



UNIVERSITÀ
DEGLI STUDI
DI PADOVA



DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE

CORSO DI LAUREA IN INGEGNERIA BIOMEDICA

**IMPLEMENTAZIONE E VALUTAZIONE DI UN ALGORITMO PER
LA RILEVAZIONE DI MALFUNZIONAMENTI NEI SET DI
INFUSIONE PER IL DIABETE**

Relatore: Prof. Simone Del Favero

Laureando: Gabriele Gradara

ANNO ACCADEMICO 2021 – 2022

Data di laurea 14/03/2022

Ringrazio la mia famiglia ed i
miei amici per il supporto che
mi hanno fornito durante i miei
studi

Indice

1	Introduzione al diabete	5
1.1	Il diabete di tipo 1	5
1.2	Il diabete di tipo 2	6
1.3	La IGT e la IFG	7
1.4	Metodi di diagnosi	7
1.5	Complicanze	7
1.6	Costi umani ed economici	9
2	La pompa di insulina e l'infusion set	11
3	L'algoritmo per la rilevazione di LISA	14
3.1	Notazione	14
3.2	Metriche di valutazione utilizzate	15
3.2.1	Metrica di guasto legata al glucosio (Glucose Fault Metric, GFM)	15
3.2.2	Metrica di guasto legata all'insulina (Insulin Fault Metric, IFM)	15
3.2.3	Metrica di guasto legata alla variazione di glicemia (Glucose Slope, GS)	16
3.3	Performance dell'algoritmo	17
4	Valutazione dell'algoritmo per la rilevazione dei LISA	19
4.1	Introduzione	19
4.2	Proprietà dei dati utilizzati	19
4.3	La struttura del programma	20
4.4	L'analisi del Continuous Glucose Monitoring	21
4.5	L'analisi della Glucose Slope	22
4.6	L'analisi della Insulin Fault Metric	23
4.7	La valutazione degli allarmi	24
4.8	Grafici	26
4.9	Analisi dei risultati	27
5	Conclusione	28

Sommario

Il pancreas artificiale e la pompa di insulina sono dispositivi biomedici che hanno migliorato di molto la qualità della vita dei pazienti malati di diabete di tipo 1. Questi strumenti sono però soggetti ad occasionali malfunzionamenti meccanici che possono incidere gravemente sul controllo del glucosio nel paziente, soprattutto se crede di ricevere regolarmente la terapia.

Grazie all'utilizzo di sensori per il glucosio minimamente invasivi abbiamo accesso ad un monitoraggio continuo del glucosio (Continue Glucose Monitoring CGM), ed è quindi possibile implementare degli algoritmi per l'autodiagnostica per allertare il paziente di un possibile guasto.

In questo elaborato analizzeremo l'algoritmo per la rilevazione di malfunzionamenti nell'attuazione dell'infusion set o della pompa di insulina ideato da Daniel P. Howsmon ed i suoi collaboratori, per verificarne l'efficacia. Per farlo utilizzeremo i dati di pazienti *in silico* generati con il simulatore UVa/Padova.

Capitolo 1

Introduzione al diabete

1.1 Il diabete di tipo 1

Il diabete di tipo 1 è una malattia metabolica causata dall'insufficiente secrezione di insulina dovuta alla distruzione delle β -cellule pancreatiche, le cellule adibite alla produzione di insulina, che provoca un'iperglicemia cronica. A causa di diversi fattori sia genetici che ambientali si manifesta nel paziente una risposta immunitaria anomala che tende a ridurre drasticamente il numero di β -cellule, arrivando anche ad eliminarle completamente [1].

Le ragioni di questa risposta immunitaria non sono ancora chiare ma le teorie più accreditate prevedono la presentazione di peptidi delle β -cellule da parte di cellule dendritiche a dei linfociti T. Questi linfociti T a questo punto riconoscono le β -cellule come corpo estraneo e le attaccano, distruggendole.

Un'altra ipotesi riguarda la possibile presenza di un'infezione virale cronica delle β -cellule, questo porterebbe con il tempo a sviluppare una risposta autoimmune verso di loro. Molte infezioni virali, con l'enterovirus tra le più comuni, sono state associate al diabete di tipo 1 ma questa ipotesi rimane difficile da provare. Nonostante questo si stanno sviluppando terapie antivirali e vaccini contro l'enterovirus proprio per queste ragioni [2].

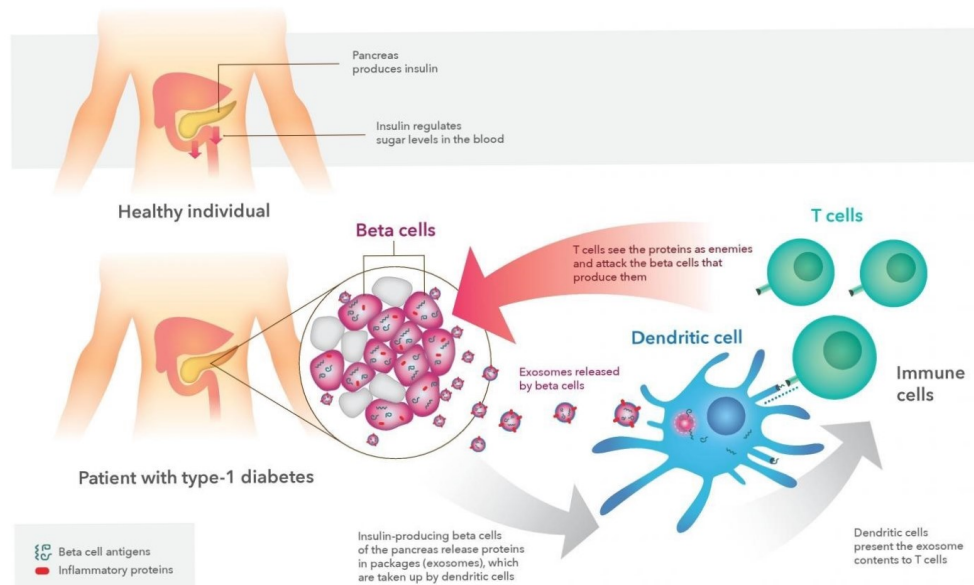


Figura 1.1. Rappresentazione di uno dei processi immunitari che portano il paziente ad avere il diabete di tipo 1 [3].

Questa attività rilascia nel sangue degli anticorpi che possono essere utilizzati come utili biomarcatori per riconoscere la malattia prima dell'effettiva manifestazione dei sintomi o di una iperglicemia tale da permettere la diagnosi della malattia [4]. La rilevazione di uno di questi anticorpi può precedere l'onset della malattia anche di anni o di decenni [1]. Subito dopo la diagnosi può avvenire una fase di "luna di miele" in cui le β -cellule riescono ad aumentare la secrezione di insulina. In questo periodo la necessità di assumere insulina si riduce ed è più semplice per il paziente gestire la propria glicemia. Tuttavia questo periodo ha una durata limitata e presto ridiventa necessaria una terapia più intensiva a causa dello stress che la richiesta maggiorata impone alle β -cellule sommato alla continua perdita di queste [1][2]. Quando il numero di β -cellule si riduce troppo diventa definitivamente necessaria una assunzione esogena di insulina per il mantenimento dell'omeostasi glicemica.

La quantità di C-peptide¹ prodotta autonomamente dal paziente può essere molto eterogenea, con pazienti che dopo decenni riescono ancora a produrne una minima parte. Le ragioni che hanno permesso ad alcune β -cellule di salvarsi dagli attacchi del sistema immunitario non sono ancora chiari, ma potrebbero avere del potenziale terapeutico [2]. Abbiamo inoltre delle prove che ci dicono che nei bambini è presente una rigenerazione della massa di β -cellule perse, questo però non accade negli adolescenti o negli adulti [6].

1.2 Il diabete di tipo 2

Nel diabete di tipo 2 invece si ha una riduzione progressiva nella produzione e/o nell'efficacia di insulina, l'obesità (soprattutto l'obesità addominale) [7], l'inattività fisica, una dieta ipercalorica e la familiarità con un malato di diabete di tipo 2 rappresentano i principali fattori di rischio [5]. L'obesità infatti richiede un'attività pancreatiche maggiorata per mantenere una glicemia adeguata, capacità che si riduce gradualmente con il tempo, portando al diabete. L'accumulo di grassi induce una resistenza all'insulina tramite cambiamenti in alcuni secreti, tra cui degli ormoni. Il rilascio di ormoni protettivi come l'adiponectina si riduce con l'aumento di adipociti, soprattutto nell'addome. L'inattività fisica oltre a favorire l'obesità induce anch'essa la resistenza all'insulina [7].

Non esistono solo due tipologie di diabete² ma queste sono le più comuni, molte inoltre condividono alcune similarità. E' importante fare presente che questa malattia si presenta in una forma molto individuale e che quindi il decorso e le terapie necessarie per due pazienti anche con la stessa diagnosi può essere molto differente.

¹Il C-peptide è un sottoprodotto della produzione di insulina ed è rilasciato insieme a questa nel sangue. La quantità di C-peptide rilasciata è equimolare a quella dell'insulina [5]. Il C-peptide ha un ruolo importante perché ci permette di determinare la produzione endogena di insulina del paziente, in quanto l'insulina commerciale è sprovvista di C-peptide.

²Giusto per citare un paio di esempi: esistono anche il diabete gestazionale che colpisce le donne in gravidanza ed il diabete di tipo 1B, una variante molto più rara del diabete di tipo 1 convenzionale. Questa variante appare per lo più in pazienti di discendenza asiatica o africana e la sua eziologia è ancora sconosciuta [1].

1.3 La IGT e la IFG

Prima di arrivare al diabete di tipo 2 un individuo sano può soffrire di intolleranza al glucosio (Impaired Glucose Tolerance, IGT) o da glicemia a digiuno alterata (Impaired Fasting Glucose, IFG). Queste patologie sono asintomatiche e reversibili, infatti oltre il 30% dei pazienti ritorna ad avere una tolleranza adeguata dopo molti anni, ma chi ne soffre ha un alto rischio di sviluppare il diabete di tipo 2 [4]. Il termine usato per indicare i pazienti affetti da una od entrambe le condizioni è prediabete [8]. La IFT e la IGT sono già di per se fattori di rischio per lo sviluppo di malattie cardiovascolari e disturbi metabolici, è quindi consigliabile ripristinare uno stato di salute sano al più presto [9].

1.4 Metodi di diagnosi

Sono disponibili vari modi per diagnosticare il diabete: [6][2]

- Glicemia a digiuno superiore alle 7,0 mmol/dL (126mg/dL)
- Glicemia in qualsiasi momento superiore a 11,1 mmol/dL (200mg/dL)
- Un risultato anomalo in un test di tolleranza al glucosio
- Una concentrazione di emoglobina glicata³ (HBA_{1c}) superiore a 48 mmol/dL (6,5%).

1.5 Complicanze

L'insulina è fondamentale per il corretto funzionamento del metabolismo del glucosio, legandosi a degli specifici recettori sulla superficie della membrana permette a gran parte dei tessuti (con alcune importanti eccezioni come il cervello) di incamerare il glucosio e sfruttarlo come fonte di energia. L'insulina è inoltre responsabile per la trasformazione del glucosio in glicogeno nei muscoli e nel fegato (glicogenesi) e dell'inibizione della produzione di glucosio tramite l'utilizzo di grassi ed amminoacidi (gluconeogenesi) bloccando parzialmente la metabolizzazione di grassi e proteine. In sua assenza il tessuto è costretto a un utilizzo spropositato di acidi grassi e proteine per sopperire ai bisogni energetici, provocando un accumulo di sottoprodotti acidi. Ciò causa una chetoacidosi che se non viene trattata può diventare molto pericolosa per il paziente, anche potenzialmente letale.

³L' HBA_{1c} è una molecola di emoglobina legata ad una molecola di glucosio che perdura per tutta la durata vitale dell'eritrocita. Questo legame è favorito soprattutto se la glicemia è elevata, e dato che HBA_{1c} rimane in circolo per 2-3 mesi è un buon indicatore della glicemia media passata [4].

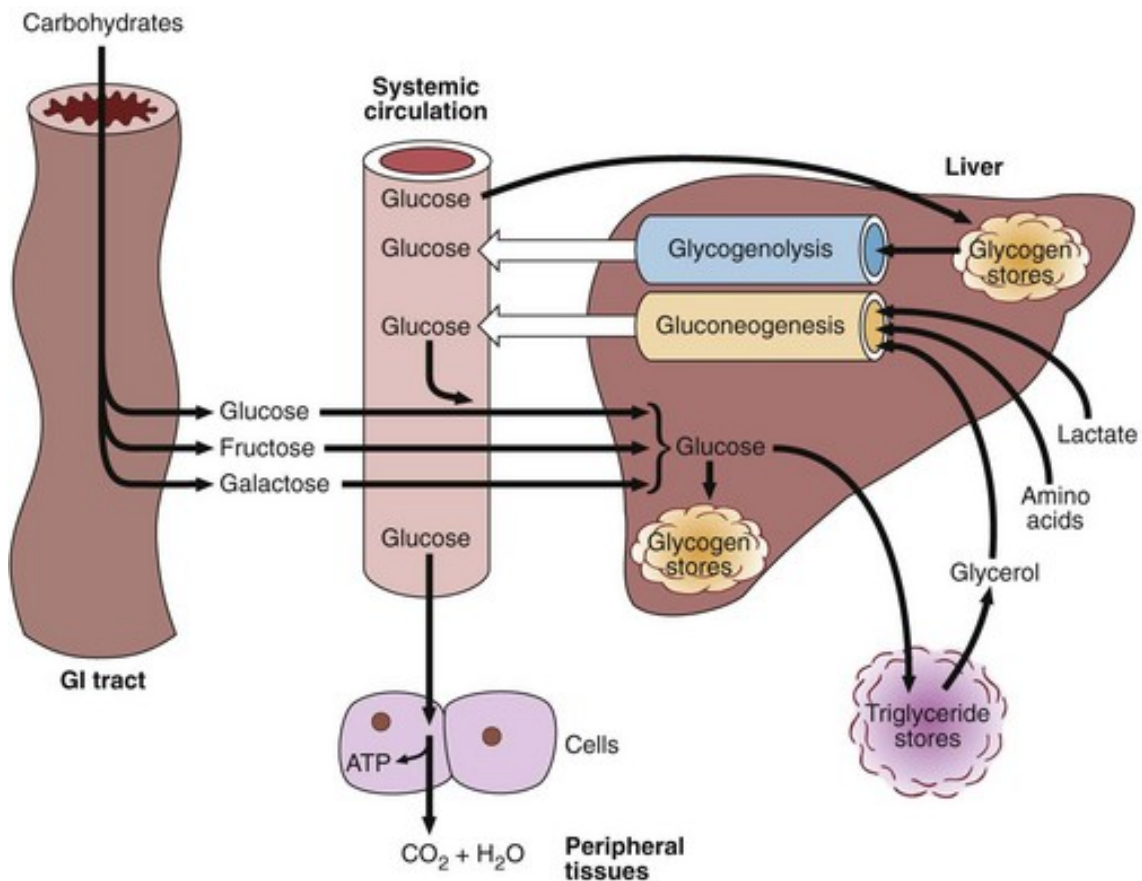


Figura 1.2. Rappresentazione schematica del metabolismo dei carboidrati.

L'iperglicemia tende inoltre ad aumentare l'osmolalità del sangue arrivando a casi estremi con glicemia superiore a 50 mmol/L (900 mg/dL) in cui può disidratare gravemente le cellule, in questi casi a causa della disidratazione delle cellule cerebrali si può arrivare anche al coma [5].

La riduzione nell'erogazione o nell'efficacia dell'insulina può compromettere l'omeostasi e portare all'iperglicemia con gravi complicazioni per il paziente, che possono manifestarsi nel lungo periodo. I meccanismi che mettono in moto l'insorgere di queste complicazioni sono molto complessi e non del tutto chiari [4].

Le principali sono:

- **Neuropatia**

Il diabete ha vari effetti sul sistema nervoso periferico, ma il più comune è la perdita di sensibilità agli arti (in particolare ai piedi ed in misura minore alle mani). Spesso la neuropatia non ha sintomi immediatamente riconoscibili anche se occasionalmente può manifestarsi molto dolorosamente, ma in ogni caso mette il paziente a rischio di ulcera, infezione e successiva amputazione dell'arto coinvolto.

Le persone con il diabete sono soggette a 25 volte il rischio di amputazione del resto della popolazione.

- **Nefropatia**

Il diabete è una delle cause più importanti di insufficienza renale. Infatti la causa più comune di malattia renale cronica in forma terminale, con necessità di sottoporsi a frequenti dialisi o ad un trapianto, è proprio il diabete.

- Retinopatia

La retinopatia diabetica è una delle complicazioni più caratteristiche del diabete nonché una delle più semplici da trattare. Dato che spesso il diabete di tipo 2 rimane non diagnosticato per molti anni, molti pazienti possono sviluppare alcune complicazioni, tra cui la retinopatia, ancora prima della diagnosi.

- Malattie cardiovascolari

Le malattie cardiovascolari, tra cui l'infarto del miocardio e l'ictus sono la causa principale di morte nei pazienti diabetici. I pazienti con il diabete che non hanno mai avuto un infarto del miocardio sono considerati soggetti allo stesso rischio di pazienti senza diabete ma che hanno avuto un infarto del miocardio in passato.

Sono anche possibili altre complicazioni di natura meno grave ma che possono compromettere comunque la salute e la qualità della vita del paziente: come la disfunzione erettile [10].

Un'eccessiva somministrazione di insulina, alcuni fattori ambientali, lo stress, l'attività fisica, o il mancato consumo di pasti da parte del paziente possono ridurre drasticamente la glicemia. Se scende sotto i 70mg/dL il paziente rischia di incorrere in ipoglicemia, una condizione molto pericolosa perché può portare al coma e diventare rapidamente letale. E' quindi importante che il paziente sappia distinguere i sintomi che precedono una carenza grave di glucosio per poter intervenire rapidamente, spesso consumando dello zucchero, in modo da reintegrarlo nell'organismo. Il momento più pericoloso per l'insorgenza di ipoglicemia è la notte durante il riposo.

L'ipoglicemia va evitata ad ogni costo perché ripetuti episodi ipoglicemici possono portare ad una inconsapevolezza dell'ipoglicemia riducendo i sintomi che possono mettere in allarme il paziente, mettendolo in pericolo soprattutto se non è costantemente monitorato [5].

1.6 Costi umani ed economici

Si stima che ad oggi 537 milioni di adulti (di età tra i 20 ed i 79 anni) in tutto il mondo abbiano il diabete, ovvero il 10.5% degli adulti [8]. Si stima che circa il 90% dei pazienti malati di diabete sia affetto da diabete di tipo 2. Questi numeri sono ancora più sorprendenti se consideriamo che nel 2003 la previsione per il 2025 sarebbe stata di 333 milioni di adulti con il diabete [7]. Quindi trovare metodi efficaci per migliorare la qualità della vita e massimizzare l'efficacia delle cure è sicuramente di grande interesse per contrastare quella che a conti fatti è una pandemia [7].

Per trattare i pazienti adulti che soffrono di diabete si sono spesi a livello mondiale USD 232 miliardi nel 2007 e USD 966 miliardi nel 2021 con un aumento della spesa sostenuta dai sistemi sanitari mondiali del 315% raggiunto in soli 14 anni [8].

L'aumento della prevalenza del diabete si ritiene sia maggiore nei paesi che passano da livelli bassi a medi di reddito, con una popolazione che sta invecchiando e che adotta progressivamente uno stile di vita sempre più legato all'urbanizzazione. Le previsioni ci dicono che il diabete prenderà piede in maniera allarmante in Africa, dove si stimano che ora ci siano 23,6 milioni di persone con il diabete, [8] dove delle cure di qualità non possono essere garantite a gran parte dei pazienti mettendo in ginocchio un sistema sanitario già compromesso [11].

Un approccio basato sulla prevenzione, l'istruzione del paziente riguardo il diabete ed una terapia adatta per la riduzione di complicanze si è rivelato efficace nel

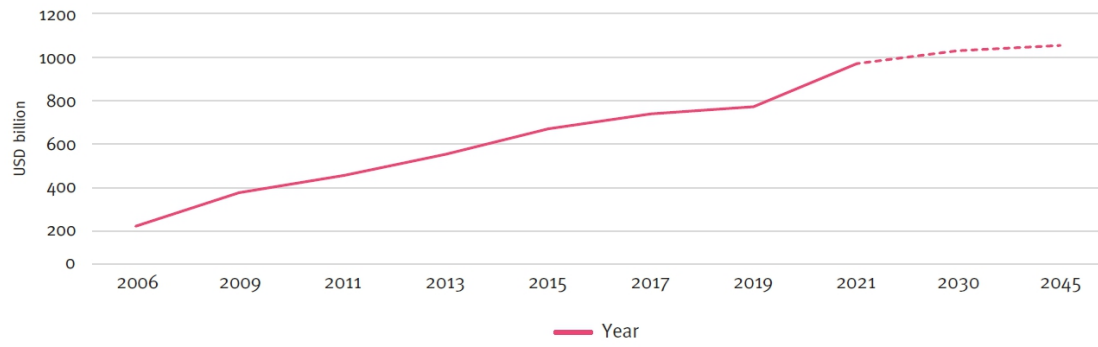


Figura 1.3. Spese sanitarie totali legate al diabete per degli adulti (età 20-79 anni) con il diabete dal 2006 al 2045

ridurre i costi. Gran parte dei costi umani ed economici è dovuta proprio all'emergere delle complicanze tipiche del diabete, è quindi preferibile mettere il paziente in condizione di cambiare le proprie abitudini ed informarlo esaustivamente sulla propria situazione di salute [4].

Da questi dati si evince quindi un forte interesse nello sviluppare tecnologie innovative per trattare ancora più efficacemente tutti i tipi di diabete e scoprire in maniera approfondita i meccanismi della malattia.

Capitolo 2

La pompa di insulina e l'infusion set

La terapia di solito consiste in una somministrazione di insulina (od un suo omologo con un comportamento differente per ottimizzarne l'efficacia terapeutica) per poter contenere la glicemia dentro un intervallo sicuro. La somministrazione avveniva tradizionalmente con l'iniezione di più boli nel corso della giornata ma con l'avvento delle pompe di insulina ed il pancreas artificiale si è passati ad un'infusione di insulina basale con occasionali boli di insulina in prossimità dei pasti. In base al tipo di sistema usato può essere o meno richiesto l'intervento dell'utente per l'inserimento della stima di carboidrati ingeriti.

Per trattare il paziente diabetico si può utilizzare la pompa di insulina e l'infusion set open-loop (la terapia con pompa di insulina convenzionale) o closed-loop (noto anche come pancreas artificiale).



Figura 2.1. Una pompa di insulina collegata al set per l'infusione. È presente anche il dispositivo per l'applicazione dell'ago minimamente invasivo per la somministrazione sottocutanea.

La terapia tradizionale prevede la somministrazione di tre boli di insulina durante la giornata, prima dei pasti. Questo approccio però non permette uno stretto controllo della glicemia che può molto spesso uscire dal range prefissato. È infatti provato che l'approccio migliore sia quello di simulare il naturale comportamento del pancreas, con un rilascio lento e controllato durante la giornata dell'insulina basale ed una somministrazione di boli al momento del pasto. In alternativa si possono assumere dosi di insulina di natura diversa alternando iniezioni di insulina a lunga azione con iniezioni di insulina ad azione rapida prima dei pasti [4]. Esistono in commercio molti tipi di insulina prodotti da aziende farmaceutiche diverse con proprietà, come la durata dell'effetto o il tempo necessario per avere effetto, differenti. Si può quindi assumere più tipi di insulina diversi durante la giornata per personalizzare la propria terapia.

La terapia open-loop non necessita di CGM o se vengono utilizzati non ne prevede una collaborazione con il software della pompa stessa, come nel caso della Sensor Augmented Pump therapy (SAP) perché eroga una terapia gestita dal paziente stesso. È comune che la pompa richieda una stima dei carboidrati ingeriti durante i pasti per poter dosare accuratamente il bolo. La pompa permette di somministrare facilmente più boli al giorno insieme all'insulina basale. La pompa di insulina inoltre è molto più precisa nelle dosi di una iniezione manuale, con la possibilità di pompare anche solo 0.1 unità di insulina l'ora [4]. A differenza della terapia standard non è possibile utilizzare durante la giornata più tipi di insulina perché le pompe hanno un solo serbatoio.

La terapia closed-loop invece prevede l'uso di CGM che diventa fondamentale per poter monitorare in tempo reale l'andamento della glicemia e di conseguenza reagire tempestivamente ai pasti o agli snack, ad eventi iperglicemici (aumentando la dose erogata) o agli eventi ipoglicemici (cessando la somministrazione di insulina) di modo da poter massimizzare il tempo in cui la glicemia è in un intervallo accettabile. Il pancreas artificiale si è rivelato la terapia al momento più efficace per tenere sotto stretto controllo la glicemia del paziente limitando gli eventi iperglicemici ed ipoglicemici. Grazie alla potenza di calcolo degli smartphone è possibile processare gli algoritmi di controllo su uno smartphone Android[®] off the shelf. In questo modo il paziente può inoltre avere accesso ai dati dell'AP e modificarne i parametri tramite un'apposita interfaccia. Le funzionalità di connettività del dispositivo aprono molte possibilità: come il monitoraggio da remoto utilizzando la connessione internet del cellulare [12].

Rispetto alle iniezioni multiple giornaliere (Multiple Daily Injections, MDI) l'infusione sottocutanea continua di insulina (Continuous Subcutaneous Insuline Infusion) è un regime molto più costoso. È necessario infatti sostenere il costo della pompa (che varia tra i 2000€ e 3000€) ed il costo ricorrente del set di infusione (variabile tra 10€ e 20€) che va cambiato regolarmente dopo 2-3 giorni per evitare occlusioni ed infezioni. Va inoltre considerato il costo del sensore di glucosio per il CGM (mediamente sui 50€) che ha una durata limitata (spesso di 14 giorni) e che quindi va cambiato per mantenere alta l'affidabilità delle letture. Il costo rappresenta uno dei più grandi limiti all'adozione su vasta scala di questa tecnologia [13].

Sono disponibili anche delle pompe dual-hormone che prevedono l'impiego dell'insulina e del glucagone (l'ormone antagonista dell'insulina) per tutelare il paziente da una possibile ipoglicemia. La sua utilità rimane comunque dubbia in quanto si ritiene che un controllo sufficientemente adeguato del pancreas artificiale con la sola insulina possa comunque prevenire gli episodi ipoglicemici [14].

Sul mercato esistono vari tipi di set per l'infusione di insulina (Insuline Infusion Set, IIS). Principalmente differiscono per il materiale del catetere, di solito Teflon o acciaio e per l'angolo di ingresso sotto cute che può essere perpendicolare o inclinato. Per i cateteri di Teflon è suggerito l'uso di un applicatore automatico per ridurre il dolore, semplificare l'applicazione e ridurre la possibilità di inserimento errato. A causa della consistenza morbida del materiale è infatti comune che la cannula si pieghi o arricci anche con l'uso dell'apposito applicatore (fino al 10% delle volte). I produttori rendono noto che le cannule di entrambi i materiali possono incorrere in occlusioni.

Il tempo di utilizzo suggerito del IIS per limitare l'insorgere di dermatiti, infezioni ed irritazioni è di 2-3 giorni. Il sito di applicazione può essere soggetto ad infezione, infatti l'IIS rimane più giorni nello stesso posto e sotto l'adesivo si forma un microclima caldo e umido perfetto per la proliferazione batterica. Il trauma di una applicazione dura più giorni mentre quello di un'iniezione si limita a qualche secondo, questo può comportare un'irritazione locale ed un rilascio maggiore di citochine che favoriscono l'irritazione¹. Questo rischio cresce più aumenta il tempo di utilizzo dello stesso IIS, per questo si consiglia di cambiare spesso il punto di applicazione e di disinfettare correttamente la zona prima di inserire l'ago [15].

Similmente alle iniezioni non tutti i punti di applicazione sono uguali. Applicare l'IIS all'addome è il metodo più efficace per abbassare la glicemia il più rapidamente possibile. La scelta di altri luoghi (ad esempio la coscia) potrebbe avere lo stesso impatto di una riduzione della dose di insulina iniettata o agire in ritardo [15].

¹Durante la sperimentazione è stato usato un dispositivo dall'ago più grande e capace di aspirare e pompare fluido attraverso il catetere. Non è quindi del tutto chiaro se è possibile considerare questi dati validi anche per l'IIS.

Capitolo 3

L'algoritmo per la rilevazione di LISA

Le perdite nell'attuazione del set di infusione (Losses in Infusion Set Actuation, LISA) comportano una assenza totale o parziale della somministrazione dell'insulina e quindi il paziente tenderà ad andare in iperglicemia. Un'iperglicemia non controllata dalla terapia può essere una causa di complicazioni del quadro clinico del paziente nel lungo termine e più raramente nel breve termine. Il vero pericolo però risiede nel fatto che il paziente è convinto di ricevere regolarmente la terapia e quindi potrebbe inavvertitamente peggiorare la propria situazione. Questo algoritmo vuole essere robusto sia riguardo la variabilità tra pazienti (o con lo stesso paziente con il passare del tempo) sia riguardo il tipo di set-up usato per la terapia. Infatti nonostante sia previsto l'utilizzo del CGM non è richiesto che il dispositivo sia in closed-loop od in open-loop, è sufficiente che l'algoritmo sia a conoscenza della terapia correntemente erogata al paziente.

In breve, l'algoritmo valuta la glicemia e l'insulina stimata presente nel paziente nel breve periodo rispetto ai suoi valori storici, facendo così può confrontarli. Se le letture sono troppo alte rispetto ai valori passati e se rileva un aumento brusco della glicemia, fa scattare l'allarme.

Questo algoritmo è stato creato da Howsmon e dai suoi collaboratori, in seguito ne discuteremo i dettagli [16].

3.1 Notazione

Definiamo il significato delle seguenti notazioni:

- k : rappresenta il tempo in minuti
- LW : rappresenta la lunghezza della finestra lunga
- SW : rappresenta la lunghezza della finestra corta
- $\bar{x}_{i|j}$: rappresenta la media di x nella finestra di tempo che va da j a i
- Δx : rappresenta il valore $x_k - x_{k-1}$

3.2 Metriche di valutazione utilizzate

3.2.1 Metrica di guasto legata al glucosio (Glucose Fault Metric, GFM)

Gli aumenti repentini della glicemia sono un mezzo fondamentale per riconoscere un LISA, in quanto ci si può aspettare che in seguito ad un'erogazione insufficiente di insulina si alzi improvvisamente la glicemia.

Per tenere conto delle variabilità ci affidiamo ad un sistema che considera due diverse finestre di tempo: una lunga ed una breve. Vengono poi considerate tutte le misure CGM presenti in queste finestre e ne calcoliamo la media. Le indicheremo come $\overline{CGM}_{k|k-LW}$ per la media del CGM nella finestra lunga e $\overline{CGM}_{k|k-SW}$ per la media del CGM nella finestra corta.

La finestra lunga ci permette di capire la media glicemica del paziente sul lungo periodo e quindi calibrare l'algoritmo attorno alle sue personali necessità. La finestra breve invece ci permette di avere il quadro corrente della sua glicemia.

Quando il $\overline{CGM}_{k|k-SW}$ supera il $\overline{CGM}_{k|k-LW}$ sapremo che il paziente sta superando la sua media glicemica e ne terremo conto calcolando la differenza tra le aree sotto la curva (AUC_k) delle due finestre di tempo. Consideriamo un Δt_k tra le acquisizioni di 5m.

$$AUC_k = (\overline{CGM}_{k|k-SW} - \overline{CGM}_{k|k-LW}) \times \Delta t_k$$

$$GFM_k = \begin{cases} GFM_{k-1} + AUC_k & \text{se } \frac{\overline{CGM}_{k|k-SW}}{\overline{CGM}_{k|k-LW}} > 1 \\ 0 & \text{altrimenti} \end{cases}$$

Utilizzeremo poi la GFM così ottenuta in sede di valutazione per determinare se è presente un guasto.

3.2.2 Metrica di guasto legata all'insulina (Insulin Fault Metric, IFM)

Un monitoraggio costante dell'andamento glicemico non è però sufficiente per determinare la presenza o meno di un guasto, è possibile infatti che il paziente stia consumando un pasto oppure uno snack.

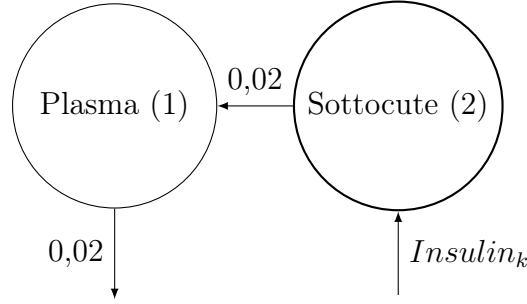
Molti dispositivi non richiedono l'inserimento di dati riguardo il pasto e per quelli che lo necessitano non si può dare per scontato che il paziente utilizzi sempre il calcolatore di bolo presente sulla sua pompa di insulina. Quindi per rendere l'utilizzo di questo algoritmo il più ampio possibile è stata ideata una metrica aggiuntiva che considera la presenza di insulina nel paziente, la IFM.

L'insulina viene somministrata per via sottocutanea, quindi per stimare l'insulina presente nel sangue è necessario utilizzare un modello compartimentale. Nel seguente sistema x_{k+1}^1 e x_{k+1}^2 sono la stima dei valori al momento discreto successivo dell'omonimo compartimento 1 o 2. Il compartimento 1 rappresenta l'insulina in circolo nel plasma mentre il compartimento 2 rappresenta l'insulina presente sotto cute, in sede di somministrazione. L'insulina proveniente dalla pompa può andare

solo nel compartimento sottocutaneo, per poi trasferire il 2% del proprio contenuto al plasma. Ad ogni istante discreto il plasma però ne deteriora il 2% tramite i naturali processi metabolici. Non è prevista l'azione di disturbi.

A differenza dell'acquisizione CGM gli intervalli tra i valori non sono di 5m ma di 1m, è quindi necessario fare particolare attenzione in fase di esecuzione dell'algoritmo.

Figura 3.1. Rappresentazione schematica del modello compartimentale. Il compartimento 1 rappresenta il plasma, mentre il compartimento 2 è quello sottocutaneo.



$$\begin{bmatrix} x_{k+1}^1 \\ x_{k+1}^2 \end{bmatrix} = \begin{bmatrix} 0,98 & 0,02 \\ 0 & 0,98 \end{bmatrix} \begin{bmatrix} x_k^1 \\ x_k^2 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} Insulin_k$$

$$PIE_k = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} x_k^1 \\ x_k^2 \end{bmatrix}$$

Dalla PIE calcoliamo poi le medie $\overline{PIE}_{k|k-SW}$ e $\overline{PIE}_{k|k-LW}$ per ragioni identiche al precedente CGM. Utilizziamo poi le medie ottenute per calcolare IFM.

$$IFM_k = \frac{\overline{PIE}_{k|k-SW} - \overline{PIE}_{k|k-LW}}{\overline{PIE}_{k|k-LW}} = \frac{\overline{PIE}_{k|k-SW}}{\overline{PIE}_{k|k-LW}} - 1$$

Se nel paziente è presente una glicemia anomala molto alta rispetto alla sua media nonostante l'insulina somministrata sia superiore alla media è plausibile credere che sia avvenuto un guasto.

3.2.3 Metrica di guasto legata alla variazione di glicemia (Glucose Slope, GS)

Per tutelare il paziente da un'eccessiva presenza di falsi allarmi utilizzeremo un ultimo valore, la GS. La GS è semplicemente costituita da una stima della derivata della glicemia.

$$GS_k = \frac{CGM_k - CGM_{k-1}}{\Delta t_k}$$

È infatti ragionevole pensare che durante un guasto la glicemia del paziente tenda a salire abbastanza rapidamente.

3.3 Performance dell'algoritmo

Riassumendo dalla ricerca di Howsmon e collaboratori: l'algoritmo è stato testato su un totale di 51 pazienti, provenienti da 3 studi diversi. Il dataset di uno degli studi, contenente 20 pazienti è stato utilizzato per la scelta delle soglie in modo da ottimizzare la performance. Il valore dei parametri è stato analizzato tramite la curva pROC¹ e sono stati scelti i valori che avrebbero garantito una sensibilità buona ed un numero accettabile di falsi positivi al giorno.

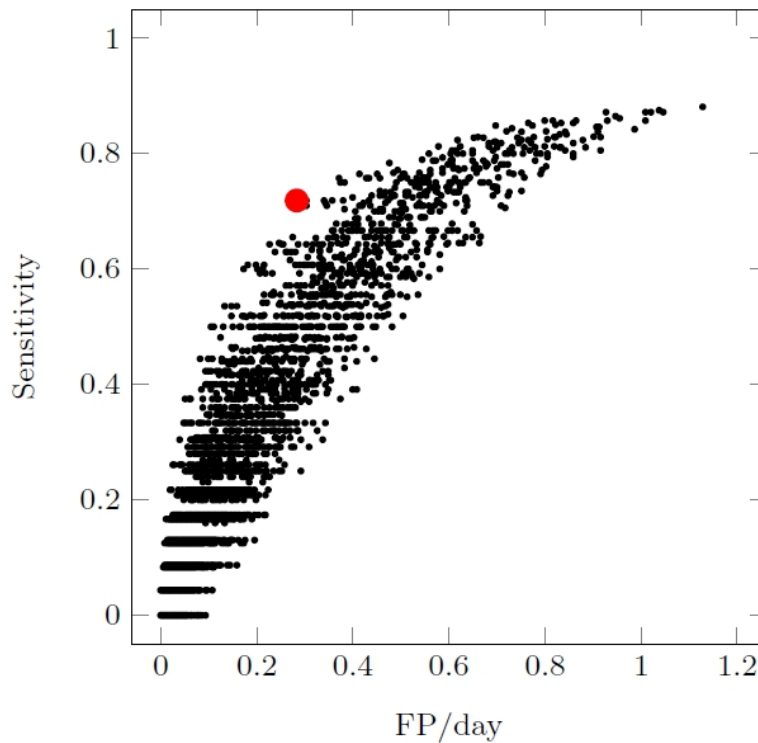


Figura 3.2. Il grafico della curva pROC usato da Howsmon ed i suoi colleghi per scegliere i valori più adatti per i coefficienti. Il valore evidenziato in rosso è quello scelto.

L'algoritmo dotato di queste soglie è stato poi utilizzato per rilevare la presenza di guasti negli altri 31 pazienti, che fanno parte di un data set completamente indipendente. Ciò ci permette di poter garantire una buona generalizzazione verso i dati di altri pazienti ed escludere un semplice adattamento ai dati già in possesso.

¹La pseudo Receiver Operating Characteristic curve (pROC curve) è un grafico che mostra la probabilità di rilevazione contro la probabilità di falso allarme per ciascun set di parametri. Differisce dalla ROC curve perché non ha la specificità sull'asse delle ascisse ma il numero di falsi positivi al giorno, questo tipo di grafico è uno strumento comunemente usato nella teoria delle decisioni per ottimizzare la scelta dei parametri.

Tabella 3.1. Parametri scelti per ottimizzare l'algoritmo

Nome del parametro	Unità	Valore scelto
LW	h	24
SW	h	1
Soglia di GFM	$\frac{mg}{dL} \times min$	100
Soglia di IFM	numero puro	0,4
Soglia di GS	$\frac{mg}{dL} \times min$	0,3

Tutti i dati sono stati ottenuti da delle ricerche per la rilevazione clinica di guasti nel set di infusione. Nonostante questo però non ci è noto l'esatto momento in cui questi guasti sono comparsi. Si è potuto quindi solo stimare il momento in retrospettiva, generalmente il team di diagnosi ha scelto l'istante precedente al guasto con il valore del CGM inferiore.

Ai fini della valutazione della performance valuteremo come allarmi accettabili (e quindi contati ai fini della sensibilità) tutti quelli che sarebbero scattati precedentemente all'intervento del personale medico che partecipava agli studi.

Con i valori scelti si è ottenuta una sensibilità al 71,8% e 0,28 falsi positivi al giorno. L'allarme tende a scattare quando il paziente è già in iperglicemia e quindi l'algoritmo non riesce a prevenirla, considerando però che la risposta dell'algoritmo è generalmente più rapida di quella del paziente, è ragionevole credere che questa implementazione possa comunque limitare le conseguenze da iperglicemia da LISA. Grazie al rapido intervento del personale medico si stima che in una situazione non altrettanto strettamente controllata la sensibilità possa essere ancora più alta. Infatti la media della durata dei guasti mancati dall'algoritmo è di 4,9h mentre quella dei guasti correttamente riconosciuti è di 10,6h.

In uno degli studi considerati per la valutazione delle performance dell'algoritmo è stata usata un'insulina associata a della ialuronidasi per ridurre il tempo necessario per l'attivazione dell'effetto dell'insulina. Questa somministrazione aggiuntiva ha cambiato il profilo farmacocinetico dell'insulina e reso molto meno efficace l'algoritmo, che raggiunge una sensibilità del 44% con 0,33 falsi positivi al giorno, un risultato decisamente peggiore rispetto alla somministrazione di sola insulina. Quindi l'utilizzo di questo algoritmo andrebbe limitato alla sola insulina che non possiede un profilo farmacocinetico troppo diverso da quella assunta nelle ipotesi.

Nonostante questo algoritmo sia stato sviluppato per la rilevazione di guasti in presenza di pasti ha rilevato con successo tutti e tre i guasti comparsi durante la notte, è però probabile che un algoritmo pensato ad hoc per la rilevazione notturna performi meglio.

Sono stati rilevati anche delle buone performance in sistemi closed-loop. Questi risultati sono però preliminari e richiedono ulteriore ricerca.

Capitolo 4

Valutazione dell'algoritmo per la rilevazione dei LISA

4.1 Introduzione

Per produrre questo elaborato e valutare la qualità dell'algoritmo questo è stato ricreato utilizzando MATLAB. L'ambiente di lavoro utilizzato è stato sempre MATLAB versione R2019b. Il testo integrale del programma utilizzato potrà essere consultato nell'appendice A.

Questo programma ha come obiettivo la replica accurata dell'algoritmo ideato da Howsmon et al in modo da verificarne la performance. Per farlo si utilizzerà un dataset di 5 pazienti *in silico* prodotti con il simulatore UVa/Padova. Riterremo l'allarme generato corretto se verrà lanciato entro 6h dall'onset del guasto.

Sarà inoltre possibile monitorare tramite grafici ogni singola metrica utilizzata per il lancio dell'allarme di modo da poter apprezzarne gli andamenti e le differenze tra i pazienti.

4.2 Proprietà dei dati utilizzati

I dati utilizzati da questo programma sono quelli di 5 pazienti *in silico* monitorati costantemente per 1 mese, questi pazienti sono dotati di pancreas artificiale e quindi la terapia si adatta alla glicemia, regolandola di conseguenza [9]. I campionamenti del CGM vengono effettuati ogni 5m ed inoltre conosciamo esattamente l'entità della terapia alla quale sono sottoposti in tempo reale.

I guasti alla pompa possono avvenire a mezzanotte o a mezzogiorno ed i guasti durano esattamente 6h, questi limiti rendono più semplice la programmazione e l'interpretazione del codice ma non minano la qualità dei risultati raggiunti.

La terapia dei pazienti consiste in una quantità variabile di insulina infusa in stato stazionario con un bolo somministrato al momento del pasto. Nonostante nei dati siano presenti le informazioni sui carboidrati consumati rimanendo in linea con l'algoritmo originale non ne faremo uso. I pazienti possono assumere una quantità variabile di pasti durante il giorno ma consumano sempre colazione, pranzo e cena. Inoltre anche l'orario di consumo dei pasti e la quantità di carboidrati ingeriti sono variabili.

Nel dataset fornito sono presenti anche dei dati accessori inutili ai fini del programma e che quindi ignoreremo.

Il file dal quale estrarremo i dati è un file CSV, le colonne separano i tipi di dati mentre le righe segnano il trascorrere del tempo. Dopo che è stato mostrato l'intero mese di acquisizioni di un paziente si passa alle acquisizioni del paziente successivo.

4.3 La struttura del programma

Per rendere flessibile l'utilizzo del programma si è limitato al minimo l'uso di costanti. Gran parte dei valori infatti sono variabili e personalizzabili in base alle caratteristiche dei dati di cui si è in possesso.

Si possono modificare:

- Le finestre di tempo, sia quella corta (SW) che quella lunga (LW).
- La durata in giorni del monitoraggio
- Tutte e 3 le soglie di allarme (CGM, IFM e GS)
- Il numero di giorni mostrati nei grafici (che saranno gli ultimi "N") ed il numero di tacche sull'asse dei tempi dei suddetti grafici.

Questo programma inoltre riconosce automaticamente il numero di pazienti che partecipano al monitoraggio (ricavando in automatico l'ID paziente più alto).

Di default i parametri puramente legati all'algoritmo (come SW, LW, le soglie ed il tempo di aggiornamento del sistema compartimentale dell'insulina) sono quelli suggeriti da Howsmon et al.

È però importante sottolineare alcune importanti restrizioni presenti nell'algoritmo:

- È previsto un monitoraggio glicemico 24h su 24h, se i dati inseriti non seguono questo andamento (anche se ad intervalli di campionamento del CGM diversi) è necessario modificare il codice.
- È necessario che l'intervallo di campionamento della glicemia sia esattamente divisibile per il tempo di aggiornamento del sistema compartimentale per la stima dell'insulina. ES: Se il tempo trascorso tra i campionamenti CGM è di 10 min ed il tempo trascorso tra aggiornamenti del sistema dell'insulina è di 2 min, è accettabile. Mentre una coppia di valori di 7 min e 3 min non è accettabile.
- È possibile aggiornare la quantità di insulina infusa nel paziente solo ad ogni intervallo principale, quello previsto per il CGM. Ovvero: se la lettura CGM avviene ogni 5 min, la terapia sarà aggiornabile solo ogni 5 min.
- Il programma è pensato per adattarsi al dataset generato dal simulatore UVa/-Padova. Potrebbe essere quindi necessario modificare i riferimenti alle colonne dalle quali vengono estratti i vari dati.

- È prevista la presenza di una colonna contenente l'ID del paziente. L'ID del paziente deve essere un numero naturale che parte da 1 per arrivare all'ID N dell'ennesimo paziente. In questo modo è possibile estrarre il numero di pazienti dall'ID più alto.

Prima che il programma elabori dei dati utili è necessario che sia passato il tempo della LW per permettere l'ottenimento della media più accurata possibile. Il programma registra ed immagazzina questi dati intermedi che quindi sono visionabili, ad esempio per aiutare nella ricerca di bug, ma non ne è previsto l'uso per il lancio di allarmi. Questo vuol dire che il programma è vulnerabile ai guasti nelle prime ore (dipendenti da LW), ed è quindi considerabile pienamente funzionante solo dopo che è passato questo periodo di tempo.

Nonostante siano inutilizzati per determinare gli allarmi questo programma genera una vasta gamma di registri, uno per ogni variabile di interesse, tutti pienamente consultabili per avere un accesso totale ai dati ed al loro andamento.

In questo capitolo verranno fatti dei commenti sulle varie componenti dell'algoritmo. Non considereremo più i parametri come se avessero un valore generico, ma gli saranno attribuiti i valori suggeriti dalla ricerca di Howsmon ed i suoi collaboratori oppure i valori necessari per far funzionare correttamente l'algoritmo con il dataset utilizzato.

In particolare:

- SW = 60; lunghezza in minuti della finestra breve (1h)
- LW = 1440; lunghezza in minuti della finestra lunga (24h)
- time step = 5; minuti trascorsi tra le misurazioni CGM
- fast step = 1; minuti necessari per il calcolo dell'aggiornamento dell'insulina
- days = 30; giorni simulati

4.4 L'analisi del Continuous Glucose Monitoring

Per calcolare il CGM genero due array: CGM_{LW} e CGM_{SW} uno per LW ed uno per SW. Dentro questi array inserisco il valore CGM più recente ed elimino il più vecchio se l'array ha raggiunto la capienza massima. Si effettua quindi una media integrale per stimare il valore medio di $\overline{CGM}_{k|k-LW}$ e $\overline{CGM}_{k|k-SW}$ accuratamente. Una volta calcolata la media di entrambe le finestre posso usarla per elaborare la GFM, verificare se ha superato il valore di soglia, ed eventualmente attivare o disattivare il suo flag di allarme.

L'aggiornamento del flag avviene ad ogni campionamento del CGM, nel nostro caso ogni 5 min.

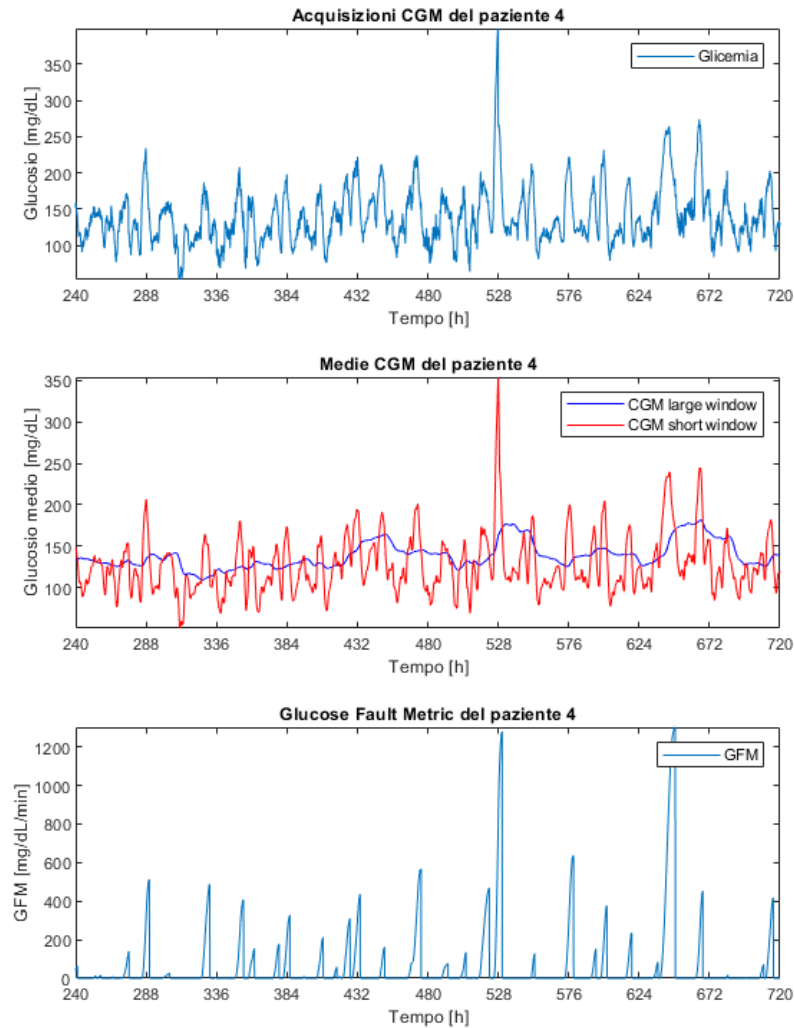


Figura 4.1. Esempio di grafici prodotti dal programma per visualizzare i dati correlati alla GFM di un paziente.

4.5 L'analisi della Glucose Slope

Per il calcolo della GS ci limitiamo a calcolare la differenza tra il CGM corrente e il precedente per poi dividere il valore per il tempo trascorso tra i campionamenti, per ottenere il valore di variazioni al minuto.

In fase di progettazione preliminare sono state considerate anche altre modalità di calcolo. In particolare è stata considerata una linearizzazione delle N pendenze precedenti, questo per limitare l'effetto degli errori casuali commessi dal sensore. Dato che questi sensori possono arrivare ad avere errori anche superiori al 10% è plausibile pensare che una rilevazione errata per difetto susseguita da una rilevazione errata per eccesso possa attivare il flag inavvertitamente. Questo però avrebbe implicato un trade-off in responsività dell'algoritmo comportando un'ulteriore ritardo nella rilevazione, e come problema di entità minore un aumento delle risorse di

calcolo necessarie. Visto che questo flag non ha un ruolo attivo nella rilevazione dei guasti, ma è solo stato inserito come ulteriore salvaguardia dai falsi allarmi, si è ripiegati su un’elaborazione più semplice.

Va in oltre fatto notare che il sensore di glicemia impiegato dal simulatore ha come tetto massimo i 399 mg/dL. Con la ratio che un dettaglio superiore è fondamentale ininfluente nel determinare lo stato di salute del paziente, che si presenta in grave iperglicemia. Questo però ha spesso comportato in fase di progettazione dei falsi negativi perché nonostante i flag GFM e IFM fossero ampiamente sopra la soglia il plateau glicemico a 399 mg/dL non faceva scattare l’allarme. Per risolvere il problema è stata aggiunta la clausola aggiuntiva di attivazione del flag nel caso si fossero raggiunto i 399 mg/dL. Successivamente a questa aggiunta abbiamo riportato un ampio miglioramento sia della precisione che nel numero dei falsi positivi al giorno.

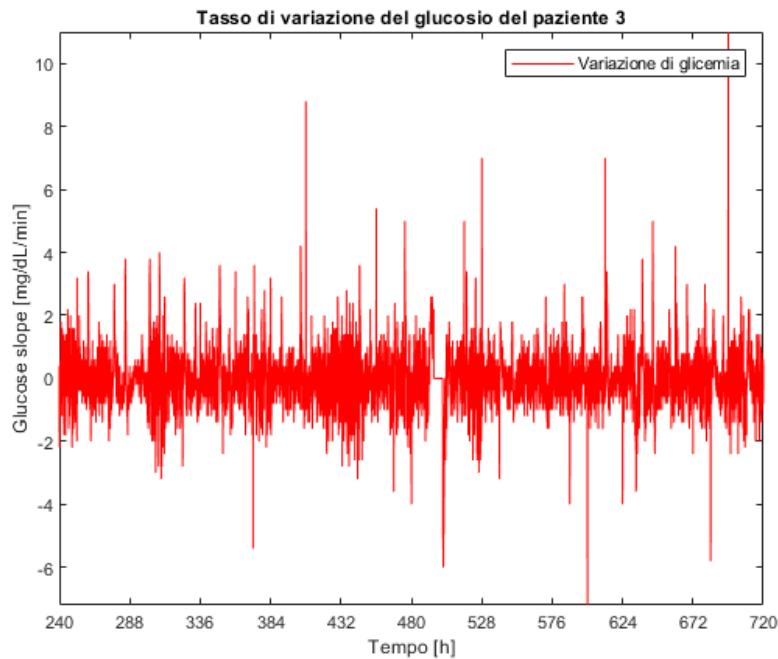


Figura 4.2. Esempio di grafico prodotto dal programma per visualizzare la GS di un paziente. È possibile anche vedere il plateau di variazione glicemica dovuto al limite di rilevazione del sensore.

4.6 L’analisi della Insulin Fault Metric

Per calcolare la PIE aggiornata l’algoritmo importa ad ogni acquisizione di CGM la terapia correntemente somministrata al paziente. A questo punto utilizziamo i dati per aggiornare la stima PIE, ogni minuto. Con un metodo identico a quello utilizzato per la GFM mantengo gli array PIE_{SW} e PIE_{LW} sempre aggiornati inserendo i valori nuovi e rimuovendo i più vecchi. Calcolo inoltre $\overline{PIE}_{k|k-SW}$ e $\overline{PIE}_{k|k-LW}$ utilizzando una media integrale per massimizzare l’accuratezza. Adesso posso ricavare IFM e confrontarla con la soglia.

Se durante questi cinque minuti la IFM supera la soglia almeno 2 volte fa attivare il flag, a quel punto riporto il flag attivato nel ciclo principale che prosegue ad intervalli di 5m. Questa scelta lascia reagire l'algoritmo anche a superamenti di soglia molto brevi permettendo comunque l'attivazione dell'allarme principale di guasto. La scelta di questi valori è arbitraria e fatta per massimizzare la precisione dell'algoritmo, in quanto in seguito ad alcune valutazioni empiriche è stato possibile determinare che scegliere valori tra il 2 ed il 5 (quindi non 1) permette un marginale aumento della precisione. Sarebbe però necessario fare una valutazione con un pool di pazienti più ampio per verificare il valore ottimale.

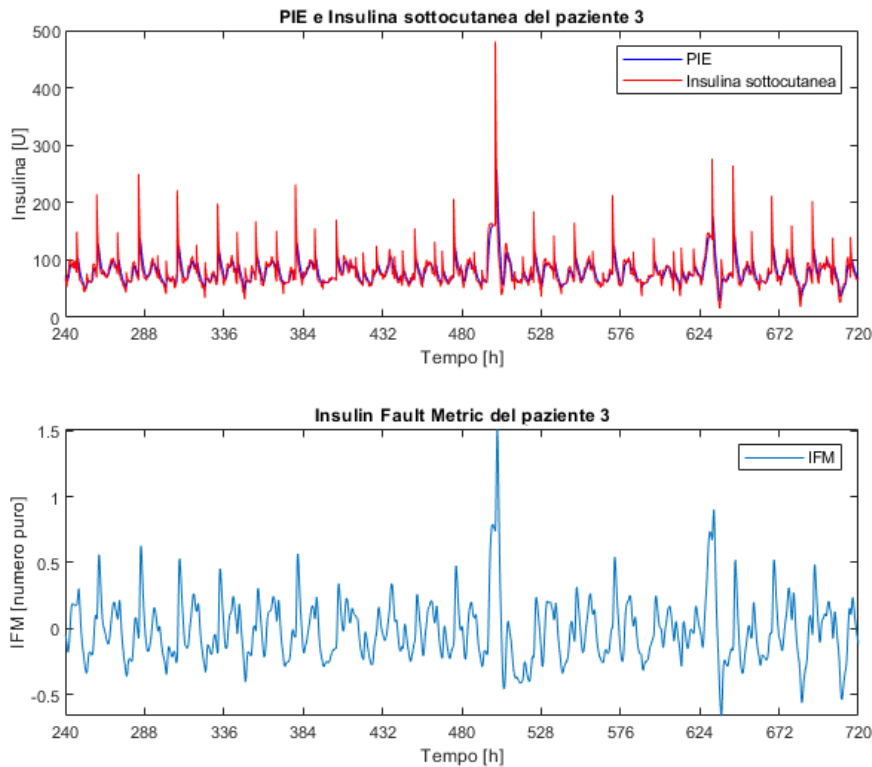


Figura 4.3. Esempio di grafici prodotti dal programma per visualizzare i dati correlati alla IFM di un paziente.

4.7 La valutazione degli allarmi

Gli allarmi vengono aggiornati ad ogni CGM e se tutti e tre sono attivi scatta l'allarme del guasto vero e proprio.

Per trattare l'efficacia degli allarmi è stata implementato un sistema di finestre di tempo di 6h, ovvero della stessa durata dei guasti simulati. Dato che i guasti iniziano a mezzanotte o a mezzogiorno sono interamente contenuti in un'unica finestra. L'obiettivo è far scattare l'allarme all'interno della stessa finestra temporale in cui è comparso il guasto.

All'atto pratico si tratta di due array, ciascuno contenente 1 (guasto presente) o 0 (guasto assente) in ogni cella. Un array contiene i dati sui guasti effettivi e

viene creato basandosi sui dati presenti nel dataset, l'altro invece è il frutto dell'elaborazione dell'algoritmo. Questi array vengono poi confrontati per valutare le performance dell'algoritmo.

In base allo status dell'infusion set nella finestra di tempo considerata e quello stimato dall'algoritmo sono state attribuite a ciascuna finestra una di queste quattro etichette: vero negativo (TN), vero positivo (TP), falso negativo (FN) e falso positivo (FP).

Queste regole sono:

- TN: L'etichetta della finestra viene considerata come vero negativo se non sono stati simulati guasti nel dataset e se l'algoritmo ne ha correttamente segnalato l'assenza.
- TP: L'etichetta della finestra viene considerata come vero positivo se sono stati simulati guasti nel dataset e se l'algoritmo ne ha correttamente segnalato la presenza. Questo vale anche per le segnalazioni avvenute con una finestra di ritardo. Questa scelta è stata fatta per non penalizzare troppo le risposte lente dell'algoritmo. Senza questa clausola una risposta lenta sarebbe scaturita in un falso negativo nella casella dove è apparso il guasto ed un falso positivo in quella in cui è stato rilevato in ritardo, mentre in questo modo penalizziamo l'algoritmo solo con un falso negativo nella finestra in cui non è scattato l'allarme.
- FN: L'etichetta della finestra viene considerata come falso negativo se sono stati simulati guasti nel dataset e se l'algoritmo ne ha erratamente segnalato l'assenza.
- FP: L'etichetta della finestra viene considerata come falso positivo se non sono stati simulati guasti nel dataset e se l'algoritmo ne ha erratamente segnalato la presenza.

Inoltre prima di effettuare il giudizio vengono eliminati tutti gli allarmi che perdurano per due finestre consecutive. In fase di elaborazione la seconda finestra viene considerata come priva di allarmi dall'algoritmo. È stata fatta questa scelta sia per simulare l'intervento dell'utente che prontamente dopo aver sentito l'allarme rimette in sesto il set di infusione, sia per tutelare i risultati dal periodo di transizione verso la situazione normale. I flag IFM e GFM si basano su array intermedi che immagazzinano dati di 24h ed 1h è quindi necessario tamponare gli effetti di queste letture anomale che potrebbero portare ad avere degli allarmi che si susseguono dopo che sia trascorso poco tempo. Ciò ci permette di migliorare la valutazione dei risultati successivi anche a fronte di queste problematiche.

4.8 Grafici

Questo programma dopo aver raccolto i dati ed elaborato gli allarmi disegna i grafici di ogni paziente. Ogni grafico di ciascun paziente contiene: le acquisizioni CGM, le medie CGM in entrambe le finestre di tempo, la GFM, la PIE e l'insulina sottocutanea, la IFM e la GS. I grafici si basano sugli ultimi N giorni di acquisizioni (20 di default), e hanno l'asse delle ascisse suddiviso in ore, il numero di tacche può essere deciso dall'utente (11 di default).

I grafici sono ottimizzati automaticamente da MATLAB nell'utilizzo dello spazio e nell'impostazione della scala degli assi.

Ogni paziente ha una proprio finestra in cui appaiono questi sei grafici. Non sono però presenti finestre che coinvolgono più di un paziente o che evidenziano nel grafico stesso la presenza di guasti.

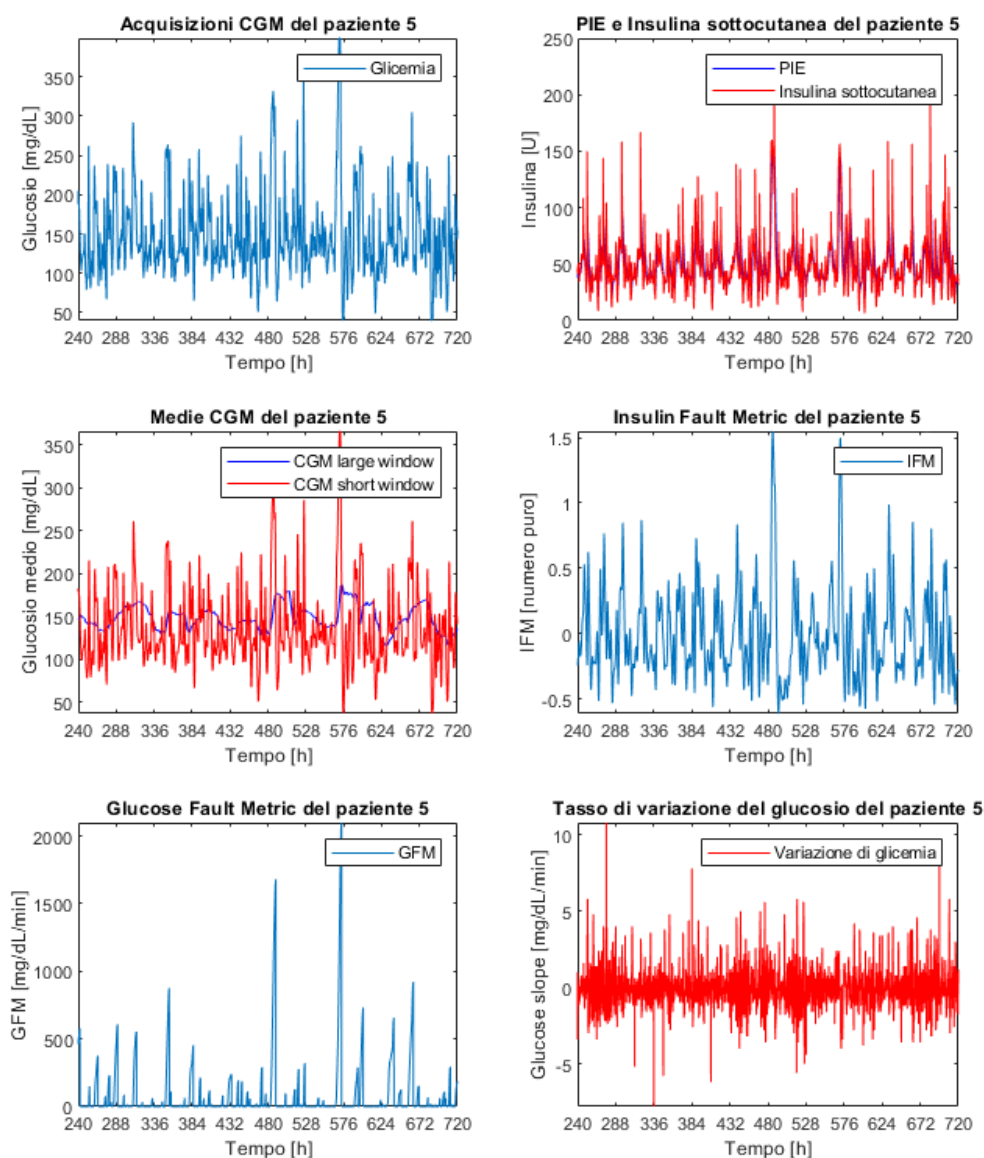


Figura 4.4. Esempio di tutti i grafici prodotti per un paziente dal programma.

4.9 Analisi dei risultati

Il programma è riuscito a raggiungere una sensibilità del 81,82% (con 9 veri positivi su un totale di 11 guasti totali) ed una precisione del 22,50% (con 9 veri positivi su un totale di 40 allarmi lanciati). Normalizzando rispetto il numero di pazienti ed il numero di giorni in cui sono stati monitorati si ottengono 0,207 falsi positivi al giorno per paziente.

A causa del dataset sbilanciato, sono presenti relativamente pochi guasti rispetto a quando l'infusion set e la pompa sono in funzione, quindi ignoriamo la specificità che apparirebbe naturalmente alta.

Il guasto meccanico dell'infusion set o della pompa è un'occorrenza rara anche nella pratica ma anche particolarmente grave. Possiamo quindi accettare come un buon risultato anche delle precisioni relativamente basse, soprattutto se normalizzando in falsi allarmi al giorno otteniamo valori contenuti. Confrontando questi risultati con quelli trovati da Howsmon ed i suoi colleghi possiamo apprezzare un miglioramento nella sensibilità (che trovano al 73%) e dei FP/day (che trovano a 0.27). È molto probabile, però, che questi risultati migliori siano dovuti al fatto che i dati simulati sono più semplici da analizzare.

Tabella 4.1. Confronto delle performance

Algoritmo	LISA	Simulazione
Sensibilità	73%	81,8%
FP/al giorno	0,27	0,21

Considerando solo gli allarmi veri positivi notiamo che la mediana della glicemia al momento dell'allarme è un po' più bassa con 260 mg/dL mentre il tempo mediano impiegato per far scattare l'allarme è molto più alto, con 305 min. Se invece consideriamo tutti gli allarmi senza distinzione di correttezza si ottiene una media glicemica ancora inferiore con 227 mg/dL ed un tempo richiesto decisamente inferiore, di 150 min. Queste forti disparità mi hanno spinto a controllare i valori legati ai soli falsi allarmi. Come sospettavo presentano una mediana glicemica di molto inferiore con 219 mg/dL ed un tempo mediano necessario per l'attivazione dell'allarme drasticamente ridotto, di soli 125 min. È quindi probabile che l'algoritmo sia pronò a dare falsi allarmi durante forti sbalzi glicemici che appaiono in un breve periodo di tempo. Questo tipo di falso allarme è simile a quello riportato nel documento originale, in cui si dice che la maggior parte dei falsi allarmi dell'algoritmo per la rilevazione dei LISA è dovuta a boli di insulina mal dosati rispetto al pasto consumato. Potrebbe essere interessante investigare su possibili criteri di protezione legati al tempo necessario per l'attivazione degli allarmi, un ambito che ritengo possa avere del potenziale per aumentare la precisione dell'algoritmo.

Tabella 4.2. Confronto dei valori mediani ottenuti all'istante dell'allarme

Tipologia di dati	LISA	solo TP	solo FP	TP + FP
Mediana dei tempi in minuti	210	305	125	150
Rilevazione glicemica (mg/dL)	300	260	219	227

Capitolo 5

Conclusione

Alla luce dei risultati ottenuti possiamo dire che l'algoritmo per la rilevazione dei LISA ha una buona performance e che può essere usato efficacemente nei contesti per cui è stato pensato. Il dataset su cui è stato testato qui si basa su pazienti che utilizzano un pancreas artificiale, situazione per la quale gli autori originali hanno espresso la necessità di ulteriore ricerca. Nonostante il pool ridotto di pazienti analizzati l'algoritmo ha avuto una buona prestazione, in linea con i dati ottenuti da Howsmon ed i suoi colleghi. Purtroppo però i risultati di queste analisi sono limitate dall'uso dei dati simulati, che hanno sicuramente contribuito ad una performance migliore, e quindi meno realistica. Nonostante questo l'efficacia dell'algoritmo può ritenersi soddisfacente. È sicuramente necessaria dell'altra ricerca ma questi risultati preliminari sono promettenti.

Bibliografia

- [1] D. Daneman, «Type 1 diabetes,» *Lancet*, n. 367, pp. 847–858, 2006. DOI: doi: 10.1016/S0140-6736(06)68341-4.
- [2] L. A. DiMeglio, C. Evans-Molina e R. A. Oram, «Type 1 diabetes,» *Lancet*, n. 391, pp. 2449–2462, 2018. DOI: 10.1016/S0140-6736(18)31320-5.
- [3] N. Papageorgiou, «Protein packaging may cause the immune attacks of type 1 diabetes,» *Diabetes*, n. 21, 2016. DOI: 10.2337/db16-0671.
- [4] K. Ruder, *Complete Guide to Diabetes, fifth edition*. American Diabetes Association, 2011, ISBN: 9781580403306.
- [5] M. A. Crook, *Clinical Biochemistry and Metabolic Medicine, eight edition*. Hodder Arnold, 2012, ISBN: 9781444144147.
- [6] M. A. Atkinson, G. S. Eisenbarth e A. W. Michels, «Type 1 diabetes,» *Lancet*, n. 383, pp. 69–82, 2014. DOI: 10.1016/S0140-6736(13)60591-7.
- [7] A. Hornsby, A. Pérez-Comas, A. Shera et al., *Diabetes Atlas, second edition*. International Diabetes Federation, 2003, ISBN: 2-930229-27-6.
- [8] *Diabetes Atlas, tenth edition*. International Diabetes Federation, 2021, ISBN: 978-2-930229-98-0.
- [9] C. Cobelli, C. D. Man, G. Sparacino, L. Magni, G. D. Nicolao e B. P. Kovatchev, «Diabetes: Models, Signals, and Control,» *IEEE reviews in biomedical engineering*, n. 2, pp. 54–96, 2009. DOI: 10.1109/RBME.2009.2036073.
- [10] N. V. Chu e S. V. Edelman, «Diabetes and Erectile Dysfunction,» *Clinical Diabetes*, n. 19, pp. 45–47, 2001. DOI: <https://doi.org/10.2337/diaclin.19.1.45>.
- [11] H. A. Elrayah-Eliadarous, C.-G. östenson, M. Eltom, P. Johansson, V. Sparring e R. Wahlström, «Economic and social impact of diabetes mellitus in a low-income country: A case-control study in Sudan,» *Journal of diabetes*, n. 9, pp. 1082–1090, 2017. DOI: 10.1111/1753-0407.12540.
- [12] S. D. Favero, C. Toffanin, L. Magni e C. Cobelli, «Deployment of Modular MPC for Type 1 Diabetes Control: the Italian experience 2008-2016.,» *The Artificial Pancreas*, pp. 153–182, 2019. DOI: 10.1016/B978-0-12-815655-1.00016-8.
- [13] A. Addala, S. C. Suttiratana, J. J. Wong et al., «Cost considerations for adoption of diabetes technology are pervasive: A qualitative study of persons living with type 1 diabetes and their families,» *Diabetic Medicine*, n. 38, 2021. DOI: <https://doi.org/10.1111/dme.14575>.

- [14] A. Haidar, L. Legault, V. Messier, T. M. Mitre, C. Leroux e R. Rabasa-Lhoret, «Comparison of dual-hormone artificial pancreas, single-hormone artificial pancreas, and conventional insulin pump therapy for glycaemic control in patients with type 1 diabetes: an open-label randomised controlled crossover trial,» *Lancet Diabetes Endocrinology*, vol. 3, 2015. DOI: [http://dx.doi.org/10.1016/S2213-8587\(14\)70226-8](http://dx.doi.org/10.1016/S2213-8587(14)70226-8).
- [15] L. Heinemann e L. Krinelke, «Insulin Infusion Set: The Achilles Heel of Continuous Subcutaneous Insulin Infusion,» *Journal of Diabetes Science and Technology*, n. 4, 2016. DOI: [10.1177/193229681200600429](https://doi.org/10.1177/193229681200600429).
- [16] D. P. Howsmon, F. Cameron, N. Baysal et al., «Continuous Glucose Monitoring Enables the Detection of Losses in Infusion Set Actuation (LISAs),» *Sensors*, 2017. DOI: <https://doi.org/10.3390/s17010161>.

Appendice A

fault_algorithm.m

```
%FAULT DETECTION
```

```
%Questo algoritmo valuta la presenza di possibili anomalie  
%possono manifestarsi durante il funzionamento di una pompa di insulina  
%tramite l'uso di tre flag:  
%-Glucose Fault Metric, GFM>100(mg*min/dL)  
%-Glucose Slope Threshold, GST>0.3 (mg*min/dL)  
%-Insuline Fault Metric, IFM>0.4
```

```
clearvars  
close all  
clc
```

```
%Inizializzazione dati  
dataset = readtable('dataset.csv');
```

```
%parametri temporali  
SW = 60; %minuti della finestra breve (def: 1h)  
LW = 1440; %minuti della finestra lunga (def: 24h)  
time_step = 5; %minuti tra le misure della glicemia  
fast_step = 1; %minuti necessari per l'aggiornamento dell'insulina  
days = 30; %giorni simulati  
minutes_in_hour = 60;  
hours_in_day = 24;  
marks = 11; %tacche sull'asse dei tempi del grafico  
minutes_in_day = minutes_in_hour*hours_in_day;  
days_shown = 20; %giorni mostrati nei grafici  
rows = days*hours_in_day*minutes_in_hour/time_step;
```

```
%ricavo il numero di pazienti dal dataset  
n_patients = max(dataset{:,14}, [], 'all');
```

```
%finestre relative al time step  
nstepLW = LW/time_step;  
nstepSW = SW/time_step;
```

```
%finestre relative al fast step  
nstepLW_ins = LW/fast_step;  
nstepSW_ins = SW/fast_step;
```

```
%alert, valori al di sopra dei quali scatta l'allarme  
GFM_threshold = 100; %(mg/dL)*min  
IFM_threshold = 0.4; %numero puro migliore con 0.2  
GS_threshold = 0.3; %(mg/dL)*min migliore con -0.00001
```

```
%sono vettori di modo da permettermi il monitoraggio  
GFM_alert = nan(rows, n_patients);
```

```

IFM_alert = nan(rows, n_patients);
GS_alert = nan(rows, n_patients);
fault_alert = nan(rows, n_patients);

GFM_alert(1) = 0;
IFM_alert(1) = 0;
GS_alert(1) = 0;
fault_alert(1) = 0;

%Inizializzazioni
meanCGMlw = nan(rows, n_patients);
meanCGMsw = nan(rows, n_patients);
GFM = nan(rows, n_patients);
GS = nan(rows, n_patients); %inizializzo il tasso di cambiamento del glucosio
GS(1,:) = 0;
meanPIElw = nan(rows*(time_step/fast_step), n_patients);
meanPIEsw = nan(rows*(time_step/fast_step), n_patients);
IFM = nan(rows*(time_step/fast_step), n_patients); %Insuline fault metric
assign = 0; %variabile per l'assegnazione di IFM

length_window = 360; %minuti durata finestra (def: 6h)
log_check_sim=nan(rows, n_patients);
log_check_abs=nan(rows, n_patients);

%log dei malfunzionamenti
permanent_log_sim = nan(days*hours_in_day / ...
    (length_window/minutes_in_hour), n_patients);
permanent_log_abs = nan(days*hours_in_day / ...
    (length_window/minutes_in_hour), n_patients);

log_CGM = nan(rows, n_patients);
%(fast_step/time_step)
sub_ins_log = nan (rows*(time_step/fast_step), n_patients);
%(time_step/fast_step)
PIE_log = nan (rows*(time_step/fast_step), n_patients);
basal_ins_log = nan (rows*(time_step/fast_step), n_patients);
bolus_ins_log = nan (rows*(time_step/fast_step), n_patients);

first_alarm_of_the_window = 1; %inizia attivato
%Tempo necessario per accorgersi del guasto e a che glicemia è successo
detected_glucose = zeros(days*hours_in_day / ...
    (length_window/minutes_in_hour), n_patients);
time_for_detection = zeros(days*hours_in_day / ...
    (length_window/minutes_in_hour), n_patients);
d_g_temp = 0;
t_f_d_temp = 0;

for p=1:n_patients
    start_window = 0; %minuti ai quali è iniziata la finestra corrente
    current_window = 1; %la finestra considerata al momento

    temp_log_sim = nan(rows, 1); %log temporaneo calcolato dalla simulazione
    temp_log_abs = nan(rows, 1); %log temporaneo recuperato dal dataset

    CGMlw = nan(nstepLW, 1);
    CGMsw = nan(nstepSW, 1);

    sub_Ins = nan(rows*(time_step/fast_step), 1); %Insulina sottocutanea
    sub_Ins(1) = 0; %Ipotizzo 0 come condizione iniziale
    PIE = nan(rows*(time_step/fast_step), 1); %Plasma insuline estimate

```



```

PIE(1) = 0;
PIElw = nan(nstepLW_ins,1);
PIEsw = nan(nstepSW_ins,1);

CGM=[dataset{((p-1)*rows+1:p*rows),2}];
log_CGM(:,p) = CGM;

for k1 = 1:rows
%Verifica assoluta guasto
    temp_log_abs(k1) = dataset{k1+rows*(p-1),11};

%Analisi della GFM
    %Shift a sinistra del contenuto e sovrascrivo l'ultima casella
    %Gli array così contengono i dati aggiornati in ordine
    CGMlw = circshift(CGMlw,-1);
    CGMsw = circshift(CGMsw,-1);
    CGMlw(nstepLW) = CGM(k1);
    CGMsw(nstepSW) = CGM(k1);

%Se vettori sono pieni posso ricavare la media
%più precisa usando un integrale
    if sum(isnan(CGMlw))==0 && sum(isnan(CGMsw))==0
        meanCGMlw(k1,p) = time_step*trapz(CGMlw)/LW;
        meanCGMsw(k1,p) = time_step*trapz(CGMsw)/SW;
    else %altrimenti uso degli placeholder
        %Non uso la media integrale per gli placeholder dato che non
        %necessito di elevata precisione e non voglio appesantire i calcoli
        meanCGMlw(k1,p) = nanmean(CGMlw);
        meanCGMsw(k1,p) = nanmean(CGMsw);
    end

%Se i vettori sono pieni, le medie sono
%affidabili e posso procedere al confronto
    if sum(isnan(CGMlw))==0 && sum(isnan(CGMsw))==0
        if meanCGMsw(k1,p) > meanCGMlw(k1,p)
            if isnan(GFM(k1-1,p))
                GFM(k1,p) = 0 + (meanCGMsw(k1,p) - meanCGMlw(k1,p)) /
                    time_step;
            else
                GFM(k1,p) = GFM(k1-1,p) + ...
                    (meanCGMsw(k1,p) - meanCGMlw(k1,p)) / time_step;
            end
        else
            GFM(k1,p) = 0;
        end
    end

%Calcolo del GS
    if not(k1 == 1)
        GS(k1,p) = (CGM(k1) - CGM(k1-1)) / time_step;
    else
        GS(1,p) = 0;
    end

%Controllo soglia GFM
    if GFM(k1,p) > GFM_threshold
        GFM_alert(k1,p) = 1;
    else

```

```

    GFM_alert(k1,p) = 0;
end

%Controllo soglia GS
if GS(k1,p) > GS_threshold || CGM(k1) == 399
    %ovvero il massimo misurabile dal sensore
    GS_alert(k1,p) = 1;
else
    GS_alert(k1,p) = 0;
end

%Aggiornamento in tempo reale della terapia
basal_Ins = dataset{(p-1)*rows+k1,4};
bolus_Ins = dataset{(p-1)*rows+k1,5};

%Stima PIE e calcolo della IFM
for k2 = 1:(time_step/fast_step) %Contatore dei sottocicli
    k3 = (k1-1)*(time_step/fast_step)+k2; %Contatore globale
    if k1 == 1 && k2 == 1
        continue %salta il primo ciclo
    end
    PIE(k3) = 0.98*PIE(k3-1)+0.02*sub_Ins(k3-1);
    sub_Ins(k3) = 0.98*sub_Ins(k3-1)+basal_Ins+bolus_Ins;
    bolus_Ins = 0;

    basal_Ins_log(k3,p) = basal_Ins;
    bolus_Ins_log(k3,p) = bolus_Ins;
    sub_Ins_log(k3,p) = sub_Ins(k3);
    PIE_log(k3,p) = PIE(k3);

    PIElw = circshift(PIElw,-1);
    PIEsw = circshift(PIEsw,-1);
    PIElw(nstepLW_ins) = PIE(k3-1);
    PIEsw(nstepSW_ins) = PIE(k3-1);

    %Se vettori sono pieni posso ricavare la media
    %più precisa usando un integrale
    if sum(isnan(PIElw))==0 && sum(isnan(PIEsw))==0
        meanPIElw(k3,p) = fast_step*trapz(PIElw)/LW;
        meanPIEsw(k3,p) = fast_step*trapz(PIEsw)/SW;
    else %altrimenti uso degli placeholder
    %Non uso la media integrale per gli placeholder dato che non
    %necessito di elevata precisione e non voglio appesantire i
    %calcoli
        meanPIElw(k3,p) = nanmean(PIElw);
        meanPIEsw(k3,p) = nanmean(PIEsw);
    end

    %Considero affidabili le IFM ottenute solo da vettori PIE pieni
    if sum(isnan(PIElw))==0 && sum(isnan(PIEsw))==0
        IFM(k3,p) = (meanPIEsw(k3,p)/meanPIElw(k3,p)) - 1;
    else
        IFM(k3,p) = NaN;
    end

    %Controllo soglia IFM
    if IFM(k3,p) > IFM_threshold

```

```

%Mi dice se la soglia è stata superata almeno una volta
    assign = assign+1;
else
    assign = 0;
end

end

%Assegnazione soglia IFM
if assign >= 3
    IFM_alert(k1,p) = 1;
else
    IFM_alert(k1,p) = 0;
end
assign = 0;

%Verifica complessiva
if GFM_alert(k1,p) == 1 && IFM_alert(k1,p) == 1 ...
    && GS_alert(k1,p) == 1
    fault_alert(k1,p) = 1;
    temp_log_sim(k1) = 1;
    if first_alarm_of_the_window == 1
        d_g_temp = CGM(k1);
        t_f_d_temp = k1*time_step-start_window;
        first_alarm_of_the_window = 0;
    end

else
    fault_alert(k1,p) = 0;
    temp_log_sim(k1) = 0;
end

%Se il prossimo ciclo mi fa cambiare finestra tiro le somme della
%finestra corrente
if (k1+1)*time_step > start_window+length_window

%controllo la presenza dei guasti dai dati forniti
if not(sum(temp_log_abs(((start_window/time_step)+1):...
    ((start_window+length_window)/time_step)))==0)
    permanent_log_abs(current_window,p) = 1; %registro il guasto
else
    permanent_log_abs(current_window,p) = 0; %nessun guasto
end

%registro i dati avvenuti al momento del guasto
detected_glucose(current_window,p) = d_g_temp;
time_for_detection(current_window,p) = t_f_d_temp;
d_g_temp = 0;
t_f_d_temp = 0;
first_alarm_of_the_window = 1;
%controllo i guasti rilevati dalla simulazione
%migliori risultati con 1
if sum(temp_log_sim(((start_window/time_step)+1):...
    ((start_window+length_window)/time_step)))>0
    %controllo la possibile persistenza dalla finestra
    precedente
    if not(current_window-1 == 0) &&...
        permanent_log_sim(current_window-1,p) == 1
        permanent_log_sim(current_window,p) = 0;
    else
        %registro il guasto
    end
end

```

```

        permanent_log_sim(current_window,p) = 1;
    end
    else
        permanent_log_sim(current_window,p) = 0; %nessun guasto
    end
    start_window = start_window+length_window;
    current_window = current_window+1;
end
%Progresso caricamento
completamento = 100*((p-1)*rows+k1)/(n_patients*rows);
if not(k1==1) && not(round(completamento)==...
    round(100*((p-1)*rows+k1-1)/(n_patients*rows)))
    disp(['Caricamento... ' num2str(completamento, '%0.0f') '%'])
end
end

log_check_sim(:,p)=temp_log_sim;
log_check_abs(:,p)=temp_log_abs;

%Pannello di controllo
%Considero solo gli ultimi giorni

time = minutes(minutes_in_day*(days-days_shown):time_step:...
    minutes_in_day*days); %1440 minuti in un giorno
time = hours(time);
time_shown = minutes(linspace(minutes_in_day*(days-days_shown),...
    minutes_in_day*days,marks)); %tempo mostrato sul grafico
time_shown = hours(time_shown);
time_ins = minutes(minutes_in_day*(days-days_shown):fast_step:...
    minutes_in_day*days); %1440 minuti in un giorno
time_ins = hours(time_ins);

%Genero dei vettori che contengono i dati
%degli ultimi giorni per graficarli
CGM_plot = CGM(minutes_in_day*(days-days_shown)/time_step:...
    minutes_in_day*days/time_step);
meanCGMlw_plot = meanCGMlw(minutes_in_day*(days-days_shown)/...
    time_step:minutes_in_day*days/time_step,p);
meanCGMsw_plot = meanCGMsw(minutes_in_day*(days-days_shown)/...
    time_step:minutes_in_day*days/time_step,p);
GFM_plot = GFM(minutes_in_day*(days-days_shown)/time_step:...
    minutes_in_day*days/time_step,p);
GS_plot = GS(minutes_in_day*(days-days_shown)/time_step:...
    minutes_in_day*days/time_step,p);
PIE_plot = PIE(minutes_in_day*(days-days_shown)/fast_step:...
    minutes_in_day*days/fast_step);
sub_Ins_plot = sub_Ins(minutes_in_day*(days-days_shown)/...
    fast_step:minutes_in_day*days/fast_step);
IFM_plot = IFM(minutes_in_day*(days-days_shown)/fast_step:...
    minutes_in_day*days/fast_step,p);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
figure %figura completa
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

subplot(3,2,1)
plot(time,CGM_plot)
%variabile di appoggio per i titoli
title_name = ['Acquisizioni_CGM_del_paziente_' num2str(p)];

```

```

title(title_name)
xlabel('Tempo_[h]')
ylabel('Glucosio_[mg/dL]')
legend('Glicemia')
set(gca,'FontSize',8.5,'XLim',[time(1),time(end)],'XTick',...
      time_shown(:),'XTickLabel',time_shown(:));
axis tight

subplot(3,2,3)
plot(time,meanCGMlw_plot,'-b',time,meanCGMsw_plot,'-r')
title_name = ['Medie_CGM_del_paziente_' num2str(p)];
title(title_name)
xlabel('Tempo_[h]')
ylabel('Glucosio_medio_[mg/dL]')
legend('CGM_large_window','CGM_short_window')
set(gca,'FontSize',8.5,'XLim',[time(1),time(end)],'XTick',...
      time_shown(:),'XTickLabel',time_shown(:));
axis tight

subplot(3,2,5)
plot(time,GFM_plot)
title_name = ['Glucose_Fault_Metric_del_paziente_' num2str(p)];
title(title_name)
xlabel('Tempo_[h]')
ylabel('GFM_[mg/dL/min]')
legend('GFM')
set(gca,'FontSize',8.5,'XLim',[time(1),time(end)],'XTick',...
      time_shown(:),'XTickLabel',time_shown(:));
axis tight

subplot(3,2,2)
plot(time_ins,PIE_plot,'-b',time_ins,sub_Ins_plot,'-r')
title_name = ['PIE_e_Insulina_sottocutanea_del_paziente_' num2str(p)];
title(title_name)
xlabel('Tempo_[h]')
ylabel('Insulina_[U]')
legend('PIE','Insulina_sottocutanea')
set(gca,'FontSize',8.5,'XLim',[time(1),time(end)],'XTick',...
      time_shown(:),'XTickLabel',time_shown(:));

subplot(3,2,4)
plot(time_ins,IFM_plot)
title_name = ['Insuline_Fault_Metric_del_paziente_' num2str(p)];
title(title_name)
xlabel('Tempo_[h]')
ylabel('IFM_[numero_puro]')
legend('IFM')
set(gca,'FontSize',8.5,'XLim',[time(1),time(end)],'XTick',...
      time_shown(:),'XTickLabel',time_shown(:));
axis tight

subplot(3,2,6)
plot(time,GS_plot,'-r')
title_name = ['Tasso_di_variazione_del_glucosio_del_paziente_' num2str(p)
             ');
title(title_name)
xlabel('Tempo_[h]')
ylabel('Glucose_slope_[mg/dL/min]')
legend('Variazione_di_glicemia')
set(gca,'FontSize',8.5,'XLim',[time(1),time(end)],'XTick',...

```

```

        time_shown(:), 'XTickLabel', time_shown(:));
    axis tight

end

%%Valutazione risultati
results = nan((days*hours_in_day/(length_window/minutes_in_hour)),n_patients
);

%%Legenda
% 1 = Vero negativo
% 2 = Vero positivo
% 3 = Falso negativo
% 4 = Falso positivo
for c1=1:n_patients
    for c2=1:(days*hours_in_day/(length_window/minutes_in_hour))
        if permanent_log_abs(c2,c1)==0 && permanent_log_sim(c2,c1)==0
            results(c2,c1) = 1;           % 1 = Vero negativo
        end
        if permanent_log_abs(c2,c1)==1 && permanent_log_sim(c2,c1)==1
            results(c2,c1) = 2;           % 2 = Vero positivo
        end
        if permanent_log_abs(c2,c1)==1 && permanent_log_sim(c2,c1)==0
            results(c2,c1) = 3;           % 3 = Falso negativo
        end
        if permanent_log_abs(c2,c1)==0 && permanent_log_sim(c2,c1)==1
            if c2==1
                results(c2,c1) = 4;       % 4 = Falso positivo
                %controllo per la depenalizzazione degli allarmi ritardati
                elseif results(c2-1,c1)==3
                    results(c2,c1) = 2;
                    %Inserisco l'errore per gli allarmi
                    %scattati nella finestra successiva
                    time_for_detection(c2,c1) = ...
                        time_for_detection(c2,c1)+length_window;
                else
                    results(c2,c1) = 4;
                end
            end
        end
    end
end

true_positive_array = find(results==2); %indici dei veri positivi
false_positive_array = find(results==4); %indici dei falsi positivi

%rendo le matrici degli array così sono compatibili con find
time_for_detection = reshape(time_for_detection,[],1);
detected_glucose = reshape(detected_glucose,[],1);

TN = length(find(results==1)); %Conta dei veri negativi
TP = length(true_positive_array); %Conta dei veri positivi
FN = length(find(results==3)); %Conta dei falsi negativi
FP = length(false_positive_array); %Conta dei falsi positivi

med_det_glucose = nan(TP,1); %median_detected_glucose
median_time_det = nan(TP,1); %median_time_detection
med_det_glucose_FP = nan(FP,1);
median_time_det_FP = nan(FP,1);

```

```

%calcoli i tempi e la glicemia mediane al momento dell'allarme
for c3 = 1:TP
    med_det_glucose(c3) = detected_glucose(true_positive_array(c3));
    median_time_det(c3) = time_for_detection(true_positive_array(c3));
end
med_det_glucose_TP = median(med_det_glucose);
median_time_det_TP = median(median_time_det);
med_det_glucose = cat(1, med_det_glucose, nan(FP,1));
median_time_det = cat(1, median_time_det, nan(FP,1));

for c4 = 1:FP
    med_det_glucose_FP(c4) = detected_glucose(false_positive_array(c4));
    median_time_det_FP(c4) = time_for_detection(false_positive_array(c4));
    med_det_glucose(TP+c4) = detected_glucose(false_positive_array(c4));
    median_time_det(TP+c4) = time_for_detection(false_positive_array(c4));
end
med_det_glucose_FP = median(med_det_glucose_FP);
median_time_det_FP = median(median_time_det_FP);
med_det_glucose_tot = median(med_det_glucose);
median_time_det_tot = median(median_time_det);

sensitivity = TP/(TP+FN);
precision = TP/(TP+FP);
FP_day = FP/(days*n_patients);

disp(' ||||| ')
disp(['La sensibilità dell'' algoritmo sui dati simulati è' ...
    num2str(sensitivity*100, '%.2f') '%'])
disp(['La precisione dell'' algoritmo sui dati simulati è' ...
    num2str(precision*100, '%.2f') '%'])
disp(['Si stimano' num2str(FP_day, '%.3f') ...
    ' falsi positivi al giorno per ciascun paziente'])

disp(' ||||| ')
disp(['La mediana del glucosio al momento dell'' allarme (considerando' ...
    ' solo i TP) è' num2str(med_det_glucose_TP, '%.0f') ' mg/dL'])
disp(['La mediana del tempo impiegato per far scattare l'' allarme' ...
    '(considerando solo i TP) è' num2str(median_time_det_TP, '%.0f') ' min'
    ])
disp(' ||||| ')
disp(['La mediana del glucosio al momento dell'' allarme' ...
    '(considerando solo i FP) è' num2str(med_det_glucose_FP, '%.0f') ' mg/dL'
    ''])
disp(['La mediana del tempo impiegato per far scattare l'' allarme' ...
    '(considerando solo i FP) è' num2str(median_time_det_FP, '%.0f') ' min'
    ''])
disp(' ||||| ')
disp(['La mediana del glucosio al momento dell'' allarme (considerando' ...
    ' tutti gli allarmi) è' num2str(med_det_glucose_tot, '%.0f') ' mg/dL'])
disp(['La mediana del tempo impiegato per far scattare l'' allarme' ...
    '(considerando tutti gli allarmi) è'])
disp(['num2str(median_time_det_tot, '%.0f') ' min'])
disp(' ||||| ')

```