



UNIVERSIDAD NACIONAL DE COLOMBIA

**DESARROLLO DE ESTRATEGIAS PARA EL USO  
EFICIENTE DE LOS RECURSOS ENERGÉTICOS  
EN UN CONTROL PREDICTIVO BASADO EN  
MODELO IMPLEMENTADO EN UN SISTEMA  
EMBEBIDO PARA EL TRATAMIENTO DE LA  
DIABETES MELLITUS TIPO I.**

**Jhon Edison Goez Mora**

Universidad Nacional de Colombia  
Facultad de Minas, Departamento de Energía Eléctrica y Automática  
Medellín, Colombia  
2022



**DESARROLLO DE ESTRATEGIAS PARA EL USO  
EFICIENTE DE LOS RECURSOS ENERGÉTICOS EN UN  
CONTROL PREDICTIVO BASADO EN MODELO  
IMPLEMENTADO EN UN SISTEMA EMBEBIDO PARA EL  
TRATAMIENTO DE LA DIABETES MELLITUS TIPO I**

**Jhon Edison Goez Mora**

Tesis presentada como requisito parcial para optar al título de:  
**Magister en Ingeniería - Automatización Industrial**

Directores:

Dra. Mónica Ayde Vallejo Velásquez

Dr. Pablo Santiago Rivadeneira Paz

Línea de Investigación:

Ingeniería biomédica

Universidad Nacional de Colombia  
Facultad de Minas, Departamento de Energía Eléctrica y Automática  
Medellín, Colombia

2022



# Agradecimientos

Agradezco a la Doctora Mónica Ayde Vallejo Velásquez y al Doctor Pablo Santiago Rivadeneira Paz, por su diligente acompañamiento en mi proceso formativo tanto en mi pregrado como en mi posgrado como directora y codirector de esta tesis de maestría. Además, agradezco su orientación, su paciencia y la forma en la que me ayudaron en el aprendizaje de nuevas metodologías que permitieron combinar desarrollos y conceptos teóricos con nuevas tecnologías que permiten su implementación en sistemas embebidos.

A mis familiares y personas cercanas . . . , por el constante apoyo y acompañamiento durante mi crecimiento personal.



# Resumen

Los avances tecnológicos actuales han acercado a la realidad el proyecto de un páncreas artificial (PA) seguro, portátil y eficiente para personas con diabetes tipo 1. Entre las estrategias de control desarrolladas para la diabetes tipo 1, el control predictivo basado en modelo (MPC por sus siglas en inglés) se ha enfatizado en la literatura como un control prometedor para la regulación de la glucosa. Sin embargo, estas estrategias de control se diseñan comúnmente en un entorno simulado, independientemente de las limitaciones de un dispositivo portátil. En este trabajo, se evalúa el rendimiento de seis sistemas embebidos, tres paquetes de optimización de código abierto y cuatro formulaciones de MPC con un esquema de hardware en el ciclo (HIL por sus siglas en inglés), para encontrar la mejor combinación tomando como criterios de selección la temperatura del procesador, el tiempo de ejecución, el coeficiente de variación, el porcentaje de tiempo en normoglucemia, la energía consumida, la cantidad de eventos de hiperglucemia y la diferencia respecto a la evolución obtenida en MATLAB. Al escoger la mejor combinación se aplica la estrategia de eventos de activación con el fin de reducir el número de veces que se ejecuta el cálculo de la dosificación de insulina óptima permitiendo que durante los periodos de tiempo en los que no es necesario llevar a cabo acciones de control el dispositivo ahorre energía.

Durante el desarrollo de las pruebas los controladores son expuestos a variaciones fisiológicas simuladas en los pacientes virtuales, a ingesta de carbohidratos y a ejecutarse con y sin anuncio de comida. Los primeros resultados muestran que la Raspberry pi 3 B, el paquete quadprog y la estrategia de eliminación de *offset* son la mejor combinación resaltando el bajo consumo energético del dispositivo. Con esta base se integra la estrategia de activación de eventos y se realizan las pruebas poblacionales encontrando una reducción significativa en el número de controles calculados, aunque se presenta una pérdida de desempeño en el controlador al elevarse los niveles promedios de glucemia. Por último, se realiza una emulación del PA con un paciente virtual en donde se implementa un sensor inteligente, un micromotor paso a paso como actuador y una batería con la que se determina el consumo real del dispositivo y su tiempo de autonomía con y sin el nuevo controlador basado en MPC.

**Palabras clave:** Páncreas artificial, sistemas embebidos, Control predictivo basado en modelo, Paquetes de optimización, Diabetes tipo 1, recursos energéticos.

**DEVELOPMENT OF STRATEGIES FOR THE EFFICIENT  
USE OF ENERGY RESOURCES IN A MODEL PREDICTIVE  
CONTROL IMPLEMENTED IN AN EMBEDDED SYSTEM  
FOR THE TREATMENT OF TYPE I DIABETES MELLITUS**

**Jhon Edison Goez Mora**

Directors:

Dra. Mónica Ayde Vallejo Velásquez

Dr. Pablo Santiago Rivadeneira Paz

Universidad Nacional de Colombia  
Facultad de Minas, Departamento de Energía Eléctrica y Automática  
Medellín, Colombia

2022

# Abstract

Current technological advances have brought closer to reality the project of a safe, portable, and efficient artificial pancreas for people with type 1 diabetes (T1D). Among the developed control strategies for T1D, model predictive control (MPC) has been emphasized in literature as a promising control for glucose regulation. However, these control strategies are commonly designed in a computer environment, regardless of the limitations of a portable device. When choosing the best combination, the event-triggering strategy is applied in order to reduce the number of times the calculation of the optimal insulin dosage is executed, allowing during the periods of time in which it is not necessary to carry out control actions the device save energy.

During the development of the tests, the controllers are exposed to simulated physiological variations in virtual patients, carbohydrate intake and to execute with and without announced meals. The first results show that the Raspberry pi 3 B, the quadprog package, and the offset-free strategy are the best combination highlighting the low power consumption of the device. With this base, the event-triggering strategy is integrated and population tests are carried out, finding a significant reduction in the number of calculated controls, although there is a loss of performance in the controller as average blood glucose levels rise. Finally, an emulation of the artificial pancreas is carried out with a virtual patient where an intelligent sensor, a micro stepper motor as an actuator, and a battery in implemented with which the real consumption of the device and its autonomy time is determined whit and without the new controller base on MPC.

**Keywords:** Artificial Pancreas, Embedded control systems, Model predictive control, Optimization solver packages, Type 1 diabetes, energetic resources

# Contenido

<b>Agradecimientos</b>	<b>v</b>
<b>Resumen</b>	<b>vii</b>
<b>1 Introducción</b>	<b>2</b>
1.1 Justificación . . . . .	3
1.2 Objetivos . . . . .	4
1.2.1 Objetivo general . . . . .	4
1.2.2 Objetivos específicos . . . . .	4
1.3 Marco teórico . . . . .	5
1.4 Antecedentes . . . . .	7
1.5 Alcance . . . . .	9
<b>2 Modelo, Controlador, estimador y paquetes de optimización</b>	<b>10</b>
2.1 Modelos . . . . .	10
2.1.1 Modelo de Bergman . . . . .	11
2.1.2 Modelo de Magdelaine . . . . .	12
2.1.3 Modelo de Hovorka . . . . .	13
2.1.4 Modelo implementado . . . . .	14
2.2 Monitoreo . . . . .	15
2.2.1 Sensores para CGM . . . . .	16
2.2.2 Modelo de ruido sensor CGM . . . . .	17
2.3 Estrategias de control . . . . .	18
2.3.1 Control predictivo basado en modelo . . . . .	18
2.3.2 Control predictivo basado en modelo por zonas . . . . .	20
2.3.3 Control predictivo basado en modelo por zonas con variables artificiales	22
2.3.4 Control predictivo basado en modelo por zonas con variables artificia- les y Offset-free . . . . .	27
2.4 Estimador . . . . .	32
2.5 Paquetes de optimización . . . . .	34
2.5.1 CVXOPT . . . . .	35
2.5.2 quadprog . . . . .	37
2.5.3 OSQP . . . . .	40

<b>3</b>	<b>Sistemas embebidos</b>	<b>43</b>
3.1	Embebidos . . . . .	43
3.1.1	Estructura de los sistemas embebidos . . . . .	44
3.1.2	Sistema operativo embebido . . . . .	46
3.2	Computadores de placa única . . . . .	47
3.2.1	Raspberry Pi 3 B . . . . .	48
3.2.2	Raspberry Pi 4 . . . . .	48
3.2.3	Organge pi PC + . . . . .	49
3.2.4	Jetson Nano . . . . .	49
3.2.5	ODROID-XU4 . . . . .	50
3.2.6	Tinker Board S . . . . .	50
3.2.7	Resumen . . . . .	51
3.3	Lenguaje de programación . . . . .	52
3.4	Periféricos . . . . .	52
<b>4</b>	<b>Evaluación de los controladores MPC en los sistemas embebidos</b>	<b>55</b>
4.0.1	Sensores en el esquema de HIL . . . . .	56
4.0.2	Patrón de referencia (Matlab) . . . . .	57
4.0.3	Configuración de los sistemas embebidos . . . . .	57
4.1	Escenario de simulación y métricas . . . . .	59
4.2	Resultados . . . . .	62
4.3	Discusión . . . . .	69
4.4	Resumen . . . . .	72
<b>5</b>	<b>Evaluación de las estrategias de uso eficiente de la energía en controladores MPC para la aplicación del PA</b>	<b>73</b>
5.1	Estrategia de activación de eventos para los recursos energéticos del dispositivo	73
5.1.1	Configuración del sistema embebido y periféricos . . . . .	75
5.1.2	Batería . . . . .	75
5.1.3	Actuador . . . . .	77
5.1.4	Escenario de simulación y métricas . . . . .	79
5.1.5	Efectos en la reducción de cálculos del controlador . . . . .	82
5.2	Resultados escenario A . . . . .	85
5.2.1	Efectos de la variación del umbral . . . . .	85
5.2.2	Cambio de sintonización con la estrategia ET . . . . .	88
5.2.3	Pruebas de desempeño del nuevo controlador . . . . .	90
5.3	Resultados escenario B . . . . .	95
5.3.1	Descarga de la batería . . . . .	95
5.3.2	Prueba de desempeño en la emulación del PA . . . . .	96

---

5.4	Opciones alternativas para el uso de los recursos limitados en el páncreas artificial . . . . .	102
5.4.1	Esquema de eventos de activación y restricciones para manejar recursos limitados . . . . .	102
5.4.2	Resultados y discusión . . . . .	104
5.5	Resumen . . . . .	107
<b>6</b>	<b>Conclusiones y recomendaciones</b>	<b>109</b>
6.1	Conclusiones . . . . .	109
6.2	Recomendaciones . . . . .	110
	<b>Glosario</b>	<b>114</b>
	<b>Bibliografía</b>	<b>114</b>

# Lista de Figuras

2-1	Efectos de los casos de hiper e hipoglucemia . . . . .	11
2-2	Uso del sensor <i>FreeStyle Libre</i> . Esta foto de Autor desconocido está bajo licencia CC BY-SA. <a href="https://creativecommons.org/licenses/by-sa/3.0/">https://creativecommons.org/licenses/by-sa/3.0/</a> . . . . .	16
2-3	Uso del sensor Dexcom G5. Imagen tomada de [Dexcom, 2015] . . . . .	17
2-4	Esquema de control para el PA con MPC . . . . .	19
2-5	Esquema de control para el PA usando el controlador ZMPC-AV-OF . . . . .	28
2-6	Algoritmo del filtro de Kalman . . . . .	33
2-7	Criterio de terminación en MATLAB, basado en [MathWorks, 2021b] . . . . .	35
3-1	Procesador ARM. Esta imagen de Autor desconocido está bajo licencia CC BY-NC-ND. <a href="https://creativecommons.org/licenses/by-nc-nd/3.0/">https://creativecommons.org/licenses/by-nc-nd/3.0/</a> . . . . .	44
3-2	Diagrama de bloque del núcleo de un procesador Intel con tecnología Híbrida. Imagen tomada de [Intel, 2020] . . . . .	45
3-3	Las seis SBC seleccionadas para las pruebas de desempeño . . . . .	47
3-4	Diagrama de bloques del procesador de la Raspberry Pi 3 B . . . . .	48
3-5	Diagrama de bloques del procesador de la Raspberry Pi 4 . . . . .	49
3-6	Diagrama de bloques del procesador de la Tinker Board S . . . . .	50
3-7	Diagrama de puertos, conexiones y periféricos de la Raspberry Pi 3 B . . . . .	53
3-8	Tarjeta para conversión USB-Serial. Imagen tomada de [I+D, 2015] . . . . .	54
4-1	Implementación de hardware-in-the-loop para emular el AP. Elaboración propia a partir de imágenes bajo licencia CC BY-SA . . . . .	55
4-2	Conexión del sensor Gravity: I2C Digital Wattmeter. Imágenes de [the Future, 2021] . . . . .	56
4-3	Temperatura en el procesador de los 6 SBC al usar el ZMPC-AV-OF . . . . .	63
4-4	Tiempo de procesamiento promedio de los 6 SBC con las 4 estrategias . . . . .	63
4-5	Energía promedio consumida en los 6 SBC con las 4 estrategias . . . . .	64
4-6	Evolución de la glucemia bajo la estrategia MPC con los tres paquetes . . . . .	64
4-7	Evolución de la glucemia bajo la estrategia ZMPC con los tres paquetes . . . . .	65
4-8	Evolución de la glucemia bajo la estrategia ZMPC-AV con los tres paquetes de optimización . . . . .	66
4-9	Evolución de la glucemia usando ZMPC-AV-OF con el paquete quadprog . . . . .	66
4-10	Resultados con ZMPC-AV-OF, (A) CV, (B) Tiempo en normoglucemia, (C) Eventos sobre 180 mg/dl y (D) Error respecto a MATLAB. . . . .	67
4-11	Ponderación obtenida de los sistemas embebidos en los criterios de desempeño. . . . .	68

5-1	Gestor y batería Pisugar 2 pro . . . . .	76
5-2	Dimensiones del micromotor paso a paso en $[mm]$ , con escala 2:1 . . . . .	78
5-3	Conexión del driver para motor paso a paso TB6612FNG . . . . .	79
5-4	Esquemas de simulación A. Elaboración propia a partir de imágenes CC-BY-SA	80
5-5	Esquemas de simulación B. Elaboración propia a partir de imágenes CC-BY-SA	81
5-6	Diagrama de flujo para el esquema de simulación B . . . . .	82
5-7	Dinámica de la glucemia con variaciones en el tiempo de muestreo $T_s$ . . . . .	83
5-8	Inyección de insulina con variaciones en el tiempo de muestreo $T_s$ . . . . .	83
5-9	Tiempo de simulación y energía consumida por $\epsilon$ . . . . .	85
5-10	Cantidad de controles calculados por $\epsilon$ . . . . .	86
5-11	Resultados promedio por $\epsilon$ , <b>(A)</b> CV, <b>(B)</b> Tiempo en normoglucemia, <b>(C)</b> Casos sobre 180 mg/dl y <b>(D)</b> Glucemia promedio. . . . .	86
5-12	Glucemia promedio de los diez pacientes por $\epsilon$ . . . . .	87
5-13	Comparación en la dinámica de la glucemia entre el TT y el ET . . . . .	87
5-14	Comparación en la dinámica de la insulina entre el TT y el ET . . . . .	87
5-15	Cantidad de controles promedio calculados por paciente . . . . .	88
5-16	Resultados promedio por paciente, <b>(A)</b> Coeficiente de variación, <b>(B)</b> Porcen- taje de tiempo en normoglucemia, <b>(C)</b> Número de casos por encima de 180 mg / dl y <b>(D)</b> Glucemia promedio. *Pac.=Paciente . . . . .	89
5-17	Glucemia promedio por paciente con variaciones del 10 % y anuncio de comida	90
5-18	Cantidad de controles calculados: Anun=Anuncio, SinA=Sin Anuncio, Varia- ciones paramétricas del 20 y 30 % . . . . .	91
5-19	Resultados con/sin anuncio y variaciones del 20/30 %, <b>(A)</b> Coeficiente de variación, <b>(B)</b> Porcentaje de tiempo en normoglucemia, <b>(C)</b> Número de casos por encima de 180 mg/dl y <b>(D)</b> Número de casos por debajo de 70 mg/dl . . . . .	92
5-20	Glucemia promedio con/sin anuncio de comida y variaciones del 20/30 % . . . . .	93
5-21	Dinámica de la glucemia con/sin anuncio de comida y variaciones del 20/30 %	93
5-22	Dinámica de la descarga de la batería en voltaje . . . . .	95
5-23	Dinámica de la descarga de la batería en porcentaje . . . . .	96
5-24	Prueba base de la estrategia TT . . . . .	97
5-25	Prueba inicial del PA . . . . .	98
5-26	Emulación del PA usando ET, anuncio de comida y variaciones del 10 % . . . . .	99
5-27	Comparación entre el ET y el TT en la glucemia e insulina . . . . .	100
5-28	Comparación entre el ET y el TT en la energía consumida . . . . .	100
5-29	Comparación de ET (Anuncio/10 %, Sin Anuncio/30 %), glucemia . . . . .	101
5-30	Comparación de ET (Con/sin anuncio, Variaciones 10 %/30 % ), energía . . . . .	102
5-31	Resultado del desempeño del ZMPC al implementar diferentes valores de $\epsilon$ . . . . .	105
5-32	Rendimiento del MPC al incluir la limitación de energía de la batería . . . . .	106
5-33	Rendimiento del MPC considerando la restricción de suministro de insulina . . . . .	106

---

<b>5-34</b> Comparación entre la ZMPC estándar y el controlador propuesto con todas las estrategias . . . . .	107
---	-----

# Lista de Tablas

<b>2-1</b>	Descripción de variables y parámetros del modelo. . . . .	15
<b>2-2</b>	Parámetros estimados de 10 pacientes adultos . . . . .	15
<b>3-1</b>	Diferencias entre RISC y CISC . . . . .	45
<b>3-2</b>	Características de los sistemas embebidos . . . . .	51
<b>4-1</b>	Tolerancias de los paquetes de optimización . . . . .	60
<b>4-2</b>	Parámetros de sintonía de las estrategias por paciente. . . . .	61
<b>4-3</b>	Evaluación del desempeño general para la selección de los sistemas embebidos. . . . .	68
<b>4-4</b>	Complejidad de las estrategias MPC . . . . .	70
<b>5-1</b>	Rangos de carga/descarga de la batería. . . . .	77
<b>5-2</b>	Parámetros de sintonía de ET con anuncio y variaciones del 10% por paciente. . . . .	84
<b>5-3</b>	Parámetros de sintonía de ET con y sin anuncio de comida y variaciones del 20% / 30% por paciente. . . . .	84
<b>5-4</b>	Comparación del desempeño del controlador con diferentes esquemas de MPC de estudios reportados en la literatura. . . . .	94

# 1 Introducción

Durante los últimos años se ha presentado un incremento de la población joven y adulta que padece de diabetes mellitus tipo 1 (T1DM, de sus siglas en inglés) [IDF, 2019, Hei et al., 2015]. Esta enfermedad autoinmune que destruye las células  $\beta$  producidas por el páncreas genera una deficiencia en la producción de insulina, hormona necesaria para el procesamiento y regulación de la glucosa en la sangre [Gondhalekar et al., 2016], la consecuencia de esta deficiencia hace que el cuerpo almacene altos niveles de glucosa que provocan daños en los órganos, venas y nervios, deteriorándolos hasta el punto de llegar a ser mortal, esta enfermedad disminuye considerablemente la calidad de vida de las personas [Shi et al., 2019].

El padecimiento de T1DM obliga a los pacientes a tener una dependencia de insulina exógena generalmente aplicada con inyecciones periódicas o a través del uso de dispositivos como las bombas de insulina, que son programadas para la regulación mediante un suministro controlado [Toffanin et al., 2019]. Algunas de las ventajas de usar una bomba de insulina es que permite mantener los niveles de glucosa del paciente dentro de un rango seguro con mayor frecuencia, no tienen que utilizar inyecciones de forma periódica (lo cual genera molestia en el usuario y por ende baja adherencia al tratamiento), tienen mayor flexibilidad en la cantidad de comida que ingieren y en los periodos de tiempo entre comidas, además de adaptarse mejor a la regulación durante y después del ejercicio y se tiene una mayor precisión cuando se quiere disminuir altos niveles de glucosa. Sin embargo, el uso de la bomba tiene las siguientes desventajas; se requiere tiempo para que los pacientes aprendan a usarlas, tienen que cargarlas consigo todo el tiempo, la cánula conectada al usuario puede generar infección y se puede desarrollar lipohipertrofia (condición que afecta la absorción de insulina y puede generar un mayor riesgo de hipoglucemia o hiperglucemia), además en algunas ocasiones tendrán que recurrir nuevamente a las inyecciones [operating as Diabetes UK, 2018]. Así mismo, al ser un equipo portable, se debe recargar constantemente tanto de insulina como de nuevas baterías.

A pesar de las ventajas que se tiene al usar una bomba de insulina, muchos investigadores continúan centrando sus esfuerzos por desarrollar estrategias, métodos y dispositivos que mitiguen los padecimientos de las personas que sufren esta enfermedad mediante el monitoreo y control continuo y automático de la glucosa en la sangre, enfrentando problemas como el ruido de medición, la respuesta automática ante perturbaciones por ingesta de carbohidratos, actividad física, cambios de humor, discontinuidad en las mediciones de glucemia, variaciones

en la sensibilidad de los pacientes ante la insulina entre otros. Sin embargo, la gran mayoría de estas estrategias se limitan a ser aplicadas en simulaciones y no integran variables o problemas que deben resolverse para poder ser implementados en un dispositivo portátil; esta falta de acercamiento a las condiciones reales de operación de lo que se denomina páncreas artificial (PA) desde su etapa de diseño, puede generar un retraso en el desarrollo de un producto que pueda ser usado de forma segura por quienes tienen este tipo de diabetes. Esto hace necesario que todas estas técnicas, métodos y desarrollos comiencen a enfocarse en la creación de un PA real mediante un dispositivo embebido que se encargue del manejo de las señales del sensor, dosificación de la insulina y ejecute los algoritmos de control y estimación.

Para un buen desarrollo del PA se debe analizar el desempeño de las estrategias de control en sistemas embebidos con diferentes arquitecturas y varios paquetes de optimización, que permitan evaluar dicho rendimiento bajo criterios como el costo computacional, el tiempo de procesamiento, el consumo energético, los objetivos de control y la temperatura del procesador para determinar una combinación viable en la emulación del PA a la que se le puedan adaptar estrategias sobre el uso eficiente de los recursos disponibles, para que no solo satisfaga los requerimientos de regulación y control sino también las variables como la independencia energética y el control de las unidades de insulina que hacen posible en un futuro obtener un PA portátil, accesible y confiable.

## 1.1. Justificación

Durante los últimos años se ha presentado un incremento de la población que padece de diabetes, para 2019 se tienen datos de alrededor de 463 millones de adultos con esta enfermedad y bajo proyecciones se estima que para 2045 esta cifra ascienda a 700 millones de personas, que representan el 10.9% de la población mundial en las edades entre 20 y 79 años. Este incremento no solo se presenta en adultos, el número de niños y adolescentes que padecen de T1DM es estimado en 1.1 millones, en donde se encuentran alrededor de 128.900 casos nuevos por año; Además el número de muertes estimadas por la diabetes y sus complicaciones solo en 2019 fue de 4.2 millones de personas. Esta enfermedad genera un gran impacto económico en el mundo, los costos anuales calculados que genera la diabetes y sus complicaciones son de 760 billones de dólares y se proyecta que esa suma aumente a 845 billones de dólares para 2045, en donde cerca del 19.4% de los costos totales pertenecen a centro y sur américa [IDF, 2019].

En Colombia por cada 100 habitantes se ha tenido un incremento del 1.3 al 2.2, en los casos de personas que tienen diabetes durante el periodo 2013-2018. Antioquia se encuentra entre los departamentos que tiene un mayor incremento. El costo total estimado de esta enfermedad para Colombia es de \$5.227.460.800.000 pesos lo que representa un costo por caso al año de \$4.800.000 pesos. Adicional la medida de morbilidad (mortalidad) por cada 100.000

habitantes para 2019 es de 27.656 casos. [de Alto Costo, 2020, de Alto Costo, 2018]

Uno de los principales problemas presentes en el tratamiento de la T1DM es que las bombas de insulina y los tratamientos para mitigar los efectos de esta enfermedad son muy costosos y el sistema de salud o los seguros médicos no cubren a todos los pacientes que cada año aumentan. Los costos de una bomba de insulina para particulares oscilan aproximadamente entre 8 y 26 millones de pesos, teniendo en cuenta que se debe de renovar el set de infusión y las baterías, lo que suma un costo anual de alrededor de 6 millones de pesos [CostHelper, 2013], esta situación hace que muchos de los que padecen de T1DM no puedan acceder a este tipo de equipos avanzados e incluso a otras alternativas como las inyecciones, lo que disminuye considerablemente la calidad de vida de las personas. Los precios de las bombas de insulina residen en la calidad del equipo, las funciones que puede cumplir como el cálculo del bolo o el monitoreo continuo de la glucosa y el software que usan.

Con los avances y desarrollos tecnológicos en el área de los microprocesadores y microcontroladores se ha tenido un incremento en las aplicaciones de los sistemas embebidos en el área de la biomedicina y muchos otros campos. Este constante avance ha permitido que los sistemas embebidos hagan parte de múltiples soluciones debido a su flexibilidad en la arquitectura y en la capacidad de analizar señales en tiempo real, adaptándose a procesos y equipos específicos [García, 2019]. Estos trabajos en investigación y los desarrollos de los equipos electrónicos forman una base confiable para llevar la implementación de las estrategias de monitoreo y control de un PA inteligente a un sistema embebido dedicado a la regulación de la glucosa y la administración eficiente de los recursos.

## **1.2. Objetivos**

### **1.2.1. Objetivo general**

Desarrollar un control predictivo basado en modelo que integre estrategias de uso eficiente de los recursos energéticos en sistemas embebidos para la emulación de un páncreas artificial.

### **1.2.2. Objetivos específicos**

- Seleccionar un paquete solucionador y un sistema embebido evaluando estrategias estándar y avanzadas de control predictivo ya desarrolladas.
- Diseñar una estrategia de control predictivo que optimice el consumo energético del sistema embebido seleccionado.
- Evaluar el desempeño de la estrategia de control predictivo desarrollada mediante la

emulación del páncreas artificial, en términos de tiempo de ejecución, consumo energético y objetivos de control de la aplicación.

### 1.3. Marco teórico

Uno de los principales enfoques para abordar el tratamiento de la T1DM mediante un control automático, es el desarrollo de modelos matemáticos que representen adecuadamente el comportamiento de las hormonas de la glucosa y la insulina en la sangre. La mayoría de los controladores se basan en estos modelos para determinar el suministro adecuado de inyección de insulina. Por esta razón los trabajos dedicados a proponer un modelo matemático se construyen mediante una base de datos de pacientes reales. Un ejemplo de esto es el desarrollo realizado en 2017 [Ruan et al., 2017], en donde se propuso el modelo usando datos de personas adultas con este tipo de diabetes, a los que se les midió los niveles de glucosa usando la administración continua de insulina en un ciclo cerrado durante 24 horas por 12 semanas. Este tipo investigaciones y los modelos que plantean, componen una base sólida para iniciar el proceso de construcción de un páncreas artificial. Además ha sido acogido por la comunidad científica un simulador de pacientes diabéticos aprobado por la Administración de medicamentos y alimentos (FDA) el UVA/Padova de la universidad de virginia [Dalla Man et al., 2014]. Por otro lado, se innova en las técnicas y desarrollos tecnológicos para el monitoreo continuo de la glucosa (CGM por sus siglas en inglés) y los modelos del ruido presente en los sensores con el fin de mitigar efectos adversos en la dosificación de insulina [Vettoretti et al., 2018]. Adicionalmente se ha hecho énfasis en el estudio de los estados que no pueden medirse como la insulina subcutánea y obtener su valor usando métodos de estimación que al mismo tiempo ayudan en el tratamiento de los problemas de medición y en la incertidumbre del modelo [Hajizadeh et al., 2017].

Todas las investigaciones orientadas en los temas de modelado, medición y estimación, complementan la concepción de diferentes estrategias de control aplicadas al problema de regulación; A pesar de la existencia de un gran número de propuestas en este tema sobresalen tres grandes corrientes: el control proporcional integrador derivativo (PID), el control difuso y el control predictivo basado en modelo (MPC por sus siglas en inglés) [Colmegna et al., 2018]. Los controladores PID se han establecido como una solución confiable en muchos de los problemas encontrados en la actualidad, incluso para un problema complejo como este, se diseñan controladores robustos; algunos autores afirman que no es necesario el uso de controladores complejos y que las condiciones y desempeños del PA son perfectamente ejecutados por este método [Huyett et al., 2015]. Pero estas afirmaciones son refutadas en otras investigaciones, en las que se realizaron comparaciones que demuestran que el MPC tiene ventajas significativas sobre el desempeño de las otras dos propuestas [Pinsker et al., 2016, Doyle et al., 2014, Radziuk, 2012], una de ellas es la integración de restricciones físicas que permiten regular la cantidad parcial y total de insulina suministrada,

esto representa una gran ventaja porque el suministro de insulina es limitado y además el algoritmo de este controlador obtiene su exactitud de un proceso de optimización. Por estos motivos muchos autores han optado por escoger al MPC como la estrategia de control adecuada para su aplicación en el PA.

Algunos desarrollos que resaltan el uso del MPC para el PA son el propuesto en 2016, nombrado como control predictivo basado en modelo por zonas (ZMPC), que usa una función de costo asimétrica para suspender o reanudar las acciones de control dependiendo de los niveles de glucosa [Gondhalekar et al., 2016]. El algoritmo de control fue probado con simulaciones dando como resultado una regulación que no se limita a seguir una referencia, sino que mantiene al paciente en un rango estable. Sin embargo, se tienen problemas durante los periodos nocturnos al observar que el controlador podía inducir casos de hipoglucemia. Este problema se ha tratado posteriormente usando nuevas estrategias que integran dos MPC acoplados permitiendo la regulación adecuada y respondiendo también ante la ingesta de comida sin previo aviso [Gondhalekar et al., 2018], este trabajo es un referente debido a que es uno de los primeros en los que se programa la estrategia de control en un dispositivo móvil y las pruebas son realizadas en un estudio clínico de corta duración que suministra información del comportamiento del control y de los niveles de estas hormonas en pacientes reales. Para 2019 proponen una mejora al MPC modificando la función de costo que la hace adaptable a las velocidades de la tasa de cambio de la glucosa en la sangre [Shi et al., 2019]. Esto permite mejorar la respuesta del controlador ante los problemas que traen los incrementos y decrementos repentinos de glucosa debido a eventos de comida o ejercicio físico, pero su desventaja es que incrementa la sensibilidad del controlador ante posibles ruidos en el sensor.

También se han desarrollado controladores MPC pero con base en modelos impulsivos, este enfoque tiene validez en esta aplicación debido a que el tiempo de muestreo para la regulación de glucosa en un dispositivo automático está en alrededor de 5 min y el suministro de insulina se da en cuestión de segundos y en el caso de las inyecciones se hacen cada 6 u 8 horas por lo que considerar al proceso de regulación de la glucemia mediante la inyección de insulina tiene más sentido si se plantea desde el punto de vista de un impulso al de continuo o discreto. Otros trabajos presentan los resultados de este enfoque y como se han mejorado las estrategias para incluir variables artificiales que lleven al sistema entre equilibrios [Rivadeneira et al., 2018]. Incluso controladores denominados ZMPC con offset-free ZMPC-OF que tienen en cuenta la diferencia entre el modelo y la planta real permitiendo regular adecuadamente a los pacientes en periodos estacionarios cuando se presentan variaciones paramétricas por el estado de ánimo, el ciclo circadiano de la sensibilidad a la insulina, actividad física entre otros [F. et al., 2020].

En paralelo con los trabajos anteriores se han presentado desarrollos que dan una mejor claridad de cómo funciona la estrategia del MPC y sobre los métodos de optimización que

pueden ser usados para solucionar el problema cuadrático específicamente programado en sistemas embebidos. Estos dispositivos electrónicos son diseñados con propósitos específicos definidos por el usuario y dependiendo de la aplicación se pueden encontrar arquitecturas que integran un sistema operativo, generalmente distribuciones de Linux y Windows. En algunos trabajos se plantea el uso de paquetes de optimización programados en lenguajes de bajo nivel (C, C++) y discuten el desempeño de controladores enfocado a la función de costo, a las restricciones y las fallas que pueda tener en las pruebas de rendimiento del sistema embebido mientras ejecuta los algoritmos de control [Johansen, 2017]. Por otro lado, también se han enfocado a la implementación del MPC no lineal usando métodos de primer orden en donde se da el panorama de la complejidad del cálculo de optimización en línea por un sistema embebido y sobre su desempeño numérico, encontrando que se debe manejar una buena relación entre la precisión versus el rendimiento [Kouzoupis et al., 2015]. Estos trabajos a pesar de no estar enfocados al tratamiento de la T1DM desarrollan bases para el uso de los sistemas embebidos implementando al MPC como estrategia de control.

## 1.4. Antecedentes

En la literatura se presenta una gran cantidad de estrategias de control propuestas para regular la glucosa en la sangre y mitigar los impactos de la ingesta de comida, actividad física, periodos nocturnos, entre otros problemas asociados al desarrollo del PA en el tratamiento de la T1DM. La gran mayoría de estos trabajos se limitan a probar la confiabilidad, estabilidad y desempeño de los controladores propuestos mediante simuladores como el UVA/Padova [Dalla Man et al., 2014], el cual es aprobado y ampliamente aceptado por la comunidad científica. Sin embargo, a pesar de obtener buenos resultados en la etapa de simulación, se evidencia la poca o nula transferencia tecnológica de estos controladores en implementaciones reales, lo que puede generar resultados sesgados basados en supuestos, ya que no se consideran los aspectos reales asociados al hardware en el que se implementan como el consumo energético, la capacidad de cómputo de los procesadores y las condiciones de operación de los sensores y actuadores de la bomba. Algunos de los trabajos que llevaron a cabo la implementación de los algoritmos de control en tabletas o teléfonos inteligentes [Toffanin et al., 2019, Doyle et al., 2014, Gondhalekar et al., 2018, Berian et al., 2019] y reportan resultados favorables en la regulación de glucosa de los pacientes, pero no se pone a prueba la eficiencia de los dispositivos para realizar los cálculos ni la autonomía energética del dispositivo.

Al hablar de transferencia tecnológica en relación a los dispositivos médicos, surge la pregunta de qué tipo de elemento electrónico es el más apropiado, y aunque estudios demuestran que las personas se sienten menos avergonzadas por usar un celular que un dispositivo electrónico embebido, la discusión sobre la seguridad del paciente y cuál debe ser el elemento adecuado para esta aplicación sigue abierta hasta el día de hoy [Anderson et al., 2019].

La implementación de la estrategia del MPC en sistemas embebidos comenzó en 2016 añadiendo el concepto de hardware en el ciclo o por sus siglas en inglés (HIL) a las pruebas de PA. Los autores presentaron el panorama general de los algoritmos de control más usados para el tratamiento de la T1DM [Zavitsanou et al., 2016], tomando como base el ZMPC y realizando pruebas con varios paquetes de problemas de optimización de los cuales identifican a los llamados CVXOPT y al CVXPY como los mejores paquetes para la solución del problema cuadrático; Estas pruebas fueron ejecutadas sin considerar el ruido en la medición y solo se programó una familia de sistemas embebidos (Raspberry Pi 3 y zero). Este trabajo da un primer paso y abre las puertas para la solución de los problemas y retos que se tiene en la selección adecuada del procesador en un PA real y portátil.

En trabajos más recientes en la implementación de los algoritmos en sistemas embebidos se propuso la integración del MPC con la estrategia de eventos de activación (ET por sus siglas en inglés) que no dependen del tiempo, permitiendo reducir la frecuencia del cálculo del control debido a que ya no se hace de forma periódica si no que se toman como referencia otros parámetros que generan el evento de activación [Li and Shi, 2014, Eqdami et al., 2010]. Integrado al problema del tratamiento de la T1DM permite reducir el gasto energético asociado al procesador evitando que se haga el computo de la óptima entrega de insulina en cada instante de tiempo y solo se ejecute con eventos específicos, además se implementa un predictor lineal que muestra cuando los niveles deseados de glucosa se encuentran por debajo de 80 mg/dL para desactivar las funciones de la bomba previniendo eventos de hipoglucemia. Para el conjunto de pruebas se usa un modelo de ruido en el sensor de glucosa, una base de datos de 17 pacientes con T1DM y se compara el algoritmo de las estrategias por tiempo de activación y el de eventos de activación con el paquete solucionador CVXOPT, el controlador se implementa en una Raspberry Pi 3 usando HIL integrado con el simulador UVA/Padova [Chakrabarty et al., 2018]. En los resultados aseguran que la estrategia de activación de eventos reduce en aproximadamente 30% el gasto energético a pesar de que no se comprueban otro tipo de controladores o diferentes sistemas embebidos que pueden dar una mayor claridad en la selección adecuada de los componentes para el PA. Por último se presenta la implementación de un ZMPC en una plataforma embebida de bajos recursos liviana y portátil en el que se usa el ET y se prueba mediante HIL entre el arduino Feather M0 y el simulador UVA/Padova, mostrando la viabilidad de incorporar algoritmos de control de retroalimentación de glucosa en dispositivos médicos y de salud corporal [Chakrabarty et al., 2020].

Estas investigaciones acercan cada vez más a la implementación de los algoritmos de control en un PA portátil que cumpla con las características de regulación y sea eficiente en el uso de los recursos. Sin embargo, ninguno de estos trabajos hace énfasis en la selección de métodos y dispositivos disponibles adecuados para la aplicación, en conjunto con la integración de

estrategias para mejorar las probabilidades de tener un PA inteligente, confiable y portátil.

## 1.5. Alcance

Los sistemas embebidos usados para la bomba de insulina que serán probados poseen un sistema operativo con base en las distribuciones Linux, además todos los paquetes, librerías y lenguajes de programación usados para el algoritmo de control son de código abierto. El alcance será la implementación de una estrategia de control predictivo que use de forma eficiente los recursos energéticos del dispositivo y será verificado mediante la emulación del PA con un paciente virtual con T1DM.

El prototipo integrará las estrategias de MPC y eficiencia energética en un sistema embebido con sistema operativo y será evaluado aplicando los criterios de tiempo de ejecución de las tareas, consumo energético, temperatura del procesador, y objetivos de regulación de glucosa reportando sus resultados en informes y publicaciones científicas.

Como resultado de este trabajo se han logrado los siguientes productos:

Un artículo publicado en la categoría A1

- Jhon E. Goez, María F. Villa-Tamayo, Monica Vallejo y Pablo S. Rivadeneira “Performance Analysis of Different Embedded Systems and Open-Source Optimization Packages Towards an Impulsive MPC Artificial Pancreas,” *Front. Endocrinol. (Lausanne)*., vol. 12, no. April, pp. 1–15, 2021, doi: 10.3389/fendo.2021.662348.

Un artículo en desarrollo

- Jhon E. Goez, Monica Vallejo y Pablo S. Rivadeneira “Emulation of an artificial pancreas for the treatment of type 1 diabetes mellitus as a portable device with limited life battery”.

Un artículo en el congreso WEA 2021, con publicación en un volumen de la revista Springer.

- Jhon E. Goez, Monica Vallejo y Pablo S. Rivadeneira “Towards event-trigger impulsive MPC for the treatment of T1DM handling limited resources.” *Applied Computer Sciences in Engineering, Springer Springer-Nature Communications in Computer and Information Sciences (CCIS)*, Pages 180-189, October 6–8, 2021, ISBN 978-3-030-86702-7.

Un poster publicado en conferencia

- Jhon E. Goez y Monica Vallejo “Programming A Predictive Control Algorithm In Embedded System for artificial pancreas (AP)” 15th International Symposium on Medical Information Processing and Analysis, Medellín Colombia. Noviembre 13 2019.

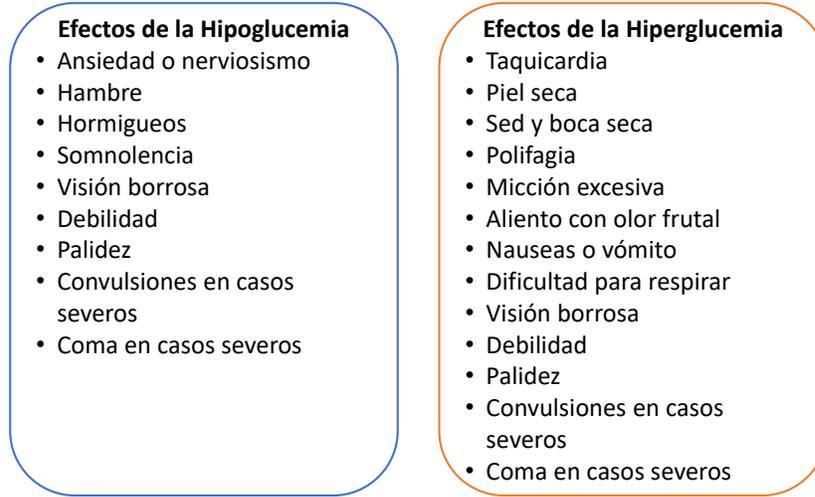
# 2 Modelo, Controlador, estimador y paquetes de optimización

## 2.1. Modelos

La representación matemática de la dinámica de las hormonas de la insulina y la glucosa en pacientes diabéticos ha sido uno de los principales objetivos en el desarrollo de tratamientos autónomos de esta enfermedad que se ha trabajado desde hace más de 30 años. Obtener un modelo que represente adecuadamente estas dinámicas ha llevado a varias investigaciones a presentar soluciones que parten desde, los modelos empíricos o de regresión denominados de caja negra en donde los parámetros no tienen una asociación a la variable física o al comportamiento fisiológico de las hormonas, hasta llegar a los modelos analíticos de los cuales parten del fenómeno que se observa y se trata de explicar mediante un modelo matemático que represente adecuadamente cada dinámica involucrada en el proceso, incluso en algunos casos se obtienen modelos que toman parte de los dos principios y combinan interpretaciones físicas de las dinámicas y datos obtenidos de la excitación de las entradas y la medición de las salidas.

Una de las razones para invertir tanto esfuerzo en obtener modelos que se ajusten adecuadamente al comportamiento de estas hormonas en los pacientes con T1DM, es que de no regularse adecuadamente el paciente puede llegar a estados de hiperglucemia: estado en el que se encuentra una alta concentración de glucosa en la sangre (aproximadamente superior a 180 mg/dl) y presentar hiperglucemia severa en niveles superiores a 250 mg/dl, por otro lado si no se regula adecuadamente y se inyectan excesivas dosis de insulina se pueden llevar al paciente a un caso de hipoglucemia con niveles inferiores a 70 mg/dl hasta llegar a una hipoglucemia severa aproximadamente por debajo de 54 mg/dl siendo este estado uno de los más peligrosos para los seres humanos, algunos de los efectos que causan estos niveles de glucemia se observan en la Figura 2-1.

En esta sección se presentan algunos de los modelos que se observan en la literatura que pueden servir como base para el desarrollo de los controladores y del observador en el PA.



**Figura 2-1:** Efectos de los casos de hiper e hipoglucemia

### 2.1.1. Modelo de Bergman

Este modelo de las hormonas de la glucosa y la insulina en forma de ecuaciones diferenciales no lineales se basa en determinar la fidelidad del comportamiento del modelo con respecto a datos, comprobando que su estimación tiene una buena reproductibilidad en inyecciones experimentales de 300mg/Kg. Este modelo está basado en experimentos con animales y posteriormente en humanos para el análisis de los datos que se obtienen de la glucosa y la insulina en la sangre de los pacientes cuando se elevan progresivamente los niveles de estas hormonas. [Bergman et al., 1979].

En la Ecuación 2-1 se describe a  $G(t)$  como la concentración de glucosa en plasma medida en [mg/dL] en el tiempo [min], a  $X(t)$  como el efecto de la insulina sobre la concentración de glucosa medida en [ $\text{min}^{-1}$ ], a  $I(t)$  como la concentración de insulina en plasma en [ $\mu$  U/ml] en el tiempo y a  $\mu(t)$  como la función que representa la ingesta de carbohidratos en el cuerpo. El parámetro  $p_1$  hace referencia al valor inicial de la iteración glucosa-insulina en [mg/dl],  $p_2$  la tasa de la disminución de glucosa por unidad de insulina en [ $\text{min}^{-1}$ ],  $p_3$  tasa de incremento de glucosa después del efecto de la insulina,  $G_b$  el nivel de glucosa basal en [mg/dl],  $I_b$  el nivel máximo de insulina basal en [ $\mu$  U/ml] y a  $n$  como el orden de decrecimiento de la insulina en plasma [ $\text{min}^{-1}$ ],

$$\begin{aligned}
 \frac{G(t)}{dt} &= -(p_1 + X(t))G(t) + p_1G_b + p(t), \\
 \frac{X(t)}{dt} &= -p_2X(t) + p_3(I(t) - I_b), \\
 \frac{I(t)}{dt} &= -n(I(t) - I_b) + \mu(t).
 \end{aligned}
 \tag{2-1}$$

### 2.1.2. Modelo de Magdelaine

Este modelo está respaldado por la identificación a partir de datos clínicos de más de dos días y se enfocan en presentar propiedades de estabilidad realistas y de ajuste a largo plazo. La tasa basal de insulina y la sensibilidad a la insulina son subproductos de este modelo matemático [Magdelaine et al., 2015]. También incluye la producción endógena de glucosa y la absorción de glucosa debido a las células del sistema nervioso. El modelo de ecuaciones diferenciales se presenta en espacios de estado como se muestra en la ecuación 2-4.

La dinámica de la glucosa se representa en una sola ecuación, en donde  $\dot{G}(t)$  es la producción menos el consumo de glucosa. El incremento involucra la producción endógena del hígado y la digestión de *CHO*  $D(t)$ , en donde toman como constante la producción de glucosa endógena  $k_1$ . La dinámica de la insulina está dada por un modelo de segundo orden, la función de transferencia de la ecuación 2-2 se obtiene de la transformada de Laplace de  $u(t)$  e  $i(t)$  como  $U(s)$  e  $I(s)$ ,

$$\begin{aligned} \dot{G} &= -k_{si}I(t) + k_i - k_b + D(t), \\ I(s) &= \frac{(K_u/V_i)}{(1+T_u \cdot s)}U(s). \end{aligned} \quad (2-2)$$

Para el modelo de la digestión se usa un sistema de segundo orden presentado en la ecuación 2-3 en espacio de estados para la relación que tiene el *CHO* en las comidas  $r(t)$  y  $D(t)$ . Además, usan como constante de tiempo a  $T_r$  y el volumen de sangre  $V_B$  para esta dinámica.  $T_r$  y la ganancia estática  $k_r/V_B$  están ajustados para reflejar el índice glucémico dependiendo de la composición de la comida.

$$\begin{bmatrix} \dot{D} \\ \dot{D} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -\frac{1}{T_r^2} & -\frac{2}{T_r} \end{bmatrix} \begin{bmatrix} D \\ \dot{D} \end{bmatrix} + \begin{bmatrix} 0 \\ -\frac{k_r}{V_B \cdot T_r^2} \end{bmatrix} r(t), \quad (2-3)$$

Estas ecuaciones son presentadas en espacios de estado para la representación total del modelo usando como vector de estados  $X(t) = [G(t) \ I(t) \ \dot{I}(t) \ D(t) \ \dot{D}(t)]'$ , en donde las entradas son  $u(t)$  y  $r(t)$  y la salida es  $y = G(t)$ ,

$$\dot{X}(t) = \begin{bmatrix} 0 & -k_{si} & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & -\frac{1}{T_r^2} & -\frac{2}{T_u} & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & -\frac{1}{T_r^2} & -\frac{2}{T_u} \end{bmatrix} X(t) + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ \frac{k_u}{V_i \cdot T_u^2} & 0 \\ 0 & 0 \\ 0 & \frac{k_r}{V_B \cdot T_r^2} \end{bmatrix} \begin{bmatrix} u(t) \\ r(t) \end{bmatrix} + \begin{bmatrix} k_1 - k_b \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \quad (2-4)$$

$$Y = CX(t) = [1 \ 0 \ 0 \ 0 \ 0]X(t).$$

### 2.1.3. Modelo de Hovorka

Este modelo se construye usando datos de personas adultas con este tipo de diabetes, a los que se les midió los niveles de glucosa usando la administración continua de insulina en un ciclo cerrado durante 24 horas por 12 semanas. Se proponen 5 ecuaciones diferenciales lineales para representar el modelo por compartimentos, los parámetros del modelo se estimaron utilizando el enfoque de Monte Carlo de la cadena de Markov dentro de un marco de modelo bayesiano jerárquico, en esta propuesta aseguran que puede facilitar las pruebas in silico para ayudar al desarrollo de sistemas de administración de insulina de circuito cerrado [Ruan et al., 2017].

El modelo para la dinámica de la glucosa-insulina lo describen en tres submodelos que se observan en las ecuaciones 2-5, en estos se toma a la administración subcutánea de insulina y la ingesta de carbohidratos como entradas y la monitorización de la glucosa como salida,

$$\begin{aligned}
 \frac{dx_1(t)}{dt} &= -\frac{1}{t_{max,IA}}x_1(t) + \frac{u_i(t)}{60}, \\
 \frac{dx_2(t)}{dt} &= \frac{1}{t_{max,IA}}(x_1(t) - x_2(t)), \\
 X(t) &= \frac{1000*x_2(t)}{t_{max,IA}MCR_IW}, \\
 \frac{da_1(t)}{dt} &= -\frac{1}{t_{max,G}}a_1(t) + \delta_{t_j}(t)u_G(t_j), \\
 \frac{da_2(t)}{dt} &= \frac{1}{t_{max,G}}(a_1(t) - a_2(t)), \\
 U_M(t) &= \frac{5,556*AGa_2(t)}{t_{max,IA}MCR_IW}, \\
 \frac{G(t)}{dt} &= -S_I(X(t) - X_b) + U_M(t) - K(G(t) - G_b),
 \end{aligned} \tag{2-5}$$

En el submodelo de absorción de insulina de la ecuación 2-5,  $x_1(t)$  y  $x_2(t)$  representa la cantidad de insulina eficaz entre el primer y segundo compartimento,  $u_i(t)$  representa la tasa de administración de insulina exógena,  $X(t)$  es la concentración de insulina efectiva,  $t_{max,IA}$  es el tiempo hasta el máximo de concentración de insulina efectiva,  $W$  es el peso del paciente y el  $MCR_I$  es la tasa de aclaramiento metabólico de la insulina efectiva fijada.

La absorción de comida es representada por dos compartimentos en donde  $a_1(t)$  y  $a_2(t)$  son la cantidad de carbohidratos en el primer y segundo compartimento de absorción de comida,  $u_G(T_j)$  representa los carbohidratos ingeridos en el tiempo  $t_j(g)$ ,  $t_{max,G}$  es el tiempo máximo de la tasa de aparición de glucosa,  $AG$  es la biodisponibilidad fraccionada,  $V_G$  es el tamaño de la reserva de glucosa plasmática y  $U_M(t)$  es la tasa de absorción intestinal de carbohidratos con la unidad convertida en tasa de cambio de concentración de glucosa.

La dinámica de la glucosa está representada por un solo compartimento en donde  $G(t)$  es la concentración de glucosa en sangre,  $S_I$  es la sensibilidad a la insulina,  $X_b$  es la concentración

de insulina basal efectiva a la que el nivel de glucosa se mantiene constante,  $G_b$  es el nivel de glucosa basal y  $K$  la tasa fraccional de autorregulación de la glucosa.

### 2.1.4. Modelo implementado

Para este trabajo se implementa un modelo fisiológico debido a que permite dar una interpretación a los parámetros del modelo y se usa como base para los controladores propuestos y el estimador. El modelo lineal desarrollado en [Hoyos et al., 2021] se basa en el modelo de Hovorka y la identificación de parámetros de la versión comercial del simulador UVA/Padova en 10 pacientes con T1DM. El modelo matemático se compone de tres partes fundamentales: La dinámica de la glucosa, la absorción de insulina y la absorción de comidas como se observa en la ecuación 2-6,

$$\begin{aligned}
\frac{dG(t)}{dt} &= \theta_0 - \theta_1 G(t) - \theta_2 Q_i(t) + \theta_3 Q_g(t), \\
\frac{dQ_i(t)}{dt} &= -\frac{1}{\theta_4} Q_i(t) + \frac{1}{\theta_4} Q_{isub}(t), \\
\frac{dQ_{isub}(t)}{dt} &= -\frac{1}{\theta_4} Q_{isub}(t) + \frac{1}{\theta_4} u(t), \\
\frac{dQ_g(t)}{dt} &= -\frac{1}{\theta_5} Q_g(t) + \frac{1}{\theta_5} Q_{sto}(t), \\
\frac{dQ_{sto}(t)}{dt} &= -\frac{1}{\theta_5} Q_{sto}(t) + \frac{1}{\theta_5} r(t).
\end{aligned} \tag{2-6}$$

La descripción de las variables y parámetros de las ecuaciones se presenta en la Tabla **2-1**. Para usarlo en la propuesta de los controladores el modelo se expresa en espacios de estado (ecuación 2-7), en donde  $X(t) = [x_1(t) \ x_2(t) \ x_3(t) \ x_4(t) \ x_5(t)]'$  con  $x_1(t) = G$ ,  $x_2(t) = Q_i$ ,  $x_3(t) = Q_{isub}$ ,  $x_4(t) = Q_g$  y  $x_5(t) = Q_{sto}$ . La salida  $y(t)$  es la glucemia a controlar  $x_1(t)$  y  $E$  es una matriz constante (término afín) que representa la producción de glucosa endógena basal en estado estacionario.

$$\begin{aligned}
x(k+1) &= Ax(k) + B_u u(k) + B_r r(k) + E, \\
y(k) &= Cx(k),
\end{aligned} \tag{2-7}$$

Las matrices del modelo están relacionadas con su contraparte continua considerando un tiempo de muestreo fijo  $T$  como  $A = e^{A_c T}$ ,  $B_u = e^{A_c T} B_{uc}$ ,  $B_r = \int_0^T e^{A_c s} ds B_{rc}$ ,  $E = e^{A_c T} E_c$ , con:

$$A_c = \begin{bmatrix} -\theta_1 & -\theta_2 & 0 & \theta_3 & 0 \\ 0 & \frac{-1}{\theta_4} & \frac{1}{\theta_4} & 0 & 0 \\ 0 & 0 & \frac{-1}{\theta_4} & 0 & 0 \\ 0 & 0 & 0 & \frac{-1}{\theta_5} & \frac{1}{\theta_5} \\ 0 & 0 & 0 & 0 & \frac{-1}{\theta_5} \end{bmatrix}, \quad B_{uc} = \begin{bmatrix} 0 \\ 0 \\ \frac{1}{\theta_4} \\ 0 \\ 0 \end{bmatrix}, \quad B_{rc} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ \frac{1}{\theta_5} \end{bmatrix}, \quad E_c = \begin{bmatrix} \theta_0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \quad C = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}. \tag{2-8}$$

**Tabla 2-1:** Descripción de variables y parámetros del modelo.

Variable/Parámetro	Descripción	Unidades
$G(t)$	Concentración de glucosa en sangre	mg/dL
$Q_i(t)$	Tasa de administración de insulina en plasma	U/min
$Q_g(t)$	Tasa de absorción de carbohidratos del intestino	g/min
$Q_{i,sub}(t)$	Tasa de administración de insulina subcutánea	U/min
$Q_{sto}(t)$	Tasa de suministro de glucosa desde el estómago	g/min
$u(t)$	Tasa de infusión de insulina (bolo y basal)	U/min
$r(t)$	Tasa de ingesta de carbohidratos (CHO)	g/min
$G_b$	Tasa de glucosa basal	mg/dL.min
$U_b$	Tasa de insulina basal	mg/dL.min
$\theta_0$	Producción endógena de glucosa	1/min
$\theta_1$	Autorregulación hepática	1/min
$\theta_2$	Sensibilidad a la insulina	mg/dL.U
$\theta_3$	Biodisponibilidad de carbohidratos	mg/dL.g
$\theta_4$	Tiempo hasta la máxima concentración efectiva de insulina	min
$\theta_5$	Tasa de aparición máxima de glucosa en tiempo	min

Este es el modelo que se usará en todas las simulaciones y emulaciones en el presente trabajo tomando como base que se ha usado previamente en trabajos como [Villa-Tamayo et al., 2021b, Villa-Tamayo and Rivadeneira, 2020b], en las pruebas se usan los parámetros de cada uno de los 10 pacientes identificados en la Tabla 2-2 y la representación en espacios de estados de la ecuación 2-8 para los controladores y el observador en cada prueba.

**Tabla 2-2:** Parámetros estimados de 10 pacientes adultos

Paciente	1	2	3	4	5	6	7	8	9	10
$\theta_0$	1.327	2.011	0.757	1.252	1.036	2.181	1.901	1.057	2.073	1.074
$\theta_1$	0.0034	0.0063	0.0010	0.0027	0.00218	0.0081	0.0018	0.0028	0.0058	0.0032
$\theta_2$	44.223	54.185	25.271	58.349	49.843	37.488	79.44	38.425	88.105	34.591
$\theta_3$	2.308	2.410	1.737	2.961	3.871	4.173	4.379	4.370	4.459	2.51
$\theta_4$	56.001	40.004	52.202	59.502	46.782	52.503	47.505	50.007	50.505	50.503
$\theta_5$	21.840	14.624	21.516	25.429	28.638	23.511	22.023	26.854	24.461	23.317

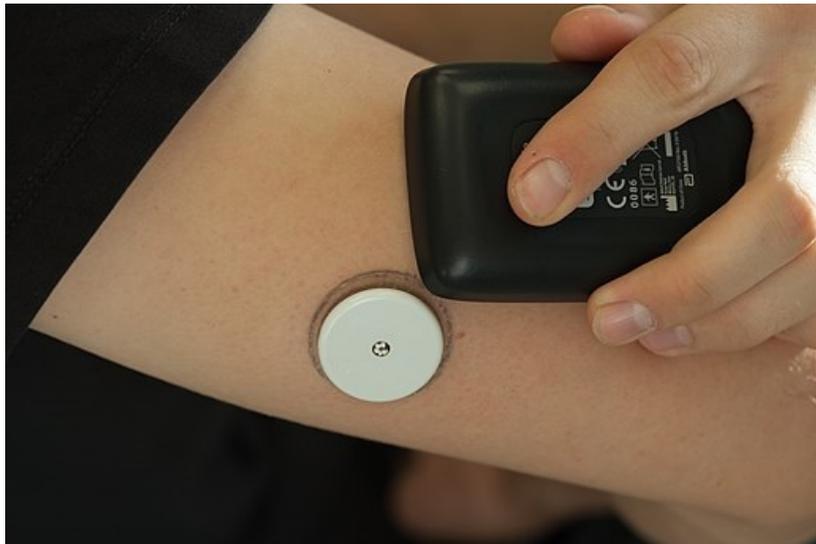
## 2.2. Monitoreo

Tener un sistema que permita obtener valores confiables y a tiempo es imprescindible para realizar un proceso de regulación o variación de referencia. La mayoría de los CGM usan un periodo de muestreo de 5 minutos y obtienen los niveles de glucosa intersticial a través de un sensor de invasión mínima subcutánea comparado con el automonitoreo de glucosa

en sangre (SMBG por sus siglas en inglés), que requiere de contacto con la sangre para obtener la medición. Actualmente la adquisición de datos mediante CGM se considera lo suficientemente precisa para confiar y tomar decisiones sobre la inyección de insulina a pesar de que se conoce el retardo que tienen este tipo de mediciones y que su bio-fluido para reaccionar en la toma de la muestra solo duran una algunos días según el fabricante [Jung, 2016, Vettoretti et al., 2018].

### 2.2.1. Sensores para CGM

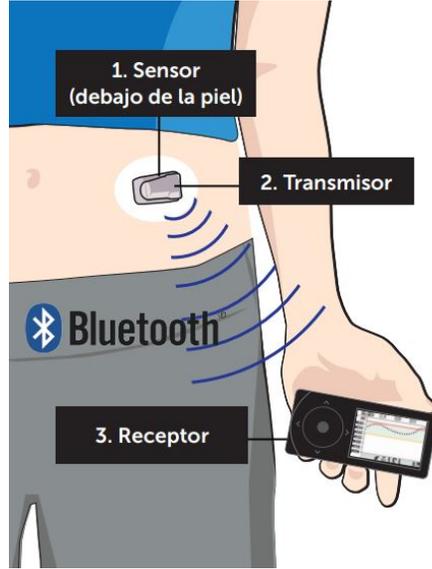
Uno de los sensores disponibles actualmente en el mercado para la medición de glucosa subcutánea es el *FreeStyle Libre* de Abbott, este sensor de 35mm de diámetro y 5mm de espesor se adhiere a la piel como se observa en la Figura 2-2, Toma medidas cada minuto y almacena las lecturas en intervalos de 15 minutos, transcurridos 15 minutos se calcula un promedio de la glucosa del paciente almacenando datos hasta 8 horas seguidas, luego de esto el espacio de memoria se sobrescribe. La duración del sensor es de máximo 14 días y los datos pueden leerse mediante un dispositivo de lectura de comunicación de campo cercano (NFC por sus siglas en inglés) o una aplicación móvil (LibreLink App).



**Figura 2-2:** Uso del sensor *FreeStyle Libre*. Esta foto de Autor desconocido está bajo licencia CC BY-SA.<https://creativecommons.org/licenses/by-sa/3.0/>

También se encuentra el *Dexcom G5 mobile*, este sistema permite tener lecturas continuas del sensor de glucosa, en tiempo real cada cinco minutos durante un máximo de siete días. El uso del *Dexcom G5* se compone de 3 partes fundamentales: el sensor y el aplicador, son elementos desechables y se usan para insertar el sensor que tomará las medidas durante 7 días. El transmisor, que se inserta en el compartimento del sensor para enviar la información

de la glucosa a su receptor de forma inalámbrica, este es reutilizable durante los tres meses que dura la batería. Por último el receptor de información, que es el dispositivo que almacena y muestra el histórico de la glucosa del paciente. Figura 2-3.



**Figura 2-3:** Uso del sensor Dexcom G5. Imagen tomada de [Dexcom, 2015]

### 2.2.2. Modelo de ruido sensor CGM

En este trabajo no se va a implementar físicamente un sensor, sin embargo, para darle validez al escenario de emulación del PA se implementa un modelo de ruido del CGM inherente a todos los procesos de medición. Para esto se consideró el modelo de ruido del sensor detallado en [Vettoretti et al., 2018]. Esta señal de ruido se simula junto con el paciente virtual (en Matlab) en cada instante de tiempo  $k$ , y es el mismo modelo para todos los experimentos en este trabajo. El modelo del ruido CGM tiene en cuenta tres componentes de inexactitud: La distorsión debida a la cinética sangre-glucosa intersticial ( $BG$ -a- $IG$ ), las subestimaciones / sobrestimaciones sistemáticas (con amplitud variable en el tiempo) debido a errores en la calibración del sensor y el ruido de medición aleatorio. Los parámetros del modelo se basaron en una base de datos recopilada de 76 adultos monitoreados por el dispositivo *Dexcom G5 Mobile*, y cada conjunto de datos también incluye mediciones YSI (*YSI, Yellow Springs*) de alta exactitud y precisión que se registraron en una sesión clínica de 12 h. La ecuación 2-9 describe el ruido de medición.

$$E_{CGM}(kT) = (a_0 + a_1 kT)x_{IG}(kT) + b_0 + r(kT), \quad (2-9)$$

$$r(kT) = c_1 r((k-1)T) + c_2 r((k-2)T) + \epsilon(kT).$$

En donde  $T$  es el período de muestreo del sensor,  $\epsilon(kT)$  representa el ruido de medición aditivo siguiendo la distribución gaussiana con media cero y varianza  $\sigma_{in}^2$ , y  $x_{IG}(kT)$  es el

valor de glucosa intersticial ( $IG$ ) en el momento  $kT$ . El perfil de  $IG$ ,  $x_{IG}(kT)$ , se describe mediante la convolución del perfil de  $BG$ ,  $x_{BG}(kT)$  y la respuesta al impulso del sistema cinético de  $BG$  a  $IG$  que se describe mediante un sistema dinámico de primer orden con ganancia unitaria y constante de tiempo  $\tau[1/min]$ ,

$$x_{IG}(kT) = x_{BG} \frac{1}{\tau} e^{-kT/\tau}. \quad (2-10)$$

Posteriormente la señal es sumada con la medida de glucosa  $G(t)$  como  $x_{IG}(kT)$  y esta es la que se envía al controlador-estimador como medida del CGM.

## 2.3. Estrategias de control

Con base en los artículos e investigaciones encontradas en la literatura, en este trabajo se prueban inicialmente cuatro estrategias de control, todas enmarcadas dentro del MPC y con diferentes niveles de complejidad que involucran mayor cantidad de variables de decisión o mayor número de restricciones lineales. La idea simple de un MPC es realizar una predicción de los estados mediante un proceso de optimización que obtiene una cadena de controles en una ventana de tiempo  $H_c$ , pero solo aplica al paciente el primer valor  $H_c[1]$  en el instante de tiempo  $k$ . La función de costo a minimizar se basa en el modelo del sistema y en las restricciones físicas del mismo. Se debe tener en cuenta que para todos los controladores probados se usa un modelo impulsivo.

### 2.3.1. Control predictivo basado en modelo

El MPC estándar consiste en una función de costo que penaliza la desviación del estado de una referencia  $x_{ref}$ , la desviación de la entrada  $u$  a una entrada de referencia  $u_{ref}$  en un horizonte de control  $H_c$ , y la diferencia entre el estado y el objetivo al final del horizonte de predicción  $H_p$  [Maciejowski, 2001]. También se consideran las restricciones de entrada y el estado estimado  $\hat{x}$  en cada instante de tiempo  $k$ , como se observa en la ecuación 2-11.

$$\begin{aligned} \min_{\mathbf{u}, \mathbf{x}} \quad & \sum_{j=0}^{H_p-1} \|x(j) - x_{ref}\|_Q^2 + \sum_{j=0}^{H_c-1} \|u(j) - u_{ref}\|_R^2 + \|x(H_p) - x_{ref}\|_P^2 \\ \text{s.t.} \quad & x(0) = \hat{x}(k), \\ & x(j+1) = Ax(j) + B_u u(j) + B_r r(j) + E, \\ & u(j) \in \mathcal{U}, \quad x(j) \in \mathcal{X}. \end{aligned} \quad (2-11)$$

La solución de este problema es la trayectoria de entrada óptima  $\mathbf{u} = \{u(0), \dots, u(H_c - 1)\}$ , y solo el primer elemento de la secuencia se aplica a la planta. Cuando hay nueva información

de medición disponible, el problema de optimización se reformula y se resuelve en el siguiente instante de tiempo  $k + 1$ , y así sucesivamente. El esquema general del PA usando MPC se observa en la Figura 2-4.

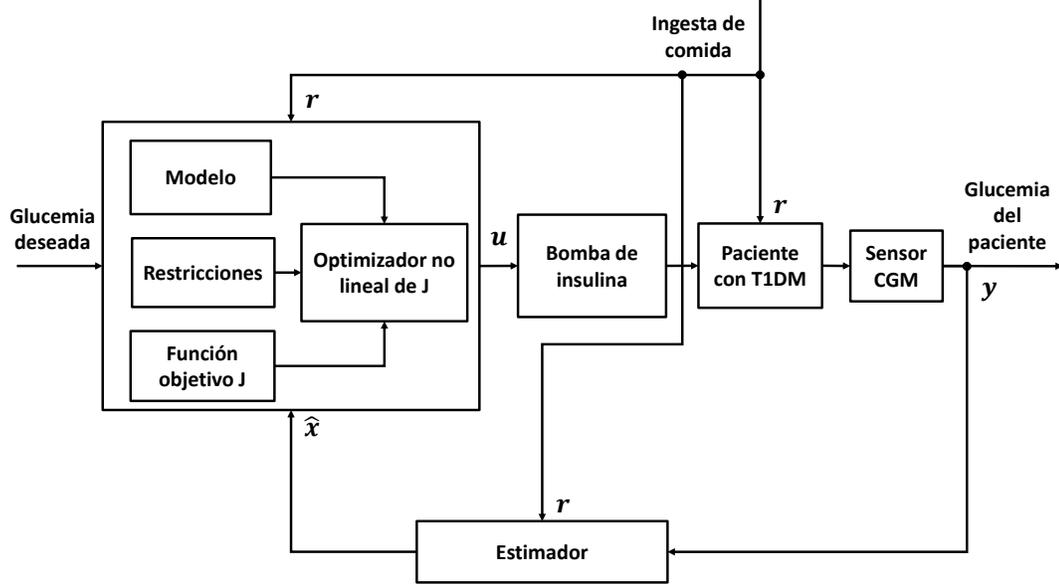


Figura 2-4: Esquema de control para el PA con MPC

A continuación, se muestra la representación matricial del problema de optimización tomando la expansión del estado en el horizonte de predicción como:

$$X_M = A_M x(k) + B_{uM} U + B_{rM} \Gamma + E_M. \quad (2-12)$$

$$\begin{bmatrix} x(k+1) \\ \vdots \\ x(k+H_c) \\ \vdots \\ x(k+H_p) \end{bmatrix} = \begin{bmatrix} A \\ \vdots \\ A^{H_c} \\ \vdots \\ A^{H_p} \end{bmatrix} x(k) + \begin{bmatrix} B_u & \cdots & 0 \\ \vdots & \ddots & \vdots \\ A^{H_c-1} B_u & \cdots & B_u \\ \vdots & \ddots & \vdots \\ A^{H_p-1} B_u & \cdots & \sum_{i=0}^{H_p-H_c} A^i B_u \end{bmatrix} \begin{bmatrix} u(k) \\ \vdots \\ u(k+H_c-1) \end{bmatrix} + \begin{bmatrix} B_r & \cdots & 0 \\ \vdots & \ddots & \vdots \\ A^{H_c-1} B_r & \cdots & 0 \\ \vdots & \ddots & \vdots \\ A^{H_p-1} B_r & \cdots & B_r \end{bmatrix} \begin{bmatrix} r(k) \\ \vdots \\ r(k+H_p-1) \end{bmatrix} + \begin{bmatrix} E \\ \vdots \\ \sum_{i=0}^{H_p-1} A^i E \end{bmatrix}. \quad (2-13)$$

Con esto se puede reescribir la función de costo como:

$$V(k) = \frac{1}{2}x^T Hx + f^T x, \quad (2-14)$$

Donde

$$H = 2 * (B_{uM}^T Q_{ex} B_{uM} + R_{ex}), f = 2 * E_{err} Q_{ex} B_{uM}, \quad (2-15)$$

Con  $E_{err} = [I * u_{ref}] - A_M x(k) - E_M - B_{rM} \Gamma$  y las matrices de sintonía:

$$Q_{ex} = \begin{bmatrix} Q & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & Q \end{bmatrix}_{H_p * n_x \times n_x * H_p}, \quad R_{ex} = \begin{bmatrix} Q & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & Q \end{bmatrix}_{H_c * n_u \times n_u * H_c}. \quad (2-16)$$

Y para las restricciones de desigualdad

$$\begin{bmatrix} a_y \\ a_u \\ I \\ -I \end{bmatrix} \leq \begin{bmatrix} b_y \\ b_u \\ u_{max} \\ -u_{min} \end{bmatrix}, \quad (2-17)$$

Tomando como  $I$  la matriz identidad

$$a_y = \begin{bmatrix} B_{uM} \\ -B_{uM} \end{bmatrix}, \quad a_u = \begin{bmatrix} M \\ -M \end{bmatrix}, \quad M = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ -1 & 1 & & \\ 0 & -1 & & \vdots \\ \vdots & & \ddots & \\ 0 & \cdots & & 1 & 0 \\ & & & -1 & 1 \end{bmatrix}, \quad (2-18)$$

$$b_u = \begin{bmatrix} u_{max} + T_{aux} u(k) \\ -T_{aux} u(k) - u_{min} \end{bmatrix}, \quad b_y = \begin{bmatrix} B_2 \\ -B_1 \end{bmatrix},$$

Con  $T_{aux} = ones(H_c, 1)$ ,  $B_1 = Y_{min} - CA_M x(k) - E_M - B_{rM} \Gamma$  y  $B_2 = Y_{max} - CA_M x(k) - E_M - B_{rM} \Gamma$ .

### 2.3.2. Control predictivo basado en modelo por zonas

Posteriormente se le da al MPC un enfoque más realista en donde se amplía el espacio viable al no considerar el seguimiento de una referencia sino la permanencia de la planta en una

zona segura (ZMPC), en donde se desea mantener a la variable controlada, en este caso los niveles glucemia del paciente [Grosman et al., 2010]. El ZMPC se formula agregando una nueva variable de decisión  $\delta$  y definiendo sus límites superior e inferior. Esto crea una zona tal que, cuando las variables predichas están dentro de ella, el costo es igual a cero. El problema de ZMPC a resolver en cada instante de tiempo  $k$  es descrito en la ecuación 2-19.

$$\begin{aligned} \min_{\mathbf{u}, \mathbf{x}, \delta} \quad & \sum_{j=0}^{H_p-1} \|x(j) - x_{ref} + \delta\|_Q^2 + \sum_{j=0}^{H_c-1} \|u(j) - u_{ref}\|_R^2 + \|x(H_p) - x_{ref} + \delta\|_P^2 \\ \text{s.t.} \quad & x(0) = \hat{x}(k), \\ & x(j+1) = Ax(j) + B_u u(j) + B_r r(j) + E, \\ & u(j) \in \mathcal{U}, \quad x(j) \in \mathcal{X}, \\ & \delta_{min} \leq \delta \leq \delta_{max}. \end{aligned} \tag{2-19}$$

Los parámetros  $x_{ref}$  y  $u_{ref}$  se seleccionan como referencia dentro de la zona objetivo. Este planteamiento modifica a la Figura 2-4 en donde se usa la Glucosa deseada como *set-point* remplazándolo por el de rango de glucemia segura.

A continuación, se muestra la representación matricial del problema de optimización tomando la expansión del estado en el horizonte de predicción como:

$$X_M = A_M x(k) + B_{uM} U + B_{rM} \Gamma + E_M, \tag{2-20}$$

$$\begin{aligned} \begin{bmatrix} x(k+1) \\ \vdots \\ x(k+H_c) \\ \vdots \\ x(k+H_p) \end{bmatrix} &= \begin{bmatrix} A \\ \vdots \\ A^{H_c} \\ \vdots \\ A^{H_p} \end{bmatrix} x(k) + \begin{bmatrix} B_u & \cdots & 0 \\ \vdots & \ddots & \vdots \\ A^{H_c-1} B_u & \cdots & B_u \\ \vdots & \ddots & \vdots \\ A^{H_p-1} B_u & \cdots & \sum_{i=0}^{H_p-H_c} A^i B_u \end{bmatrix} \begin{bmatrix} u(k) \\ \vdots \\ u(k+H_c-1) \end{bmatrix} + \\ & \begin{bmatrix} B_r & \cdots & 0 \\ \vdots & \ddots & \vdots \\ A^{H_c-1} B_r & \cdots & 0 \\ \vdots & \ddots & \vdots \\ A^{H_p-1} B_r & \cdots & B_r \end{bmatrix} \begin{bmatrix} r(k) \\ \vdots \\ r(k+H_p-1) \end{bmatrix} + \begin{bmatrix} E \\ \vdots \\ \sum_{i=0}^{H_p-1} A^i E \end{bmatrix}. \end{aligned} \tag{2-21}$$

Con esto se puede reescribir la función de costo como:

$$V(k) = \frac{1}{2} x^T H x + f^T x, \tag{2-22}$$

Donde

$$H = 2 * \begin{bmatrix} B_{uM}^T Q_{ex} B_{uM} + R_{ex} & B_{uM}^T Q_{ex} \rho \\ (B_{uM}^T Q_{ex} B_{uM})^T & \rho^T Q_{ex} \rho \end{bmatrix}, \quad (2-23)$$

$$f = -2 * [E_{err}^T Q_{ex} B_{uM} + u_{ref} R_{ex} \quad E_{err}^T Q_{ex} \rho]^T,$$

Con  $E_{err} = [I * u_{ref}] - A_M x(k) - E_M - B_{rM} \Gamma$  y las matrices de sintonía:

$$Q_{ex} = \begin{bmatrix} Q & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & Q \end{bmatrix}_{H_p * n_x \times n_x * H_p}, \quad R_{ex} = \begin{bmatrix} Q & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & Q \end{bmatrix}_{H_c * n_u \times n_u * H_c}. \quad (2-24)$$

Y para las restricciones de desigualdad:

$$\begin{bmatrix} a_y \\ a_u \\ I \\ -I \\ \delta_{min} \end{bmatrix} \leq \begin{bmatrix} b_y \\ b_u \\ u_{max} \\ -u_{min} \\ \delta_{max} \end{bmatrix}, \quad (2-25)$$

Tomando como  $I$  la matriz identidad

$$a_y = \begin{bmatrix} B_{uM} \\ -B_{uM} \end{bmatrix}, \quad a_u = \begin{bmatrix} M \\ -M \end{bmatrix}, \quad M = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ -1 & 1 & & \\ 0 & -1 & & \vdots \\ \vdots & & \ddots & \\ 0 & \cdots & & 1 & 0 \\ 0 & & & -1 & 1 \end{bmatrix}, \quad (2-26)$$

$$b_u = \begin{bmatrix} u_{max} + Tu(k) \\ -Tu(k) - u_{min} \end{bmatrix}, \quad b_y = \begin{bmatrix} B_2 \\ -B_1 \end{bmatrix},$$

Con  $T = ones(H_c, 1)$ ,  $B_1 = Y_{min} - CA_M x(k) - E - Br_{rM} \Gamma$  y  $B_2 = Y_{max} - CA_M x(k) - E_M - B_{rM} \Gamma$ .

### 2.3.3. Control predictivo basado en modelo por zonas con variables artificiales

La idea principal del ZMPC con variables artificiales (ZMPC-AV) es dirigir el sistema desde un punto  $(x_t, u_t)$  en el conjunto objetivo  $(\mathcal{X}_s^{Tar}, \mathcal{U}_s^{Tar})$  a través de un equilibrio del sistema impulsivo. Para esto se usa la función de costo de la ecuación 2-27 en donde se introducen nuevas variables de decisión  $x_a, u_a$  que son los equilibrios del sistema [Rivadeneira et al., 2018].

$$\min_{\mathbf{u}, \mathbf{x}, x_a, u_a, x_t, u_t} \sum_{j=0}^{H_p-1} \|x(j) - x_a\|_Q^2 + \sum_{j=0}^{H_c-1} \|u(j) - u_a\|_R^2 + P(\text{dist}_{\mathcal{X}_s^{Tar}}(x_a) + \text{dist}_{\mathcal{U}_s^{Tar}}(u_a))$$

$$\begin{aligned} s.t. \quad & x(0) = \hat{x}(k), \\ & x(j+1) = Ax(j) + B_u u(j) + B_r r(j) + E, \\ & u(j) \in \mathcal{U}, \quad x(j) \in \mathcal{X}, \\ & x(H_p) = x_a, \\ & x_a = Ax_a + B_u u_a + E. \end{aligned} \tag{2-27}$$

Las variables de equilibrio artificial ahora son variables de decisión junto con la entrada de control  $\mathbf{u}$ . La restricción  $x(H_p) = x_a$  asegura que el estado alcance las variables artificiales y la restricción  $x_a = Ax_a + B_u u_a + E$  asegura que el par  $(x_a, u_a)$  sea un equilibrio del sistema.

En la representación matricial del problema de optimización del ZMPC-AV se define el estado aumentado estimado  $\tilde{x}(k)$ , la trayectoria de control  $U = \{u(k, \dots, u(k + H_c - 1))\}$ , las variables de equilibrio  $x_a, u_a$  y el punto objetivo  $x_t, u_t$ , que conforman las variables de decisión  $x_d = [U \ x_a \ u_a \ x_t \ u_t]^T$ . Tomando en consideración lo anterior se expone la siguiente expansión matricial del estado en el horizonte de predicción:

$$X = A_{au}x(k) + B_{uau}U + B_{rau}\Gamma + E_{au}, \tag{2-28}$$

$$\begin{aligned} \begin{bmatrix} x(k+1) \\ \vdots \\ x(k+H_c) \\ \vdots \\ x(k+H_p) \end{bmatrix} &= \begin{bmatrix} A \\ \vdots \\ A^{H_c} \\ \vdots \\ A^{H_p} \end{bmatrix} x(k) + \begin{bmatrix} B_u & \cdots & 0 \\ \vdots & \ddots & \vdots \\ A^{H_c-1}B_u & \cdots & B_u \\ \vdots & \ddots & \vdots \\ A^{H_p-1}B_u & \cdots & \sum_{i=0}^{H_p-H_c} A^i B_u \end{bmatrix} \begin{bmatrix} u(k) \\ \vdots \\ u(k+H_c-1) \end{bmatrix} + \\ &\begin{bmatrix} B_r & \cdots & 0 \\ \vdots & \ddots & \vdots \\ A^{H_c-1}B_r & \cdots & 0 \\ \vdots & \ddots & \vdots \\ A^{H_p-1}B_r & \cdots & B_r \end{bmatrix} \begin{bmatrix} r(k) \\ \vdots \\ r(k+H_p-1) \end{bmatrix} + \begin{bmatrix} E \\ \vdots \\ \sum_{i=0}^{H_p-1} A^i E \end{bmatrix}. \end{aligned} \tag{2-29}$$

Adicionalmente se considera la matriz  $C_c = [I_{n_{xC}} \ O_{n_{xC} \times (n_x + n_d) - n_{xC}}]$  donde  $n_{xC}$  es el número de estados controlables,  $I_{n_{xC}}$  es la matriz identidad de tamaño  $n_{xC} \times n_{xC}$  y  $O_{n_{xC} \times (n_x + n_d) - n_{xC}}$

es una matriz de ceros de dimensiones  $n_{xC} \times ((n_x + n_d) - n_{xC})$  por lo que la expansión en el horizonte de predicción queda como:

$$C_{au} \begin{bmatrix} C_c & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & C_c \end{bmatrix}. \quad (2-30)$$

Luego se reemplaza la expansión matricial en la función de costo como:

$$V = V_1 + V_2 + V_3 + V_4. \quad (2-31)$$

Donde

$$\begin{aligned} V_1 &= (C_{au}X - I_{xa}x_a)^T \tilde{Q}(C_{au}X - I_{xa}x_a), \\ &= (C_{au}(A_{au}x(k) + B_{uau}U + B_{rau}\Gamma + E_{au}) - I_{xa}x_a)^T \tilde{Q}(\cdot), \\ &= (a + bx_d)^T \tilde{Q}(a + bx_d), \end{aligned} \quad (2-32)$$

con  $a = C_{au}(A_{au}x(k) + B_{rau}\Gamma + E_{au})$ ,  $b = [C_{au}B_{uau} \quad -I_{xa} \quad O_{H_p n_{xC} \times (2n_u + n_x)}]$ .

$$\begin{aligned} V_2 &= (U - I_{ua}u_a)^T \tilde{R}(U - I_{ua}u_a), \\ &= (U + 0x_a - I_{ua}u_a + 0x_t + 0u_t)^T \tilde{R}(\cdot), \\ &= (cx_d)^T \tilde{Q}(cx_d), \end{aligned} \quad (2-33)$$

Con  $c = [I_{H_u n_u} \quad O_{H_u n_u \times n_{xC}} \quad -I_{ua} \quad O_{H_u n_u \times (n_{xC} + n_u)}]$ .

$$\begin{aligned} V_3 &= (x_a - x_t)^T \tilde{P}(x_a - x_t), \\ &= (0U + I_{n_{xC}}x_a + 0u_a - I_{n_{xC}}x_t + 0u_t)^T \tilde{P}(\cdot), \\ &= (ex_d)^T \tilde{Q}(ex_d), \end{aligned} \quad (2-34)$$

Con  $e = [O_{c_{xC} \times H_u n_u} \quad I_{n_{xC}} \quad O_{c_{xC} \times n_u} \quad -I_{n_{xC}} \quad O_{c_{xC} \times n_u}]$ .

$$\begin{aligned} V_4 &= (u_a - x_t)^T \tilde{P}(u_a - u_t), \\ &= (0U + 0x_a + I_{n_u}u_a + 0x_t - I_{n_u}ut)^T \tilde{P}(\cdot), \\ &= (dx_d)^T \tilde{Q}(dx_d). \end{aligned} \quad (2-35)$$

Con  $d = [O_{n_{xC} \times H_u n_u} \quad O_{n_u \times n_{xC}} \quad I_{n_u} \quad O_{n_u \times n_{xC}} \quad -I_{n_u}]$ . De los términos anteriores la matriz  $I_{xa}$  se establece como la expansión en el horizonte de predicción de la matriz identidad de dimensión

$n_{xC}$ , y la matriz  $I_{ua}$  se establece como la expansión en el horizonte de control de la matriz identidad de dimensión  $n_u$ :

$$I_{xa} = \begin{bmatrix} I_{nxC} \\ \vdots \\ I_{nxC} \end{bmatrix}, I_{ua} = \begin{bmatrix} I_{nu} \\ \vdots \\ I_{nu} \end{bmatrix}. \quad (2-36)$$

Definiendo las matrices de sintonía como:

$$\tilde{Q} = \begin{bmatrix} C_c Q & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & C_c Q \end{bmatrix}, \tilde{R} = \begin{bmatrix} R & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & R \end{bmatrix}, \tilde{P} = P I_{nxC}, \tilde{\tilde{P}} = P I_{nu}. \quad (2-37)$$

Remplazando las ecuaciones de la 2-32 a la 2-35 en 2-31 se obtiene la siguiente función de costo:

$$V(k) = \frac{1}{2} x^T H x + f^T x + cte,$$

$$H = 2(b^T \tilde{Q} b + c^T \tilde{R} c + e^T \tilde{P} e + d^T \tilde{\tilde{P}} d), f = 2(a^T \tilde{Q} b)^T, cte = a^T \tilde{Q} a. \quad (2-38)$$

Para las restricciones de desigualdad los límites para las variables de decisión a lo largo del horizonte de control son:

$$LB_u = \begin{bmatrix} u_{min} \\ \vdots \\ u_{min} \\ C_c x_{min} \\ u_{min} \\ x_{min}^{Tar} \\ u_{min}^{Tar} \end{bmatrix}, UB_u = \begin{bmatrix} u_{max} \\ \vdots \\ u_{max} \\ C_c x_{max} \\ u_{max} \\ x_{max}^{Tar} \\ u_{max}^{Tar} \end{bmatrix}. \quad (2-39)$$

Para las restricciones de igualdad se tiene la terminal  $C_c x(k + H_p) = x_a$ , en donde el estado en  $H_p$  equivale a la última fila de la expansión matricial en 2-28, obteniendo:

$$C_c x(k + H_p) = C_c A^{H_p} x(k) + C_c [A^{H_p-1} B_u \cdots \sum_{i=0}^{H_p-H_u} A^i B_u] U, \quad (2-40)$$

$$+ C_c [A^{H_p-1} B_r \cdots B_r] \Gamma + C_c \sum_{i=0}^{H_p-1} A^i E = x_a.$$

Representando la restricción de la forma  $A_{eq}x_d = b_{eq1}$  se definen como:

$$A_{eq1} = [C_c[A^{H_p-1}B_u \cdots \sum_{i=0}^{H_p-H_u} A^i B_u] - I_{n_{xC}} O_{n_{xC} \times (2n_u+n_{xC})}],$$

$$b_{eq1} = -C_c A^{H_p} x(k) - C_c [A^{H_p-1} B_r \cdots B_r] \Gamma - C_c \sum_{i=0}^{H_p-1} A^i E. \quad (2-41)$$

Las variables de decisión que deben satisfacer la condición de equilibrio son las variables artificiales  $x_a, u_a$  ( $x_a = Ax_a + B_u u_a + E = 0$ ), y las variables objetivo  $x_t, u_t$  ( $x_t = Ax_t + B_u u_t + E = 0$ ). Para ambos casos debe tenerse solo en cuenta la sección controlable de las matrices  $A, B_u$  y  $E$ . Con esto la representación matricial queda como:

$$A_{eq2} = [O_{n_{xC} \times (H_u n_u)} (A_{n_{xC}} - I_{n_{xC}}) B_{un_{xC}} O_{n_{xC} \times (n_{xC} + n_u)}],$$

$$b_{eq2} = -E_{n_{xC}}. \quad (2-42)$$

$$A_{eq3} = [O_{n_{xC} \times (H_u n_u + n_{xC} + n_u)} (A_{n_{xC}} - I_{n_{xC}}) B_{un_{xC}}],$$

$$b_{eq3} = -E_{n_{xC}}. \quad (2-43)$$

Luego se concatenen las matrices para tenerlas de la forma:

$$A_{eq} = \begin{bmatrix} A_{eq1} \\ A_{eq2} \\ A_{eq3} \end{bmatrix}, \quad b_{eq} = \begin{bmatrix} b_{eq1} \\ b_{eq2} \\ b_{eq3} \end{bmatrix}. \quad (2-44)$$

Para que las restricciones de desigualdad del estado de la forma  $x_{min} \leq x(j) \leq x_{max}$  se satisfagan durante todo el horizonte de predicción y teniendo solo en cuenta los estados controlables se presenta la siguiente representación matricial:

$$C_{au}(A_{au}x(k) + B_{uau}U + B_{rau}\Gamma + E_{au}) \leq X_{max},$$

$$X_{min} \leq C_{au}(A_{au}x(k) + B_{uau}U + B_{rau}\Gamma + E_{au}). \quad (2-45)$$

En donde

$$X_{max} = \begin{bmatrix} C_c x_{max} \\ \vdots \\ C_c x_{max} \end{bmatrix}, \quad X_{min} = \begin{bmatrix} C_c x_{min} \\ \vdots \\ C_c x_{min} \end{bmatrix}. \quad (2-46)$$

Y estas se llevan a la forma  $A_{des}x_d \leq b_{des}$  dando como resultado:

$$\begin{aligned}
A_{des} &= \begin{bmatrix} C_{au}B_{uau} & O_{H_p n_x C \times (2n_x C + 2n_u)} \\ -C_{au}B_{uau} & O_{H_p n_x C \times (2n_x C + 2n_u)} \end{bmatrix}, \\
b_{des} &= \begin{bmatrix} -C_{au}A_{au}x(k) - C_{au}B_{rau}\Gamma - C_{au}E_{au} + X_{max} \\ C_{au}A_{au}x(k) + C_{au}B_{rau}\Gamma + C_{au}E_{au} - X_{max} \end{bmatrix}.
\end{aligned} \tag{2-47}$$

### 2.3.4. Control predictivo basado en modelo por zonas con variables artificiales y Offset-free

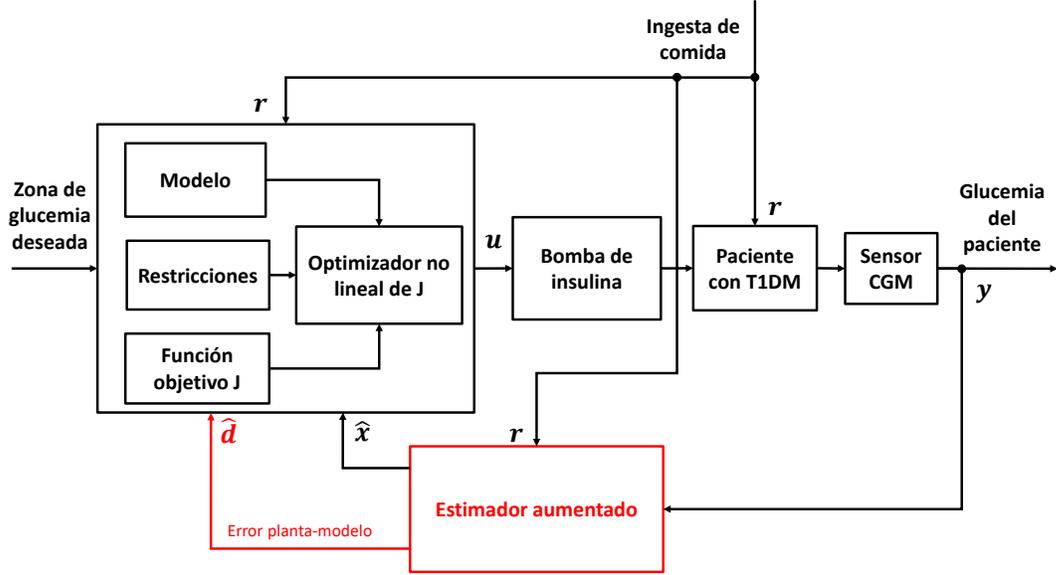
Por último, se tiene el ZMPC-AV con Offset-free (ZMPC-AV-OF) que es formulado para compensar el efecto de la diferencia (*mismatch* en inglés) entre el modelo y la planta real en la regulación de glucemia. La idea de la estrategia consiste en aumentar el estado con una perturbación (que representa el desajuste), estimarlo mediante el observador de estado y proporcionar esta información al problema MPC en el modelo de predicción y restricciones [F. et al., 2020, Villa-Tamayo and Rivadeneira, 2020b]. Por tanto, el modelo discreto se reescribe y se llega a la ecuación 2-48.

$$\begin{aligned}
x(k+1) &= Ax(k) + B_u u(k) + B_r r(k) + B_d d(k) + E, \\
y(k) &= Cx(k) + C_d d(k).
\end{aligned} \tag{2-48}$$

En donde la perturbación satisface  $d(k+1) = d(k)$  y  $B_d = e^{A_c T} B_{dc}$ . y se considera la función de costo de la ecuación 2-49.

$$\begin{aligned}
&\min_{\mathbf{u}, \mathbf{x}, x_a, u_a, x_t, u_t, \mathbf{d}} \sum_{j=0}^{H_p-1} \|x(j) - x_a\|_Q^2 + \sum_{j=0}^{H_c-1} \|u(j) - u_a\|_R^2 + P(\text{dist}_{\mathcal{X}_s^{Tar}}(x_a) + \text{dist}_{\mathcal{U}_s^{Tar}}(u_a)) \\
&s.t. \quad x(0) = \hat{x}(k), \quad d(0) = \hat{d}(k), \\
&\quad x(j+1) = Ax(j) + B_u u(j) + B_r r(j) + B_d d(j) + E, \\
&\quad d(j+1) = d(j), \\
&\quad u(j) \in \mathcal{U}, \quad x(j) \in \mathcal{X}, \\
&\quad x(H_p) = x_a, \\
&\quad x_a = Ax_a + B_u u_a + B_d d + E, \quad y_a = Cx_a + C_d d(j).
\end{aligned} \tag{2-49}$$

Al esquema de control de la Figura 2-4, se le añade el estado aumentado del error planta modelo en el estimador que realimenta los estados al MPC, como se observa en la Figura 2-5. Las garantías de eliminación de offset para sistemas impulsivos de este controlador y sus demostraciones se pueden observar en la tesis de maestría [Villa-tamayo, 2021], que es de donde se toman algunas de las formulaciones matemáticas de modelo-controlador para ser implementadas en este trabajo.



**Figura 2-5:** Esquema de control para el PA usando el controlador ZMPC-AV-OF

La ecuación 2-48 en donde  $d$  es la perturbación y tiene matrices asociadas  $B_d \in R^{n_x \times n_d}$  y  $C_d \in R^{n_y \times n_d}$ , con las que se puede construir el subsistema subyacente en tiempo discreto primario y de salida como:

$$\tilde{x}(k+1) = \tilde{A}\tilde{x}(k) + \tilde{B}_u u(k) + \tilde{B}_r r(k) + \tilde{E}, \quad \tilde{y}(k) = \tilde{C}\tilde{x}(k).$$

Donde las matrices del sistema aumentadas están dadas por:

$$\tilde{x} = \begin{bmatrix} x \\ d \end{bmatrix}, \quad \tilde{A} = \begin{bmatrix} A & B_d \\ 0 & I_d \end{bmatrix}, \quad \tilde{B}_u = \begin{bmatrix} B_u \\ 0 \end{bmatrix}, \quad \tilde{B}_r = \begin{bmatrix} B_r \\ 0 \end{bmatrix},$$

$$\tilde{E} = \begin{bmatrix} E \\ 0 \end{bmatrix}, \quad \tilde{C} = [C \quad C_d], \quad y \quad B_d = e^{AT} B_d. \quad (2-50)$$

En la representación matricial del problema de optimización del ZMPC-AV-OF se define el estado aumentado estimado  $\tilde{x}(k)$ , la trayectoria de control  $U = \{u(k), \dots, u(k+H_c-1)\}$ , las variables de equilibrio  $x_a$ ,  $u_a$  y el punto objetivo  $x_t$ ,  $u_t$ , que conforman las variables de decisión  $x_d = [U \quad x_a \quad u_a \quad x_t \quad u_t]^T$ . Considerando la siguiente expansión del estado en el horizonte de predicción:

$$\tilde{X} = A_{au}x(k) + B_{uau}U + B_{rau}\Gamma + E_{au}, \quad (2-51)$$

$$\begin{aligned}
\begin{bmatrix} \tilde{x}(k+1) \\ \vdots \\ \tilde{x}(k+H_c) \\ \vdots \\ \tilde{x}(k+H_p) \end{bmatrix} &= \begin{bmatrix} \tilde{A} \\ \vdots \\ \tilde{A}^{H_c} \\ \vdots \\ \tilde{A}^{H_p} \end{bmatrix} x(k) + \begin{bmatrix} \tilde{B}_u & \cdots & 0 \\ \vdots & \ddots & \vdots \\ \tilde{A}^{H_c-1}\tilde{B}_u & \cdots & \tilde{B}_u \\ \vdots & \ddots & \vdots \\ \tilde{A}^{H_p-1}\tilde{B}_u & \cdots & \sum_{i=0}^{H_p-H_c} \tilde{A}^i \tilde{B}_u \end{bmatrix} \begin{bmatrix} u(k) \\ \vdots \\ u(k+H_c-1) \end{bmatrix} + \\
&\begin{bmatrix} \tilde{B}_r & \cdots & 0 \\ \vdots & \ddots & \vdots \\ \tilde{A}^{H_c-1}\tilde{B}_r & \cdots & 0 \\ \vdots & \ddots & \vdots \\ \tilde{A}^{H_p-1}\tilde{B}_r & \cdots & \tilde{B}_r \end{bmatrix} \begin{bmatrix} r(k) \\ \vdots \\ r(k+H_p-1) \end{bmatrix} + \begin{bmatrix} \tilde{E} \\ \vdots \\ \sum_{i=0}^{H_p-1} \tilde{A}^i \tilde{E} \end{bmatrix}.
\end{aligned} \tag{2-52}$$

Adicionalmente se considera la matriz  $C_c = [I_{n_{xC}} \ O_{n_{xC} \times (n_x + n_d) - n_{xC}}]$  donde  $n_{xC}$  es el número de estados controlables,  $I_{n_{xC}}$  es la matriz identidad de tamaño  $n_{xC} \times n_{xC}$  y  $O_{n_{xC} \times (n_x + n_d) - n_{xC}}$  es una matriz de ceros de dimensiones  $n_{xC} \times ((n_x + n_d) - n_{xC})$  por lo que la expansión en el horizonte de predicción queda como:

$$C_{au} \begin{bmatrix} C_c & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & C_c \end{bmatrix}. \tag{2-53}$$

Luego se reemplaza la expansión matricial en la función de costo como:

$$V = V_1 + V_2 + V_3 + V_4. \tag{2-54}$$

Donde:

$$\begin{aligned}
V_1 &= (C_{au}\tilde{X} - I_{xa}x_a)^T \tilde{Q} (C_{au}\tilde{X} - I_{xa}x_a), \\
&= (C_{au}(A_{au}\tilde{x}(k) + B_{uau}U + B_{rau}\Gamma + E_{au}) - I_{xa}x_a)^T \tilde{Q}(\cdot), \\
&= (a + bx_d)^T \tilde{Q}(a + bx_d),
\end{aligned} \tag{2-55}$$

Con  $a = C_{au}(A_{au}\tilde{x}(k) + B_{rau}\Gamma + E_{au})$ ,  $b = [C_{au}B_{uau} \ - I_{xa} \ O_{H_p n_{xC} \times (2n_u + n_{xC})}]$ .

$$\begin{aligned}
V_2 &= (U - I_{ua}u_a)^T \tilde{R}(U - I_{ua}u_a), \\
&= (U + 0x_a - I_{ua}u_a + 0x_t + 0u_t)^T \tilde{R}(\cdot), \\
&= (cx_d)^T \tilde{Q}(cx_d),
\end{aligned} \tag{2-56}$$

Con  $c = [I_{H_u n_u} \ O_{H_u n_u \times n_{xC}} \ - I_{ua} \ O_{H_u n_u \times (n_{xC} + u_u)}]$ .

$$\begin{aligned}
V_3 &= (x_a - x_t)^T \tilde{P}(x_a - x_t), \\
&= (0U + I_{n_x C} x_a + 0u_a - I_{n_x C} x_t + 0u_t)^T \tilde{P}(\cdot), \\
&= (ex_d)^T \tilde{Q}(ex_d),
\end{aligned} \tag{2-57}$$

Con  $e = [O_{n_x C \times H_u n_u} \quad I_{n_x C} \quad O_{c_x C \times n_u} \quad -I_{n_x C} \quad O_{c_x C \times n_u}]$ .

$$\begin{aligned}
V_4 &= (u_a - x_t)^T \tilde{P}(u_a - u_t), \\
&= (0U + 0x_a + I_{n_u} u_a + 0x_t - I_{n_u} u_t)^T \tilde{P}(\cdot), \\
&= (dx_d)^T \tilde{Q}(dx_d),
\end{aligned} \tag{2-58}$$

Con  $d = [O_{c_x C \times H_u n_u} \quad O_{n_u \times n_x C} \quad I_{n_u} \quad O_{n_u \times n_x C} \quad -I_{n_u}]$ . De los términos anteriores las matriz  $I_{xa}$  se establece como la expansión en el horizonte de predicción de la matriz identidad de dimensión  $n_x C$  y la matriz  $I_{ua}$  se establece como la expansión en el horizonte de control de la matriz identidad de dimensión  $n_u$ .

$$I_{xa} = \begin{bmatrix} I_{n_x C} \\ \vdots \\ I_{n_x C} \end{bmatrix}, I_{ua} = \begin{bmatrix} I_{n_u} \\ \vdots \\ I_{n_u} \end{bmatrix}. \tag{2-59}$$

Definiendo las matrices de sintonía como:

$$\tilde{Q} = \begin{bmatrix} C_c Q & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & C_c Q \end{bmatrix}, \tilde{R} = \begin{bmatrix} R & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & R \end{bmatrix}, \tilde{P} = P I_{n_x C}, \tilde{\tilde{P}} = P I_{n_u}. \tag{2-60}$$

Remplazando las ecuaciones de la 2-55 a la 2-58 en 2-54 se obtiene la siguiente función de costo:

$$V(k) = \frac{1}{2} x^T H x + f^T x + cte,$$

$$H = 2(b^T \tilde{Q} b + c^T \tilde{R} c + e^T \tilde{P} e + d^T \tilde{\tilde{P}} d), \quad f = 2(a^T \tilde{Q} b)^T, \quad cte = a^T \tilde{Q} a. \tag{2-61}$$

Para las restricciones de desigualdad los límites para las variables de decisión a lo largo del horizonte de control son:

$$LB_u = \begin{bmatrix} u_{min} \\ \vdots \\ u_{min} \\ C_c x_{min} \\ u_{min} \\ x_{min}^{Tar} \\ u_{min}^{Tar} \end{bmatrix}, \quad UB_u = \begin{bmatrix} u_{max} \\ \vdots \\ u_{max} \\ C_c x_{max} \\ u_{max} \\ x_{max}^{Tar} \\ u_{max}^{Tar} \end{bmatrix}. \quad (2-62)$$

Para las restricciones de igualdad se tiene la terminal  $C_c \tilde{x}(k + H_p) = x_a$ , en donde el estado en  $H_p$  equivale a la última fila de la expansión matricial en 2-51, obteniendo:

$$C_c \tilde{x}(k + H_p) = C_c \tilde{A}^{H_p} x(k) + C_c [\tilde{A}^{H_p-1} \tilde{B}_u \cdots \sum_{i=0}^{H_p-H_u} \tilde{A}^i \tilde{B}_u] U, \quad (2-63)$$

$$+ C_c [\tilde{A}^{H_p-1} \tilde{B}_r \cdots \tilde{B}_r] \Gamma + C_c \sum_{i=0}^{H_p-1} \tilde{A}^i \tilde{E} = x_a.$$

Luego se pasa a representar la restricción de la forma  $A_{eq} x_d = b_{eq1}$ , dando como resultado:

$$A_{eq1} = [C_c [\tilde{A}^{H_p-1} \tilde{B}_u \cdots \sum_{i=0}^{H_p-H_u} \tilde{A}^i \tilde{B}_u] - I_{n_{xC}} \quad O_{n_{xC} \times (2n_u + n_{xC})}], \quad (2-64)$$

$$b_{eq1} = -C_c \tilde{A}^{H_p} x(k) - C_c [\tilde{A}^{H_p-1} \tilde{B}_r \cdots \tilde{B}_r] \Gamma - C_c \sum_{i=0}^{H_p-1} \tilde{A}^i \tilde{E}.$$

Las variables de decisión que deben satisfacer la condición de equilibrio son las variables artificiales  $x_a, u_a$  ( $x_a = Ax_a + B_u u_a + E = 0$ ), y las variables objetivo  $x_t, u_t$  ( $x_t = Ax_t + B_u u_t + E = 0$ ). Para ambos casos debe tenerse solo en cuenta la sección controlable de las matrices  $A, B_u, E$  y  $B_d$  representado por el subíndice  $n_{xC}$ . Con esto la representación matricial queda como:

$$A_{eq2} = [O_{n_{xC} \times (H_u n_u)} \quad (A_{n_{xC}} - I_{n_{xC}}) \quad B_{un_{xC}} \quad O_{n_{xC} \times (n_{xC} + n_u)}], \quad (2-65)$$

$$b_{eq2} = -E_{n_{xC}} - B_{d_{n_{xC}}} d(k).$$

$$A_{eq3} = [O_{n_{xC} \times (H_u n_u + n_{xC} + n_u)} \quad (A_{n_{xC}} - I_{n_{xC}}) \quad B_{un_{xC}}], \quad (2-66)$$

$$b_{eq3} = -E_{n_{xC}} - B_{d_{n_{xC}}} d(k).$$

Luego se concatenen las matrices para tenerlas de la forma:

$$A_{eq} = \begin{bmatrix} A_{eq1} \\ A_{eq2} \\ A_{eq3} \end{bmatrix}, \quad b_{eq} = \begin{bmatrix} b_{eq1} \\ b_{eq2} \\ b_{eq3} \end{bmatrix}. \quad (2-67)$$

Para que las restricciones de desigualdad del estado de la forma  $x_{min} \leq x(j) \leq x_{max}$  se satisfagan durante todo el horizonte de predicción y teniendo solo en cuenta los estados controlables se presenta la representación matricial de la ecuación 2-68.

$$\begin{aligned} C_{au}(A_{au}\tilde{x}(k) + B_{uau}U + B_{rau}\Gamma + E_{au}) &\leq X_{max}, \\ X_{min} &\leq C_{au}(A_{au}\tilde{x}(k) + B_{uau}U + B_{rau}\Gamma + E_{au}). \end{aligned} \quad (2-68)$$

En donde

$$X_{max} = \begin{bmatrix} C_c x_{max} \\ \vdots \\ C_c x_{max} \end{bmatrix}, \quad X_{min} = \begin{bmatrix} C_c x_{min} \\ \vdots \\ C_c x_{min} \end{bmatrix}. \quad (2-69)$$

Y estas se llevan a la forma  $A_{des}x_d \leq b_{des}$  dando como resultado:

$$\begin{aligned} A_{des} &= \begin{bmatrix} C_{au}B_{uau} & O_{H_p n_x C \times (2n_x C + 2n_u)} \\ -C_{au}B_{uau} & O_{H_p n_x C \times (2n_x C + 2n_u)} \end{bmatrix}, \\ b_{des} &= \begin{bmatrix} -C_{au}A_{au}\tilde{x}(k) - C_{au}B_{rau}\Gamma - C_{au}E_{au} + X_{max} \\ C_{au}A_{au}\tilde{x}(k) + C_{au}B_{rau}\Gamma + C_{au}E_{au} - X_{max} \end{bmatrix}. \end{aligned} \quad (2-70)$$

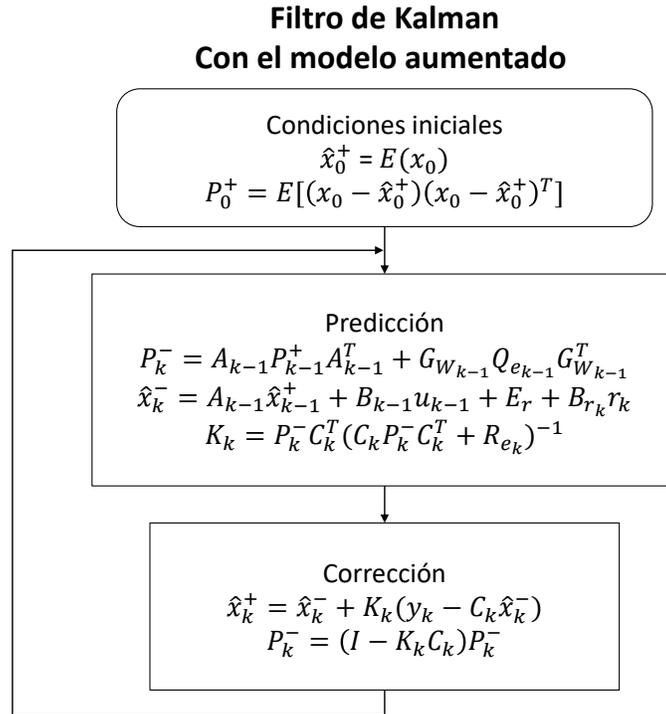
## 2.4. Estimador

En la estrategia de control del MPC se requiere una realimentación de todos los estados para calcular la dosificación óptima de insulina, sin embargo, el CGM es el único sensor que se tiene para conocer el estado del paciente, todos los demás estados que representan el comportamiento de las hormonas y sus tasas de crecimiento o decrecimiento son difíciles de medir o simplemente no se tiene la forma de hacerlo en tiempo real, por lo que se hace indispensable el uso de un estimador, tal y como se observa en las Figuras 2-4 y 2-5. Con esto a partir de la información que se obtiene del sensor de glucosa se estiman los otros estados complementando la información necesaria para la realimentación total del modelo hacia el controlador. Además de esto siempre que se realiza una medición los datos obtenidos del sensor contienen ruido que distorsiona el valor real de la medición, por este motivo se decide usar un filtro de Kalman como estimador, que ayude a la reconstrucción de la información

del modelo actual y filtre el ruido de medición.

El filtro de Kalman ha sido ampliamente usado en muchos de los problemas que requieren de un control con observador, las ventajas de su estructura de filtro son claves para el manejo de los datos obtenidos que contienen ruido inherente a cualquier proceso de medición. Este estimador tiene la estructura del algoritmo descrito en la Figura 2-6, en donde se le dan unas condiciones iniciales “aleatorias.” al modelo y luego de eso se entra a un bucle donde los datos apriori se usan para la etapa de predicción usando al modelo y luego con los datos de la medición se calculan los valores aposteriori en la etapa de corrección, este ciclo se da previo a cada cálculo de la dosificación de insulina [Simon, 2006].

En la Figura 2-6, se describe el algoritmo de predicción y corrección del filtro de Kalman que se calcula en cada instante de tiempo  $k$ , tomando a  $P_k$  como la matriz de covarianza del error, las matrices de sintonía del estimador son las matrices de covarianza  $Q_k$ ,  $R_k$  y  $G_W$ , esta ultima matriz para el caso de los controladores MPC, ZMPC y ZMPC-AV es la matriz identidad. Esto con el fin de calcular la ganancia  $K_k$  óptima en cada iteración y obtener el estado estimado  $\hat{x}_k^+$ .



**Figura 2-6:** Algoritmo del filtro de Kalman

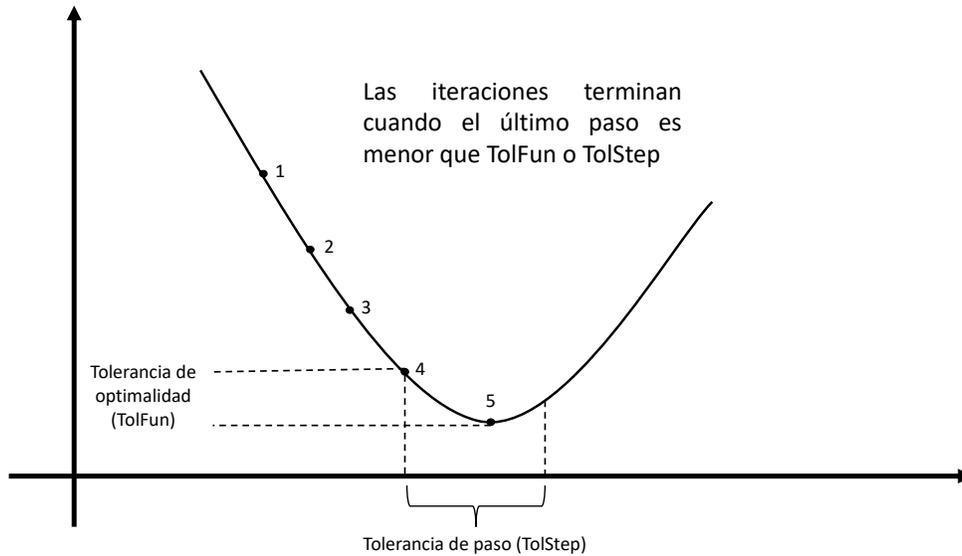
## 2.5. Paquetes de optimización

Debido a que todos los controladores en este trabajo son de tipo MPC y estos se basan en la minimización de una función de costo, para todos ellos es necesario implementar un método que dé una solución óptima. La referencia con la que se comprueba el funcionamiento de los sistemas embebidos es un computador en el que se ejecuta el software MATLAB R2020b y en este usa el paquete quadprog que tiene por defecto el método de punto interior convexo. Todos los paquetes de optimización implementados para los sistemas embebidos son de código abierto, lo que permite que su uso no incremente el costo del dispositivo final. De los paquetes actualmente disponibles se tomaron tres que implementan diferentes métodos de solución del problema de optimización (CVXOPT, quadprog, OSQP) para Python.

El paquete quadprog de MATLAB soluciona el problema de la forma 2-71 en donde  $H$  es una matriz simétrica,  $f$  es un vector en términos lineales,  $A$  es la matriz de desigualdades lineales,  $b$  es el vector de desigualdad lineal,  $Aeq$  es la matriz de igualdades lineales,  $beq$  es el vector de igualdad lineal,  $lb$  es el vector de límites inferiores y  $ub$  es el vector de límites superiores. Una descripción sencilla del algoritmo de punto interior convexo es: tratar de remover redundancias, simplificar restricciones, generar un punto inicial, entrar a la fase predictor/corrector, evaluar criterios de parada y determinar infactibilidades. Este método presenta soluciones de problemas grandes de forma rápida y ocupa poco espacio de memoria, a pesar manejar una precisión un poco menor a otros algoritmos debido a que la función barrera mantiene las iteraciones alejadas de los límites de la restricción de desigualdad.

$$\begin{aligned}
 \text{Min} \quad & \frac{1}{2}x^T Hx + f^T x \\
 \text{Sujeto a} \quad & Ax \leq b \\
 & Aeqx = beq, \\
 & lb \leq x \leq ub.
 \end{aligned} \tag{2-71}$$

Los criterios de parada de este algoritmo dependen del número máximo de iteraciones  $MaxIterations$  que por defecto son 200 y las tolerancias asociadas al problema: tolerancia de terminación en la optimalidad de primer orden para la terminación en el valor de la función  $TolFun$ . Tolerancia de terminación en un paso, es un límite inferior del tamaño de un paso  $TolStep$  Si el solucionador intenta dar un paso menor que  $TolFun$  o  $TolStep$ , las iteraciones finalizan ver Figura 2-7. Tolerancia para la infracción de restricción  $TolCon$  y la tolerancia para la terminación de las iteraciones en el método del gradiente conjugado pre-seleccionado  $TolPCG$ , que es la parte del algoritmo que resuelve un sistema de ecuaciones lineales con matrices simétricas y definidas positivas [MathWorks, 2021b].



**Figura 2-7:** Criterio de terminación en MATLAB, basado en [MathWorks, 2021b]

### 2.5.1. CVXOPT

Este es un paquete gratuito para la optimización convexa basado en el lenguaje de programación Python, fue desarrollado como propósito personal de los diseñadores, pero fue puesto a disposición para cualquiera que lo considere útil, a pesar de que sigue en constante desarrollo y reciben reporte de errores y mejoras, ofrecen un soporte muy limitado. Este paquete para optimización convexa implementa un algoritmo de punto interior predictor-corrector estándar de Mehrotra. Permite al usuario especificar un problema de optimización a través de una descripción del operador, es decir, proporcionando funciones para evaluar las asignaciones lineales en las restricciones y proporcionar un método personalizado para resolver las ecuaciones de Newton.

Este algoritmo es programado para resolver el problema cuadrático de la ecuación 2-72 en donde  $x$  es la variable de decisión primaria,  $s$  es la variable de holgura,  $y$ ,  $z$  son las variables duales y con  $P$  matriz semidefinida positiva. El programa para la solución de problemas cuadráticos (coneqp de CVXOPT), puede manejar un término cuadrático en la función objetivo, pero necesita que el problema sea estrictamente factible primario y dual y no se detecten infactibilidades.

$$\begin{aligned}
 & \text{Minimizar} && (1/2)x^T P x + c^T x \\
 & \text{Sujeto a} && Gx + s = h \\
 & && Ax = b, \\
 & && s \succeq 0.
 \end{aligned} \tag{2-72}$$

Se asumen que  $\text{rank}(A) = p$  y que el  $\text{rank}([PA^T G^T]) = n$  en donde  $p$  es la dimensión de la

fila de  $A$  y  $n$  es la dimensión de  $x$ , lo cual es equivalente a asumir que la matriz  $M$  es no singular para cualquier  $Q$  semidefinida positiva. Si  $\text{rank}(A) < p$  entonces las restricciones de igualdad en el problema primario son inconsistentes y si  $\text{rank}([PA^T G^T]) < n$  entonces las restricciones de desigualdad en el problema dual son inconsistentes.

$$M = \begin{bmatrix} P & A^T & G^T \\ A & 0 & 0 \\ G & 0 & -Q \end{bmatrix}.$$

Los métodos que usan el punto interior son bastante implementados porque en general alcanzan una alta precisión en un número pequeño de iteraciones, en cada iteración el algoritmo soluciona un conjunto de ecuaciones lineales y la clave de este método son estas ecuaciones que muchas veces se denominan ecuaciones de Newton porque pueden ser interpretadas como una linealización de las ecuaciones no lineales que caracterizan la ruta principal o también llamadas ecuaciones Karush-Kuhn-Tucker (KKT), porque pueden ser consideradas como condiciones de optimalidad de un problema cuadrático con restricciones de igualdad, a pesar de esto una de las grandes desventajas conocidas de este paquete son los altos requisitos de memoria y la complejidad del álgebra lineal asociada a las ecuaciones lineales que se resuelven en cada iteración, el paquete CVXOPT implementa un algoritmo para el método de punto interior primal-dual basado en la escala Nesterov-Todd [Andersen et al., 2003]. Las ecuaciones de Newton tienen la forma de 2-73.

$$\begin{bmatrix} P & A^T & G^T \\ A & 0 & 0 \\ G & 0 & -W^T W \end{bmatrix} \begin{bmatrix} \Delta_x \\ \Delta_y \\ \Delta_z \end{bmatrix} = \begin{bmatrix} r_x \\ r_y \\ r_z \end{bmatrix}, \quad (2-73)$$

Pero esta ecuación se puede reducir de la forma:

$$\begin{bmatrix} P + G^T W^{-1} W^{-T} G & A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} b_x + P + G^T W^{-1} W^{-T} G b_z \\ b_y \end{bmatrix}. \quad (2-74)$$

Y si  $P + G^T W^{-1} W^{-T} G$  es no singular la ecuación 2-74 se puede resolver con la factorización Cholesky ( $P + G^T W^{-1} W^{-T} G = LL^T$ ), en donde se computa  $P + G^T W^{-1} W^{-T} G + A^T A = LL^T$  y se soluciona  $AL^T L^{-1} A^T y = AL^{-1} L^{-1} (b_x + P + G^T W^{-1} W^{-T} G b_z + A b_y) - b_y$  usando la factorización Cholesky  $AL^T L^{-1} A^T$  para obtener  $y$  y luego  $LL^T x = b_x + P + G^T W^{-1} W^{-T} G b_z + A^T (b_y - y)$  para obtener  $x$  [Sra et al., 2011].

Este paquete cuenta con parámetros modificables relacionados con los criterios de parada del algoritmo. Tolerancia absoluta  $\epsilon_{abs}$ , tolerancia relativa  $\epsilon_{rel}$ , la tolerancia a las condiciones de viabilidad  $\epsilon_{feas}$  y el número máximo de iteraciones  $maxiters$  que por defecto es 100. El

algoritmo de CVXOPT termina con un estado de “óptimo” si satisface las condiciones del siguiente algoritmo.

---

**si** ( $s \succeq 0$ ,  $\frac{\|Gx+s-h\|_2}{\max\{1,\|h\|_2\}} \leq \epsilon_{feas}$ ,  $\frac{\|Ax-b\|_2}{\max\{1,\|b\|_2\}} \leq \epsilon_{feas}$  &  $z \succeq 0$ ,  $\frac{\|Px+G^Tz+A^Ty+q\|_2}{\max\{1,\|q\|_2\}} \leq \epsilon_{feas}$ )  
**entonces**  
**si**  $s^Tz \leq \epsilon_{abs}$  ||  $(\frac{1}{2}x^TPx + q^T < 0$ ,  $y \frac{s^Tz}{-(1/2)x^TPx+q^T} \leq \epsilon_{rel}$ ) ||  $(L(x, y, z) > 0$ ,  $y \frac{s^Tz}{L(x,y,z)} \leq \epsilon_{rel}$ ) **entonces**  
Punto óptimo.  
**fin si**  
**si no**  
Siguiente iteración.  
**fin si**

---

Con  $L(x, y, z) = \frac{1}{2}x^TPx + q^T + z^T(Gx - h) + y^T(Ax - b)$ .

## 2.5.2. quadprog

El paquete quadprog (*Quadratic Programming Solver*), para Python es un repositorio que se encuentra de forma gratuita bajo licencia (GNU *General Public License* v2.0) en la plataforma GitHub, sin embargo, no se tiene mucha información al respecto, solo se conoce al contribuidor principal a Robert T. McGibbon y otras 6 personas que se basan en la publicación de [Goldfarb and Idnani, 2011], para la programación del algoritmo dual basado en el lenguaje de programación Fortran y convertido a C para ser ejecutado en Python.

Este es un algoritmo dual eficiente y numéricamente estable para programación cuadrática definida positiva de la ecuación 2-75, en donde  $x$  y  $a$  son  $n$ -vectores,  $G$  es una matriz simétrica definida positiva a ser minimizada de dimensión  $n \times n$ ,  $C$  es la matriz que define las restricciones bajo las cuales se minimiza la función de dimensión  $n \times m$ ,  $b$  es el  $m$ -vector que define las restricciones. Adicional el vector de variables  $x$  también puede estar sujeto a restricciones lineales de la forma  $\hat{C}^T x = \hat{b}$ .

$$\begin{aligned} & \text{Minimizar} && 1/2x^T Gx - a^T x \\ & \text{Sujeto a} && C^T x \geq b \end{aligned} \tag{2-75}$$

El algoritmo principal del método se describe como un método de programación cuadrática secuencial, (Active set) en donde  $H$  es definida como el operador hessiano inverso reducido,  $A \subseteq K$  para los índices del conjunto activo,  $N$  es la matriz de vectores normales de las restricciones en el conjunto activo indexado por  $A$ ,  $N^*$  es la pseudo inversa de  $N$  en el espacio de variables obtenidas en la transformación  $y = G^{\frac{1}{2}}x$ ,  $q$  es la cardinalidad de  $A$ ,  $K$  denota el conjunto  $\{1, 2, \dots, m\}$  de los índices de las restricciones, el subproblema  $P(J)$  se convierte en la función objetivo de 2-75 en donde  $J \subset K$  y  $S - pair$  es el par solución  $(x, A)$  linealmente

independiente de  $x$  en el subproblema  $P(J)$  con  $A \subseteq K$ . En este algoritmo se prueba en el paso 1 si en el punto actual  $x$  se soluciona el subproblema  $P(A)$  y si cumple con todas las restricciones  $x$  es el punto óptimo, de lo contrario se darán pasos completos o intermedios para llegar a un nuevo  $S - pair$   $(\bar{x}, \bar{A})$  y volver al paso 1 a no ser de que se detecten infactibilidades y como se define que  $(f(\bar{x}) > f(x))$  nunca se repetirá un  $S - pair$  y se tienen un número finito de  $S - pair$  el algoritmo converge a una solución o indica que no tiene una solución factible en un número finito de pasos aunque no se define cual es este número.

Adicional para implementación del cálculo numérico en este método también se implementa la factorización Cholesky  $G = LL^T$  para la matriz simétrica hessiana y la factorización QR 2-76 de la matriz  $nxq$   $B = L^{-1}N$ , en donde  $L$  es la matriz triangular inferior de  $nxn$ ,  $R$  es la matriz triangular superior de  $qxq$  y  $Q = [Q_1|Q_2]$  una matriz ortogonal particionada de forma que  $Q_1$  tiene  $q$  columnas. Con esto se calcula la matriz Hessiana  $H = J_2J_2^T$  y la pseudo inversa de la matriz de los vectores normales de las restricciones en el conjunto activo  $A$  se calcula de la forma  $N^* = R^{-1}J_1^T$  en donde  $[J_1|J_2] = [L^{-T}Q_1|L^{-T}Q_2] = L^{-T}Q = J$ .

$$B = Q \begin{bmatrix} R \\ 0 \end{bmatrix} = [Q_1|Q_2] \begin{bmatrix} R \\ 0 \end{bmatrix}. \quad (2-76)$$

Esta factorización se obtienen el resultado de las matrices de las ecuación 2-77, que permiten el cálculo en cada iteración de mínimos locales y el cumplimiento de restricciones para el algoritmo del conjunto activo en el paquete quadprog.

$$R^- = \begin{bmatrix} R_1 & S \\ 0 & R_2 \end{bmatrix} \quad J^- = J \begin{bmatrix} I_{|A|} & 0 & 0 \\ 0 & \bar{Q}^T & 0 \\ 0 & 0 & I_{n-q} \end{bmatrix}. \quad (2-77)$$

---

**Algoritmo Dual:**

**Paso 0:** *Encontrar el mínimo sin restricciones*

*Fijar  $x \leftarrow -G^{-1}a$ ,  $f \leftarrow \frac{1}{2}a^T x$ ,  $H \leftarrow G^{-1}$ ,  $A \leftarrow \phi$ ,  $q \leftarrow 0$ .*

---

**Paso 1:** *Seleccionar una restricción de violación :*

*Computar  $s_j(x)$  para todo  $j \in KA$*

**si**  $(V = \{j \in KA | s_j(x) < 0\})$  **entonces**

*Pare, la solución actual  $x$  es factible y óptima.*

**si no**

*Escoger  $p \in V$  y fijar  $n^+ \leftarrow n_p$  y  $u^+ \leftarrow (\frac{u}{0})$*

**si**  $q = 0$  **entonces**

*fijar  $u \leftarrow 0$ . ( $A^+ = A \cup \{P\}$ )*

**fin si**

**fin si**

---

---

**Paso 2:** *Mirar la factibilidad y determinar un nuevo  $S - pair$  :*

(a) *Determinar la dirección del paso*

*Computar  $z = Hn^+$  (dirección en el espacio primario)*

**si  $q > 0 \parallel r = N * n^+$  entonces**

(b) *Computar el tamaño de paso*

(i) Paso parcial de tamaño:  $t_1$ ; (Paso máximo en el espacio dual sin violar la factibilidad dual)

**si  $r \leq 0 \parallel q = 0$  entonces**

*fijar  $t_1 \leftarrow \infty$*

**si no**

*$t_1 \leftarrow \min_{r_j > 0, j=1, \dots, q} \left\{ \frac{u_j^+(x)}{r_j} \right\} = \frac{u_i^+(x)}{r_i}$*

**fin si**

*En el paso 2(c) de abajo*

(ii) Tamaño de paso completo:  $t_2$ ; (mínimo paso en el espacio primario en donde la restricción  $p_{th}$  se vuelve factible)

**si  $|z| = 0$  entonces**

*Fijar  $t_2 \leftarrow \infty$*

**si no**

*Fijar  $t_2 \leftarrow -s_p(x)/z^T n^+$*

**fin si**

(iii) Tamaño de paso,  $t$ :

*Fijar  $t \leftarrow \min(t_1, t_2)$*

**si no**

(c) *Determinar el nuevo  $S - pair$  y tomar el paso :*

(i) *Sin pasos en espacio primario o dual*

**si  $t = \infty$  entonces**

*Parar, problema infactible*

**fin si**

(ii) *Paso en espacio dual*

**si  $t_2 = \infty$  entonces**

*Fijar  $u^+ \leftarrow u^+ + t \left( \frac{-r}{i} \right)$  y soltar la restricción  $k$ , fijar  $A \leftarrow A \setminus \{k\}$ ,  $q \leftarrow q - 1$ , actualizar  $H$  y  $N^*$  e ir al paso 2(a).*

**fin si**

(iii) *Paso en el espacio primario y dual.*

*fijar  $x \leftarrow x + t z$ ,*

*$f \leftarrow f + t z^T n^+ \left( \frac{1}{2} t + u_{q+1}^+ \right)$*

*$u^+ \leftarrow u^+ + t \left( \frac{-r}{1} \right)$*

**si  $t = t_2$  entonces**

*(paso completo) fijar  $u \leftarrow u^+$  y añadir las restricciones;  $A \leftarrow A \cup \{P\}$ ,  $q \leftarrow q + 1$ , actualizar  $H$  y  $N^*$  e ir al Paso 1.*

**fin si**

**si  $t = t_1$  entonces**

*(paso parcial) soltar las restricciones  $K$  y fijar  $A \leftarrow A \setminus \{K\}$ ,  $q \leftarrow q - 1$ , actualizar  $H$  y  $N^*$  e ir al Paso 2(a).*

**fin si**

**fin si**

---

### 2.5.3. OSQP

El OSQP (*Operator Splitting Quadratic Program*) es un paquete solucionador numérico de optimización de propósito general que implementa un método de primer-orden, diseñado para resolver programas cuadráticos convexos de la forma 2-78, este paquete emplea una técnica de división de operadores que resuelve un sistema lineal cuasi-definido con la misma matriz de coeficientes en cada iteración; a diferencia de los paquetes anteriores, este no requiere de una función de costo definida positiva o restricciones linealmente independientes. También afirman que este algoritmo es especialmente útil para programación en sistemas embebidos que trabajen en tiempo real debido a que puede ser configurado para no realizar divisiones una vez se haya ejecutado la matriz inicial de factorización y tiene un generador de código personalizado integrable en C sin necesidad de usar un administrador de memoria, además de puede trabajar con C/C++, Python, Julia y Matlab. En algunos casos el algoritmo ha demostrado ser diez veces más rápido que los algoritmos basados en el método de punto interior [Stellato et al., 2020]. Este paquete de código abierto fue lanzado en 2017 bajo la licencia Apache Versión 2.0.

$$\begin{aligned} \text{Minimizar} \quad & (1/2)x^T P x + q^T x \\ \text{Sujeto a} \quad & A x \in C. \end{aligned} \tag{2-78}$$

En la Ecuación 2-78 se define a  $x \in R^n$  como la variable de decisión, a  $P \in S_+^n$  como una matriz semidefinida positiva, el vector  $q \in R^n$ , las restricciones como la matriz  $A \in R^{m \times n}$  y un conjunto cerrado y convexo  $C \subseteq R^m$  en donde si  $C$  toma la forma de  $C = [l, u] = \{z \in R^m | l_i \leq z_i \leq u_i, i = 1, \dots, m\}$  se puede escribir las restricciones de la Ecuación 2-78 como *Sujeto a*  $l \leq A x \leq u$  y para las restricciones de igualdad  $l_i = u_i$ .

El OSQP usa el algoritmo de divisor de operadores que se conoce como el método de multiplicadores de dirección alternativa (ADMM por sus siglas en inglés). Como está basado en un método de primer-orden el ADMM se combina con una integración auto-dual homogénea para resolver problemas de optimización cónica en donde es necesario calcular la factibilidad primaria y/o dual. Uno de los inconvenientes mencionados en la literatura es que el número de iteraciones necesarias para converger dependen de los datos del problema y de la elección de los usuarios en los parámetros de tamaño de paso para el algoritmo y a pesar de algunos resultados teóricos como los de [Giselsson and Boyd, 2017] no se puede afirmar cual es la selección adecuada de estos parámetros para optimizar la convergencia numérica del algoritmo, lo que hace que este paquete no sea un programa cónico de propósito general confiable y puede que no sea computacionalmente eficiente [Stellato et al., 2020].

El desarrollo de la solución en el OSQP comienza usando el ADMM e introduciendo una nueva variable de decisión  $z \in C$  con  $z \in R^m$  y las variables auxiliares  $\tilde{x} \in R^n$  y  $\tilde{z} \in R^m$  para

reescribir el problema en la forma de la ecuación 2-79, en donde  $I_{Ax=z}$  y  $I_C$  son funciones a tramos donde si  $Ax = z$  y  $z \in C$  entonces  $I_{Ax=z} = I_C = 0$ , de lo contrario  $I_{Ax=z} = I_C = +\infty$ .

$$\begin{aligned} \text{Minimizar} \quad & (1/2)\tilde{x}^T P \tilde{x} + q^T \tilde{x} + I_{Ax=z}(\tilde{x}, \tilde{z}) + I_C(z) \\ \text{Sujeto a} \quad & (\tilde{x}, \tilde{z}) = (x, z). \end{aligned} \quad (2-79)$$

A partir de la ecuación 2-79 se construye la condición inicial del algoritmo ADMM que involucra solucionar la ecuación 2-80, en donde  $\rho > 0$  y  $\sigma > 0$  son parámetros del tamaño de paso y con  $y \in R^m$  como  $y \in N_C(z)$  sabiendo que  $N_C(z)$  denota el cono normal de  $C$  en  $z$ . Para solucionar este sistema con restricción de igualdad se usa el método de multiplicadores de Lagrange para obtener el término  $v^{k+1}$  asociado a  $Ax = z$  en el que se obtiene el sistema lineal reducido de la matriz 2-81 en donde a los coeficientes se les refiere como la matriz  $KKT$ , matriz que siempre tiene rango completo debido a la definición de  $\rho$  y  $\sigma$  positivos.

$$\begin{aligned} \text{Minimizar} \quad & (1/2)\tilde{x}^T P \tilde{x} + q^T \tilde{x} + (\sigma/2)\|\tilde{x} - x^k\|_2^2 + (\rho/2)\|\tilde{z} - z^k + \rho^{-1}y^k\|_2^2 \\ \text{Sujeto a} \quad & A\tilde{x} = \tilde{z} \end{aligned} \quad (2-80)$$

$$\begin{bmatrix} P + \sigma I & A^T \\ A & -\rho^{-1}I \end{bmatrix} \begin{bmatrix} \tilde{x}^{k+1} \\ v^{k+1} \end{bmatrix} = \begin{bmatrix} \sigma x^k - q \\ z^k - \rho^{-1}y^k \end{bmatrix}. \quad (2-81)$$

Los pasos del algoritmo ADMM se especifican en el siguiente algoritmo, donde  $\alpha \in (0, 2)$  es un parámetro de relajación y  $\prod$  denota la proyección euclidiana sobre  $C$ , la introducción de la variable separada  $\tilde{x}$  asegura que el subproblema relacionado al paso 3 siempre tiene solución. Los pasos 4, 5, 6 y 7 son operaciones vectoriales sencillas que incluso pueden ser paralelizadas, la operación que tiene mayor costo computacional es la solución del sistema lineal en el paso 3, que se soluciona mediante un método directo o indirecto en el caso de tener un problema cuadrático de gran escala en el que es más conveniente solucionar la ecuación 2-82 en vez del sistema lineal 2-81.

---

1 : *Dados los valores iniciales  $x^0, z^0, y^0$  y los parámetros  $\rho > 0, \sigma > 0, \alpha \in (0, 2)$ .*

2 : *Repetir.*

3 :  $(\tilde{x}^{k+1}, v^{k+1}) \leftarrow \begin{bmatrix} P + \sigma I & A^T \\ A & -\rho^{-1}I \end{bmatrix} \begin{bmatrix} \tilde{x}^{k+1} \\ v^{k+1} \end{bmatrix} = \begin{bmatrix} \sigma x^k - q \\ z^k - \rho^{-1}y^k \end{bmatrix}.$

4 :  $\tilde{z}^{k+1} \leftarrow z^k + \rho^{-1}(v^{k+1} - y^k).$

5 :  $x^{k+1} \leftarrow \alpha \tilde{x}^{k+1} + (1 - \alpha)x^k.$

6 :  $z^{k+1} \leftarrow \prod(\alpha \tilde{z}^{k+1} + (1 - \alpha)z^k + \rho^{-1}y^k).$

7 :  $y^{k+1} \leftarrow y^k + \rho(\alpha \tilde{z}^{k+1} + (1 - \alpha)z^k - z^{k+1}).$

8 : *Hasta satisfacer los criterios de terminación.*

---

$$(P + \sigma I + \rho A^T A) \tilde{x}^{k+1} = \sigma x^k - q A^T (\rho z^k - y^k). \quad (2-82)$$

Los criterios de parada mencionados en el algoritmo del OSQP, hacen referencia a la detección de convergencia e infactibilidad, definiendo los residuos primarios y duales del problema como  $r_{prim} = Ax - z$ ,  $r_{dual} = Px + q + A^T y$ , permitiendo comprobar la factibilidad cuando los residuales converjan a cero  $r_{prim}^k = r_{dual}^k$  cuando  $k \rightarrow \infty$ , y el algoritmo se detiene cuando la norma del residuo sobrepasa niveles específicos de tolerancia  $\epsilon_{prim} > 0$   $\epsilon_{dual} > 0$ .

$$\|r_{prim}^k\|_\infty \leq \epsilon_{prim}, \|r_{dual}^k\|_\infty \leq \epsilon_{dual}.$$

Estos niveles de tolerancia se definen en la ecuación 2-83:

$$\begin{aligned} \epsilon_{prim} &= \epsilon_{abs} + \epsilon_{rel} \max\{\|A^k\|_\infty, \|z^k\|_\infty\}, \\ \epsilon_{dual} &= \epsilon_{abs} + \epsilon_{rel} \max\{\|Px^k\|_\infty, \|A^T y^k\|_\infty, \|q\|_\infty\}. \end{aligned} \quad (2-83)$$

Para garantizar una rápida convergencia  $\rho$  se define de forma adaptativa equilibrando los residuos, el número de iteraciones en las que se actualiza  $\rho$  es una medición de tiempo que se compara con una fracción de tiempo definida, la actualización de  $\rho$  se muestra en la ecuación 2-84.

$$\rho^{k+1} \leftarrow \rho \sqrt{\frac{\|r_{prim}\|_\infty}{\|r_{dual}\|_\infty}}. \quad (2-84)$$

## 3 Sistemas embebidos

La idea de usar sistemas embebidos como la base para el desarrollo de un PA es respaldada por las múltiples ventajas, versatilidad y facilidades que presentan estos sistemas a muchos de los problemas y procesos que tenemos en la actualidad. También se encuentran soluciones directamente relacionadas con la biomedicina, en donde se usan para la adquisición y procesamiento de señales en tiempo real [García, 2019]. En este capítulo se mostrará un panorama general de los sistemas embebidos, las razones para la selección de embebidos con sistema operativo (SO), la descripción de la arquitectura de los procesadores, así como el lenguaje de programación y los periféricos que se implementan en el desarrollo del PA.

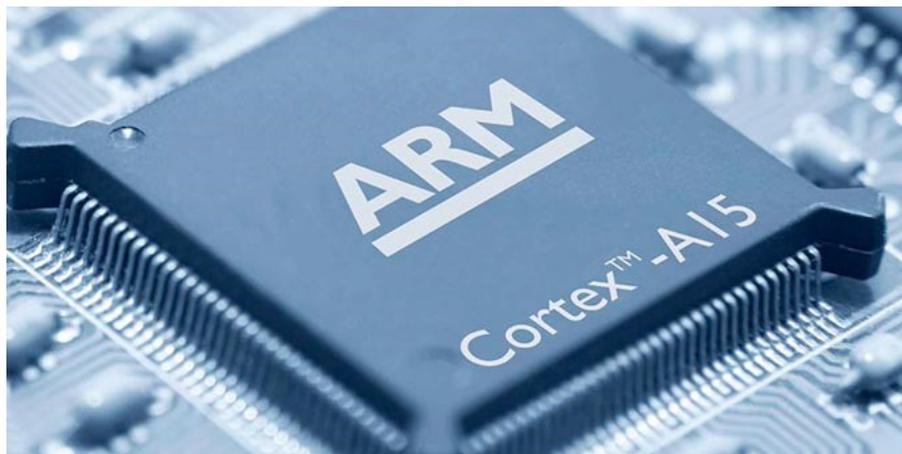
### 3.1. Embebidos

Son sistemas desarrollados para aplicaciones de propósito específico, a diferencia de los computadores personales (PC por sus siglas en inglés), que son sistemas de propósito general y pueden manejar una amplia gama de tareas. La board de un embebido generalmente contiene la mayoría de los componentes y periféricos del dispositivo, entre estos componentes sobresalen la unidad de procesamiento central (CPU por sus siglas en inglés), contenido en un microprocesador o microcontrolador con memorias de almacenamiento interno o externo (RAM, ROM, Flash, EEPROM, eMMC, SD), puertos de entradas y salidas digitales (E/S) o análogas y módulos de comunicación TCP/IP, I2C, SPI, Bluetooth, Wifi, serial (RS232 RS485), infrarrojo entre otros. La inclusión de estos u otros módulos dependen de la aplicación en la que se van a usar y muchas veces también del tamaño que se dispone para la tarjeta impresa.

A principios de los años 2000 se contaba con embebidos que son programados directamente en el procesador o el en microcontrolador mediante instrucciones en ensamblador o través de un compilador en lenguajes como C/C++, que han demostrado una gran ventaja en la rapidez de ejecución de las instrucciones del algoritmo. La arquitectura de estos procesadores se encuentran generalmente en el rango de 4, 8, 16, 32 y 64-bit, sin embargo, uno de los más usados son los ARM (Maquina avanzada RISC) de 32-bit, Figura 3-1. Una arquitectura de 32-bit en la CPU hace referencia a que sus registros internos, las operaciones en entero, las direcciones, el bus de datos pueden tomar  $2^{32}$  valores y aunque hay algunas excepciones lo que no necesariamente se cumple con esta arquitectura es el módulo de operador en punto flotante de 32-bit, como este tipo de operaciones son complejas la mayoría de las arquitectu-

ras de 32-bit tienen operador de punto flotante de 64-bit. Hoy en día el set de instrucciones de la mayoría de los procesadores es para una arquitectura de 64-bit, por lo que lentamente se ha desplazado y dejado a un lado al popular procesador de 32-bit.

Adicionalmente también se encuentran procesadores que no necesitan propiamente de un compilador para trabajar si no que se usan como una computadora de board simple (SBC por sus siglas en inglés), en el cual se puede instalar directamente un sistema operativo que generalmente es una distribución de Linux, Debian, Android y en algunos casos soportan Windows 10 o MAC. Esta versatilidad amplió el panorama de los lenguajes de programación usados en sistemas embebidos como Python, java, Basic, LabVIEW, Matlab entre otros, aunque C/C++ siguen teniendo preferencia debido a su velocidad [García, 2019].



**Figura 3-1:** Procesador ARM. Esta imagen de Autor desconocido está bajo licencia CC BY-NC-ND.<https://creativecommons.org/licenses/by-nc-nd/3.0/>

Algunos de los puntos críticos que presentan los sistemas embebidos son: a. La CPU está restringida en velocidad de procesamiento por el reloj interno y una longitud de palabra, esto consecuente con las restricciones en el consumo energético, b. Presentan dimensiones físicas acotadas, c. Ejecución de tareas en tiempo real, d. Aplicaciones críticas, monitoreo o tareas de alta prioridad que no pueden ser interrumpidas, e. Gestión de los recursos de cómputo, memoria, E/S, procesos, almacenamiento masivo y comunicaciones.

### 3.1.1. Estructura de los sistemas embebidos

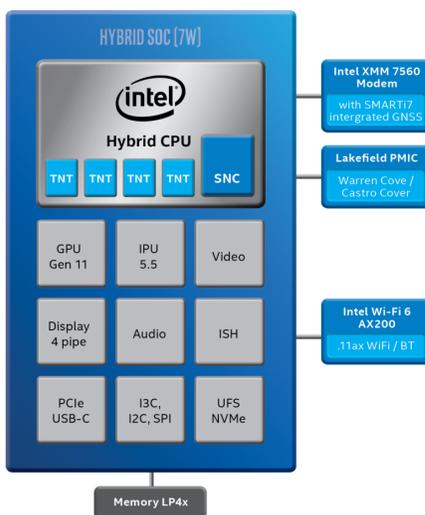
La mayoría de los sistemas embebidos tienen una arquitectura de procesador ARM, en general este tipo de procesadores se divide en sectores para cada tipo de tarea y estas se ejecutan bajo la arquitectura RISC (Conjunto de instrucciones reducidas) de 32 y 64-bit, en donde cada comando es procesado una instrucción a la vez. A diferencia de estos procesadores los denominados x86 ejecutan comandos compuestos y los procesadores son núcleos de propósito general que manejan las tareas mediante una arquitectura CISC (Conjunto de instrucciones

complejas), que son los que usualmente se emplean en PC. Algunas de las diferencias más significativas de estas dos arquitecturas se muestran en la Tabla 3-1

**Tabla 3-1:** Diferencias entre RISC y CISC

Parámetro	RISC	CISC
Tipo de instrucción	Simple	Compleja
Número de instrucciones	Reducidas (30-40)	Extendidas (100-200)
Duración por instrucción	Un ciclo	Más ciclos (4-120)
Formato de instrucción	Fijo	Variable
Ejecución de instrucción	En paralelo	Secuencial
Modos de direccionamiento	Simple	Complejo
Set de registros	Múltiples	Únicos
Complejidad	En el compilador	En la CPU

Con el paso del tiempo a pesar de estas diferencias en la forma de procesar instrucciones y de manejar la información, el resultado y los rendimientos de estas dos arquitecturas han llegado a ser en parte imperceptibles por los usuarios e incluso en algunos casos se toman conceptos de ambas arquitecturas para la concepción de procesadores híbridos como ese el caso del Galaxy Book S LTE, Figura 3-2, que se enmarca dentro de lo que se conoce como SoC (sistema en un chip), que trae la filosofía de bajo consumo energético e integra componentes como los de los microcontroladores, pero con procesadores más complejos y memorias RAM y Flash mucho más grandes. La ventaja de este tipo de arquitecturas es que mantienen a los núcleos potentes del procesador en reposo y todas las tareas se les asigna de entrada a núcleos de procesamiento liviano, cuando estos núcleos determinan que la tarea es demasiado compleja o tarda demasiado, activan los núcleos potentes y delegan las tareas, esto permite reducir considerablemente el consumo de los procesadores y mantener su rendimiento.



**Figura 3-2:** Diagrama de bloque del núcleo de un procesador Intel con tecnología Híbrida. Imagen tomada de [Intel, 2020]

### 3.1.2. Sistema operativo embebido

Como se ha mencionado, hoy en día tenemos sistemas embebidos catalogados en los SBC que usan un SO, sin embargo, cabe aclarar que a pesar de que contienen distribuciones de Linux, Windows o MAC, son sistemas operativos que se ejecutan de una forma completamente diferente a los que tenemos en los PC debido al cambio de instrucciones, para sistemas embebidos (RISC). Esto limita un poco a las SBC porque como el conjunto de instrucciones cambia no se pueden usar los software diseñados para PC en por ejemplo una Raspberry Pi; si hay un software que funciona en ambos dispositivos es porque se diseñó con el mismo propósito con el que funciona en PC pero para ser ejecutado de forma diferente por el procesador, muchos diseñadores de software, empresas y algunos contribuyentes al desarrollo de código abierto se han dado a la tarea de diseñar programas específicamente para este tipo de arquitectura, el resultado de esto es que este tipo de embebidos sean atractivos para el desarrollo de aplicaciones a pesar de que en general se tienen consumos de energía elevados con respecto a sistemas embebidos que no tienen SO, a cambio de una mayor potencia de cómputo y un sistema de memoria en general más amplio.

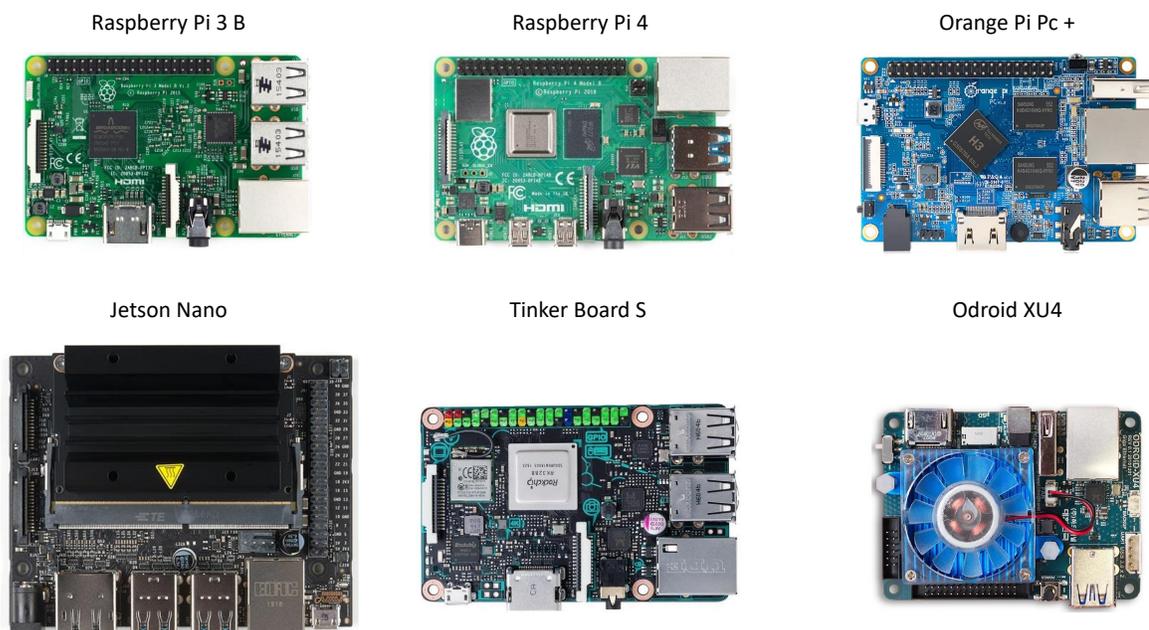
En el tema específico del PA es claro que se busca que el dispositivo sea confiable, robusto y de bajo consumo energético, una primera mirada llevaría a tomar la decisión de optar por un sistema embebido sin SO, sin embargo, hay que tener en cuenta que el proyecto del PA está en desarrollo, los modelos matemáticos de los pacientes diabéticos, los algoritmos de control, el observador, la memoria y los periféricos involucrados en el sistema aún son objeto de estudio y a pesar de que hay muchas propuestas teóricas, pocas investigaciones han llegado a tener una aplicación real en un caso de estudio clínico, esto significa que lo que se busca en este trabajo al implementar los algoritmos en un sistema embebido, es dar un paso intermedio y evaluar el desempeño y la fidelidad con la que estos sistemas ejecutan los planteamientos teóricos o de simulaciones en un esquema un poco más real, esto con el fin de identificar las fortalezas y los retos que hay que afrontar antes de obtener un producto que pueda ser implementado de forma segura. Al momento de desarrollar este trabajo el proyecto del PA encuentra en una fase intermedia y por esto se opta por implementar los algoritmos del PA en sistemas embebidos con SO, lo que permite tener una mayor flexibilidad en la programación usando lenguajes interpretados y multiparadigma, facilitar el almacenamiento de datos y permita realizar ajustes o variaciones de forma sencilla a los algoritmos de control y estimación. Posteriormente cuando se hable de la fabricación de un producto lo más recomendable será implementarlo en un sistema embebido de propósito específico sin SO y diseñado a la medida para la aplicación del PA como lo es la bomba de insulina de Medtronic [Medtronic, 2017].

La mayoría de los SO que implementan estos embebidos de arquitectura RISC, son distribuciones de GNU/Linux basado en Debian que suelen ser de código abierto y multitarea (GNU

es un sistema operativo basado en Unix que está implementado completamente en software libre). Estos SO suelen dividirse entre las versiones con entorno gráfico que contiene menús, ventanas, iconos, fondos de pantalla, entre otros, que es lo que se usa normalmente en un PC, y están las versiones reducidas sin entorno gráfico, por lo que la ejecución de algoritmos, instalación de paquetes, gestión de archivos y demás actividades se realizan por comandos en modo consola. Como en la aplicación de PA no es de utilidad tener un entorno gráfico, si es posible se instalarán este tipo de SO en los embebidos para la ejecución de los algoritmos de control en modo consola. Los gráficos, estadísticas y demás elementos resultados de las pruebas se visualizan en un computador.

## 3.2. Computadores de placa única

Con el objetivo de probar el desempeño y la fidelidad de los sistemas embebidos al ejecutar los algoritmos de control. Se usan seis SBC disponibles en el mercado con diferentes arquitecturas para realizar las pruebas, compararlas y seleccionar la de mejor desempeño. Las board son: la Raspberry Pi 3 B, la Raspberry Pi 4, la Orange Pi PC +, la Jetson Nano, el Odroid XU4 y la Tinker board S, Figura 3-3. De estos cabe resaltar que los procesadores de 32-bit son los de la Orange Pi y la Tinker board S y que la mayoría de ellos tienen unidad de procesamiento gráfico (GPU por sus siglas en inglés), periféricos como el puerto HDMI, cámara, entre otros, pero no serán puestos a prueba porque no tienen ninguna finalidad en esta aplicación.



**Figura 3-3:** Las seis SBC seleccionadas para las pruebas de desempeño

### 3.2.1. Raspberry Pi 3 B

Este SBC es uno de los más conocidos, principalmente porque fue el primero en muchos de los conceptos contenidos en este tipo de embebidos, muchas otras marcas lo usan como referencia para sacar al mercado sus propios SBC, copiando dimensiones físicas, protocolos de comunicación, periféricos e incluso el procesador. El modelo 3 B fue lanzado en 2016 [FOUNDATION, 2016], usa un procesador Quad Core de 1.2GHz (Broadcom BCM2837) de 64-bit, este procesador de 4 núcleos de tipo Cortex-A53 soporta el juego de instrucciones de 64-bit ARMv8, cuenta con memoria interna de tipo caché de nivel 1 de 8 a 64 KB para Instrucciones y de 8 a 64 KB para Datos, puede contar con hasta 2 MB de caché de nivel 2 Figura 3-4. Para el cálculo de las operaciones de punto flotante usa el vector de punto flotante VFPv4 y usa el conjunto avanzado de instrucciones (instrucciones simples, múltiples datos SIMD), que incluyen la doble precisión en operaciones vectoriales de punto flotante compatibles con el estándar IEEE 754 [ARM, 2012c].

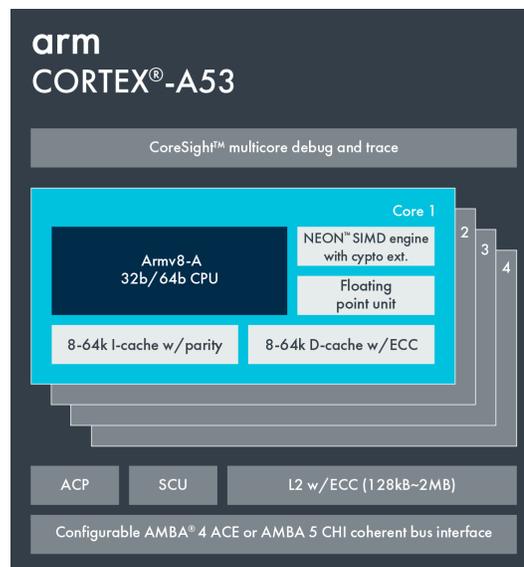
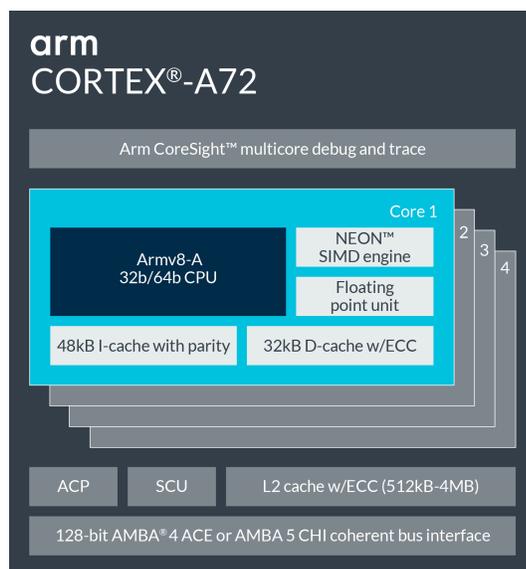


Figura 3-4: Diagrama de bloques del procesador de la Raspberry Pi 3 B

### 3.2.2. Raspberry Pi 4

El modelo 4 es a la fecha de este trabajo la última plataforma lanzada por Raspberry Pi Foundation en 2019, en hardware contiene muchas mejoras con respecto al modelo 3 B. Su procesador es el Broadcom BCM2711, quad core Cortex-A72 ARMv8 64-bit SoC @ 1.5GHz Figura 3-5, lo que se destaca además de la frecuencia del procesador es su aumento en la dirección física de 40 a 44-bit, la caché nivel 1 de 148kB/32-64kB y hasta 4 MB de caché de nivel 2 [ARM, 2016b]. Presenta mejoras en los periféricos como la inclusión de puertos USB 3.0, Bluetooth vs 5.0, aumento en los puertos UART y modelos compatibles hasta con 8GB LPDDR4-3200 de SDRAM que teóricamente mejoran el desempeño de los modelos

anteriores [FOUNDATION, 2018], además cambia el puerto para monitor, permitiendo hasta dos conexiones micro HDMI.



**Figura 3-5:** Diagrama de bloques del procesador de la Raspberry Pi 4

### 3.2.3. Organge pi PC +

Las Orange Pi son SBC basadas en el desarrollo de las Raspberry Pi fabricadas por Shenzhen Xunlong Software CO [CO, 2017], pero que en general manejan un menor precio. Trabajan SO como Android 4.4, Ubuntu, Debian y Raspbian. El procesador es un H3 Quad-core Cortex-A7 H.265/HEVC 4K con una arquitectura de 32-bit Armv7-A, este es el predecesor del procesador Cortex-A53 del que se afirma que fue superado en un 50 % en su el desempeño, posee una extensión VFP y un avanzado SIMD para realizar operaciones de punto flotante de simple y de doble precisión[ARM, 2011].

### 3.2.4. Jetson Nano

El kit de desarrollo de Nvidia la Jetson Nano del 2019, es una de las pocas SBC enfocadas en el procesamiento digital de imágenes, robótica, el aprendizaje de máquina, aprendizaje profundo y procesamiento en paralelo con el módulo Cuda, su procesador es un quad-core ARM A57 @ 1.43GHz de 64-bit [ARM, 2016a], en general tiene la misma estructura que el cortex A53 pero aprovecha un poco mejor su arquitectura, el diseño de A57 está dirigido generalmente a servidores y se basa en un proceso de 28 nm, la fortaleza de esta board no radica tanto en su procesador si no en su GPU 128-core NVIDIA Maxwell @ 921MHz[NVIDIA, 2019], como se observa en la Figura 3-3 de fabrica se implementa un disipador para la CPU y la GPU considerablemente grande comparado con el tamaño de la propia board.

### 3.2.5. ODROID-XU4

El ODROID-XU4 viene equipado con SoC Samsung Exynos5422 (Cortex A15 2Ghz y Cortex A7) Octa core CPUs que básicamente contiene dos procesadores Armv7-A [ARM, 2012a] [ARM, 2011] y una GPU integrada ARM Mali-T628 MP6, el procesador Cortex-A15 se puede implementar en una configuración de uno o varios núcleos y se puede emparejar con el Cortex-A7 para permitir configuraciones big LITTLE, esto significa que estos procesadores pueden funcionar de forma independiente como en el caso de la Orange Pi PC + o se pueden combinar para funcionar como en el Odroid-XU4. La configuración de los pines de E/S y las dimensiones físicas cambian con respecto a la referencia de los SBC (la Raspberry Pi) como se observa en la Figura 3-3, sin embargo, los periféricos y el diseño sigue la misma línea de este tipo de embebidos [HardKernel, 2015].

### 3.2.6. Tinker Board S

la Tinker Board S es una placa de Asus con arquitectura de 32 bits. Como una extensión de la arquitectura ARMv7, esta placa implementa la tecnología *Advanced Single Instruction Multiple Data* (SIMDv2) que permite soportar operaciones vectoriales con enteros, puntos flotantes y también utiliza el Vector de punto flotante versión 4 (VFPv4) para el cálculo del punto flotante que es totalmente compatible con el estándar IEEE-754, el procesador que usa esta SBC es el Cortex-A17 Quad-core 1.8GHz [ARM, 2012b], a pesar de manejar 32-bit en su arquitectura global, la memoria caché de nivel 1 es prácticamente igual a del Cortex A53 (32-64KB) pero la memoria caché integrada en la capa nivel 2 puede llegar hasta 8M, superando incluso al Cortex A72 Figura 3-6, el aumento en la memoria caché básicamente significa que se almacenan datos para que las operaciones futuras se puedan atender con mayor rapidez y no se tenga que recurrir a la lectura de la RAM que es mucho más lenta, la Dual-CH LPDDR3 de 2GB para la Tinker Board S [ASUS, 2018].

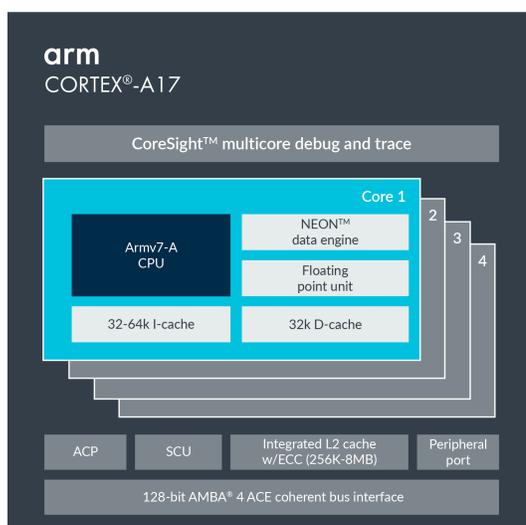


Figura 3-6: Diagrama de bloques del procesador de la Tinker Board S

### 3.2.7. Resumen

Estos seis embebidos son seleccionados debido a las diferencias de sus procesadores, en donde se presentan arquitecturas desde 32-bit y 64-bit hasta híbridos que combinan dos tipos de procesadores. El resumen de las características de estos embebidos se observa en la Tabla 3-2, en donde la principal diferencia entre los embebidos es el procesador seguido de algunas características de conectividad como el uso de Bluetooth, además la fuente de alimentación para este tipo de sistemas oscila entre 2 a 4 A con una tensión de 5v, este dato es relevante debido a que es uno de los factores más importantes en el momento de pensar en el diseño de un dispositivo portátil.

**Tabla 3-2:** Características de los sistemas embebidos

Embebido	CPU	Frecuencia	RAM	Almacenaje	Conexión	Fuente	OS
Raspberry Pi 3 B	Quad Core Broadcom BCM2837 64bit	1.2 GHz	1GB LPDDR2 SDRAM	16 GB Mi- croSD	Ethernet, BlueTooth 4.0	5v 2.5A	GNU/Linux, Windows, MAC
Raspberry Pi 4	Quad core Cortex- A72 (ARM v8) 64-bit	1.5 GHz	4 GB LPDDR4- 3200 SDRAM	16 GB Mi- cro SD	Gigabit Ethernet, BlueTooth 5.0	5V 3A	Raspbian, Windows, MAC
Orange Pi PC +	Quad- core Cortex- A7 H.265 32-bit	1.6 GHz	2GB DDR3	8GB EMMC Flash / MicroSD	Ethernet, BlueTooth 4.0	5V 3 A	Android Lubuntu, Debian, Raspbian
Jetson Nano	Quad-core ARM A57	1.43 GHz	4 GB LPDDR4 25.6 GB/s	microSD	Gigabit Ethernet	5V 4A	Distribución Linux NVIDIA Jetson Software
ODROID-XU4	Exynos5422 Cortex- A15- Cortex-A7	2 GHz	2Gbyte LPDDR3	Flash 5.0, 16 GB Mi- croSD	Gigabit Ethernet	5V 4A	Distribución GNU/Linux Debian
Tinker Board S	Rockchip RK3288 Cortex- A17 Quad Core 32-bit	1.8 GHz	2GB dual channel LPDDR3	16GB eMMC / MicroSD slot	Gigabit Ethernet, BlueTooth 4.0	5V/2-3A	Linux dis- tributions, TinkerOS

### 3.3. Lenguaje de programación

Como software base para la programación y ejecución de los algoritmos de control y estimación se usa Python 3.x, que por defecto viene en la mayoría de los SO para este tipo de embebidos. Python es un lenguaje de programación interpretado multiparadigma (que permite varios estilos de programación) y es multitarea, en la actualidad existen varios entornos de desarrollo integrado (IDE) para Python, sin embargo, debido a que la interacción con el código que contienen los embebidos no se hace mediante una interfaz gráfica si no mediante una consola en vez de usar un IDE para la programación se usa un editor de texto para Linux (Nano).

Python 3.x utiliza por defecto el estándar IEEE-754 para representación en coma flotante. Asimismo, los sistemas embebidos probados en este trabajo tienen en su arquitectura módulos compatibles con el estándar IEEE-754. Este estándar utiliza un formato de punto flotante (precisión doble) Aunque en este se utilizan 53 bits para la precisión de los datos calculados, debe tenerse en cuenta que, en general, Python aproxima los números significativos hasta 17 dígitos [S.F., 2021].

Este lenguaje posee una gran comunidad y soporte, esto permite que muchas de las librerías, funciones o programas desarrollados en otros lenguajes de programación estén disponibles en Python. Librerías como “Python Control Systems”, “scipy”, “numpy”, “qpssolvers”, entre otras son las que permiten una fácil conversión de los algoritmos de control y estimación desarrollados para el PA en computador (muchos de ellos en el software Matlab) a su implementación en un sistema embebido. Además, permite el uso de lenguajes como C/C++, que implementa algunos de los paquetes de optimización debido a la rapidez y eficiencia en la ejecución de los códigos, esto representa una gran ventaja a la hora de buscar una implementación que tarde el menor tiempo posible en encontrar la solución al problema de la dosificación óptima de insulina en cada periodo de muestreo.

### 3.4. Periféricos

En general el esquema de periféricos se resume bajo la Figura 3-7, en donde se muestran los módulos de comunicación, puertos externos y los pines GPIO. Existen algunas variaciones a este esquema como el que se presenta en la Raspberry Pi 4 que contiene varios módulos de UART o el caso del Odroid XU4 que maneja un puerto de conexión diferente pero que contienen casi los mismos elementos. Para el desarrollo de este proyecto se usan específicamente los protocolos de comunicación (I2C, UART, Bluetooth, SSH) y los pines de propósito general del GPIO para la conexión del motor paso a paso.

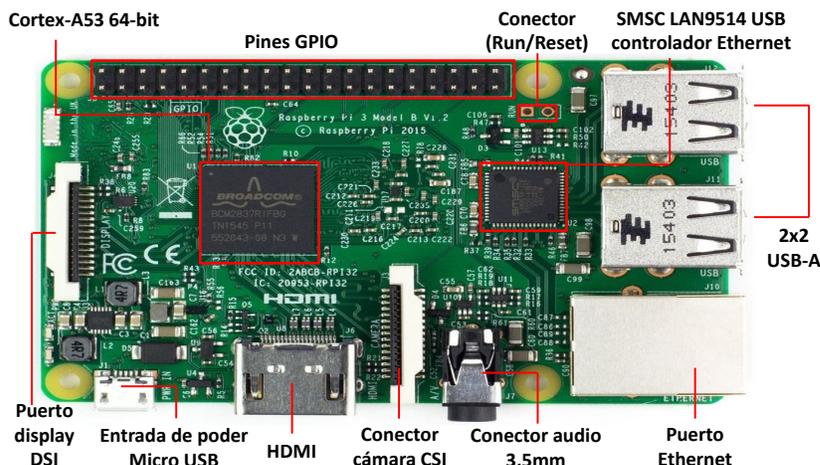
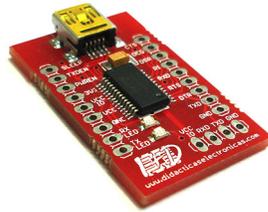


Figura 3-7: Diagrama de puertos, conexiones y periféricos de la Raspberry Pi 3 B

- $I^2C$  es la abreviación de (*Inter integrated circuit*), este es un tipo de bus de datos diseñado para la combinación entre circuitos integrados (Maestro-Esclavo), en su concepción permite el multimaestro, lo que significa que solo hace falta comenzar la comunicación para actuar como maestro, el tipo de comunicación es en serie y síncrona. La conexión se hace por medio de tres puntos, el pin SDA (Dato serial) por donde se envía la información, el pin SCL (Reloj serial), que es con el cual se coordina la emisión y recepción de información y por último un pin de GND (masa) para manejar la misma referencia sobre los niveles de voltaje de los pines SCL y SDA. Las velocidades de transmisión de datos van desde 0.1 hasta 5.0-Mbit/s, hay protocolos definidos para la escritura y lectura de información. Este protocolo se usa para la lectura del vatímetro con el que se mide el consumo energético, el sensor de temperatura infrarrojo, el gestor energético de la batería y la RTC (reloj de tiempo real), más adelante se explicará cada uno de estos elementos y en que parte de las pruebas se ven involucrados.
- UART Transmisor-Receptor Asíncrono Universal, bajo el estándar de voltaje RS232 que dicta la norma para el intercambio binario, el UART toma bytes de datos y transmite los bits individuales (Transmisión-Recepción) de forma secuencial. Se emplea para la comunicación en la transmisión de datos entre el PC y el embebido usando un conversor USB-Serial que implementa el integrado FT232RL Figura 3-8, el protocolo se configura de la siguiente forma (RS232, Rata de baudio 115200, sin paridad, bit de parada = 1, Tiempo de espera = 90 s) [Limited, 2011].
- Bluetooth es una red de área personal inalámbrica (WPAN por sus siglas en inglés) creado por *Bluetooth Special Interest Group, Inc.* Bluetooth permite la transmisión de datos entre dos dispositivos ubicados en cierta distancia, usando un sistema inalámbrico que trabaja con un ancho de banda entre los 2,402 GHz y los 2,480 GHz. es usado para la comunicación entre el sensor inteligente y el controlador, esto se explica más adelante

en el capítulo 5, la versión que usa la Raspberry Pi 3 B es la 4.0, que permite manejar un volumen de datos mayor a las versiones anteriores y maneja protocolos de bajo consumo.

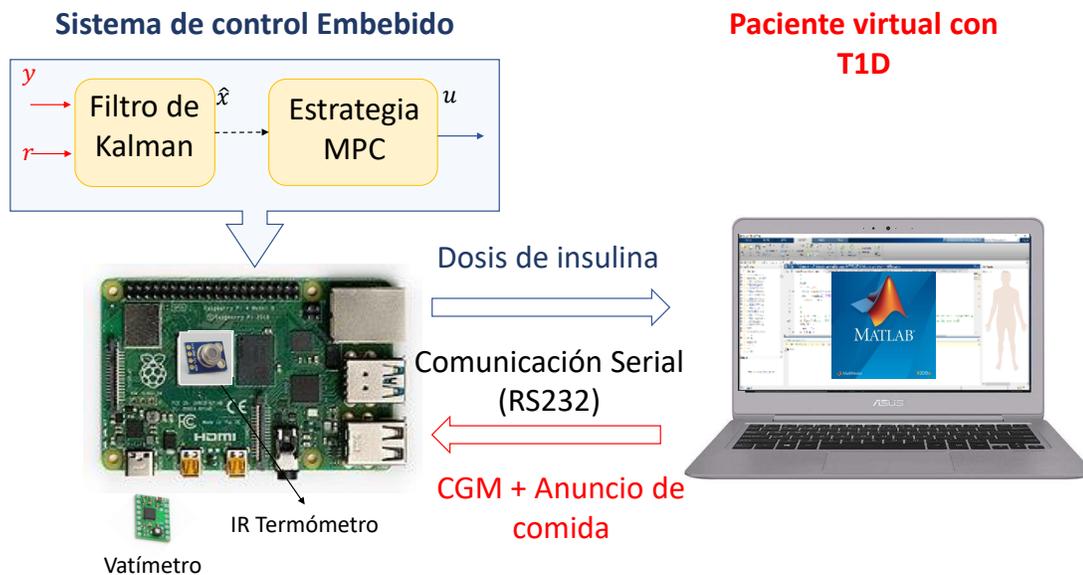
- SSH *Secure Shell* (SSH) es el nombre del protocolo cuya función es el acceso remoto a un dispositivo por medio de un canal seguro en el que toda la información es cifrada. Se usa básicamente para la programación, manejo de datos, y visualización de los registros de los SBC.



**Figura 3-8:** Tarjeta para conversión USB-Serial. Imagen tomada de [I+D, 2015]

## 4 Evaluación de los controladores MPC en los sistemas embebidos

En aplicaciones de seguridad críticas, como en el tratamiento de enfermedades, HIL se ha considerado gradualmente como un paso intermedio necesario entre las simulaciones virtuales y la creación de prototipos físicos completos [Zavitsanou et al., 2016]. Por esta razón, el enfoque de este trabajo se basa en implementar los algoritmos de control en sistemas embebidos para analizar las fortalezas y los desafíos de la implementación del PA a través de simulaciones HIL. Se probaron seis sistemas embebidos, cuyas características se muestran en la Tabla 3-2. El esquema de simulación se describe en la Figura 4-1.



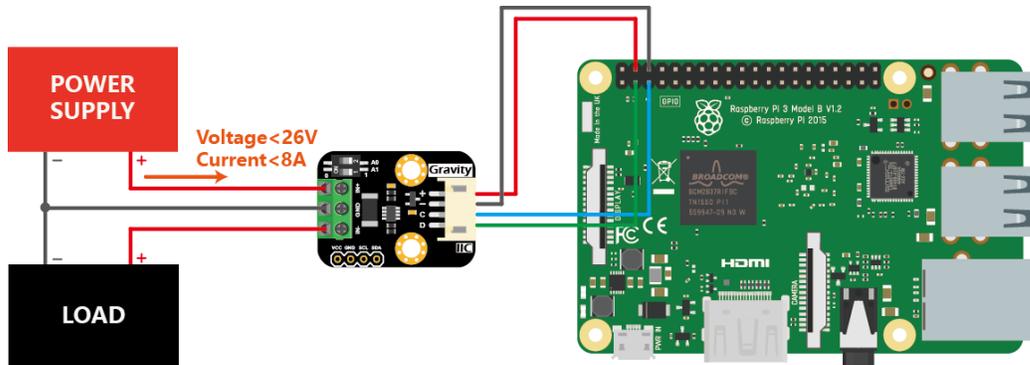
**Figura 4-1:** Implementación de hardware-in-the-loop para emular el AP. Elaboración propia a partir de imágenes bajo licencia CC BY-SA

HIL es una simulación en tiempo real entre un computador que cumple con la representación virtual de la planta y una versión real (hardware) del controlador, que en este estudio corresponde a cada sistema embebido que contiene las estrategias MPC y el estimador [MathWorks, 2021a]. Este tipo de simulaciones se realiza porque la validación de estos controladores usados en pacientes reales puede resultar costosa, complicada o incluso peligrosa para las personas. Además, el trabajo se centra precisamente en evaluar la capa física relativa

a la plataforma embebida y los paquetes de código abierto utilizados para resolver el problema de optimización involucrado. Por lo tanto, el objetivo principal es generar confianza en el diseño desarrollado en Matlab y su futura implementación en controladores físicos.

#### 4.0.1. Sensores en el esquema de HIL

El sensor usado para adquirir la información del consumo energético es el sensor Gravity I2C vatímetro digital, este sensor de alta precisión y exactitud puede medir voltajes en los rangos de  $0 \sim 26V$  con una resolución de  $4mV$  y un error relativo típico  $< \pm 0,2\%$ , también puede medir corriente bidireccional en los rangos de  $0 \sim \pm 8A$  con una resolución de  $1mA$  y un error relativo  $< \pm 0,2\%$ , los rangos de potencia son de  $0 \sim 206W$  con resoluciones por hardware de  $20mW$  y con corrección por software de  $4mW$ , el consumo de corriente en reposo es de  $0,7mA$ , dentro de las aplicaciones de este sensor se encuentra la evaluación del consumo de energía de motores, controladores o módulos electrónicos o la vida útil de la batería de sistemas de energía solar [the Future, 2021]. El esquema de conexión se muestra en la Figura 4-2, en donde cada sistema embebido por separado es alimentado con una fuente de poder de  $5V \approx 3000mA$  que es conectada al embebido por medio del sensor Gravity.



**Figura 4-2:** Conexión del sensor Gravity: I2C Digital Wattmeter. Imágenes de [the Future, 2021]

Para la toma de muestras de temperatura en el procesador se usa el sensor infrarrojo de alta precisión sin contacto MLX90614 (GY - 906), es calibrado de fabrica para manejar rangos de medición de objetos entre  $-70 \sim 380 \text{ }^{\circ}C$  con rangos de temperatura ambiente de  $-40 \sim 125 \text{ }^{\circ}C$ , con una precisión de  $0,5 \text{ }^{\circ}C$  y una resolución de  $0,02 \text{ }^{\circ}C$  en el que usan 17-bits para la conversión Análogo Digital (ADC) [melexis microelectronic systems, 2006]. Este sensor es ubicado aproximadamente a  $2cm$  del procesador o disipador de cada uno de los embebidos y se toman muestras cada  $0,006seg$  durante 1 minuto en el que se toma la temperatura del procesador en estado de reposo y cuando se estresa resolviendo el problema de optimización más complejo (ZMPC-AV-OF), registrando la temperatura ambiente actual antes de cada una de las pruebas.

## 4.0.2. Patrón de referencia (Matlab)

Se estableció la referencia de desempeño obtenida en simulación con Matlab, es decir, tanto la planta como el controlador están funcionando en Matlab, utilizando quadprog como algoritmo de optimización. Se usó un computador con las siguientes características: Core i9-9880H con 16 GB de RAM. El sistema operativo del PC es Windows 10 y la versión de Matlab es la R2020a. Los datos obtenidos con este PC se usan como referencia para comparar los rendimientos obtenidos en cada prueba con los sistemas embebidos. Además en la simulación de HIL de la Figura 4-1 este es el computador que hace las veces de paciente virtual.

## 4.0.3. Configuración de los sistemas embebidos

Cada uno de los embebidos es configurado para tener una dirección IP estática con el fin de realizar la conexión SSH y configurar o ejecutar los algoritmos de control, para esto se modificó el archivo dhcpcd.conf para incluir la IP correspondiente *-interface eth0 - static ip\_address=192.168.1.X*.

Como ya se ha definido anteriormente los SO que usan los sistemas embebidos son de distribuciones UNIX/Linux y para los algoritmos de control y estimación se usa el lenguaje de programación Python3.x. Para ejecutar estos códigos que involucran las operaciones matriciales, representación en espacios de estado, discretización impulsiva o de Zoh, conversión de matrices de formato de columna dispersa comprimida (CSC por sus siglas en inglés), ejecutar los algoritmos de optimización (quadprog, CVXOPT, OSQP), control de los pines GPIO y el uso de los canales de comunicación  $I^2C$ , Serial y Bluetooth es necesaria la instalación de las siguientes librerías o paquetes usando los comandos:

- *sudo apt-get install python3-pip*
- *sudo apt-get install python3-dev python3-rpi.gpio*
- *sudo apt-get install python3-numpy*
- *sudo apt-get install python3-scipy*
- *sudo apt-get install python3-matplotlib*
- *sudo apt-get install cython*
- *sudo apt-get install git*

- *git clone https://github.com/python-control/python-control.git*  
*cd python-control*  
*sudo python3 setup.py build*  
*sudo python3 setup.py install*
- *sudo apt-get install python3-dev*
- *sudo apt-get install build-essential*
- *sudo apt-get install python3 libopenblas-dev*
- *sudo apt-get install python3 libatlas-base-dev*
- *sudo apt-get install python3 libblas-dev*
- *sudo apt-get install python3 liblapack-dev*
- *sudo apt-get install python3 libsuitesparse-dev*
- *sudo apt-get install python3 libsdsp-dev*
- *sudo apt-get install python3 libfftw3-dev*
- *sudo apt-get install python3 libglpk-dev*
- *sudo apt-get install python3 libgsl-dev*
- *git clone https://github.com/cvxopt/cvxopt.git*  
*cd cvxopt*  
*sudo nano setup.py*  
“Reemplazar por la dirección de instalación de la librería **suitesparse**”  
# Guess SUITESPARSE\_LIB\_DIR and SUITESPARSE\_INC\_DIR  
if glob(“/usr/lib/arm-linux-gnueabi/hf/libsuitesparse\*”):  
# Ubuntu/Debian  
SUITESPARSE\_LIB\_DIR = “/usr/lib/arm-linux-gnueabi/hf”
- *sudo pip3 install Wheel*
- *sudo git clone https://github.com/rmcgibbo/quadprog.git* *cd quadprog/quadprog* *sudo python3 setup.py build* *sudo python3 setup.py install*
- *sudo pip3 install qpsolvers*
- *sudo pip3 install pyserial*
- *sudo pip3 install setuptools-scm*

- *sudo pip3 install setuptools*
- *sudo pip3 install glpk*
- *sudo pip3 install PICOS*
- *sudo pip3 install ecos*
- *sudo pip3 install smcp*
- *sudo apt-get install cmake*
- *git clone --recurse-submodules https://github.com/oxfordcontrol/osqp-python*  
*cd osqp-python*  
*sudo python3 setup.py install*
- *sudo apt-get install python3-bluetooth*

## 4.1. Escenario de simulación y métricas

Para definir completamente el escenario se usa el esquema de HIL con seis embebidos (Raspberry Pi 3, Raspberry Pi 4, Orange Pi PC +, Jestson Nano, Odroid XU4, Tinker Board S) con diferentes arquitecturas de procesador, en el que se implementan cuatro estrategias de control enmarcadas en el control predictivo por modelo (MPC, ZMPC, ZMPC-AV, ZMPC-AV-OF) y se usan tres paquetes de optimización CVXOPT V1.2.6, quadprog V0.1.8 y OSQP V0.6.2 para resolver el problema de la dosificación de insulina óptima (En la Tabla 4-1 se observan los valores iniciales de las tolerancias de cada paquete), tomando a un computador como la representación de paciente virtual con T1DM (10 modelos lineales identificados con el modelo de Ruan a partir del simulador UVA/Padova de 2014 que es el primero aprobado por la FDA).

Los parámetros del modelo se varían para emular el efecto neto de las variaciones fisiológicas en el sujeto que inducen hipoglucemia o hiperglucemia. Para ello, se utiliza un escenario de simulación de 3 días en el que se varían los parámetros de la planta para inducir hipoglucemia en el primer día, en el segundo día no hay variaciones paramétricas y se cambian los parámetros para inducir un estado de hiperglucemia en los sujetos en el tercer día. Los parámetros se modifican simultáneamente en  $\pm 10\%$  de sus valores nominales (los identificados para ajustarse a los datos [Villa-tamayo, 2021]). La información de la modificación de los parámetros no se proporciona al controlador.

**Tabla 4-1:** Tolerancias de los paquetes de optimización

Paquete	Valor de la tolerancia			
<b>quadprog (MATLAB)</b>	TolPGG =1e-5	Tolcon =1e-4	TolX =1e-4	Tolfun =1e-4
<b>quadprog (Python)</b>	N.A.	N.A.	N.A.	N.A.
<b>OSQP</b>	Eps_abs =1e-3	Eps_rel =1e-3		
<b>CVXOPT</b>	Feastol <1e-7	Abstol <1e-7	Reltol <1e-6	

Se proporcionan comidas con diferente contenido de carbohidratos a los sujetos virtuales. Las comidas se establecieron a las 7, 10, 13 y 19 horas con un contenido de carbohidratos de 55, 20, 90 y 70 g, respectivamente. Se consideró una duración de 15 min para cada comida. las comidas se anunciaron a las estrategias de control, es decir, esta información se envía al sistema embebido.

Se considera que el ruido del sensor en la medición se envía a través de la señal CGM que contiene la glucemia actual de los pacientes. La señal CGM se ha estudiado en base a los dispositivos móviles Dexcom G5. Posteriormente, la señal CGM se transmite a través de un protocolo de comunicación (RS232) al controlador físico, donde se realizan los cálculos de las acciones de control. El controlador consta de un filtro Kalman para estimar los estados no medidos además de reducir el efecto del ruido de medición, y una estrategia MPC que calcula la dosis de insulina a inyectar por medio del problema de optimización. A continuación, en cada instante de tiempo  $k$ , se envía la cantidad de insulina que se administrará al paciente virtual usando el mismo protocolo de comunicación (RS232).

En cuanto a la configuración de las simulaciones, los parámetros de control utilizados para cada estrategia MPC son: un horizonte de predicción  $H_p = 80$ , un horizonte de control  $H_c = 10$ , un tiempo de muestreo  $T_s = 5$  y el estado inicial dado por  $X_0 = [115, x_{2ss}, x_{3ss}, 0, 0]^T$ , donde  $x_{2ss}$  y  $x_{3ss}$  se refieren al estado de equilibrio del modelo 2-8, dado por  $x_{2ss} = x_{3ss} = (\theta_{a2} - 115 \theta_{a3}) / \theta_{a0}$ . Para el MPC la referencia se estableció en  $95mg/dL$ , y para las otras tres estrategias, la zona objetivo se estableció en  $[85 - 105]mg/dl$ . Los valores de las matrices de peso  $Q$ ,  $R$ ,  $P$  y  $\gamma$  para el controlador y las matrices  $Q_e$ ,  $R_e$ , y  $G_w$  para el estimador se observan en la Tabla 4-2. Se ajustaron en Matlab para cada paciente y luego, estos valores se utilizaron en las pruebas de los sistemas embebidos y los paquetes de optimización.

**Tabla 4-2:** Parámetros de sintonía de las estrategias por paciente.

MPC	Pac. 1	Pac. 2	Pac. 3	Pac. 4	Pac. 5	Pac. 6	Pac. 7	Pac. 8	Pac. 9	Pac. 10
Q	100	100	100	100	100	100	100	100	50	100
R	3 <sup>6</sup>	3 <sup>6</sup>	3 <sup>6</sup>	4 <sup>6</sup>	3 <sup>6</sup>	1,1 <sup>5</sup>	1,46 <sup>6</sup>	1 <sup>6</sup>	3 <sup>5</sup>	3 <sup>6</sup>
Qe	$MPC_{Qe}$	$MPC_{Qe}$	$MPC_{Qe}$	$MPC_{Qe}$	$MPC_{Qe}$	$MPC_{Qe}$	$MPC_{Qe}$	$MPC_{Qe}$	$MPC_{Qe}$	$MPC_{Qe}$
Re	6,5 <sup>7</sup>	6,5 <sup>7</sup>	3,5 <sup>7</sup>	1,05 <sup>8</sup>	5,2 <sup>7</sup>	1,83 <sup>8</sup>	9,83 <sup>8</sup>	4,6 <sup>8</sup>	8,86 <sup>8</sup>	6,8 <sup>7</sup>
ZMPC	Pac. 1	Pac. 2	Pac. 3	Pac. 4	Pac. 5	Pac. 6	Pac. 7	Pac. 8	Pac. 9	Pac. 10
Q	3,9 <sup>3</sup>	1,0 <sup>3</sup>	1,0 <sup>2</sup>	5,8 <sup>2</sup>	2,5 <sup>2</sup>	3,2 <sup>2</sup>	3,5 <sup>3</sup>	1,0 <sup>3</sup>	1,5 <sup>2</sup>	1,0 <sup>2</sup>
R	5,0 <sup>5</sup>	3,0 <sup>6</sup>	3,0 <sup>6</sup>	1,0 <sup>7</sup>	6,0 <sup>6</sup>	1,1 <sup>5</sup>	4,6 <sup>5</sup>	6,5 <sup>6</sup>	3,0 <sup>5</sup>	3,0 <sup>6</sup>
Qe	$MPC_{Qe}$	$MPC_{Qe}$	$MPC_{Qe}$	$MPC_{Qe}$	$MPC_{Qe}$	$MPC_{Qe}$	$MPC_{Qe}$	$MPC_{Qe}$	$MPC_{Qe}$	$MPC_{Qe}$
Re	6,5 <sup>8</sup>	6,5 <sup>7</sup>	3,5 <sup>7</sup>	6,5 <sup>7</sup>	6,2 <sup>7</sup>	1,5 <sup>8</sup>	8,83 <sup>8</sup>	9,0 <sup>7</sup>	8,86 <sup>8</sup>	6,8 <sup>7</sup>
ZM-AV	Pac. 1	Pac. 2	Pac. 3	Pac. 4	Pac. 5	Pac. 6	Pac. 7	Pac. 8	Pac. 9	Pac. 10
Q	3,0 <sup>2</sup>	3,0 <sup>2</sup>	1,0 <sup>2</sup>	2,0 <sup>3</sup>	2,0 <sup>2</sup>	3,0 <sup>3</sup>	3,0 <sup>2</sup>	1,0 <sup>4</sup>	3,0 <sup>2</sup>	3,0 <sup>3</sup>
R	5,0 <sup>5</sup>	5,0 <sup>5</sup>	9,0 <sup>4</sup>	5,0 <sup>7</sup>	6,0 <sup>5</sup>	5,0 <sup>6</sup>	4,6 <sup>6</sup>	6,5 <sup>6</sup>	3,0 <sup>6</sup>	3,0 <sup>4</sup>
Qe	$MPC_{Qe}$	$MPC_{Qe}$	$MPC_{Qe}$	$MPC_{Qe}$	$MPC_{Qe}$	$MPC_{Qe}$	$MPC_{Qe}$	$MPC_{Qe}$	$MPC_{Qe}$	$MPC_{Qe}$
Re	6,8 <sup>8</sup>	6,8 <sup>8</sup>	9,8 <sup>6</sup>	6,8 <sup>9</sup>	6,2 <sup>8</sup>	6,8 <sup>9</sup>	6,5 <sup>8</sup>	9,0 <sup>8</sup>	8,8 <sup>8</sup>	6,8 <sup>8</sup>
$\gamma$	10 <sup>4</sup>	10 <sup>4</sup>	10 <sup>4</sup>	10 <sup>4</sup>	10 <sup>4</sup>	10 <sup>4</sup>	10 <sup>4</sup>	10 <sup>4</sup>	10 <sup>4</sup>	10 <sup>4</sup>
AV-OF	Pac. 1	Pac. 2	Pac. 3	Pac. 4	Pac. 5	Pac. 6	Pac. 7	Pac. 8	Pac. 9	Pac. 10
Q	1,3 <sup>3</sup>	3,0 <sup>3</sup>	2,0 <sup>2</sup>	7,4 <sup>2</sup>	2,5 <sup>2</sup>	2,2 <sup>3</sup>	1,6 <sup>2</sup>	1,8 <sup>3</sup>	6,0 <sup>3</sup>	5,0 <sup>3</sup>
R	1,0 <sup>7</sup>	5,0 <sup>6</sup>	5,0 <sup>6</sup>	4,5 <sup>6</sup>	1,5 <sup>6</sup>	4,0 <sup>6</sup>	3,6 <sup>6</sup>	7,0 <sup>6</sup>	5,0 <sup>7</sup>	5,0 <sup>7</sup>
Qe	$OF_{Qe}$	$OF_{Qe}$	$OF_{Qe}$	$OF_{Qe}$	$OF_{Qe}$	$OF_{Qe}$	$OF_{Qe}$	$OF_{Qe}$	$OF_{Qe}$	$OF_{Qe}$
Re	6,5 <sup>9</sup>	6,8 <sup>9</sup>	6,8 <sup>9</sup>	3,6 <sup>9</sup>	3,2 <sup>9</sup>	8,8 <sup>9</sup>	6,5 <sup>9</sup>	8,0 <sup>9</sup>	6,8 <sup>9</sup>	6,8 <sup>9</sup>
$\gamma$	10 <sup>4</sup>	10 <sup>4</sup>	10 <sup>4</sup>	10 <sup>4</sup>	10 <sup>4</sup>	10 <sup>4</sup>	10 <sup>4</sup>	10 <sup>4</sup>	10 <sup>4</sup>	10 <sup>4</sup>
Gw	$OF_{Gw}$	$OF_{Gw}$	$OF_{Gw}$	$OF_{Gw}$	$OF_{Gw}$	$OF_{Gw}$	$OF_{Gw}$	$OF_{Gw}$	$OF_{Gw}$	$OF_{Gw}$

Las matrices constantes en los diez pacientes se describen como:  $MPC_{Qe} = [1^3 \ 1 \ 1 \ 1 \ 2^6]$   $AV - OF_{Qe} = [10 \ 1 \ 1 \ 1 \ 200 \ 5]$ ,  $OF_{Gw} = [10 \ 10 \ 10 \ 10 \ 10 \ 10]$ . \* Pac = Paciente. Para todos los números  $A^n = Ax10^n$

Finalmente, se diseñó un índice cuantitativo para seleccionar el sistema embebido más adecuado para el sistema del PA. El índice está compuesto por siete criterios ponderados según su importancia. Para la selección de la mejor plataforma se consideraron los siguientes criterios:

- i) Porcentaje de tiempo en rango.
- ii) Número de eventos que sobrepasan 180mg/dL.
- iii) Coeficiente de variación de la glucosa en sangre.
- iv) Tiempo de simulación.
- v) Energía consumida.
- vi) Temperatura del procesador.
- vii) La precisión de la respuesta glucémica obtenida con cada sistema embebido con respecto a la referencia Matlab (medido con el error absoluto medio).

El criterio del error absoluto medio tiene la ponderación más alta destacando su importancia en la selección (se estableció una ponderación de 30), este término está directamente

relacionado con el objetivo principal de este capítulo, que es evaluar el rendimiento de los controladores diseñados por computadora frente a su implementación en sistemas físicos embebidos. El segundo término más importante fue ponderado con 20, y es el consumo de energía, ya que la visión del páncreas artificial es ser implementado como un dispositivo portátil con un suministro de energía limitado.

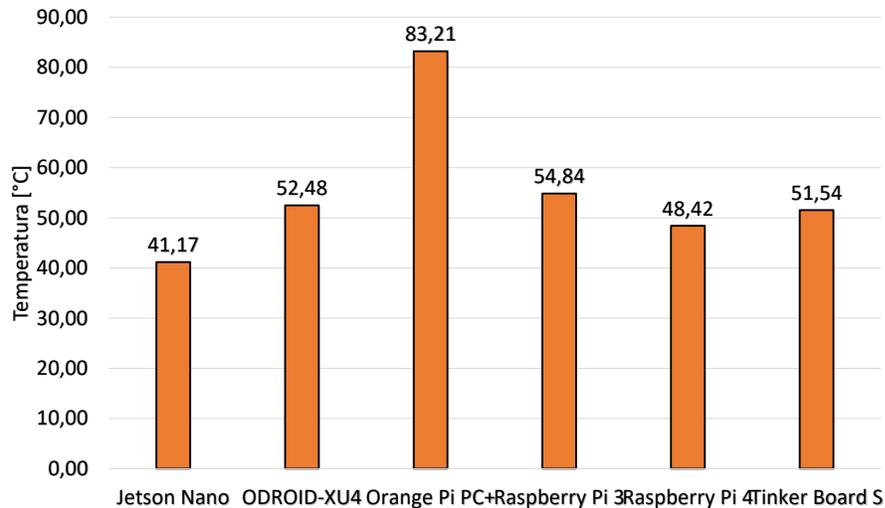
El resto de los factores son de igual importancia y se establecieron con un peso de 10. Los términos (i) - (iii) pueden evaluarse y mejorarse en pasos anteriores (es decir, en la fase de diseño) a la implementación de la estrategia de control en el sistema embebido. Los términos (iv) y (v) están relacionados con características menores que resultan ser menos importantes en los SBC. El tiempo de simulación se consideró menos importante ya que en una bomba de insulina comercial, el tiempo de muestreo es de 1 o 5 minutos, y de todas las pruebas experimentales, el tiempo máximo obtenido para una iteración fue de 6,9 segundos. El criterio (vi), aunque afecta el rendimiento del dispositivo portátil, puede manejarse con disipadores de calor o ventiladores de ser el caso, aunque es preferible no tener que usarlos debido al espacio que ocupan.

El procedimiento de ponderación para cada sistema embebido consiste en determinar la plataforma con mejor desempeño en cada criterio y asignarle el peso total de ese término. Los valores para el resto de los sistemas embebidos en cada criterio son proporcionales a la diferencia con el mejor.

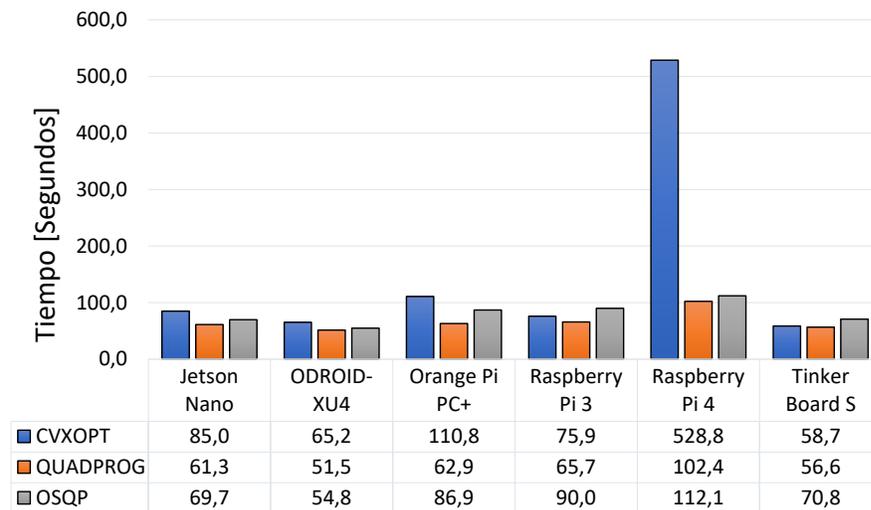
## 4.2. Resultados

Los resultados de temperatura de cada sistema embebido al estresar sus procesadores con la estrategia más compleja (el ZMPC-AV-OF) se pueden observar en la Figura 4-3, en donde el sistema embebido con el menor aumento de temperatura al ejecutar el código fue la Jetson Nano registrando  $41 \pm 0,3^{\circ}C$ . En contraste, el sistema embebido con la temperatura más alta fue el Orange Pi PC + registrando valores promedios de  $83 \pm 2,245^{\circ}C$ . La medición promedio de la temperatura ambiente para todas las pruebas fue de  $25 \pm 0,04^{\circ}C$ , y la temperatura en reposo para todos los sistemas se mantuvo cerca de  $42^{\circ}C$ .

En la Figura 4-4 se presenta el promedio de tiempo total de los cálculos para ejecutar las cuatro estrategias MPC usando los tres paquetes de optimización y los seis sistemas embebidos. El algoritmo de quadprog resultó ser el paquete más rápido en todos los embebidos. Además, de los SBC, el ODROID-XU4 presentó el tiempo total de simulación más bajo con 51,5s, seguido por el Tinker Board S con 56,62s. Por otro lado, la Raspberry Pi 4 tuvo el mayor tiempo total de simulación con los tres paquetes solucionadores, especialmente al utilizar el paquete CVXOPT con el que tuvo un tiempo de simulación total cinco veces mayor que con los otros paquetes.



**Figura 4-3:** Temperatura en el procesador de los 6 SBC al usar el ZMPC-AV-OF



**Figura 4-4:** Tiempo de procesamiento promedio de los 6 SBC con las 4 estrategias

Los resultados del consumo de energía se visualizan en la Figura 4-5. En esta se presenta el valor promedio de la energía consumida al ejecutar las cuatro estrategias MPC. En la mayoría de los casos, existe un mayor consumo vinculado a las estrategias más complejas (ZMPC-AV y ZMPC-AV-OF). El paquete quadprog generó el menor consumo en todos los sistemas embebidos y se destaca la Raspberry Pi 3 con un consumo promedio de  $2,05W/min$ , seguida por la Orange Pi PC + con  $3W/min$ . Aunque el ODROID-XU4 presentó la potencia instantánea más alta de todos los sistemas embebidos, no es el de mayor consumo de energía debido a la velocidad con la que ejecuta los algoritmos. La Raspberry Pi 4 (especialmente con el paquete CVXOPT) fue el sistema de mayor consumo energético.

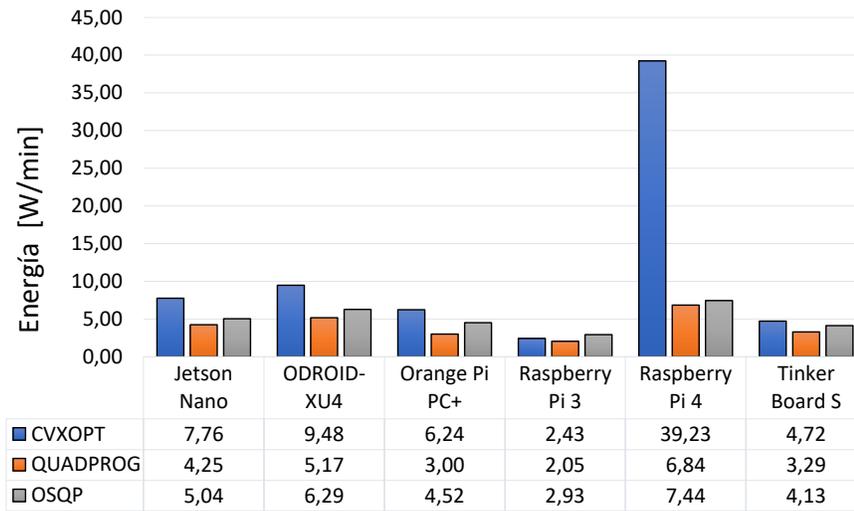


Figura 4-5: Energía promedio consumida en los 6 SBC con las 4 estrategias

De la Figura 4-6 a la 4-9, se presenta la evolución de la glucemia controlada con las cuatro estrategias MPC para cada SBC y paquete de optimización. Cada figura muestra el nivel promedio de glucosa en sangre en la población adulta. En la Figura 4-6 se observa el resultado del desempeño de la estrategia MPC en donde se muestra un comportamiento típico y esperado, ya que este controlador base no tiene en cuenta las variaciones paramétricas por lo que se ve claramente afectado en el primer día en donde se induce un caso de hipoglucemia la mayoría de los embebidos alcanzan a tener niveles por debajo de  $70mg/dl$ , así mismo para el tercer día se llegan a niveles de hiperglucemia superiores a  $240mg/dl$ . Hay que notar que para los tres paquetes de optimización y los seis embebidos el desempeño es prácticamente el mismo, en este punto no se observan diferencias significativas entre ellos.

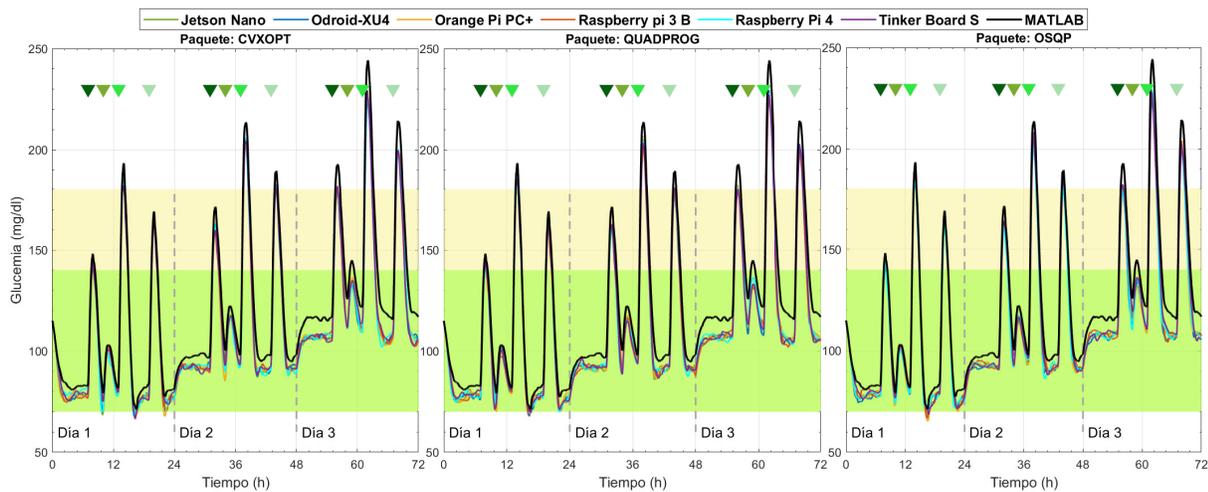
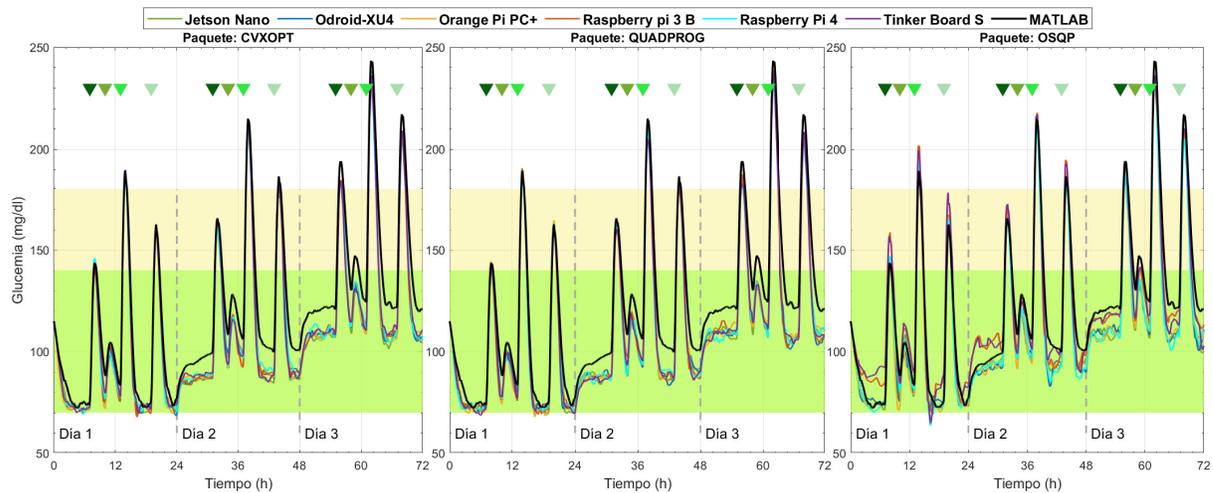


Figura 4-6: Evolución de la glucemia bajo la estrategia MPC con los tres paquetes

En la Figura 4-7 se muestran el rendimiento de los sistemas controlados por la estrategia ZMPC. Esta estrategia comparada con el MPC base no presenta muchas diferencias, en

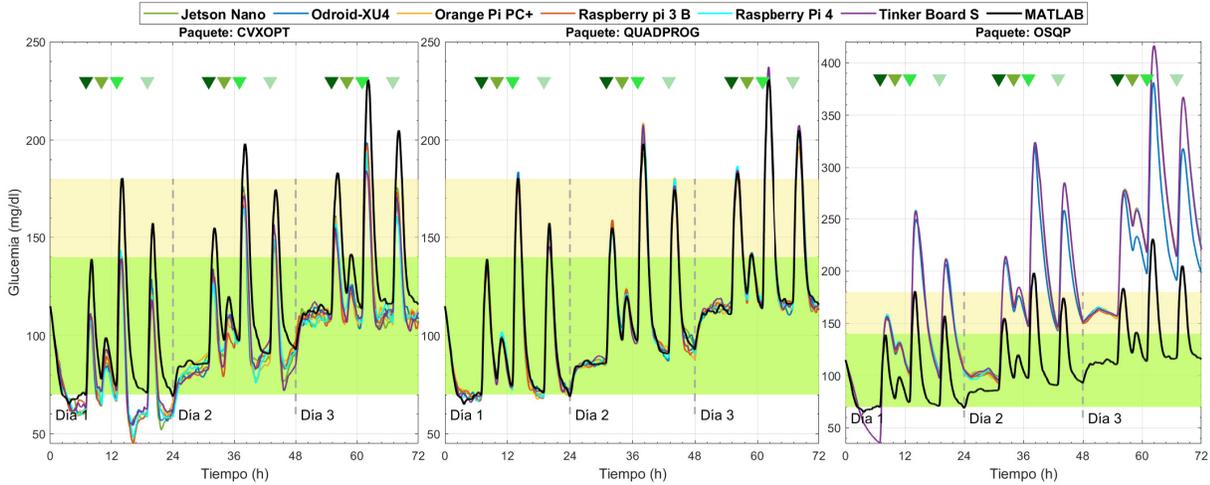
ambos se nota claramente el efecto que tiene la variación paramétrica para los casos de hiper e hipoglucemia y aunque en ambas estrategias hay casos de hipoglucemia, los niveles de glucemia registrados en el ZMPC no llegan a ser tan bajos como en el caso anterior. En general los niveles de glucemia en estado estacionario crecen debido a que el controlador no sigue una referencia si no un rango y este se mantiene en el rango superior  $105\text{mg/dl}$  en los días 2 y 3, Los niveles de glucosa en sangre se mantienen por debajo de los obtenidos con la referencia Matlab (líneas negras). A pesar de que no hay grandes diferencias entre todos los paquetes y entre las estrategias MPC base y ZMPC, en el paquete OSQP y la estrategia ZMPC se empiezan a notar algunas discrepancias en las trayectorias del registro de glucemia.



**Figura 4-7:** Evolución de la glucemia bajo la estrategia ZMPC con los tres paquetes

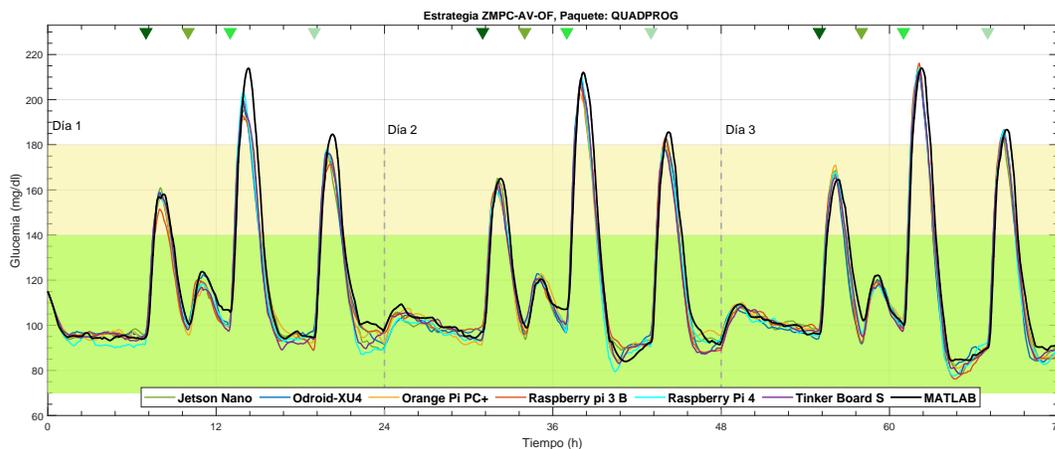
La Figura 4-8 muestran los resultados de la estrategia ZMPC-AV en donde se observan los problemas en el desempeño de los paquetes OSQP y CVXOPT, las tolerancias que inicialmente se plantearon en la Tabla 4-1 se relajaron para obtener soluciones factibles, es decir, para evitar la divergencia del solucionador y tiempos de cálculo excesivamente altos (más de 20 min por simulación). Para el paquete CVXOPT, las tolerancias se establecieron como  $Abstol = 3e - 7$ ,  $Reltol = 4e - 2$  y  $Feastol = 4e - 2$ , y para el paquete OSQP como  $Eps_{abs} = 90$  y  $Eps_{rel} = 9,9$ . Con estos dos paquetes, la evolución de la glucemia controlada con los sistemas embebidos varía considerablemente con respecto al rendimiento de referencia (obtenido con Matlab). Al usar el paquete CVXOPT a pesar de que en los días 2 y 3 se obtienen resultados aceptables en el día 1 se presentan casos de hipoglucemia severa. Por otro lado, al usar el paquete OSQP se sigue una tendencia, pero no lo logra cumplir con los objetivos de control en los tres días llevando a los pacientes virtuales a casos de hipo e hiperglucemia severa. En contraste, con el paquete quadprog a pesar de tener algunos casos de hipoglucemia la evolución de la glucemia controlada con cualquiera de los embebidos resultó ser muy similar a la obtenida con Matlab incluso en los periodos de estado estacionario, a diferencia de las dos estrategias anteriores. Los resultados obtenidos con los paquetes

CVXOPT y OSQP con la estrategia ZMPC-AV-OF no se muestran porque generaron los mismos problemas ilustrados con la estrategia ZMPC-AV.



**Figura 4-8:** Evolución de la glucemia bajo la estrategia ZMPC-AV con los tres paquetes de optimización

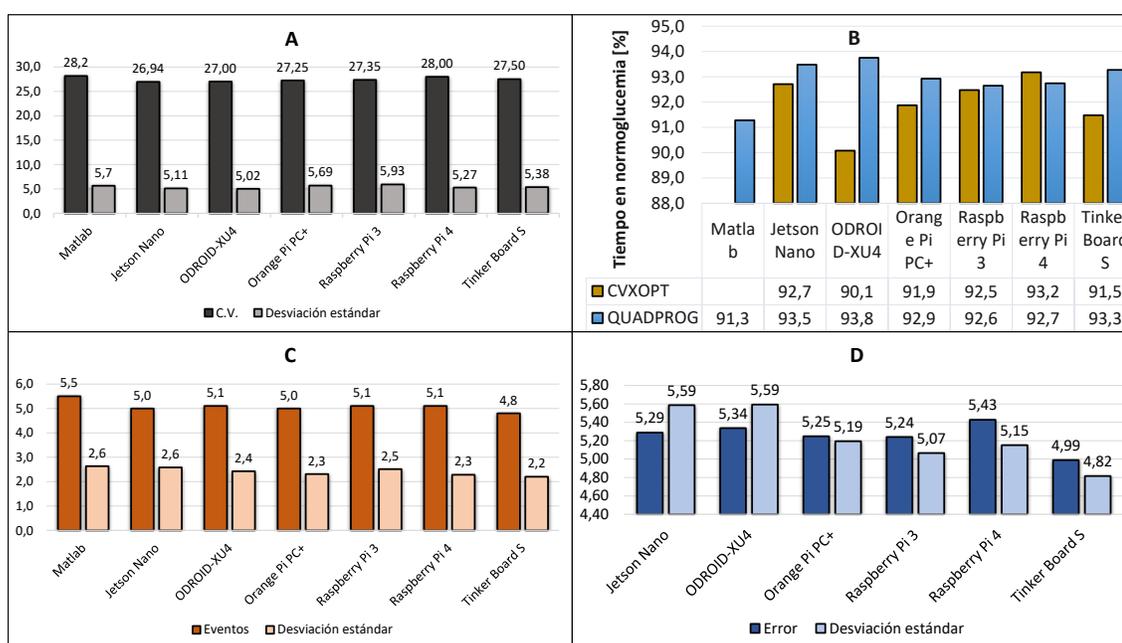
Los resultados obtenidos con la estrategia ZMPC-AV-OF se pueden visualizar en la Figura 4-9 cuando se utiliza el solucionador quadprog, en la figura se observa que todos los SBC logran tener un resultado muy cercano al arrojado en las simulaciones en computador usando Matlab, con lo que se logra evitar llevar a los pacientes virtuales a casos de hipoglucemia y reducir los eventos de hiperglucemia además de manejar adecuadamente las variaciones paramétricas en el día 1 y 3.



**Figura 4-9:** Evolución de la glucemia usando ZMPC-AV-OF con el paquete quadprog

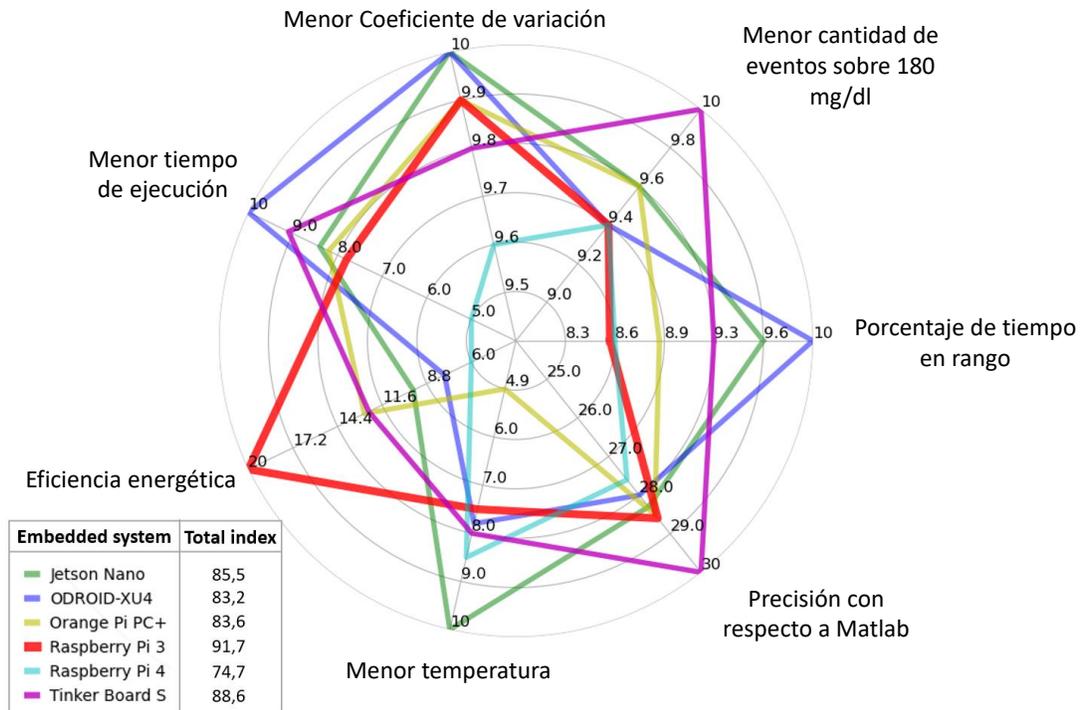
La Figura 4-10.A muestra que se obtiene un coeficiente de variación (CV) similar entre los pacientes con todos los SBC. El CV se mantiene por debajo del 30%; por tanto, se puede considerar que la glucemia no varía significativamente respecto al valor medio de  $213\text{mg/dl}$ . En la Figura 4-10.B se presenta el porcentaje de tiempo en normoglucemia. En general, al

utilizar el paquete quadprog, se obtiene un mayor tiempo en normoglucemia; y al utilizar los sistemas embebidos Jetson Nano, ODRROID-XU4 y Tinker Board S, se obtuvo un tiempo superior al 93 %. Además, en la Figura 4-10.C se muestran el número de eventos por encima de  $180\text{mg/dl}$ , donde se destaca a la Tinker Board S por tener el menor número de eventos hiperglucémicos ( $4.8 \pm 2.4$  eventos). Por último, la Figura 4-10.D muestra el error promedio de cada SBC con respecto a la evolución de la glucemia simulada en Matlab. Aunque este resultado se ve afectado por el ruido de medición, todos los sistemas embebidos tienen una buena aproximación a la referencia (Matlab). La Tinker Board S se destaca con el error promedio más bajo con:  $4,99\text{mg/dl} \pm 4,82\text{mg/dl}$ .



**Figura 4-10:** Resultados con ZMPC-AV-OF, (A) CV, (B) Tiempo en normoglucemia, (C) Eventos sobre 180 mg/dl y (D) Error respecto a MATLAB.

Finalmente, en la Figura 4-11, se muestra la ponderación obtenida por cada sistema embebido en cada criterio evaluado. Para cada criterio, se representa el peso máximo asignado, la sección exterior del gráfico indica un mejor desempeño en el criterio evaluado, por el contrario, la sección interior de la trama de la araña está relacionada con los valores menos atractivos. Por ejemplo, recordemos que el consumo energético tiene un peso máximo de 20, es decir, al sistema embebido con menor consumo energético (o más eficiencia energética) se le asigna un valor de 20, que corresponde a la Raspberry Pi 3, a los otros SBC se les asignan un valor más bajo en este criterio ya que consumen una mayor cantidad de energía. La Raspberry Pi 4 es el embebido con mayor consumo energético y por tanto con menor rendimiento en el criterio energético, obteniendo un valor de 6. La asignación de pesos a cada SBC se realiza de forma similar con el resto de los criterios. Al final, se suman los valores obtenidos por cada sistema embebido para encontrar la mejor opción.



**Figura 4-11:** Ponderación obtenida de los sistemas embebidos en los criterios de desempeño.

El resultado total de la suma muestra que la Raspberry Pi 3 es la mejor opción con una puntuación de 91,7, seguida de la Tinker Board S con 88,6. La Jetson Nano, la Orange Pi PC + y el ODROID-XU4 obtuvieron una puntuación de 85,5, 83,6 y 83,2, respectivamente. El sistema integrado con el peor rendimiento es Raspberry Pi 4. Los valores de los factores de ponderación y el índice de rendimiento final se informan en la **Tabla 4-3**.

**Tabla 4-3:** Evaluación del desempeño general para la selección de los sistemas embebidos.

Criterio	Peso	Jetson Nano	ODROID-XU4	Orange Pi PC+	Raspberry Pi 3	Raspberry Pi 4	Tinker Board S
% Tiempo en rango	10,0	9,6	10,0	8,8	8,5	8,6	9,3
Eventos >180	10,0	9,6	9,4	9,6	9,4	9,4	10,0
CV	10,0	10,0	10,0	9,9	9,9	9,6	9,8
Tiempo de simulación	10,0	8,4	10,0	8,2	7,8	5,0	9,1
Energía consumida	20,0	9,6	7,9	13,6	20,0	6,0	12,5
Temperatura	10,0	10,0	7,8	4,9	7,5	8,5	8,0
Precisión respecto a Matlab	30,0	28,3	28,0	28,5	28,6	27,6	30,0
<b>Total</b>	<b>100,0</b>	<b>85,5</b>	<b>83,2</b>	<b>83,6</b>	<b>91,7</b>	<b>74,7</b>	<b>88,6</b>

### 4.3. Discusión

En cuanto a la prueba de temperatura, la Jetson Nano presentó el menor aumento de temperatura porque viene con un disipador de calor que abarca gran parte de la placa base Figura 3-3, lo que evita el calentamiento del procesador pero lo hace impráctico para un dispositivo portátil debido a su tamaño. Del mismo modo, el ODROID-XU4 tiene un disipador de calor y un ventilador significativamente grande en comparación con el tamaño de la placa. Por el contrario, la alta temperatura obtenida con el Orange Pi pc + sugiere que para su funcionamiento requeriría de un disipador de calor y posiblemente un ventilador.

Aunque la arquitectura en hardware de la Raspberry Pi 4 ha mejorado con respecto a la Raspberry Pi 3 3-5, es el embebido con el mayor consumo energético, aunque esto no se debe a un aumento de potencia en el momento de solucionar el proceso de optimización, este consumo de energía está más relacionado con el tiempo de ejecución del algoritmo. De la Figura 4-4, se puede ver que el tiempo requerido para completar la simulación es sustancialmente más largo cuando se implementa el paquete CVXOPT en la Raspberry Pi 4 que con cualquier otra SBC, lo que resulta en un mayor consumo de energía, como se ha visto en la Figura 4-5. Este comportamiento podría sugerir que no existe una completa compatibilidad entre el sistema operativo, el lenguaje de programación o el paquete de optimización utilizado al momento de realizar la parte experimental de este estudio. Por tanto, se obtiene un rendimiento bastante deficiente con periodos prolongados calculando la dosis óptima de insulina y generando el alto consumo energético registrado.

Se obtuvo una clara diferencia en los tiempos de procesamiento y el consumo de energía al ejecutar las estrategias de control complejas (ZMPC-AV y ZMPC-AV-OF) en comparación con las estrategias más simples (MPC y ZMPC), que es lo que se esperaba. Esta diferencia se debe al mayor costo computacional con las estrategias complejas ya que el problema de optimización a resolver en cada paso de tiempo tiene más restricciones y más variables de decisión; por lo tanto, el problema de optimización a resolver tiene mayores dimensiones (como se evidencia en la Tabla 4-4). Sin embargo, el tiempo por iteración es corto en comparación con el tiempo de muestreo (5 minutos) incluso para la estrategia más compleja, lo que sugiere que no hay ningún impedimento para implementar estas estrategias MPC en tiempo real. Los resultados presentados en [Zavitsanou et al., 2016] son comparables con los obtenidos en este trabajo cuando se usa Raspberry Pi 3 con el paquete de optimización CVXOPT para una estrategia ZMPC en un solo sujeto. El tiempo promedio por iteración se informó como  $50 \text{ ms} \pm 10 \text{ ms}$ , mientras que en este trabajo, el tiempo promedio por iteración fue de  $78 \text{ ms} \pm 17 \text{ ms}$ , sin embargo, el horizonte de predicción utilizado en [Zavitsanou et al., 2016] es mucho más pequeño ( $H_p = 20$ ) lo que reduce significativamente las dimensiones de las matrices y con esto el tiempo de simulación.

**Tabla 4-4:** Complejidad de las estrategias MPC

Estrategia	Variables de decisión	Número de variables de decisión	Número de restricciones de desigualdad	Número de restricciones de igualdad
MPC	$\mathbf{u}, \mathbf{x}$	$H_c + n_x H_p$	$2H_c + 2n_x H_p$	0
ZMPC	$\mathbf{u}, \delta, \mathbf{x}$	$H_c + 1 + n_x H_p$	$2(H_c + 1) + 2n_x H_p$	0
ZMPC-AV	$\mathbf{u}, x_a, u_a, x_t, u_t, \mathbf{x}$	$H_c + 2n_x + 2 + n_x H_p$	$2(H_c + 2n_x + 2) + 2n_x H_p$	$3n_x$
ZMPC-AV-OF	$\mathbf{u}, x_a, u_a, x_t, u_t, \mathbf{x}, \mathbf{d}$	$H_c + 2n_x + 2 + (n_x + n_d)H_p$	$2(H_c + 2n_x + 2) + 2n_x H_p$	$3n_x$

$H_c$ : Horizonte de control,  $H_p$ : Horizonte de predicción,  $n_x$ : Dimensión de los estados,  $n_d$ : Dimensión de las perturbaciones consideradas en el ZMPC-AV-OF.

En cuanto al desempeño de las estrategias de control con cada paquete solucionador, se obtuvo un comportamiento similar entre los paquetes con las estrategias más simples: el MPC y ZMPC. Los resultados de la Figura 4-8 sugieren que el paquete OSQP presenta problemas cuando aumenta la complejidad del problema de optimización (como se muestra para las estrategias ZMPC-AV y ZMPC-AV-OF). Para ambas estrategias de control, los sujetos virtuales no pueden mantenerse dentro de los rangos normales de glucosa y se desvían sustancialmente de los resultados de Matlab. Este bajo rendimiento sugiere una incompatibilidad entre el solucionador OSQP y las estrategias más complejas, en las que se agregan más variables de decisión y más restricciones de igualdad y desigualdad. El algoritmo OSQP no puede encontrar una solución adecuada, incluso cuando las tolerancias se redujeron significativamente. Otros trabajos corroboran esto. Por ejemplo, en [Chakrabarty et al., 2018], donde se implementó una estrategia ZMPC, se prefiere usar el algoritmo de minimización alternante rápido (FAMA) en sistemas de control integrados porque, en general, el algoritmo ADMM presenta limitaciones en los límites complejos, en el número de iteraciones y tamaño de paso. Además, en [Stellato et al., 2018], se menciona que el paquete OSQP presenta un rendimiento sobresaliente (comparable con los solucionadores comerciales) para problemas de pequeño-mediano tamaño .

Una situación similar ocurrió con el paquete CVXOPT. Con este paquete se obtienen resultados satisfactorios al implementar las estrategias sencillas (MPC y ZMPC). Estos buenos resultados también se obtuvieron cuando se implementó este paquete usando una formulación de ZMPC en trabajos como [Zavitsanou et al., 2016, Chakrabarty et al., 2018]. Pero en ninguno de los documentos se probó el rendimiento del paquete CVXOPT con problemas de optimización más complejos. El ZMPC usado en [Zavitsanou et al., 2016, Chakrabarty et al., 2018] tiene  $4H_c + 2(H_p - H_u)$  restricciones de desigualdad, mientras que el ZMPC-AV explicado en el capítulo 2 tiene  $2(H_c + 2n_x + 2) + 2n_x H_p$  restricciones de desigualdad más las  $3n_x$  restricciones de igualdad debido a las variables artificiales de equilibrio, ( $n_x$  se refiere a la dimensión del estado). Cuando se usa este paquete en la estrategia ZMPC-AV-OF, fue posible ajustar el rendimiento modificando las tolerancias para obtener soluciones factibles. Sin embargo, se presentó una sensibilidad muy alta en este ajuste, es decir, se obtuvieron cambios significativos sobre la glucemia para pequeños cambios de las

tolerancias, típicamente con una magnitud de 0.01, y especialmente en la tolerancia relativa del gab. Además, las mismas tolerancias no funcionan bien para todos los pacientes. Estos resultados sugieren que los paquetes OSQP y CVXOPT no son confiables para el tratamiento de la diabetes Tipo 1, donde la robustez y la estabilidad son de primordial importancia. Por lo tanto, ambos paquetes se consideran opciones no válidas para implementar un PA con estrategias de control complejas.

Otro factor que se destaca de los resultados es que una arquitectura compleja y cara no es necesariamente la mejor para todas las aplicaciones. Por ejemplo, la Jetson Nano tiene una GPU diseñada para el procesamiento paralelo, enfocada en el aprendizaje automático y el procesamiento de imágenes digitales. Sin embargo, estos componentes son irrelevantes para la aplicación de este estudio donde el algoritmo y su ejecución son secuenciales. Por tanto, solo se considera el rendimiento del procesador principal, que, si bien presentó buenos resultados, no fue el mejor ni en tiempo de ejecución ni en rendimiento. Así, en este caso, la Jetson Nano no representa una ventaja sino un costo extra para el dispositivo. Por otro lado el ODROID-XU4 presenta un gran consumo de energía instantáneo que se ve compensado por su velocidad de procesamiento y a pesar de tener las mejores especificaciones de procesador, este sistema embebido no presenta una ventaja sustancial en rendimiento sobre otros sistemas embebidos más simples y económicos como la Tinker Board S o la Raspberry Pi 3.

Por otro lado, un sistema con arquitectura de procesador de 32 bits que use (VFPv4) para calcular el punto flotante bajo el estándar IEEE-754 (Doble precisión) permite que la SBC no solo se compare con las otras plataformas de 64 bits, sino también presenta un desempeño sobresaliente en muchos aspectos evaluados en las pruebas. Por ejemplo, la Tinker board S junto con la Raspberry Pi 3, logran tener un buen equilibrio entre la energía consumida, el tiempo de procesamiento y los objetivos de control de esta aplicación.

En cuanto a la transmisión de datos en los resultados obtenidos, no se observaron diferencias significativas en las velocidades de transmisión de datos vía comunicación serial entre cada sistema embebido y el computador.

Los resultados también permiten evaluar el nivel de confianza en los sistemas embebidos al compararlos con el rendimiento en el computador. Existen grandes diferencias entre un computador y un SBC. El computador usa un procesador x86 con la arquitectura CISC para ejecutar el algoritmo mediante comandos. Por el contrario, los sistemas integrados tienen un procesador ARM que usa la arquitectura RISC. Sin embargo, a pesar de estas diferencias, se pudo ver que los resultados obtenidos en el computador (en Matlab) se pueden reproducir en los sistemas embebidos cuando se utiliza el paquete de solucionador adecuado. Por lo tanto, al diseñar y ajustar las estrategias de control en los computadores y luego implementarlas en los sistemas embebidos, existe la confianza de que estos van a ejecutar las estrategias de

forma adecuada.

Al considerar las prioridades para la aplicación T1DM y los resultados en cada uno de los criterios de las pruebas, la Raspberry Pi 3 y la Tinker Board S resultan como las mejores opciones para implementarlo en el PA cuando se utilizan estrategias de control avanzadas. Cabe destacar que, a pesar de que la Tinker Board S y otros sistemas embebidos tienen un mejor desempeño que la Raspberry Pi 3 en tiempos de regulación de glucosa y ejecución de algoritmos, la diferencia entre ellos no es tan significativa como el bajo consumo energético que presenta la Raspberry Pi 3. Este último criterio es la razón por la que la Raspberry Pi 3 se ubica como el más adecuado para implementarlo en el PA.

## 4.4. Resumen

En este capítulo se analizó el desempeño de seis sistemas embebidos y tres paquetes de optimización de código abierto para su implementación en la emulación de un PA utilizando la metodología HIL probando cuatro estrategias MPC. Los resultados muestran que bajo el paquete solucionador adecuado los procesadores de arquitectura ARM cumplen satisfactoriamente los objetivos de control de la aplicación.

El paquete quadprog es el que presenta un mejor rendimiento, en tiempo de ejecución, menor consumo energético, y desempeño en la regulación de glucemia de los pacientes virtuales comparado con los otros dos paquetes. Este paquete demuestra ejecutar de forma fiel los algoritmos de control diseñados en computador dando resultados similares a los obtenidos en Matlab a pesar de la diferencia que existe entre el quadprog de Python y el de Matlab. Con respecto a los paquetes CVXOPT y OSQP debido a los problemas y variaciones que presentaron durante las pruebas los hace inviables para su uso en el tratamiento de la diabetes tipo 1 bajo el esquema de simulación de este trabajo.

# 5 Evaluación de las estrategias de uso eficiente de la energía en controladores MPC para la aplicación del PA

Se considera que el páncreas artificial se implemente en sistemas embebidos que funcionan como dispositivos portátiles con batería de duración limitada y capacidad de cómputo reducida con respecto a un computador (especialmente cuando los algoritmos de estimación y control se vuelven más complejos). Por esta razón, los algoritmos de control deben incorporar características adicionales que extiendan al máximo la vida útil de la batería, permitiéndole al paciente tener mayor tiempo de autonomía y le brinde una mayor holgura en la recarga de la batería. Con este propósito se plantea agregar a la estrategia de control base ZMPC-AV-OF una estrategia de activación de eventos (ET por sus siglas en inglés), para disminuir el número de cálculos de la dosificación de insulina óptima y con esto el consumo energético del dispositivo.

## 5.1. Estrategia de activación de eventos para los recursos energéticos del dispositivo

La idea es cambiar el concepto de activación por tiempo (TT por sus siglas en inglés) en donde el control se ejecuta en cada instante de tiempo, el evento que desata el cálculo en el controlador es el tiempo que transcurre desde un instante  $k$  hasta el otro  $k + 1$ , por el contrario, la estrategia de ET usa un evento externo al tiempo para determinar que es necesario realizar los cálculos. Para esto se incorpora la función de activación de eventos en donde se toma la decisión si el problema de optimización del MPC debe resolverse en un instante de tiempo  $k$  o no; esta estrategia disminuye el consumo del dispositivo debido a que no es necesario resolver el problema de optimización en cada instante de tiempo para lograr los objetivos de control.

En este esquema el instante de tiempo discreto en donde ocurre un evento se denota como  $k_e$  y se asume que el primer evento  $e = 0$  ocurre en el tiempo  $k_0 = 0$ , luego el siguiente evento  $k_{e+1}$  se da cuando la diferencia entre la planta  $y$  y el estado estimado  $\tilde{x}C$  supere el umbral  $\epsilon \geq 0$  en el instante  $k$ , como se muestra en la ecuación 5-1.

$$\|y(k) - \tilde{y}(k)\| \geq \epsilon. \quad (5-1)$$

El estado estimado  $\tilde{y} = \tilde{x}C$  se obtiene del resultado del proceso de optimización en donde la salida  $U$  se calcula a lo largo del horizonte de control  $H_c$ , de esta forma el vector  $U : \{u(k+1), \dots, u(k+H_c)\}$  es usado para encontrar el vector de predicción del estado mediante el modelo de la planta  $\tilde{x}(k+1) = Ax(k) + B_u u(k) + B_r r(k) + E$ , encontrando el vector  $\tilde{x}$  con cada  $\{u(k+1), \dots, u(k+H_c)\}$ . Este vector es el que se compara con el estado actual en cada instante  $k$  para determinar si el  $\Delta y_k = y_k - \tilde{y}_k$  es lo suficientemente grande para resolver la trayectoria óptima del MPC y actualizar el vector  $U$  [Lehmann et al., 2013].

La propuesta del nuevo controlador que implementa la estrategia de eventos de activación se observa en el siguiente algoritmo. En donde el cálculo de la dosificación de insulina óptima del ZMPC-AV-OF se ve condicionado por el umbral en cada instante de tiempo  $k$ , para el tiempo de simulación  $N$ , resolviendo la función de costo para  $x, u, x_a, u_a, x_t, u_t$  y  $d$  solo cuando sea necesario, de lo contrario no se realizan acciones de control y no se inyectan unidades de insulina al paciente virtual.

---

**para**  $k=0$  **hasta**  $k=N$  **con paso** 1 **hacer**  
**si** ( $\|y(k) - \tilde{y}(k)\| \geq \epsilon$  **o**  $k = 0$ ) **entonces**  
 Resuelva:  

$$\min_{\mathbf{u}, \mathbf{x}, x_a, u_a, x_t, u_t, d} \sum_{j=0}^{H_p-1} \|x(j) - x_a\|_Q^2 + \sum_{j=0}^{H_c-1} \|u(j) - u_a\|_R^2 + P(\text{dist}_{\mathcal{X}_s^{\text{Tar}}}(x_a) + \text{dist}_{\mathcal{U}_s^{\text{Tar}}}(u_a))$$
  
*s.t.*  $x(0) = \hat{x}(k), \quad d(0) = \hat{d}(k)$   
 $x(j+1) = Ax(j) + B_u u(j) + B_r r(j) + B_d d(j) + E$   
 $d(j+1) = d(j)$   
 $u(j) \in \mathcal{U}, \quad x(j) \in \mathcal{X}$   
 $x(H_p) = x_a$   
 $x_a = Ax_a + B_u u_a + B_d d + E, \quad y_a = Cx_a + C_d d(j)$   
**si no**  
 $u(k) = 0$   
**fin si**  
**fin para**

---

Desde el punto de vista de la estabilidad de los sistemas lineales impulsivos en lazo cerrado, se define la atractividad débil solo en los instantes impulsivos, que en este caso sería en los eventos  $k_e$ , dejando entre los intervalos que el sistema se mueva de forma libre, dando como resultado la estabilidad asintótica débil del controlador base ZMPC-AV-OF [Rivadeneira et al., 2018]. Sin embargo, hay que tener en cuenta que la magnitud del umbral

de activación, el horizonte de control y las perturbaciones pueden afectar la factibilidad del problema a resolver en el MPC, de esta misma forma estas variables afectan el desempeño del control y el costo computacional. Una alta perturbación (gran ingesta de carbohidratos) normalmente resulta en un peor desempeño del controlador, al aumentar la magnitud del umbral se reduce la frecuencia del cálculo de insulina óptima que al mismo tiempo reduce la carga computacional disminuyendo la energía consumida. Por otro lado, en general se tiene un mejor desempeño en el controlador cuando el horizonte de predicción aumenta, esto permitiría aumentar el nivel del umbral de activación teniendo menos carga computacional, pero esto puede disminuir la precisión del control. En resumen, se deben equilibrar estos factores porque al aumentar el umbral de activación se reduce la carga computacional pero el desempeño del controlador disminuye [Li and Shi, 2014].

### 5.1.1. Configuración del sistema embebido y periféricos

Se implementa la SBC Raspberry Pi 3 B por sus ventajas frente a los otros embebidos resaltando el bajo consumo energético presentado en las pruebas de rendimiento. El sistema de almacenamiento usado es una memoria microSD de 32GB clase 10. Por los puertos GPIO se configura la comunicación serial (RS232, Bd, 115200, sin paridad, bit de parada = 1, tiempo de terminación 90) y los puertos de propósito general para controlar el motor paso a paso. Por último, para la comunicación inalámbrica se usa el Bluetooth de bajo consumo BLE v4.2.

### 5.1.2. Batería

El PA se debe considerar como un dispositivo portátil que permita mejorar la calidad de vida de las personas, por esta razón se debe de integrar una batería en el esquema de emulación que permita visualizar las ventajas, el rendimiento, la autonomía y las limitaciones que tienen este tipo de sistemas embebidos y sus periféricos en la aplicación. Para esto se usa la batería de polímero de litio Pisugar2 Pro. Este sistema cuenta con una batería de litio de  $5000mAh/18,5Wh$ , a  $3,7V$  Figura 5-1, con un voltaje de carga límite de  $4,2V$  y un voltaje de descarga de  $3,2V$ , además contiene el gestor de batería IP5312 que tiene una eficiencia de carga del 97%, una precisión del voltaje de carga de  $\pm 0,5$ , contiene un conversor análogo digital de 14 bits para las mediciones de voltaje y corriente, se puede personalizar la curva de energía de la batería, tiene protección contra cortocircuitos (sobrevoltaje y sobre corriente), maneja una interfaz I2C con dirección 0x75 y un reloj de tiempo real (RTC por sus siglas en inglés) DS3078 con dirección 0x32.



**Figura 5-1:** Gestor y batería Pisugar 2 pro

La caracterización de la batería se hace de acuerdo con la programación sugerida por el fabricante, del algoritmo V obtiene el valor actual de voltaje en la batería como un entero multiplicado por mil y con base en este se usa el algoritmo B para obtener el nivel en porcentaje actual de carga. Para los rangos de porcentaje de carga se usan los valores de la tabla 5-1.

---

**Algoritmo V**

```

low = bus.read(0x75, 0xd0)
low = bus.read(0x75, 0xd0)
si (high & 0x20): entonces
    low = ~ low & 0xFF
    high = (~ high) & 0x1F
    v = -((high & 0x1F) * 256 + low + 1) * 0,26855 + 2600
si no
    v = ((high & 0x1F) * 256 + low + 1) * 0,26855 + 2600
fin si
BatV = v
    
```

---



---

**Algoritmo B**

```

batter_level = 0
para range in batter_curve : hacer
    si range[0] < BatV/1000 <= range[1] : entonces
        level_base = ((BatV/1000 - range[0]) / (range[1] - range[0])) * (range[3] - range[2])
    fin si
fin para
batter_level = level_base + range[2]
devolver batter_level
    
```

---

**Tabla 5-1:** Rangos de carga/descarga de la batería.

V_inferior range 0	V_superior range 1	%_inferior range 2	%_superior range 3
4.16	5.0	100	100
4.05	4.16	95.0	100
3.90	4.05	88.0	95.0
3.80	3.90	77.0	88.0
3.70	3.80	65.0	77.0
3.62	3.70	55.0	65.0
3.58	3.62	49.0	55.0
3.49	3.58	25.6	49.0
3.32	3.49	4.5	25.6
3.1	3.32	0	4.5
0	3.1	0	0

El modelo de la batería es una 955465 de Shenzhen Wonzen Technology Co.Ltd con dimensiones  $65mm \times 54mm \times 9,5mm$ , los resultados de las pruebas de la batería para la pérdida de masa, cortocircuito externo y temperaturas se pueden encontrar en [EMC, 2020, DougGuan, 2020]. Las condiciones de seguridad y la operación estable de las baterías se hacen especialmente relevantes cuando se trata de una aplicación que involucra tener un constante contacto con el cuerpo humano.

### 5.1.3. Actuador

Uno de los principales objetivos en este trabajo es determinar qué tan viable es la implementación de las estrategias de control no tradicionales basadas en MPC en sistemas embebidos, por lo que es imperativo la implementación del actuador de la bomba de insulina para considerar su efecto sobre el consumo de energía y en los tiempos de aplicación de la dosis de insulina. Se implementa el micromotor paso a paso GM24FM08BY en forma de U desarrollado por “Shenzen Shunli Motor Co. LTD”, ver Figura 5-2, este es un mecanismo rotacional que transmite su potencia mecánica a través de una caja reductora con acoples piñón-engranaje que finalmente llega a un eje roscado encargado de realizar el movimiento de salida de la insulina.

Este motor bipolar de 20 pasos por revolución tiene un torque de arrastre de  $68.6 \text{ [mN x M]}$ , una velocidad nominal de  $3600 \text{ [RPM]}$ , un paso de eje de  $0.6mm$ , una resistencia en los devanados de  $20 \Omega \pm 10 \%$ , funciona a  $5V$  y maneja una corriente de  $300 \text{ mA}$  por fase. Teóricamente cuenta con una caja reductora de 6 piñones y 6 engranajes con una relación

1:500, pero en [Lopez-Valencia, 2021], se obtiene una relación aproximada de 1:497 con lo que se obtiene un menor error en el suministro de insulina 0.298[UI - U100]. Bajo estas condiciones se requieren de 1988.0716 Pasos/U, en donde cada unidad de insulina es suministrada en 13.9722 segundos. Las cantidades de insulina calculadas en cada instante de tiempo  $k$  son redondeadas a un mínimo de variación de 0.1 debido a que las pequeñas magnitudes en la inyección de insulina son difíciles de ejecutar para el mecanismo de inyección.

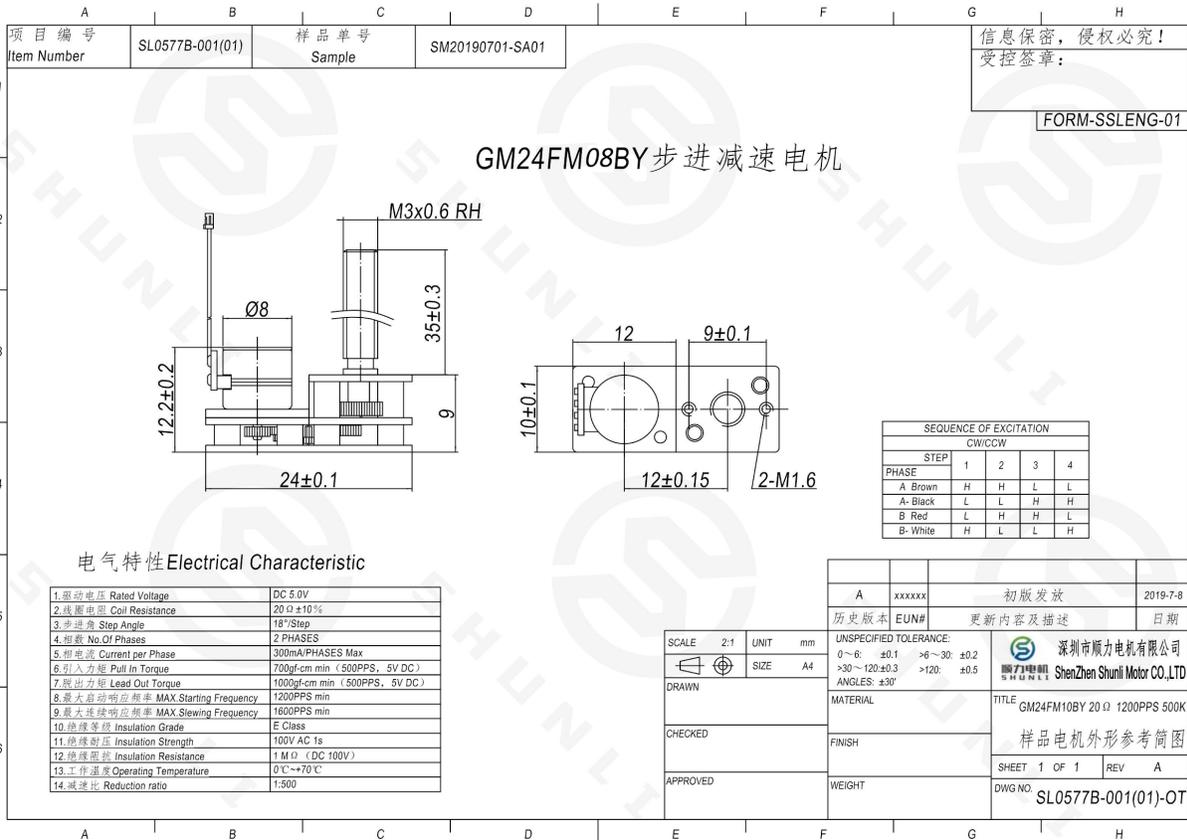


Figura 5-2: Dimensiones del micromotor paso a paso en [mm], con escala 2:1

Para el control del motor se usa el controlador para motores paso a paso TB6612FNG de la marca pololu que implementa un puente H con transistores tipo MOSFET. Puede trabajar con un voltaje en el motor de 5V a una corriente de 1,2A. Incorpora un circuito de apagado térmico, capacitores para las líneas de alimentación y protección de potencia inversa en la alimentación del motor, el esquema de conexión entre el motor y el controlador se observa en la Figura 5-3.

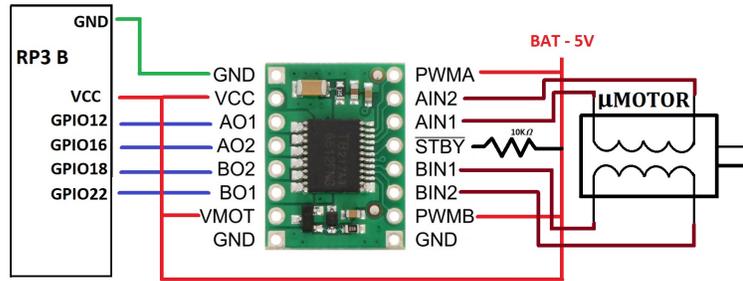


Figura 5-3: Conexión del driver para motor paso a paso TB6612FNG

#### 5.1.4. Escenario de simulación y métricas

Para esta última etapa se aplican dos esquemas para las pruebas, ambos basados en HIL. Se sigue teniendo una simulación en tiempo real entre una computadora (Core i9-9880H, con 16 GB de RAM, SO Windows 10, versión de Matlab R2020a) como la representación virtual de la planta (paciente virtual con T1DM) y una versión real del controlador, que en este estudio corresponde al sistema embebido que contiene las estrategias MPC y el estimador [MathWorks, 2021a]. En la Figura 5-4, se muestra el esquema para las pruebas poblacionales en donde se obtienen las estadísticas del desempeño en la regulación de los 10 pacientes virtuales y del número de controles realizados en cada simulación. Por otro lado en la Figura 5-5, se plantean pruebas en donde se implementan el actuador, la batería y se ejecutan las instrucciones en tiempo real, este esquema se centra en el consumo energético y la comparación del controlador basado en TT y el de ET.

Para ambos esquemas se tienen en cuenta las siguientes consideraciones: la representación matemática del paciente con T1DM se implementa mediante la ecuación (2-7) usando un computador con el software Matlab. Varios estudios previos han demostrado que bajo las consideraciones apropiadas este modelo es válido para la emulación de pacientes con T1DM ([Villa-Tamayo and Rivadeneira, 2020a, Villa-Tamayo et al., 2021a]). El modelo (2-7) está personalizado para 10 pacientes adultos cuyos parámetros se han identificado para ajustarse a los datos de la versión comercial del simulador UVA/Padova (2014). La simulación tiene una duración de 3 días en los cuales se hacen variaciones paramétricas de hasta el 30 % sobre los valores nominales para inducir casos de hipoglucemia en el primer día e inducir casos de hiperglucemia en el tercer día. Durante la simulación cuatro diferentes cantidades de carbohidratos son suministrados por día a los pacientes con una duración de 15 minutos cada una. Las comidas de magnitud 55, 20, 90, y 70g se suministran a las 7, 10, 13, y 19 horas. La información de la ingesta de carbohidratos es suministrada al controlador en algunos escenarios y en otros no.

Los parámetros de control utilizados para las pruebas en este trabajo son: un horizonte de predicción  $H_p = 80$ , un horizonte de control  $H_c = 10$ , un tiempo de muestreo  $T_s = 5$  y el

estado inicial dado por  $X_0 = [115, x_{2ss}, x_{3ss}, 0, 0]'$ , en donde  $x_{2ss}$  y  $x_{3ss}$  se refieren al estado de equilibrio del modelo (2-7) dado por  $x_{2ss} = x_{3ss} = (\theta_1 - 115\theta_2)/\theta_3$ . La zona objetivo se estableció en  $[85 - 105]$  mg/dl. Las matrices de peso  $Q$ ,  $R$ ,  $P$ ,  $Q_e$ ,  $R_e$  y  $Q_w$  correspondientes al control y el estimador se ajustaron en Python bajo el esquema de simulación A, tomando como referencia los valores usados para el ZMPC-AV-OF.

### Simulación bajo el esquema A

Este esquema de simulación se presenta para facilitar el proceso de sintonización de las matrices de peso del controlador, el estimador y el épsilon usando la estrategia ET. El computador que hace las veces de paciente virtual contiene el modelo de paciente con T1DM que envía la información de la glucemia actual después de pasar por el modelo de ruido del CGM, esta información es enviada por comunicación serial mediante el protocolo (RS232) hacia el controlador físico. En este se ejecuta el algoritmo del filtro de Kalman y con base en la glucemia actual obtenida se compara y se determina si es necesario llevar a cabo el cálculo de la dosificación de insulina óptima mediante el algoritmo del MPC usando el paquete quadprog para luego enviar las unidades de insulina de nuevo a la planta por (RS232); si por el contrario el paciente se encuentra dentro del rango permisible de variación no se realizan los cálculos y la insulina aplicada es igual a cero.

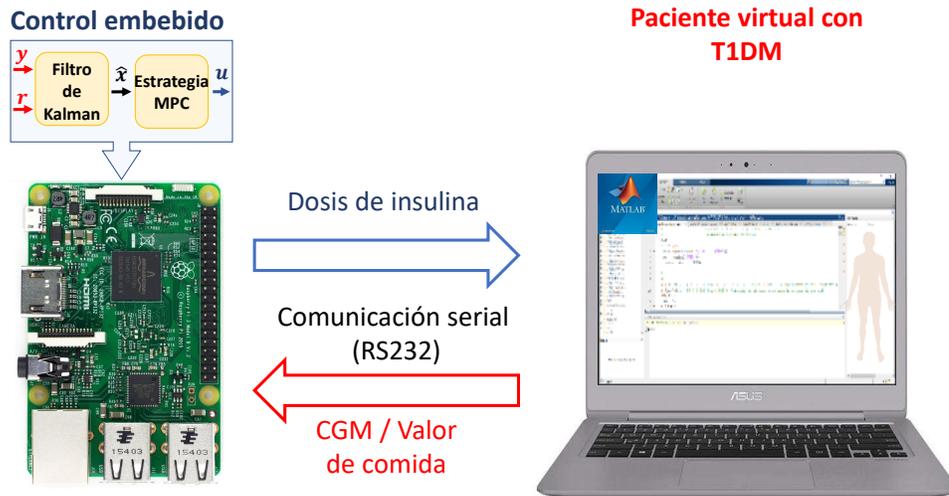


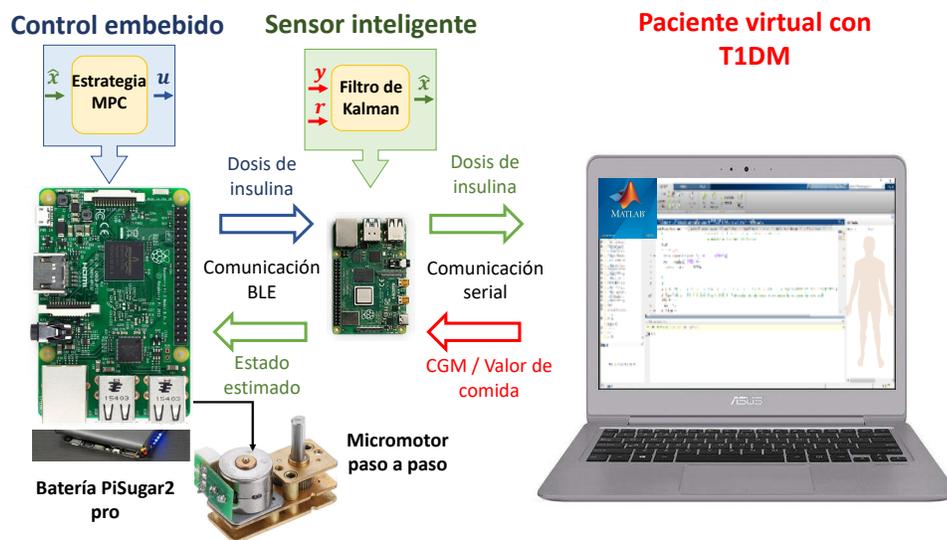
Figura 5-4: Esquemas de simulación A. Elaboración propia a partir de imágenes CC-BY-SA

### Simulación bajo el esquema B

Bajo este esquema se implementa el diagrama de flujo de la Figura 5-6, este esquema involucra elementos como la batería y el motor de la bomba que permiten tener un acercamiento más realista a lo que sería una prueba del PA. Se implementa lo que se denomina como un

sensor inteligente, este sensor recibe la medición actual de glucemia del paciente mediante el protocolo serial (RS232) con el computador, procesa la información con el algoritmo del filtro de Kalman para obtener el estado estimado actual del paciente y luego se usa la estrategia ET para determinar si es necesario el cálculo de la dosificación de insulina o no.

En caso de que el paciente se encuentre dentro del rango admisible de glucosa en sangre no se realiza el cálculo y no se inyectan unidades de insulina al paciente virtual, luego de esto el sensor permanece inactivo hasta que pasen 5 minutos, se realice una nueva medida y se vuelva a tomar la decisión, en caso de que los niveles de glucemia sobrepasen el error de tolerancia épsilon, el sensor envía una señal para que el embebido que contiene la estrategia de control se energice, el sensor espera a que el controlador cargue sus funciones principales y ejecute de forma automática el algoritmo de control, iniciando por un ciclo infinito en donde se espera a tener una conexión inalámbrica exitosa por medio de Bluetooth, cuando se establece la conexión el sensor inteligente envía la información del estado actual del paciente al controlador, este realiza el cálculo de la dosificación de insulina óptima usando el paquete quadprog, se llama a la función que determina y ejecuta la cantidad de pasos del motor por unidad de insulina calculada, luego envía esta información al sensor nuevamente por Bluetooth antes de guardar los registros, las variables y apagarse. Una vez el sensor recibe la información de la dosificación de insulina envía esta información al paciente virtual por medio de la comunicación serial, repitiendo el ciclo nuevamente.



**Figura 5-5:** Esquemas de simulación B. Elaboración propia a partir de imágenes CC-BY-SA

La finalidad de las emulaciones bajo este esquema es determinar el desempeño real del sistema cuando se implementa una batería y se tienen en cuenta el consumo de la dosificación de insulina, en donde se involucran el gasto energético del embebido al realizar los cálculos y el del motor al ejecutar los pasos necesarios para la administración de insulina.

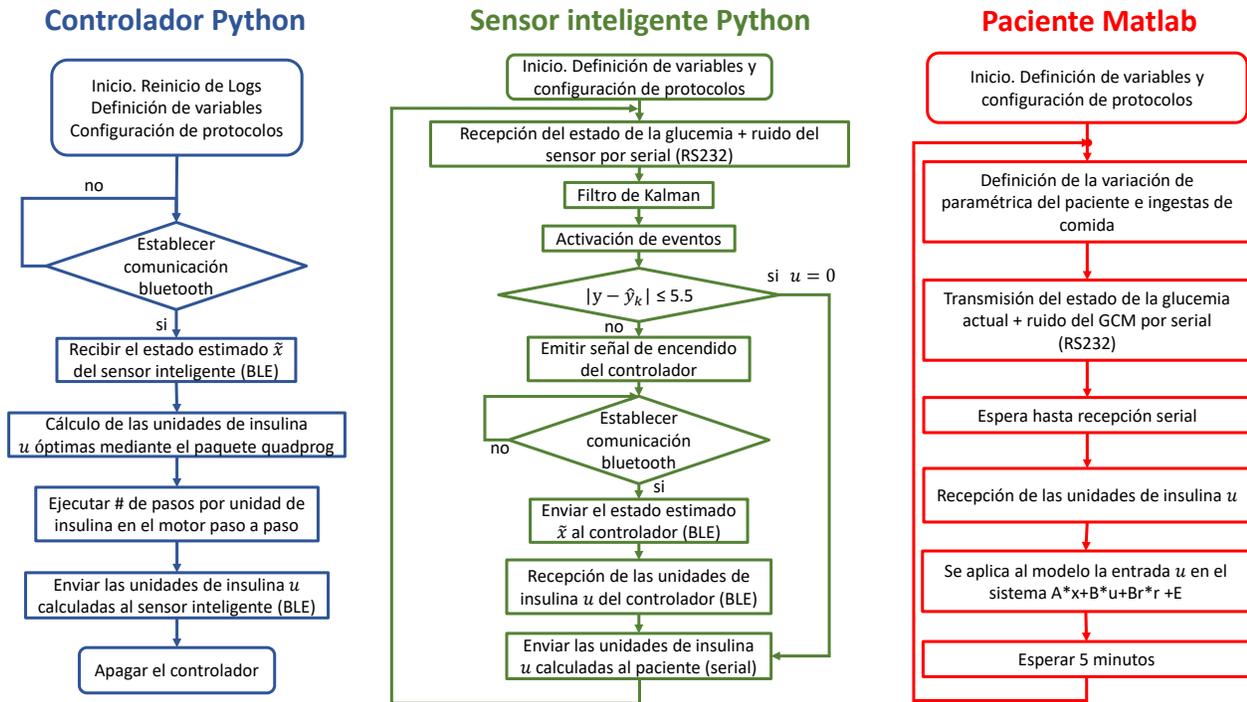


Figura 5-6: Diagrama de flujo para el esquema de simulación B

### 5.1.5. Efectos en la reducción de cálculos del controlador

Durante las pruebas anteriores se ha mantenido constante el tiempo de muestreo  $T_s = 5$  minutos y con esto la frecuencia con la que se calcula la dosificación de insulina. En la estrategia ET se obtiene una frecuencia de control que varía conforme el estado estimado discrepa del estado real del paciente. Para observar la dinámica de la glucemia y el comportamiento del controlador base ZMPC-AV-OF se realiza una prueba en donde se usa un esquema básico de simulación y se incrementa el tiempo de muestreo en 10, 15, 20 y 30 minutos sin modificar las matrices de sintonía del controlador base. En la Figura 5-7 se observa en las franjas verdes la zona objetivo (85-105 mg/dl) que sirve como referencia para identificar que tanto se desvían los controladores de la zona deseada conforme aumenta el tiempo de muestreo, notando la relación del aumento del  $T_s$  con el aumento en los niveles máximos y promedios de glucemia. En la Figura 5-8 se observa la diferencia en magnitudes y en frecuencia de inyecciones de insulina que genera el cambio del tiempo de muestreo.

De estas dos figuras se puede evidenciar el efecto que tiene el aumento del tiempo de muestreo y con esto la respuesta del controlador, al aumentar el  $T_s$  el controlador recibe información del estado actual del paciente con mucha menos frecuencia, retardando su respuesta y con esto prolongando y elevando los niveles de glucemia del paciente, este efecto es más notable cuando se generan eventos de ingesta de carbohidratos. El controlador para compensar la elevación de los niveles de glucemia inyecta mayores cantidades de insulina con menor frecuencia, este

resultado es el que se espera obtener cuando se implemente la estrategia de ET pero teniendo en cuenta que en esta prueba se tienen tiempos prolongados en el que el controlador no obtiene información del sistema de ninguna forma y no es posible responder adecuadamente ante perturbaciones de pequeñas o grandes magnitudes, por el contrario con la estrategia de ET si se cuenta con un monitoreo continuo precisamente para detectar el momento en el que el controlador debe realizar el cálculo y evitar que las condiciones del paciente se agraven.

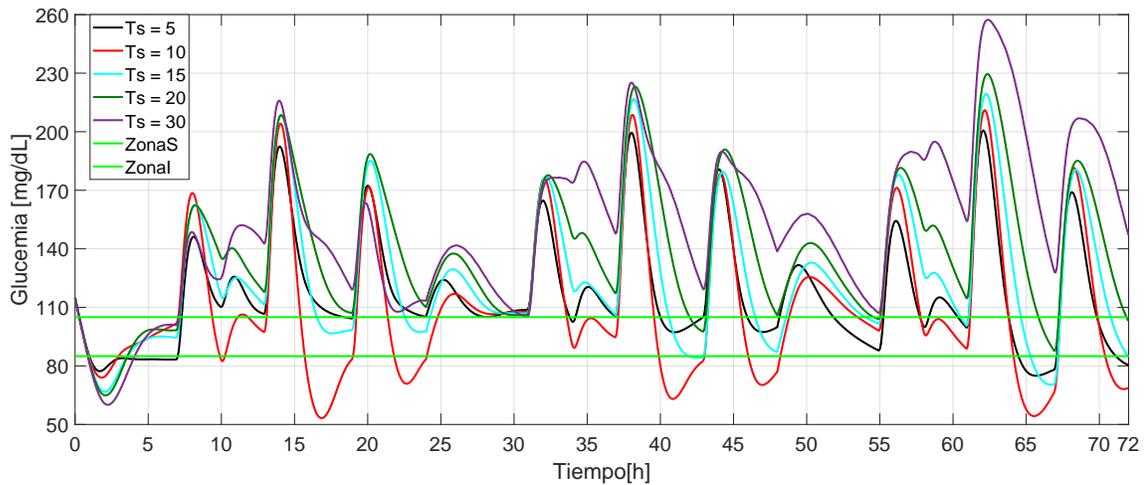


Figura 5-7: Dinámica de la glucemia con variaciones en el tiempo de muestreo  $T_s$

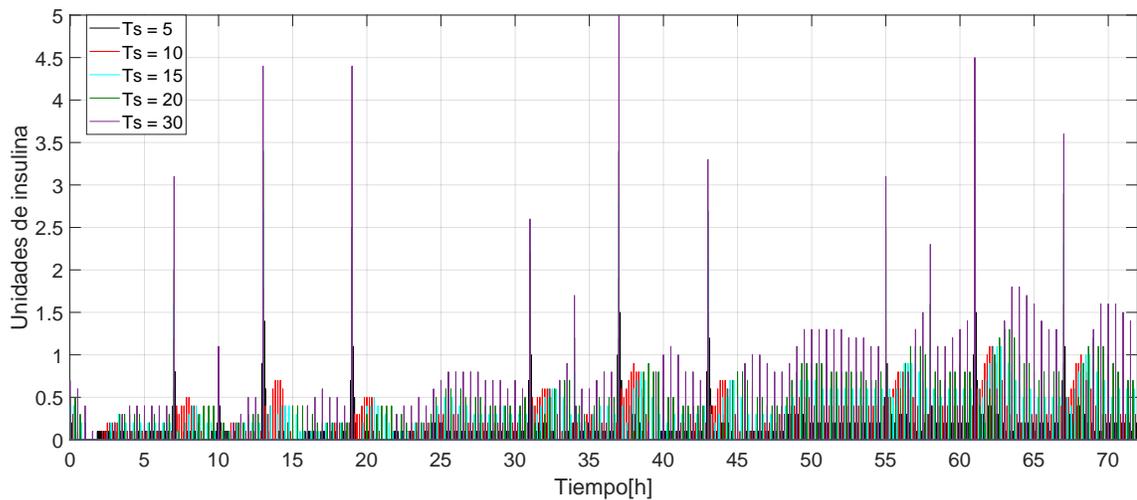


Figura 5-8: Inyección de insulina con variaciones en el tiempo de muestreo  $T_s$

Con la introducción de la estrategia de ET se evidencia la necesidad de modificar las matrices de sintonía con el fin de que respondan adecuadamente a los cambios en la glucemia del paciente que son significativamente mayores a los que se notan cuando se tiene un tiempo fijo para la dosificación. En la Tabla 5-2 se muestra la sintonización para cada paciente, estos nuevos valores comparados con los de la Tabla 4-2 que pertenecen al controlador base son menores en la matriz de sintonía  $Q$ , indicando que el controlador debe de reducir su

“agresividad” al momento de tratar de estabilizar a los pacientes luego de pasar un tiempo mayor al  $T_s$  sin conocer el estado actual de glucemia.

**Tabla 5-2:** Parámetros de sintonía de ET con anuncio y variaciones del 10% por paciente.

AV-OF-ET	Q	R	Qe	Re	$\gamma$	Gw
Pac. 1	$13x10^2$	$10x10^6$	[10, 1, 1, 1, 200, 1]	$65x10^8$	$10x10^4$	[10, 10, 10, 10, 10, 10]
Pac. 2	$3,5x10^1$	$80x10^3$	[10, 1, 1, 1, 400, 1]	$98x10^8$	$10x10^4$	[100, 10, 10, 10, 10, 10]
Pac. 3	$2,0x10^1$	$19x10^3$	[10, 1, 1, 1, 300, 5]	$98x10^8$	$10x10^4$	[100, 10, 10, 10, 10, 10]
Pac. 4	$1,4x10^1$	$145x10^2$	[10, 1, 1, 1, 300, 5]	$96x10^8$	$10x10^4$	[10, 10, 10, 10, 100, 10]
Pac. 5	$1,3x10^1$	$18x10^3$	[10, 1, 1, 1, 400, 1]	$192x10^8$	$10x10^4$	[100, 10, 10, 10, 100, 10]
Pac. 6	$23,3x10^1$	$7x10^2$	[100, 1, 1, 1, 200, 5]	$809x10^8$	$10x10^4$	[200, 10, 10, 10, 300, 100]
Pac. 7	$6x10^1$	$50x10^3$	[200, 10, 10, 10, 100, 80]	$70x10^8$	$10x10^4$	[100, 10, 10, 10, 110, 1]
Pac. 8	$12x10^1$	$70x10^3$	[10, 1, 1, 1, 200, 90]	$80x10^8$	$10x10^4$	[50, 10, 10, 10, 200, 1]
Pac. 9	$30x10^1$	$16x10^6$	[10, 1, 1, 1, 20, 50]	$798x10^8$	$10x10^4$	[10, 10, 10, 10, 100, 1]
Pac. 10	$15x10^1$	$80x10^2$	[100, 1, 1, 1, 200, 5]	$98x10^8$	$10x10^4$	[100, 10, 10, 10, 100, 1]

\* Pac. = Paciente

Posteriormente se realizarán pruebas en donde se introducen variaciones paramétricas del 20 y 30%, además de trabajar con y sin anuncio de comida. Para este conjunto de variaciones se realiza una última sintonización, los valores de las matrices de sintonía para el ZMPC-AV-OF con ET se observan en la Tabla 5-3, en donde se nota que las magnitudes de las matrices  $Q$  y  $R$  se ven reducidas comparadas con el controlador base, además de modificar de forma drástica y personalizada a la matriz de pesos del estimador  $Gw$  que contiene la información del error planta-modelo.

**Tabla 5-3:** Parámetros de sintonía de ET con y sin anuncio de comida y variaciones del 20% / 30% por paciente.

AV-OF-ET	Q	R	Qe	Re	$\gamma$	Gw
Pac. 1	$0,3x10^2$	$20,5x10^3$	[20, 1, 0,1, 0,1, 180, 1]	$20x10^8$	$10x10^4$	[10, 1, 0,1, 0,2, 300, 4,3]
Pac. 2	$0,4x10^2$	$20x10^3$	[20, 1, 0,1, 0,1, 190, 1]	$18x10^8$	$10x10^4$	[10, 1, 0,1, 0,1, 300, 4,0]
Pac. 3	$90x10^2$	$28x10^3$	[100, 1, 0,1, 0,1, 380, 1]	$28x10^8$	$10x10^4$	[10, 1, 0,1, 0,1, 300, 4,6]
Pac. 4	$0,3x10^2$	$22,3x10^2$	[20, 1, 0,1, 0,1, 180, 1]	$80x10^8$	$10x10^4$	[10, 1, 0,1, 0,2, 300, 4,2]
Pac. 5	$1,3x10^1$	$18x10^3$	[10, 1, 1, 1, 400, 1]	$7x10^8$	$10x10^3$	[100, 10, 10, 10, 100, 2,5]
Pac. 6	$2,22x10^2$	$20x10^3$	[10, 1, 0,1, 0,1, 200, 1]	$88x10^8$	$10x10^4$	[20, 0,1, 0,1, 0,1, 250, 4,4]
Pac. 7	$5,5x10^2$	$10x10^3$	[10, 1, 0,1, 0,1, 100, 1]	$17x10^8$	$10x10^3$	[10, 10, 10, 10, 300, 3,9]
Pac. 8	$5,5x10^2$	$900x10^3$	[10, 1, 1, 1, 100, 1]	$80x10^8$	$10x10^3$	[10, 10, 10, 10, 300, 3,9]
Pac. 9	$1,8x10^2$	$25,3x10^3$	[20, 1, 0,1, 0,1, 180, 1]	$798x10^8$	$10x10^3$	[10, 10, 10, 10, 300, 4,3]
Pac. 10	$1,5x10^2$	$30x10^3$	[10, 1, 1, 1, 200, 1]	$98x10^8$	$10x10^4$	[10, 10, 10, 10, 100, 3,9]

\* Pac. = Paciente

## 5.2. Resultados escenario A

### 5.2.1. Efectos de la variación del umbral

Con el fin de determinar el efecto en la dinámica de la glucemia de los pacientes, las magnitudes, la frecuencia en las inyecciones de insulina y el consumo de energía se plantea una prueba inicial bajo el esquema de simulación de la Figura 5-4, en donde se realizan variaciones en el umbral  $\epsilon$ , pero tanto la sintonización como los parámetros del horizonte de predicción, horizonte de control entre otros parámetros permanecen igual que en el controlador base (ZMPC-AV-OF). Para esta prueba se realiza una simulación por paciente y se presenta el promedio de los valores presentados en los 10 pacientes virtuales. El resultado parcial del consumo energético consumido por el sistema embebido es directamente proporcional al tiempo que tarda el sistema en terminar la simulación, en la Figura 5-9 se observa una disminución en la cantidad de energía consumida cuando se implementa un umbral  $> 0$ , debido a que se realiza con menor frecuencia el cálculo de la dosificación de insulina óptima reduciendo así el tiempo de simulación.

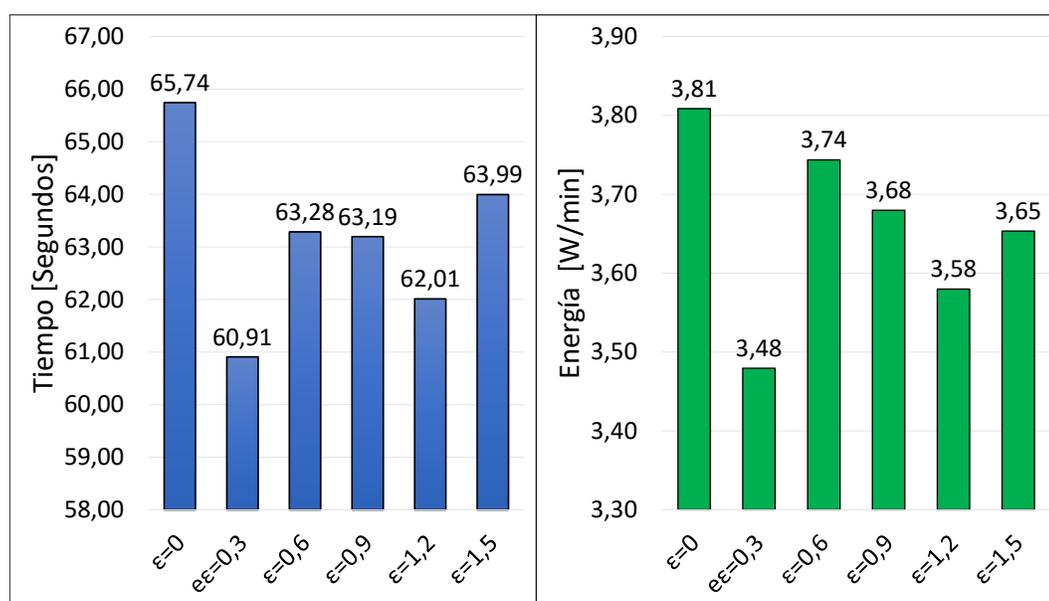


Figura 5-9: Tiempo de simulación y energía consumida por  $\epsilon$

En la Figura 5-10, se evidencia una clara tendencia en la disminución del número de veces que se calcula la dosificación de insulina óptima conforme se incrementa el valor del  $\epsilon$ , pasando de 864 valores calculados con el TT a un promedio de  $581,6 \pm 26,4$  con el ET usando un umbral de 1,5. También se muestra para el caso del paciente 1, que se presenta una menor cantidad de cálculos en el día 1, acorde con la variación paramétrica que induce un caso de hipoglucemia, del mismo modo en el tercer día el promedio de cálculos se incrementa para tratar de mitigar el efecto de la inducción de un estado de hiperglucemia.

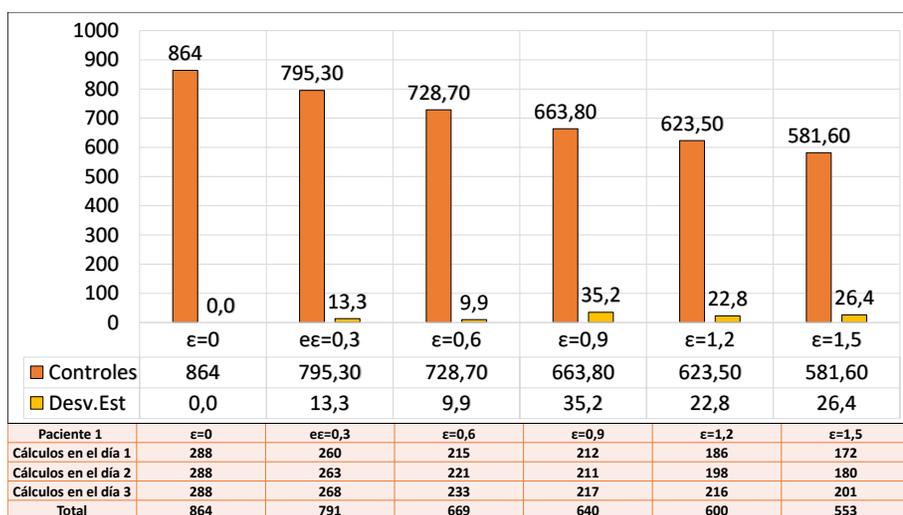


Figura 5-10: Cantidad de controles calculados por  $\epsilon$

Al mantener las matrices de sintonía constantes se encuentra que a medida que se incrementa la magnitud del umbral se obtiene un decremento en el desempeño del controlador, en la Figura 5-11 C, se observa el aumento en los casos de hiperglucemia ( $glucemia > 180m/dl$ ), llegando hasta un promedio de  $10,8 \pm 4,3$  casos en los 10 pacientes, a su vez se nota el decremento de porcentaje de tiempo que pasan los pacientes en un estado estable de normoglucemia 5-11 B. Acorde con lo anterior los niveles promedio de glucemia aumentan pasando de estar en  $113,83mg/dl$  a  $127,141mg/dl$  5-11 D. La única métrica con respecto al desempeño que presenta un mejor resultado es el coeficiente de variación, presentando valores promedios de hasta  $25,53 \pm 5,02$  con un  $\epsilon = 0,9$  5-11 A.

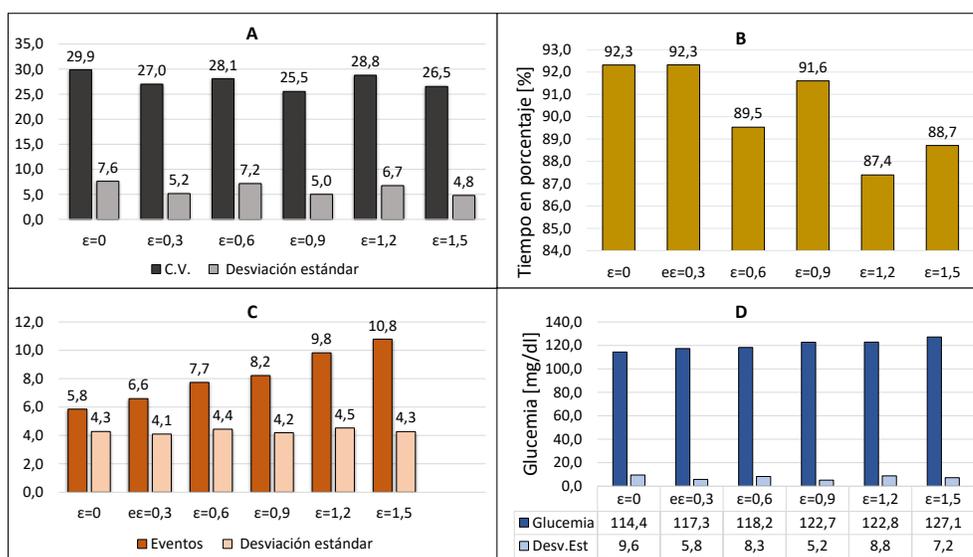


Figura 5-11: Resultados promedio por  $\epsilon$ , (A) CV, (B) Tiempo en normoglucemia, (C) Casos sobre 180 mg/dl y (D) Glucemia promedio.

La dinámica promedio de los 10 pacientes virtuales en cada umbral se muestra en la Figura 5-12. Coherente con las estadísticas de la Figura 5-11, se muestra la dinámica de los pacientes

y como de forma progresiva con el incremento del umbral de la estrategia ET se aumentan los nivel de glucemia, sobre todo en el estado estacionario en donde no se tienen ingesta de carbohidratos. Por último, se realiza una comparación entre la estrategia TT y la ET con un umbral  $\epsilon = 5,5$ , manteniendo los mismos valores en la sintonización. De la Figura 5-13 se observa la pérdida de desempeño al incrementar el umbral y con esto los tiempos en los que el controlador no calcula ni inyecta las unidades de insulina al paciente virtual, pasando de 864 controles con el controlador base a 275 con el ET. En la Figura 5-14 observan los intervalos entre inyecciones de insulina menos frecuentes pero con magnitudes mayores cuando se implementa la estrategia de ET, esto es lo que se esperaba al probar la dinámica y el desempeño del controlador base cuando se da un incremento en el tiempo de muestreo  $T_s$ ; mayores magnitudes de insulina en intervalos de tiempo prolongados.

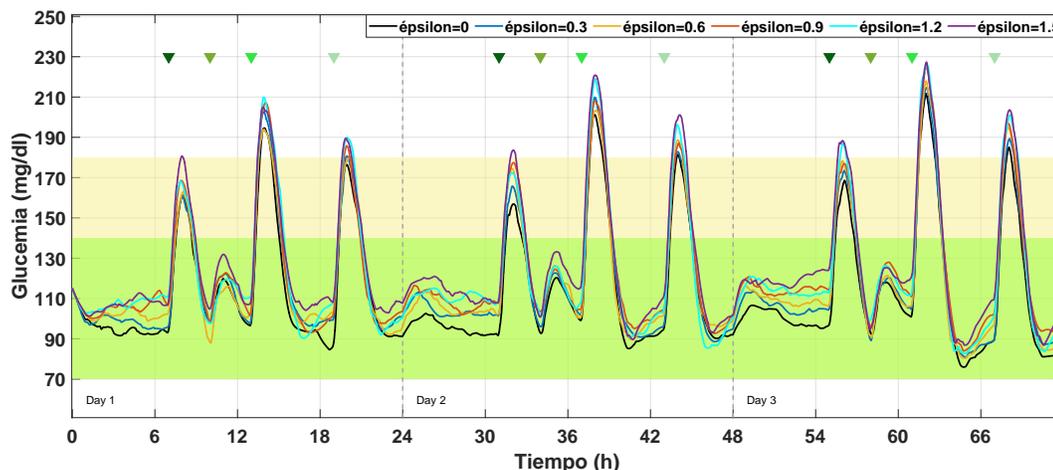


Figura 5-12: Glucemia promedio de los diez pacientes por  $\epsilon$

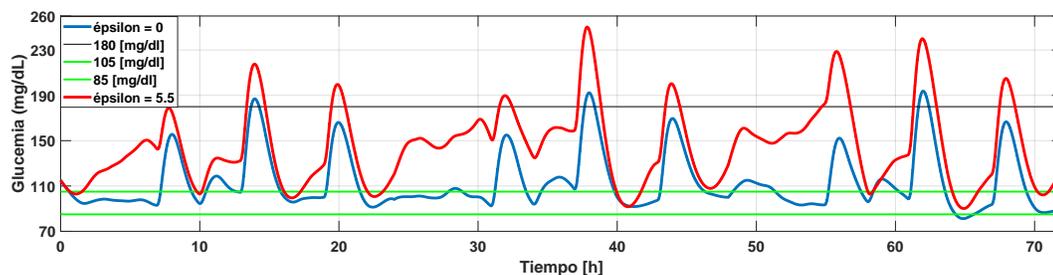


Figura 5-13: Comparación en la dinámica de la glucemia entre el TT y el ET

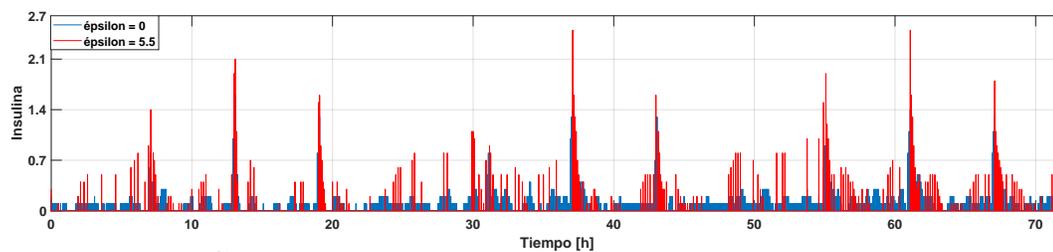


Figura 5-14: Comparación en la dinámica de la insulina entre el TT y el ET

### 5.2.2. Cambio de sintonización con la estrategia ET

Los valores de las matrices de sintonía se modifican para evitar el efecto que tiene la estrategia ET en el incremento en los valores promedios de glucemia implementando los valores de la Tabla 5-2. En las pruebas se mantiene el esquema de simulación de la Figura 5-4, en donde la información del consumo de carbohidratos es suministrada al controlador y las variaciones paramétricas que inducen los casos de hipoglucemia e hiperglucemia son del 10%. Se realizan 10 pruebas por cada paciente con un umbral de  $\epsilon = 5,5$ , en la Figura 5-15 se observa la cantidad promedio de cálculos por paciente que realiza el controlador al modificar las matrices de sintonía y manejando un umbral de 5,5, los pacientes que presentan una menor cantidad de cálculos son el paciente 1 y el 10 teniendo cálculos promedios de  $235,8 \pm 12,6$  y  $208,5 \pm 8,1$  respectivamente, por el contrario los pacientes que presentan la mayor cantidad de cálculos por simulación son el 7 y el 8 con un promedio de  $332,6 \pm 10,7$  y  $312,1 \pm 8,4$ . La cantidad de cálculos promedios por paciente en el primer día es de 86.15, para el segundo día es de 88.82, para el tercer día de 92 y en los tres días se presentan  $266,98 \pm 36,27$  cálculos promedio en el total de las simulaciones. Este cambio representa aproximadamente una reducción del 69% del número de cálculos efectuados por el controlador comparado con la estrategia base que realiza 864 controles en los tres días de simulación.

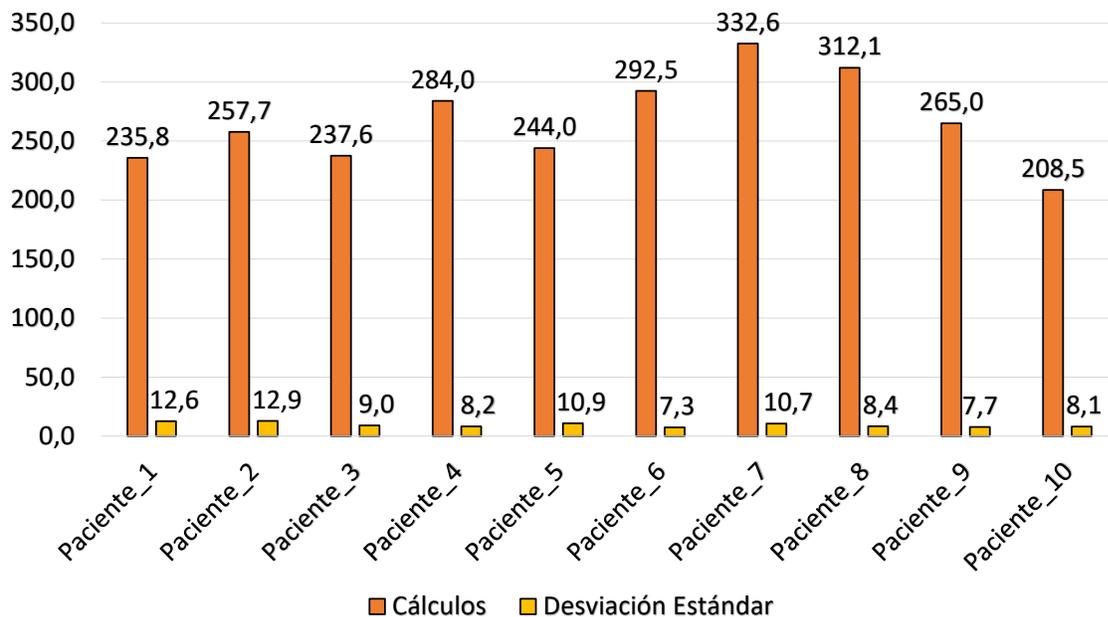
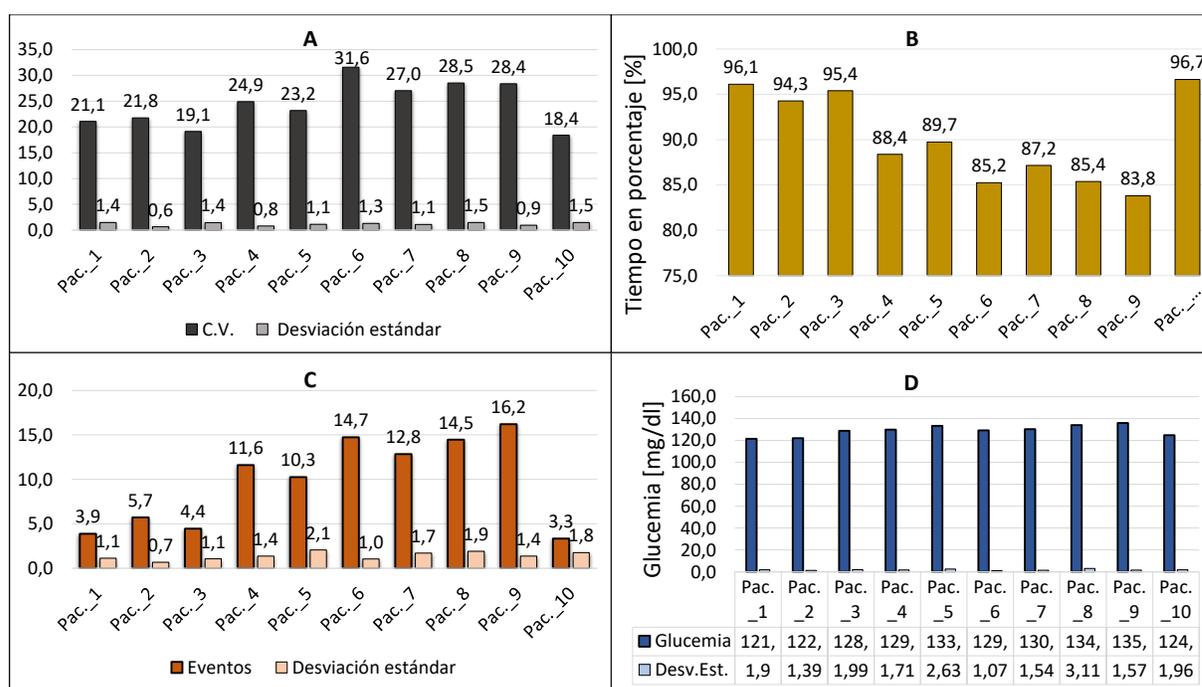


Figura 5-15: Cantidad de controles promedio calculados por paciente

Con esta nueva configuración en la sintonía del controlador implementando la estrategia ET en la Figura 5-16 se observa el resultado del desempeño del controlador por paciente. Los pacientes que presentan un mayor coeficiente de variación (pacientes del 4 al 9) en donde sobresale el paciente 6 con un C.V. de  $31,58 \pm 1,28$  5-16 A, tienden a estar un menor porcentaje de tiempo en normoglucemia resaltando al paciente 9 con un 83,8% del tiempo 5-

**16 B** y también a presentar un mayor número de casos de hiperglucemia (mayor a 180 mg/dl) como el paciente 9 con  $16,2 \pm 1,4$  casos **5-16 C**. Estos pacientes que presentan problemas en el desempeño muestran niveles de glucemia promedio mayores (hasta de  $9mg/dl$  que el resto de los pacientes) **5-16 D**, aún después de modificar las matrices del control y el estimador. Son estos casos los que precisamente muestran dificultades en la estabilidad del paciente en donde se ejecuta un mayor número de cálculos e insulina aplicada por el controlador.



**Figura 5-16:** Resultados promedio por paciente, **(A)** Coeficiente de variación, **(B)** Porcentaje de tiempo en normoglucemia, **(C)** Número de casos por encima de 180 mg / dl y **(D)** Glucemia promedio. \*Pac.=Paciente

Por último en la Figura 5-17 se observa la dinámica promedio de los diez pacientes implementando el umbral de 5,5 de donde se puede observar por las estadísticas anteriores que sí hay una pérdida en el desempeño del controlador, permitiendo a los pacientes alcanzar valores promedios más altos de glucemia y valores pico mayores en la ingesta de carbohidratos, sin embargo, esta pérdida en el desempeño no llega a ser representativa comparada con la cantidad de controles que disminuyen durante de la simulación, un aumento en la glucemia promedio y el número de casos de hiperglucemia que no superan aproximadamente el 30% comparado con un 69% de disminución de número de cálculos en el controlador. Parcialmente la implementación de la estrategia ET muestra ventajas significativas en la reducción de cálculos e inyecciones de insulina con un decremento en el desempeño del controlador que es parcialmente admisible y que no involucra casos de hipoglucemia bajo el esquema de simulación A.

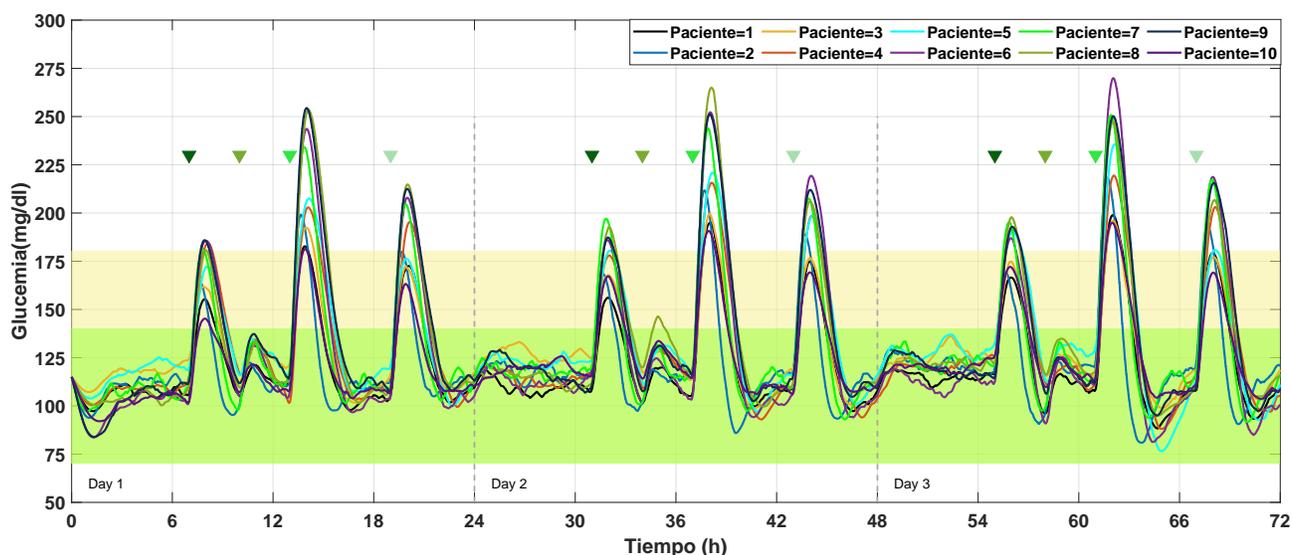
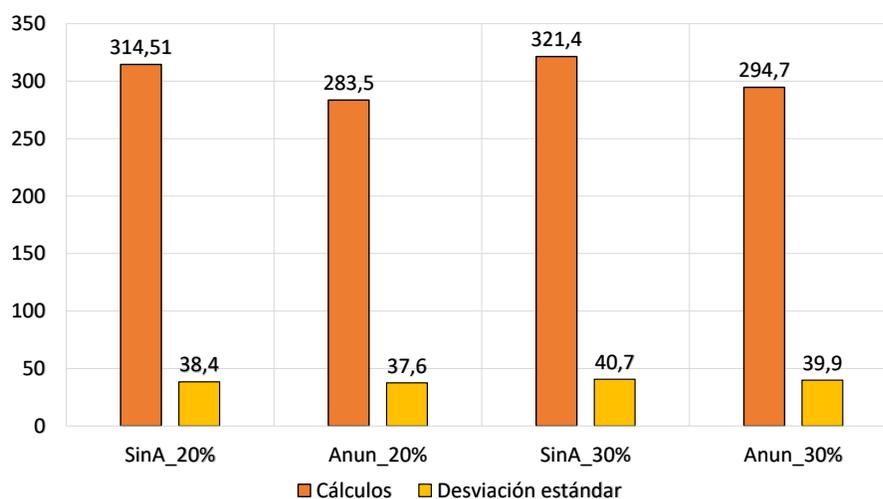


Figura 5-17: Glucemia promedio por paciente con variaciones del 10% y anuncio de comida

### 5.2.3. Pruebas de desempeño del nuevo controlador

Con la estrategia analizada, los valores promedios en el número de controles calculados y verificando el efecto que tiene el umbral en el desempeño del controlador se realiza la última prueba bajo el esquema de simulación de la Figura 5-4. En esta prueba se usan variaciones paramétricas del 20% y 30% y además se plantean escenarios con y sin anuncio de ingesta de carbohidratos al controlador. Para estas nuevas modificaciones en las condiciones a las que se expone el controlador se usan las matrices de sintonía de la Tabla 5-3, notando la alta sensibilidad del controlador ante la matriz de sintonía  $Gw$  que involucra el error planta modelo.

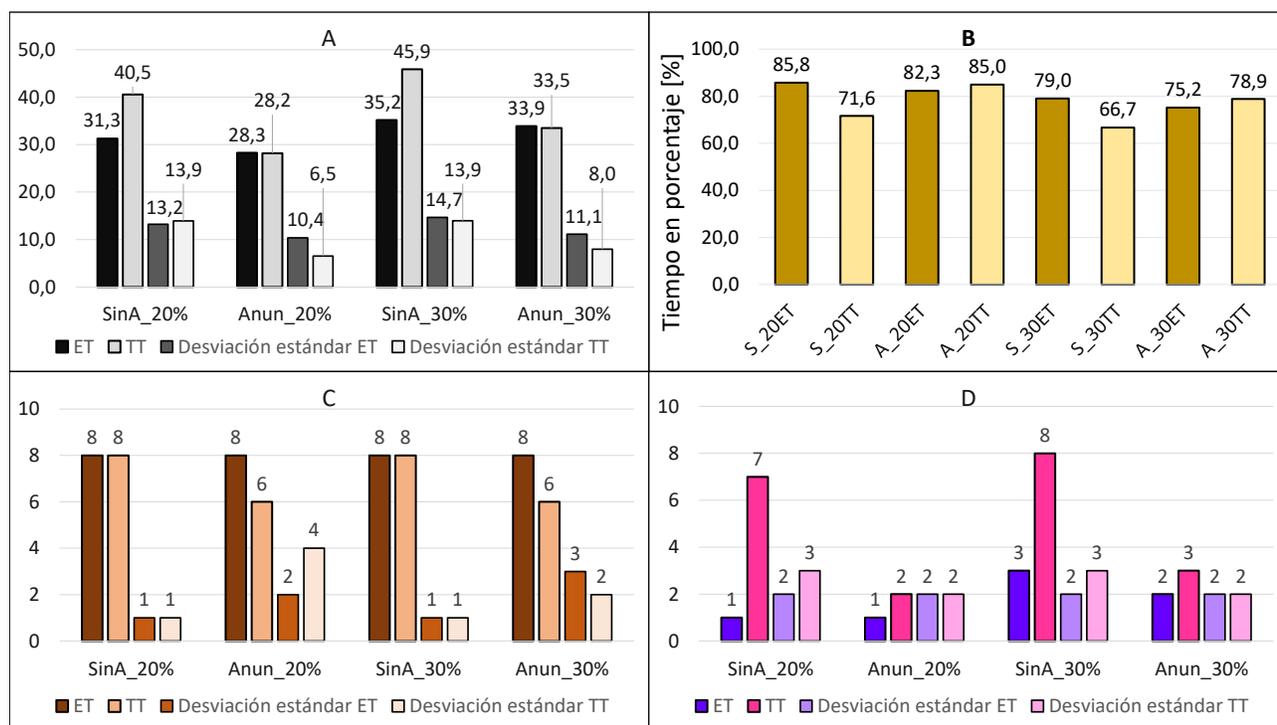
En la Figura 5-18 se observa la cantidad de cálculos de la dosificación de insulina óptima promedio realizados por el controlador con/sin anuncio de carbohidratos y con variaciones del 20/30%, de esta figura se puede notar que el número de cálculos realizados por el controlador se incrementa al no tener información de las comidas consumidas, esta información permite tener una proyección más acertada sobre la evolución de la glucemia en los pacientes virtuales. cabe recordar que bajo el esquema tradicional de activación por tiempo el número de controles calculados en las simulaciones es de 864 y que con la estrategia de ET esto se reduce en el peor de los casos hasta un 60% del total de los cálculos cuando se tienen variaciones del 30% y no se anuncia la comida al sistema de control. Por otra parte, es claro que con una variación paramétrica más grande se incrementa el número de cálculos realizados por el controlador debido a la discrepancia entre la planta y el modelo, además del esfuerzo del controlador por regular los niveles de glucemia.



**Figura 5-18:** Cantidad de controles calculados: Anun=Anuncio, SinA=Sin Anuncio, Variaciones paramétricas del 20 y 30 %

Los resultados del desempeño del controlador en el porcentaje de tiempo que pasan los pacientes virtuales en normoglucemia se muestran en la Figura 5-19 B, donde se compara el resultado de las variaciones paramétricas y el anuncio de comida con las estrategias ET y TT, resaltando que el porcentaje de tiempo es mayor para la estrategia ET cuando no se realiza el anuncio de comida y por el contrario la estrategia TT da mejores resultados en las pruebas cuando se da el anuncio de comida. El coeficiente de variación que presenta la estrategia ET es menor considerando el aumento en las variaciones paramétricas frente al TT aunque presenta una desviación estándar mayor, las menores variaciones para la estrategia de activación de eventos se da con un esquema de anuncio de comida y variaciones paramétricas de 20 % con un valor de  $28,3 \pm 10,4$ . Por otra parte el coeficiente de variación más alto se da cuando se tienen variaciones paramétricas del 30 % y no se anuncian las comidas al controlador presentando valores de  $45,9 \pm 13,9$  en la estrategia TT 5-19 A. Estos resultados muestran que la estrategia ET se comporta mejor cuando se trabaja un esquema sin anuncio de comidas y que en general tiene una menor variación de los valores medios de glucemia de los pacientes.

De las Figuras 5-19 C y D se observa el número de casos de hiperglucemia e hipoglucemia respectivamente que presentan las dos estrategias de control. Se observa que el controlador que usa ET con casi todas las ingestas de carbohidratos genera un caso de hiperglucemia dando un total de 8 casos por cada simulación en cada paciente, pero para el controlador base TT cuando se realiza el anuncio de comida con variaciones del 20 y 30 % se logra reducir el número de eventos a 6, también se muestra que para estas condiciones de simulación se dan casos de hipoglucemia, en donde el ET presenta una menor cantidad de eventos en los cuatro escenarios y que no anunciar la comida en el controlador base TT provoca que las magnitudes de insulina inyectadas al paciente generen entre 7 y 8 casos de hipoglucemia.



**Figura 5-19:** Resultados con/sin anuncio y variaciones del 20/30 %, (A) Coeficiente de variación, (B) Porcentaje de tiempo en normoglicemia, (C) Número de casos por encima de 180 mg/dl y (D) Número de casos por debajo de 70 mg/dl

Los niveles promedio de glucemia de los dos controladores bajo los cuatro escenarios se muestran en la Figura 5-20. La estrategia base TT logra mantener niveles de glucemia menores que el ET en los cuatro escenarios, sin embargo, el controlador con ET logra tener un buen desempeño si se comparan las estrategias manejando una variación paramétrica del 20% cuando no se anuncia la comida en el ET y cuando se anuncia la comida en el TT dando valores de  $124,0 \pm 8,4$  y  $124,6 \pm 10,9$  respectivamente. Del mismo modo cuando se presentan variaciones paramétricas del 30% cuando no se anuncia la comida en el ET y cuando se anuncia la comida en el TT se obtienen los valores promedio de  $127,3 \pm 11,8$  y  $126,7 \pm 10,5$  respectivamente. Los resultados dan el indicio de que bajo la sintonización usada para la estrategia ET cuando no se anuncia la ingesta de carbohidratos se obtienen resultados que se aproximan a los del controlador base cuando si se envía esta información al control.

Por último en la Figura 5-21 se observa la dinámica de la glucemia promedio de los diez pacientes en los cuatro escenarios usando la estrategia ET. Ya sea en el escenario que usa variaciones paramétricas del 20 o del 30%, el nuevo controlador responde mejor cuando no se realiza el anuncio de comida, esto debido a que en la sintonización es difícil encontrar unas matrices que logren un buen desempeño en los casos en los que se hace el anuncio y

en los que no, de igual forma se nota el incremento de los niveles de glucemia en los cuatro casos debido al tiempo que pasa el sistema sin calcular ni inyectar unidades de insulina, sin embargo, no se muestran valores constantes de hiperglucemia severa y en los casos de hipoglucemia que se ven al inicio de la simulación se deben al cambio abrupto en el que en un instante de tiempo se hace una variación paramétrica del 30 % que sería el peor de los casos en un escenario real.

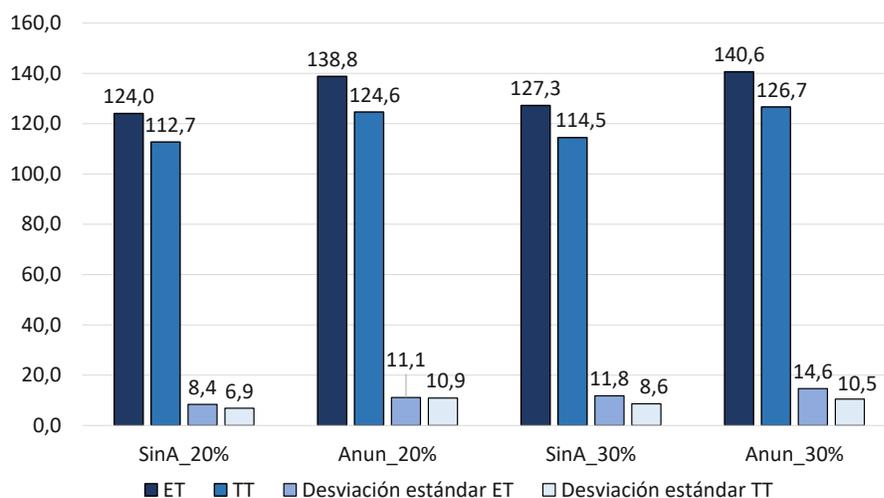


Figura 5-20: Glucemia promedio con/sin anuncio de comida y variaciones del 20/30 %

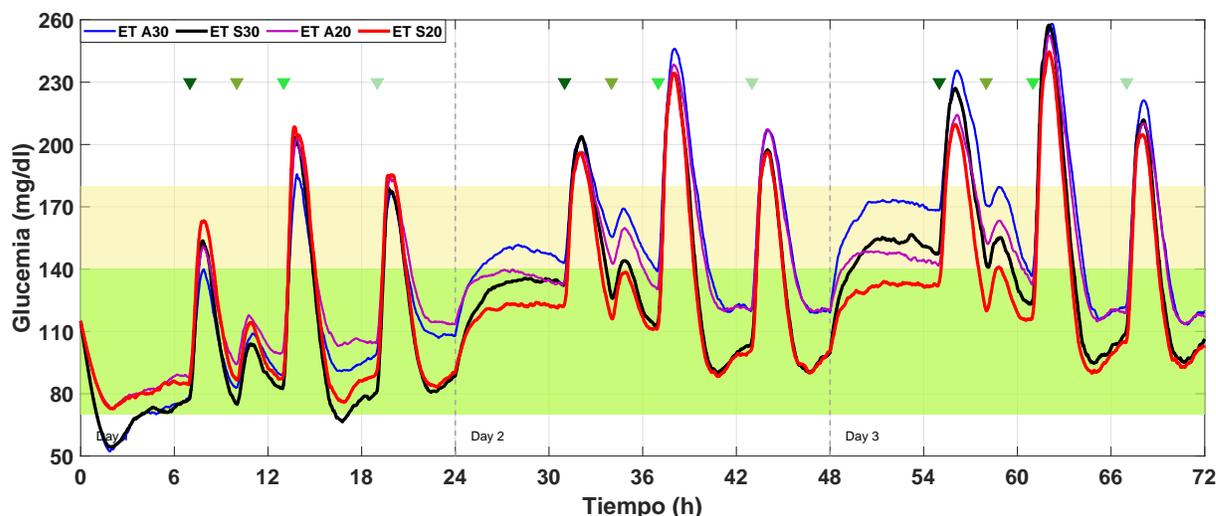


Figura 5-21: Dinámica de la glucemia con/sin anuncio de comida y variaciones del 20/30 %

Con el fin de comparar el desempeño obtenido por el controlador que usa la estrategia ET con algunos de los resultados encontrados y previamente comparados en la literatura se construye la Tabla 5-4, tomando en cuenta que los resultados del día 1 se cambian con el día 2 para manejar el mismo formato (día 1: parámetros normal, día 2: variación hipoglucemia del 30 %, día 3: variación hiperglucemia del 30 %). En la tabla se muestran controladores

basados en MPC y algunos de ellos integran estrategias adaptativas para regular los niveles de glucemia en la sangre. Se puede observar que, a diferencia de las estrategias encontradas en la literatura, la estrategia ET al implementar un umbral de  $\epsilon = 5,5$  se presentan casos de hipoglucemia con mayor frecuencia cuando se tienen variaciones paramétricas que inducen a este estado, pero responde adecuadamente cuando las variaciones inducen a la hiperglucemia tanto cuando se anuncia la comida como cuando no se hace. En los niveles medios de glucemia para el día que no tiene variaciones y para el tercer día se evidencian valores que se encuentran dentro de los rangos de los controladores encontrados en la literatura con 169,2 mg/dl con anuncio de comida y 148,9 mg/dl sin anuncio de comida, para el caso de la variación que lleva a hipoglucemia los valores medios son muy bajos tanto en las pruebas con anuncio como en las pruebas sin anuncio de comida con 102,0 mg/dl y 96,5mg/dl respectivamente. Hay que recordar que los controladores encontrados están enfocados solo en la regulación del paciente y no orientados a su implementación en un dispositivo portátil, sin embargo, es bueno tener presente cual es el nivel de los controladores solo enfocados al objetivo de regulación del paciente para tratar de encontrar un balance entre el desempeño deseado y la implementación real de la estrategia en un sistema embebido.

**Tabla 5-4:** Comparación del desempeño del controlador con diferentes esquemas de MPC de estudios reportados en la literatura.

Estrategia	BG media	* 70-180 mg/dl	* < 70 mg/dl	* >250 mg/dl
<b>Con anuncio de comida</b>				
ZMPC-AV-OF-ET (Día 1)	152.5 (22.3)	80.4 (17.9)	0 (0)	0 (5.4)
ZMPC-AV-OF-ET (Día 2)	102.0 (16.1)	86.2 (18.8)	10.3 (18.3)	0 (0)
ZMPC-AV-OF-ET (Día 3)	169.2 (29.9)	67.6 (37.7)	0 (0)	3.2 (14.1)
ZMPC-OF-AD [Villa-tamayo, 2021] (Día 1)	127.1 (4.7)	93.1 (5.0)	0 (0)	0 (0)
ZMPC-OF-AD [Villa-tamayo, 2021] (Día 2)	113.1 (10.4)	94.2 (6.4)	0 (0)	0 (0)
ZMPC-OF-AD [Villa-tamayo, 2021] (Día 3)	156.6 (4.0)	78.3 (4.1)	0 (0)	0 (0)
[Shi et al., 2019](basal 100 %)	135.9 (8.0)	90.6 (11.7)	0.0 (0.0)	0.0
[Shi et al., 2019](basal 130 %)	126.5 (9.3)	93.3 (11.5)	0.0 (2.4)	0.0
[Shi et al., 2019](basal 70 %)	146.1 (9.6)	85.8 (12.6)	0.0 (0.0)	0.0
[Gondhalekar et al., 2016, Shi et al., 2019] (basal 100 %)	136.5(8.4)	90.5 (12.1)	0.0(0.0)	0.0(0.0)
[Gondhalekar et al., 2016, Shi et al., 2019] (basal 130 %)	126.0 (9.2)	93.3 (11.4)	0.2 (2.7)	0.0(0.0)
[Gondhalekar et al., 2016, Shi et al., 2019] (basal 70 %)	148.9 (10.3)	84.7 (13.6)	0.0 (0.0)	0.0(0.0)
<b>Sin anuncio de comida</b>				
ZMPC-AV-OF-ET (Día 1)	135.4 (17.69)	85.12 (11.2)	0 (0)	0 (4.7)
ZMPC-AV-OF-ET (Día 2)	96.5 (6.1)	77.6 (21.7)	15.5 (13.5)	0 (0)
ZMPC-AV-OF-ET (Día 3)	148.9 (17.6)	79.7 (10.58)	0 (0)	1.2 (9.8)
ZMPC-OF-AD [Villa-tamayo, 2021] (Día 1)	127.3 (10.6)	89.9 (12.7)	0 (0)	0 (0)
ZMPC-OF-AD [Villa-tamayo, 2021] (Día 2)	107.4 (17.0)	93.7 (11.1)	0 (0)	0 (0)
ZMPC-OF-AD [Villa-tamayo, 2021] (Día 3)	159.7 (23.0)	74.7 (24.0)	0 (0)	0 (1.2 )
[Shi et al., 2019](basal 100 %)	153.8 (13.5)	70.5 (11.5)	0.0(0.0)	1.9 (7.1)
[Shi et al., 2019](basal 130 %)	143.6 (12.7)	73.6 (11.7)	0.0 (2.1)	0 (4.7)
[Shi et al., 2019](basal 70 %)	166.4 (21.9)	64.8 (18.0)	0.0(0.0)	5.9 (13.4)
[Gondhalekar et al., 2016, Shi et al., 2019] (basal 100 %)	159.0 (18.1)	66.3 (14.3)	0.0 (0.0)	3.6 (10.8)
[Gondhalekar et al., 2016, Shi et al., 2019] (basal 130 %)	145.8 (13.8)	71.8 (12.8)	0.0 (1.5)	0.0 (5.8)
[Gondhalekar et al., 2016, Shi et al., 2019] (basal 70 %)	180.6 (31.6)	56.1 (21.1)	0.0 (0.0)	12.2 (17.2)

\*% de tiempo

### 5.3. Resultados escenario B

Con la implementación de la estrategia de activación de eventos analizada y teniendo los resultados poblacionales de los pacientes virtuales sobre la glucemia promedio, CV, el tiempo en normoglucemia y cantidad de controles calculados se pasa al escenario de emulación de la Figura 5-5. Bajo este esquema se realizan pruebas en tiempo real en donde un minuto corresponde a un minuto real y no uno simulado como en las pruebas anteriores, es importante entender el comportamiento de la batería y el efecto que tiene esta estrategia en el tiempo y la energía disponible para el uso del PA.

#### 5.3.1. Descarga de la batería

Esta prueba se realiza para identificar los valores máximos, mínimos y la forma en la que la batería gradualmente pierde la capacidad de reacción electroquímica. Se limitan las tareas del sistema embebido al mínimo, la única tarea asignada al controlador es la de recepción de datos por parte del gestor de la batería mediante comunicación I2C y el almacenamiento de estos datos en un archivo de texto .txt, las variables de voltaje y el porcentaje de energía actual se toman cada un segundo. La temperatura promedio del procesador fue de  $52^{\circ}C$ , solo al inicio del proceso la frecuencia del procesador alcanzó  $1,2GHz$  pero durante la operación de la toma de datos las frecuencias se mantienen en valores normales de operación sin mucha carga computacional (frecuencia en el procesador de  $0,6GHz$ ). En la Figura 5-22 se observa la gráfica de la descarga de la batería en voltaje, tomando como punto de máxima carga  $4,2081V$  y como punto mínimo de carga (batería totalmente descargada)  $2,9360V$ , la batería tarda aproximadamente 12 horas y media en descargarse totalmente, el rizado de la figura se debe al ruido de medición y a los procesos internos del sistema operativo y sus periféricos.

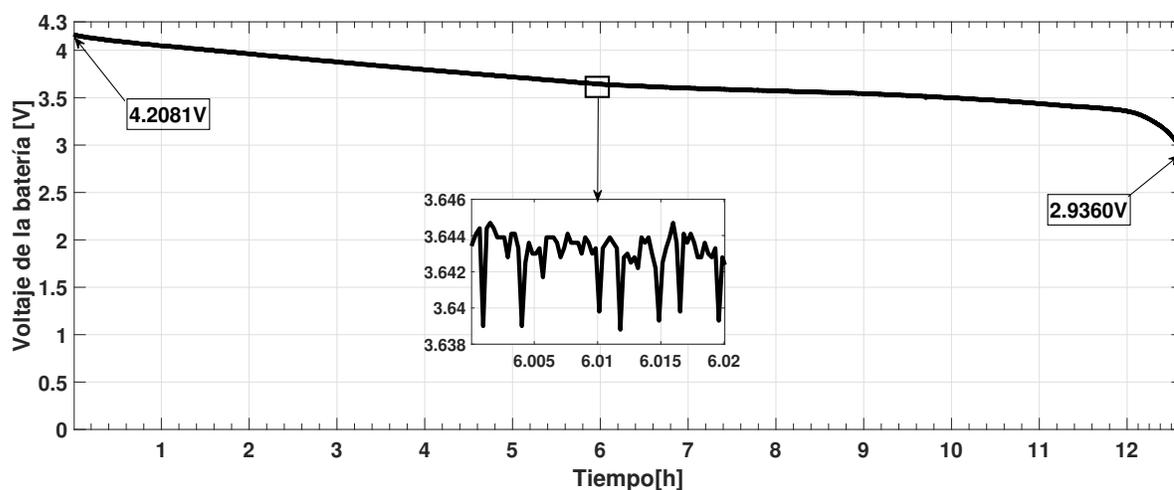


Figura 5-22: Dinámica de la descarga de la batería en voltaje

De la misma prueba anterior se registra la dinámica de la descarga de la batería en porcentaje, como se observa en la Figura 5-23, de acuerdo a los valores obtenidos por el algoritmo B que usa los rangos de valores de la Tabla 5-1 se da el comportamiento de la descarga de la batería a medida que se consume el material activo y la superficie funcional hasta que su capacidad de reacción electroquímica alcanza su límite y se corta la corriente hacia el embebido. De la figura se observa que en las tres primeras horas la batería se descarga aproximadamente un 15 %, posteriormente de la tercer hasta la sexta hora se consume aproximadamente un 27 %, de la sexta hora hasta la novena se decrementa la velocidad de descarga consumiendo un 18 %, de este punto en adelante el consumo de energía se acelera y se consume aproximadamente un 32 % hasta llegar has las doce horas dejando para la última media hora una descarga súbita del nivel de la batería restante. Es claro que la descarga de la batería no es lineal, sin embargo, al sacar un promedio se obtiene un 1 % de descarga en aproximadamente 7,58 minutos y el tiempo de carga total de la batería tarda alrededor de 3 horas.

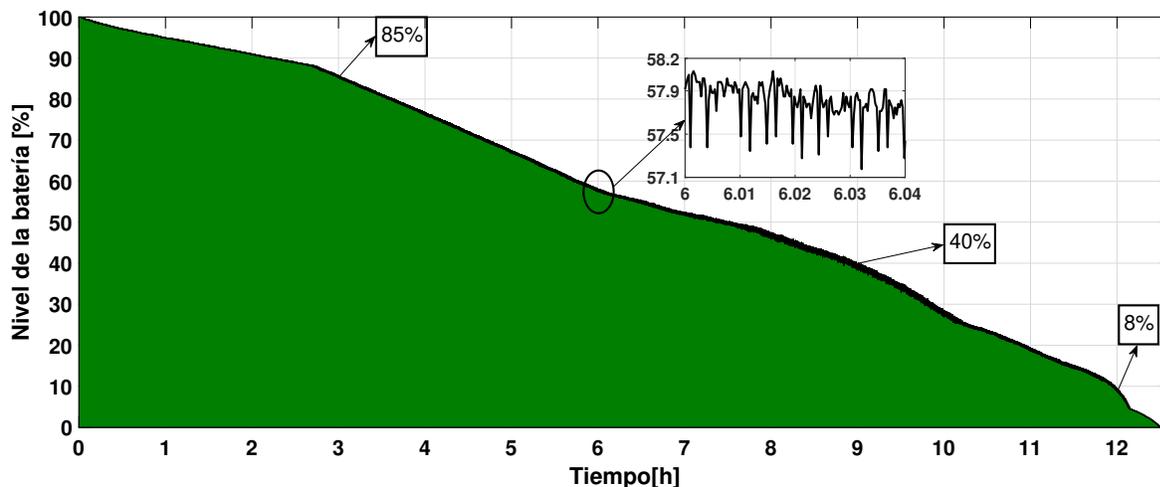
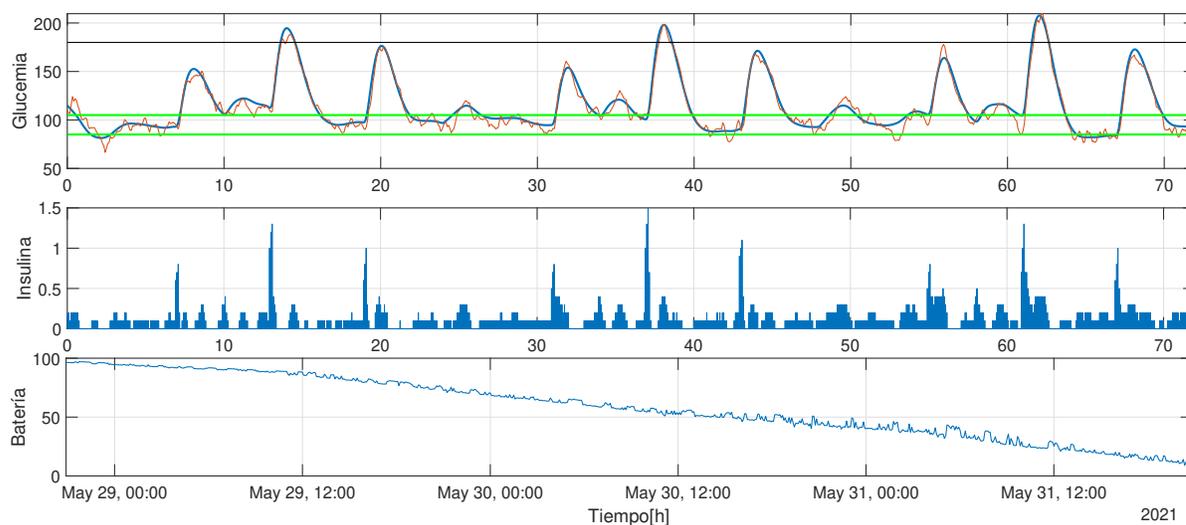


Figura 5-23: Dinámica de la descarga de la batería en porcentaje

### 5.3.2. Prueba de desempeño en la emulación del PA

Para las pruebas de desempeño se toma como referencia el resultado de la emulación del controlador base TT de la Figura 5-24, la prueba comienza del 28-May-2021 20:53:51 hasta el 31-May-2021 20:51:07 para un total de 72 horas en donde se usan las matrices de sintonía de la Tabla 4-2 en el ZMPC-AV-OF, para esta prueba se usa una variación paramétrica del 10 % y anuncio de comida. El controlador base regula adecuadamente la glucemia del paciente virtual y al finalizar la prueba el nivel de energía de la batería es de 10.79 % lo que permite tener al paciente una autonomía aproximada de 80 horas de funcionamiento.



**Figura 5-24:** Prueba base de la estrategia TT

En la etapa inicial de la emulación del PA se cuenta con la batería (PiSugar2 Pro), el sensor inteligente y el micro-motor paso a paso siguiendo el esquema de emulación de la Figura 5-5, en donde se aplica el algoritmo de la Figura 5-6 entre el paciente virtual, el sensor inteligente y el controlador embebido. Para esta prueba se hace el anuncio de ingesta de carbohidratos al controlador, las variaciones paramétricas son del 10% y se maneja un umbral  $\epsilon = 5,5$ , para este control se usan las matrices de sintonía de la Tabla 5-2 y se registran los datos del desempeño en términos de la glucemia de un paciente virtual, las cantidades de insulina inyectadas, el nivel de la batería en porcentaje y el tiempo transcurrido. Los resultados de la primera prueba se muestran en la Figura 5-25. La prueba se inicia 20-May-2021 16:35:32 y termina el 21-May-2021 22:17:48 con la descarga total de la batería, la emulación dura un total de 29.7 horas en las cuales el sistema logra controlar al paciente aplicando altas unidades de insulina en intervalos de tiempo prolongados en donde solo se realiza el cálculo de la dosificación de insulina óptima 80 veces por el controlador. Sin embargo, a pesar de que el controlador está “*apagado*” durante los periodos de tiempo en los que no requiere el cálculo de la dosificación de insulina este embebido (Raspberry Pi 3 B), al usar el comando “*sudo shutdown now*”, apaga el procesador pero algunas funciones aún continúan consumiendo energía, aproximadamente unos 100mA instantáneos más el consumo de los periféricos que mantienen energizados debido a que la SBC no hace un apagado global, el resultado de esto es que el sistema a pesar de estar implementando la estrategia ET sigue consumiendo energía en los periodos entre cálculos del problema de optimización  $k_e$  y  $k_{e+1}$ . El problema con esto es que no solo el consumo de energía es constante si no que adicional se consume mucha más energía por el encendido del dispositivo provocando que en vez de realizar un ahorro energético se aumente el consumo y la batería dure mucho menos.

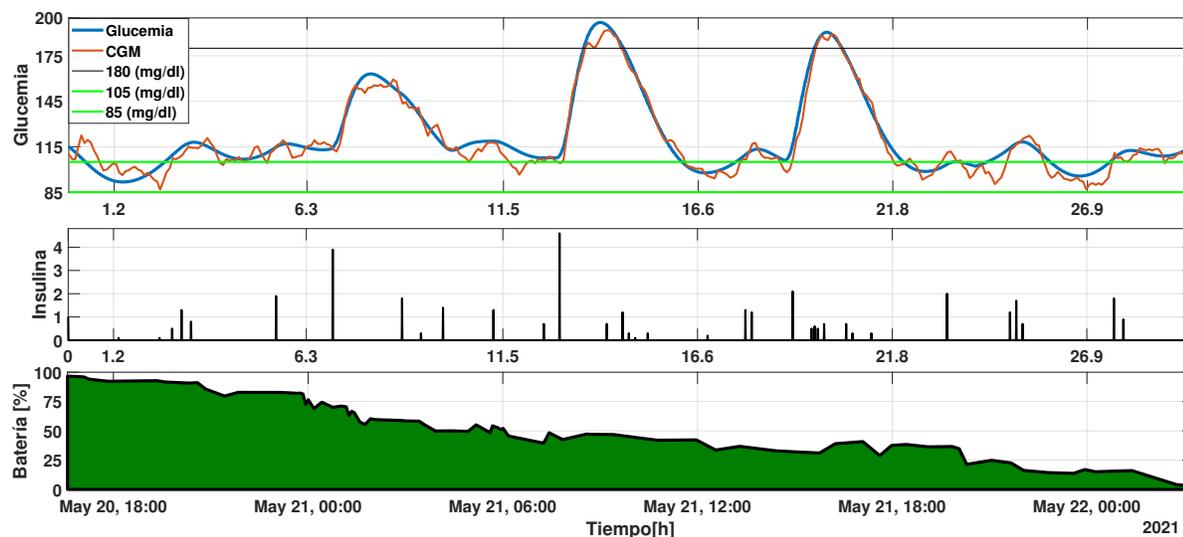


Figura 5-25: Prueba inicial del PA

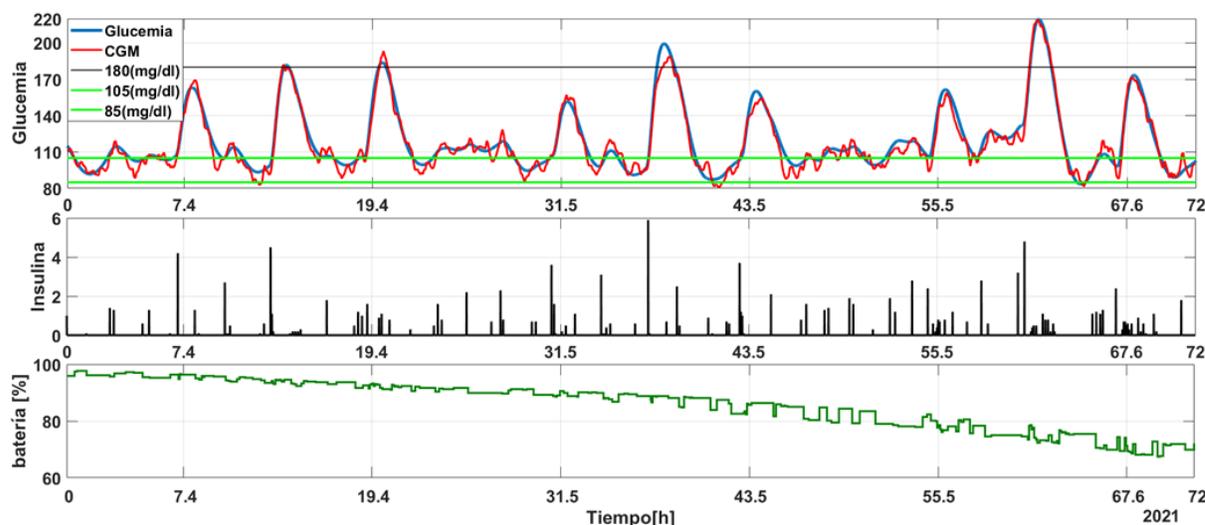
Con el fin de eliminar el consumo de energía mientras no se usa el controlador se hace uso del gestor de la batería IP5312 para controlar el apagado y encendido del embebido, la configuración aplicada involucra el nivel de apagado seguro, el retardo del apagado entre otros, la configuración aplicada en “*sudo nano /etc/pisugar-server/config.json*” es la siguiente:

```
{
  "auto_wake_time": null,
  "auto_wake_repeat": 0,
  "double_tap_enable": false,
  "auto_shutdown_level": 1.0,
  "auto_shutdown_delay": 0.5,
  "auto_charging_range": [80.0,100.0]
  "full_charge_duration": null,
  "auto_power_on": false}

```

Con la configuración implementada en el gestor de la batería se reanuda la prueba bajo las mismas condiciones dando como resultado la dinámica de la glucemia, la insulina y el nivel de la batería de la Figura 5-26. La emulación comienza desde 25-May-2021 16:35:15 hasta el 28-May-2021 16:35:14 con un total de 72 horas, los objetivos de control de la aplicación se cumplen bajo las condiciones de emulación en el paciente usado para la prueba. La batería al finalizar el tercer día termina con aproximadamente 70% de energía, en la dinámica de la descarga se muestra la frecuencia y las variaciones que genera el apagado y encendido del embebido, También se ve reflejado el esfuerzo que requiere el procesador para el cálculo de la dosificación de insulina elevando la frecuencia del procesador hasta  $1,2GHz$  cuando se realiza el proceso de optimización, esto sumado al consumo del motor al suministrar la insulina calculada. En esta figura se observa que durante los periodos entre comidas la frecuencia en el

cálculo de dosificación de insulina disminuye y cuando se ingieren carbohidratos la frecuencia del cálculo de la dosificación se ejecuta en cada instante de tiempo  $k_e = k$ , comportándose como el controlador TT por cortos periodos de tiempo, el número de veces que se enciende, se calcula y se inyectan las cantidades de insulina óptimas es de 251 veces en los tres días.



**Figura 5-26:** Emulación del PA usando ET, anuncio de comida y variaciones del 10 %

En las Figuras 5-27 y 5-28 se observan las diferencias en el desempeño de la regulación de glucemia del paciente virtual, la insulina inyectada y la energía consumida de la estrategia ZMPC-AV-OF- ET y TT. En la tarea de regulación del paciente ambos controladores cumplen con la estabilidad, con la respuesta ante la ingesta de carbohidratos y a las variaciones paramétricas del 10 % que inducen hiper e hipoglucemia, la forma en la que se regula al paciente cambia de inyecciones constantes y en pequeñas magnitudes a unas mucho menos frecuentes pero con magnitudes mayores, a pesar de esta diferencia en la forma en la que se inyecta la insulina las cantidades suministradas por el ET son de 128,8 unidades de insulina y para el TT son de 131,2 unidades de insulina.

El TT finalizó la prueba con casi un 11 % de energía, esto comparado con un 70 % registrado para ET evidencia que aproximadamente se ahorra un 60 % de energía al implementar la estrategia en el controlador base. A pesar de las no linealidades presentadas en la reacción electroquímica al momento de la descarga de energía se puede aproximar a que el dispositivo del PA consume alrededor de un 0.105 a un 0.11 % de energía por cálculo y ejecución de control, esta variación se da porque no en todos los casos hay necesidad de aplicar insulina y por la diferencia entre las magnitudes cuando si hay que hacerlo, teniendo en cuenta que el motor tarda 13,9722 segundos en suministrar una unidad de insulina y que en algunos casos se llega a aplicar hasta casi 6 unidades de insulina. Cuando la estrategia TT alcanza un 30 % de energía consumida han transcurrido 25 horas, mientras que el ET alcanza esos niveles de consumo a las 72 horas. Bajo esta estrategia es posible pasar de una proyección de 80 horas

de funcionamiento cuando se usa el TT a una proyección de 240 horas de funcionamiento del PA cuando se usa el ET, tomando en cuenta que esto puede variar según la muestra poblacional dando un estimado entre el 180 horas y 266 horas según las condiciones de operación del dispositivo (con/sin anuncio de comida, magnitud de las variaciones paramétricas que pueda tener el paciente debido a ejercicios y demás acciones que afectan el metabolismo y la interacción de estas hormonas).

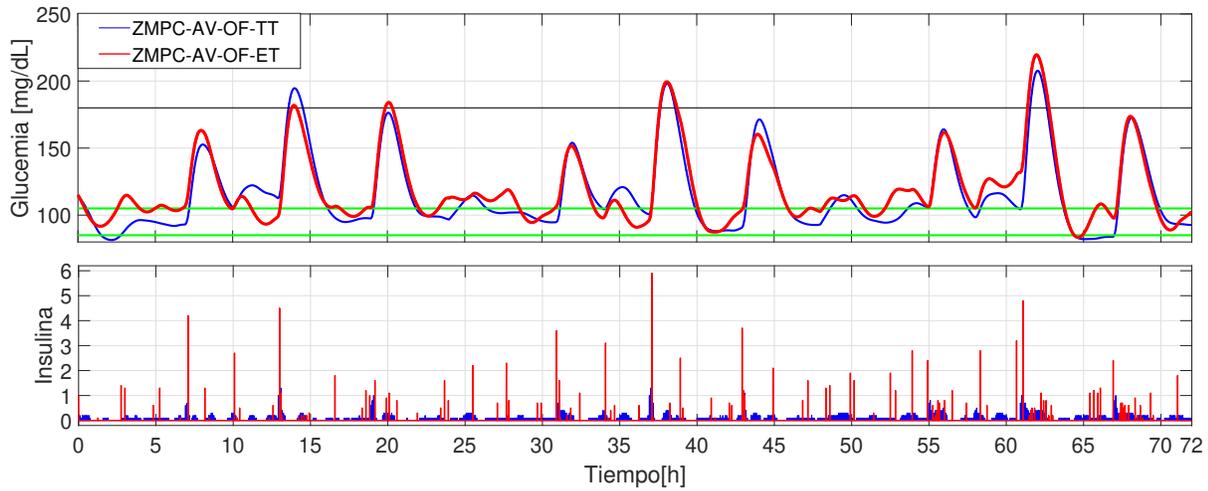


Figura 5-27: Comparación entre el ET y el TT en la glucemia e insulina

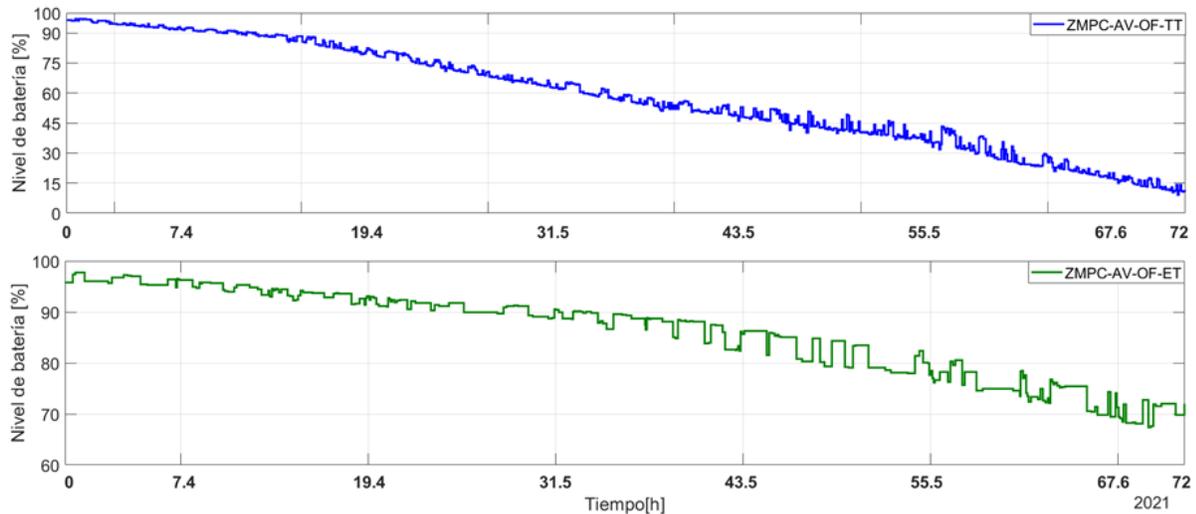
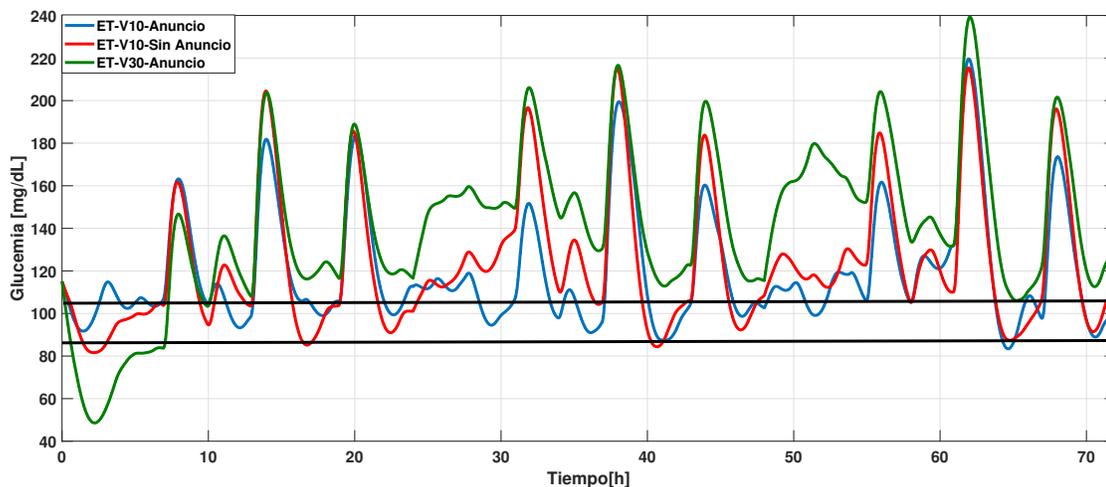


Figura 5-28: Comparación entre el ET y el TT en la energía consumida

Por último se realiza una prueba con el mismo escenario de simulación de la Figura 5-5, pero sin anunciar la comida al controlador y manejando variaciones paramétricas del 30%, para esto se usan las matrices de sintonía de la Tabla 5-3. Los resultados se comparan con los obtenidos cuando se da el anuncio de comida y se tienen variaciones paramétricas del 10%. En la Figura 5-29 se observa la comparación en la dinámica de la glucemia en el mismo paciente pero con diferentes variaciones para métricas y con anuncio de comida al controlador, de esta prueba se evidencia que para variaciones paramétricas grandes es difícil para el controlador mantener al paciente en la zona objetivo (105 - 85 mg/dl) incluso para los periodos entre comidas, incrementando los niveles promedios de glucemia y llegando a tener casos de hipoglucemia. Para el escenario en el que se manejan variaciones paramétricas del 10% pero no se anuncia la comida el controlador es capaz de mantener al paciente en una zona segura sin presentar casos de hipoglucemia graves ni aumentar considerablemente los niveles de glucemia promedio. La cantidad de controles en el esquema con anuncio y variaciones de 10% es de 251, para el esquema sin anuncio y variaciones del 10% se realizan 294 controles y para el que tiene variaciones de 30% y anuncio de comida se ejecutan 284 controles.

En la Figura 5-30 se observa la comparación en el consumo de energía de las tres condiciones a las que es sometido el controlador que implementa la estrategia ET, mostrando que evidentemente al manejar mayores variaciones paramétricas y tener que realizar un esfuerzo más grande por controlar los niveles de glucemia del paciente se incrementa el costo computacional y con esto la energía consumida por el dispositivo, pasando de aproximadamente 30% de energía consumida a casi un 40% en los tres días de emulación para los esquemas que no manejan anuncio de comida o tienen variaciones paramétricas del 30%.



**Figura 5-29:** Comparación de ET (Anuncio/10%, Sin Anuncio/30%), glucemia

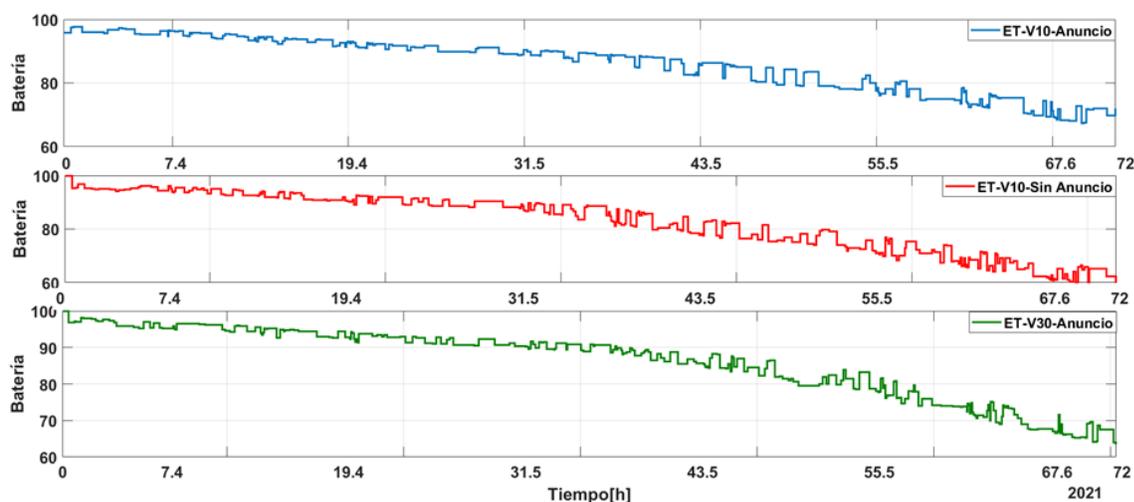


Figura 5-30: Comparación de ET (Con/sin anuncio, Variaciones 10%/30%), energía

## 5.4. Opciones alternativas para el uso de los recursos limitados en el páncreas artificial

Como medida alternativa para administrar de forma eficiente los recursos limitados (Energía e insulina) en el PA, se toma como inspiración la forma en la que los dispositivos portátiles actuales implementan el modo “ahorro de energía” para limitar sus funciones y extender la duración de la batería en los celulares y computadores portátiles. La ejecución de un modo ahorro de energía en un dispositivo como el PA requiere de la implementación de capas de seguridad y una estructura en la programación que regule el balance entre mantener estable al paciente y extender las horas de funcionamiento del dispositivo.

En este trabajo se da un primer acercamiento a la implementación de estrategias que permitan extender las horas de funcionamiento cuando se agota la energía de la batería o se agotan las unidades de insulina disponibles en el reservorio del dispositivo. Para esto se realiza una prueba piloto usando una estrategia de control MPC estándar. Para la prueba se implementa el mismo modelo de paciente diabético de la ecuación 2-6, y se usa como base el controlador ZMPC de la ecuación 2-19 pero solo se usa al paciente 1 para la prueba de las estrategias.

### 5.4.1. Esquema de eventos de activación y restricciones para manejar recursos limitados

En esta prueba se agrega la estrategia de ET y se añaden dos restricciones para manejar dispositivos portátiles con batería de duración limitada, cantidad limitada de insulina y capacidad de computo limitada. En primer lugar, se incorpora la función de ET para decidir si el problema de MPC debe resolverse en un instante de tiempo  $k$  como en el esquema

anterior.

$$\|y(k) - \tilde{y}(k)\| \geq \epsilon \quad \text{o} \quad k \geq k_l + H_c. \quad (5-2)$$

Junto con la estrategia ET, la primera restricción agregada tiene en cuenta la duración de la batería del dispositivo portátil. Esta restricción se activará cuando la vida útil restante de la batería este por debajo del 15% de la energía total de la batería, implementando la siguiente inecuación.

$$\sum_{i=1}^{H_c} \alpha u(i) \leq W_{min} - W_{act} - W_{proc}. \quad (5-3)$$

En donde la parte izquierda de la ecuación. 5-3, es decir,  $\sum_{i=1}^{H_c} \alpha u_i$ , es la proyección del consumo del sistema de energía debido a la inyección de insulina para el horizonte de control  $H_c$ . El dispositivo portátil funciona con un motor paso a paso con engranajes. El coeficiente  $\alpha$  define la relación entre los pasos necesarios del motor para inyectar una cantidad específica de insulina, es decir, define el consumo de energía  $W$  en [mWh], del motor paso a paso para aplicar una unidad de insulina.

En la parte derecha de la ecuación. 5-3, el término  $W_{proc}$  denota el consumo de energía debido a los cálculos del procesador para generar la dosis de insulina en el instante  $k$ . Se supone que el tiempo ( $t^*$ ) para encontrar la solución del problema de optimización es aproximadamente el mismo en todos los instantes de tiempo  $k$ , y por lo tanto, el valor de  $W_{proc} = W_p t^*$  resulta constante, donde  $W_p$  es la energía consumida por el procesador en cada instante de tiempo  $k$ .

El término  $W_{act}$  es el consumo total de energía de la batería en el instante de tiempo  $k$ , que se calcula como la energía total disponible en la batería (mAh) menos un consumo constante debido al funcionamiento básico del sistema integrado multiplicado por el tiempo de muestreo, y menos la energía consumida al generar la acción de control en el instante  $k$  y la aplicación de dicha dosis por el dispositivo portátil. Finalmente, el término  $W_{min}$  representa el valor mínimo seguro de vida útil de la batería que se puede alcanzar durante el horizonte de predicción  $H_p$ , considerando el consumo básico del sistema integrado y el cálculo de control en el horizonte de predicción.

La segunda restricción intenta prolongar el suministro de insulina y se activa dependiendo del consumo actual. Normalmente, la insulina usada en los tratamientos de T1DM es la insulina de acción corta llamada U-100 y cada reservorio contiene 300 unidades. La restricción se activará cuando la insulina inyectada alcance un porcentaje específico y se incorporará a la formulación de MPC cambiando el valor del límite superior  $u_{max}$ . Esta variable se cambia al 3% de su valor nominal cuando el consumo de insulina alcanza el 80 % de su capacidad

total (es decir, 240 U), y se cambia a 2% cuando alcanza el 90% .

En resumen, el nuevo controlador basado en MPC a resolver en cada instante de tiempo  $k$  esta dado por:

---

**si**  $|y(k) - \tilde{y}(k)| \leq \epsilon$  **o**  $k \geq k_l + H_c$  **entonces**  
 $\min_{u, \delta} V(y, u, \delta)$   
*s.t.*  
 $x(0) = \hat{x}, r(0) = r, r(i+1) = 0, \quad j \in I_{1:H_p-1},$   
 $x(i+1) = Ax(i) + Bu(i) + B_r r(i) + E, \quad j \in I_{0:H_p-1},$   
 $y(i) = Cx(i),$   
 $u_{min} \leq u(i) \leq u_{max}, |\delta| \leq \delta_{max},$   
**si** *insulinsupply*  $\leq 20\%$  **entonces**  
 $u_{max} = 0,03u_{max},$   
**si no, si** *insulinsupply*  $\leq 10\%$  **entonces**  
 $u_{max} = 0,02u_{max},$   
**fin si**  
**si** *batterylife*  $\leq 15\%$  **entonces**  
 $\alpha \sum_{i=1}^{H_u} u(i) \leq W_{min} - W_{act} - W_{proc}.$   
**fin si**  
**fin si**

---

### 5.4.2. Resultados y discusión

Para esta sección se consideraron elementos de hardware específicos, aunque el rendimiento se evaluó solo mediante simulación. Se entiende que el dispositivo portátil está equipado con un sistema integrado Raspberry Pi 3 B, un micro-motor paso a paso con una caja de reductora que genera 30 rpm como máximo y una batería de polímero de iones de litio de capacidad 6800 mAh. Las mediciones de consumo de energía del sistema integrado en reposo y procesamiento se tomaron fuera de línea con el sensor “Gravity: I2C Digital Wattmeter”. Con base en estos elementos, los términos de consumo de energía de las nuevas restricciones son:  $\alpha = (1657,22051 \text{step}/U)$  ( $0,01333 \text{mWh}/\text{step}$ )  $\approx 22,0907 \text{mWh}/U$ ,  $G_{proc} = (600 * 2,7833 * 10^{-4} * H_u) \approx 1,7 \text{mWh}$ , y  $W_{act} = 1500 * 0,083333 \text{mWh}$ .

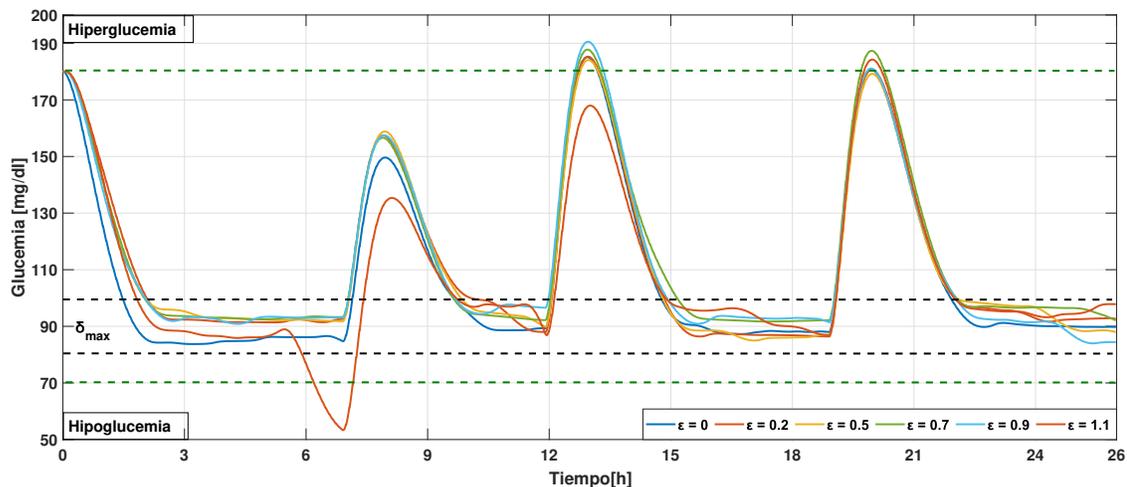
El tiempo de muestreo es  $T = 5$  minutos, y los parámetros de ajuste del controlador utilizados son:  $H_p = 80$ ,  $H_u = 10$ ,  $Q = 100$ ,  $R = 1 * 10^5$ ,  $S = 1 * 10^5$ . La referencia se eligió como 90 mg/dl y  $\delta_{max} = 10$ . El tiempo de simulación fue de 26 horas y se hace la suposición de que el total de unidades de insulina disponibles era de 43 U. La glucemia inicial se tomó como  $x_1(0) = 180 \text{mg}/\text{dl}$ , y se consideraron tres ingestas de comida a las 7 a.M. Con 51 g,

a las 12 p.M. Con 73 g y a las 7 p.M. Con 70 g.

El desempeño de las estrategias se evaluó mediante las siguientes métricas: i) las unidades de insulina consumidas, ii) la energía consumida, iii) el estado de glucemia, y iv) los tiempos en que se resolvió el problema de optimización.

Para evaluar el desempeño de la nueva propuesta y el efecto de cada restricción, se llevaron a cabo cuatro simulaciones para el escenario de simulación descrito anteriormente manteniendo los mismos valores en la sintonización. La primera evaluó solo el efecto de la estrategia ET, la segunda exploró la restricción del consumo de energía, la tercera evaluó la restricción del consumo de insulina y la última la estrategia completa. En todos los casos, los resultados se compararon con el ZMPC estándar.

En la Figura 5-31 se ilustra la primera simulación para los valores de  $\epsilon$  que van desde cero (estrategia estándar) hasta 1.1. Ambas formulaciones de MPC estabilizan al paciente con un rendimiento y métricas similares, por ejemplo, el consumo de insulina fue de aproximadamente 45,5 U para cualquier valor de  $\epsilon$ . Sin embargo, la energía total consumida estuvo en el rango de 1074,8 mAh ( $\epsilon = 0$ ) y 768 mAh ( $\epsilon = 1, 2$ ), ahorrando aproximadamente un 28 % de energía. Esto es esperado ya que el problema de optimización se calculó solo 183 veces para  $\epsilon = 1,1$ , mientras que se calculó 312 veces para un  $\epsilon = 0$ . Sin embargo, los eventos de hipoglucemia ocurren con más frecuencia con un  $\epsilon \geq 1,1$ . Además, se observó que para todos los valores de  $\epsilon$ , existen valores de glucemia por encima del umbral de hiperglucemia, especialmente después de las ingestas de comida. Pero, no se consideraron eventos de hiperglucemia ya que no duraron más de 40 min.



**Figura 5-31:** Resultado del desempeño del ZMPC al implementar diferentes valores de  $\epsilon$

En la Figura 5-32, se muestra el comportamiento de MPC incluyendo la restricción de

consumo de energía. Tal como fue diseñado, cuando la duración de la batería es inferior al 15%, la restricción 5-3 se activaba. Como resultado, se calcularon dosis menores logrando un ahorro energético del 4,5% y, de manera equivalente, el tratamiento se mantuvo durante casi una hora más. Después de que se agotó la batería, el tiempo para llegar a la zona de hiperglucemia fue mayor usando el controlador propuesto, lo que le puede dar al paciente más tiempo para reemplazar o recargar la batería.

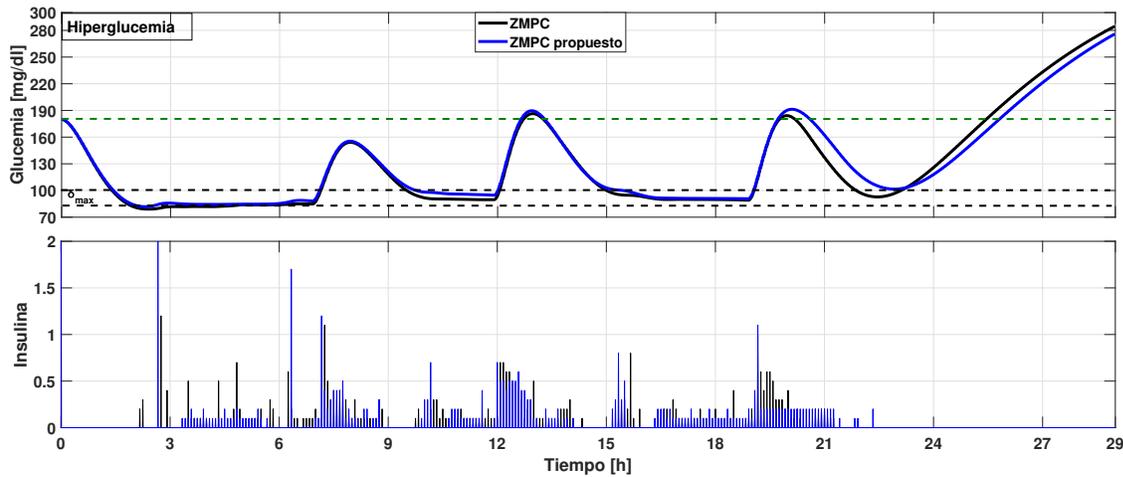


Figura 5-32: Rendimiento del MPC al incluir la limitación de energía de la batería

En la Figura 5-33, se muestra el efecto de la restricción en el suministro de insulina. Se asumió que el reservorio de insulina tenía 43 unidades. El MPC estándar (línea negra) y el que considera esta restricción (línea azul) agotaron las 43 unidades a 23,17h y 24,42h respectivamente. Aunque el tratamiento se prolongó debido a la restricción durante 1 hora y 15 minutos más, se produjo un episodio de hiperglucemia.

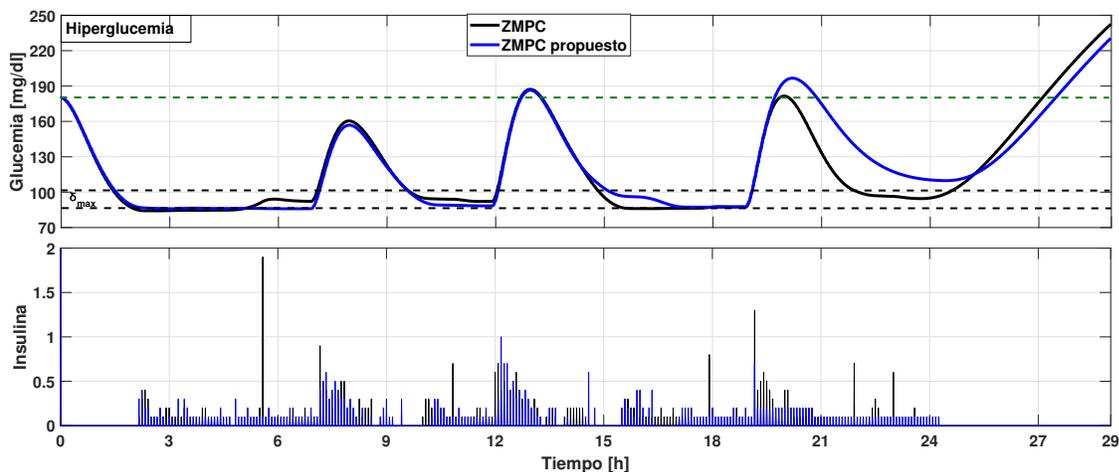
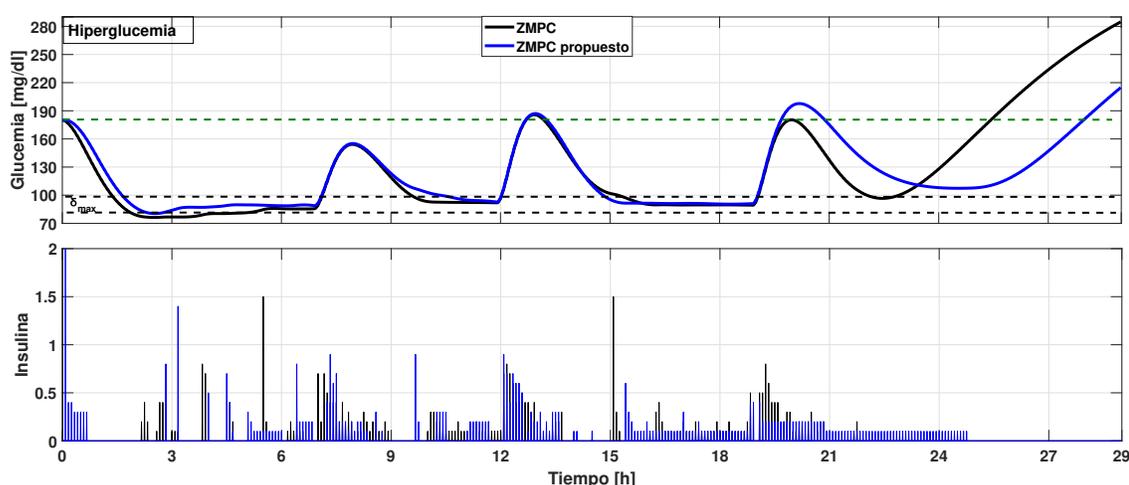


Figura 5-33: Rendimiento del MPC considerando la restricción de suministro de insulina

Finalmente, el desempeño de la estrategia completa, es decir, el MPC, incluido el componente

ET y las dos restricciones, se ilustra en la Figura 5-34. Al principio, el MPC estándar consigue llevar la glucemia a la zona normal más rápido que la estrategia propuesta debido al hecho de que el control no se actualiza con nuevas mediciones. Después de eso, el desempeño de ambas estrategias es similar, hasta que se activaron las restricciones para la duración de la batería y el suministro de insulina. En esta sección se pueden observar dosis más pequeñas de insulina. El reservorio de insulina se agotó totalmente para ambas estrategias después de las 22.08 y 23.5 horas, respectivamente. La estrategia propuesta logró extender el tratamiento por 85 minutos (este valor es la diferencia de tiempo cuando la glucemia supera el límite de 180 mg/dl en ambos casos), como se muestra en la figura. El consumo de energía de la estrategia propuesta fue del 10 % con respecto a la estrategia estándar. Cabe señalar que, aparte de este ahorro de energía, el tratamiento se mantuvo durante 85 minutos y durante este tiempo se mantuvo el consumo de energía.



**Figura 5-34:** Comparación entre la ZMPC estándar y el controlador propuesto con todas las estrategias

## 5.5. Resumen

En este capítulo se desarrolla y se integra la estrategia de activación de eventos con el fin de determinar si es necesario realizar el cálculo de la dosificación de insulina óptima según el nivel de glucemia actual del paciente comparado con la proyección que se hace con el modelo. Las pruebas se dividen en dos etapas, una en donde se realizan las pruebas poblacionales sacando las estadísticas sobre el número de cálculos del controlador, la glucemia promedio, casos de hipo e hiperglucemia, entre otras métricas enfocadas al desempeño del control en la regulación, además se hace una comparación del desempeño obtenido con otras estrategias que están enfocadas solo a la regulación del paciente con T1DM encontrando que, para un controlador que implementa estrategias enfocadas a reducir el consumo energético presenta

valores promedios en los mismos rangos que los encontrados en la literatura con excepción de los casos de hipoglucemia, en donde si hay un aumento en los casos que llevan al paciente a este estado y es algo en lo que se debe de trabajar para tratar de encontrar un balance entre el desempeño del controlador y el desempeño que tiene el embebido como dispositivo portátil.

En la segunda parte de la prueba se usa un esquema más realista en donde se emula el funcionamiento del páncreas artificial al implementar un micro-motor paso a paso como actuador, un sensor inteligente y una batería para el funcionamiento del dispositivo. Se obtienen los resultados de la dinámica de la descarga de la batería implementada, además de cuál es el comportamiento de la descarga cuando se ejecutan los cálculos de forma periódica y cuando se implementa la estrategia de activación de eventos en el que el número de veces que se ejecuta el cálculo y la inyección de las unidades de insulina disminuyen hasta un 70 % en algunos casos. Bajo las condiciones de emulación de este trabajo el consumo disminuye hasta un 60 %, que de acuerdo con los resultados poblacionales dependiendo del paciente, su sensibilidad a la insulina y la cantidad de carbohidratos que consuma puede llegar a un tiempo de funcionamiento del dispositivo de entre 180 y 266 horas, a diferencia del controlador base que alcanza unas 80 horas de funcionamiento.

Por último para las estrategias adicionales, se proporciona una primera aproximación a un MPC impulsivo que se enfrenta a recursos limitados. Esta nueva fórmula es atractiva especialmente para la aplicación del AP, que debe funcionar con una bomba de insulina portátil con una batería de vida útil y suministro insulina limitados. Como resultado principal, este nuevo primer enfoque ofrece al paciente más tiempo para recargar la batería o el suministro de insulina. Sin embargo, es necesario extender funciones de protección y supervisión a este tipo de estrategias que aseguren que el controlador en ningún momento priorice la duración de funcionamiento del dispositivo sobre un estado crítico del paciente que lo pueda llevar o mantener en niveles de glucosa peligrosos para su salud.

# 6 Conclusiones y recomendaciones

## 6.1. Conclusiones

- En este trabajo se analizó el desempeño de seis sistemas embebidos y tres paquetes de optimización de código abierto en cuanto a su implementación en la emulación de un PA utilizando la metodología HIL. Los sistemas embebidos y los paquetes de optimización se probaron con cuatro estrategias MPC de complejidad creciente. Los resultados muestran que, cuando se utiliza el paquete de optimización adecuado, los procesadores ARM pueden cumplir con éxito el objetivo de regular la glucemia en pacientes con diabetes Tipo 1, incluso bajo estrategias de control complejas.
- El paquete quadprog resultó ser el mejor solucionador con respuestas más rápidas y confiables con respecto a los otros paquetes. Los resultados obtenidos fueron muy similares a los obtenidos en Matlab y dieron garantías de aplicar fielmente el algoritmo en el PA a pesar de que el método de optimización utilizado en el quadprog de Matlab es diferente al disponible para Python. Por el contrario, los paquetes OSQP y CVXOPT muestran dificultades para manejar problemas complejos con respecto al número de variables de decisión y restricciones, lo que los hace inaceptables para el tratamiento de la diabetes tipo 1.
- Los resultados obtenidos al utilizar la estrategia de control más compleja y el paquete quadprog no muestran diferencias significativas en el rendimiento de los sistemas embebidos. Sin embargo, se encuentra una diferencia sustancial en la energía consumida durante la ejecución de las pruebas; Así, el sistema embebido recomendado para ser implementado como dispositivo portátil en las condiciones expuestas en este trabajo es la Raspberry Pi 3 B seguido de la Tinker Board S.
- Con el sistema embebido y el paquete de optimización seleccionado se toma como base la estrategia de control con eliminación de *offset* para implementar la estrategia de activación de eventos para un uso eficiente de los recursos energéticos del dispositivo, en donde se realiza una muestra poblacional que involucra la cantidad de veces y la frecuencia con la que se ejecuta el cálculo de la dosificación de insulina óptima mostrando que, bajo los esquemas de simulación planteados se puede llegar a reducir hasta en un 60 % la cantidad de veces que es necesario realizar el control pero notando

una disminución en el desempeño del controlador, concretamente al permitir casos de hipoglucemia cuando se tienen variaciones paramétricas del 30 %.

- Al momento de integrar la estrategia ET al funcionamiento del PA involucrando al actuador y a la batería se observa que el consumo de energía disminuye, aumentando el tiempo de funcionamiento del sistema embebido entre 100 y 186 horas dependiendo de los factores a los que se vea expuesto el paciente (ejercicios, horas nocturnas, ingesta de carbohidratos, entre otros), a pesar de la pérdida de desempeño en los resultados, se observa que es posible encontrar un balance entre los objetivos de control y la duración de funcionamiento del dispositivo, combinando la elección del umbral con las matrices de sintonía del control y el estimador.
- Tener en cuenta los aspectos reales de operación del PA permiten identificar las fortalezas y las debilidades de todo el conjunto de elementos involucrados en la regulación de glucosa del paciente, no solo la glucemia y las cantidades de insulina inyectadas son tenidas en cuenta si no también el consumo que genera el cálculo de la dosificación de insulina, el consumo y el tiempo que tarda el actuador en inyectar las cantidades calculadas, los protocolos de comunicación para el control y monitoreo, la temperatura y frecuencia del procesador, así como dinámica de la descarga de la batería influyen para que el PA cumpla satisfactoriamente su propósito.

## 6.2. Recomendaciones

Al desarrollo de las estrategias de control/estimación y a los elementos involucrados en la implementación real como el sistema embebido, la batería y demás elementos necesarios para la implementación del PA deben de estar en una constante mejora para garantizar la estabilidad de las posibles personas que hagan uso de un sistema como este y genere una mejora en su calidad de vida, algunos aspectos a tener en cuenta son:

- Desarrollar o integrar tecnologías y trabajos que permitan al menos igualar el rendimiento de las estrategias enfocadas a la regulación de glucosa en la sangre como la restricción de insulina a bordo (IOB) junto con las que están enfocadas al correcto funcionamiento del dispositivo embebido (capas de seguridad), así como las que incrementan la eficiencia energética (modo ahorro de energía), son fundamentales para encontrar un balance entre los objetivos de control de la aplicación y la ejecución de estas en un dispositivo portátil.
- Con el fin de acercarse cada vez más a obtener un producto que puedan usar con seguridad las personas que padecen esta enfermedad se deben de realizar pruebas que no solo se ejecuten mediante simulaciones sino pruebas que integren componentes reales, que lidien con los problemas de hardware de un dispositivo real. Esto con el fin de que

el PA sea concebido como un conjunto integrado y no como una estrategia diseñada para ejecutarse en un computador que luego debe ser adaptada para ejecutarse en un sistema embebido.

- Tener en cuenta el tamaño del dispositivo también es algo que influye en la comodidad de las personas que hagan uso de un dispositivo como este. Al momento de este trabajo la prioridad es tratar de encontrar un balance entre la regulación del paciente y el consumo energético por la operación del dispositivo, sin embargo, cuando se haya establecido una estrategia de control/estimación que cumpla con el objetivo de regulación y se integre con las estrategias del uso eficiente de los recursos energéticos del dispositivo sin perder desempeño, es importante la selección adecuada del procesador, sus periféricos, el actuador y la batería para el diseño de una tarjeta impresa que se acerque más a un producto, que a una prueba o a un prototipo.
- Para tratar de minimizar la pérdida de desempeño del controlador cuando se prueba en algunos pacientes y bajo ciertas circunstancias se puede tratar de integrar un método que evalúe la sensibilidad del paciente ante la insulina y con esto tener un umbral adaptable a cada paciente que permita regular adecuadamente la glucosa en sangre y prolongue la duración de funcionamiento del dispositivo.

# Glosario

**Sistema embebido:** Dispositivo electrónico diseñado para cumplir con funciones de propósito específico que puede ejecutar tareas en tiempo real, así como el manejo de protocolos de comunicación y periféricos para el control en diversas aplicaciones.

**CPU:** La unidad de procesamiento central (CPU) es la encargada de interpretar las instrucciones de un hardware haciendo uso de distintas operaciones aritméticas y matemáticas.

**Software:** Comprende el conjunto de los componentes lógicos necesarios que hacen posible la realización de tareas específicas, (organiza y envía las instrucciones hacia el Hardware).

**Hardware:** Partes físicas de un sistema informático, sus componentes eléctricos, electrónicos, electromecánicos, mecánicos y periféricos componen el soporte físico que ejecuta las instrucciones enviadas por el software.

**Control predictivo basado en modelo:** Control que se basa en hallar una secuencia de control al resolver un problema de optimización en un horizonte finito que usa como función de costo el modelo de la planta e incluye restricciones físicas del sistema.

**Diabetes mellitus tipo 1:** Enfermedad autoinmune que afecta a las células beta generadas por el páncreas para la regulación de glucosa en la sangre, provocando deficientes niveles de insulina y evitando la regulación de estas hormonas en la sangre.

**Filtro de Kalman:** Algoritmo desarrollado para identificar estados no medibles en un sistema dinámico lineal basado en un modelo estocástico.

**Hiper glucemia:** Estado de una persona que presenta niveles altos de glucosa en sangre  $\text{glucemia} > 180 \text{ mg/dl}$ .

**Hiper glucemia severa:** Estado de una persona que presenta niveles muy altos de glucosa en sangre  $\text{glucemia} > 250 \text{ mg/dl}$ .

**Hipoglucemia:** Estado de una persona que presenta niveles bajos de glucosa en sangre  $\text{glucemia} < 70 \text{ mg/dl}$ .

**Hipoglucemia severa:** Estado de una persona que presenta niveles muy bajos de glucosa en sangre glucemia  $< 54$  mg/dl.

**Normoglucemia:** Estado del paciente en donde los niveles de glucosa en sangre permanecen en rangos normales para el ser humano. Generalmente se establece entre 70mg/dl y 180mg/dl.

**Offset:** Error o desplazamiento en estado estacionario que hay entre la respuesta del sistema y el objetivo.

**Paciente Virtual:** Simulador basado en un modelo que representa las dinámicas de la glucosa, la insulina y los carbohidratos en pacientes con diabetes tipo 1.

**Páncreas Artificial:** Sistema encargado de la regulación autónoma de los pacientes con T1DM, consta de una bomba de infusión, un sensor de monitoreo continuo de los niveles de glucosa en sangre y un dispositivo que contiene las estrategias de control/estimación para la regulación del paciente.

**Sensibilidad a la insulina:** Habilidad de la insulina para estimular la utilización de glucosa e inhibir su producción, regulando que tan sensible es el cuerpo a los efectos de la insulina.

**Sistema de control impulsivo:** Sistemas en donde la entrada manipulada se considera como un impulso debido a su corta duración en relación al tiempo de muestreo. La aplicación de una entrada impulsiva genera un sistema caracterizado por una respuesta libre y una respuesta forzada en forma de discontinuidad.

# Bibliografía

- [Andersen et al., 2003] Andersen, E., Roos, C., and Terlaky, T. (2003). On implementing a primal-dual interior-point method for conic quadratic optimization. *Mathematical Programming*, 95:249–277.
- [Anderson et al., 2019] Anderson, S. M., Dassau, E., Raghinaru, D., Lum, J., Brown, S. A., Pinsker, J. E., Church, M. M., Levy, C., Lam, D., Kudva, Y. C., Buckingham, B., P., F. G., Wadwa, R. P., Laffel, L., Doyle, F. J., HansDevries, J., Renard, E., Cobelli, C., Boscari, F., Del Favero, S., and Kovatchev, B. P. (2019). The internationaldiabetes closed-loop study: Testing artificial pancreas component interoperability. *Diabetes Technology and Therapeutics*, 2:73–80.
- [ARM, 2011] ARM (2011). *ARM® Cortex™-A7 MPCore™ Processor Technical Reference Manual*. <https://developer.arm.com/ip-products/processors/cortex-a/cortex-a7>.
- [ARM, 2012a] ARM (2012a). *ARM® Cortex®-A15 MPCore™ Processor Technical Reference Manual*. <https://developer.arm.com/ip-products/processors/cortex-a/cortex-a15>.
- [ARM, 2012b] ARM (2012b). *ARM® Cortex®-A17 MPCore™ Processor Technical Reference Manual*. <https://developer.arm.com/ip-products/processors/cortex-a/cortex-a17>.
- [ARM, 2012c] ARM (2012c). *Arm® Cortex®-A53 MPCore Processor Technical Reference Manual*. <https://developer.arm.com/ip-products/processors/cortex-a/cortex-a53>.
- [ARM, 2016a] ARM (2016a). *ARM® Cortex®-A57 MPCore Processor Technical Reference Manual*. <https://developer.arm.com/ip-products/processors/cortex-a/cortex-a57>.
- [ARM, 2016b] ARM (2016b). *ARM® Cortex®-A72 MPCore Processor Technical Reference Manual*. <https://developer.arm.com/ip-products/processors/cortex-a/cortex-a72>.
- [ASUS, 2018] ASUS (2018). *Tinker Board S Tech specs*. <https://www.asus.com/us/Motherboards-Components/Single-Board-Computer/All-series/Tinker-Board-S/>.
- [Bergman et al., 1979] Bergman, R., Ider, Y., Bowden, C., and Cobelli, C. (1979). Quantitative estimation of insulin sensitivity. *The American journal of physiology*, 236:67–77.

- [Berian et al., 2019] Berian, J., Bravo, I., Gardel, A., Lázaro, J., and Hernández, S. (2019). A wearable closed-loop insulin delivery system based on low-power socs. *Electronics*, 8.
- [Chakrabarty et al., 2020] Chakrabarty, A., Healey, E., Shi, D., Zavitsanou, S., J., D. F., and Dassau, E. (2020). Embedded model predictive control for a wearable artificial pancreas. *IEEE TRANSACTIONS ON CONTROL SYSTEMS TECHNOLOGY*, 28:2600–2607.
- [Chakrabarty et al., 2018] Chakrabarty, A., Zavitsanou, S., Doyle, F., and Dassau, E. (2018). Event-triggered model predictive control for embedded artificial pancreas systems. *IEEE Transactions on Biomedical Engineering*, 65:575–586.
- [CO, 2017] CO, S. X. S. (2017). *Orange Pi Pc Plus Tech specs*. <http://www.orangepi.org/orangepipcplus/>.
- [Colmegna et al., 2018] Colmegna, P., Garelli, F., Battista, H. D., and Sánchez peña, R. (2018). Automatic regulatory control in type 1 diabetes without carbohydrate counting. *Control Engineering Practice*, 74:22–32.
- [CostHelper, 2013] CostHelper, I. (2013). *Insulin Pump Cost*. <https://health.costhelper.com/insulin-pump.html>.
- [Dalla Man et al., 2014] Dalla Man, C., Micheletto, F., Lv, D., Breton, M., Kovatchev, B., and Cobelli, C. (2014). The uva/padova type 1 diabetes simulator: New features. *Diabetes Science and Technology*, 8:26–34.
- [de Alto Costo, 2018] de Alto Costo, C. (2018). Dia mundial de la diabetes. boletín de información técnica especializada. Technical report, Cuenta de Alto Costo, Fondo colombiano de enfermedades de alto costo.
- [de Alto Costo, 2020] de Alto Costo, C. (2020). Cuenta de alto costo asamblea general ordinaria 2020 anexo técnico. Technical report, Cuenta de Alto Costo, Fondo colombiano de enfermedades de alto costo.
- [Dexcom, 2015] Dexcom (2015). *Sensor Dexcom G5*. <https://s3-us-west-2.amazonaws.com/dexcompdf/Downloads+and+Guides+Updates/LBL013358+G5+Mobile+UG+OUS+ES+mgdL.pdf>.
- [DougGuan, 2020] DougGuan, D. A. E. T. C. L. (2020). Report n°: Sa2007199m 01001 lithium polymer battery 955465. Technical report, DougGuan Anci Electronic Technology Co. Ltd.
- [Doyle et al., 2014] Doyle, F. J., Huyett, L. M., Lee, J. B., Z., C., H., and Dassau, E. (2014). Closed-loop artificial pancreas systems: Engineering the algorithms. *Diabetes Care*, 37:1191–1197.

- [EMC, 2020] EMC, E. T. R. (2020). Report n°: Htt202003477er model pisugar2-pro. Technical report, Shenzhen HTT Technology Co., Ltd.
- [Eqdami et al., 2010] Eqdami, A., Dimarogonas, D. V., and Kyriakopoulos, K. J. (2010). Event-triggered control for discrete-timesystems. *Proceedings of the 2010 American Control Conference*.
- [F. et al., 2020] F., V.-T. M., Caicedo, M. A., and Rivadeneira, P. S. (2020). Offset-free mpc strategy for nonzero regulation of linear impulsive systems. *ISA Transactions*, 101:91–101.
- [FOUNDATION, 2016] FOUNDATION, R. P. (2016). *Raspberry Pi 3 model B Tech specs*. <https://www.raspberrypi.org/products/raspberry-pi-3-model-b/>.
- [FOUNDATION, 2018] FOUNDATION, R. P. (2018). *Raspberry Pi 4 Tech specs*. <https://www.raspberrypi.org/products/raspberry-pi-4-model-b/specifications/>.
- [García, 2019] García, P. A. (2019). *Sistemas Embebidos de Tiempo Real con aplicaciones en Bioingeniería*. Tesis de doctorado, Universidad Nacional de la Plata.
- [Giselsson and Boyd, 2017] Giselsson, P. and Boyd, S. (2017). Linear convergence and metric selection for douglas–rachford splitting and admm. *IEEE Trans. Autom. Control*, 62:532–544.
- [Goldfarb and Idnani, 2011] Goldfarb, D. and Idnani, A. (2011). A numerically stable dual method for solving strictly convex quadratic programs. *Mathematical Programming*, 27:1–33.
- [Gondhalekar et al., 2016] Gondhalekar, R., Dassau, E., and Doyle, F. J. (2016). Periodic zone-mpc with asymmetric costs for outpatient-ready safety of an artificial pancreas to treat type 1 diabetes. *Automatica*, 71:237–246.
- [Gondhalekar et al., 2018] Gondhalekar, R., Dassau, E., and Doyle, F. J. (2018). Velocity-weighting & velocity-penalty mpc of an artificialpancreas: Improved safety & performance. *Automatica*, 91:105–117.
- [Grosman et al., 2010] Grosman, B., Dassau, E., Zisser, H. C., and Jovanović, L. and Doyle, F. J. (2010). Zone model predictive control: a strategy to minimize hyper- and hypoglycemic events. *J. Diabetes Sci. Technol.*, 4:961–975.
- [Hajizadeh et al., 2017] Hajizadeh, I., Rashid, M., Turksoy, K., Samadi, S., Feng, J., Frantz, N., Sevil, M., Cengiz, E., and Cinar, A. (2017). Plasma insulin estimation in people with type 1 diabetes mellitus. *Industrial and Engineering ChemistryResearch*, 56:9846–9857.
- [HardKernel, 2015] HardKernel, L. (2015). *ODROID-XU4 Tech specs*. <https://wiki.odroid.com/odroid-xu4/hardware/hardware>.

- [Hei et al., 2015] Hei, X., Du, X., Lin, S., Lee, I., and Sokolsky, O. (2015). Patient infusion pattern based access control schemes for wireless insulin pump system. *IEEE Transactions on Parallel and Distributed Systems* 26, pages 3108–3121.
- [Hoyos et al., 2021] Hoyos, J., Villa-Tamayo, M., Builes-Montaña, C., Ramirez-Rincón, A., Godoy, J., Carcia-Tirado, J., and Rivadeneira, P. (2021). Identifiability of control oriented glucose insulin linear models: review and analysis. *IEEE ACCESS*, pages 1–16.
- [Huyett et al., 2015] Huyett, L. M., Dassau, E., Zisser, H. C., and Doyle, F. J. (2015). Design and evaluation of a robust pid controller for a fully implantable artificial pancreas. *Industrial and Engineering Chemistry Research*, 54:10311–10321.
- [I+D, 2015] I+D, E. (2015). Tarjeta para interfaz usb - serial. Technical report, Electrónica I+D, Medellín.
- [IDF, 2019] IDF (2019). Idf diabetes atlas ninth edition. Technical report, International Diabetes Federation.
- [Intel, 2020] Intel (2020). New intel® core™ processors with intel® hybrid technology. Technical report, © Intel Corporation.
- [Johansen, 2017] Johansen, T. A. (2017). Toward dependable embedded model predictive control. *IEEE Systems Journal*, 11:1208–1219.
- [Jung, 2016] Jung, C. A. and Lee, S. J. (2016). Design of automatic insulin injection system with continuous glucose monitoring (cgm) signals. *3rd IEEE EMBS Int. Conf. Biomed. Heal. Informatics*, pages 102–105.
- [Kouzoupis et al., 2015] Kouzoupis, D., Ferreau, H. J., Peyrl, H., and Diehl, M. (2015). First-order methods in embedded nonlinear model predictive control. *2015 European Control Conference (ECC)*.
- [Lehmann et al., 2013] Lehmann, D., Henriksson, E., and Johansson, K. (2013). Event-triggered model predictive control of discrete-time linear systems subject to disturbances. *European Control Conference (ECC)*, page 1156–1161.
- [Li and Shi, 2014] Li, H. and Shi, Y. (2014). Event-triggered robust model predictive control of continuous-time nonlinear systems. *Automatica*, 50:1507–1513.
- [Limited, 2011] Limited, F. T. D. I. (2011). *Future Technology Devices International Ltd. FT232R USB UART IC Datasheet*. <https://www.raspberrypi.org/products/raspberry-pi-4-model-b/specifications/>.

- [Lopez-Valencia, 2021] Lopez-Valencia, A. (2021). Construcción de un prototipo de infusor de insulina e integración con un sensor de glucosa en sangre freestyle libre mediante comunicación inalámbrica. Master's thesis, Universidad Nacional de Colombia.
- [Maciejowski, 2001] Maciejowski, J. (2001). *Predictive Control with Constraints*. Pearson College Div 1er edición.
- [Magdelaine et al., 2015] Magdelaine, N., Chaillous, L., Guilhem, I., J.-Y., P., Krempf, M., Moog, C. H., and Le Carpentier, E. (2015). A long-term model of the glucose-insulin dynamics of type 1 diabetes. *IEEE TRANSACTIONS ON BIOMEDICAL ENGINEERING*, 62:1546–1552.
- [MathWorks, 2021a] MathWorks (2021a). *Hardware-In-The-Loop Simulation*. <https://la.mathworks.com/help/phymod/simscape/ug/what-is-hardware-in-the-loop-simulation.html>.
- [MathWorks, 2021b] MathWorks (2021b). Optimization toolbox documentation, quadprog. Technical report, The MathWorks, Inc.
- [Medtronic, 2017] Medtronic (2017). Minimed 640g system user guide. Technical report, The MathWorks, Inc, Los Angeles.
- [melexis microelectronic systems, 2006] melexis microelectronic systems (2006). *MLX90614 family Single and Dual Zone Infra Red Thermometer in TO-39*. <https://pdf1.alldatasheet.com/datasheet-pdf/view/218977/ETC2/MLX90614.html>.
- [NVIDIA, 2019] NVIDIA (2019). *Jetson Nano Tech specs*. <https://developer.nvidia.com/embedded/jetson-nano-2gb-developer-kit>.
- [operating as Diabetes UK, 2018] operating as Diabetes UK, T. B. D. A. (2018). *Insulin pumps*. <https://www.diabetes.org.uk/guide-to-diabetes/managing-your-diabetes/treating-your-diabetes/insulin-pumps>.
- [Pinsker et al., 2016] Pinsker, J. E., Lee, J. B., Dassau, E., Seborg, D. E., Bradley, P. K., Gondhalekar, R., Bevier, W. C., Huyett, L., Z., C., H., and Doyle, F. J. (2016). Randomized crossover comparison of personalized mpc and pid control algorithms for the artificial pancreas. *Diabetes Care*, 39:1135–1142.
- [Radziuk, 2012] Radziuk, J. (2012). The artificial pancreas. *Diabetes*, 61:2221–2224.
- [Rivadeneira et al., 2018] Rivadeneira, P. S., Ferramosca, A., and González, A. H. (2018). Control strategies for non-zero set-point regulation of linear impulsive systems. *IEEE Transactions on Automatic Control*, 63:2994–3001.

- [Ruan et al., 2017] Ruan, Y., Wilinska, M. E., Thabit, H., and Hovorka, R. (2017). Modeling day-to-day variability of glucose-insulin regulation over 12-week home use of closed-loop insulin delivery. *IEEE Transactions on Biomedical Engineering*, 64:1412–1419.
- [S.F., 2021] S.F., P. (2021). *Floating Point Arithmetic: Issues and Limitations*. <https://docs.python.org/3/tutorial/float.html>.
- [Shi et al., 2019] Shi, D., Dassau, E., and Doyle, F. J. (2019). Adaptive zone model predictive control of artificial pancreas based on glucose- and velocity-dependent control penalties. *IEEE transactions on biomedical engineering*, 66:1045–1054.
- [Simon, 2006] Simon, D. (2006). *Optimal State Estimation: Kalman, H Infinity, and Nonlinear Approaches*. Wiley-Interscience 1st Edición.
- [Sra et al., 2011] Sra, S., Nowozin, S., and Wright, J. (2011). *Optimization for Machine Learning*. Neural Information Processing series.
- [Stellato et al., 2020] Stellato, B., Banjac, G., Goulart, P., Bemporad, A., and Boyd, S. (2020). OSQP: an operator splitting solver for quadratic programs. *Mathematical Programming Computation*, 12:637–672.
- [Stellato et al., 2018] Stellato, B., Naik, V., Bemporad, A., Goulart, P., and Boyd, S. (2018). Embedded mixed-integer quadratic optimization using the osqp solver. *European Control Conference, ECC*, in press:1–6.
- [the Future, 2021] the Future, D. D. (2021). *Gravity: I2C Digital Wattmeter*. <https://wiki.dfrobot.com/Gravity>
- [Toffanin et al., 2019] Toffanin, C., Aiello, E. M., Del Favero, S., Cobelli, C., and Magni, L. (2019). Multiple models for artificial pancreas predictions identified from free-living condition data: A proof of concept study. *ProcessControl*, 77:29–37.
- [Vettoretti et al., 2018] Vettoretti, M., Fachinetti, A., Sparacino, G., and Cobelli, C. (2018). Type-1 diabetes patient decision simulator for in silico testing safety and effectiveness of insulin treatments. *IEEE Transactions on Biomedical Engineering*, 65:1281–1290.
- [Villa-tamayo, 2021] Villa-tamayo, M. (2021). Compensación del efecto de variaciones fisiológicas en la glucemia de pacientes diabéticos tipo 1 utilizando control predictivo con entradas impulsivas. Master’s thesis, Universidad Nacional de Colombia.
- [Villa-Tamayo et al., 2021a] Villa-Tamayo, M. F., Leon-Vargas, F., Garcia-Jaramillo, M., and Rivadeneira, P. S. (2021a). Glycemic control strategy based on an impulsive mpc with safety layer coupling for iob limitation. *IEEE Control Systems Letters*, 5(5):1669–1674.

- 
- [Villa-Tamayo et al., 2021b] Villa-Tamayo, M. F., Leon-Vargas, F., Garcia-Jaramillo, M., and Rivadeneira, P. S. (2021b). Glycemic control strategy based on an impulsive mpc with safety layer coupling for iob limitation. *IEEE Control Systems Letters*, pages 1669–1674.
- [Villa-Tamayo and Rivadeneira, 2020a] Villa-Tamayo, M. F. and Rivadeneira, P. S. (2020a). Adaptive impulsive offset-free mpc to handle parameter variations for type 1 diabetes treatment. *Ind. Eng. Chem. Res.*, 59(13):5865–5876.
- [Villa-Tamayo and Rivadeneira, 2020b] Villa-Tamayo, M. F. and Rivadeneira, P. S. (2020b). Adaptive impulsive offset-free mpc to handle parameter variations for type 1 diabetes treatment. *Ind. Eng. Chem. Res.*, 59:5865–5876.
- [Zavitsanou et al., 2016] Zavitsanou, S., Chakrabarty, A., Dassau, E., and Doyle, F. J. (2016). Embedded control in wearable medical devices: Application to the artificial pancreas. *Processes*, 4.