle routines che vengono eseguite in risposta ad un evento generato dall'utente, da cui deriva il concetto di programmazione event-driven, e in corrispondenza di un componente specifico della GUI. Una callback può essere definita, se l'interfaccia da costruire è relativamente semplice, durante la creazione dell'oggetto corrispondente, altrimenti è preferibile codificare le callbacks come funzioni a sé stanti attraverso una propria specifica sintassi.



## Capitolo 4

# CGMLab: una GUI MatLab per l'analisi e l'elaborazione di segnali continui di glicemia

CGMLab (*Continuous Glucose Monitoring Laboratory*) è un software per il processamento e l'elaborazione di serie temporali acquisite da sensori per la misura continua della glicemia. Il programma, realizzato con il software MatLabR2012b, è nato allo scopo di:

- fornire uno strumento user-friendly per lo studio del segnale glicemico;
- rendere robusta e modulare l'elaborazione di serie temporali glicemiche;
- consentire una più rapida gestione di segnali provenienti da tecnologie differenti;
- ridurre i tempi di applicazione di algoritmi e di confronto dei risultati;
- implementare in un'unica piattaforma software gli algoritmi propri del gruppo di ricerca dell'Università di Padova;
- consentire la simulazione di un'elaborazione algoritmicamente smart;
- fornire un supporto ulteriore alla didattica.

Il sCGM proposto in [23] è composto da più moduli di processamento operanti in cascata sul segnale quasi-continuo misurato e trasmesso da dispositivi CGM commerciali. In questa tesi sono stati scelti e realizzati i moduli di denoising e predizione, lasciando aperta la possibilità di codificare anche la parte relativa alla calibrazione e alla generazione di allarmi.

La GUI che è stata realizzata è composta da tre diverse sotto-interfacce. La prima di queste rappresenta la finestra di pre-processing e caricamento dei dati, mentre le altre due, appartenenti alla finestra di elaborazione del segnale,

implementano i moduli di smoothing e predizione. Di seguito vengono introdotte brevemente le tre interfacce create, rimandando ai paragrafi e ai capitoli successivi l'approfondimento relativo alla loro progettazione, organizzazione funzionale interna e codifica.

La Figura 4.1 mostra la finestra di inserimento dei dati. Due parti distinte sono riservate, rispettivamente, all'inserimento di dati CGM e di altri tipi di dati, mentre la parte sottostante permette di visualizzare, sia in maniera tabellare, sia graficamente, il segnale glicemico così come esso si presenta in seguito al pre-processing e prima di essere elaborato.

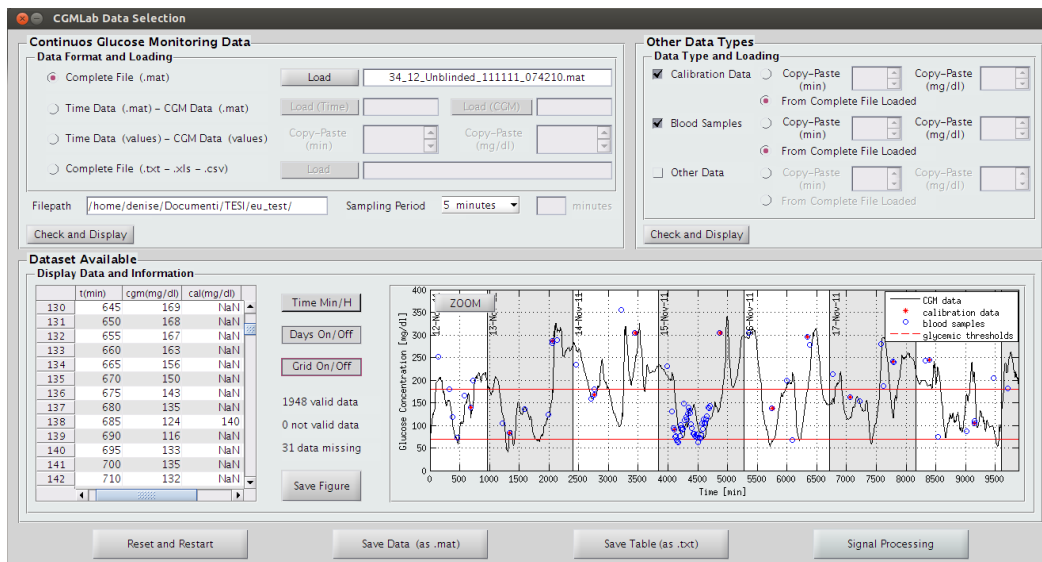


Figura 4.1: Finestra di inserimento e caricamento dei dati.

La seconda finestra invece dà, come accennato, la possibilità di attivare i quattro moduli descritti precedentemente, di cui due sono quelli effettivamente implementati. Il modulo per lo smoothing, di cui si mostra un'istantanea in Figura 4.2, consente, attraverso una libera gestione delle impostazioni di regolarizzazione, di visualizzare il confronto tra segnale CGM e corrispondente segnale regolarizzato. Quantitativamente, il risultato grafico è accompagnato dai risultati della stima parametrica, siano essi frutto di una stima unica iniziale o di più stime effettuate nel tempo. Infine, come si intuisce dalla sotto-interfaccia raffigurata in Figura 4.3, per quanto riguarda la predizione temporale si è ritenuta utile l'implementazione di tre diversi algoritmi, eventualmente da porre a confronto, anche a seconda dei valori di PH e di  $\mu$  impostati.



#### 4.1. Progettazione del modulo per il pre-processing e l'inserimento dei dati 33

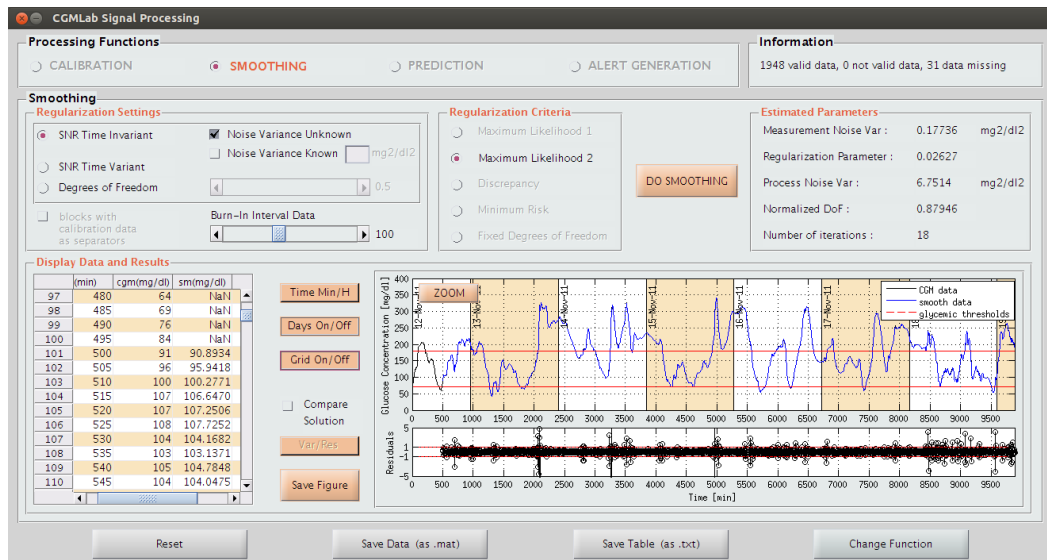


Figura 4.2: Finestra di elaborazione con il modulo di smoothing abilitato.

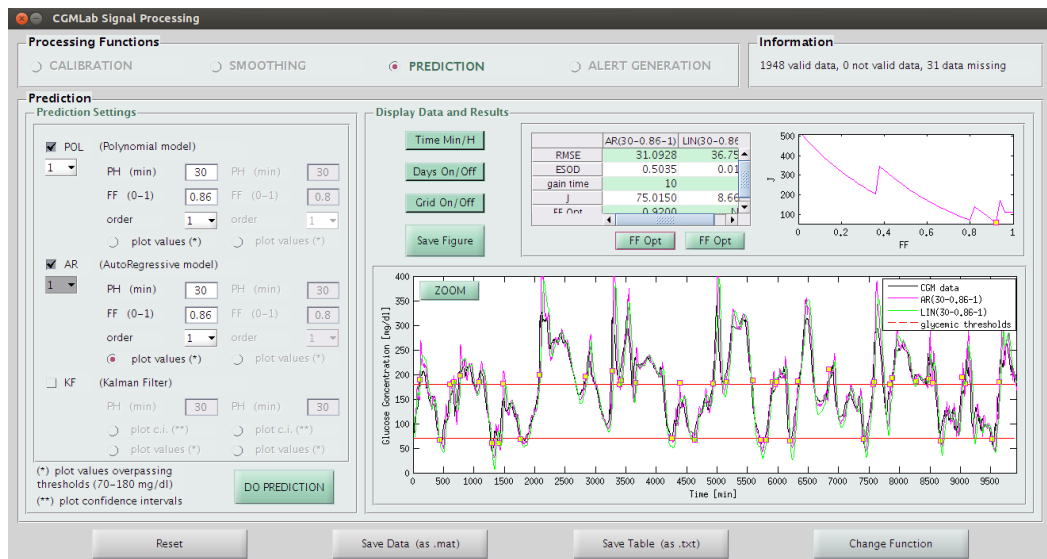


Figura 4.3: Finestra di elaborazione con il modulo di predizione abilitato.

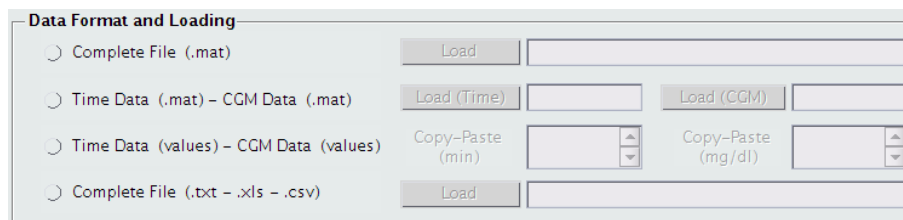
## 4.1 Progettazione del modulo per il pre-processing e l'inserimento dei dati

Dovendo considerare la complessità del funzionamento di una GUI così come fino ad ora è stata brevemente presentata, la parte di caricamento dei dati sembrerebbe quella più immediata da realizzare. In realtà è stata dedicata a questo passaggio un'intera figura grafica, proprio per la propedeuticità imprescindibile che esso ha rispetto alle successive elaborazioni.

Dal punto di vista della progettazione grafica la figura principale *CGMLab Data Selection* (Figura 4.1) è stata suddivisa in tre pannelli principali, due dedicati al caricamento dei dati e il terzo utilizzato per una prima presa visione della serie temporale completa da elaborare. Una volta caricati i dati, i pulsanti sul fondo dell'interfaccia danno la possibilità di resettare quanto inserito, di salvare i dati inseriti o di passare alla seconda finestra grafica per attuare smoothing o predizione.

#### 4.1.1 Layout: *CGM Data Selection* (pannello 1)

Il pannello *Continuous Glucose Monitoring Data* per l'inserimento dei dati CGM, assieme a quello per l'inserimento di altri tipi di dati, è l'unico che compare nella finestra grafica al lancio in esecuzione del software. La progettazione della finestra in esame è stata infatti gestita in maniera tale da indirizzare subito l'utente verso il caricamento dei dati CGM. Come si vede dalla Figura 4.4, supportando l'idea di realizzare una GUI il più possibile versatile e completa, si è pensato di agevolare l'inserimento di dati che possono potenzialmente provenire da tipi di sensori diversi e che possono presentarsi in formati differenti. Nel concreto, attraverso un sotto-pannello di pulsanti radio mutuamente esclusivi, è possibile scegliere se inserire i dati da un MAT-file unico (che può essere anche quello salvato da una precedente sessione di CGMLab), se caricarli da due MAT-file distinti (uno per i valori temporali in minuti e l'altro per le corrispondenti concentrazioni glicemiche in mg/dl), se copiare i dati direttamente in due caselle editabile oppure se caricare i dati da file di un altro formato (.txt,.xls o.csv).

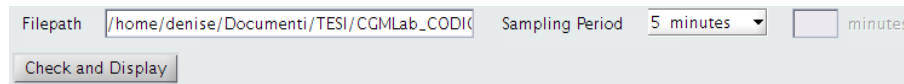


**Figura 4.4:** Sotto-pannello *Data Format and Loading*.

Tutti i radio button, eccetto il terzo, se selezionati abilitano i corrispondenti push button e editable text, lasciando all'utente la possibilità di digitare direttamente il nome del file o di aprire e caricare il file tramite una finestra di dialogo preimpostata.

Oltre al sotto-pannello di scelta del formato di dati da inserire, sono presenti altre due funzionalità, non necessariamente da modificare, ma indispensabili per il caricamento. Nella parte sinistra di Figura 4.5 è stato definito un editable text etichettato per visualizzare il percorso di localizzazione del file o dei file che si stanno inserendo, mentre immediatamente di seguito, allo stesso livello, un pop-up menu ed eventualmente una casella di testo consentono di indicare il

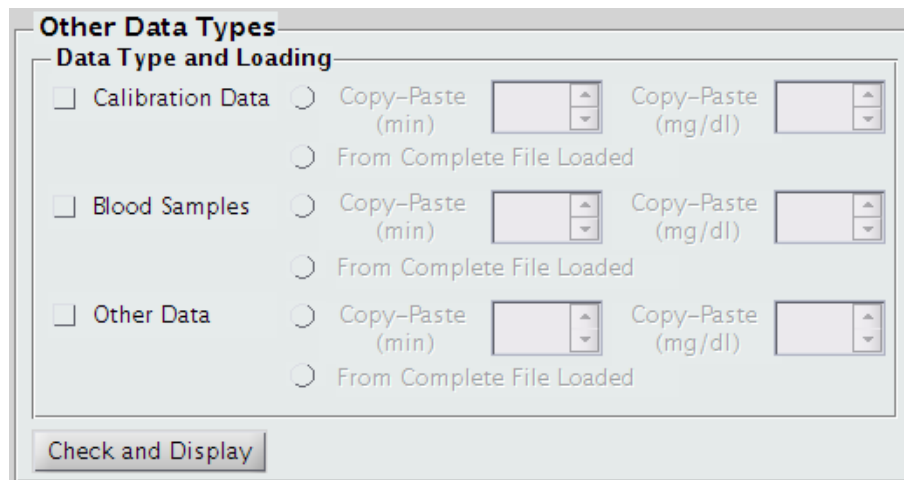
periodo di campionamento della serie temporale acquisita. Completati questi passaggi, è possibile procedere al caricamento dei dati con un click del mouse sul push button *Check and Display*.



**Figura 4.5:** Sezione inferiore del pannello *Continuous Glucose Monitoring Data*.

#### 4.1.2 Layout: *Other Data Types* (pannello 2)

Accanto al pannello per i dati CGM è stato costruito il pannello *Other Data Types* (Figura 4.6), che consente l'inserimento di altre serie temporali oltre a quella relativa alla concentrazione di glucosio.



**Figura 4.6:** Pannello *Other Data Types*.

Nonostante i moduli implementati in CGMLab non utilizzino, in questa prima e parziale versione del software, altri dati oltre a quelli CGM per la regolarizzazione e la predizione temporale, è utile e interessante poter visualizzare e salvare anche i valori di calibrazione e quelli provenienti da campionamento plasmatico, che sicuramente verranno considerati nella codifica futura del modulo di calibrazione. Inoltre, si è definito un terzo check box per l'inserimento di un altro tipo di dati a scelta, eventualmente dedicabile alla visualizzazione degli istanti relativi a iniezioni insuliniche o a interventi terapeutici di altro tipo. L'inserimento di serie temporali in questo pannello può avvenire direttamente richiamando il file completo precedentemente caricato, se il caso specifico lo consente, altrimenti è realizzabile con un semplice copia-incolla dei valori. Come per il pannello precedente, anche in questo caso, una volta completate le impostazioni, è possibile procedere al caricamento dei dati premendo il pulsante *Check and Display*.

### 4.1.3 Layout: *Dataset Available* (pannello 3)

A questo punto, sebbene i dati da utilizzare siano stati caricati e sia direttamente possibile applicarvi gli algoritmi di predizione e smoothing, si è scelto di dedicare la parte inferiore della prima figura alla visualizzazione di quanto inserito. L'utilità di poter visualizzare l'andamento del segnale, i corrispondenti valori puntuali in una tabella e le sue caratteristiche globali ha condotto alla codifica del pannello *Dataset Available* (Figura 4.7), che fornisce una panoramica immediata del profilo glicemico in esame.

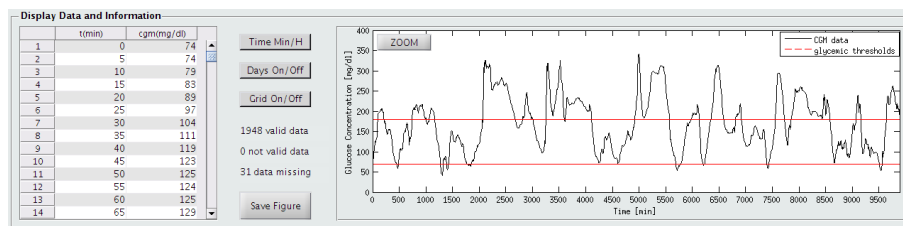


Figura 4.7: Pannello *Dataset Available*.

La tabella visualizzata riporta tutti i valori caricati. Le righe tengono memoria del numero di campioni, mentre la prima colonna visualizza l'incremento temporale corrispondente in minuti. Le colonne successive contengono i dati CGM e gli altri dati, se presenti. Per una visualizzazione ottimale, si è scelto di dare una colorazione alternata alle righe della tabella. Uno spazio all'interno del pannello è stato destinato, inoltre, all'oggetto grafico che permette la visualizzazione del segnale glicemico, il quale può essere anche aperto in un'altra finestra ed ispezionato nel dettaglio con lo zoom. In aggiunta, sono stati inseriti tre pulsanti toggle per rendere possibile una più completa e funzionale visualizzazione del profilo glicemico:

- *Time Min/H*: consente di visualizzare l'asse delle ascisse in minuti o in ore;
- *Days On/Off*: consente di visualizzare dei riquadri a colori alterni corrispondenti ai giorni, se i campioni sono in numero minimo da ricoprire temporalmente almeno 24 ore di acquisizione continua;
- *Grid On/Off*: consente di visualizzare la griglia sullo sfondo del segnale.

Attraverso il push button *Save Figure* è data la possibilità di salvare la figura in diversi formati immagine. Infine, per fornire qualche essenziale informazione sul segnale inserito vengono visualizzate in figura tre componenti statiche testuali che riportano rispettivamente il numero di dati validi, cioè compresi tra le soglie di 39 e 400 mg/dl, la quantità, complementare a quest'ultima, di dati non validi e il numero di dati mancanti, nel caso ci siano problemi nell'acquisizione o il campionamento sia volutamente non uniforme.

## 4.2 Dettagli implementativi del modulo per il pre-processing e l'inserimento dei dati

Nei paragrafi precedenti è stata presentata in maniera del tutto generale la sub-interfaccia che consente il caricamento e la visualizzazione del dataset scelto e si è discusso circa ciò che riguarda il suo scopo funzionale e la sua progettazione grafica. Per comprendere, però, i motivi per cui è stata programmata proprio questa specifica interfaccia presentata, è indispensabile approfondire alcuni aspetti legati alla sua programmazione.

Il software CGMLab consiste in una serie di funzioni, che coprono in totale all'incirca 8000 righe di codice MatLab, la cui esecuzione viene lanciata dalla funzione principale CGMLab, così definita

```
function CGMLab
% main function calling the GUI1 for data selection
% and visualization

close all
clear all
clc

GUI1
```

Tale funzione, che non prevede la presenza di argomenti in ingresso o in uscita, invoca direttamente la funzione *GUI1*, preposta all'inizializzazione delle variabili scalari, vettoriali e matriciali e dei componenti grafici utilizzati nella finestra *CGMLab Data Selection*. Per quanto riguarda questi ultimi, i dettagli più utili sulla loro inizializzazione saranno approfonditi nei paragrafi dedicati ai singoli pannelli.

Di seguito vengono illustrate dettagliatamente le variabili suddette, il loro valore di inizializzazione e il loro contenuto. Si è scelto di mantenerne memoria in una struttura dati globale nominata *init* affinché fosse possibile richiamarle in qualunque parte e in qualunque funzione del codice in maniera immediata. C'è da ricordare, inoltre, che i valori di inizializzazione sono gli stessi valori che le variabili riassumono quando la sessione di lavoro viene resettata e rilanciata cliccando sul push button *Reset and Restart*.

Nome	Tipo	Contenuto	Inizializzazione
<code>init.file_path</code>	stringa	Percorso del file o dei file da caricare	<code>pwd</code>
<code>init.Ts</code>	scalare	Periodo di campionamento	5

<code>init.t_scal</code>	scalare		Minuti da scalare a ciascun valore temporale per ottenere una griglia in minuti a partire dal minuto 0	0
<code>init.data_load.file1</code>	scalare o stringa	o	Nome del primo file da caricare (file completo o file temporale), vale 0 se non è stato indicato nessun file	0
<code>init.data_load.file2</code>	scalare o stringa	o	Nome dell'eventuale secondo file da caricare (dei valori CGM), vale 0 se non è stato indicato nessun file	0
<code>init.data_load.vect1</code>	vettoriale		Valori dei tempi nel caso vengano caricati manualmente con copia-incolla (minuti)	[ ]
<code>init.data_load.vect2</code>	vettoriale		Valori CGM nel caso vengano caricati manualmente con copia-incolla (mg/dl)	[ ]
<code>init.cgm_values.cgm</code>	vettoriale		Dati CGM in mg/dl	[ ]
<code>init.cgm_values.time</code>	vettoriale		Tempi corrispondenti in minuti	[ ]
<code>init.cal_values.cal</code>	vettoriale		Dati di calibrazione in mg/dl	[ ]
<code>init.cal_values.time</code>	vettoriale		Tempi corrispondenti in minuti	[ ]
<code>init.bs_values.bs</code>	vettoriale		Campioni sanguigni in mg/dl	[ ]
<code>init.bs_values.time</code>	vettoriale		Tempi corrispondenti in minuti	[ ]
<code>init.oth_values.oth</code>	vettoriale		Terzo tipo di dati eventualmente da inserire	[ ]
<code>init.oth_values.time</code>	vettoriale		Tempi corrispondenti in minuti	[ ]
<code>init.Ncgm</code>	scalare		Numero di dati CGM validi ed effettivamente elaborabili	0
<code>init.Ncal</code>	scalare		Numero di dati di calibrazione	0

<code>init.Nbs</code>	scalare	Numero di campioni sanguigni	0
<code>init.Nothing</code>	scalare	Numero di dati di altro tipo	0
<code>init.id</code>	scalare	Identificativo della scelta del formato di inserimento dei dati CGM	0
<code>init.data_table</code>	matriciale	Dati da visualizzare in tabella	[ ]
<code>init.Nmissing</code>	scalare	Numero di data missing	0
<code>init.Nthre</code>	scalare	Numero di dati oltre soglia, cioè al di fuori della fascia 39 – 400 mg/dl (dati il cui numero viene segnalato ma che vengono eliminati e posti come data missing)	0
<code>init.first00</code>	scalare	Minuto esatto, anche infra-campionamento, corrispondente al primo minuto del primo giorno completo	1
<code>init.min_init</code>	scalare	Minuto esatto, anche infra-campionamento, corrispondente al primo minuto del blocco di 6 ore completo	1
<code>init.h_label</code>	vettoriale	Etichette testuali presenti nel grafico sull'asse delle ascisse quando questo viene visualizzato in ore	[ ]
<code>init.date_string</code>	vettoriale	Stringhe con giorno-mese-anno, visualizzate se si abilita la visualizzazione dei blocchi di 24 ore nel grafico	[ ]

---

A seconda dei dati inseriti, un click su *Save Data (as .mat)* salva le variabili di interesse in un MAT-file. Ipotizzando di aver inserito tutti i tipi di dati, la codifica del salvataggio dei dati risulta essere

```

function save_structure_GUI1 (init , complete_name)

cgm_time = init.cgm_values.time;
cgm_data = init.cgm_values.cgm;
Ts = init.Ts;
cal_time = init.cal_values.time;
cal_data = init.cal_values.cal;
bs_time = init.bs_values.time;
bs_data = init.bs_values.bs;
oth_time = init.oth_values.time;
oth_data = init.oth_values.oth;
data_missing = init.Nmissing;
data_overthre = init.Nthre;
hours_label = init.h_label;

save (complete_name , 'cgm_time' , 'cgm_data' , 'Ts' , 'data_missing' ,
      'data_overthre' , 'hours_label' , 'cal_time' , 'cal_data' , '
      bs_time' , 'bs_data' , 'oth_time' , 'oth_data')

end

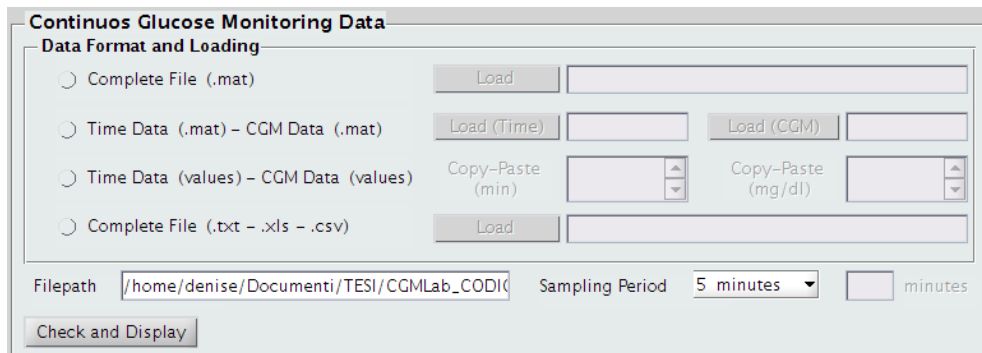
```

A questo punto è possibile scandagliare ulteriormente il lavoro fatto, presentando le scelte implementative riguardanti il comportamento dei singoli pannelli, precedentemente introdotti dal punto di vista del layout. Per ogni pannello si procederà con un'iniziale descrizione del comportamento dei componenti grafici, mentre a seguire saranno presentate in linea generale le scelte riguardanti la gestione dei dati e la codifica relativa.

#### 4.2.1 Dati e callbacks: *Continuous Glucose Monitoring Data* (pannello 1)

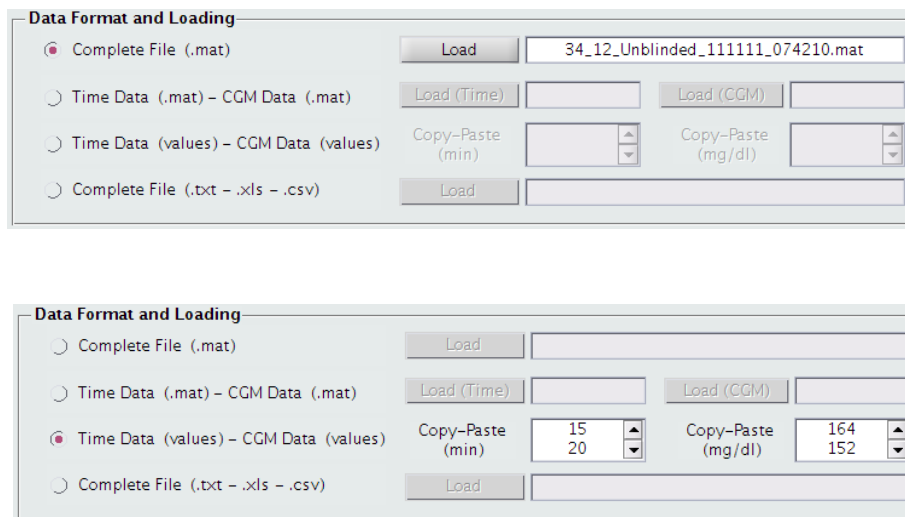
Il pannello *Continuous Glucose Monitoring Data*, dedicato all'inserimento dei dati CGM, appare inizialmente con i quattro radio button deselezionati, i corrispondenti pulsanti e editor di inserimento disabilitati e vuoti, il percorso di file indicante la directory da cui si lancia il software e il periodo di campionamento inizializzato a 5 minuti, come mostrato in Figura 4.8. Come precedentemente detto, i radio button possono essere selezionati in maniera esclusiva, quindi ad ogni sessione di CGMLab è possibile, come è logico attendersi, inserire i dati di interesse di un solo tipo di formato. Conseguentemente alla selezione di uno dei radio button presenti, si abilita il solo gruppo di caselle ed eventualmente di pulsanti a questo corrispondenti e a questo punto è possibile inserire i nomi dei file da caricare.





**Figura 4.8:** *Pannello Continuous Glucose Monitoring Data, come si presenta all'avvio del software.*

A titolo d'esempio, la Figura 4.9 illustra il comportamento grafico dei componenti in seguito a selezione del primo pulsante radio e in seguito a selezione del terzo.



**Figura 4.9:** *Sopra: esempio di inserimento di un MAT-file unico. Sotto: esempio di inserimento manuale dei dati temporali, espressi in minuti, e dei dati di concentrazione glicemica, in mg/dl.*

Per quanto riguarda la casella che visualizza il percorso del file da caricare, è bene precisare che il percorso viene aggiornato automaticamente solo nel caso in cui i nomi dei file vengano inseriti tramite pulsante *Load* e relativa finestra di dialogo. Qualora invece si scegliesse di inserire manualmente da editor il nome dei file allora è di conseguenza necessario modificare sempre manualmente anche il percorso relativo, se questo è diverso dalla cartella da cui si esegue CGMLab. Il problema non si pone se viene selezionato il terzo radio button, dato che in questo caso i valori non devono essere estratti da file, ma vengono direttamente inseriti nella GUI tramite copia-incolla. In basso a sinistra nel pannello troviamo il pop-up menu che permette di indicare il periodo con cui

sono stati campionati i dati. Considerando i periodi di campionamento dei dispositivi CGM in commercio, dal menu è possibile selezionare i valori di 5 minuti, 3 minuti e 1 minuto. Nell'eventualità di voler studiare serie temporali con un campionamento differente è possibile indicarne il valore manualmente, tramite la casella a fianco. Arrivati a questo punto, cliccando su *Check and Display* vengono avviati il caricamento dei dati, un controllo su di essi e una eventuale pre-elaborazione per ottenere i due vettori `init.cgm_values.time` e `init.cgm_values.cgm` contenenti, rispettivamente, una griglia in minuti in cui `init.cgm_values.time(1)=0` e i valori CGM corrispondenti espressi in mg/dl.

Sulla gestione dei dati e la parte algoritmica di pre-elaborazione riguardanti questo primo pannello, è indispensabile fornire alcune precisazioni. La callback invocata dalla pressione del push button *Check and Display* è definita come

```
function CgmCheckLoadCallback(Object , eventdata)
global init;
global g1;

% [codice]

end
```

dove `init` è la struttura dati che contiene le variabili precedentemente elencate e `g1` è la struttura dati che memorizza e permette di richiamare i componenti grafici di layout di tutta la prima interfaccia. `CgmCheckLoadCallback` è una delle funzioni cardini di questo software, sia relativamente alle circa 350 righe di codice di cui è composta sia considerando il fatto che il controllo dei dati e la manipolazione iniziale che si effettua su di essi influenza completamente la gestione degli stessi dati nelle successive elaborazioni. Nelle prime righe di questa funzione vengono inizializzati a 0 il numero di dati mancanti, che possono essere dovuti ad un campionamento non uniforme o a momentanee lacune nell'acquisizione del sensore, e il numero di dati al di fuori del range 39 – 400 mg/dl. Vengono abilitati e resi visibili i componenti del pannello *Dataset Available* e di seguito avviene il check dei dati, a seconda del loro formato di inserimento, attraverso l'utilizzo della variabile `init.id`. Ottenuti i vettori contenenti tempi e valori si procede alla visualizzazione automatica dei dati in tabella e nel grafico. Il codice della funzione assume l'aspetto

```
function CgmCheckLoadCallback(Object , eventdata)
global init;
global g1;

% [inizializzazioni]
% [manipolazione dei componenti]
```

```

% case 'Complete File (.mat)'
if init.id == 1
% [codice]
end
% case 'Time Data (.mat) - CGM Data (.mat)'
if init.id == 2
% [codice]
end
% case 'Time Data (values) - CGM Data (values)'
if init.id == 3
% [codice]
end
% case 'Complete File (.txt - .xls - .csv)'
if init.id == 4
% [codice]
end
% [aggiornamento dei componenti]
end

```

Senza entrare eccessivamente nel dettaglio della codifica, è utile fornire alcune informazioni sull'elaborazione dei dati a seconda del loro formato.

1. Selezionando il radio button 1, vengono inseriti dati contenuti in un unico MAT-file. Inizialmente questa opzione è stata pensata per caricare i dati salvati di una precedente sessione di lavoro del software qui presentato. In questo modo, essendo completamente gestito dal programmatore il salvataggio di variabili e vettori, il caricamento dei dati è immediato. Considerando, però, i file a disposizione per le simulazioni di funzionamento della GUI, sono stati utilizzati anche alcuni dati di dispositivi Dexcom contenuti in un file il cui formato è quello utilizzato internamente dal gruppo di ricerca di Padova. In questo caso, vengono estratti valori identificati da etichette note e la parte più laboriosa riguarda l'elaborazione dei tempi. Dando la possibilità di approfondire dal codice completo quanto qui si accenna, è stata definita una funzione

```

function [data_time] = extract_minutes(name_var,
    time_sampling);
% [codice]
end

```

che fornisce in uscita il vettore dei tempi in minuti `data_time`, a partire dalla variabile `name_var` che contiene il vettore dei tempi in formato MatLab (da convertire con la funzione interna `datevec` che fornisce una matrice a sei colonne [anno mese giorno ora minuto secondo]) e dal periodo di campionamento `time_sampling`. Questo tipo di elaborazione è necessaria in quanto il formato temporale MatLab è accettabile per la visualizzazione dei dati, ma non utilizzabile per l'elaborazione del segnale.

Da questi dati vengono esclusi, sempre in maniera codificata e controllata, i primi valori della serie in quanto non validi e vengono assegnati a `init.t_scal` i minuti da scalare a ciascun tempo per ottenere una griglia temporale che parta da `init.cgm_values.time(1)=0`. Inoltre, con la funzione `function [time data] = over_threshold(time, data, 400, 39)` i dati oltre soglia vengono eliminati e considerati alla pari di dati mancanti. I valori di 400 e 39 mg/dl sono stati fissati in riferimento alle soglie di saturazione dei sensori.

2. Selezionando il radio button 2, vengono inseriti dati contenuti da due MAT-file separati, il primo contenente il vettore dei minuti e il secondo contenente i corrispondenti valori di concentrazione glicemica. Il primo valore del vettore dei tempi viene assegnato alla variabile `init.t_scal`, anche in questo caso per ottenere una griglia temporale che parta da `init.cgm_values.time(1)=0`. Una volta caricati tutti i dati, valutando quelli mancanti e quelli oltre soglia, un check sulla lunghezza dei due vettori `init.cgm_values.time` e `init.cgm_values.cgm` ne consente o ne impedisce l'utilizzo.
3. Selezionando il radio button 3, il vettore dei tempi e il vettore delle concentrazioni CGM vengono inseriti manualmente nei componenti editabile text prestabiliti tramite un semplice comando di copia-incolla. In questo caso il procedimento di pre-elaborazione per ottenere i due vettori `init.cgm_values.time` e `init.cgm_values.cgm` è del tutto simile a quello descritto al punto precedente.
4. Selezionando il radio button 4, vengono inseriti dati provenienti da un unico file in formato testuale o tabellare.

### 4.2.2 Dati e callbacks: *Other Data Types* (pannello 2)

Il pannello *Other Data Types*, dedicato all'inserimento di altre serie temporali, appare inizialmente con i tre check box disabilitati e i corrispondenti pulsanti radio e editor di inserimento oscurati, come mostrato precedentemente in Figura 4.6.

A differenza dei pulsanti radio, i tre check box consentono di scegliere una, due o tutte le tre opzioni presentate. Corrispondentemente alla scelta del tipo di dati da inserire, vengono visualizzati e abilitati i pulsanti radio adiacenti. Si è scelto di lasciare libertà di inserire i dati in modo automatico o manuale qualora venga caricato un file completo nel pannello *Continuous Glucose Monitoring Data*, quindi nel caso in cui vengano scelti come radio button il *Complete File (.mat)* oppure il *Complete File (.txt-.xls-.csv)*. Negli altri due casi è d'obbligo l'inserimento manuale dei valori di interesse e il radio button per l'inserimento diretto da file non compare. In corrispondenza della selezione del terzo tipo di dati, *Other Data*, si apre una finestra di dialogo,

mostrata in Figura 4.10, che dà la possibilità di inserire un'etichetta testuale che identifichi più propriamente i valori inseriti.



**Figura 4.10:** Esempio di inserimento di un'etichetta per il terzo tipo di dati, nel pannello *Other Data Types*.

Completata la scelta di inserimento nel pannello *Data Type and Loading* si clicca su *Check and Display* per caricare e visualizzare i dati.

Anche in questo caso è utile approfondire, almeno per sommi capi, la parte relativa alla codifica di inserimento. La callback invocata dalla pressione del push button *Check and Display* è definita come

```
function CheckLoadOtherCallback(Object , eventdata)
global init;
global g1;

% [codice]

end
```

dove *init* e *g1* sono, come già detto, le strutture dati di variabili e componenti. In *CheckLoadOtherCallback* un controllo iniziale sui dati inseriti, impedisce il caricamento delle serie temporali indicate se non risultano ancora inseriti i dati CGM oppure se il vettore dei tempi e quello dei valori in mg/dl dei tipi di dati indicati non hanno la medesima lunghezza. Tabella e plot del pannello *Dataset Available* vengono aggiornati automaticamente nel momento in cui i dati vengono caricati. Per quanto riguarda la doppia modalità di inserimento dei dati, è utile precisare quanto segue.

- Inserimento copia-incolla da editor: si è accennato al contenuto della variabile *init.t\_scal*, indispensabile se si vuole ottenere un vettore dei tempi il cui primo elemento sia 0. Se i minuti vengono inseriti manualmente devono essere inseriti relativamente al riferimento temporale originario, cioè quello in cui non necessariamente *init.cgm\_values.time(1)=0*. I minuti vengono poi scalati opportunamente per far corrispondere le griglie temporali di dati CGM e dati di altro tipo inseriti.
- Inserimento da file pre-caricato: se il file in questione è relativo a una sessione precedente di CGMLab il caricamento è immediato. Se esso, invece, è un MAT-file di dati Dexcom, il caricamento dei dati avviene

con la chiamata alla funzione `extract_other` così definita

```
function [data_time] = extract_other(name_var_oth,
    name_var_cgm)
global init;

% [codice]

end
```

che fornisce in uscita il vettore dei tempi in minuti relativi ai nuovi dati inseriti, a partire dalla variabile `name_var_oth` che contiene il vettore dei tempi relativo a tali dati e dalla variabile `name_var_cgm` che contiene il vettore dei tempi dei dati CGM. I tempi dei dati da inserire, campionati non uniformemente, e dei dati CGM sono in formato MatLab e non necessariamente i valori di calibrazione o i campioni sanguigni sono prelevati a istanti temporali multipli del periodo di campionamento. Queste considerazioni portano alla codifica, nella funzione `extract_other` definita, di una serie di cicli logici di processamento per estrarre un vettore di valori temporali per i dati scelti che sia compatibile con il riferimento temporale precedentemente codificato. Se invece il file completo pre-caricato è inserito dal pulsante radio *Complete File (.txt-.xls-.csv)* è necessario estrarre i valori per colonna, in relazione all'etichetta delle singole colonne, tenendo in considerazione il fatto che i differenti tipi di dati possono essere di vario tipo, non temporalmente omogenei e separati da virgole, tabulazioni o spazi.

### 4.2.3 Dati e callbacks: *Dataset Available* (pannello 3)

Il pannello *Dataset Available*, che permette di visualizzare i dati a disposizione, appare inizialmente vuoto, consentendo un approccio iniziale con la GUI più immediato perché indirizzato esclusivamente all'inserimento dei dati CGM. Contemporaneamente al click di *Check and Display* ne vengono abilitati i componenti. Vengono resi visibili la tabella, i pulsanti toggle per variare la grafica di visualizzazione del segnale, le informazioni su di esso, il pulsante radio per salvare il pannello grafico e il plot dei dati con il pulsante per lo zoom. In questo caso, non sono presenti comportamenti esclusivi tra i componenti.

La tabella `g1.Table` di visualizzazione dati contiene inizialmente i soli valori CGM, cioè la tabella, all'interno della function `CgmCheckLoadCallback`, è inizializzata con la codifica

```
init.colnames = { 't (min)', 'cgm(mg/dl)' };
init.colformat = { 'numeric', 'numeric' };
init.colwidth = { 100, 100 };
init.coledit = [ false false ];
```

```

init.data_table = [init.cgm_values.time init.cgm_values.cgm];

set(g1.Table, 'Enable', 'on', 'Data', init.data_table, 'ColumnName', ...
    init.colnames, 'ColumnFormat', init.colformat, 'ColumnWidth', ...
    init.colwidth, 'ColumnEditable', init.coledit)

```

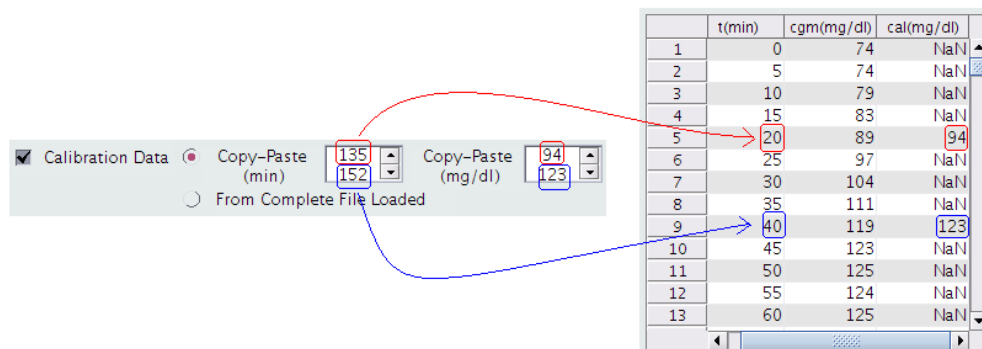
e viene aggiornata interattivamente all'invocazione di `CheckLoadOtherCallback` tramite la funzione

```

[init.colnames init.colformat init.colwidth init.coledit init.
 data_table] ...
    = control_table(init);

```

che, in base alle serie di dati caricati, gestisce l'inserimento di dati nella tabella. Bisogna precisare che in tabella i dati non CGM i cui tempi non corrispondono a multipli del periodo di campionamento vengono visualizzati in corrispondenza del valore temporale CGM futuro più prossimo. Questa scelta è praticamente obbligata per la visualizzazione tabellare dei dati, ma nonostante questo il vettore dei tempi memorizzato contiene i minuti corretti, anche se infra-campionamento. Tenendo conto anche dell'eventuale scalamento dell'asse dei tempi espresso da `init.t_scal`, un esempio schematico di quanto spiegato è illustrato in Figura 4.11.



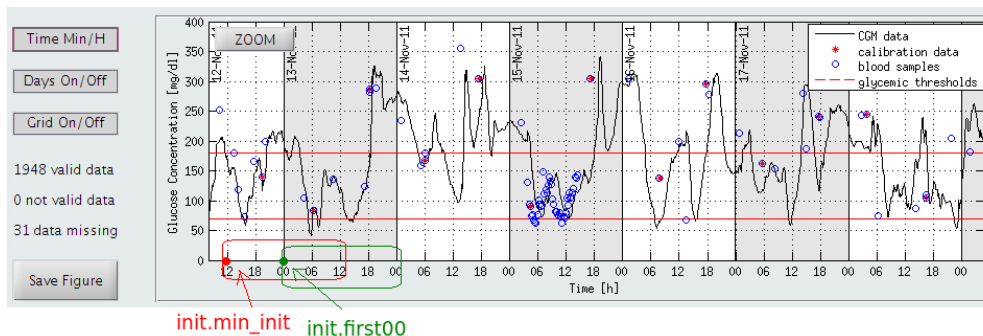
**Figura 4.11:** Esempio di inserimento di dati di calibrazione da editor, con corrispondente visualizzazione dei dati in tabella. In questo specifico caso  $T_s = 5$  minuti e `init.t_scal=115` minuti.

Per quanto riguarda il plot del segnale, si è scelto di settare l'asse dei tempi in base all'ampiezza dell'intervallo temporale da esso ricoperto, a prescindere dal periodo di campionamento, in particolare:

- se `init.cgm_values.time(end) ≥ 1440`: le etichette dei minuti sull'asse x vengono posizionate ogni 500 minuti, quelle orarie ogni 6 ore

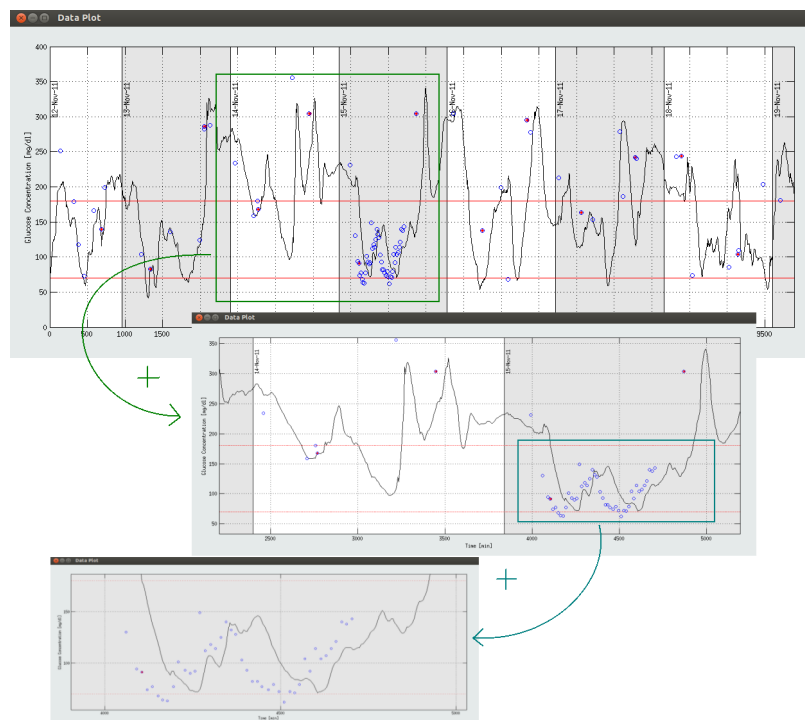
- se `init.cgm_values.time(end) < 1440 & init.cgm_values.time(end) ≥ 200`: le etichette dei minuti sull'asse x vengono posizionate ogni 60 minuti, quelle orarie ogni ora
- se `init.cgm_values.time(end) < 200`: le etichette dei minuti sull'asse x vengono posizionate ogni 10 minuti, quelle orarie ogni ora

Di conseguenza, cliccando su *Time Min/H* si sceglie se visualizzare il profilo glicemico rispetto ai riferimenti temporali in minuti o in ore presentati. Il toggle button *Days On/Off*, invece, è abilitato solo per segnali sufficientemente lunghi in termini di tempo, sempre a prescindere dal numero di campioni che contengono, quindi per quelli in cui `init.cgm_values.time(end) ≥ 1440`. La funzione di questo componente si traduce nella possibilità di visualizzare o meno aree di colore alterno della durata di 24 ore sullo sfondo del grafico, aspetto che facilita la rilevazione di trend del segnale che potrebbero ripetersi quotidianamente. In aggiunta alla semplice visualizzazione di blocchi di 24 ore, se i dati caricati risultano temporalmente localizzati non solo in termini di ore e minuti, ma anche di giorno, mese e anno, allora la suddivisione in giorni rappresenta senza dubbio un valore aggiunto. In particolare, si ricorda che alla variabile `init.first00` viene assegnato il minuto corrispondente al primo minuto del primo giorno completo, mentre a `init.min_init` viene assegnato il minuto corrispondente al primo minuto del blocco di 6 ore completo. In Figura 4.12 viene illustrata la posizione delle due variabili. *Grid On/Off* consente di visualizzare la griglia sullo sfondo del segnale. *Save Figure* invece permette di salvare il grafico in esame in uno dei formati .jpg, .eps o .pdf, da scegliere in una finestra di dialogo interattiva e appositamente abilitata. Il pulsante utilizzato per lo zoom del grafico permette di visualizzare il segnale in un pannello più ampio e consente anche di zoomare specifiche aree del segnale selezionandole con il mouse. Un esempio è illustrato in Figura 4.13.



**Figura 4.12:** Schema che illustra la differenza tra le variabili `init.first00` e `init.min_init`.





**Figura 4.13:** Esempio di zoom di diverse aree del grafico.



# Capitolo 5

## Modulo di CGMLab per lo smoothing

### 5.1 Filtraggio stocastico con filtro di Kalman

Uno degli elementi che limitano l'utilizzo on-line dei sensori CGM è la presenza di rumore di misura. Come verrà delineato in seguito, uno degli aspetti più indagati negli ultimi anni riguarda lo sviluppo di algoritmi di filtraggio o smoothing in grado di rimuovere il rumore di misura dai dati CGM adattandosi alla variabilità inter-individuale e intra-individuale del SNR, problema introdotto nel Capitolo 2.

Gli approcci deterministici per il filtraggio, come si è visto presentando in maniera concisa il filtro a media mobile (che si trova nella maggior parte dei dispositivi CGM), essendo completamente empirici non utilizzano nessun tipo di conoscenza statistica a priori né relativamente alla regolarità del profilo glicemico vero né per quanto riguarda il rumore di misura. Questi sono i motivi per cui la ricerca nel nostro dipartimento si è orientata negli ultimi anni, più precisamente dal 2010, verso un approccio allo smoothing di tipo stocastico, implementato attraverso l'utilizzo del filtro di Kalman (KF). La caratteristica principale di questo algoritmo è la capacità di stimare in tempo reale il valore dei parametri in ciascun individuo in modo da ottimizzare l'azione del filtro sulla base del SNR individuale. Ciò è possibile grazie ad una procedura di stima automatica del SNR, e quindi dei parametri, basata sulla massimizzazione della verosimiglianza della misura ed eseguita in un intervallo di burn-in. Questo permette di fornire, in un contesto di stima bayesiana, un filtraggio ottimo del segnale e di raggiungere il compromesso ottimale tra riduzione del rumore e distorsione del segnale.

#### 5.1.1 Richiami teorici

Il filtro di Kalman è un algoritmo che consente di stimare ricorsivamente e linearmente lo stato  $x \in \mathbb{R}^n$  di un sistema dinamico a partire dalla relazione che

lo lega ad una serie di misure  $y \in \mathbb{R}^n$  soggette a rumore. Per poter essere risolto, il problema richiede di essere formulato come un modello stocastico nella forma di un sistema di stato, quest'ultimo in grado di contenere completamente nel vettore di stato  $x(t)$  tutta la storia passata del sistema in esame. In questo modo il problema di filtraggio ottimo si risolve come problema di stima dello stato del sistema.

Si fa riferimento ad una situazione in cui il segnale incognito e le misure a partire dalle quali vogliamo stimarlo possono essere messi in relazione mediante un modello della forma

$$x(t+1) = Fx(t) + w(t) \quad (5.1a)$$

$$y(t) = Hx(t) + v(t) \quad (5.1b)$$

dove  $x(t)$  è il vettore di stato da stimare, di dimensione  $N$ ,  $F$  è la matrice di transizione di stato, di dimensione quadrata  $N \times N$ ,  $y(t)$  è il vettore delle misure, di dimensione  $n$  (solitamente  $n \ll N$ ) e  $H$  è la matrice di trasferimento stato-uscita, di dimensione rettangolare  $n \times N$ . La prima equazione permette di calcolare lo stato del sistema ad un certo istante a partire dallo stato all'istante precedente e viene denominata, per questo, equazione di aggiornamento dello stato. Si parla, invece, di equazione delle misure o equazione di uscita riferendosi alla seconda delle due. In aggiunta,  $w(t)$  è il vettore, di dimensione  $N$ , che rappresenta il rumore del processo ed è aleatorio, considerato in prima approssimazione a media nulla e a campioni scorrelati, con matrice di covarianza  $\text{cov}(w) = Q$  incognita e di dimensione  $N \times N$ . Similmente,  $v(t)$ , rumore di misura, è un vettore di dimensione  $n$ , aleatorio, non correlato a  $w(t)$ , a media nulla e a campioni scorrelati, con matrice di covarianza  $\text{cov}(v) = R$  di dimensione  $n \times n$ . Assumendo per entrambi gli errori una distribuzione gaussiana, la loro descrizione statistica si può riassumere con le espressioni

$$p(w) \sim N(0, Q)$$

$$p(v) \sim N(0, R)$$

Ponendoci in un contesto stazionario, l'obiettivo per risolvere il sistema è dato dal costruire delle espressioni ricorsive che, ad un certo istante  $t+1$ , forniscano la stima lineare a minima varianza d'errore (LMVE) dello stato  $x$  sfruttando lo stesso tipo di stima dello stato all'istante  $t$  precedente. Ciò è possibile in quanto il filtro di Kalman stima lo stato utilizzando una sorta di controllo a retroazione tenendo in considerazione l'errore di misura. In relazione a ciò, le equazioni del KF sono ottenute in due passi: un primo passo predittivo ed un passo correttivo successivo. Siamo al tempo  $t$  e disponiamo della stima LMVE dello stato in  $t$ ,  $\hat{x}(t|t) = E[x(t|t)]$ , originata dalla conoscenza di tutti i dati misurati fino al tempo  $t$  compreso. Tale stima è affetta da errore di stima, a media nulla e matrice di covarianza  $P(t|t) = E[(x(t) - \hat{x}(t|t))(x(t) - \hat{x}(t|t))^T]$  che per semplicità esprimiamo con  $P(t)$ . In un contesto bayesiano, ci si pone il problema di calcolare una stima dello stato valida al tempo  $t+1$ ,  $\hat{x}(t+1|t+1) =$

$E[x(t+1|t+1)]$ , sfruttando come informazione a priori la predizione dello stato valida per il tempo  $t+1$  ma fatta sulla base dei dati di misura collezionati fino a  $t$ ,  $\hat{x}(t+1|t) = E[x(t+1|t)]$ . Senza guardare, quindi, i dati al tempo  $t+1$  è possibile fare una predizione ad un passo, sostituendo  $w(t)$  con il suo valore atteso, cioè imponendo  $w(t) = 0$ , e ottenendo

$$\hat{x}(t+1|t) = F\hat{x}(t|t) \quad (5.2)$$

La covarianza dell'errore di predizione ad un passo è ottenibile propagando le covarianze tramite l'espressione

$$P(t+1|t) = FP(t)F^T + Q \quad (5.3)$$

Successivamente, disponendo della misura al tempo  $t+1$ , utilizzando la predizione ad un passo come prior bayesiano e sfruttando l'equazione delle misure al tempo  $t+1$ , si ottiene che la stima LMVE dello stato al tempo  $t+1$  si può scrivere come

$$\begin{aligned} \hat{x}(t+1|t+1) &= \hat{x}(t+1, t) + \\ &+ P(t+1, t)H^T [HP(t+1|t)H^T + R]^{-1} [y(t+1) - H\hat{x}(t+1|t)] \end{aligned} \quad (5.4)$$

ed è affetta da un errore di misura a media nulla e matrice di covarianza data da

$$\begin{aligned} P(t+1) &= P(t+1|t) \\ &- P(t+1|t)H^T [HP(t+1|t)H^T + R]^{-1} HP(t+1|t) \end{aligned} \quad (5.5)$$

In aggiunta, è possibile definire la matrice di guadagno del filtro, di dimensione  $N \times n$ , e il vettore di errore di predizione, di dimensione  $m$ , rispettivamente come

$$K(t+1) = P(t+1|t)H^T [HP(t+1|t)H^T + R]^{-1} \quad (5.6)$$

$$e(t+1) = y(t+1) - H\hat{x}(t+1|t) \quad (5.7)$$

Queste espressioni consentono di compattare lo step correttivo e di ottenere le espressioni

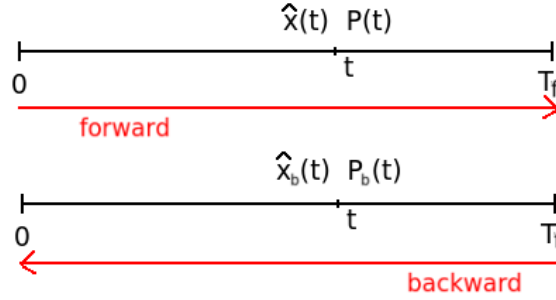
$$P(t+1) = [I_N - k(t+1)H]P(t+1|t) \quad (5.8)$$

$$\hat{x}(t+1|t+1) = \hat{x}(t+1|t) + K(t+1)e(t+1) \quad (5.9)$$

L'applicazione di un algoritmo ricorsivo richiama anche il problema delle condizioni iniziali, da fissare in maniera consona al contesto di stima in esame. L'inizializzazione è del tipo  $P(0) = P_0$  e  $\hat{x}(0|0) = x_0$ . I dettagli delle scelte implementative verranno discussi in seguito.

Relativamente allo smoothing implementato nel software che in questa tesi viene presentato, si è scelto di applicare una variante del filtro di Kalman

per ottenere una stima migliorata rispetto a quella ottenuta con il solo KF in avanti visto fino ad ora. Questo tipo di filtraggio deriva da uno studio pubblicato nel 1965 (poi sistemare riferimento bibliografico) e prende il nome di Rauch-Tung-Striebel Two-Pass Smoother da chi lo ha implementato per la prima volta. Il filtro di Rauch-Tung-Striebel permette di migliorare le stime ottenute con il filtro di Kalman forward (in avanti) completandolo con un filtraggio backward (all'indietro). Concretamente, la prima stima è ottenuta con il procedimento appena descritto, utilizzando all'istante  $t$  solo le misure precedenti, mentre la seconda stima è ricavata all'indietro sfruttando, al tempo  $t$ , solo le misure future mediante una modifica dell'algoritmo che gli consente di operare ricorsivamente all'indietro a partire dall'ultimo tempo di osservazione  $T_f$ . In Figura 5.1 viene illustrata un'interpretazione grafica del fenomeno, in cui la stima  $\hat{x}(t)$  in avanti con matrice di covarianza dell'errore data da  $P(t)$  si combina ad una stima  $\hat{x}_b(t)$  con matrice di covarianza associata  $P_b(t)$ .



**Figura 5.1:** Interpretazione grafica del comportamento del filtro di Rauch-Tung-Striebel.

La stima finale è poi ottenuta pesando in maniera ottima le due stime, cioè con peso inversamente proporzionale alla propria varianza d'errore, attraverso l'espressione

$$\hat{x}(t|T_f) = P(t|T_f) [P^{-1}(t)\hat{x}(t) + P_b^{-1}(t)\hat{x}_b(t)] \quad (5.10)$$

dove

$$P(t|T_f) = AP(t)A^T + (I_N - A)P_b(t)(I_N - A)^T \quad (5.11)$$

con

$$A = P_b(t) [P(t) + P_b(t)]^{-1} \quad (5.12)$$

Affinché sia possibile utilizzare il filtro di Kalman, così come è stato presentato, per ridurre il rumore nei dati CGM, è necessario approfondire due aspetti fondamentali:

- l'applicazione dell'algoritmo per lo smoothing richiede che sia fornita una descrizione a priori sul segnale glicemico reale, affinché sia possibile l'espressione del problema tramite modello di stato;

- le matrici di covarianza  $Q$  e  $R$  rappresentano due parametri chiave nella determinazione della stima dello stato ed è necessario, pertanto, determinarle

### 5.1.2 Descrizione statistica di $u(t)$

Il primo aspetto da sviluppare richiede che venga formulata una descrizione del reale, ma incognito, profilo glicemico  $u(t)$ . Dagli studi effettuati in questi anni [31], fondati su simulazioni e supportati da un background di conoscenze fisiologiche, si è stabilita la possibilità di esprimere la regolarità del segnale incognito pensando quest'ultimo come realizzazione di un processo stocastico in cui tra un campione e il successivo la velocità varia in maniera regolare e continua. Ciò si traduce nel formulare  $u(t)$  come un processo di rumore bianco integrato due volte, cioè come modello random-walk integrato, di seguito espresso come

$$u(t) = 2u(t-1) - u(t-2) + w(t) \quad (5.13)$$

dove  $w(t)$  è un rumore bianco, a media nulla, con varianza incognita uguale a  $\lambda^2$ . Diversi sono i modelli che si possono ottenere integrando un processo di rumore bianco e questa diversità risiede nella scelta del numero  $m$  di integratori, indice della continuità del segnale. Modelli di questo tipo vengono utilizzati come informazione a priori sui dati perché descrivono non tanto il segnale, ma piuttosto il suo modo di variare tra un campione e l'altro. La scelta di 2 integratori emerge da studi di simulazione in cui si è applicata la Cross-Validation (CV) e in cui si evince che questo tipo di modello rappresenta il giusto compromesso tra random-walk ( $m = 1$ ), ottimo per descrivere processi stocastici molto irregolari, e modello random-walk doppiamente integrato ( $m = 3$ ), che risulta eccessivamente regolare per poter rappresentare la variabilità del profilo glicemico.

Cercando di portare l'espressione (5.13) in forma di modello di stato, sono necessarie due variabili: una variabile di stato  $x_1(t) = u(t)$ , per il segnale incognito al tempo  $t$ , e una variabile addizionale per lo stato all'istante precedente,  $x_2(t) = u(t-1)$ . L'equazione (5.13) diventa

$$x(t+1) = Fx(t) + Gw(t) \quad (5.14)$$

dove  $x(t) = [x_1(t)x_2(t)]^T$  è il vettore di stato al tempo  $t$ , mentre  $F$  e  $G$  sono date da

$$F = \begin{bmatrix} 2 & -1 \\ 1 & 0 \end{bmatrix} \quad G = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \quad (5.15)$$

Ricordando il fatto che la misura CGM ad un certo istante rappresenta il solo valore in uscita del sistema, la misura  $y(t)$  si riduce a uno scalare, perciò l'equazione di misura si esprime come

$$y(t) = Hx(t) + v(t) \quad (5.16)$$

con  $H = [0 \quad 1]$ .

### 5.1.3 Stima delle matrici Q ed R e criteri di regolarizzazione

Nel momento in cui l'ordine del modello a priori per  $u(t)$  è stato fissato, anche le matrici di covarianza dell'errore Q e R assumono una loro espressione, data da

$$Q = \begin{bmatrix} \lambda^2 & 0 \\ 0 & 0 \end{bmatrix} \quad R = \sigma^2 \quad (5.17)$$

Per poter calcolare una stima del glucosio  $\hat{u}(t)$ , entrambi i valori  $\lambda^2$  e  $\sigma^2$  (varianza del rumore di processo e di misura) devono essere noti. Sfortunatamente tali parametri sono generalmente incogniti e devono, quindi, essere stimati in tempo reale a partire dai dati a disposizione. In aggiunta a questo, il rapporto segnale-rumore varia da sensore a sensore e da individuo a individuo, cosa che rende impensabile l'impiego di  $\lambda^2$  e  $\sigma^2$  universali, ma che invece richiede una procedura di stima individualizzata e realizzabile in tempo reale. Il maggior problema nell'applicazione online del filtro di Kalman rimane, quindi, la determinazione di Q ed R, aspetto che si presenta del tutto simile alla determinazione dei parametri di regolarizzazione nello smoothing bayesiano.

L'ingresso incognito di un sistema endocrino-metabolico lineare, quale appunto la secrezione di un ormone, è stimato a partire da un modello delle misure espresso, nel nostro caso, nella forma matriciale-vettoriale

$$y = Gu + v \quad (5.18)$$

dove  $y$  e  $v$  sono i vettori  $n$ -dimensionali dei dati e dell'errore di misura,  $u$  è il vettore  $N$ -dimensionale dell'ingresso incognito ( $N \geq n$  perché, essendo in presenza di segnali con possibili dati mancanti, viene impiegata una griglia virtuale nei calcoli) e  $G$  è una matrice  $n \times N$  ottenuta da una matrice identità  $N \times N$ , eliminando le righe corrispondenti ai dati mancanti. Considerando in generale la stima di  $u$  attraverso approssimazione dei dati  $y$  come problema di stima lineare a minima varianza d'errore (LMVE), ciò che si fa è cercare la funzione  $\hat{u}$  che meglio approssima questi dati, cioè che minimizza  $E[\|u - \hat{u}\|^2]$ . L'espressione è data da

$$\hat{u} = \underset{u}{\operatorname{argmin}} \{ (y - Gu)^T (y - Gu) + \gamma u^T L^T L u \} \quad (5.19)$$

in cui la funzione costo tiene in considerazione sia la fedeltà ai dati sia la regolarità del segnale approssimante.  $L$  è una matrice triangolare inferiore di Toeplitz di dimensione  $N \times N$ , dipendente da  $m$  (numero di integratori), la cui prima colonna è data da  $[1 \quad -2 \quad 1 \quad 0 \quad \dots \quad 0]^T$  e  $\gamma = \frac{\sigma^2}{\lambda^2}$  è chiamato parametro di regolarizzazione. La soluzione dell'equazione (5.19) è

$$\hat{u} = (G^T G + \gamma L^T L)^{-1} G^T y \quad (5.20)$$

ed è chiaramente una funzione dei due parametri  $m$ , fissato a 2, e  $\gamma$ . A seconda del fatto che  $\sigma^2$  sia o meno incognita, sono diversi i criteri a base statistica che



permettono di trovare il valore di  $\gamma$  che determina un compromesso ottimale tra bias e varianza della soluzione.

Quando sia  $\lambda^2$  che  $\sigma^2$  sono incogniti, il problema di minimizzazione espresso dalla (5.19) viene risolto ripetutamente per diversi valori del parametro di regolarizzazione finché

$$\frac{\text{WRSS}(\gamma)}{n - q(\gamma)} = \gamma \frac{\text{WESS}(\gamma)}{q(\gamma)} \quad (5.21)$$

dove  $n$  è il numero dei dati di misura nell'intervallo predeterminato in cui si effettua la stima dei parametri,  $\text{WESS}(\gamma) = \hat{u}^T L^T L \hat{u}$  è la somma pesata delle stime al quadrato (weighted estimates sum of squares),  $\text{WRSS}(\gamma) = (y - G(\hat{u}))^T (y - G(\hat{u}))$  è la somma pesata dei residui al quadrato (weighted residuals sum of squares) e  $q(\gamma) = \text{Trace}[G^T G + \gamma L^T L]^{-1}$  è il termine reale positivo che rappresenta i gradi di libertà associati a  $\gamma$ . I gradi di libertà variano con continuità da 0 a  $n$  in maniera monotona decrescente all'aumentare di  $\gamma$ . Questo criterio è il cosiddetto criterio Maximum Likelihood 2 (ML2) in quanto il valore di  $\gamma$  stimato è quello che massimizza la verosimiglianza del vettore misurato  $y$ . Una volta determinato  $\gamma$ , la stima a posteriori di  $\sigma^2$  e  $\lambda^2$  è ottenibile come

$$\sigma^2 = \frac{\text{WRSS}(\gamma)}{n - q(\gamma)} \quad \lambda^2 = \frac{\sigma^2}{\gamma} \quad (5.22)$$

Quando  $\sigma^2$  è noto, il parametro  $\gamma$  può essere stimato scegliendo tre metodi diversi. Volendo proseguire con criteri statisticamente consistenti, il criterio Maximum Likelihood 1 (ML1) calcola il valore di  $\gamma$  che massimizza la verosimiglianza di  $y$  rispetto al parametro  $\lambda^2$  quando  $\sigma^2$  è noto. In particolare il problema di minimizzazione viene anche in questo caso risolto ripetutamente per diversi valori del parametro di regolarizzazione finché

$$\text{WESS}(\gamma) = \lambda^2 q(\gamma) \quad (5.23)$$

con  $\lambda^2 = \frac{\sigma^2}{\gamma}$ .

Parallelamente a questo criterio ne esiste un altro che calcola il valore di  $\gamma$  massimizzando la verosimiglianza di  $y$  rispetto a  $\sigma^2$  quando  $\lambda^2$  è noto. Nella realtà, però,  $\lambda^2$  non è mai noto perciò si sceglie di utilizzare questo approccio nel caso in cui  $\sigma^2$  è noto, scelta che rende questo criterio più debole dal punto di vista teorico rispetto ai precedenti. Tale criterio coincide con il criterio Minimum Risk (MR) sviluppato da Hall e Titterington nel 1987. Esso prevede di scegliere  $\gamma$  in modo tale che sia soddisfatta, a meno di un valore di tolleranza, l'uguaglianza

$$\text{WRSS}(\gamma) = \sigma^2 n - q(\gamma) \quad (5.24)$$

con  $\sigma^2 = \gamma \lambda^2$ .

Un altro criterio implementabile solo se si conosce la varianza dell'errore di misura è il cosiddetto criterio di discrepanza, proposto da Twomey nel 1965, che suggerisce di scegliere  $\gamma$  in modo che

$$\text{WRSS}(\gamma) = \sigma^2 n \quad (5.25)$$

Tale criterio nasce dall'osservazione che, essendo il vettore dei residui  $(y - \hat{u})$  interpretabile come una stima dell'errore di misura  $v$ , è ragionevole attendersi che, se  $\gamma$  produce una stima  $\hat{u}$  abbastanza accurata allora l'energia dei residui pesati  $(y - G\hat{u})^T(y - G\hat{u})$  dovrebbe essere all'incirca uguale a  $E[v^T v] = \sigma^2 n$ . Senza approfondire oltremodo la cosa, è giusto sottolineare che il criterio di discrepanza, non tenendo conto dell'errore di bias nel calcolo della covarianza dell'errore di stima, conduce in media a oversmoothing.

Si è accennato in precedenza ai gradi di libertà associati al parametro di regolarizzazione, espressi dal termine  $q(\gamma)$ , i quali variano con continuità da 0 a  $n$ , dove  $n$ , ricordiamo, è il numero di misure. I criteri di regolarizzazione esposti, alcuni più di altri, essendo implementati mediante algoritmi ricorsivi, richiedono un aggiustamento iterativo della stima fino alla convergenza verso il valore ottimo di  $\gamma$ . In alcuni casi è possibile che non si giunga a convergenza, non per problemi algoritmici, ma per il fatto che approcci che portano ad un completo successo con dati simulati possono non dare i risultati attesi, in piccola percentuale, nel momento in cui sono applicati a dati reali, considerando anche il fatto che le approssimazioni che vengono fatte sul segnale non sempre descrivono perfettamente la situazione reale. Per far fronte a questa evenienza, si è scelto di dare la possibilità, parallelamente alla scelta degli altri criteri, di fissare i gradi di libertà per la funzione da stimare, agendo in maniera forzata sull'intensità della sua regolarizzazione. Essendo, però,  $q(\gamma)$  dipendente da  $n$ , il valore che assume non permette di confrontare in maniera significativa profili glicemici diversi. Si preferisce quindi, per comparare il grado di regolarizzazione introdotto in due diversi problemi, utilizzare al posto di  $q(\gamma)$  il termine  $q(\gamma)/n$ , altrimenti detto gradi di libertà normalizzati. Questo indice, più intuitivo del precedente, assume valori da 0 (se  $\gamma \rightarrow +\infty$ , nessuna libertà) a 1 (se  $\gamma \rightarrow 0$ , massima libertà).

## 5.2 Implementazione dello smoothing e variabilità del SNR

Dopo aver fissato le caratteristiche a priori di  $u(t)$ , averne determinato il corrispondente modello di stato e aver individuato la procedura di stima dei parametri incogniti, rimane da definire la metodologia implementativa di smoothing da applicare ai dati. In linea del tutto generale, su una serie temporale di dati di questo tipo è possibile:

- fare denoising a posteriori, off-line;
- applicare un denoising causale, tra una calibrazione e l'altra (numerosità elevata di campioni), con riconciliazione a posteriori;
- applicare un denoising causale, a compartimenti stagni (numerosità contenuta di campioni), senza riconciliazione.

Coerentemente con gli scopi di questo lavoro, data la necessità di poter simulare un funzionamento in tempo reale degli algoritmi impiegati, si è scelto di percorrere la strada dello smoothing causale senza riconciliazione dei blocchi. In questo contesto, ci si è indirizzati verso i due approcci presentati negli articoli di Facchinetti et al. [26] [27], applicando in entrambi i casi la nuova procedura di filtraggio stocastico basata su filtro di Kalman.

Il primo tipo di implementazione, identificato in [32] con il nome Burn-In Interval Approach, prevede di dedicare una finestra iniziale di qualche ora di acquisizione alla stima dei parametri incogniti  $\lambda^2$  ed eventualmente  $\sigma^2$ . Il processo di stima avviene una sola volta, inizialmente, tiene conto della variabilità del rapporto segnale-rumore tra sensori differenti e tra soggetti diversi e ipotizza che tale rapporto si mantenga costante durante tutta l'acquisizione. Per una stima sufficientemente affidabile, gli algoritmi visti in (sezione in cui si parla dei criteri di stima – convergenza) vengono generalmente applicati su un numero di dati iniziali variabile tra 80 e 120. Ciò non impedisce tuttavia, per scopi didattici, di estendere la scelta del numero di dati di burn-in ad un intervallo più ampio. Ottenuta la stima dei parametri individualizzata, viene applicato il filtro di Kalman al resto dei dati. Tuttavia, bisogna considerare il fatto che la condizione di SNR costante ipotizzata in questo approccio è nel concreto difficilmente riscontrabile in dati CGM.

Per gestire la presenza di un rapporto segnale-rumore variabile all'interno di una stessa seduta di monitoraggio è quindi necessario seguire un secondo approccio, denominato Sliding Window Approach [32]. Anche in questo tipo di implementazione alcuni dati del primo blocco di smoothing vengono impiegati per la stima dei parametri incogniti, ma ciò si ripete ulteriormente dopo un certo intervallo di tempo. Detto altrimenti, in questo specifico caso, in una finestra scorrevole di lunghezza fissata i dati iniziali sono utilizzati per la stima di  $\lambda^2$  ed eventualmente di  $\sigma^2$ , ed ogni volta che viene acquisito un nuovo dato tale finestra scorre in avanti nel tempo di un campione e gli stessi parametri vengono stimati nuovamente. Nonostante il filtro venga applicato su tutta la finestra selezionata, l'algoritmo tiene in memoria solo la stima dell'ultimo campione della finestra mobile, cioè del nuovo campione acquisito dal sensore. Si può intuire facilmente che maggiore sarà la lunghezza della finestra, maggiore sarà l'informazione a disposizione e più accurata risulterà la stima ottenuta. Questo tipo di approccio risulta molto utile là dove la variabilità del rapporto segnale-rumore è elevata, perché consente di adattare il filtraggio ai cambiamenti della componente di rumore. Tuttavia, un procedimento di questo tipo risulta computazionalmente molto dispendioso, soprattutto se l'intenzione è quella di eseguire la procedura più velocemente del periodo di campionamento, per simularne il funzionamento real-time. Essendo necessario, in relazione alla variabilità intra-individuale del SNR, riadattare periodicamente i parametri da utilizzare nel filtro di Kalman, di seguito vengono presentate alcune accortezze per velocizzare il processo di stima. La differenza tra i due approcci presentati si evince dalla Figure 5.2 e 5.3.

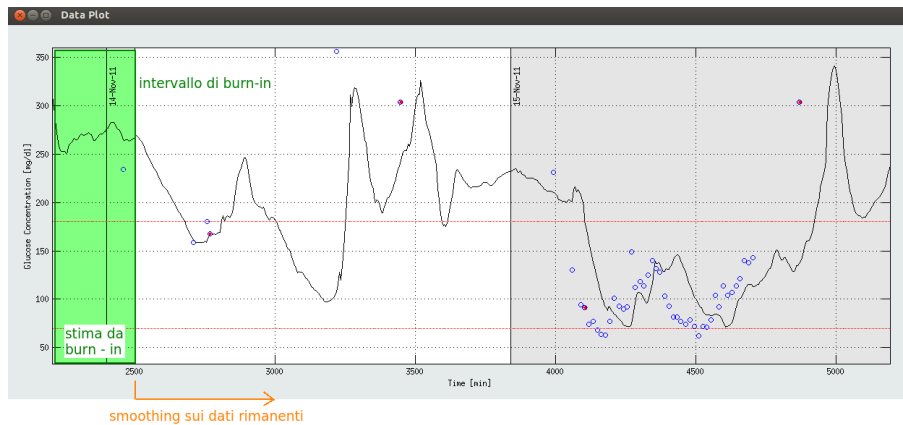


Figura 5.2: Schema intuitivo dell'applicazione del Burn-In Interval Approach.

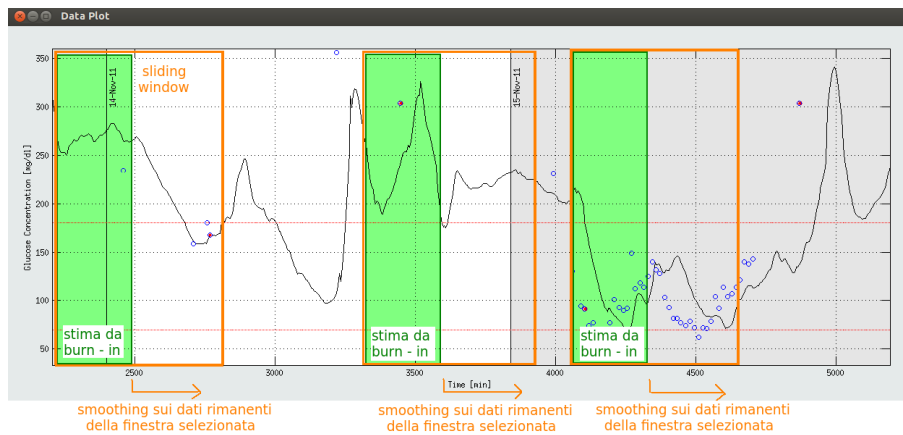


Figura 5.3: Schema intuitivo dell'applicazione dello Sliding Window Approach.

### 5.3 Aspetti numerici: diagonalizzazione

In questo paragrafo verrà mostrato come alcune scelte algoritmiche possano rendere più veloce ed efficiente il calcolo della soluzione al variare del parametro di regolarizzazione, accelerando il processo di convergenza e quindi il conseguimento della stima finale dell'ingresso. Relativamente a quanto detto in precedenza, tra i calcoli necessari a perseguire lo smoothing del segnale di misura ci si trova a risolvere l'equazione

$$\hat{u} = (G^T G + \gamma L^T L)^{-1} G^T y \quad (5.26)$$

in cui il calcolo della matrice inversa, da effettuare ad ogni  $\gamma$ -tentativo, richiede  $O(N^3)$  operazioni. Una complessità così elevata rende inefficiente l'algoritmo, soprattutto considerando la necessità di lavorare sul segnale CGM in tempo reale. Risulta conveniente effettuare preliminarmente una procedura di diagonalizzazione, cioè un cambio di coordinate che porti il problema in

forma per l'appunto diagonalizzata, permettendo così di determinare  $\gamma$  utilizzando semplici espressioni scalari in maniera tale da ridurre la complessità computazionale.

Si definisca la matrice  $H = GL^{-1}$ , di dimensione  $n \times N$ . Si consideri ora la Singular Value Decomposition (SVD) di  $H$  e siano  $U$  e  $V$  le due matrici unitarie, di dimensione  $n \times n$  e  $N \times N$  rispettivamente, tali che

$$U^T U = I_n \quad V^T V = I_N \quad U^T V = D \quad (5.27)$$

dove  $D$  è una matrice diagonale  $n \times N$  i cui elementi diagonali sono  $d_i$ , con  $i = 1, 2, \dots, n$  cioè

$$D = \text{diag}\{d_i\} = \begin{bmatrix} d_1 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 \\ 0 & d_2 & 0 & 0 & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & d_{n-1} & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & 0 & d_n & 0 & \cdots & 0 \end{bmatrix} \quad (5.28)$$

La complessità di questo step è  $O(n^3)$ . Si consideri adesso il cambio di base nello spazio delle misure  $\xi = U^T B^{-1/2} y, \varepsilon = U^T B^{-1/2} v$  e nel dominio dell'ingresso  $\eta = V^T F U$ . Nelle nuove coordinate l'equazione  $y = u + v$  diventa

$$\xi = D\varepsilon + \eta \quad (5.29)$$

con  $\text{cov}(\varepsilon) = \sigma^2 I_n$  e  $\text{cov}(\eta) = \lambda^2 I_N$ . Nelle nuove basi, i vettori aleatori  $\varepsilon$  e  $\eta$  hanno quindi componenti estratte da due processi aleatori di rumore bianco a varianza  $\sigma^2$  e  $\lambda^2$  rispettivamente. Nelle nuove coordinate la stima dell'ingresso regolarizzata ottenuta dal modello delle misure (5.29) è

$$\hat{\eta} = (D^T D + \gamma I_N)^{-1} D^T \xi \quad (5.30)$$

La struttura diagonale della matrice  $D$  consente di ottenere la stima in maniera disgiunta, componente per componente, attraverso un il sistema di  $n$  equazioni scalari indipendenti

$$\hat{\eta}_i = \begin{cases} \frac{d_i}{d_i^2 + \gamma} \xi_i & i = 1, \dots, n \\ 0 & i = n + 1, \dots, N \end{cases} \quad (5.31)$$

Si noti che le misure  $\xi_i$ , con  $i = 1, 2, \dots, n$ , non recano informazione a posteriori sulle variabili  $\eta_i$  per  $i = n+1, n+2, \dots, N$ . Non essendo disponibile informazione a posteriori su tali variabili, lo stimatore bayesiano non può che attribuirgli come stima il suo valore atteso a priori, cioè zero, con varianza pari a quella a priori  $\lambda^2$ . Tramite l'ausilio delle nuove coordinate è facile calcolare le quantità WESS e WRSS, ricavando immediatamente

$$WESS(\gamma) = \sum_{i=1}^n \left( \frac{d_i \xi_i}{d_i^2 + \gamma} \right)^2 \quad (5.32)$$

$$WRSS(\gamma) = \sum_{i=1}^n \left( \frac{\gamma \xi_i}{d_i^2 + \gamma} \right)^2 \quad (5.33)$$

I gradi di libertà equivalenti necessari a verificare che siano soddisfatti i criteri di stima si ottengono anch'essi mediante operazioni scalari come

$$q(\gamma) = \sum_{i=1}^n \frac{d_i^2}{d_i^2 + \gamma} \quad (5.34)$$

Una volta trovato il valore ottimo di  $\gamma$ , la stima dell'ingresso nelle coordinate originali è ottenuta da

$$\hat{u} = F^{-1}V\hat{\eta} \quad (5.35)$$

Riassumendo, eseguito il cambio di coordinate, la stima dell'ingresso nella nuova base può essere quindi ottenuta al variare di  $\gamma$  con sole  $O(n)$  operazioni scalari tramite l'equazione (5.31). Allo stesso modo, con sole  $O(n)$  operazioni scalari si calcolano residui, gradi di libertà e le due quantità WESS e WRSS. Lo sforzo principale di questo algoritmo è quindi il calcolo della SVD della matrice  $H$ . Questo richiede  $O(n^3)$  operazioni ma, essendo indipendente da  $\gamma$ , viene eseguito una sola volta. Ogni volta che  $\gamma$  viene calcolato sono quindi richieste solo  $O(n)$  operazioni, a differenza delle  $O(N^3)$  dell'algoritmo originale. Si osservi inoltre che l'efficienza della procedura di diagonalizzazione nella selezione del parametro di regolarizzazione non dipende dal fatto che  $G$  è di Toeplitz, cosicché l'algoritmo possa essere applicabile anche nel caso di sistemi tempo-varianti e con griglia di campionamento, come possibile nel nostro caso, non uniforme.

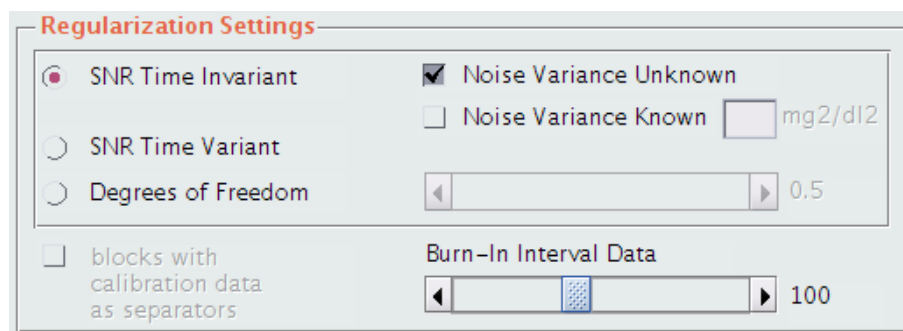
## 5.4 Progettazione del modulo per lo smoothing

Il primo modulo di processamento realizzato è quello dedicato alla regolarizzazione del segnale CGM. Tale modulo visualizzato come sotto-pannello della seconda finestra grafica, identificata con il nome *CGMLab Signal Processing*, e viene abilitato in seguito alla selezione del radio button corrispondente, situato nella parte superiore della finestra assieme agli altri radio button identificanti i moduli di elaborazione.

Osservando la finestra così come è stata presentata in §3.4, si nota che la sub-interfaccia *Smoothing* è composta da quattro pannelli principali. Un pannello permette di impostare il tipo di regolarizzazione da effettuare, in relazione al rapporto segnale-rumore che caratterizza i dati CGM e di cui si è parlato in §4.2.1. Altri due pannelli sono dedicati rispettivamente alla selezione del criterio di regolarizzazione da applicare per la stima dei parametri e alla visualizzazione di tali stime. Nella metà inferiore della finestra, infine, vengono visualizzati i risultati in forma tabellare e in forma grafica. In maniera del tutto simile alla finestra di inserimento dei dati, i pulsanti posizionati all'estremo inferiore permettono di resettare quanto calcolato, di salvare i risultati o di cambiare modulo funzionale.

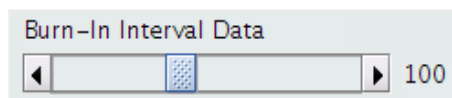
### 5.4.1 Layout: *Regularization Settings* e *Regularization Criteria* (pannello 1 e 2)

Il pannello *Regularization Settings*, mostrato in Figura 5.4, è composto da due parti entrambe preposte all'impostazione del tipo di regolarizzazione da effettuare. Tre pulsanti radio mutuamente esclusivi permettono di gestire le caratteristiche del profilo stimato regolarizzato, consentendo di ipotizzare che il rapporto segnale-rumore dei dati CGM acquisiti si mantenga costante nel tempo o sia variabile all'interno di una stessa acquisizione oppure dando la possibilità di fissare il grado di regolarizzazione del profilo da stimare. Nel caso di rapporto segnale-rumore tempo invariante è possibile considerare la varianza del rumore di misura costante, fissata e nota oppure sconosciuta e da stimare, attraverso due check box anch'essi esclusivi. Per quanto riguarda, invece, il terzo pulsante radio, la scelta del grado di regolarizzazione del profilo glicemico coincide con la scelta, attraverso la manipolazione di una barra scorrevole, dei gradi di libertà normalizzati, di cui si è parlato in §4.1.3.



**Figura 5.4:** *Pannello Regularization Settings.*

Nella parte inferiore del pannello viene richiesto di fissare il numero di dati di burn-in in relazione ai quali stimare i parametri incogniti del segnale CGM. Anche in questo caso, come è possibile vedere in Figura 5.5, la scelta avviene intervenendo sulla barra scorrevole dedicata, cioè trascinando il riferimento centrale o agendo sulle frecce direzionali agli estremi.

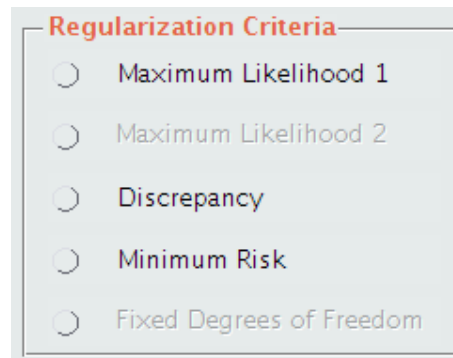


**Figura 5.5:** *Slider per l'impostazione del numero di dati di burn-in.*

Si è scelto di presentare, ma per il momento di non implementare, la possibilità di effettuare lo smoothing del segnale per compartimenti separati dagli istanti di calibrazione, qualora inseriti, come si può osservare nell'angolo sinistro inferiore di Figura 5.4. Questa idea, da implementare eventualmente in futuro, è nata con l'obiettivo di ottenere un profilo CGM regolarizzato che

tenga conto dell'inaccuratezza dei dati immediatamente precedenti alle calibrazioni e aggiusti in maniera automatica l'andamento del profilo regolarizzato in corrispondenza di tali dati nel momento di riconciliazione dei blocchi.

Adiacente al pannello *Regularization Settings* è stato costruito il pannello *Regularization Criteria*, dedicato alla scelta del criterio di regolarizzazione attraverso il quale stimare i parametri associati ai dati CGM (Figura 5.6).

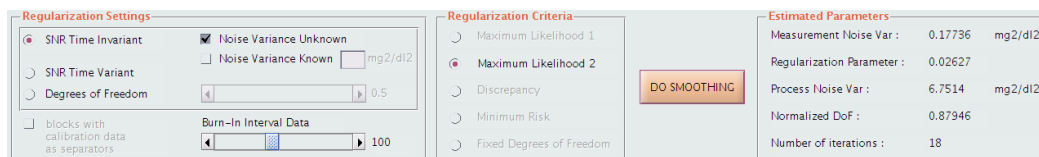


**Figura 5.6:** *Pannello Regularization Criteria.*

Tali criteri sono associati a componenti radio mutuamente esclusivi e vengono abilitati qualora le impostazioni settate nel pannello precedente siano effettivamente compatibili con la loro applicazione.

### 5.4.2 Layout: Smoothing Button and *Estimated Parameters* (pannello 3)

Completata la fase di impostazione dello smoothing e di gestione a priori del profilo regolarizzato, la disposizione grafica dei componenti orienta l'utente verso il pulsante di attuazione dello smoothing. Premendo il push button *DO SMOOTHING*, nel pannello *Estimated Parameters*, posizionato di seguito sulla destra, vengono visualizzati i risultati della stima parametrica svolta. I valori stimati sono associati a componenti di testo statico che rendono il pannello informativo, schematico e intuitivo. Ulteriori dettagli saranno forniti nei paragrafi successivi. La Figura 5.7 consente di avere un colpo d'occhio completo su tutta la sezione superiore del modulo di smoothing presentata fino ad ora, comprendente pannelli di impostazione, pulsante per avviare il calcolo e pannello dei parametri stimati.



**Figura 5.7:** *Panoramica completa dei pannelli per le impostazioni dello smoothing, del pulsante di avvio dell'algoritmo e del pannello di visualizzazione delle stime.*



### 5.4.3 Layout: *Display Data and Results* (pannello 4)

Il pannello *Display Data and Results* è quello che occupa lo spazio maggiore all'interno del modulo di smoothing. Quanto si ottiene dal procedimento algoritmico di regolarizzazione di un segnale deve poter essere apprezzato nell'immediato da un punto di vista grafico e qualitativo ed è utile che l'immagine sia allo stesso tempo completata e accompagnata dai valori puntuali ottenuti. Queste esigenze hanno portato alla costruzione del pannello così come viene presentato in Figura 5.9.

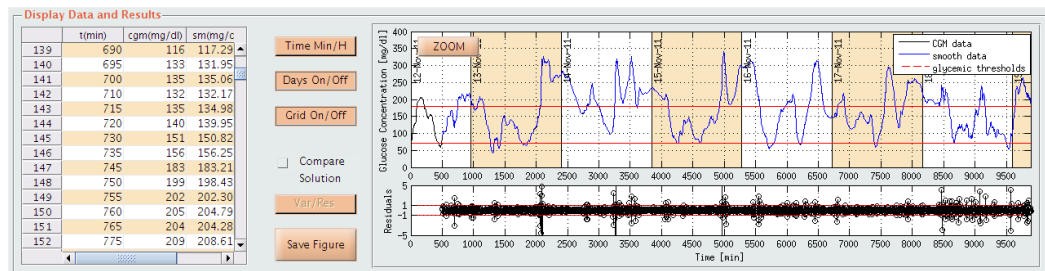


Figura 5.8: Pannello *Display Data and Results*.

La tabella visualizza i dati CGM originari e i dati CGM regolarizzati. In maniera del tutto simile all'interfaccia di inserimento e caricamento dei dati, le righe seguono la progressione della numerosità dei campioni, mentre le colonne contengono, nell'ordine, l'incremento temporale in minuti, i valori originari in mg/dl, i valori filtrati in mg/dl e i residui pesati. Se si è ipotizzata la presenza di un rapporto segnale-rumore in cui la varianza del rumore in ingresso varia nel tempo, in tabella compaiono anche le colonne relative ai valori stimati della varianza, espressi in  $\text{mg}^2/\text{dl}^2$ , ed ai gradi di libertà normalizzati. Accanto alla tabella sono stati posizionati i pulsanti *Time Min/H*, *Days On/Off*, *Grid On/Off* e *Save Figure*, identici a quelli precedentemente presentati, con il duplice scopo di personalizzare e variare la visualizzazione grafica della soluzione e di riconoscere un'impronta organizzativa nel layout di base che sia comune a tutte le sub-interfacce realizzate. In aggiunta a tali pulsanti, sono stati inseriti due ulteriori componenti funzionali:

- *Compare Function*: check box che, se contrassegnata da segno di spunta, apre una seconda soluzione di regolarizzazione, a scelta tra quelle precedentemente salvate, e la confronta con i risultati attuali, sovrappo-  
nendone l'andamento nel grafico;
- *Var/Res*: toggle button che, nel caso di rapporto segnale-rumore tempo  
variante, abilita la possibilità di visualizzare, nel plot inferiore, l'anda-  
mento della varianza del rumore di misura nel tempo o quello dei residui  
pesati.

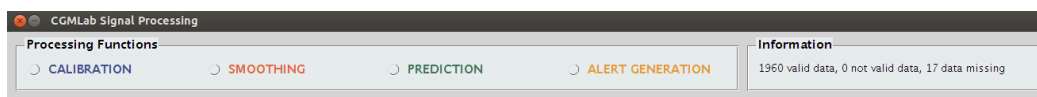
Come si vede dalla Figura 5.9, il riquadro per il plot del segnale contiene due grafici: uno per visualizzare l'andamento temporale dei dati CGM originari

e del segnale corrispondente regolarizzato e l'altro per poter identificare la distribuzione nel tempo dei residui pesati, o alternativamente della varianza dell'errore di misura, là dove possibile. Come nelle altre finestre, anche nel modulo di smoothing il pannello contenente i due grafici è dotato di pulsante per lo zoom dell'immagine. Ulteriori dettagli implementativi saranno scandagliati nei paragrafi seguenti.

## 5.5 Dettagli implementativi del modulo per lo smoothing

Ricalcando in parte la procedura descrittiva del capitolo precedente, di seguito verranno approfonditi gli aspetti più rilevanti riguardanti la codifica di quanto presentato per ora da un punto di vista puramente grafico.

La pressione del push button *Signal Processing*, appartenente alla finestra di selezione e inserimento dei dati, determina l'apertura della seconda sub-interfaccia di CGMLab, la quale appare inizialmente con i soli pulsanti radio relativi ai moduli di processamento disabilitati e con le stesse informazioni testuali essenziali sui dati CGM visualizzate durante la fase di caricamento degli stessi (Figura 5.9).



**Figura 5.9:** Sezione superiore della finestra *CGMLab Signal Processing*, nella quale si può scegliere il modulo di elaborazione da lanciare in esecuzione.

La funzione GUI2 contiene tutto ciò che riguarda la definizione e il comportamento funzionale dei componenti di ciascuno dei moduli di elaborazione realizzati. Per quanto riguarda il modulo in questione, anche in questo caso due sono le variabili globali dedicate alla gestione e alla memorizzazione di variabili numeriche o testuali, `sm`, e dei singoli componenti grafici, `g2`. Di seguito vengono presentate le variabili numeriche, scalari, vettoriali o matriciali, e le variabili testuali così come esse sono inizializzate all'avvio della finestra di smoothing e così come vengono reimpostate nel momento in cui si resettano i calcoli e si rilancia il procedimento di smoothing, tramite il pulsante *Reset*.

Nome	Tipo	Contenuto	Inizializzazione
<code>sm.data_table</code>	matriciale	Dati da visualizzare in tabella	[ ]
<code>sm.sigma2</code>	scalare o vettoriale	Varianza dell'errore di misura presente nei dati CGM originari	[ ]

<code>sm.lambda2</code>	scalare o vettoriale	Varianza del rumore di processo	[ ]
<code>sm.gamma</code>	scalare o vettoriale	Parametro di regola- rizzazione	[ ]
<code>sm.dof_norm</code>	scalare o vettoriale	Gradi di libertà nor- malizzati	[ ]
<code>sm.iter</code>	scalare o vettoriale	Numero di tentativi per arrivare a conver- genza della stima	[ ]
<code>sm.smooth_values.cgm</code>	vettoriale	Valori CGM regolariz- zati	[ ]
<code>sm.smooth_values.time</code>	vettoriale	Valori temporali cor- rispondenti ai valori CGM regolarizzati	[ ]
<code>sm.smooth_values.res</code>	vettoriale	Residui pesati	[ ]
<code>sm.n_burnin</code>	scalare	Numero di dati utiliz- zati per stimare i pa- rametri incogniti della serie temporale	100
<code>sm.slider_dof_norm</code>	scalare	Gradi di libertà nor- malizzati, qualora si vogliono fissare e non stimare	0.5
<code>sm.id_set</code>	scalare		0
<code>sm.compare_values.cg</code>	vettoriale	Valori CGM regola- rizzati della soluzio- ne che eventualmen- te si visualizza per confronto	[ ]
<code>sm.compare_values.time</code>	vettoriale	Valori temporali cor- rispondenti ai valo- ri CGM regolarizza- ti della soluzione di confronto	[ ]
<code>sm.compare_values.res</code>	vettoriale	Residui pesati della soluzione di confronto	[ ]
<code>sm.plotsm</code>	variabile identifica- tiva	Identificativo, per la visualizzazione della legenda, del plot del segnale regolarizzato	[ ]
<code>sm.plotcgm</code>	scalare	Identificativo, per la visualizzazione della legenda, del plot del segnale originario	[ ]

<code>sm.plotth</code>	scalare	Identificativo, per la visualizzazione della legenda, del plot delle soglie di ipoglicemia (70 mg/dl) e di iperglicemia (180 mg/dl)	[ ]
<code>sm.plotcompare</code>	scalare	Identificativo, per la visualizzazione della legenda, del plot del segnale regolarizzato di confronto	[ ]
<code>sm.sigma2plot</code>	vettoriale	Stime puntuali della varianza dell'errore di misura, nel caso di SNR tempo variante, rese congruenti e corrispondenti con i valori temporali del segnale originario affinché ne sia agevolata l'eventuale visualizzazione grafica	[ ]

Relativamente ai pulsanti sul fondo della figura, si è già accennato al ruolo del pulsante *Reset*. Cliccando, invece, sul pulsante *Save Data (as .mat)* è possibile salvare il segnale di riferimento originario e i risultati del processo di smoothing, grazie alla callback che richiama internamente la funzione

```
function save_structure_GUI2_Smoothing( init ,sm,complete_name)
% function that saves original and smoothed data
% useful for the check box 'Compare Function'

    cgm_time = init.cgm_values.time;
    cgm_data = init.cgm_values.cgm;
    cgm_time_smooth = sm.smooth_values.time;
    cgm_data_smooth = sm.smooth_values.cgm;
    residuals = sm.smooth_values.res;

    Ts = init.Ts;

    % sigma2, gamma, lambda2, dof_norm
    % can be scalars (ordinary KF) or vectors (sliding KF)

    sigma2 = sm.sigma2;
    gamma = sm.gamma;
```

```

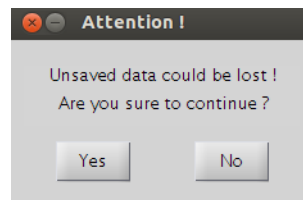
lambda2 = sm.lambda2;
dof_norm = sm.dof_norm;

save(complete_name, 'cgm_time', 'cgm_data', 'cgm_time_smooth',
      'cgm_data_smooth', 'Ts', ...
      'residuals', 'sigma2', 'gamma', 'lambda2', 'dof_norm')

end

```

In aggiunta, il pulsante *Change Function*, posto nell'estremo inferiore della finestra, consente di passare ad un altro modulo di processamento, dopo aver ricordato all'utente con una finestra di dialogo (Figura 5.10) della possibilità di salvare i dati prima che le variabili a cui questi sono stati assegnati vengano reinizializzate con il lancio del nuovo modulo di elaborazione.

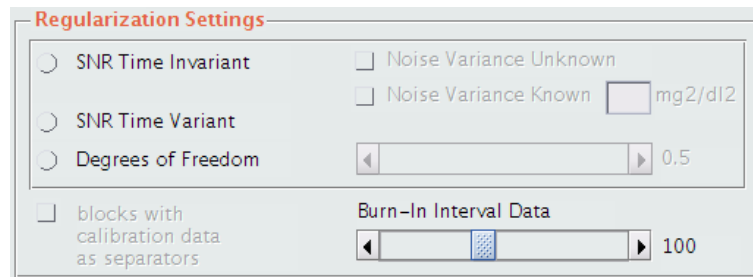


**Figura 5.10:** Finestra di dialogo che ricorda all'utente della possibilità di salvare la sessione di lavoro prima di passare ad un altro modulo di processamento.

Di seguito vengono presentati i dettagli implementativi più significativi sulla codifica dei singoli pannelli della sub-interfaccia di smoothing. Ripercorrendo il percorso di presentazione del Capitolo 3 relativamente alla prima finestra grafica, anche in questo contesto per ogni pannello verrà fornita una descrizione sull'inizializzazione e sul comportamento dei componenti di controllo e di visualizzazione, mentre successivamente verranno commentati alcuni dettagli di programmazione e le scelte progettuali ad essi collegate.

### 5.5.1 Dati e callbacks: *Regularization Settings* e *Regularization Criteria* (pannello 1 e 2)

Il pannello *Regularisation Settings* è visibile, al lancio del modulo di smoothing, così come appare in Figura 5.11. I radio button preposti alla gestione delle ipotesi sul segnale CGM appaiono inizialmente disabilitati, così come i corrispondenti check box o componenti di controllo loro associati. In particolare, la slider relativa all'imposizione dei gradi di libertà normalizzati è disabilitata e inizializzata al valore intermedio 0.5. Anche il check box relativo alla possibilità di effettuare lo smoothing tra una calibrazione e l'altra è disabilitato e in una versione futura di CGMLab verrà eventualmente abilitato qualora siano stati inseriti precedentemente i dati di calibrazione. A prescindere dalle ipotesi sul SNR o sui gradi di libertà del segnale, la barra scorrevole riguardante i dati di burn-in è direttamente accessibile all'utente ed inizializzata a 100 dati.



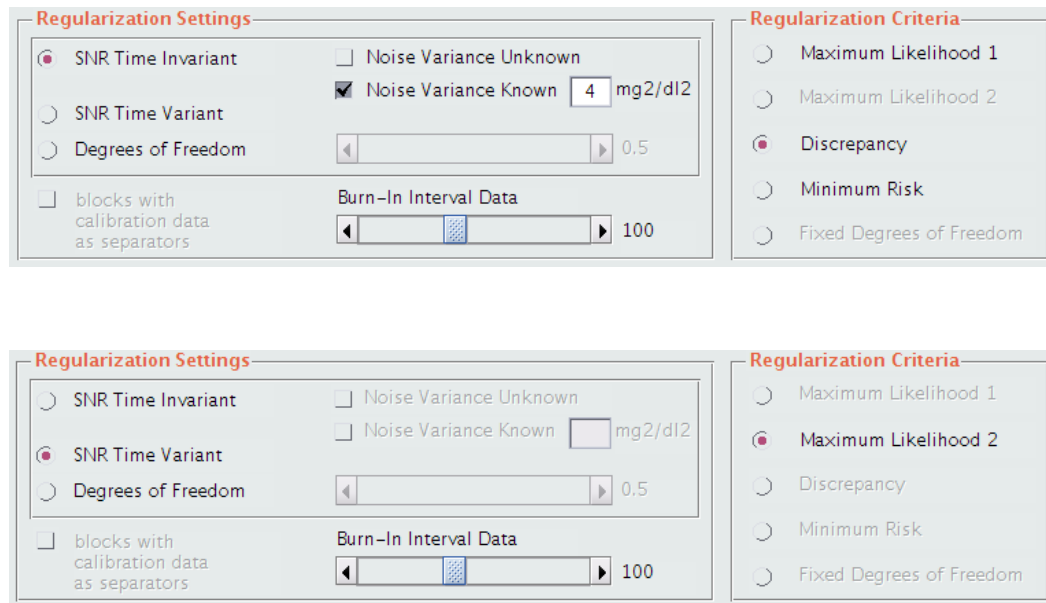
**Figura 5.11:** Pannello *Regularization Settings*, così come esso si presenta all'avvio del modulo di smoothing.

Per quanto riguarda il pannello di selezione dei criteri di stima, *Regularization Criteria*, i radio button presenti sono tutti inizialmente schermati e la loro abilitazione dipende dalle impostazioni settate nel pannello precedente.

Considerando gli aspetti fondamentali riguardanti la codifica di questa coppia di pannelli collegati, uno solo tra i pulsanti radio relativi al SNR e ai gradi di libertà può essere selezionato e ciò comporta l'abilitazione di uno o più criteri di regolarizzazione, tra cui selezionarne uno. In particolare:

- selezionando *SNR Time Invariant*: viene data la possibilità di abilitare uno dei check box adiacenti, a seconda che la varianza dell'errore di misura sia sconosciuta o sia nota e costante. Nel primo caso l'unico criterio di stima applicabile, tra quelli presentati in §4.1.3, è il criterio Maximum Likelihood 2. In alternativa, è possibile scegliere se selezionare Maximum Likelihood 1, Discrepancy o Minimum Risk;
- selezionando *SNR Time Variant*: il valore della varianza dell'errore deve essere stimato ad ogni nuovo campione acquisito, perciò l'unico criterio applicabile è il Maximum Likelihood 2;
- selezionando *Degrees of Freedom*: l'utente può spostare e fissare la barra corrispondente, con l'aiuto del mouse o delle frecce agli estremi dei componenti, ad un valore compreso tra 0.1 e 0.9. Si è scelto di restringere il range 0-1 a questi valori per evitare problemi di calcolo durante la stima. Conseguentemente a questa scelta, l'unico criterio selezionabile è quello identificato da Fixed Degrees of Freedom.

È importante sottolineare il fatto che, qualora primi due casi ci siano problemi di convergenza nella stima parametrica iniziale, il software indirizza l'utente all'impostazione obbligatoria dei gradi di libertà normalizzati mediante una finestra di dialogo e l'interruzione momentanea del processo di calcolo avviato. La Figura 5.12 mostra due esempi di selezione, tra quelli possibili descritti.



**Figura 5.12:** *Sopra: imposizione di un valore costante della varianza dell'errore di misura, con conseguente possibilità di scelta tra diversi criteri di stima. Sotto: imposizione tempo-variante del SNR, con conseguente scelta obbligata sul criterio Maximum Likelihood 2.*

Relativamente alla parte algoritmica i criteri di regolarizzazione presentati, sono invocati, rispettivamente, nel codice mediante la sintassi

```
[sm.gamma,sm.lambda2,sm.dof_norm,sm.iter,sm.x0] = ML1_svd(
    time_burnin,...
    CGM_burnin,sm.sigma2,init.Ts,2,1e-4,1e+4);

[sm.gamma,sm.sigma2,sm.lambda2,sm.dof_norm,sm.iter,sm.x0] =
    ML2_svd(time_burnin,...
    CGM_burnin,init.Ts,2,1e-4,1e+4);

[sm.gamma,sm.lambda2,sm.dof_norm,sm.iter,sm.x0] = discrepancy_svd(
    time_burnin,...
    CGM_burnin,sm.sigma2,init.Ts,2,1e-4,1e+4);

[sm.gamma,sm.lambda2,sm.dof_norm,sm.iter,sm.x0] = minrisk_svd(
    time_burnin,...
    CGM_burnin,sm.sigma2,init.Ts,2,1e-4,1e+4);

[sm.gamma,sm.sigma2,sm.lambda2,sm.iter,sm.x0] = qgamma_norm_fixed(
    sm.degrees_norm,...
    time_burnin,CGM_burnin,init.Ts,2,1e-4,1e+4);
```

Le cinque function, codificate in M-file separati, differiscono tra loro solo per il termine di riferimento in relazione al quale viene identificato il parametro di regolarizzazione. Tutti i file di codice applicano il criterio di stima alla

porzione di burn-in del segnale CGM, relativamente al quale a ciascun dato è sottratta la quantità corrispondente al valore iniziale, affinché sia possibile implementare un algoritmo causale su un segnale non causale. Per ciascun gamma-tentativo, viene stimata una funzione approssimante attraverso risoluzione ai minimi quadrati mediante l'utilizzo di una griglia virtuale uniforme di passo pari al periodo di campionamento. Per ridurre la complessità computazionale dei calcoli la stima viene svolta mediante la SVD, di cui si è parlato in §4.2.2 e, una volta ottenuto il valore ottimale del parametro di regolarizzazione, al segnale filtrato viene aggiunto nuovamente il valore iniziale e gli ultimi di questi dati costituiscono il valore iniziale dello stato del sistema, a partire dai quali viene stimato il segnale smooth con filtro di Kalman. Quanto detto è esplicito dal codice seguente, comune ai cinque M-file presentati.

```
function [...] = name_function(...)

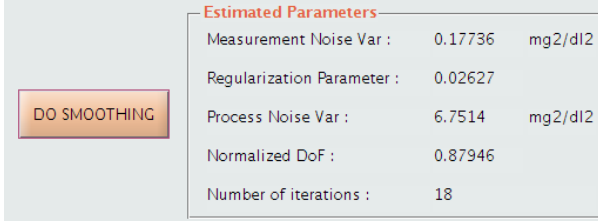
y = yburnin;
yb = y(1);
ym = y - yb;
...
% data missing, virtual grid
tv = (t(1):Tc:t(end));
...
% F
...
F = toeplitz(p,r);
% virtually missing samples
...
Gv = eye(nv);
G = Gv(vt == 1,:);
% singular value decomposition
H = G*inv(F);
[U S V] = svd(H);
D = U'*H*V;
eta = U'*ym;
ni_est = zeros(nv,1);
% gamma
itergamma = 0;
convergenza = 0;
while convergenza == 0
    ... % regularization criteria
end % while
...
uhat = uhat + yb;
lambda2 = sigma2/gamma;
x0 = [uhat(end-1) uhat(end)]';
```



### 5.5.2 Dati e callbacks: Smoothing Button and *Estimated Parameters* (pannello 3)

Il push button *DO SMOOTHING* è sempre abilitato e, osservando la disposizione grafica che assume all'interno della finestra, si può dire che esso riassume in sé il processo di calcolo per cui il modulo di smoothing è stato realizzato. Di seguito si trova il pannello *Estimated Parameters* che, a seconda del caso in esame, consente di visualizzare, a scopo statistico e informativo, alcuni risultati della stima parametrica effettuata da burn-in. Al lancio del modulo, o nel momento in cui questo viene resettato, il pannello appare completamente vuoto, sia perché nessun parametro è stato ancora stimato, sia per indicare all'utente una ben precisa consequenzialità da seguire nella manipolazione dell'interfaccia.

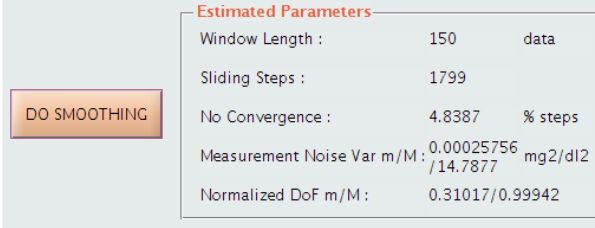
Per quanto riguarda la codifica, questa sezione rappresenta la parte più corposa e centrale del modulo. Cliccando sul pulsante radio preposto all'azione di smoothing, questa viene esplicitata in maniere differente a seconda delle impostazioni fissate nei due pannelli presentati precedentemente e anche le informazioni che compaiono nel pannello *Estimated Parameters* assumono forme diverse. Nell'ipotesi di SNR tempo-invariante, con  $\sigma^2$  incognita o nota, o scelta dei gradi di libertà normalizzati da imporre, la stima parametrica viene effettuata una sola volta da intervallo di burn-in, perciò nel pannello *Estimated Parameters* compaiono le informazioni relative ai valori, fissati o stimati a seconda del caso, di  $\sigma^2, \gamma, \lambda^2, q(\gamma)/n$  e del numero di tentativi per arrivare a convergenza della stima (Figura 5.13). Il filtro di Kalman viene applicato in maniera ordinaria al resto dei dati.



Estimated Parameters		
Measurement Noise Var :	0.17736	mg2/dl2
Regularization Parameter :	0.02627	
Process Noise Var :	6.7514	mg2/dl2
Normalized DoF :	0.87946	
Number of iterations :	18	

**Figura 5.13:** Esempio di visualizzazione dei risultati di stima parametrica nel caso di SNR tempo-invariante.

Se, in alternativa, si ipotizza che il rapporto segnale-rumore sia variabile nel tempo, il valore dei parametri viene stimato ad ogni nuovo campione acquisito, perciò nel pannello *Estimated Parameters* compaiono le informazioni relative alla lunghezza della finestra scorrevole, al numero di passi di scorrimento, alla percentuale di passi in cui non si raggiunge la convergenza della stima e ai valori minimo e massimo rispettivamente di  $\sigma^2$  e  $q(\gamma)/n$ , parametri il cui significato è più intuitivo e immaginabile di altri (Figura 5.14).



Estimated Parameters		
Window Length :	150	data
Sliding Steps :	1799	
No Convergence :	4.8387	% steps
Measurement Noise Var m/M :	0.00025756 /14.7877	mg2/dl2
Normalized DoF m/M :	0.31017/0.99942	

**Figura 5.14:** Esempio di visualizzazione dei risultati di stima parametrica nel caso di SNR tempo-variante.

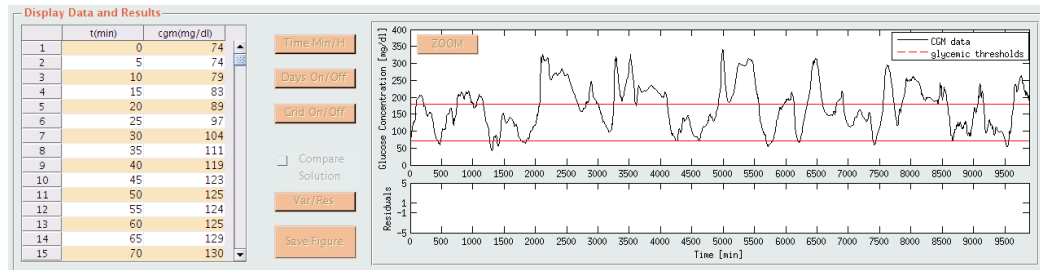
Il filtro di Kalman viene applicato ai dati rimanenti di ciascuna finestra, una volta esclusi quelli di burn-in da cui sono stati stimati i parametri e si serve di tali campioni per ottenere una stima attendibile del solo ultimo campione. Si è scelto, una volta fissato il numero di dati di burn-in, di impostare la lunghezza della finestra ad un numero di campioni pari a `sm.n_burnin + 50`. Considerando la grande mole di dati a disposizione, si è scelto di procedere nel seguente modo nel caso in cui non si arrivasse a convergenza della stima. Nella situazione in cui la stima iniziale, sia il SNR costante o variabile nel tempo, dovesse non convergere entro il 200-esimo tentativo, si è già accennato alla comparsa di una finestra di dialogo che obbliga l'utente a fissare il valore dei gradi di libertà e procedere allo smoothing in quel modo. Se i parametri invece venissero inizialmente stimati senza problemi, ma durante uno o più tentativi di stima intermedi, durante lo scorrere della finestra, la stima non dovesse andare a buon fine, si è scelto di mantenere come valore dei parametri attuali l'ultimo valore valido di ciascuno. Questa scelta è dettata dal fatto che, avendo riscontrato con le serie di prova una bassissima percentuale di step a mancata convergenza (1-5%), si vuole dare la possibilità di portare a termine lo smoothing in tempo reale di un segnale CGM, senza che il calcolo si interrompa per problemi riguardanti una minima percentuale delle finestre.

Come verrà illustrato in dettaglio nel paragrafo successivo, l'avvio dello smoothing implica anche l'aggiornamento del pannello *Display Data and Results*, sia per quanto riguarda la tabella in esso presente, sia per quanto riguarda la componente grafica.

### 5.5.3 Dati e callbacks: *Display Data and Results* (pannello 4)

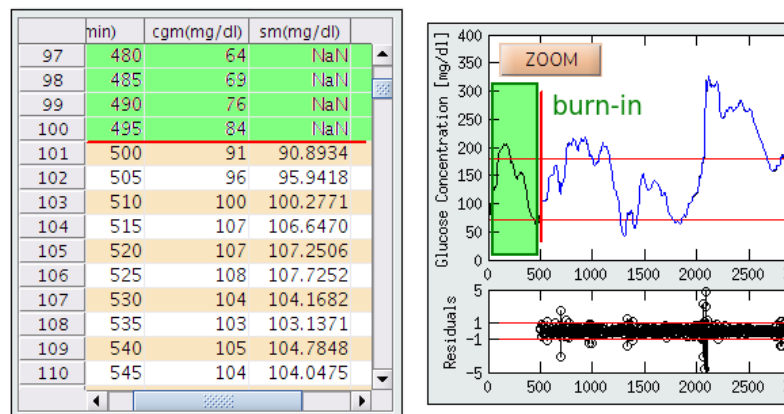
Il pannello *Display Data and Results* è il più articolato e il più vasto, essendo realizzato allo scopo di visualizzare, secondo differenti modalità, i risultati ottenuti. Essendoci già, ancora prima di effettuare il filtraggio, dei dati di riferimento a disposizione, in questa finestra il pannello di visualizzazione dei dati e dei risultati viene mostrato fin dall'inizio. All'avvio del modulo di smoothing esso appare come in Figura 5.15: la tabella contiene i dati CGM caricati nella finestra precedente, i pulsanti presenti sono tutti disabilitati, il compo-

nente grafico superiore raffigura l'andamento della serie di dati CGM, mentre il pannello grafico inferiore si presenta vuoto.



**Figura 5.15:** Pannello *Display Data and Results* del modulo di smoothing, così come si presenta al suo avvio.

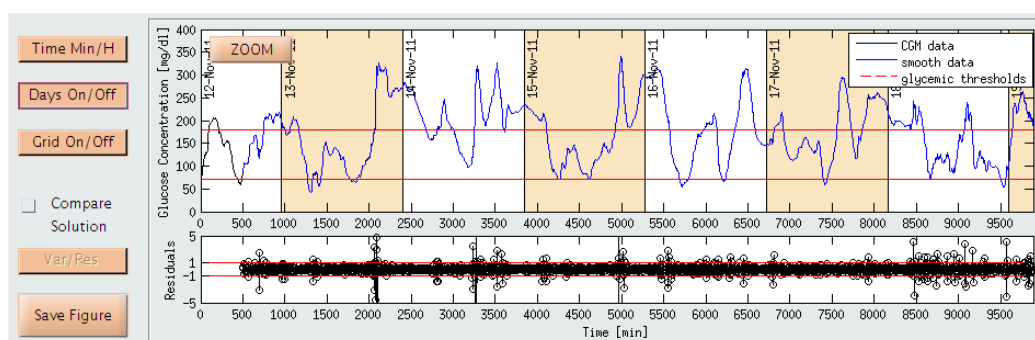
L'azione sul push button *DO SMOOTHING* comporta l'aggiornamento e il completamento del contenuto del pannello *Display Data and Results*. La tabella `g2.Table` di visualizzazione dati viene aggiornata con i valori stimati del profilo regolarizzato, con i valori dei residui pesati corrispondenti e, se il rapporto segnale-rumore è ipotizzato essere tempo-variante, con i valori stimati di  $\sigma^2$  e dei gradi di libertà normalizzati. La codifica di aggiornamento della tabella, di cui non si riportano i dettagli, è del tutto simile a quella presentata in §3.6.3. In Figura 5.16 si può notare che in corrispondenza dei dati di burn-in, non compaiono valori di regolarizzazione e di residui, aspetto che viene identificato da caselle contenenti il riferimento NaN.



**Figura 5.16:** Schema che illustra la corrispondenza tra intervallo di burn-in e dati registrati con NaN in tabella.

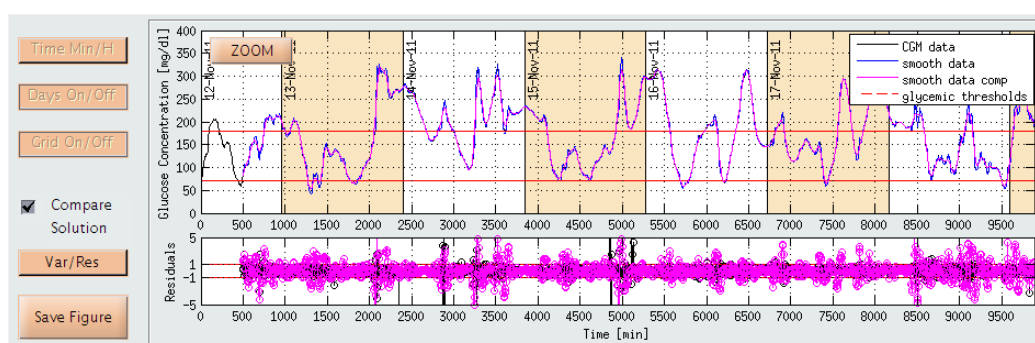
Per quanto riguarda il plot del segnale, identicamente alla prima finestra presentata, l'asse delle ascisse viene settato, a prescindere dal periodo di campionamento, a seconda della durata dell'acquisizione in esame, come viene descritto in (riferimento al paragrafo del capitolo 3 in cui si parla del set asse

x). Anche per quanto riguarda i pulsanti *Time Min/H*, *Days On/Off*, *Grid On/Off*, *Save Figure* e *ZOOM*, le dinamiche e le condizioni sono esattamente le medesime di quelle presentate in (riferimento al paragrafo del capitolo 3 in cui si parla del set asse x). Per quanto concerne il pulsante *Days On/Off*, lo sfondo a colori alterni indicante i giorni di acquisizione viene proiettato solo sul pannello del segnale CGM e non su quello dei residui, come si vede dalla Figura 5.17, aspetto che altrimenti ne peggiorerebbe la qualità grafica. Osservando il grafico del segnale è importante notare, inoltre, che dati filtrati e residui vengono visualizzati da un certo istante temporale in poi, coerentemente con l'approccio di burn-in di cui si è ampiamente parlato.



**Figura 5.17:** *Pannello Display Data and Results: lo sfondo alternato dei giorni, per mantenere una certa qualità grafica complessiva, viene visualizzato solo sul segnale e non sul pannello dei residui.*

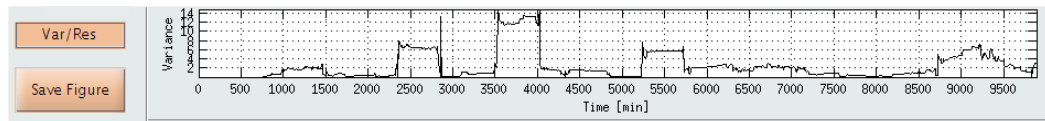
In aggiunta a quanto mostrato fino ad ora, nel pannello *Smoothing* sono stati inseriti un check box per il confronto della soluzione corrente con un altro segnale regolarizzato precedentemente salvato e un toggle button, *Var/Res*, introdotti in precedenza.



**Figura 5.18:** *Pannello Display Data and Results: esempio di comparazione tra soluzione corrente (in blu) e una soluzione precedentemente salvata (in fucsia), entrambe provenienti dallo stesso segnale originario.*

Relativamente al primo componente, se *Compare Function* viene selezionato, una finestra di dialogo consente all'utente di caricare un segnale salvato da

una precedente sessione di smoothing. Il plot di questo segnale e dei residui corrispondenti, in sovrapposizione a quello corrente, avviene solo se c'è corrispondenza tra dati CGM originari del profilo attuale e dati CGM originari del segnale di confronto. Ogni dato necessario al plot è salvato nel MAT-file che viene caricato. Quanto detto è illustrato in Figura 5.18. Se si sceglie di operare un confronto, il background grafico corrente (asse dei tempi, sfondo alternato dei giorni, griglia) viene mantenuto tale finché il confronto non viene disabilitato. Per quanto riguarda, invece, il toggle button *Var/Res*, esso si abilita solo se il SNR è supposto essere tempo-variante e, se premuto, visualizza alternativamente l'andamento della varianza del rumore di misura stimata o l'andamento dei residui pesati. In Figura 5.19 è illustrato un esempio di quanto detto. Anche in questo caso il valore della varianza viene plottato solo da un determinato istante in poi, dipendente dal numero di dati scelti per la stima parametrica.



**Figura 5.19:** Secondo componente grafico del pannello *Display Data and Results*: esempio di visualizzazione temporale, nel caso di SNR tempo-invariante, della varianza dell'errore di misura stimata.



# Capitolo 6

## Modulo di CGMLab per la predizione

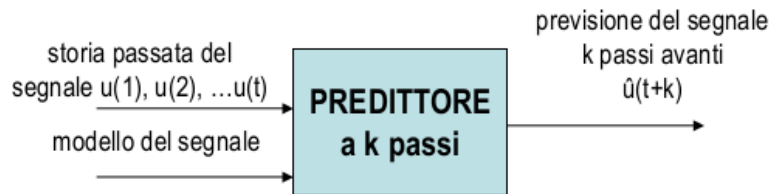
### 6.1 Predizione della glicemia

I sensori CGM contengono al loro interno un dispositivo in grado di generare un allarme acustico e visivo nel momento in cui le soglie di ipoglicemia e iperglicemia vengono raggiunte. Dal punto di vista terapeutico, ciò consente al paziente di agire prontamente, per esempio con l'assunzione di zuccheri nel caso di ipoglicemia, senza però evitare che la concentrazione di glucosio rimanga per un certo periodo all'interno delle zone di rischio. Per limitare anche l'insorgere di complicanze sarebbe utile riuscire a generare gli allarmi di ipoglicemia ed iperglicemia in anticipo rispetto alla loro occorrenza. Questo è possibile se la generazione dell'allarme si basa non sull'andamento corrente della glicemia, ma su quello che si prevede, in tempo reale, essere l'andamento in un futuro prossimo, generalmente 30 o 45 minuti dopo. Alcuni dispositivi CGM sono già dotati di semplici algoritmi di predizione, che rilevano il trend degli ultimi campioni e generano un'allerta sulla base del valore successivo predetto. Il software che viene presentato in questa tesi ha invece l'obiettivo di implementare algoritmi di predizione basati sull'impiego e l'identificazione di modelli matematici di dati, che siano in grado di fornire una stima più raffinata della glicemia futura.

Come illustrato schematicamente in Figura 6.1, per calcolare la predizione di un segnale che varia nel tempo sono necessari:

1. la storia passata del segnale
2. un modello matematico

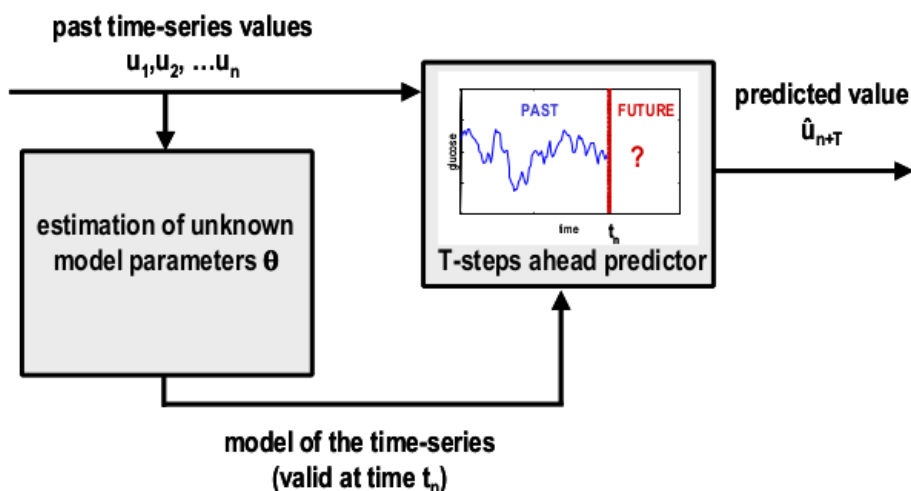
In questo caso, la storia passata della serie glicemica viene fornita da un sensore CGM. Ciò che rimane soggetta a scelta è la determinazione di un opportuno modello matematico che sia in grado di descrivere i dati a disposizione e sia utilizzabile al tempo corrente per predire il valore del segnale al un certo istante futuro.



**Figura 6.1:** Schema a blocchi di un predittore discreto. La predizione avviene  $k$  passi in avanti, dove  $k$  è dato dal rapporto tra orizzonte di predizione espresso e periodo di campionamento della serie.

## 6.2 Algoritmi di predizione di letteratura

Negli ultimi quindici anni diversi algoritmi di predizione sono stati proposti nella letteratura. Il software CGMLab vuole dare, in riferimento alla molteplicità di metodi esistenti, la possibilità di confrontare profili predittivi relativi ad uno stesso segnale ottenuti con algoritmi predittivi differenti, cioè con tipi di modello o parametri diversi. Per realizzare un software il più possibile esaustivo sono stati scelti e implementati modelli polinomiali (ordini 1, 2 e 3), modelli AR (ordini 1, 2 e 3) e predittori di Kalman. Ciò che accomuna gli algoritmi selezionati e inseriti nel modulo predittivo di CGMLab è, come viene presentato in Figura 6.2, il loro funzionamento tempo-variante. A prescindere, cioè, dal tipo di elaborazione adottata, ad ogni nuovo istante di tempo in cui perviene un nuovo dato i parametri del modello vengono riaggiustati, in maniera tale che quest'ultimo sia in grado di adattarsi dinamicamente, in tempo reale, ai cambiamenti del segnale glicemico.



**Figura 6.2:** Schema a blocchi di un predittore discreto adattativo.

La struttura specifica del modello, assieme ai parametri di volta in volta stimati, descrivono quindi la serie temporale in esame localmente. Tale descrizione, valida solo per un intervallo di campionamento e poi di volta in volta



riaggiustata, permette di lavorare con un modello locale che, rispetto ad uno globale e tempo-invariante, presenta complessità parametrica ridotta e, conseguentemente, maggiore facilità e rapidità nell'identificazione, aspetti che ne permettono un utilizzo in online.

Di seguito vengono presentati alcuni algoritmi predittivi, così come sono stati implementati nella GUI. Sono stati presi in considerazione i modelli AR e polinomiali presentati in letteratura, in particolare in [33], ampliandone l'applicazione anche a ordini superiori, ed è stato implementato anche un algoritmo di predizione a partire dalla teoria sul KF presentata in [26] e descritta nel capitolo precedente.

### 6.2.1 Modelli Auto-Regressivi (AR)

In un modello autoregressivo di un generico ordine  $p$  il campione di  $y$  all'istante  $t$  è descritto dall'equazione ingresso-uscita

$$y(k) = a_1 y(k-1) + a_2 y(k-2) + \dots + a_p y(k-p) + v(k) \quad k = p+1, \dots, n \quad (6.1)$$

Ciò equivale ad affermare che il campione  $k$ -esimo è descritto come una combinazione lineare dei dati agli istanti precedenti e del valore  $k$ -esimo di un rumore bianco,  $v(k)$ . Nel modello AR di ordine 1 il dato al generico istante dipende esclusivamente dal dato all'istante precedente e dal rumore bianco in ingresso. Per adattarsi progressivamente ai cambiamenti del segnale, ad ogni istante di campionamento si dovrà prima identificare il modello dai dati passati e successivamente calcolare la predizione del segnale ad un certo numero di passi in avanti, dove tale numero è dato dal rapporto tra PH e il periodo di campionamento, utilizzando l'equazione alle differenze corrispondente al modello (assumendo nullo il rumore bianco che alimenta il modello). Il modello non viene, quindi, identificato una volta per tutte all'inizio del processo di previsione, ma viene identificato ogni volta che viene acquisita una nuova misura.

Per l'identificazione del modello AR al generico tempo  $t_i$  viene utilizzata la tecnica dei Recursive Least Squares (RLS) che permette, rispetto ad una semplice identificazione ai minimi quadrati, di ridurre il peso dell'algoritmo su processore e RAM, agevolandone l'applicabilità online. Infatti, la stima dei parametri del modello ad un certo istante viene ottenuta sfruttando ricorsivamente la stima all'istante precedente, così da abbattere il costo computazionale relativo all'inversione di una matrice che, altrimenti, aumenterebbe di dimensione ad ogni nuovo istante di campionamento. Indichiamo con  $\psi_k$  il vettore (colonna) contenente i  $p$  campioni che precedono il campione corrente e con  $a$  il vettore (colonna) dei  $p$  coefficienti del modello

$$\psi_k = [y(k-1) \quad y(k-2) \quad \dots \quad y(k-p)]^T \quad k = p+1, \dots, n \quad (6.2)$$

$$a = [a_1 \quad a_2 \quad \dots \quad a_p] \quad (6.3)$$

Consideriamo di avere a disposizione  $n$  campioni. Per ogni  $k$ , da  $p+1$  a  $n$ , lo scalare  $y(k)$  si può esprimere in funzione del prodotto interno tra i due vettori

$$y(k) = \psi_k^T a + v(k) \quad (6.4)$$

Mettendo i campioni  $y(k)$ , con  $k = p+1, p+2, \dots, n$ , in un vettore colonna  $y$ , si può passare ad un'espressione matriciale ottenendo, rispettivamente in forma esplicita e in forma compatta,

$$\begin{bmatrix} y(p+1) \\ y(p+2) \\ \vdots \\ y(n) \end{bmatrix} = \begin{bmatrix} y(p) & y(p-1) & \cdots & y(1) \\ y(p+1) & \ddots & & y(2) \\ \vdots & & \ddots & \vdots \\ y(n-1) & y(n-2) & \cdots & y(n-p) \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_p \end{bmatrix} + \begin{bmatrix} v(p+1) \\ v(p+2) \\ \vdots \\ v(n) \end{bmatrix} \quad (6.5)$$

$$y = Ga + v \quad (6.6)$$

in cui  $G$  ha dimensione  $(n-p) \times p$ , con  $n \gg p$ . L'identificazione del modello attraverso i minimi quadrati porta a risolvere l'equazione

$$\hat{a} = (G^T G)^{-1} G^T y = Q^{-1} b \quad (6.7)$$

Se non si adottasse la procedura di identificazione ricorsiva, all'arrivo di ogni nuovo campione si riapplicherebbe lo stesso procedimento. Visto che l'ordine  $p$  del modello nel nostro caso può essere 1, 2 o 3 l'inversa della matrice  $Q = (G^T G)$  sarebbe non eccessivamente onerosa da calcolare ( $O(p^3)$ ). Tuttavia è computazionalmente dispendioso creare tale matrice, in quanto ad ogni nuovo campione  $n$  aumenta e ottenere  $Q$  diventa sempre più pesante ( $O(p^{2n})$ ).

Con l'identificazione RLS si parte dai suddetti  $n$  campioni acquisiti, per cui si definisce

$$y_n = [y(p+1), y(p+2), \dots, y(n)]^T \quad (6.8)$$

$$y_n = G_n a_n + v_n \quad (6.9)$$

$$\hat{a}_n = (G_n^T G_n)^{-1} G_n^T y_n = Q_n^{-1} b \quad (6.10)$$

Nel momento in cui viene acquisito il campione  $n+1$ -esimo, detto  $y_{n+1} = [y(p+1), y(p+2), \dots, y(n), y(n+1)]^T$  ci si trova a risolvere l'equazione corrispondente  $\hat{a}_{n+1} = (G_{n+1}^T G_{n+1})^{-1} G_{n+1}^T y_{n+1} = Q_{n+1}^{-1} b$  che, con una serie di accortezze e sostituzioni, diventa

$$\hat{a}_{n+1} = \hat{a}_n + [y(n+1) - \psi_{n+1}^T \hat{a}_n] Q_{n+1}^{-1} \psi_{n+1} \quad (6.11)$$

Definendo lo scalare  $e(n+1) = y(n+1) - \psi_{n+1}^T \hat{a}_n$ , errore di stima nella predizione a un passo, e  $k_{n+1} = Q_{n+1}^{-1} \psi_{n+1}$ , vettore di guadagno, l'espressione (6.11) diventa

$$\hat{a}_{n+1} = \hat{a}_n + e(n+1) k_{n+1} \quad (6.12)$$

Al crescere di  $n$ , ciò porta ad avere un costo computazionale  $O(p^3)$  invece che  $O(p^{2n})$ . Applicando il lemma di inversione di matrice si mette tutto in funzione

di  $P_n = Q_n^{-1}$ , cosa che permette di non calcolare poi inverse di matrici. Includendo nella stima ai minimi quadrati una matrice di pesi i cui pesi sono, come verrà illustrato in seguito, del tipo  $\mu_k$  (peso del campione glicemico relativo a  $k$  istanti prima del tempo corrente di campionamento), si ottiene, riepilogando quanto visto, il sistema di equazioni

$$P_{n+1} = \frac{1}{\mu} \left[ P_n - \frac{P_n \psi_{n+1} \psi_{n+1}^T P_n}{\mu + \psi_{n+1}^T P_n \psi_{n+1}} \right] \quad (6.13)$$

$$k_{n+1} = \frac{P_n \psi_{n+1}}{\mu + \psi_{n+1}^T P_n \psi_{n+1}} \quad (6.14)$$

$$e(n+1) = y(n+1) - \psi_{n+1}^T \hat{a}_n \quad (6.15)$$

$$\hat{a}_{n+1} = \hat{a}_n + e(n+1) k_{n+1} \quad (6.16)$$

dove  $\psi_{n+1} = [y(n) \quad y(n-1) \quad \dots \quad y(n-p+1)]^T$ .

Dopo aver identificato il modello, per effettuare predizione un certo numero di istanti in avanti, ad esempio  $k$  passi, è sufficiente utilizzare l'equazione (6.1) e alimentare il modello con rumore bianco nullo in ingresso. Nel semplice caso di modello AR(1), data la stima del coefficiente  $a_n$  all'istante attuale,  $\hat{a}_n$ , la predizione  $k$  passi in avanti è data semplicemente da

$$y_{n+k} = a_n^k y_n \quad (6.17)$$

### 6.2.2 Modelli polinomiali

Un modello polinomiale di grado generico  $m$  è descritto dall'equazione

$$u(t) = a_1 t^m + a_2 t^{m-1} + \dots + a_m t + a_{m+1} \quad (6.18)$$

Ad ogni nuovo istante di misura  $t_i$ :

- con la tecnica dei minimi quadrati pesati, fittando i dati relativi ai tempi passati ( $t_i, t_{i-1}, t_{i-2}, \dots$ ), viene identificata una nuova  $n$ -pla di parametri ( $a_1, a_2, \dots, a_{m+1}$ );
- il modello così determinato, valido al tempo  $t_i$ , è impiegato per predire la glicemia al tempo  $t_i + PH$ . Il Prediction Horizon (PH) è un parametro fissabile e, sommato all'istante corrente, rappresenta il tempo futuro in minuti in corrispondenza del quale voglio conoscere il valore glicemico

Come verrà approfondito in seguito, un ruolo chiave nella predizione del segnale glicemico è attribuito al valore di PH e a come vengono pesati i dati storici nel processo di identificazione del modello. Nel caso di modello lineare ( $m = 1$ ) la curva parametrica si riduce ad una semplice retta di regressione

$$u(t) = a_1 t + a_2 \quad (6.19)$$

Nel caso quadratico ( $m = 2$ ) e cubico ( $m = 3$ ) l'equazione del modello si riduce invece, rispettivamente, alle espressioni

$$u(t) = a_1 t^2 + a_2 t + a_3 \quad (6.20)$$

$$u(t) = a_1 t^3 + a_2 t^2 + a_3 t + a_4 \quad (6.21)$$

Per identificare il modello è possibile procedere con la soluzione ai minimi quadrati, vista con l'espressione in forma esplicita (6.10) oppure applicare anche in questo caso il metodo di stima ricorsiva. Come si può facilmente verificare, per identificare ricorsivamente il modello a partire dalla stima all'istante precedente si possono utilizzare le stesse identiche equazioni utilizzate per i modelli autoregressivi, con l'unica variante del vettore  $\psi$ , che non conterrà più i  $p$  campioni precedenti a quello attuale, ma conterrà istanti temporali elevati alla  $m, m - 1, \dots, 0$ . In particolare si ha

$$\psi_k = [t_k^p \quad t_k^{p-1} \quad \dots \quad t_k \quad 1]^T \quad k = 1, \dots, n \quad (6.22)$$

$$a = [a_1 \quad a_2 \quad \dots \quad a_{p+1}] \quad (6.23)$$

$$y(k) = \psi_k^T a \quad (6.24)$$

$$\begin{bmatrix} y(1) \\ y(2) \\ \vdots \\ y(n) \end{bmatrix} = \begin{bmatrix} t_1^p & t_1^{p-1} & \dots & t_1 & 1 \\ t_2^p & \ddots & & t_2 & 1 \\ \vdots & & \ddots & \vdots & \\ t_n^p & t_n^{p-1} & \dots & t_n & 1 \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_{p+1} \end{bmatrix} \quad (6.25)$$

$$y = Ga \quad (6.26)$$

$$\hat{a} = (G^T G)^{-1} G^T y = Q^{-1} b \quad (6.27)$$

Dopo aver identificato il modello, per effettuare predizione un certo numero di istanti in avanti, per esempio  $k$ , è sufficiente calcolare il valore della curva di regressione in quell'istante. Nel caso di modello lineare la predizione al tempo  $t_{n+k}$  sarà data da

$$u(t_{n+k}) = a_n(1)t_{n+k} + a_n(2) \quad (6.28)$$

Dagli studi effettuati e abbondantemente presenti in letteratura è possibile affermare che i modelli polinomiali, rispetto ai modelli AR, a parità di percentuale di falsi allarmi e allarmi mancati, portano ad un guadagno temporale inferiore, che si traduce in un ritardo maggiore, di circa il 30%. Dagli studi effettuati e abbondantemente presenti in letteratura è possibile affermare che i modelli polinomiali, rispetto ai modelli AR, a parità di percentuale di falsi allarmi e allarmi mancati, portano ad un guadagno temporale inferiore, che si traduce in un ritardo maggiore, di circa il 30% [31].

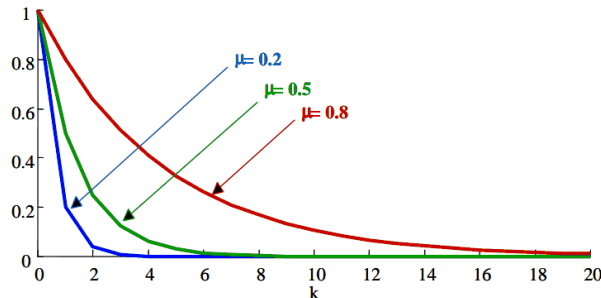
### 6.2.3 Forgetting Factor ( $\mu$ ) e Prediction Horizon (PH)

Negli algoritmi presentati si è visto come assumono fondamentale importanza i valori del Prediction Horizon (PH) e il valore del Forgetting Factor ( $\mu$ ), altrimenti detto coefficiente di oblio. La scelta dei due parametri risulta determinante per la conseguente predizione. Il coefficiente di oblio rappresenta, in un certo senso, la memoria del sistema, mentre l'orizzonte di predizione indica quanti minuti in avanti si vuole predire.

Come accennato nei paragrafi precedenti, nell'identificazione ricorsiva di un modello AR o polinomiale è possibile pesare in maniera non uniforme i dati passati. Si ritiene, infatti, ragionevole pensare che nella stima dei coefficienti del modello all'istante attuale i campioni più remoti nel tempo debbano essere considerati in misura minore rispetto a quelli più recenti. Di conseguenza, appare sensato attribuire un peso a ciascun campione inversamente proporzionale alla distanza che esso ha dal campione corrente, in particolare

$$\text{peso} = \mu^k \quad (6.29)$$

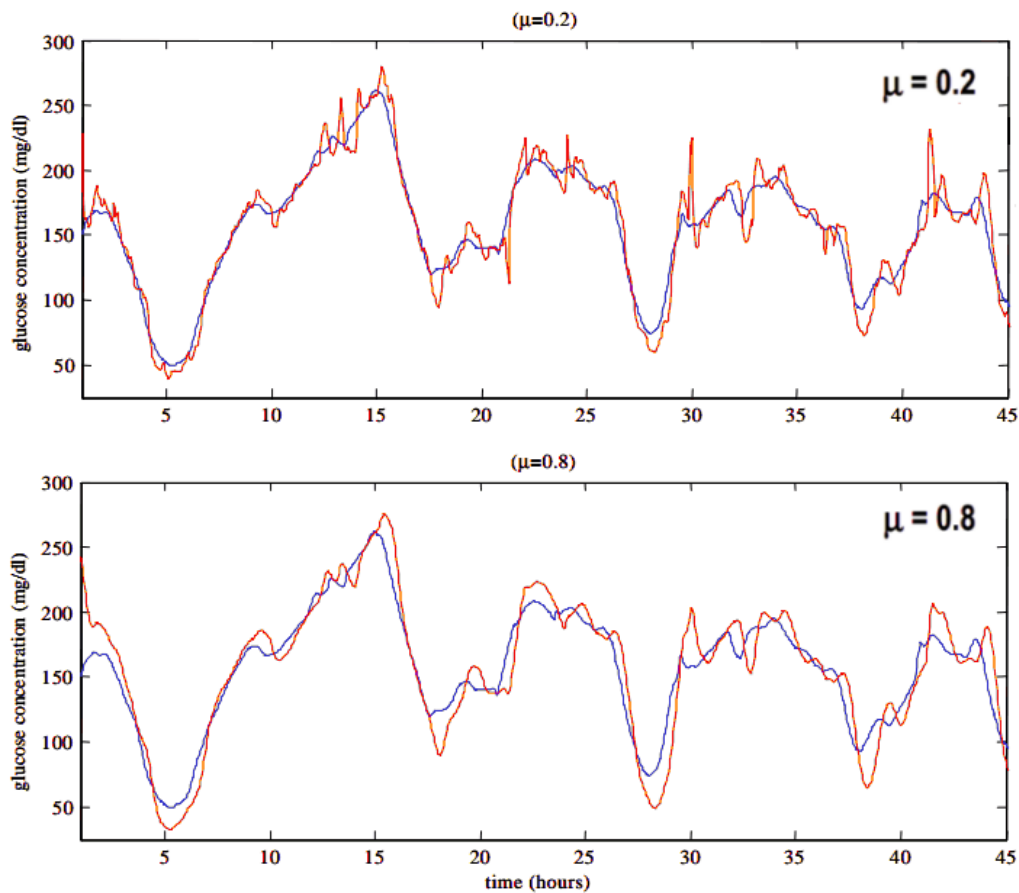
dove  $k$  è proprio la distanza (in termini di passi o istanti di campionamento) del campione in esame dalla misura corrente. Come illustra la Figura 6.3 il peso dei parametri diminuisce più o meno rapidamente man mano che ci si allontana dall'istante presente.



**Figura 6.3:** *Peso esponenziale decrescente in funzione della distanza  $k$  dal campione attuale.*

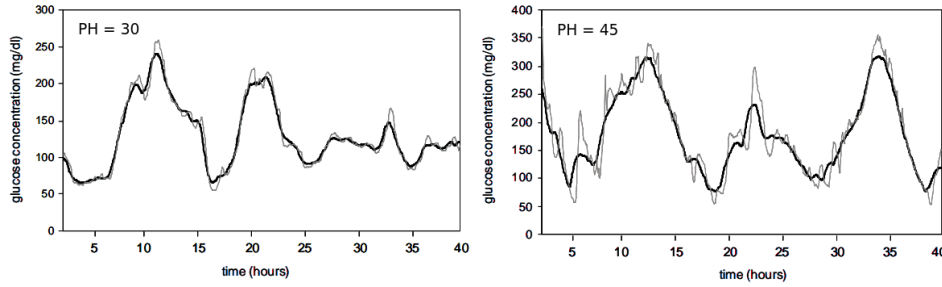
Il parametro  $\mu$  assume valori compresi tra 0 e 1 e rappresenta la rapidità con cui ci si dimentica dei dati passati. Più alto è  $\mu$ , più è lunga la memoria e quindi più lentamente ci si dimentica dei dati remoti, mentre più  $\mu$  tende a 0, più in fretta ci si dimentica precedenti. Nel primo caso si rileva la difficoltà a cogliere le variazioni del segnale e ciò si traduce in un ritardo temporale più marcato. Al contrario, nel secondo caso, la maggior reattività del modello comporta però una elevata sensibilità al rumore che si manifesta in una marcata irregolarità del profilo predetto. In Figura 6.4 è mostrato un esempio di ciò che si ottiene fissando, verso l'uno e l'altro estremo il valore di  $\mu$ , a parità di tipo di modello e di PH scelti. Nell'immagine superiore, a causa di un valore troppo basso del coefficiente, il segnale predetto appare eccessivamente

rumoroso, mentre nella secondo grafico appare un ritardo del profilo di predizione, dovuto a un valore di  $\mu$  elevato. In entrambi i casi la predizione non è del tutto utilizzabile clinicamente, in questo secondo perché essa non consente di avere un guadagno temporale sufficiente a prevenire un evento iperglicemico o ipoglicemico, mentre nel primo caso perché un profilo irregolare, generando in particolar modo molti falsi positivi, non può essere preso come riferimento per stabilire con affidabilità il superamento delle soglie di normalità.



**Figura 6.4:** Esempi di predizione con differenti valori di  $\mu$  fissati.

Il secondo parametro di cui si è accennato è il cosiddetto Prediction Horizon (PH) o orizzonte di predizione, ovvero quanto in avanti nel tempo, in termini di minuti, si vuole predire la concentrazione di glucosio. Generalmente [31] [33] viene spesso indicato per il PH un valore di 30 minuti o 45 minuti, più raramente di 60. C'è da dire che più elevato è il PH, cioè più in avanti ci si spinge nella ricerca del valore della glicemia, più la predizione appare instabile. D'altra parte, un PH troppo basso renderebbe temporalmente inutilizzabile la predizione. La Figura 6.5 confronta, in relazione a quanto detto, due profili predetti con due valori di PH diversi.



**Figura 6.5:** A sinistra: predizione con modello AR,  $\mu=0.5$ ,  $PH=30$ , soggetto 1. A destra: predizione con modello AR,  $\mu=0.5$ ,  $PH=45$ , soggetto 2.

### 6.2.4 Predizione con filtro di Kalman

Il filtro di Kalman presentato nel Capitolo (riferimento al capitolo/paragrafo) può assumere il ruolo di predittore, oltre che di semplice filtro. L'impostazione di stima dello stato a partire dalle osservazioni a disposizione rimane esattamente la stessa. In particolare, si consideri la predizione ad un passo espressa dalla (5.2)

$$\hat{x}(t+1|t) = F\hat{x}(t|t) \quad (6.30)$$

Volendo, al tempo  $t$ , ottenere una predizione  $k$  passi in avanti, con  $k$  dato dal rapporto tra orizzonte di predizione e periodo di campionamento, si ottiene la progressione di calcolo data da

$$\hat{x}(t+1|t) = F\hat{x}(t|t) \quad (6.31)$$

$$\hat{x}(t+2|t) = F\hat{x}(t+1|t) \quad (6.32)$$

...

$$\hat{x}(t+k|t) = F\hat{x}(t+k-1|t) \quad (6.33)$$

che porta all'espressione del valore predetto

$$\hat{x}(t+k|t) = F^k\hat{x}(t|t) = F^{PH/T_s}\hat{x}(t|t) \quad (6.34)$$

## 6.3 Indici quantitativi per la valutazione di un predittore

Dopo aver scelto più modelli da impiegare per effettuare la predizione della glicemia, averne identificato il valore dei parametri e aver determinato il profilo glicemico predetto, occorre disporre di uno strumento oggettivo che sia in grado di analizzare le prestazioni di un predittore, ovvero di valutare la bontà della predizione in funzione dei parametri in esame.

L'indice quantitativo impiegato più comunemente riguarda l'errore di predizione, ovvero la differenza tra il valore di misura e il corrispondente valore

predetto. È possibile valutare l'errore di predizione tramite l'MSPE (Mean Square Prediction Error), ovvero la norma dell'errore di predizione medio, data da

$$MSPE = \frac{1}{N} \sum_{i=1}^N (y_{pi} - y_i)^2 \quad (6.35)$$

dove  $N$  indica il numero di campioni,  $y$  i dati CGM e  $y_p$  i dati predetti. Nel nostro caso si è presa in considerazione la radice quadrata di tale indice, cioè l'RMSE (Root Mean Square Error)

$$RMSE = \sqrt{MSPE} = \sqrt{\frac{1}{N} \sum_{i=1}^N (y_{pi} - y_i)^2} \quad (6.36)$$

L'RMSE fornisce un'indicazione sulla bontà di predizione poiché minore è il suo valore migliore dovrebbe risultare nel complesso la predizione. Può quindi anche essere impiegato a posteriori per determinare il set ottimo di parametri osservando in corrispondenza di quali valori di  $\mu$  e PH l'RMSE presenta il valore minimo. Tuttavia tale indice non guarda né alla regolarità del profilo predetto, né al guadagno di tempo che la predizione consente di ottenere, rendendo necessaria la definizione di ulteriori indici che tengano in considerazione l'effettiva utilizzabilità e utilità dei risultati.

Uno degli aspetti che si auspica sia presente in un profilo di predizione è la regolarità del segnale, affinché questo possa generare allarmi certi. Per misurare la regolarità di un segnale si utilizza l'energia delle derivate di secondo ordine, ESOD, ovvero l'energia dell'accelerazione del segnale, definita come

$$ESOD = \frac{1}{N} \sum_{i=1}^N (d_{2i})^2 \quad (6.37)$$

dove  $d_{2i}$  esprime la derivata seconda del segnale

$$d_{2i} = \frac{y_{pi} - 2y_{pi-1} + y_{pi-2}}{T_s^2} \quad (6.38)$$

con  $T_s$  periodo di campionamento. Un valore di energia elevato è sintomo di un'irregolarità marcata del segnale, che si manifesta in frequenti cambi di pendenza del segnale stesso. Viceversa, più l'ESOD è basso, più il segnale è regolare.

L'altro aspetto da considerare è il guadagno temporale della predizione di un evento sul suo avverarsi, ovvero quanto anticipo la predizione consente di ottenere sul segnale reale. Il guadagno viene semplicemente espresso come differenza tra l'orizzonte di predizione e il ritardo della predizione. Nel lavoro di Sparacino et al. [33] viene proposto un metodo per il calcolo del ritardo di predizione che consiste nel valutare il ritardo come media dei ritardi di attraversamento di soglie poste sui fronti di salita di discesa del segnale. Questo



approccio risulta tuttavia laborioso, perciò nel presente lavoro questo valore viene ricavato come shift temporale che minimizza l'errore di predizione. A livello pratico, una volta effettuata la predizione, il profilo predetto viene tralato all'indietro di diversi shift temporali, da 0 a PH, per ognuno dei quali viene calcolato l'RMSE. Lo shift per il quale tale indice è minimo costituisce il ritardo di predizione. Di conseguenza appare chiaro come un buon predittore debba essere associato a un ritardo il più piccolo possibile.

In un lavoro di qualche anno fa [34], svolto presso il Dipartimento di Padova, sono stati proposti dei nuovi indici atti a quantificare la bontà di un predittore. La necessità di ottenere una soluzione ottima mediante un compromesso tra regolarità e, contemporaneamente, buon guadagno, ha portato in particolare al tentativo di definire un indice che integrasse questi due aspetti appartenenti a domini e ordini di grandezza differenti. Essendo l'ordine di grandezza dell'ESOD superiore a quella del guadagno, per conferire a entrambi gli indici lo stesso peso, è stato presentato, come indice migliore tra diverse proposte, quello definito come il rapporto tra ESOD normalizzato all'energia dei dati CGM e guadagno temporale al quadrato, normalizzato al PH. Tale indice, che viene chiamato indice J, è espresso come

$$J = \frac{ESOD_{norm}}{guadagno_{norm}^2} \quad (6.39)$$

con

$$ESOD_{norm} = \frac{ESOD_{predizione}}{ESOD_{CGM}} \quad (6.40)$$

$$guadagno_{norm} = \frac{PH - ritardo}{PH} \quad (6.41)$$

In aggiunta, sempre in (tesi trifoglio), è stato dimostrato che J consente di indicare a posteriori il valore del  $\mu$  ottimo in relazione all'esigenza di avere il migliore compromesso tra regolarità dell'andamento della predizione e tempestività dello stesso.

## 6.4 Progettazione del modulo per la predizione

Quanto presentato fino ad ora è stato implementato nel modulo *Prediction*, a cui si accede dalla seconda finestra grafica, dopo aver selezionato il radio button corrispondente. Tale modulo, illustrato in Figura 4.2 è composto da due macro-pannelli che consentono, rispettivamente, di scegliere l'algoritmo di predizione da utilizzare e di visualizzare risultati grafici e indici quantitativi relativi al profilo predetto. Identicamente a quanto presentato nei capitoli precedenti, i pulsanti posizionati all'estremo inferiore permettono di resettare quanto calcolato, di salvare i risultati ottenuti o di cambiare modulo di elaborazione.

### 6.4.1 Layout: *Prediction Settings* (pannello 1)

Il pannello *Prediction Setting* è composto da un gruppo funzionale di componenti indirizzati all'impostazione dell'algoritmo o degli algoritmi da utilizzare per predire i valori futuri di glicemia. In particolare, come si evince dalla Figura 6.6, il modulo di predizione è stato pensato allo scopo di poter scegliere tra più algoritmi di predizione e di confrontare tra loro eventualmente in maniera diretta due soluzioni.

**Prediction Settings**

POL (Polynomial model)

PH (min)  PH (min)

FF (0-1)  FF (0-1)

order  order

plot values (\*)  plot values (\*)

AR (AutoRegressive model)

PH (min)  PH (min)

FF (0-1)  FF (0-1)

order  order

plot values (\*)  plot values (\*)

KF (Kalman Filter)

PH (min)  PH (min)

plot c.i. (\*\*)  plot c.i. (\*\*)

plot values (\*)  plot values (\*)

(\*) plot values overpassing thresholds (70-180 mg/dl)

(\*\*) plot confidence intervals

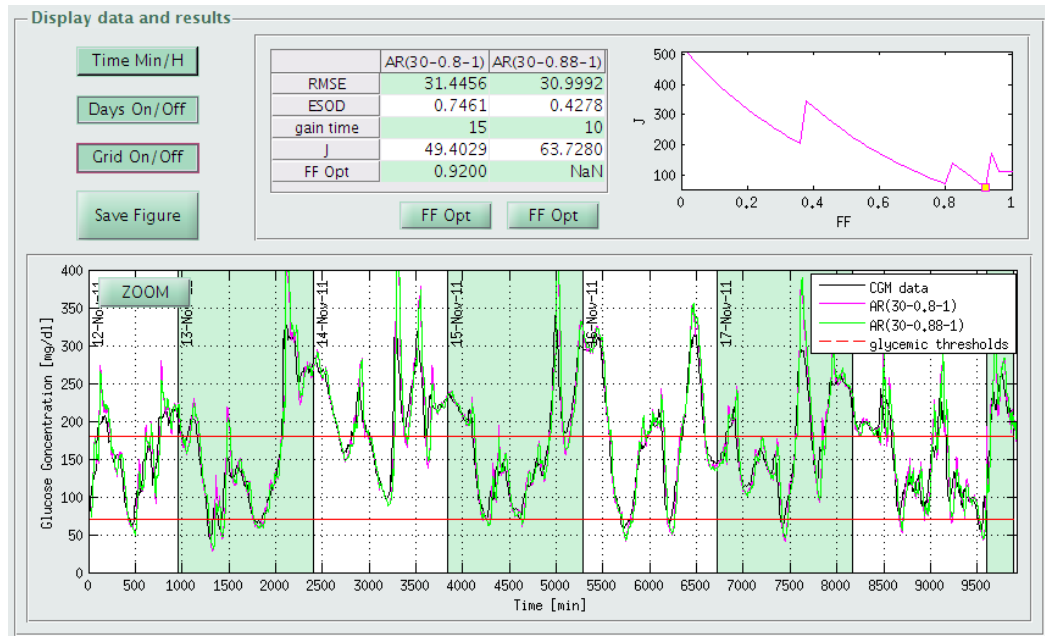
**Figura 6.6:** *Pannello Prediction Settings.* Nell'esempio sono selezionati due set predittivi basati su modello autoregressivo.

tre algoritmi predittivi si basano, come abbondantemente presentato nei paragrafi precedenti, su modelli polinomiali, modelli autoregressivi e filtro di Kalman e viene lasciata la possibilità all'utente di gestire in completa autonomia il set di parametri di ciascun algoritmo selezionato. La scelta di effettuare e visualizzare contemporaneamente al massimo due predizioni, non di più, è dettata dalla necessità di poter valutare i profili risultanti mantenendo una sufficiente qualità grafica del plot, senza eccessive sovrapposizioni di serie temporali diverse. Infine, all'estremo inferiore del pannello, il pulsante radio *DO PREDICTON* consente di avviare la predizione.

### 6.4.2 Layout: *Display Data and Results* (pannello 2)

Il pannello *Display Data and Results* (Figura 6.7) racchiude al suo interno il componente grafico che visualizza il segnale CGM originario e quello predetto,

i soliti pulsanti per gestirne la grafica o salvarla, una tabella contenente gli indici quantitativi calcolati per ciascun profilo implementato e un ulteriore componente grafico per plottare la funzione risultante dal calcolo del forgetting factor ottimo in funzione dell'indice J.



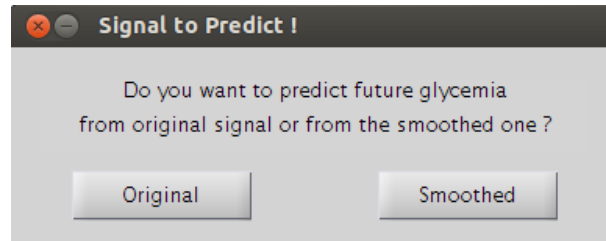
**Figura 6.7:** *Display Data and Results* relativo al modulo di processamento predittivo.

In particolare la tabella è suddivisa in due colonne, delle quali una o entrambe vengono riempite, a seconda del numero di profili di predizione selezionati. Le righe sono dedicate alla visualizzazione dei valori della radice dell'errore quadratico medio (RMSE), dell'energia delle derivate seconde (ESOD), del guadagno temporale (gain time), dell'indice J e dell'eventuale valore ottimo del coefficiente di oblio, calcolabile solo nel caso di modelli polinomiali o autoregressivi di ordine 1, come verrà specificato in seguito.

## 6.5 Dettagli implementativi del modulo per la predizione

Il modulo di predizione si abilita nel momento in cui viene selezionato il radio button corrispondente nell'interfaccia *CGMLab Signal Processing*. Da un punto di vista logico, la predizione può essere effettuata nell'immediato, sul segnale CGM originario, oppure può esserci l'interesse di predire la glicemia a partire dal segnale già regolarizzato. Per rendere accessibile questa doppia possibilità, nel momento in cui si seleziona il pulsante *PREDICTION*:

- se non è stato effettuato smoothing sul segnale, il modulo di predizione appare direttamente manipolabile dall'utente;
- se la scelta della predizione è conseguente ad una precedente sessione di smoothing, prima che si possa accedere al modulo di predizione compare una finestra di dialogo (Figura 6.8) che indica la possibilità di lavorare sul segnale originario o su quello filtrato.



**Figura 6.8:** Finestra di dialogo relativa alla scelta del segnale su cui effettuare predizione, qualora fosse già stato precedentemente effettuato denoising.

La funzione `GUI2` contiene al suo interno, oltre alla codifica di visualizzazione e di controllo del pannello di smoothing, anche tutto ciò che riguarda la definizione e il comportamento logico e funzionale dei componenti del modulo predittivo. Similmente ai capitoli precedenti, anche in questo caso sono state definite due strutture di dati per inizializzare e memorizzare variabili e oggetti. La variabile globale `g2` tiene in memoria qualunque componente grafico dell'interfaccia *CGMLab Signal Processing*, mentre alla variabile globale `pr` sono assegnati i valori di variabili numeriche o testuali relative al contenuto gestito dai componenti grafici. Nella tabella seguente vengono presentate tutte le variabili afferenti alla struttura globale `pr`, così come esse sono inizializzate e così come vengono reimpostate nel momento in cui si preme il pulsante *Reset*.

Nome	Tipo	Contenuto	Inizializzazione
<code>pr.n_profile</code>	scalare	Indicatore del primo o del secondo profilo di predizione	0
<code>pr.PH1</code>	scalare	Prediction Horizon del primo profilo di predizione	30
<code>pr.PH2</code>	scalare	Prediction Horizon del secondo profilo di predizione	30
<code>pr.mu1</code>	scalare	Forgetting Factor del primo profilo di predizione (per polinomiali e autoregressivi)	0.8

<code>pr.mu2</code>	scalare	Forgetting Factor del secondo profilo di predizione (per polinomiali e autoregressivi)	0.8
<code>pr.ord1</code>	scalare	Ordine del primo profilo di predizione (per polinomiali e autoregressivi)	1
<code>pr.ord2</code>	scalare	Ordine del secondo profilo di predizione (per polinomiali e autoregressivi)	1
<code>pr.valthre1</code>	scalare	Indicatore della selezione del radio button per la visualizzazione dei valori oltre soglia del primo profilo predetto	0
<code>pr.valthre2</code>	scalare	Indicatore della selezione del radio button per la visualizzazione dei valori oltre soglia del secondo profilo predetto	0
<code>pr.confint1</code>	vettoriale	Indicatore della selezione del radio button per la visualizzazione degli intervalli di confidenza del primo profilo predetto (KF)	0
<code>pr.confint2</code>	scalare	Indicatore della selezione del radio button per la visualizzazione degli intervalli di confidenza del secondo profilo predetto (KF)	0
<code>pr.mod1</code>	scalare	Valore che identifica il tipo di algoritmo scelto per il primo profilo di predizione (1: polinomiale, 2: autoregressivo, 3: predittore di Kalman)	0

<code>pr.mod2</code>	scalare	Valore che identifica il tipo di algoritmo scelto per il secondo profilo di predizione (1: polinomiale, 2: autoregressivo, 3: predittore di Kalman)	0
<code>pr.pred_values1.time</code>	vettoriale	Tempi corrispondenti ai dati CGM regolarizzati del primo profilo (minuti)	[ ]
<code>pr.pred_values1.cgm</code>	vettoriale	Dati CGM regolarizzati del primo profilo (mg/dl)	[ ]
<code>pr.pred_values2.time</code>	vettoriale	Tempi corrispondenti ai dati CGM regolarizzati del secondo profilo (minuti)	[ ]
<code>pr.pred_values2.cgm</code>	vettoriale	Dati CGM regolarizzati del secondo profilo (mg/dl)	[ ]
<code>pr.n_burnin</code>	scalare	Numero di dati utilizzati per stimare i parametri incogniti della serie temporale	100
<code>pr.RMSE1</code>	scalare	RMSE del primo profilo predetto	0
<code>pr.RMSE2</code>	scalare	RMSE del secondo profilo predetto	0
<code>pr.ESOD1</code>	scalare	ESOD del primo profilo predetto	0
<code>pr.ESOD2</code>	scalare	ESOD del secondo profilo predetto	0
<code>pr.gain1</code>	scalare	Guadagno temporale del primo profilo predetto	0
<code>pr.gain2</code>	scalare	Guadagno temporale del secondo profilo predetto	0
<code>pr.J1</code>	scalare	Indice J del primo profilo predetto	0
<code>pr.J2</code>	scalare	Indice J del secondo profilo predetto	0

<code>pr.mu_opt1</code>	scalare	Forgetting Factor eventualmente calcolato per il primo profilo predetto	0
<code>pr.mu_opt2</code>	scalare	Forgetting Factor eventualmente calcolato per il secondo profilo predetto	0
<code>pr.square_values1.time</code>	vettoriale	Tempi dei valori oltre soglia relativi al primo profilo predetto (minuti)	[ ]
<code>pr.square_values1.cgm</code>	vettoriale	Valori oltre soglia relativi al primo profilo predetto (mg/dl)	[ ]
<code>pr.square_values2.time</code>	vettoriale	Tempi dei valori oltre soglia relativi al secondo profilo predetto (minuti)	[ ]
<code>pr.square_values2.cgm</code>	vettoriale	Valori oltre soglia relativi al secondo profilo predetto (mg/dl)	[ ]
<code>pr.mu_opt1_plot.x</code>	vettoriale	Valori delle ascisse relative alla progressione del Forgetting Factor per il primo profilo predetto	[ ]
<code>pr.mu_opt1_plot.y</code>	vettoriale	Valori corrispondenti dell'indice J	[ ]
<code>pr.mu_opt2_plot.x</code>	vettoriale	Valori delle ascisse relative alla progressione del Forgetting Factor per il primo profilo predetto	[ ]
<code>pr.mu_opt2_plot.y</code>	vettoriale	Valori corrispondenti dell'indice J	[ ]
<code>pr.plotpr1</code>	variabile identificativa	Identificativo, per la visualizzazione della legenda, del plot del primo profilo predetto	[ ]

<code>pr.plotpr2</code>	variabile identifica- tiva	Identificativo, per la visualizzazione della legenda, del plot del secondo profilo predetto	[ ]
<code>pr.plotcgm</code>	variabile identifica- tiva	Identificativo, per la visualizzazione della legenda, del plot del segnale originario	[ ]
<code>pr.plotth</code>	variabile identifica- tiva	Identificativo, per la visualizzazione della legenda, del plot delle soglie di ipoglice- mia (70 mg/dl) e di iperglicemia (180 mg/dl)	[ ]
<code>pr.plotsm</code>	variabile identifica- tiva	Identificativo, per la visualizzazione della legenda, del plot del segnale regolarizzato	[ ]

---

In aggiunta alle variabili illustrate, sono state definite nella funzione `GUI2` altre variabili globali temporanee, indispensabili per non perdere informazioni e dati utili alla predizione relativa ad un segnale precedentemente regolarizzato. Tali variabili consentono, infatti, di tenere in memoria i risultati pregressi del filtraggio effettuato senza che essi vengano modificati o sovrascritti al nuovo lancio di `GUI2`, cioè nel momento in cui si sceglie di cambiare modulo di elaborazione mediante il pulsante *Change Function*. Tali variabili sono presentate nella tabella seguente.

Nome	Tipo	Contenuto	Inizializzazione
<code>tmp_time</code>	vettoriale	Tempi relativi al se- gnale CGM regolariz- zato (minuti)	-
<code>tmp_cgm</code>	vettoriale	Dati CGM regolariz- zati (mg/dl)	-
<code>tmp_id</code>	scalare	Identificatore di smoo- thing, vale 0 se non è stato ancora effettua- to, 1 altrimenti	-
<code>tmp_s2</code>	scalare	Varianza del rumore di misura	-
<code>tmp_l2</code>	scalare	Varianza di processo	-



<code>tmp_g</code>	scalare	Parametro di regola- rizzazione	-
<code>tmp_df</code>	scalare	Gradi di libertà nor- malizzati	-

---

Come si può notare, le stime dei parametri assumono un unico valore, anche nel caso in cui venga implementato l'approccio con sliding window in presenza di SNR variabile. Questa scelta è dovuta alla necessità di rendere il calcolo predittivo il più possibile efficiente, rapido e uniforme, perciò nella codifica dello smoothing assumono i valori

```
tmp_s2 = mean(sm.sigma2)
tmp_l2 = mean(sm.lambda2)
tmp_g = mean(sm.gamma)
tmp_df = mean(sm.dof_norm)
```

Guardando il modulo *Prediction*, si nota come una parte dei componenti presenti sia esattamente identica, dal punto di vista sia grafico che funzionale, alla finestra di caricamento dei dati e alla finestra di smoothing. In particolare modo, i pulsanti situati nell'estremo inferiore dell'interfaccia, permettono di resettare i calcoli, di salvare i risultati in un MAT-file unico, di salvare la tabella contenente gli indici di confronto o di cambiare modulo. Come per le precedenti finestre, anche in questo caso si riporta una parte del codice della callback attivata dal pulsante *Save Data (as .mat)*, affinché si abbia un riferimento riguardante ciò che viene salvato in automatico dal programma. La funzione di salvataggio tiene in considerazione i profili selezionati per la predizione, in maniera tale da salvare solo le variabili necessarie tra quelle a disposizione. Nella situazione, ad esempio, in cui il primo profilo selezionato corrisponde a un predittore di Kalman e in cui il secondo è dato da un modello polinomiale, il codice relativo è dato da

```
function save_structure_GUI2_Prediction (init ,sm,pr ,complete_name)
% function that saves original data , smoothed data if selected ,
% and predictive data and parameters

    cgm_time = init.cgm_values.time;
    cgm_data = init.cgm_values.cgm;
    if pr.tmp == 1
    cgm_time_smooth = sm.smooth_values.time;
    cgm_data_smooth = sm.smooth_values.cgm;
    end

    Ts = init.Ts;

    cgm_time_pred1 = pr.pred_values1.time;
    cgm_data_pred1 = pr.pred_values1.cgm;
```

```

cgm_time_pred2 = pr.pred_values2.time;
cgm_data_pred2 = pr.pred_values2.cgm;

PH1 = pr.PH1;
PH2 = pr.PH2;
mu1 = pr.mu1;
mu2 = pr.mu2;
ord1 = pr.ord1;
ord2 = pr.ord2;

RMSE1 = pr.RMSE1;
RMSE2 = pr.RMSE2;
ESOD1 = pr.ESOD1;
ESOD2 = pr.ESOD2;
gain1 = pr.gain1;
gain2 = pr.gain2;
J1 = pr.J1;
J2 = pr.J2;

if pr.mod1 == 3 & pr.mod2 ~= 3 % model 1: KF
    if pr.tmp == 1
        if isempty(pr.pred_values2.time) == 1 % 1 prediction
            from smoothed profile
                save(complete_name, 'cgm_time', 'cgm_data', '
cgm_time_smooth', 'cgm_data_smooth', 'Ts', ...
                    'cgm_time_pred1', 'cgm_data_pred1', ...
                    'PH1', ...
                    'RMSE1', 'ESOD1', 'gain1', 'J1')
            else % 2 predictions from smoothed profile
                save(complete_name, 'cgm_time', 'cgm_data', '
cgm_time_smooth', 'cgm_data_smooth', 'Ts', ...
                    'cgm_time_pred1', 'cgm_data_pred1', '
cgm_time_pred2', 'cgm_data_pred2', ...
                    'PH1', 'PH2', 'mu2', 'ord2', ...
                    'RMSE1', 'ESOD1', 'gain1', 'J1', 'RMSE2', 'ESOD2', '
gain2', 'J2')
            end
        else
            if isempty(pr.pred_values2.time) == 1 % 1 prediction
                from original profile
                    save(complete_name, 'cgm_time', 'cgm_data', 'Ts'
, ...
                        'cgm_time_pred1', 'cgm_data_pred1', ...
                        'PH1', ...
                        'RMSE1', 'ESOD1', 'gain1', 'J1')
            else % 2 predictions from original profile
                save(complete_name, 'cgm_time', 'cgm_data', 'Ts'
, ...
                    'cgm_time_pred1', 'cgm_data_pred1', '
cgm_time_pred2', 'cgm_data_pred2', ...
                    'PH1', 'PH2', 'mu2', 'ord2', ...
                    'RMSE1', 'ESOD1', 'gain1', 'J1', 'RMSE2', 'ESOD2', '
gain2', 'J2')
            end
        end
    end
end

```

```

        end
    end
else if pr.mod1 ~= 3 & pr.mod2 == 3 % model 2: KF
. . .
else if pr.mod1 == 3 & pr.mod2 == 3 % model 1 and 2: KF
. . .
    end
end
end
end
end

```

Di seguito vengono presentati alcuni dettagli implementativi sulla codifica dei due pannelli che compongono la sub-interfaccia di predizione. Ad un’iniziale presentazione dei componenti e del loro comportamento, seguiranno alcune spiegazioni più dettagliate sulle scelte implementative e la relativa codifica.

### 6.5.1 Dati e callbacks: *Prediction Settings* (pannello 1)

Il pannello *Prediction Settings* appare, all’avvio del modulo, come illustrato in Figura 6.9. I check box relativi ai tre algoritmi di predizione appaiono disabilitati, mentre ciascuna casella editabile contiene il valore di inizializzazione delle variabili corrispondenti.

**Prediction Settings**

POL (Polynomial model)

PH (min)  PH (min)

FF (0-1)  FF (0-1)

order  order

plot values (\*)  plot values (\*)

AR (AutoRegressive model)

PH (min)  PH (min)

FF (0-1)  FF (0-1)

order  order

plot values (\*)  plot values (\*)

KF (Kalman Filter)

PH (min)  PH (min)

plot c.i. (\*\*)  plot c.i. (\*\*)

plot values (\*)  plot values (\*)

(\*) plot values overpassing thresholds (70-180 mg/dl)  
 (\*\*) plot confidence intervals

**DO PREDICTION**

**Figura 6.9:** Pannello *Prediction Settings*, così come appare al lancio del modulo di predizione.

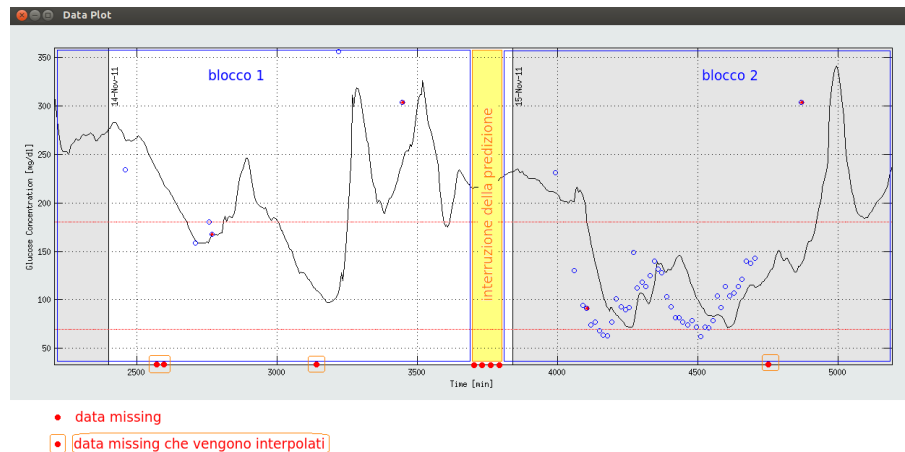
Il pannello permette all'utente di selezionare uno o due algoritmi o di gestire, mediante il medesimo algoritmo, uno o due profili di predizione con parametri differenti, per un massimo di due profili di predizione complessivamente. In pratica, selezionando uno dei check box appare un pop-up menu che consente di abilitare uno o due set di predizione relativi a quel determinato algoritmo. Qualora si tenti di abilitare un terzo set predittivo, una finestra di dialogo ricorda che non è possibile effettuare più di due predizioni contemporaneamente. Ciascuno dei sei set comprende anche un radio button *plot values* che, se selezionato, visualizza sul segnale predetto i dati oltre soglia mediante quadratini evidenziati. Inoltre è presente anche un controllo sull'inserimento dei parametri, che accetta solo valori di PH multipli del periodo di campionamento e valori del coefficiente di oblio compresi tra 0 e 1. Come si vede, i set dedicati a predizione mediante filtro di Kalman non contemplano la presenza del coefficiente di oblio, ma in aggiunta contengono un radio button che permette la visualizzazione congiunta di predizione e intervalli di confidenza corrispondenti. Viene lasciata la libertà all'utente di plottare o meno gli intervalli di confidenza, anche se è caldamente consigliato visualizzarli solo se si effettua un'unica predizione affinché i risultati siano graficamente apprezzabili e utilizzabili. Riguardo la codifica specifica di questo gruppo di componenti, si può dire che la parte più onerosa e lunga riguarda il controllo di selezione, deselezione, abilitazione, re-inizializzazione, dei componenti specifici e delle relative variabili assegnate. Questa parte occupa all'incirca 700 righe di codice.

Il pulsante *DO PREDICTION*, una volta premuto, comporta l'avvio della predizione e i conseguente aggiornamento del pannello adiacente. Relativamente alla codifica di predizione c'è da precisare quanto segue. Per quanto riguarda la predizione con filtro di Kalman, il codice riprende in parte alcuni passaggi implementati durante la codifica dello smoothing, con l'aggiunta del calcolo iterativo che consente di predire PH minuti in avanti. L'aspetto, invece, più rilevante riguardante le scelte di implementazione riguarda gli algoritmi polinomiale e autoregressivo. Questi, infatti, non prevedono nessun approccio stocastico per il calcolo ottimo dei parametri incogniti e si basano esclusivamente sulla formulazione e sull'identificazione di un modello di dati. Affinché si mantenga una continuità del segnale per l'applicazione dell'algoritmo, si è pensato di effettuare un pre-processing del segnale assegnando ai dati mancanti il valor medio del precedente e del successivo campione utile, ma interrompendo la predizione nel caso di buchi nell'acquisizione di più di 15 minuti. Il segnale CGM viene scannerizzato progressivamente, attraverso la funzione

```
[ virt_grid1 values_grid1 init_end1 ] = pred_blocks_15min ...
    (pr.topred_values.time, pr.topred_values.cgm, init.Ts);
```

la quale riceve in ingresso tempi, concentrazioni CGM e periodo di campionamento e restituisce in uscita una griglia virtuale con i corrispondenti valori

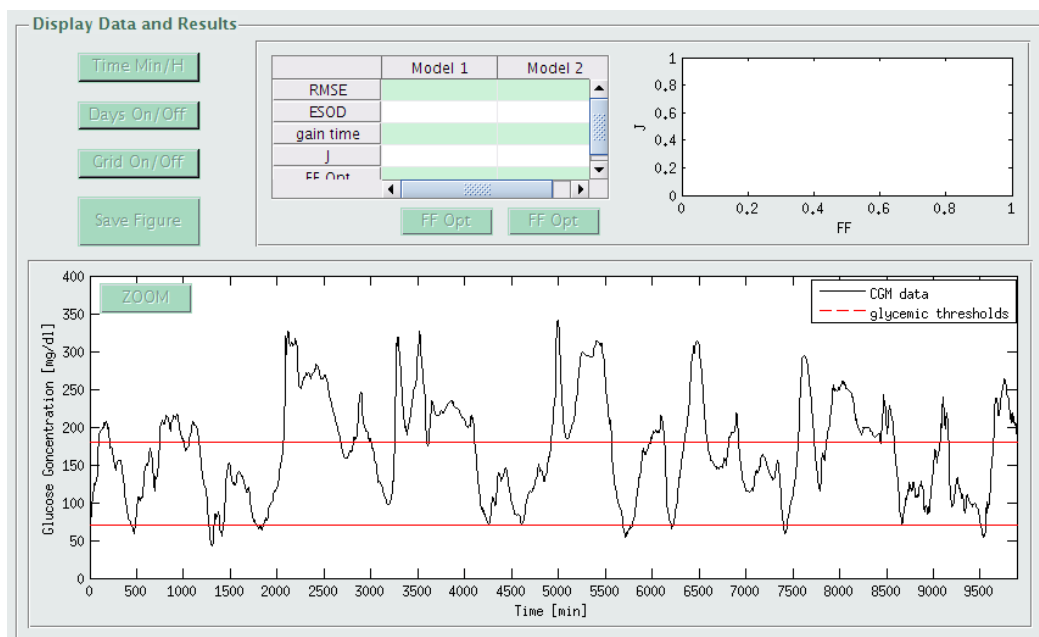
CGM e una matrice contenente gli istanti di inizio e di fine di ciascun blocco di predizione individuato. La scelta del limite di 15 minuti vale per ogni serie di dati caricata, poiché la variabilità del profilo glicemico è la medesima a prescindere dalla frequenza di campionamento adottata. La predizione viene applicata a ciascun blocco individuato e, attraverso un controllo automatico robusto degli indici, i valori vengono alla fine salvati nelle strutture dati `pr.pred_values1` e `pr.pred_values2`. Un esempio schematico del procedimento illustrato è mostrato in Figura 6.10.



**Figura 6.10:** Esempio schematico che illustra la suddivisione in blocchi predittivi del segnale CGM a seconda della posizione e del numero dei data missing.

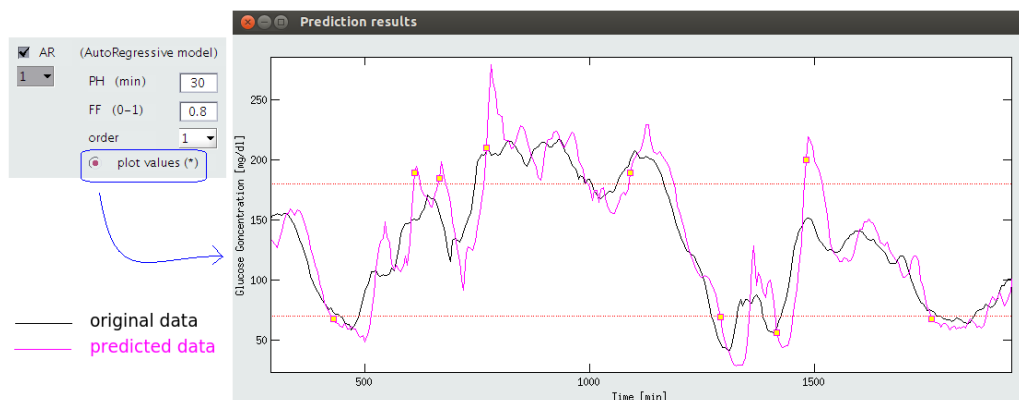
### 6.5.2 Dati e callbacks: *Display Data and Results* (pannello 2)

Brevemente viene presentato il pannello *Display Data and Results*, la cui codifica riprende alcuni passaggi implementativi delle interfacce precedentemente presentate. Esso si presenta inizialmente come mostrato in Figura 6.11. I pulsanti sono tutti disabilitati, la tabella e il pannello adiacente appaiono vuoti, mentre il grafico principale contiene, a seconda della scelta iniziale, il segnale originario oppure quest'ultimo sovrapposto a quello regolarizzato. Alla pressione del push button *DO PREDICTION* i pulsanti vengono abilitati e il pannello viene aggiornato. La codifica dei pulsanti *Time Min/H*, *Days On/Off*, *Grid On/Off*, *Save Figure* e *ZOOM* ricalca esattamente quella vista nei paragrafi §3.6.3 e §4.4.3, anche se il contesto di dati assume una forma differente. Per quanto riguarda il grafico temporale, l'avvio della predizione comporta la sovrapposizione del segnale relativamente al quale essa viene calcolata e del singolo profilo o dei due profili predetti. In aggiunta, per ciascuno degli algoritmi scelti, possono essere plottati i marker dei valori che superano le soglie di 70 e 180 mg/dl. In particolare, all'interno della function `PredictionButtonCallback` man mano che viene calcolata la predizione vengono salvati il primo valore che supera i 180 mg/dl per ogni trend crescente



**Figura 6.11:** Pannello *Display Data and Results*, così come appare al lancio del modulo di predizione.

del profilo e il primo valore che scende al di sotto dei 70 mg/dl per ogni suo trend decrescente. Un esempio di visualizzazione con marker oltre soglia si evince dalla Figura 6.12.



**Figura 6.12:** Esempio di selezione dell'opzione che consente di plottare con marker i valori che superano le soglie e conseguente visualizzazione nel pannello grafico.

La codifica relativa risulta essere data da

```
% values that pass thresholds (plot with squares)
k = 1;
for i = 2:length(tipred1)
    if yipred1(i) <= 70
```

```

        if yipred1(i-1) > 70
            pr.square_values1.cgm(k) = yipred1(i);
            pr.square_values1.time(k) = tipred1(i);
            k = k+1;
        else
            continue
        end
    end
end
if yipred1(i) >= 180
    if yipred1(i-1) < 180
        pr.square_values1.cgm(k) = yipred1(i);
        pr.square_values1.time(k) = tipred1(i);
        k = k+1;
    else
        continue
    end
end
end
end
end

```

Nel caso di predizione con filtro di Kalman l'utente può scegliere di visualizzare gli intervalli di confidenza della stima. A tal proposito, è consigliabile il plot di tali intervalli solo se si sceglie un unico set di predizione poiché altrimenti il risultato grafico sarebbe inutilizzabile e graficamente non apprezzabile. In aggiunta a quanto detto, considerando il fatto che è possibile predire la glicemia mediante più algoritmi con orizzonti di predizione anche diversi, gli assi del grafico vengono in automatico settati sulla base dell'istante temporale più avanzato tra quelli dei profili a disposizione (6.13). La codifica di impostazione degli assi è quindi data da

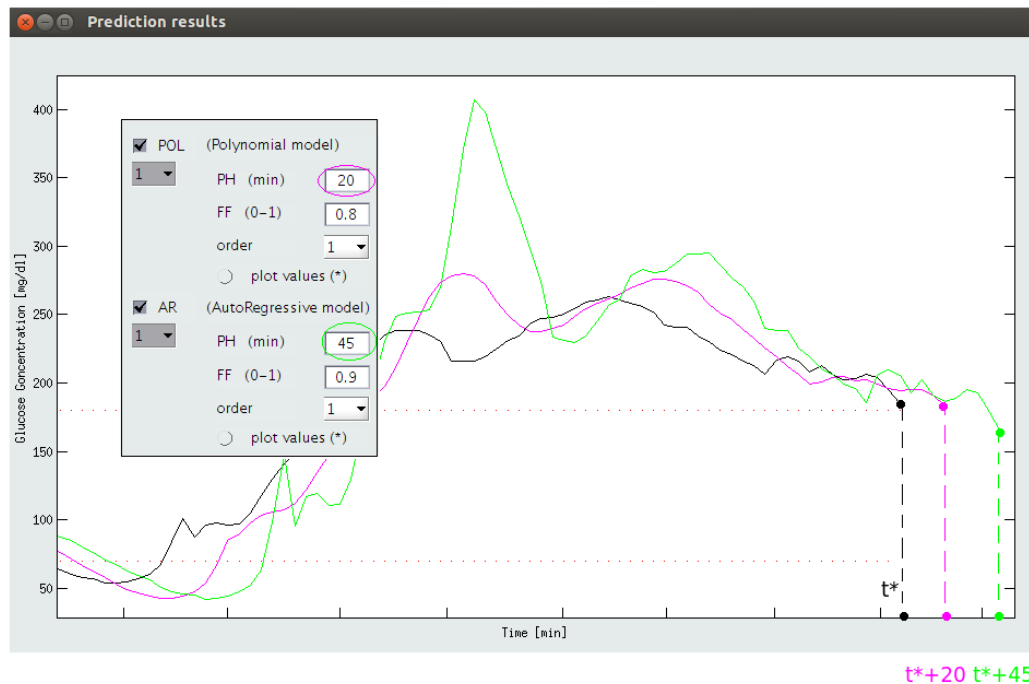
```

tend = max([pr.pred_values1.time(end) pr.pred_values2.time(end)]);
axis([init.cgm_values.time(1) tend 0 400])

```

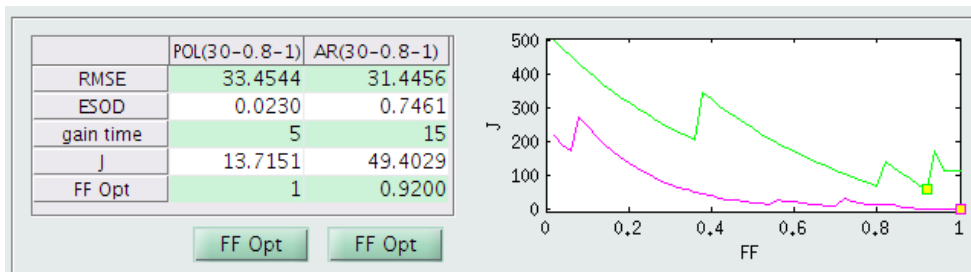
Successivamente al calcolo di predizione, dopo che sono stati uniformati e resi corrispondenti in maniera automatica i vettori dei risultati, si procede al calcolo degli indici quantitativi definiti in §5.3. La tabella viene aggiornata con i valori calcolati e le etichette relative alle colonne variano a seconda dell'algoritmo implementato e dei parametri fissati:

- i modelli polinomiali vengono identificati con POL(PH-FF-ordine);
- i modelli autoregressivi vengono identificati con AR(PH-FF-ordine);
- i predittori alla Kalman vengono etichettati con KF(PH).



**Figura 6.13:** Plot del segnale originario e dei profili predittivi con asse delle ascisse settato sulla base dell'ultimo valore temporale utile.

I pulsanti che consentono di calcolare il coefficiente di oblio ottimale in funzione di  $J$  vengono abilitati solo in presenza di modelli polinomiali o autoregressivi del primo ordine. Considerando il fatto che la predizione con modelli di ordine superiore è utile più a scopo didattico che clinico, essendo i profili risultanti sempre più irregolari e variabili, si è infatti pensato che avesse senso procedere al calcolo del forgetting factor ottimale solo per quei modelli che effettivamente potessero sfruttarne in maniera anche clinicamente utile la conoscenza. In Figura 6.14 viene mostrato il confronto quantitativo tra un modello polinomiale e un modello autoregressivo, con conseguente calcolo del forgetting factor ottimale per ciascuno dei due profili.



**Figura 6.14:** Tabella con gli indici quantitativi e corrispondente pannello grafico in cui è plottato l'andamento dell'indice  $J$  in funzione del forgetting factor. Il confronto avviene, in questo esempio, tra un modello polinomiale  $POL(30-0.8-1)$  e un modello autoregressivo  $AR(30-0.8-1)$ .



# Conclusioni

Nel presente lavoro si è visto come i dispositivi CGM rappresentino potenzialmente una notevole innovazione nella terapia del diabete. Un miglioramento nella gestione della patologia consente di ottenere considerevoli vantaggi sulla qualità della vita dei soggetti diabetici e sui costi economici di cui il sistema sanitario nazionale deve farsi carico. Tuttavia, la precisione e l'accuratezza dei dati CGM e la tempestività nella generazione degli allarmi di ipo e iperglicemia sono ancora subottime e ciò non consente di poter sfruttare a pieno le innumerevoli potenzialità che questi sistemi di monitoraggio offrono. Negli ultimi anni sono stati proposti diversi algoritmi per il denoising, la calibrazione, la predizione temporale e la generazione di allarmi, funzionanti in tempo reale ed in cascata all'output del sensore CGM, che consentono di superare, o quantomeno di migliorare, questi aspetti critici. Tali algoritmi, basati su metodi numerici avanzati, tecniche di filtraggio e modelli di dati, risultano codificati solo a livello prototipale e quindi, oltre a dover essere riadattati e reimplementati ad ogni nuovo studio di segnale, essi risultano praticamente non accessibili ad un utente medio e ciò indebolisce il meccanismo di trasmissione e di comunicazione efficace dei risultati al di fuori dell'ambiente di ricerca. Scopo del progetto di CGMLab in cui si inserisce la presente tesi è rendere più veloce ed efficiente lo studio dei segnali CGM, automatizzando l'implementazione dei suddetti algoritmi, e creando uno strumento user-friendly affinché anche un utente inesperto possa intuirne il funzionamento. CGMLab è stata realizzata per portare avanti gli obiettivi sopra elencati e si è visto essere costituita da due figure principali, comprendenti i quattro moduli di processamento dei segnali.

In particolare, nel software realizzato in questo lavoro di tesi, inizialmente è stata presentata la finestra di inserimento e caricamento dei dati. Si è visto come il software sia stato progettato per inserire dati di qualsiasi tipo, e di come questi dati vengano poi caricati in maniera robusta e automatica, tenendo conto dei dati mancanti e di eventuali sfasamenti temporali. Contemporaneamente è possibile inserire qualsiasi altro tipo di dato inerente allo studio del profilo glicemico. Questa prima finestra consente di rendere versatile e veloce il caricamento delle serie temporali da esaminare, ne favorisce la visualizzazione completa, eventualmente suddivisa in giorni, e il salvataggio in un file unico e predefinito. Alternativamente si dovrebbe procedere ricodificando di volta in volta i singoli passaggi e adattando variabili e parametri al caso in esame, con

conseguente perdita di efficienza e maggiore dispersività nell'analisi.

Successivamente è stato presentato il processo di denoising, implementato con filtro di Kalman mediante due approcci differenti a seconda delle ipotesi sul rapporto segnale-rumore. Questo tipo di filtraggio consente ricorsivamente di regolarizzare il segnale CGM, eventualmente simulandone il funzionamento in tempo reale.

Alternativamente si è parlato della possibilità di effettuare predizione sul segnale CGM, attraverso l'implementazione contemporanea di più modelli predittivi, alcuni di utilizzo comune, ma pur sempre da adattare allo specifico ambito e allo specifico segnale in esame, l'altro basato ancora una volta sul filtro di Kalman. È comprensibile come siano imparagonabili la situazione in cui ci si ritrova a dover gestire più script o funzioni diverse, cercando le righe di codice in cui modificare i parametri da settare, e la situazione in cui, in maniera immediata, intuitiva e completa, si vanno ad impostare i parametri desiderati, si aziona un pulsante e si visualizzano istantaneamente i risultati, con la possibilità di valutare e comparare tra loro quantitativamente i vari predittori.

I moduli di CGMLab realizzati in questa tesi sono stati testati da alcuni membri del Gruppo di Bioingegneria del DEI di Padova. In questo modo è stato possibile valutare in via preliminare la flessibilità e l'affidabilità dell'interfaccia realizzata, la sua accessibilità anche da parte di studiosi di altri settori, nonché il suo design e le sue mancanze. I risultati del test sono stati positivi e hanno suggerito anche delle vie da intraprendere per rendere ulteriormente più robusta la codifica in alcuni controlli.

Sviluppi futuri e miglioramenti dell'interfaccia illustrata possono indirizzarsi verso la sistemazione dei dettagli di codifica segnalati in fase di test e verso il completamento della finestra di processamento, cioè dei moduli di calibrazione e di generazione di allarmi. Il codice è open-source e quindi modificabile in ogni sua parte, affinché sia possibile innovare e arricchire il software a seconda delle nuove esigenze di ricerca che potrebbero di volta in volta presentarsi. CGMLab è un punto di partenza che potrebbe, attraverso la sua semplicità e consistenza, favorire in modo più immediato e convincente la trasmissibilità e l'utilizzabilità di risultati potenzialmente utili ad incrementare l'efficacia dei sensori CGM nella terapia del diabete.

# Bibliografia

- [1] Sito web: <http://www.idf.org>
- [2] Sito web: <http://www.diabetes.org>
- [3] Sito web: <http://www.bostonscientific-international.com>
- [4] Sito web: <http://www.who.int/diabetes/en>
- [5] Sito web: <http://www.epicentro.iss.it>
- [6] Sito web: <http://www.nhs.uk/Pages/HomePage.aspx>
- [7] Sito Web: <http://abbottdiabetescare.it/freestyle.htm>
- [8] Sito web: <https://www.abbottdiabetescare.com>
- [9] Sito web: <http://www.dexcom.com>
- [10] Sito web: <http://www.lifescan.com>
- [11] Sito web: <http://www.menarini.com>
- [12] Sito web: <http://www.medtronicdiabetes.com/products/continuousglucosemonitoring>
- [13] Amir O., Weinstein D., Zilberman S., Less M., Perl.Treves D., Primack H., Weinstein A., Gabis E., Fikhte B., Karasik A., *Continuous Noninvasive Glucose Monitoring Technology Based on Occlusion Spectroscopy*, J Diabetes Sci Technol, Vol. 1, No. 4, Luglio 2007
- [14] Tierney M.J., Tamada J.A., Potts R.O., Eastman R.C., Pitzer K., Ackerman N.R., Fermi S.J., *The GlucoWatch Biographer: a frequent, automatic and noninvasive glucose monitor*, Ann Med, 32(9), Dicembre 2000
- [15] Weiss R., Yegorchikov Y., Shusterman A., Raz. I., *Noninvasive continuous glucose monitoring using photoacoustic technology-results from the first 62 subjects*, Diabetes Technol Ther, 9(1), Febbraio 2007
- [16] Lesperance L.M. et al., *Calibration of the continuous glucose monitoring system for transient glucose monitoring*, Diabetes Technol Ther, 9(2), Aprile 2007

- [17] Ryan C. et al., *Modeling of Calibration Effectiveness and Blood-to-Interstitial Glucose Dynamics as Potential Confounders of the Accuracy of Continuous Glucose Sensors during Hyperinsulinemic Clamp*, J Diabetes Sci Technol, 7(1), Febbraio 2005
- [18] Knobbe E.J., Buckingham B., *The Extended Kalman filter for continuous glucose monitoring*, Diabetes Technol Ther, 1(3), Maggio 2007
- [19] A. Facchinetti et al., *Reconstruction of Glucose in Plasma from Interstitial Fluid Continuous Glucose Monitoring Data: Role of Sensor Calibration*, Journal of Diabetes Science and Technology Volume 1, Pubblicazione 5, Settembre 2007
- [20] A. Facchinetti et al., *On-Line Time-Series Prediction Models for Continuous Glucose Monitoring (CGM) Data*, Proceedings of 7<sup>th</sup> Annual Diabetes Technology Meeting (DTM), San Francisco (CA), Ottobre 2007
- [21] Guerra S., Facchinetti A., Sparacino G., De Nicolao G., Cobelli C., *Enhancing the accuracy of subcutaneous glucose sensors: a real-time deconvolution-based approach*, IEEE Transaction on Biomedical Engineering, 59, 2012
- [22] G. Sparacino et al., *Smart Continuous Glucose Monitoring Sensors: On-Line Signal Processing Issues*, Sensors 2010, 10, 6751-6722
- [23] G. Sparacino et al., *Italian Contributions to the Development of Continuous Glucose Monitoring Sensors for Diabetes Management*, Sensors 2012, 12, 13753-13780
- [24] Sparacino G., Facchinetti A., Maran A., Cobelli C., *Continuous glucose monitoring time series and hypo/hyperglycemia prevention: requirements, methods, open problems*, Curr Diabetes Rev, Vol. 4, No. 3, Agosto 2008
- [25] Appunti del corso di *Modelli e Controllo dei Sistemi Biologici*, 2010-2011
- [26] A. Facchinetti et al., *An Online Self-Tunable Method to Denoise CGM Sensor Data*, IEEE Transaction on Biomedical Engineering, Vol. 57, No. 3, Marzo 2010
- [27] A. Facchinetti et al., *Online Denoising Method to Handle Intraindividual Variability of Signal-to-Noise Ratio in Continuous Glucose Monitoring*, IEEE Transaction on Biomedical Engineering, Vol. 58, No. 9, Settembre 2011
- [28] Sito web: <http://www.mathworks.com>
- [29] P. Marchand et al., *Graphics and GUIs with MATLAB® (third edition)*, Chapman & Hall/CRC, 2003

- [30] W. O. Galitz, *The Essential Guide to User Interface Design - Second Edition - An Introduction to GUI Design - Principles and Techniques*, Wiley, 2002
- [31] Appunti del corso di *Analisi di Dati Biologici*, 2011-2012
- [32] A. Facchinetti, *On-line Filtering Algorithms for Continuous Glucose Monitoring*, tesi di dottorato presso l'Università di Padova, ciclo XXI, Gennaio 2009
- [33] Sparacino G., Zanderigo F., Corazza S., Maran A., Facchinetti A., Cobelli C., *Glucose concentration can be predicted ahead in time from continuous glucose monitoring sensor time-series*, IEEE Transaction on Biomedical Engineering, Vol. 54, No. 5, Maggio 2007
- [34] E. Trifoglio, *Il Monitoraggio Continuo del Glucosio: Sviluppo per una Metodologia per il Confronto e l'Ottimizzazione di Algoritmi Predittivi*, Tesi di laurea Magistrale in Bioingegneria, Università di Padova, A.A. 2009-2010