



**UNIVERSITÀ  
DEGLI STUDI  
DI PADOVA**



**DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE**

**CORSO DI LAUREA IN INGEGNERIA BIOMEDICA**

**ALGORITMI PER LA RILEVAZIONE DI MALFUNZIONAMENTI DEL SET  
DI INFUSIONE DI INSULINA NEI PAZIENTI DIABETICI BASATI SUL  
MONITORAGGIO CONTINUO DI GLUCOSIO**

**Relatore: Prof. Del Favero Simone**

**Laureando: Bizzotto Davide**

**ANNO ACCADEMICO 2023 – 2024**

**Data di laurea 26/09/2024**



# Abstract

L'utilizzo crescente di sistemi autonomi nella somministrazione di insulina e dell'innovativo pancreas artificiale in pazienti diabetici ha portato un notevole miglioramento nella qualità di vita delle persone affette da diabete. Tuttavia, questi dispositivi possono essere soggetti a malfunzionamenti, portando a condizioni di iperglicemia dovute ad una mancata somministrazione di insulina. È necessario quindi, tutelare la sicurezza e la salute del paziente attraverso dei sistemi in grado di ridurre il verificarsi di tali situazioni.

In questa tesi è stato valutato un algoritmo per la rilevazione di malfunzionamenti del set di infusione di insulina in pazienti affetti da diabete di tipo 1. L'algoritmo preso in esame è quello proposto da Howsmon et al., che attraverso il continuo monitoraggio di glucosio (CGM) è in grado di individuare e segnalare al paziente un guasto nel set di infusione (Losses in Infusion Set Actuation, LISA).

L'algoritmo è stato implementato nell'ambiente di lavoro MATLAB R2023b e testato utilizzando i dati di pazienti in *silico* generati con il simulatore UVa/Padova. Il programma sfrutta tre metriche di guasto per valutare i parametri del paziente e segnalare, eventualmente, un guasto. Le tre metriche sono: Glucose Fault Metric (GFM), Glucose Slope (GS) e Insulin Fault Metric (IFM).

Con una soglia IFM pari a 0,4 è stata riscontrata una sensibilità del 30% con 0,253 FP/giorno, mentre abbassando la soglia a 0,3, la sensibilità ottenuta è stata del 60% con 0,287 FP/giorno.



# Indice

<b>INTRODUZIONE AL DIABETE .....</b>	<b>7</b>
1.1 DIABETE DI TIPO 1.....	7
1.2 DIABETE DI TIPO 2.....	9
1.3 IL PREDIABETE (IGT E IFG).....	10
1.4 SINTOMI E DIAGNOSI .....	10
1.5 COMPLICANZE .....	11
1.6 IL DIABETE NEL MONDO E IN ITALIA .....	13
1.7 TRATTAMENTO DEL DIABETE DI TIPO 1: IL PANCREAS ARTIFICIALE .....	14
<b>ALGORITMO PER INDIVIDUAZIONE LISA .....</b>	<b>17</b>
2.1 INTRODUZIONE ALL'ALGORITMO .....	17
2.2 NOTAZIONE.....	19
2.3 SVILUPPO DELL'ALGORITMO.....	19
2.4 METRICHE DI GUASTO.....	20
2.4.1 <i>Glucose Fault Metric (GFM)</i> .....	20
2.4.2 <i>Insulin Fault Metric (IFM)</i> .....	22
2.4.3 <i>Glucose slope (GS)</i> .....	24
2.5 PRESTAZIONE DELL'ALGORITMO.....	24
<b>IMPLEMENTAZIONE ALGORITMO PERSONALE.....</b>	<b>29</b>
3.1 INTRODUZIONE .....	29
3.2 SIMULAZIONE DEI SEGNALI.....	29
3.3 FINESTRE DI ANALISI E SOGLIE DI ALLARME.....	31
3.4 CALCOLO METRICHE DI GUASTO .....	32
3.4.1 <i>Glucose Fault Metric (GFM)</i> .....	32
3.4.2 <i>Glucose Slope (GS)</i> .....	35
3.4.3 <i>Insulin fault metric (IFM)</i> .....	36

3.5	SEGNALAZIONE GUASTO .....	38
3.6	IMPLEMENTAZIONE ALGORITMO CON DATI GENERATI DAL SIMULATORE UVA/PADOVA.....	38
3.6.1	<i>Dati utilizzati</i> .....	38
3.6.2	<i>Struttura del programma</i> .....	39
3.6.3	<i>Grafici</i> .....	40
3.7	VALUTAZIONE DELLE PRESTAZIONI.....	42
	<b>CONCLUSIONI .....</b>	<b>45</b>
	<b>BIBLIOGRAFIA .....</b>	<b>47</b>
	<b>APPENDICE A .....</b>	<b>51</b>

# Capitolo 1

## Introduzione al diabete

Il diabete mellito, o più semplicemente diabete, è una malattia cronica caratterizzata da aumento dei livelli di glucosio nel sangue e si verifica quando il corpo non produce più insulina o non è in grado di utilizzarla correttamente. [16][17].

L'insulina è un ormone essenziale prodotto dal pancreas e permette al glucosio presente nel sangue di entrare nelle cellule del corpo ed essere convertito in energia o immagazzinato, quando questo meccanismo è alterato, il glucosio si accumula nel circolo sanguigno e porta a condizioni di iperglicemia [16][18].

Una quantità insufficiente di insulina e il prolungarsi di tale condizione per un tempo prolungato può portare a gravi complicanze nell'organismo del paziente, tra cui malattie cardiovascolari, neuropatia, danni ai reni e retinopatia che può portare anche a cecità [16]. Il deficit di insulina induce l'organismo a metabolizzare gli acidi grassi al posto del glucosio per produrre energia. Dalla degradazione di quest'ultimi, si generano delle sostanze di scarto dette chetoni, che provocano uno stato di acidosi metabolica. Il cervello stimola la produzione di ormoni quali glucagone, catecolamine, cortisolo e l'ormone della crescita (GH), peggiorando ulteriormente l'iperglicemia, in quanto, sono ormoni che stimolano la gluconeogenesi (generazione di glucosio da aminoacidi, glicerolo o lattato) e la glicogenolisi (scissione del glicogeno nel fegato e nei muscoli per ottenere glucosio), portando alla chetoacidosi diabetica, che nei casi più gravi può essere letale [21].

La crescita del diabete è provocata soprattutto dall'invecchiamento della popolazione e da abitudini alimentari scorrette, tuttavia grazie all'aumento delle diagnosi precoci, si è abbassato il tasso di mortalità dei pazienti, che in Italia sono più di 3,5 milioni di persone [17].

### 1.1 Diabete di tipo 1

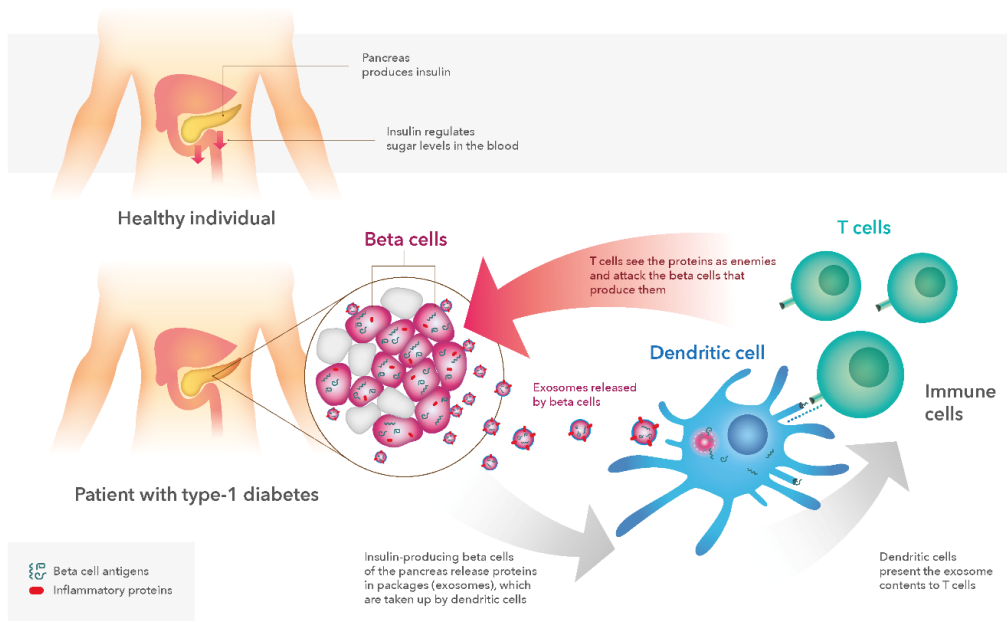
Il diabete di tipo 1 riguarda circa il 10% delle persone con diabete e in genere viene diagnosticato nell'infanzia o nell'adolescenza [18]. Il pancreas non produce insulina a

causa della distruzione delle  $\beta$ -cellule che producono questo ormone. Il sistema immunitario effettua un attacco selettivo, attraverso le cellule T e B. Tali cellule si infiltrano nelle isole pancreatiche e rilasciano proteine infiammatorie (citochine e chemochine) che interagiscono con le  $\beta$ -cellule pancreatiche fino a distruggerle [19]. La distruzione avviene in quanto è stato riscontrato che le cellule pancreatiche possono secernere le proteine note per essere soggette all'attacco da parte del sistema immunitario. Oltre alle proteine, il problema potrebbe essere dato anche dal loro "imballaggio" chiamato esosomi, che si presentano come piccole vescicole. Le  $\beta$ -cellule pancreatiche rilasciano tre proteine negli esosomi e, in condizioni di forte stress, tale rilascio può avvenire in grandi quantità richiamando l'attivazione del sistema immunitario e la distruzione delle cellule stesse. Queste proteine fortemente proinfiammatorie possono essere coinvolte nell'induzione di un'autoimmunità [20].

Il diabete viene classificato tra le malattie *autoimmuni*, cioè dovute ad una reazione immunitaria diretta contro l'organismo stesso [18]. Le possibili cause del diabete di tipo 1 sono ancora sconosciute, ma potrebbero essere legate a fattori ambientali (alimentazione, stile di vita o il contatto con specifici virus) o a fattori genetici [17]. È necessario che i pazienti diabetici si iniettino ogni giorno della loro vita l'insulina, senza la quale non sopravvivrebbero, attraverso penne insuliniche, microinfusori o all'innovativo sistema di pancreas artificiale, il quale permette un'infusione continua e automatica di insulina controllata tramite un algoritmo [18].

## Diabetes: Why does the immune system attack the pancreas?

In type-1 diabetes, cells of the immune system attack the insulin-producing beta cells of the pancreas. Scientists have now found how this may happen.



**Figura 1.1** Rappresentazione dell'attacco immunitario che porta ad avere il diabete di tipo 1 [20].

## 1.2 Diabete di tipo 2 ed altri tipi di diabete

Il diabete di tipo 2 è determinato da una ridotta sensibilità dell'organismo all'insulina da parte di fegato, muscolo e tessuto adiposo o da una ridotta secrezione di insulina da parte del pancreas. Questa tipologia interessa oltre il 90% delle persone affette da diabete ed è una patologia che può progressivamente peggiorare nel tempo [17]. Le cause del diabete di tipo 2 sono ancora ignote, ma sembra esserci una correlazione con l'obesità, la scarsa attività fisica, un eccessivo apporto calorico, l'etnia di appartenenza e una familiarità con parenti di primo grado affetti da diabete di tipo 2 [15][17]. Questa patologia si sviluppa principalmente con l'aumentare dell'età (dopo i 30-40 anni), ma può essere diagnosticata anche in bambini e adolescenti [17]. Spesso il diabete di tipo 2 non viene diagnosticato per molti anni in quante le condizioni iperglicemiche si sviluppano gradualmente e inizialmente non si verificano i classici sintomi del diabete. Viene casualmente diagnosticato in concomitanza a situazioni di stress fisico, infezioni o interventi chirurgici [18].

Non esistono solo due tipologie di diabete, ma queste sono le più comuni. Altre tipologie di diabete esistenti sono: il diabete gestazionale che colpisce le donne in gravidanza, diabete causato da utilizzo di droghe o sostanze chimiche che rendono

inefficace l'insulina, diabete legato a malattie pancreatiche come il cancro e altre ancora [16].

### 1.3 Il prediabete (IGT e IFG)

Il glucosio a digiuno alterato (IFG) e la tolleranza al glucosio alterata (IGT) sono condizioni in cui i livelli di glucosio nel sangue sono al di sopra dei valori ottimali ma al di sotto della soglia per la diagnosi del diabete, questa condizione viene definita *prediabete* [16]. È una situazione intermedia tra la normalità ed il diabete, ma con un alto rischio di progredire verso il diabete di tipo 2, anche se è possibile evitare tale peggioramento [15]. Queste due patologie sono collegate all'aumento di malattie cardiovascolari e disturbi metabolici. L'IFG ha un rischio del 20-30% di progressione verso il diabete entro 5-10 anni ed è quindi molto importante ridurre il più possibile l'aggravarsi di tale situazione [15].

### 1.4 Sintomi e diagnosi

Nel caso del diabete di tipo 1 i sintomi tendono ad insorgere in maniera molto rapida e grave, spesso in concomitanza con febbre, aumento della sete, aumento della quantità di urine, stanchezza, perdita di peso, pelle secca, aumento della fame, difficoltà alla vista, nausea, vomito, dolori addominali [17,18]. Nel caso invece, del diabete di tipo 2, generalmente non viene diagnosticato per molti anni poiché l'iperglicemia si sviluppa molto gradualmente, ma i sintomi sono i medesimi, con effetti più leggeri [17].


I criteri per la diagnosi del diabete sono [16,17,18]:

- Sintomi di diabete associati ad un valore di glicemia superiore a 200 mg/dL
- Glicemia a digiuno superiore a 126 mg/dL (almeno 8 ore dall'ultima assunzione di cibo)
- Glicemia superiore a 200 mg/dL durante una curva da carico, test effettuato somministrando 75 grammi di glucosio
- Concentrazione di emoglobina glicata (HbA<sub>1c</sub>) superiore al 6,5%

Per la diagnosi di condizioni di prediabete i criteri sono i seguenti [18]:

- Alterata glicemia a digiuno (IFG) per valori di glicemia a digiuno compresi tra 100 e 125 mg/dL

- Alterata tolleranza al glucosio (IGT) per valori di glicemia due ore dopo il carico di glucosio compresi tra 140 e 200 mg/dL.

Test	<b>Diabetes</b> Should be diagnosed if ONE OR MORE of the following criteria are met	<b>Impaired Glucose Tolerance (IGT)</b> Should be diagnosed if BOTH of the following criteria are met	<b>Impaired Fasting Glucose (IFG)</b> Should be diagnosed if THE FIRST OR BOTH of the following are met
 Fasting plasma glucose	<b>≥7.0</b> mmol/L (126 mg/dL)	<b>&lt;7.0</b> mmol/L (126 mg/dL)	<b>6.1 – 6.9</b> mmol/L (110 – 125 mg/dL)
or			
 Two-hour plasma glucose after 75g oral glucose load (oral glucose tolerance test (OGTT))	<b>≥11.1</b> mmol/L (200 mg/dL)	<b>≥7.8 and &lt;11.1</b> mmol/L (140-200 mg/dL)	<b>&lt;7.8</b> mmol/L (140 mg/dL)
or			
 HbA <sub>1c</sub>	<b>≥48</b> mmol/mol (equivalent to 6.5%)		
or			
 Random plasma glucose in the presence of symptoms of hyperglycaemia	<b>≥11.1</b> mmol/L (200 mg/dL)		

*Figura 1.2 Criteri di diagnosi per il diabete [16]*

## 1.5 Complicanze

Le complicanze croniche del diabete più diffuse sono quelle vascolari ed oculari, sono più frequenti nei pazienti affetti da diabete di tipo 2 e si manifestano solitamente dopo 10-15 anni dalla comparsa della malattia [22]. Tuttavia, le condizioni più acute si verificano nei pazienti affetti da diabete di tipo 1 in relazione alla carenza quasi totale dell'insulina. Il paziente nei casi più gravi può andare in contro a coma chetoacidotico, dovuto all'accumulo di chetoni, causando perdita di coscienza, disidratazione e gravi alterazioni ematiche [18]. Nel paziente diabetico di tipo 2 le complicanze acute sono rare, ma sono molto frequenti le complicanze croniche che colpiscono occhi, reni, cuore, vasi sanguigni e nervi periferici [18].

Le complicanze principali sono [18]:

- Retinopatia: un danno a carico dei vasi sanguigni che scorrono nella retina, causando la perdita parziale o totale della vista. Le persone diabetiche hanno maggiori probabilità di sviluppare glaucoma e cataratta

- Nefropatia diabetica: è una riduzione progressiva della funzione di filtro del rene e se non trattata può portare all'insufficienza renale con la necessità di dialisi o trapianto
- Malattie cardiovascolari: il rischio di infarto del miocardio o ictus è da 2 a 4 volte nelle persone diabetiche, provocando oltre il 50% di morti per diabete. Questo porta a considerare i pazienti diabetici allo stesso rischio di pazienti non diabetici che hanno avuto un infarto in passato.
- Neuropatia diabetica: secondo l'Organizzazione Mondiale della Sanità è una complicanza che colpisce il 50% dei diabetici. Può causare intorpidimento e formicolio degli arti, perdita di sensibilità, dolore e danni agli arti, nei casi più gravi si raggiunge l'amputazione, disfunzioni del cuore, occhi, stomaco ed è una causa dell'impotenza maschile [22].
- Piede diabetico: a causa dei carichi che sopporta il piede può essere soggetto ad ulcerazioni, portando ad amputazione nei casi peggiori. È dovuto alla modificazione della struttura dei vasi sanguigni e dei nervi

È importante quindi, che il paziente sia molto attento al monitoraggio dei livelli di glucosio nel sangue e che effettui correttamente le somministrazioni di insulina per evitare l'insorgere di complicanze da iperglicemia. Tuttavia, una somministrazione eccessiva di insulina, la mancata consumazione di un pasto, l'attività fisica, una situazione di stress o il consumo di alcoolici, possono portare all'abbassamento dei livelli di glicemia, che scendendo al di sotto di 70 mg/dL portano il paziente ad una condizione di ipoglicemia [23]. È importante curare tempestivamente le condizioni di ipoglicemia, consumando una bustina di zucchero o una bibita ricca di zuccheri, in quanto ripetuti episodi ipoglicemici possono causare gravi problemi nel paziente [23].

L'ipoglicemia può verificarsi in tre gradi [23]:

- Lieve: accompagnata da sudorazione, tremore, palpitazioni e fame intensa
- Moderata: disturbi oculari e offuscamento della vista, formicolio, difficoltà di concentrazione, sonnolenza, irritabilità e mal di testa
- Grave: porta alla perdita di conoscenza e, in un paziente fragile, può essere fatale. Può favorire infarti ed ictus, e causare seri danni al sistema nervoso, cuore e cellule nervose.

L'ipoglicemia colpisce spesso anche durante il sonno, in maniera inconsapevole da parte del paziente. Sono le più pericolose, soprattutto per i bambini, in quanto non si

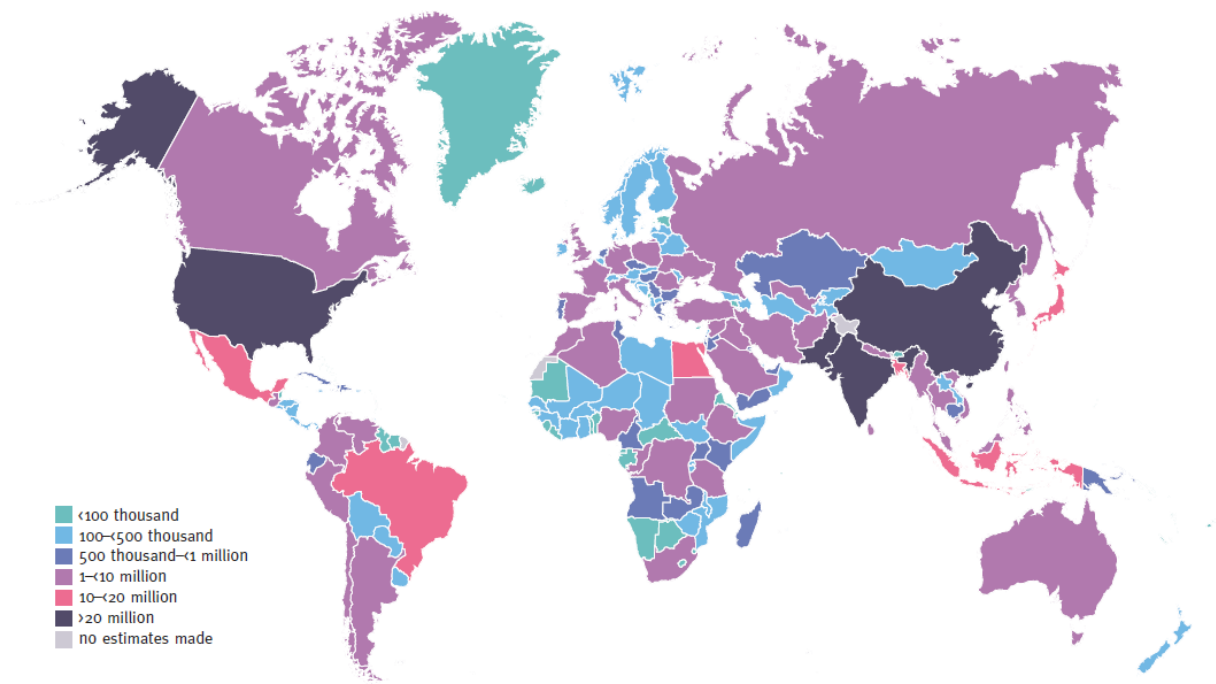
manifestano sintomi evidenti durante il periodo notturno e i genitori non sono in grado di rilevare tale problematica [24].

## 1.6 Il diabete nel mondo e in Italia

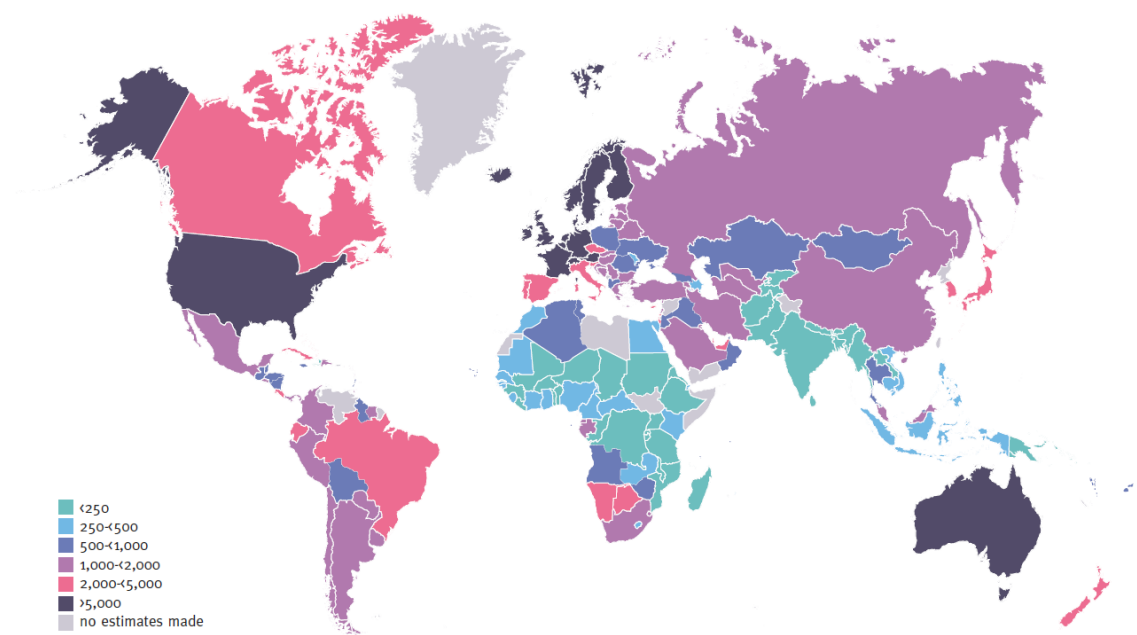
L'International Diabetes Federation (IDF), nel 2021, ha stimato che nel mondo sono presenti 537 milioni di persone affette da diabete, con un'età compresa tra 20-79 anni, corrispondente al 10,5% delle persone in questa fascia d'età e oltre 1,2 milioni di bambini e adolescenti affetti da diabete di tipo 1. Le previsioni stimano una crescita della popolazione diabetica adulta fino a 643 milioni (11,3%) nel 2030 e 783 milioni (12,2%) nel 2045. Il dato preoccupante è il numero di persone affette da diabete non diagnosticato, stimato a 240 milioni di persone, ovvero quasi 1 persona su 2 affetta da diabete è inconsapevole, il che la rende una malattia ancora più pericolosa. La maggior parte proviene soprattutto da paesi con situazioni economiche di livello medio-basso, come Africa, sud-est asiatico e la regione dell'Ovest Pacifico [16]. Sebbene i metodi di assistenza verso i pazienti diabetici siano in continuo sviluppo e con la nuova tecnologia la qualità di vita è migliorata, il diabete è ancora una delle principali cause di morte. L'Organizzazione mondiale della Sanità nel 2019 ha constatato 1,5 milioni di decessi a causa del diabete, la nona causa di morte nel mondo [25].

I costi economici legati al diabete sono cresciuti considerevolmente, crescendo del 316% in 15 anni. La spesa totale, sia pubblica che privata, è passata da 232 miliardi di dollari nel 2007 a 966 miliardi nel 2021, per adulti con un'età compresa tra 20 e 79 anni. L'IDF stima un'ulteriore crescita fino a 1,03 trilioni di dollari nel 2030 e 1,05 trilioni nel 2045. L'aumento considerevole è da attribuirsi al miglioramento della qualità di vita dei pazienti e alle tecnologie emergenti applicate alla cura del diabete, come l'innovativo pancreas artificiale [16].

In Italia l'indagine ISTAT del 2020 stima una prevalenza del diabete pari al 5,9%, che corrisponde a oltre 3,5 milioni di persone, di cui il 21% ultrasettantacinquenni [26]. In Italia il costo è di oltre 20 miliardi di euro l'anno, per farmaci, ospedalizzazioni e assistenza, ma anche a causa della perdita di produttività e delle spese a carico del sistema previdenziale. Il costo per il monitoraggio e la cura del diabete è di circa 2800 euro a persona, più del doppio di una persona non diabetica [27].



*Figura 1.3 Stima del numero di adulti (20-79 anni) con diabete nel 2021 [16]*



*Figura 1.4 Spesa pro capite per la cura del diabete nei pazienti di età tra i 20 e i 79 anni [16]*

## 1.7 Trattamento del diabete di tipo 1: Il Pancreas Artificiale

Per il trattamento del diabete di tipo 1 è necessario che il paziente si somministri delle dosi di insulina per abbassare il livello di glucosio nel sangue e portarlo a livelli

ottimali. La somministrazione può avvenire manualmente attraverso delle penne da insulina, pre-riempite o ricaricabili. Il trattamento consiste nel somministrare l'insulina basale, lenta o mista, ed effettuare dei boli di insulina rapida, di solito al momento dei pasti o dello spuntino. La quantità di insulina da somministrare dipende dalla quantità di carboidrati assunti [28]. Una possibile alternativa sono i microinfusori, dispositivi che erogano insulina ad azione rapida continuamente. I microinfusori sono delle pompe per l'insulina che consentono l'infusione continua nel tessuto sottocutaneo (CSII). Sono dei dispositivi indossabili, possono essere a cerotto o con dei tubicini, e somministrano una quantità fissa di insulina in maniera continuativa in modo da simulare l'azione del pancreas, e dei boli in corrispondenza dei pasti o come correzione della glicemia. Il microinfusore viene accoppiato ad un sensore per il monitoraggio continuo della glicemia (CGM), costituendo il cosiddetto sistema semi-integrato (open loop system). La quantità di insulina da somministrare richiede, anche in questo caso, l'intervento del paziente in base al livello di glucosio nel sangue o alla quantità di carboidrati assunti [28][31].

Negli ultimi anni sta prendendo sempre più spazio un dispositivo in grado di monitorare il livello di glicemia e somministrare in maniera autonoma la quantità di insulina da somministrare, per mantenere i parametri in condizioni ottimali, il pancreas artificiale. Il pancreas artificiale migliora notevolmente la qualità di vita del paziente, riducendo l'onere e lo stress che ne deriva dalla cura di questa malattia. Questo sistema libera le persone dal monitoraggio del glucosio, dalle iniezioni sull'addome e dal peso della gestione della malattia [29].

Il pancreas artificiale viene definito anche come sistema di controllo della glicemia ad ansa chiusa, *closed-loop system*. Esistono due tipologie di sistemi ad ansa chiusa, l'*hybrid close-loop* e il *full closed loop*. Il sistema ibrido controlla autonomamente la somministrazione di insulina basale, ma è richiesto l'intervento umano al momento del pasto per il calcolo dei carboidrati e, di conseguenza, il bolo da somministrare. I sistemi ibridi in commercio sono mono-ormonali, ovvero permettono l'infusione della sola insulina, ma dal 2014 si stanno sperimentando sistemi bi-ormonali, in grado di somministrare il glucagone, l'ormone antagonista dell'insulina e che permette l'innalzamento dei livelli di glucosio per prevenire episodi di ipoglicemia. Il sistema completo è totalmente automatico e non richiede interventi umani, tuttavia presenta un ritardo nell'infusione sottocutanea di insulina e non è possibile garantire un buon controllo in ogni circostanza [30].

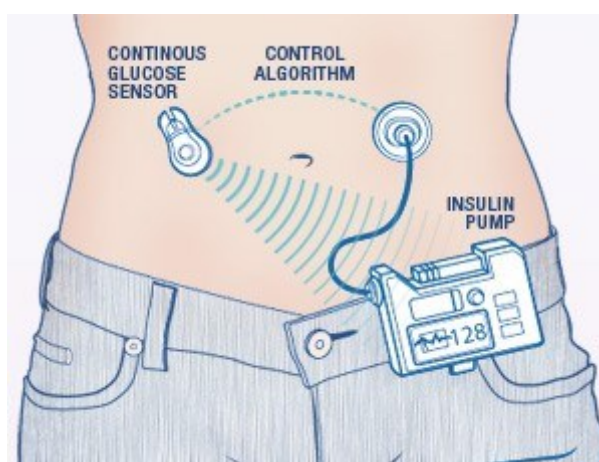
Il sistema è costituito da tre elementi principali [30]:

- Un sensore sottocutaneo per il monitoraggio continuo del glucosio (CGM)
- Un microinfusore per la somministrazione di insulina
- Un algoritmo di controllo, integrato nel microinfusore o in un dispositivo esterno

I tre elementi comunicano in maniera wireless. Il sensore CGM monitora il livello di glucosio con un periodo di campionamento che varia tra 1-5 minuti, i più comuni in commercio monitorano i valori ogni 5 minuti. Il valore viene comunicato all'algoritmo e, se il valore è troppo alto (iperglicemia), il microinfusore effettua un'iniezione della quantità corretta di insulina così da abbassare il livello di glicemia, ma non scendere al di sotto della soglia di ipoglicemia [30]. L'algoritmo è il fulcro del funzionamento del pancreas artificiale, in quanto deve determinare la quantità corretta di insulina da infondere, così da reagire tempestivamente a condizioni di iperglicemia in corrispondenza dei pasti, ma deve essere anche in grado di cessare la somministrazione se i livelli di glucosio sono troppo bassi, così da evitare episodi di ipoglicemia [30].

Tuttavia, sia i sensori CGM che le pompe CSII sono soggetti a guasti, guasti che possono essere critici per la sicurezza del paziente. È quindi necessario sviluppare dei sistemi di monitoraggio in grado di rilevare malfunzionamenti o anomalie.

Principalmente i guasti interessano le pompe per l'infusione di insulina, malfunzionamenti dovuti a difetti, occlusioni o all'estrazione del catetere dal sito di infusione possono portare ad una somministrazione insufficiente di insulina e quindi, a casi di iperglicemia [14].



*Figura 1.5* Illustrazione schematica delle componenti del pancreas artificiale [30]

# Capitolo 2

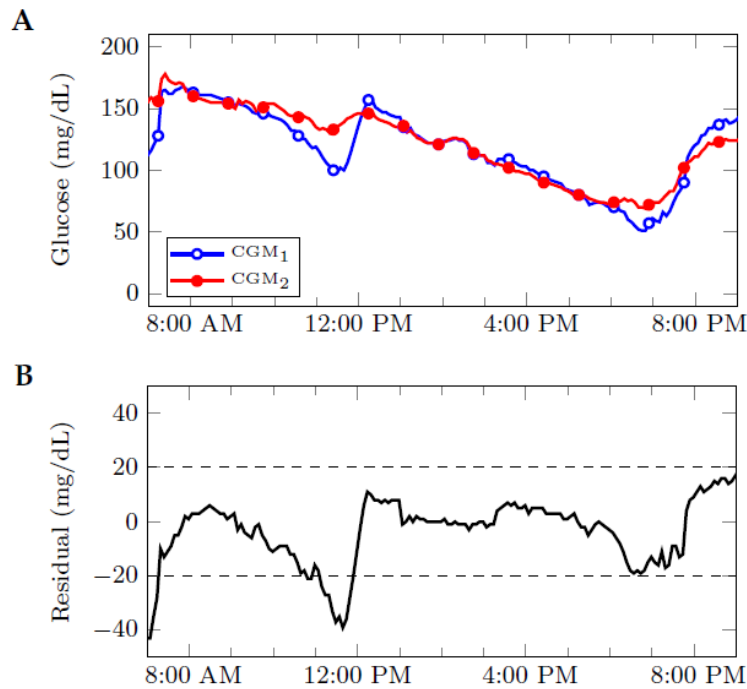
## Algoritmo per individuazione LISA

### 2.1 Introduzione all'algoritmo

I pancreas artificiali hanno migliorato notevolmente la cura dei pazienti affetti da diabete di tipo 1, grazie al progresso dei dispositivi medici esterni, tra cui il sensore per il monitoraggio continuo del glucosio (CGM) e le pompe per l'infusione di insulina (CSII) [1,2]. L'obiettivo di questi dispositivi è di ridurre gli eventi avversi dovuti alle oscillazioni dei valori di glicemia nei pazienti diabetici. Escursioni glicemiche molto elevate e valori di glicemia troppo elevati possono portare a serie complicanze nel paziente. I pancreas artificiali hanno come primo obiettivo preservare la sicurezza del paziente, ciò nonostante, è comunque richiesto un coinvolgimento della persona per cambiare i sensori, scegliere i siti per l'infusione di insulina e diagnosticare eventuali problemi. Uno dei principali problemi con i pancreas artificiali, è la perdita della capacità del sistema di attivare la pompa di insulina, un problema che può verificarsi per numerose ragioni, tra cui il catetere del sistema di infusione spostato, perdita di insulina, lipoipertrofia (accumulo di tessuto adiposo) nel sito di infusione o un guasto alla pompa. Di conseguenza segnalare al paziente che un potenziale guasto è stato individuato, può prevenire effetti avversi dovuti ad escursioni iperglicemiche. [9]

Le strategie per rilevare i guasti del sistema vengono definite come approcci hardware o analiticamente ridondanti. La ridondanza hardware comporta l'aggiunta di hardware extra per il solo scopo di individuare un determinato guasto. Nel caso di PISAs [3] (pressure induced sensor attenuations), si utilizzano due sensori per il monitoraggio della glicemia (CGMs, continuous glucose monitors) e il guasto è individuato confrontando le uscite dei due sensori. I PISAs sono dei guasti che avvengono soprattutto di notte quando il paziente dorme sopra il sensore, l'algoritmo sottostima il valore di glucosio nel sangue e può causare lo spegnimento indesiderato della pompa per l'infusione di insulina [3].

La figura 2.1 illustra come si può sviluppare un approccio simile [9].



**Figura 2.1** Rappresentazione del rilevamento di un guasto nel sensore per il monitoraggio della glicemia (CGM) attraverso un approccio hardware ridondante. (A) Un paziente indossa due sensori CGM in posizioni diverse e (B) questi segnali sono confrontati per generare un residuo. L'approccio HW ridondante analizza il residuo per rilevare dei segni di guasto. In questo caso è stata utilizzata una soglia di  $\pm 20\text{mg/dL}$  [9].

Tale strategia è stata applicata utilizzando i dati in *silico* ottenuti dal simulatore UVa/Padova. Il simulatore è in grado di generare un set di dati contenente i risultati in *silico* del trattamento con insulina di pazienti affetti da diabete mellito di tipo 1, utilizzando un algoritmo per pancreas artificiale completamente automatizzato.[10]. Gli autori assumono che il paziente cambi la pompa di infusione ad ogni segnale di allarme, e hanno constatato una sensibilità del 55% e una specificità del 66% [9].

Le scarse conoscenze riguardo le cause dei LISAs e la loro probabilità di verificarsi creano ancora molta incertezza sulla modellizzazione dei guasti di tipo LISA, necessari negli studi di simulazione. Uno dei principali vantaggi del lavoro svolto da Howsmon e dai suoi collaboratori, è la possibilità di valutare l'algoritmo di rilevamento LISA proponendo dati clinici piuttosto che degli scenari simulati. In aggiunta, la semplicità dell'algoritmo è di facile integrazione all'interno dei sistemi di pancreas artificiale, richiedendo risorse computazionali trascurabili.

## 2.2 Notazione

- $k$ : 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 quantità media di  $x$  in una finestra dal tempo  $j$  al tempo  $i$
- $\Delta x_k$ : rappresenta la quantità  $x_k - x_{k-1}$

## 2.3 Sviluppo dell'algoritmo

Le perdite nell'attivazione del set di infusione (Losses in Infusion Set Actuation, LISA) [9] possono portare ad una somministrazione insufficiente, o talvolta assente, dell'insulina nel paziente portando ad un innalzamento del livello di glucosio nel sangue. È importante intervenire tempestivamente per ridurre il prolungarsi di condizioni iperglicemiche nel paziente, infatti, il vero pericolo sta nel fatto che il paziente non si curi di tale guasto, cosa che potrebbe aggravare ulteriormente la situazione. Ogni paziente ha un diverso livello di controllo dei propri parametri ed è quindi importante che l'algoritmo sia abbastanza robusto da poter essere applicato ad ogni diversa situazione e paziente. Durante la giornata il paziente può essere impegnato a lavorare, può fare attività fisica o riposare, tutte queste diverse situazioni comportano livelli di controllo differenti durante la giornata, che possono essere più o meno attenti.

Per individuare un guasto si analizzano i parametri del paziente in una finestra recente, Short Window, e li si confrontano con i parametri di una finestra più lunga, Long Window. La SW rappresenta i parametri del paziente più recenti e che possono quindi indicare un possibile guasto, mentre la LW rappresenta i parametri del paziente su una finestra di analisi più lunga e permette quindi di stimare le abitudini e i parametri medi del paziente. Confrontando queste due finestre è possibile individuare se il valore di glucosio nel sangue del paziente è nella media con le abitudini o è troppo alto. Tuttavia, solo questo confronto non è sufficiente per segnalare un guasto al paziente. E' necessario analizzare anche il tasso di crescita del glucosio nel sangue (ossia la sua derivata), infatti, un brusco aumento della glicemia può essere un ulteriore sintomo di un malfunzionamento del sistema. Il terzo e ultimo confronto riguarda il livello di insulina nel sangue. Infatti, un picco glicemico può essere dovuto ad un pasto od uno spuntino, è quindi importante stimare la quantità di insulina attiva nel sangue, piuttosto

che quella totale somministrata, per capire se l'aumento improvviso del glucosio sia dovuto ad un pasto o ad un malfunzionamento.

Sono state quindi stabilite tre metriche di guasto, una per il glucosio, una per l'insulina e una per la pendenza del segnale di glucosio. Se in una determinata finestra il paziente presenta dei parametri che superano contemporaneamente queste tre metriche, viene inviato un segnale di allarme che avvisa la presenza di un malfunzionamento nell'attivazione del set di infusione.

## 2.4 Metriche di guasto

### 2.4.1 Glucose Fault Metric (GFM)

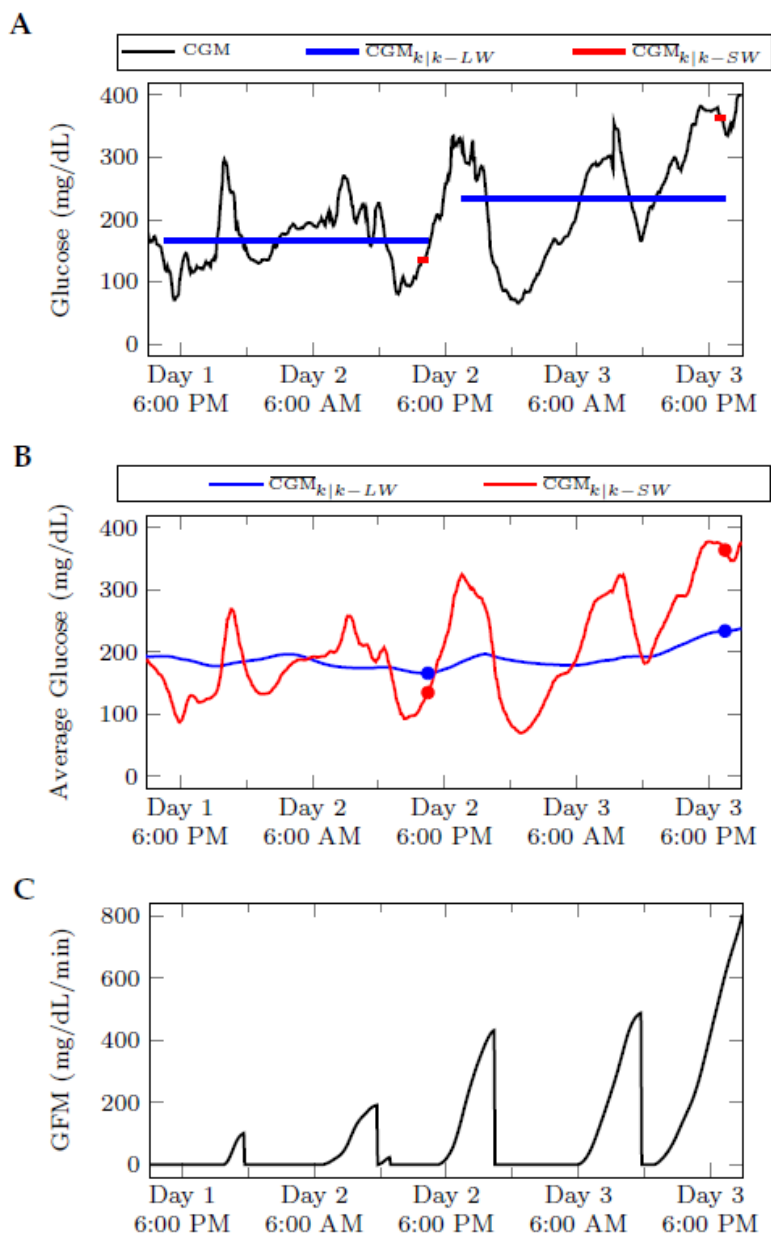
Un primo segnale di una perdita nell'attivazione del set di infusione (LISA) sono i picchi glicemici, infatti una mancata somministrazione di insulina o una somministrazione insufficiente, può portare ad un brusco aumento della glicemia, che si può individuare analizzando il valore di glucosio (CGM) nelle finestre di osservazione, Long Window e Short Window [9]. A causa dei diversi livelli di controllo tra pazienti e la possibilità che il paziente sia più o meno attento al controllo dei parametri, è necessario calcolare il valore medio del glucosio all'interno di una finestra lunga di analisi,  $\overline{CGM}_{k|k-LW}$ . La media del CGM nella finestra lunga fornisce il valore medio del glucosio nel paziente e consente di calibrare l'algoritmo sulle specifiche esigenze della persona. La media nella finestra di osservazione corta,  $\overline{CGM}_{k|k-SW}$ , è utilizzata per rilevare i valori più recenti di glucosio nel paziente. Se il valore medio del glucosio nella finestra corta,  $\overline{CGM}_{k|k-SW}$ , è maggiore del valore medio del glucosio nella finestra lunga,  $\overline{CGM}_{k|k-LW}$ , l'area tra le due curve  $AUC_k$ , è accumulata nella metrica di guasto del glucosio, Glucose Fault Metric (GFM). Se il valore di  $\overline{CGM}_{k|k-SW}$  scende al di sotto di  $\overline{CGM}_{k|k-LW}$ , il GFM viene resettato a 0. Tipicamente i sensori per il monitoraggio del valore di glucosio nel sangue hanno un periodo di campionamento pari a 5 minuti, assumiamo quindi  $\Delta t_k = 5 \text{ min}$  [9].

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

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

Tuttavia, questa metrica da sola non è sufficiente, per eliminare la possibilità di segnalare dei falsi guasti, l'allarme è inviato solo se il GFM e la pendenza delle letture di glucosio nella finestra corta sono al di sopra di determinate soglie.

Nella figura 2.2 è illustrato il calcolo della GFM eseguito da Howsmon e i suoi collaboratori.



**Figura 1.2** Illustrazione del glucose fault metric (GFM) calcolato per  $LW=24$  h e  $SW=1$  h. (A) La lunghezza della linea orizzontale blu indica la durata della finestra di osservazione lunga, pari a 24h, corrispondente al  $\overline{CGM}_{k|k-LW}$ , mentre la linea rossa rappresenta la durata della finestra corta, pari a 1 h, corrispondente al  $\overline{CGM}_{k|k-SW}$ . (B) Le medie dei valori di glucosio nelle due finestre di osservazione sono calcolate punto per punto. I punti evidenziati corrispondono alle linee orizzontali della figura soprastante. (C) Se  $\overline{CGM}_{k|k-SW} > \overline{CGM}_{k|k-LW}$  l'area tra le due curve è accumulata nel GFM. Quando  $\overline{CGM}_{k|k-SW} < \overline{CGM}_{k|k-LW}$ , il GFM viene resettato a 0 [9].

## 2.4.2 Insulin Fault Metric (IFM)

I picchi glicemici da soli non riescono ad indicare con precisione la presenza di un guasto nella pompa di insulina. Infatti, durante la giornata abituale di un paziente, possiamo individuare delle escursioni di glucosio molto elevate dovute ai pasti o a degli spuntini. Molto spesso i pazienti non utilizzano il calcolatore del bolo del pasto, per mancanza di tempo o, magari, per dimenticanza, e alcuni algoritmi per pancreas artificiali non necessitano di utilizzare le informazioni dei pasti per poter funzionare correttamente [4,5][9]. È quindi difficile capire se il picco iperglicemico sia dovuto ad un pasto o meno e per far fronte a queste situazioni e rendere l'algoritmo il più ampiamente applicabile, è stata introdotta una nuova metrica di guasto che riguarda l'insulina (IFM) [9]. Tale metrica di guasto si basa sulla quantità di insulina attiva nel sangue. L'insulina viene iniettata per via sottocutanea mediante una pompa di insulina e prima di poter influenzare il livello di glucosio nel sangue del paziente deve attraversare il compartimento sottocutaneo. L'insulina iniettata non penetra immediatamente all'interno del sangue del paziente, ma viene rilasciata gradualmente [11]. Inoltre, in caso di iniezioni multiple di insulina a brevi intervalli di tempo, può verificarsi una sovrapposizione delle curve di azioni dell'insulina, con un conseguente fenomeno di accumulo di insulina [12]. Questo implica che l'insulina precedentemente iniettata influenzi ancora la glicemia del paziente, è importante quindi definire la quantità di insulina attiva nel sangue al momento dell'analisi dei parametri del paziente, piuttosto che la quantità totale di insulina iniettata. Tale insulina prende il nome di Plasma Insulin Estimate (PIE), e permette il calcolo della relativa metrica di guasto legata all'insulina. Per i motivi descritti precedentemente, è necessario definire la quantità di insulina attiva nel sangue attraverso un modello compartimentale.

$$\begin{bmatrix} x_{k+1}^1 \\ x_{k+1}^2 \end{bmatrix} = \begin{bmatrix} 0.98 & 0.02 \\ 0 & 0.98 \end{bmatrix} \begin{bmatrix} x_k^1 \\ x_k^2 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} Insulin_k$$

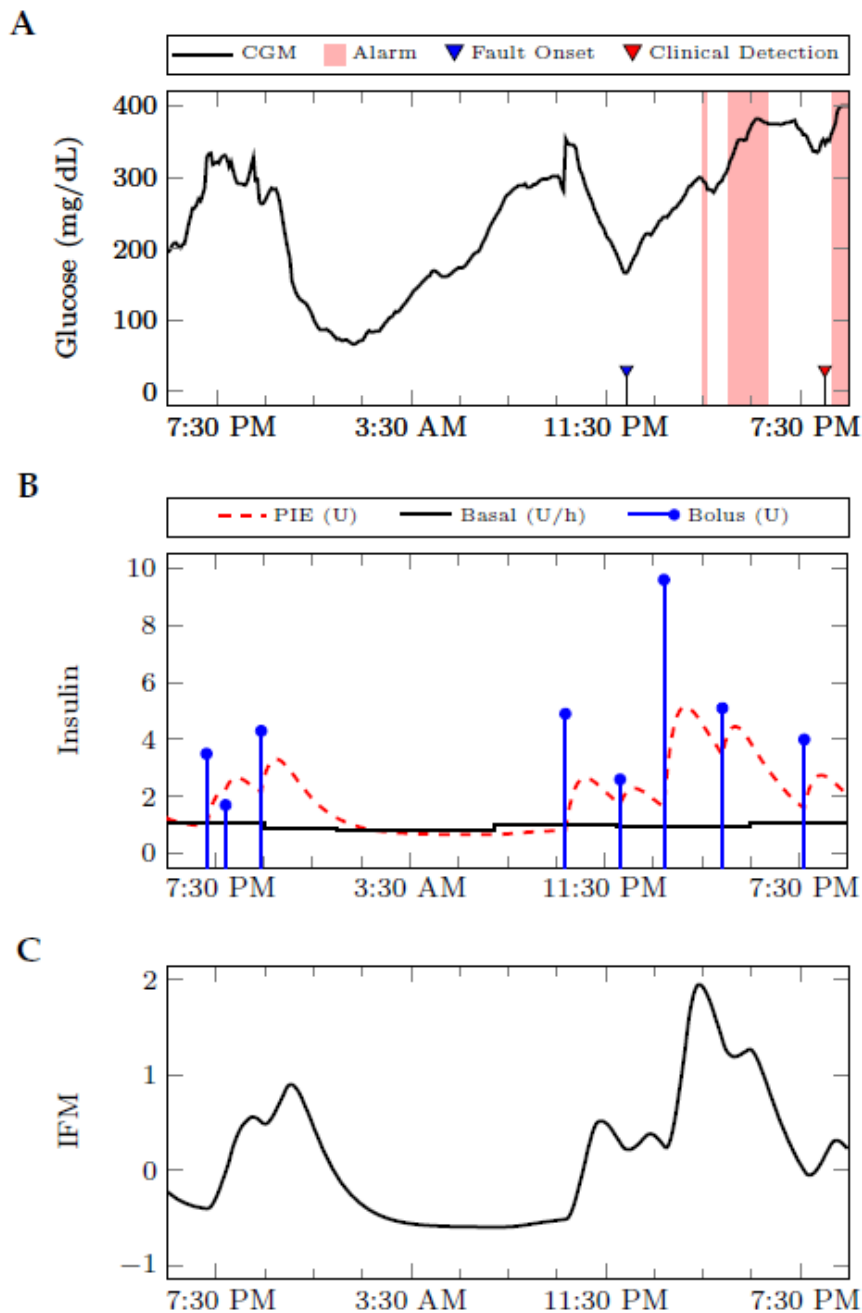
$$PIE_k = [1 \quad 0] \begin{bmatrix} x_k^1 \\ x_k^2 \end{bmatrix}$$

Il modello è diviso in due compartimenti, 1 e 2. Il compartimento 1 rappresenta l'insulina attiva nel plasma mentre il compartimento 2 è la quantità di insulina depositata nel sottocute. La terza variabile,  $Insulin_k$ , rappresenta la quantità totale di insulina, basale e bolo, iniettata nel sottocute, ad un istante k. Analizzando il modello è possibile notare come nell'istante successivo alla somministrazione dell'insulina, il 98% permane all'interno del sottocute,  $x^2$ , mentre il restante 2% viene trasferito al plasma e

può quindi influenzare i livelli di glicemia del paziente. A causa di processi metabolici il 2% dell'insulina nel plasma,  $x^1$ , viene consumata. Per il calcolo dell'Insulin Fault Metric (IFM), la media del plasma insulin estimate (PIE) nella finestra corta è confrontata con la media della stessa nella finestra lunga.

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

Un esempio del calcolo dell'IFM calcolata da Howsmon e i suoi collaboratori è riportata nella figura 2.3.



**Figura 2.3** Illustrazione dell'Insulin Fault Metric (IFM) calcolata per  $LW=24$  h e  $SW=1$  h. (A) Livello di glucosio del paziente durante la giornata, con indicazione dell'inizio del guasto, l'individuazione clinica

*da parte dello staff medico e le segnalazioni di allarme. (B) Insulina basale e bolo di insulina utilizzate per determinare il Plasma Insulin Estimate (PIE) (C) Rappresentazione metrica di guasto dell'insulina (IFM), calcolata con il modello precedentemente illustrato [9].*

La metrica di guasto legata all'insulina attiva nel sangue è stata utilizzata per individuare i momenti in cui i livelli di glucosio sono più alti della media, nonostante la quantità di insulina somministrata sia al di sopra della media ed è quindi plausibile la presenza di un guasto. Per il calcolo è stato utilizzato un tempo di campionamento di 1 minuto, in quanto i cambiamenti nel bolo di insulina e nell'insulina basale, possono avvenire in un tempo molto inferiore rispetto i 5 minuti dei sensori CGM.

### 2.4.3 Glucose slope (GS)

Le metriche di guasto legate al glucosio e all'insulina sono state definite, ma è necessario definire la terza metrica di guasto per ridurre l'insorgere di falsi allarmi. Tale metrica è la Glucose Slope (GS), ovvero la misura del tasso di crescita del segnale di glucosio (CGM). È plausibile supporre che, in presenza di una somministrazione insufficiente di insulina, i valori di glucosio crescano in maniera anomala. Valutando quindi la derivata del segnale di glucosio, è possibile individuare i momenti in cui si ha un picco glicemico al di fuori della normalità.

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

Se le tre metriche vengono superate contemporaneamente all'interno di una finestra corta, viene inviato un allarme al paziente, che segnala la presenza di un malfunzionamento nella pompa di insulina.

## 2.5 Prestazione dell'algoritmo

Per valutare le prestazioni dell'algoritmo per il rilevamento di LISA (Losses in Infusion Set Actuation) sono stati utilizzati tre differenti set di dati. Il primo set di dati T1 è stato preso da uno studio sugli effetti della lipoipertrofia nei siti di infusione di insulina [6], è stato utilizzato per il primo test dell'algoritmo e per l'impostazione dei parametri iniziali riportati in tabella 2.2. Successivamente l'algoritmo è stato testato sul set di dati ottenuto dallo studio V1, il quale confrontava i set di infusione in Teflon con quelli in metallo [7], e sullo studio V2, il quale analizzava l'effetto della ialuronidasi sulla farmacocinetica dell'insulina sottocutanea [8]. Lo studio ha riscontrato che la ialuronidasi influenza la farmacocinetica dell'insulina e di conseguenza, le prestazioni dell'algoritmo, per cui è stato utilizzato solo il gruppo di controllo dello studio V2, nel

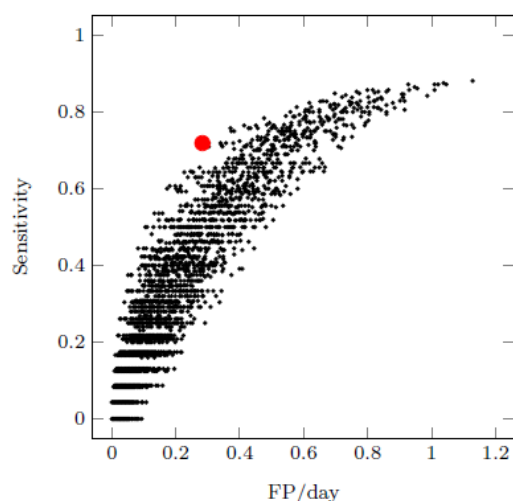
quale la sostanza non è somministrata. L’algoritmo è stato testato su un totale di 51 pazienti appartenenti a tre studi differenti, questa indipendenza tra i vari studi garantisce un’ottima generalizzazione dell’algoritmo a diverse tipologie di pazienti e non ad uno specifico data set.

I risultati sono riportati nella tabella 2.1.

	T1	V1	V2
Reference	[23]	[24]	[25]
Number of patients	20	18	13
Number of infusion sets	62	49	22
Total patient days	352.7	275.7	106.9
Number of infusion set failures	23	15	10
Algorithm Sensitivity	71.8%	73.3%	71.4%
Algorithm FP/day	0.28	0.27	0.28
Algorithm Median Minutes to Detect	262	210	280
Algorithm Glucose at Detection (mg/dL)	289	300	264

**Tabella 2.1** Prestazioni algoritmo e caratteristiche del set di dati. FP: falso positivo, tratta da [9]

Tutti i set di dati sono stati ricavati da studi che hanno individuato clinicamente un malfunzionamento, tuttavia non è possibile stabilire con precisione l’inizio del guasto in quanto non sono stati indotti dallo staff medico, ma sono sorti in maniera spontanea. Di conseguenza, lo staff di ingegneri e medici ha generalizzato l’inizio del guasto in maniera retrospettiva. La sensibilità è stata calcolata basandosi sul fatto che l’algoritmo abbia segnalato o meno un guasto tra il tempo stimato di inizio e l’effettiva individuazione da parte dello staff. Mentre i falsi positivi sono stati calcolati sulla base della segnalazione di falsi allarmi durante il periodo di analisi. Per massimizzare la sensibilità e diminuire il numero di falsi positivi giornalieri, i parametri dell’algoritmo sono stati scelti all’interno di una curva pROC (pseudo receiver operating characteristic), riportata in figura 2.5 [9]. La curva pROC rappresenta la probabilità di individuare un vero positivo in funzione della probabilità di un falso positivo [32]. Il parametro scelto è rappresentato dal pallino rosso in figura, le migliori prestazioni si hanno per valori vicini all’angolo in alto a sinistra.



**Figura 2.5** Illustrazione della curva pROC. La curva confronta il numero di falsi positivi giornalieri con la sensibilità dell' algoritmo, ovvero i veri positivi [9].

Nella tabella 2.2 troviamo indicati i parametri scelti per ottimizzare le performance dell' algoritmo. Con tali parametri l' algoritmo è stato in grado di raggiungere una sensibilità del 71,8% con 0.28 falsi positivi giornalieri [9]. La segnalazione dei guasti è avvenuta a livelli di glucosio già alti, questo indica che l' algoritmo per la rilevazione di LISA, non è in grado di prevenire completamente l' iperglicemia, tuttavia è più veloce nel rilevare il guasto da parte del paziente e questo permette di mitigare parzialmente gli effetti negativi provocati da situazioni iperglicemiche.

**Tabella 2.2** Parametri scelti per ottimizzare l' algoritmo [9]

Nome del parametro	Unità	Valore scelto
LW	h	24
SW	h	1
Soglia GFM	mg/dL*h	100
Soglia IFM	senza unità	0,4
Soglia GS	mg/dL*min	0,3

È importante precisare che, i risultati ottenuti, sono dovuti anche al pronto intervento da parte dello staff medico, per evitare il prolungarsi di situazioni potenzialmente pericolose per il paziente. In effetti la durata media dei guasti mancati dall' algoritmo è di 4,9 ore, mentre quella dei guasti correttamente riconosciuti è di 10,6 ore [9]. Questo suggerisce che, senza l' intervento dello staff, i guasti mancati sarebbero stati correttamente individuati da parte dell' algoritmo, influenzando positivamente la sensibilità dello stesso.

Quando la ialuronidasi è stata somministrata, per ridurre il tempo di azione dell'insulina, l'algoritmo ha riscontrato una sensibilità del 44,4% con 0,33 falsi positivi giornalieri [9], risultati decisamente peggiori rispetto allo studio in cui la ialuronidasi non è somministrata. Di conseguenza l'algoritmo va ricalibrato sul nuovo profilo farmacocinetico dell'insulina se differisce significativamente dal profilo inizialmente predisposto.

L'obiettivo futuro di questo algoritmo è la possibilità di poter integrare le informazioni riguardanti i pasti al fine di ridurre il numero di falsi positivi, per i pazienti che desiderassero poter integrare questi parametri. Infatti, la maggior parte dei falsi positivi è stata attribuita a pasti sotto-bolizzati, in quanto la segnalazione di allarme è stata riscontrata entro le due ore dal pasto.

L'algoritmo è stato sviluppato per poter funzionare in presenza di pasti, ma ha riscontrato prestazioni soddisfacenti anche in condizioni notturne, rilevando correttamente tre malfunzionamenti del set di infusione. Tuttavia, un algoritmo sviluppato specificatamente per l'utilizzo notturno, quando solo l'insulina basale è somministrata, è sicuramente consigliato.

Il principale campo di applicazione di questo algoritmo sono i sistemi open-loop, al fine di migliorare l'attuale livello di cura dei pazienti diabetici. È stato testato anche in scenari closed-loop riscontrando risultati simili, tuttavia le informazioni e i set di dati disponibili sono ancora limitati e non è possibile stabilire con esattezza le prestazioni dell'algoritmo con questi sistemi.



# Capitolo 3

## Reimplementazione e test dell'algoritmo

### 3.1 Introduzione

L'obiettivo di questo elaborato è di ricreare e valutare l'algoritmo per rilevazione LISA, descritto nel capitolo 2, all'interno dell'ambiente di programmazione MATLAB. In particolare, l'ambiente di lavoro utilizzato è MATLAB versione R2023b. Il testo del programma potrà essere consultato nell'appendice A.

L'algoritmo è stato testato, e valutato, utilizzando un dataset di 5 pazienti in *silico* generato attraverso il simulatore UVa/Padova [10]. In questo elaborato vengo riportati i grafici del paziente numero e sarà possibile consultare i grafici di ogni singola metrica utilizzata per la segnalazione di un guasto. Il simulatore ha generato, all'interno di una finestra di 30 giorni, due guasti dalla durata di 6 h ciascuno, a mezzogiorno o a mezzanotte. Ai fini della valutazione delle prestazioni, è stata ritenuta corretta una segnalazione di allarme entro 6h dall'inizio del guasto.

Tutti i parametri presenti all'interno dell'algoritmo e descritti nell'elaborato, sono modificabili al fine di rendere il programma flessibile e applicabile a più contesti.

### 3.2 Simulazione dei segnali

Per facilitare l'iniziale implementazione dell'algoritmo si è proceduto con la simulazione dei parametri del paziente, il dataset ottenuto dal simulatore non è stato impiegato immediatamente, ma solo dopo aver verificato il corretto funzionamento del programma. È stato quindi necessario creare manualmente i segnali necessari per il calcolo delle metriche di guasto, ovvero il CGM (continuous glucose monitoring), insulina basale e bolo di insulina.

Il primo passo è stato la generazione del segnale di debug, utile per calcolare le metriche di guasto legate al glucosio, il segnale CGM e confrontarle con quanto previsto dalla teoria. È stato utilizzato un segnale sinusoidale di questo tipo:

$$CGM = A_o + A \sin\left(\frac{2\pi t}{T}\right)$$

I parametri utilizzati sono:

- $A_o = 150 \text{ mg/dL}$
- $A = 50 \text{ mg/dL}$
- $T = 8 \text{ h}$

Il parametro  $A_o$  rappresenta il valore medio del segnale di glucosio simulato, il parametro  $A$  è stato utilizzato per avere un'oscillazione del segnale compresa tra 100 mg/dL e 200 mg/dL, il parametro  $T$  rappresenta il periodo del segnale sinusoidale utilizzato per l'analisi.

Infine, sono stati stabiliti gli ultimi due parametri, la durata totale dell'analisi pari a 3 giorni o 72 ore e il passo di campionamento del segnale pari a 5 minuti, siccome la maggior parte dei sensori per la misurazione del glucosio presentano un periodo di campionamento pari a 5 minuti. All'interno dell'algoritmo questo valore è stato convertito in ore, pari a  $\frac{1}{12} \text{ h}$ , ovvero in un'ora sono stati rilevati 12 campioni di glucosio. Questi due parametri sono stati utilizzati per la generazione del vettore dei tempi  $t$ , in ore. Il vettore parte dall'ora 0 del primo giorno e termina dopo 72 ore di analisi, con una distanza tra i campioni pari a 5 minuti. È importante precisare che il tempo di analisi corrispondente ad un giorno inizia alle ore 0:00 e termina alle ore 23:55, corrispondente a  $n = 288$  campioni. Tale valore è stato ottenuto utilizzando la seguente formula,  $n = \frac{h}{T_s}$ . Dove  $h$  corrisponde a 24 ore e  $T_s$  al tempo di campionamento pari a  $\frac{1}{12}$ .

Successivamente è stato necessario generare i segnali necessari al calcolo della metrica di guasto legata all'insulina, ovvero l'insulina basale del paziente e il bolo di insulina. Il periodo di campionamento di questi due segnali differisce dal precedente, in quanto è stato stabilito pari ad 1 minuto o  $\frac{1}{60} \text{ h}$ , ovvero 60 campioni all'ora, come descritto nel paragrafo 2.4.2. L'insulina basale è un segnale costante di valore basso, pari ad 1 U/h, che permane per tutta la durata dell'analisi. Il bolo di insulina, invece, è una quantità di insulina molto maggiore che viene somministrata in concomitanza con i pasti. Il bolo di insulina è una quantità misurata in Unità, è stato quindi necessario effettuare la conversione nella stessa unità di misura dell'insulina basale, per poter sommare i due valori e calcolare l'insulina totale somministrata. Il valore stabilito per questa

somministrazione è di 1 Unità pari a 60 U/h. Gli orari scelti per la somministrazione del bolo sono tre, colazione ore 8, pranzo ore 12 e cena ore 20.

Una volta predisposti i segnali, sono state calcolate le metriche di guasto.

### 3.3 Finestre di analisi e soglie di allarme

Per l'analisi dei parametri del paziente e la segnalazione di eventuali guasti, sono state stabilite le finestre di osservazione per eseguire i confronti necessari al calcolo delle metriche di guasto. Tali finestre di osservazione sono finestre mobili, ovvero vengono traslate di un valore pari al passo di campionamento, in modo da eseguire l'analisi dei parametri del paziente per la durata totale delle misurazioni, pari al numero di giorni osservati (in questo caso 3 giorni). Come descritto nell'elaborato di Howsmon et al [9], la Long Window (LW) scelta ha una durata di 24 h e la Short Window (SW) di 1h. La Long Window (LW) definisce i parametri medi del paziente all'interno di una giornata abituale. Per la segnalazione di un allarme di guasto è necessario confrontare gli attuali parametri medi del paziente, calcolati nella Short Window (SW), con i parametri medi all'interno della finestra lunga di osservazione. Questo implica che, per poter stabilire i parametri medi con i quali eseguire il confronto, siano necessarie almeno 24 ore di analisi. Di conseguenza, prima che l'algoritmo possa elaborare i dati e segnalare un guasto, è necessario che sia trascorso il tempo della LW stabilita. Il programma risulta quindi vulnerabile a guasti nelle prime ore corrispondenti alla finestra lunga (in questo caso 24 ore) ed è considerabile funzionante a partire dal secondo giorno di analisi.

All'interno dell'algoritmo, per facilitare il calcolo dei valori medi nelle due finestre di osservazione, è stata implementata una funzione denominata *calc\_Avg*. Tale funzione riceve tre parametri in input, il numero totale di giorni di analisi nei quali calcolare i valori medi, la lunghezza in ore della finestra nella quale calcolare il valore medio e il segnale, in formato vettore riga, dal quale ricavare i dati per il calcolo. In automatico la funzione ricava il periodo di campionamento del segnale e restituisce in uscita un vettore riga contenente i valori medi del segnale per tutta la durata dello stesso, tali valori sono ricavati all'interno della finestra prescelta e traslata di un passo pari al periodo di campionamento del segnale. Per eseguire questo calcolo è stata utilizzata la funzione *mean* fornita all'interno dell'ambiente di lavoro MATLAB.

Le soglie da impostare per la segnalazione di un guasto sono tre, due relative al glucosio e una relativa all'insulina. Per massimizzare le prestazioni dell'algoritmo sono state

scelte le tre soglie indicate nell'elaborato di Howsmon et al. [9]. Quando i parametri del paziente in un dato istante sono al di sopra di tutte e tre le soglie contemporaneamente, l'algoritmo emette un segnale d'allarme che avverte la possibile presenza di un guasto all'interno del set di infusione.

I valori scelti per le soglie sono i seguenti:

- Soglia Glucose Fault Metric (GFM) = 100 mg/dl\*h
- Soglia Insulin Fault Metric (IFM) = 0,4
- Soglia Glucose Slope (GS) = 0,3 mg/dL\*min

## 3.4 Calcolo metriche di guasto

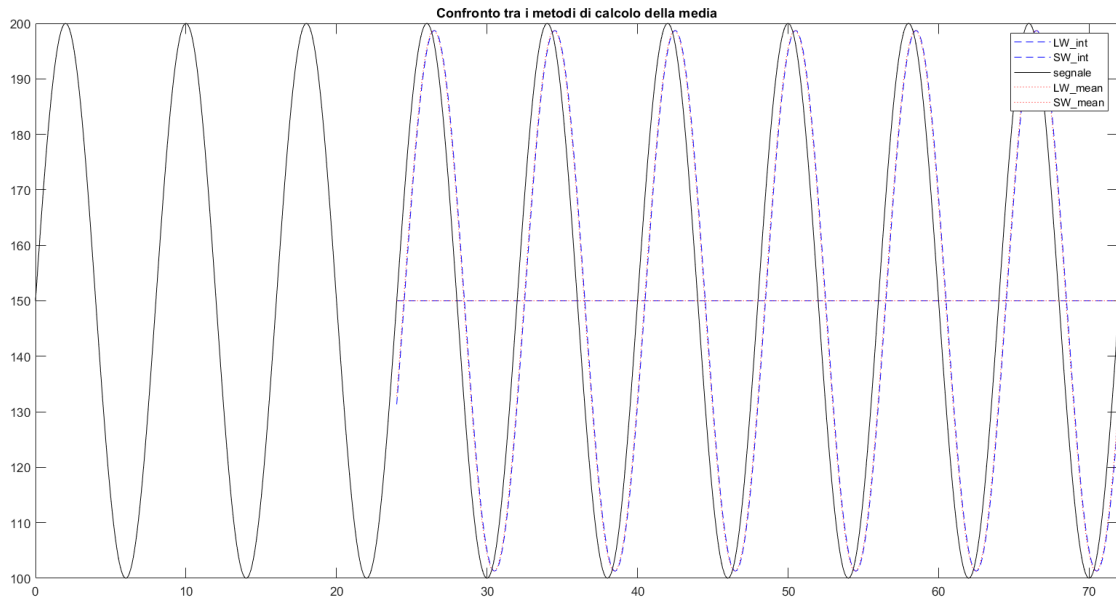
### 3.4.1 Glucose Fault Metric (GFM)

Per il calcolo della prima metrica di guasto sono stati utilizzati i due vettori riga contenenti i valori medi del CGM nelle due finestre di osservazione, rispettivamente la finestra lunga,  $\overline{CGM}_{k|k-LW}$ , e la finestra corta  $\overline{CGM}_{k|k-SW}$ . Per semplicità all'interno dell'algoritmo tali parametri sono indicati rispettivamente come  $CGM_{LW}$  e  $CGM_{SW}$ . Per il calcolo iniziale di tali parametri sono stati utilizzati due metodi differenti e posti a confronto. Il primo metodo utilizza la funzione *calc\_Avg* spiegata nel paragrafo 3.3, per il secondo metodo è stata utilizzata la strategia della media integrale. È stato calcolato il valore dell'area del segnale, all'interno di una determinata finestra ed è stato diviso per la lunghezza della stessa.

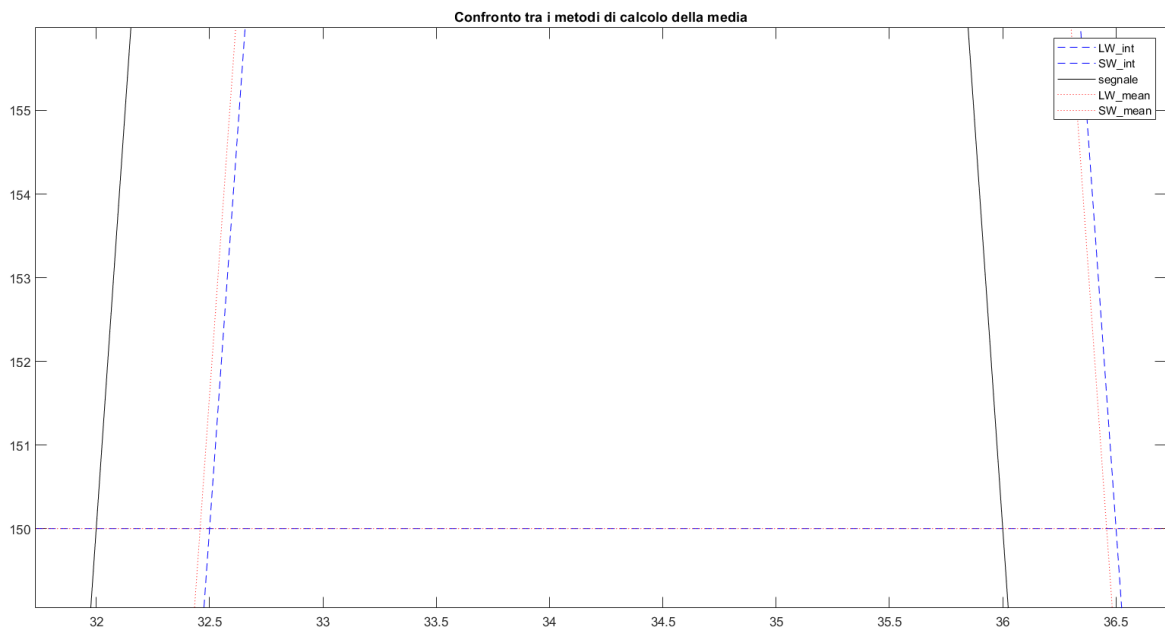
$$CGM_{SW} = \frac{\int_{k-SW}^k A_o + A \sin\left(\frac{2\pi t}{T}\right) dt}{SW} \quad CGM_{LW} = \frac{\int_{k-LW}^k A_o + A \sin\left(\frac{2\pi t}{T}\right) dt}{LW}$$

I due metodi a confronto sono illustrati nella figura 3.1. Come si può notare nella figura 3.1, i valori medi del segnale CGM nelle due finestre mobili partono dalla 24esima ora, ovvero dal secondo giorno. Durante il primo giorno analisi non si ha alcun parametro con il quale eseguire il confronto e poter segnalare un guasto. La figura 3.2 mette in evidenza come i due metodi differiscano leggermente nel calcolo del valore medio del glucosio all'interno della finestra corta, mentre sono identici nella finestra lunga.

Per calcolare analiticamente l'integrale del segnale di glucosio è necessario conoscere la forma analitica della funzione CGM nel tempo, tuttavia, cambiando da paziente a paziente, non è possibile riconoscere un'equazione per tale segnale. È stato quindi preferito il metodo di calcolo attraverso la funzione *calc\_Avg*.

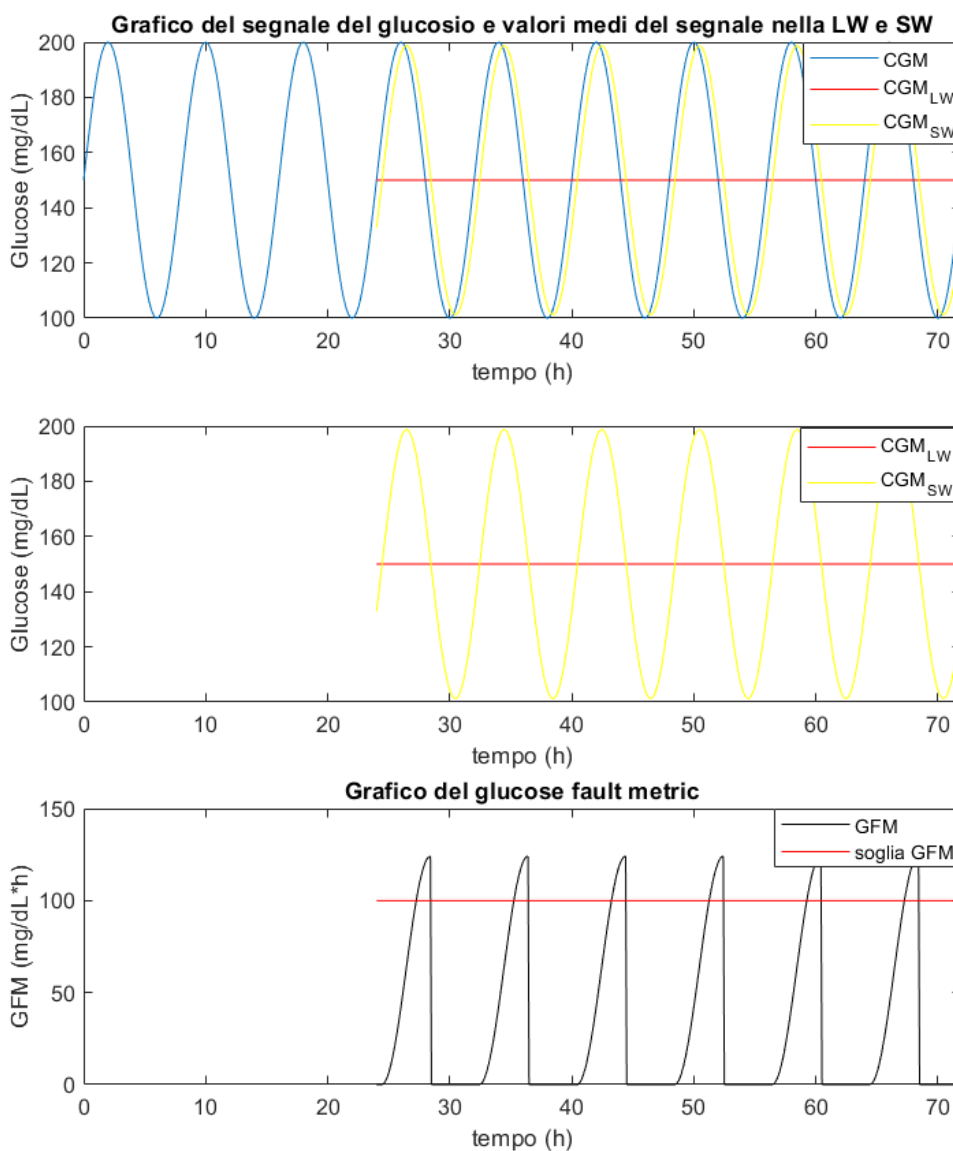


**Figura 3.1** Illustrazione dei due metodi di calcolo del valore medio del segnale CGM nelle due finestre mobili. In nero è indicato il segnale sinusoidale di partenza, in blu il valore medio calcolato con la media integrale e in rosso il valore medio calcolato con la funzione mean



**Figura 3.2** Zoom sull'illustrazione dei due metodi di calcolo del valore medio del segnale CGM nelle due finestre mobili. In nero è indicato il segnale sinusoidale di partenza, in blu il valore medio calcolato con la media integrale e in rosso il valore medio calcolato con la funzione mean. I grafici della LW risultano sovrapposti, mentre i grafici della SW risultano traslati.

Successivamente è stata calcolata la glucose fault metric (GFM) con il metodo descritto nel paragrafo 2.4.1, a differenza dell'intervallo di tempo  $\Delta t$ . La soglia GFM è stabilita in mg/dL\*h, mentre l'intervallo di tempo  $\Delta t$  era stato stabilito pari a 5 min. È stato quindi necessario, effettuare una conversione in ore per poter impostare correttamente la soglia oltre la quale segnalare il guasto. Nella figura 3.3 è rappresentato il grafico del GFM confrontato con la relativa soglia, oltre alla rappresentazione del segnale CGM e il valore medio nelle due finestre mobili. Come si può notare nel grafico C della figura 3.2 si hanno degli istanti in cui i valori del GFM oltrepassano la soglia pari a 100 mg/dL, in corrispondenza di quei valori potrebbe essere emesso un segnale d'allarme se viene oltrepassata anche la soglia relativa alla metrica di guasto legata all'insulina e alla glucose slope.



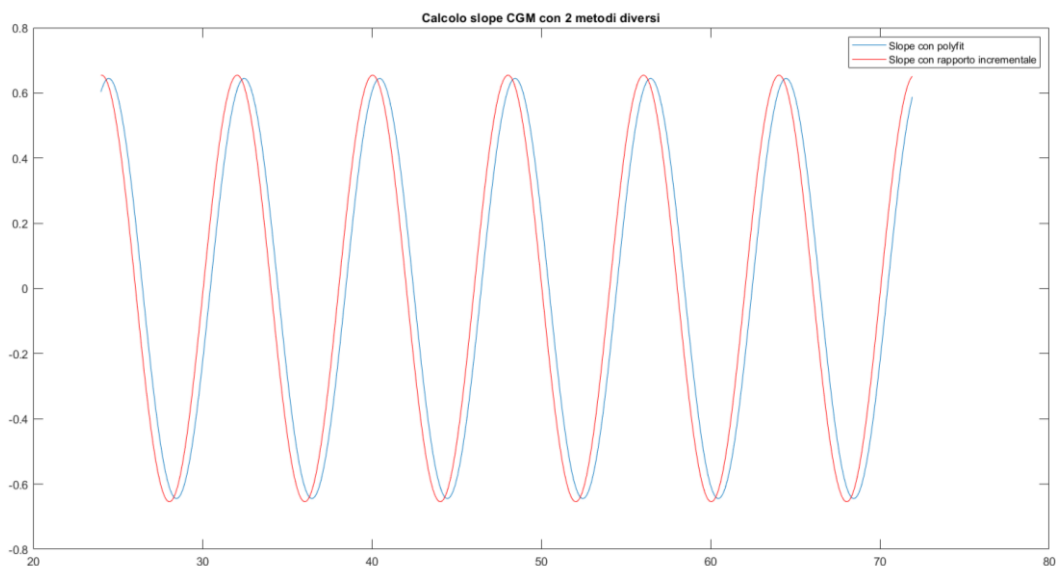
**Figura 3.3** Rappresentazione del segnale di glucosio (CGM) in mg/dL, attraverso un segnale sinusoidale con relativi valori medi nelle due finestre mobili, LW e SW, e metrica di guasto del glucosio (GFM) confrontata con la relativa soglia pari a 100 mg/dL\*h.

### 3.4.2 Glucose Slope (GS)

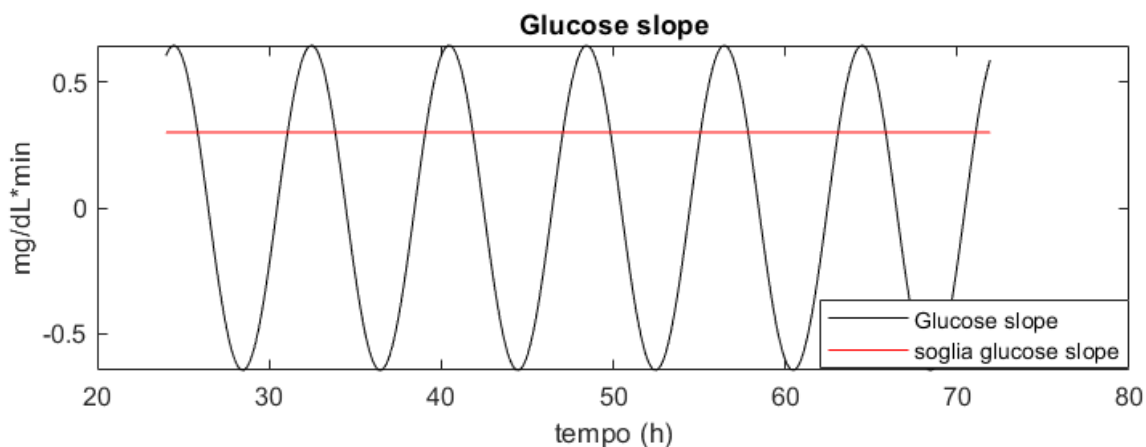
La seconda metrica di guasto legata al glucosio, la Glucose Slope (GS), è rappresentata dal tasso di crescita del segnale di glucosio (CGM). Il tasso di crescita è stato calcolato come la derivata della funzione da esaminare. Per fare ciò sono stati utilizzati due metodi differenti e confrontati. Per il primo metodo è stata utilizzata la definizione di derivata, ovvero il limite del rapporto incrementale per  $\Delta t$  che tende a zero. È stata calcolata la differenza tra due campioni successivi,  $CGM_k$  e  $CGM_{k-1}$ , e divisa per l'intervallo di tempo  $\Delta t$  pari a 5 minuti,  $\frac{1}{60}$  di ora. La formula è riportata di seguito, analogamente al paragrafo 2.4.3.

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

Nel secondo metodo è stata utilizzata una funzione fornita dall'ambiente di lavoro MATLAB. Tale funzione è denominata *polyfit*, riceve in ingresso tre parametri  $x$ ,  $y$  e  $n$ , e restituisce i coefficienti di un polinomio di grado  $n$  che costituisce il miglior adattamento dei valori in  $y$ . È stato così calcolato il coefficiente angolare della retta che collega ogni coppia di punti appartenenti al segnale CGM, ovvero la derivata del nostro segnale. Come coppia di punti sono stati scelti valori a distanza di un ora, così da eseguire la linearizzazione di 12 campioni per volta, corrispondenti a  $CGM_k$  e  $CGM_{k-11}$ . Il secondo metodo ha permesso un calcolo della pendenza del glucosio più preciso rispetto al primo, il quale utilizza un metodo più approssimativo. Il confronto è illustrato in figura 3.4. In figura 3.5 è rappresentata la Glucose slope calcolata con la funzione *polyfit* e confrontata con la relativa soglia.



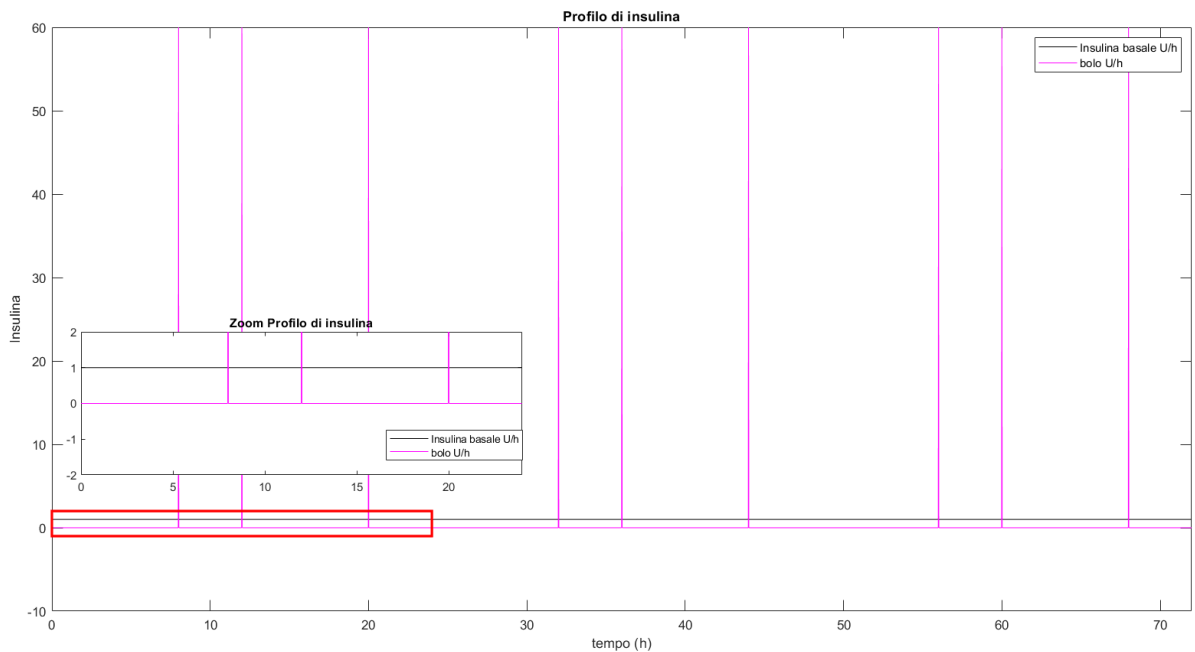
**Figura 3.4** Rappresentazione del confronto tra i due metodi utilizzati per il calcolo della Glucose Slope (GS). In blu la derivata ottenuta con la funzione polyfit, in rosso ottenuta con la definizione (formula riportata anche nel paragrafo 2.4.3)



**Figura 3.5** Illustrazione Glucose Slope calcolata con polyfit confrontata con la relativa soglia.

### 3.4.3 Insulin fault metric (IFM)

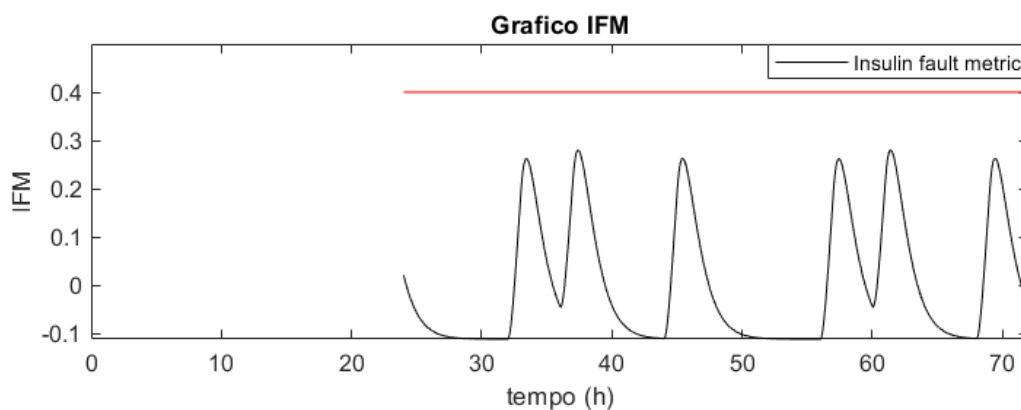
Per il calcolo della metrica di guasto legata all'insulina è necessario conoscere il profilo di insulina totale somministrata al paziente, costituita dalla somma tra bolo di insulina e insulina basale. Come descritto nel paragrafo 3.3 dell'elaborato, sono stati simulati tali segnali con un passo di campionamento pari ad 1 minuto. Nella figura 3.6 è possibile notare un segnale costante pari ad 1 U/h, rappresentante l'insulina basale e gli impulsi relativi alle somministrazioni periodiche del bolo di insulina, del valore di 60 U/h.



**Figura 3.6** Illustrazione del profilo di insulina costituito dall'insulina basale e il bolo di insulina somministrato ad ora prestabiliti.

Sviluppato il profilo di insulina, è stato possibile calcolare il Plasma Insulin Estimate (PIE) attraverso il modello compartimentale definito nel paragrafo 2.4.2, e successivamente, il valore medio all'interno delle due finestre mobili. Analogamente al calcolo del GFM, è stata utilizzata la funzione *calc\_Avg*, al fine di snellire la struttura del programma. Una volta calcolati i valori medi del PIE nelle due finestre,  $PIE_{SW}$  e  $PIE_{LW}$ , è stata calcolata la metrica di guasto legata all'insulina (IFM), riportata in figura 3.7 e confrontata con la relativa soglia.

Come si può notare in figura, utilizzando un segnale simulato, la metrica di guasto dell'insulina è sempre al di sotto della soglia, per cui non verrà emesso nessun allarme in questa analisi simulata.



**Figura 3.7** Illustrazione Insulin Fault Metric (IFM) confrontata con la relativa soglia.

## 3.5 Segnalazione guasto

Per la segnalazione di un guasto nel set di infusione è necessario che tutte e tre le metriche siano al di sopra della relativa soglia nello stesso istante. Al fine di verificare tale condizione sono stati creati due array, uno per il glucosio e uno per l'insulina, che tengono in memoria se in un dato istante la soglia è stata sorpassata o meno. L'array relativo al glucosio assume valore 1 se sia la GFM che la GS sono al di sopra della soglia nello stesso istante, assume valore 0 se almeno una delle due condizioni non è verificata. L'insulina, a differenza del segnale CGM, presenta un passo di campionamento pari ad 1 minuto. È necessario quindi utilizzare un secondo array, il quale tenga in memoria il superamento della soglia da parte della sola IFM, infatti quando si rileva un valore di glucosio attraverso il sensore, sono stati già rilevati 5 valori di insulina. All'interno dell'algoritmo viene analizzata una finestra di 5 minuti e se un solo valore di IFM supera la soglia, allora in quella finestra l'array assume valore 1, altrimenti 0. I due array sono stati confrontati tra di loro, eseguendo un'operazione di *AND*, e il risultato è stato memorizzato all'interno di un nuovo vettore chiamato *FAULT*. L'allarme è segnalato se in un dato istante il vettore *FAULT* ha valore 1, ovvero tutte e tre le soglie sono state superate contemporaneamente.

Con la segnalazione dei guasti e la verifica del funzionamento del programma, la struttura principale dell'algoritmo è stata sviluppata ed è ora possibile, passare all'implementazione con dati reali.

## 3.6 Implementazione algoritmo con dati generati dal simulatore UVa/Padova

### 3.6.1 Dati utilizzati

Sviluppato l'algoritmo e testato il funzionamento, è stato possibile procedere con l'implementazione dei dati generati attraverso il simulatore UVa/Padova. Il simulatore ha generato un dataset di 5 pazienti dotati di un sistema di pancreas artificiale, il quale raccoglie le misurazioni della concentrazione di glucosio nel sangue e dell'insulina, in un periodo totale di trenta giorni, con un passo di campionamento di cinque minuti, come previsto nei convenzionali sistemi di pancreas artificiali.

La terapia dei pazienti consiste in una quantità variabile di insulina basale, che permette di mantenere i livelli glicemici in un range di valori quasi normali, e un bolo somministrato durante il pasto. All'interno del database vengono fornite le informazioni sui carboidrati consumati, ma non verranno utilizzate all'interno dell'algoritmo.

Il file generato dal simulatore è un file CSV, le colonne separano la tipologia di dati, mentre le righe indicano gli orari precisi di rilevazione. Raggiunta la visualizzazione di tutti i dati mensili di un paziente si passa al paziente successivo. Nell'algoritmo non sono stati utilizzati tutti i dati generati dal simulatore, ma solo quelli utili al fine del calcolo delle metriche di guasto e per la segnalazione degli allarmi.

All'interno del dataset, i guasti avvengono a mezzanotte o a mezzogiorno, e hanno una durata di 6 ore, questa conoscenza permette una semplificazione nella realizzazione dell'algoritmo senza intaccare la qualità dei risultati raggiunti.

### 3.6.2 Struttura del programma

La struttura principale del programma, ed il suo funzionamento, sono rimaste pressoché invariate a quelle descritte nel paragrafo 3.4. Tuttavia, avendo un dataset fornito dal simulatore UVa/Padova, è stato necessario eseguire alcune modifiche al fine di adattare l'algoritmo al nuovo set di dati. Siccome questo algoritmo è basato sul database generato dal simulatore UVa/Padova, l'utilizzo di un simulatore differente potrebbe richiedere ulteriori modifiche all'algoritmo.

Anzitutto è stato importato il file CSV contenente il dataset generato, ogni colonna della tabella importata è stata convertita in un vettore riga così da facilitarne l'analisi degli elementi. L'algoritmo individua autonomamente il passo di campionamento delle rilevazioni ed il numero totale di pazienti. Ogni paziente è identificato da un numero intero e disposto in ordine crescente all'interno della tabella. Il programma esegue automaticamente l'analisi dei parametri di ogni paziente, rappresentando i grafici delle metriche di guasto e segnalando le presenze di guasto. Inoltre, terminata l'analisi di tutti i pazienti, effettua il calcolo delle prestazioni dell'algoritmo stesso.

Prima di eseguire il calcolo delle metriche di guasto, il programma esegue un'interpolazione dei valori di insulina, passando da un tempo di campionamento di 5 minuti ad un tempo di campionamento di 1 minuto. Questo passaggio è necessario in quanto le variazioni di insulina sono più rapide rispetto a quelle di glucosio, come descritto nell'articolo di Howsmon et al. [9].

Analizzando il segnale CGM generato dal simulatore è stato possibile notare che il sensore di glicemia impiegato presenta un valore di saturazione pari a 399 mg/dL, oltre il quale le misurazioni non sono più accurate. Se il segnale rileva tale valore, è importante segnalare immediatamente al paziente che si trova in una grave condizione di iperglicemia. Questo accorgimento è stato necessario, in quanto tale situazione, ha spesso portato alla segnalazione di falsi negativi in fase di progetto. Il segnale CGM, saturando a 399 mg/dL, può presentare dei tratti costanti in corrispondenza del valore di saturazione e, nel calcolo della Glucose Slope, comporta una derivata nulla. Questo risultato esclude la segnalazione di qualsiasi allarme e porta alla presenza di falsi negativi. Per risolvere il problema è stata aggiunta una clausola aggiuntiva per segnalazione di un guasto nel caso si fossero raggiunti i 399 mg/dL. Successivamente a questa modifica è stato riscontrato un notevole miglioramento della precisione e del numero di falsi positivi giornalieri.

I valori di default impostati nell'algoritmo sono i medesimi descritti nell'articolo citato precedentemente, tuttavia è possibile modificarli a discrezione dell'utente:

- Ampiezza finestra breve SW = 1 h
- Ampiezza finestra lunga LW = 24 h
- Soglia GFM = 100 mg/dL\*h
- Soglia IFM = 0,4
- Soglia GS = 0,3 mg/dL\*min
- Passo di campionamento glucosio = 5 minuti (12 campioni all'ora)
- Passo di campionamento insulina = 1 minuto (60 campioni all'ora)

Gli ultimi due parametri non sono modificabili in quanto ricavati automaticamente dalla colonna del tempo nel dataset.

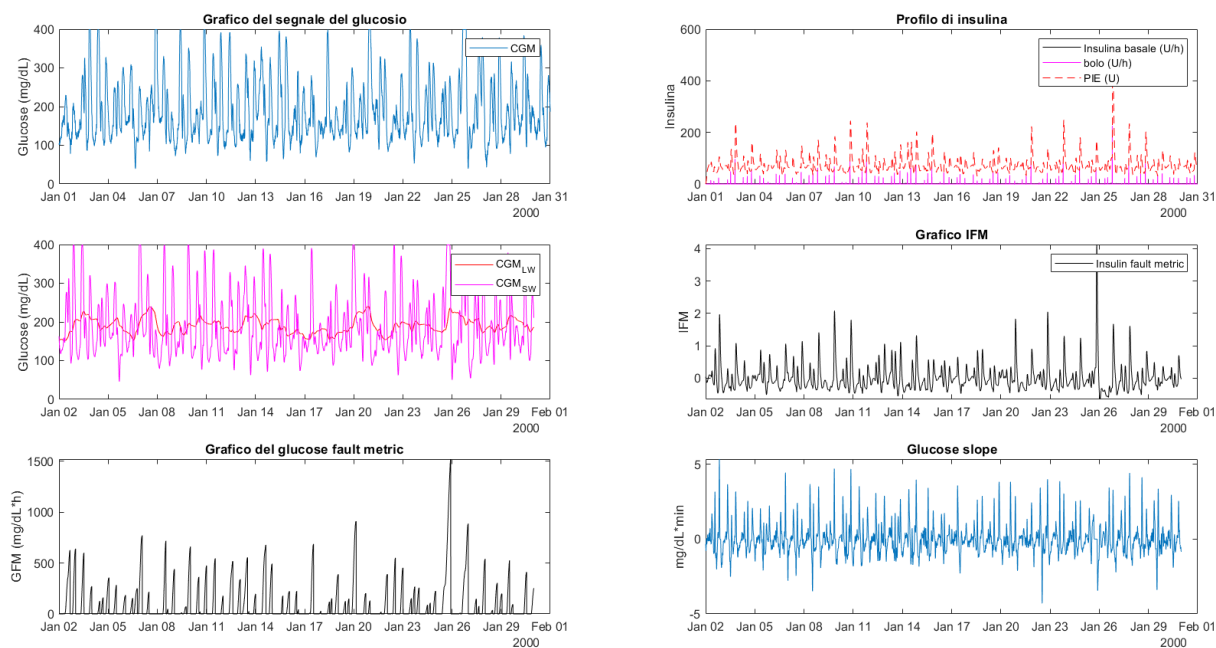
### 3.6.3 Grafici

L'algoritmo, dopo aver raccolto i dati ed elaborato gli allarmi, rappresenta i grafici di ogni paziente. I grafici rappresentati contengono: le acquisizioni CGM e le medie nelle due finestre mobili, la GFM, la PIE e il profilo di insulina con un relativo zoom nelle prime 24 ore, la IFM, la Glucose Slope, le tre metriche di guasto confrontate con le relative soglie e le segnalazioni di guasto. Quest'ultimo rappresenta i guasti individuati dall'algoritmo come degli impulsi diretti nel verso positivo delle ordinate, mentre i guasti reali, ottenuti dal simulatore, sono indicati verso il basso. I grafici si basano su N-

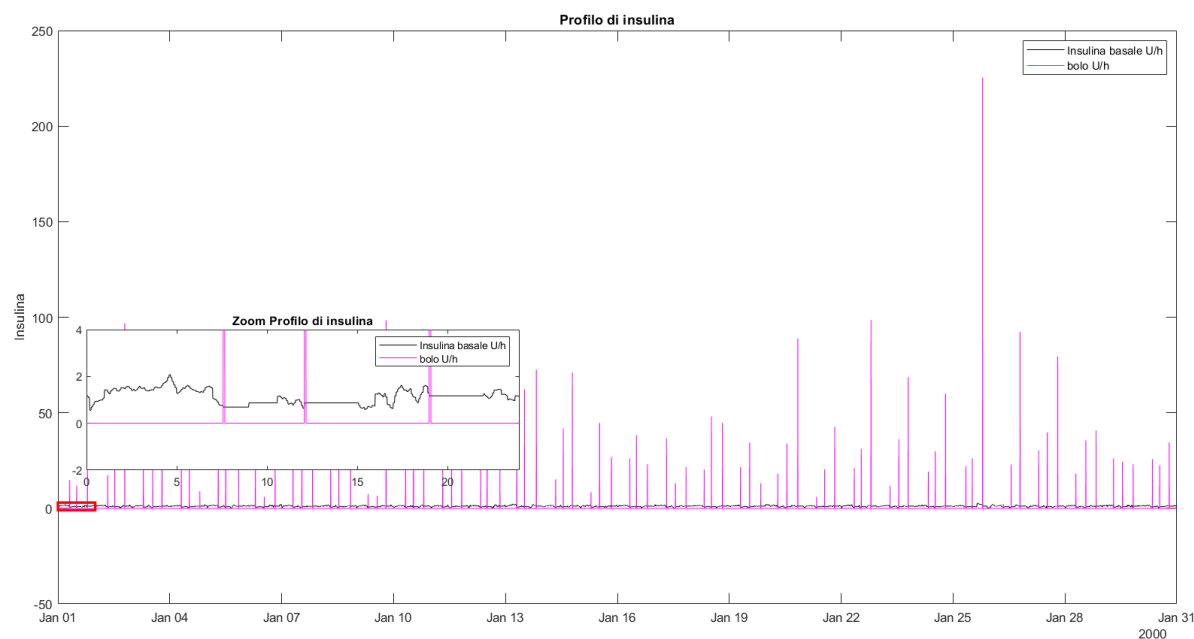
1 giorno di analisi (in questo caso 29), ad eccezione del grafico del glucosio e dell'insulina, i quali vengono rappresentati su tutti gli N giorni.

I grafici vengono rappresentati un paziente per volta e non è possibile vedere le rappresentazioni di più pazienti simultaneamente.

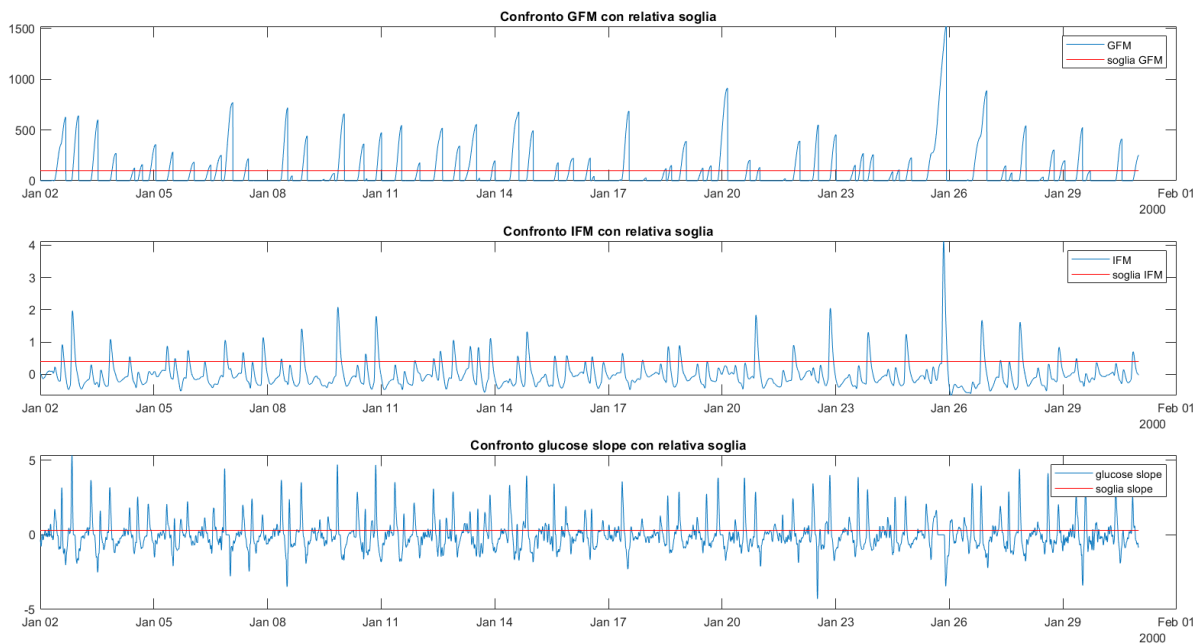
Terminata la rappresentazione, l'algoritmo calcola sensibilità, specificità, precisione e falsi positivi giornalieri e li restituisce in *output* come messaggio.



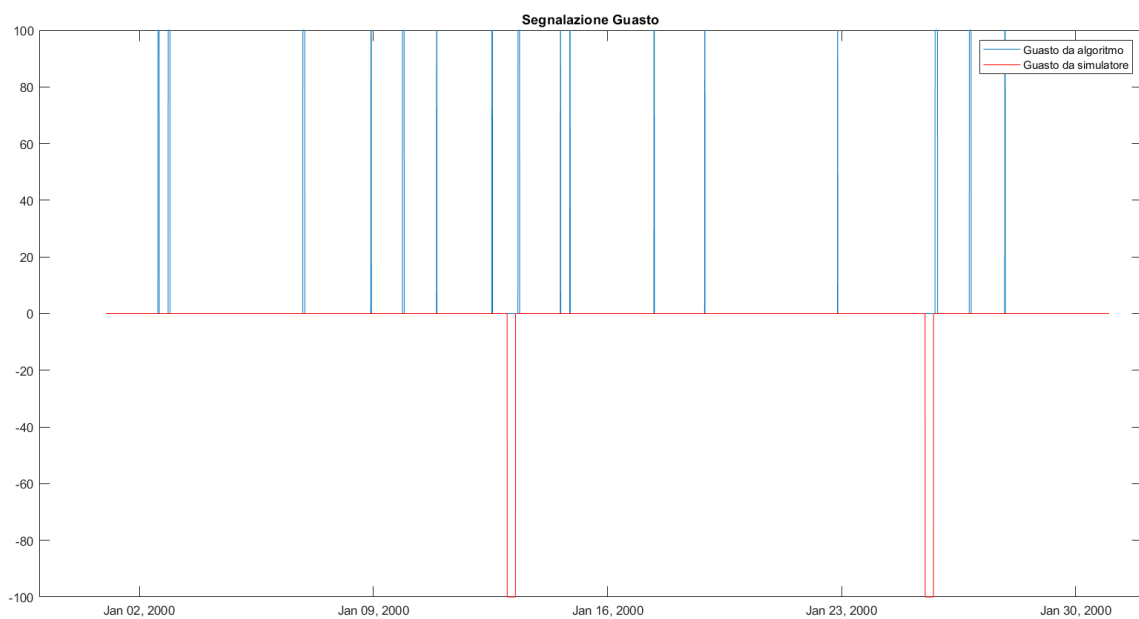
**Figura 3.8** Grafici relativi al paziente numero 1



**Figura 3.8** Profilo di insulina con relativo zoom nelle prime 24 ore del paziente 1



**Figura 3.9** Confronto metriche di guasto con relative soglie del paziente 1



**Figura 4** Rappresentazione delle segnalazioni di guasto e dei guasti simulati del paziente 1

### 3.7 Valutazione delle prestazioni

Gli allarmi vengono aggiornati ogni 5 minuti e se tutte e tre le soglie di guasto sono superate contemporaneamente avviene la segnalazione di guasto.

Per valutare l'efficacia dell'algoritmo sono state predisposte delle finestre di analisi della durata di 6 ore, ovvero la stessa durata dei guasti simulati. Tali guasti iniziano a mezzanotte o a mezzogiorno per cui sono interamente contenuti all'interno di un'unica

finestra. L'obiettivo è che l'algoritmo riesca a far scattare l'allarme nella finestra in cui è presente il guasto. Il programma suddivide quindi il periodo di analisi (29 giorni) in sotto intervalli della durata di 6 ore.

A tale scopo sono stati predisposti due vettori, ogni cella rappresenta una finestra. Il primo individua le finestre all'interno delle quali è presente il guasto simulato, assumendo valore 1 se il guasto è presente altrimenti 0, il secondo vettore è ottenuto dall'elaborazione dell'algoritmo e indica le finestre all'interno delle quali è stato emesso almeno un allarme di guasto, assumendo valore 1 se è stato segnalato un guasto. I due vettori vengono poi confrontati per valutare le prestazioni dell'algoritmo.

Per il calcolo delle performance dell'algoritmo ad ogni finestra è stata attribuita una di queste quattro etichette:

- Vero positivo (TP): Se all'interno della finestra è presente il guasto simulato e l'algoritmo ha emesso un segnale d'allarme.
- Falso positivo (FP): Se all'interno della finestra non è presente il guasto ma l'algoritmo ha emesso un segnale d'allarme. Tuttavia, se nella precedente finestra è presente un vero positivo, il falso positivo non viene conteggiato. Questa convenzione è stata adottata in quanto si presume che, nelle 6 ore successive al guasto nel set di infusione, il paziente sia ancora in una condizione di anomalia ed è quindi ragionevole supporre che i parametri non siano nella norma.
- Vero negativo (TN): Se all'interno della finestra non è presente un guasto e l'algoritmo non ha emesso un segnale dall'allarme
- Falso negativo (FN): Se all'interno della finestra è presente un guasto ma l'algoritmo non ha emesso un segnale d'allarme

Infine, per valutare le prestazioni dell'algoritmo sono state calcolate la sensibilità e il numero di falsi positivi giornalieri per tutti i pazienti ottenuti dal dataset. La sensibilità è il rapporto tra il numero di veri positivi e il numero di guasti totali.

$$\text{Sensibilità} = \frac{TP}{TP + FN}$$

I falsi positivi giornalieri sono calcolati come la somma dei falsi positivi normalizzati con il numero di giorni di analisi. L'algoritmo ha riscontrato una sensibilità del 30% rilevando correttamente 3 guasti con 0,2533 FP/giorno. Confrontando i valori ottenuti con quelli trovati da Howsmon e i suoi colleghi, i quali hanno riscontrato una sensibilità

del 73% con 0,27 FP/giorno, è possibile notare una sensibilità considerevolmente peggiore, ma con meno falsi positivi giornalieri. Questa differenza è dovuta al fatto che, in alcune circostanze, l'Insulin Fault Metric è al di sotto della relativa soglia per tutta la durata del guasto. Infatti, abbassando la soglia a 0,3 l'algoritmo ha riscontrato una sensibilità del 60% con 0,2867.

Le performance potrebbero migliorare ulteriormente se l'algoritmo fosse testato su un dataset di più pazienti. La limitazione a 5 pazienti fornisce un numero ristretto di guasti simulati, con un numero maggiore di guasti da individuare l'algoritmo potrebbe essere testato in maniera più efficiente e ottenere una sensibilità maggiore.

# Capitolo 4

## Conclusioni

Alla luce dei risultati ottenuti è possibile dire che l'implementazione dell'algoritmo di Howsmon et al. nel presente elaborato ha riportato prestazioni peggiori, in quanto, le soglie per la segnalazione dei guasti non sono ottimali. Apportando delle modifiche alle soglie prefissate, sono stati riscontrati dei risultati promettenti, sia per la sensibilità sia per i falsi positivi. Il dataset su cui è stato testato l'algoritmo si basa su pazienti che utilizzano un pancreas artificiale, per il quale gli autori originali esprimono una necessità di ulteriori e più approfondite ricerche. In ogni caso i risultati sono promettenti, ma sono necessari ulteriori studi al fine di massimizzare le prestazioni e ridurre gli errori.



# Bibliografia

1. Bequette, B.W. “Challenges and recent progress in the development of a closed-loop artificial pancreas”, *Annu. Rev. Control* **2012**, 36, 255–266.  
doi:10.1016/j.arcontrol.2012.09.007.
2. Peyser, T.; Dassau, E.; Breton, M.; Skyler, J.S. “The artificial pancreas: Current status and future prospects in the management of diabetes”, *Ann. N.Y. Acad. Sci.* **2014**, 1311,102–123. doi: 10.1111/nyas.12431.
3. Baysal, N.; Cameron, F.; Buckingham, B.A.; Wilson, D.M.; Chase, H.P.; Maahs, D.M.; Bequette, B.W. “A Novel Method to Detect Pressure-Induced Sensor Attenuations (PISA) in an Artificial Pancreas”, *J. Diabetes Sci. Technol.* **2014**, 8,1091–1096. doi: 10.1177/1932296814553267.
4. Cameron, F.; Bequette, B.W.; Wilson, D.M.; Buckingham, B.A.; Lee, H.; Niemeyer, G. “A closed-loop artificial pancreas based on risk management”, *J. Diabetes Sci. Technol.* **2011**, 5, 368–379. doi: 10.1177/193229681100500226.
5. Cameron, F.; Niemeyer, G.; Wilson, D.M.; Bequette, B.W.; Benassi, K.S.; Clinton, P.; Buckingham, B.A. “Inpatient Trial of an Artificial Pancreas Based on Multiple Model Probabilistic Predictive Control with Repeated Large Unannounced Meals”, *Diabetes Technol. Ther.* **2014**, 16, 728–734. doi: 10.1089/dia.2014.0093.
6. Ly, T.T.; Maahs, D.M.; Messer, L.; DeSalvo, D.; Payne, S.; Clinton, P.; Block, J.; Buckingham, B.A. “Duration of Infusion Set Survival in Lipohypertrophy versus Nonlipohypertrophied Tissue in Patients with Type 1 Diabetes”, *Diabetes Technology and Therapeutics.* **2016**. doi: 10.1089/dia.2015.0432.
7. Patel, P.J.; Benasi, K.; Ferrari, G.; Evans, M.G.; Shanmugham, S.; Wilson, D.M.; Buckingham, B.A. “Randomized Trial of Infusion Set Function: Steel Versus Teflon”, *Diabetes Technol. Ther.* **2014**, 16, 15–19. doi: 10.1089/dia.2013.0119.
8. DeSalvo, D.J.; Ly, T.T.; Maahs, D.M.; Messer, L.H.; Pyle, L.; Wadwa, P.; Westfall, E.; Hanes, S.; Buckingham, B.A. “Duration of Infusion Set Survival with Hyaluronidase Injected at Insulin Pump Infusion Site”, *Diabetes*: Boston, MA, USA, **2015**.

9. 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: 10.3390/s17010161.
10. C. Dalla Man, F. Micheletto, D. Lv, M. Breton, B. Kovatchev e C. Cobelli, “*The UVA/PADOVA Type 1 Diabetes Simulator*”, *J. Diabetes Sci Technol.*, **2014**. doi: 10.1177/1932296813514502.
11. C. Vannini. “*Insulina, l’educazione del paziente per la somministrazione*”. Nurse24.it, ultima modifica: novembre **2018**.  
<https://www.nurse24.it/dossier/diabete/insulina-tipologie-come-si-somministra.html>
12. Ypsomed. “*Insulina attiva e terapia multi-iniettiva giornaliera*”. Mylife-diabetescare, ultima modifica: ottobre **2023**. <https://www.mylife-diabetescare.com/it-CH/sul-diabete/terapia-con-penna/insulina-attiva-e-terapia-multi-iniettiva-giornaliera.html>
13. 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*, **2019**. <https://doi.org/10.1016/B978-0-12-815655-1.00016-8>.
14. L. Meneghetti, M. Terzi, S. Del Favero, G. A. Susto and C. Cobelli, “*Data-Driven Anomaly Recognition for Unsupervised Model-Free Fault Detection in Artificial Pancreas*,” *IEEE Transactions on Control Systems Technology*, **2020**. doi: 10.1109/TCST.2018.2885963.
15. C. Cobelli, C. D. Man, G. Sparacino, L. Magni, G. D. Nicolao e B.P. Kovatchev, “*Diabetes: models, signals and control*”, *IEEE Rev. Biomed. Eng.*, **2009**. doi: 10.1109/RBME.2009.2036073.
16. *Diabetes Atlas, tenth edition*. International Diabetes Federation, **2021**, ISBN: 978-2-930229-98-0.
17. IRCCS Istituto Clinico Humanitas. “*Diabete*”. Humanitas.it, consultato il 5 agosto **2024**. <https://www.humanitas.it/malattie/diabete/>
18. Istituto Superiore di Sanità. “*Diabete*”. Epicentro.iss.it, novembre **2022**. <https://www.epicentro.iss.it/diabete/>
19. Diabetes Research Institute. “*Le cellule beta pancreatiche rispondo all’infiammazione: implicazioni nello sviluppo del diabete di tipo 1*”. Dri.hsr.it, novembre **2019**. <https://dri.hsr.it/news/le-cellule-beta-pancreatiche-rispondono->

all'infiammazione-implicazioni-nello-sviluppo-del-diabete-tipo-1/#:~:text=Nel%20diabete%20di%20tipo%201,insulina%20e%20controllare%20la%20glicemia

20. Diabetes Research Institute. “*Un errore di imballaggio può causare gli attacchi del sistema immunitario nel diabete di tipo 1*”, dri.hsr.it, consultato il 5 agosto **2024**.

<https://dri.hsr.it/news/un-errore-imbollaggio-puo-causare-gli-attacchi-del-sistema-immunitario-nel-diabete-tipo-1/>

21. S. Stabellini. “*Chetoacidosi diabetica*”, Nurse24.it, ultima modifica: 19 agosto **2020**. <https://www.nurse24.it/studenti/patologia/chetoacidosi-diabetica.html>

22. Ascensia Diabetes Care. “*Le complicanze croniche del diabete*”. Diabete.net, consultato il 5 agosto **2024**. <https://www.diabete.net/le-complicanze-croniche-del-diabete/conoscere-il-diabete/complicanze/31297/>

23. I. Savulescu. “*Ipoglicemia: che cos'è e come trattarla*”. Grupposandonato.it, settembre **2020**. <https://www.grupposandonato.it/news/2020/settembre/ipoglicemia-cose-come-riconoscerla-trattarla>

24. Redazione diabete.com. “*Ipoglicemia: cause, tipi, rischi*”. Diabete.com, marzo **2015**. <https://www.diabete.com/ipoglicemia-cause-tipi-rischi/>

25. Istituto Superiore di Sanità. “*Aspetti epidemiologici nel mondo*”. Epicentro.iss.it, ultima modifica giugno **2022**. <https://www.epicentro.iss.it/diabete/epidemiologia-mondo>

26. Istituto Superiore di Sanità. “*Aspetti epidemiologici in Italia*”. Epicentro.iss.it, ultima modifica giugno **2022**. <https://www.epicentro.iss.it/diabete/epidemiologia-italia#:~:text=In%20Italia%2C%20in%20base%20ai,tra%20le%20persone%20ultra%2075enni>

27. Redazione diabete.com. “*I numeri del diabete in Italia*”. Diabete.com, novembre **2022**. <https://www.diabete.com/impatto-del-diabete-italia/>

28. Diabeloop. “*Dispositivo per il trattamento del diabete: penna da insulina o microinfusore?*”. Dbl-diabete.it, maggio **2023**. <https://www.dbl-diabete.it/tutto-sul-diabete/abc-del-diabete/trattamento/dispositivo-per-il-trattamento-del-diabete-penna-da-insulina-o-microinfusore>

29. Sanità24. “*Avogaro (Sid): con il pancreas artificiale si apre una nuova era per contrastare il diabete*”. Sanita24.ilsole24ore.com, gennaio **2024**.

<https://www.sanita24.ilsole24ore.com/art/medicina-e-ricerca/2024-01-08/avogaro-sid-il-pancreas-artificiale-si-apre-nuova-era-contrastare-diabete-163517.php?uuid=AFiASQHC>

30. Redazione diabete.com. “*6 domande sul pancreas artificiale*”. Diabete.com, gennaio **2021**. <https://www.diabete.com/6-domande-sul-pancreas-artificiale/>

31. S. Maltoni, A. Negro, F. Trimaglio, L. Vignatelli, L. Ballini. “*Dispositivi medici innovativi nella gestione del diabete*”, Short Report n. 6, Agenzia Sanitaria e Sociale Regionale, Regione Emilia-Romagna. Bologna, Ottobre **2012**

32. Coffin, Marie, and Shashikala Sukhatme. “*Receiver Operating Characteristic Studies and Measurement Errors.*” *Biometrics*, vol. 53, no. 3, 1997, pp. 823–37. *JSTOR*, <https://doi.org/10.2307/2533545>. Consultato il 5 agosto **2024**.

# Appendice A

## % Algoritmo per rilevazione LISA

```
clear all
close all
clc

% Parametri: giorni totali, giorno di partenza per l'analisi, ore in un
% giorno, minuti in un ora, lunghezza LW e SW. Le prime 23 ore e 55 min
% fanno parte del primo giorno, la mia analisi parte dalle ore 24. Un
% giorno è composto da 288 campioni, 24 ore *12 campioni/ora
gg=30; % giorni di rilevazione del glucosio tramite CGM
ga=2; % giorno attuale di partenza per l'analisi dei dati
h=24; % numero di ore in un giorno
min=60; % numero minuti in un'ora
L_LW=24;% definisco la lunghezza della LW in ore
L_SW=1; % definisco la lunghezza della SW in ore
% Leggo la tabella contenete i dati ottenuti attraverso il simulatore, il
% quale genera dati su 30 giorni per 5 pazienti differenti.
dataset = readtable('simulator_dataset_2018_rng1.csv', 'VariableNamingRule',
'preserve');
% Le colonne che mi servono sono: time, CGM, bolus,
% basal, fault pump e ID.
% Effettuo la conversione in vettore attraverso il comando table2array
% che restituisce un vettore colonna, effettuo la trasposizione
% tramite "."
time=table2array(dataset(:,1)).'; %importo l'intera colonna dei tempi per
% il calcolo di Ts
Ts=minutes(time(2)-time(1))/min; % periodo di campionamento dei segnali
% pari a 5 min, ricavato dal vettore time
ID=table2array(dataset(:,14)).'; % importo colonna ID paziente
max_Paziente=max(ID); % numero massimo di pazienti
cicli_Paz=1; % Paziente da cui partire per analisi
TP_tot=0; % Conteggio guasti rilevati correttamente

% Per effettuare analisi del solo paziente 1 aggiungere "%" DA RIGA 45 e
% da 478 a 490
```

```

% Definisco un vettore contenente le prestazioni dell' algoritmo,
% il primo valore sono i FP totali, il secondo la sensibilità, il terzo
% la specificità e il quarto la precisione.
% Imposto un ciclo while per analizzare tutti i pazienti, dal primo
% all'ultimo in ordine crescente

performance=[0 0 0 0];

while cicli_Paz<=max_Paziente
    disp(['Paziente numero ', num2str(cicli_Paz)])
    Paziente=cicli_Paz; % scelgo il paziente da ricercare e prendere come
                        % riferimento per l'analisi.
    for i=1:length(ID)
        if ID(i)==Paziente
            rif_min=i; % la prima volta che trovo l'ID corretto sarà
                       % la prima riga da utilizzare
            break
        end
    end
    rif_max=rif_min+h*gg/Ts-1; % per trovare l'ultima riga appartenente
                              % ad un paziente, aggiungo 8639=24*30*12-1,
                              % la distanza tra il primo campione di un
                              % paziente e l'ultimo dello stesso paziente.
    time=table2array(dataset(rif_min:rif_max,1)).'; % tempo di analisi da
                                                    % 1/1/2000 0:00 a
                                                    % 30/1/2000 23:55
    CGM=table2array(dataset(rif_min:rif_max,2)).'; % segnale glucosio
                                                    % con Ts=5 min

    % I segnali basal e bolus nella tabella hanno un tempo di campionamento
    % pari a 5 minuti, devo effettuare una conversione attraverso un
    % interpolazione per avere un Ts=1 min
    basal=table2array(dataset(rif_min:rif_max,4)).';
    bolus=table2array(dataset(rif_min:rif_max,5)).';
    % fault_pump se -100 è presente un guasto
    fault_pump=table2array(dataset(rif_min:rif_max,11)).';

    % creo i vettori contenenti le medie del segnale CGM nelle due
    % finestre mobili di confronto
    CGM_LW=[]; % vettore long window
    CGM_SW=[]; % vettore short window

```

```

% Utilizzo la Funzione calc_AVG. Calcola il valore medio di un segnale
% s, all'interno di una finestra L, per un numero di giorni pari a gg
CGM_LW=calc_Avg(gg,L_LW,CGM);
CGM_SW=calc_Avg(gg,L_SW,CGM);

% plot valore medio CGM nelle due finestre
% definisco il periodo di plot dei segnali a partire dal secondo giorno
tp_1=time((h*(1/Ts)+1):(gg*h*(1/Ts)));
f1=figure(1);
f1.Position=[20 350 625 400]; % i primi due valori è la posizione
                                % dall'angolo in basso a sx, gli ultimi
                                % due valori sono la
                                % dimensione della finestra

subplot(3,2,1);
plot(time,CGM);
legend('CGM');
title('Grafico del segnale del glucosio')
ylabel('Glucose (mg/dL)');
subplot(3,2,3);
plot(tp_1,CGM_LW,'r',tp_1,CGM_SW,'m');
legend('CGM_L_W','CGM_S_W'); % L'underscore mi scrive in pedice il
                                % carattere successivo

ylabel('Glucose (mg/dL)');

% Calcolo del glucose fault metric GFM creo due vettori AUC e GFM
% contenenti solo zeri, della stessa lunghezza
% del vettore CGM medio nelle due finestre di osservazione

if isequal(size(CGM_SW),size(CGM_LW))
    AUC=zeros(size(CGM_LW));
    GFM=zeros(size(CGM_LW));
else
    display('I vettori del CGM medio nella LW e nella SW sono di lunghezza diversa');
end

% inizio ad analizzare i valori del CGM medio nelle due finestre, partendo
% dal secondo campione fino all'ultimo,il numero di campioni sarà dato
% dal numero di giorni di analisi escluso il primo giorno. Parto dal
% secondo campione e non dal primo poiche il primo valore del GFM sarà
% sicuramente 0 non avendo dati precedenti

```

```

for i=1:((gg-1)*h*1/Ts)
    % calcolo l'area tra le curve del CGM medio nella LW e nella SW
    % moltiplicando per 5 minuti, il GFM ha come le ore.
    AUC(i)=(CGM_SW(i)-CGM_LW(i)).*(Ts);
    if i==1
        GFM(i)=0;
        % se il rapporto all'istante i è maggiore di 1, il GFM a
        % quell'istante sarà dato da quello precedente i-1 sommato con
        % l'area AUC nel dato istante. Se minore di 1, il GFM viene
        % resettato a 0
    elseif CGM_SW(i)/CGM_LW(i) > 1
        GFM(i)=GFM(i-1)+AUC(i);
    end
end
% plot del GFM
figure(1);
subplot(3,2,5);
plot(tp_1,GFM,'k');
ylabel('GFM (mg/dL*h)');
title('Grafico del glucose fault metric');

% Profilo di insulina, costituito da un segnale costante di valore basso,
% l'insulina basale, e dal bolo di insulina somministrato a determinati
% orari, Tempo campionamento pari a 1 minuto
% Importo il vettore dei tempi con passo di campionamento pari a 1 minuto.
% 43200 campioni sarebbero 60 campioni/ora*24 ore*30 giorni.
% Il vettore dei tempi è stato creato su un file excel dal 1 gennaio
% 2000 al 30 gennaio 2000 con passo pari ad un minuto.

time_i=readtable('time_i.xlsx', 'VariableNamingRule', 'preserve');
time_i=table2array(time_i(1:43200,1)).';

Ts_i=minutes(time_i(2)-time_i(1))/min; % tempo di campionamento
    % pari a 1 minuto, 1/60 di ora
% creo i vettori dei tempi aventi con passo di campionamento 1 minuto
% e 5 minuti così da fare l'interpolazione del segnale di basal e bolus
% con Ts=5 minuti e il nuovo tempo di campionamento di 1 minuto. Questi
% vettori servono anche per fare lo zoom sul grafico

```

```

% NB:l'ultimo campione, quindi nel nostro caso l'ora 72, fa parte del
% giorno successivo. La lunghezza del vettore è di 8640 campioni,
% ovvero 24*1/Ts*gg.
t=[0:Ts:(gg*h)]; % vettore con passo 5 minuti
t=t(1:gg*h*1/Ts); % elimino l'ultimo elemento che fa parte del
                % giorno successivo
t_i=0:Ts_i:(gg*h); % vettore con passo 1 minuto
t_i=t_i(1:gg*h*1/Ts_i); % elimino l'ultimo elemento che fa parte
                        % del giorno successivo

% Converto basal e bolus che hanno un Ts di 5 min, in segnali che hanno
% un tempo di campionamento pari a 1 minuto attraverso un'interpolazione
basal_new=interp1(time,basal,time_i,'previous');
bolus_new=interp1(time,bolus,time_i,'previous');
% plotto il segnale dell'insulina basale e il bolo
figure(1)
subplot(3,2,2);
plot(time_i,basal_new,'k',time_i,bolus_new,'m');
title('Profilo di insulina');
ylabel('Insulina');
legend('Insulina basale U/h','bolo U/h')

% realizzo lo zoom sul grafico del profilo di insulina,
% nell'intervallo di ore 0-24
f2=figure(2);
f2.Position=[350 50 625 400];
plot(time_i,basal_new,'k',time_i,bolus_new,'m');
ax1=gca; % immagazzino le informazioni che riguardano gli assi
        % maggiori,mi servirà in seguito
title('Profilo di insulina');
legend('Insulina basale U/h','bolo U/h');
ylabel('Insulina');
hold on
% disegno un rettangolo
axes(ax1);
rectangle('Position',[0, -1, 1, 4],'EdgeColor','r','LineWidth',2);
% definisco la posizione dove disegnare i nuovi assi e
% disegno un riquadro
ax2=axes('Position', [.15 .3 .30 .20]);
box on

```

```

% seleziono la porzione di grafico da zoomare, da ora 0 a ora 24
% (campione 24*60)
t_zoom=t_i(1:(h*1/Ts_i));
bolus_zoom=bolus_new(1:(h*1/Ts_i));
basal_zoom=basal_new(1:(h*1/Ts_i));
plot(t_zoom,basal_zoom,'k',t_zoom,bolus_zoom,'m');
legend('Insulina basale U/h','bolo U/h');
title('Zoom Profilo di insulina');
ax2.XLim=[0 24];
ax2.YLim=[-2 4];

% Insulina totale è data dalla somma di bolo e insulina basale
Insulin=bolus_new+basal_new;

% Definisco il PIE, Plasma insulin estimate=X1, ovvero la quantità di
% insulina attiva nel plasma. X2 è l'insulina presente
% nel sottocute prima di entrare in circolazione nel plasma
% Definisco i due vettori aventi lunghezza uguale al vettore insulin,
% profilo di insulina
x_1=zeros(size(Insulin));
x_2=zeros(size(Insulin));
% Inizializzazione dei valori
x_1(1)=0;
x_2(1)=0;
% Calcolo valori tramite le equazioni riportate nell'elaborato
for i=1:(length(Insulin)-1)
    x_1(i+1)=0.98*x_1(i)+0.02*x_2(i);
    x_2(i+1)=0.98*x_2(i)+Insulin(i);
end
PIE=x_1;
% Calcolo le medie nelle finestre di analisi
PIE_SW=calc_Avg(gg,L_SW,PIE);
PIE_LW=calc_Avg(gg,L_LW,PIE);
% Calcolo Insulin fault metric
IFM=[];
for i=1:length(PIE_LW)
    IFM(end+1)=((PIE_SW(i))/(PIE_LW(i)))-1;
end
% definisco il tempo di plot dell'IFM,corrisponde al numero di
% giorni meno il primo,come per le finestre di analisi LW e SW, ma con

```

```

% passo 1 minuto
tp_2=time_i((h*(1/Ts_i)+1):(gg*h*(1/Ts_i)));
% Plot dei segnali
figure(1);
subplot(3,2,2);
plot(time_i,basal_new,'k',time_i,bolus_new,'m',time_i,PIE,'--r');
legend('Insulina basale (U/h)','bolo (U/h)','PIE (U)');
title('Profilo di insulina');
ylabel('Insulina');
subplot(3,2,4);
plot(tp_2,IFM,'k');
legend('Insulin fault metric');
title('Grafico IFM');
ylabel('IFM');

% Calcolo glucose slope, pendenza segnale glucosio CGM.
% Inizio il calcolo a partire dal secondo giorno, in quanto il
% segnale GFM e IFM sono attivi dal secondo giorno, per cui non avrei
% segnalazioni di allarme il primo giorno
GS=[]; % Vettore contenente il valore della derivata
for i=h/Ts+1:length(CGM)
    x=(t(i-11:i)).*60; % NB:il vettore dei tempi è in ore mentre la
    % soglia del glucose slope in mg/dL*min,
    % multiplico per 60 per convertire
    % Utilizzo i-11 poichè se sono al
    % campione x, devo analizzare
    % l'andamento del segnale di glucosio nei
    % 60 minuti precedenti al campione x,
    % ovvero 12 campioni

    y=CGM(i-11:i);
    p=polyfit(x,y,1);
    GS(end+1)=p(1);
end
figure(1)
subplot(3,2,6);
plot(tp_1,GS);
title('Glucose slope');
ylabel('mg/dL*min');

```

```

% Confronto dei segnali con relative soglie e segnalazione guasto
% tramite allarme, a partire dal secondo giorno di analisi
% SOGLIE ALLARMI
GFM_threshold=100; % mg/dL*h
IFM_threshold=0.4 ;
GS_threshold=0.3; % mg/dL*min
sensor_threshold=398; % massimo valore di glucosio rilevato dal segnale
% oltre il quale le misure non sono più accurate

% Creo vettori costanti per il plot delle soglie
GFM_cost=ones(size(GFM))*GFM_threshold;
IFM_cost=ones(size(IFM))*IFM_threshold;
GS_cost=ones(size(GS))*GS_threshold;
FAULT=[]; % vettore contenente le segnalazioni di allarme

% Dato che GFM e slope hanno un tempo di campionamento di 5 minuti
% mentre l'IFM ha un tempo pari ad un minuto, effettuo due cicli for
% diversi con passo diverso. Utilizzo due vettori, uno per ciclo for,
% nei quali assegno valore 1 se le soglie sono state superate, valore 0
% se invece i valori sono inferiori.
% Infine faccio una moltiplicazione tra i 2 vettori come fosse un AND,
% per individuare gli istanti in cui le soglie sono superate
% contemporaneamente e mandare un segnale di allarme
First_cond=[]; % superamento soglia GFM e slope
Second_cond=[]; % superamento soglia IFM
for i=1:(gg-1)*h*1/Ts % escludo il primo giorno di analisi
    % se il valore di glucosio è superiore alla soglia del sensore
    % bypasso il segnale di glucose slope per evitare falsi negativi

    if (GFM(i)>=GFM_threshold) & ...
        ((GS(i)>=GS_threshold) | (CGM(i+h/Ts)>sensor_threshold))
        First_cond(end+1)=100; % assegno 100 così moltiplicando per 1
            % il valore rimane 100 e posso
            % plottarlo assieme a fault_pump
    else
        First_cond(end+1)=0;
    end
end

% Essendo IFM campionato su 1 minuto, quando per il GFM ho un campione,
% per l'IFM ho 5 campioni, basta che uno di questi sia oltre la soglia.

```

```

% Passo=5 minuti ovvero il rapporto tra i tempi di campionamento
for i=1:(Ts/Ts_i):(gg-1)*h*1/Ts_i
    if ((IFM(i)>IFM_threshold) | (IFM(i+1)>IFM_threshold) | ...
        (IFM(i+2)>IFM_threshold) | (IFM(i+3)>IFM_threshold) | ...
        (IFM(i+4)>IFM_threshold))
        Second_cond(end+1)=1;
    else
        Second_cond(end+1)=0;
    end
end
FAULT=First_cond.*Second_cond;

% Se si vuole visualizzare i giorni e le ore in cui viene emesso un
% segnale di allarme dall'algorithm togliere "%"
for i=1:length(FAULT)-1
    if FAULT(i)==100
        % Print dei giorni in cui viene segnalato un guasto.
        % sommo h/Ts=288 poichè nella prima giornata non ho GFM e
        % IFM quindi non posso avere segnalazioni di
        % errore. Infatti il campione 1 del GFM parte dal giorno 2
        % fprintf('GUASTO %s \n',time(i+h/Ts));
    end
end

% Se si vuole visualizzare i giorni e le ore in cui è presente
% un guasto da simulatore togliere "%"
% Print allarmi da simulatore
% disp('SEGNALAZIONE GUASTO DA SIMULATORE')
for i=1:length(fault_pump)-1
    if fault_pump(i)==-100
        % fprintf('GUASTO %s \n',time(i));
    end
end

% Plot segnale d'allarme
f3=figure(3);
f3.Position=[900 150 625 400];
plot(tp_1,FAULT,time,fault_pump,'r');
title('Segnalazione Guasto');
xscale linear

```

```

legend('Guasto da algoritmo','Guasto da simulatore');
f5=figure(5);
f5.Position=[700 350 625 400];
subplot(3,1,1)
plot(tp_1,GFM,tp_1,GFM_cost,'r')
legend('GFM','soglia GFM')
title('Confronto GFM con relativa soglia')
subplot(3,1,2)
plot(tp_2,IFM,tp_2,IFM_cost,'r')
legend('IFM','soglia IFM')
title('Confronto IFM con relativa soglia')
subplot(3,1,3)
plot(tp_1,GS,tp_1,GS_cost,'r')
legend('glucose slope','soglia slope')
title('Confronto glucose slope con relativa soglia')

% Conteggio positivi e negativi. FP=false positive,
% FN=false negative,TP=true positive e TN=true negative
% Divido l'asse dei tempi in intervalli di 6h, verifico se è presente o
% meno il guasto attraverso fault_pump e se è stato segnalato o meno il
% guasto dall'algoritmo attraverso il vettore FAULT
fault_pump=fault_pump(h/Ts+1:h/Ts*gg); % elimino i primi 288 campioni,
% ovvero il primo
% giorno non utile all'analisi

%Se è presente un guasto fault_pump=-100, se è stato segnalato un
% guasto FAULT=100. Se è stato segnalato un true positive attraverso
% fault_pump, nelle 6 ore successive ignoro eventuali false positive.
frame=6; % ore di suddivisione asse dei tempi
FP=0;
FN=0;
TP=0;
TN=0;
% Individuo i sotto intervalli dove è presente un guasto. frame_guasto
% è un vettore di lunghezza 116, ogni campione rappresenta un
% intervallo in cui posso avere una segnalazione di guasto
frame_guasto=[];
m=0;
i=1;
% 8352 (8640-288) diviso 72 sono 116 sottointervalli
frame_guasto=zeros(1,(h/Ts*(gg-1))/(frame/Ts));

```

```

while i<=length(tp_1) % divido asse dei tempi in intervalli da 6 ore
    % ovvero 72 campioni
    % aggiungendo a fine ciclo frame/Ts=72 campioni
    m=m+1; % numero cicli, ovvero il conteggio del numero di
        % sottointervali esaminati
    for j=i:frame/Ts*m
        if fault_pump(j)==-100 % è presente un guasto
            frame_guasto(m)=1; % memorizzo il guasto nel sottointervallo
            break
        end
    end
    i=i+frame/Ts;
end
% Creo due vettori per memorizzare quando il mio algoritmo manda
% un segnale di guasto, positive, o meno, negative. Di default
% negative è tutto pari a 1, quando viene segnalato un guasto, il
% relativo campione di positive diventa 1 mentre quello di negative
% diventa 0, sono complementari. Entrambi i vettori hanno 116 campioni
positive=zeros(1,(h/Ts*(gg-1))/(frame/Ts));
negative=ones(1,(h/Ts*(gg-1))/(frame/Ts));
i=1;
n=0;
% divido asse dei tempi in intervalli da 6 ore ovvero 72 campioni
while i<=length(tp_1)
    n=n+1; % numero cicli, ovvero il conteggio del numero di
        % sottointervali esaminati
    for j=i:frame/Ts*n
        if FAULT(j)==100 % è presente un guasto
            positive(n)=1;
            negative(n)=0;
            break
        end
    end
    i=i+frame/Ts; % aggiungo 72 così da andare al sottointervallo successivo
end

for i=1:length(frame_guasto)
    % se nel sottointervallo è presente un guasto e viene segnalato
    % abbiamo un true positive

```

```

if frame_guasto(i)==1 & positive(i)==1
    TP=TP+1;
% se nel sottointervallo è presente un guasto ma non viene segnalato
% abbiamo un false negative
elseif frame_guasto(i)==1 & positive(i)==0
    FN=FN+1;
% se nel sottointervallo non è presente un guasto e non
% viene segnalato abbiamo un true negative
elseif frame_guasto(i)==0 & positive(i)==0
    TN=TN+1;
% se nel sottointervallo non è presente un guasto ma viene
% segnalato dall'algorithm abbiamo un true positive.
% attenzione: se il sottointervallo esaminato è successivo ad un
% true positive, ignoro il false positive
elseif frame_guasto(i)==0 & positive(i)==1
    if i==1 % se siamo nel primo intervallo segnala FP
        FP=FP+1;

        % non siamo più nel primo intervallo, devo verificare che
        % nel precedente intervallo non ci sia un TP, se è presente
        % non segnalo FP
    elseif frame_guasto(i-1)==1
        %non faccio nulla, non segnalo FP
    else
        FP=FP+1;
    end
end
end
end

% Calcolo performance algoritmo
TP_tot=TP_tot+TP;
disp(['FP=',num2str(FP), ' TP=',num2str(TP), ' FN=',num2str(FN), ...
    ' TN=',num2str(TN)])
specificity=TN/(TN+FP)
precision=TP/(TP+FP)
recall=TP/(TP+FN);
sensitivity=TP/(TP+FN)
disp(['FP/day=', num2str(FP/gg)]);

performance(1)=performance(1)+FP;

```

```

performance(2)=performance(2)+sensitivity;
performance(3)=performance(3)+specificity;
performance(4)=performance(4)+ precision;
cicli_Paz=cicli_Paz+1;
disp('Per proseguire con il prossimo paziente premere un tasto')
pause
end
disp('Pazienti terminati, performance algoritmo:')
disp(['sensibilità=',num2str(performance(2)/max_Paziente),' FP/day=', ...
      num2str(performance(1)/(max_Paziente*gg)),...
      ' precisione=',num2str(performance(4)/max_Paziente), ...
      ' specificità=',num2str(performance(3)/max_Paziente)]);
disp(['Guasti correttamente individuati: ',num2str(TP_tot(1))])

```

## %Funzione calc\_Avg

```

% definisco una funzione per il calcolo del valore medio di un segnale
% sotto forma di vettore, all'interno di una finestra mobile arbitraria,
% la quale trasla di un certo tempo

```

```

% Definizione parametri in input

```

```

% gg sono i giorni totali di analisi del segnale, dai quali escludo in
% automatico le prime 24h siccome non sono utili per il confronto e arrivo
% fino alle 23:55 dell'ultimo giorno, siccome le ore 24 fanno parte del
% giorno successivo. Nel nostro caso saranno 72h ma posso utilizzare
% finestre di analisi più ampie.
% L è la lunghezza della finestra per il calcolo del valore medio in ore
% s è il segnale in forma vettoriale, costituito dai campioni di cui
% bisogna fare la media entro una determinata finestra

```

```

% Definizione parametri in output

```

```

% AVG vettore di output contenete la media del segnale all'interno di una
% finestra di lunghezza L e traslata di un tempo T, dall'ora 24 del primo
% giorno(inizio secondo giorno) all'ora 23:55 dell'ultimo giorno

```

```

function AVG=calc_Avg(gg,L,s)

```

```

h=24; % ore in un giorno

```

```

% avendo il segnale ed il numero di giorni mi ricavo Ts in ore, il tempo di
% campionamento del segnale, mi serve per calcolare correttamente il numero

```

```
% di campioni all'interno di un giorno ed eseguire una traslazione della  
% finestra pari proprio a Ts tempo di campionamento
```

```
Ts=1/((length(s))/(gg*24));
```

```
AVG=[]; % vettore vuoto all'interno del quale inserire i valori medi
```

```
% Analizzo Dal primo campione del secondo giorno (ore 24),
```

```
% alle ore 23:55 dell'ultimo giorno, penultimo campione del segnale s
```

```
for t_W=(h*1/Ts+1):(gg*h*1/Ts)
```

```
    s_temp=s((t_W-L*1/Ts+1):t_W); % utilizzo una variabile temporanea
```

```
                                % per prendere i campioni nelle L ore
```

```
                                % precedenti, L*1/Ts campioni
```

```
    % calcolo del valore medio del CGM nella finestra
```

```
    AVG(end+1)=mean(s_temp);
```

```
end
```

```
end
```