Universidad de Alcalá

Programa de Doctorado en Electrónica: Sistemas Electrónicos Avanzados. Sistemas Inteligentes

Diseño de un controlador electrónico para la dosificación automática de insulina basado en estimación de niveles basales para diabetes tipo 1

Tesis Doctoral presentada por

Jesús Berián Múgica

2020

Universidad de Alcalá

Programa de Doctorado en Electrónica: Sistemas Electrónicos Avanzados. Sistemas Inteligentes

Diseño de un controlador electrónico para la dosificación automática de insulina basado en estimación de niveles basales para diabetes tipo 1

Tesis Doctoral presentada por

Jesús Berián Múgica

Director

Ignacion Bravo Muñoz / Alfredo Gardel Vicente

Alcalá de Henares, 25 de Noviembre de 2020

A mi mujer y a mis hijos, el motor que me impulsa día a día.

Agradecimientos

A todos los que la presente vieren y entendieren.

Inicio de las Leyes Orgánicas. Juan Carlos I

Esta tesis doctoral es el fruto de muchas horas de trabajo. Horas que han sido robadas a mi mujer Soraya y a mis hijos Jesús y Julia. Nada de lo que esta tesis doctoral contiene podría ver la luz si no fuera por el esfuerzo que han hecho ellos al permitirme dedicarme a este estudio. Es a ellos a los que principalmente quiero dar mis agradecimientos, pues son los que, con total seguridad, más han aportado a la causa.

Mención especial merecen mis directores de tesis, Nacho y Alfredo. Muchas horas son las que hemos pasado trabajando y revisando para que todo el trabajo tuviera sentido.

Quiero también agradecer a todos aquellos que en algún momento me han escuchado cuando me he enfrentado a algún problema nuevo o han aportado alguna idea para continuar. Sergio, Eduardo y Miguel siempre han estado ahí cuando lo he necesitado y su apoyo ha sido fundamental para llegar aquí.

Gracias a todos mis amigos ″pancreáticos″ por acompañarme y, como no podía ser de otra manera, a Mercedes por siempre estar dispuesta a ayudarme revisando documentación o escuchando nuevas ideas.

Gracias a todos.

Resumen

Esta tesis se centra en el campo de la bioingeniería en diabetes y, para ser más preciso, en el desarrollo de un sistema de control en lazo cerrado para el control de la glucemia de un paciente con diabetes tipo 1. Se propone un nuevo método de estimación de parámetros y control capaz de adaptarse mejor a las necesidades de insulina del paciente.

En esta tesis se desarrolla un algoritmo capaz de estimar las necesidades basales de insulina para el paciente y estimar la sensibilidad a la insulina en cada momento. Junto a estas estimaciones se implementa un método para efectuar la regulación de insulina y mantener estables las glucemias del paciente de manera segura.

A nivel de diseño electrónico, se desarrolla un sistema embebido capaz de comunicarse con varios de los dispositivos necesarios para obtener los datos de entrada del algoritmo y además poder efectuar los cambios necesarios para el control.

Para finalizar, se realizan diferentes simulaciones para validar el algoritmo propuesto y el autor de esta tesis auto-experimenta con la plataforma hardware y el algoritmo para validar a su vez las simulaciones.

Palabras clave: Sistema embebido, diabetes, tipo 1, lazo cerrado, basal.

Abstract

This thesis focuses on the field of bioengineering in diabetes and, to be more precise, in the development of a closed-loop control system to regulate a patient's glycemia suffering type-1 diabetes. A new method to estimate the patient's insulin basal needs, along with a new control system, is designed allowing to better adapt to the patient's conditions.

This thesis develops a new method capable of estimating the insulin basal needs for a patient in real-time and also accordingly adapt the insulin sensitivity factor. Along with this method, a new control algorithm is implemented to help regulate the insulin delivery and stabilize the patient's glycemia in a safe way.

As of electronic designs, a new wearable embedded system is developed capable of communicating with several of the compatible devices to obtain input data and, also, make the necessary changes to perform the control.

Several simulation batches were performed to validate the performance and safety of the algorithm and the thesis' author auto-experiments with the embedded platform and the algorithm to validate the simulations.

Keywords: Embedded system, diabetes, type 1, closed loop, basal.

Índice general

Resumen **ix**

Abstract **xi**

Índice general **xiii**

Índice de figuras **xix**

Índice de tablas **xxiii**

1 Introducción **1**

1.1 Motivación 1

1.2 Alcance y objetivos generales de la tesis 4

1.3 Descripción del problema 5

1.4 Estructura de la memoria 6

2 Estado del arte **7**

2.1 Evolución de los sistemas de control de la glucemia 7

2.2 Implementaciones de los lazos de control 9

2.3 Tipos de páncreas artificiales 12

2.3.1 Número de hormonas 12

2.3.2 Tipo de algoritmo 13

2.3.3 Tipo de lazo de control 13

2.3.4 Tipo de objetivo: rango o valor 14

2.3.5 Resumen de Sistemas de Páncreas Artificial 15

2.4 Pruebas realizadas con sistemas de páncreas artificial 16

2.5 Patentes relacionadas con sistemas de páncreas artificial 18

2.6 Soluciones disponibles para el paciente 21

2.6.1 Comerciales 21
2.6.2 Do-It-Yourself 22
2.6.2.1 Nightscout 22
2.6.2.2 Sistemas de control en lazo cerrado 23
2.6.2.3 OpenAPS 25
2.7 Objetivos de la tesis 29

3 Propuesta de sistema de control: Plataforma *wearable* de bajo consumo 31
3.1 Introducción 31
3.2 Propuesta de sistema en lazo cerrado 33
3.3 Requisitos del sistema *wearable* 36
3.4 Prestaciones y sistema *wearable* propuesto para la dosificación de insulina . 37
3.5 Descripción del diseño electrónico 39
3.6 Arquitectura firmware básica para el soporte del lazo de control 44
3.6.1 CC1110 - SoC de comunicaciones con dispositivos Medtronic 44
3.6.1.1 Gestión de potencia del procesador 46
3.6.1.2 Gestión de tiempos 46
3.6.1.3 Recepción de datos serie (puente BLE) 47
3.6.1.4 Recepción de datos via Radio 47
3.6.1.5 SmartRecovery 51
3.6.1.6 Consulta de datos de bomba de insulina 53
3.6.1.7 Procesado de información 55
3.6.1.8 Envío de mensajes por Radio 57
3.6.2 CC2540 - Comunicaciones por Bluetooth Low Energy 59
3.6.3 CC2510 - Comunicaciones por Radio 2.4GHz 60
3.7 Seguridad en las comunicaciones 60
3.8 Integración con Nightscout 62
3.9 Conclusiones 64

4 Algoritmia de control 67
4.1 Introducción 67
4.2 Descripción de parámetros de entrada al sistema 68
4.3 Conceptos básicos aplicables al algoritmo 70

4.3.1 Curva de acción de la insulina 70
4.3.2 Insulina Activa o Insulina a Bordo 71
4.3.3 Gestión de ingestas 73
4.3.4 Necesidades basales de insulina 78
4.3.5 Insulina Activa negativa 80
4.3.6 Factor de Sensibilidad a la Insulina 81
4.3.7 Estado glucémico del paciente 81
4.3.8 Objetivo glucémico 82
4.3.9 Gestión de ejercicio físico 83
4.3.10 Suspensión por hipoglucemia 84
4.3.11 Cálculo de corrección 84
4.4 Flujo lógico del algoritmo 87
4.5 Seguridad médica del paciente 89
4.6 Implementación de algoritmos alternativos 90
4.7 Conclusiones 90

5 Estimación dinámica de necesidades basales de insulina 93
5.1 Introducción 93
5.2 Importancia de las necesidades basales de insulina como parámetro 94
5.3 Algoritmia de estimación de necesidades basales 96
5.3.1 Descomposición de bolos de insulina 96
5.3.2 Estimación de necesidades basales y el factor de sensibilidad a la insulina 98
5.3.3 Problemas derivados de la regresión lineal 99
5.3.4 Mitigación de problemas mediante sensibilidad prefijada 100
5.3.5 Extrapolación de necesidades basales actuales 101
5.3.6 Ajuste dinámico del factor de sensibilidad a la insulina 103
5.3.7 Efectos causados sobre la estimación de las necesidades basales debidas a parámetros erróneos 105
5.3.8 Objetivo glucémico dinámico y umbral de suspensión por hipoglucemia 105
5.4 Inclusión de la estimación en el algoritmo básico de control 108
5.5 Conclusiones 109

6 Resultados **111**

6.1 Introducción 111

6.2 Entorno de simulación 111

6.2.1 Fases de pruebas 113

6.2.2 Simulador y modelo de controlador 114

6.2.2.1 Generación de nuevo controlador 114

6.2.2.2 Definición de simulación 116

6.2.2.3 Ajuste de parámetros por paciente 118

6.2.2.4 Simulación del bloqueo de la infusión 120

6.2.3 Implementación del controlador 123

6.2.3.1 Inicialización del controlador 123

6.2.3.2 Método de cálculo 125

6.2.4 Ejecución de la simulación 131

6.2.5 Estrategia de simulación 133

6.3 Resultados de las simulaciones 134

6.3.1 Interpretación de gráficas de resultados 134

6.3.2 Simulación de ingestas de hidratos de carbono 135

6.3.3 Simulaciones sobre el modelo de controlador básico 136

6.3.3.1 Simulaciones en lazo abierto 136

6.3.3.2 Simulaciones en lazo abierto con correcciones post-prandiales 137

6.3.3.3 Simulaciones en lazo cerrado 139

6.3.4 Simulaciones sobre el modelo de controlador con estimación de necesidades basales y bloqueo parcial de la infusión 140

6.3.4.1 Simulaciones en lazo cerrado sin bloqueo de insulina ni ingestas de hidratos de carbono (Batch 1) 140

6.3.4.2 Simulaciones en lazo cerrado sin bloque de insulina y con ingestas de hidratos de carbono (Batch 2) 142

6.3.4.3 Simulaciones en lazo cerrado con bloqueo de insulina y sin ingestas de hidratos de carbono (Batch 3) 143

6.3.4.4 Simulaciones en lazo cerrado con bloqueo de insulina y con ingestas de hidratos de carbono (Batch 4) 147

6.4 Entorno de experimentación real 149

6.4.1 Equipos utilizados 149

6.4.2 Estrategia de experimentación 149

6.5 Resultados de la experimentación 150
6.6 Conclusiones 153

7 Conclusiones y líneas futuras 155
7.1 Conclusiones 155
7.2 Líneas futuras 157
7.3 Publicaciones derivadas de la tesis doctoral 158
7.4 Disponibilidad de diseños y código fuente 159

Bibliografía 161

A Detalles técnicos de la implementación de la plataforma *wearable* 169
A.1 Flags de control de potencia del procesador 169
A.2 Temporizadores internos 170
A.3 Funciones de soporte al cómputo de parámetros 172
A.4 Variables globales accesibles para el algoritmo de control 174
A.5 Funciones de control de terapia 176

B Listado de términos 179
B.1 Terminología aplicable 179

Índice de figuras

2.1 Primer esquema de páncreas artificial [1] 7
2.2 Páncreas artificial de Arnold Kadish [2] 8
2.3 Primer páncreas artificial con computador (Biostator) [3] 9
2.4 Acción insulina Humalog® [4] y retardo 10
2.5 Diagrama de bloques del páncreas artificial descrito en la patente US 4515584 A [5] 19
2.6 Diagrama de bloques del páncreas artificial descrito en la patente US 5298022 A [6] 20
2.7 Nightscout [7] 22
2.8 Primera versión del setup de OpenAPS [8] 24
2.9 Ejemplo de sistema Loop [9] 25
2.10 Ejemplo 1 de sistema AndroidAPS [10] 26
2.11 Ejemplo 2 de sistema AndroidAPS[10] 27
2.12 Ejemplo de sistema HAPP [11] 28

3.1 Esquema genérico de un sistema híbrido *wearable* para la infusión de insulina en lazo cerrado 33
3.2 Esquema de controlador simplificado indicando retardos relacionados con la persona y el sistema sensor usado en lazo cerrado 35
3.3 Efecto relativo de varios tipos de insulina en el tiempo 36
3.4 Esquema de controlador simplificado indicando retardos y la entrada de información del paciente 37
3.5 Diagrama de bloques del sistema *wearable* propuesto 40
3.6 Diagrama de bloques de interconexión de SoCs 41
3.7 Esquemas de la plataforma - interconexión SoCs 42
3.8 Esquemas de la plataforma - Regulación de potencia y carga de batería . . 43
3.9 Plataforma hardware desarrollada para el sistema de control en lazo cerrado 43

3.10 Circuito impreso de la plataforma - Capa Top 44
3.11 Diagrama de ejecución principal - CC1110 45
3.12 Ejemplo de flujo de información entre mensajes del MCG 51
3.13 Diagrama de flujo de la gestión de la transmisión por radio 58
3.14 Nightscout 63

4.1 Ejemplo de absorcion 1 unidad de insulina Humalog 71
4.2 Ejemplo de absorcion de insulina Humalog - Duración de la insulina 72
4.3 Ejemplo de cálculo de insulina activa 73
4.4 Ejemplo de absorción de hidratos con distinto índice glucémico 74
4.5 Simulación de ingesta con dosificación perfecta sin adelanto de insulina . . 76
4.6 Simulación de ingesta con dosificación perfecta con adelanto de insulina de 25 minutos 77
4.7 Simulación de ingesta con dosificación perfecta con adelanto de insulina de 40 minutos 78
4.8 Simulación de ingesta con dosificación perfecta con adelanto de insulina de 65 minutos 79
4.9 Ejemplo de acciones correctivas 80
4.10 Ejemplo de cálculo de corrección - Paso 1 85
4.11 Ejemplo de cálculo de corrección - Paso 2 86
4.12 Diagrama del flujo lógico del algoritmo 88

5.1 Diagrama de flujo del algoritmo incluyendo la determinación de necesidades basales de insulina. 94
5.2 Ejemplo de curva de bolos instantáneos de una unidad de insulina con diferentes duraciones. 96
5.3 Ejemplo cálculo de iIOB con dos bolos de insulina. 97
5.4 Ejemplo. Relación entre dSGV e iIOB 99
5.5 Ejemplo de resultados erróneos debidos a número insuficiente de duplas en la regresión lineal. a) Usando subconjunto 1 de datos, b) Usando subconjunto 2 de datos 100
5.6 Ejemplo de estimación de necesidades basales utilizando ISF prefijado . . . 101
5.7 Representación gráfica del cálculo de $\mathrm{ISF}_{\mathrm{final}}$ 104
5.8 Ejemplo de error en la estimación de iIOB debido a IAT incorrecta 106
5.9 Relación entre el $objetivo_{final}$, el flag LTS_{flag} y el valor de $glucemia_{final}$ . . . 107

6.1 Ejemplo de clase tipo controlador en simglucose 115
6.2 Ejemplo simulación en simglucose 118
6.3 Modificación de simglucose.controller.base 121
6.4 Modificación de simglucose.controller.env.T1DSimEnv - 1 121
6.5 Modificación de simglucose.controller.env.T1DSimEnv - 2 122
6.6 Modificación de lsimglucose.controller.env.T1DSimEnv - 3 122
6.7 Modificación de simglucose.controller.t1dpatient.T1DPatient - 1 123
6.8 Modificación de simglucose.controller.t1dpatient.T1DPatient - 2 123
6.9 Inicialización del controlador 125
6.10 Resultados de la simulación en lazo abierto 137
6.11 Resultados de la simulación en lazo abierto con correcciones post-prandiales 138
6.12 Resultados de la simulación en lazo cerrado 140
6.13 Resultados del Batch 1 de simulaciones 141
6.14 Resultados del Batch 2 de simulaciones 143
6.15 Resultados del Batch 3 de simulaciones 144
6.16 Comparación entre insulina bloqueada y TDD en batch 3 147
6.17 Resultados del Batch 4 de simulaciones 148
6.18 Resultados de 3 meses de experimentación - Distribución 151
6.19 Resultados de 3 meses de experimentación - Percentiles 153

Índice de tablas

2.1 Listado de sistemas de Páncreas Artificial [12] 15

3.1 Comparación de duraciones de baterías LiPo estándar en el sistema final con consumos medios 41
3.2 Tabla de comandos implementados para la recepción BLE 48
3.3 Tabla de notificaciones implementadas para la transmisión BLE 48
3.4 Codificación de línea 4b6b 50
3.5 Ejemplo de Tabla de Factores de Sensibilidad a la Insulina 53
3.6 Ejemplo de Tabla de Objetivos Glucémicos 53
3.7 Ejemplo de Perfil basal 54

6.1 Resultados de tiempo en rango en el Batch 1 de simulaciones 141
6.2 Resultados de tiempo en rango en el Batch 2 de simulaciones 142
6.3 Resultados de tiempo en rango en el Batch 3 de simulaciones 144
6.4 Comparación de los Batches 1 y 3 en términos de TDD 145
6.5 Resultados de tiempo en rango en el Batch 4 de simulaciones 147

Capítulo 1

Introducción

Donde quiera que se ama el arte de la medicina se ama también a la humanidad.

Platón

1.1 Motivación

La OMS (Organización Mundial de la Salud) define la Diabetes como una enfermedad crónica que aparece cuando el páncreas no produce insulina suficiente o cuando el organismo no utiliza eficazmente la insulina que produce. [13]

Aunque no se dispone de cifras oficiales de la OMS en cuanto al impacto de la enfermedad a fecha de hoy, la OMS afirma que esta enfermedad afecta a más de 400 millones de personas en el mundo (aproximadamente el 8 % de la población mundial en 2014) y las cifras de afección aumentan año a año. Los enfermos de diabetes tienen una mayor probabilidad de sufrir complicaciones como ceguera, infarto de miocardio, insuficiencia renal, accidente cerebro-vascular y amputación de los miembros inferiores. Se estima que en 2012 hubo 1,5 millones de muertes causadas directamente por la diabetes y que, otros 2,2 millones, fueron atribuibles a hiperglucemias.[13]

La diabetes se divide en 3 tipos o variantes:

- Tipo 1 o Insulino-dependiente: Se caracteriza por la imposibilidad del páncreas de producir insulina. Se trata de un problema auto-inmune en el que las células beta del páncreas (responsables de la producción de insulina) son destruidas de manera selectiva. La insulina es una hormona que permite que la glucosa existente en el torrente sanguíneo entre en las células del organismo. Sin insulina la glucosa se acumula en la sangre y puede ocasionar daños en el organismo a corto y largo plazo. Entre los síntomas para el diagnóstico se encuentran:
  - Incremento del apetito y la sed (casi continuo)

  - Pérdida de peso
  - Cansancio muy elevado
  - Aumento de la cantidad de orina

  El tratamiento consiste principalmente en:

  - Inyecciones de insulina
  - Controles glucémicos periódicos
  - Dieta

  Aproximadamente el 5 % de las personas con diabetes sufren esta variedad. [14]

- Tipo 2 o no-insulino-dependiente: Es un trastorno metabólico en el que el paciente sufre de un aumento de la resistencia a la insulina que produce su páncreas y/o una reducción de la producción de insulina. Debido a esto sufre hiperglucemias debidas al mal control. El tratamiento principal en estos casos consiste en:

  - Dieta
  - Ejercicio físico
  - Tratamientos orales como la Metformina
  - Insulina si fuera necesario

  Aproximadamente el 95 % de los pacientes con diabetes sufren esta variante. [14]

- Gestacional: Es una condición que, de producirse, aparece normalmente sobre la semana 24 del embarazo y por la que los niveles en sangre de glucosa se elevan al aumentar la resistencia a la insulina de la madre. La resistencia a la insulina se eleva por el efecto de otra serie de hormonas que bloquean la absorción de la insulina en el cuerpo. Aproximadamente un 4 % de las mujeres embarazadas sufren este tipo de diabetes. Los efectos desaparecen después del parto.[14]

Cuando una persona es diagnosticada con diabetes debe ser formado en todas las áreas que van a ser parte de su día a día a partir de ese momento: tipos de insulina, administración de insulina, medicación oral, medición de glucosa capilar, medición de glucosa continua, conteo de raciones de hidratos de carbono, cálculo de bolos de insulina para compensar comidas o bolos correctores, mantenimiento del equipamiento, cambio de agujas y/o catéteres, ajustes de parámetros como ratios para las ingestas, niveles basales y/o sensibilidades a la insulina, reconocimiento de síntomas de hipoglucemia o hiperglucemia, etc.

Un paciente con diabetes puede llegar a tener que tomar más de 180 decisiones diarias [15] de las cuales dependerá su bienestar, en el momento y en el futuro. Entre estas decisiones podemos encontrar:

- Controles capilares de glucemia (medición de glucemia en sangre mediante un glucómetro)
- Mantenimiento de un sistema MCG (Monitor Continuo de Glucemia)
  - Calibración del sensor cuando sea necesario (algunos requieren calibraciones cada 12 horas)
  - Ajustar tu ritmo de vida para que el cambio de sensor (cuando caduca) no coincida con un periodo de gran variabilidad glucémica.
  - Mantenimiento del stock de sensores y transmisores
  - Monitorización manual de si los datos del sensor corresponden con las sensaciones esperadas en el cuerpo del paciente para detectar desviaciones de los datos y la necesidad de una re-calibración
  - Gestión de alarmas por hipoglucemia o hiperglucemia
- Mantenimiento de un sistema de infusión de insulina
  - Control de la insulina restante y accesible al paciente en cada momento
  - Control del estado del set de infusión de insulina (equipo que permite transportar la insulina desde el reservorio de la bomba de insulina hasta el tejido subcutáneo del paciente)
  - Planificar su cambio cada pocos días o cuando se encuentre deteriorado.
  - Mantenimiento de la batería de la bomba de insulina
  - Mantenimiento térmico de la insulina para evitar su degradación
- Gestión de ingestas
  - Planificación temporal de ingestas para poder administrar insulina con cierto tiempo de adelanto
  - Estimación del contenido de las ingestas para una correcta dosificación de insulina
  - Monitorización exhaustiva de la glucemia tras las ingestas para evitar hiperglucemias post-prandiales
- Planificación temporal del ejercicio para adecuar la insulina en el cuerpo del paciente a sus necesidades durante el ejercicio
- Gestión de hipoglucemias mediante ingesta de hidratos de carbono de recuperación cuando sea necesario.
- Etc

Es imposible que una persona sea capaz de calcular todos los parámetros necesarios para controlar su glucemia en función de lo que ocurre en su vida: su vida pasaría a ser hacer esa única tarea. Por otro lado, esta repetitiva y matemática tarea puede ser efectuada por un sistema electrónico, liberando así al paciente de esta carga. Es importante remarcar que no todos los pacientes tienen las mismas capacidades y necesidades: por ejemplo, hay personas mayores o niños que necesitan de un cuidador para lograr mantener un buen control o, llegando al extremo, mantenerse vivos. Un sistema automatizado es de gran utilidad en este tipo de casos pues mejora tanto su vida como la de sus cuidadores. Es a este tipo de sistemas a los que se ha denominado comúnmente como sistemas de páncreas artificial, y oficialmente como sistemas de control de glucemia en lazo cerrado.

Es por esto que la objetivo principal de esta tesis doctoral es diseñar un sistema capaz de calcular cada pocos minutos, de manera autónoma, los cambios en terapia necesarios para mantener al paciente con un buen control glucémico. Este sistema no sólo tendrá un impacto directo sobre la salud física en el momento presente y futuro, sino también de manera casi inmediata en la salud mental al lograr desprenderse, por lo menos parcialmente, de esa carga que supone la gestión de la enfermedad. La motivación es única: mejorar la calidad de vida y el bienestar del paciente y/o sus cuidadores e intentar normalizar sus vidas en la mayor manera posible.

1.2 Alcance y objetivos generales de la tesis

En esta tesis se pretende obtener un sistema de control que, con ciertos datos de entrada, pueda ayudar al control de la glucemia de un paciente de manera semi-automática. Debido a que será necesaria información por parte del paciente a este tipo de lazo de control se denomina híbrido.

Si bien es cierto que disponemos de dispositivos y tecnología capaz de suministrar insulina con una alta precisión (0.025 U[1]) y medir la glucemia del paciente con MARD[2] inferiores a 10 %[16], lo cual nos permite tener datos válidos como representación del estado glucémico real del paciente y actuar con precisión suficiente para considerar el proceso seguro, la latencia del proceso de medida sumada al tiempo de respuesta de las insulinas sintéticas hace inviable un sistema totalmente autónomo si no deseamos superar ciertos límites de concentración de glucosa en sangre. Es por este motivo que se le requerirá al paciente un cierto grado de interacción para poder responder adecuadamente a los eventos que ocurran: aviso previo de ingestas y su estimación de contenido en hidratos de carbono.

Se abordará el diseño de una plataforma HW/SW capaz de implementar un algoritmo de control y se harán pruebas simuladas para obtener datos y analizar si los resultados son los esperados. Posteriormente, se diseñará un algoritmo capaz de adaptarse en tiempo real a las necesidades basales de insulina del paciente, reduciendo de esta manera los

[1] U = Unidades de insulina.
[2] MARD = Mean Absolute Relative Difference

parámetros a configurar por el paciente, y se analizarán los resultados de la simulación de dicho algoritmo. En el apartado 2.7 se enumeran uno a uno los objetivos particulares que la tesis pretende aportar una vez revisado el estado de la cuestión.

1.3 Descripción del problema

Cualquier persona que viva con diabetes tiene que aprender la relación existente entre la insulina y la glucosa, y esta relación cambia de persona a persona. No existe una pauta estándar o perfecta que valga para todos los pacientes. Es más, ni una misma pauta es válida para un mismo paciente exactamente de un día a otro.

La insulina es una hormona anabólica que permite el aprovechamiento de la glucosa por parte de las células del cuerpo en los procesos de síntesis con gasto energético. Dicho con otras palabras, es la hormona necesaria en el cuerpo para que las células puedan hacer uso de la glucosa que se encuentra en el torrente sanguíneo y, por consiguiente, no se acumule en la sangre después de una ingesta.

Por otro lado, con cada ingesta de hidratos de carbono se produce un incremento de la concentración de glucosa en sangre. Este aporte es necesario para sostener las funciones básicas del cuerpo.

El glucagón, al contrario de la insulina, tiene el efecto de liberar cantidades grandes de glucosa (previamente almacenada en el hígado) en el torrente sanguíneo. Suele utilizarse dentro de la terapia del paciente únicamente como procedimiento de seguridad.

Como parece lógico, cada vez que el paciente tiene una ingesta de hidratos de carbono, es necesario realizar un aporte proporcional de insulina para mantener en rango válido la concentración de glucosa en sangre: si la insulina inyectada es excesiva, la concentración de glucosa en sangre (o glucemia) se verá reducida llevando al paciente a lo que se conoce como hipoglucemia. Si la glucemia cae a niveles muy bajos el paciente podría llegar a sufrir una hipoglucemia severa. Una hipoglucemia severa [17] es una condición del paciente en la cual no es capaz de valerse por sí mismo, necesitando de la ayuda de terceras personas. Por contrario, si la insulina inyectada es insuficiente, la glucemia subirá por encima de los rangos recomendables y puede causar trastornos médicos a largo plazo.

El paciente tiene que tener calculados lo que se denomina comúnmente como ratios. Estas ratios son la relación de insulina necesaria por cada ración de hidratos de carbono. Estas ratios dependen de muchos factores y normalmente varían a lo largo del día. Una ración de hidratos de carbono equivale a 10gr de hidratos en España [18] y a 15gr de hidratos en algunas otras partes del mundo [19]. Utilizando estas ratios, ante una ingesta, el paciente estimará (normalmente pesando o “a ojo” según le dicte la experiencia) cuántas raciones de hidratos de carbono pretende ingerir. Multiplicando el número de raciones estimadas por la ratio de la franja horaria obtendrá las unidades de insulina necesarias para esa ingesta.

Existe otro motivo para administrar insulina al paciente y es en caso de correcciones glucémicas. Si el paciente se encuentra por encima del rango glucémico deseable se podrá administrar lo que se denomina coloquialmente como bolo corrector. Este bolo (término equivalente a inyección) se calculará utilizando las sensibilidades del paciente. Al igual que ocurre con las ratios, las sensibilidades establecen la variación de glucemia esperada (siempre reducción) frente a una unidad de insulina. Cuando el paciente quiera corregir su glucemia calculará el exceso de glucemia que quiere compensar y, utilizando la sensibilidad para esa franja horaria, calculará las unidades de insulina extra que necesita. Si la sensibilidad real es inferior a la utilizada para el cálculo no tendremos toda la corrección necesaria y, si la sensibilidad real es superior a la del cálculo, podríamos acabar sufriendo una hipoglucemia.

Además, existe lo que se llaman niveles basales de insulina. Estos niveles basales determinan la cantidad de insulina necesaria por el cuerpo a lo largo del día para poder mantener una glucemia estable. Este parámetro, al igual que la sensibilidad, es cambiante a lo largo del día y se ve influido por multitud de factores: ejercicio físico, fármacos, enfermedades, etc.

1.4 Estructura de la memoria

Esta memoria ha sido organizada en los siguientes capítulos

En el Capítulo 2 se revisa el estado del arte y se describe la propuesta de esta tesis doctoral dentro de la actual situación del campo de estudio.

En el Capítulo 3 se describe cuales son las principales características que debe cumplir un sistema de control *wearable* y se diseña una plataforma que cumple con ellas y que posibilita la implementación de un sistema genérico de control en lazo cerrado.

En el Capítulo 4 se explica el funcionamiento de un sistema de control básico en lazo cerrado, haciendo uso de la plataforma *wearable*. Este algoritmo hace uso únicamente de parámetros previamente configurados por el paciente.

En el Capítulo 5 se analiza la relación entre las necesidades basales de insulina del paciente, su evolución glucémica y la historia de infusión de insulina. Con esta información se desarrolla un algoritmo capaz de adaptarse a las necesidades basales de insulina del paciente en tiempo real, eliminando la necesidad de configurar un perfil de dosificación de insulina basal por parte del paciente.

El Capítulo 6 describe cómo, haciendo uso de un simulador, se validan los algoritmos desarrollados. En su última parte, el autor de esta tesis doctoral auto-experimenta con la plataforma real y el algoritmo con determinación automática de necesidades basales y se comprueba que los resultados obtenidos en la vida real concuerdan con los arrojados por las simulaciones.

Capítulo 2

Estado del arte

El progreso de la medicina nos depara el fin de aquella época liberal en la que el hombre aún podía morirse de lo que quería.

Stanislaw Jerzy Lec

2.1 Evolución de los sistemas de control de la glucemia

En el año 1959 aparece publicada por primera vez la idea de un sistema automático capaz de regular la infusión de insulina a un paciente [1] (ver Figura 2.1). Aunque en aquel momento no existían los medios que tenemos hoy en día, ya se hablaba de la posibilidad de desarrollar algún dispositivo capaz de suplir a un páncreas, al igual que se había logrado con éxito en el caso de los pulmones, riñones y corazones artificiales.

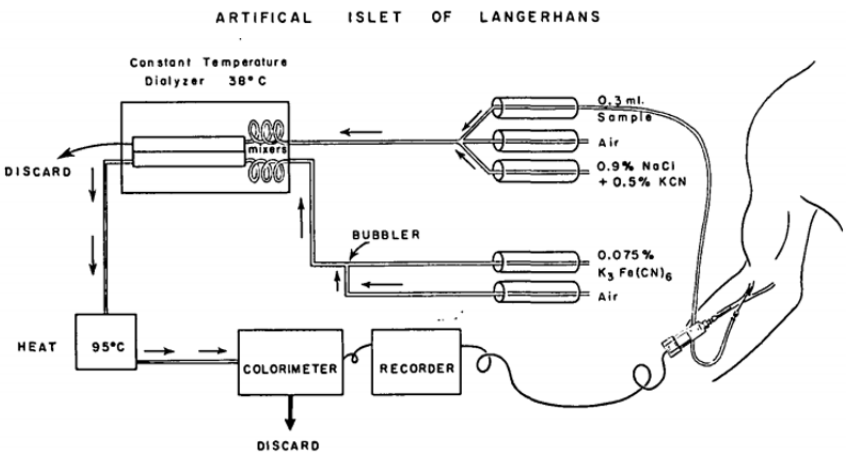


Figura 2.1: Primer esquema de páncreas artificial [1]

Ya en el año 1963, Arnold Kadish [2] diseñó lo que sería la primera bomba de insulina portable con control en lazo cerrado (ver Figura 2.2). Tenía el tamaño de una mochila militar y era capaz de analizar la concentración de glucosa en sangre y controlar la infusión

de insulina mediante un servo mecanismo de dos estados: encendido y apagado. No gozó de gran atención dado que su uso no parecía muy sencillo dado el tamaño y su mantenimiento.

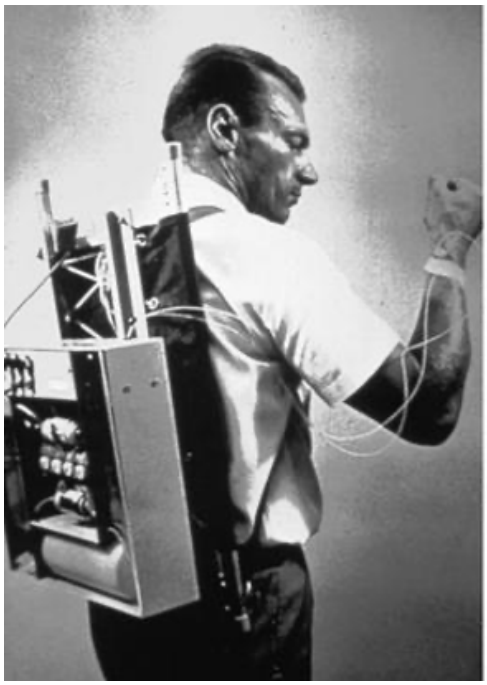


Figura 2.2: Páncreas artificial de Arnold Kadish [2]

Hubo que esperar hasta el año 1977 a que A.H. Clemens, P. H. Chang y R. W. Myers hicieran públicos sus ensayos [3] con lo que conocemos hoy como el primer páncreas artificial controlado por un computador: Biostator (ver Figura 2.3). Realizan sus pruebas con un sistema complejo y no portable que regula los niveles de glucosa en sangre de manera intravenosa, tanto la medida de la glucemia como la administración de la insulina. Era un computador el encargado de calcular las dosis de insulina necesarias para el control glucémico y todo quedaba registrado en papel mediante una impresora. Al igual que sucedió con el páncreas artificial de Arnold Kadish, su uso quedó altamente restringido por su complejidad, tamaño y poca portabilidad. Se destinó únicamente a estudios de investigación de corta duración.

Como no podía ser de otra manera, también existen intentos de conseguir reproducir los mismos efectos regulatorios mediante mecanismos biológicos. Muchos avances se han hecho en el ámbito del trasplante de células beta [20], pero el uso de inmunosupresores es un gran problema: pueden tener grandes efectos secundarios y riesgos. Otro de estos intentos trata de encapsular tejido pancreático sano dentro de una membrana para que segregue insulina una vez implantado en el cuerpo [21]. A esta membrana se le aplica una protección para evitar su rechazo por parte del cuerpo humano, pero siguen adoleciendo del mismo problema que tienen las células betas del páncreas: el sistema inmunológico destruye estos tejidos y es necesario el uso de inmunosupresores.

Como efectos secundarios de los inmunosupresores encontramos un mayor riesgo de infecciones y mayor riesgo de cáncer. También existen otros de carácter menor como

Figura 2.3: Primer páncreas artificial con computador (Biostator) [3]

pérdida de apetito, vómitos y nauseas, aumento del crecimiento del pelo y temblor de manos.

Dado que tenemos métodos para que las personas que padecen diabetes tipo 1 puedan vivir con una buena calidad de vida sin necesidad de inmunosupresores, y son grandes los avances que se han hecho en el campo de la bioingeniería, no es normal su tratamiento por esta vía. Actualmente existen a disposición de los pacientes insulinas sintéticas de efecto rápido, lento y mixto. Hace muchos años que se comercializan dispositivos capaces de administrar insulina de manera subcutánea (jeringuillas, viales, bolígrafos y bombas de insulina) y, mediante los dispositivos de medida de glucosa en sangre (medidores capilares y sensores continuos de glucemia), cualquier persona es capaz de vivir con diabetes sin que esto suponga directamente un riesgo para su vida.

Especialmente desde la aparición de los sensores continuos de glucemia, ya que las bombas de insulina ya llevan mucho más tiempo en el mercado, empezaron los desarrollos de los sistemas de control en lazo cerrado (páncreas artificial). Son muchos los diseños que se han hecho de manera experimental [22][23][24][25][26][27][28][29] pero sólo unos pocos a día de hoy se comercializan y están aprobados por las autoridades sanitarias en EE.UU. (FDA): Sistema Medtronic Minimed 670g [30] y Tandem t:slim X2 con Control-IQ [31]. Ambos sistemas están disponibles para el paciente a día de hoy y ofrecen un tiempo en rango (TIR) medio de 73 % [32] y 71 % [33] respectivamente.

2.2 Implementaciones de los lazos de control

Aunque la tecnología ha avanzado notablemente en cuanto a los elementos necesarios para poder realizar un control en lazo cerrado, sigue existiendo una serie de puntos en los que todavía se sigue encontrando un problema para realizar un buen control. Entre estos

problemas se encuentran:

- Insulinas con efecto no suficientemente inmediato: A diferencia de como ocurre en el caso de un páncreas sano o en los primeros sistemas de páncreas artificial en los que la insulina se administraba directamente en el torrente sanguíneo, en estos sistemas la insulina se aplica de manera subcutánea por su mayor facilidad de uso. Esta forma de aplicación introduce un retardo desde que es inyectada hasta que realmente comienza a hacer efecto en el cuerpo. El pico máximo de acción de la insulina se encuentra aproximadamente después de 30-45 minutos de su aplicación y puede llegar a suceder hasta 120 minutos después de la aplicación en algunos casos [34]. Este retardo, al igual que otros que existen, reduce la capacidad de reacción del sistema de control [35]. Se muestra en la Figura 2.4 la respuesta de una unidad de insulina Humalog® [4] con el tiempo en la que se puede apreciar este efecto.

- Insulinas con efecto duradero en el tiempo: además del retardo en su acción, las insulinas más rápidas tienen un tiempo total de acción en el cuerpo cercano a las 8 horas. Cierto es que las últimas horas tienen un efecto casi residual, pero podemos tener entre 3 y 5 horas con un efecto no despreciable. Esto hace que el lazo de control tenga que tener en cuenta que durante todo ese tiempo va a tener una cierta cantidad de insulina haciendo efecto y, como es lógico, no se puede eliminar [35].

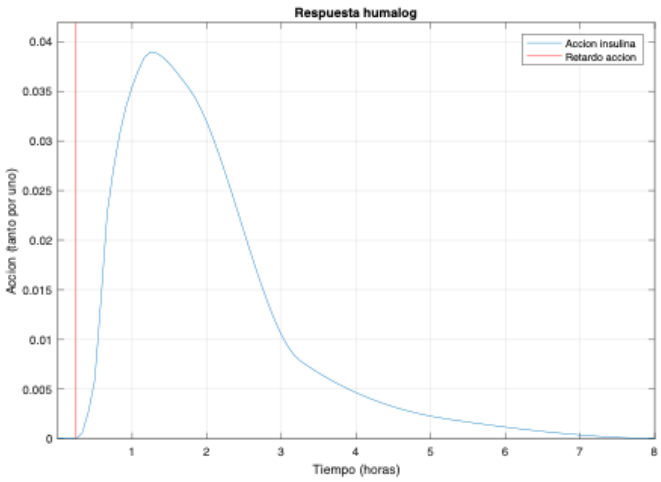


Figura 2.4: Acción insulina Humalog® [4] y retardo

- Sensores continuos de glucemia y las medidas intersticiales: Los sensores continuos de glucemia (CGM) realizan una medida de la concentración de glucosa en sangre en el líquido intersticial y no de manera intravenosa. Esto hace que exista un retraso de aproximadamente 15 minutos en la medida de la glucemia con respecto a la medida en sangre. Al igual que sucede en el caso de las insulinas, esto reduce la capacidad de reacción del lazo de control: cuando el páncreas artificial detecte un aumento de la glucemia del paciente mediante la medida del sensor, hace 15 minutos que

realmente empezó y, debido al retraso en la acción de la insulina, no tendremos un efecto que contrarreste la subida hasta unos 30 minutos después. Esto es de vital importancia a la hora de reducir los picos post-prandiales [26]. Son muchos los estudios que constatan que el uso de lazos híbridos, en los que se anuncian las ingestas, reducen considerablemente este efecto [22][26][35]. También hay estudios para diseñar algoritmos que detectan ingestas no anunciadas para poder actuar sobre ellas [36].

- Ruido, calidad de señal y calibración en el sensor continuo de glucemia: El sensor continuo de glucemia juega un papel crucial en el sistema de control. La mayor parte de los sensores proporcionan una medida de la corriente que circula por un electrodo introducido en el cuerpo de la persona. Esta corriente depende directamente de una reacción química que sucede en ese electrodo, el cual está siendo desgastado por el propio proceso, y por un posible proceso de rechazo a cuerpo extraño en el cuerpo del paciente. Esto hace que sea necesario calibrar la relación corriente <->glucemia cada cierto número de horas (típicamente 12 horas como máximo). Además, una vez se acerca el fin de vida del sensor, se hace menor la relación señal a ruido existente en las medidas y provoca una gran dispersión de los datos entregados. La mala calidad de los datos del sensor provocará efectos no deseados sobre el lazo de control y, por ese motivo, es de vital importancia mantener en buen estado el sistema de medida.

- Algoritmos de control robustos: Dado que el sistema a controlar (la glucemia del paciente) se ve afectada por múltiples factores como las ingestas, el ejercicio físico, y otros muchos factores no medibles como el estrés, desajustes hormonales y ciertas enfermedades, los sistemas de medida disponibles pueden introducir grandes errores. Es necesario que el algoritmo de control sea lo más robusto posible y tolere un alto grado de error en sus entradas y, aun en ese caso, siga siendo lo más seguro posible. Cualquier decisión tomada por el algoritmo debe tener como prioridad la seguridad del paciente.

- Plataforma integrada: Los sistemas de páncreas artificial existentes a día de hoy están compuestos por más de un módulo o subsistema. Estos subsistemas típicamente son un sensor continuo de glucemia y una bomba de insulina. En algunos casos hay que añadir un tercer subsistema que es el controlador donde se realizan los cálculos del lazo de control. Todos estos subsistemas deben comunicarse entre ellos para poder trabajar de manera coordinada y en tiempo real. Esto hace que el buen funcionamiento de las comunicaciones pase a tener un carácter vital: El sistema debe tolerar fallos de comunicaciones y, en la medida de lo posible, mantener la seguridad del paciente.

- Alimentación de los subsistemas: el páncreas artificial debe ser portátil. Para que el sistema sea portátil es necesario que todos los subsistemas dispongan de un medio de almacenamiento de energía suficiente para poder trabajar de manera autónoma el mayor tiempo posible (mínimo deseable de 24 horas). Normalmente todos los

subsistemas harán uso de baterías, pero, para reducir el impacto en peso y volumen que tiene que cargar el paciente, la electrónica usada deberá ser de bajo consumo.

2.3 Tipos de páncreas artificiales

El término de páncreas artificial lleva utilizándose desde hace muchos años para referirse a dispositivos capaces de regular la infusión de insulina al paciente. En el año 2010 Minimed/Medtronic sacó al mercado su modelo de bomba de insulina subcutánea MiniMed Paradigm REAL-Time Revel, la cual dispone de un modo de suspensión de infusión por hipoglucemia. En este caso, aunque se cumple el criterio de regular la infusión de insulina, únicamente se realiza en un sentido (cortando la infusión de insulina en caso de hipoglucemia) pero, al faltar la regulación para el caso en que sea necesaria más insulina, no se considera sistema de páncreas artificial. En años posteriores se fueron añadiendo funciones avanzando en el concepto de páncreas artificial hasta llegar al modelo 670g que es capaz de compensar también hiperglucemias.

Los páncreas artificiales se pueden clasificar en función de una serie de parámetros [12]:

- Número de hormonas utilizadas
- Tipo de algoritmo de control
- Tipo de lazo: completo o híbrido
- Tipo de objetivo: rango o valor

A continuación se describen cada uno de ellos.

2.3.1 Número de hormonas

Las hormonas de las que dispone normalmente una persona para el control de la glucemia son la insulina y el glucagón. La insulina, como ya hemos explicado anteriormente, es la hormona que produce una reducción de la concentración de la glucosa en sangre. Por otro lado, tenemos otra hormona llamada glucagón que actúa sobre el metabolismo del glucógeno, también producida en el páncreas. Esta hormona estimula un proceso llamado gluconeogénesis e incrementa la producción de glucosa en el hígado, con el consiguiente aumento de la concentración de glucosa en sangre.

Un gran número de los páncreas artificiales existentes utilizan únicamente insulina como hormona a regular y se denominan de “hormona única”. El resto utilizan insulina y glucagón y regulan las dos hormonas de manera independiente, denominándose de “hormona doble”. [27]

Desde el punto de vista puramente de control parece lógico pensar que es más interesante utilizar las dos hormonas para ser capaz de hacer correcciones en ambos sentidos,

pero el uso de glucagón tiene un problema importante: el glucagón suele distribuirse como un vial con polvo blanco estéril liofilizado, el cual debe ser diluido con un disolvente precargado en una jeringuilla justo antes de su uso; la solución de glucagón no es estable en el tiempo y no aguanta mucho tiempo diluido. Muchos avances se están haciendo para conseguir un glucagón que aguante preparado durante el tiempo necesario. Además, el uso de dos hormonas implica el uso de dos bombas separadas o bombas duales (no comunes) con dos sets de infusión[1] distintos y separados. Independientemente de estos problemas, son varios los sistemas que han optado por esta opción.

Los sistemas de hormona única utilizan insulina como hormona a regular. Esta decisión de diseño limita la posible actuación a reducción de la concentración de glucosa en sangre y, por ese motivo, debe ser dosificada con máxima precaución para no provocar hipoglucemias no deseadas. Como ventaja estos sistemas únicamente necesitan de una bomba y la insulina es mucho más estable en el tiempo.

2.3.2 Tipo de algoritmo

A la hora de clasificar los algoritmos de control implementados, se establecen las siguientes categorías:

- MPC (Model Predictive Control) [37]: Son sistemas que calculan el valor de glucemia próximo en el futuro y actúan en consecuencia. Algunos pueden incluso "aprender" del pasado y adaptarse al paciente.
- PID (Proportional Integral Derivative) [38]: Utilizan el error calculado sobre la glucemia del paciente y se calculan correcciones proporcionales, integrales y derivativas. Responden directamente, y de manera continua, a los valores de glucemia que reciben del sensor.
- Sistemas expertos [39]: Al igual que pasa con todos los sistemas expertos en otras áreas de conocimiento, intentan reproducir las acciones que tomaría un paciente experto para controlar su terapia.
- Redes neuronales [40]: De manera experimental, y únicamente en simulación, se ha probado a entrenar redes neuronales para que sean capaz de predecir la respuesta de la glucemia del paciente y reaccionar ante ella [41]. El problema de estos sistemas es que dependen de un entrenamiento previo que puede no ser trivial.

2.3.3 Tipo de lazo de control

El objetivo de cualquier sistema de páncreas artificial es el control de la glucemia y cuanto más automático y más transparente para el paciente mucho mejor.

[1] set de infusión: equipo fungible que conecta el reservorio de la bomba de insulina con el cuerpo del paciente, permitiendo la infusión insulina desde el reservorio hasta el tejido subcutáneo del paciente

El problema aparece cuando la respuesta del sistema, como ya se describió en el apartado de problemas de implementación, es demasiado lenta para adaptarse a las perturbaciones. Probablemente las ingestas de hidratos de carbono de alto índice glucémico sean las perturbaciones más grandes a la que se pueda enfrentar un sistema de estas características. Ante una ingesta de hidratos de carbono podemos sufrir lo que se denomina como "pico post-prandiales": aumento de la glucemia posterior a la ingesta de hidratos de carbono. Este aumento suele ser pronunciado y las latencias en los procesos de medida y actuación no permiten que la respuesta sea lo suficientemente rápida como para evitar sufrir una hiperglucemia si únicamente actúa el lazo de control (lazo completo o "full closed loop" - FCL).

Existen estudios que, dadas las limitaciones que presentan los dispositivos actuales, justifican la necesidad de usar un lazo híbrido (o "hybrid closed loop" − HCL) en el cual el paciente anuncie las comidas con antelación [22][26]. De esta manera el sistema puede dosificar insulina con antelación para esa ingesta y reducir el pico post-prandial.

El parámetro que se utiliza como referencia para evaluar el control glucémico de una persona es una prueba llamada Hemoglobina Glicosilada o HbA1c. Este parámetro está directamente relacionado con la media glucémica ponderada sobre los 3 últimos meses y se expresa en forma de porcentajes [42]. Los valores típicos de una persona no diabética son inferiores a 5.7 % mientras que en el caso de sufrir diabetes pueden subir hasta más de 12 % en los peores casos. Se suele tomar como valor de poco riesgo de sufrir complicaciones a largo plazo una HbA1c inferior a 7 %.

Se estima que la reducción en Hb1Ac es del orden de 0.5 % por el hecho de usar un lazo híbrido.

2.3.4 Tipo de objetivo: rango o valor

Como ya se ha mencionado anteriormente, el objetivo de un páncreas artificial es mantener la glucemia del paciente dentro del rango óptimo para el correcto funcionamiento del cuerpo. Para este fin, los algoritmos de control pueden tener objetivos definidos por rango o por valor:

Los algoritmos con objetivos definidos por rango utilizan los valores máximo y mínimo de glucemia a obtener y todas las actuaciones van orientadas a que el valor actual de glucemia se encuentre entre esos dos valores.

Cuando la glucemia es superior al valor máximo, la actuación irá orientada a conseguir llegar al valor máximo válido. De esta manera, si algún otro parámetro (como la sensibilidad) no se encuentra perfectamente bien ajustado, el error cometido en la corrección únicamente requerirá una corrección adicional si no fue suficiente con la primera o, si el error estaba acotado, la glucemia terminará en algún punto dentro del rango válido. Por contrario, si la glucemia es inferior al valor mínimo, la actuación irá orientada a conseguir

Nombre del sistema	Desarrollador comercial (si aplicable)	Gen.	Etapa	Aproximación hormonal	Tipo de algoritmo de control	Estrategia de control
Inreda artificial pancreas	Inreda Diabetic BV	3	6	Insulina + glucagón	PID	Objetivo
iLet (Bionic Pancreas)		3	6	Insulina + glucagón	MPC/PD	Objetivo
GlucoSitter	DreaMed Diabetes Ltd; Medtronic Diabetes (licensee)	2	4 o 5	Insulina	FL	Objetivo / Rango
Florence		2	4	Insulina	MPC	Objetivo
Diabetes Assistant (DiAs) Including inControl	TypeZero Technologies LLC (licensee)	2	4	Insulina	MPC	Objetivo / Rango
Closed-Loop Glucose-Sensing Insulin-Delivery System		2	5	Insulina	PID	Objetivo
Closed-Loop Assessment (CLASS)		2/3	5 + 6	Insulina + glucagón	MPC	Objetivo
ZMPC		2	5	Insulina	MPC	Rango
Portable Artificial Pancreas System (pAPS)		2	5	Insulina	MPC/PID/FL	Objetivo / Rango
Hybrid Closed-Loop (HCL) system	Medtronic Diabetes	2	4	Insulina	PID	Objetivo
Oregon bihormonal closed-loop system	Legacy Health Systems	3	6	Insulina + glucagón	PID	Rango
Bio-inspired Artificial Pancreas (BiAP)		3	6	Insulina + glucagón	Bio-inspirado	Objetivo
Integrated Multivariable Adaptive Artificial Pancreas (IMA-AP)		3	6	Insulina + glucagón	MPC	Objetivo
Dose Safety Controller (DSC)	Dose Safety Inc.	2	5	Insulina	FL	Objetivo
Physiologic Insulin Delivery with Adaptive Pancreas (PIDAB)		2	5	Insulina	PID	Objetivo
ClosedLoop4Meal Controls (CL4M)		2	5	Insulina	PID	Objetivo
Predictive Rule-Based Algorithm (pRBA)		2	4	Insulina	FL	Rango
Diabeloop		2	4	Insulina	MPC	Rango

Tabla 2.1: Listado de sistemas de Páncreas Artificial [12]

el valor mínimo.

Los algoritmos con objetivos definidos por valor actuarán de manera constante hasta conseguir que el valor de glucemia actual del paciente sea exactamente la del valor definido como objetivo.

2.3.5 Resumen de Sistemas de Páncreas Artificial

Gran parte de los sistemas de páncreas artificial que se han desarrollado y/o estudiado se encuentran en la Tabla 2.1. En esta tabla se puede ver cómo el autor ha categorizado según la información proporcionada por los desarrolladores.

Los sistemas de páncreas artificiales de esta lista han sido categorizados por la JDRF (*Juvenile Diabetes Research Foundation*) [43], entre otros factores, por su etapa de desarrollo. Esta etapa hace referencia al nivel de automatización del sistema según la siguiente lista:

- Etapa 1 (Primera generación): Suspensión de infusión por hipoglucemia
- Etapa 2 (Primera generación): Suspensión de infusión por previsión de hipoglucemia
- Etapa 3 (Primera generación): Lazo cerrado híbrido (HCL - *Hybrid Closed Loop*)

- Etapa 4 (Segunda generación): Lazo cerrado híbrido avanzado (AHCL - *Advanced Hybrid Closed Loop*)
- Etapa 5 (Segunda generación): Lazo cerrado totalmente automatizado
- Etapa 6 (Tercera generación): Lazo cerrado totalmente automatizado multi-hormonal

Cabe resaltar que la lista de sistemas en estudio o comercializados no es reducida y que existen opciones incluso en etapa 6. También es interesante ver la cantidad de estudios multi-hormonales que existen cuando todavía no se ha encontrado una método para mantener el glucagón estable y diluido durante mucho tiempo. El glucagón, hormona que eleva los niveles de glucosa en sangre, se encuentra disponible para su uso en forma de compuesto que hay que diluir justo antes de su uso y no puede ser almacenado diluido por su pérdida de eficacia. En cuanto a los algoritmos, se puede apreciar una gran variedad de estrategias de control y tipos de algoritmos. Esto da una idea del gran campo de estudio que abarca la materia.

A nivel comercial también hay otras dos soluciones más a día de hoy, aunque no aparecen en la Tabla 2.1 debido a su fecha de publicación y no han sido incluidas en la tabla por mantener la integridad de los datos de la publicación [12]:

- Bigfoot Biomedical: hormona simple, lazo híbrido. [44]
- Insulet Horizon: hormona simple, lazo híbrido [45]

Se debe resaltar el hecho de que, de entre estos sistemas, únicamente las soluciones de Medtronic (MiniMed 670G [30]) y Tandem Diabetes (t:slim X2 - Control IQ [31]) están disponibles a día de hoy para el público en general. Esto ha provocado que la comunidad de personas con diabetes a nivel mundial desarrolle sus propios sistemas DIY (Do-It-Yourself) y se estén usando cada día con mayor número de usuarios. Entre estos se encuentran:

- OpenAPS: hormona simple, lazo híbrido, lógica difusa, objetivo por rango. [8]
- Loop: hormona simple, lazo híbrido, lógica difusa, objetivo por rango. [9]
- AndroidAPS: hormona simple, lazo híbrido, lógica difusa, objetivo por rango. [10]

2.4 Pruebas realizadas con sistemas de páncreas artificial

Desde los comienzos del desarrollo de los sistemas de páncreas artificial se han hecho pruebas para intentar validar las distintas técnicas y tecnologías [46].

Son múltiples los estudios que se han realizado y con finalidades distintas. En el año 1977 se hicieron las primeras pruebas en lazo cerrado con un ser humano [3] en el que

el objetivo era mantener la glucemia del paciente estable y en rango. Al ser las primeras pruebas en realizarse con un sistema de estas características, se consideraron los resultados como satisfactorios al lograr reducir en número de hiperglucemias y aumentar el tiempo dentro del rango aceptable.

Normalmente estas pruebas se realizan con personas voluntarias elegidas según ciertos requisitos que, a su vez, dependen del objetivo de la prueba. Los criterios suelen ser:

- Edad
- Sexo
- Nivel de actividad física
- Tipo de alimentación
- Otras afecciones añadidas, como por ejemplo en situaciones en las que se busque analizar posibles relaciones entre enfermedades.

Aunque al comienzo las pruebas eran más generalistas en cuanto a sus objetivos, en años siguientes se realizan pruebas para analizar otros parámetros como el rendimiento de los distintos algoritmos [22][23][24][25][26][28]. A medida que los algoritmos empiezan a dar resultados, las pruebas empiezan a enfocarse en temas más concretos como:

- Control nocturno: tema especialmente interesante para evitar las hipoglucemias nocturnas, las cuales tienen una relevancia importante al suceder mientras el paciente no está consciente y puede que este no sea capaz de corregirlas [25][26][29]. Podemos decir que la mayor parte de las pruebas con estos sistemas de páncreas artificial dan resultados muy positivos durante la noche puesto que el nivel de actividad del paciente es mínimo, no hay ingestas y el número de variables a controlar es menor, con lo que el control automático es más sencillo que durante el día..
- Control ante ingestas no anunciadas: Como parece natural, muchos intentos existen por hacer un sistema de control eficiente en lazo totalmente cerrado al que no haya que anunciarle las comidas. Este sistema proporcionaría al paciente la experiencia más parecida a no padecer la enfermedad pero, como ya se describió en los tipos de algoritmo de control, no es una tarea sencilla. Existen pruebas se sistemas que intentan detectar las ingestas y determinar su aporte para posteriormente administrar un bolo que la compense [36]. Por desgracia, el asegurar que estos algoritmos son seguros no es trivial y el mero hecho de su estudio para su desarrollo comercial exige una gran apuesta por los fabricantes de dispositivos.
- Regulación del pico post-prandial: Muchos estudios intentan demostrar que los pico post-prandiales se reducen con el uso de los sistemas en lazo cerrado. Es unánime la conclusión de que los picos se ven reducidos e incluso más si el lazo es híbrido (se anuncian las ingestas). [22][23][26]

- Número de hormonas: Aunque todos los sistemas en lazo cerrado utilizan insulina como hormona para controlar la glucemia del paciente, también existen pruebas con sistemas que hacen uso de una segunda hormona, con efecto contrario a la insulina, para la ayudar a dicho control. Esta hormona se llama glucagón y, aunque al lector le podría parecer más sencillo el control al poder compensar efectos de una hormona con su contraria, el glucagón tiene un problema todavía no resuelto: no existe una manera que permita mantener la hormona estable una vez diluida. En la actualidad esta hormona se obtiene en forma de polvo que debe ser diluido justo antes de su uso. [22][27]

En cuanto a los criterios de aceptación/comparación hay consenso: se considera rango válido o normoglucemia a las concentraciones comprendidas entre 70 mg/dl y 130 mg/dl antes de las ingestas, permitiendo subir hasta 180 mg/dl hasta dos horas después de la ingesta. Los resultados utilizando un sistema de páncreas artificial se comprenden típicamente entre un 80 % y un 90 % del tiempo dentro del rango válido.

Aunque el objetivo final de la comunidad científica es llegar a la automatización total (Sistema de tercera generación y etapa 6 según JDRF), los sistemas actuales se encuentran lejos de dicho objetivo. Es por ello que esta tesis se centra en buscar un algoritmo que mejore el control glucémico de sistemas parecidos a los disponibles a día de hoy, en etapa 3/4 y que utilicen únicamente insulina, mediante la automatización del ajuste de ciertos parámetros que afectan a la terapia del paciente. Las mejoras propuestas, además, tienen aplicación en sistemas de siguientes etapas cuando estos estén disponibles.

2.5 Patentes relacionadas con sistemas de páncreas artificial

Dado el gran número de personas que, por desgracia, sufren esta patología y el gran campo de estudio y avance disponible, son multitud de empresas y entidades las interesadas en el desarrollo de estas tecnologías. Como es normal, estas empresas están generando patentes intentando a la vez avanzar en el estudio compartiendo información y proteger sus resultados para poder explotarlos.

Son múltiples las patentes existentes sobre los sistemas en lazo cerrado, aunque la mayor parte de ellas no dan demasiados detalles en un intento de cubrir el máximo contenido posible:

- US 4515584 A (Artificial pancreas) [5]: En esta patente se describe un páncreas artificial como un dispositivo capaz de medir la glucosa del paciente mediante un electrodo introducido en el cuerpo del ser vivo, una unidad de control capaz de calcular la dosis de insulina o glucagón necesaria para realizar los ajustes y una unidad capaz de dosificar los fármacos al ser vivo (ver Figura 2.5). Se patenta en el año 1982 el concepto de páncreas artificial y es de libre uso a día de hoy.

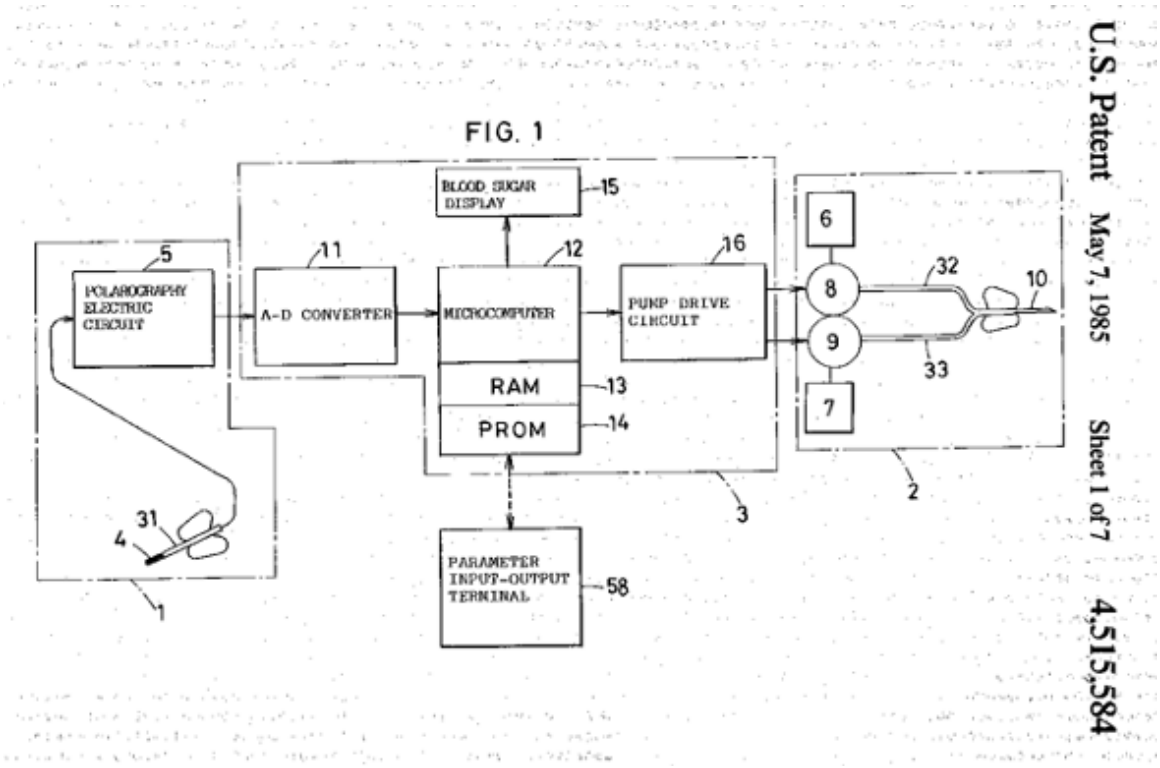


Figura 2.5: Diagrama de bloques del páncreas artificial descrito en la patente US 4515584 A [5]

- US 5298022 A (Weareable artificial pancreas) [6]: En esta patente se describe un dispositivo portable capaz de determinar la concentración de glucosa en sangre de una persona durante un periodo de 24 a 36 horas. Se inyecta una solución salina con heparina en el torrente sanguíneo utilizando una aguja de micro diálisis y una bomba. Posteriormente, utilizando una membrana semipermeable externamente a la aguja, se realiza un proceso de diálisis por el que sólo las partículas con bajo peso molecular (por debajo de 100.000 daltons) son capaces de atravesar dicha membrana. Se puede medir la concentración de glucosa midiendo la corriente eléctrica producida por los electrones liberados en el proceso de oxidación de la glucosa en ácido glucónico y peróxido de hidrógeno y la posterior descomposición del peróxido de hidrógeno para dar H2O y oxígeno. Además, añaden que, una vez obtenida la medida de la concentración de glucosa en sangre, esta medida se puede utilizar para dosificar insulina y, por tanto, tener un páncreas artificial (ver Figura 2.6).

- US 6023009 A (Artificial pancreas) [47]: En esta patente se describe el proceso por el cual se encapsulan isletas pancreáticas viables y fisiológicamente activas dentro de una membrana esférica semipermeable para posteriormente ser implantadas en el paciente.

La patente US 6023009 A [47] describe una aproximación al problema radicalmente distinta a la propuesta en esta tesis y, lamentablemente para los pacientes con diabetes tipo 1, no ha dado frutos a día de hoy.

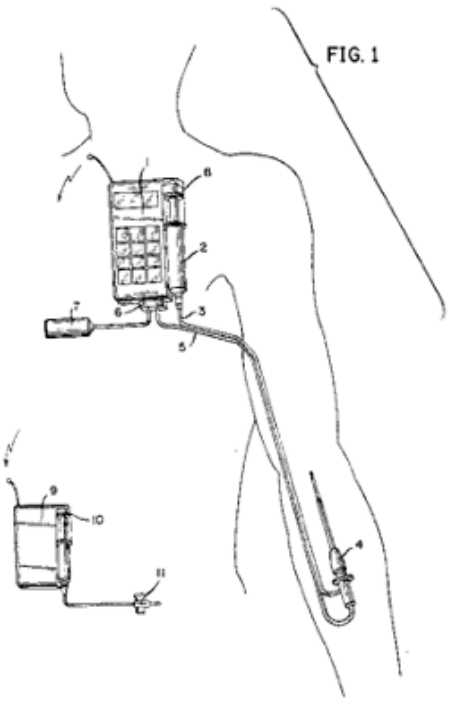


Figura 2.6: Diagrama de bloques del páncreas artificial descrito en la patente US 5298022 A [6]

Por otro lado, las patentes US 4515584 A [5] y US 5298022 A [6] describen tecnologías más similares a las propuestas en esta tesis aunque en un estadio de desarrollo más primigenio. La primera patente describe un sistema capaz de calcular la dosis de insulina necesaria para el paciente mediante las lecturas de glucemia realizadas con un electrodo. Esta es la base de los sistemas actuales de control, incluido el aquí propuesto, y establece los bloques principales de un sistema de estas características. En esta patente no se entra a discutir si el sistema debe o no ser portable y no se hace mención a ningún algoritmo avanzado de control, dejando espacio para la investigación en este ámbito.

La segunda patente [6] hace referencia a la portabilidad del sistema ya que desarrolla el concepto de un sensor de glucemia capaz de funcionar entre 24 y 36 horas que el paciente puede usar de manera continua. Esta patente es el primer paso en el estudio de los sistemas MCG, aunque los sistemas actuales utilizan distintos sistemas para la medición de la glucosa en sangre. Como se puede intuir por la figura 2.6, aunque el sistema es portable, su uso con agujas de micro diálisis no lo hace sencillo ni, muy probablemente, cómodo. Un sistema portable, como el propuesto en esta tesis, debe reducir al máximo el impacto en la vida del paciente e intentar conseguir que su mera existencia sea casi inapreciable.

Existen otras patentes que protegen funcionalidades concretas de los sistemas de infusión de insulina y de monitorización continua de glucosa pero no se hace referencia a ellas en este documento por no estar directamente ligadas a los sistemas de control en lazo cerrado y/o por ser muy específicas en cuanto a implementaciones propietarias de ciertas compañías o grupos.

2.6 Soluciones disponibles para el paciente

Muy recientemente han aparecido sistemas de control en lazo cerrado que el paciente puede usar de manera regular. Esta aparición viene de la mano de la difusión de la tecnología de sensores continuos de glucemia y las bombas de insulina. Aunque no son muchas las opciones, las podemos agrupar en dos categorías: comerciales y DIY (Do-It-Yourself o Haztelo-Tú-Mismo).

Un sistema de control en lazo cerrado, como los descritos en los apartados siguientes, puede ser descrito como un sistema en el que un sensor continuo de glucemia envía datos de glucemia del paciente a un sistema controlador, una bomba de insulina que ejecuta las órdenes o actuaciones provenientes de ese mismo controlador y el controlador en si mismo, encargado de calcular las actuaciones pertinentes para mantener en control la glucemia del paciente. En algunos sistemas comerciales, el controlador se integra dentro de la bomba de insulina. Al integrar el controlador en la bomba de insulina se obtiene como resultado un sistema más compacto, con menos dispositivos y menor probabilidad de fallo en comunicaciones.

2.6.1 Comerciales

Son muchas las empresas que están trabajando para que sean aprobadas sus soluciones de páncreas artificial por los organismos reguladores pero, a día de hoy, las dos únicas disponibles por parte de los pacientes son las soluciones de la empresa Medtronic (MiniMed 670G) y de la empresa Tandem (t:slim X2 − ControlIQ).

Es poca la información existente sobre sus algoritmos de control pero entre las características conocidas tenemos:

1. Medtronic Minimed 670G
   - Algoritmo Medtronic ePID [26][12]
   - El sistema debe aprender parámetros del paciente durante un periodo no inferior a 2 semanas antes de poder activar el lazo cerrado.
   - El lazo de control es híbrido, por lo que debemos anunciar y estimar las ingestas.
   - Utiliza una única hormona (insulina).
   - El lazo de control es un PID.
   - El objetivo es una concentración fija de 120 mg/dl.
2. Tandem t:slim − ControlIQ
   - Algoritmo TypeZero
   - El lazo de control es híbrido, por lo que debemos anunciar y estimar las ingestas.
   - Utiliza una única hormona (insulina).

- El lazo de control es un MPC (*Model-Predictive-Controller*)
- El objetivo es un rango fijo de 112.5-160 mg/dl durante el día y 112.5-120 mg/dl durante la noche.
- Dosifica bolos de corrección automáticos cuando la predicción de glucemia supere los 180 mg/dl.

2.6.2 Do-It-Yourself

Es tal la necesidad de soluciones por parte de los pacientes que un grupo de ellos, muy motivados y con conocimientos suficientes de los dispositivos y la enfermedad, han decidido crear sus propios sistemas de monitorización y control en lazo cerrado al margen de los organismos reguladores.

2.6.2.1 Nightscout

Aunque no se trate de un sistema de control en lazo cerrado, es importante hablar de Nightscout [7] (ver Figura 2.7) . Nightscout es una plataforma open-source creada por y para los pacientes que permite la monitorización remota de sensores continuos de glucemia. Aunque a día de hoy ya existen plataformas comerciales similares, esta es especialmente importante por ser la primera en proporcionar este tipo de monitorización y por ser modificable/ampliable para soportar nuevos sistemas.

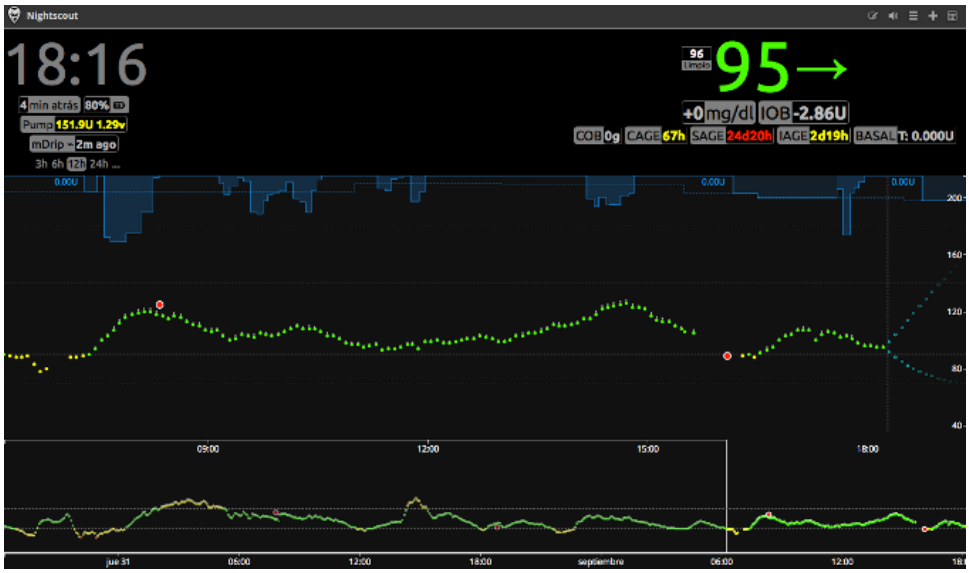


Figura 2.7: Nightscout [7]

Nightscout es un sistema compuesto por 3 elementos principales:

1. Base de datos mongo. Los datos de los pacientes son almacenados en una base de datos tipo mongo. En ella se almacenan tanto los datos del sensor continuo de glucemia como los tratamientos realizados por el paciente o un sistema de control en lazo cerrado.

2. Hosting web con soporte para NodeJS. El servidor web será el encargado de las comunicaciones con la base de datos y de generar las páginas web dinámicas que dan soporte a la aplicación escrita en NodeJS. El código es muy modular y fácilmente ampliable

3. Dispositivo *uploader*. Es el dispositivo que recibe los datos del paciente (datos del sensor continuo de glucemia, tratamientos, acciones del lazo de control, etc.) y los sube a la base de datos a través de la web de Nightscout. Este dispositivo normalmente es un teléfono con posibilidad de llevar asociado algún transceiver dependiendo del sensor/dispositivo a conectar.

Gracias a la cantidad de desarrolladores que tiene el proyecto, disponemos a día de hoy de una plataforma suficientemente versátil como para poder adaptar cualquier sistema de lazo cerrado y poder analizar su comportamiento en tiempo real con muy poco esfuerzo. Es por ello que todos los sistemas DIY hacen uso de Nightscout para la monitorización y gestión de los sistemas de control.

Es importante mencionar que el autor de la tesis ha participado en este proyecto con el diseño del *uploader* para bombas Medtronic Paradigm y anteriores [48].

2.6.2.2 Sistemas de control en lazo cerrado

A nivel de dispositivos necesarios los requisitos de estos sistemas son distintos (tarjetas procesadoras, bombas de insulina, teléfonos móviles, etc) pero a nivel de algoritmia todos comparten muchos conceptos y características:

- Utilizan una única hormona (insulina)
- Son controladores basados en modelos predictivos (MPC).
- El objetivo se define por rango y es configurable
- Pueden utilizar cualquier sensor de glucemia continuo comercial de los existentes actualmente en el mercado
- Necesitan una bomba Medtronic Paradigm con una versión antigua de firmware, una bomba Insulet Omnipod o, según el caso, una bomba Dana-R con control por Bluetooth.

Entre estos sistemas encontramos:

- OpenAPS [8]: Sistema basado de una tarjeta procesadora tipo Raspberry Pi o similar capaz de ejecutar scripts de Python sobre Linux, conexión con un sensor continuo de glucemia y con una bomba de Medtronic con firmware apropiado para poder ser controlada por RF (ver Figura 2.8). La tarjeta controladora junto con los transceivers

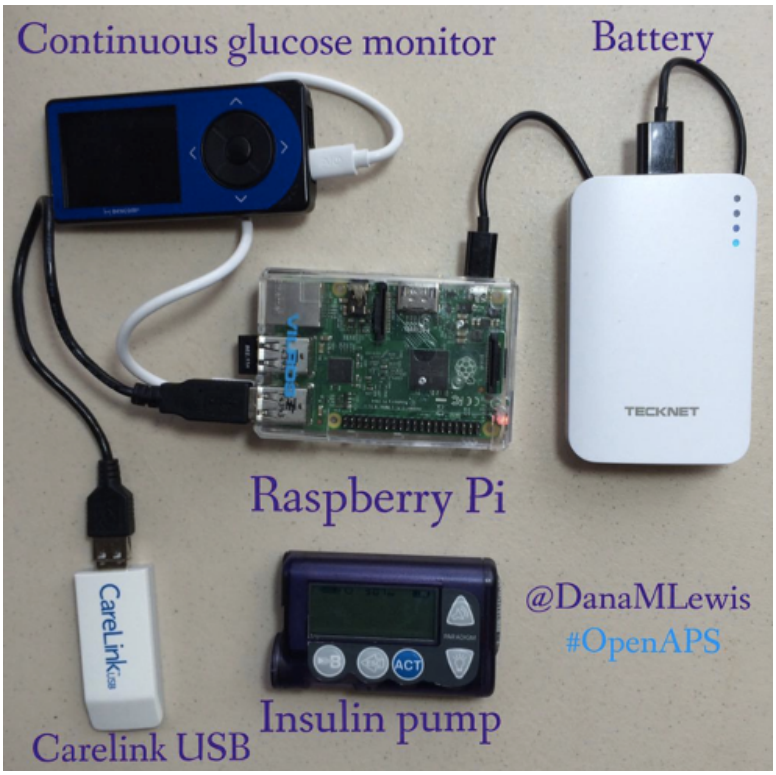


Figura 2.8: Primera versión del setup de OpenAPS [8]

necesarios son alimentados por una batería portátil. Los datos del control deben ser configurados en la bomba y los resultados se pueden consultar a través de Nightscout.

- Loop [9]: basándose en la algoritmia de OpenAPS, se desarrolla un sistema de control en lazo cerrado basado en un terminal iOS que se conecta a una bomba Medtronic compatible con control por RF utilizando un bridge denominado RileyLink (interfaz Bluetooth Low-Energy a RF compatible Medtronic) y con un sensor continuo de glucemia por Bluetooth, a través de Internet o a través del RileyLink (ver Figura 2.9). Todas las operaciones como comandar bolos, basales temporal y el ajuste de parámetros se realiza mediante la App instalada en el terminal iOS. Los datos del control se pueden consultar tanto en la App como en una web Nightscout si está disponible.

- AndroidAPS [10]: Se trata de una adaptación de OpenAPS a un terminal Android y para su uso con una bomba Dana-R controlada por Bluetooth. Puede obtener datos de glucemia de un sensor Dexcom G5 por Bluetooth o desde cualquier sensor conectado a Nightscout (ver Figuras 2.10 y 2.11). Todo el control se realiza desde la App Android y los resultados se pueden consultar en la propia App o en Nightscout.

- HAPP [11]: No se trata de un control en lazo cerrado propiamente dicho ya que no controla directamente ninguna bomba de insulina. Se trata de una aplicación Android que, haciendo uso de los datos provenientes de un sensor continuo de glucemia y los tratamientos introducidos manualmente por el usuario, es capaz de realizar

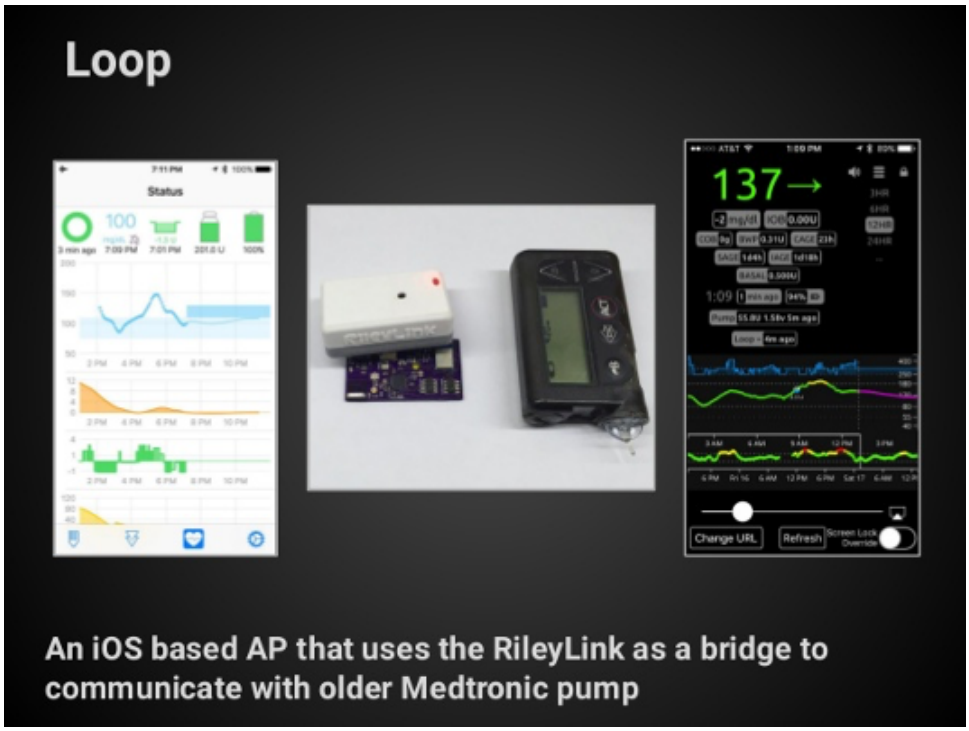


Figura 2.9: Ejemplo de sistema Loop [9]

sugerencias al usuario para realizar un mejor control (ver Figura 2.12). El algoritmo utilizado para realizar las recomendaciones es el de OpenAPS. Se pueden obtener los datos directamente de un sensor conectado por bluetooth al teléfono o a través de Nightscout.

En la pantalla de HAPP se pueden observar los siguientes puntos:

1. Nivel de infusión basal activo en el momento
2. Botón para la introducción de tratamientos y cálculo de bolos
3. Ultima lectura del sensor de glucemia, últimos resultados de la predicción, insulina a bordo y carbohidratos a bordo.
4. Menú de selección de gráficos informativos
5. Gráfico de glucemias

2.6.2.3 OpenAPS

El primer sistema de este tipo en estar disponible fue OpenAPS [8] y, quizá por ese motivo, tiene los algoritmos más avanzados y es el sistema que será estudiado en este apartado.

OpenAPS es un sistema que se aprovecha del conocimiento que tiene la comunidad de diabéticos para controlar un cierto tipo de bombas de insulina (Medtronic Paradigm) y de la posibilidad de obtener los datos de glucemia de un medidor continuo (Medtronic Minilink o Dexcom G4/G5) para integrar un sistema de control en lazo cerrado.

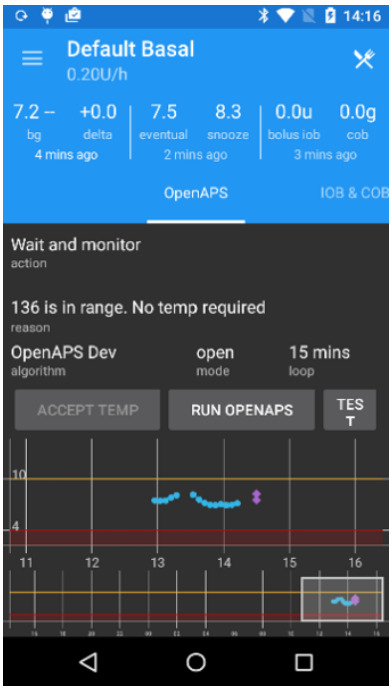


Figura 2.10: Ejemplo 1 de sistema AndroidAPS [10]

En el caso de OpenAPS, los creadores han intentado plasmar su pauta de control (la que realizan a mano) en un sistema experto. Este sistema hace uso de las glucemias recibidas del sensor y calcula la pauta para realizar correcciones si es necesario.

El sistema se compone de una batería recargable, una tarjeta procesadora corriendo Linux (Raspberry PI, Intel Edison, etc.), un receptor de un sensor continuo de glucemia conectado por USB o un sensor continuo de glucemia directamente conectado por Bluetooth y un adaptador para las comunicaciones con la bomba.

OpenAPS lo forman un conjunto de scripts escritos en Python que, ejecutado en el sistema operativo Linux, se encargan de:

1. Obtener los datos de la bomba: Es necesario conocer parámetros como tasas basales programadas, sensibilidades a la insulina, objetivos de glucemia programados y las últimas acciones realizadas sobre la bomba.

2. Obtener datos del sensor: Para realizar los cálculos del lazo de control es imprescindible tener los datos de glucemia lo más actualizados posible. Los scripts descargarán el histórico de las glucemias medidas por el sensor con el fin de poder calcular tendencias y futuros estados.

3. Realizar los cálculos del lazo de control: Una vez obtenidos todos los datos de los dispositivos anteriores, se ejecuta el lazo de control para determinar las acciones a tomar en cuanto a la infusión de insulina para los siguientes 30 minutos en forma de basales temporales.

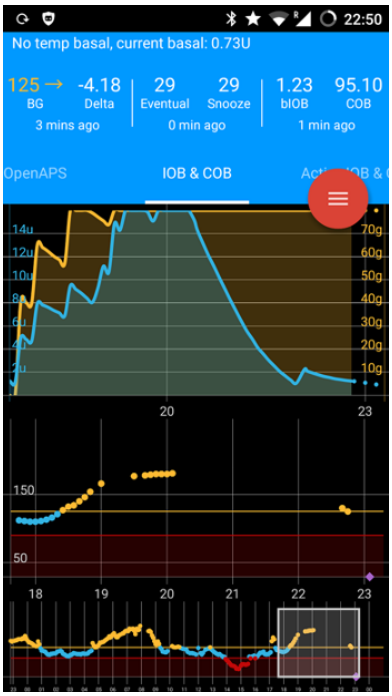


Figura 2.11: Ejemplo 2 de sistema AndroidAPS[10]

4. Aplicar el resultado de los cálculos: Una vez se determina la acción a tomar, existen scripts para comunicar a la bomba dicha acción.

5. Subir los informes a Internet (Nightscout): Para poder monitorizar el estado del controlador, se comunica a la web de Nightscout los últimos datos recibidos de los sensores y la bomba junto con la acción tomada y ciertas estimaciones como la insulina-a-bordo estimada y los hidratos-de-carbono-a-bordo.

6. Subir los informes a Internet (Nightscout): Para poder monitorizar el estado del controlador, se comunica a la web de Nightscout los últimos datos recibidos de los sensores y la bomba junto con la acción tomada y ciertas estimaciones como la insulina-a-bordo estimada y los hidratos-de-carbono-a-bordo.

La primera versión del algoritmo de OpenAPS (denominado por sus autores como oRef0) se basa en un set de reglas muy sencillo. Como los sensores continuos miden la glucemia en el líquido intersticial en vez de en sangre, existe un retardo aproximado de 15 minutos entre medidas. Para intentar compensar este retardo, se calcula lo que se denomina como “eventual glucose” o “glucosa prevista”. Las correcciones se configurarán como basales temporales de media hora (que es el mínimo sin usar directamente bolos), lo que tiene un doble beneficio:

1. Si se aplica una basal temporal después de haber puesto otra antes, el efecto de la primera queda anulado. En el supuesto caso de un error de ejecución del lazo de control será imposible llegar a aumentar la infusión de manera peligrosa: una basal temporal comandada anula a la anterior. En el caso de utilizar bolos de insulina para

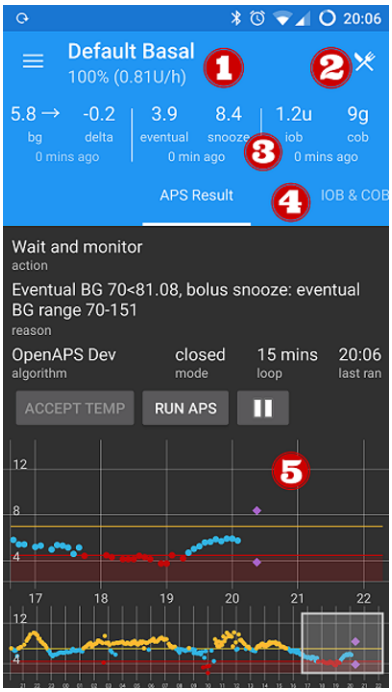


Figura 2.12: Ejemplo de sistema HAPP [11]

el control, sería posible inyectar insulina de manera reiterada, poniendo en grave riesgo al paciente.

2. Si el sistema dejase de funcionar, la bomba volverá a usar las pautas programadas de manera normal. En el caso de un sistema basado en micro-bolos, no sería posible tener programada ninguna pauta basal puesto que es necesario poder reducir la basal en casos de hipoglucemia o tendencia a ella. En caso de fallo del algoritmo o las comunicaciones, el paciente no recibiría infusión alguna de insulina pudiendo llevar al paciente a sufrir una hiperglucemia o incluso cetoacidosis diabética.

El algoritmo de OpenAPS opera según el siguiente listado de operaciones:

1. Primero se calcula la tendencia media de los últimos datos de glucosa y se extrapola la glucemia prevista en 15 minutos.

2. Se leen de la bomba los valores máximo y mínimo para el objetivo del paciente.

3. Si la glucemia está subiendo pero el valor de glucemia prevista está por debajo del objetivo se cancelan las basales temporales.

4. Si no, si la glucemia está bajando pero el valor de glucemia prevista está por encima del objetivo se cancelan las basales temporales

5. Si no, si la glucemia prevista está por encima del objetivo:

   (a) Se calcula la basal temporal a 30 minutos que hará bajar la glucemia al objetivo

   (b) Si la basal temporal requerida es mayor que la basal existente se comandará la nueva basal temporal

6. Si no, si la glucemia actual está por debajo del objetivo:
   (a) Se calcula la basal temporal a 30 minutos que nos hará subir hasta el objetivo
   (b) Si la basal temporal requerida es menor que la basal existente, se comandará la nueva basal temporal:
      i. Si se necesita estar con una basal temporal a 0 durante más de media hora, se extiende la duración de la basal temporal.

El principal "problema" que tiene este sistema es que depende totalmente de tener perfectamente ajustados todos los parámetros del paciente. Las correcciones se aplican sobre los datos de la pauta normal programada. Si esta pauta no está bien ajustada el sistema no tiene por qué funcionar como se espera.

2.7 Objetivos de la tesis

Como se ha explicado a lo largo de este capítulo, son múltiples las opciones disponibles actualmente para el control en lazo cerrado de la glucemia en diabetes tipo 1 pero, dado que son las primeras en ser desarrolladas, su configuración y ajuste requiere de amplios conocimientos sobre diabetes y sobre la respuesta del cuerpo de cada paciente.

Se podría decir que este es el comienzo en cuanto a la automatización de la terapia y, por tanto, son muchos los caminos a explorar a partir de aquí. En los sistemas disponibles en la actualidad (sistemas de segunda generación), de los que su algoritmia es de dominio público, todavía son muchos los parámetros que deben ser ajustados por el paciente/médico y que tienen un impacto considerable en los resultados de la terapia. Es objetivo de esta tesis avanzar en la automatización de algunos de estos parámetros y, con ello, mejorar los resultados de estos algoritmos y la calidad de vida del paciente. También se diseñará una plataforma *wearable* que permitirá el desarrollo y test de estos nuevos algoritmos y que podrá servir como plataforma para futuros investigadores en la materia.

En particular, dentro de la presente tesis doctoral se desean abordar las siguientes cuestiones:

- Conectividad con los elementos necesarios del AP
- Desarrollo de una plataforma *wearable* de bajo consumo
  - Firmware de bajo consumo en todos los elementos
  - Integración con sistemas de gestión ya existentes
- Mejoras en los algoritmos de control

  - Determinación automática de ajustes
  - Adaptación en tiempo real a cambios fisiológicos

- Validación de algoritmos de control en lazo cerrado

Como resultado se obtiene un sistema controlador *wearable* que reduce la complejidad de configuración del lazo cerrado. Se ha desarrollado una nueva plataforma de bajo consumo y un nuevo algoritmo que determina las necesidades basales de insulina del paciente, reduciendo la complejidad de configuración del sistema y permitiendo un mejor ajuste de la terapia en tiempo real.

Para la validación del algoritmo propuesto, también se han hecho modificaciones a un simulador especializado en diabetes tipo 1 que nos permite simular las condiciones en las que este algoritmo ofrece ventajas.

Capítulo 3

Propuesta de sistema de control: Plataforma *wearable* de bajo consumo

La ciencia puede divertirnos y fascinarnos, pero es la Ingeniería la que cambia el mundo.

Isaac Asimov

3.1 Introducción

En los últimos años, hemos podido presenciar la aparición en el mercado de los sistemas de monitorización continua de glucemia (MCG) junto con nuevos dispositivos de infusión de insulina más precisos. Los sistemas MCG son unos pequeños dispositivos conformados por un electrodo o conjunto de electrodos, los cuales son insertados en el tejido subcutáneo y son capaces de medir una corriente proporcional a los niveles de glucosa en dicho tejido. Un pequeño transmisor envía el resultado de la medición vía radiofrecuencia al receptor. Sensores como los Dexcom G4,G5 y G6, Medtronic Enlite y Guardian Sensor 3 o Abbott Freestyle Libre y Libre2 tienen una vida que oscila entre los 6 y 15 días y algunos de ellos deben ser calibrados mediante controles capilares de glucemia entre 3 y 4 veces al día. Aunque las bombas de insulina han estado disponibles desde los años 60, su precisión y funcionalidades han mejorado hasta el punto de, por ejemplo, ser capaz de dosificar insulina con precisiones de 0,025 unidades, recibir y procesar datos provenientes de un MCG o un glucómetro, y mantener el historial de eventos del paciente para su posterior análisis.

La tecnología ha evolucionado y estos dispositivos abren nuevas posibilidades. El desarrollo de sistemas de control en lazo cerrado, que reduzcan la carga a la que se enfrentan las personas que viven con diabetes tipo 1, es una prioridad para las compañías de tecnología médica. La mayor parte de las bombas de insulina actualmente disponibles en

el mercado no hacen uso de los datos de un sensor MCG para calcular la cantidad de insulina necesaria para el paciente, incluso en muchos casos en los que los datos del sensor también están siendo recibidos por dicha bomba. Cambios en la dosificación de insulina podrían tener efectos letales sobre el paciente en caso de error. Los sistemas actuales no tienen manera de cancelar el efecto de una insulina previamente dosificada, más allá de la sugerencia al paciente de que haga una ingesta de hidratos de carbono de corrección. En otras palabras, es casi insignificante el riesgo asumido al reducir la cantidad de insulina que recibe una paciente puesto que el sistema siempre puede incrementarla más tarde, pero una vez la insulina ha sido dosificada no existe manera para eliminarla del cuerpo del paciente. Esta es la razón principal por la que los fabricantes de estos dispositivos están desarrollando estas funciones por etapas; aunque existen sistemas comerciales que funcionan en lazo cerrado, la mayor parte de ellos permiten lo que se denomina como "suspensión-por-hipoglucemia" o "suspensión-por-predicción-de-hipoglucemia": la bomba de insulina suspenderá la infusión cuando la glucemia del paciente, o su predicción, sea inferior a un cierto umbral de seguridad. Esta funcionalidad de seguridad es el primer paso hacia conseguir un control en lazo cerrado. Existen unos pocos modelos de bomba de insulina, comercializados desde hace recientemente poco tiempo, que son capaces de incrementar la insulina a dosificar dependiendo de la glucemia del paciente [30][31][49].

Cada paciente es distinto, al igual que lo son sus necesidades [36]. Esta es la razón por la que son varias las comunidades de pacientes activas (Comunidad de Páncreas Artificial DIY : Loop [9] / OpenAPS [8] / AndroidAPS [10] / Nightscout [7]) las que han desarrollado sus propias plataformas de control y las han hecho configurables de manera que, con apropiados conocimientos, cada paciente puede ajustarlo a sus necesidades. Estos sistemas han sido diseñados para funcionar sobre plataformas comerciales como Raspberry Pi, Intel Edison o smartphones, normalmente usando Linux o sistemas operativos de alto nivel, y no han sido diseñados desde un inicio como sistemas alimentados por baterías en los que es necesario controlar el consumo al máximo. Otros trabajos [50][51] analizan soluciones más complejas que tampoco se focalizan en hacer un diseño *wearable*. Lo que sí comparten todos estos sistemas es que utilizan la glucemia del paciente y la historia de dosificación de insulina para calcular las acciones necesarias para mantener al paciente dentro de un rango predefinido (configurable o no por el usuario). El disponer de un conjunto de parámetros correctos para el paciente se convierte en una tarea crítica que se encuentra fuera de la temática de esta tesis doctoral.

Este capítulo se estructura en 6 apartados:

1. La propuesta del sistema, en la que se describe el funcionamiento general del sistema y las partes que lo componen.

2. Descripción del sistema *wearable* propuesto como controlador.

3. Descripción del diseño electrónico del módulo controlador.

4. Arquitectura firmware base del sistema. Este firmware es el encargado del mantenimiento de las comunicaciones y la información necesaria para la ejecución del algoritmo de control.

5. Seguridad implementada en el sistema en cuanto a comunicaciones.

6. Integración con la plataforma Nightscout. Esta plataforma proporciona las capacidades necesarias para realizar análisis de los datos, almacenamiento en la nube y monitorización remota del paciente.

3.2 Propuesta de sistema en lazo cerrado

El sistema propuesto en esta tesis doctoral se compone de 4 elementos diferenciados (ver Figura 3.1), aunque algunos (o incluso todos) podrían ser integrados dentro del mismo dispositivo:

- Bomba de insulina
- Monitor Continuo de Glucemia
- Controlador
- Dispositivo de entrada de datos

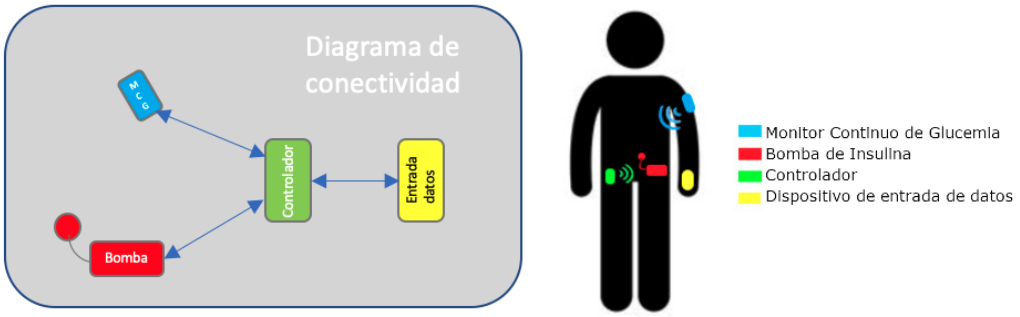


Figura 3.1: Esquema genérico de un sistema híbrido *wearable* para la infusión de insulina en lazo cerrado

La mayor parte de las bombas de insulinas comerciales actuales fueron diseñadas para trabajar en lazo abierto, por consiguiente, operan de una manera que se ajusta a las necesidades de los pacientes en esta situación: la bomba de insulina permite bolos de insulina (equivalente a una inyección de insulina) y la infusión de insulina siguiendo un patrón definido como "basal". Un perfil basal (comúnmente denominado "basales") define las diferentes cantidades de insulina que necesita el paciente a lo largo del día para mantener su glucemia estable. Como las necesidades basales cambian dependiendo de muchos factores externos, las bombas de insulina típicamente soportan tener configurado más un perfil basal, permitiendo de esta manera que el paciente pueda elegir de manera manual

el perfil que mejor se adapta a su situación en cada momento. Esta selección de perfil basal se realiza normalmente según un criterio de "tipo de día" pero, si el paciente necesita modificar su perfil basal por un periodo de tiempo más reducido (de 30 minutos a algunas horas) la bomba de insulina permite el uso de lo que se denomina como "basales temporales". Mediante el uso de estas basales temporales el paciente puede incrementar o decrementar manualmente la infusión de insulina sin necesidad de modificar el perfil basal que se esté utilizando en ese momento. Una bomba de insulina será apta para su uso en lazo cerrado en nuestro sistema si permite ser controlada por un dispositivo externo y soporta el cambio de sus parámetros de infusión de manera remota: típicamente mediante el uso de basales temporales.

Como se muestra en la Figura 3.2, se debe especificar un objetivo a conseguir por el controlador. En la figura se muestra de manera genérica y conceptual pero, en un sistema real, este objetivo debe ser fijado por el paciente/médico en aquel dispositivo que haga las funciones de controlador. Este objetivo podrá ser configurado en la propia bomba de insulina o mediante alguna entrada de datos al controlador. Este tema será analizado en apartados posteriores.

El Monitor Continuo de Glucemia ofrece al sistema de control información en tiempo real sobre la glucemia del paciente a través de un sensor. Este es un módulo compuesto por un filamento insertado debajo de la piel del paciente, el cual es capaz de medir la concentración de glucosa en el líquido intersticial, y la electrónica asociada para poder realizar la medida y la transmisión por radiofrecuencia de los datos. Los sistemas MCG actuales obtienen una nueva medida cada 5 minutos y muchos de ellos deben ser calibrados por el paciente para obtener resultados suficientemente exactos. Puesto que el sensor no está midiendo directamente la concentración de glucosa en el flujo sanguíneo, hay un retardo variable en los datos de glucemia recibidos por el controlador: existe un retardo biológico entre la concentración de glucosa en el líquido intersticial y la concentración en el flujo sanguíneo de unos 10/15 minutos. Este retardo es lo que ocasiona que las calibraciones deban hacerse cuando la glucemia del paciente esté estable: si la glucemia está estable, ambas medidas deberían coincidir al calibrar independientemente del retardo.

Este tipo de sistema normalmente requiere información introducida por el paciente y es por lo que se denominan como sistemas en lazo cerrado híbridos. La principal razón para necesitar esta entrada de datos del paciente es el retardo que existe en el proceso de medida de la glucemia y en la acción del sistema (ver Figura 3.2).

Los retardos mostrados en la Figura 3.2 corresponden a los siguientes efectos:

- Retardo en la información de glucemia (azul): como ya se explicó anteriormente, existe un retardo de entre 10 y 15 minutos inherente al proceso de medida. Esto reduce la velocidad con la que el controlador puede actuar para compensar cambios rápidos de glucemia.

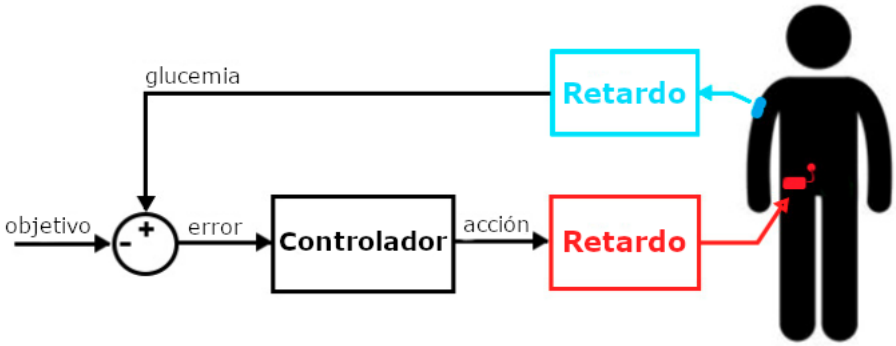


Figura 3.2: Esquema de controlador simplificado indicando retardos relacionados con la persona y el sistema sensor usado en lazo cerrado

- Respuesta de la insulina (rojo): El paciente recibe la infusión de insulina de manera subcutánea. Este proceso introduce un nuevo retardo en el lazo del sistema: el efecto de la insulina no se verá materializado en el cuerpo del paciente hasta que esta no empiece a ser absorbida, mientras que en una persona sin diabetes y con insulina endógena no existe dicho retardo [52]. La Figura 3.3 compara los efectos relativos en el tiempo de varios tipos de insulina. Como se puede apreciar, la insulina endógena (la que es generada por un páncreas sano) tiene efecto casi inmediato que termina pasada aproximadamente una hora desde su liberación en el cuerpo. El resto de insulinas son algunos de los tipos sintéticos que una persona con diabetes puede utilizar. Los usuarios de bomba de insulina normalmente utilizan las insulinas más rápidas disponibles (siempre que no exista alguna otra contra-indicación) ya que los efectos de las insulinas de acción lenta se pueden conseguir con los perfiles basales programados. Incluso en el caso de las insulinas más rápidas el pico de acción ocurre aproximadamente pasada una hora desde su infusión y puede causar efecto hasta 8 horas.

Aunque un sistema en lazo cerrado podría responder de manera efectiva incluso con estos retardos, estos harían que la glucemia del paciente alcanzara ciertos niveles que son considerados como peligrosos o, por lo menos, no recomendados. Existen situaciones en los que los pacientes experimentan cambios rápidos en su glucemia: principalmente ingestas de comida y ejercicio físico. Una comida rica en hidratos de carbono hará incrementar rápidamente la glucemia del paciente y, para el momento en el que el controlador sea capaz de reaccionar, el paciente habrá alcanzado con facilidad niveles de hiperglucemia o cercanos a ellos. Durante el ejercicio físico las necesidades de insulina del cuerpo se ven reducidas y, si la infusión de insulina no se ve reducida con suficiente antelación, el paciente podría sufrir una hipoglucemia. El controlador detectará la caída en los niveles de glucemia del paciente pasados unos 15 minutos desde el comienzo real de la caída (debido al retardo del MCG) y disminuirá la infusión de insulina pero, dado que la insulina permanece causando efecto por muchas horas (dependiendo del tipo de insulina y de su

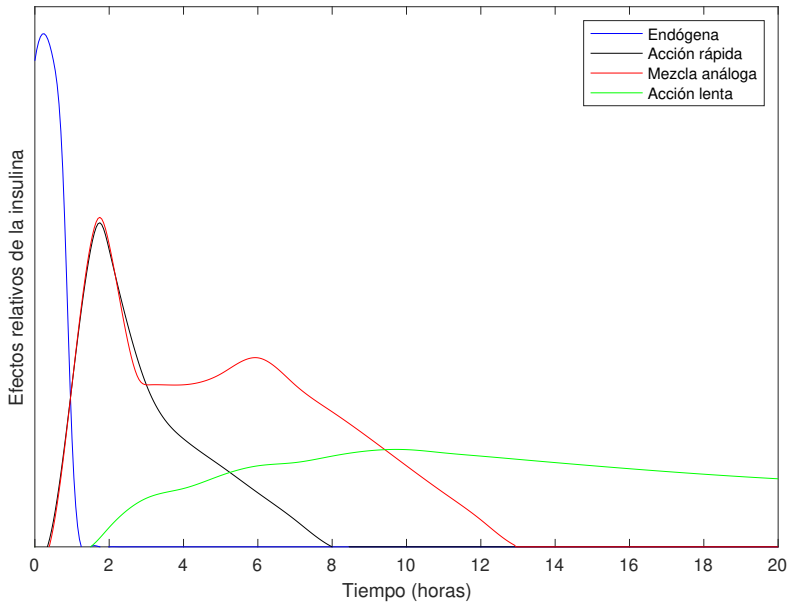


Figura 3.3: Efecto relativo de varios tipos de insulina en el tiempo

efecto concreto en el paciente), el efecto de la disminución de la infusión podría no ser suficiente. Para minimizar estos problemas, el controlador recibirá información generada por el paciente en forma de anuncio de ingestas y/o ejercicio antes de que estos eventos sucedan mediante un smartphone u otro dispositivo habilitado para ello (ver Figura 3.4). El tiempo de antelación para el anuncio del evento varía según la situación concreta del paciente pero, por ejemplo, el tiempo de adelanto para ingestas debe tener en cuenta el tiempo que transcurre desde la infusión de insulina hasta que sucede su máximo de acción, el estado glucémico del paciente (en hipoglucemia puede que sea preferible no adelantar insulina) y el tipo de ingesta que se prevea realizar. Es común que los pacientes utilicen adelantos entre 15 y 20 minutos con uso de insulinas rápidas y comidas con bajo contenido en grasas. Estos temas se abordarán en apartados posteriores. El controlador, de esta manera, será capaz de adaptar la infusión al futuro cercano del paciente sin necesidad de esperar a cambios en la glucemia.

3.3 Requisitos del sistema *wearable*

El controlador es el módulo responsable del análisis de la información y del cálculo de las operaciones del lazo de control. Para que este pueda ser considerado *wearable* debe cumplir con los siguientes requisitos:

- Disponer de una **batería** con capacidad suficiente para poder operar durante al menos 24 horas. Veinticuatro horas es el mínimo periodo de tiempo que debe ser capaz de operar de manera continua para permitir que el paciente recargue la batería del controlador, a diario, mientras duerme.

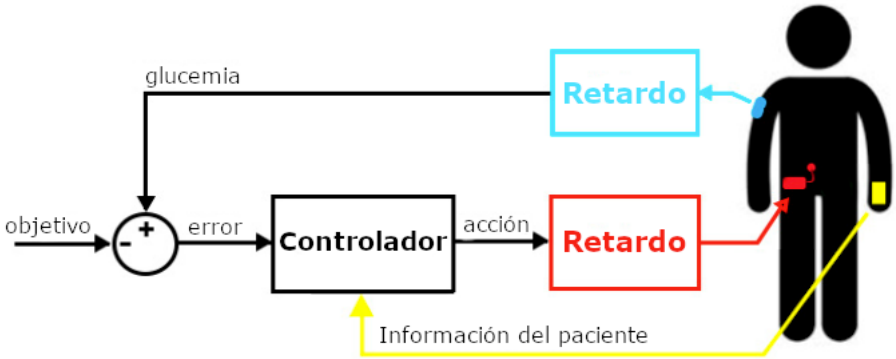


Figura 3.4: Esquema de controlador simplificado indicando retardos y la entrada de información del paciente

- Su **tamaño** debe ser el menor posible. Los pacientes deben ser capaces de guardarlo en algún lugar en su ropa. Para ser considerado *wearable*, el sistema debe ser cómodo, ligero y casi inapreciable mientras el paciente lo lleve consigo.
- Ser **tolerante a fallos de comunicación**. El controlador comunicará con el MCG y la bomba de insulina mediante radiofrecuencia, por consiguiente, la probabilidad de que existan errores de comunicación es considerable. El controlador debe ser tolerante a fallos y no debería comandar acciones que, en el caso de una pérdida de comunicación con la bomba de insulina, pudieran dejar al resto del sistema o al paciente en una situación peligrosa.
- Disponer de un **interfaz de usuario** que permita al paciente anunciar diversas situaciones, antes las cuales el sistema deba tomar alguna acción con cierta antelación, y poder configurar ciertos parámetros si fuera necesario.

3.4 Prestaciones y sistema *wearable* propuesto para la dosificación de insulina

Una vez establecidos los requisitos para la plataforma, pasamos a describir la plataforma en si misma. Está compuesta por una bomba de insulina, un Monitor Continuo de Glucemia, y un sistema controlador ad-hoc basado en procesadores system-on-chip de bajo consumo.

El sistema propuesto utiliza una bomba de insulina comercial con un Monitor Continuo de Glucemia integrado de Medtronic (Medtronic Paradigm Veo + Sensor Enlite) debido a que su protocolo de comunicaciones por radiofrecuencia ha sido documentado [48][53] y se puede comandar basales temporales, suspensión y reanudación de infusión y bolos de insulina de manera remota.

Tan importante como tener un sistema de infusión controlable es tener un sistema correctamente calibrado. Los sensores Medtronic Enlite [54] requieren ser calibrados cada

12 horas (como máximo) y deben ser calibrados por el paciente siguiendo una serie de reglas:

- El Monitor Continuo de Glucemia calculará el factor de traducción de sus señales internas a glucemias intersticiales en función de la glucemia introducida por el paciente en el sistema. Este valor de glucemia debe ser obtenido mediante un glucómetro siguiendo los pasos del fabricante del mismo, típicamente: Lavar y limpiar las manos antes de obtener una gota de sangre y descartar la primera gota para evitar posibles agentes externos presentes en la piel.
- Debido al retardo entre la glucemia medida directamente en el torrente sanguíneo y en el líquido intersticial, las calibraciones solo deben realizarse cuando la glucemia esté estable. De esta manera, el error introducido por el ratio de cambio de la glucemia y el retardo variable entre los puntos de medidas se minimiza al tener un valor casi invariante con el tiempo.

Este proceso es necesario para mantener la coherencia entre las medidas en el líquido intersticial (realizadas por el MCG)) y en sangre (realizadas mediante el glucómetro durante la calibración). La calibración del MCG es de vital importancia en un sistema de control en lazo cerrado: si el sensor se ve degradado con el paso del tiempo o no se calibra correctamente, el uso de datos erróneos puede llevar al paciente a hipoglucemia o hiperglucemia.

Una vez el paciente está haciendo uso de un sistema integrado de infusión de insulina y monitorización continua de glucemia, confiable y controlable, el siguiente paso es obtener una plataforma *wearable* que permita al paciente "cerrar el lazo": obtener datos de glucemia en tiempo real, junto con los parámetros relevantes de terapia del paciente y actuar sobre la infusión de insulina.

Otras plataformas Do-It-Yourself (Hazlo-Tú-Mismo) [8][10][9] utilizan microprocesadores potentes y sistemas operativos como Linux, Android o iOS. Estas plataformas no están pensadas para un uso de bajo consumo y deben mantener su sistema operativo corriendo, teniendo otros procesos corriendo en segundo plano y consumiendo energía de una manera poco adecuada. Esta es la razón por la que estos sistemas no pueden ser considerados como *wearable*. En muchos casos, para ser capaces de mantener un funcionamiento continuado durante 24 horas, el paciente debe hacer uso de baterías relativamente voluminosas, haciendo el sistema menos portable y, quizá, no confiable a largo plazo. Cuando el controlador se ejecuta en un smartphone existen dos problemas adicionales:

1. El teléfono del paciente no puede quedarse sin batería o, en caso contrario, el paciente se quedará en lazo abierto.
2. Existe la posibilidad de que el teléfono sea *hackeado* y el atacante logre modificar la terapia del paciente de manera remota. Aunque la seguridad de estos sistemas ha

mejorado con el paso del tiempo, estos sistemas reciben actualizaciones constantes de seguridad. La superficie de ataque es tan grande que es muy complicado hacer una implementación segura del controlador en un teléfono que está permanentemente conectado a Internet.

Estos dos problemas han hecho que la arquitectura elegida en esta tesis doctoral sea una basada en un diseño embebido En esta propuesta, con el objetivo del bajo consumo en mente, se ha diseñado una nueva tarjeta adaptada a las necesidades expuestas. Esta tarjeta está dividida en 4 partes bien diferenciadas (ver diagrama en la Figura 3.5).

1. Gestión de energía y batería: dado que es un modulo alimentado por una batería, se añadieron los componentes electrónicos necesarios para ser capaz de cargarla, monitorizarla y regular su salida hacia el resto del sistema.

2. Microcontrolador de bajo consumo: el algoritmo de control se codifica para ser ejecutado en un microprocesador de bajo consumo basado en Intel 8051. Como el controlador recibe una nueva medición cada 5 minutos, el procesador puede permanecer dormido durante la espera y reducir al máximo la energía consumida. Aunque este procesador utiliza tecnología diseñada hace muchos años, su bajo consumo hace de él una plataforma perfecta.

3. Comunicaciones por Bluetooth Low Energy (BLE): Esta tarjeta dispone de un interfaz Bluetooth Low Energy capaz de conectarse a un smartphone. Utilizando el smartphone, el paciente puede configurar el controlador, notificar eventos como ingestas o ejercicio y ser notificado de posibles alarmas. También puede ser utilizado para propósitos como monitorización remota, por ejemplo usando Nightscout [7].

4. Comunicaciones con la bomba de insulina y con el MCG, ambos de Medtronic: Este es el interfaz utilizado para recibir información del MCG y de la bomba de insulina, y para comandar las acciones necesarias a la bomba de insulina de Medtronic. Cada fabricante de bomba de insulina y sensor MCG utiliza su sistema de comunicaciones adaptado a sus necesidades. En caso de desear migrar a dispositivos de otro fabricante, se deben analizar sus comunicaciones y, quizá, utilizar otro microprocesador existente en esta plataforma. En el caso de bombas de insulina Paradigm de Medtronic, el protocolo de comunicaciones es propietario y será descrito posteriormente en el apartado 3.6.1.4.

3.5 Descripción del diseño electrónico

Como resultado del análisis de requisitos y diseño de diagrama de bloques, se diseñó una plataforma hardware nueva (ver Figura 3.6) con el objetivo de ser capaces de implementar

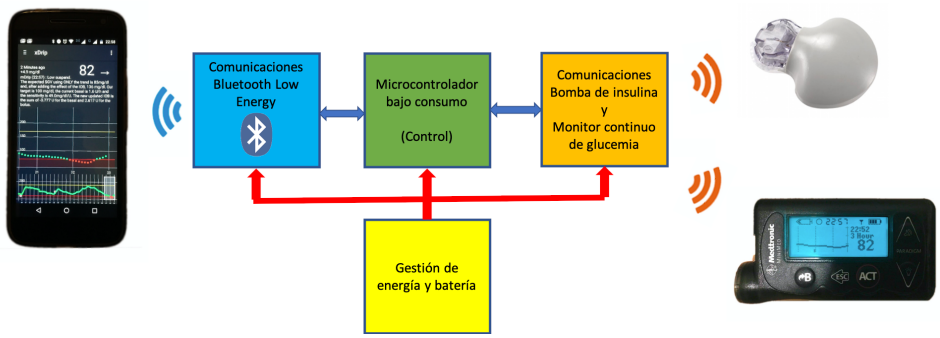


Figura 3.5: Diagrama de bloques del sistema *wearable* propuesto

un controlador en lazo cerrado para infusión de insulina. Tres *System-On-Chip* han sido incluidos para dar soporte a las comunicaciones con los dispositivos de Medtronic (Texas Instruments CC1110 [55]), comunicaciones Bluetooth Low Energy en modo periférico (Texas Instruments CC2540 [56]) y, aunque no es necesario para el sistema tal y como está descrito, otras comunicaciones en la banda de 2.4GHz para soportar otros dispositivos en el futuro (Texas Instruments CC2510 [57]).

Los tres SoCs, a parte del transceiver de radiofrecuencia, incorporan un procesador Intel 8051 y han sido utilizados para implementar, entre otras funcionalidades, el lazo de control (ver Figuras 3.6 y 3.7). Todos ellos pueden entrar en distintos modos de bajo consumo y pueden reducir su consumo hasta 0.3 μA (modo PM3 [55]) dependiendo de las necesidades de cada diseño. Existen 3 modos de funcionamiento: uno de funcionamiento normal y dos de bajo consumo (PM2 y PM3). Esto es de vital importancia ya que, como se explicará en capítulos posteriores, se ejecuta el algoritmo de control cada 5 minutos (con cada dato nuevo del MCG) y se debe reducir el consumo lo máximo posible entre ejecuciones del lazo para maximizar la duración de la carga de la batería.

Como también se verá más adelante, la batería se conecta a la tarjeta mediante un conector, permitiendo así elegir su capacidad a posteriori dependiendo del consumo medio del sistema completo. Aunque todavía no se haya descrito el funcionamiento de la plataforma ni del algoritmo completo, es relevante hacer referencia a los consumos obtenidos con el sistema final: el sistema opera en un bucle infinito en el que se realiza, aproximadamente, el mismo conjunto de operaciones cada 5 minutos. Esto permite calcular el consumo medio por hora y así poder dimensionar correctamente la batería para el sistema.

Las pruebas de consumo dieron como resultado un valor promedio por hora de, aproximadamente, 20 mA utilizando una batería LiPo de 3.7V. Con estos datos, se puede evaluar la duración de las capacidades de las baterías más estándar existentes en el mercado (ver Tabla 3.1). Las filas en color rojo indican las capacidades que no proporcionarían una duración de 24 horas entre recargas y, por tanto, quedan descartadas. En color naranja se indica la batería que, aun soportando 24 horas de ejecución, no dejan margen para que el consumo medio pueda crecer debido a otros factores y con ello aumenta el riesgo de

Modelo	Capacidad (mAh)	Tamaño (mm)	Horas de funcionamiento
222770	350	2,2 * 27 * 72	17,5
502530	400	5 * 25 * 37	20
901937	500	9 * 19 * 39	25
403048	600	4 * 30 * 50	30
602543	**650**	**6 * 25 * 45**	**32,5**
902540	700	9 * 25 * 42	35

Tabla 3.1: Comparación de duraciones de baterías LiPo estándar en el sistema final con consumos medios

no alcanzar las 24 horas de ejecución. En verde se indican las capacidades a partir de las cuales se garantizan las 24 horas de ejecución y, además, se dispone de cierto margen para soportar las siguientes situaciones:

1. Aumento del consumo medio debido a **problemas de comunicaciones por radio**: si la comunicación por radio requiere retransmisiones o se pierde la comunicación con la bomba o el MCG, el sistema deberá mantener el interfaz de radio activo por más tiempo. El tener activo el interfaz de radio tiene un impacto muy grande en el consumo medio del sistema.

2. Necesidad de incrementar la **complejidad del algoritmo** y los cálculos que realiza: Esto reducirá el tiempo que el microprocesador puede permanecer en modo de bajo consumo.

3. **Degradación de la batería**: Las baterías se degradan con el tiempo y más aún en sistemas que operan las 24 horas del día de manera continuada.

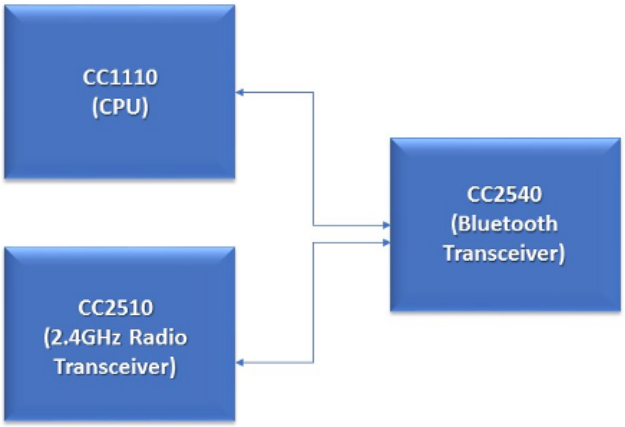


Figura 3.6: Diagrama de bloques de interconexión de SoCs

Como este módulo debe ser *wearable*, se opta por incluir una batería de polímero de litio de 650 mAh de capacidad junto con su electrónica de regulación y carga asociada (ver U1 y U3 en la Figura 3.8). Esta batería provee de un tiempo de operación de más de 24 horas, permitiendo al paciente llevar el módulo encendido durante todo el día y recargarlo por la

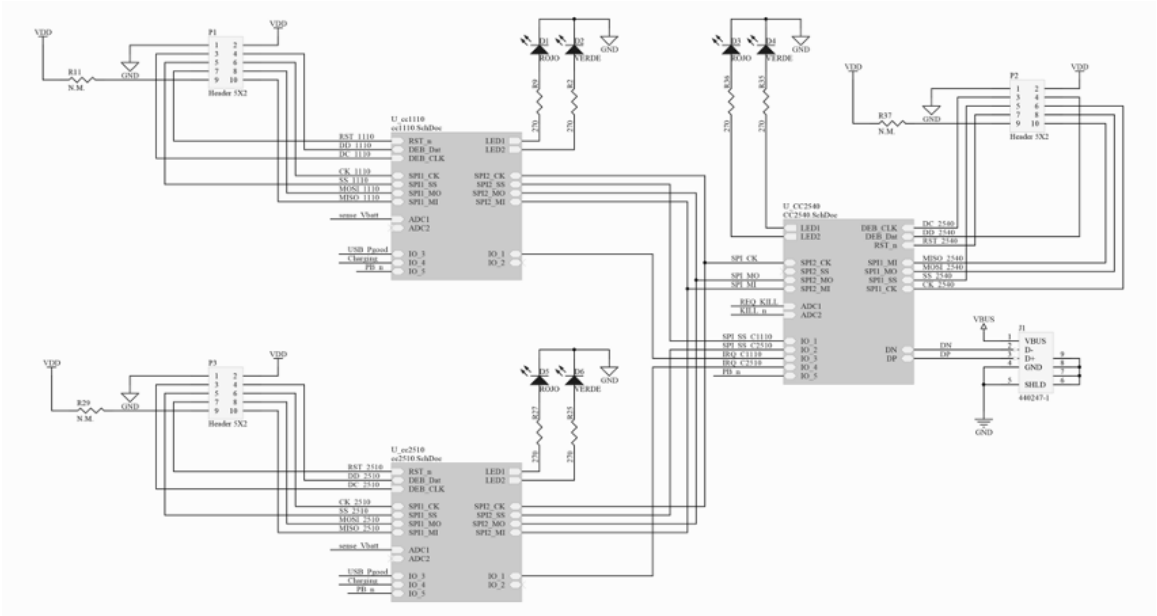


Figura 3.7: Esquemas de la plataforma - interconexión SoCs

noche mientras, además, sigue operando. Además se permite cierto margen de incremento de consumo para dar respuesta a los problemas de comunicaciones y a la edad de la batería con los años.

Se ha optado por un sistema de carga de batería por USB, con conector micro-USB como se puede apreciar en la Figura 3.7, y se ha aprovechado para conectar las lineas de datos al procesador U_cc2540 que dispone de interfaz USB por si fuera de interés para el futuro.

A nivel general, tal y como se puede apreciar en la Figura 3.7, se interconectan los procesadores U_cc1110 y U_cc2510 a través del procesador U_cc2540 que hace las funciones de procesador central. Estos procesadores se comunican a través de buses serie y un par de líneas discretas auxiliares que permiten notificar eventos entre procesadores. Todos los procesadores han sido equipados con dos LEDs (cada uno) para generar notificaciones luminosas y también cada uno dispone de su conector de depuración/programación independiente. Al pertenecer todos a la misma familia de procesadores, todos hacen uso del mismo cable y entorno de depuración. Al tener conectores de depuración independientes es posible depurar todos los procesadores de manera simultánea. Las líneas de monitorización de carga por USB y tensión de la batería también se distribuyen entre los microprocesadores.

La electrónica de regulación y carga de la batería se detalla en la Figura 3.8. Se hace uso de un gestor de baterías LiPo BQ24232 y se incluyen dos LEDs para indicar el estado del proceso de carga (uno para indicar que el proceso de carga está en marcha y otro para indicar que la carga ha finalizado). La tensión de la batería se regula y filtra mediante un LDO TPS79301 y se genera una tensión de 3.3V para alimentar al resto del circuito. La batería se conecta a través de un conector al gestor de la batería.

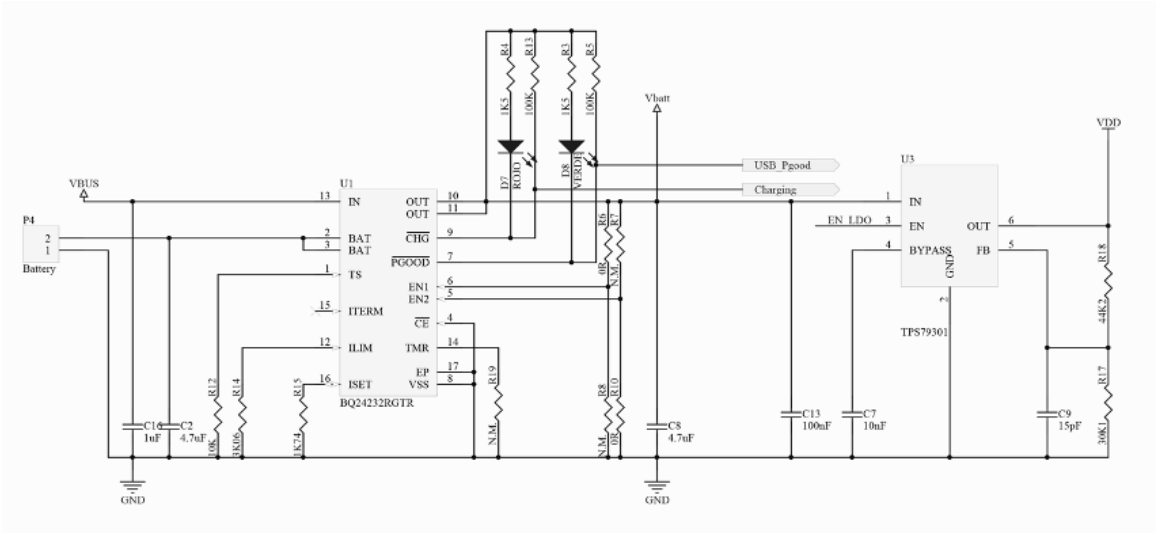


Figura 3.8: Esquemas de la plataforma - Regulación de potencia y carga de batería

Además, esta nueva plataforma ha sido integrada con la plataforma Nightscout [7] para permitir monitorización remota y con xDrip [58] como plataforma móvil para controlarlo y subir información a la nube. Por tanto, esta plataforma (mostrada en las Figuras 3.9 y 3.10) puede ser conectada con las soluciones disponibles en la nube para compartir y analizar información y resultados. El formato de los datos y los mensajes intercambiados será descrito en posteriores apartados.

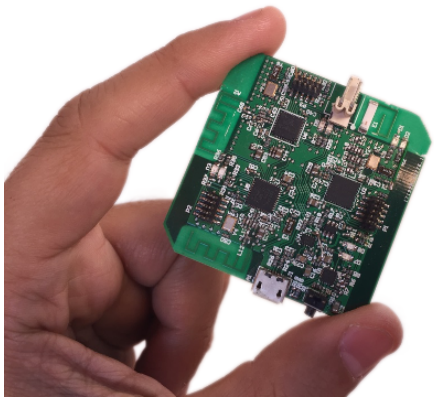


Figura 3.9: Plataforma hardware desarrollada para el sistema de control en lazo cerrado

Tal y como se puede apreciar en las Figuras 3.9 y 3.10, las antenas utilizadas para los interfaces de radio son antenas diseñadas directamente sobre el PCB siguiendo las indicaciones del fabricante [59]. Esto nos garantiza un buen rendimiento, radio-eléctricamente hablando, y asegurarnos de que no existen problemas a nivel de EMI (*Electromagnetic interference*)

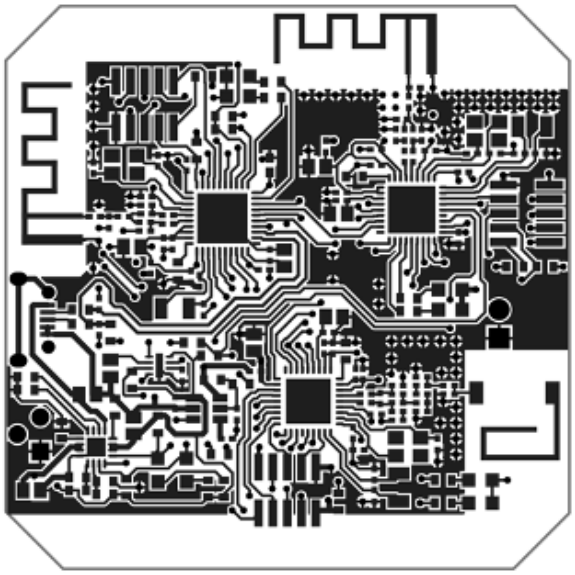


Figura 3.10: Circuito impreso de la plataforma - Capa Top

3.6 Arquitectura firmware básica para el soporte del lazo de control

3.6.1 CC1110 - SoC de comunicaciones con dispositivos Medtronic

El SoC que controla las comunicaciones con la bomba de insulina y el MCG (U_CC1110 en la Figura 3.6) es el encargado de, además de las comunicaciones con estos dispositivos, del proceso de la información recibida y la ejecución del lazo de control.

La Figura 3.11 muestra el diagrama de ejecución principal del SoC CC1110. Como se puede apreciar en la parte izquierda del diagrama, primero se inicializa el hardware y las variables globales del sistema, para después dar paso a la ejecución de un bucle infinito en el que se controlan las comunicaciones, los tiempos, la ejecución del lazo de control y se realiza la gestión de potencia.

Se hace uso de un LED verde para indicar el estado del procesador. El LED tiene 3 posibles estados:

- Apagado - el sistema se encuentra en estado de bajo consumo
- Encendido - el sistema se encuentra activo y a la espera de la recepción de mensajes por radio
- Parpadeando - el sistema tiene la radio activa y transmitiendo comandos a la bomba de insulina

Los mensajes provenientes del interfaz de Bluetooth (U_CC2540 en la Figura 3.6) se reciben a través de un interfaz serie configurado para trabajar a 115200 bps con 8 bits, sin paridad y un bit de stop. La recepción de datos serie provoca una interrupción que hace

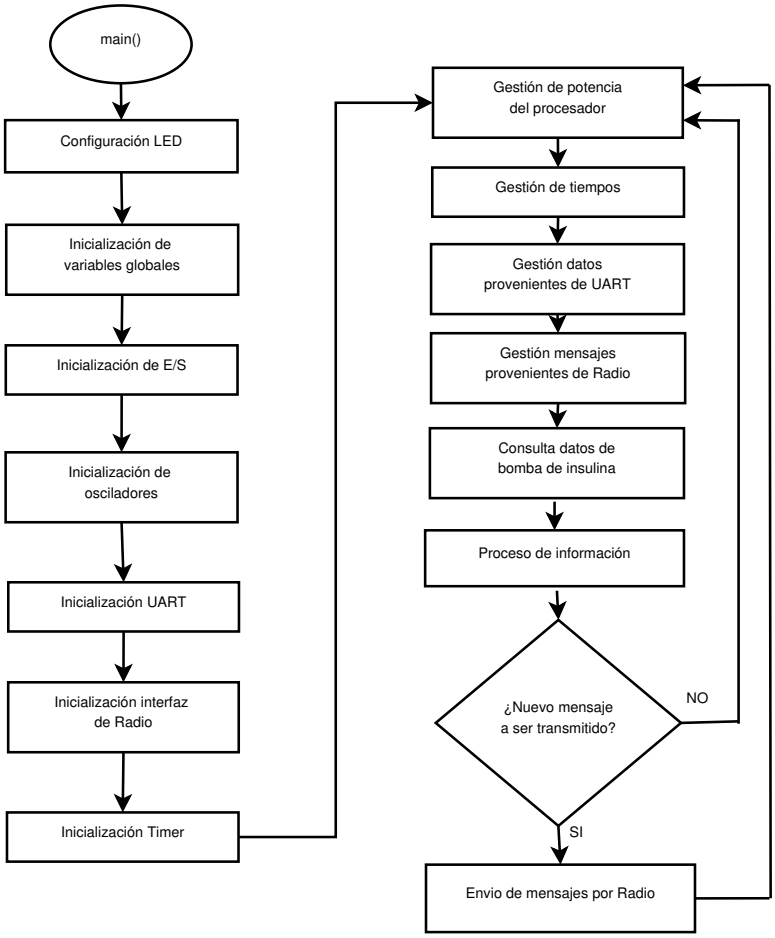


Figura 3.11: Diagrama de ejecución principal - CC1110

salir al SoC U_C1110 de su estado de bajo consumo, evitando hacer polling del interfaz y reduciendo el consumo.

El interfaz de radio se configura para la recepción y transmisión de mensajes según las características establecidas por Medtronic y documentadas online [48]. Debido a que la intención es mantener el consumo lo más reducido posible, el interfaz de radio se mantiene apagado mientras no se espere ningún mensaje entre los dispositivos. Sin contar con los periodos de transmisión y recepción del sistema, únicamente se permitirá el encendido, de manera periódica y por un breve instante de tiempo, para implementar los mecanismos de seguridad descritos más adelante en el apartado 3.7.

La medida de los tiempos del sistema se realiza mediante un *timer* que, configurado de manera apropiada, permite sacar al procesador de su estado de bajo consumo por interrupción. Debido a que los tiempos a medir son relativamente grandes con respecto a la resolución del *timer*, se realiza un escalado para reducir la resolución a 200 ms.

Una vez la plataforma ha sido inicializada, la ejecución entra en un bucle infinito en el que se realizan las siguientes funciones:

1. Gestión de potencia del procesador

2. Gestión de tiempos
3. Recepción de datos de serie (Puente con Bluetooth Low Energy)
4. Recepción mensajes de Radio (Bomba de insulina y/o MCG)
5. Proceso de información
6. Envío de mensajes por Radio (si fuera necesario)

3.6.1.1 Gestión de potencia del procesador

La gestión de potencia del procesador es el proceso que controla las tareas pendientes por tratar y, en caso de no existir ninguna pendiente, hace entrar al procesador en modo de bajo consumo hasta que suceda una interrupción de un periférico.

Esta gestión se realiza mediante la activación y desactivación de unos flags que, cuando se encuentran activos, indican la necesidad de mantener el procesador activo.

Los motivos que pueden mantener el procesador activo son:

- Necesidad de enviar o recibir mensajes por radio
- Ajuste de frecuencia para las comunicaciones
- Transmisión o recepción de datos por el interfaz serie
- Actualización de temporizaciones internas

Estos *flags* junto con su descripción pueden ser consultados en el apartado A.1.

3.6.1.2 Gestión de tiempos

Una correcta medida del tiempo es lo que permite que se pueda reducir el consumo al máximo, operando únicamente cuando es necesario.

Como ya se indicó previamente, el tiempo se mide en *slots* de 200 ms mediante el escalado de un *timer* del procesador. Este escalado tiene una parte *hardware* y otra *firmware*, y un flag se activa cada vez que un slot de tiempo transcurre.

Cada vez que transcurre un *slot* de tiempo, a parte de mantener un reloj de tiempo real sincronizado con la bomba de insulina, se actualizan unos contadores internos que mantienen la temporización de todos los procesos internos. Estos temporizadores y sus descripciones pueden ser consulados en el apartado A.2.

El conocimiento preciso de la hora del día en el que el sistema está trabajando (reloj en tiempo real) es importante para poder determinar correctamente qué parámetros debe tomar como activos cuando se consulte las configuraciones de la bomba de insulina.

Hay parámetros que vienen codificados como tablas por horas, entre ellos: perfiles basales, sensibilidades a la insulina y objetivos glucémicos. Estos parámetros, junto con la hora de la bomba de insulina, se consultan por radio cuando el *flag* `queryPumpFlag` se activa (ver apartado A.1). Esto sucede normalmente en el inicio de funcionamiento de la plataforma. Todos estos datos quedan accesibles para el algoritmo en caso de que sean de utilidad.

3.6.1.3 Recepción de datos serie (puente BLE)

Las comunicaciones con el *smartphone* se realizan a través de un interfaz serie del SoC. Para ello se ha diseñado un protocolo sencillo en el que se ha seguido la siguiente estructura:

1. Byte 0: Longitud del mensaje en bytes (N)
2. Byte 1: Identificador del mensaje
3. Bytes 2 a N-2: Contenido del mensaje
4. Bytes N-1 y N: CRC-16

Sobre este protocolo se han definido una serie de mensajes que se han creído útiles para el funcionamiento del sistema y se enumeran en la Tabla 3.2. Como es lógico, las necesidades de comunicaciones es muy dependiente del algoritmo a implementar y, por ello, la implementación de nuevos mensajes es tan sencilla como modificar la función `void receiveBLEMessage (char *message, unsigned int messageLen)` y añadir los mensajes que se crean necesarios.

La comunicación con este sistema no depende de que el *smartphone* haga *polling* de manera periódica. Este sistema enviará mensajes de manera "no solicitada", utilizando el mismo protocolo, cuando exista alguna actualización en su estado. En este caso, se han implementado las notificaciones enumeradas en la Tabla 3.3.

3.6.1.4 Recepción de datos via Radio

Las comunicaciones con los dispositivos de Medtronic [48], representadas en la Figura 3.5 por el bloque color naranja, hacen uso de las bandas ISM[60] (*Industrial, Scientific and Medical*) por debajo de 1 GHz. Estas bandas son:

- 868 MHz: Europa, China, India, Malasia, Singapur, África, Emiratos Árabes
- 915 MHz: EE.UU., Canada, Paises de Sudamérica, Nueva Zelanda, Taiwan, Hong Kong, Japón, Israel

Identificador	Descripción
0x00	Identificación del sistema y versión de firmware
0x10	Lectura de la tabla de tiempos del MCG
0x20	Configuración de la tabla de tiempos del MCG
0x11	Lectura de los contadores de tiempo del sistema
0x12	Lectura de IDs de la bomba de insulina, MCG y glucómetro asociados.
0x22	Configuración de IDs de la bomba de insulina, MCG y glucómetro asociados.
0x13	Lectura de las últimas 16 medidas recibidas del MCG
0x14	Lectura de Ias últimas 16 medidas recibidas del MCG (RAW)
0x15	Lectura de Ias últimas 16 medidas recibidas del MCG (iSIG)
0x27	Activación temporal de la radio
0x18	Consulta de los ajustes finos de frecuencia para los dispositivos asociados.
0x28	Configuración de los ajustes finos de frecuencia para los dispositivos asociados.
0x19	Consulta de la tabla de RSSI asociada al MCG.
0x1A	Consulta de la tabla de RSSI asociada a la bomba de insulina.
0x1B	Consulta de la tabla de RSSI asociada al glucómetro.
0x2C	Inicio del proceso de ajuste fino de frecuencia para la bomba de insulina.
0x1E	Lectura de la configuración del modo repetidor de MCG.
0x2E	Configuración del modo repetidor de MCG.
0x35	Lectura de los últimos resultados del lazo de control.

Tabla 3.2: Tabla de comandos implementados para la recepción BLE

Identificador	Descripción
0x01	Actualización de datos glucémicos del paciente
0x02	Recibida lectura del glucómetro
0x03	Notificación de acción sobre la bomba de insulina
0x35	Notificación de cambios del algoritmo de control

Tabla 3.3: Tabla de notificaciones implementadas para la transmisión BLE

El sistema está preparado para trabajar en ambas bandas y se configura mediante un comando por el interfaz de Bluetooth. Las comunicaciones son bastante sencillas y sus características son las siguientes:

- Modulación OOK [61][55]
- Longitud variable
- Codificación de línea 4b6b
- 16.384 bps
- Preámbulo 0xFF00
- CRC-8-WCDMA para comprobación de integridad

El Firmware configura el interfaz de radio para trabajar con estas características y, mediante una variable global de selección de modo, es capaz de variar de manera dinámica la frecuencia central en la que se sintoniza y el ancho de banda del canal. Como ya se ha explicado anteriormente, el sistema intenta ajustar de manera periódica las frecuencias

centrales a utilizar con cada uno de los dispositivos del sistema. La variación de ancho de banda de canal es una funcionalidad que permite al sistema empezar a trabajar sin conocimiento exacto de las frecuencias de los dispositivos pero, una vez estas son conocidas, se puede aumentar la relación señal a ruido de la comunicación reduciendo el ancho de banda del canal. Esto, como es lógico, permite tener comunicaciones más fiables y resistentes a ruidos en canales adyacentes o lograr una comunicación a mayor distancia.

Se han tomando los siguientes anchos de banda para la operación del sistema:

- Ancho : 600 KHz
- Estrecho : 63 KHz

El ajuste de la frecuencia central de la bomba de insulina es sencillo de hacer puesto que es este sistema el que controla las comunicaciones. El proceso consiste en:

1. Configurar un ancho de banda de canal estrecho
2. Seleccionar la frecuencia más baja según la banda de trabajo (868 o 915 MHz)
3. Sintonizar la frecuencia a probar
4. Enviar de manera consecutiva 256 mensajes para “despertar” la bomba de insulina
5. Esperar la respuesta de la bomba y, si se recibe, anotar en una tabla la RSSI recibida
6. Si no se ha llegado a la última frecuencia a probar, incrementar la frecuencia un salto de 25KHz y volver al paso 3.

Al terminar este proceso, bastará con tomar la frecuencia con mayor RSSI como la adecuada para las comunicaciones con la bomba y usar un ancho de banda estrecho.

En el caso del MCG el problema es ligeramente más complicado pues no controlamos las comunicaciones y no hay manera de forzar al transmisor a que mande ningún mensaje. Debemos aprovechar las comunicaciones tal y como fueron diseñadas. Por suerte, y debido a que el transmisor del MCG utiliza poca potencia para maximizar la duración de su pequeña batería, cada nueva lectura del sensor se envía dos veces separadas unos pocos segundos. De esta manera, en caso de interferencia o colisión durante la primera transmisión, se puede intentar una segunda vez en otro momento. En el diseño del sistema se tuvo este hecho en cuenta y, mientras no se tenga la tabla completa de RSSI para el MCG, se utiliza un ancho de banda grande para intentar maximizar las posibilidades de recibir los datos y poder operar. En caso de recibir el primer mensaje de datos del MCG, se utilizará el segundo mensaje para ir probando el ancho de banda estrecho sobre todas las frecuencias y así lograr determinar la mejor frecuencia a utilizar sin perjudicar la operatividad del sistema. Una vez se tiene la frecuencia exacta a utilizar, siempre se utilizará esta con un ancho de banda estrecho.

En aquellos sistemas que utilicen glucómetro el procedimiento será similar. Mientras se desconozca la frecuencia exacta, se utilizará el ancho de banda grande para maximizar las posibilidades de recepción. Pero, una vez se ha detectado el mensaje del glucómetro, se comenzará a probar todas las frecuencias de manera consecutiva para determinar todas las RSSI. El glucómetro manda cientos de veces el mismo mensaje para así lograr que los datos sean recibidos por la bomba de insulina en uno de sus periodos de escucha. En nuestro caso, se aprovecharán estos mensajes para el ajuste de RSSI. Una vez determinada la frecuencia del glucómetro, siempre se utilizará esta junto con el ancho de banda estrecho.

Las comunicaciones por radio, en nuestro caso, no contienen la longitud del mensaje. Para identificar el final de cada mensaje se ha optado por utilizar una codificación de línea poco habitual: 4b6b. Esta codificación expande el espacio de símbolos de 4 bits a 6 bits de manera que sólo en caso de finalizar el mensaje encontraremos más de 3 ceros seguidos. La Tabla 3.4 muestra la conversión de símbolos en esta codificación de línea.

Símbolo 4 bits	Símbolo 6 bits
0000	010101
0001	110001
0010	110010
0011	100011
0100	110100
0101	100101
0110	100110
0111	010110
1000	011010
1001	011001
1010	101001
1011	001011
1100	101100
1101	001101
1110	001110
1111	011100

Tabla 3.4: Codificación de línea 4b6b

La recepción de mensajes de radio se inicia cuando el detector de preámbulos del SoC detecta el preámbulo configurado (en nuestro caso 0xFF00) y, a partir de ahí, se muestrea la señal demodulada a una tasa de 16.384 bits por segundo. Cada vez que se obtienen 8 bits se interrumpe al microprocesador y el firmware lo almacena en un buffer de recepción. Cuando el la rutina de interrupción detecta 4 o más ceros consecutivos en los datos recibidos, se para la recepción de la radio y se decodifica 4b6b el mensaje.

El mensaje decodificado debe contener un CRC-8 (semilla 0x9B) como último byte. Si el CRC-8 coincide con el último byte recibido, el mensaje se transfiere al proceso de recepción de mensajes de Medtronic.

En este proceso, se comprueba que el identificador incluido en el mensaje (los tres primeros bytes) coinciden con los de uno de los dispositivos de nuestro sistema. En caso

afirmativo, se analiza el mensaje y se almacena la información relevante para su posterior procesamiento en los buffers internos destinados a ello.

3.6.1.5 SmartRecovery

Se ha comprobado de manera empírica que la comunicación entre la bomba de insulina y el transmisor del MCG falla con relativa facilidad si estos dispositivos no se encuentran muy próximos. En este sistema, tal y como fue diseñado por Medtronic, no existe mucho problema pues el transmisor del MCG envía con cada mensaje los últimos 45 minutos de datos del sensor. El problema se incrementa ligeramente cuando el sistema trabaja en lazo cerrado tomando decisiones cada 5 minutos: por ello en este sistema se intentar reducir el número de comunicaciones fallidas al mínimo posible.

En este intento, dado que el controlador es capaz de ajustar de manera fina la frecuencia con la que hay que transmitir a la bomba de insulina y la frecuencia con la que se deben recibir los mensajes del MCG, se puede utilizar el controlador como una ayuda al resto de elementos del sistema. El controlador, cuando recibe el mensaje del MCG, procederá a la retransmisión del mismo inmediatamente después utilizando la frecuencia exacta de la bomba de insulina y la potencia máxima de transmisión. De esta manera se maximizan las probabilidades de que la bomba de insulina reciba la información del sensor.

Además, gracias a la estructura muy regular del mensaje del MCG [48], al hecho de que gran parte de los contenidos del mensaje se pueden obtener del mensaje anterior o de manera programática por ser contadores o CRCs, se implementa un mecanismo de recuperación de mensajes que se ha denominado como *SmartRecovery*.

Este proceso, en caso de que el mensaje recibido del MCG se reciba con errores y el CRC no coincida con el esperado, procederá a intentar recuperar el mensaje modificando la información que es conocida por otros medios.

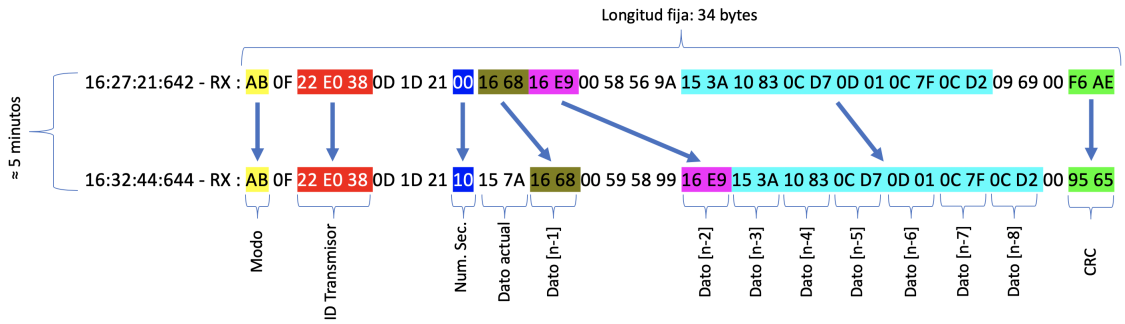


Figura 3.12: Ejemplo de flujo de información entre mensajes del MCG

La Figura 3.12 muestra un ejemplo en el que se puede apreciar cómo la información de un mensaje recién recibido puede ser reconstruido, en caso de error, utilizando la información del mensaje anterior (o anteriores):

- El byte que indica el modo de transmisión (marcado en amarillo) únicamente puede tener dos valores difiriendo en el último bit (0xAA o 0xAB). Con este byte el transmisor especifica si los datos pueden ser utilizados para determinar correctamente valores de glucemia (0xAB) o si todavía no ha transcurrido el tiempo definido como *warm-up* y los datos no son todavía válidos (0xAA). Ambas opciones serán probadas para ver si el CRC resultante es el recibido y, con ello, el error queda corregido.
- Los tres bytes marcados en rojo identifican el número de serie del transmisor. Estos bytes no pueden cambiar y, por tanto, se pueden reemplazar directamente en el mensaje recibido con errores.
- El byte marcado en azul oscuro determina el número de secuencia del mensaje. El nibble superior se incrementa en una unidad con cada nuevo dato utilizando los números de 0 a 7 y el nibble inferior sólo puede utilizar los valores 0 o 1 para indicar si es el primer mensaje mandado por el transmisor o el segundo. El nibble superior puede ser deducido mediante el mensaje anterior y el tiempo que ha pasado desde la recepción de dicho mensaje (+1 % 8 cada 5 minutos). El nibble inferior se puede deducir probando sus dos únicas opciones.
- Los bytes marcados en color verde oscuro, fucsia y azul claro corresponden con los datos históricos de glucemia del mensaje recibido. El único dato de glucemia que no se puede recuperar es el actual, pero todos los anteriores pueden ser recuperados y comprobar si el CRC coincide.
- La longitud del mensaje es fija y debe ser de 34 bytes. Si el mensaje recibido contiene más información que la esperada, el mensaje puede ser truncado para que tenga la longitud correcta.

El CRC está marcado en verde claro en la Figura 3.12. Si el error se produce en cualquiera de los bytes no marcados con color, no será posible la recuperación del mensaje pues la información es desconocida y no puede ser derivada/recuperada usando mensajes anteriores. Si el error ocurre en el CRC mismo, tampoco será posible la recuperación del mensaje.

En caso de error, los cambios realizados al mensaje son:

1. Se trunca la longitud del mensaje para que sea exactamente 34 bytes.
2. El primer byte debe ser 0xAA o 0xAB. Se prueba con ambos casos.
3. Se recupera el número de secuencia y se prueba con las dos opciones para su bit bajo.
4. Los datos del sensor en la trama recibida anteriormente (salvo el último) deben ser los que se encuentren en este mensaje como historia.

Si después de estos cambios el CRC coincide con el esperado, se da el mensaje como recuperado y se reenvía a la bomba para intentar que esta reciba también la información.

Hora de inicio	Sensibilidad (mg/dl/U)
00:00	70
08:00	57
11:00	55
16:00	51
21:00	60
22:30	70

Tabla 3.5: Ejemplo de Tabla de Factores de Sensibilidad a la Insulina

Hora de inicio	Objetivo mínimo (mg/dl)	Objetivo máximo (mg/dl)
00:00	80	110
14:30	90	120
18:00	80	110

Tabla 3.6: Ejemplo de Tabla de Objetivos Glucémicos

3.6.1.6 Consulta de datos de bomba de insulina

Este proceso comprueba que el sistema disponga de datos actualizados de la bomba de insulina antes de ejecutar ningún cálculo. El sistema depende de tener todos los datos configurados por el paciente (o su médico) para poder realizar cálculos correctos. Estos parámetros son:

- *Tabla de Factores de Sensibilidad a la Insulina*: El factor de sensibilidad a la insulina determina cuanta reducción en la glucemia del paciente es esperada tras la infusión de una unidad de insulina. Este parámetro se configura en forma de tabla pues puede tener diversos valores a lo largo del día. Se puede ver un ejemplo de este tipo de tabla en la Tabla 3.5.

- *Tabla de Objetivos Glucémicos*: Al igual que sucede con la sensibilidad a la insulina, el paciente o el médico pueden configurar objetivos glucémicos distintos a lo largo del día. Los objetivos glucémicos vienen determinados en una tabla por tres valores: hora de inicio del tramo, objetivo mínimo y objetivo máximo. Se puede ver un ejemplo de este tipo de tabla en la Tabla 3.6.

- *Basal máxima permitida*: Como medida de seguridad la bomba de insulina limita la cantidad de insulina que se puede dosificar en calidad de basal. Este límite se denomina basal máxima y equivale al número máximo de unidades de insulina que la bomba permitirá dosificar en una hora.

- *Bolo máximo permitido*: Al igual que ocurre con la basal máxima, existe otra medida de seguridad en la bomba de insulina que se denomina bolo máximo. Establece el número máximo de unidades de insulina que es capaz de dosificar la bomba de insulina cada vez que se comanda un bolo de insulina.

- *Duración de Insulina Activa*: Este parámetro determina el número de horas que la bomba de insulina considera que un bolo de insulina permanece activo en el cuerpo

Hora de inicio	Tasa de infusión (U/hr)
00:00	1,45
04:00	1,20
11:00	0,575
13:00	0,725
15:30	1,35
16:30	1,70
18:30	1,90
20:00	1,60

Tabla 3.7: Ejemplo de Perfil basal

del paciente tras su infusión. En el caso de este sistema, se utiliza este parámetro para poder calcular la curva de acción de la insulina adaptada al tiempo necesario para el paciente. Esta adaptación es un sencilla interpolación en el tiempo de una curva tomada como patrón (explicada más adelante en el punto 4.3.1) junto con un ajuste en los valores de cada punto de la curva para mantener constante el valor de su integral (la curva de acción calculada siempre debe ser la equivalente a una unidad de insulina).

- *Perfil basal*: El paciente tendrá configurado en forma de tabla, en función de la hora del día, la cantidad de insulina que se cree necesaria para mantener sus niveles glucémicos estables a lo largo del día. Estos valores vienen configurados en Unidades por Hora. Se puede ver un ejemplo de perfil basal en la Tabla 3.7.

- *Factor de calibración actual del MCG*: Los datos en "crudo" que envía el transmisor del MCG corresponden a lo que Medtronic denomina como *iSig*. iSig es la corriente, medida en nanoamperios, que circula por el sensor al aplicar un tensión apropiada para desencadenar la reacción buscada. Esta corriente es directamente proporcional a la concentración de glucosa en el líquido intersticial y dicha proporción se llama factor de calibración. El conocer el factor de calibración tiene un doble propósito en este sistema:

  1. Poder calcular valores de glucemia directamente con los valores del transmisor sin que la bomba de insulina sea responsable de esa tarea (normalmente lo es).

  2. Detectar cuándo ha habido un cambio en el factor de calibración (debido a una calibración del sensor) y poder re-calibrar datos pasados. De esta manera podemos evitar saltos abruptos de glucemia que puedan tener efectos indeseados en el algoritmo de control. Para esto es importante que el algoritmo de control contemple la posibilidad de que los datos pasados de glucemia puedan variar.

Otros parámetros, igualmente importantes para el correcto funcionamiento del sistema pero no relacionados con parámetros específicos del paciente, deben ser obtenidos para asegurar que la terapia puede ser llevada a cabo. Estos parámetros son:

- *Insulina restante en el reservorio*: Con esta información podemos calcular si la bomba de insulina dispone de insulina suficiente para cierto tiempo de terapia y, en caso contrario, poder mandar una alarma al paciente.
- *Voltaje de la batería de la bomba de insulina*: Al igual que sucede con la insulina restante en el reservorio, es importante avisar al paciente cuando la batería de la bomba empiece a estar descargada. Un efecto que se ha visto en este tipo de sistemas es que el consumo de energía se eleva considerablemente, debido a las periódicas conexiones entre la bomba y el sistema de control.
- *Hora del reloj de tiempo real*: La hora del reloj en tiempo real es de vital importancia para poder seleccionar, dentro de ciertas tablas de parámetros como la sensibilidad a la insulina, objetivos glucémicos o nivel basal programado, el valor que aplique para cada momento. Además, podría ser de utilidad dentro del algoritmo de control dependiendo de la algoritmia diseñada.

3.6.1.7 Procesado de información

Una vez se ha recibido toda la información necesaria para un ciclo de ejecución del lazo de control (glucemia actual, insulina activa actual medida por la bomba de insulina, etc) se pasa al proceso de dicha información. El procesado está dividido en 4 subprocesos:

1. Actualización de datos históricos
2. Actualización de vectores de insulina activa
3. Envío de actualizaciones por Bluetooth
4. Ejecución del algoritmo de control

La actualización de datos históricos hace el mantenimiento de los buffers destinados a mantener datos glucémicos pasados del paciente. Se almacenan tres vectores con datos distintos:

- Datos glucémicos: Son los datos de glucemia pasados del paciente teniendo un periodo de muestreo de 5 minutos, tal y como los proporciona el MCG. Además se codifica con un bit si el dato almacenado en cada punto del histórico es válido o no. Esto es importante para tener en cuenta datos perdidos o efectos de borde en los cálculos cuando se inicia el sistema.
- Datos glucémicos "crudos": En el caso de los sensores de Medtronic, se guardan los datos glucémicos resultantes de multiplicar directamente el iSig recibido del transmisor por el factor de calibración. Estos datos no han sido tratados por la bomba de insulina y podrían contener más ruido. La medida del ruido puede proporcionar al algoritmo una estimación de la "credibilidad" de los datos para tomar decisiones de

una u otra manera. El periodo de muestreo es, al igual que en el caso anterior, de 5 minutos y también se almacena la validez de cada punto del buffer.

- Datos "crudos" o iSig: En el caso de sensores de Medtronic, se almacenan las corrientes recibidas por el transmisor permitiendo al algoritmo hacer filtrados o medidas adicionales si fueran necesarias. Al igual que en los casos anteriores, el periodo de muestreo es de 5 minutos y también se almacena el bit de validez.

En la implementación actual, debido a la limitación en memoria del procesador, se almacenan únicamente los datos correspondientes a los últimos 80 minutos (16 valores).

La actualización de vectores de insulina activa mantiene dos buffers que almacenan la cantidad de insulina dosificada en cada periodo de muestreo (5 minutos igualmente) en calidad de insulina basal e insulina destinada a bolos.

De manera añadida, también se monitoriza la entrada de bolos de manera manual. Esto también permite controlar el temporizador `timeCounterBolusSnooze` (explicado en el apartado 3.6.1.2). Cuando la insulina activa indicada por la bomba es incrementada podemos derivar que se ha dosificado un bolo de insulina igual a la cantidad en la que ha incrementado la insulina activa.

El envío de actualizaciones por Bluetooth consiste en mandar aquellos mensajes que permiten al *smartphone* o al sistema de monitorización remoto seguir la evolución de la glucemia del paciente y el estado general del sistema, como son la insulina activa marcada por la bomba de insulina, voltaje de la batería de la bomba y la cantidad de insulina restante en el reservorio.

Una vez el sistema está actualizado con los últimos datos recibidos de todos los dispositivos, se pasa a la ejecución del algoritmo de control llamando a la función `void controlLoopDataUpdate (void)`. Dentro de esta función se puede implementar el algoritmo deseado, haciendo uso de una librería de funciones específicas que facilitan cálculos estándar si es necesario. Se puede consultar más información sobre esta librería de funciones en el apartado A.3.

Por motivos de restricciones de espacio en memoria Flash y RAM, se ha tomado la decisión de poder acceder a ciertas variables como globales directamente. La actualización de estas variables se ejecuta de manera automática durante la consulta de datos de la bomba de insulina (Apartado 3.6.1.6) y no deben ser modificadas por el algoritmo de control. La lista de variables globales disponibles puede ser consultada en el apartado A.4.

Como resultado de la ejecución del lazo de control, puede ser necesario comandar a la bomba de insulina bolos de insulina, basales temporales, cancelarlas y/o mandar notificaciones al *smartphone* para que dichas acciones y resultados del lazo queden registrados y accesibles al usuario. Para ejecutar las acciones de control sobre la bomba de insulina, se ha diseñado una librería de funciones que permite el control de una manera sencilla

y transparente para el diseñador del algoritmo. Esta librería puede ser consultada en el apartado A.5.

3.6.1.8 Envío de mensajes por Radio

En cuanto al proceso de transmisión, el firmware dispone de una cola de mensajes a ser transmitidos. De esta manera, un mensaje puede ser incluido en dicha cola en cualquier momento de la ejecución y las capas de firmware más en contacto con el hardware se encargan de la codificación, temporización del envío y gestión de la respuesta.

La cola de envio de mensajes por radio sigue un estricto orden FIFO (*First In First Out*) y está dividida en lo que se ha denominado como *slots*. Un *slot* contiene toda la información necesaria para la correcta transmisión de un mensaje y la gestión de la respuesta por parte de la bomba de insulina.

La estructura de un *slot* es la siguiente:

- `unsigned char pumpCmdQueuePending`: Determina si el mensaje almacenado en este *slot* está pendiente de ser enviado. Un valor `0` indicará que no está pendiente y un valor `1` indicará lo contrario.
- `char pumpCmdQueueMsg [ 80 ]`: Este es el buffer en el que se almacena el mensaje a ser transmitido.
- `unsigned int pumpCmdQueueLen`: Esta variable almacena el número de bytes que han sido almacenados en el buffer `pumpCmdQueueMsg [80]` y que se desea transmitir.
- `char pumpCmdQueueExpectAns`: La bomba de insulina devuelve una respuesta por cada comando que recibe. En esta variable se debe especificar el byte que devuelve la bomba de insulina como respuesta a nuestra transmisión. De esta manera el sistema es capaz de discriminar en recepción si el mensaje recibido es la respuesta al comando enviado o, por contrario, otro mensaje no relacionado y que se debe tratar por separado.
- `int pumpCmdQueueTimes`: Dependiendo del modo en el que se encuentre la bomba de insulina, puede que sea suficiente con enviar el mensaje únicamente una vez o, como en el caso del comando para "despertar" a la bomba, sea necesario el envío múltiples veces seguidas. Esta variable determina el número de veces que se desea enviar el mensaje por cada intento.
- `unsigned long pumpCmdQueueTimeout`: Esta variable establece el tiempo que se permite esperar como máximo para recibir la respuesta de la bomba de insulina desde que se finaliza la transmisión del mismo. Al igual que en el resto de *timers* del sistema, la base de tiempos es de 200 ms. Si pasado este tiempo no se ha rebido la

respuesta esperada, se procederá a reintentar la transmisión o a dar la transmisión por fallida si ya se han agotado el número de retransmisiones permitidas.

- `unsigned char pumpCmdQueueRetries`: Con esta variable se determina el número máximo de reintentos que permitimos al sistema antes de dar la transmisión por fallida.

El manejo de una cola de estas características puede no ser trivial cuando, además, el sistema está gestionando el envío y modifica parámetros de la cola en tiempo real. Es por este motivo que se ha desarrollado una función que permite añadir mensajes a la cola de manera sencilla y segura: `char addMsgToQueue (char *msg, unsigned int length, int times, char expectedAnswer, unsigned long timeout, unsigned char retries)`.

La función `addMsgToQueue` añade el mensaje a la cola de transmisión y devuelve el número de `slot` en el que se almacenó el mensaje. En caso de existir algún error como la falta de slots para transmisión o algún parámetro incorrecto, el valor devuelto por la función es `-1`.

Una vez los mensajes se encuentran en la cola de transmisión, el proceso de transmisión de mensajes sigue el flujo mostrado en la Figura 3.13.

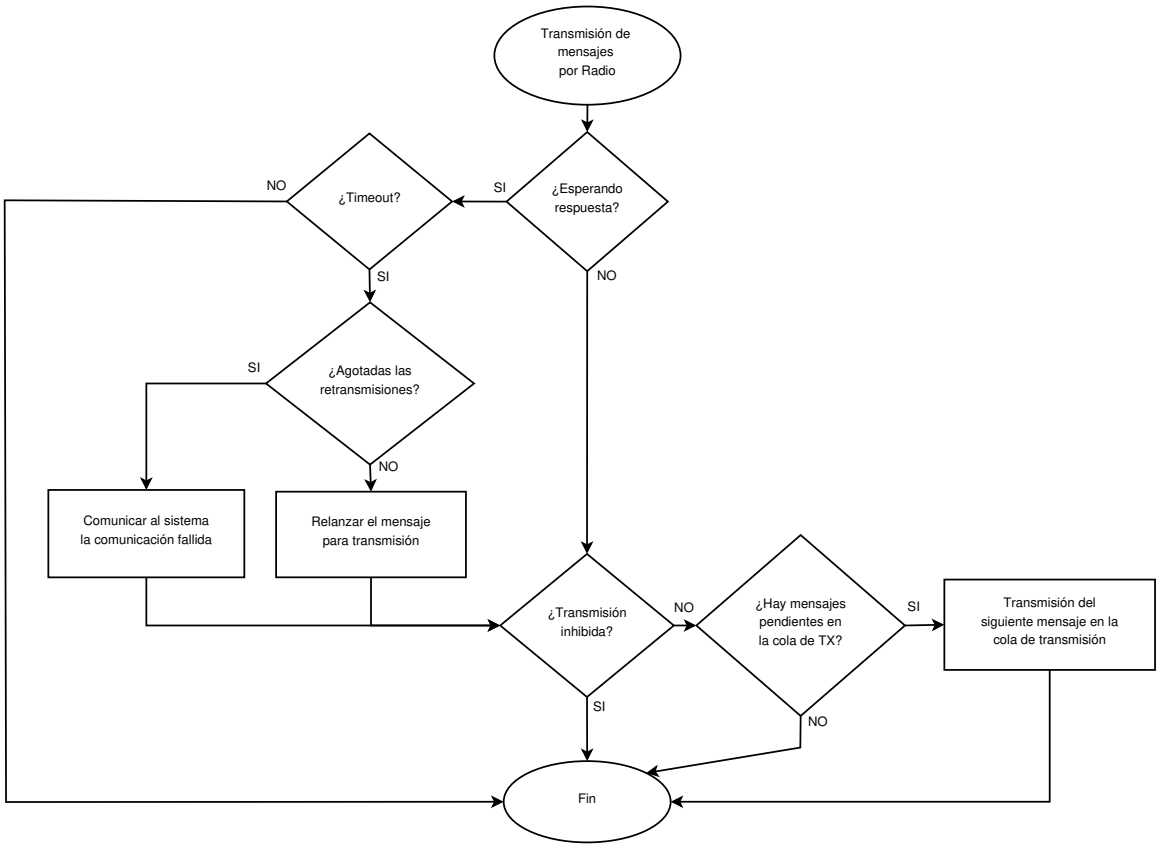


Figura 3.13: Diagrama de flujo de la gestión de la transmisión por radio

El primer paso es comprobar si el sistema se encuentra esperando la respuesta a un mensaje previamente enviado. Para ello se espera el tiempo especificado como `timeout`

y, mientras tanto, no se procede a la transmisión de más mensajes. Si este tiempo vence sin haber recibido la respuesta esperada por parte de la bomba de insulina, se consulta si ha agotado el número de reintentos especificados como `times`. En caso de no haberlos agotado se solicitará el reenvío por radio y, en caso contrario, se notifica la transmisión fallida al sistema y se elemina el mensaje de la cola.

El proceso de recepción de mensajes por radio comprobará si alguno de los mensajes recibidos corresponde con la respuesta esperada de la bomba de insulina y, en caso positivo, se notifica al sistema para que tome la acción pertinente y se elimina el mensaje de la cola de transmisión.

Posteriormente se comprueba si el sistema se encuentra en el periodo de tiempo en el que se inhibe la transmisión por radio (ver apartado 3.6.1.2) y, en caso de poder transmitir, se comprueba la existencia de mensajes a ser transmitidos. En caso de tener mensajes pendientes de transmisión, se procede a la transmisión del siguiente en la cola.

3.6.2 CC2540 - Comunicaciones por Bluetooth Low Energy

En la implementación actual del controlador este procesador se limita a presentar un interfaz sencillo para hacer de puente entre una conexión Bluetooth Low Energy y un interfaz serie.

Texas Instruments, en su entorno de desarrollo software para este procesador [62], ofrece una serie de proyectos de ejemplo y, entre ellos, se encuentra uno que realiza gran parte de las funcionalidades necesarias para este sistema. En este caso, el interfaz the Bluetooth hace uso del proyecto de ejemplo `SimpleBLEPeripheral` como base y se realizan los cambios necesarios para:

- Modificar el nombre identificador del dispositivo que presenta el dispositivo como periférico Bluetooth para tener uno más reconocible.
- Modificar los servicios y características Bluetooth para que se disponga de un servicio con una única característica capaz de:
  - Ser leible (Propiedad *Read*) y que permita al smartphone leer su contenido.
  - Ser modificable (Propiedad *Write without response*) y que permita enviar información desde el smartphone
  - Mandar notificaciones (Propiedad *Notify*) al smartphone cuando suceda algún cambio.
- Añadir la inicialización y funciones de tratamiento del interfaz serie.

Para la implementación se utilizaron los siguientes UUID:

- Servicio: `0xFFE0`

- Característica: `0xFFE1`

Como ya se ha comentado anteriormente, este dispositivo tiene el rol de periférico de manera que sea el smartphone quien se conecte con él cuando sea necesario. El emparejamiento se realiza mediante el sistema más sencillo posible en *Bluetooth Low Energy 4.0*: *BLE JustWorks*.

Como todo el tratamiento de mensajes y *timeouts* se realiza en el SoC CCC1110, en el firmware para el interfaz de Bluetooth no se hace ningún tratamiento de la información y simplemente se actua como puente entre las dos tecnologías.

Para comunicarse con el sistema, el smartphone únicamente deberá escribir los datos que desee mandar en la característica `0xFFE1` y subscribirse a las notificaciones para que, cuando el sistema envíe datos al smartphone, este pueda ser notificado y pueda leer la característica.

3.6.3 CC2510 - Comunicaciones por Radio 2.4GHz

Este SoC ha sido incluido en la plataforma *hardware* pensando en la posibilidad futura de conectar otros sistemas MCG de otros fabricantes (como Dexcom), pero actualmente se encuentra desactivado. El software diseñado únicamente entra en estado de bajo consumo nada más iniciarse la ejecución.

En caso de necesitar comunicaciones con otros dispositivos utilizando este SoC, el procesador encargado de las comunicaciones Bluetooth Low Energy debe actuar como concentrador o puente pues son los dos procesadores conectados mediante un interfaz serie. Si se deseara utilizar datos de un MCG de Dexcom, los datos deberían ser recibidos por el SoC CC2510 y transmitidos via serie al procesador de Bluetooth. Desde el procesador de Bluetooth se pueden enviar los datos al smartphone y/o implementar mensajes nuevos para transferir los datos al procesador CC1110 que actualmente está ejecutando el algoritmo.

Mediante las lineas discretas se pueden implementar protocolos para poner el procesador en bajo consumo y despertarlo o generar interrupciones ante eventos como la recepción de datos nuevos.

Son muchas las posibilidades que ofrece este procesador en la plataforma.

3.7 Seguridad en las comunicaciones

Aunque el aspecto de la seguridad en las comunicaciones se encuentra fuera de los objetivos principales de esta tesis doctoral, es importante remarcar que la seguridad del paciente siempre es lo primero a tener en cuenta y, como no puede ser de otra manera, las comunicaciones deben ser lo más seguras posibles para evitar incidentes indeseados.

Es por ello que se ha hecho un gran esfuerzo por analizar el problema y llegar a la mejor solución posible.

Una de las principales preocupaciones con respecto a los sistemas DIY que utilizan bombas de insulina antiguas de Medtronic [8][9], a parte de la seguridad a nivel glucémico, es la falta de sistemas de seguridad en las comunicaciones entre el controlador y la bomba de insulina. Las comunicaciones no están encriptadas de ninguna manera y, además de hacer que la bomba de insulina sea fácilmente accesible por el controlador, también la hace vulnerable a posibles atacantes que podrían intentar controlar de manera remota la infusión de insulina y, por consiguiente, ser un peligro potencial para la salud del paciente.

El controlador se comunica con la bomba de insulina cada 5 minutos y, al hacerlo, el identificador de la bomba de insulina está siendo anunciado múltiples veces con cada intercambio de datos. Cada mensaje contiene el identificador de la bomba para que pueda ser reconocido por el receptor y/o cualquier otro sistema que pueda estar esperando esos mensajes. Los periodos de tiempo en los que existen comunicaciones con la bomba de insulina son críticos, ya que el identificador de la bomba está siendo expuesto y el atacante puede estar escuchando a la espera de recibirlo para, posteriormente, utilizarlo de manera maliciosa y mandar comandos que podrían resultar peligrosos para la salud del paciente.

Medtronic ha creado un nuevo protocolo de comunicaciones para dar respuesta a este problema encriptando el contenido de los mensajes pero, como el sistema aquí propuesto hace uso de una bomba de insulina con el protocolo anterior, se ha introducido un sistema de detección de intrusiones y un perturbador como contramedida.

El controlador está siempre escuchando, buscando mensajes de/para la bomba de insulina. Si se recibe un mensaje y no es respuesta a ningún comando enviado por el controlador, se manda una alarma de intrusión al smartphone para que se pueda dar la alarma a su vez al paciente. Al mismo tiempo, se comienza el proceso de perturbación: el controlador manda de manera continua el comando de "suspensión de infusión" a la bomba de insulina en un intento por cancelar cualquier comando peligroso mandado por el atacante, dando tiempo de esta manera al paciente para que tome acción y se proteja a sí mismo de la situación. Esta transmisión de manera continua corromperá cualquier otro mensaje mandado por el atacante, bloqueando de esta manera el canal de comunicaciones por algunos minutos.

Esta protección puede ser deshabilitada para permitir la descarga de datos desde la bomba cuando el usuario lo necesite (por ejemplo durante una cita médica).

Por otro lado, este controlador también dispone de comunicaciones por Bluetooth y también podrían llegar a ser comprometidas. Como no es necesario comandar la bomba de insulina directamente por Bluetooth, pues todos los cálculos se realizan internamente en el controlador, no se ha implementado mensaje alguno que lo permita (ver apartado 3.6.1.3) manteniendo así un interfaz seguro.

3.8 Integración con Nightscout

La integración con el sistema de monitorización open-source Nightscout [7] permite al paciente, y a quien necesite acceder a los datos del control, el acceso fácil a los datos resultantes de los cálculos y a las entradas al algoritmo (parámetros del paciente, datos del mcg, infusión de insulina, etc). Además, permite recibir alarmas del sistema cuando es necesario.

La forma en la que se ha realizado la integración de ambas plataformas ha sido mediante la modificación del software open-source xDrip [58]. Este software permite la recepción de datos de sistemas MCG con interfaz Bluetooth Low Energy y, además de mostrar la información de manera local al smartphone, dispone de una API que permite subir la información a Nightscout. xDrip modulariza la conectividad con cada dispositivo o servicio mediante clases que proporcionan un interfaz común. Este diseño facilita el intercambio que dispositivos o la inclusión de nuevos dispositivos dentro del software. En esta tesis doctoral se ha modificado xDrip añadiendo la plataforma *wearable* como una nueva clase que gestiona la conexión Bluetooth Low Energy e interpreta los mensajes para convertirlos en datos o eventos a ser almacenados. También se aprovecha la API de Nightscout para subir la información a la nube. Los mensajes interpretados por esta nueva clase son los anteriormente enumerados en la Tabla 3.2. Una vez los datos han sido convertidos y subidos a Nightscout, se puede hacer uso de todos los servicios que ofrece la plataforma como monitorización en tiempo real o generación de informes.

Nightscout, como plataforma, es un sistema perfecto para ajustar parámetros y además permite al paciente registrar toda su información relativa a su terapia: ingestas, bolos de insulina, cambios de set de infusión, cambios de reservorio de insulina, ejercicio físico, basales temporales y cualquier otra información que pueda ser de utilidad. Endocrinólogos se pueden beneficiar también de esta herramienta para hacer ajustes, ya que disponen de información en tiempo real y de manera remota.

La Figura 3.14 muestra la típica web de Nightscout que un paciente podría estar utilizando. El número grande verde de la esquina superior derecha (elemento número 1) muestra la glucemia actual y la flecha indica la tendencia utilizando los últimos datos recibidos del MCG. La línea de puntos (elemento número 7) muestra los datos de glucemia del paciente en tiempo real. Cada punto equivale a una medida del MCG. En el caso del ejemplo, el rango objetivo (cuando los puntos son de color verde) está ajustado para el rango 90-140 mg/dL. La línea de abajo (elemento número 8) muestra los datos de glucemia de los dos últimos días consecutivos, mientras que la gráfica del centro puede ajustarse para mostrar las últimas 3, 6, 12 o 24 horas de datos glucémicos utilizando un selector (elemento número 6). El gráfico azul sobre la representación de las glucemias (elemento número 9) muestra cómo se ha ido modificando la infusión de insulina (insulina basal). Otros parámetros que pueden ser monitorizados son:

- Hora actual (elemento número 3)
- IOB: *Insulin-On-Board* o Insulina Activa
- Tendencia numérica (elemento número 2)
- COB: *Carbohydrates-On-Board* o Hidratos de Carbono activos (siempre que el usuario introduzca información sobre sus ingestas)
- CAGE: *Cannula Age* o Edad del set de infusión. Se utiliza como recordatorio para que el paciente sepa cuando debe cambiar su set de infusión.
- SAGE: *Sensor Age* o Edad del sensor. Recordatorio del tiempo que lleva el sensor en el cuerpo del paciente
- IAGE: *Insulin Age* o Edad de la insulina. Muestra el tiempo que la insulina lleva introducida en el reservorio.
- BASAL: Muestra el ratio de infusión basal actual o si se ha configurado una basal temporal.
- PUMP: Muestra el estado actual de la bomba de insulina (insulina restante en el reservorio y voltaje de batería)
- Tiempo desde la última medida recibida del MCG y desde la última acción tomada por el controlador (elemento número 4).
- Nivel de batería del smartphone que está subiendo los datos del paciente a Internet (elemento número 5)

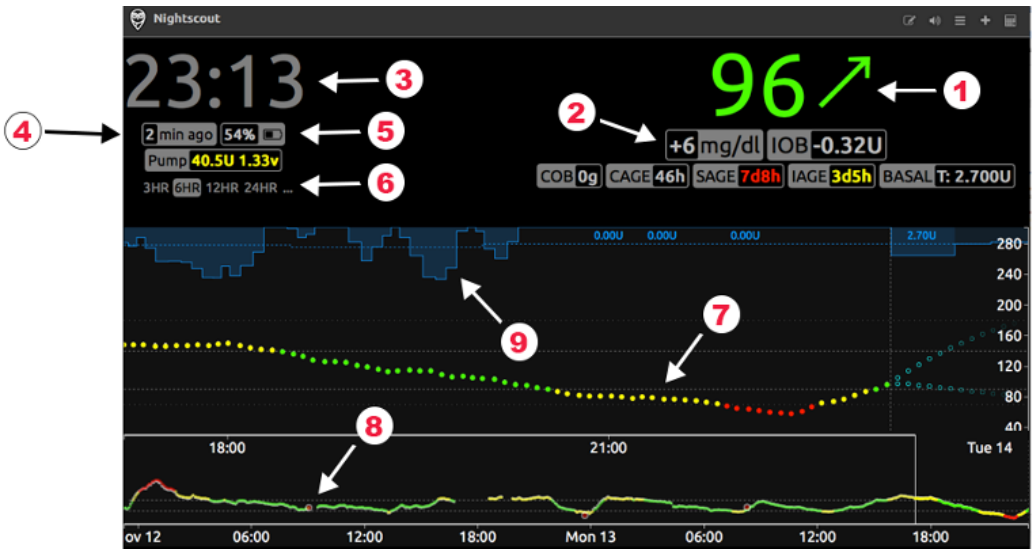


Figura 3.14: Nightscout

3.9 Conclusiones

Este capítulo se puede dividir en dos partes diferenciadas: una primera parte en la que se describe el concepto general de lo que es un sistema de control en lazo cerrado para diabetes tipo 1 y los requisitos asociados para que el controlador pueda denominarse *wearable*, y una segunda parte en la que se realiza el desarrollo de una plataforma HW/FW capaz de cumplir con dichos requisitos.

En la primera parte, tras explicar el concepto de control en lazo cerrado para diabetes, se realiza una descripción de cada uno de los elementos involucrados en el sistema y los problemas fundamentales asociados según el estado actual de la tecnología. Estos problemas radican básicamente en los retardos existentes tanto en el proceso de medida de la glucemia del paciente como en la acción de la insulina en el cuerpo una vez esta ha sido dosificada. Estos problemas tienen como consecuencia que sea necesario anunciar ciertos eventos al controlador para que este pueda actuar antes de que los efectos sean evidentes en los datos.

Posteriormente se describen los requisitos establecidos para considerar que una plataforma de control es *wearable* y, en función de los mismos, se desarrolla la plataforma hardware necesaria. Estos requisitos se pueden resumir en:

- Duración mínima de batería de 24 horas o superior.
- Menor tamaño posible.
- Tolerante a fallos de comunicaciones
- Debe permitir conectar una interfaz con el paciente para anuncio de eventos.

La segunda parte corresponde con la descripción de la plataforma implementada. Esta plataforma tiene dos partes: una parte hardware en la que se hace una selección de procesadores, tecnologías y batería y una parte firmware, que da soporte al funcionamiento de la plataforma y que prepara el entorno para que sea sencillo el diseño de nuevos algoritmos de control.

Dentro de la plataforma firmware se desarrollan los siguientes conceptos:

- Comunicaciones via Bluetooth y monitorización remota
- Descripción del bucle general de operación del controlador
- Gestión de energía y tiempos
- Comandos implementados para el interfaz Bluetooth
- Descripción de comunicaciones con dispositivos Medtronic
- Algoritmo de recuperación de mensajes con errores (SmartRecovery)

- Procesado de datos relacionados con el paciente y su terapia
- Implementación de funciones de cálculo generales y reutilizables entre algoritmos
- Gestión de cola de envío de mensajes via radio
- Seguridad en las comunicaciones

En resumen, este capítulo presenta las siguientes contribuciones a la tesis:

- Desarrollo de una plataforma *wearable* de bajo consumo
- Conectividad con los elementos necesarios del APS
- Firmware de bajo consumo en todos los elementos
- Integración con sistemas de gestión ya existentes
- Posibilidad de prueba de algoritmos de control en lazo cerrado

Como conclusión podemos añadir que se ha demostrado la viabilidad de desarrollar una plataforma *wearable* que permite a un paciente llevar un controlador las 24 horas del día con un mínimo impacto en su día a día. Además, esta misma plataforma es una excelente opción para que la comunidad científica pueda recopilar datos de pacientes reales y, además, desarrollar nuevos algoritmos de manera sencilla. Es importante también remarcar que esta plataforma no necesita de un *smartphone* para poder operar y que todos los cálculos del controlador se pueden realizar en la plataforma. Esto aumenta la seguridad del sistema en cuanto a posibles ataques de *hackers*, no existe relación entre la batería disponible del *smartphone* y la disponibilidad del controlador a lo largo del día y permite llevar un controlador en situaciones en las que el uso de *smartphones* pudiera estar prohibido (por ejemplo en colegios).

Capítulo 4

Algoritmia de control

Medicina sólo hay una, y es efectiva cuando tiene una evidencia científica detrás que la respalde.

J.M. Mulet

4.1 Introducción

Una vez se dispone de un controlador *wearable* capaz de mantener las comunicaciones y controlar una bomba de insulina y recibir información de un MCG, el siguiente paso natural es implementar un sistema de control en lazo cerrado. La plataforma descrita en el capítulo 3 dispone de las funcionalidades y capacidades necesarias para la implementación de un algoritmo de control que permita mantener la glucemia del paciente en un rango seguro.

El *firmware* base del controlador es el encargado de:

- Gestionar las comunicaciones con el *smartphone*.
- Gestionar el control de potencia y los modos de bajo consumo.
- Gestionar el interfaz de radio con la bomba de insulina y el MCG
- Decodificación y tratamiento de la información recibida por radio
- Mantenimiento de los datos históricos del paciente
- Proporcionar un conjunto de funciones que comanden acciones en la bomba de insulina
- Actualización periódica de parámetros de terapia configurados por el paciente en la bomba de insulina
- Proporcionar un conjunto de funciones que permitan hacer cálculos complejos como insulina activa o consultar tablas de configuración

Con este conjunto de funcionalidades resueltas, el diseñador de la algoritmia únicamente necesita hacer uso de ellas para implementar su algoritmo y no tener que preocuparse por el "bajo nivel" del sistema. Cada 5 minutos, los datos recibidos de la bomba de insulina y el MCG son actualizados en las estructuras internas del *firmware* y la función que realiza los cálculos del lazo de control es llamada.

Este capítulo se ha dividido en las siguientes secciones:

1. Descripción de parámetros de entrada al sistema y cómo hacer uso de ellos.
2. Algoritmo base de toma de decisiones. En este apartado se diseña un algoritmo de control en lazo cerrado básico en el que se hace uso de los parámetros programados por el paciente y se toman decisiones para obtener el objetivo glucémico deseado.
3. Seguridad médica del paciente. Es de vital importancia explicar las medidas de seguridad implementadas en el algoritmo base para justificar que este sistema podría ser utilizado en la vida real.

4.2 Descripción de parámetros de entrada al sistema

Como ya se ha explicado con anterioridad, existe un conjunto de variables y funciones que permiten al diseñador del algoritmo obtener información actualizada, de manera transparente, con cada ejecución de la función de cálculo del lazo de control (`void controlLoopDataUpdate(void)`).

La función es llamada cada 5 minutos y será ella misma la responsable de almacenar toda aquella información propia que sea necesaria entre llamadas. No será necesario almacenar datos de dosificación de insulina ni las glucemias pasadas del paciente pues estos datos son almacenados por el sistema y pueden ser consultados.

Existe un conjunto de variables globales (anteriormente detallado en el apartado 3.6.1.7) que puede ser consultado por el algoritmo pero que no debe ser alterado durante su ejecución. Estas variables son:

- Hora del reloj en tiempo real: permite conocer la hora en la que se está trabajando y consultar tablas de parámetros que se obtienen configuradas por tramos horarios (sensibilidad a la insulina, perfil basal y objetivos glucémicos). Esta información puede ser importante para hacer análisis más en profundidad de otros parámetros biológicos como el ciclo circadiano.
- Tabla de sensibilidad a la insulina: conjuntamente con la hora en tiempo real, permite conocer el efecto esperado de una unidad de insulina sobre la glucemia del paciente.
- Perfil basal: Este perfil es una tabla configurada en la bomba de insulina que determina las necesidades de insulina por hora del cuerpo del paciente para mantener su glucemia estable. Normalmente este parámetro es configurado por un endocrinólogo.

- Tabla de objetivos glucémicos: Al igual que sucede con la tabla de sensibilidad a la insulina y el perfil basal, esta tabla determina, según la hora del día, la glucemia mínima y máxima que se establecen como objetivos para dicha franja horaria. Estos datos son utilizados normalmente para cálculo de estadísticas, como el tiempo dentro de rango, pero el algoritmo de control puede hacer uso de estos datos para ajustar su paramétrica si lo considera necesario.

- Historia de datos glucémicos del MCG: Están actualizados y disponibles los datos recibidos del sensor hasta una antigüedad de 80 minutos. Cada dato de la historia va acompañado de un bit que indica si el dato es válido o no.

- Historia de dosificación de insulina: Se dispone de un array historial de la insulina dosificada como basal y otro de la dosificada como bolo. El periodo de muestreo es de 5 minutos y se almacenan las últimas 8 horas de información.

- Curva de respuesta a la insulina: El sistema genera un array con la cantidad de insulina, relativa a una única unidad de insulina, que se asume está causando efecto con un periodo de muestreo de 5 minutos. Esta curva siempre debe tener una integral igual a 1 (por ser la relativa a una unidad de insulina) y se adapta en función de la duración a la insulina activa configurada en la bomba de insulina. Al igual que sucede con el historial de infusión de insulina, existe espacio para almacenar la respuesta hasta de 8 horas después de la infusión (que en insulinas rápidas debería ser casi nula llegado ese punto).

Aunque estos datos de partida deberían ser suficientes para cualquier algoritmo de control, el sistema también proporciona otra serie de funciones que, debido a su carácter genérico, se ha creído podían ser de utilidad y aumenta la legibilidad del código del algoritmo. Estas funciones (descritas en detalle en el apartado 3.6.1.7) permiten realizar las siguientes tareas:

- Obtener la insulina activa:
  - Correspondiente a la insulina dosificada total
  - Correspondiente a la insulina dosificada en calidad de basal
  - Correspondiente a la insulina dosificada como bolos
  - Correspondiente a la insulina programada para ser dosificada en el futuro cercano como basal.
  - Correspondiente a toda la insulina dosificada y la programada para su dosificación en el futuro cercano como basal.

- Obtener la tasa basal programada para:
  - El momento de la consulta.

  - Una hora del día en concreto.

- Obtener la sensibilidad a la insulina programada para:
  - El momento de la consulta.
  - Una hora del día en concreto.
- Obtener el objetivo glucémico programado para:
  - El momento de la consulta.
  - Una hora del día en concreto.
- Obtener el tiempo que transcurre desde la infusión de una unidad de insulina hasta que se alcanza el pico de acción.
- Obtener la hora del día actual

Si fuera necesario configurar más parámetros adicionales, se deberían configurar utilizando el *smartphone* y su conexión *Bluetooth Low Energy*. Para ello, sería necesario definir nuevas variables globales y mensajes para su modificación y consulta en la función `void receiveBLEMessage (char *message, unsigned int messageLen)` (descrito en el apartado 3.6.1.3).

4.3 Conceptos básicos aplicables al algoritmo

Antes de explicar el flujo lógico de cálculo del algoritmo, es importante hacer hincapié en los distintos conceptos que serán utilizados de ahora en adelante en esta tesis. Estos conceptos serán de vital importancia para la correcta compresión de las diversas variables que se tendrán en cuenta en la toma de decisiones y en el funcionamiento general del sistema.

4.3.1 Curva de acción de la insulina

El algoritmo controla la dosificación de insulina en el cuerpo del paciente, y dicha insulina tendrá un efecto u otro dependiendo del tipo de insulina utilizado y su respuesta concreta en el cuerpo del paciente.

En esta tesis doctoral se supondrá el uso de insulina tipo Humalog® [4][63] por ser probablemente una de las insulina rápidas más extendidas.

La Figura 4.1 muestra la curva típica de absorción de una unidad de insulina en intervalos de 5 minutos. La suma de todos los puntos mostrados en la figura da como resultado la unidad. Se puede ver cómo, desde el momento de la dosificación de la insulina, el efecto comienza a ascender llegando a su máximo de acción aproximadamente a la hora y posteriormente desciende en lo que comúnmente se denomina como la “cola de acción”.

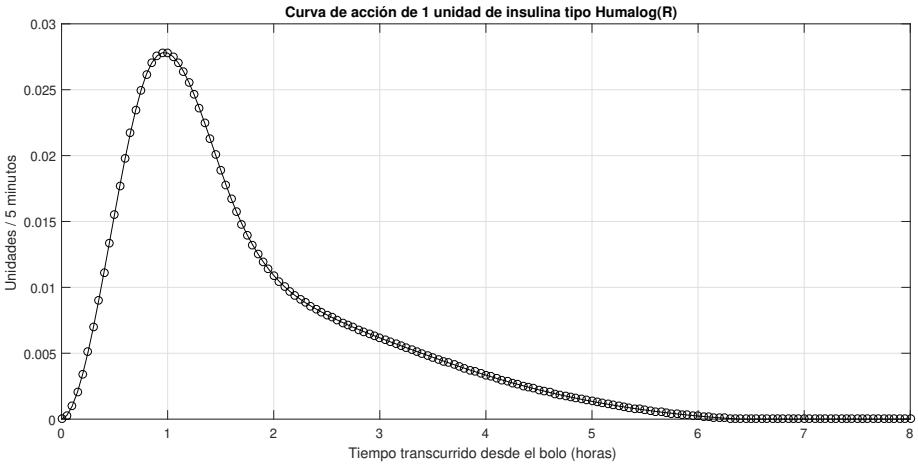


Figura 4.1: Ejemplo de absorcion 1 unidad de insulina Humalog

Esta curva debe ser adaptada a cada paciente mediante el parámetro "Duración de la insulina". En sistemas comerciales en lazo abierto es común encontrar que esta "curva" se implementa como una recta en la que el efecto de la insulina decrece de manera lineal hasta que se anula transcurrido el tiempo definido como duración. No existe un consenso en cuanto a cómo definir este tiempo utilizando curvas reales de respuestas a la insulina, pero en esta tesis se toma como duración el tiempo que debe transcurrir hasta que se alcanza el 90 % del efecto esperado. En el caso de la curva utilizada en el ejemplo, la duración aproximada es de 3,5 horas (ver Figura 4.2).

En función de la duración de la insulina configurada para el paciente, se extiende o contrae la curva de acción mediante interpolación y se re-escalan proporcionalmente los valores de los puntos resultantes para que el sumatorio de todos sus puntos continúe siendo la unidad. Este proceso tiene como resultado la curva de acción de la insulina que será utilizada en el algoritmo a partir de ahora.

4.3.2 Insulina Activa o Insulina a Bordo

Una vez conocida la curva de acción de la insulina, es de vital importancia que el algoritmo/sistema conozca la insulina que, a pesar de haber sido dosificada en el pasado, todavía está causando efecto en ese preciso momento y en el futuro. Esta insulina es a la que se denomina como "insulina activa" o "insulina a bordo" (IOB - *Insulin-On-Board*).

Ya se ha explicado en apartados anteriores que el sistema guarda un histórico de todas las dosificaciones de insulina que han sucedido en el pasado cercano (8 horas). Esta información se almacena como un array de dosificaciones puntuales cada 5 minutos. Para obtener la insulina activa o, dicho de otra manera, la cantidad de insulina que todavía tiene que causar efecto en el cuerpo del paciente, se debe tener en cuenta la curva de acción de la insulina. Al haber guardado la insulina dosificada como eventos puntuales sin tener en cuenta la curva de acción, es posible el reconfigurar la curva de acción en

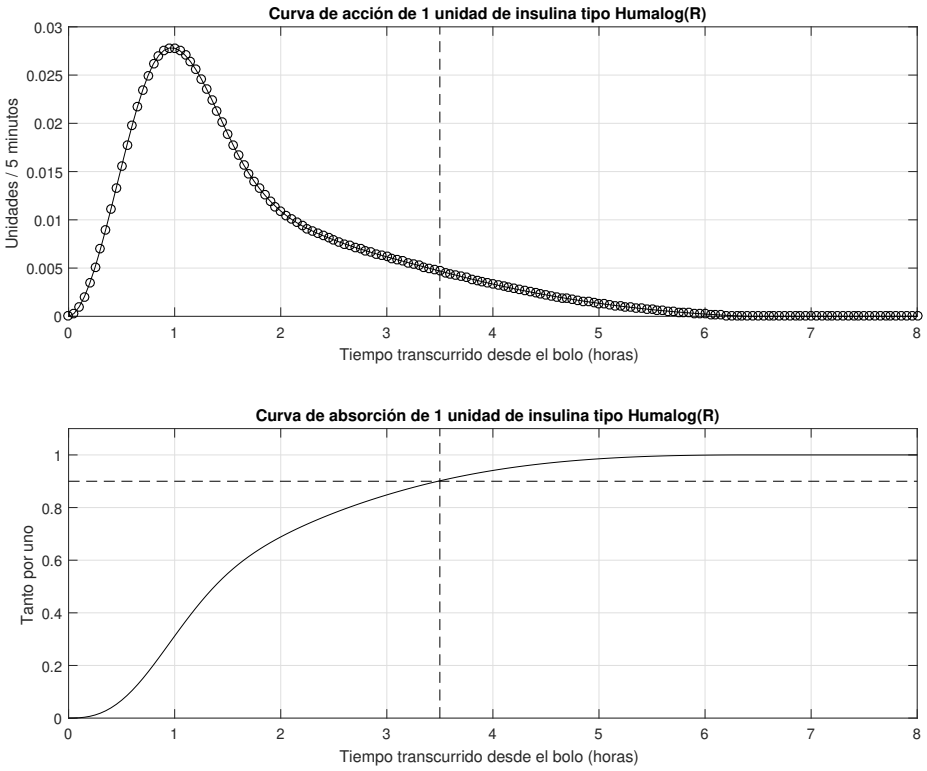


Figura 4.2: Ejemplo de absorcion de insulina Humalog - Duración de la insulina

cualquier momento (de manera manual o automática si fuera necesario) y tener los valores actualizados en cualquier momento.

Para realizar el cálculo de IOB, basta con calcular la convolución del vector de historia de dosificaciones de insulina con la curva de acción de la misma y hacer el sumatorio de todo efecto posterior al momento presente (ver Ecuación 4.1). En las ecuaciones que describen este sistema se toma como referencia de tiempos (*t=0*) el momento presente, quedando representado el pasado por tiempos negativos y el futuro por tiempos positivos.

Ecuación 4.1: Cálculo de insulina activa (IOB)

$$IOB = \int_{t=0}^{\infty} historiaDosificacion(t) * curvaAccionInsulina(t) \tag{4.1}$$

En la Figura 4.3 se muestra un ejemplo en el que se dosifican dos bolos de insulina (2.0 y 3.0 unidades respectivamente) con aproximadamente una hora de distancia temporal entre ellos. Hora y media después de la dosificación del último bolo se calcula la insulina activa. El efecto esperado de estos dos bolos de insulina responde a la convolución de la curva de acción de la insulina y el vector que almacena los bolos. Todo efecto que tenga lugar en el futuro (tiempo relativo >= 0) debe ser considerado como insulina activa o insulina a bordo. En este ejemplo, si se integra el efecto en el futuro, obtenemos una insulina activa igual a 1.78 unidades. Esta información debe ser tenida en cuenta en el

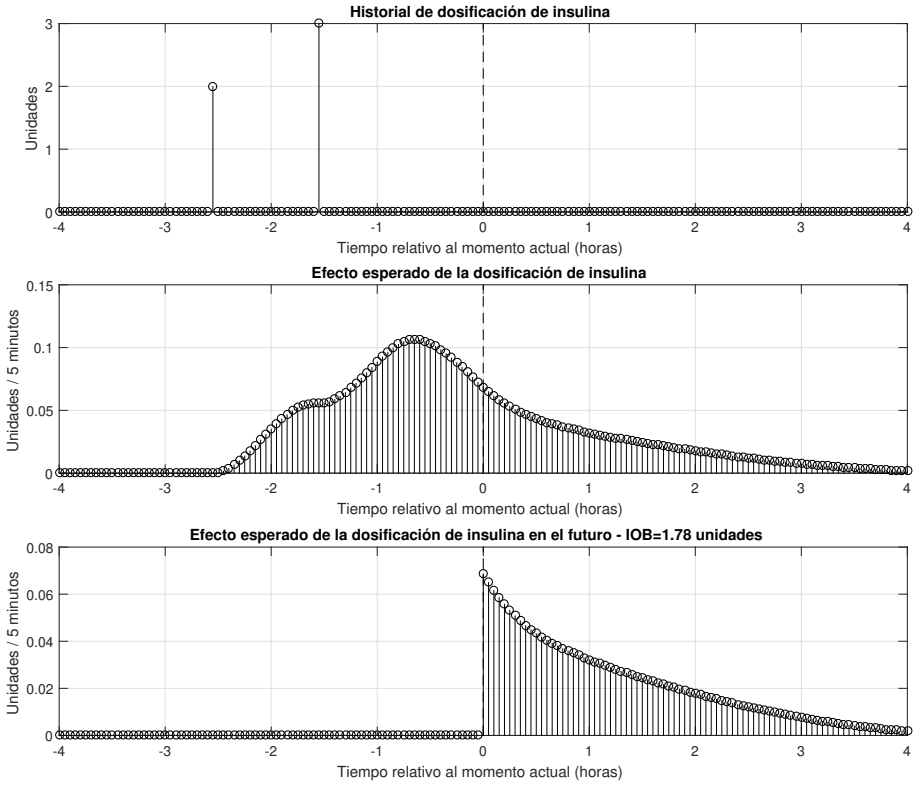


Figura 4.3: Ejemplo de cálculo de insulina activa

algoritmo de control.

4.3.3 Gestión de ingestas

Es conocido que toda ingesta, a medida que va siendo absorbida en el cuerpo, repercute incrementando la glucemia. Es común visualizar el efecto de la ingesta, junto con el de la insulina dosificada para ella, como dos fuerzas contrapuestas en el que el objetivo final del paciente es su perfecta cancelación. El paciente debe calcular la insulina necesaria para "cubrir" dicha ingesta mediante el uso de los denominados "Ratios HC/U" o Ratios de hidratos de carbono por unidad de insulina y, además, ajustar el tiempo existente entre la dosificación de la insulina y la ingesta para que el pico de acción de la insulina coincida con el de absorción de los hidratos de carbono.

Existen diversas maneras de realizar la gestión de las ingestas del paciente: se puede intentar modelar la absorción de los hidratos de carbono o se puede considerar que el paciente es capaz de calcular los bolos de insulina necesarios para su correcta compensación.

Existen varios problemas fundamentales con el modelado de la absorción de los hidratos de carbono:

- La estimación de la cantidad de hidratos de carbono ingerida suele tener un alto

porcentaje de error. Es el paciente el que debe estimar esta cantidad (en múltiples ocasiones "a ojo") y el grado de error suele ser elevado [64].

- La velocidad de absorción de los alimentos, también denominado como Índice Glucémico, varía de producto a producto y también, en algunos casos, dependiendo del grado de maduración (por ejemplo en la fruta) o del grado de cocción (por ejemplo en la pasta) (ver ejemplos en la Figura 4.4) [65].

- Las grasas añadidas a los alimentos (por ejemplo aceites o salsas) tienen un efecto retardador de la absorción y es dificil calcular tanto el contenido en grasas como el efecto que las mismas tienen en cuanto a tiempo de retardo[66].

- Los distintos alimentos de los que se compone la ingesta, y el orden en el que son ingeridos, hacen que la curva de absorción sea una composición de las curvas individuales y, además, pueden afectarse entre si.

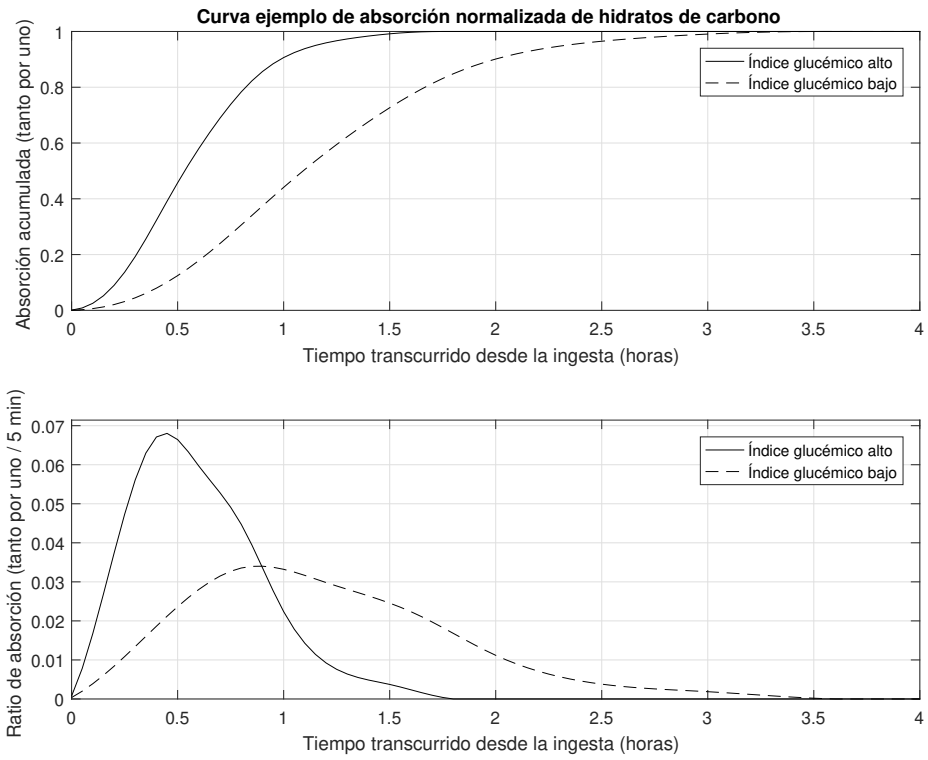


Figura 4.4: Ejemplo de absorción de hidratos con distinto índice glucémico

Además, el tiempo de adelanto con el que se dosifique la insulina tiene un gran impacto sobre el resultado glucémico final. Las curvas de acción de la insulina (ver ejemplo de la insulina Humalog en la Figura 4.1) no coinciden con las curvas de absorción de los hidratos de carbono (salvo en el caso de que la ingesta se ajuste para ello, no siendo lo normal en la vida del paciente) y el paciente debe ajustar el tiempo existente entre la dosificación de insulina y la ingesta para intentar obtener la mejor respuesta glucémica posible. En

caso de que el paciente tenga su glucemia dentro de su rango deseado, se deberá ajustar el tiempo de adelanto de la dosificación de la insulina de tal manera que los picos máximos de acción de la insulina y la absorción de los hidratos coincida en el tiempo. Por contrario, si la glucemia del paciente se encuentra por debajo de dicho rango, deberá reducir el tiempo entre la ingesta y la dosificación de la insulina. De esta manera, los hidratos comenzarán a actuar antes que la insulina, incrementando la glucemia del paciente y eliminando el riesgo de hipoglucemia, y después la insulina hará su efecto. En caso de encontrase por encima del rango deseable, quizá sería interesante retrasar la ingesta para dejar que la insulina haga efecto primero, normalizando un poco la glucemia previamente, y después realizar la ingesta.

A modo ilustrativo, se ha realizado la simulación de una misma ingesta de hidratos de carbono en la que el bolo de insulina se ha ajustado de manera perfecta y se ha variado únicamente el tiempo de adelanto de la insulina con respecto a la ingesta. No se ha simulado la posible variación de sensibilidad a la insulina en función de la glucemia del paciente, situación que ocurre en la vida real [67], pero aún así se puede observar el gran impacto que tiene la modificación de únicamente uno de los parámetros involucrados en el proceso de ingesta.

En las Figuras 4.5, 4.6, 4.7 y 4.8 se muestra el resultado de simular el efecto de un bolo de insulina Humalog de 3 unidades, utilizando una sensibilidad a la insulina de 36 mg/dl/U, junto con una ingesta de hidratos de carbono de absorción rápida de cantidad relativa equivalente a la insulina dosificada de una manera totalmente exacta. Si prestamos atención a la evolución glucémica simulada en todas las figuras, vemos que se parte de una glucemia de 100 mg/dl en todos los casos y, debido a que los efectos de los hidratos de carbono y la insulina se han ajustado de manera exacta, la glucemia final siempre es 100 mg/dl. En estas figuras también se puede ver el efecto causado por la insulina y por la ingesta de manera independiente y el efecto conjunto de ambas. El momento de referencia t=0 corresponde en todos los casos con la dosificación del bolo de insulina y, dado que siempre se dosifica la misma cantidad de insulina, todas las figuras representan en color azul la misma curva. Esta curva está relacionada directamente con la respuesta en el tiempo de la insulina utilizada (ver Figura 4.1) y se representa con valores negativos haciendo referencia a que el efecto de la insulina provoca una reducción de la glucemia del paciente. El efecto de la ingesta de hidratos de carbono, representado en color rojo, se desplaza en el tiempo dependiendo del adelanto de insulina simulado.

En la Figura 4.5 se puede apreciar que, debido a que el efecto de la ingesta se produce con más antelación que el del pico de la insulina dosificada, obtenemos una subida pronunciada de la glucemia hacia la media hora después de la ingesta y después un lento descenso que, tras varias horas, lleva al paciente a su objetivo glucémico. El paciente, en este caso, supera los 160 mg/dl acercándose a una situación de hiperglucemia.

En la Figura 4.6 el adelanto de la insulina es de 25 minutos y se puede observar como,

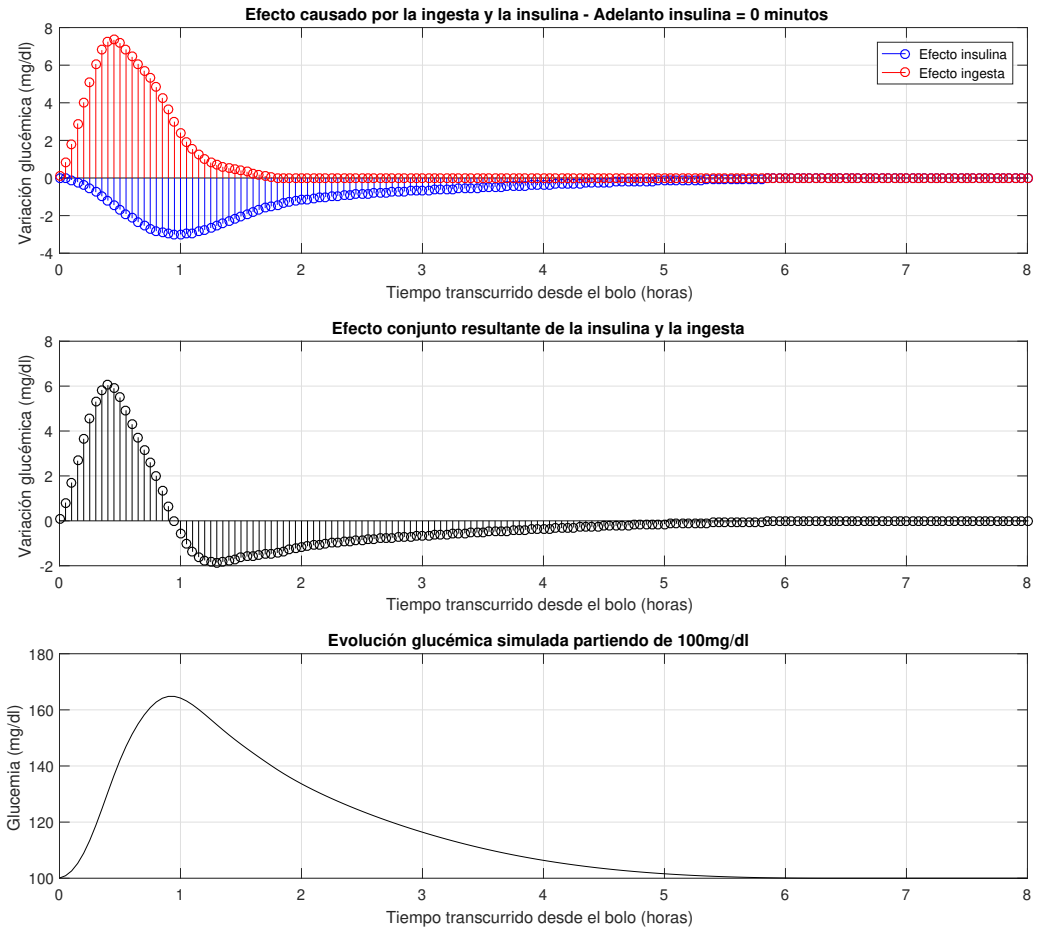


Figura 4.5: Simulación de ingesta con dosificación perfecta sin adelanto de insulina

sin ser suficiente el adelanto debido a que los hidratos siguen causando efecto antes que la insulina, la glucemia del paciente no supera los 160 mg/dl y acaba llegando a su objetivo tras unas horas.

En el caso de la Figura 4.7, se puede apreciar cómo los máximos de acción de la insulina y la ingesta coinciden aproximadamente en el tiempo y la glucemia del paciente llega a ligeramente superar los 140 mg/dl. Hacia la media hora desde la dosificación de la insulina se aprecia cómo únicamente la insulina está causando efecto, reduciendo ligeramente la glucemia del paciente por debajo de los 100 mg/dl, y justo después la ingesta produce la subida.

En la Figura 4.8 el adelanto de la insulina es, probablemente, excesivo y la insulina lleva a una situación de casi-hipoglucemia previa a la ingesta. Aunque a nivel de resultado glucémico (numéricamente hablando) pueda parecer que el caso de la Figura 4.8 es el más idóneo, en realidad no lo es: se ha aumentado mucho el riesgo de hipoglucemia para obtener una pequeña reducción del máximo glucémico debido a la ingesta.

Es importante hacer hincapié en el grado de indeterminación de todos estos parámetros:

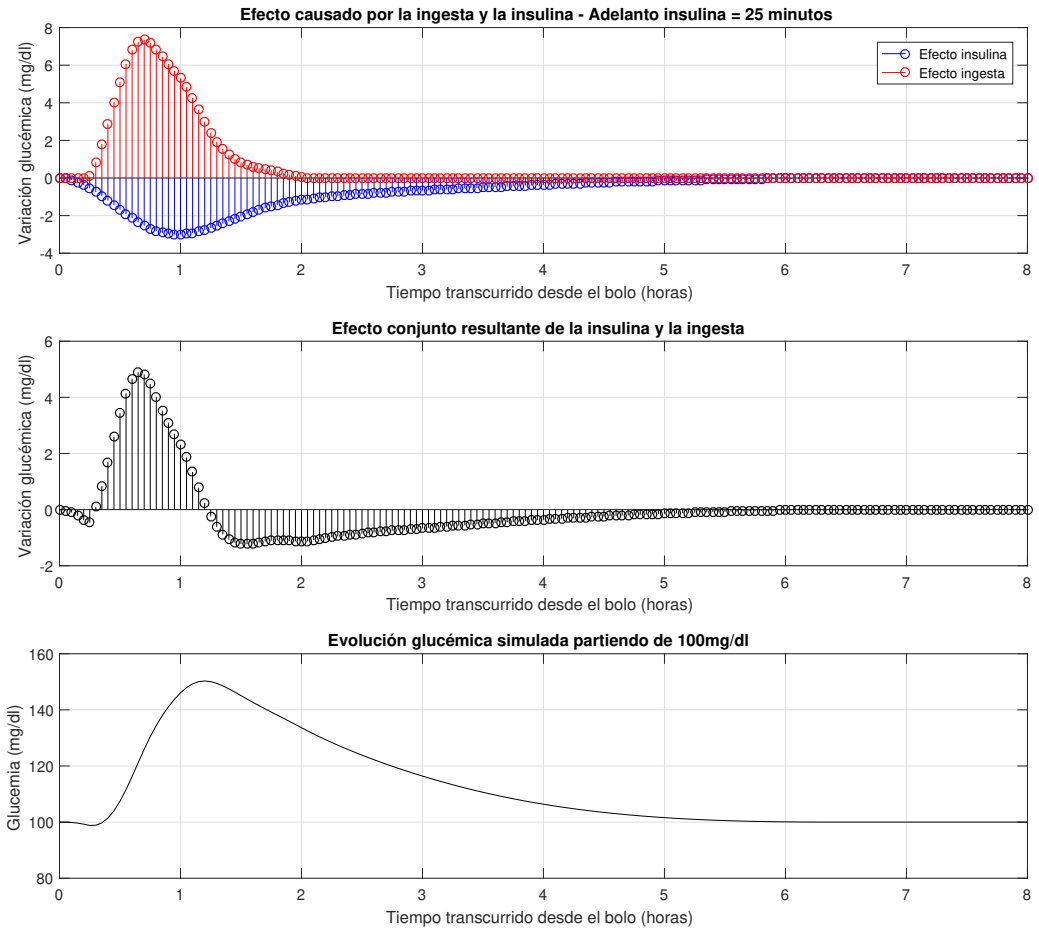


Figura 4.6: Simulación de ingesta con dosificación perfecta con adelanto de insulina de 25 minutos

la curva de acción de la insulina varía de persona a persona, la curva de absorción de los hidratos depende de múltiples parámetros que no son sencillos de determinar (contenido en gramos de cada ingrediente, índice glucémico, nivel de cocción, maduración, contenido en grasas, etc).

Puede parece que modelar la absorción de los hidratos de carbono es la manera apropiada de proceder pero el problema es, quizá, demasiado complejo para un algoritmo básico y, por ello, en esta tesis se hará uso de la segunda opción: suponer la ingesta compensada por los bolos de insulina dosificados por el paciente. Además, en siguientes apartados se explicará una técnica que ayudará a minimizar los problemas derivados de esta forma de actuar.

Al suponer que la ingesta y su bolo de insulina asociado están ajustados en el tiempo de manera que los máximos de acción coinciden en el tiempo y que los efectos de ambas partes se compensan perfectamente, sea o no sea así, podemos inhibir el lazo de control hasta que el pico de acción del bolo de insulina haya transcurrido. De esta manera el algoritmo no se verá afectado por las grandes fluctuaciones causadas por el comienzo del

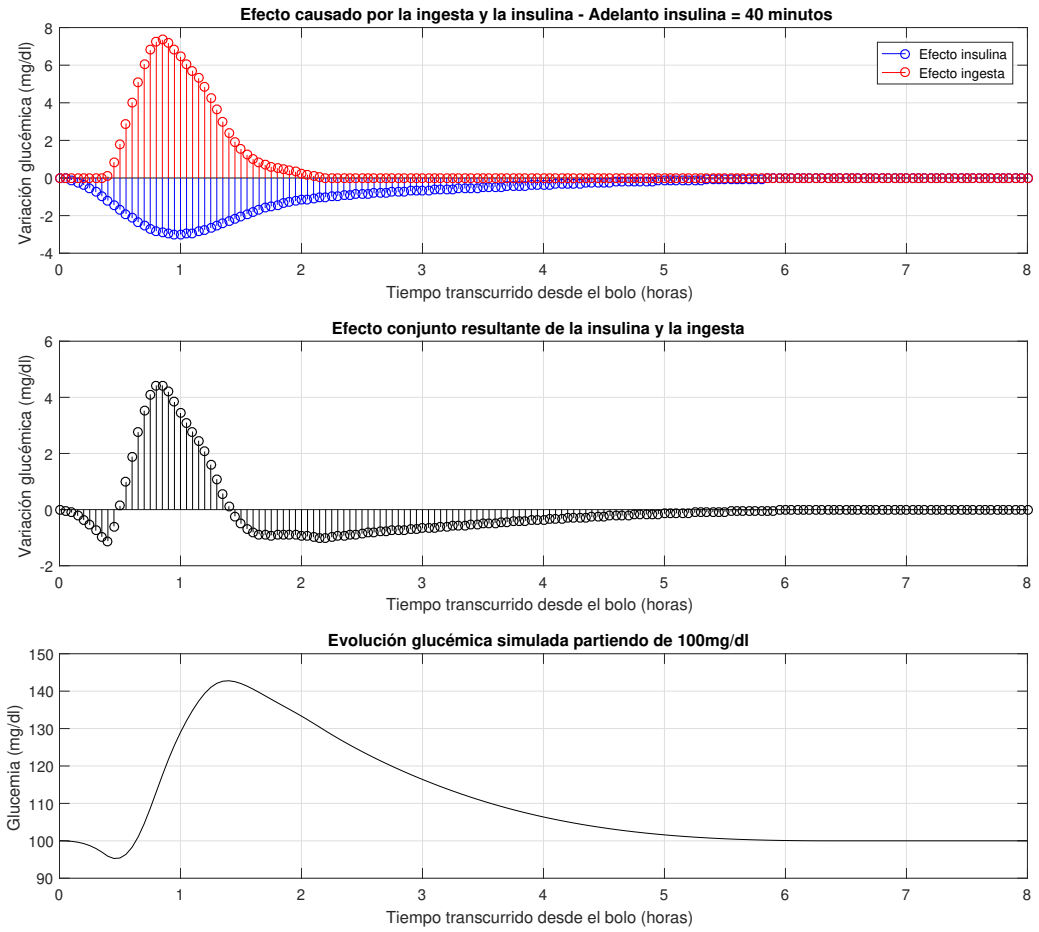


Figura 4.7: Simulación de ingesta con dosificación perfecta con adelanto de insulina de 40 minutos

proceso y podrá ayudar a compensar la evolución posterior.

4.3.4 Necesidades basales de insulina

Se define como Necesidades Basales a aquella cantidad de insulina que necesita el cuerpo del paciente en cada momento para mantener su glucemia estable. Estas necesidades basales se ven modificadas a lo largo del día por múltiples motivos, entre los que se encuentran los ciclos circadianos[68], el ejercicio físico[69][70][71][72], estrés y otras enfermedades[73].

Para dar respuesta a la variación de necesidades basales del paciente dentro en un mismo día, es común encontrar en los sistemas de lazo abierto actuales de uno a varios perfiles basales diarios programables. Cada perfil basal determina, según la hora del día, el número de unidades de insulina por hora que necesita el paciente para cubrir sus necesidades basales. Como dependiendo de otros factores estas necesidades se pueden ver alteradas, es también común encontrar que los pacientes pre-configuren perfiles basales adicionales para ajustarse a dichos cambios. Si el paciente consigue ajustar su perfil de

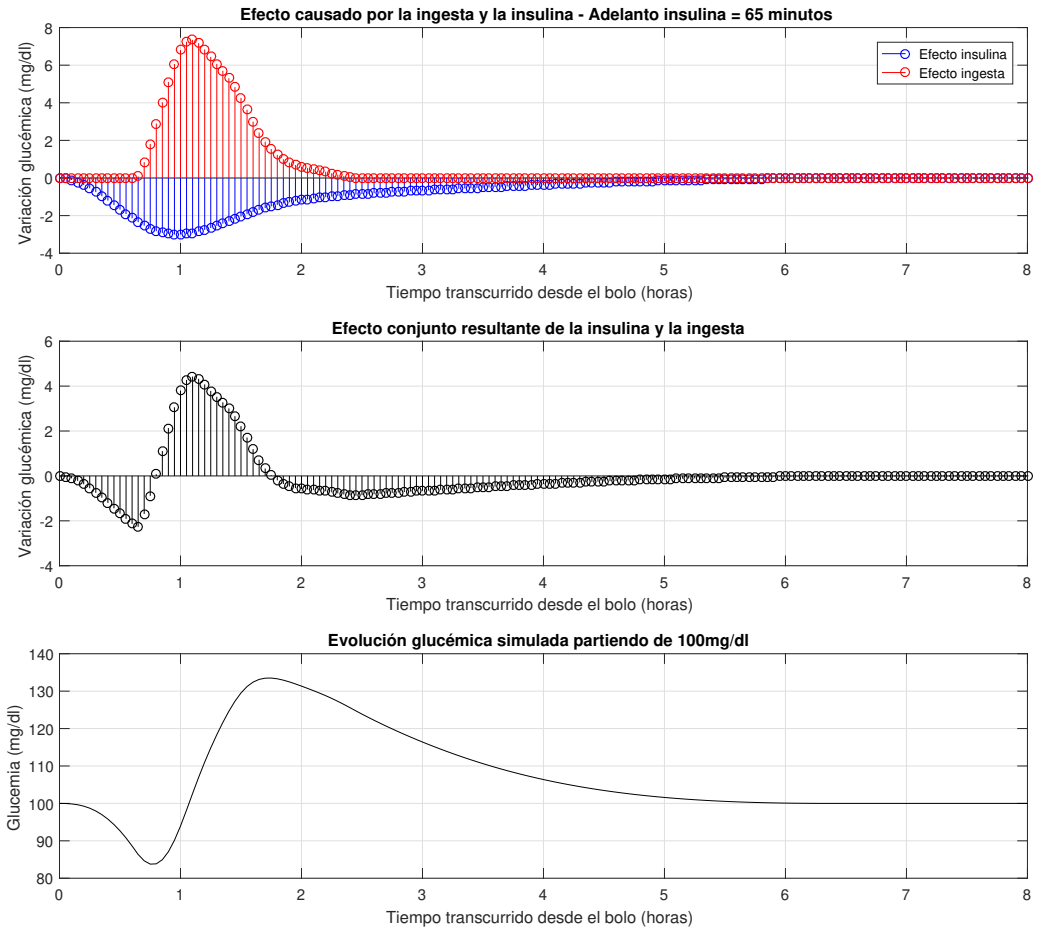


Figura 4.8: Simulación de ingesta con dosificación perfecta con adelanto de insulina de 65 minutos

dosificación basal a sus necesidades basales, logrará mantener su glucemia estable siempre que se encuentre fuera de la influencia de factores como las ingestas de hidratos de carbono o el ejercicio físico, los cuales tienen un gran impacto sobre el control glucémico.

Aquellos pacientes que hacen uso de bomba de insulina en lazo abierto suelen ajustar sus perfiles, con la ayuda de un médico endocrino, en función de resultados previos y análisis de datos históricos. Si el paciente espera a tener la ayuda de un médico endocrino para realizar estos ajustes, podría darse el caso de que el paciente viva varios meses (tiempo habitual entre consultas) con unos perfiles basales no ajustados. El procedimiento de ajuste de perfiles basales se escapa al alcance de esta tesis doctoral, pero cabe remarcar que el proceso es complejo y depende de una buena toma de datos históricos y un análisis experto de los mismos.

4.3.5 Insulina Activa negativa

El concepto de Insulina Activa, descrito en apartados anteriores (ver apartado 4.3.2), hace referencia a aquella insulina que, debido a su curva de acción y el tiempo transcurrido desde su dosificación, todavía debe causar efecto y sigue, por tanto, activa.

En sistemas de lazo abierto actuales sólo se toma como insulina que "cause efecto", y por tanto deba considerarse para el cómputo de la Insulina Activa, aquella insulina dosificada en calidad de bolo. Toda la insulina dosificada como perfil basal para cubrir las necesidades basales del paciente se considera la referencia sobre la que se calcula la insulina activa y, por tanto, no se incluye en el cómputo de la misma. De igual manera es normal que tampoco se considere como Insulina Activa la modificación temporal del perfil basal.

Con la aparición de los sistemas en lazo cerrado estas asunciones ya no son válidas. El sistema de control básico asume que las necesidades basales son las especificadas en el perfil basal programado en la bomba de insulina, pero podrá aumentar o reducir la tasa basal que recibe el paciente de manera temporal y con ello "esperar un efecto". Es por esto que en el cómputo de la insulina activa se deben considerar todas las modificaciones temporales del perfil basal.

Es sencillo comprender que, si se aumenta temporalmente la dosificación basal por encima de lo programado en el perfil, estamos dosificando insulina extra que aumentará la insulina activa. Pero, de manera opuesta, si se reduce la insulina dosificada por debajo de lo programado en el perfil basal obtendremos lo que se denomina como Insulina Activa negativa (ver Corrección de hipoglucemia en Figura 4.9). Esta reducción de insulina por debajo de las necesidades basales deberá provocar un incremento de la glucemia del paciente.

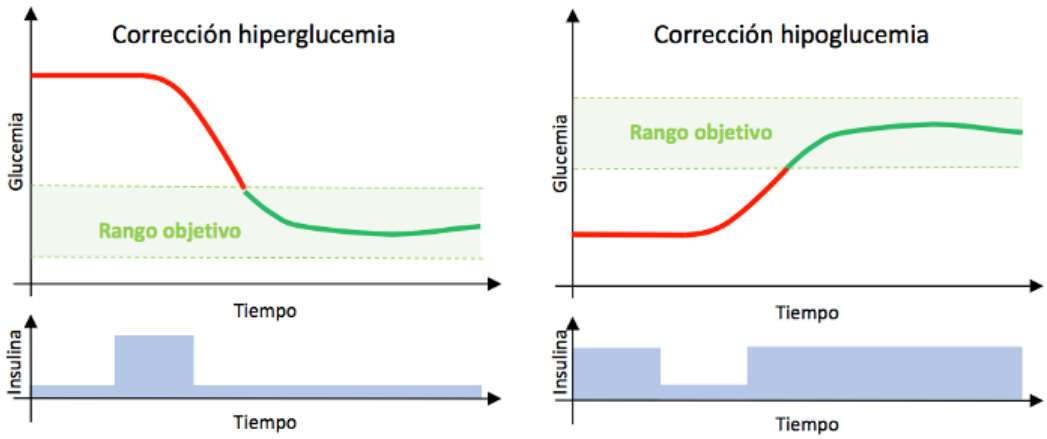


Figura 4.9: Ejemplo de acciones correctivas

Desgraciadamente, no es igual el grado de maniobra que se obtiene en un sentido que en

el otro. Mientras que el sistema de control puede aumentar en gran medida la insulina que recibe el paciente, y provocar con ello grandes decrementos de la glucemia del paciente, la insulina activa negativa tiene como límite las necesidades basales del paciente. La mayor reducción posible es igual a sus necesidades (o lo que es lo mismo, hacer que el perfil basal del paciente sea 0 unidades por hora) y estas pueden no ser suficientes para considerarlo una verdadera contra-actuación a la insulina. Es por ello que los sistemas de control con una única hormona (como es el descrito en esta tesis) deben ser muy garantistas a la hora de aumentar la insulina dosificada al paciente.

4.3.6 Factor de Sensibilidad a la Insulina

Se define como Factor de Sensibilidad a la Insulina (ISF - *Insulin Sensitivity Factor*) a la relación que existe entre una unidad de insulina y la reducción de glucemia esperada por ella. Sus unidades típicas son mg/dl/U. Al igual que sucede con los perfiles basales, es común encontrar que los sistemas en lazo abierto ya proporcionen la capacidad de configurar este parámetro como una tabla en función de la hora del día. El Factor de Sensibilidad a la Insulina también se ve afectado por los mismos condicionantes que las necesidades basales.

Gracias a este parámetro podemos estimar el efecto esperado de la insulina activa en la glucemia del paciente o calcular la insulina necesaria para compensar un cierto error glucémico en el lazo de control.

4.3.7 Estado glucémico del paciente

En el sistema propuesto se hace uso de un MCG para obtener la concentración de glucosa en el líquido intersticial‘ del paciente. Como ya se explicó en apartados anteriores (ver apartado 3.2), existe un retardo biológico entre la concentración en el liquido intersticial y la concentración en sangre de unos pocos minutos. El primer paso es, por tanto, intentar minimizar el error cometido en la estimación de la glucemia en sangre actual del paciente. Esto es de gran importancia en momentos en los que la glucemia intersticial tiene una tendencia descendiente y el riesgo de hipoglucemia se incrementa.

Dado que el retardo típico es de unos 10/15 minutos [74], el algoritmo básico de control tomará como estimación de glucemia actual a aquella resultante de la extrapolación lineal de la glucemia en los próximos 15 minutos partiendo de la última lectura del sensor (*SGV - Sensor Glucose Value*). Para ello se hace uso de la historia reciente de los datos del sensor y el gradiente de los mismos.

El algoritmo básico de control haría uso únicamente de los dos últimos datos del MCG para tener la referencia glucémica y poder calcular el último gradiente. A partir de ahí se extrapola el valor en 15 minutos.

Durante la fase de simulación del algoritmo (desarrollada en apartados posteriores) se probó a hacer medias ponderadas (filtro FIR) de los datos del MCG en vez de usar únicamente los dos últimos datos para, con ello, intentar minimizar el efecto del posible ruido contenido en las medidas. El efecto introducido por el filtrado en los datos del MCG hace que, en ciertas situaciones donde existe un cambio brusco de tendencia hacia la hipoglucemia, este retardo se traduzca en un retardo en la detección y prevención de hipoglucemias. Por este motivo se toma la decisión de usar únicamente los dos últimos datos del MCG y, con ello, primar la velocidad en la respuesta y con ello la seguridad del paciente.

La Ecuación 4.2 describe el procedimiento para calcular la $glucemia_{actual}$ del paciente para cada nuevo dato recibido del sensor. Para ello, tomando el último dato recibido del sensor (sgv_0) como punto de referencia, se añade el efecto equivalente a que se mantenga la tendencia actual (primera derivada de los datos del sensor) durante los siguiente 15 minutos. Para calcular dicho efecto se obtiene la modificación de glucemia ocurrida en los últimos 5 minutos (sgv_0 - sgv_{-1}), se divide por el periodo de muestreo del sensor (que en este caso es 5 minutos) para obtener la tasa de cambio por minuto y se multiplica por 15 para obtener la modificación de la glucemia del paciente esperada en 15 minutos.

$$glucemia_{actual} = sgv_0 + (sgv_0 - sgv_{-1}) \times \frac{15}{periodo_muestreo_minutos} \tag{4.2}$$

El sistema de control también calcula un tercer valor de glucemia que denominado glucemia final. La glucemia final es el valor de glucemia que se estima tendrá el paciente una vez la insulina activa (positiva o negativa) cause todo su efecto esperado. Para su cálculo se toma el valor de glucemia actual (el extrapolado anteriormente) y se le aplica el efecto de la insulina activa mediante el factor de sensibilidad a la insulina (ver Ecuación 4.3). Es sobre esta glucemia sobre la que el lazo de control calculará el error.

$$glucemia_{final} = glucemia_{actual} - IOB \times ISF \tag{4.3}$$

4.3.8 Objetivo glucémico

El objetivo glucémico puede ser definido como un rango de valores (típicamente 70 a 180 mg/dl) o un valor de glucemia concreto.

En el algoritmo de control básico, a la hora de tomar decisiones en cuanto a actuación, se toma un valor concreto como objetivo glucémico aunque se tenga definido un rango. El criterio seguido es tomar el valor medio del rango para así intentar mantener al paciente en el punto medio y tener margen suficiente para tolerar desviaciones.

En este sistema, dado que la bomba de insulina elegida nos ofrece los objetivos glucémicos en forma de rangos y organizados por tramos horarios en forma de tabla, existe

la posibilidad de variar el objetivo según la hora del día para intentar ajustarse de una manera más eficiente al modo de vida del paciente.

4.3.9 Gestión de ejercicio físico

El ejercicio físico, junto con las ingestas, es uno de los eventos que más impacto tiene sobre la glucemia de un paciente [72]. El ejercicio físico, en líneas generales, debe ser anunciado con cierto tiempo de antelación para permitir al sistema prepararse para los cambios que se van a producir.

Siendo la reducción de la glucemia el efecto más directo del ejercicio físico sobre el paciente, sobre todo en ejercicio cardiovascular, no es el único. El ejercicio físico suele venir acompañado de una reducción de las necesidades basales y un aumento significativo del factor de sensibilidad a la insulina. Estos últimos efectos pueden permanecer activos hasta varias horas una vez cese el ejercicio físico.

La reducción de necesidades basales y el aumento del factor de sensibilidad a la insulina suponen un problema importante en el caso de no hacer un anuncio previo del ejercicio. El sistema tiene establecidas unas necesidades, sensibilidades y objetivos glucémicos para su funcionamiento normal. En el caso de tener un buen control, la glucemia del paciente será relativamente baja y las cantidades de insulina ajustadas a sus necesidades. Si las necesidades basales se ven reducidas drásticamente (en comparación a los tiempos de acción de la insulina utilizada y descritos en la curva de acción), la situación es equivalente a tener un valor de insulina activa superior a lo deseado. Además, el factor de sensibilidad a la insulina se ve incrementado, con lo que el efecto de dicha insulina activa sobre la glucemia del paciente se ve magnificado. Esto puede llevar al paciente a una situación de hipoglucemia.

En el caso del sistema de control básico, debemos hacer todo lo posible para mantener la seguridad del paciente. El modelado del ejercicio físico no es sencillo: el efecto que éste tiene en el cuerpo depende muchos factores sobre los que ni el paciente ni el sistema tiene suficiente información. Es por ello que, ante el anuncio del ejercicio físico por parte del paciente, el controlador configura de manera inmediata un objetivo glucémico temporal superior (150 mg/dl) aumentando el margen glucémico con respecto al nivel de hipoglucemia y, con ello, reduciendo el riesgo de hipoglucemia. El anuncio debe ser realizado con el tiempo suficiente para que el controlador logre reducir la insulina activa y llevar al paciente a un rango “más seguro”. Este objetivo permanecerá establecido mientras dure el ejercicio físico o hasta que el paciente lo considere apropiado. Este objetivo temporal tiene varios propósitos:

1. El tener un objetivo glucémico superior al normal dará cierto margen extra durante el ejercicio, glucémicamente hablando, y existirá menor riesgo de hipoglucemia si el ejercicio provoca una disminución directa de la glucemia del paciente.

2. Como el objetivo glucémico temporal debe activarse previo al ejercicio, el controlador reducirá la insulina activa para intentar llevar al paciente a dicho objetivo. Al tener menor insulina activa, el paciente verá reducida su probabilidad de hipoglucemia debida simplemente a la disminución de sus necesidades basales y a su incremento de sensibilidad.

4.3.10 Suspensión por hipoglucemia

Como se ha descrito previamente en apartados anteriores, la prioridad del algoritmo de control básico es mantener la seguridad del paciente. Es por ello que se implementa un sistema denominado "suspensión por hipoglucemia" y tiene prioridad a cualquier otro procedimiento o cálculo existente en el sistema.

Esta protección consiste en la suspensión de cualquier infusión de insulina al paciente si su estado glucémico indica que corre riesgo de sufrir una hipoglucemia o se encuentra en estado de hipoglucemia actualmente. Para tomar la decisión de suspender la infusión se establece un umbral glucémico por debajo del cual se activa esta protección. En el sistema de control básico se utiliza el nivel mínimo del rango objetivo como umbral de suspensión por hipoglucemia.

Si la glucemia reportada por el MCG, la glucemia actual o la glucemia final (ver apartado 4.3.7), o cualquier combinación de ellas, se encuentra por debajo del umbral de suspensión, se comanda directamente una basal temporal de 0 U/hr durante 30 minutos a la bomba de insulina. De esta manera se suspende la infusión durante 30 minutos, renovable cada 5 minutos si hiciera falta, hasta que desaparezca esta situación. Si se perdiera la comunicación entre la bomba de insulina y el controlador, tras esos 30 minutos se volvería automáticamente a la terapia programada en la bomba de insulina y así se elimina correr el riesgo de llevar al paciente a hiperglucemia o incluso a cetoacidosis por dejar la infusión suspendida de manera permanente y sin control. Sólo en el caso de que las tres glucemias se encuentren por encima del umbral de suspensión se permitirá al lazo de control tomar otro tipo de decisiones.

4.3.11 Cálculo de corrección

Como ya se ha explicado en el apartado 4.3.7, el error glucémico se calcula como la diferencia entre la glucemia final del paciente (aquella obtenida después de aplicar los efectos esperados de la insulina activa) y su objetivo glucémico (ver apartado 4.3.8).

En la figura 4.10 se muestra, a nivel gráfico y conceptual, un ejemplo en el que se calcula el error glucémico a ser compensado. En esta figura se pueden apreciar dos gráficas distintas mostrando la evolución con el tiempo de la glucemia del paciente (gráfica superior) y la insulina que está siendo dosificada en cada momento (gráfica inferior). La línea de puntos vertical indica en ambas gráficas el momento actual en el tiempo, indicando

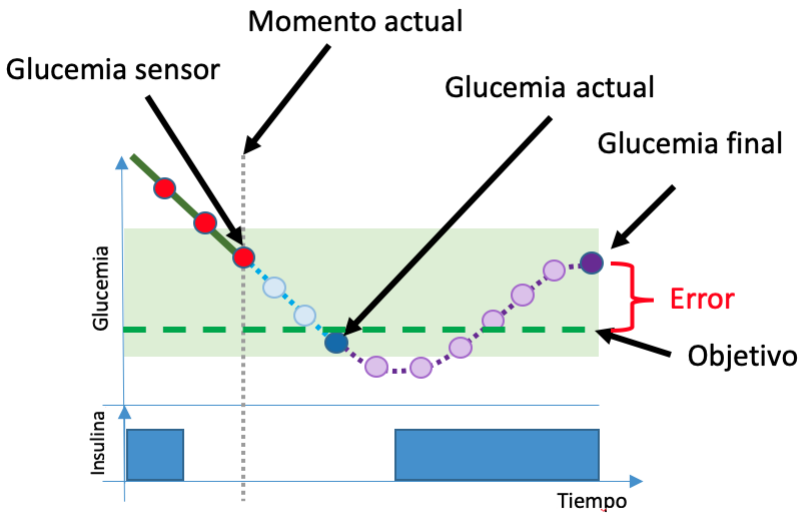


Figura 4.10: Ejemplo de cálculo de corrección - Paso 1

también que ambas gráficas están sincronizadas en el tiempo. La gráfica de la glucemia muestra las tres últimas lecturas del MCG (representadas mediante puntos rojos) siendo la última lectura la que coincide con la linea del momento actual. Siguiendo la tendencia de los datos recibidos del MCG, se calcula la evolución esperada de la glucemia en los siguientes 15 minutos (representada mediante puntos azules), obteniendo la glucemia actual del paciente (representada mediante un punto azul oscuro). Partiendo de la glucemia actual y la historia de dosificación de insulina , entre otros parámetros, se calcula la evolución de la glucemia del paciente (representada mediante puntos violetas) obteniendo así el valor de glucemia final (representado por un punto violeta oscuro). La gráfica de dosificación de insulina muestra cómo, en un tiempo pasado, se comandó una reducción de la insulina basal en previsión de una posible hipoglucemia y esta se mantiene hasta un cierto momento en el futuro. Es por este motivo que la evolución de la glucemia del paciente se ha representado de manera que se evita la hipoglucemia e incluso se llega a superar el objetivo glucémico marcado (insulina activa negativa - ver Apartado 4.3.5). Como ya se ha explicado en apartados anteriores, el error se obtiene como la diferencia entre el valor de glucemia final y el objetivo glucémico (representado en la gráfica mediante una línea horizontal verde).

Mediante el factor de sensibilidad a la insulina (ver apartado 4.3.6) se deriva la cantidad de insulina necesaria para compensar dicho error.

El sistema de control básico comanda las correcciones en forma de basales temporales. Debido a que el mínimo tiempo configurable para una basal temporal en la bomba de insulina utilizada es de 30 minutos, se toma este tiempo como referencia para todas las correcciones. Además, el comandar una basal temporal no implica que sea de obligado cumplimiento hasta el final de la misma: se pueden cancelar en cualquier momento o

comandar otra basal temporal que reemplace a la primera. Con cada ciclo de ejecución del lazo de control (cada 5 minutos aproximadamente) se re-evalúa la corrección necesaria y se puede alterar la terapia si fuera preciso.

La corrección será comandada como una basal temporal de 30 minutos con un ritmo de infusión tal que aglutine los efectos de las necesidades basales más la corrección calculada. En el algoritmo de control básico, las necesidades basales vienen determinadas por el perfil basal configurado en la bomba de insulina. Y la insulina necesaria para la corrección deberá ser distribuida en esos 30 minutos.

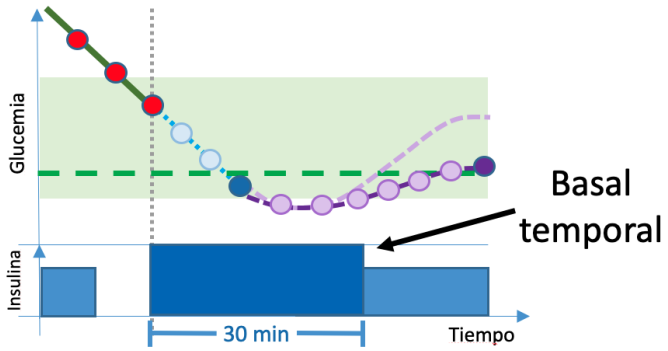


Figura 4.11: Ejemplo de cálculo de corrección - Paso 2

La figura 4.11, al igual que sucede con la figura 4.10, muestra de manera conceptual la aplicación de una basal temporal como corrección. La cantidad de insulina programada para su dosificación como basal temporal debe alterar el cálculo de glucemia final de manera que la glucemia tienda al valor objetivo. Los puntos violetas en la figura han sido actualizados para ilustrar el cambio en la estimación de la evolución glucémica y se ha mantenido un linea violeta de guiones como referencia de la evolución prevista previa a la corrección. Este cálculo se repetirá con el siguiente dato recibido por parte del MCG y se volverá a compensar el nuevo error calculado.

Las correcciones tienen dos niveles de saturación:

- Las reducciones de insulina están limitadas a, como máximo, reducir a 0 U/hr la infusión. Al ser un sistema de única hormona, no existe mecanismo controlable por el sistema para forzar un incremento de la glucemia del paciente. Es por este motivo que, como se habrá podido apreciar en el desarrollo de la algoritmia, siempre se prioriza la seguridad y el evitar situaciones de sobre-dosificación de insulina.

- Los incrementos de la dosificación de insulina están limitados por la basal temporal máxima configurada en la bomba. Este parámetro tiene el doble cometido de servir como mecanismo de seguridad, pues nadie puede aumentar la dosificación basal por encima de ese parámetro, y como limitador de la “agresividad” del algoritmo. Al tener

una basal temporal máxima, independientemente de que el controlado considere que sería necesario dosificar más insulina, sólo se permitirá esta cantidad. De esta manera, en el siguiente ciclo de ejecución y sólo si sigue siendo necesaria, podrá ser dosificada. Pero, si en el siguiente ciclo de ejecución del lazo de control la situación glucémica es distinta, todavía existe la posibilidad de no dosificarla y mantener seguro al paciente. Este límite, al igual que el resto de configuraciones sensibles en la bomba de insulina, debe ser ajustado con precaución y, preferiblemente, por un médico endocrino.

4.4 Flujo lógico del algoritmo

El algoritmo que se ha denominado como "básico" modifica la tasa basal que recibe el paciente para tratar de compensar las desviaciones glucémicas con respecto al objetivo glucémico configurado.

Existe un proceso de inicialización que debe ser ejecutado para que el algoritmo parta de una situación conocida y así se eviten problemas de condiciones de borde:

- *Buffer de historia de datos del sensor*: El sistema debe asegurarse de que los buffers que contienen los datos del MCG no contienen datos y, de existir, deben ser borrados. También se debe borrar el buffer que determina la validez de los datos del MCG. De esta manera el sistema no genera glucemias futuras erróneas debido al uso de datos incorrectos.

- *Buffer de historia de dosificación de insulina*: Ya que la historia de dosificación de insulina es convolucionada con la curva de acción de la insulina (ver figura 4.1) durante la ejecución del lazo de control, no se puede borrar estos datos como sucede con los datos provenientes del MCG. Al convolucionar los vectores se entremezclan los datos válidos con los no válidos. Es por este motivo que en el proceso de inicialización se opta por simular que se ha estado dosificando las cantidades especificadas en el perfil basal del paciente e inicializar la historia con estos datos. Al inicializar la historia con el perfil basal, la insulina activa resultante del cálculo en la primera ejecución del lazo será nula (pues la dosificación hasta ese momento coincidirá con las necesidades basales).

Una vez el sistema ha sido inicializado, se espera a la recepción de datos nuevos del MCG. Una vez se han recibido los datos, se procede a la ejecución del lazo de control. El flujo de ejecución del lazo de control, el cual contiene parte del firmware de soporte básico de la plataforma y la función que contiene el código para el lazo cerrado, ha sido representado en la figura 4.12 y contiene los siguientes pasos:

1. *Ajuste temporal de los buffers de historia*: Estos buffers son aquellos que contienen toda la historia de datos del MCG y las dosificaciones de insulina realizadas y pre-

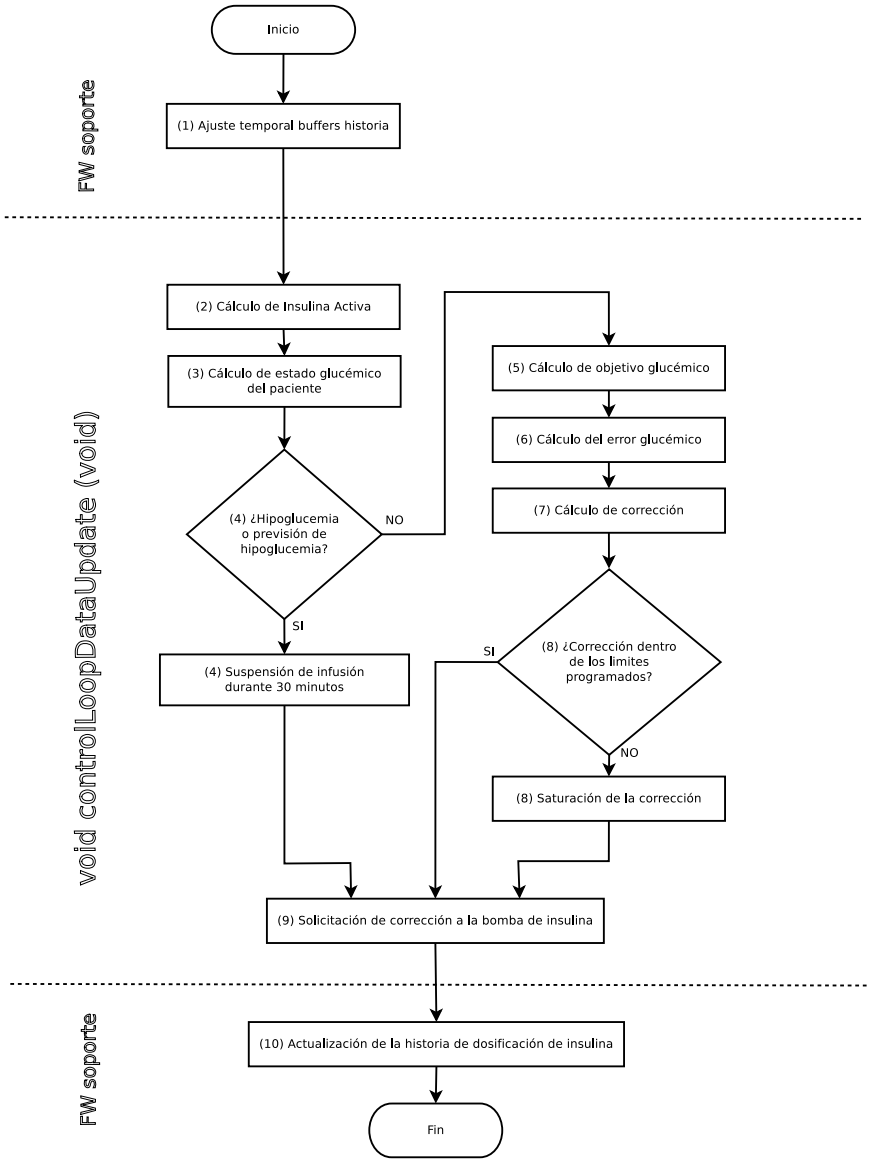


Figura 4.12: Diagrama del flujo lógico del algoritmo

vistas. Este ajuste consiste en desplazar los datos en los buffers una posición hacia el pasado de manera que se refleje en los mismos que ha pasado un periodo tiempo.

2. *Cálculo de Insulina Activa*: El controlador hará uso de la curva de acción de la insulina utilizada y la historia de dosificación para realizar el cálculo de insulina activa.

3. *Cálculo de estado glucémico del paciente*: Mediante los datos actuales e históricos del MCG, y la insulina activa, se calculan la glucemia actual y final del paciente.

4. *Comprobación de suspensión por hipoglucemia*: Si cualquiera de las glucemias calculadas en este periodo de ejecución son inferiores al límite inferior del rango objetivo de glucemia, se suspende la infusión de insulina por 30 minutos y se termina la ejecución del lazo de control hasta el siguiente periodo. En caso contrario, se continúa con el siguiente paso.

5. *Cálculo del objetivo glucémico*: Haciendo uso del rango objetivo configurado en la bomba de insulina (ver apartado 4.3.8), se calcula el valor exacto del objetivo glucémico para este periodo de ejecución del lazo de control.

6. *Cálculo del error glucémico*: Comparando el objetivo glucémico y la glucemia final calculada para el paciente, se calcula el error glucémico a compensar en este periodo de ejecución.

7. *Cálculo de la corrección*: El error glucémico se traduce a necesidades de insulina mediante el factor de sensibilidad a la insulina y se calcula la basal temporal necesaria para dosificar dicha cantidad de insulina en 30 minutos.

8. *Saturación de la corrección*: Si la corrección calculada supera el nivel definido como basal temporal máxima, se satura el valor de la corrección a dicho nivel máximo.

9. *Solicitar la corrección a la bomba de insulina*: Haciendo uso de las funciones proporcionadas por el *firmware* base de la plataforma *wearable*, se solicita el inicio de una basal temporal acorde a la corrección calculada.

10. *Actualización de la historia de dosificación de insulina*: Sólo en el caso de recibir la confirmación por parte de la bomba de insulina, notificando que la basal temporal ha sido configurada, se actualiza la historia de dosificación de insulina para que este cambio quede reflejado y sea tenido en cuenta en la siguiente ejecución del lazo de control. En caso de no recibir la confirmación por parte de la bomba de insulina, la historia permanecerá sin cambio alguno pues no se llegó a comandar la corrección.

4.5 Seguridad médica del paciente

Cualquier sistema que pueda alterar la medicación de un paciente, y tenga la capacidad de poner en peligro su vida, debe tener como prioridad la seguridad médica de dicho paciente.

En este sistema son diversos los sistemas y decisiones de funcionamiento que se han implementado para maximizar la seguridad:

- *Sistema de suspensión por hipoglucemia*: Si el estado glucémico del paciente confirma una hipoglucemia o el riesgo de hipoglucemia, el sistema suspenderá la infusión de insulina hasta que la situación quede controlada y el paciente fuera de peligro (ver apartado 4.3.10).

- *Correcciones basadas en basales temporales*: Podría parecer apropiado implementar un sistema en lazo cerrado basado en bolos de insulina (también llamados microbolos), pero esta manera de funcionamiento conlleva ciertos riesgos cuando no se puede garantizar la comunicación entre el algoritmo y la bomba de insulina. Para poder

realizar una terapia basada en microbolos se debería configurar un perfil basal de 0 U/hr durante todo el día. Esta sería la única manera de poder reducir la infusión cuando fuera necesario (ya que la terapia se basa en bolos de insulina). Pero, en el caso de una pérdida de comunicaciones entre el controlador y la bomba, se estaría dejando al paciente sin cubrir sus necesidades basales y podríamos poner en riesgo de cetoacidosis diabética al paciente. En un sistema basado en basales temporales, si sucede el fallo de comunicaciones, tras los 30 minutos de la última basal temporal la bomba recupera sus funciones normales y se dosificaría el perfil basal programado, eso sí, en lazo abierto. Es una aproximación al problema mucho más segura.

- *No apilamiento de correcciones*: El sistema de correcciones, como ya se ha explicado, está basado en basales temporales y no en microbolos. Aunque la cantidad máxima de insulina por bolo está limitada por un parámetro en la bomba de insulina, no ocurre lo mismo con el número máximo de bolos que se pueden dosificar. Si el sistema estuviera basado en microbolos, no existiría manera de asegurar que ante una mala ejecución del algoritmo no se pudiera comandar más de un bolo de insulina: si se comanda por error varias veces la misma corrección, estaríamos sumando los efectos de todos los bolos. En caso de las basales temporales, una basal temporal nueva cancela la anterior, con lo cual podemos comandar la misma basal varias veces seguidas que los efectos no se suman como resultado.
- *Límites de actuación en basales temporales*: La bomba de insulina limita el número de unidades por hora que se pueden programar en calidad de basal temporal. Este parámetro, configurable por el paciente en la bomba de insulina, evita que se lleguen a niveles de actuación peligrosos y se permite controlar así la "agresividad" del algoritmo.

4.6 Implementación de algoritmos alternativos

La plataforma desarrollada en esta tesis sirve de soporte para la implementación del algoritmo de control descrito en este capítulo, pero ello no quiere decir que sea el único algoritmo implementable. La plataforma *hardware* y el *firmware* básico de soporte dan cabida a implementar algoritmos diversos con metodologías distintas.

Es importante mencionar que, en caso de que se desee reutilizar esta plataforma para pruebas de otro algoritmo, es suficiente con implementar la función de proceso del nuevo algoritmo dentro de la función `void controlLoopDataUpdate (void)`.

4.7 Conclusiones

En este capítulo se parte de una plataforma *wearable* capaz de comunicarse con los dispositivos existentes en el sistema y de ofrecer capacidades firmware suficientes para ejecutar

un algoritmo de control.

El comienzo del capítulo responde a la necesidad de explicar al lector la situación que encontraría el diseñador de un algoritmo al utilizar esta plataforma. En resumen, existe un único punto de entrada a la función que debe realizar los cálculos con cada iteración del lazo de control y existe una librería de funciones que permite el hacer cálculos comunes en cuanto a diabetes. Están también a disposición del diseñador un conjunto de variables globales (sólo lectura) que permite un sencillo acceso a las configuraciones del paciente. Estas variables globales permiten el acceso a:

- Hora del reloj en tiempo real
- Tabla de sensibilidad a la insulina
- Tabla de objetivos glucémicos
- Perfil basal programado
- Historia de datos del MCG
- Historia de dosificación de insulina
- Curva de respuesta a la insulina

Una vez descrito el punto de partida, y antes de desarrollar el flujo lógico del algoritmo, se describen conceptos básicos necesarios para la correcta compresión del mismo. Estos conceptos son:

- Curva de acción de la insulina
- Insulina activa e Insulina activa negativa
- Necesidades basales de insulina
- Factor de sensibilidad a la insulina
- Estado glucémico del paciente
- Modelos de gestión de ingestas
- Gestión de ejercicio físico
- Objetivo glucémico
- Suspensión por hipoglucemia
- Cálculo de correcciones

Con estos conceptos descritos, se desarrolla un algoritmo básico de control y se detalla su flujo lógico.

En capítulos anteriores se ha analizado el sistema desde el punto de vista de la seguridad en las comunicaciones. En este capítulo se ha hecho hincapié en describir cómo toda actuación siempre está considerada desde el punto de vista de maximizar la seguridad médica, pues el sistema además de seguro debe ser confiable, y en ser conservador a la hora de dosificar insulina al paciente. La seguridad médica se proporciona mediante los siguientes conceptos:

- Sistema de suspensión por hipoglucemia
- Correcciones basadas en basales temporales en vez de microbolos
- No apilamiento de correcciones
- Límites de actuación en basales temporales

Todo ello ofrece como resultado un algoritmo seguro e implementable en una plataforma *wearable* que será simulado y evaluado en consecutivos capítulos.

Capítulo 5

Estimación dinámica de necesidades basales de insulina

Los científicos estudian el mundo tal como es; los ingenieros crean el mundo que nunca ha sido.

Theodore Von Karman

5.1 Introducción

Existen estudios que muestran que las necesidades basales de insulina cambian dependiendo de factores como la hora del día, el ejercicio físico[69][70][71][72], estrés, otras enfermedades[73], etc. La mayor parte de los sistemas de control en lazo cerrado estudiados (principalmente sistemas Do-It-Yourself, los cuales son los que están públicamente documentados[8][9][10]) hacen la asunción de que las necesidades basales para un paciente en concreto son las mismas de un día para otro. Es por este motivo que normalmente se utiliza un único perfil basal y este es refinado iterativamente con información de días pasados. El paciente es responsable de activar los modificadores pertinentes para estos parámetros en situaciones de ejercicio físico o enfermedad y así compensar los efectos derivados de estas situaciones y lograr un buen control glucémico.

En esta tesis doctoral se propone un algoritmo que ha sido diseñado para evaluar de manera dinámica las necesidades basales de insulina del paciente y ajustar su dosificación en lazo cerrado. Este algoritmo reemplaza el perfil basal programado en la bomba de insulina como referencia de necesidades basales y se ejecuta en la plataforma *wearable*, dentro del algoritmo de control, con cada iteración del mismo, tal y como se puede apreciar en la figura 5.1. Esta solución ofrece ventajas como ajustarse dinámicamente en situaciones de enfermedad, ejercicio físico, ingestas con alto contenido en grasas o incluso a situaciones en las que la absorción de insulina en el punto de infusión comienza a no ser óptima. Por otro lado, existe todavía un conjunto de parámetros que deben ser ajustados manualmente para lograr la correcta determinación de las necesidades basales. Algunos

de estos parámetros son los factores de sensibilidad a la insulina y la curva de acción del tipo de insulina que se está dosificando al paciente (normalmente definido por el tipo de insulina y la duración de su efecto en el paciente).

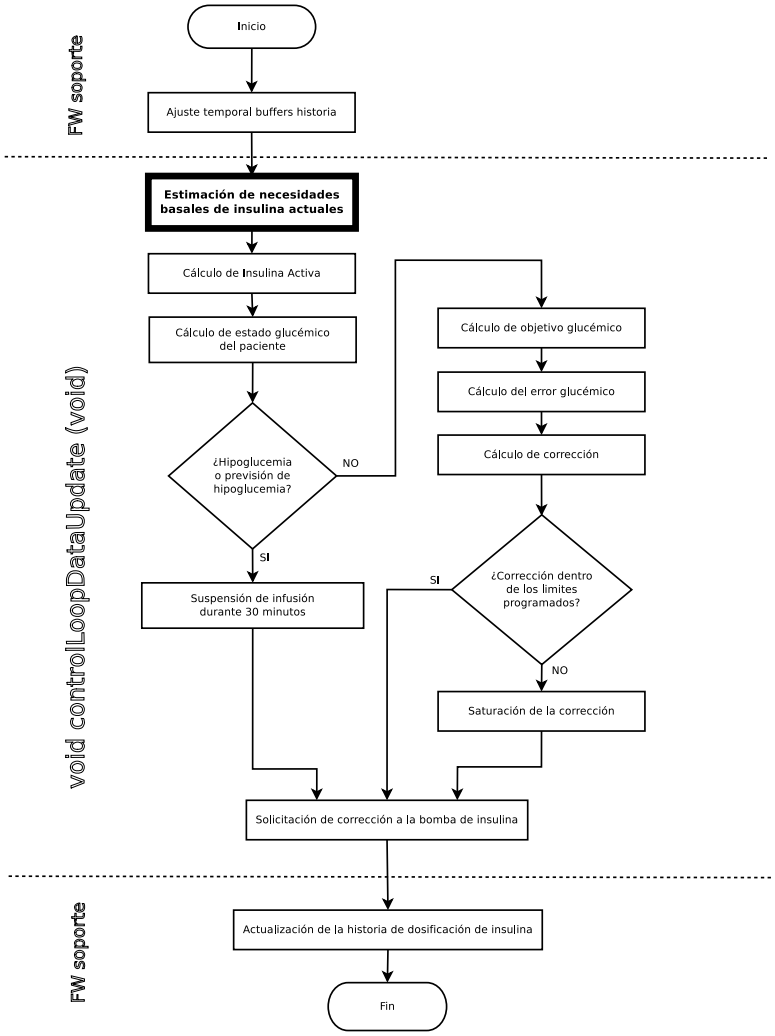


Figura 5.1: Diagrama de flujo del algoritmo incluyendo la determinación de necesidades basales de insulina.

El método propuesto hace uso de un conjunto de parámetros predefinidos para crear un modelo de cómo se espera que cause efecto la insulina dosificada y, desde ese punto, obtener las correspondientes necesidades basales. Estas necesidades basales serán utilizadas posteriormente, utilizando como base el algoritmo de control básico ya descrito (ver Capítulo 4), para determinar la acción apropiada a ser tomada.

5.2 Importancia de las necesidades basales de insulina como parámetro

Para comprender la importancia de un correcto ajuste de las necesidades basales de insulina es importante comprender el origen de este tipo de terapia.

Cuando un paciente depende de una terapia basada en múltiples dosis o inyecciones diarias de insulina (MDI) para controlar sus niveles de glucosa en sangre, las necesidades basales se cubren normalmente utilizando insulinas de larga duración [75][76][77]. Este tipo de insulinas tienen una respuesta más o menos plana hasta 24 horas tras su dosificación, permitiendo al paciente el uso de una o dos inyecciones diarias como insulina basal y minimizando el número de veces que el paciente debe inyectarse manualmente insulina. Mientras que es cierto que un paciente puede cubrir sus necesidades de manera más o menos cómoda con una única inyección diaria, al mismo tiempo, casi no existe flexibilidad en cuanto al ajuste de necesidades basales específicas por las cuales un paciente pudiera necesitar más o menos insulina en ciertos periodos del día.

Si se utiliza terapia basada en bomba de insulina [78], también denominada Infusión Continua Subcutánea de Insulina (ICSI), el paciente utilizará un único tipo de insulina (típicamente insulina rápida o ultra-rápida). Mediante la dosificación automática de pequeños microbolos cada pocos minutos se puede personalizar la insulina basal que recibe el paciente en cada momento (perfil basal), permitiendo a su vez una mayor adaptación a las necesidades reales y ofreciendo la posibilidad de tener un mejor control glucémico. Si el paciente prevee que sus necesidades basales van a cambiar en el futuro cercano, deberá hacer uso de las basales temporales para lograr el control glucémico deseado. Esta terapia ofrece una mejor adaptación y control, pero también aumenta la carga de trabajo y carga mental del paciente.

Este tipo de terapias requieren un ajuste preciso de los perfiles basales para conseguir la mejor adaptación posible a las necesidades basales del paciente [79]. Si la insulina basal, siendo esta insulina de larga duración en MDI o cualquier otra insulina dosificada por una bomba de insulina, no cubre las necesidades basales la glucemia del paciente aumentará. De manera contraria, si la insulina basal dosificada es superior a las necesidades basales, la glucemia del paciente decrecerá y aumentará el riesgo de hipoglucemia. En el primer caso el paciente corregirá su estado hiperglucémico utilizando lo que se conoce como bolo corrector. El objetivo de estos bolos correctores es compensar el exceso de glucosa en sangre del paciente y recuperar los niveles normales. En el segundo caso, si el paciente recibe demasiada insulina, la glucemia decrecerá pudiendo llegar a sufrir una hipoglucemia y, en ese caso, el paciente se verá obligado a ingerir hidratos de carbono o utilizar glucagón para corregir la situación.

Un ajuste incorrecto de los perfiles basales provocará que el paciente se vea forzado a una monitorización más intensiva de su glucemia para evitar grandes desviaciones y a usar hidratos de carbono de recuperación y/o bolos correctores de manera habitual. Esto se traduce en una mayor carga mental para el paciente y un decremento de su calidad de vida. Con este algoritmo se determina de manera automática las necesidades basales mejorando el control glucémico del paciente y, no con menor importancia, reduciendo la carga mental que la enfermedad provoca.

5.3 Algoritmia de estimación de necesidades basales

Las necesidades basales del paciente no solo varían de día a día sino que también pueden variar a cada momento dependiendo de muchos factores. Para abordar este problema, el algoritmo propuesto determina las necesidades basales de manera dinámica permitiendo un mejor ajuste de la terapia en cada momento.

El algoritmo de estimación de necesidades basales hace uso de información conocida del paciente como sus factores de sensibilidad a la insulina y la curva de acción de la misma para determinar cuanta insulina está causando efecto en el cuerpo en cada momento. Evaluando esta cantidad de insulina y el resultado obtenido sobre la glucemia se derivan las necesidades del basales del paciente. Además, se utiliza esta información y su historial para hacer otros ajustes que aumenten la seguridad del paciente.

5.3.1 Descomposición de bolos de insulina

Como ya se ha explicado en capítulos anteriores, cada combinación de tipo de insulina y paciente tiene su propia respuesta en el tiempo. El efecto de una unidad de insulina puede tardar hasta varias horas en desaparecer del cuerpo y, por consiguiente, debe ser tenida en cuenta. El efecto de cada bolo de insulina puede ser traducido o descompuesto en lo que se denomina en esta tesis como serie de "bolos instantáneos endógeno-equivalentes" o, simplemente, "bolos instantáneos".

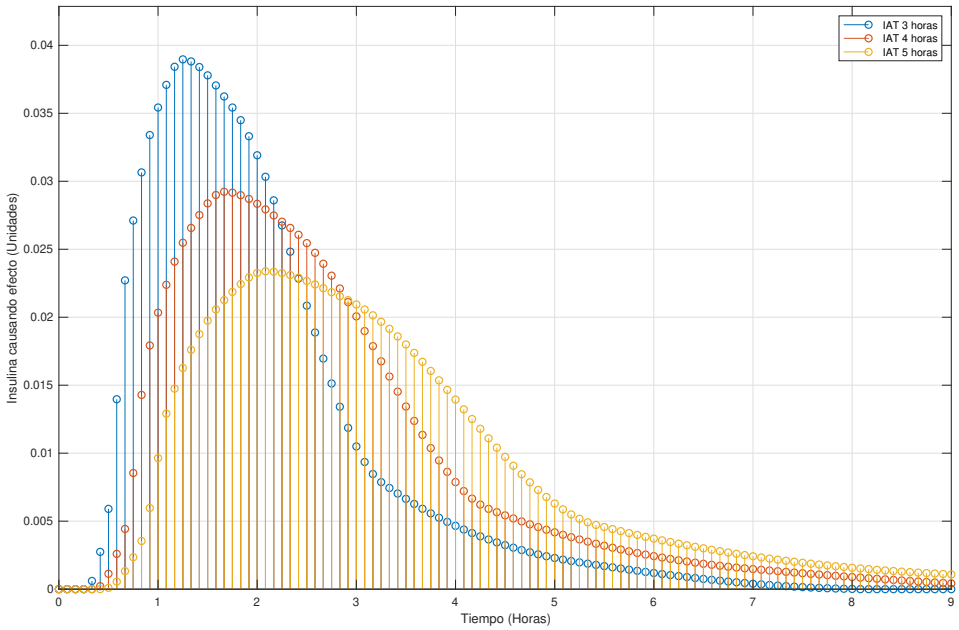


Figura 5.2: Ejemplo de curva de bolos instantáneos de una unidad de insulina con diferentes duraciones.

En la Figura 5.2, cada punto corresponde con lo un "bolo instantáneo". Un bolo instantáneo (iBolus) se define como la cantidad de insulina que, de acuerdo a la curva de

acción del tipo de insulina, está causando efecto en el cuerpo del paciente en un momento concreto del tiempo. De manera análoga, se define "Insulina Activa Instantánea" (iIOB) como la suma de todos los bolos instantáneos, provenientes de distintos bolos, que ocurren al mismo tiempo. En esta Figura también se puede observar cómo, dependiendo del tiempo que permanezca activa la insulina en el cuerpo (IAT - *Insulin Action Time*), la distribución de bolos instantáneos cambia, incluso para la misma cantidad de insulina (una unidad en el ejemplo).

A modo de ejemplo, asumiendo que el paciente tiene una duración de la insulina (IAT) de 4 horas, si el paciente recibió un bolo de 0.5 unidades hace dos horas y otro bolo de 2 unidades hace 1 hora, los bolos instantáneos que estan causando efecto en el cuerpo del paciente pueden ser calculados como:

- iBolus1 = 0.5 * IAT_{4h} (2.0)
- iBolus2 = 2.0 * IAT_{4h} (1.0)

La insulina activa instantánea (iIOB) para el paciente en ese preciso momento será igual al sumatorio de todos los bolos instantáneos (iBolus1 e iBolus2 en el ejemplo).

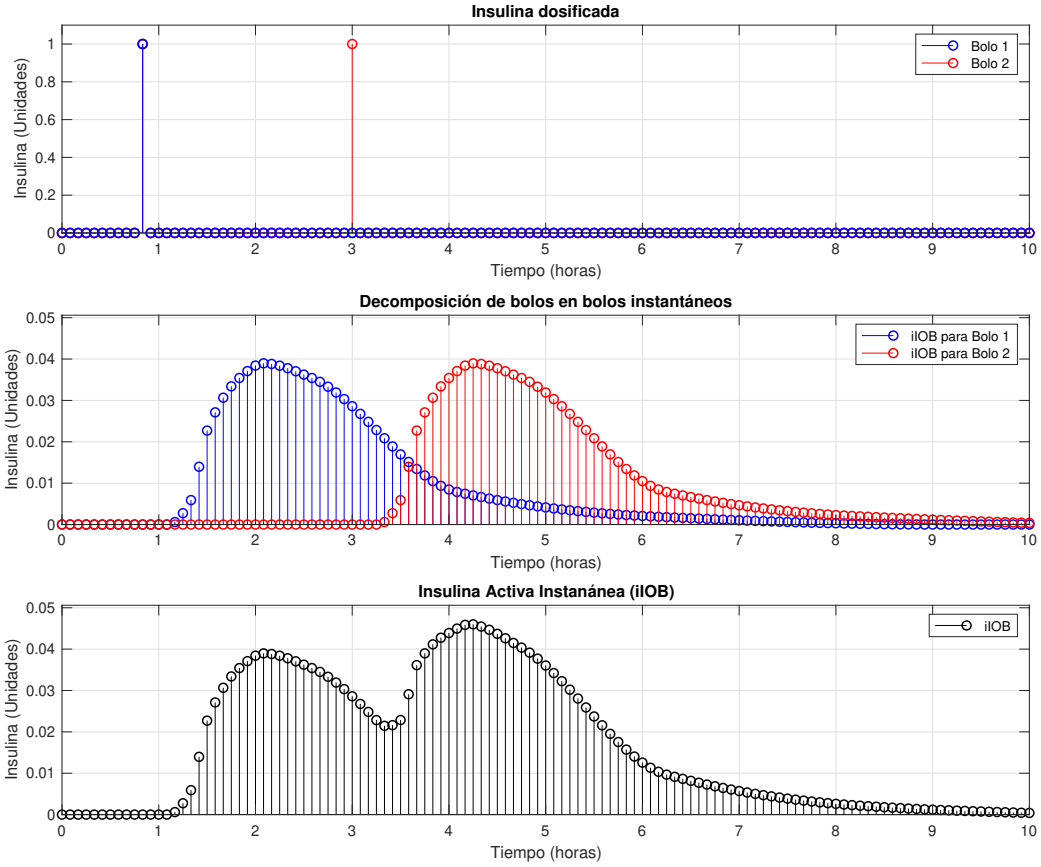


Figura 5.3: Ejemplo cálculo de iIOB con dos bolos de insulina.

La Figura 5.3 muestra un ejemplo de cómo dos bolos de insulina, representados en color azul y rojo respectivamente, son descompuestos en bolos instantáneos y, después, todos los efectos son agregados para dar lugar a las insulinas activas instantáneas (iIOB).

Es importante diferenciar entre Insulina Activa (IOB) e Insulina Activa Instantánea (iIOB): La insulina activa es la estimación de toda la insulina que todavía debe ser absorbida, una vez la insulina basal ha sido descontada (ver apartado 4.3.2). Por otro lado, la insulina activa instantánea es la estimación de la insulina que está causando efecto en un momento concreto debido a dosificaciones previas. La insulina activa también puede ser calculada como el sumatorio de todas las insulinas activas instantáneas desde el momento actual hacia el futuro sustrayendo la cantidad correspondiente a la insulina basal.

$$IOB[t] = \sum_{n=t}^{\infty}(iIOB[n] - Basal[n]) \tag{5.1}$$

La Ecuación 5.1 describe la relación entre *IOB* e *iIOB*. La *IOB* evaluada en un tiempo t es igual al sumatorio de todos los *iIOB* desde ese momento en adelante sustrayendo las necesidades basales (*Basal*). Es importante recordar que las necesidades basales deben ser convertidas de sus unidades típicas (U/h) de manera que el periodo de muestreo del sistema coincida.

5.3.2 Estimación de necesidades basales y el factor de sensibilidad a la insulina

Las necesidades basales fueron definidas en capítulos previos como la cantidad de insulina que necesita un paciente para mantener su glucemia estable en el tiempo. La estabilidad de la glucemia puede ser calculada como la primera derivada de los niveles de glucemia reportados por el MCG (*dSGV* - derivative of Sensor Glucose Value). Por otro lado, la insulina que está causando efecto por cada dato reportado por el MCG puede ser calculada como la iIOB utilizando el periodo de muestreo que corresponda a los datos del MCG (típicamente 5 minutos). Para que el cálculo de la iIOB sea correcto, es de vital importancia tener bien caracterizada la curva de acción de la insulina para el paciente, o lo que es lo mismo, es importante conocer el tipo de insulina utilizado por el paciente y el tiempo de acción de la misma (IAT).

De manera ideal, utilizando las últimas duplas dSGV-iIOB en una regresión lineal, se pueden derivar tanto las necesidades basales como el factor de sensibilidad a la insulina (*ISF*). Las necesidades basales serán equivalentes a la iIOB que anule la dSGV en la recta. Y el ISF deberá coincidir con la pendiente de la recta resultante de la regresión lineal.

En el ejemplo de la Figura 5.4, se ha representado las duplas dSGV-iIOB utilizando puntos rojos. Como es de esperar, a mayor iIOB más negativa la dSGV, significando que a más insulina activa instantánea mayor es la reducción de la glucemia del paciente. Mediante la regresión lineal (línea morada en la Figura 5.4), se puede calcular la cantidad

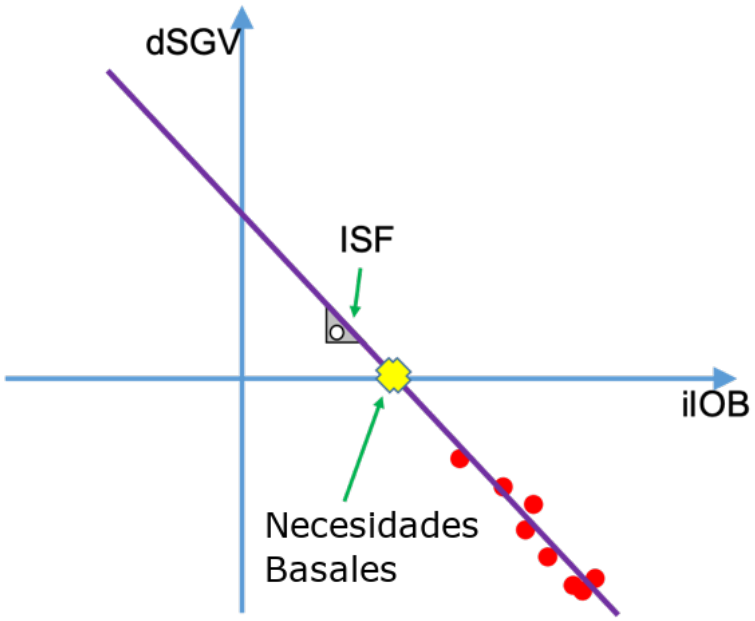


Figura 5.4: Ejemplo. Relación entre dSGV e iIOB

de insulina instantánea que hace cero la dSGV (cruz amarilla en la Figura 5.4). Al obtener una dSGV igual a cero podemos decir que la glucemia del paciente es estable y, por tanto, esa será la cantidad de iIOB correspondiente a las necesidades basales del paciente en ese momento.

El ISF o factor de sensibilidad a la insulina puede ser calculado como la pendiente de la linea resultantes de la regresión lineal multiplicada por menos uno.

$$dSGV = -ISF \times (iIOB - Basal) \tag{5.2}$$

$$ISF = -1 \times \frac{N \times \sum_{i=0}^{N-1}(dSGV_i \times iIOB_i) - \sum_{i=0}^{N-1} dSGV_i \times \sum_{i=0}^{N-1} iIOB_i}{N \times \sum_{i=0}^{N-1}(iIOB_i)^2 - (\sum_{i=0}^{N-1} iIOB_i)^2} \tag{5.3}$$

$$Basal = \frac{\sum_{i=0}^{N-1} dSGV_i - (-ISF) \times \sum_{i=0}^{N-1} iIOB_i}{N \times ISF} \tag{5.4}$$

Las Ecuaciones 5.2, 5.3 y 5.4 describen el procedimiento matemático para calcular las necesidades basales y el ISF utilizando el método de regresión lineal por mínimos cuadrados, siendo N el número de duplas o punto utilizados en el algoritmo de regresión.

5.3.3 Problemas derivados de la regresión lineal

El problema aparece cuando las necesidades basales cambian de manera abrupta: haciendo uso de la tecnología disponible en la actualidad, el algoritmo puede obtener un nuevo dato del MCG (SGV) cada 5 minutos. Para que la regresión lineal ofrezca resultados

correctos, el algoritmo necesita múltiples puntos o duplas, entre otros motivos debido al ruido introducido por el MCG en sus medidas. Por otro lado, cuanto mayor sea el número de puntos utilizado en la regresión menor será la velocidad con la que se podrán detectar cambios en las necesidades basales.

Si no se utilizan suficientes duplas, y/o las que usamos están muy próximas entre si, la regresión lineal podría arrojar resultados erráticos. Si el número de puntos o duplas utilizado es pequeño, podría ser difícil detectar violaciones de las asunciones de la regresión lineal [80]. Efectos como la heterocedasticidad de varianzas o la no-normalidad pueden ser difíciles de detectar. Y, lo que es incluso más importante, es difícil saber si los resultados en ese caso coinciden con las necesidades reales del paciente.

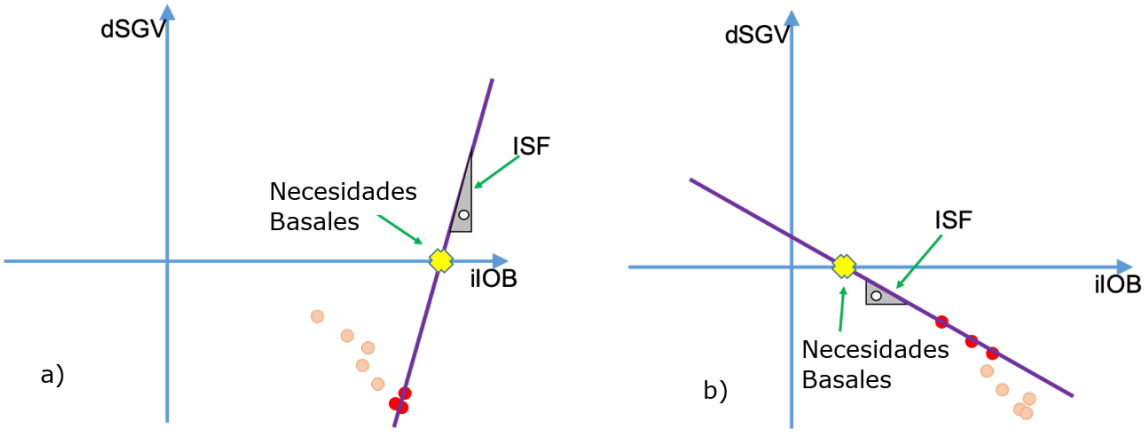


Figura 5.5: Ejemplo de resultados erróneos debidos a número insuficiente de duplas en la regresión lineal. a) Usando subconjunto 1 de datos, b) Usando subconjunto 2 de datos

Como se puede ver en los ejemplos de las Figuras 5.5-a y 5.5-b, dependiendo de las duplas utilizadas en la regresión lineal, los resultados pueden variar de manera drástica hasta el punto de no tener sentido. En la Figura 5.5-a, los resultados de la regresión lineal indicarían un valor mucho más alto de necesidades basales que las reales y, lo que es todavía más inquietante, un ISF sin lógica: el cambio en la pendiente de la recta sugiere que la glucemia del paciente se vería incrementada con un incremento en la insulina activa instantánea, lo cual es obviamente incorrecto. En la Figura 5.5-b, el subconjunto de duplas elegido nos llevaría a determinar unas necesidades basales muy inferiores a las que el paciente realmente necesita y, por consiguiente, a un empeoramiento del control.

5.3.4 Mitigación de problemas mediante sensibilidad prefijada

Como se ha venido repitiendo a lo largo de esta tesis, existen diversos factores que alteran en gran medida las necesidades basales de insulina de una persona. Entre estos factores encontramos las ingestas de hidratos de carbono, el ejercicio físico, el estado físico del set de infusión y la posible degradación de la insulina. Pero el ISF o factor de sensibilidad a la insulina se ve afectado en mucha menor medida por estos factores: las guías para

ajustes de los calculadores de bolos incluso las consideran constantes basadas en la dosis total diaria de insulina del paciente[81]. Para evitar situaciones parecidas a las mostradas en las Figuras 5.5-a y 5.5-b, el algoritmo propuesto fija el ISF para el paciente y, junto con ello, la pendiente de la linea resultante en el proceso de regresión. El ISF se obtiene de un conjunto de ISFs predefinidos para el paciente en forma de tabla, permitiendo tener un ISF para distintos tramos horarios dentro del mismo día. El cálculo de estos factores se realiza utilizando métodos clásicos fuera del alcance de esta tesis [82].

Utilizando factores de sensibilidad a la insulina predefinidos, las estimaciones de necesidades basales todavía siguen afectadas por el ruido del MCG, pero los efectos son minimizados considerablemente.

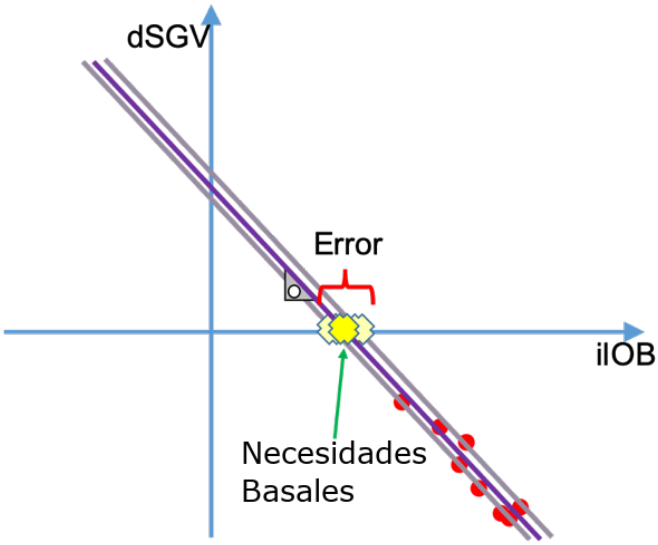


Figura 5.6: Ejemplo de estimación de necesidades basales utilizando ISF prefijado

La Figura 5.6 muestra un ejemplo de cómo, fijando el factor de sensibilidad a la insulina, el error se reduce y podemos tomar el compromiso de utilizar menos duplas para calcular las necesidades basales de insulina. Esta simplificación permite al algoritmo usar menos puntos/duplas y, con ello, ganar en velocidad de respuesta frente a cambios drásticos en las necesidades del paciente.

$$eBasal = \frac{\sum_{i=0}^{N-1} dSGV_i - (-ISF_{predefinido}) \times \sum_{i=0}^{N-1} iIOB_i}{N \times ISF_{predefinido}} \tag{5.5}$$

La Ecuación 5.5 describe el método matemático para calcular las necesidades basales estimadas ($eBasal$) usando un ISF predefinido por el paciente ($ISF_{predefinido}$).

5.3.5 Extrapolación de necesidades basales actuales

El sistema de control propuesto en esta tesis cae dentro de la categoría de controlador de lazo cerrado híbrido de única hormona. Esto significa que el controlador sólo actúa sobre

la infusión de una hormona y, en este caso, es insulina. Existen otros sistemas llamados de doble-hormona que también controlan la infusión de una segunda hormona, típicamente glucagón. Glucagón es una hormona cuyo efecto es contrario a la insulina elevando los niveles de glucosa en sangre de manera muy rápida y permitiendo al controlador actuar en ambos sentidos, disminuyendo o aumentando la glucemia como sea necesario.

Los datos obtenidos del MCG sufren un retardo frente a los valores de glucemia en sangre y este retardo puede variar entre 10 y 20 minutos dependiendo de diversos factores [74]. Cualquier estimación realizada sobre estos datos llevará implícitamente este mismo retardo y, si no se tiene en cuenta, en situaciones en las que las necesidades basales estén disminuyendo rápidamente el controlador podría dosificar más insulina de la necesaria, pudiendo llevar al paciente a sufrir una hipoglucemia.

Para compensar este problema, se analiza la tendencia en la estimación de necesidades basales con cada iteración del lazo de control. Al utilizar únicamente insulina para el control en el sistema, la estimación de necesidades basales debe ser conservadora: si las necesidades basales están incrementando con el tiempo, la tendencia no será tenida en cuenta pero, si están decreciendo, la tendencia será incorporada al cálculo de las necesidades para reducir la insulina dosificada al paciente lo antes posible. De esta manera, el algoritmo de control siempre mantendrá las necesidades basales lo más bajas y seguras posible.

$$tendencia_{eBasal} = \overline{\left(\frac{\sum_{i=-(N-1)}^{0} (eBasal(i) - eBasal(i-1))}{N} \right)} \tag{5.6}$$

La ecuación 5.6 muestra cómo se calcula la tendencia de las necesidades basales estimadas. La primera derivada de las necesidades basales determina cómo estas se ven modificadas en el tiempo. Haciendo uso únicamente del último punto de la derivada se obtendría la respuesta más rápida frente al cambio pero, como los datos recibidos del sensor podrían verse afectados por el ruido y la curva de acción de la insulina podría estar ligeramente mal configurada, se obtendrían grandes fluctuaciones en el resultado e impactaría negativamente en el control. Afortunadamente, las necesidades basales no cambian rápidamente en comparación con el periodo de muestreo del MCG, y el periodo de muestreo es suficientemente corto como para permitir cierto nivel de filtrado. Una simple media de los últimos 15 minutos de la primera derivada muestra buenos resultados balanceando tiempo y reducción de ruido

$$eBasal_{final}(eBasal, tendencia_{eBasal}) = \begin{cases} eBasal + (tendencia_{eBasal} \times \frac{15}{samplingPeriod}) & : \ tendendia_{eBasal} < 0 \\ eBasal & : \ tendencia_{eBasal} \geqslant 0 \end{cases} \tag{5.7}$$

Las necesidades basales finales (ver Ecuación 5.7) se calculan utilizando la estimación de necesidades basales y su tendencia. Aunque podría parecer razonable compensar siempre

el retardo del MCG en la estimación de necesidades basales, incrementar las necesidades basales iría en contra de asegurar la seguridad del paciente; la insulina no puede ser eliminada del cuerpo del paciente una vez se ha dosificado y siempre existe la opción de dosificar más insulina en el futuro, si es necesaria, cuando el incremento en necesidades basales se haya confirmado. El mismo razonamiento nos lleva a actuar de manera opuesta cuando se detecta un decremento de las necesidades basales: el algoritmo de control puede reducir la infusión de insulina tanto como la predicción sugiera, reduciendo el riesgo de hipoglucemia, porque esta insulina podría ser dosificada posteriormente si se confirma que la tendencia era incorrecta. Como resultado, si la tendencia es igual o mayor que cero, la estimación de necesidades basales no se ve modificada y se convierte directamente en su versión final. Si la tendencia es menor que cero, significando que el paciente necesita menos insulina a medida que el tiempo pasa, las necesidades basales se reducen para acomodarse a esta tendencia, calculando mediante extrapolación lineal el decremento esperado en los siguientes 15 minutos (siendo este el tiempo de retardo medio en sistemas MCG actuales) e incorporando el efecto a la estimación de necesidades basales.

5.3.6 Ajuste dinámico del factor de sensibilidad a la insulina

Es también conocido que el factor de sensibilidad a la insulina (ISF) decrece cuando la glucemia del paciente aumenta [67]. Para contrarrestar este efecto, el algoritmo propuesto hace uso de los factores de sensibilidad a la insulina configurados como aquellos factores a utilizar cuando la glucemia del paciente es igual a la glucemia objetivo ($ISF_{objetivo}$). A medida que la glucemia del paciente aumenta por encima del objetivo, el factor de sensibilidad a la insulina se ve linealmente decrementado de acuerdo a un ratio predefinido ($ratio_{ISF}$). Por otro lado, cuando la glucemia decrece por debajo del objetivo, el algoritmo también reducirá, quizá contra-intuitivamente, el factor de sensibilidad a la insulina. Tal y como se puede apreciar en la Ecuación 5.8, estas modificaciones del ISF son equivalentes a decir que la estimación de ISF ($ISF_{estimado}$) es igual al ISF configurado para cuando el paciente se encuentra en su objetivo glucémico menos el valor absoluto del error glucémico ($|error_{glucémico}|$) multiplicado por el ratio de corrección del ISF ($ratio_{ISF}$). El error glucémico, como se ha explicado con anterioridad, resulta de restar el objetivo glucémico de la glucemia final del paciente.

$$\begin{aligned} ISF_{estimado} = ISF_{objetivo} - ratio_{ISF} \times |error_{glucémico}| = \\ ISF_{objetivo} - ratio_{ISF} \times |sgv_{actual} - sgv_{objetivo}| \end{aligned} \tag{5.8}$$

Este ajuste es opcional y puede ser deshabilitado configurando el ratio de corrección del ISF a cero. El valor mínimo de ISF (ISF_{min}) también debe ser ajustado para evitar resultados de ISF extremos que pudieran conducir al situaciones indeseadas.

$$ISF_{final}(ISF_{estimado}, ISF_{min}) = \begin{cases} ISF_{min} & : ISF_{estimado} < ISF_{min} \\ ISF_{estimado} & : ISF_{estimado} \geqslant ISF_{min} \end{cases} \tag{5.9}$$

La Ecuación 5.9 describe la manera en que el ISF estimado (*ISFestimated*) se ve limitado por el ISF mínimo (*ISFmin*) obteniendo como resultado el ISF final (*ISFfinal*) que será utilizado en los cálculos del lazo de control.

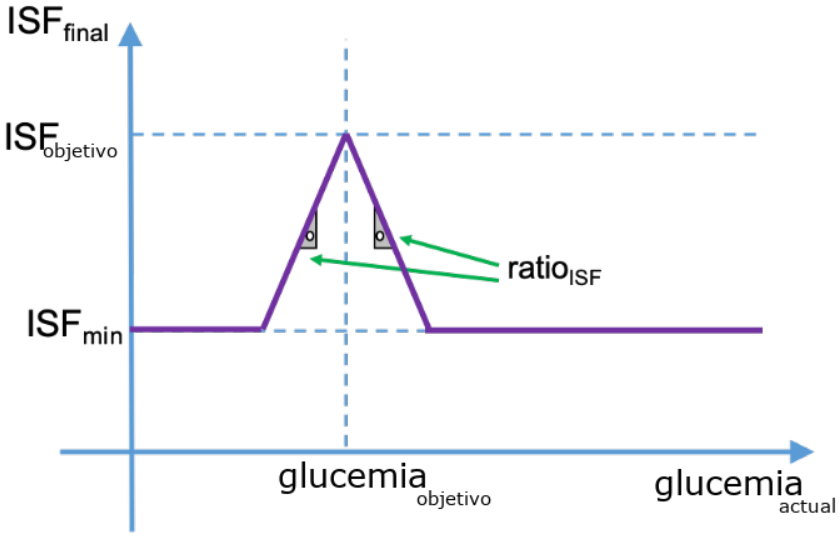


Figura 5.7: Representación gráfica del cálculo de ISF_{final}

La Figura 5.7 muestra, a modo de clarificación, una representación gráfica de la relación entre la glucemia final el paciente y el ISF_{final}.

Por ejemplo, si la glucemia de un paciente ($glucemia_{actual}$) se encuentra en su objetivo glucémico($glucemia_{objetivo}$) el factor de sensibilidad a la insulina resultante será el definido para el objetivo ($ISF_{objetivo}$). Este valor de factor de sensibilidad a la insulina será el máximo posible, obteniendo así las correcciones menos agresivas posibles. A medida que la glucemia del paciente ($glucemia_{actual}$) se aleja del objetivo glucémico, el factor de sensibilidad a la insulina decrece linealmente, siguiendo el ratio configurado para ello ($ratio_{ISF}$), hasta alcanzar el factor de sensibilidad a la insulina configurado como mínimo (ISF_{min}). El factor de sensibilidad a la insulina mínimo es el factor que proporciona la mayor agresividad en las correcciones.

Parece lógico que a medida que la glucemia del paciente aumenta por encima del objetivo las necesidades de insulina para la corrección son mayores, pero el caso contrario puede no ser tan evidente. El reducir el factor de sensibilidad a la insulina cuando la glucemia se encuentra por debajo del objetivo response a una necesidad de seguridad médica: las correcciones a ejecutar cuando la glucemia se encuentra por debajo del objetivo equivalen a reducir la insulina administrada al paciente para intentar incrementar la glucemia hasta el objetivo. Para asegurarnos de que se evitan hipoglucemias, se disminuye el factor

de sensibilidad para reducir aún más la cantidad de insulina administrada (corrección más negativa), obteniendo un comportamiento mucho más conservador en cuanto a la dosificación de insulina.

5.3.7 Efectos causados sobre la estimación de las necesidades basales debidas a parámetros erróneos

La curva de acción de la insulina, definida por la curva de absorción ofrecida por el fabricante de la misma y el ajuste proporcionado por el paciente mediante su duración (IAT), se convierte en una parte crítica de la estimación de necesidades basales. Si la curva de acción de insulina programada no concuerda con la respuesta real de la insulina en el cuerpo del paciente, debido a la forma más o menos genérica de las curvas de acción, se producen oscilaciones en las estimaciones y, con ellas, oscilaciones en las glucemias del paciente. El desajuste entre curvas de acción (configurada y real) crea áreas en las que el algoritmo considera que tiene una $iIOB$ superior a la real, resultado en estimaciones de necesidades basales superiores a las reales, y áreas con $iIOB$ más bajo del real, dando como resultado estimaciones de necesidades basales inferiores a las reales. Este efecto puede ser apreciado en el ejemplo de la Figura 5.8. Estas subidas y bajadas causan un efecto de rizado en la estimación de necesidades basales y, por consiguiente, también sobre las actuaciones del algoritmo de control y en la glucemia del paciente.

La Figura 5.8 muestra un ejemplo en el que la respuesta real del paciente coincide con una IAT de 4 horas pero el lazo de control está operando con una curva de IAT de 3 horas. En la gráfica con las dos curvas superpuestas se ha representado con círculos negros la curva para una IAT de 3 horas y con cruces azules la curva para una IAT de 4 horas. La zona marcada por las líneas azules corresponde con la iIOB común entre ambas curvas. El área definida por las líneas rojas define la parte de iIOB que se estaría estimado en exceso a causa de usar una IAT inferior a la real. El área definido por las líneas verdes determina la cantidad de iIOB que no se está teniendo en cuenta a causa de que se incluyó previamente, y por error, en el área de líneas rojas. La zona de líneas verdes se traduce como un error negativo en la estimación de iIOB. La gráfica inferior en la Figura 5.8 representa el error obtenido en la estimación de iIOB de una unidad de insulina debido a utilizar una IAT de 3 horas en vez de 4 horas.

5.3.8 Objetivo glucémico dinámico y umbral de suspensión por hipoglucemia

El último parámetro que se ajusta dinámicamente es el objetivo glucémico. El paciente fija el objetivo glucémico como un rango considerado como aceptable y definido por sus valores máximo y mínimo. Este algoritmo calcula el objetivo glucémico deseado como el punto medio de este rango definido por el paciente, siendo entonces este punto el valor de glucemia a conseguir mientras el paciente se encuentre dentro del rango objetivo (ver

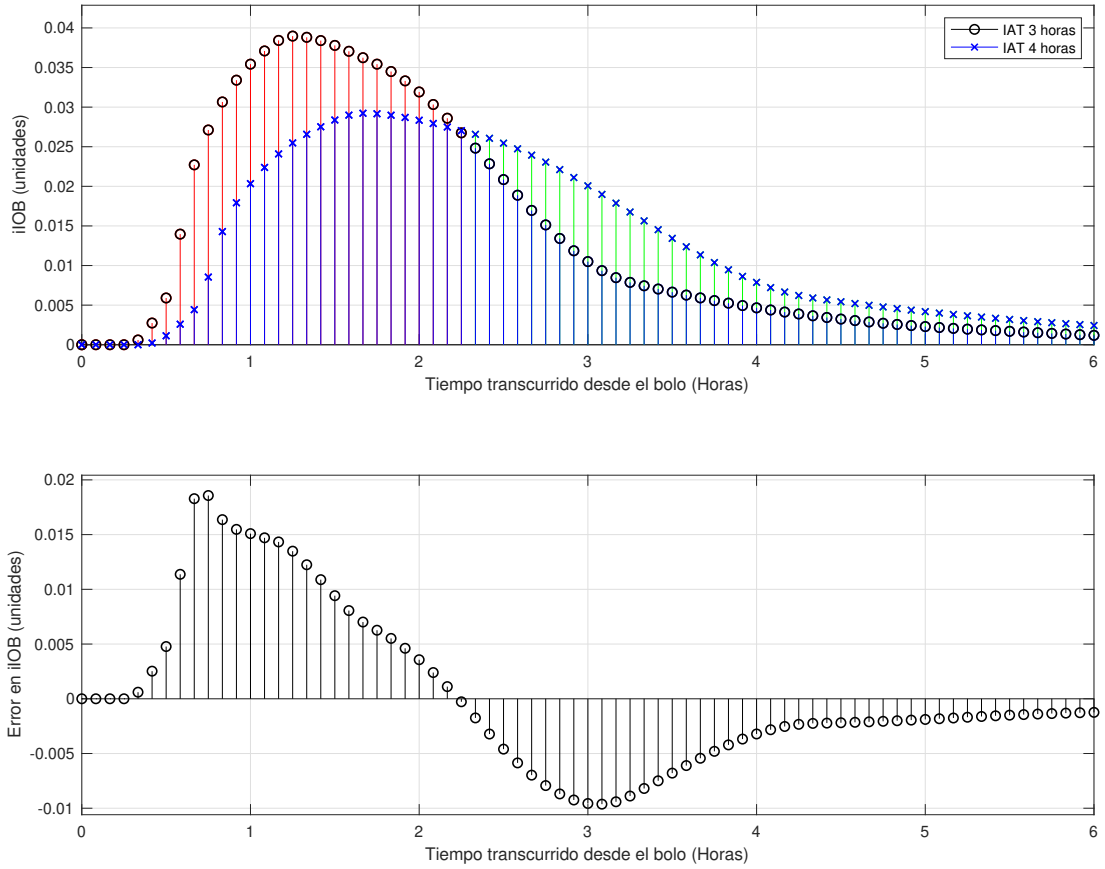


Figura 5.8: Ejemplo de error en la estimación de iIOB debido a IAT incorrecta

Ecuación 5.10).

Como ya se ha explicado en apartados anteriores (ver apartado 4.3.10), el valor mínimo del rango se utiliza también como nivel de suspensión por hipoglucemia. Esto significa que, si la glucemia del paciente se encuentra o se prevee que esté por debajo de este umbral, se suspenderá la infusión de insulina independientemente a que el resto de parámetros indiquen que la glucemia del paciente va a subir. El algoritmo también implementa un flag (LTS_{flag} - *Low-Threshold-Suspend Flag*) que es activado en cuanto actúa la protección. Una vez la glucemia del paciente se encuentra fuera de riesgo de hipoglucemia, el LTS_{flag} se mantendrá activo durante otros 15 minutos. Mientras el flag esté activo, el objetivo glucémico es modificado a uno más alto (el valor máximo del rango objetivo) con la intención de reducir la agresividad del lazo de control tras salir de la hipoglucemia. Se permitirá al lazo tender a una glucemia más alta, con lo que hará falta menos insulina y, transcurrido el tiempo de seguridad, podemos volver al objetivo deseado y recuperar la insulina no dosificada que sea necesaria.

$$objetivo_{deseado} = \frac{objetivo_{max} + objetivo_{min}}{2} \tag{5.10}$$

Durante la operación normal del sistema, el algoritmo calcula el objetivo glucémico dinámico (ver Ecuación 5.11) como el punto medio entre la glucemia final del paciente y el valor mínimo del rango objetivo.

$$objetivo_{dinámico} = \frac{sgv + objetivo_{min}}{2} \tag{5.11}$$

El objetivo glucémico final se encuentra limitado y sólo puede tomar valores dentro del rango objetivo fijado por el paciente: el objetivo glucémico final nunca será superior al valor máximo del rango objetivo o menor que el objetivo deseado (punto medio del rango objetivo). El flag LTS_{flag} actúa como un modificador temporal que fuerza el objetivo glucémico final a ser el valor máximo del rango objetivo. Este comportamiento se describe en la Ecuación 5.12.

$$objetivo_{final}(objetivo_{dinámico}, objetivo_{deseado}, objetivo_{max}, LTS_{flag}) =$$
$$\begin{cases} objetivo_{max} & : LTS_{flag} = Activo \\ objetivo_{deseado} & : LTS_{flag} = Inactivo, objetivo_{dinámico} \leq objetivo_{deseado} \\ objetivo_{dinámico} & : LTS_{flag} = Inactivo, objetivo_{deseado} < objetivo_{dinámico} \leq objetivo_{max} \\ objetivo_{max} & : LTS_{flag} = Inactivo, objetivo_{max} < objetivo_{dinámico} \end{cases} \tag{5.12}$$

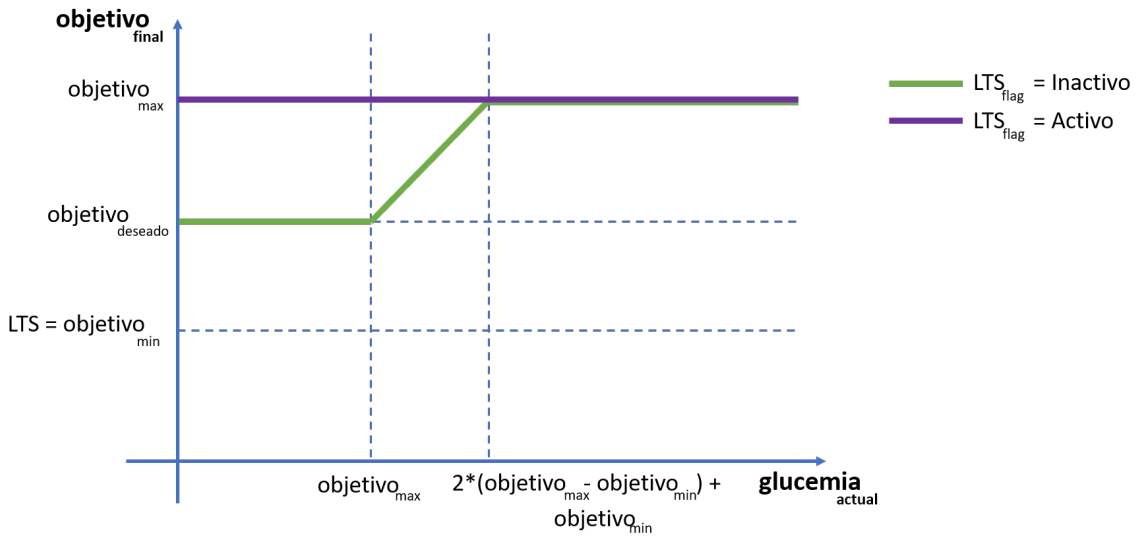


Figura 5.9: Relación entre el $objetivo_{final}$, el flag LTS_{flag} y el valor de $glucemia_{final}$

La Figura 5.9 es una representación gráfica de la relación existente entre el valor de la glucemia final del paciente, el flag *LTSFlag* y el objetivo glucémico final resultante.

A modo de ejemplo, pensemos en el caso de un paciente que se encuentra en hipoglucemia. En este caso el indicador LTS_{flag} se encontrará activo y, por tanto, el controlador tendrá como objetivo glucémico ($objetivo_{final}$) el máximo valor del rango definido ($objetivo_{max}$). En esta situación el controlador mantendrá suspendida la infusión hasta

que la condición de hipoglucemia desaparezca pero, en el momento que esto ocurra, el controlador utilizará este objetivo durante los primeros minutos de ejecución. Este mecanismo permite recuperarse de la hipoglucemia con seguridad y hacer que el controlador sea menos agresivo a la hora de dosificar insulina. Cuando el tiempo programado pasa y el indicador LTS_{flag} se desactiva, se procede al cálculo del objetivo en función de la glucemia actual del paciente ($glucemia_{actual}$). El cálculo es sencillo:

1. Si la glucemia del paciente se encuentra dentro del rango, el objetivo siempre será el punto medio del rango definido como objetivo (punto medio entre $objetivo_{min}$ y $objetivo_{max}$).

2. Si la glucemia supera el valor máximo del rango objetivo ($objetivo_{max}$), el objetivo glucémico pasa a ser el punto medio entre la glucemia actual del paciente y el valor mínimo del objetivo. Esto provoca que el objetivo crezca de manera proporcional a la glucemia actual del paciente

3. Si el objetivo calculado supera el valor definido como máximo ($objetivo_{max}$), el objetivo resultante será dicho valor máximo.

Esta forma de modificar el objetivo glucémico permite al paciente configurar el sistema de manera que las correcciones hasta llegar al objetivo deseado sean más lentas en el tiempo y, a la vez, minimicen la probabilidad de hipoglucemia causadas por una sobreactuación del controlador.

5.4 Inclusión de la estimación en el algoritmo básico de control

La estimación de las necesidades basales de insulina del paciente debe ser parte del algoritmo de control y, al igual que sucede con el resto de parámetros del controlador, debe ser actualizado con cada dato recibido o con cada ejecución del lazo de control. Gracias a esta forma de operar, el controlador obtiene estimaciones de las necesidades basales de insulina del paciente de manera dinámica.

El algoritmo básico hace uso de las necesidades basales al comienzo de su ejecución para realizar los cálculos de insulina activa (ver apartados 4.3.2, 4.3.4 y 4.3.5). Esto fuerza a que la estimación de necesidades basales sea calculada al principio de la ejecución del lazo de control y con cada nuevo dato del sensor (aproximadamente 5 minutos). Como los datos de entrada para la estimación son las historias de datos del MCG y de dosificación de insulina, y son conocidos al comienzo de la ejecución, puede realizarse el cálculo al comienzo de la ejecución sin problema alguno.

Se puede visualizar el punto donde se incluye la estimación de necesidades basales en la figura 5.1 marcado en negrita.

5.5 Conclusiones

Esta modificación al algoritmo básico de control determina de manera automática la cantidad de insulina que necesita el paciente para mantener su glucemia estable (necesidades basales de insulina) llevando la glucemia del paciente a su glucemia objetivo mediante objetivos dinámicos. Este nuevo algoritmo ayuda al lazo de control ajustando dinámicamente tanto las necesidades basales como el factor de sensibilidad a la insulina y el objetivo glucémico que, en el algoritmo básico de control, estos parámetros eran prefijados por el paciente o su médico.

Este algoritmo se adapta a cambios en las necesidades basales de insulina producidos por factores como enfermedades o alteraciones hormonales pero también al ejercicio físico y a las ingestas, los cuales son eventos mucho más frecuentes en la vida del paciente. Es importante remarcar que, debido a los retardos inherentes al sistema (ver figura 3.4), sigue siendo importante anunciar al controlador cualquier cambio brusco para que el control glucémico sea seguro. Entre estos cambios bruscos podemos encontrar el comienzo repentino de ejercicio físico intenso o la ingesta de hidratos de absorción rápida. Por otro lado, este algoritmo debería ser capaz de responder apropiadamente, y sin intervención del paciente, ante ejercicio físico ligero o ingestas de alimentos con absorción lenta. En sistemas que carecen de estas características el ajuste de la terapia para estos casos es manual y complicado y se suele realizar mediante modificaciones temporales de los objetivos glucémicos o la programación de basales temporales calculadas por el paciente.

Es importante remarcar que los beneficios no sólo son puramente glucémicos: el que los ya explicados parámetros se ajusten de manera automática y transparente al usuario del sistema permite reducir la carga mental y devolver un poco de normalidad a la vida del paciente. Además, situaciones complicadas como son la gestión de ingestas con alto contenido en grasas o situaciones necesidades insulínicas incrementadas a causas de otras enfermedades, quedan muy aliviadas por la determinación automática y dinámica de parámetros y por no ser necesario esperar días para poder hacer un análisis retrospectivo de los datos.

Las diferencias que existen entre sistemas con estimación de necesidades basales de insulina y sin ellas se pueden resumir en:

1. En un sistema SIN estimación de necesidades basales, es tarea del paciente o su médico el correcto ajuste del perfil basal (que típicamente ocurre cada varios meses). Cualquier alteración de las necesidades basales o un mal ajuste se traducen directamente en un deterioro del control glucémico del paciente. En un sistema CON estimación de necesidades basales el perfil se determina de manera automática y transparente para el paciente, estando siempre adaptado a las necesidades en todo momento.

2. En un sistema SIN estimación de necesidades basales, el paciente está encargado de

modificar su perfil basal ante alteraciones como enfermedades o estrés. Es por ello que se suele disponer de más de un perfil basal programado para poder ir alternando entre ellos según sea necesario y esto obliga al paciente/médico a mantener correctamente configurados todos los perfiles. La toma de datos de situaciones de enfermedad o estrés también es un proceso realizado por el paciente y del que dependerá la calidad del perfil basal programado para dicha situación. En un sistema CON estimación de necesidades basales este proceso no es necesario y se libera al paciente de toda su carga asociada.

3. En un sistema SIN estimación de necesidades basales, la ingesta de grasas o alimentos de largo tiempo de absorción debe ser tenida en cuenta a la hora de programar los bolos de insulina para dichas ingestas. Es común tener que programar bolos cuadrados (aquellos bolos de insulina que se dosifican durante un periodo de tiempo largo) para asegurarse de que la insulina hace efecto de manera más retardada y coincidente con el efecto de dichos alimentos. En un sistema CON estimación de necesidades basales los efectos causados por estos alimentos de largo tiempo de absorción se asimilan a un periodo de necesidades basales incrementadas y, por tanto, el sistema se adapta a ellos de manera automática y sin necesidad de intervención por parte del paciente. En un sistema CON estimación de necesidades basales únicamente se dosifica como bolo normal la insulina necesaria para los hidratos de carbono de absorción rápida.

4. En un sistema SIN estimación de necesidades basales es tarea del paciente detectar situaciones en las que el punto de infusión de insulina o el set de infusión en si mismo empiezan a estar degradados, requiriendo más insulina como resultado. En un sistema CON estimación de necesidades basales se asimila este incremento de necesidades basales debidos a la mala infusión con necesidades reales del cuerpo. Sigue siendo importante mantener el buen estado del sitio de infusión por parte del paciente, pero el algoritmo se adaptará a esta situación e intentará mantener el buen control glucémico hasta que la degradación sea excesiva y con ello sea imposible mantener el control.

En resumen, este capítulo desarrolla un algoritmo de estimación de necesidades basales de insulina que puede ser incorporado a multitud de sistemas de control, incluido el desarrollado en el capítulo 4. En el próximo capítulo (Capítulo 6) se validará el proceso de estimación de las necesidades basales de insulina y se evaluarán las mejoras aquí estimadas.

Capítulo 6

Resultados

No basta tener un buen ingenio, lo principal es aplicarlo bien.

René Descartes

6.1 Introducción

En este capítulo se hace un análisis de cuales son los resultados esperables del sistema mediante simulaciones, utilizando el simulador *simglucose* [83], y mediante los resultados obtenidos en la vida real por un paciente (el autor de la tesis). El realizar ensayos clínicos se encuentra fuera del ámbito de esta tesis doctoral.

Primeramente se explicará el entorno de pruebas y simulación utilizado. En esta tesis se desarrolla un sistema completo de control y es necesario tener una estrategia clara de simulación y validación de cada parte.

Posteriormente se explicarán los lotes de simulaciones realizadas sobre los algoritmos, en qué consisten, su orden y cuáles son sus objetivos y sus resultados.

Haciendo uso del simulador y del código a ejecutar en la plataforma *wearable*, se ejecutan ambos bajo el mismo entorno para ver si arrojan los mismos resultados y el código embebido es confiable.

Finalmente, el sistema real se pone en funcionamiento en un ejercicio de auto-experimentación y se documentan los resultados.

6.2 Entorno de simulación

El primer paso en el proceso de validación es encontrar el entorno apropiado para llevarlo a cabo. En cuanto a la simulación del algoritmo existían varias opciones para probar el flujo del algoritmo en un paciente sintético:

1. Crear un simulador propio
2. Usar un simulador existente
   (a) Usar el simulador de UVA/PADOVA
   (b) Usar el simulador simglucose

La opción de crear un simulador propio fue descartada rápidamente, pues era necesario tener un simulador validado cuyos resultados fueran confiables por la comunidad científica. Un simulador propio quizás permitiría adaptar de una mejor manera los escenarios y modelos a los casos que se desean mejorar con el algoritmo propuesto, pero sería un esfuerzo no despreciable que en principio ha quedado fuera del desarrollo de la presente tesis doctoral al no ser un objetivo perseguido en la misma, aunque puede ser abordado como un trabajo futuro.

En cuanto a los simuladores existentes, el simulador diseñado por las Universidades UVA/PADOVA [84] está aprobado por la FDA y es el estándar de facto, permitiendo al investigador incluso saltarse la fase de experimentación animal (en caso de éxito con las simulaciones). Funciona sobre Matlab, lo cual le añade grandes capacidades analíticas y de cómputo, sus modelos de simulación son probablemente los más complejos y avanzados, pero tiene un coste elevado, exigiendo una licencia para su uso. Incluye una base de datos de modelos de pacientes con 300 pacientes virtuales (100 adultos, 100 niños y 100 adolescentes).

Por otro lado, existe un simulador Open Source llamado simglucose [83]. Este simulador implementa una versión anterior (2008) y algo más sencilla de los modelos del simulador UVA/PADOVA y está limitado a un menor número de pacientes simulables. En el caso de simglucose, el simulador incluye 30 pacientes virtuales: 10 adolescentes, 10 adultos y 10 niños. El entorno software utilizado para este simulador es Python y ofrece también grandes capacidades de cómputo e interacción con otros entornos externos.

Simglucose ofrece las siguientes ventajas:

- Utiliza los mismos modelos validados de paciente que los desarrollados en el simulador de UVA/PADOVA.
- Es una plataforma de código abierto que permite su modificación para adecuarlo a las necesidades concretas de cada simulación. En el caso de esta tesis, esta característica permite:
  - Simular bloqueos parciales de la infusión de insulina como equivalente a simular condiciones en las que se alteran las necesidades basales de insulina del paciente (desarrollado en apartados posteriores).
  - Utilizar el simulador para validar el código implementado para la plataforma *wearable*. Se comprueba que el simulador y el código desarrollado para la plata-

forma *wearable* responden de manera equivalente ante los mismos escenarios de simulación.

- No requiere de ninguna licencia para su uso

Como desventaja de simglucose frente al simulador UVA/PADOVA se encuentra que el número de pacientes que pueden ser simulados es inferior (30 pacientes frente a 300). A pesar del menor número de pacientes simulados en simglucose, los tres grupos principales de pacientes (niños, adolescentes y adultos) se encuentran representados en los 30 modelos de paciente, permitiendo igualmente obtener una idea clara de los resultados obtenibles con el algoritmo.

Dada esta lista de ventajas y desventajas, se decide utilizar el simulador simglucose para llevar a cabo los ejercicios de simulación.

6.2.1 Fases de pruebas

Por todo lo anterior, en esta tesis doctoral se hará uso del simulador en Python simglucose. Una vez validado el algoritmo, es necesario validar que la implementación en C que se ejecuta en la plataforma *wearable* responde con las mismas actuaciones ante los mismos datos de entrada y, por tanto, podemos suponer que el algoritmo teórico y su implementación real son equivalentes. Para ello se dividen las pruebas sobre la plataforma *wearable* en varias fases que irán confiriendo fiabilidad al desarrollo final obtenido::

1. Puesta en marcha y ejecución real de la plataforma *wearable* con intención únicamente de realizar monitorización y comprobar la fiabilidad del software y hardware del sistema real. El algoritmo de control en lazo cerrado no se ejecuta.

2. Simulación del algoritmo en C para comprobar que las medidas de protección funcionan y que los cálculos realizados son correctos y ofrecen los mismos resultados que las simulaciones en Python. Para ello habrá de realizarse una serie de pruebas que llevan al sistema a tener que aplicar dichas medidas de protección.

3. Activación del algoritmo en lazo cerrado, con entradas de datos reales, pero sin permitir la actuación sobre la bomba de insulina. Las actuaciones únicamente se registran sus valores para su posterior análisis. A partir de esta fase se puede considerar como fiable la plataforma y conexión con los elementos componentes del páncreas artificial. En las siguientes fases se tienen las fases de pruebas de auto-experimentación.

4. Fase de auto-experimentación: Activación del algoritmo en lazo cerrado, con entradas de datos reales y permitiendo la actuación sobre la bomba de insulina. El ajuste de parámetros se hizo de manera conservadora y siempre bajo extrema vigilancia.

6.2.2 Simulador y modelo de controlador

Como ya se ha comentado en el apartado anterior, el simulador utilizado en esta tesis es simglucose [83]. Entre sus principales características se encuentran:

- Es una versión Open Source de los modelos de simulación diseñados por las Universidades UVA/PADOVA en 2008 para su simulador.
- Estos modelos están aprobados por la FDA.
- Dispone de 30 modelos de pacientes para simular: 10 adolescentes, 10 adultos y 10 niños.
- Está codificado en Python.
- Soporta computación en paralelo, por lo que es habitual preparar un lote de diferentes pacientes y simular una cierta situación de ingesta, control, etc.
- Los escenarios pueden ser aleatorios o predefinidos en el simulador.
- Proporciona un interfaz muy sencillo que permite implementación de controladores basados en PID, Model Predictive Control (MPC) o incluso Reinforcement Learning Control.
- Genera distintos tipos de gráficas para mostrar los resultados de todo lo lanzado en el lote de simulación de una manera cómoda. Estas gráficas incluyen:
  - Gráfico de evolución de glucemia en sangre
  - Rejilla de análisis de variabilidad en el control
  - Histograma de tiempo en rangos
  - Índices de riesgo de hipoglucemia e hiperglucemia

6.2.2.1 Generación de nuevo controlador

El simulador permite la creación de controladores pero estos deben ser implementados utilizando unos métodos predefinidos, los cuales permiten al núcleo del simulador llamar al controlador de una manera estándar.

A continuación, presentamos las principales cuestiones técnicas que nos permiten abordar el flujo de datos y control en tiempo real implementado en la tesis doctoral.

La Figura 6.1 muestra un ejemplo de clase controlador que se proporciona con el simulador y puede ser usada como esqueleto para programar un nuevo tipo de control, como es nuestro caso.

Deben ser implementados 3 métodos dentro de la clase (`init`, `policy` y `reset`):

```python
from simglucose.simulation.user_interface import simulate
from simglucose.controller.base import Controller, Action


class MyController(Controller):
    def __init__(self, init_state):
        self.init_state = init_state
        self.state = init_state

    def policy(self, observation, reward, done, **info):
        '''
        Every controller must have this implementation!
        ----
        Inputs:
        observation - a namedtuple defined in simglucose.simulation.env. For
                      now, it only has one entry: blood glucose level measured
                      by CGM sensor.
        reward      - current reward returned by environment
        done        - True, game over. False, game continues
        info        - additional information as key word arguments,
                      simglucose.simulation.env.T1DSimEnv returns patient_name
                      and sample_time
        ----
        Output:
        action - a namedtuple defined at the beginning of this file. The
                 controller action contains two entries: basal, bolus
        '''
        self.state = observation
        action = Action(basal=0, bolus=0)
        return action

    def reset(self):
        '''
        Reset the controller state to inital state, must be implemented
        '''
        self.state = self.init_state


ctrller = MyController(0)
simulate(controller=ctrller)
```

Figura 6.1: Ejemplo de clase tipo controlador en simglucose

- **Método `__init__`**: En este método se debe declarar e inicializar todas las variables internas que necesite el controlador (por ejemplo los ajustes del controlador para cada paciente, declaración de variables internas que deban permanecer entre ejecuciones del lazo de control, declaración de buffers para almacenar datos históricos del sensor y actuaciones previas, etc.). Este método es llamado únicamente una vez cuando se crea el objeto controlador al inicio de la simulación.

- **Método `reset`**: Este método debe permitir al simulador llevar al controlador a un estado inicial conocido. Es decir, se configuran los parámetros y valores de funcionamiento del controlador a unos valores por defecto, iniciales.

- **Método `policy`**: Este método es el encargado de procesar los eventos del sistema. Cada vez que exista un dato nuevo del MCG (también llamado CGM en inglés - *Continous Glucose Monitor*) se llamará a este método y los datos se pasarán mediante el parámetro `observation`. Otro parámetro que será de gran utilidad es `info`. Gracias a este parámetro el controlador puede conocer el nombre del paciente y modificar sus parámetros en función del paciente simulado, como por ejemplo el

factor de sensibilidad a la insulina, los ratios para calcular los bolos para las ingestas, etc. Esto permite tener al simulador configurar los parámetros necesarios para cada paciente en el controlador simulado de una manera cómoda.

Con cada nuevo dato del MCG, se ejecuta el método `policy` del controlador. Este proceso es el responsable de implementar el lazo de control siguiendo una filosofía *run-to-completion* [85]. Esto se traduce en que el proceso deberá mantener su histórico de datos para siguientes ejecuciones futuras de sí mismo cuando se reciban nuevos datos y, a su vez, tiene la opción con cada ejecución de comandar modificaciones a la infusión de insulina (si el algoritmo implementado lo considera necesario). La ejecución del proceso se lanza con la recepción de un nuevo dato y debe finalizar para permitir la continuación de la simulación: no se puede quedar el proceso esperando a ningún evento.

La modificación de la infusión de insulina en este simulador se realiza mediante lo que se denomina como "acciones". Esta acción se comunica devolviendo, desde el método de cálculo del lazo de control, un objeto tipo `Action` (definido por el simulador) que encapsula el perfil basal deseado a ser ejecutado por la bomba de insulina a partir de ese momento y la cantidad de insulina a dosificar como bolo si fuera necesario. Este objeto contiene dos variables: una determina la tasa basal a ser ejecutada por la bomba de insulina, siendo el mínimo valor 0, y la otra variable permite comandar bolos de insulina a la bomba si el algoritmo lo cree necesario. Si no es necesario comandar un bolo de insulina en ese ciclo de ejecución del lazo de control, bastará con configurar ese parámetro como 0.

Esta arquitectura que tiene la clase controlador para realizar su funcionamiento en el simulador concuerda exactamente con la forma en la que se ha preparado el *firmware* básico de control en la plataforma *wearable*: existe una función/método que es llamada con cada nuevo dato recibido del MCG y permite la ejecución del código del lazo de control y actuar (o no) sobre la cantidad de insulina a ser proporcionada por la bomba. En el sistema propuesto, la función `void controlLoopDataUpdate(void)` es equivalente al método `policy` del controlador en el simulador (ver apartado 3.6.1.7). Esto facilita en gran manera tanto la recodificación del algoritmo sobre la plataforma HW real (una vez validado en el simulador) como la ejecución de simulaciones en paralelo utilizando ambos códigos para realizar la validación cruzada.

6.2.2.2 Definición de simulación

Una vez se dispone de una implementación de un controlador, es necesario crear un entorno de simulación en el que se incluyan todos los parámetros necesarios para llevarla a cabo. Simglucose permite generar un entorno completo de simulación codificado en Python.

Simglucose proporciona un conjunto de clases que permite generar entornos completos de simulación:

- `T1DPatient`: Clase que simula un paciente virtual de entre los disponibles en el simulador. Se identifican por un nombre compuesto por el tipo de paciente (*adult*,*adolescent* o *child*) y un número entre 001 y 010.

- `CGMSensor`: Clase que simula los efectos que introduce un MCG en la medida de la glucemia intersticial: ruido, retardo, etc. Los distintos tipos de MCG simulables se identifican con un nombre.

- `InsulinPump`: Clase que simula la infusión de insulina al paciente. Al igual que sucede con los modelos de MCG, los modelos de bomba de insulina simulables se identifican por un nombre.

- `CustomScenario`: Clase en la que se establecen los tiempos de simulación y los momentos en los que suceden las ingestas de hidratos de absorción rápida.

- `T1DSimEnv`: Clase que encapsula a los objetos que forman la simulación y puede ser pasado al motor del simulador para su ejecución.

- `Controller`: Clase base que sirve para generar controladores.

- `SimObj`: Clase a la que se le pasa el objeto T1DSimEnv junto con el controlador a simular.

En el ejemplo de la Figura 6.2 se puede ver cómo se define el paciente a simular (que debe ser uno de los incluidos en el simulador) junto con el sistema MCG y bomba de insulina a utilizar en la simulación.

Posteriormente se define el escenario de ingestas (`scen` en el código del ejemplo) como una lista de duplas (tiempo, tamaño comida) y se genera un entorno de simulación con todos los parámetros.

Se crea una instancia del controlador, que en este caso deberá corresponder con el creado para nuestra simulación, y junto con el entorno se crea un objeto simulación. Este objeto simulación es el que será usado por el simulador.

```
# --------- Create Custom Scenario --------------
# Create a simulation environment
patient = T1DPatient.withName('adolescent#001')
sensor = CGMSensor.withName('Dexcom', seed=1)
pump = InsulinPump.withName('Insulet')
# custom scenario is a list of tuples (time, meal_size)
scen = [(7, 45), (12, 70), (16, 15), (18, 80), (23, 10)]
scenario = CustomScenario(start_time=start_time, scenario=scen)
env = T1DSimEnv(patient, sensor, pump, scenario)

# Create a controller
controller = BBController()

# Put them together to create a simulation object
s2 = SimObj(env, controller, timedelta(days=1), animate=False, path=path)
results2 = sim(s2)
print(results2)


# --------- batch simulation --------------
# Re-initialize simulation objects
s1.reset()
s2.reset()

# create a list of SimObj, and call batch_sim
s = [s1, s2]
results = batch_sim(s, parallel=True)
print(results)
```

Figura 6.2: Ejemplo simulación en simglucose

Las simulaciones se pueden lanzar como lotes o *batches*. Lanzar *batches* permite simular el mismo entorno sobre múltiples pacientes y presentar los resultados de todas las simulaciones de manera agregada. Son especialmente interesantes para ver los resultados de algoritmos de manera global a conjuntos de pacientes y no únicamente sobre pacientes individuales. Es por ello que en esta tesis todas las simulaciones se realizarán en *batches* en los que se incluirán todos los pacientes disponibles en el simulador.

El propio simulador generará las gráficas estándar con todas las estadísticas al finalizar la simulación.

6.2.2.3 Ajuste de parámetros por paciente

El sistema propuesto en esta tesis necesita que se determinen ciertos parámetros del paciente y ciertas configuraciones como datos de entrada, tal y como se explica en el apartado 4.3. Para habilitar que el controlador pueda acceder a estos datos, dado que se pretende simular más de un paciente, se ha creado en el simulador un fichero auxiliar en formato CSV (`AP_params.CSV`) en el que se almacenan y configuran los parámetros que sean necesarios para cada paciente y datos como la insulina utilizada por dicho paciente o la activación/desactivación de ciertos procesos del controlador. De esta manera el controlador puede, haciendo uso del nombre (ID) del paciente (incluido en el parámetro `info`), recuperar su configuración con cada llamada a la función de cálculo del lazo de control.

Los parámetros almacenados por paciente en este fichero son:

- **Nombre (*Name*)**: Este campo contiene el id del paciente simulado y se utiliza como referencia para la búsqueda del resto de parámetros en el controlador.
- **Necesidades basales (*Basal*)**: Las necesidades basales del paciente se especifican aquí para su uso en dos situaciones:
  1. Cuando no se realizan estimaciones en tiempo real de las necesidades basales es necesario proporcionar dicho dato al controlador.
  2. En cualquier caso para inicializar la historia de dosificación de insulina y evitar los efectos de borde al inicio de la simulación.
- **Factor de sensibilidad a la insulina objetivo (*ISFtarget*)**: Factor de sensibilidad a la insulina para el caso en el que la glucemia del paciente se encuentre justo en la glucemia objetivo deseada.
- **Ratio de cambio del factor de sensibilidad a la insulina (*ISFratio*)**: Ratio modificador del factor de sensibilidad a la insulina con el error glucémico del paciente. Para desactivar la modificación del factor de sensibilidad a la insulina se debe configurar este parámetro a `0`.
- **Factor de sensibilidad a la insulina mínimo (*ISFmin*)**: Valor mínimo permitido para el factor de sensibilidad a la insulina tras la modificación via el ratio.
- **Objetivo glucémico máximo (*Targetmax*)**: Valor superior del rango glucémico objetivo.
- **Objetivo glucémico mínimo (*Targetmin*)**: Valor inferior del rango glucémico objetivo.
- **Perfil basal máximo (*maxBasal*)**: Valor máximo para las correcciones del lazo de control. No se pueden configurar basales temporales con valores superiores a este límite.
- **Duración de la insulina (*DIA*)**: Número de horas de duración de la insulina en el paciente.
- **Tipo de insulina utilizado (*InsulinType*)**: Tipo de insulina utilizada en la simulación. En este caso sólo existe un tipo de insulina: Humalog. Mediante este parámetro y la duración a la insulina, el controlador es capaz de calcular la curva de acción de la insulina para el paciente.
- **Ratio de Hidratos de carbono por unidad de insulina (*CR*)**: Ratio mediante el cual el controlador es capaz de calcular la insulina necesaria para “cubrir” una ingesta ante un anuncio de la misma.

- **Tiempo de adelanto de insulina en ingestas (*Prebolusing*)**: Tiempo en minutos del adelanto que se debe utilizar para la dosificación de insulina ante el anuncio de una ingesta.
- **Habilitación del tiempo de adelanto en ingestas (*PreBolusEnable*)**: Habilitación booleana del tiempo de adelanto para la dosificación de insulina ante ingestas.
- **Habilitación del cálculo de necesidades basales en tiempo real (*AutoBasalEnable*)**: Habilitación booleana de la determinación automática de necesidades basales.
- **Habilitación del control en lazo cerrado (*APEnable*)**: Habilitación booleana del control en lazo cerrado.

No se simula la modificación de ninguno de estos parámetros con el tiempo dado que los modelos de los pacientes en este simulador no varían con el tiempo y, por consiguiente, un único valor para cada parámetro es suficiente. Todos estos parámetros son constantes para cada paciente durante toda la simulación. Esta ausencia de variabilidad de parámetros no es extrapolable a un paciente real, y es por ello que el controlador implementado sobre la plataforma *wearable* sí permite la configuración de ciertos parámetros como tablas organizadas por franjas horarias (tal y como se ha descrito en capítulos anteriores).

6.2.2.4 Simulación del bloqueo de la infusión

Aunque el simulador no proporciona la funcionalidad necesaria para incluir efectos de cambios de necesidades basales con el tiempo, y la validación del algoritmo propuesto para la determinación de necesidades basales se puede hacer con un único valor de necesidades, se creyó oportuno habilitar una forma de simular la alteración esas necesidades durante la ejecución de la simulación de manera controlada y ver si el algoritmo se adapta a ellas correctamente.

Para conseguir este efecto, se ha modificado el simulador de manera que el controlador comunica en el objeto de acción de la siguiente manera:

```
action = Action(basalNeedsDetected, actionBolus, blockage)
```

Además del perfil basal y los bolos, se comunica un valor de bloqueo de insulina. Este parámetro actuará reduciendo la insulina efectiva que llega al modelo de simulación pero permitiendo el registro de las cantidades de insulina dosificadas por el controlador.

Existía la posibilidad de reducir artificialmente la insulina resultante de los cálculos dentro del propio controlador, pero en ese caso no existe manera de incluir en los informes la insulina “extra” que el controlador está proporcionando, pues ese valor nunca llega al

simulador. Esta modificación permite simular bloqueos parciales de la infusión de insulina y realizar el registro de dicho bloqueo para su posterior análisis.

Los cambios realizados al simulador se resumen en:

1. Modificación de la clase `simglucose.controller.base` para ampliar el número de parámetros permitidos por el objeto `Action` e incluir el bloqueo. Se puede apreciar en la Figura 6.3 cómo se ha añadido el parámetro `'blockage'` al final de la linea recuadrada en rojo.

```
1  from collections import namedtuple
2
3  Action = namedtuple('ctrller_action', ['basal', 'bolus', 'blockage'])
4
5
6  class Controller(object):
7      def __init__(self, init_state):
8          self.init_state = init_state
9          self.state = init_state
10
```

Figura 6.3: Modificación de simglucose.controller.base

2. Modificación de la clase `simglucose.simulation.env.T1DSimEnv` para registrar y propagar al modelo del paciente la información de bloqueo de insulina (ver Figuras 6.4, 6.5 y 6.6)). Se ha recuadrado en rojo aquellas líneas que han sido modificadas o añadidas para permitir la propagación de los datos de bloqueo (Figura 6.4) y el registro de los bloqueos junto con el resto de resultados de la simulación (Figuras 6.5 y 6.6).

```
48      def mini_step(self, action):
49          # current action
50          patient_action = self.scenario.get_action(self.time)
51          basal = self.pump.basal(action.basal)
52          bolus = self.pump.bolus(action.bolus)
53          blockage = action.blockage
54          insulin = basal + bolus
55          CHO = patient_action.meal
56          patient_mdl_act = Action(insulin=insulin, CHO=CHO, blockage=blockage)
57
```

Figura 6.4: Modificación de simglucose.controller.env.T1DSimEnv - 1

```python
    def step(self, action, reward_fun=risk_diff):
        '''
        action is a namedtuple with keys: basal, bolus and blockage
        '''
        CHO = 0.0
        insulin = 0.0
        BG = 0.0
        CGM = 0.0
        blockage = 0.0

        for _ in range(int(self.sample_time)):
            # Compute moving average as the sample measurements
            tmp_CHO, tmp_insulin, tmp_BG, tmp_CGM, tmp_blockage = self.mini_step(action)
            CHO += tmp_CHO / self.sample_time
            insulin += tmp_insulin / self.sample_time
            BG += tmp_BG / self.sample_time
            CGM += tmp_CGM / self.sample_time
            blockage += tmp_blockage / self.sample_time

        # Compute risk index
        horizon = 1
        LBGI, HBGI, risk = risk_index([BG], horizon)

        # Record current action
        self.CHO_hist.append(CHO)
        self.insulin_hist.append(insulin)
        self.blockage_hist.append(blockage)

```

Figura 6.5: Modificación de simglucose.controller.env.T1DSimEnv - 2

```python

    def show_history(self):
        df = pd.DataFrame()
        df['Time'] = pd.Series(self.time_hist)
        df['BG'] = pd.Series(self.BG_hist)
        df['CGM'] = pd.Series(self.CGM_hist)
        df['CHO'] = pd.Series(self.CHO_hist)
        df['insulin'] = pd.Series(self.insulin_hist)
        df['blockage'] = pd.Series(self.blockage_hist)
        df['LBGI'] = pd.Series(self.LBGI_hist)
        df['HBGI'] = pd.Series(self.HBGI_hist)
        df['Risk'] = pd.Series(self.risk_hist)
        df = df.set_index('Time')
        return df

```

Figura 6.6: Modificación de lsimglucose.controller.env.T1DSimEnv - 3

3. Modificación de la clase `simglucose.simulation.t1dpatient.T1DPatient` para implementar el bloqueo de insulina en el modelo del paciente (ver Figuras 6.7 y 6.8). La Figura 6.8 muestra parte de la implementación del modelo del paciente y se ha remarcado en rojo la parte del código que ha sido añadida para simular el bloqueo de insulina. La variable `insulin` es la que proporciona al modelo la insulina que está recibiendo. Para simular el bloqueo, se ha generado una nueva variable llamada `final_action_insulin` que se obtiene como resultado de restar el bloqueo de insulina a la insulina que entra al modelo. Se puede apreciar dentro del recuadro rojo la línea original del simulador que implementaba la misma función sin tener en cuenta el bloqueo. También se implementa un proceso de saturación por debajo de cero ya que no tiene sentido físico una infusión de insulina negativa que, por otro lado, podría ser resultado de restar un cierto bloqueo en casos de que la infusión

programada de insulina sea de un valor inferior a la programada como bloqueo.

```
1  from .base import Patient
2  import numpy as np
3  from scipy.integrate import ode
4  import pandas as pd
5  from collections import namedtuple
6  import logging
7  import pkg_resources
8
9  logger = logging.getLogger(__name__)
10
11 Action = namedtuple("patient_action", ['CHO', 'insulin','blockage'])
12 Observation = namedtuple("observation", ['Gsub'])
13
14 PATIENT_PARA_FILE = pkg_resources.resource_filename(
15     'simglucose', 'params/vpatient_params.csv')
16
17
18 class T1DPatient(Patient):
19     SAMPLE_TIME = 1  # min
20     EAT_RATE = 5     # g/min CHO
```

Figura 6.7: Modificación de simglucose.controller.t1dpatient.T1DPatient - 1

```
115
116     @staticmethod
117     def model(t, x, action, params, last_Qsto, last_foodtaken):
118         dxdt = np.zeros(13)
119         d = action.CHO * 1000  # g -> mg
120         final_action_insulin = action.insulin - action.blockage
121         if (final_action_insulin < 0.0):
122             final_action_insulin = 0.0
123         #insulin = action.insulin * 6000 / params.BW  # U/min -> pmol/kg/min
124         insulin = final_action_insulin * 6000 / params.BW # U/min -> pmol/kg/min
125         basal = params.u2ss * params.BW / 6000  # U/min
126
```

Figura 6.8: Modificación de simglucose.controller.t1dpatient.T1DPatient - 2

6.2.3 Implementación del controlador

Tanto la clase que implementa los controladores descritos en esta tesis, como la declaración del entorno de simulación, se han incluido en un único fichero Python que ejecuta la simulación (`apply_customized_controller.py`).

6.2.3.1 Inicialización del controlador

En el método de inicialización se crean las variables necesarias para mantener los datos que necesiten persistencia entre ejecuciones del lazo de control (ver Figura 6.9):

- `self.state`: Almacena el estado interno del controlador para el simulador.
- `self.AP_params`: Almacena los parámetros específicos del controlador propuesto en la tesis para cada paciente. Los parámetros incluidos en esta variable quedan descritos en el apartado 6.2.2.3.

- `self.prevMeal`: Almacena el último dato de ingesta simulado para detectar cambios en la variable y poder hacer detección de ingestas.
- `self.simulationTime`: El controlador lleva la cuenta de tiempo de simulación por varios motivos:
  1. En el caso de utilizar tiempo de adelanto para las ingestas es necesario conocer, además del escenario programado, el tiempo en el que se encuentra la ejecución. De esta manera la simulación puede tomar la decisión de dosificar el bolo de la comida antes de que esta realmente sea efectiva. Es la manera más sencilla de simular la intencionalidad del paciente a la hora de hacer *Pre-Bolusing* (poner insulina antes de la ingesta).
  2. En ciertas simulaciones se provocará un bloqueo parcial de la insulina a partir de una hora en concreto. Para poder aumentar el parámetro del bloqueo desde el controlador es necesario conocer el tiempo de simulación. La modificación realizada al simulador en los apartados anteriores permite que sea el mismo controlador quien, con cada ejecución del lazo de control, determine la cantidad de insulina a ser bloqueada. Es posible que delegar esta funcionalidad al controlador no sea lo más apropiado desde un punto de vista puramente de simulación, pero sí nos permite un sencillo control del bloqueo sin tener que incluir nuevos elementos y modificaciones mucho más profundas en el simulador.
  3. En tiempo de inicialización del controlador todavía no se conoce el paciente que se va a simular, es por ello que ciertas variables son declaradas e inicializadas a valores irreales durante la inicialización y, durante el primer ciclo de ejecución del lazo de control cuando se conoce el paciente simulado, se inicializan a su valor real. Para detectar la primera ejecución del lazo de control se utiliza el tiempo de simulación.
- `self.cgmHistory`: Buffer en el que se almacenan los últimos datos recibidos del MCG.
- `self.cgmDeriv`: Buffer en el que se almacena la derivada de los últimos datos recibidos del MCG.
- `self.basalHistory`: Buffer en el que se almacenan las últimas dosificaciones de insulina basal.
- `self.basalNeedsHistory`: Buffer en el que se almacenan los últimos cálculos de necesidades basales de insulina.
- `self.bolusHistory`: Buffer en el que se almacenan los últimos bolos dosificados.
- `self.DIA`: Valor de duración de la insulina en el paciente. Como no se conoce este parámetro en tiempo de inicialización, se asigna un valor aleatorio y se actualizará posteriormente.

- `self.insulinType`: Tipo de insulina utilizado por el paciente. Como no se conoce este parámetro en tiempo de inicialización, se asigna un valor aleatorio y se actualizará posteriormente.

- `self.insulinResponse`: Buffer en el que se almacenan la curva de acción a la insulina una vez adaptada al paciente. Esta curva se calcula mediante el método `calculateInsulinResponse(self, DIA, insulinType)` incluido también en el controlador. Este método toma la curva prefijada para el tipo de insulina utilizado y la adapta a la duración de insulina del paciente. Al depender del tipo de insulina y su duración, esta curva deberá ser recalculada cuando estos parámetros se ajusten para el paciente.

- `self.mealSchedule`: Buffer de duplas que notifican al controlador en qué momento sucederán las ingestas. Este buffer únicamente será de utilidad en caso de que el paciente simulado utilice tiempo de adelanto para la insulina.

- `self.simBlockage`: Como ya se ha comentado anteriormente, habrá ciertas simulaciones en las que desearemos incluir un cierto bloqueo de insulina a partir de cierto tiempo de simulación. En esta variable booleana almacenaremos si deseamos activar el bloqueo o no.

```
111     def __init__(self, init_state, batch_sim):
112         self.init_state = init_state
113         self.state = init_state
114         self.AP_params = pd.read_csv(AP_PARA_FILE)
115
116         self.prevMeal = 0
117         self.simulationTime = 0
118         self.cgmHistory = np.zeros(12*24,np.int16,'C')
119         self.basalHistory = np.ones(12*24+6,np.float64,'C')
120         self.basalNeedsHistory = np.zeros(12*24+6,np.float64,'C')
121         self.cgmDeriv = np.zeros(12*48,np.float64,'C')
122         self.bolusHistory = np.zeros(12*24,np.float64,'C')
123         self.insulinResponse = np.zeros(12*24,np.float64,'C')
124         self.DIA = 6
125         self.insulinType = 'humalog'
126         self.calculateInsulinResponse(self.DIA,self.insulinType)
127         if (batch_sim == 2) or (batch_sim == 4):
128             self.mealSchedule = dict([(7*60, 30), (13*60, 20), (17*60, 5), (20*60, 20), (23*60, 0)])
129         else:
130             self.mealSchedule = dict([(7*60,  0), (13*60,  0), (17*60, 0), (20*60,  0), (23*60, 0)])
131         if (batch_sim == 3) or (batch_sim == 4):
132             self.simBlockage = True
133         else:
134             self.simBlockage = False
135
```

Figura 6.9: Inicialización del controlador

6.2.3.2 Método de cálculo

El método `policy` del controlador es el encargado de realizar los cálculos con cada iteración del lazo de control. La primera parte del código, mostrada más abajo, recupera el nombre del paciente a partir de la información pasada al controlador. Haciendo uso de la variable `self.AP_params` se recuperan los parámetros de configuración del paciente

y se asignan a variables locales.

```
pname = info.get('patient_name')
meal = info.get('meal')

ap = self.AP_params[self.AP_params.Name.str.match(pname)]

  # Una vez obtenida la tabla de parámetros para el paciente (estructura ap),
  # se recupera cada parámetro de manera individual para mayor claridad del código

insulinType = ap.InsulinType.values[0]
dia = ap.DIA.values[0]
maxBasal = ap.maxBasal.values[0]
ISFtarget = ap.ISFtarget.values[0]
ISFratio = ap.ISFratio.values[0]
ISFmin = ap.ISFmin.values[0]
targetMax = ap.TargetMax.values[0]
targetMin = ap.TargetMin.values[0]
carbRatio = ap.CR.values[0]
preBolusingMinutes = ap.PreBolusing.values[0]
preBolusingEnable = ap.PreBolusEnable.values[0]
autoBasalEnable = ap.AutoBasalEnable.values[0]
APEnable = ap.Enable.values[0]
scen_dict = self.mealSchedule
```

Una vez obtenidos los parámetros de configuración del paciente, se comprueba si el tiempo de simulación es 0 (inicio de la simulación). En caso de que la simulación acabe de iniciarse, se procede a la inicialización de los buffers de historia basal y de necesidades basales. Como ya se explicó en apartados anteriores, ambas historias se inicializan como si se hubiera dosificado las necesidades basales configuradas para el paciente en la variable `self.AP_params`.

```
if (self.simulationTime == 0):
        # np = numpy. Librería estándar de Python
        self.basalHistory = np.ones(12*24+6,np.float64,'C') * (ap.Basal.values[0]+0.2)/12
        self.basalNeedsHistory = np.ones(12*24+6,np.float64,'C') * ap.Basal.values[0]
```

El tiempo de simulación se actualiza justo después de la comprobación de inicio de simulación incrementándolo en un `sample_time` o tiempo de muestra.

```
self.state = observation
self.simulationTime += info.get('sample_time')
```

A partir de este momento en el código se van a ejecutar los pasos descritos en los capítulos 4 y 5 siguiendo el diagrama de flujo de la Figura 5.1. Los primeros cálculos a realizar son todos los relacionados con la insulina activa. Para comenzar, y para dar

respuesta al problema de no poder calcular ciertos parámetros durante la inicialización del controlador, se comprueba si la duración del efecto de la insulina o el tipo de insulina utilizado por el paciente son distintos a los utilizados durante la inicialización del controlador. Si fuera así, se procede a re-calcular la curva de acción de la insulina para el paciente y a actualizar la duración y tipo de insulina en las variables internas del controlador.

```
if (self.DIA != dia) or (self.insulinType != insulinType):
        self.calculateInsulinResponse(dia,insulinType)
```

Una vez el controlador dispone de la curva de acción apropiada, se calculan los vectores de insulina activa instantánea tanto para las dosificaciones basales (`basaliIOB`) como para los bolos de insulina (`bolusiIOB`) convolucionando sus respectivas historias de dosificación con la curva de acción de la insulina. Utilizando la insulina activa instantánea, se calculan los vectores de insulina activa para las dosificaciones basales (`basalIOB`) y para los bolos de insulina (`bolusIOB`) como el sumatorio de las insulinas activas instantáneas desde el índice correspondiente al momento actual de simulación hasta el final de los buffers. En el caso de la insulina activa correspondiente a las basales, además, es necesario descontar las últimas necesidades basales calculadas a cada valor de la insulina activa instantánea (ver Ecuación 5.1). Para finalizar, se calcula la insulina activa total (`totalIOB`) como la suma de las insulinas activas anteriores.

```
# np = numpy. Librería estándar de Python
basaliIOB = np.convolve(self.insulinResponse,self.basalHistory,'full')
bolusiIOB = np.convolve(self.insulinResponse,self.bolusHistory,'full')
futureBasalIOB = basaliIOB[24*12:24*12+6] - np.ones(6,np.float64,'C') *
                                (self.basalNeedsHistory[-1])/12
futureBolusIOB = bolusiIOB[24*12:-1]
basalIOB = np.sum(futureBasalIOB)
bolusIOB = np.sum(futureBolusIOB)
totalIOB = basalIOB + bolusIOB
```

Utilizando el nuevo dato proveniente del MCG, se calcula su derivada con respecto al último dato del sensor y se actualiza el buffer de derivadas del MCG (`self.cgmDeriv`). Este buffer se actualiza desplazando los datos una posición hacia la izquierda (simulando el paso del tiempo) y colocando la nueva derivada en la última posición del buffer.

```
# np = numpy. Librería estándar de Python
self.cgmDeriv = np.roll(self.cgmDeriv,-1)
self.cgmDeriv[-1] = observation.CGM - self.cgmHistory[-1]
```

De manera similar al resto de buffers de historias, se desplaza una posición el buffer de historia de datos del MCG y se coloca el último dato recibido al final del buffer. Con los

datos del sensor ya actualizados, se calcula cuanto fue la última desviación (dato actual del sensor menos el dato previo) y se calcula la glucemia actual (`eventualCGMWoIOB`) como el último dato del sensor continuando la desviación 15 minutos (3 pasos de simulación).

```
# np = numpy. Librería estándar de Python
self.cgmHistory = np.roll(self.cgmHistory,-1)
self.cgmHistory[-1] = observation.CGM
deviation = sum(self.cgmHistory[-3:-1] - self.cgmHistory[-4:-2])
eventualCGMWoIOB = observation.CGM + deviation*3
```

El siguiente paso consiste en calcular el objetivo glucémico final (`targetFinal`), evaluando directamente las Ecuaciones 5.10, 5.11 y 5.12, y el factor de sensibilidad a la insulina final (`ISFfinal`), haciendo uso de las Ecuaciones 5.8 y 5.9.

```
targetDesired = (targetMax + targetMin)/2.0
targetDynamic = (observation.CGM + targetMin)/2.0
targetFinal = targetMax
if (observation.CGM >= targetMin):
        if (targetDynamic <= targetDesired):
                targetFinal = targetDesired
        else:
                if (targetDynamic >= targetMax):
                        targetFinal = targetMax
                else:
                        targetFinal = targetDynamic

ISFfinal = ISFtarget - ISFratio*(np.abs(eventualCGMWoIOB - targetDesired))
if (ISFfinal < ISFmin):
        ISFfinal = ISFmin
```

Utilizando el factor de sensibilidad a la insulina final, la insulina activa y la glucemia actual, se calcula la glucemia final del paciente (`eventualCGM`) utilizando las Ecuaciones 4.2 y 4.3. Y, haciendo uso de la glucemia final del paciente ya se posible calcular el error glucémico (`error`) a compensar por el algoritmo.

```
eventualCGM = eventualCGMWoIOB - totalIOB*ISFfinal
error = eventualCGM - targetFinal
```

Llegados a este punto, el controlador puede calcular las necesidades basales utilizando el método descrito en los apartados 5.3.2 y 5.3.4 o continuar usando las necesidades basales aportadas en la configuración para el paciente (simulando de esta manera el algoritmo de control básico). En caso de habilitar la determinación automática de necesidades basales (`autoBasalEnable == True`), se toman los últimos datos de la historia de iIOB y datos del MCG para tener la información correspondiente a las duplas de la regresión y, si se detecta que los datos del MCG han variado dentro

de la ventana de tiempo (`thereWasAChange >0`) y por tanto no se viola ninguna condición, se procede al cálculo de las necesidades basales (`basalNeeds`) siguiendo la Ecuación 5.5. En caso de no poder calcular las necesidades basales debido a condiciones de los datos de entrada, se utiliza como estimación la última estimación obtenida.

```
if (autoBasalEnable == True):
        lastCGMDeriv = self.cgmDeriv[-6:-1]
        lastiIOB = basaliIOB[24*12-5:24*12]

        thereWasAChange = sum(abs(lastCGMDeriv[0:-2]
                                - lastCGMDeriv[1:-1]))

        if (self.simulationTime < info.get('sample_time')*7):
                thereWasAChange = 0

        if (thereWasAChange > 0):
                basalNeeds = (np.mean(lastiIOB) +
                        (np.mean(lastCGMDeriv)/ISFfinal))*12
        else:
                basalNeeds = self.basalNeedsHistory[-1]
```

Una vez obtenidas las necesidades basales, estas se almacenan en la historia de necesidades basales y se evalua la tendencia de las mismas según el método descrito en el apartado 5.3.5 en la Ecuación 5.6. Tal y como se describe en la Ecuación 5.7, si la tendencia es descendente se procede a la extrapolación de las nuevas necesidades basales, dando como resultado las necesidades basales finales (`basalNeeds`).

```
# np = numpy. Librería estándar de Python
self.basalNeedsHistory = np.roll(self.basalNeedsHistory,-1)
self.basalNeedsHistory[-1] = basalNeeds

if (thereWasAChange > 0):
        needsDeriv = np.mean(self.basalNeedsHistory[-3:-1]
                                - self.basalNeedsHistory[-4:-2])
        if (needsDeriv < 0):
                basalNeeds = basalNeeds + 3*needsDeriv
```

Haciendo uso de las necesidades basales del paciente, del error glucémico a compensar y el factor de sensibilidad final obtenidos en este ciclo de ejecución del lazo de control, se calcula la acción correctora a aplicar sobre las necesidades basales (`actionBasal`).

```
actionBasal = (error/ISFfinal)
if (actionBasal > (maxBasal - basalNeeds)):
        actionBasal = (maxBasal - basalNeeds)
if (actionBasal < -basalNeeds):
        actionBasal = -basalNeeds
```

El controlador es responsable de dosificar los bolos para las ingestas (lo cual difiere ligeramente del algoritmo propuesto), pues no existe otra forma de simular la interacción del paciente. El controlador permite implementar tiempo de adelanto de la insulina para las ingestas (`preBolusingEnable`) y, en caso de activarse, la simulación debe ser capaz de determinar si es el momento preciso para dosificar la insulina. El simulador ofrece información de las ingestas justo en el momento en el que ocurren y, como es obvio, esa información no es suficiente. Para lograr el objetivo, se configura durante la inicialización del controlador un vector que contiene los momentos y cantidades en los que ocurrirán las ingestas. Haciendo uso de esa información, junto con el tiempo de adelanto programado para el paciente y el tiempo de simulación, el controlador es capaz de dosificar la insulina en el momento justo. Se utiliza una variable intermedia (`internalMeal`) para regenerar la misma información que nos ofrece el simulador pero con el tiempo de adelanto. Si no se desea adelantar la insulina, simplemente se copia el valor ofrecido por el simulador en esta variable intermedia.

Es importante remarcar que el proporcionar esta información al controlador no simplifica ni la implementación ni el problema de base. Lo deseable sería que el simulador proporcionara esta información en el momento apropiado pero, como no es el caso, esta implementación nos permite simular este escenario sin hacer cambios importantes al simulador.

```
if (preBolusingEnable == True):
        if ((self.simulationTime + preBolusingMinutes) in scen_dict):
                internalMeal = scen_dict[(self.simulationTime + preBolusingMinutes)]
        else:
                internalMeal = 0
else:
        internalMeal = meal
```

La acción en forma de bolo (`actionBolus`), que únicamente se utiliza en esta simulación para gestión de ingestas, se calcula directamente multiplicando los gramos ingeridos por el ratio de hidratos de carbono/insulina configurado para el paciente.

```
actionBolus = internalMeal / carbRatio
```

Llegado este punto se comprueba si se ha habilitado el control en lazo cerrado (`APEnable = True`) y, en caso afirmativo, se hace la primera comprobación de seguridad: Suspensión por hipoglucemia. Siguiendo la descripción del apartado 4.3.10, si cualquiera de las glucemias calculadas para el paciente se encuentran por debajo del nivel de suspensión por hipoglucemia, o lo que es lo mismo el valor mínimo del rango objetivo, la acción basal se hará cero independientemente de los cálculos previos del algoritmo. En caso contrario, se configura la historia basal como una basal temporal de 30 minutos que incluye la acción calculada previamente. En el supuesto caso de no activar el control en

lazo cerrado, se aplica directamente las necesidades basales programadas para el paciente.

```
if (APEnable == True):

        if ((observation.CGM < targetMin) or
            (eventualCGMWoIOB < targetMin) or
            (eventualCGM < targetMin)) :
                self.basalHistory[24*12:24*12+6] = np.zeroes(6)
        else:
                self.basalHistory[24*12:24*12+6] = np.ones(6)*
                                (actionBasal/6 + basalNeeds/12)

        basalNeedsDetected = self.basalHistory[24*12]/5

else:
        basalNeedsDetected = basalNeeds/60

# np = numpy. Librería estándar de Python
self.basalHistory = np.roll(self.basalHistory,-1)
```

El último cálculo a realizar por el controlador es la cantidad de insulina a bloquear. Por decisiones que serán explicadas posteriormente, el controlador permite que se obstruya una cantidad equivalente a la mitad de las necesidades basales programadas para el paciente y sólo tras simular 12 horas. El bloqueo de insulina será 0 a no ser que el bloqueo esté activo (`self.simBlockage == True`) y la condición temporal también se cumpla.

```
blockage = 0.0
if ((self.simulationTime >= (12*60)) and (self.simBlockage == True)):
        blockage = (ap.Basal.values[0]/2)/60
```

Una vez calculadas las acciones de esta iteración se actualizan los buffers de historias y se comanda la acción con su bloqueo.

```
self.prevMeal = internalMeal
# np = numpy. Librería estándar de Python
self.bolusHistory = np.roll(self.bolusHistory,-1)
self.bolusHistory[len(self.bolusHistory)-1] = actionBolus

action = Action(basalNeedsDetected, actionBolus, blockage)
```

6.2.4 Ejecución de la simulación

Como ya se explicó anteriormente, el mismo fichero Python que implementa el controlador también define cómo se debe ejecutar la simulación. El simulador permite simular el

número de pacientes que se consideren oportunos y, en este caso, se seleccionan todos los pacientes para tener todos los datos agregados.

```
patient_names=['adolescent#001','adolescent#002','adolescent#003',
        'adolescent#004','adolescent#005','adolescent#006',
        'adolescent#007','adolescent#008','adolescent#009',
        'adolescent#010',
        'adult#001','adult#002','adult#003','adult#004',
        'adult#005','adult#006','adult#007','adult#008',
        'adult#009','adult#010','child#001','child#002',
        'child#003','child#004','child#005','child#006',
        'child#007','child#008','child#009','child#010']
```

El tiempo de simulación es de 24 horas y el escenario de ingestas dependerá del *batch* de simulación deseado. Con todo ello se crea un escenario *custom*.

```
now = datetime.now()
start_time = datetime.combine(now.date(), datetime.min.time())
sim_time=timedelta(days=1)

batch_number = 4

if (batch_number == 2) or (batch_number == 4):
        scen = [(7, 30), (13,20), (17, 5), (20, 20), (23, 0)]
else:
        scen = [(7, 00), (12,00), (17, 0), (20, 00), (23, 0)]

scenario = CustomScenario(start_time=start_time, scenario=scen)
```

Se define un array de entornos de simulación con la lista de pacientes a simular, el tipo de MCG simulado (GuardianRT en nuestro caso), el modelo de bomba de insulina simulado (Insulet) y el escenario previamente creado.

```
def local_build_env(pname):
        patient = T1DPatient.withName(pname)
        cgm_sensor = CGMSensor.withName('GuardianRT', seed=1)
        insulin_pump = InsulinPump.withName('Insulet')
        scen = copy.deepcopy(scenario)
        env = T1DSimEnv(patient, cgm_sensor, insulin_pump, scen)
        return env

envs = [local_build_env(p) for p in patient_names]
```

Se crea el controlador en base al código que describe el algoritmo propuesto y se lanza la ejecución de las simulaciones y generación de informes.

```
controller = MyController(0,batch_number)
ctrllers = [copy.deepcopy(controller) for _ in range(len(envs))]
path = './results_batch' + str(batch_number)
sim_instances = [SimObj(e,c,sim_time,animate=False,path=path) for (e, c) in zip(envs, ctrllers)]
results = batch_sim(sim_instances, parallel=False)
df = pd.concat(results, keys=[s.env.patient.name for s in sim_instances])
results, ri_per_hour, zone_stats, figs, axes = report(df, path)
```

6.2.5 Estrategia de simulación

Las simulaciones realizadas muestran, haciendo uso de las estadísticas ofrecidas por el simulador, las diferentes mejoras que se obtienen en terapia con cada incremento de capacidades del lazo de control. Podemos agrupar las simulaciones en los siguientes grupos:

1. **Lazo abierto**: Como primer paso se evalúa cómo sería un control en lazo abierto en el que el paciente no interviene para hacer correcciones. Este caso podría ser el de un paciente que no disponga de un sistema MCG y por tanto desconozca su estado glucémico.

2. **Lazo abierto con correcciones post-prandiales**: Al caso de simulación anterior se le añade la posibilidad de dosificar bolos correctores de insulina transcurridas dos horas desde la ingesta. Este caso podría ser el de un paciente con alta motivación que compruebe su glucemia capilar después de las comidas o que haga uso de un sistema MCG.

3. **Lazo cerrado sin ajustes dinámicos**: Al mismo escenario de simulación del caso anterior se le permite hacer correcciones en lazo cerrado pero sin determinar las necesidades basales de manera automática ni dosificar bolos correctores.

4. **Lazo cerrado con ajustes dinámicos**: En este conjunto de simulaciones se activa la determinación automática de necesidades basales pero, para poder hacer un estudio más profundo de las posibilidades del controlador, se subdividen estas pruebas en lo que se ha denominado como *batches* o lotes de simulación.

   - **Sin bloqueo de insulina y sin ingestas de hidratos de carbono (Batch 1)**: En este *batch* se simula el escenario sin activar el bloqueo de insulina y, además, se evita el simular ingestas para comprobar el comportamiento del lazo de control en ausencia de agentes externos.
   - **Sin bloqueo de insulina y con ingestas de hidratos de carbono (Batch 2)**: En el *batch* 2 se simulan los mismos parámetros que en el *batch* 1 pero se introducen ingestas de hidratos de carbono. Para la gestión de las ingestas se utiliza tiempo de adelanto para la insulina.

- **Con bloqueo de insulina y sin ingestas de hidratos de carbono (Batch 3)**: El *batch* 3 es equivalente al *batch* 1 pero incluyendo un bloqueo parcial de la basal a partir de las 12 horas. Este bloqueo es proporcional a las necesidades basales programadas para cada paciente: debido a la gran disparidad de valores entre pacientes y a que lo deseable es simular unas situaciones equivalentes para cada uno, el bloqueo generado es igual a la mitad de las necesidades basales programadas en el fichero de parámetros de los pacientes.
- **Con bloqueo de insulina y con ingestas de hidratos de carbono (Batch 4)**: En el último *batch* de simulaciones se inluye el efecto de las ingestas además del bloqueo parcial de la basal.

Los objetivos a conseguir con estas simulaciones son:

1. Demostrar que el control en lazo cerrado ofrece resultados superiores frente a la terapia en lazo abierto.
2. Demostrar que la determinación automática de necesidades basales en tiempo real es factible y ofrece resultados comparables a un control en lazo cerrado con necesidades basales correctamente configuradas.
3. Demostrar que la determinación automática de necesidades basales en tiempo real es capaz de adaptarse a los cambios en necesidades basales.

6.3 Resultados de las simulaciones

En este apartado se incluyen los datos y gráficas resultantes de las simulaciones, así como la interpretación de los mismos.

6.3.1 Interpretación de gráficas de resultados

Los resultados de las simulaciones se muestran como figuras y tablas con resultados de “tiempo en rango". Las figuras de resultados se dividen en cuadro subfiguras:

1. *Blood Glucose (mg/dl)*: Muestra las curvas de glucemia resultantes de la simulación de los pacientes (en color gris) en unidades de mg/dl. Al haber simulado más un paciente, también se ofrecen las curvas de media de todos los pacientes (línea azul) y la región que coincide con la desviación estándar de todas (región azul). Los niveles marcados como hiperglucemia (180 mg/dl) e hipoglucemia (70 mg/dl) se representan como líneas horizontales de puntos rojos y verdes respectivamente.
2. *CGM (mg/dl)*: Muestra los datos recibidos por el MCG simulado para cada paciente en mg/dl. Tanto esta gráfica como la superior deberían mostrar los mismos datos

si el sensor fuera ideal pero, como no se simula de tal manera, se puede apreciar variaciones debidas a ruido en las medidas. La manera de representar los datos es equivalente a la subfigura superior.

3. *Insulin (units)*: Esta subfigura muestra la insulina que ha sido dosificada con el tiempo medida en unidades de insulina (U). Es importante remarcar que el simulador utiliza una resolución temporal de 1 minuto y, por tanto, las cantidades basales aparecen representadas en U/min. Cada paciente aparece representado con un color distinto.

4. *CHO (g)*: Muestra la ingesta de hidratos de carbono, medida en gramos, que se ha simulado. Las ingestas de hidratos de carbono tampoco se simulan como si ocurrieran de manera instantánea. Es por ello que se simula que la ingesta ocurre en 5 minutos y los hidratos representados se miden en gr/min. Al igual que en el caso de las representaciones de insulina, cada paciente aparece representado en un color distinto.

6.3.2 Simulación de ingestas de hidratos de carbono

Es conocido que las dietas o hábitos alimenticios de niños, adolescentes y adultos pueden ser diferentes. Es por ello que encontrar la correcta distribución de hidratos de carbono que se ajusta a las necesidades de simulación de todos los grupos de pacientes no es un problema trivial. La evidencia [86] también sugiere que no hay un porcentaje ideal de calorías provenientes de hidratos de carbono, proteína y grasa para personas con diabetes y las dietas deben ser individualizadas. Esta misma evidencia indica que reducir la ingesta global de hidratos de carbono mejora el control glucémico.

Dada la complejidad del problema de ajustar la simulación de las ingestas para los distintos grupos de paciente simulados y, dado que el objetivo no es demostrar un cierto rendimiento del algoritmo en cuanto a TIR sino demostrar la efectividad de la determinación de necesidades basales en presencia de perturbaciones como ingestas, se toma la decisión de diseñar un plan de ingestas medio y utilizarlo para todos los pacientes.

Las ingestas simuladas se dividen en cuatro momentos de simulación, con igual contenido de hidratos de carbono iguales para todos los pacientes y con tiempos de adelanto (pre-bolusing) individualizados según las necesidades de cada uno. El plan de ingestas elegido contiene 140 gramos de hidratos de carbono de alto índice glucémico que equivale a una dieta de 2000 kcal con un 28 % de su contenido en hidratos de carbono o a una dieta de 1500 kcal con un 37 % de su contenido en hidratos de carbono. La distribución simulada es la siguiente:

- 07:00 ->56 gramos de hidratos de carbono
- 13:00 ->37 gramos de hidratos de carbono
- 17:00 ->9 gramos de hidratos de carbono

- 20:00 ->38 gramos de hidratos de carbono

6.3.3 Simulaciones sobre el modelo de controlador básico

Se han realizado simulaciones con configuraciones similares a las descritas en el algoritmo básico de control para utilizar los resultados como base sobre los que poder comparar los resultados utilizando determinación automática de necesidades basales. Se considera que el control en lazo cerrado con parámetros fijos, aunque no esté suficientemente extendido o popularizado, es una realidad a día de hoy y es la base sobre la que se deben comparar el resto de sistemas.

6.3.3.1 Simulaciones en lazo abierto

En la simulación del sistema en lazo abierto, donde no hay ningún tipo de corrección ni automática ni por parte del paciente, existe una tendencia más o menos generalizada hacia la hiperglucemia. Como se puede apreciar en la Figura 6.10, la media de la glucemia de todos los pacientes aumenta con el tiempo (linea azul en la subfigura *Blood Glucose*). Es por este motivo que no se muestra la línea de límite de hipoglucemia: todos los pacientes se alejan de ella y no aparece ese nivel de glucemia en la gráfica. También se puede apreciar cómo las ingestas de hidratos de carbono simuladas en el escenario del simulador, correspondientes al desayuno, comida, merienda y cena, aparecen en la subfigura *CHO* y el controlador dosifica los bolos necesarios, cada paciente con su tiempo de adelanto correspondiente, para cubrir las ingestas utilizando los factores de sensibilidad correspondientes a cada paciente. Estos bolos se pueden ver en la subfigura *Insulin* como ”picos” de diferentes colores. Al tener factores de sensibilidad y tiempos de adelanto distintos para cada paciente, los bolos aparecen con cantidades distintas y en tiempos distintos. También se puede ver cómo cada paciente, representado cada uno con un color distinto, recibe distintas cantidades de insulina para las mismas cantidades de hidratos de carbono. Al simular los mismos hidratos de carbono para todos los pacientes, sólo se ve la gráfica de hidratos de carbono correspondiente a uno de los pacientes pues están todas superpuestas.

Siempre será discutible si con otro ajuste de perfiles basales o ratios de hidratos de carbono se hubiera podido conseguir mejores resultados, pero se considera que los resultados son relevantes pues las necesidades basales, como ya se ha comentado en apartados anteriores, cambian con el tiempo y es muy probable que el paciente no tenga un perfil perfectamente ajustado para las necesidades del día en concreto.

La conclusión que se obtiene de esta simulación es que, si no existe ningún tipo de control por parte del paciente u otro sistema, los pequeños desajustes en el perfil basal pueden provocar grandes desajustes glucémicos en los pacientes.

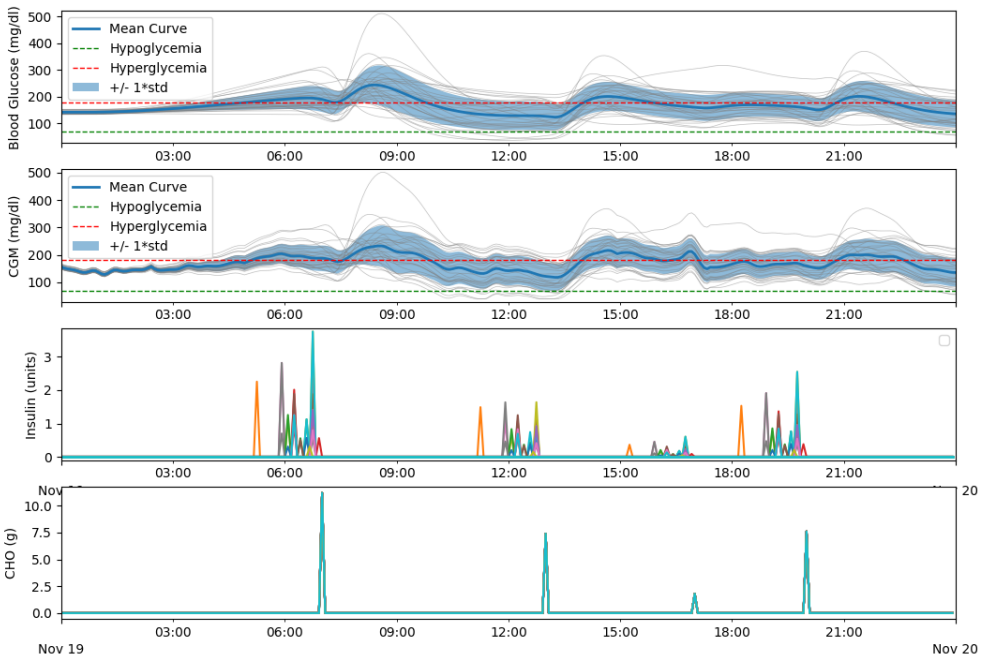


Figura 6.10: Resultados de la simulación en lazo abierto

6.3.3.2 Simulaciones en lazo abierto con correcciones post-prandiales

Las simulaciones en lazo abierto con correcciones post-prandiales corresponden a un escenario en el que el paciente, aun siguiendo una terapia en lazo abierto, hace correcciones mediante bolos correctores cada cierto tiempo. Los resultados están claramente condicionados por la exactitud de los ajustes pero, de igual manera que en el caso anterior, los resultados nos indican que la inexactitud conlleva mal control y que cuantas más correcciones se hacen, más mejora el control.

Con hacer una corrección transcurridas dos horas después de cada ingesta ya se puede apreciar a simple vista la mejoría (ver Figura 6.11): las ingestas provocan una desviación glucémica a la alza que, posteriormente, es compensada con un bolo corrector. Esto hace que la media de la glucemia de los pacientes vuelva a entrar dentro del rango objetivo (70-180 mg/dl) y que la desviación estándar sea inferior al caso del lazo abierto sin correcciones. También se pueden observar los bolos correctores tras las ingestas en la subfigura *Insulin*. En esta simulación se han administrado los bolos correctores una vez transcurridas dos horas desde las ingestas programadas en el escenario, que no desde el bolo correspondiente a la ingesta pues se administra la insulina con adelanto (pre-bolusing), y por ello todo los bolos correctores de todos los pacientes están superpuestos a las mismas horas (09:00, 15:00, 19:00 y 22:00). Si se comparan las subfiguras *Insulin* de las Figuras 6.10 y 6.11 se puede comprobar que la única diferencia son los bolos correctores a dichas horas.

Este tipo de terapia y resultados son los que podrían equivaler a un paciente que realiza un control manual con cierta periodicidad y es capaz de ajustar perfectamente sus necesidades basales, calcular sus dosis de insulina y las cantidades de hidratos que ingiere correctamente cuando es necesario. El grado de error que típicamente se comete al hacer estas estimaciones provoca que, por desgracia, el paciente que sigue este tipo de terapia no logre un control como el de la Figura 6.11 de manera habitual.

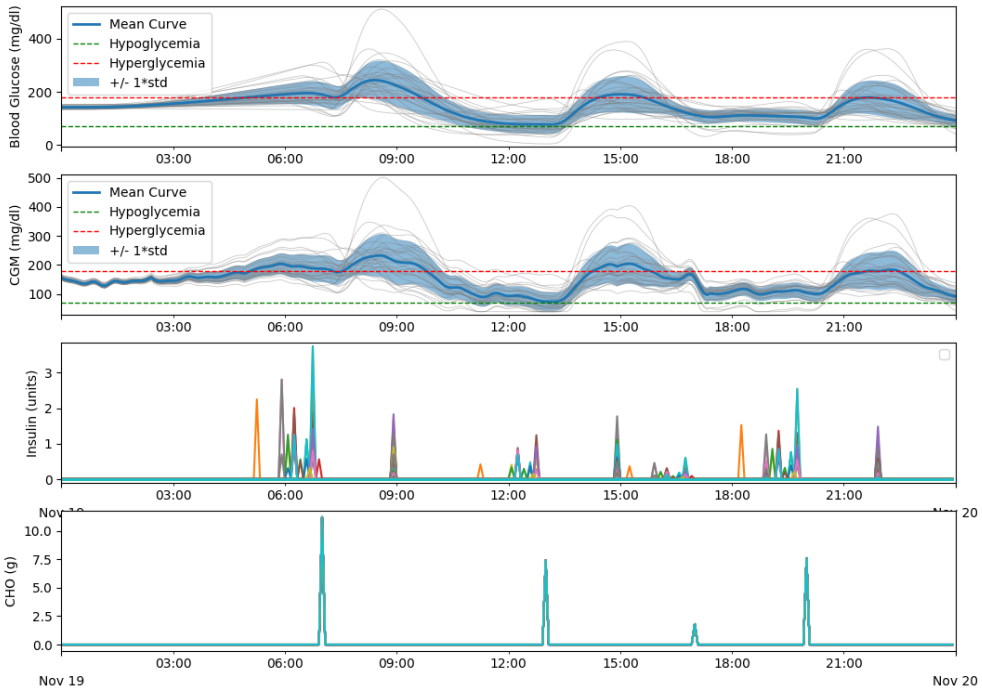


Figura 6.11: Resultados de la simulación en lazo abierto con correcciones post-prandiales

6.3.3.3 Simulaciones en lazo cerrado

En el momento que la terapia utiliza un lazo cerrado de control los resultados cambian drásticamente.

Si se observa la Figura 6.12, se puede ver cómo el lazo de control mantiene estable la glucemia muy próxima al objetivo y únicamente se ve alterada cuando sucede una ingesta. La ingesta provoca casi de manera inmediata una subida que después compensa el lazo de control mediante modificaciones de la insulina basal administrada al paciente. Las actuaciones del lazo de control se pueden observar como pequeños aumentos de insulina en la subfigura *Insulin*. Por lo demás, las ingestas siguen siendo las mismas que en las simulaciones anteriores y los bolos asociadas a estas siguen siendo calculados con los mismos parámetros: ratios de hidratos de carbono y tiempos de adelanto. Si se comparan las subfiguras *Insulin* de las Figuras 6.10 y 6.12 se puede apreciar que la única diferencia son los pequeños aumentos de basal que ocurren en lazo cerrado.

Es importante mencionar también que, aunque el caso de lazo abierto con correcciones post-prandiales y el caso de lazo cerrado parecen ofrecer resultados glucémicos similares, existen dos diferencias fundamentales:

1. El obtener los resultados aquí simulados para el caso de lazo abierto con correcciones post-prandiales está totalmente ligado a una perfecta caracterización de los parámetros del paciente en todo momento, y estos pueden derivar considerablemente con el tiempo. En el caso del lazo cerrado, las correcciones del lazo de control podrán compensar en mejor grado esas desviaciones de parámetros y permitirá tener mejores resultados glucémicos.

2. En el caso del lazo cerrado, en vez de una corrección después de cada comida, se está permitiendo al lazo de control corregir cada 5 minutos si es necesario. Y además estas correcciones son automáticas, con lo que la carga mental del paciente es mucho menor.

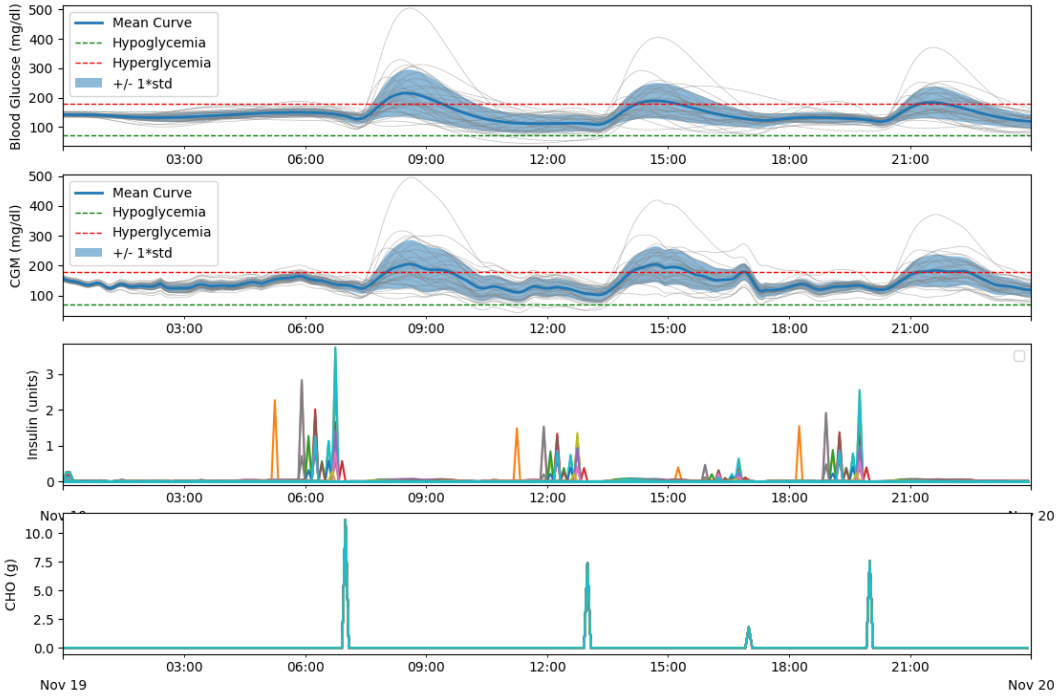


Figura 6.12: Resultados de la simulación en lazo cerrado

6.3.4 Simulaciones sobre el modelo de controlador con estimación de necesidades basales y bloqueo parcial de la infusión

Una vez se ha establecido los resultados de referencia mediante las simulaciones en lazo abierto y lazo cerrado con parámetros fijos del paciente, pasamos a utilizar la nueva algoritmia de detección de necesidades basales y a, en ciertos casos, simular un bloqueo parcial de la infusión.

6.3.4.1 Simulaciones en lazo cerrado sin bloqueo de insulina ni ingestas de hidratos de carbono (Batch 1)

El primer *batch* de simulaciones se realiza sin simular ningún tipo de bloqueo de insulina ni ninguna ingesta de hidratos de carbono. El objetivo de esta prueba es demostrar que, mediante el algoritmo de detección de necesidades basales, el controlador es capaz de mantener estable al paciente. No se simulan ingestas todavía para evitar otros efectos que pudieran complicar el análisis de los resultados.

Se presentan los resultados de tiempo en rango en la Tabla 6.1. Se puede apreciar tanto en la Tabla 6.1 como en la Figura 6.13 que se mantiene a todos los pacientes dentro del rango objetivo estándar (70 - 180 mg/dl) el 100 % del tiempo. Lógicamente esto no quiere decir que el algoritmo final vaya a ofrecer tiempos en rango del 100 % sino que, en ausencia

de grandes perturbaciones a nivel glucémico (ingestas y ejercicio físico), el sistema es capaz de determinar las necesidades basales y mantener al paciente con pequeños cambios de infusión cada 5 minutos.

También se puede ver la influencia del ruido del MCG en la insulina dosificada. Por suerte, como el efecto de la insulina actúa como filtro paso bajo debido a su largo tiempo de acción y el sistema es muy garantista en cuanto a dosificación de insulina, se pueden corregir las fluctuaciones indeseadas a posteriori.

Se puede por tanto concluir, a partir de los resultados, que la determinación dinámica de necesidades basales de insulina es factible.

	BG <50	BG <70	70 <= BG <= 180	BG >180	BG >250
Niños (n=10)	0 %	0 %	100 %	0 %	0 %
Adolescentes (n=10)	0 %	0 %	100 %	0 %	0 %
Adultos (n=10)	0 %	0 %	100 %	0 %	0 %

Tabla 6.1: Resultados de tiempo en rango en el Batch 1 de simulaciones

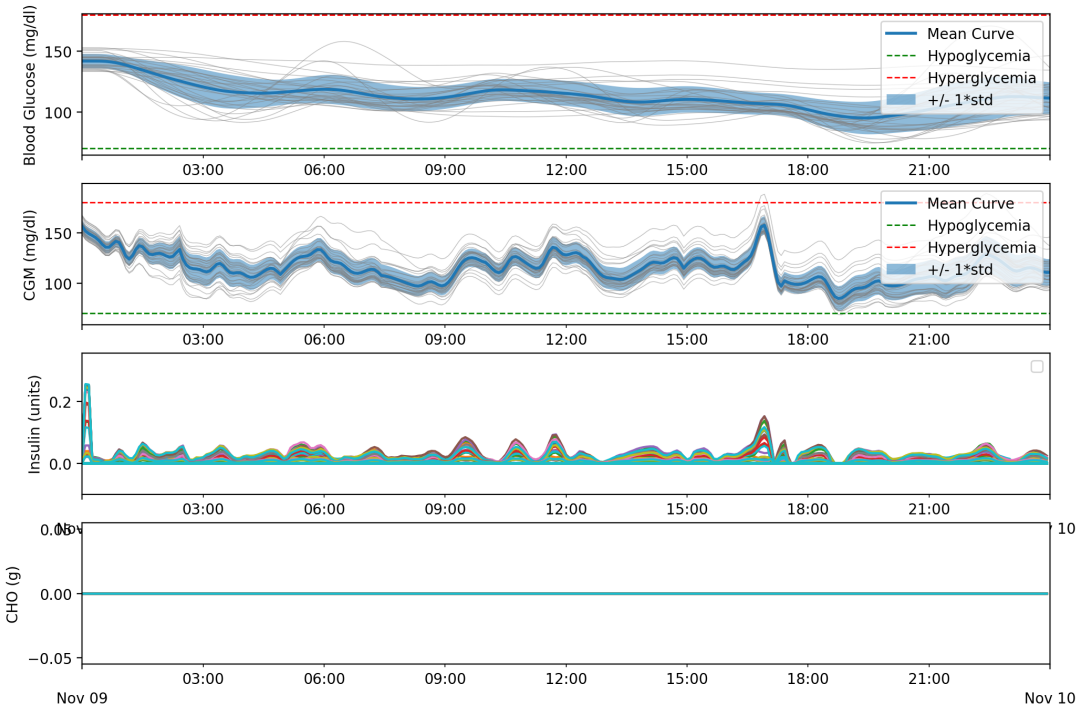


Figura 6.13: Resultados del Batch 1 de simulaciones

6.3.4.2 Simulaciones en lazo cerrado sin bloque de insulina y con ingestas de hidratos de carbono (Batch 2)

En el segundo *batch* de simulaciones se incluyen la ingestas de hidratos de carbono (como se puede ver en la subfigura *CHO* de la Figura 6.14).

La Tabla 6.2 muestra los porcentajes de tiempo en rango para estas simulaciones. Al introducir ingestas en la simulación se obtiene como resultado una disminución del tiempo en rango y un incremento del tiempo en hiperglucemia. Los resultados también muestran cómo el control es mejor cuanto mayor es la edad del paciente [87].

Los resultados esperables serían similares a los obtenidos en las pruebas en lazo cerrado con perfiles basales perfectamente ajustados (ver apartado 6.3.3.3), pero en realidad se mejoran ligeramente (como se puede apreciar comparando los resultados glucémicos de las Figuras 6.12 y 6.14).

La principal diferencia en los resultados se puede ver en cómo el valor medio de la glucemia después de las ingestas es ligeramente más próximo al objetivo glucémico (100 mg/dl) en el caso de tener ajuste dinámico de necesidades basales que en el caso de perfil basal prefijado. Es probable que el ajuste de necesidades basales hecho para el caso de los apartados 6.3.3.1, 6.3.3.2 y 6.3.3.3 no fuera muy preciso y por eso el ajuste dinámico obtiene resultados ligeramente más óptimos.

Este simulador no ofrece la posibilidad de simular ingestas de largo tiempo de absorción (como pueden ser alimentos con alto contenido en grasas) y por tanto no se han simulado pero, en el caso de que fuera posible, esa situación podría equivaler a una elevación temporal de las necesidades basales del paciente. De este razonamiento se deriva que, mediante la determinación dinámica de las necesidades basales, el algoritmo podrá obtener una mejor respuesta ante este tipo de ingestas.

Se puede concluir por tanto que, el uso de la determinación dinámica de necesidades basales no sólo no empeora el control, sino que ayuda a compensar posibles variaciones sobre lo programado por el paciente en su bomba de insulina.

	BG <50	BG <70	70 <= BG <= 180	BG >180	BG >250
Niños (n=10)	0 %	0 %	82 %	18 %	6 %
Adolescentes (n=10)	0 %	0 %	92 %	8 %	0 %
Adultos (n=10)	0 %	0 %	96 %	4 %	0 %

Tabla 6.2: Resultados de tiempo en rango en el Batch 2 de simulaciones

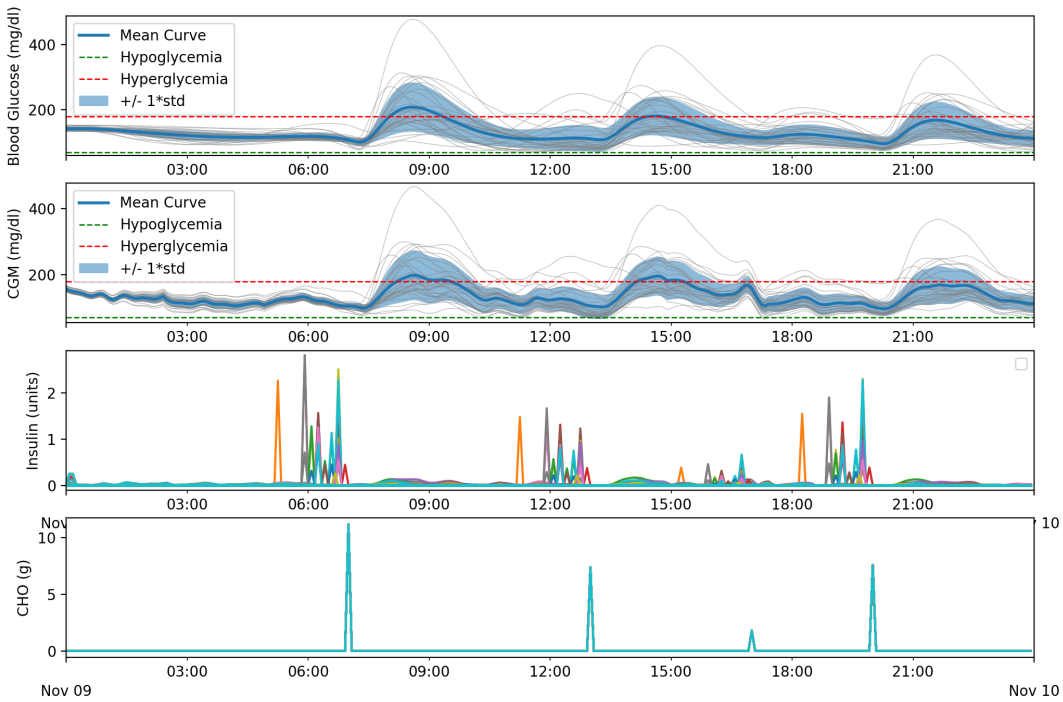


Figura 6.14: Resultados del Batch 2 de simulaciones

6.3.4.3 Simulaciones en lazo cerrado con bloqueo de insulina y sin ingestas de hidratos de carbono (Batch 3)

El tercer *batch* de simulaciones se ejecuta en las mismas condiciones que el *batch* 1 pero incluyendo un bloqueo parcial de la infusión a partir de las 12:00 horas. El objetivo con esta prueba es demostrar que el sistema no sólo es capaz de determinar las necesidades basales desde el comienzo de la simulación sino que, además, puede adaptarse con rapidez a cambios bruscos en las necesidades basales de insulina del paciente.

Para ello, se eliminan las ingestas de hidratos de carbono, en un intento de eliminar factores que compliquen el análisis del resto de parámetros, y se aplica un bloqueo de la infusión a partir de las 12:00 horas. De esta manera, durante las primeras 12 horas de simulación las necesidades basales serán las mismas que en el *batch* 1 y, una vez transcurridas, se verán incrementadas.

Se ha optado por simular un bloqueo proporcional a las necesidades programadas del paciente y no un bloqueo de una cantidad absoluta de unidades de insulina por hora debido a la gran dispersión de necesidades basales de insulina entre pacientes [88]: los niños suelen tener unas necesidades basales mucho más bajas que los adultos y, si se utilizara una cantidad de bloqueo fija para todos los pacientes, existiría un efecto no proporcional entre edades. Una misma cierta cantidad de bloqueo puede ser apropiada para un niño y casi inapreciable para un adulto, o apropiada para un adulto y totalmente

excesiva para un niño. Es por este motivo que se ha tomado la decisión de bloquear una cantidad de insulina equivalente a la mitad de las necesidades basales programadas para cada paciente. Esta programación a la que se hace referencia es la realizada en el fichero `AP_params.CSV` como `Necesidades basales` y descrita en el apartado 6.2.2.3. Este bloqueo equivale a decir que, a partir de las 12:00, las necesidades basales de insulina de cada paciente se ven incrementadas hasta el 150 % de lo que estuviera programado.

	BG <50	BG <70	70 <= BG <= 180	BG >180	BG >250
Niños (n=10)	0 %	0 %	100 %	0 %	0 %
Adolescentes (n=10)	0 %	0 %	100 %	0 %	0 %
Adultos (n=10)	0 %	0 %	100 %	0 %	0 %

Tabla 6.3: Resultados de tiempo en rango en el Batch 3 de simulaciones

Se obtiene un 100 % de tiempo en rango en todos los casos como se puede apreciar en la Tabla 6.3. Al igual que sucede en el caso del *Batch* 1, este 100 % de tiempo en rango únicamente indica que, en ausencia de grandes alteraciones en el control, este sistema es capaz de determinar dinámicamente las necesidades basales de insulina y, además, es capaz de adaptarse rápidamente a cambios en situaciones en las que dichas necesidades no sean constantes.

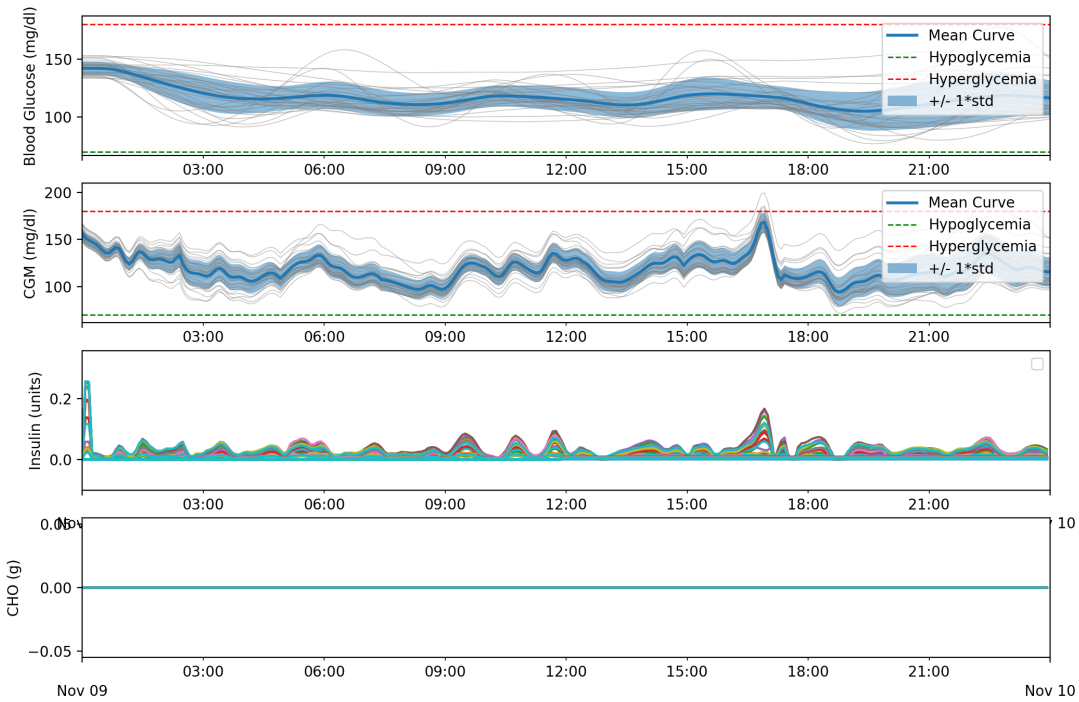


Figura 6.15: Resultados del Batch 3 de simulaciones

Si se comparan las evoluciones glucémicas en las Figuras 6.13 y 6.15 veremos que el

resultado es casi idéntico. Pero, aún existiendo un aumento de dosificación de insulina en la segunda mitad de la simulación del *batch* 3 con respecto al primero, no es sencillo ver el incremento de dosificación de insulina en la subfigura *Insulin* de la Figura 6.15. Es por ello que se ha generado la Tabla 6.4 con los consumos de insulina totales desglosados por paciente, *batch* de simulación y rango de tiempos de simulación.

En la Tabla 6.4 se puede ver el detalle de cuanta insulina total diaria (TDD - *Total Daily Dose*) ha usado cada paciente por cada *batch* de simulación (columnas "`TDD`"), cuanta insulina fue utilizada de 00:00 a 12:00 (columnas "`TDD de 0 a 12`") y cuanta fue utilizada de 12:00 en adelante (columnas "`TDD desde 12`"). Como parámetro añadido, se puede consultar cuanta insulina en términos absolutos fue bloqueada en el *batch* 3 de 12:00 en adelante (columna "`BLOQUEO`").

PACIENTE	BATCH 1 DE SIMULACIÓN			BATCH 3 DE SIMULACIÓN				INCREMENTO
	TDD	TDD de 0 a 12	TDD desde 12	TDD	TDD de 0 a 12	TDD desde 12	BLOQUEO	TDD desde 12
adolescent#001	33,36	17,39	15,98	35,71	17,39	18,32	3,62	2,34
adolescent#002	29,62	14,78	14,84	30,80	14,78	16,03	3,62	1,18
adolescent#003	17,65	9,47	8,18	19,35	9,47	9,88	2,24	1,70
adolescent#004	25,64	13,47	12,17	28,04	13,47	14,57	3,02	2,41
adolescent#005	22,84	11,90	10,94	25,12	11,90	13,22	3,02	2,41
adolescent#006	31,78	15,57	16,20	34,03	15,57	18,46	3,62	2,26
adolescent#007	20,54	10,00	10,55	21,02	10,00	11,03	3,02	0,48
adolescent#008	20,09	9,79	10,30	20,44	9,79	10,65	3,02	0,35
adolescent#009	17,88	8,60	9,28	18,85	8,60	10,25	2,42	0,97
adolescent#010	21,35	11,04	10,32	23,24	11,04	12,21	2,90	1,89
adult#001	41,13	21,69	19,44	44,36	21,69	22,67	4,83	3,23
adult#002	45,42	23,67	21,75	48,44	23,67	24,77	4,83	3,02
adult#003	44,14	22,97	21,17	47,69	22,97	24,72	4,83	3,54
adult#004	26,49	13,88	12,61	28,78	13,88	14,90	3,02	2,29
adult#005	46,79	24,05	22,74	50,18	24,05	26,13	4,83	3,39
adult#006	44,53	24,87	19,66	49,00	24,87	24,13	6,04	4,47
adult#007	36,20	19,07	17,13	39,85	19,07	20,78	4,53	3,65
adult#008	35,98	18,78	17,19	38,58	18,78	19,80	3,63	2,60
adult#009	41,90	22,39	19,51	44,50	22,39	22,11	3,62	2,60
adult#010	43,22	22,42	20,80	45,87	22,42	23,45	3,63	2,65
child#001	10,22	5,32	4,90	11,67	5,32	6,35	1,81	1,45
child#002	10,37	5,32	4,90	11,67	5,32	6,35	1,81	1,45
child#003	7,67	3,96	3,71	8,49	3,96	4,54	0,97	0,83
child#004	12,19	6,31	5,89	14,32	6,31	8,01	2,42	2,13
child#005	13,85	6,99	6,86	15,08	6,99	8,09	1,63	1,23
child#006	11,58	5,97	5,61	12,72	5,97	6,75	1,39	1,14
child#007	13,90	6,92	6,98	14,81	6,92	7,89	1,51	0,91
child#008	10,81	5,73	5,08	11,84	5,73	6,11	1,51	1,02
child#009	11,10	5,76	5,34	12,95	5,76	7,19	2,42	1,85
child#010	11,71	5,97	5,74	12,72	5,97	6,74	1,45	1,00

Tabla 6.4: Comparación de los Batches 1 y 3 en términos de TDD

Tal y como era de esperar, se puede ver en la Tabla 6.4 que las columnas `TDD de 0 a 12` en ambos *batches* de simulación ofrecen los mismos resultados, pues las necesidades

basales de insulina no se ven alteradas en ese tramo horario. En cambio, se puede apreciar un ligero aumento de TDD en las columnas `TDD desde 12` del *batch* 1 al *batch* 3. Este incremento de TDD está provocado por el bloqueo de insulina que se produce en el *batch* 3 a partir de las 12:00 (y que no ocurre en el *batch* 1). Únicamente el *batch* 3 fue simulado realizando un bloqueo parcial de insulina, motivo por el cual solo dicho *batch* dispone de una columna indicando la insulina bloqueada para cada paciente (columna `BLOQUEO`). Al igual que sucede con las columnas `TDD desde 12`, se puede apreciar un aumento de TDD entre el *batch* 1 y el 3 en las columnas `TDD` ya que estas columnas son el resultado de sumar las otras dos TDD parciales.

Se puede ver cláramente cómo todos los pacientes requirieron más insulina en la simulación del *batch* 3 y, aun así, obtuvieron los mismos resultados glucémicos.

También es importante hacer notar la correlación existente entre las cantidades de insulina bloqueada y el incremento de TDD (*Total Daily Dose*) desde las 12:00. Las cifras no son exactamente iguales, siendo el incremento en TDD ligeramente inferior al bloqueo, existiendo una correlación de ambas cifras muy clara. El motivo por el que el incremento de TDD es inferior a la cantidad de insulina bloqueada se asocia al hecho de que durante el cambio de necesidades basales el controlador debe adaptarse al cambio. En ese cambio la glucemia del paciente también aumenta ligeramente y con ello el objetivo glucémico. El llevar al paciente al objetivo deseado lleva al controlador algo más de tiempo, para ser garantista con la infusión de insulina, y eso provoca un uso ligeramente inferior de insulina al del bloqueo.

La Figura 6.16 muestra de manera gráfica los bloqueos de insulina y el incremento de TDD a partir de las 12:00 para todos los pacientes simulados en el *batch* 3. El índice del eje X corresponde con el orden en el que aparecen los pacientes en la Tabla 6.4. Por ejemplo, el paciente con índice 15, que sufrió un bloqueo de 6,04 unidades, corresponde con el paciente `adult#006`. El anterior, `adult#005`, sufrió un bloqueo de 4,83 unidades y aparece en la Figura 6.16 justo con el índice anterior (14). Tal y como se ha explicado, se puede ver gráficamente que existe una correlación entre el bloqueo producido y el incremento en insulina utilizada por el controlador, quedando siempre la insulina utilizada por debajo del bloqueo.

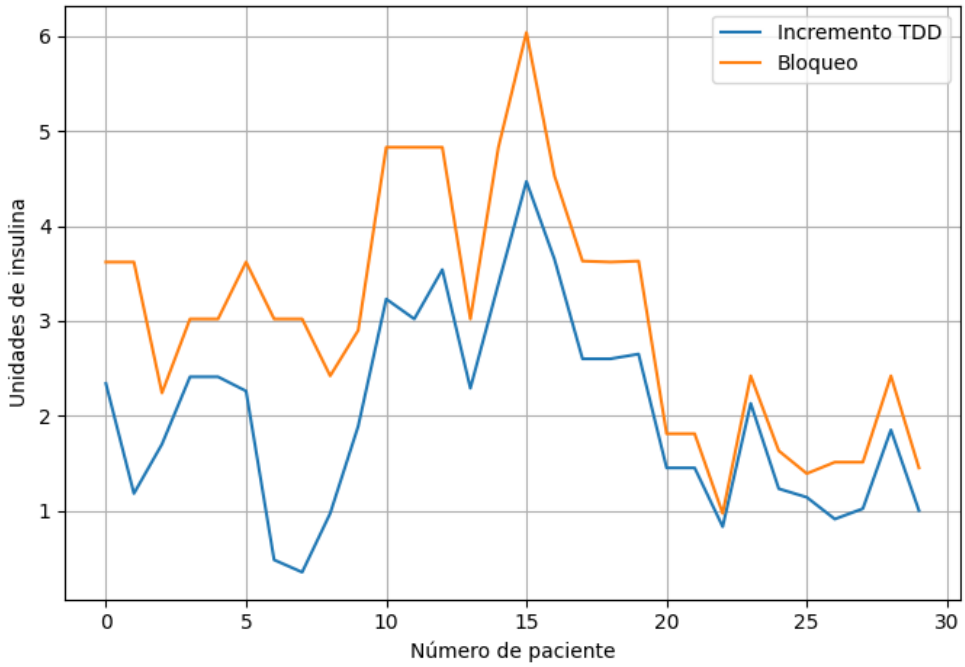


Figura 6.16: Comparación entre insulina bloqueada y TDD en batch 3

Como resultado final de esta prueba obtenemos que el algoritmo es capaz de adaptarse a cambios en las necesidades basales de insulina del paciente y que el control casi no se ve afectado.

6.3.4.4 Simulaciones en lazo cerrado con bloqueo de insulina y con ingestas de hidratos de carbono (Batch 4)

El último *batch* de simulaciones intenta replicar una situación más próxima a la realidad. En esta simulación, partiendo de la simulación del *batch* 3, se incluyen las ingestas de hidratos de carbono.

Lo que se puede apreciar, comparando las Tablas 6.2 y 6.5, es que los tiempos en rango son casi idénticos (superiores al 90 % en todos los casos) y que, a simple vista sobre las Figuras 6.14 y 6.17, el control glucémico es equivalente a pesar de haber incluido un bloqueo parcial de la infusión de insulina en el *batch* 4.

	BG <50	BG <70	70 <= BG <= 180	BG >180	BG >250
Niños (n=10)	0 %	0 %	79 %	21 %	6 %
Adolescentes (n=10)	0 %	0 %	91 %	9 %	0 %
Adultos (n=10)	0 %	0 %	96 %	4 %	0 %

Tabla 6.5: Resultados de tiempo en rango en el Batch 4 de simulaciones

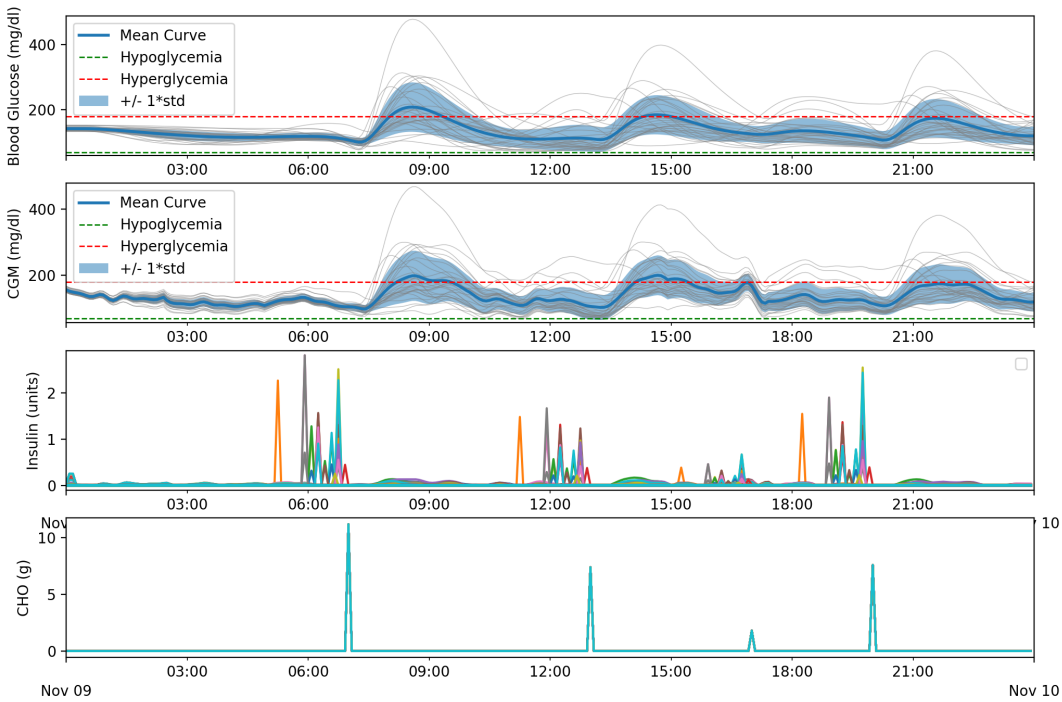


Figura 6.17: Resultados del Batch 4 de simulaciones

Si se comparan las Figuras 6.14 y 6.17, es complicado encontrar diferencias apreciables. Ambos *batches* de simulación utilizan las mismas ingestas, a las mismas horas, y es por ello que la subfigura *CHO* y los bolos correspondientes a las ingestas en la subfigura *Insulin* son exactamente iguales. Todas las subfiguras son exactamente iguales hasta las 12:00 que es cuando se produce en bloqueo de insulina en el *batch* 3. Es a partir de esa hora cuando se puede ver que, en las subfiguras *Blood Glucose* y *CGM*, algunos pacientes incrementan ligeramente sus glucemias (sobre las 18:00 por ejemplo) y también aumenta ligeramente el área que representa la desviación estándar de las glucemias de los pacientes. Se debería también apreciar un aumento en la insulina dosificada como basal a partir de las 12:00 en la subfigura *Insulin*, pero como los bolos son proporcionalmente mucho mayores que el incremento de basal, este incremento es inapreciable en la figura (motivo por el cual se ofrecen más datos en la Tabla 6.5).

Como conclusión de esta prueba obtenemos que el algoritmo propuesto es capaz de determinar las necesidades basales de insulina del paciente y ofrece resultados óptimos de control incluyendo todos los efectos disponibles en el simulador.

6.4 Entorno de experimentación real

El algoritmo simulado tiene la estructura perfecta para ser portado a la plataforma *wearable* descrita en el Capítulo 3. Haciendo uso de las funcionalidades implementadas en el *firmware* base del sistema, se re-codifica el algoritmo en C de manera que pueda ser ejecutado directamente por la plataforma.

Una vez listo el sistema, haciendo uso de una bomba de insulina compatible y un sensor MCG también compatible, se realizan ejercicios de auto-experimentación en vida real.

6.4.1 Equipos utilizados

Los equipos utilizados durante la experimentación son:

- Módulo plataforma *wearable* alimentado con una batería LiPo de 640mAh y cerrado en una caja para facilitar su transporte.
- Bomba de insulina Medtronic Paradigm Veo compatible.
- Sensor Medtronic Enlite con transmisor Medtronic Minilink.
- *Smartphone* Android con software xDrip para monitorizar y registrar datos de la plataforma.
- Insulina Humalog

6.4.2 Estrategia de experimentación

El dar el paso a la auto-experimentación no es sencillo y existen muchos pasos y fases que hay que realizar antes de permitir que un sistema electrónico tome decisiones de terapia sobre un paciente real.

El primer paso que se debe dar es garantizar la estabilidad de la plataforma *wearable* en si misma. El objetivo en esta fase es conseguir un sistema electrónico que se pueda mantener funcionando las 24 horas del día sin que existan fallos (*hardware o firmware*) y no requiera del paciente para su correcto funcionamiento. En esta primera fase, el autor de la tésis llevó el módulo registrando datos de la bomba de insulina y del MCG durante varias semanas, sin tomar decisión alguna sobre la terapia, hasta que se obtuvo este nivel de confianza/disponibilidad.

El segundo paso es simular el código del lazo de control y comprobar que, ante situaciones de riesgo, las protecciones se activan. Utilizando el código portado a C, se generaron simulaciones en las que era necesario que las siguientes protecciones fueran puestas a prueba:

- Supensión por hipoglucemia

- Límite de actuación por basal máxima
- Límite de actuación por basal mínima

Como tercer paso, sin dejar que el sistema actúe sobre la bomba de insulina, se comienza dejando el que lazo de control haga sus cálculos y estos queden registrados por el smartphone en Nightscout. El autor estuvo varias semanas analizando los resultados para comprobar que eran coherentes con el algoritmo.

El cuarto paso fue dejar al algoritmo operar y actuar sobre la bomba de insulina pero bajo las siguientes restricciones:

1. Únicamente se permite al lazo actuar mientras el paciente está consciente y puede prestar atención a las decisiones tomadas.
2. El objetivo glucémico deseado se configura suficientemente alto como para evitar posibles problemas.
3. El límite de basal máxima se configuró lo suficientemente bajo como para que en caso de error no supusiera ningún problema.

El autor estuvo un par de semanas validando el funcionamiento de esta manera hasta que se comprobó que no existían problemas.

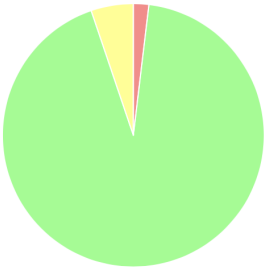
La quinta fase consistió en aumentar la agresividad del lazo de control, subiendo el límite de basal máxima, hasta un nivel que se podría denominar como "normal" y se configuraron objetivos glucémicos óptimos. Este nivel de agresividad permite corregir hiperglucemias en tiempos similares a un control manual. Todavía se mantiene el lazo activo únicamente mientras el paciente está consciente, aunque el nivel de atención se reduce considerablemente dado el grado de confianza obtenido.

La sexta y última fase implica dejar actuar al lazo de control las 24 horas del día con monitorización puntual.

6.5 Resultados de la experimentación

En este apartado se presentan los resultados de la experimentación con el sistema en la última fase de pruebas (Fase sexta). En esta fase ya se hace uso del sistema durante las 24 horas del día y con todos los parámetros ajustados al paciente. El sistema de control en lazo cerrado propuesto permite resultados tan óptimos que el autor de esta tesis todavía sigue utilizándolo.

Distribución de glucemias (90 Total de días)



Intervalo	% de valores	N° de valores	Media	Mediana	Desviación estándar	Estimación de HbA1c*
Bajo (<70):	1.9%	459	64.7	66.0	4.4	
Normal:	93.0%	22383	111.7	108.0	24.4	
Alto (>=180):	5.1%	1259	207.7	202.0	24.4	
General:		24457	115.8	109.0	33.0	5.7%$_{DCCT}$ | 38$_{IFCC}$

Variación media total diaria	Tiempo fluctuando (>5 mg/dl/5m)	Tiempo fluctuando rápido (>10 mg/dl/5m)
735.39 mg/dl	10.0%	2.0%
Variación media total por horas	**GVI**	**PGS**
30.64 mg/dl	1.18	9.5

Figura 6.18: Resultados de 3 meses de experimentación - Distribución

Las Figuras 6.18 y 6.19 son parte de los informes ofrecidos por la herramienta Nightscout [7] y han sido generados con los datos almacenados de los últimos 3 meses durante el último periodo de experimentación. La Figura 6.18 muestra la distribución de valores de glucemia que se ha obtenido en dicho periodo. Se muestra en color verde los valores que se encontraron dentro del rango objetivo (70-180 mg/dl), en rojo los que se encontraron por encima y en amarillo los que se encontraron por debajo. La columna “ % de valores” nos ofrece, para cada intervalo de datos, el porcentaje del total de valores que se encontraron en dicho rango. En otras palabras, podemos comprobar que el tiempo en rango durante ese periodo fue de un 93 %. También se nos ofrece el número de valores (o valores del sensor MCG) que se encontraron dentro de cada intervalo, así como sus medias, medianas y desviaciones estándar respectivas. La fila “General” nos ofrece las mismas estadísticas pero para el conjunto total de valores. Se puede comprobar en esta fila que la media y la mediana globales de los datos se encuentran muy próximas a 110 mg/dl (dentro de rango objetivo) y que la desviación estándar indica que la glucemia se encuentra normalmente dentro del rango.

La prueba clínica que ha servido desde hace años para evaluar la calidad del control glucémico en diabetes tipo 1 es la medida de la denominada Hemoglobina Glicosilada (HbA1c). Hay estudios que relacionan el valor de la hemoglobina glicosilada con el valor medio de glucemia durante los últimos 3 meses [42]. Según el *National Institute of Diabetes and Digestive and Kidney Diseases* [89], se considera una HbA1c de persona sana aquella que está por debajo de 5.7 %, pre-diabetes cuando se encuentra entre 5.7 % y 6.4 % y diabetes cuando se encuentra por encima de 6.4 %. En este caso, Nightscout nos ofrece la estimación de hemoglobina glicosilada correspondiente a los datos analizados que, en este caso, es de 5.7 %. Estos resultados, de acuerdo al criterio anteriormente descrito, se encontrarían al límite de lo aceptable para una persona sin diabetes tipo 1.

También nos ofrece las estadísticas de variabilidad en forma de variación glucémica media total diaria y los porcentajes de tiempo que le paciente se encuentra con tasas de

cambio por encima de 5 mg/dl/5min y por encima de 10 mg/dl/5min. Cuanto menor sean estos valores, mayor estabilidad y mejor control.

Las medidas de GVI y PGS responden a unas métricas no estándar creadas por la compañía Dexcom [90]. Según la documentación de Dexcom, estos parámetros deben ser interpretados siguiendo las siguientes reglas:

- GVI
  - De 1.0 a 1.2 ->Variabilidad baja (no-diabético)
  - De 1.2 a 1.5 ->Variabilidad media
  - Superior a 1.5 ->Variabilidad alta
- PGS
  - Inferior a 35 ->Estado glucémico excelente (no-diabético)
  - Entre 35 y 100 ->Estado glucémico bueno (diabético)
  - Entre 100 y 150 ->Estado glucémico pobre (diabético)
  - Superior a 150 ->Estado glucémico muy pobre (diabético)

Como se puede apreciar en la Figura 6.18, se han obtenido un GVI de 1.18 y un PGS de 9.5. Ambos resultados, según su interpretación, corresponden con resultados ”no-diabéticos”.

La Figura 6.19 nos muestra los percentiles de los últimos tres meses organizados según la hora del día. Este tipo de gráfica es especialmente útil para detectar periodos del día en los que se encuentran problemas de manera más o menos habitual. La gráfica muestra los límites del rango objetivo como una línea de color amarillo (límite superior) y otra de color rojo (límite inferior). La mediana de los valores está representada por una línea sólida de color negro y se añaden dos regiones de colores violeta que representan los percentiles 25 %/75 % y 10 %/90 %. Aunque la mediana es más o menos constante y cercana al objetivo programado (100mg/dl), se observa que hay mayor variabilidad a las pocas horas después de las ingestas. En cualquier caso, esta gráfica muestra que el paciente pasa la mayor parte del tiempo dentro del rango objetivo (como también mostraban las estadísticas de la Figura 6.18). Es importante remarcar también cómo el lazo de control reduce la variabilidad de manera importante durante las noches gracias a la ausencia de actividad y de ingestas. A las 09:00 de la mañana se podría decir que los percentiles coinciden con la mediana.

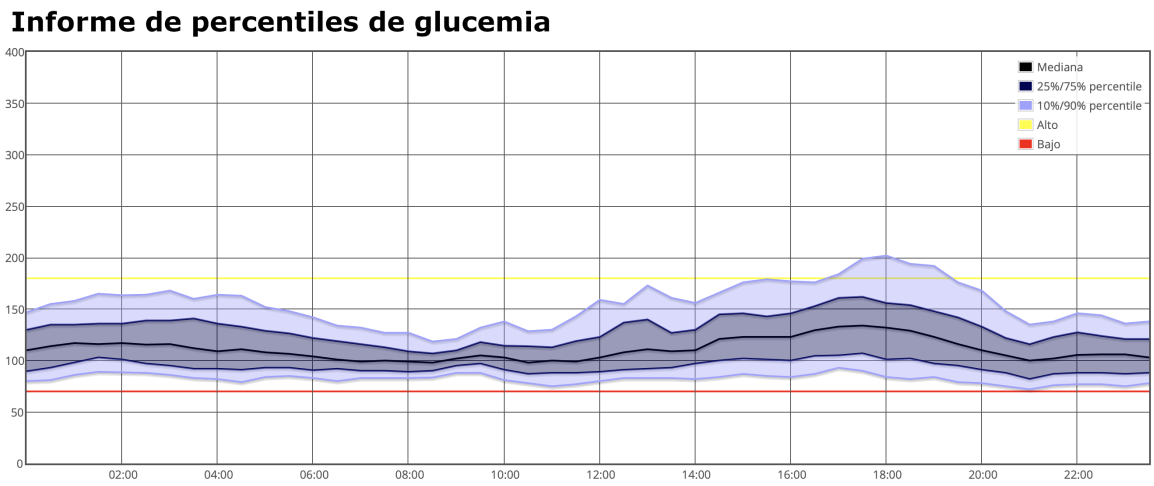


Figura 6.19: Resultados de 3 meses de experimentación - Percentiles

Como resumen de los resultados podemos destacar los siguientes puntos:

1. En los últimos 3 meses, el autor consiguió un tiempo en rango del 93 % con únicamente un 1.9 % de hipoglucemias. Estas cifras cuadran con las resultantes de la simulación del apartado 6.3.4.4 aun teniendo en cuenta que el resultado de la experimentación implica 3 meses de situaciones reales con comidas con alto y bajo índice glucémico, enfermedades, sets de infusión parcialmente obstruidos, ejercicio físico y estrés entre otros factores que no fueron simulados.

2. El objetivo glucémico deseado es de 100mg/dl y se obtienen una media de 115.8 mg/dl y una mediana de 109.0 mg/dl con una desviación estándar de 33 mg/dl.

3. La Hemoglobina glicosilada estimada (HbA1c), que es la medida que ha servido de referencia del control glucémico durante años, es de 5.7. En diabetes tipo 1 se considera como excelente control.

4. Los percentiles de la Figura 6.19 muestran una mediana que se aproxima al objetivo en horas donde no hay influencia de ingestas y ligeramente superior al objetivo justo durante y después de las ingestas con una distribución casi plana.

6.6 Conclusiones

En este capítulo se ha desarrollado un sistema de simulación del algoritmo propuesto y, junto con una estrategia de pruebas apropiada, se ha demostrado la viabilidad de la determinación automática en tiempo real de las necesidades basales de insulina del paciente.

Para poder llevar a cabo las simulaciones necesarias, se ha modificado el simulador simglucose añadiendo efectos de bloqueo de infusión de insulina. Gracias a esta modificación

es posible probar escenarios en los que las necesidades basales de insulina del paciente cambian con el tiempo.

Se ha comparado este algoritmo con el que hace uso de parámetros prefijados y no sólo no se ha empeorado el control, sino que mejora en cuanto las necesidades basales cambian.

Se ha establecido una estrategia de auto-experimentación que ha permitido al autor de la tesis evaluar el rendimiento real del algoritmo en una plataforma *wearable* y se ha comprobado que los resultados obtenidos durante 3 meses, no solo concuerdan con los arrojados por la simulación, sino que además corresponden con un control glucémico excelente.

Es importante remarcar que, a parte de los resultados numéricos, también existen resultados no medibles como la reducción de carga mental del paciente. El simple hecho de tener un sistema en lazo cerrado, aunque sea con parámetros fijos, descarga mucha presión del paciente: no existe manera alguna de que un paciente haga todos los cálculos que hace un sistemas de estas características 288 veces al día. Si a este sistema se le añade el hecho de que el paciente ya no necesita hacer ajustes de necesidades basales y que el sistema se adapta en tiempo real a los cambios, se reduce aún más la carga del paciente y se le acerca a lo que podríamos denominar una “vida más normal”.

Capítulo 7

Conclusiones y líneas futuras

En este capítulo se hace un resumen de las conclusiones de todos los trabajos realizados en esta tesis y se proponen futuras líneas de investigación.

7.1 Conclusiones

Esta tesis doctoral nace de un doble propósito:

1. Avanzar en el desarrollo de nuevas tecnologías en el campo de la diabetes.
2. Ayudar a los millones de personas que conviven con esta enfermedad a intentar llevar una vida un poco más normal y con menos complicaciones.

Como resultado de este estudio se ha obtenido un sistema completo que es capaz de controlar la glucemia del paciente de manera automática, eficiente y segura, obteniendo excelentes resultados en cuanto a tiempo en rango y, sobre todo, reduciendo la carga mental que supone el ajuste periódico de los perfiles basales de insulina.

Entendemos por carga mental todos aquellos procesos o eventos que suponen un esfuerzo por parte del paciente para mantener su buen estado de salud. Pudiera parecer que el paciente únicamente toma decisiones en ciertos momentos del día (como antes y/o después de las ingestas) pero en realidad es una tarea continua en la que muchos factores pueden tener repercusión durante muchas horas. Estos factores, en muchos casos, pueden requerir que el paciente necesite tomar acciones con mucho tiempo de adelanto y, en algunas de esas ocasiones, también pueden forzar al paciente a que ciertas acciones tengan que suceder de manera forzosa en el futuro. Un ejemplo de estas situaciones podría ser una intención de realizar ejercicio físico y después realizar una ingesta: se debe tener en cuenta el ejercicio físico con tiempo de antelación, evaluar la posterior reducción de necesidades basales post-ejercicio y calcular correctamente la insulina necesaria para la ingesta bajo dichas necesidades basales reducidas. Situaciones similares ocurren también en otros eventos en los que se ven afectadas las necesidades basales de insulina como

enfermedades, infusión de insulina parcialmente deteriorada, etc. En estos eventos el paciente debe estar permanentemente controlando que los resultados de su terapia a nivel glucémico concuerdan con lo esperado y, de no ser así, tomar decisiones lo antes posible. Gracias a este nuevo algoritmo, el sistema es capaz de realizar esta monitorización de manera continua y ajustarse a los cambios, permitiendo al paciente reducir el tiempo que dedica a monitorizar su enfermedad.

Los resultados de las simulaciones demuestran la viabilidad del uso de este nuevo sistema y, mediante los periodos de auto-experimentación, el autor de la tesis pudo comprobar que los resultados en un paciente real coinciden con los arrojados por las simulaciones. También se experimentó, aunque de una manera menos cuantificable, la considerable reducción de carga mental en el día a día gracias a la automatización de todos los procesos desarrollados en la tesis.

Para lograr el objetivo ha sido necesario desarrollar una plataforma hardware *wearable* nueva, capaz de operar de manera continuada y con un tamaño y peso suficientemente reducidos como para que un paciente pueda llevarla consigo sin que sea molesto. Dicha plataforma se ha diseñado y fabricado cumpliendo con dichos requisitos y, además, se ha realizado de manera que puede ser reutilizada para futuros estudios en este campo. Estas características hacen que esta plataforma sea perfecta no sólo para su uso real en pacientes sino para líneas futuras de investigación.

Sobre esta plataforma se ha creado un firmware básico de control que, maximizando la seguridad del paciente, es capaz de realizar las tareas y cálculos básicos que cualquier controlador pudiera necesitar. Este firmware se ha desarrollado teniendo en mente que debe servir de base para futuras líneas de investigación y, además, servir para las pruebas de los algoritmos desarrollados en esta tesis. Se ha conseguido una plataforma hardware + firmware capaz de desempeñar todas las tareas de comunicaciones con los diversos componentes del sistema y ofrecer una interfaz y librería de funciones reutilizables en futuros proyectos. El consumo de este firmware en la plataforma se ha comprobado que cumple con el requisito de mantener su operación más de 24 horas sin recargar, lo cual mantiene la denominación de *wearable*.

La integración de esta plataforma con Nightscout ofrece las capacidades necesarias para monitorizar y analizar los resultados del paciente, o del algoritmo que se esté diseñando, de manera cómoda y mediante herramientas *open-source* disponibles actualmente por la comunidad de pacientes con diabetes. Esta integración incrementa considerablemente el número de herramientas disponibles para ser utilizadas conjuntamente con la plataforma.

Una parte fundamental de los trabajos realizados en esta tesis es el algoritmo de estimación de necesidades basales de insulina y los ajustes dinámicos de parámetros como objetivos glucémicos o ISF. Estos nuevos algoritmos son la clave para haber conseguido los objetivos a nivel de control glucémico y la mejor adaptación de la terapia al paciente. Los cambios producidos en las necesidades basales de insulina por cambios hormonales,

ejercicio físico, enfermedades o ingestas de hidratos de carbono de bajo índice glucémico pueden ser asimilados a modificaciones temporales de las necesidades basales de insulina y, por tanto, el sistema se encarga de adaptarse automáticamente de estas situaciones. Estos algoritmos también suponen una reducción del número de parámetros a ajustar en la terapia, pues ya no es necesario ni estimar manualmente las necesidades basales para ajustar un perfil basal ni dejarlos programados en la bomba de insulina. El reducir el número de parámetros a ajustar tambien tiene la ventaja, posiblemente obvia, de que existen menos parámetos en los que se pueden cometer errores. Y, si se tiene en cuenta que este proceso de ajuste de perfiles basales puede llegar a ocurrir con varios meses entre ajustes, la relevancia de tener un buen perfil basal se incrementa. Es importante remarcar la importancia que tiene esta automatización en la vida del paciente: la reducción del nivel de intervención que es necesario acerca más al paciente a lo que sería una vida sin diabetes.

Para poder simular correctamente las situaciones en las que estos nuevos algoritmos muestran su potencial fue necesario modificar el simulador simglucose. Esta modificación permite la simulación de bloqueos parciales de la insulina que recibe el paciente y, con ello, se pueden simular las situaciones descritas anteriormente.

Cada parte del desarrollo de esta tesis ha sido un pequeño paso en el camino para conseguir los propósitos iniciales: avanzar en la tecnología y servir de ayuda a los pacientes. Una vez se observa de manera conjunta todos los resultados parciales de la tesis podemos concluir que se han logrado alcanzar dichos propósitos.

"Donde quiera que se ama el arte de la medicina se ama también a la humanidad." (Platón).

7.2 Líneas futuras

Son muchas las líneas de estudio que se pueden abordar en el futuro. La tecnología está comenzando a entrar en el mundo de la diabetes y hay amplio margen de mejora en muchos aspectos.

En el apartado 2.7 ya se enumeraron posibles áreas de mejora. Muchas de ellas corresponden a campos de estudio fuera de la ingeniería electrónica como la mejora en la respuesta de las insulinas o la mejora de los sistemas MCG en factores como retardo en las medidas, relación señal a ruido de los sensores o duración de los mismos. Pero, por otro lado, existen líneas de investigación muy interesantes que cubren desde algoritmos para eliminar las calibraciones de los sensores MCG mediante el sensado de otras variables biológicas y el uso de redes neuronales, hasta el diseño de nuevos algoritmos de control, de hormona única o multihormonales, que permitan una mayor automatización de todos los procesos necesarios para obtener un buen control glucémico.

Es necesario investigar sobre la determinación automática de más ajustes e incluso de la predicción de eventos. Por ejemplo, la predicción de ingestas o ejercicio físico para una preparación previa automática del paciente. O utilizar parámetros como la geolocalización para hacer modificaciones temporales de parámetros. Aunque la tecnología de los sensores MCG está más ligada al rendimiento bioquímico del sensor en el cuerpo del paciente, se pueden diseñar algoritmos que permitan reducir ruido en las medidas o incluso detectar cambios en el cuerpo y compensarlos matemáticamente. Algoritmos de Deep Learning o Machine Learning pueden ser aplicados a todos estos problemas.

También sería interesante el estudio de nuevos simuladores que permitieran escenarios de simulación más complejos y que permitieran simular situaciones como los bloqueos parciales simulados en esta tesis doctoral o ingestas de comida con distintos índices glucémicos.

7.3 Publicaciones derivadas de la tesis doctoral

De esta tesis doctoral se derivan las siguientes publicaciones en revistas indexadas JCR y congresos de investigación:

- J. Berián, I. Bravo, A. Gardel, J.L. Lázaro y S. Hernández, "A Wearable ClosedLoop Insulin Delivery System Based on LowPower SoCs", *Electronics*, 8 (6), p. 612, 2019. DOI:10.3390/electronics8060612 . Revista indexada JCR Q2
- J. Berián, A. Gardel, I. Bravo y C. Fernández. "Sistema para la adquisición de datos CGM", *JCER*, 2016.

También se han llevado a cabo las siguientes conferencias de divulgación relacionadas con la temática de la tesis:

- "Una mirada hacia el futuro. El futuro de las terapias de lazo cerrado". Diabetes Evolution. Coruña. 5/5/2019
- "Una mirada hacia el futuro. Próximos sensores, personalización y asa cerrada". Conferencia SED. Sevilla. 26/4/2019
- "Nightscout: una apuesta por el futuro". Universidad de Alcalá. 5/11/2018
- "Cerrando el asa". Hospital La Paz. Madrid. 19/10/2018
- "Experiencia de control en lazo cerrado: ¿Sistemas Do-It-Yourself?". Hospital Parc Tauli. Sabadell. 16/6/2018
- "Monitorización continua de glucemia y su posibilidad de visualización de forma remota a través de dispositivos". Hospital 12 de Octubre. Madrid. 31/5/2018

- "Estimación de niveles basales de insulina para el control en lazo cerrado de la glucemia en diabetes tipo 1". Universidad de Alcalá. Departamento de Electrónica. 13/11/2017

7.4 Disponibilidad de diseños y código fuente

Todos los diseños realizados durante esta tesis doctoral han sido publicados en un repositorio de libre acceso para todo aquel que quiera seguir trabajando en esta temática.

Los contenidos ofrecidos son:

1. **Demo** - Video time-lapse de un día completo de auto-experimentación del autor de la tesis con el sistema en pleno funcionamiento.
2. **Firmware** - Firmware descrito en la tesis incluyendo el controlador con ajustes dinámicos de necesidades basales.
3. **Platform** - Diseño eléctrico de la plataforma *wearable* junto con sus ficheros de fabricación.
4. **Simulation** - Simulador *simglucose* modificado para soportar simular bloqueos de infusión de insulina junto con el modelo del controlador desarrollado en la tesis para poder ser simulado.
5. **Uploader** - Código fuente de una aplicación para iOS capaz de recibir información de la plataforma *wearable* y subirla a Nightscout para su monitorización y análisis posterior.

El repositorio con todos estos contenidos se encuentra en la siguiente dirección: `https://github.com/jberian/closed_loop`

Bibliografía

[1] E. P. McCullagh, "Hormones-whence and whither: Presidential address," *The Journal of Clinical Endocrinology & Metabolism*, vol. 19, no. 12, p. 1571, December 1959.

[2] Medscape. Continuous subcutaneous insulin infusion: Intensive treatment, flexible lifestyle. [Online]. Available: https://www.medscape.org/viewarticle/460365_2

[3] A. Clemens, P. H. Chang, and R. W. Myers, "The development of biostator, a glucose controlled insulin infusion system (gciis)," *Hormone and Metabolic Research*, no. 7 (Supplement), pp. 23–33, 1977.

[4] Lilly. Insulina humalog. [Online]. Available: https://www.humalog.com

[5] H. Abe, M. Shichiri, R. Kawamori, T. Ichoh, K. Iwatani, M. Higuchi, K. Takagi, N. Okamoto, and T. Okudaira, "Artificial pancreas," US Patent US4 515 584A, Julio 6, 1983.

[6] L. Bernardi, "Wearable artificial pancreas," US Patent US5 298 022A, Mayo 29, 1989.

[7] N. Foundation. Nightscout. [Online]. Available: http://www.nightscout.info

[8] Openaps. [Online]. Available: https://www.openaps.org

[9] Loop. [Online]. Available: https://github.com/LoopKit/Loop

[10] Androidaps. [Online]. Available: https://github.com/MilosKozak/AndroidAPS/wiki/AndroidAPS

[11] Happ. [Online]. Available: https://github.com/timomer/HAPP

[12] S. Trevitt, S. Simpson, and A. Wood, "Artificial pancreas device systems for the closedloop control of type 1 diabetes," *Journal of Diabetes Science and Technology*, vol. 10, pp. 714–723, Mayo 2016. [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5038530/

[13] W. H. Organization. Diabetes facts. [Online]. Available: http://www.who.int/mediacentre/factsheets/fs312/en/

[14] A. D. Association. [Online]. Available: http://www.diabetes.org/

[15] E. Digitale. New research keeps diabetics safer during sleep. [Online]. Available: http://scopeblog.stanford.edu/2014/05/08/new-research-keeps-diabetics-safer-during-sleep

[16] J. J. Chamberlain, S. M. Hospital, and S. M. D. Center, "Continuous glucose monitoring systems: Categories and features," *American Diabetes Association - DiabetesPro*, 2018.

[17] A. E. Gold, K. M. Macleod, and B. M. Frier, "Frequency of severe hypoglycemia in patients with type i diabetes with impaired awareness of hypoglycemia," *Diabetes Care*, vol. 17, pp. 697–703, 1994.

[18] F. para la Diabetes. Tabla de raciones de hidratos de carbono. [Online]. Available: https://www.fundaciondiabetes.org/upload/publicaciones_ficheros/71/TABLAHC.pdf

[19] N. D. S. S. Australia. Carbohydrate counting fact sheet. [Online]. Available: https://www.ndss.com.au/about-diabetes/resources/find-a-resource/carbohydrate-counting-fact-sheet/

[20] D. Spero, BSN, and RN. New beta cell transplant appears to cure type 1 diabetes. [Online]. Available: https://www.diabetesselfmanagement.com/blog/new-beta-cell-transplant-appears-to-cure-type-1-diabetes/

[21] C. Colton and E. Avgoustiniatos, "The development of biostator, a glucose controlled insulin infusion system (gciis)," *Journal of Biomechanical Engineering*, vol. 2, no. 113, pp. 152–170, Mayo 1991. [Online]. Available: http://biomechanical.asmedigitalcollection.asme.org/article.aspx?articleid=1398578

[22] F. H. El-Khatib, S. J. Russel, D. M. Nathan, R. G. Sutherlin, and E. R. Damiano, "A bihormonal closed-loop artificial pancreas for type 1 diabetes," *Science Translational Medicine*, vol. 2, no. 27, p. 27ra27, Abril 2010. [Online]. Available: http://stm.sciencemag.org/content/2/27/27ra27.short

[23] R. Hovorka, K. Kumareswaran, J. Harris, J. M. Allen, D. Elleri, D. Xing, C. Kollman, M. Nodale, H. R. Murphy, D. B. Dunger, S. A. Amiel, S. R. Heller, M. E. Wilinska, and M. L. Evans, "Overnight closed loop insulin delivery (artificial pancreas) in adults with type 1 diabetes: crossover randomised controlled studies," *BMJ*, no. 342, Abril 2011. [Online]. Available: http://www.bmj.com/content/342/bmj.d1855

[24] S. D. Favero, D. Bruttomesso, F. D. Palma, G. Lanzola, R. Visentin, A. Filippi, R. Scotton, C. Toffanin, M. Messori, S. Scarpellini, P. Keith-Hynes, B. P. Kovatchev, J. H. DeVries, E. Renard, L. Magni, A. Avogaro, and C. Cobelli, "First use of model predictive control in outpatient wearable artificial pancreas," *Diabetes Care*, vol. 37, pp. 1212–1215, Mayo 2014. [Online]. Available: http://care.diabetesjournals.org/content/37/5/1212.short

[25] R. Nimri, I. Muller, E. Atlas, S. Miller, O. Kordonouri, N. Bratina, C. Tsioli, M. A. Stefanija, T. Danne, T. Battelino, and M. Phillip, "Night glucose control with md-logic artificial pancreas in home setting: a single blind, randomized crossover trial. interim analysis," *Pediatric Diabetes*, vol. 15, pp. 91–99, Agosto 2013. [Online]. Available: http://onlinelibrary.wiley.com/doi/10.1111/pedi.12071/full

[26] S. A. Weinzimer, G. M. Steil, K. L. Swan, J. Dziura, N. Kurtz, and W. V. Tamborlane, "Fully automated closed-loop insulin delivery versus semiautomated hybrid control in pediatric patients with type 1 diabetes using an artificial pancreas," *Diabetes Care*, vol. 31, pp. 934–939, Mayo 2008. [Online]. Available: http://care.diabetesjournals.org/content/31/5/934.short

[27] A. Haidar, L. Legault, V. Messier, T. M. Mitre, C. Leroux, and R. Rabasa-Lhoret, "Comparison of dual-hormone artificial pancreas, single-hormone artificial pancreas, and conventional insulin pump therapy for glycaemic control in patients with type 1 diabetes: an open-label randomised controller crossover trial," *The Lancet. Diabetes & Endocrinology*, vol. 3, pp. 17–26, Enero 2015. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S2213858714702268

[28] F. de Canete J., G.-P. S., and R.-D. J.C., "Artificial neural networks for closed loop control of in silico and ad hoc type 1 diabetes," *Computer Methods and Programs in Biomedicine*, vol. 106, pp. 55–66, Abril 2012. [Online]. Available: https://www.ncbi.nlm.nih.gov/pubmed/22178070

[29] M. Phillip, T. Battelino, E. Atlas, O. Kordonouri, N. Bratina, S. Miller, T. Biester, M. A. Stefanija, I. Muller, R. Nimri, and T. Danne, "Nocturnal glucose control with an artificial pancreas at a diabetes camp," *The New England Journal of Medicine*, vol. 368, pp. 824–833, Febrero 2013. [Online]. Available: https://www.ncbi.nlm.nih.gov/pubmed/23445093?access_num=23445093&link_type=MED&dopt=Abstract

[30] Medtronic. Medtronic minimed 670g insulin pump system. [Online]. Available: https://www.medtronicdiabetes.com/products/minimed-670g-insulin-pump-system

[31] T. Diabetes. Tandem t:slim x2 with control-iq. [Online]. Available: https://www.tandemdiabetes.com/products/t-slim-x2-insulin-pump/control-iq

[32] M. P. Stone, P. Agrawal, X. Chen, M. Liu, J. Shin, T. L. Cordero, and F. R. Kaufman, "Retrospective analysis of 3-month real-world glucosedata after the minimed 670g system commercial launch," *Diabetes Technology and Therapeutics*, vol. 20, pp. 1–4, 2018.

[33] S. A. Brown, B. P. Kovatchev, D. Raghinaru, J. W. Lum, B. A. Buckingham, Y. C. Kudva, L. M. Laffel, C. J. Levy, J. E. Pinsker, R. P. Wadwa, E. Dassau, and F. J. Doyle, "Six-month randomized, multicenter trial of closed-loop control in type 1 dia-

betes," *The New England Journal of Medicine*, vol. 381, pp. 1707–1717, Octubre 2019.

[34] J. Pickup and H. Keen, "Continuous subcutaneous insulin infusion at 25 years. evidence base for the expanding use of insulin pump therapy in type 1 diabetes," *Diabetes Care*, vol. 25, pp. 592–598, Marzo 2002. [Online]. Available: http://care.diabetesjournals.org/content/25/3/593

[35] B. W. Bequette, "A critical assessment of algorithms and challenges in the development of a closed-loop artificial pancreas," *Journal of Diabetes Technology & Therapeutics*, vol. 7, pp. 28–47, Febrero 2005. [Online]. Available: http://online.liebertpub.com/doi/abs/10.1089/dia.2005.7.28

[36] H. Lee, B. A. Buckingham, D. M. Wilson, and B. W. Bequette, "A closed-loop artificial pancreas using model predictive control and a sliding meal size estimator," *Journal of Diabetes Technology & Therapeutics*, vol. 3, pp. 1082–1090, Septiembre 2009. [Online]. Available: http://journals.sagepub.com/doi/abs/10.1177/193229680900300511

[37] Wikipedia. Model predictive control. [Online]. Available: https://en.wikipedia.org/wiki/Model_predictive_control

[38] ——. Pid controller. [Online]. Available: https://en.wikipedia.org/wiki/PID_controller

[39] D. Linkens and M. Chen, "Expert control systems i. concepts, characteristics and issues," *Engineering Applications of Artificial Intelligence*, vol. 8, pp. 413–421, 1995.

[40] Wikipedia. Artificial neural network. [Online]. Available: https://en.wikipedia.org/wiki/Artificial_neural_network

[41] A. K. El-Jabali, "Neural network modelling and control of type 1 diabetes mellitus," *Bioprocess and Biosystems Engineering*, vol. 27, pp. 75–9, Diciembre 2004. [Online]. Available: https://www.ncbi.nlm.nih.gov/pubmed/15578231

[42] D. M. Nathan, J. Kuenen, R. Borg, H. Zheng, D. Schoenfeld, R. J. Heine, and A.-D. A. G. A. S. Group, "Translating the a1c assay into estimated average glucose values," *Diabetes Care*, vol. 31, pp. 1473–8, Agosto 2008. [Online]. Available: https://www.ncbi.nlm.nih.gov/pubmed/18540046

[43] Wikipedia. Jdrf - juvenile diabetes research foundation. [Online]. Available: https://www.jdrf.org

[44] B. Biomedical. [Online]. Available: https://www.bigfootbiomedical.com

[45] Insulet. Insulet omnipod dash system. [Online]. Available: https://diatribe.org/insulet-omnipod-dash-system-to-add-touchscreen-and-bluetooth

[46] E. W. Kraegen, L. Campbell, Y. O. Chia, H. Meier, and L. Lazarus, "Control of blood glucose in diabetics using an artificial pancreas," *Internal Medicine Journal*, vol. 7, pp. 280–286, Junio 1977. [Online]. Available: http://onlinelibrary.wiley.com/doi/10.1111/j.1445-5994.1977.tb03687.x/full

[47] J. P. StegemannJohn, J. O'NeilClaudy, and J. P. Mullon, "Artificial pancreas," US Patent US6 023 009A, Febrero 23, 1996.

[48] J. Berian. Medtronic paradigm veo rf transceiver. github repository. [Online]. Available: https://github.com/jberian/mmcommander

[49] Diabeloop. Diabeloop. [Online]. Available: https://www.diabeloop.com

[50] F. J. Doyle, L. M. Huyett, J. B. Lee, H. C. Zisser, and E. Dassau, "Closed-loop artificial pancreas systems: Engineering the algorithms," *Diabetes Care*, vol. 37, pp. 1191–1197, Junio 2014. [Online]. Available: https://www.ncbi.nlm.nih.gov/pubmed/24757226

[51] Y. C. Kudva, R. E. Carter, C. Cobelli, R. Basu, and A. Basu, "Closed-loop artificial pancreas systems: Physiological input to enhance next-generation devices." *Diabetes Care*, vol. 37, pp. 1184–1190, Junio 2014. [Online]. Available: https://care.diabetesjournals.org/content/37/5/1184.long

[52] J. S. Freeman, "Insulin analog therapy: Improving the match with physiologic insulin secretion devices," *The Journal of the American Osteopathic Association*, vol. 109, pp. 26–36, Enero 2009. [Online]. Available: https://jaoa.org/article.aspx?articleid=2093701

[53] B. West. Medtronic paradigm veo commands. github repository. [Online]. Available: https://github.com/bewest/decoding-carelink/tree/master/decocare

[54] M. Diabetes. Medtronic enlite sensors. [Online]. Available: https://www.medtronicdiabetes.com/customer-support/sensors-and-transmitters-support

[55] T. Instruments. Texas instruments low-power soc cc111x datasheet. [Online]. Available: http://www.ti.com/lit/ds/symlink/cc1110-cc1111.pdf

[56] ——. Texas instruments low-power soc cc2540 datasheet. [Online]. Available: http://www.ti.com/lit/ds/symlink/cc2540.pdf

[57] ——. Texas instruments low-power soc cc2510 datasheet. [Online]. Available: http://www.ti.com/lit/ds/symlink/cc2510.pdf

[58] S. Black. xdrip - using your dexcom g4 with android. [Online]. Available: http://stephenblackwasalreadytaken.github.io/xDrip

[59] T. Instruments. Application note an043. [Online]. Available: https://www.ti.com/lit/an/swra117d/swra117d.pdf

[60] Wikipedia. Ism band. [Online]. Available: https://en.wikipedia.org/wiki/ISM_band

[61] ——. On-off keying modulation. [Online]. Available: https://en.wikipedia.org/wiki/On-off_keying

[62] T. Instruments. Bluetooth low energy software stack. [Online]. Available: http://www.ti.com/tool/BLE-STACK?DCMP=wbu-blestack&HQS=blestack

[63] M. Hermans, F. Nobels, and I. D. Leeuw, "Insulin lispro (humalog®), a novel fast-acting insulin analogue for the treatment of diabetes mellitus: Overview of pharmacological and clinical data," *Acta Clinica Belgica*, vol. 54, pp. 230–240, 1999.

[64] A. Brazeau, H. Mircescu, K. Desjardins, C. Leroux, I. Strychar, J. EkoÃ©, and R. Rabasa-Lhoret, "Carbohydrate counting accuracy and blood glucose variability in adults with type 1 diabetes," *Diabetes Research and Clinical Practice*, vol. 99, pp. 19–23, 2013.

[65] T. M. Wolever, D. J. Jenkins, A. L. Jenkins, and R. G. Josse, "The glycemic index: methodology and clinical implications," *The American Journal of Clinical Nutrition*, vol. 54, pp. 846–854, Noviembre 1991.

[66] G. Collier, A. McLean, and K. O'Dea, "Effect of co-ingestion of fat on the metabolic responses to slowly and rapidly absorbed carbohydrates," *Diabetologia*, vol. 26, 1984.

[67] H. Yki-JÃ¤rvinen, E. Helve, and V. A. Koivisto, "Hyperglycemia decreases glucose uptake in type 1 diabetes," *Diabetes*, vol. 36, pp. 892–896, Agosto 1987.

[68] W. WaldhÃ¤usl, "Circadian rhythms of insulin needs and actions," *Diabetes Research and Clinical Practice*, vol. 6, pp. S17–S24, Septiembre 1989. [Online]. Available: https://doi.org/10.1016/0168-8227(89)90074-0

[69] A. J. Roberts, G. P. Forlenza, D. Maahs, and C. E. Taplin, "Type 1 diabetes mellitus and exercise," *Diabetes and Exercise. Contemporary Diabetes*, 2018.

[70] P. Galassetti, D. Tate, R. A. Neil, A. Richardson, S.-Y. Leu, and S. N. Davis, "Effect of differing antecedent hypoglycemia on counterregulatory responses to exercise in type 1 diabetes," *American Journal of Physiology Endocrinology and Metabolism*, vol. 290, pp. E1109–E1117, Junio 2006. [Online]. Available: https://journals.physiology.org/doi/full/10.1152/ajpendo.00244.2005

[71] W. L. Clarke, D. J. Cox, L. A. Gender-Frederick, D. Julian, D. Schlundt, and W. Polonsky, "The relationship between nonroutine use of insulin, food and exercise and the occurrence of hypoglycemia in adults with iddm and varying degrees of hypoglycemic awareness and metabolic control," *The Diabetes Educator*, vol. 23, pp. 55–58, Enero 1997. [Online]. Available: http://tde.sagepub.com/content/23/1/55

[72] G. E. Sonnenberg, F. W. Kemmer, and M. Berger, "Exercise in type 1 (insulin-dependent) diabetic patients treated with continuous subcutaneous insulin infusion: prevention of exercise-induced hypoglycemia," *Diabetologia*, vol. 22, pp. 609–703, Junio 1990.

[73] L. Laffel, "Sick-day management in type 1 diabetes," *Endocrinology and Metabolism Clinics of North America*, vol. 29, pp. 707–723, Diciembre 2000.

[74] G. Schmelzeisen-Redeker, M. Schoemaker, H. Kirchsteiger, G. Freckmann, L. Herinemann, and L. del Re, "Time delay of cgm sensors: Relevance, causes and countermeasures," *Journal of Diabetes Science and Technology*, vol. 9, pp. 1006–1015, Agosto 2015.

[75] J. E. Gerich, "Insulin glargine: long-acting basal insulin analog for improved metabolic control," *Current Medical Research and Opinion*, vol. 20, pp. 31–37, 2004.

[76] T. Heise and L. Heinemann, "Rapid and long-acting analogues as an approach to improve insulin therapy: An evidence-based medicine assessment," *Current Pharmaceutical Design*, vol. 7, pp. 1303–1325, 2001.

[77] M. Lepore, S. Pampanelli, and C. F. et al, "Pharmacokinetics and pharmacodynamics of subcutaneous injection of long-acting human insulin analog glargine, nph insulin, and ultralente human insulin and continuous subcutaneous infusion of insulin lispro," *Diabetes*, vol. 49, pp. 2142–2148, Diciembre 2000.

[78] J. Walsh and R. Roberts, *Pumping Insulin.* Torrey Pines Pr, 2016.

[79] G. Scheiner and B. A. Boyer, "Characteristics of basal insulin requirements by age and gender in type-1 diabetes patients using insulin pump therapy," *Diabetes Research and Clinical Practice*, vol. 69, pp. 14–21, Julio 2005.

[80] R. J. Casson and L. D. Farmer, "Understanding and checking the assumptions of linear regression: a primer for medical researchers," *Clinical & Experimental Ophthalmology*, vol. 42, pp. 590–596, Abril 2014.

[81] J. Walsh, R. Roberts, and T. Bailey, "Guidelines for optimal bolus calculator settings in adults," *Journal of Diabetes Science and Technology*, vol. 5, pp. 129–135, Enero 2011.

[82] F. Belfiore, S. Iannello, and G. Volpicelli, "Insulin sensitivity indices calculated from basal and ogtt-induced insulin, glucose and ffa levels," *Molecular Genetics and Metabolism*, vol. 63, pp. 134–141, Febrero 1998.

[83] J. Xie. Simglucose v0.2.1 (2018). [Online]. Available: https://github.com/jxx123/simglucose

[84] C. D. Man, F. Micheletto, D. Lv, M. Breton, B. Kovatchev, and C. Cobelli, "The uva/padova type 1 diabetes simulator," *Journal of Diabetes Science and Technology*, vol. 8, pp. 26–34, Enero 2014.

[85] Wikipedia. Run to completion scheduling. [Online]. Available: https://en.wikipedia.org/wiki/Run_to_completion_scheduling

[86] A. D. Association, "Facilitating behaviour change and wellbeing to improve health outcomes: Standards of medical care in diabetes 2020," *Diabetes Care*, vol. 43, pp. S48–S65, 2020.

[87] A. Casu, L. G. Kanapka, N. C. Foster, I. B. Hirsch, L. M. Laffel, V. N. Shah, D. J. DeSalvo, S. K. Lyons, F. Vendrame, G. Aleppo, L. D. Mastrandrea, R. E. Pratley, M. R. Rickels, and A. L. Peters, "Characteristics of adult compared to childhood onset type 1 diabetes," *Diabetic Medicine*, Abril 2020.

[88] C. Snider. Let's talk about your insulin pump data. [Online]. Available: https://www.tidepool.org/blog/lets-talk-about-your-insulin-pump-data

[89] N. I. of Diabetes, Digestive, and K. Diseases. The a1c test & diabetes. [Online]. Available: https://www.niddk.nih.gov/health-information/diagnostic-tests/a1c-test?dkrd=/health-information/diabetes/overview/tests-diagnosis/a1c-test

[90] D. Mine. A new view of glycemic variability: How long is your line? [Online]. Available: https://web.archive.org/web/20160523152519/http://www.healthline.com/diabetesmine/a-new-view-of-glycemic-variability-how-long-is-your-line

Apéndice A

Detalles técnicos de la implementación de la plataforma *wearable*

A.1 Flags de control de potencia del procesador

En este apartado se detallan los *flags* que utiliza el proceso de control de potencia del procesador para determinar si existen tareas pendientes o, por contra, se puede pasar el procesador a modo de bajo consumo.

1. `startRFTxFlag`: Este flag indica la necesidad de enviar mensajes por Radio a la bomba de insulina. Además permite mantener al procesador activo mientras se espera después de la recepción o transmisión de otro mensaje, evitando así posibles colisiones en la transmisión.

2. `adjustPumpFreqFlag`: El sistema permite realizar un ajuste de la frecuencia de comunicaciones con la bomba de insulina. Este proceso se inicia bajo demanda del usuario mediante un comando por el interfaz BLE y se activa este *flag* durante su ejecución. El objetivo es buscar la frecuencia con la que se reciben datos de la bomba de insulina con mejor RSSI y, para ello, se manda un comando de *WAKE_UP* en cada frecuencia y se analiza la respuesta.

3. `sendFlag`: Indica la necesidad de enviar los resultados de la operación del lazo de control via Bluetooth a un posible *smartphone* que esté conectado. Este proceso permite subir los datos a la nube, ser almacenados y tener alarmas en el *smartphone*.

4. `queryPumpFlag`: Este flag indica la necesidad, por parte del algoritmo de control, de obtener los parámetros configurados en la bomba de insulina en cuanto a ajustes de terapia del paciente (perfiles basales, sensibilidades a la insulina, objetivos glucé-

micos,...). Estos parámetros son básicos para la operación del lazo de control y, en caso de no haberlos recibidos, el sistema debe permanecer activo hasta su obtención.

5. `rfMessageLength`: El número de bytes recibidos por Radio es un indicativo directo de que hay información a ser procesada.

6. `timeUpdatedFlag`: Debido al gran escalado que se necesita, se ha dividido en dos partes: una realizada mediantes las capacidades del propio *timer* y otra implementada en software mediante un contador. Cuando este contador alcanza el nivel de escalado buscado, se activa este flag para permitir la actualización de todas las variables de tiempo del sistema.

7. `bleRxFlag`: Este *flag* indica que se han recibido bytes por el interfaz serie (puente Bluetooth Low Energy) y deben ser tratados.

8. `bleMessageLength`: El número de bytes que quedan por ser transmitidos via serie (puente Bluetooth Low Energy) es un indicativo directo de que hay información a ser procesada. Si el procesador pasa a modo de bajo consumo antes de haber transmitido todos los bytes existentes en el buffer se perderá la información.

A.2 Temporizadores internos

Los temporizadores que controlan los procesos internos son los siguientes:

1. `rfOnTimer`: Este contador de tiempo se inicializa a un valor equivalente al tiempo que debe estar la radio encendida (cuando es oportuno) y va decreciendo hasta que se hace cero y se apaga la radio.

2. `txTimer`: Cuando se transmite un mensaje a la bomba de insulina, la bomba debe responder con datos o, al menos, un mensaje de aceptación. Cuando el sistema manda un mensaje a la bomba, inicializa este contador de tiempo como TimeOut y, si no se ha recibido respuesta cuando se hace cero, se indica internamente que el mensaje no ha sido recibido por la bomba de insulina. Esto da la oportunidad al controlador para que tome la acción que crea oportuna: reintentar o dar por inválida la transmisión.

3. `reCalTimer`: Mientras el interfaz de radio del SoC esté encendido, debe ser recalibrado cada minuto. Este contador de tiempo cuenta el tiempo que lleva la radio encendida y desencadena el proceso de calibración cada minuto.

4. `bleCommsWatchdogTimer`: Existen muchos motivos por los que una transmisión de datos via serie (puente BLE) puede quedar incompleta (fallos de transmisión, protocolo mal implementado, uso de diferentes versiones del protocolo...). Este contador de tiempo se inicializa a un valor predeterminado cada vez que se recibe un byte por el interfaz serie. Si el contador se hace cero se elimina cualquier byte que quede

almacenado en el buffer interno y se comunica el error. Este mecanismo permite el dejar un mensaje sin terminar o con errores y recuperarse para el siguiente dado que, de manera normal, existen grandes periodos de tiempo sin comunicaciones por este interfaz. Normalmente existen comunicaciones cada vez que se recibe un dato nuevo del MCG y/o el algoritmo de control toma alguna acción (aproximadamente 5 minutos).

5. `lookForMinilinkTimer`: El sistema permite tener ajustes finos de frecuencia para el MCG (que en nuestro caso es un Minilink de Medtronic) y para la bomba de insulina. Cuando se activa la radio para recepción de nuevos datos, el primer dispositivo en transmitir es el MCG. Es por esto que se ajusta la frecuencia de la radio a la del MCG y se activa este contador como TimeOut para la recepción de estos datos. Si este contador se hace cero, se asume que no se han recibido los datos del sensor y pasamos a esperar la recepción de datos de la bomba de insulina.

6. `lookForMySentryTimer`: El protocolo utilizado por la bomba de insulina para comunicar, entre otras cosas, el último par de datos del sensor y datos como la insulina restante o el voltaje de la batería se llama MySentry. La bomba de insulina, justo después de recibir los datos del MCG, envía sus datos utilizando este protocolo. Al igual que en el caso del MCG, una vez vence el TimeOut del Minilink o hemos recibido los datos, pasamos a ajustar la frecuencia para una mejor recepción de datos de la bomba e inicializamos este contador como TimeOut.

7. `timeCounterBolusSnooze`: Si el algoritmo lo cree oportuno, se da la posibilidad de medir el tiempo desde que se realizó el último bolo manual desde la bomba. Este mecanismo puede ser de utilidad para no tomar acciones automáticas hasta pasado un tiempo desde que comenzó la acción de la insulina del bolo (que suele ser una cantidad importante en comparación con los niveles basales) y/o la acción de la ingesta. No es obligatorio su uso.

8. `timeCounterPumpAwake`: El primer paso para comunicarse con la bomba de insulina es ″despertarla″. El proceso comienza mandando de manera continuada un mensaje concreto para que, en uno de los periodos de escucha periódicos de la bomba, la bomba deje activo su interfaz de radio. Una vez la bomba recibe este mensaje, contesta con una mensaje de aceptación y permanecerá con la radio activa unos segundos permitiendo una comunicación de manera más continuada. Para que el sistema sepa cuando es necesario ″despertar″ a la bomba de insulina, se utiliza este timer que cuenta el tiempo desde que se despertó por última vez. Si, cuando se quiere enviar un mensaje, este contador ya es cero entonces se procede a ″despertar″ a la bomba antes del envío del mensaje.

9. `timeCounterTxInhibit`: Es importante no comprometer las comunicaciones ″oficiales″ entre los dispositivos de Medtronic. Para ello, se establece un periodo

de tiempo (controlado por este timer) en el que nuestro sistema no podrá transmitir desde la recepción del último mensaje del MCG o la bomba de insulina. De esta manera garantizamos que nuestras comunicaciones siempre quedarán dentro de los periodos de silencio entre estos dispositivos.

A.3 Funciones de soporte al cómputo de parámetros

Una vez el sistema está actualizado con los últimos datos recibidos de todos los dispositivos, se pasa a la ejecución del algoritmo de control llamando a la función `void controlLoopDataUpdate (void)`. Dentro de esta función se puede implementar el algoritmo deseado haciendo uso de las siguientes funciones diseñadas para dar soporte al cálculo:

- `void setIOBvectors (float rate)`:
  Esta función permite el inicializar los *buffers* donde se almacenan los datos de insulina basal dosificada a un valor equivalante a haber tenido una basal de valor `rate` desde el comienzo del tiempo del histórico. Esto es util si, por ejemplo, se desea comenzar la ejecución como si se hubiera dosificado anteriormente los niveles basales y, de esta manera, partir de un estado de insulina activa nula. El parámetro de ritmo de infusión se establece en Unidades de insulina por hora (U/h).
- `float getDeliveredIOBBasal (float basal)`:
  Esta función permite obtener la insulina activa en el momento de su ejecución, únicamente teniendo en cuenta las aportaciones de insulina basales dosificadas previamente y tomando como necesidades basales las proporcionadas en el parámetro `basal` en unidades de insulina por hora (U/h). El resultado vendrá expresado en Unidades de Insulina.
- `float getDeliveredIOBBolus (void)`:
  Esta función permite obtener la insulina activa en el momento de su ejecución, únicamente teniendo en cuenta las aportaciones de insulina dosificadas previamente en calidad de bolo. El resultado vendrá expresado en Unidades de Insulina.
- `float getDeliveredIOB (float basal)`:
  Esta función permite obtener la insulina activa en el momento de su ejecución teniendo en cuenta toda la insulina dosificada previamente. El resultado vendrá expresado en Unidades de Insulina.
- `float getToBeDeliveredIOB (float basal)`:
  Esta función permite obtener la insulina activa que se obtiene únicamente como resultado de las programaciones de basales temporales a ejecutar en el futuro. Las necesidades basales se establecen en el parámetro `basal` en Unidades de insulina por hora (U/h). El resultado vendrá expresado en Unidades de Insulina.

- `float getTotalBasalIOB (float basal)`:
  Esta función permite obtener el total de insulina activa en concepto de insulina basal en el momento de su ejecución, teniendo en cuenta toda la insulina dosificada previamente como basal y la programada como basales temporales a falta de dosificar. Las necesidades basales se establecen en el parámetro `basal` en Unidades de insulina por hora (U/h). El resultado vendrá expresado en Unidades de Insulina.

- `float getTotalIOB (float basal)`:
  Esta función calcula el valor total de Insulina Activa del paciente tomando el valor del parámetro `basal` como las necesidades basales actuales del paciente medidas en Unidades de Insulina por hora (U/h). El resultado devuelto por la función viene expresado en Unidades de Insulina.

- `float getInstantIOB (void)`: Esta función devuelve la cantidad de Insulina Activa Instantánea (ver apartado 5.3.1) que está causando efecto en el cuerpo del paciente según sus parámetros y la insulina previamente dosificada. El valor devuelto por la función viene expresado en Unidades de Insulina.

- `float getBasalForTime (unsigned char hour, unsigned char minute)`:
  Mediante esta función se puede consulta el nivel basal programado en la bomba de insulina para una hora en concreto del día. La hora del día se proporciona como dos parámetros tipo `unsigned char`: uno para la hora y otro para los minutos. El resultado se devuelve en Unidades de Insulina por hora y, en caso de proporcionar datos erróneos o de que no se tengan datos de la bomba, se devuelve el valor `0.0`.

- `float getCurrentBasal (void)`:
  De manera análoga a la función `getBasalForTime`, esta función devuelve el valor del nivel basal programado en la bomba para el momento concreto de la llamada a la función (suponiendo que el reloj de tiempo real esté configurado y sincronizado con la bomba de insulina). El resultado se devuelve en Unidades de Insulina por hora y, en caso de error, se devuelve `0.0`.

- `int getInsulinSensitivityForTime (unsigned char hour, unsigned char minute)`:
  Esta función permite obtener el valor de sensibilidad a la insulina programado en la bomba de insulina para una hora del día en concreto. La hora del día se proporciona como dos parámetros tipo `unsigned char`: uno para la hora y otro para los minutos. El resultado se devuelve en mg/dl por unidad de Insulina y, en caso de proporcionar datos erróneos o de que no se tengan datos de la bomba, se devuelve el valor `0`.

- `int getCurrentInsulinSensitivity (void)`:
  Esta función permite conocer el valor de sensibilidad a la insulina programado en

la bomba de insulina para el momento concreto en el que se llama a la función (suponiendo que el reloj de tiempo real esté configurado y sincronizado con la bomba de insulina). El resultado se devuelve en mg/dl por unidad de insulina. En caso de error, la función devuelve el valor `0`.

- `char getTargetsForTime (unsigned char hour, unsigned char minute, int *min, int *max)`:
  Esta función permite obtener los valores de objetivo máximo y mínimo configurados en la bomba de insulina para una hora del día en concreto. La hora del día a consultar (hora y minutos) se especifica en los dos primeros parámetros como variables de tipo `unsigned char`. Los dos últimos parámetros son los punteros a dos variables de tipo `int` donde serán devueltos los objetivos mínimo y máximo respectivamente. La función devuelve el valor `1` cuando la consulta ha sido ejecutada correctamente y `0` cuando ha sucedido algún error.

- `char getCurrentTargets (int *min, int *max)`:
  Al igual que la función `getTargetsForTime`, esta función devuelve los valores de objetivos glucémicos configurados en la bomba de insulina para el momento en concreto de la llamada a esta función (suponiendo que el reloj de tiempo real esté configurado y sincronizado con la bomba de insulina). Los dos parámetros que se pasan a la función son dos punteros a `int` donde se devolverán los valores del objetivo mínimo y máximo respectivamente. La función devuelve el valor `1` cuando la consulta ha sido ejecutada correctamente y `0` cuando ha sucedido algún error.

- `unsigned int calculateBolusSnoozeTime (void)`:
  Esta función considera que se debe esperar a que llegue el pico de acción de la insulina del bolo para poder volver a operar en lazo cerrado. La función analiza la curva programa de acción de la insulina y devuelve los minutos a los que el pico de acción ocurre.

- `char getCurrentTime (unsigned char *hour, unsigned char *minute)`:
  Esta función permite obtener la hora actual del reloj en tiempo real. Los dos parámetros pasados a la función corresponden a dos punteros a variables de tipo `unsigned char` donde se devuelven la hora y minutos actuales correspondientes. La función devuelve el valor `1` cuando se ha ejecutado correctamente o el valor `0` cuando el reloj no ha sido todavía sincronizado con la bomba de insulina.

A.4 Variables globales accesibles para el algoritmo de control

Existe un conjunto de variables globales que se han dejado accesibles al diseñador del algoritmo para reducir el uso de memoria Flash y RAM. Entre estas variables tenemos

las siguientes:

- `systemTimeSynced`: Es una variable tipo `Boolean` que indica si el reloj de tiempo real del sistema ha sido sincronizado con la bomba de insulina. Esta variable tendrá el valor `1` cuando el reloj esté sincronizado y `0` en caso contrario.
- `insulinSensValidRanges`: Esta variable indica el número de rangos de tiempo definidos para los ajustes de sensibilidad a la insulina en la bomba de insulina. Cuando el sistema está sin inicializar el valor es `0` y, cuando se obtiene la información de la bomba de insulina, su valor debe ser un número positivo mayor o igual a `1`.
- `pumpBasalsValidRanges`: Esta variable indica el número de rangos de tiempo definidos para los ajustes de niveles basales en la bomba de insulina. Cuando el sistema está sin inicializar el valor es `0` y, cuando se obtiene la información de la bomba de insulina, su valor debe ser un número positivo mayor o igual a `1`.
- `pumpTargetsValidRanges`: Esta variable indica el número de rangos de tiempo definidos para los ajustes de objetivos glucémicos en la bomba de insulina. Cuando el sistema está sin inicializar el valor es `0` y, cuando se obtiene la información de la bomba de insulina, su valor debe ser un número positivo mayor o igual a `1`.
- `historySgv[16]`: Este array de 16 valores tipo `int` almacena las 16 últimas medidas recibidas del MCG, siendo el índice 0 el último valor recibido y el 15 el valor recibido 75 minutos antes.
- `historySgvValid[16]`: Este array de 16 valores tipo `Boolean` almacena si los valores del array `historySgv[16]` con índice correspondiente son válidos o no. Esto permite tener elementos en la historia que no se reciban por, por ejemplo, problemas en las comunicaciones y, aun así poder seguir operando. Cuando el valor de uno de los elementos del array es `1`, el valor almacenado en ese mismo índice en el array `historySgv[16]` es un valor de glucemia recibido y válido. Por contrario, si el valor almacenado es `0`, el valor almacenado en ese mismo índice en el array `historySgv[16]` no es válido y debe ser ignorado o interpolado en caso de ser necesario.
- `bolusIOBvector[96]`: Este array de 96 valores tipo `Int` almacena la historia de dosificación de insulina en calidad de bolo de insulina durante las últimas 8 horas (un valor cada 5 minutos). El índice 0 representa al tiempo más cercano y, por tanto, a la última ejecución del lazo de control. El índice 95 equivale al más distante en el tiempo (8 horas). Se ha optado por utilizar variables tipo `Int` en vez de punto flotante para reducir la memoria utilizada a la mitad (en esta plataforma un `Int` ocupa 2 bytes, mientras que un `Float` ocupa 4 bytes). Para poder almacenar valores con parte fraccional utilizando variables `Int`, se hace una conversión interna que debe ser tenida en cuenta si se utiliza este array. Los datos en coma flotante se multiplican

por `40` y se redondean para que, de una manera sencilla, se mantenga la resolución de dosificación de la bomba de insulina (0.025 unidades).

- `basalIOBvector[96]`: Este array de 96 valores tipo `Int` almacena la historia de dosificación de insulina en calidad de insulina basal durante las últimas 8 horas (un valor cada 5 minutos). El índice 0 representa al tiempo más cercano y, por tanto, a la última ejecución del lazo de control. El índice 95 equivale al más distante en el tiempo (8 horas). Al igual que sucede en el caso del array `bolusIOBvector[96]`, se ha optado por utilizar variables tipo `Int` en vez de punto flotante para reducir la memoria utilizada a la mitad. El factor de conversión, de manera análoga, es el valor `40`.

- `responseToInsulin[96]`: Este array de 96 valores tipo `Int` almacena la respuesta a la insulina que utiliza el sistema para todos sus cálculos. Este array se ve afectado por la curva fijada en el sistema como curva para duración de 3 horas y, posteriormente, adaptada según la duración a la insulina programada en la bomba de insulina. Dado que estos sistemas utilizan insulinas de absorción rápida, los 96 valores del array (8 horas) deberían ser suficientes para codificar la respuesta incluso en el peor caso. Como sucede con los arrays de historia de dosificación de insulina, se ha optado por almacenar los datos en variables tipo `Int` para reducir su tamaño a la mitad frente a usar variables tipo `Float`. En este caso, dado que la curva de respuesta a la insulina está normalizada a 1, se ha optado por hacer una conversión a coma fija: 1 bit de signo (que siempre debería ser 0), 1 bit de parte entera y 14 bits de parte fraccional). Para convertir cualquier valor de este array a punto flotante basta con dividir por el coeficiente `16384`.

A.5 Funciones de control de terapia

Es necesario tener la capacidad de poder comandar a la bomba de insulina bolos de insulina, basales temporales, cancelarlas y/o mandar notificaciones al *smartphone*. Para ejecutar las acciones de control sobre la bomba de insulina, las siguientes funciones están disponibles para el diseñador del algoritmo:

- `void suspendPump (void)`:
  Esta función solicita al sistema que se comunique a la bomba de insulina la suspensión de la dosificación, independientemente de los comandos que estuviera ejecutando.

- `void resumePump (void)`:
  Esta función solicita la reanudación de la dosificación de insulina tras una suspensión. Este comando carece de efecto si la bomba de insulina no estaba en estado de suspensión.

- `void setTempBasal (unsigned int minutes, float rate)`:
  Mediante los parámetros `minutes` y `rate`, esta función solicitará a la bomba de insulina la inicialización de una basal temporal del valor establecido por `rate` que durará los minutos establecidos por `minutes`. La variable `rate` será ajustada para que las unidades coincidan con las esperadas por la bomba de insulina (resolución de 0.025 unidades). No es necesario cancelar una basal temporal para poder comandar una nueva. Cuando existe una basal temporal previa, sus efectos quedan cancelados por la orden de la nueva. Internamente se comprueba que no se superan los límites establecidos por el paciente y, en caso de superarse, se limita la acción al valor máximo o mínimo fijado. El sistema actualiza las variables internas y la historia acorde a si se pudo ejecutar el comando o no; no es necesario que el lazo de control actualice nada en la historia.

- `void cancelTempBasal (void)`:
  Esta función solicita a la bomba de insulina que cancele cualquier basal temporal que estuviera ejecutando y que vuelva a dosificar la basal programada en el perfil basal del paciente.

- `void bolus (float insulinAmount)`:
  Esta función solicita la dosificación de un bolo de insulina. La cantidad de insulina a dosificar se establece por el parámetro `insulinAmount`. Si se supera el valor de bolo máximo programado por el paciente en la bomba de insulina, se limita internamente el bolo a dicho valor. Cualquier valor de cantidad será ajustado a la resolución de dosificación de la bomba de insulina (0.025 unidades).

Apéndice B

Listado de términos

B.1 Terminología aplicable

En esta sección se explican los diferentes términos técnicos que son aplicables y que serán de amplio uso a lo largo de la memoria de la tesis:

- AHCL (Advanced Hybrid Closed Loop): Control en lazo cerrado híbrido avanzado. El término híbrido hace referencia a la necesidad añadida de control por parte del paciente en ciertas situaciones. Típicamente ingestas y ejercicio físico. El término avanzado hace referencia a la capacidad del algoritmo de dosificar de manera automática bolos correctores de insulina cuando son necesarios.
- APS (Artificial Pancreas System): Sistema de páncreas artificial o sistema de control de glucemia en lazo cerrado.
- Basal temporal: Cantidad de insulina programada en una bomba para que sea suministrada al paciente durante un tiempo a definir. Normalmente se establecen periodos en pasos de media hora y la cantidad de insulina se establece como una basal (medida en U/h).
- Bolo de insulina: Cantidad de insulina que se suministra al paciente de manera inmediata. Es el equivalente a una inyección. Se mide en unidades.
- COB (Carbohydrates-On-Board) o Hidratos de carbono activos: Cantidad de hidratos de carbono ingeridos y que aún no han sido asimilados.
- DIOB (Duration of Insulin-On-Board) o Duración de la insulina: Tiempo que permanece una unidad de insulina causando efecto en el cuerpo del paciente. Normalmente se mide en horas.
- dSGV (*derivative of Sensor Glucose Value*): Primera derivada del valor de glucemia intersticial ofrecido por el monitor continuo de glucemia.

- eBasal (*Estimated Basal*): Estimación de necesidades basales de insulina.
- Factor de sensibilidad a la insulina (ISF - *Insulin Sensitivity Factor*): Variación que produce una unidad de insulina en la glucemia del paciente. Normalmente los pacientes toman este dato como dependiente únicamente de la franja horaria, pero hay estudios que sugieren un comportamiento no lineal con la glucemia de partida del paciente. La sensibilidad es menor cuanto mayor es la glucemia. Se mide en (mg/dl)/U o en (mmol/L)/U. Suele venir determinada en forma de tabla, dando la posibilidad de especificar distintas sensibilidades a la insulina según la hora del día.
- FDA (*Food and Drug Administration*): Organismo norteamericano encargado de, entre otros asuntos, la regulación de dispositivos médicos para Estados Unidos.
- Glucemia capilar: Concentración de glucosa medida mediante la extracción de una gota de sangre en un dedo.
- Glucemia en sangre: Concentración de glucosa en el torrente sanguíneo medida en mg/dl o mmol/L (depende de la región geográfica)
- Glucemia intersticial: Concentración de glucosa en el líquido intersticial medida en mg/dl o mmol/L (depende de la región geográfica)
- GVI (*Glycemic Variability Index*): Medida desarrollada por la compañía Dexcom que, basándose en el concepto de ”longitud de la linea”, compara la línea de glucemia recorrida en el caso del paciente con una linea recta perfecta en un intento por evaluar la variabilidad.
- HbA1c: Hemoglobina glicosilada.
- HCL (Hybrid Closed Loop): Control en lazo cerrado híbrido. El término híbrido hace referencia a la necesidad añadida de control por parte del paciente en ciertas situaciones. Típicamente ingestas y ejercicio físico.
- IAT (*Insulin Action Time*): Duración de una unidad de insulina en el cuerpo del paciente.
- iBolus: Bolo instantáneo de insulina o contribución de un bolo de insulina previamente administrado en el momento actual.
- ICSI: Terapia basada en la Infusión Continua Subcutanea de Insulina
- iIOB: (*instantaneous Insulin-On-Board*): suma de la contribución instantánea de toda la insulina previamente administrada al paciente.
- Insulina basal: Cantidad de insulina necesaria para suplir las necesidades del cuerpo y mantener planos y estables las concentraciones de glucosa en sangre. Se mide en U/h (unidades por hora)

- IOB (Insulin-On-Board) o Insulina Activa: Cantidad de insulina suministrada al paciente y que aún no ha sido absorbida.
- Líquido intersticial: Liquido existente en el intersticio o espacio que rodea a las células.
- LTS (*Low Threshold Suspend*): Sistema de suspensión de la infusión de insulina debida a un descenso de la glucemia del paciente por debajo de un cierto umbral.
- MCG: Monitor Continuo de Glucemia. Dispositivo que sensa la glucemia intersticial de un paciente de manera periódica y automática.
- MDI: Terapia basada en Múltiples Dosis de Insulina
- MPC (Model Predictive Controller): Controlador predictivo basado en modelos. Los modelos típicamente utilizados son de los sensores de glucemia, del efecto de la insulina en el cuerpo y la respuesta del cuerpo del paciente en si mismo.
- Objetivo glucémico: Glucemia que se establece en el controlador como objetivo a conseguir para el paciente. Al igual que otros parámetros, suele ser configurado en forma de tabla para permitir distintos objetivos glucémicos en función de la hora del día.
- Perfil basal: Perfil de infusión de insulina basal de un paciente. Determina el número de unidades por hora que deben ser dosificadas al paciente según la hora del día. Típicamente se establecen tramos de, como mínimo, 30 minutos. También es posible configurar un único tramo de 24 horas que cubra el día completo.
- PGS (*Patient Glycemic Status*): Medida desarrollada por la compañía Dexcom que, combinando el cálculo de GVI, la media de la glucemia del paciente y el tiempo en rango, intenta ofrece un valor relativo al estado glucémico del paciente.
- Picos postprandiales: Incremento puntual de la glucemia del paciente como resultado de la absorción de hidratos de carbono después de una ingesta.
- Ratios de hidratos de carbono: Relación de insulina frente a hidratos de carbono para las ingestas. Los hidratos de carbono se pueden medir en raciones o en gramos dependiendo de la región geográfica. Según esto, las ratios se pueden medir en U/ración o U/gr
- Set de infusión: equipo, típicamente fungible, que conecta el reservorio de la bomba de insulina con el cuerpo del paciente, permitiendo la infusión insulina desde el reservorio hasta el tejido subcutáneo
- SGV (*Sensor Glucose Value*): Valor de glucemia intersticial ofrecido por el monitor continuo de glucemia.

- SoC (*System On Chip*): Componente electrónico que integra un procesador junto con ciertos periféricos que le permiten trabajar como si fueran un sistema completo. Estos perifericos típicamente incluyen memoria, interfaces de comunicaciones, conversores AD/DA, entrada/salida de propósito general, etc.
- TDD (*Total Daily Dose*): Cantidad total de insulina utilizada por un paciente en 24 horas.
- TIR (*Time In Range*): Porcentaje de tiempo diario que el paciente mantiene su glucemia dentro del rango objetivo.