

Control Predictivo Robusto de Glucosa en Pacientes con Diabetes Mellitus Tipo I: Validación in
Silico

David Alberto Padilla Toloza

Trabajo de Grado para optar al título de Magister en Ingeniería Electrónica

Director

PhD. Rodolfo Villamizar Mejía

Codirector

PhD. Jose Jorge Carreño Zagarra

Universidad Industrial de Santander

Facultad de Ingenierías Fisicomecánicas

Escuela de Ingenierías Eléctrica, Electrónica y de Telecomunicaciones

Bucaramanga

2021

Tabla de Contenido

Introducción	20
1. Objetivos	22
2. Producción académica	23
3. Marco conceptual	25
3.1. Diabetes	25
3.1.1. Diabetes a nivel mundial	25
3.1.2. ¿Qué es la diabetes?	27
3.1.3. Diabetes Mellitus Tipo I	29
3.1.4. Modelo Uva Padova	30
3.1.5. Descripción del modelo diabético	31
3.2. Control predictivo basado en el modelo (MPC)	41
3.2.1. Modelo de predicción	42
3.2.2. Horizonte de predicción	43
3.2.3. Horizonte de control	44
3.2.4. Función objetivo	45
3.2.5. Matrices de ponderación	47

3.2.6. Ley de control	48
3.3. Estimador de perturbaciones	55
3.3.1. Control feed-forward	55
3.3.2. Clasificadores	56
3.4. Observadores de estados	56
3.4.1. Filtro de Kalman	58
3.4.2. Filtro de Kalman extendido	60
4. Esquema de control propuesto	63
4.1. Problema de control	63
4.2. Propuesta de control	64
5. Diseño de sistema de control	65
5.1. Cálculo de perfil de glucosa saludable	65
5.2. MPC con modelo desacoplado	67
5.3. MPC con modelo acoplado	70
5.4. Estimador de perturbaciones	75
5.4.1. Selección de predictores	75
5.4.2. Adecuación de señales	80
5.4.3. Selección del clasificador	83
6. Validación in silico	84

6.1. Modelos in silico	84
6.2. Sintonización de Controladores	88
6.2.1. Pruebas con modelo nominal	99
6.2.2. Pruebas con modelo incierto	109
6.3. Validación del esquema de control completo	115
6.3.1. Sintonización del estimador	115
6.3.2. Pruebas con modelo nominal	116
6.3.3. Pruebas con modelo incierto	124
7. Trabajo futuro	131
8. Conclusiones	132
Referencias Bibliográficas	135
Apéndices	138

Lista de Figuras

Figura 1.	Cantidad total calculada de adultos de 20 a 79 años de edad con diabetes en el año 2019. Nota: Tomado de (Federation, 2019)	26
Figura 2.	Prueba de tolerancia a la glucosa oral. Nota: Tomado de (Barrett, 2019)	29
Figura 3.	Sistema de bloques de Simulink que modela el subsistema de glucosa.	32
Figura 4.	Sistema de bloques de Simulink que modela el subsistema de insulina.	33
Figura 5.	Sistema de bloques de Simulink que modela el paso de los alimentos por el tracto digestivo.	35
Figura 6.	Sistema de bloques de Simulink que modela la producción endógena de glucosa realizada por el hígado.	36
Figura 7.	Sistema de bloques de Simulink que modela toda la utilización de glucosa en el paciente.	37
Figura 8.	Sistema de bloques de Simulink que modela los riñones.	38
Figura 9.	Sistema de bloques de Simulink que modela la absorción de insulina exógena.	39
Figura 10.	Sistema de bloques de Simulink que modela la glucosa que pasa del plasma al tejido subcutáneo.	39
Figura 11.	Sistema de bloques de Simulink que modela el subsistema de glucagón.	41
Figura 12.	Esquema aplicado por el algoritmo de control predictivo para calcular las acciones de control.	42

- Figura 13. Predicciones de la salida a lo largo de un horizonte de predicción definido, aplicando un horizonte de control a la acción de control. 45
- Figura 14. Sistema de control en lazo cerrado con acción feed-forward. 56
- Figura 15. Sistema de control en espacio de estados con observador. 58
- Figura 16. Esquema de control. 64
- Figura 17. Comparación de la referencia generada a partir de la mejor aproximación de la regresión con la respuesta de glucosa generada por el bolo óptimo para el paciente Adult_001. 67
- Figura 18. Comparación de la referencia generada a partir de la regresión redefinida con la respuesta de glucosa generada por el bolo óptimo para el paciente Adult_001. 68
- Figura 19. Algoritmo de proyección para iteración k . 72
- Figura 20. Algoritmo de proyección para iteración $k + 1$. 73
- Figura 21. Comparación entre el control de glucosa con información de perturbación real (Ideal) e información retardada 5, 10, 15, 20 y 25 minutos. 77
- Figura 22. Comparación entre la perturbación de carbohidratos retardada 15 minutos (respuesta) con la señal de glucosa y sus dos primeras derivadas respectivamente normalizadas (predictores). 77
- Figura 23. Predictores calculados sin sistema de adecuación para una ingesta de alimentos de 60, 80 y 50 [g] de carbohidratos a las 6 a.m., 12 m. y 6 p.m. respectivamente para el paciente *adult_001*. 79

- Figura 24. Predictores calculados usando el sistema de adecuación para una ingesta de 60, 80 y 50 [g] de carbohidratos a las 6 a.m., 12 m. y 6 p.m. respectivamente para el paciente *adult_001*. 82
- Figura 25. Matriz de Butcher correspondiente al método Dorman-Prince. 86
- Figura 26. Gráfica generada por el algoritmo de optimización para la clase *MPC_QDMC* simulando una ingesta de alimentos de 50, 80 y 60 [g] de carbohidratos a las 6 a.m., 12 m. y 6 p.m. respectivamente para el paciente *adult_001*. 91
- Figura 27. Gráfica generada por el algoritmo de optimización para la clase *MPC_QDMC2* simulando una ingesta de alimentos de 50, 80 y 60 [g] de carbohidratos a las 6 a.m., 12 m. y 6 p.m. respectivamente para el paciente *adult_001*. 93
- Figura 28. Gráfica generada por el algoritmo de optimización para la clase *MPC_QDMC2* simulando una ingesta de alimentos de 50, 80 y 60 [g] de carbohidratos a las 6 a.m., 12 m. y 6 p.m. respectivamente para el paciente *adult_001*. 96
- Figura 29. Gráfica generada por el algoritmo de optimización para la clase *MPC_QDMC2* simulando una ingesta de alimentos de 50, 80 y 60 [g] de carbohidratos a las 6 a.m., 12 m. y 6 p.m. respectivamente para el paciente *adult_001*. 98
- Figura 30. Resultados del control mediante bolo óptimo del simulador T1DMS ante una ingesta de alimentos de 50, 80 y 60 [g] de carbohidratos a las 6 a.m., 12 m. y 6 p.m. respectivamente para los pacientes adultos in silico. 100

- Figura 31. Gráfica generada por el algoritmo de optimización para la clase *MPC_QDMC2* simulando una ingesta de alimentos de 50, 80 y 60 [g] de carbohidratos a las 6 a.m., 12 m. y 6 p.m. respectivamente para todo el banco de pacientes adultos el modelo nominal de cada uno. 103
- Figura 32. Gráfica generada por el algoritmo de optimización para la clase *MPC_RP* simulando una ingesta de alimentos de 50, 80 y 60 [g] de carbohidratos a las 6 a.m., 12 m. y 6 p.m. respectivamente para todo el banco de pacientes adultos el modelo nominal de cada uno. 105
- Figura 33. Gráfica generada por el algoritmo de optimización para la clase *MPC_QDMC2* simulando una ingesta de alimentos de 50, 80 y 60 [g] de carbohidratos a las 6 a.m., 12 m. y 6 p.m. respectivamente para todo el banco de pacientes adultos usando el modelo de predicción *adult_001* para todos. 109
- Figura 34. Gráfica generada por el algoritmo de optimización para la clase *MPC_QDMC2* simulando una ingesta de alimentos de 50, 80 y 60 [g] de carbohidratos a las 6 a.m., 12 m. y 6 p.m. respectivamente para todo el banco de pacientes adultos usando el modelo de predicción *adult_001* para todos. 111
- Figura 35. Resultados de estimación de perturbación y generación de referencia para una ingesta de 60, 80 y 50 [g] de carbohidratos a las 6 a.m., 12 m. y 6 p.m. respectivamente para el paciente *adult_001* usando un controlador basado en la clase *MPC_QDMC2*. 116

- Figura 36. Resultados de estimación de perturbación y generación de referencia para una ingesta de 60, 80 y 50 [g] de carbohidratos a las 6 a.m., 12 m. y 6 p.m. respectivamente para el paciente *adult_001* usando un controlador basado en la clase *MPC_QDMC2* y el estimador de perturbaciones. 117
- Figura 37. Gráfica generada por el algoritmo de optimización para la clase *MPC_QDMC2* simulando una ingesta de alimentos de 50, 80 y 60 [g] de carbohidratos a las 6 a.m., 12 m. y 6 p.m. respectivamente para todo el banco de pacientes adultos el modelo nominal de cada uno. 118
- Figura 38. Gráfica generada por el algoritmo de optimización para la clase *MPC_RP* simulando una ingesta de alimentos de 50, 80 y 60 [g] de carbohidratos a las 6 a.m., 12 m. y 6 p.m. respectivamente para todo el banco de pacientes adultos el modelo nominal de cada uno. 120
- Figura 39. Gráfica generada por el algoritmo de optimización para la clase *MPC_QDMC2* simulando una ingesta de alimentos de 50, 80 y 60 [g] de carbohidratos a las 6 a.m., 12 m. y 6 p.m. respectivamente para todo el banco de pacientes adultos usando el modelo de predicción *adult_001* para todos. 124
- Figura 40. Gráfica generada por el algoritmo de optimización para la clase *MPC_QDMC2* simulando una ingesta de alimentos de 50, 80 y 60 [g] de carbohidratos a las 6 a.m., 12 m. y 6 p.m. respectivamente para todo el banco de pacientes adultos usando el modelo de predicción *adult_001* para todos. 126

Figura 41.	Subsistema de Simulink que contiene los sistemas de bloques descritos en la sección Modelo Diabético.	139
Figura 42.	Archivo que contiene la clase UVA_PADOVA_2 en Matlab.	140
Figura 43.	Subsistema de Simulink que contiene el Matlab System en el que se instancia el objeto del modelo del paciente diabético.	140
Figura 44.	Clase UVA_PADOVA_2 instanciada en el bloque Matlab System de Simulink.	141
Figura 45.	Parámetros configurables para el objeto del modelo diabético.	141
Figura 46.	Instanciación, configuración de parámetros y uso como función de la clase UVA_PADOVA_2 en MATLAB.	142
Figura 47.	Archivo que contiene la clase MPC_QDMC en Matlab.	142
Figura 48.	Clase MPC_QDMC instanciada en el bloque Matlab System de Simulink.	143
Figura 49.	Configuración de parámetros de la clase MPC_QDMC.	144
Figura 50.	Instanciación, configuración de parámetros y uso como función de la clase MPC_QDMC en MATLAB.	145
Figura 51.	Archivo que contiene la clase MPC_RP en Matlab.	145
Figura 52.	Archivo que contiene la clase MPC_RPI en Matlab.	145
Figura 53.	Archivo que contiene la clase EKF_UVA en Matlab.	146
Figura 54.	Clase <i>EKF_UVA</i> instanciada en el bloque Matlab System de Simulink.	146
Figura 55.	Configuración de parámetros de la clase EKF_UVA.	147
Figura 56.	Archivo que contiene la clase ESTIMADOR_CHO en Matlab.	147
Figura 57.	Clase ESTIMADOR_CHO instanciada en el bloque Matlab System de Simulink.	147

- Figura 58. Parámetros configurables del objeto de la clase ESTIMADOR_CHO de señales en Simulink. 148
- Figura 59. Instanciación, configuración de parámetros y uso como función de la clase ESTIMADOR_CHO en MATLAB. 148
- Figura 60. Archivo que contiene la clase KALMAN_FMM en Matlab. 148
- Figura 61. Clase KALMA_FMM instanciada en el bloque Matlab System de Simulink. 149
- Figura 62. Parámetros configurables del objeto de la clase KALMAN_FMM de señales en Simulink. 149
- Figura 63. Instanciación, configuración de parámetros y uso como función de la clase KALMAN_FMM en MATLAB. 150
- Figura 64. contenido del bloque sensor de glucosa. 151
- Figura 65. Contenido del bloque bomba de insulina. 151
- Figura 66. Ingesta de 60, 80 y 50 gramos de carbohidratos a las 7 a.m, 12 m y 6p.m respectivamente. 152

Lista de Tablas

Tabla 1.	Estimaciones y proyecciones mundiales de diabetes Federation (2019).	27
Tabla 2.	Criterios de diagnóstico para la diabetes	28
Tabla 3.	Resultados de error de cantidad, forma en la estimación de carbohidratos y forma de la referencia usando distintas combinaciones de predictores	78
Tabla 4.	Resultados de error de cantidad, forma en la estimación de carbonidratos y forma de la referencia usando distintos clasificadores	83
Tabla 5.	H_p , H_u , Q y R óptimos, usando el algoritmo de optimización por búsqueda ciega	90
Tabla 6.	Parámetros óptimos H_p , H_u , Q and R usando el algoritmo de optimización PSO para el controlador basado en la clase MPC_QDMC	95
Tabla 7.	Parámetros óptimos H_p , H_u , Q and R usando el algoritmo de optimización PSO para el controlador basado en la clase MPC_RP	97
Tabla 8.	Medidas de hiperglucemia aplicando control mediante bolo óptimo al banco de adultos in silico, comparación de pico de glucosa (PG) en $[mg/dL]$ y tiempo del paciente en hiperglucemia en minutos.	101
Tabla 9.	Medidas de hiperglucemia aplicando control basado en la clase MPC_QDMC2 al banco de adultos in silico usando modelo nominal, comparación de pico de glucosa en $[mg/dL]$ y tiempo del paciente en hiperglucemia en minutos.	104

Tabla 10.	Medidas de hiperglucemia aplicando control basado en la clase <i>MPC_RP</i> al banco de adultos in silico usando modelo nominal, comparación de pico de glucosa (PG) en $[mg/dL]$ y tiempo del paciente en hiperglucemia en minutos.	106
Tabla 11.	Valores IAEP de los controladores usando modelo nominal	107
Tabla 12.	Unidades de insulina total administrada al paciente durante el día	108
Tabla 13.	Medidas de hiperglucemia aplicando control basado en la clase <i>MPC_QDMC2</i> al banco de adultos in silico con modelo incierto, comparación de pico de glucosa en $[mg/dL]$ y tiempo del paciente en hiperglucemia en minutos.	110
Tabla 14.	Medidas de hiperglucemia aplicando control basado en la clase <i>MPC_RP</i> al banco de adultos in silico con modelo incierto, comparación de pico de glucosa (PG) en mg/dL y tiempo del paciente en hiperglucemia en minutos.	112
Tabla 15.	Valores IAEP de los controladores usando modelo nominal	113
Tabla 16.	Unidades de insulina total administrada al paciente durante el día	114
Tabla 17.	Medidas de hiperglucemia aplicando controlador basado en la clase <i>MPC_QDMC2</i> , comparación de pico de glucosa (PG) en $[mg/dL]$ y tiempo del paciente en hiperglucemia en minutos.	119
Tabla 18.	Medidas de hiperglucemia aplicando controlador basado en la clase <i>MPC_QDMC2</i> , comparación de pico de glucosa (PG) en $[mg/dL]$ y tiempo del paciente en hiperglucemia en minutos.	121
Tabla 19.	Valores IAEP de los controladores usando modelo nominal	122
Tabla 20.	Unidades de insulina total administrada al paciente durante el día	123

- Tabla 21. Medidas de hiperglucemia aplicando controlador basado en la clase *MPC_QDMC2*, comparación de pico de glucosa (PG) en [mg/dL] y tiempo del paciente en hiperglucemia en minutos. 125
- Tabla 22. Medidas de hiperglucemia aplicando controlador basado en la clase *MPC_QDMC2*, comparación de pico de glucosa (PG) en [mg/dL] y tiempo del paciente en hiperglucemia en minutos. 127
- Tabla 23. Valores IAEP de los controladores usando modelo nominal 128
- Tabla 24. Unidades de insulina total administrada al paciente durante el día 129

Lista de Apéndices

	pág.
Apéndice A. Programación orientada a objetos	138
Apéndice B. Hardware	151

Glosario

CHO carbohidratos consumidos durante la ingesta de alimentos.

DMC Dinamic Matrix Control - Control de matriz dinámica. Algoritmo de control predictivo que se basa en la respuesta al escalón del modelo para hacer predicciones.

FDA federación de drogas y alimentos. Agencia gubernamental Estadounidense que es responsable de la regulación de alimentos, medicamentos, cosméticos, aparatos médicos, productos biológicos y derivados sanguíneos.

FID federación internacional de la diabetes.

Hardware in the loop técnica de desarrollo y prototipado rápido de controladores, que consiste en simular toda la dinámica de un proceso por medio de una computadora para cerrar el lazo con un microcontrolador programado que se desee validar.

Hessiana matriz cuadrada que agrupa las derivadas parciales de un sistema de ecuaciones.

In silico que se realiza en simulación por computadora.

Incierto Modelo del que no se sabe con certeza el valor de sus parámetros.

MPC control predictivo basado en el modelo. Estrategia de control de control avanzado que usa predicciones del comportamiento de un proceso para calcular una acción de control óptima, es popular debido al manejo que da a las restricciones del proceso.

Nominal modelo del que se conocen todos los parámetros.

Observador algoritmo que permite estimar las variables de un modelo en espacio de estados a partir de las mediciones de algunas de estas.

Predictores variables de entrada usadas por un clasificador para generar una salida.

PSO Particle Swarm Optimización - Optimización por enjambre de partículas. Método heurístico para resolver un problema de optimización que se basa en el comportamiento de las partículas.

QDMC Quadratic Dinamic Matrix Control - Control de matriz dinámica cuadrático. Algoritmo de control predictivo que se basa en la respuesta al escalón del modelo para hacer predicciones, utiliza un algoritmo de programación cuadrática para calcular la acción de control.

Robusto que es estable y presenta un transitorio adecuado en presencia de incertidumbre paramétrica, ruido y/o perturbaciones.

T1DM Type I Diabetetes Mellitus - Diabetes mellitus tipo I. Tipo de diabetes, también llamada insulino dependiente debido a que el paciente requiere de dosificación externa de insulina.

T1DMS Type I Diabetetes Mellitus Simulator - Simulador de diabetes mellitus tipo I. Simulador desarrollado en MATLAB que contiene las ecuaciones del modelo desarrollado por Boris Kovatchev cuyos parámetros fueron sintonizados para generar el modelo UVA PADOVA.

Resumen

Título: Control Predictivo Robusto de Glucosa en Pacientes con Diabetes Mellitus Tipo I: Validación in Silico *

Autor: David Alberto Padilla Toloza **

Palabras Clave: Control predictivo, Estimación de carbohidratos, In Silico, Uva Padova.

Descripción: En este proyecto de grado se propone un esquema de control para la dosificación de insulina en pacientes T1DM, que mantenga un perfil de glucosa saludable, evitando condiciones de hiperglucemia e hipoglucemia, cuyo modelo dinámico presenta fuertes retardos, no linealidades e incertidumbre paramétrica. Para enfrentar este problema de control se planteó la combinación de un controlador predictivo para seguir una referencia con un estimador basado en clasificadores. El controlador predictivo consta dos partes. La primera, cancela los efectos de la perturbación en el momento en que es detectada, aplicando una acción feed-forward; la segunda calcula la acción de control necesaria para seguir una referencia, que busca que la glucosa regrese a su estado basal imitando un perfil saludable para el paciente. Finalmente, se propusieron tres esquemas de controlador predictivo, el primero basado en el algoritmo QDMC, el segundo basado en realimentación de estados con modificaciones en el cálculo de las predicciones y el tercero es una versión del segundo con integrador. El esquema de control propuesto busca ser apto para la implementación real, por lo que se consideran efectos del hardware tales como, cuantización, muestreo, ruido y saturación.

Se validaron in silico los tres controladores usando modelos dinámicos dados por el simulador T1DMS, cuyo modelo es aceptado por la FDA, para una cohorte de 10 pacientes adultos. Dicha validación se hizo para modelo nominal e incierto, con el fin de medir el desempeño dinámico y robustez de los controladores. Se encontró que el desempeño dinámico de los dos primeros controladores fue adecuado, mientras que el desempeño del tercero no asegura estabilidad. Finalmente, el estimador demostró ser robusto ante la presencia de ruido y el error de estimación de carbohidratos fue menor al 10%, sin embargo, los 15 minutos que toma en estimar afectan el desempeño del segundo controlador.

* Trabajo de maestría

** Facultad de Ingenierías Físico-Mecánicas. Escuela de Ingenierías Eléctrica, Electrónica y telecomunicaciones. Director: PhD. Rodolfo Villamizar Mejía. Codirector: PhD. Jose Jorge Carreño Zagarra

Abstract

Title: Robust Predictive Glucose Control in T1DM Patients: Validation in Silico. *

Author: David Alberto Padilla Toloza **

Keywords: Predictive control, Carbohydrate estimation, In Silico, Uva Padova.

Description: In this degree project, a control scheme is proposed for insulin dosing in T1DM patients, which maintains a healthy glucose profile, avoiding hyperglycemic and hypoglycemic conditions, whose dynamic model presents strong delays, non-linearities and parametric uncertainty. To face this control problem, the combination of a predictive controller was proposed to follow a reference with an estimator based on classifiers. The predictive controller has two parts. The first, cancels the effects of the disturbance at the moment it is detected, applying a feedforward action; the second calculates the control action necessary to follow a reference, which seeks to return glucose to its baseline state, mimicking a healthy glucose profile for the patient. Finally, three predictive controller schemes were proposed, the first based on the QDMC algorithm, the second based on state feedback with modifications in the calculation of predictions, and the third is a version of the second with integrator. The proposed control scheme seeks to be suitable for the real implementation, for which hardware effects such as quantization, sampling, noise and saturation are considered.

The three controllers were validated in silico using dynamic models given by the type I diabetes mellitus simulator (T1DMS), whose model is accepted by the FDA, for a cohort of 10 adult patients. This validation was made for a nominal and uncertain model, in order to measure the dynamic performance and robustness of the controllers. The dynamic performance of the first two controllers was found to be adequate, while the performance of the third does not ensure stability. Finally, the estimator proved to be robust in the presence of noise and the carbohydrate estimation error was less than 10%, however, the 15 minutes it takes to estimate affect the performance of the second controller.

* Master Thesis

** Facultad de Ingenierías Físico-Mecánicas. Escuela de Ingenierías Eléctrica, Electrónica y telecomunicaciones. Director: PhD. Rodolfo Villamizar Mejía. Codirector: PhD. Jose Jorge Carreño Zagarra

Introducción

El control de variables para automatizar procesos es un reto cada día más ambicioso, llegando incluso a escenarios en los que no se espera que una maquina pueda realizar todo el trabajo sin supervisión alguna. Esto ha representado un enorme desarrollo industrial, que a generado todo tipo de dispositivos que han facilitado el diario vivir de la población. Un ejemplo interesante de estos escenarios inesperados es el área de la salud, un tema de interés mundial ya que aún existen enfermedades y condiciones del cuerpo humano que no se han podido tratar o curar mediante fármacos o procesos químicos. En estos casos la electrónica, la automatización y el control han desempeñado un papel fundamental generando un área de investigación que busca recuperar algunas funciones que el cuerpo humano ha perdido, por medio del desarrollo de nuevas tecnologías.

La diabetes mellitus tipo I (T1DM) es una patología caracterizada por altos niveles de glucosa en sangre de quien la padece. Esta condición se denomina hiperglucemia y es debida a un defecto absoluto en la secreción de insulina por parte de las células beta del páncreas, que no es reparable y hace que la enfermedad sea para toda la vida. Por esto, se han tomado diversas medidas, donde la administración de insulina exógena en el paciente es el tratamiento normal y más aceptado por los especialistas del área Federation (2019); Barrett (2019).

Hoy en día la insulina se dosifica al paciente por medio de inyecciones periódicas o bombas de infusión de insulina programables continuas o periódicas. Este trabajo de investigación se cen-

tra en estas últimas, en donde se busca una estrategia de dosificación automática basada en control realimentado que genere en un paciente con diabetes mellitus tipo I perfiles de glucosa similares a los de un sujeto sano al momento de ingerir un alimento, evitando niveles de hipoglucemia e hiperglucemia y emulando el comportamiento dinámico del páncreas Bequette (2012); Cobelli et al. (2009, 2011).

Para el diseño del algoritmo de dosificación de insulina se ha estudiado y revisado en la literatura, la técnica de control predictivo como una herramienta potencial para lograr un buen desempeño dinámico de este tipo de sistemas cuya dinámica es lenta, con grandes periodos de muestreo, lo que facilita realizar cálculos predictivos complejos entre instantes de control. Adicionalmente, la ley de control predictivo se basa en condiciones de referencia futuras, por tanto, se ajusta al problema de control bajo estudio donde se conoce de antemano la referencia que quiere seguirse para cada ingesta de comida, sujeto a las restricciones de infusión mediante bombas de insulina.

1. Objetivos

Objetivo general

Generar en pacientes con diabetes mellitus tipo I (T1DM) perfiles de glucosa en sangre saludables, a partir de un algoritmo de control predictivo robusto que calcule la dosificación de insulina y estime la ingesta de glucosa, teniendo en cuenta restricciones del sensor, actuador, así como en los niveles de glucosa e insulina.

Objetivos específicos

Estimar la ingesta de glucosa de pacientes in silico con diabetes mellitus tipo I, a partir de estimadores basados en técnicas de inteligencia artificial, con el fin de generar una señal de referencia que represente una condición saludable para el paciente;

Diseñar controladores predictivos robustos con capacidad de seguir una referencia de glucosa saludable, para la cohorte de pacientes adultos del simulador UVA PADOVA;

Validar in silico el conjunto controlador de glucosa/estimador de ingesta de carbohidratos, usando la cohorte de pacientes adultos del simulador UVA PADOVA.

2. Producción académica

Conferencias internacionales

7th International Conference on Control, Decision and Information Technologies (CoDIT). Junio 27 - 29, 2020, Praga, Republica Checa. Artículo presentado: Following glucose profiles of T1DM UVA/Padova in silico patients, by using estimated carbohydrate intake in a MPC controller. Autores: D. Padilla and R. Villamizar.

Proyectos de pregrado (Finalizados)

Simulación en tiempo real de la dinámica de glucosa para diferentes pacientes con diabetes mellitus tipo I, orientada a implementación con ?hardware in the loop?. Ingeniería Electrónica, UIS, 2018. Personas orientadas: Rafael Ely Rodriguez Iguaran. Trabajo dirigido como cotutor.

Control predictivo de glucosa en pacientes con diabetes mellitus tipo I: validación en tiempo real del controlador sobre modelos T1DM. Ingeniería Electrónica, UIS, 2018. Personas orientadas: Natalia Tarazona Cordoba y Leidy Lorena Arias Cabeza. Trabajo dirigido como cotutor.

Estrategias de control óptimo y no lineal para seguimiento de trayectorias de glucosa postprandiales en un paciente con diabetes mellitus tipo I, a partir de la estimación de ingesta de glucosa. Ingeniería Electrónica, UIS, 2019. Personas orientadas: Edward Alfonso Rodriguez Moreno y Juliana Andrea Díaz Blanco. Trabajo dirigido como cotutor.

Proyectos de pregrado (En curso)

Validación en Tiempo Real de un Controlador Predictivo de Glucosa sobre Pacientes INSILICO. Ingeniería Electrónica, UIS. Personas orientadas: Julieth Ximena Arias Guzman y Santiago Andres Vergara Hernandez. Trabajo dirigido como cotutor.

3. Marco conceptual

3.1. Diabetes

3.1.1. Diabetes a nivel mundial. Según los resultados registrados en la novena edición del Atlas de la diabetes de la FID, en el año 2019 se calculó que aproximadamente el 9.3 % de la población de adultos a nivel mundial padecía diabetes (463 millones de adultos entre 20 y 79 años de edad) y se prevé que esta cifra se eleve a 10.2 % para el año 2030 y a 10.9 % para el año 2045 Federation (2019). En la figura 1 se muestra un mapa mundi tomado del Atlas de la diabetes donde se registra la cantidad aproximada de adultos con diabetes por país en el 2019. En la imagen es claro que la región de América presenta una alta prevalencia de la enfermedad y que, específicamente en Colombia registra que de 1 a 10 millones de adultos tienen diabetes, un indicador preocupante debido a que en 2019 el DANE registró un total de 48.2 millones de colombianos, lo que en cifras porcentuales representa que de un 2 a un 20 % de la población del país son adultos con diabetes.

En cuanto a la población de personas menores de 20 años, el Atlas registra que en niños y adolescentes se calcula que 1.1 millones padecen diabetes mellitus tipo 1. Sin embargo, esta cifra se vio opacada por los 20.4 millones de nacimientos vivos en el 2019 que resultaron afectados por la hiperglucemia en el embarazo, representando el 15.8 % de los nacimientos vivos totales. Cabe recalcar fue imposible calcular la cantidad de personas en esta población que padecen diabetes mellitus tipo 2 Federation (2019).

Teniendo en cuenta todas estas cifras y otras más mencionadas en el documento, se calculó que el gasto anual en salud a nivel mundial para ese año fue de 760 mil millones de dolares estadou-

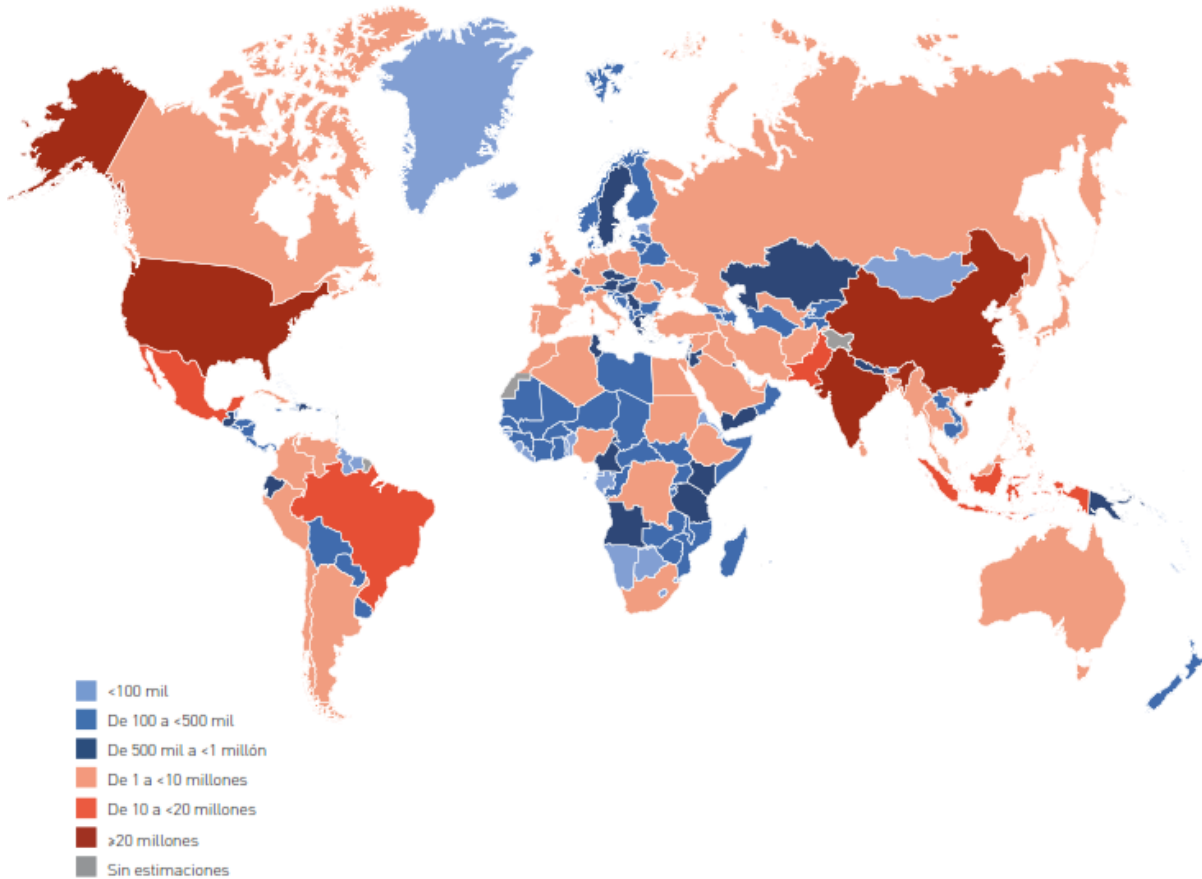


Figura 1. Cantidad total calculada de adultos de 20 a 79 años de edad con diabetes en el año 2019. Nota: Tomado de (Federation, 2019)

nidenses, lo que ahora mismo representa aproximadamente 3 mil billones de pesos colombianos Federation (2019).

La tabla 1 muestra las proyecciones y estimaciones basadas en los estudios realizados en el Atlas de la diabetes para su última edición.

Tabla 1

Estimaciones y proyecciones mundiales de diabetes Federation (2019).

DESCRIPCIÓN	2019	2030	2045
Total de población mundial [millones]	7.7 mil	8.6 mil	9.5 mil
Población adulta (20 - 79 años) [millones]	5.0 mil	5.7 mil	6.4 mil
Diabetes (20 - 79 años)			
Prevalencia mundial	9.3 %	10.2 %	10.9 %
Número de personas con diabetes [millones]	463.0	578.4	700.2
Número total de muertes por diabetes [millones]	4.2	–	–
Total de gastos en salud para la diabetes [millones de USD]	760.3 mil	–	–
Hiperglucemia en el embarazo (20 - 49 años)			
Proporción de nacidos vivos afectados	15.8 %	14.0 %	13.3 %
Cantidad de nacidos vivos afectados [millones]	20.4	18.3	18.0
Diabetes tipo I (0 - 19 años)			
Número de niños y adolescentes con diabetes tipo I	1.110.000	–	–
Cantidad de diagnósticos nuevos por año	128.900	–	–

3.1.2. ¿Qué es la diabetes?. La diabetes mellitus, llamada regularmente solo diabetes, es una patología que afecta la forma en que el organismo regula la glucosa en la sangre de una persona, ocasionado daños graves en su salud a largo plazo.

Se caracteriza principalmente por diagnosticar frecuentemente altos niveles de glucosa en sangre en el paciente, condición denominada hiperglucemia. Esto puede deberse a que no se produce suficiente insulina o a que las células del cuerpo presentan cierta resistencia a la insulina que se produce, que es una hormona fundamental para el metabolismo de proteínas y grasas, encargada de permitir la entrada de glucosa a las células. Este déficit de insulina puede llevar a la persona

afectada a presentar enfermedades cardiovasculares, lesiones en los nervios, enfermedades renales y afecciones oculares Federation (2019). En la tabla 2 se muestran criterios de diagnóstico para la diabetes, esta enfermedad se diagnostica si el paciente cumple uno o más de los criterios citados.

Tabla 2

Criterios de diagnóstico para la diabetes

Prueba	Criterio
Glucosa en sangre en ayunas (GSA)	$\geq 126\text{mg/dL}$
Prueba de tolerancia a la glucosa oral (PTGO)*	$\geq 200\text{mg/dL}$
Hemoglobina A1c**	$\geq 48\text{mmol/mol}$
Glucosa en plasma aleatoria	$> 200\text{mg/dL}$

* Debe medirse la glucosa del paciente tras haber ingerido una solución glucosada que contenga el equivalente a 75g de glucosa anhidra disuelta en agua Federation (2019).

** La prueba HbA1c debe realizarse en laboratorios certificados por entidades encargadas del estudio y tratamiento de la diabetes Federation (2019).

Existen diversos tipos de diabetes mellitus, de los que cabe resaltar 2 principalmente: Diabetes mellitus tipo I o insulino dependiente, condición en la que el paciente depende de la administración externa de insulina ya que su cuerpo deja de producirla; y diabetes mellitus tipo 2 o insulino resistente, donde a pesar de que existe producción de insulina en el paciente, la insulina producida no sirve para regular la glucosa. Las técnicas aplicadas en este proyecto para dosificar correctamente la inyección de insulina, se diseñaron únicamente para pacientes insulino dependientes, debido a que no existe producción de insulina por parte del páncreas, lo que permite que el control de glucosa dependa únicamente de la fuente externa de insulina, es decir, el sistema de control propuesto. Por lo tanto, únicamente se abordaran los conceptos relacionados a esta última en el documento.

3.1.3. Diabetes Mellitus Tipo I. También conocida como diabetes mellitus insulino dependiente, corresponde a una enfermedad metabólica caracterizada por altos niveles de glucemia en plasma (hiperglucemia), debida a la destrucción auto-inmunitaria de las células beta de los islotes pancreáticos, este fenómeno ocasiona un defecto absoluto en la secreción de insulina y por lo tanto de regulación de glucosa plasmática en la persona que la padece (Barrett, 2019; Federation, 2019). En la figura 2 se muestran las curvas de respuesta de glucosa de un sujeto sano y una persona con diabetes mellitus tipo I a la prueba de tolerancia a la glucosa oral, una manera eficaz de diagnosticar esta patología.

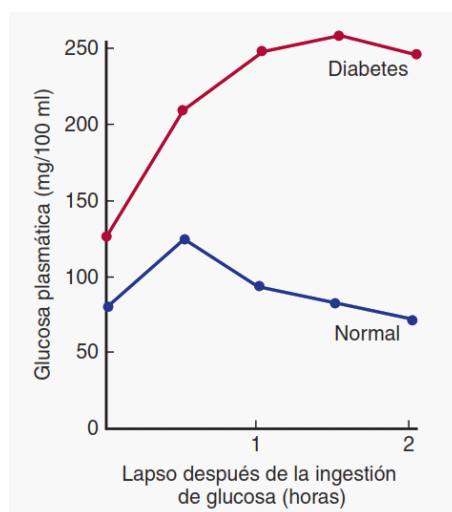


Figura 2. Prueba de tolerancia a la glucosa oral. Nota: Tomado de (Barrett, 2019)

Aunque la afección se puede presentar a cualquier edad, ocurre con mayor frecuencia en niños y jóvenes. Los síntomas característicos que acompañan a la diabetes tipo 1 incluyen sed excesiva, visión borrosa, mojar la cama, micción frecuente, falta de energía, hambre constante y pérdida de peso repentina Federation (2019).

El efecto de la insulina consiste en permitir una mayor penetración de glucosa en las células, disminuyendo la concentración de glucosa plasmática en consecuencia. Sin embargo, el exceso de esta hormona puede ocasionar un déficit de glucosa plasmática (hipoglucemia), estado que puede conllevar graves problemas de salud como letargo, coma, crisis convulsivas, daño cerebral permanente y la muerte (Barrett, 2019). Por esta razón, el tratamiento de la diabetes tipo I debe tener en cuenta evitar tanto los niveles de hiperglucemia como los de hipoglucemia, en pro de mantener la glucosa del paciente en rangos saludables (70 a 180mg/dL).

Para modelar el comportamiento de la glucosa de los pacientes diabéticos simulados, se usan las ecuaciones desarrolladas por el matemático Boris Kovatchev Kovatchev et al. (2009). Los parámetros de estas ecuaciones fueron sintonizadas, a partir de pruebas en un sujeto sano, donde ciertos parámetros se modificaron para generar 3 bases de datos de pacientes diabéticos: niños, adolescentes y adultos (Referencia). De esta manera, resulta el modelo UVA Padova, que es el modelo final con el que se realizan todas las simulaciones.

3.1.4. Modelo Uva Padova. Es el modelo implementado en el simulador metabólico de diabetes mellitus tipo I (T1DMS) desarrollado por la universidad de Padova, en Italia y la universidad de Virginia, en Estados Unidos. El T1DMS se ha convertido en una plataforma de referencia para diseñar e implementar técnicas de control en lazo cerrado para el tratamiento de la diabetes mellitus tipo I. Al contar con el aval de la federación de drogas y alimentos (FDA), se ha posicionado como sustituto de ensayos pre-clínicos y se han desarrollado diversas investigaciones en el área de la diabetes mellitus tipo I haciendo uso de esta herramienta (Molano-jiménez and León-vargas,

2017).

El UVA PADOVA es un modelo basado en compartimientos que describe la dinámica de glucosa de un paciente con diabetes mellitus tipo I por medio de 17 ecuaciones diferenciales y algunas funciones, que modelan la dinámica del tracto digestivo, el subsistema de glucosa, la utilización de glucosa, la producción endógena de glucosa efectuada por el hígado y el subsistema de insulina. El modelo fue actualizado en el 2013, donde se incluyeron las dinámicas del glucagón, la excreción renal y se complementan las dinámicas de la glucosa en condiciones de hipoglucemia (Molano-jiménez and León-vargas, 2017)(Dalla Man et al., 2014). Su más reciente versión fue lanzada en el 2017, en la que el modelo se hace variante en el tiempo y se añade la posibilidad de usar diferentes formas de administrar insulina, aparte de la subcutánea (Visentin et al., 2018). El simulador en su versión comercial cuenta actualmente con los parámetros de 30 pacientes diabéticos tipo I distintos, divididos en 3 grupos: niños, adolescentes y adultos.

3.1.5. Descripción del modelo diabético. Debido a que el simulador T1DMS no permite hacer simulaciones en tiempo real y teniendo en cuenta que se necesita tener acceso a un modelo de ecuaciones totalmente identificado, que es fundamental en el diseño del modelo de predicción del controlador predictivo, se decidió extraer un modelo de caja blanca a partir de los resultados, información que ofrece el simulador y artículos relacionados al desarrollo del mismo. Este nuevo modelo identificado se construyó a partir de un sistema bloques en Simulink, dividido en subsistemas de ecuaciones.

Subsistema de glucosa. Subsistema que modela la dinámica de la aparición, utilización, secreción y producción de glucosa por parte de diferentes órganos en el torrente sanguíneo. La

ecuación (1) representa la variación de glucosa en plasma, (2) la variación de glucosa en tejidos y músculos, y (3) es la glucosa en plasma por unidad de volumen (Visentin et al., 2018; Colmegna and Peña, 2014).

$$\dot{G}_p(t) = EGP(t) + Ra(t) - U_{ii} - E(t) - k_1 \cdot G_p(t) + k_2 \cdot G_t(t) \quad (1)$$

$$\dot{G}_t(t) = -U_{id} + k_1 \cdot G_p(t) - k_2 \cdot G_t(t) \quad (2)$$

$$G(t) = \frac{G_p(t)}{V_G} \quad (3)$$

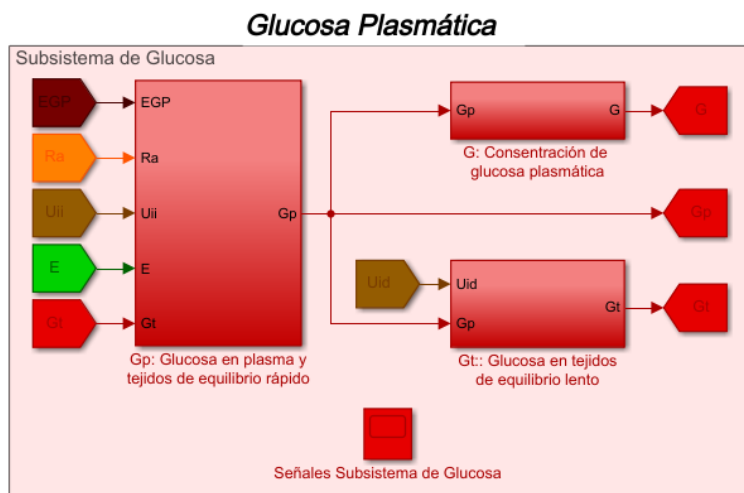


Figura 3. Sistema de bloques de Simulink que modela el subsistema de glucosa.

Subsistema de insulina. Sistema que modela la inyección y dinámica de la insulina. La ecuación (4) representa la variación de insulina en plasma, (5) la variación de insulina en el hígado,

y (6) es la insulina en plasma por unidad de volumen (Visentin et al., 2018; Colmegna and Peña, 2014).

$$\dot{I}_p(t) = -(m_2 + m_4) \cdot I_p(t) + m_1 \cdot I_l(t) + Ra_I(t) \quad (4)$$

$$\dot{I}_l(t) = -(m_1 + m_3) \cdot I_l(t) + m_2 \cdot I_p(t) \quad (5)$$

$$I(t) = \frac{I_p(t)}{V_I} \quad (6)$$

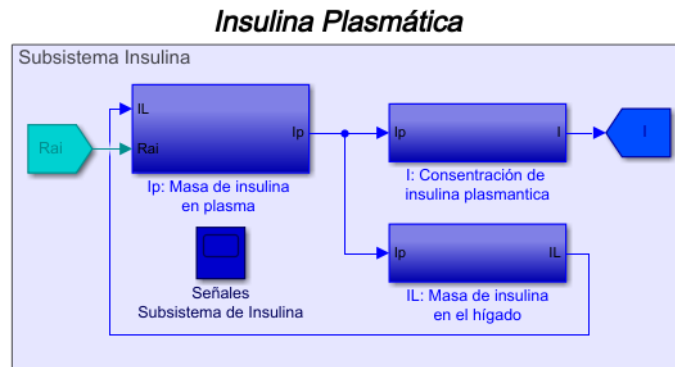


Figura 4. Sistema de bloques de Simulink que modela el subsistema de insulina.

Rata de aparición de glucosa. Representa la dinámica del tracto digestivo durante la ingesta de carbohidratos. La ecuación (7) representa cantidad de carbohidratos en el estomago, (8) y (9) son las variaciones de carbohidratos en el estomago, en estado solido y liquido respectivamente, (10) la absorción de carbohidratos por parte del intestino y (11) la rata de aparición de glucosa

en plasma (Visentin et al., 2018; Colmegna and Peña, 2014; Dalla Man et al., 2006).

$$Q_{sto}(t) = Q_{sto1}(t) + Q_{sto2}(t) \quad (7)$$

$$\dot{Q}_{sto1}(t) = -k_{max} \cdot Q_{sto1}(t) + D \cdot \delta(t) \quad (8)$$

$$\dot{Q}_{sto2}(t) = -k_{empt}(Q_{sto}) \cdot Q_{sto2}(t) + k_{max} \cdot Q_{sto1}(t) \quad (9)$$

$$\dot{Q}_{gut}(t) = -k_{abs} \cdot Q_{gut}(t) + k_{empt}(Q_{sto}) \cdot Q_{sto2}(t) \quad (10)$$

$$Ra(t) = \frac{f \cdot K_{abs} \cdot Q_{gut}(t)}{BW} \quad (11)$$

Donde la constante de vaciado gástrico $K_{empt}(Q_{sto})$ viene dada por (12):

$$k_{empt}(Q_{sto}) = k_{min} + \frac{k_{max} - k_{min}}{2} \cdot \{ \tanh[\alpha(Q_{sto} - b \cdot D)] - \tanh[\beta(Q_{sto} - c \cdot D) + 2] \} \quad (12)$$

Producción endógena de glucosa. Representa la glucosa producida por el hígado cuando las células alfa detectan niveles bajos de glucosa en sangre. La ecuación (13) representa la producción endógena de glucosa, (14) representa a la insulina proveniente del hígado, (15) representa

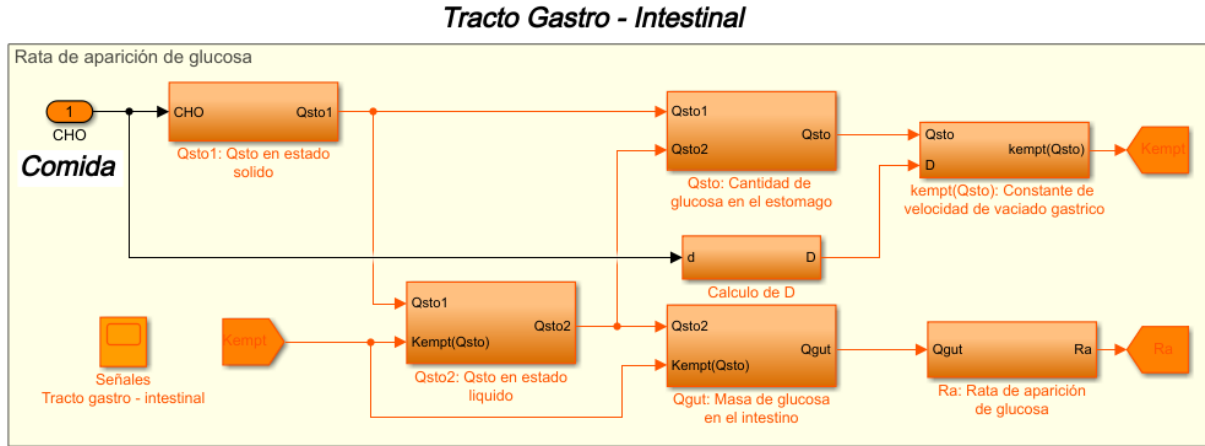


Figura 5. Sistema de bloques de Simulink que modela el paso de los alimentos por el tracto digestivo.

el retardo en la insulina inyectada y (16) la acción retardada del glucagon sobre la producción endógena de glucosa (Visentin et al., 2018; Colmegna and Peña, 2014).

$$EGP(t) = k_{p1} - k_{p2} \cdot G_p(t) - k_{p3} \cdot X^L(t) + \xi \cdot X^H(t) \quad (13)$$

$$\dot{X}^L(t) = -k_i \cdot [X^L(t) - I'(t)] \quad (14)$$

$$\dot{I}'(t) = -k_i \cdot [I'(t) - I(t)] \quad (15)$$

$$\dot{X}^H(t) = -k_H \cdot X^H(t) + -k_H \cdot \text{máx}[(H(t) - H_b), 0] \quad (16)$$

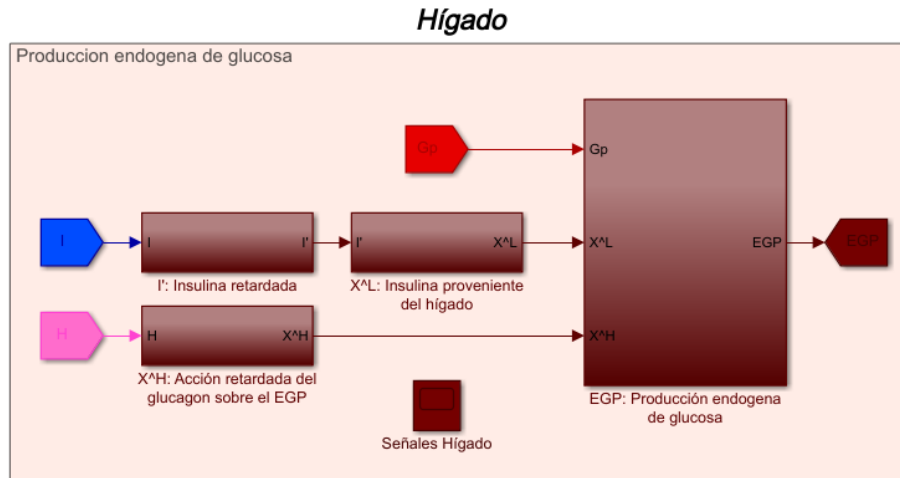


Figura 6. Sistema de bloques de Simulink que modela la producción endógena de glucosa realizada por el hígado.

Utilización de glucosa. Es el subsistema encargado de describir el uso de la glucosa en diversas funciones metabólicas del cuerpo. Donde (17) representa la utilización insulino independiente de glucosa realizada por el cerebro, (18) es la utilización insulino dependiente realizada por los músculos y tejidos del cuerpo, (19) modela la insulina activa durante la utilización de glucosa (Visentin et al., 2018; Colmegna and Peña, 2014).

$$U_{ii}(t) = F_{cns} \quad (17)$$

$$U_{id}(t) = \frac{[V_{m0} + V_{mx} \cdot X(t) \cdot (1 + r_1 \cdot risk)] \cdot G(t)}{K_{m0} + G(t)} \quad (18)$$

$$\dot{X}(t) = -p_{2U} \cdot X(t) + p_{2U} \cdot [I(t) - I_b] \quad (19)$$

$$risk = \begin{cases} 0 & \text{si } G \geq G_b \\ 10 \cdot [f(G)^2] & \text{si } G_{th} \leq G < G_b \\ 10 \cdot [f(G_{th})^2] & \text{si } G < G_{th} \end{cases} \quad (20)$$

$$f(G) = [\log(G)]^{r_2} - [\log(G_b)]^{r_2} \quad (21)$$

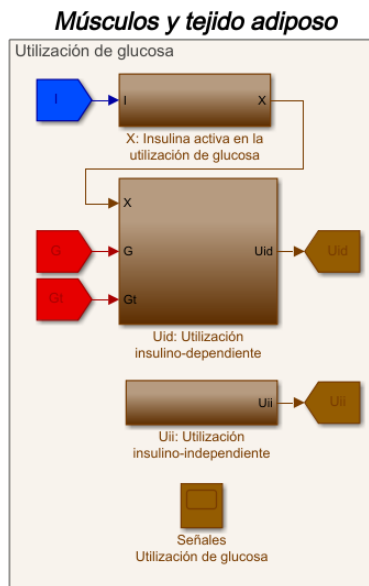


Figura 7. Sistema de bloques de Simulink que modela toda la utilización de glucosa en el paciente.

Excreción renal. Subsistema que modela la dinámica de la excreción de glucosa realizada por los riñones, representada por (22) (Visentin et al., 2018; Colmegna and Peña, 2014).

$$E(t) = \begin{cases} k_{e1} \cdot [G_p(t) - k_{e2}] & \text{si } G_p(t) > k_{e2} \\ 0 & \text{si } G_p(t) \leq k_{e2} \end{cases} \quad (22)$$

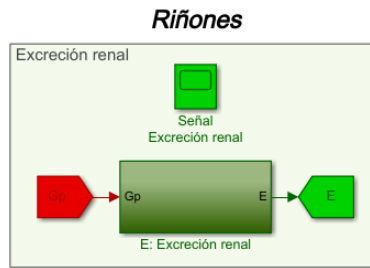


Figura 8. Sistema de bloques de Simulink que modela los riñones.

Cinética de la insulina subcutánea. Subsistema que modela la dinámica de la insulina inyectada externamente. Donde (23) representa la rata de aparición de insulina que se inyecta, (24) es la insulina no monomérica en el espacio subcutáneo y (25) la insulina monomérica en el espacio subcutáneo (Visentin et al., 2018; Colmegna and Peña, 2014).

$$Ra_I(t) = k_{a1} \cdot I_{sc1}(t) + k_{a2} \cdot I_{sc2}(t) \quad (23)$$

$$\dot{I}_{sc1}(t) = -(k_d + k_{a1}) \cdot I_{sc1}(t) + IIR(t) \quad (24)$$

$$\dot{I}_{sc2}(t) = k_d \cdot I_{sc1}(t) - k_{a2} \cdot I_{sc2}(t) \quad (25)$$

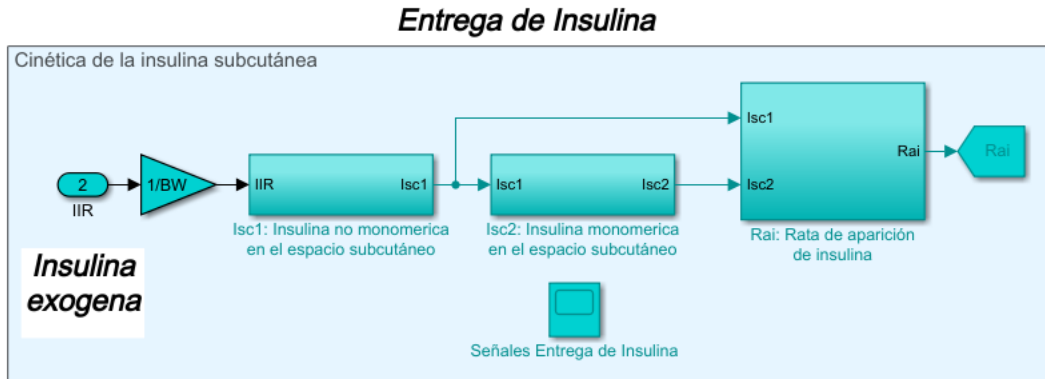


Figura 9. Sistema de bloques de Simulink que modela la absorción de insulina exogena.

Cinética de la glucosa subcutánea. Subsistema que modela la dinámica de la glucosa que pasa al espacio subcutáneo, representado por (26) (Visentin et al., 2018; Colmegna and Peña, 2014).

$$\dot{G}_{sc}(t) = -\frac{1}{T_s} \cdot G_{sc}(t) + \frac{1}{T_s} \cdot G(t) \tag{26}$$

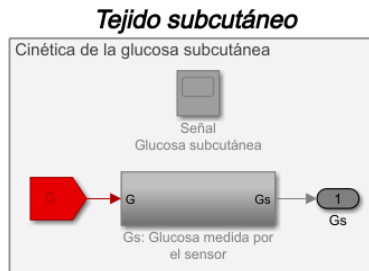


Figura 10. Sistema de bloques de Simulink que modela la glucosa que pasa del plasma al tejido subcutáneo.

Cinética y secreción del glucagón. Subsistema que modela la dinámica del glucagón producido por las células alfa de los islotes pancreáticos. Donde (27) representa la concentración de glucagón plasmático, (29) y (30) representan la síntesis de glucagón en las células alfa (Visentin

et al., 2018; Colmegna and Peña, 2014).

$$\dot{H}(t) = -n \cdot H(t) + SR_H(t) + Ra_H(t) \quad (27)$$

$$\dot{SR}_H(t) = SR_H^s(t) + SR_H^d(t) \quad (28)$$

$$\dot{SR}_H^s(t) = \begin{cases} -\rho \cdot [SR_H^s(t) - SR_H^b] & \text{si } G(t) \geq G_b \\ -\rho \cdot [SR_H^s(t) - \text{máx}(\frac{\sigma \cdot [G_{th} - G(t)]}{I(t) + 1} + SR_H^b, 0)] & \text{si } G(t) < G_b \end{cases} \quad (29)$$

$$SR_H^d(t) = \delta \cdot \text{máx}(-\frac{dG(t)}{dt}, 0) \quad (30)$$

Cinética del glucagón subcutáneo. Subsistema que modela la dinámica del glucagón inyectado de forma exógena. Donde (31) representa la tasa de aparición de glucagón inyectado, (32) y (33) representan dos compartimientos que modelan el paso del glucagón hacia el torrente sanguíneo (Visentin et al., 2018; Colmegna and Peña, 2014).

$$Ra_H(t) = k_{h3} \cdot H_{sc2}(t) \quad (31)$$

$$\dot{H}_{sc1}(t) = -(k_{h1} + k_{h2}) \cdot H_{sc1}(t) \quad (32)$$

$$\dot{H}_{sc2}(t) = k_{h1} \cdot H_{sc1}(t) - k_{h3} \cdot H_{sc2}(t) \quad (33)$$

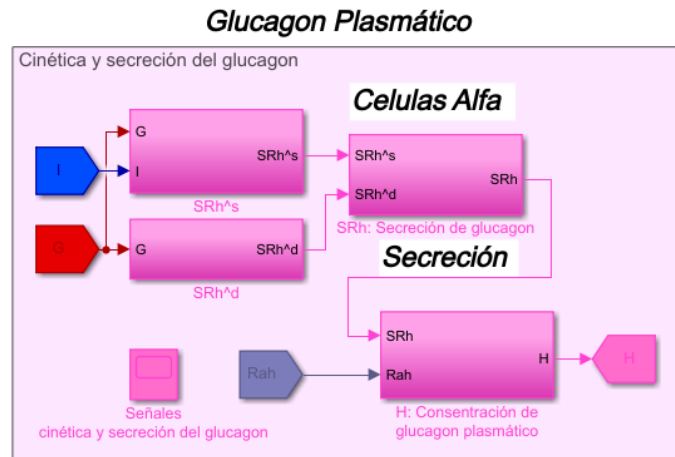


Figura 11. Sistema de bloques de Simulink que modela el subsistema de glucagon.

3.2. Control predictivo basado en el modelo (MPC)

El control predictivo basado en el modelo o MPC (*"Model Predictive Control"*), es una estrategia de control avanzada que emplea un modelo bien definido, para hacer predicciones de la salida a lo largo de una ventana de tiempo, usando la información presente y pasada del proceso que esta controlando. Es una estrategia versátil, que permite modificar su algoritmo para incluir estrategias de control óptimo, robusto, adaptativo, etc. Una de sus mayores virtudes esta en el manejo de restricciones lo que le permite mantener sin problemas condiciones de operación exigentes en procesos complejos. Maciejowski (2002).

La ley de control MPC, se calcula solucionando un problema de optimización sujeto a restricciones, usualmente aplicando programación cuadrática y siguiendo el siguiente esquema:

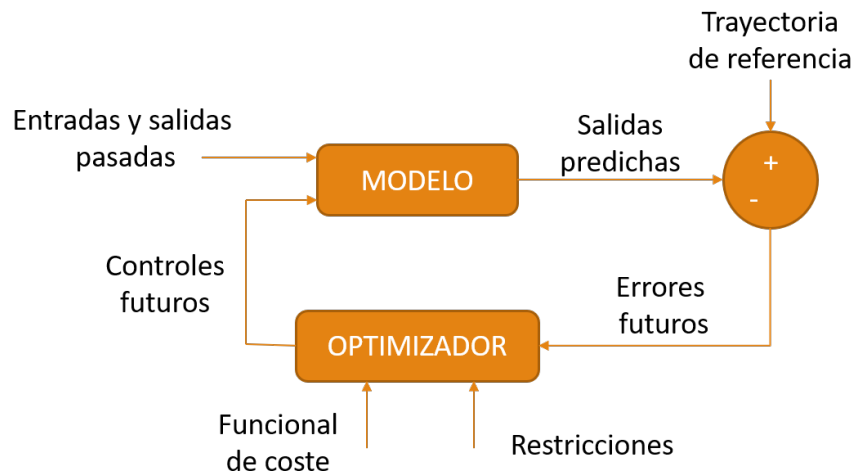


Figura 12. Esquema aplicado por el algoritmo de control predictivo para calcular las acciones de control.

Los parámetros que se tuvieron en cuenta para sintonizar el controlador predictivo son:

3.2.1. Modelo de predicción. Es la parte más importante del algoritmo de control predictivo, ya que de este dependen la fiabilidad de las predicciones, por lo tanto, a mayor precisión del modelo más efectiva será la ley de control calculada. El modelo se usa para calcular salidas predichas denominadas $\hat{y}[k+i|k]$, donde en la parte derecha de la llave se indica el instante de tiempo actual en que se encuentra el sistema, y en el izquierdo el instante de tiempo futuro estimado durante la predicción, que se indica usando el índice i .

Existen dos formas de trabajar con el modelo de predicción, la primera denominada “modelo desacoplado”, se caracteriza por calcular las condiciones iniciales del modelo de predicción para cada iteración, sin tener en cuenta lo que está sucediendo con la planta y la acción de control que se enviará al proceso. Este tipo de algoritmo es sencillo de implementar aplicando control en espacio

de estados, ya que no requiere un observador de estados, basta con tener mediciones de la salida del proceso para implementar una acción correctiva en la ley de control para compensar imprecisiones de modelado que puedan generar errores de estado estable. El concepto de modelo desacoplado se aplica en algoritmos DMC ("*Dinamic Matrix Control*") y QDMC ("*Quadratic Dinamic Matrix Control*") Maciejowski (2002).

La en la segunda, denominada "modelo acoplado", las condiciones iniciales del modelo de predicción son los estados medidos o estimados de la planta. Este tipo de algoritmo por lo general requiere de un observador de estados para poder generar condiciones iniciales para cada iteración, pero es más estable y robusta que su contra parte, debido a que en ningún momento se pierde de vista como esta evolucionando la planta.

3.2.2. Horizonte de predicción. Indica el ancho de la ventana que contiene todas las predicciones durante el calculo de la acción de control, se denota H_p . Se dice que el control predictivo aplica un horizonte deslizante porque la ventana generada por H_p avanza hacia el futuro con cada nueva iteración.

Es un parámetro de sintonización clave, por lo general, si se define un horizonte de predicción muy lejano, se consigue un desempeño mejor, pero a medida que la ventana crece, el costo computacional aumenta. Se recomienda que H_p sea lo suficientemente grande para contener todo el transitorio y un poco del estado estable de la respuesta del sistema.

Se debe tener en cuenta que el algoritmo no siempre usa toda la información de la ventana generada por el horizonte de predicción, por esta razón, es importante escoger adecuadamente los denominados “puntos de coincidencia”, que representan los instantes predichos dentro del horizonte de predicción que serán utilizados para calcular la acción de control. El concepto de aplicar puntos de coincidencia permite evitar información dentro de la predicción que no sea relevante, como por ejemplo, los primeros instantes predichos en un proceso con retardo.

3.2.3. Horizonte de control. Es una estructura que se le da al cálculo de las acciones de control futuras durante la predicción, donde solo se permitirá hacer variaciones en la acción de control a lo largo del horizonte de control, después de pasarlo, el valor de la acción de control permanecerá constante en lo que queda del horizonte de predicción. El horizonte de control se denota con H_u y su valor debe ser menor o igual al valor del horizonte de predicción H_p .

Este parámetro afecta en gran medida la forma de la acción de control generada por el controlador, haciendo que sea más suave para valores de H_u pequeños o agresiva para valores de H_u altos. Valores altos de H_u pueden conseguir respuestas con un seguimiento bastante preciso, pero en consecuencia hacen al algoritmo más susceptible a errores de modelado, cambios inesperados en las variables del proceso y aumenta considerablemente el costo computacional.

En la figura 13 se muestran los conceptos asociados a los horizontes de predicción y control. En la figura se sigue un set point denotado por $s(t)$ y se calcula una trayectoria de referencia a lo largo de la predicción denotada por $r[t|k]$, esta trayectoria de referencia es una aproximación exponencial

desde la saluda actual hasta el set point que se aplica cuando se desea evitar cambios bruscos en la referencia.

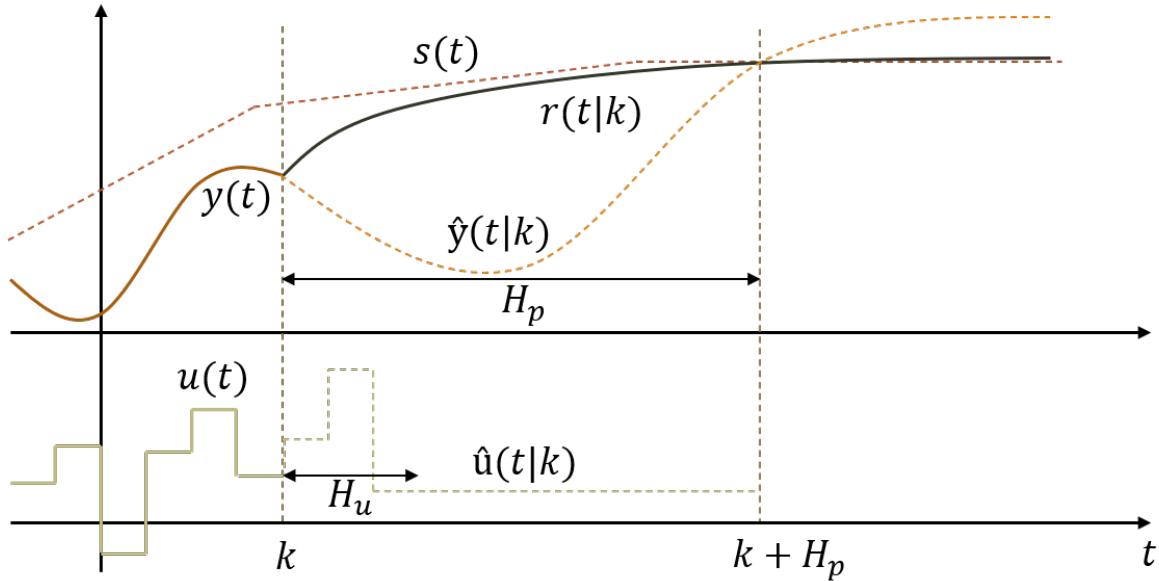


Figura 13. Predicciones de la salida a lo largo de un horizonte de predicción definido, aplicando un horizonte de control a la acción de control.

3.2.4. Función objetivo. Función que contiene las características del proceso que se desean optimizar, generalmente en forma de sumatorias cuadráticas buscando garantizar que la función sea convexa para que el problema de optimización tenga un mínimo global.

En los planteamientos más comunes de algoritmos de control predictivo se tienen en cuenta, el error de seguimiento y el esfuerzo en la variación de la acción de control como términos de la

función de coste. Un ejemplo de función de coste se muestra a continuación:

$$V[k] = \sum_{i=H_w}^{H_p} \|\hat{y}[k+i|k] - r[k+i|k]\|_{Q[i]}^2 + \sum_{i=1}^{H_u-1} \|\Delta\hat{u}[k+i|k]\|_{R[i]}^2 \quad (34)$$

Donde H_p y H_u corresponden a los horizontes de predicción y control respectivamente, H_w se usa para definir desde que punto en el futuro se empezarán a tener en cuenta las predicciones para la optimización, $r[k+i|k]$ puede ser el set point o una trayectoria de referencia definida para suavizar el seguimiento, y finalmente Q y R son matrices de ponderación. Para el problema de control de glucosa H_w se definió como 1 a pesar de presentar fuertes retardos, para que el sistema responda inmediatamente se detecte una perturbación y las matrices de ponderación se definieron constantes.

Una de las características que hacen del control predictivo una estrategia frecuentemente aplicada en procesos industriales, es que su función objetivo este sujeta a restricciones, ya sea en la acción de control o en la salida del proceso. Por esta razón es común hablar de programación cuadrática cuando se desarrollan algoritmos de control predictivo como el QDMC. Las restricciones se definen mediante desigualdades lineales y pueden ser por ejemplo:

$$y_{min} \leq y \leq y_{max} \quad (35)$$

$$u_{min} \leq u \leq u_{max} \quad (36)$$

$$\Delta u_{min} \leq \Delta u \leq \Delta u_{max} \quad (37)$$

Se debe tener en cuenta que a medida que se incluyen restricciones en el problema de optimización el costo computacional se eleva. Además, es necesario definir una variable de holgura para las restricciones a la salida, que permita que puedan llegar a ser violadas hasta cierto punto, de otra manera el algoritmo de optimización podría generar fallos durante la operación debido a que no encuentra soluciones posibles.

3.2.5. Matrices de ponderación. Las matrices de ponderación Q y R son parámetros de sintonización que se usan para dar mayor o menor peso a algún elemento de la función objetivo, permitiendo dar prioridad a objetivos de control que se consideren más importantes que otros. Se pueden definir constantes para ponderar de forma absoluta el seguimiento o el esfuerzo de control a lo largo de las predicciones o se pueden definir variables para ponderar de una manera los primeros instantes de la predicción y de otra los instantes finales.

Ambas matrices cuadradas definidas positivas, donde el tamaño de Q depende de los estados del sistema y el tamaño de R de las entradas de control.

3.2.6. Ley de control. El calculo de la ley de control se basó principalmente en control predictivo QDMC, donde la principal característica del algoritmo consiste en usar el modelo de predicción para generar sus propias condiciones iniciales en cada iteración. A continuación se explicara como se desarrolló el algoritmo de control general.

Modelo de predicción:. El modelo de predicción se definió en el dominio de espacio de estados discreto, definiendo la matriz de estado G_m , la matriz de entradas H_m , la matriz de salidas C_m que representa la medición del estado de glucosa subcutánea y la matriz D_m nula.

Para calcular las predicciones, la salida del modelo se divide en respuesta libre y forzada, donde la primera depende de las condiciones iniciales y la segunda de la entrada.

$$\hat{y}[k+i|k] = \hat{y}_{libre}[k+i|k] + \theta \Delta \hat{u}[k|k] \quad (38)$$

Matriz de respuesta forzada:. La respuesta forzada es calculada para cada instante a lo largo del horizonte de predicción, aplicando al modelo únicamente una entrada escalón de amplitud unitaria y definiendo condición inicial cero para los estados, cada valor calculado se denomina $S[k|k]$.

$$\begin{aligned} x[k|k] &= G_m x[k-1|k] + H_m \\ S[k|k] &= C_m x[k|k] \end{aligned} \quad (39)$$

Los valores $S(k)$ se agrupan dependiendo del horizonte de control y los puntos de coincidencia definidos para armar la matriz Θ Maciejowski (2002), después de ser generada esta matriz permanecerá constante durante toda la operación.

Matriz Hessiana: Es el componente cuadrático que queda luego de ajustar la función objetivo a la forma de un problema de optimización de programación cuadrática Best (2017):

$$J_{QP} = \frac{1}{2} \Delta u^T H \Delta u + f \Delta u + \frac{1}{2} \varepsilon^T P \varepsilon \quad (40)$$

De esta ecuación H corresponde al componente cuadrático o matriz Hessiana, f es el componente lineal y P es una matriz de holgura definida para permitir que el algoritmo pueda violar algunas restricciones hasta cierto punto para evitar problemas de convergencia, donde ε es una variable de optimización adicional para incluir la holgura en el calculo del algoritmo de programación cuadrática. Partiendo de la función objetivo de la sección 3.2.4 y aplicando la ecuación 38 se obtiene:

$$V(k) = \sum_{i=1}^{H_p} \|\hat{y}_{libre}[k+i|k] + \theta \Delta \hat{u} - r[k+i|k]\|_Q^2 + \sum_{i=1}^{H_u-1} \|\Delta \hat{u}[k+i|k]\|_R^2 \quad (41)$$

Definiendo una variable de error $\varepsilon(k)$ como sigue:

$$\varepsilon[k] = r[k+i|k] - \hat{y}_{libre}[k+i|k] \quad (42)$$

Y llevando la función objetivo a la forma de programación cuadrática se obtuvo:

$$V[k] = \Delta u^T (\Theta^T Q \Theta + R) \Delta u - 2(\Theta^T Q \varepsilon[k]) \Delta u + \varepsilon(k)^T Q \varepsilon[k] \quad (43)$$

La variable a optimizar es Δu , por lo tanto, el último término de la ecuación 43 será un valor constante durante los cálculos y por esta razón no se tuvo en cuenta. De este resultado obtenemos la ecuación de la matriz Hessiana:

$$H = 2(\Theta^T Q \Theta + R) \quad (44)$$

Incluyendo el término de holgura como se muestra en la ecuación 46 y creando un nuevo vector para la variable a optimizar que agrupe a Δu y ε , H resulta:

$$H = \begin{bmatrix} 2(\Theta^T Q \Theta + R) & 0_{Hu \times 1} \\ 0_{1 \times Hu} & P \end{bmatrix} \quad (45)$$

Matrices de restricciones: La programación cuadrática define las restricciones del problema de optimización en forma de desigualdades lineales de la forma:

$$A \Delta u \leq b \quad (46)$$

Donde A_c es una matriz de transformación que se usa para definir las restricciones en términos de la variable de optimización y b_c es un vector construido con los valores de las restricciones.

Las restricciones son definidas en la sección 2.6.4 donde A_c y b_c para la variable de optimización

Δu :

$$A_c = \begin{bmatrix} I_{H_u} \\ -I_{H_u} \\ I_l \\ -I_l \\ \Theta \\ \Theta \end{bmatrix} \quad (47)$$

$$b_c = \begin{bmatrix} \Delta u_{max} \times 1_{H_u \times 1} \\ -\Delta u_{min} \times 1_{H_u \times 1} \\ u_{max} \times 1_{H_u \times 1} \\ -u_{min} \times 1_{H_u \times 1} \\ y_{max} \times 1_{H_p \times 1} \\ -u_{min} \times 1_{H_p \times 1} \end{bmatrix} \quad (48)$$

Donde I_{H_u} es una matriz identidad de dimensión $H_u \times H_u$ y I_l es una matriz triangular inferior de dimensión $H_u \times H_u$. Teniendo en cuenta la variable ε añadida para agregar holgura a las

restricciones a la salida A_c y b_c resultan:

$$A_c = \begin{bmatrix} I_{Hu} & 0_{Hu \times 1} \\ -I_{Hu} & 0_{Hu \times 1} \\ I_l & 0_{Hu \times 1} \\ -I_l & 0_{Hu \times 1} \\ \Theta & -1_{Hp \times 1} \\ \Theta & -1_{Hp \times 1} \\ 0_{Hu}^T & -1 \end{bmatrix} \quad (49)$$

$$b_c = \begin{bmatrix} \Delta u_{max} \\ -\Delta u_{min} \\ u_{max} \\ -u_{min} \\ y_{max} \\ -u_{min} \\ 0 \end{bmatrix} \quad (50)$$

Matriz de respuesta libre: La respuesta libre es calculada para cada instante a lo largo del horizonte de predicción, usando las condiciones iniciales del modelo de predicción y manteniendo la condición inicial de las entradas, incluyendo la acción de control calculada en la iteración anterior y la medida actual de las entradas de perturbación $u_d(k|k)$, cada valor calculado se denomina $y_{libre}(k)$.

$$\begin{aligned} x[k|k] &= G_m x[k-1|k] + H_m u[k-1|k] + H_{md} u_d[k|k] \\ y_{libre}[k|k] &= C_m x[k|k] \end{aligned} \quad (51)$$

La respuesta libre varía en cada iteración, ya que puede depender de entradas externas, haciendo necesario definir una matriz H_{md} que modele su efecto en el sistema y de la condición inicial actual del modelo de predicción.

Calculo del coeficiente lineal de la función objetivo: Es el termino denominado f en la ecuación 34. Se obtiene de remplazar la ecuación 42 en 43 y tomando el termino lineal.

$$f = -2(\Theta^T Q(r[k+i|k] - \hat{y}_{libre}[k+i|k])) \quad (52)$$

Si se trabaja con un modelo de predicción desacoplado el termino correspondiente a ε puede incluir la variable de corrección $d(k|k)$ para eliminar el error de estado estable generado por los errores de predicción. Donde $d[k|k]$ es la diferencia entre la salida de la planta y la salida calculada

por el modelo de predicción.

$$d[k|k] = y_p[k-1|k] - y_m[k-1|k] \quad (53)$$

Donde $y_p[k-1|k]$ corresponde a la medición de la salida en la planta y $y_m(k-1|k)$ al calculo de la salida al final de la iteración anterior usando el modelo de predicción, modificando la ecuación 52 como sigue.

$$f = -2(\Theta^T Q(r[k+i|k] - d[k|k] - \hat{y}_{libre}[k+i|k])) \quad (54)$$

Finalmente, si se tiene en cuenta la matriz ampliada que incluye el termino de holgura para las restricciones a la salida, la ecuación 54 queda de la forma.

$$f = \begin{bmatrix} -2(\Theta^T Q(r[k+i|k] - d[k|k] - \hat{y}_{libre}[k+i|k])) \\ 0 \end{bmatrix} \quad (55)$$

Calculo de la acción de control. Se calcula Δu usando la función *quadprog* de MATLAB. Esta función recibe como parámetros de entrada las matrices H , f , A_c y b_c , y devuelve un vector de todos los Δu calculados de dimensión $H_u \times 1$. Del vector que arroja *quadprog* solo se toma el primer valor, que es el correspondiente al instante actual y se le suma el valor de la acción de control de la iteración anterior

$$u[k|k] = u[k-1|k] + \Delta u[k|k] \quad (56)$$

Condiciones iniciales: Si se define un modelo de predicción desacoplado, se calculan después de hallar $u(k|k)$ calculando el valor de $y_m(k|k)$ usando el modelo de predicción y las entradas actuales:

$$\begin{aligned}x[k|k+1] &= G_m x[k-1|k] + H_m u[k|k] + H_{md} u_d[k|k] \\y_m[k|k+1] &= C_m x[k|k]\end{aligned}\tag{57}$$

Si se define un modelo acoplado las condiciones iniciales serán los valores medidos de la planta Maciejowski (2002).

3.3. Estimador de perturbaciones

Para aprovechar la acción feed-forward que se puede implementar fácilmente en el algoritmo de control predictivo, se propuso el diseño de un estimador de perturbaciones basado en clasificación de señales con el objetivo de hacer que sea independiente de un modelo.

3.3.1. Control feed-forward. Es una estrategia de control de lazo abierto que se basa en anticipar el efecto de una entrada externa al sistema y compensarlo a partir de la medición o estimación de dicha entrada. Esta estrategia es fundamental en sistemas expuestos a fuertes perturbaciones o donde el efecto de las perturbaciones sea

En la figura 14 se muestra un sistema de control en lazo cerrado que aplica una acción feed-forward adicional $u_{ff}[k]$ a su acción de control $u[k]$, el objetivo es que $u_{ff}[k]$ cancele el efecto de las perturbaciones y $u[k]$ haga el que sistema tenga sea estable y tenga un transitorio adecuado.

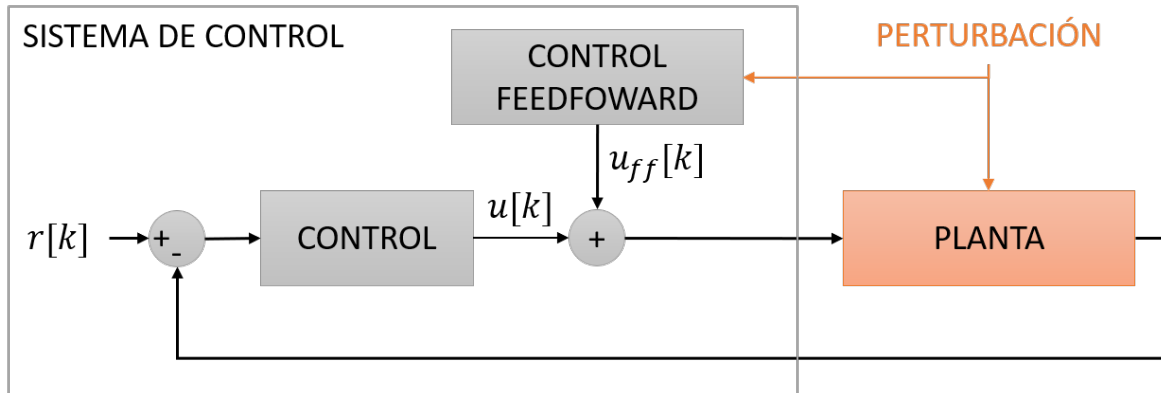


Figura 14. Sistema de control en lazo cerrado con acción feed-forward.

3.3.2. Clasificadores. Son algoritmos basados en *Machine Learning* o aprendizaje automático, que se encargan de asignar un valor o determinar a que conjunto pertenece, una serie de datos dependiendo de sus características. La selección o clasificación de los datos se hace mediante un modelo de clasificación que se genera en un proceso denominado *aprendizaje supervisado*, donde se le entrega al algoritmos una serie de datos de entrada que debe aprender a clasificar, denominados *predictores* y sus respectivas salidas que son las clases, conjuntos a los que pertenecen los datos; o los valores que deben generar estos datos, para que de esta manera, el algoritmo pueda entrenarse y posteriormente, clasificar Bishop (2006).

3.4. Observadores de estados

Los sistemas desarrollados en el dominio del espacio de estados requieren de aplicar realimentación de estados completa para aprovechar al máximo sus características. Esto es, tener medición de todas las variables de estados que representen el modelo del sistema. Sin embargo, en la mayoría de casos solo se cuenta con sensores para medir algunas de estas variables, algunas mediciones no son buenas o no miden directamente las variables requeridas. En estos casos es necesario imple-

mentar un sistema de estimación que calcule un valor aproximado de la variable de estado, estos sistemas se conocen como observadores de estados.

Partiendo de un sistema de espacio de estados básico:

$$\begin{aligned}x[k + 1] &= Gx[k] + Hu[k] \\y[k] &= Cx[k] + Du[k]\end{aligned}\tag{58}$$

Donde G , H , C y D son las matrices que representan el modelo en espacio de estados discreto. La ecuación general de un observador de espacio de estados discreta es:

$$\hat{x}[k + 1] = G\hat{x}[k] + Hu[k] + L(y[k] - C\hat{x}[k])\tag{59}$$

La expresión $G\hat{x}[k] + Hu[k]$ se usa para predecir la dinámica de las variables de estado y $L(y - C\hat{x}[k])$ hace correcciones basándose en las mediciones, donde L es la matriz de ganancia del observador, que se calcula de tal manera que sus valores propios sean estables y más rápidos que los polos del sistema en lazo cerrado. El esquema de un observador implementado para realimentación de estados se muestra en la figura 15.

Debido a que este planteamiento para el observador de estados no es suficientemente preciso para implementarlo en un sistema de control que puede presentar ruido de medición e incertidumbre en el modelo, se decidió abordar un concepto más elaborado.

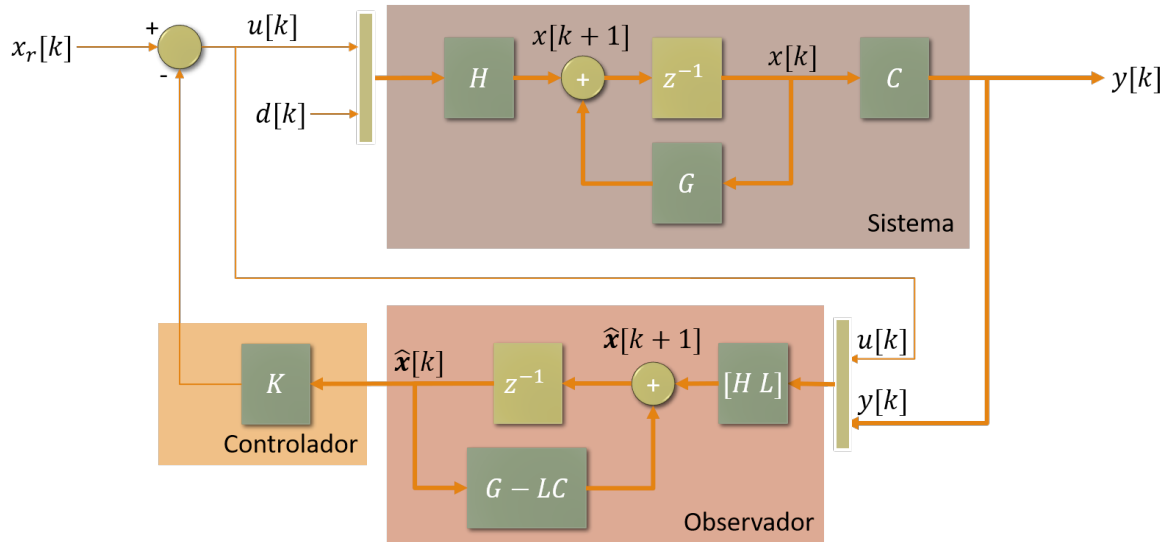


Figura 15. Sistema de control en espacio de estados con observador.

3.4.1. Filtro de Kalman. El filtro de Kalman es un observador de estados estocástico, que tiene en cuenta la presencia de variables aleatorias en sus cálculos. Este observador parte de la siguiente ecuación de espacio de estados para el modelo:

$$\begin{aligned} x[k+1] &= Gx[k] + Hu[k] + w[k] \\ y[k] &= Cx[k] + Du[k] + v[k] \end{aligned} \quad (60)$$

Donde $w[k]$ representa el ruido del proceso, que está asociado a la incertidumbre de modelado y $v[k]$ representa el ruido de medición que pueden presentar los sensores, ambas señales se representan como ruido blanco, con media cero, no correlacionadas y se asume que el valor de su covarianza es conocido, siendo Q_k para $w[k]$ y R_k para $v[k]$.

La estructura de la ecuación general del filtro de Kalman es similar a la de la ecuación 59 del observador de estados, donde una parte representa la predicción del estado a partir del modelo y otra la corrección de la estimación a partir de la medición de la salida. La forma de operar del filtro consiste en dos pasos que se repiten de forma cíclica, primero se predicen estados usando el modelo, después se actualizan y corrigen usando la información de los sensores.

$$\hat{x}^+[k+1] = \hat{x}^-[k] + K[k](y[k] - C\hat{x}^-[k]) \quad (61)$$

En la ecuación 62 el índice “-” en $\hat{x}[k]^-$ indica que es una estimación a priori, es decir, antes de hacer una medición, luego $\hat{x}[k]^-$ representa la etapa de predicción y se calcula a partir de la información pasada que se tiene del proceso.

$$\hat{x}^-[k+1] = G\hat{x}^-[k] + Hu[k] \quad (62)$$

Donde G , H y C representan las matrices de estado del modelo. Por otra parte, la etapa de corrección viene dada por $K[k](y[k] - C\hat{x}^-[k])$, donde el término $K[k]$ conocido como ganancia de Kalman, hace las veces de la ganancia L de la ecuación 59 con la diferencia de que no se calcula por ubicación de polos, sino por un proceso de optimización que busca reducir la covarianza de la estimación del error, denotada como $P[k]$. El valor a priori de $P[k]$ se calcula mediante:

$$P^-[k] = GP^+[k-1]G^T + T_s^2 Q_k \quad (63)$$

el índice “+” indica que es un calculo a posteriori, es decir, después de usar la información medida por el sensor para corregir, usualmente el valor inicial $P[k-1]^+$ se define como Q_k , luego en cada iteración se calcula usando:

$$P^+[k] = (I - K[k]C)P^-[k](I - K[k]C)^T + K[k]R_kK[k]^T \quad (64)$$

Donde la ganancia de Kalman que minimiza su valor se define como:

$$K[k] = P^-[k]C^T (CP^-[k]C^T + R_k)^{-1} \quad (65)$$

Finalmente, $\hat{x}^+[k+1]$ que es la estimación a posteriori, es decir, después de aplicar la corrección, será la información que el filtro Kalman envíe como estado estimado Simon (2006).

3.4.2. Filtro de Kalman extendido. El filtro de Kalman es un estimador de estados óptimo pero no deja de ser lineal, por esta razón, si se desea estimar estados de un sistema con fuertes no linealidades, puede no presentar buenos resultados. El filtro de Kalman extendido es una versión no lineal del filtro de Kalman, que usa un modelo no lineal para el calculo durante la etapa de predicción y linealiza dos matrices jacobianas en cada iteración para poder usar las formulas presentadas anteriormente Simon (2006).

Etapas de predicción: En la etapa de predicción se calculan los estados estimados a partir de un modelo no lineal.

$$\hat{x}^-[k+1] = f(\hat{x}^-[k], u[k]) \quad (66)$$

Para poder calcular la covarianza a priori del error de estimación $P^- [k]$ se debe hallar una matriz $G[k]$ a partir de la discretización del jacobiano de la función no lineal del proceso $f(x, u)$ evaluando como punto de operación las condiciones iniciales actuales.

$$G[k] = I + T_s \frac{\partial f}{\partial x} \Big|_{x=x^- [k]} \quad (67)$$

$$P^- [k] = G[k]P^+ [k-1]G[k]^T + T_s^2 Q_k \quad (68)$$

Etapa de actualización: Para la etapa de actualización y corrección se debe hallar una matriz $C[k]$ a partir del jacobiano de la función de los sensores $g(x, u)$ evaluando como punto de operación las condiciones iniciales actuales, si esta función es lineal se usa la matriz C del espacio de estados.

$$C[k] = \frac{\partial g}{\partial x} \Big|_{x=x^- [k]} \quad (69)$$

$$K[k] = P^- [k]C[k]^T [C[k]P^- [k]C[k]^T + R_k]^{-1} \quad (70)$$

Finalmente, se calcula la estimación y la covarianza del error a posteriori usando la función de sensores y las demás matrices anteriormente calculadas.

$$x^+ [k] = x^- [k] + K[k][y[k] - g(x^- [k], u_k)] \quad (71)$$

$$P^+[k] = [I - K[k]C[k]]P^-[k][I - K[k]C[k]]^T + K[k]R_k K[k]^T \quad (72)$$

4. Esquema de control propuesto

4.1. Problema de control

Típicamente, el control de glucosa en un paciente diabético se considera como un problema de regulación, tal que ante una condición de perturbación (Ingesta de carbohidratos) se mantenga un nivel de glucosa alrededor del valor basal del paciente (100 a 130[mg/dL]). Sin embargo, por las limitaciones de la acción de insulina y de la condición del paciente, se pueden presentar condiciones de hiperglucemia (glucosa por encima de 180[mg/dL]) o hipoglucemia (glucosa por debajo de 70[mg/dL]). Sin embargo, un esquema de este tipo no es el adecuado dado que en el caso de un sujeto sano, existen excursiones de glucosa cerca del doble del nivel basal, dependiendo de la ingesta de carbohidratos. Por tanto, el esquema de control propuesto es un servo controlador, que sigue un perfil de referencia saludable del paciente, ante una ingesta estimada de carbohidratos. Por tanto, la estrategia de control propuesta debe considerar:

- Los retardos presentes en el modelo dinámico de la glucosa; la glucosa por lo general llega a su pico máximo alrededor de una hora después de la ingesta de alimentos y regresa a su valor basal alrededor de 2 a 3 horas después de la dosificación de insulina.
- No linealidades del modelo, que además presenta discontinuidades.
- Perturbaciones asociadas con la ingesta de carbohidratos.
- La incertidumbre paramétrica del modelo; los parámetros del modelo individual del paciente varían.

4.2. Propuesta de control

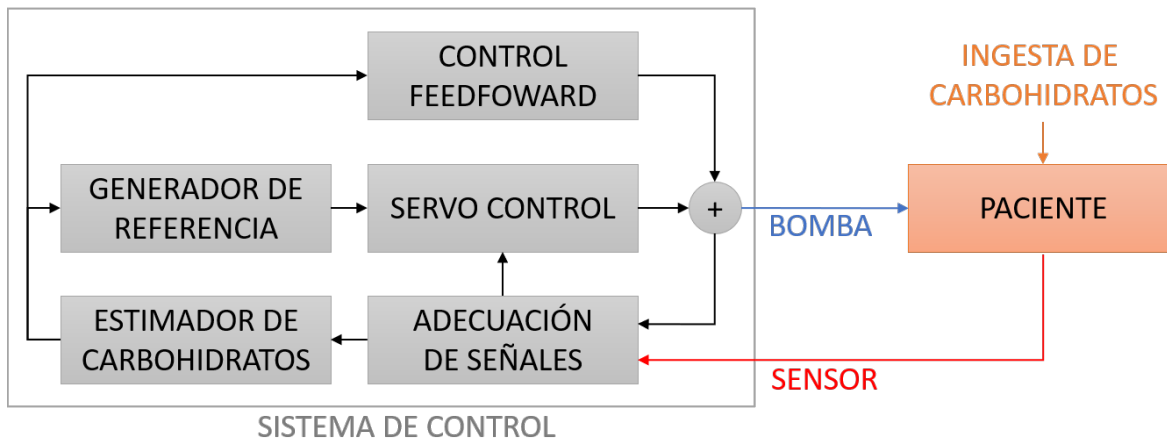


Figura 16. Esquema de control.

En la figura 16 se muestra el esquema de control propuesto, donde se busca controlar el perfil de glucosa del paciente diabético mediante la suma de 2 acciones de control. La primera basada en control feed-forward, que se encargará de cancelar los efectos de la perturbación haciendo que los valores de glucosa se eleven lo menos posible después de la ingesta y en un servo controlador que se encargará de seguir un perfil de referencia dinámico saludable para el paciente.

Teniendo en cuenta que tanto el control feed-forward, como el perfil de referencia dinámico dependen de la cantidad de carbohidratos consumidos, se propone incluir en el esquema de control un estimador de carbohidratos que no requiera ningún tipo de información adicional a la que ya suministra el sensor de glucosa. Debido a esto, se incluye adicionalmente un bloque de acondicionamiento de señales para lograr una estimación y control adecuado. El sensor de glucosa y la bomba de insulina incluyen características y limitaciones asociadas al hardware como ruido, saturaciones, cuantizaciones y muestreo.

5. Diseño de sistema de control

La estrategia de control realimentado seleccionada para dosificar la insulina en un paciente con diabetes mellitus tipo I es el control predictivo basado en el modelo. Para ello se tuvo en cuenta las características del problema de control. Por un lado, se cuenta con un modelo bien descrito, que es un requisito fundamental en un planteamiento de control predictivo, por el otro, el proceso presenta una dinámica lenta, con transitorios que pueden durar más de tres horas. Lo que permite diseñar un algoritmo que realice cálculos extensos. Adicionalmente, es una estrategia que aprovecha la estimación de perturbaciones, ya que por definición puede aplicar control feed-forward en sus predicciones. Finalmente, la estrategia seleccionada es robusta, presentando un buen desempeño a pesar de incertidumbre en el modelo de predicción y, como su ley de control depende del comportamiento predicho, se adapta muy bien a los sistemas con retardos.

Debido a que se planteó un controlador predictivo con seguimiento de referencia, es necesario que exista un perfil de glucosa saludable, por lo tanto es necesario un algoritmo que genere dicho perfil de referencia.

5.1. Cálculo de perfil de glucosa saludable

Se cuenta con una cohorte de 10 pacientes adultos in silico, por lo que se decidió generar un perfil de referencia para cada paciente, buscando que el controlador realmente pueda seguir la curva de referencia que sea generada y que esta curva, represente una condición saludable para el paciente in silico. Por lo tanto, se decidió aplicar una ley de control “ideal” en el paciente, y a partir de

esta, crear un perfil de referencia saludable. La ley de control utilizada para generar el perfil de referencia fue la del control mediante bolo óptimo que ofrece el simulador T1DMS en condiciones ideales, y evaluó su desempeño en el comportamiento de glucosa en los pacientes in silico.

El bolo óptimo se determina de tal manera que iniciando desde el valor basal, la concentración de glucosa se encuentre nuevamente entre el 85 % y 110 % de su valor basal al cabo de 3 horas, que su valor mínimo no esté en ningún momento por debajo de 90 [mg/dL] y su valor pico sea como máximo 80 [mg/dL] mayor al valor basal del paciente Dalla Man et al. (2014).

Sin embargo, el bolo óptimo es una ley de control de lazo abierto, que asume que el bolo será aplicado en el instante de la ingesta de carbohidratos y que para realizar su cálculo, se sabe la cantidad exacta de carbohidratos consumidos por el paciente y también se asume que el parámetro CR (*Carb Ratio*) del paciente, indispensable para obtener el valor del bolo, es preciso.

Debido a que en simulación estas condiciones se pueden idealizar, el desempeño del bolo óptimo se puede usar como referencia para aproximar una función de transferencia discreta que a partir de una entrada de carbohidratos (CHO), genere una curva de referencia para la glucosa. Las regresiones se realizaron usando la toolbox “*SystemIdentification*” de Matlab. En la figura 17 se muestra el resultado de la regresión para definir una función de transferencia para el paciente *Adult_001*.

Debido a que el control mediante bolo óptimo tiende a ser agresivo, se decidió modificar un poco la regresión buscando que el valor mínimo en la curva de referencia en todo momento

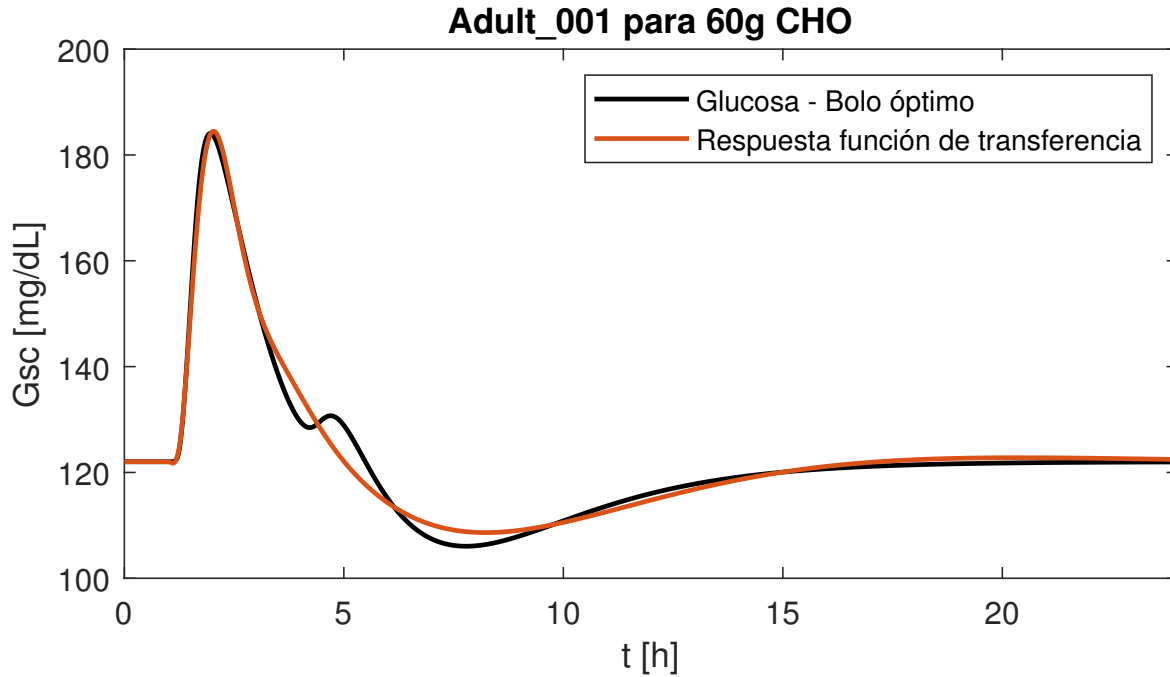


Figura 17. Comparación de la referencia generada a partir de la mejor aproximación de la regresión con la respuesta de glucosa generada por el bolo óptimo para el paciente *Adult_001*.

sea mayor o igual valor basal del paciente, para reducir cualquier tendencia a hipoglucemia en la referencia. En la figura 18 se muestra la curva de referencia final del paciente *Adult_001*.

Por tanto, en el presente trabajo de investigación se usaron 3 variaciones del algoritmo de control predictivo buscando probar diferentes grados de robustez y desempeño de control. Los controladores diseñados se pueden dividir en dos grupos, dependiendo como se use el modelo de predicción, MPC con modelo desacoplado y MPC con modelo acoplado.

5.2. MPC con modelo desacoplado

Para el diseño del primer controlador se abordó el enfoque QDMC, donde la principal característica del algoritmo consiste en usar el modelo de predicción para generar sus propias condiciones iniciales en cada iteración. Adicionalmente, este controlador aplica realimentación de salida y no

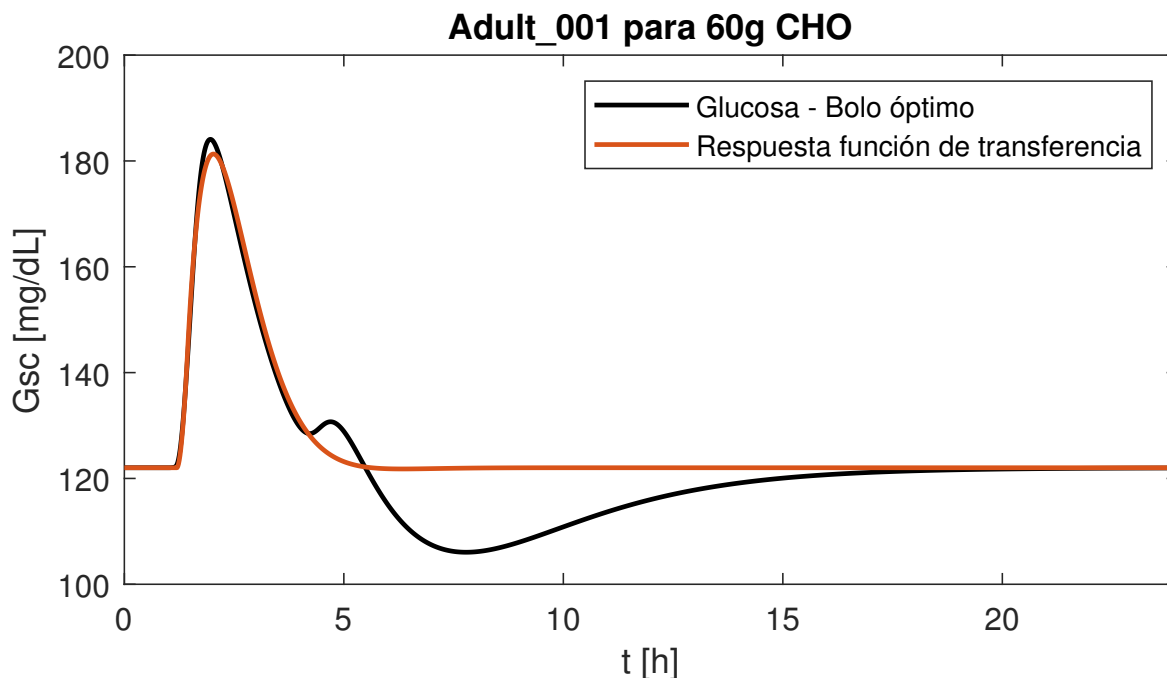


Figura 18. Comparación de la referencia generada a partir de la regresión redefinida con la respuesta de glucosa generada por el bolo óptimo para el paciente Adult_001.

de estados, por lo tanto, no necesita observador de estados. Las entradas del controlador son: la salida del proceso $y_p[k]$ (Glucosa subcutánea G_{sc} en [mg/dL]), la referencia $r[k]$ también en [mg/dL] y la perturbación de carbohidratos $CHO[k]$ en mg ; la salida del controlador será la acción de control $u[k]$ (dosificación de insulina en [pmol/min]). La clase creada para representar este controlador se denominó *MPC_QDMC* (todo lo referente a las clases creadas se expone en el ANEXO A). A continuación, se explica el funcionamiento del algoritmo de control, donde todos los algoritmos se diseñaron con dos fases de operación: la inicialización y el proceso iterativo.

Inicialización. El sistema recibe los parámetros de configuración (Periodo de muestreo T_s , Horizonte de predicción H_p , Horizonte de control H_u , Matrices de ponderación Q y R , y el nombre del archivo .mat correspondiente a los parámetros de algún paciente). Halla el modelo de

predicción, condiciones iniciales, la respuesta forzada, la matriz Hessiana y parte de las matrices de restricción.

Proceso iterativo.

- Se leen las entradas $CHO[k]$, $y_p[k]$ y $r[k]$.
- Se usan las condiciones iniciales $y_m[k-1|k]$ y $u[k-1|k]$ almacenadas en memoria y entrada de perturbación $CHO[k|k]$ para calcular la respuesta libre y_{libre} . Siguiendo el esquema de control QDMC, ya que no se conoce como evolucionará la perturbación el valor actual medido de esta, se mantendrá constante durante todo el horizonte de predicción H_p ; Este mismo concepto aplica al valor de referencia $r[k]$.
- Se calcula el término de corrección $d[k|k]$ mediante la ecuación 53.
- Se calcula el coeficiente lineal f de la función objetivo aplicando la ecuación 55.
- Se calcula el resto de la matriz b_c correspondiente a las restricciones en $u[k]$ y salida $y_p[k|k]$, que varían entre iteraciones.
- Se calcula $\Delta u[k|k]$ usando la función *quadprog* de MATLAB
- Se calcula $u[k|k]$ aplicando la ecuación 56.
- Se calculan las nuevas condiciones iniciales, $y_m[k|k+1]$ usando el modelo de predicción mediante la ecuación 57 y $u[k|k+1]$ será la acción de control $u[k|k]$ que es enviada al proceso al final de la iteración.

Este primer algoritmo de control predictivo es bastante convencional y no se aleja de los lineamientos de un controlador QDMC estándar. El objetivo de esto es poder comparar su desempeño con el de los siguientes algoritmos en los que se propusieron combinar otros conceptos y se proponen cambios que agregan originalidad al modo de calcular la acción de control.

5.3. MPC con modelo acoplado

Para el diseño del segundo controlador se abordó el enfoque de realimentación de estados con modelo acoplado. Es decir, las condiciones iniciales las genera un observador de estados a partir de las mediciones en la planta, mientras que como observador de estados se diseñó un filtro de Kalman extendido.

Se propone una modificación de este algoritmo, que consiste en no mantener constantes los valores de la referencia $r[k]$ y de perturbación $CHO[k]$ a lo largo de todo el horizonte de predicción (tal como lo plantea el algoritmo QDMC), y en su lugar, calcular la dinámica de estas variables a través de las predicciones. Por lo tanto, el algoritmo ya no recibe información externa de la referencia, sino que la calcula internamente.

Proyección de la perturbación. En la simulación de la ingesta de carbohidratos, esta se modela como un pulso con una duración de 15 minutos en promedio, donde su integral corresponde a la cantidad total de carbohidratos consumidos por el paciente in silico. Si se mantuviera el pulso durante todo el horizonte de predicción (tres horas por ejemplo), resultaría un falso consumo de doce veces más carbohidratos en comparación a los realmente consumidos. Por tanto, si el horizonte de control es mayor a 1, la acción de control se va a saturar cada vez que se detecte una

perturbación.

Para solucionar el problema de saturación de la acción de control, se diseñó un algoritmo que detecta cuando se presenta una perturbación, asumiendo que toda señal mayor a 500 [mg] se considera ingesta de alimentos, de lo contrario, será considerada ruido de medición. Cuando se detecta una ingesta, el algoritmo genera una señal interna que inicia con un pulso de 15 minutos, de altura igual a la perturbación medida y lo concatena con ceros hasta el final de la predicción. La información de cada perturbación medida almacena en registros y el algoritmo aplica el concepto de horizonte deslizante con la señal de perturbación generada. Es decir, en la siguiente iteración se usa la perturbación medida para actualizar la altura del pulso, que ahora durará 14 minutos y seguirá así sucesivamente hasta que la perturbación sea cero.

Cálculo de la referencia. El cálculo de la referencia se hace de forma similar a la proyección de la perturbación. Se aplica la ecuación recursiva que representa la función de transferencia discreta hallada en la sección 4.1.1, usando como entrada los valores de perturbación medida que se van almacenando en los registros durante la proyección de la perturbación. La ecuación recursiva se usa dos veces para calcular, por una parte la condición inicial que debe tener la referencia en la siguiente iteración y por otra la proyección de la referencia para el cálculo de la ley de control, con el fin mantener el concepto de horizonte deslizante.

En las figuras 19 y 18 se muestra como opera el algoritmo para proyectar la perturbación y la referencia a partir de las mediciones de perturbación. Únicamente se guarda en registro los valores diferentes de cero y estos registros son reiniciados cuando el valor de la condición inicial

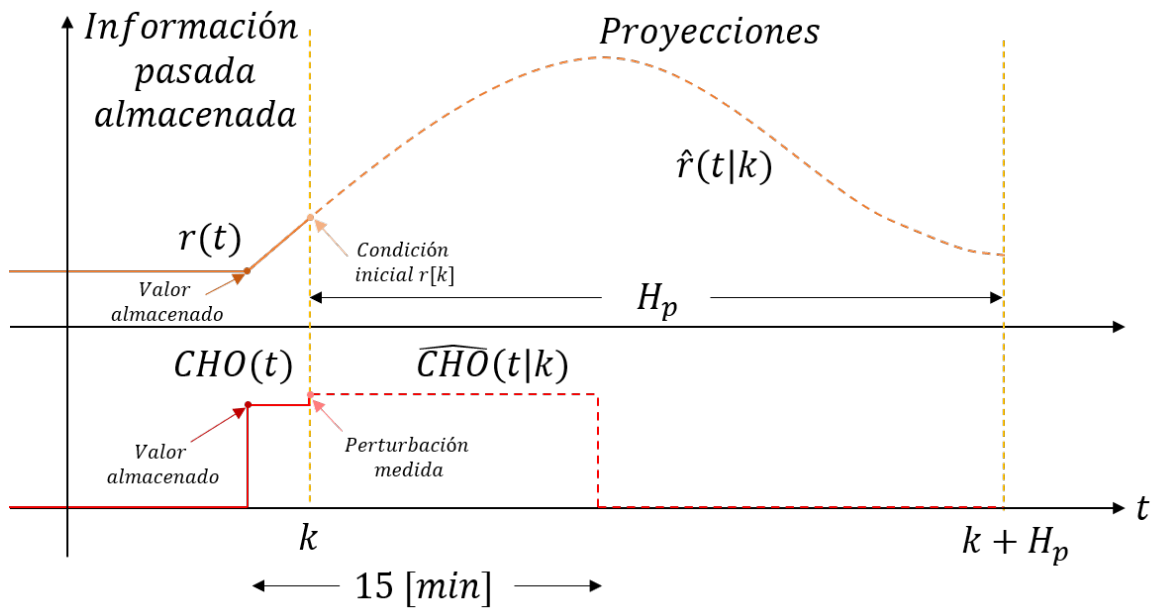


Figura 19. Algoritmo de proyección para iteración k .

de la referencia vuelve a ser menor o igual a la glucosa subcutánea basal.

Las entradas de este nuevo controlador son: los 13 estados que tiene en cuenta el modelo linealizado $x_p[k]$ y la perturbación de carbohidratos $CHO[k]$ en [mg]; la salida del controlador será la acción de control $u[k]$ (dosificación de insulina en [pmol/min]). La clase creada para representar este controlador se denominó *MPC_RP*, a continuación se explica como funciona su algoritmo de control.

Inicialización. El sistema recibe los parámetros de configuración (Periodo de muestreo T_s , Horizonte de predicción H_p , Horizonte de control H_u , Matrices de ponderación Q y R , y el nombre del archivo .mat correspondiente a los parámetros de algún paciente). Halla el modelo de predicción, la respuesta forzada, la matriz Hessiana y parte de las matrices de restricción.

- Se calcula $\Delta u[k|k]$ usando la función *quadprog* de MATLAB
- Se calcula $u[k|k]$ aplicando la ecuación 56.
- Se calculan la nueva condición inicial $u[k|k+1]$ que será la acción de control $u[k|k]$ que es enviada al proceso al final de la iteración.

Se diseñó un algoritmo adicional con modelo acoplado, esto se debe a que, se considera que la clase *MPC_RP* puede presentar error en estado estable si el modelo de predicción no es preciso, ya que no usa el coeficiente de corrección $d[k|k]$ en su algoritmo. Para solucionar este problema se decidió diseñar un algoritmo de control con integrador y se definió como la clase *MPC_RPI*.

Integrador. Para incluir el integrador en el sistema de espacio de estados de la clase *MPC_RPI* se definió una nueva matriz de estado $\xi[k]$.

$$\xi[k] = \begin{bmatrix} x[k] - x[k-1] \\ y[k-1] \end{bmatrix} \quad (73)$$

El nuevo sistema de espacio de estados aumentado queda de la siguiente forma.

$$G_i = \begin{bmatrix} G_m & 0_{13 \times 1} \\ C_m & 1 \end{bmatrix} \quad (74)$$

$$H_i = \begin{bmatrix} H_m \\ 1 \end{bmatrix} \quad (75)$$

$$C_i = \begin{bmatrix} 0_{1 \times 13} & 1 \end{bmatrix} \quad (76)$$

La forma en la que opera el algoritmo de control predictivo de la clase *MPC_RPI* es la misma de la clase *MPC_RP* usando el nuevo sistema de espacio de estados aumentado que incluye el integrador.

5.4. Estimador de perturbaciones

Se propuso un método de diseño para el estimador basado en clasificación de datos. Esta propuesta no sigue ninguna metodología establecida y su objetivo principal es evitar que el estimador dependa de un modelo bien definido para generar buenos resultados.

Para el diseño de los clasificadores se utilizó la toolbox *classificationLearner* de MATLAB. Debido a que el método es una propuesta, no se ahondó en la teoría de algún clasificador en particular, simplemente usando la toolbox se probaron todos y se seleccionó el que presentara los mejores resultados. Adicionalmente, fue necesario diseñar un sistema para filtrar y generar las señales adecuadas para la clasificación en presencia de ruido de medición. Inicialmente, se plantearon las variables de estimación o predictores.

5.4.1. Selección de predictores. Se ha evidenciado que existe una relación fuerte entre la ingesta de carbohidratos y el comportamiento de la glucosa. Por esta razón, se definió a la glucosa medida como predictor principal durante el entrenamiento del clasificador. Por otra parte, se asume que una sola señal de entrenamiento no es suficiente para obtener resultados satisfactorios. Adicio-

nalmente, los cambios en los valores de glucosa son muy lentos. Por lo tanto, se decidió usar sus derivadas como señales auxiliares. Cada derivada de la señal presenta variaciones en sus valores con mayor rapidez durante el transitorio. Por lo tanto, usar derivadas de orden elevado permitirá que el clasificador sea mucho más rápido. Sin embargo, si se halla la derivada de una señal ruidosa, la señal resultante tendrá un SNR considerablemente bajo, haciendo poco viable derivar muchas veces la señal original, debido a esto se decidió no ir más allá de la segunda derivada de la glucosa.

Al ser un algoritmo de clasificación que se basa en tendencias, el estimador necesita recibir una determinada cantidad de datos para determinar si hay o no hay perturbación. Por lo tanto, se produce un retardo de estimación y este retardo afectará el desempeño del controlador. Para saber que tanto puede disminuir el desempeño de un controlador predictivo ante una estimación de perturbación retardada, se realizaron pruebas preliminares suministrando a un controlador información de la perturbación original añadiendo diferentes retardos. El controlador con el que se hicieron todas las pruebas durante el diseño del estimador fue de la clase *MPC_QDMC*.

La figura 21 presenta la comparación de la glucosa cuando el controlador recibe información de la perturbación original, con la respuesta cuando recibe la información con diferentes retardos. Se evidencia que a mayor retardo de estimación, más tarda el controlador en suministrar el bolo de insulina y en consecuencia, el pico de glucosa crece. Por otra parte, que el estimador tenga poco retardo, implica que cuenta menos información de sus predictores para calcular una respuesta. Finalmente, se concluyó que se podía permitir un retardo de estimación de 15 minutos.

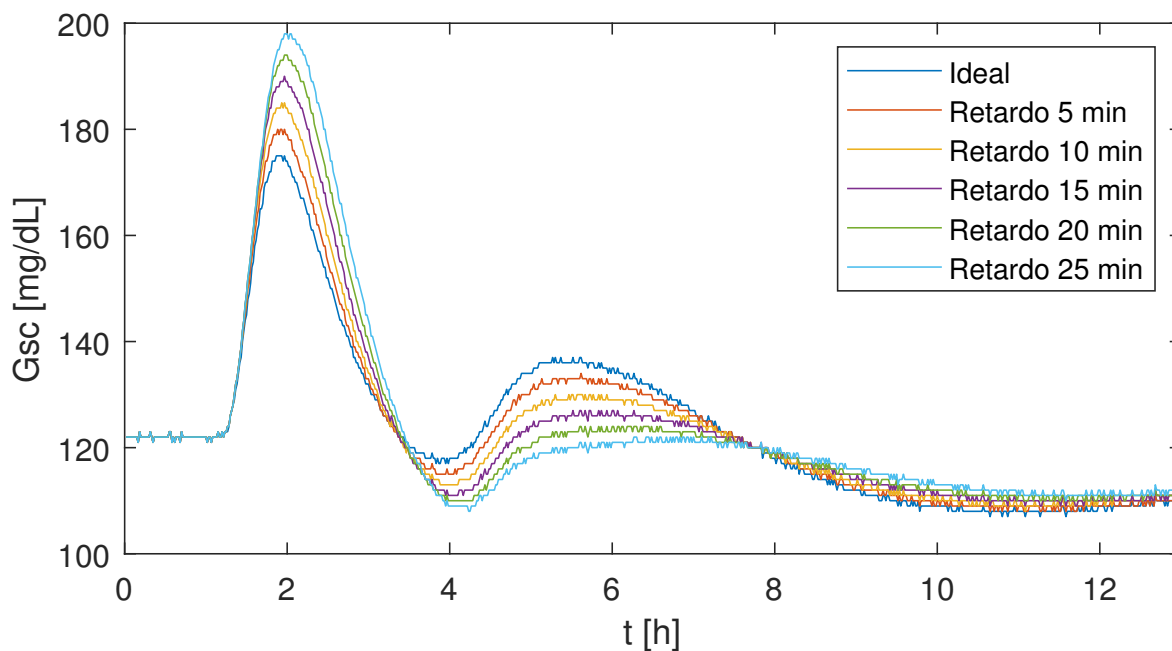


Figura 21. Comparación entre el control de glucosa con información de perturbación real (Ideal) e información retardada 5, 10, 15, 20 y 25 minutos.

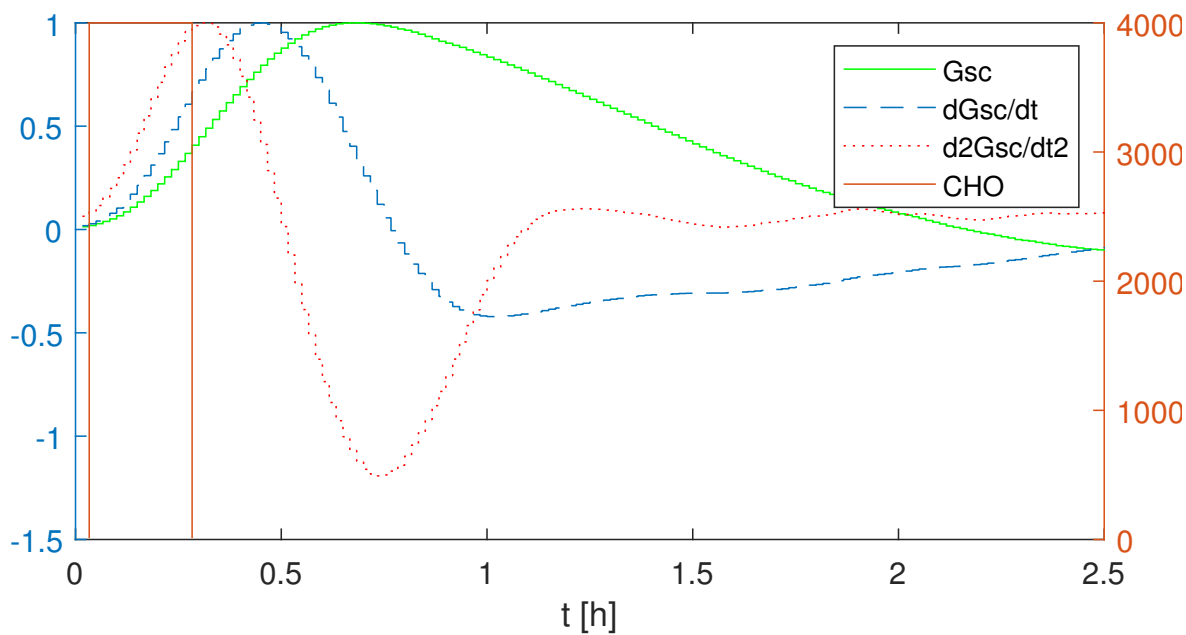


Figura 22. Comparación entre la perturbación de carbohidratos retardada 15 minutos (respuesta) con la señal de glucosa y sus dos primeras derivadas respectivamente normalizadas (predictores).

La figura 22 presenta la comparación realizada entre la perturbación de carbohidratos (CHO) retardada 15 minutos denominada *respuesta*, con señal de glucosa y sus dos primeras derivadas respectivamente normalizadas que llamaremos *predictores*. El objetivo de esta comparación es verificar que cuando los valores de perturbación sean distintos de 0, los predictores presenten una tendencia creciente en todo momento, para que no exista ambigüedad en la clasificación.

Finalmente, se entrenaron todos los clasificadores de la toolbox usando distintas combinaciones de los predictores (la señal de glucosa y sus primeras dos derivadas). Los resultados se agruparon en base a los predictores usados y se promediaron para determinar que la relevancia de cada señal durante la clasificación (Se eliminaron los datos de los clasificadores que arrojaran resultados mediocres alejados de la media). En la tabla 3 se muestran los resultados generados por las simulaciones.

Tabla 3

Resultados de error de cantidad, forma en la estimación de carbohidratos y forma de la referencia usando distintas combinaciones de predictores

Predictores	Error en la cantidad (CHO) (%)	Error de forma (CHO) (%)	Error de forma referencia (%)
G_{sc}	90.6	90.1	89.1
$\frac{dG_{sc}}{dt}$	90.7	90.2	89.3
$\frac{d^2G_{sc}}{dt^2}$	89.5	89.6	87.7
$G_{sc}, \frac{dG_{sc}}{dt}$	58.8	53.8	58.1
$G_{sc}, \frac{d^2G_{sc}}{dt^2}$	64.7	65.5	69.3
$\frac{dG_{sc}}{dt}, \frac{d^2G_{sc}}{dt^2}$	65.6	65.1	69.5
$G_{sc}, \frac{dG_{sc}}{dt}, \frac{d^2G_{sc}}{dt^2}$	35.1	37.3	35.9

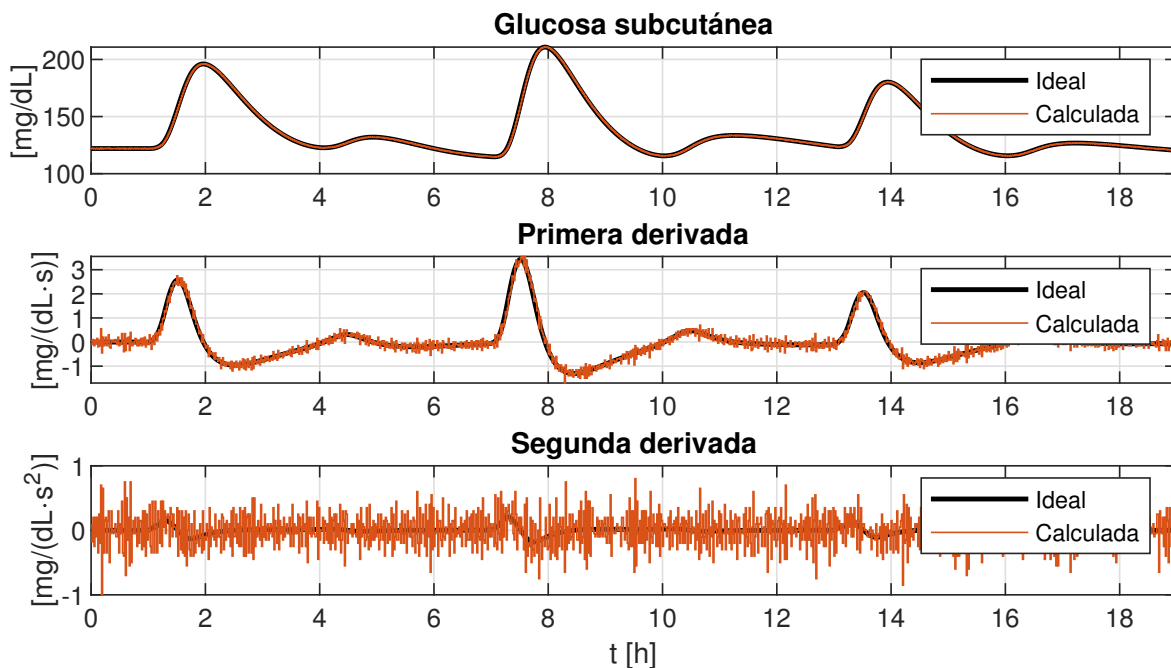


Figura 23. Predictores calculados sin sistema de adecuación para una ingesta de alimentos de 60, 80 y 50 [g] de carbohidratos a las 6 a.m., 12 m. y 6 p.m. respectivamente para el paciente *adult_001*.

En base a los resultados de la tabla 3 se decidió usar las 3 señales G_{sc} , $\frac{dG_{sc}}{dt}$, $\frac{d^2G_{sc}}{dt^2}$ para el diseño del clasificador. Inicialmente, para generar los predictores se tomó la salida de glucosa del sensor y se programó al algoritmo para que derive dos veces con un periodo de muestreo de 1 minuto (Las características del sensor se encuentran en el ANEXO B).

La figura 23 presenta una comparación de los 3 predictores generados a partir de la derivada digital con la señal de glucosa y sus derivadas sin ruido. A pesar de que el proceso de muestrear una señal filtra algo de ruido, los resultados no son aceptables. Por lo tanto, se decidió desarrollar un sistema más complejo.

5.4.2. Adecuación de señales. Para que el estimador pueda recibir predictores en condiciones satisfactorias, se diseñó un sistema de adecuación de señales que recibe la señal de glucosa del sensor la filtra y genera sus derivadas. Se usa un filtro de Kalman para generar una señal de glucosa mucho más limpia y su primera derivada. La primera derivada se deriva nuevamente de forma digital y se aplica un filtro de media móvil que limpia las señales.

Filtro de Kalman. Se diseñó un filtro Kalman lineal que usa el modelo discreto linealizado sin incluir los estados relacionados con el tracto digestivo (Q_{sto1} , Q_{sto2} y Q_{gut}), ya que no es conveniente que un sistema relacionado con el proceso de estimación dependa de los resultados de la estimación. La matriz de covarianza Rk es calculada hallando la covarianza de la señal del sensor en condición basal y la matriz de covarianza Qk se define como un valor elevado, para que el proceso de optimización de prioridad a la medición sobre la estimación.

Para evitar amplificar el ruido de la señal de glucosa medida al hallar su derivada numérica directa, se decidió hallar la primera derivada a partir de los estados estimados del filtro. Las salidas del filtro Kalman se calculan como sigue:

$$y[n] = C_m x[n] \quad (77)$$

$$dy[n] = \frac{y[n+1] - y[n]}{T_s} \quad (78)$$

$$dy[n] = \frac{C_m(G_m - I)x[n] + H_mu[n]}{T_s} \quad (79)$$

Donde G_m , H_m y C_m son las matrices que representan el modelo en espacio de estados linealizado y discretizado del paciente del simulador T1DMS. $y[n]$ representa la salida filtrada de glucosa y $dy[n]$ su primera derivada. Inicialmente, se hizo el cálculo de la segunda derivada dentro del filtro kalman. Sin embargo, los resultados de la derivada digital de $dy[n]$ aplicando el filtro de media móvil fueron mejores.

Filtro de media móvil. El filtro de media móvil o FMM, se desarrollo a partir de la función de transferencia en la ecuación 80.

$$\frac{y_{MMF}[n]}{x_{MMF}[n]} = 10 + 9z^{-1} + 8z^{-2} + 7z^{-3} + 6z^{-4} + 5z^{-5} + 4z^{-6} + 3z^{-7} + 2z^{-8} + z^{-9} \quad (80)$$

Las constantes del filtro se eligieron de forma empírica después de probar algunas definiciones para hallarlas. Se busca usar la información pasada para mejorar el SNR de la señal original alterando lo menor posible su forma. Aunque los valores máximo y mínimo de la señal se atenúan ligeramente, esto no es un problema para la estimación porque su forma permanece igual.

Se implementaron 2 filtros de media móvil en el sistema, uno para la salida de la primera derivada de glucosa del filtro Kalman y otro para la segunda derivada.

La figura 24 presenta una comparación de los 3 predictores generados con la señal de glu-

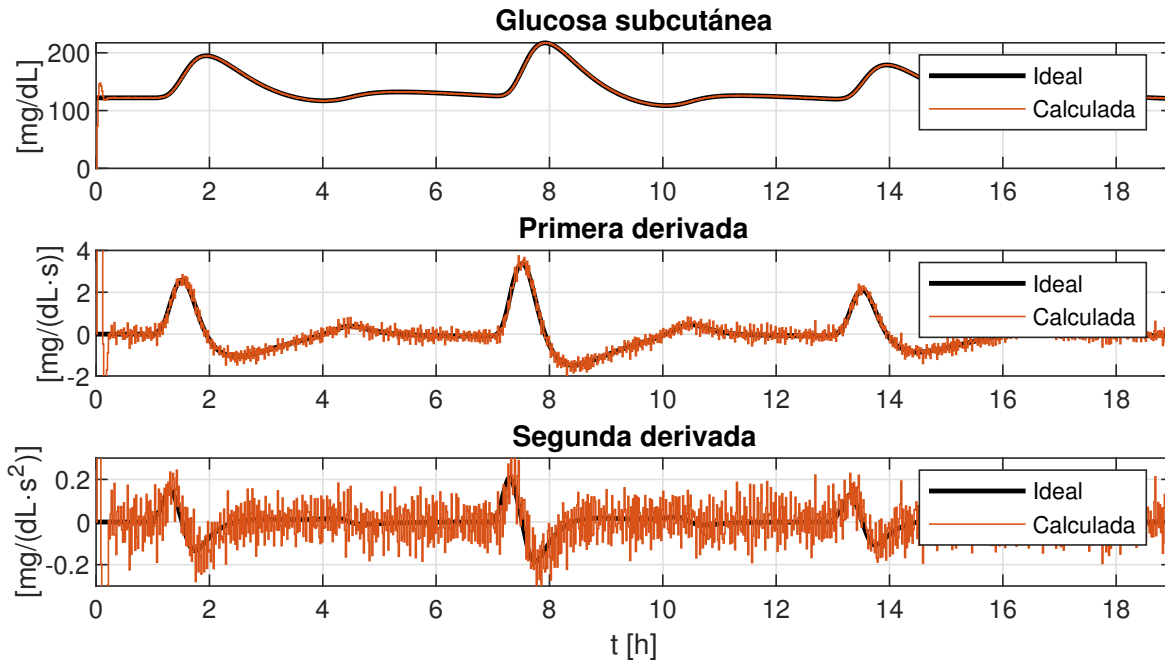


Figura 24. Predictores calculados usando el sistema de adecuación para una ingesta de 60, 80 y 50 [g] de carbohidratos a las 6 a.m., 12 m. y 6 p.m. respectivamente para el paciente adult_001.

cosa y sus derivadas sin ruido. Aunque el sistema de adecuación genera un pequeño retardo, no evita que el estimador calcule la perturbación en el margen de 15 minutos. Finalmente, se creó la clase *KALMAN_FMM* que agrupa los algoritmos del filtro de Kalman lineal, los filtros de media móvil, la segunda derivada y corrige problemas de lazos algebraicos que se generan porque el estimador depende de la acción de control y a su vez, la acción de control depende del estimador. El desempeño de estos filtros se comparó con el bloque derivador presentado en el artículo *The robust exact differentiator toolbox: Improved discrete-time realization* Reichhartinger et al. (2018), donde el sistema diseñado generó mejores resultados.

5.4.3. Selección del clasificador. Con los predictores y la señal de respuesta definidas, se hicieron varias simulaciones contrastando los 5 modelos de clasificación que mostraron el mejor desempeño en las pruebas de los predictores. Los modelos de clasificadores que se usaron en las pruebas finales fueron: fine tree, quadratic discriminant, gaussian support vector machine, weighed k-nearest neighbors y Bagged trees. En la tabla 4 se muestran los resultados generados por las simulaciones.

Tabla 4

Resultados de error de cantidad, forma en la estimación de carbonhidratos y forma de la referencia usando distintos clasificadores

Clasificadores	Error en la cantidad (CHO) (%)	Error de forma (CHO) (%)	Error de forma referencia (%)
Fine tree	35.4	34.3	32.7
Quadratic discriminant	286.3	85.2	81.2
Gaussian SMV	27.3	31.6	34.3
Weighed KNN	3.5	5.2	4.1
Bagged trees	12.3	16.5	13.6

Finalmente, se eligió el clasificador weighted KNN ya que presentó los mejores resultados.

6. Validación in silico

6.1. Modelos in silico

A partir del sistema de ecuaciones diferenciales del modelo *UVA_PADOVA* y con ayuda del software T1DMS se generaron 3 modelos para el desarrollo y validación de sistemas de control.

Modelo continuo no lineal. El primero consiste una versión actualizada de un sistema de bloques en simulink, inicialmente desarrollado en el trabajo de grado: Simulación en tiempo real de la dinámica de glucosa para diferentes pacientes con diabetes mellitus tipo I, orientada a la implementación “Hardware in the loop” Rodriguez et al. (2018), en el que se explica el desarrollo de sistema de bloques y se hacen pruebas para medir su exactitud con respecto al software T1DMS, este sistema de bloques que inicialmente se desarrollo usando las ecuaciones del modelo UVA PADOVA en su versión del año 2013 Dalla Man et al. (2014), fue ligeramente actualizado a la versión del 2017 Visentin et al. (2018), como resultado se presenta un modelo de naturaleza continua que presenta errores porcentuales menores al 1 % respecto al simulador T1DMS.

Modelo discreto no lineal. El segundo producto consiste en un script basado en programación orientada a objetos en Matlab, que es la versión final de la clase *UVA_PADOVA_2*, que fue desarrollada a partir de las ecuaciones de la versión final del modelo continuo. Por ser de naturaleza discreta, este modelo requiere de un método de integración para generar su salida, por lo general se aplica un retenedor de orden cero que es equivalente a usar el método de Runge-Kutta de primer orden, conocido como el método de Euler. Sin embargo, buscando mayor precisión en la respuesta de este modelo discreto, teniendo en cuenta que se enfrenta a dinámicas con fuertes

no linealidades se decidió usar de Dormand Prince como método de integración, que es el método de integración por defecto de MATLAB.

Programando el método Dormand-Prince: Es un método para solucionar ecuaciones diferenciales ordinarias que pertenece a la familia de métodos de Runge-Kutta. Halla soluciones de cuarto y quinto orden a partir de la evaluar seis veces la función $f(t, y)$ Dormand (1996).

La forma en que se halló el valor para el siguiente paso de la ecuación diferencial sigue esta fórmula:

$$y_{n+1} = y_n + h \sum_{i=1}^6 b_i K_i \quad (81)$$

Donde h representa el paso, que será el periodo de muestreo asociado al objeto objeto, K los coeficientes resultantes de evaluar la función.

$$K_i = f(t_n + c_i h, y_n + h \sum_{j=1}^6 a_{ij} K_j) \quad (82)$$

La imagen 25 la matriz de Butcher con los coeficientes a , b y c correspondientes al método Dormand and Prince (1980). Como se mencionó, este método genera una respuesta de cuarto orden y una de quinto orden, por esta razón resultan 2 filas de coeficientes b_j , donde la primera es la respuesta de quinto orden y la segunda la de cuarto orden. En tesis se usaron ambas respuestas, la clase `UVA_PADOVA_2` usa la respuesta de cuarto orden y otra clase creada, denominada `UVA_PADOVA_3` usa la de quinto orden. No se notó diferencia en los resultados de ambas clases,

por lo tanto, la clase usada para el desarrollo de la tesis fue *UVA_PADOVA_2*.

c_i	a_{ij}						
1/5	1/5						
3/10	3/40	9/40					
4/5	44/45	-56/15	32/9				
8/9	19372/6561	-25360/2187	64448/6561	-212/729			
1	9017/3168	-355/33	46732/5247	49/176	-5103/18656		
1	35/384	0	500/1113	125/192	-2187/6784	11/84	
b_j	5179/57600	0	7571/16695	393/640	-92097/339200	187/2100	1/40
	35/384	0	500/1113	125/192	-2187/6784	11/84	0

Figura 25. Matriz de Butcher correspondiente al método Dorman-Prince.

La clase *UVA_PADOVA_2* recibe como entradas la perturbación de carbohidratos consumidos durante la ingesta *CHO* y la insulina suministrada al paciente *u*, para genera los 16 estados del modelo diabético como salida.

Modelo discreto lineal. Teniendo en cuenta que se emplearon sistemas de naturaleza lineal, con el objetivo de hacer control en la dosificación de insulina y para estimar variables, se planteo un modelo lineal a partir de las ecuaciones especificadas en la sección 2.5, para el que se definieron las siguientes condiciones de operación:

- Las ecuaciones asociadas a la cinética y secreción del glucagón plasmático y subcutáneo no se tuvieron en cuenta, porque en ningún momento se inyectará glucagon de forma exógena durante la operación.
- Se asume que el sistema de control mantendrá al paciente in silico en condiciones de operación que eviten la secreción de glucagón interna, que se presenta cuando la concentración de glucosa plasmática baja demasiado, por lo tanto, se definió el valor del estado \dot{X}^H en cero (0).

- El valor de la constante de vaciado gástrico K_{empt} , que depende de una función que presenta fuertes no linealidades, se aproximó a un valor constante para cada paciente in silico partir de una regresión usando la toolbox idnlgrey.
- El valor de la variable $Risk$ es diferente de cero únicamente en condiciones de hipoglucemia, por lo tanto, se asumió que siempre será cero porque el sistema de control debe evitar esta condición.
- Se asumió que no habrá eliminación de glucosa en los riñones, porque el sistema de control debe evitar que la concentración de glucosa plasmática llegue a los niveles elevados donde se presenta.

En estas condiciones de operación el modelo se simplificó considerablemente, pasando de 16 a 13 estados y reduciendo elementos en algunas ecuaciones de estado. Después del ajuste, la única ecuación de estado que aún presenta no linealidades es la de \dot{G}_t , con el fin de linealizar esta ecuación, se aplican series de Taylor y se obtiene la ecuación (83).

$$\dot{G}_t(t) = k_1 G_p(t) + A_1 G_t(t) + A_2 X(t) + A_3 \quad (83)$$

Donde A_1 , A_2 y A_3 son respectivamente:

$$A_1 = -\frac{(V_{m0} + V_{mx} X_{op})}{(K_{m0} + G_{top})^2} K_{m0} - k_2 \quad (84)$$

$$A_2 = -\frac{V_{mx}G_{top}}{K_{m0} + G_{top}} \quad (85)$$

$$A_3 = \frac{(V_{m0} + V_{mx}X_{op})}{(K_{m0} + G_{top})^2} K_{m0} G_{top} - \frac{V_{m0}G_{top}}{K_{m0} + G_{top}} \quad (86)$$

Los 13 estados que representan la dinámica del modelo de glucosa linealizado son presentados en el vector (87).

$$x = [G_p, G_t, I_p, I_l, Q_{sto1}, Q_{sto2}, Q_{gut}, X^L, I', X, I_{sc1}, I_{sc2}, G_S]^T \quad (87)$$

Debido a que los sistemas aplicados en esta tesis son de naturaleza discreta, las matrices de estado A , B , C y D que resultan del modelo linealizado se discretizan aplicando el método exacto con retenedor de orden cero.

6.2. Sintonización de Controladores

Se realizaron varios conjuntos de pruebas preliminares buscando sintonizar los controladores. Inicialmente, se estableció un periodo de muestreo aplicando el criterio de $\tau/10$, asumiendo un transitorio de cuatro horas se calculó un periodo de muestreo de 6 minutos. Sin embargo, se decidió usar un periodo de muestreo más rápido porque durante las pruebas del estimador, se evidenció que periodos lentos empeoran su desempeño; se diseñó el estimador para que el retardo de estimación sea de 15 minutos, donde un periodo de muestreo de 6 minutos representa aproximadamente 2

muestras para estimar la perturbación. Por lo tanto, se decidió bajar el periodo de muestreo a 1 minuto; un periodo de muestreo menor a este puede generar problemas en la ley de control predictivo, ya que por definición el microprocesador debe ser capaz de hacer todos los cálculos en un tiempo máximo equivalente al diez por ciento del periodo de muestreo, por esta razón elegir tiempos menores a un minuto tampoco resulta conveniente Maciejowski (2002). Para todas las simulaciones de esta sección se envía al controlador la información de la perturbación ideal, esto se hizo con el fin de verificar el desempeño del algoritmo de control independientemente de la estimación de perturbaciones.

Sintonización de parámetros. Se sintonizó de forma empírica el controlador basado en la clase *MPC_QDMC*, buscando comprender el efecto de cada parámetro en la respuesta y de esta manera poder acotar el proceso de optimización para la selección de parámetros. Para esto, se desarrollo un algoritmo de optimización de parámetros por “búsqueda ciega”, permitiendo elegir rangos de valores para los parámetros, simular todos los controladores dentro de estos rangos y elegir el que proporcione la respuesta con menor error cuadrático de seguimiento. A su vez, se penaliza que la respuesta del controlador es por encima de 1 [mg/dL] del valor máximo de referencia y por debajo de 1 [mg/dL] del valor mínimo de la referencia. En principio, también se optimizaba el esfuerzo de control, pero se evidenció una correlación entre ambos elementos, es decir, un mejor seguimiento de la referencia tiende a generar un menor consumo de insulina. La tabla 5 muestra los valores óptimos encontrados para el controlador basado en la clase *MPC_QDMC* para el paciente *adult_001* simulado a partir de la clase *UVA_PADOVA_2*, el controlador también usa como modelo al paciente *adult_001*. Para las pruebas de control de glucosa, se simuló un sensor de glucosa

subcutáneo que agrega ruido blanco de amplitud 0.4 [mg/dL].

Tabla 5

H_p , H_u , Q y R óptimos, usando el algoritmo de optimización por búsqueda ciega

Parámetros	Valores óptimos
Horizonte de predicción (H_p)	280
Horizonte de control (H_u)	1
Ponderación de seguimiento (Q)	1
Ponderación de esfuerzo de control (R)	20000

En la figura 26 se presenta el algoritmo de optimización simulando una ingesta de alimentos de 50 , 80 y 60 [g] de carbohidratos a las 6 a.m., 12 m. y 6 p.m. respectivamente. La gráfica presenta la curva de referencia generada comparada con la salida de glucosa, especificando el error IAE dividido en el número de muestras que se denominó IAEP. También muestra la dosificación de insulina indicando la insulina total administrada al paciente.

En MATLAB este algoritmo de control presentó un retardo de procesamiento por iteración de entre 25 y 50 [ms] cuando el controlador no detectaba perturbación; y de 200 a 300 [ms] en presencia de perturbación. Se debe tener en cuenta que para que el modelo de predicción no pierda precisión, el retardo por procesamiento no debe superar el 10% del periodo de muestreo Maciejowski (2002), es decir 6 segundos. Los resultados se consideran favorables ya que el caso crítico requiere del 0.5% del tiempo de muestreo, es decir, 20 veces menos del tiempo permitido. Sin embargo, estas pruebas se están realizando en un computador que cuenta con un procesador de

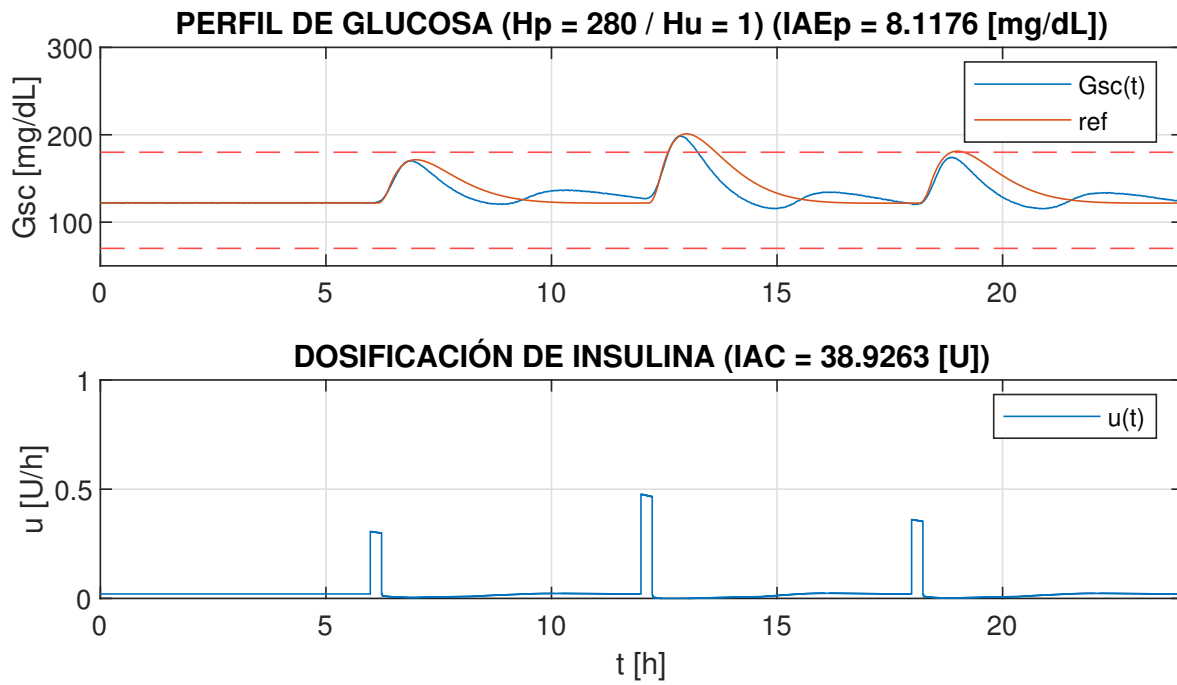


Figura 26. Gráfica generada por el algoritmo de optimización para la clase MPC_QDMC simulando una ingesta de alimentos de 50, 80 y 60 [g] de carbohidratos a las 6 a.m., 12 m. y 6 p.m. respectivamente para el paciente *adult_001*.

3[GHz] de frecuencia de reloj. Sin embargo, se espera programar el controlador en una unidad de procesamiento como por ejemplo una Raspberry PI 4, una ESP32 o un Arduino UNO, que tienen procesadores con frecuencias de reloj de 1.5, 0.24 y 0.016 [GHz] respectivamente. Por lo tanto, usando alguno de estos dispositivos se cuenta con un 50%, un 8% o un 0.53% del procesamiento con el que se realizaron las pruebas. Por lo tanto, se decidió buscar la manera de optimizar el algoritmo.

Después de realizar pruebas en todos los pacientes in silico usando distintas configuraciones en los parámetros del controlador, se concluyó que no es conveniente especificar restricciones a la salida para el modelo UVA PADOVA. Esto se debe a dos condiciones que presenta este problema de control, por una parte el fuerte retardo que presenta el modelo y por otra el hecho de haber configurado un control de seguimiento de referencia. Estas características combinadas hacen irrelevante definir restricciones a la salida del proceso. Se creó la clase *MPC_QDMC2*, que corrige el algoritmo para quitar las restricciones a la salida para compararla con los mismos pacientes y parámetros que la clase *MPC_QDMC*.

La figura 27 muestra los resultados de simulación de la nueva clase *MPC_QDMC2* en las mismas condiciones especificadas anteriormente para la clase *MPC_QDMC*. Como se ve en la figura, no existen diferencias relevantes además del aumento del 11.7% en el error IAEP, que no representa un peor comportamiento, ya que no se ven cambios en los picos máximo y mínimo de glucosa, solo se presentan diferencias durante el transitorio de regreso al estado basal. Por otra parte, el tiempo de procesamiento por iteración en ausencia de perturbación registrado fue de en-

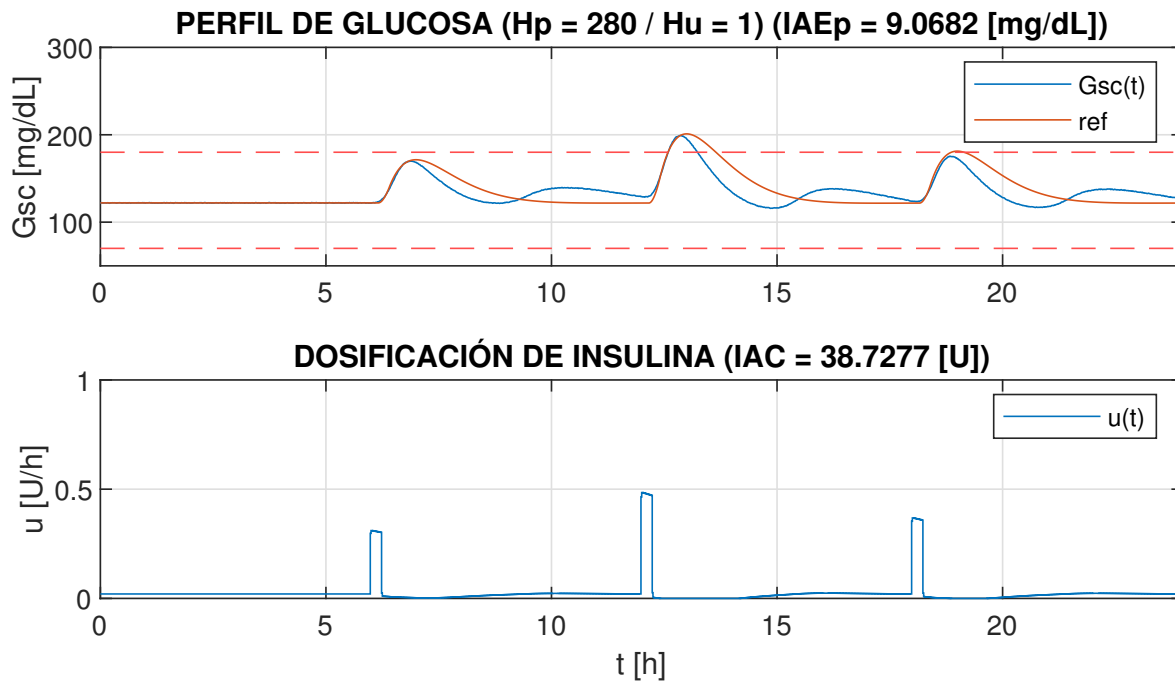


Figura 27. Gráfica generada por el algoritmo de optimización para la clase *MPC_QDMC2* simulando una ingesta de alimentos de 50, 80 y 60 [g] de carbohidratos a las 6 a.m., 12 m. y 6 p.m. respectivamente para el paciente *adult_001*.

tre 4 a 7[ms]; y se mantuvo así en presencia de perturbaciones. Por lo tanto, ahora el caso crítico usa únicamente 0.0116 % del periodo de muestreo, aproximadamente 862 veces menos del tiempo permitido. Por otra parte, que en esta prueba no se vea un aumento del tiempo de procesamiento en presencia de perturbación puede significar que la operación más costosa computacionalmente en el algoritmo de MATLAB, ya no sea la optimización con *quadprog*, sino por ejemplo, estar imprimiendo datos en pantalla, una operación que no es necesaria a la hora de implementar el controlador. En resumen, modificando el algoritmo para no tener en cuenta las restricciones a la salida, ahora es viable programar el controlador en cualquiera de las unidades de procesamiento mencionadas. Por lo tanto, también se modificaron las clases *MPC_RP* y *MPC_RPI* para quitar las restricciones a la salida de su función objetivo.

Con esta modificación, el nuevo algoritmo permite establecer horizontes de predicción y control más amplios. Esto se concluye a partir de una prueba en la que al seleccionar un H_p de 1000 pasos, con un H_u de 100 pasos, el controlador presentó un tiempo de procesamiento crítico de 300[ms], es decir, los rangos de búsqueda de H_p y H_u se podían ampliar por lo menos 5 veces cada uno. Por lo tanto, no era viable continuar optimizando a partir del algoritmo por búsqueda ciega, por lo que se desarrollaron 2 algoritmos adicionales de optimización. El primer algoritmo usa la función *fmincon* de *MATLAB* para buscar los parámetros óptimos, pero al trabajar con una función que utiliza un solo agente de búsqueda, el resultado cae fácilmente en mínimos locales. Por esta razón, se creó un segundo algoritmo que usa la función *particleswarm* de *MATLAB*, que aplica el algoritmo PSO (*Particle Swarm Optimization*) que trabaja con múltiples agentes de búsqueda pudiendo evitando

los mínimos locales que presenta el problema de optimización.

Tabla 6

Parámetros óptimos H_p , H_u , Q and R usando el algoritmo de optimización PSO para el controlador basado en la clase MPC_QDMC

Parámetros	Valores óptimos
Horizonte de predicción (H_p)	371
Horizonte de control (H_u)	1
Ponderación de seguimiento (Q)	5
Ponderación de esfuerzo de control (R)	8

La tabla 6 muestra los valores de los parámetros de sintonización calculados por el algoritmo PSO para la clase *MPC_QDMC2*. Sin embargo, el optimizador no calcula valores de horizonte de control superiores a 1 paso, esto se debe a que estos valores de H_u saturan la acción de control, provocando estados de hipoglucemia en el paciente simulado.

La figura 28 presenta la gráfica con los resultados del controlador *MPC_QDMC2* con los nuevos parámetros optimizados. Se evidencia que el error IAEP es menor, así como la inyección de insulina. Finalmente, se aplicó el algoritmo PSO para optimizar los parámetros de los demás controladores. Se programó la clase *EKF_UVA* con el algoritmo del filtro de Kalman extendido para estimar los estados de los controladores basados en las clases *MPC_RP* y *MPC_RPI*, cuyas matrices de covarianza se definieron como $R_k = 0.01$ y $Q_k = 1 \cdot 10^{-4}$.

La tabla 7 presenta los valores calculados por el algoritmo PSO para la clase *MPC_RP*. Se observa que esta clase permite horizontes de control superiores a 1 gracias a la forma en la que

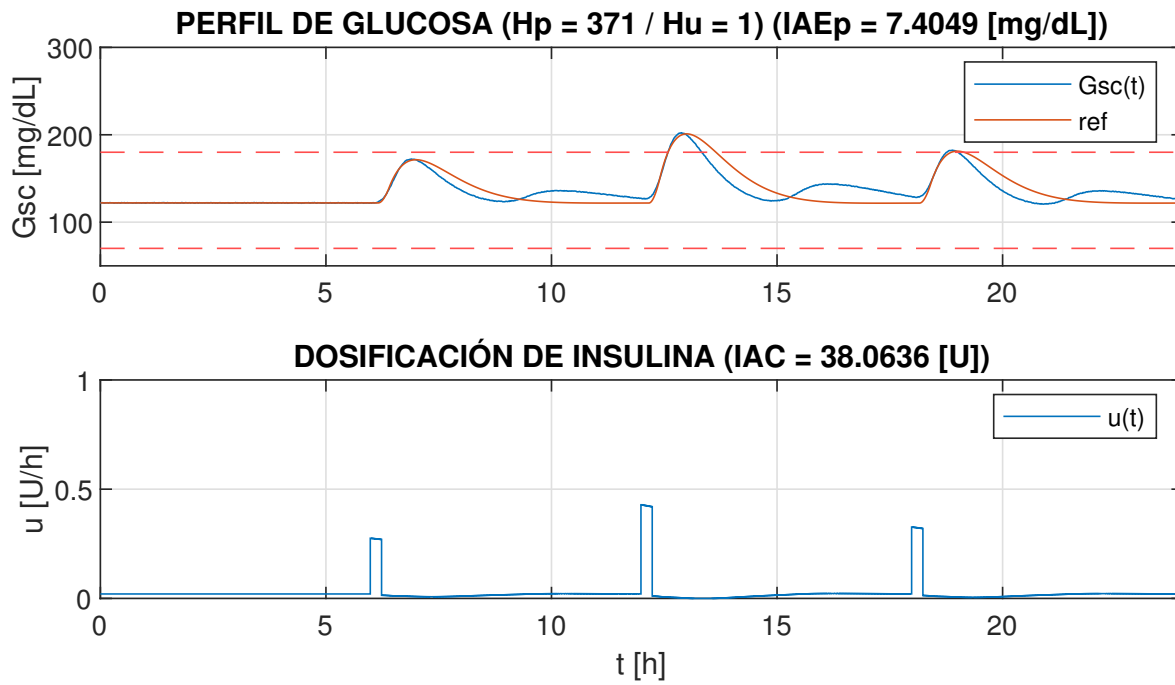


Figura 28. Gráfica generada por el algoritmo de optimización para la clase *MPC_QDMC2* simulando una ingesta de alimentos de 50, 80 y 60 [g] de carbohidratos a las 6 a.m., 12 m. y 6 p.m. respectivamente para el paciente *adult_001*.

Tabla 7

Parámetros óptimos H_p , H_u , Q and R usando el algoritmo de optimización PSO para el controlador basado en la clase MPC_RP

Parámetros	Valores óptimos
Horizonte de predicción (H_p)	950
Horizonte de control (H_u)	20
Ponderación de seguimiento (Q)	1
Ponderación de esfuerzo de control (R)	20000

proyecta las perturbaciones para hacer las predicciones, evitando saturar la acción de control.

La figura 29 presenta la gráfica con los resultados del controlador de la clase *MPC_RP* con los parámetros optimizados. Este controlador calcula una acción de control más agresiva, que genera una respuesta con menor error IAEP y con picos de glucosa menores en comparación con la respuesta de la clase *MPC_QDMC2*. Por otra parte, la inyección de insulina total fue mayor. En cuanto a la clase *MPC_RPI*, los resultados fueron insatisfactorios y no se presentan en el documento. Esto se debe a su definición de integral, ya que no basta con incluir el integrador en la matriz aumentada, se debe modificar también el calculo de las predicciones y la definición de las restricciones; se probaron diversas modificaciones sin éxitos, por lo tanto, queda por revisar este algoritmo en trabajos futuros.

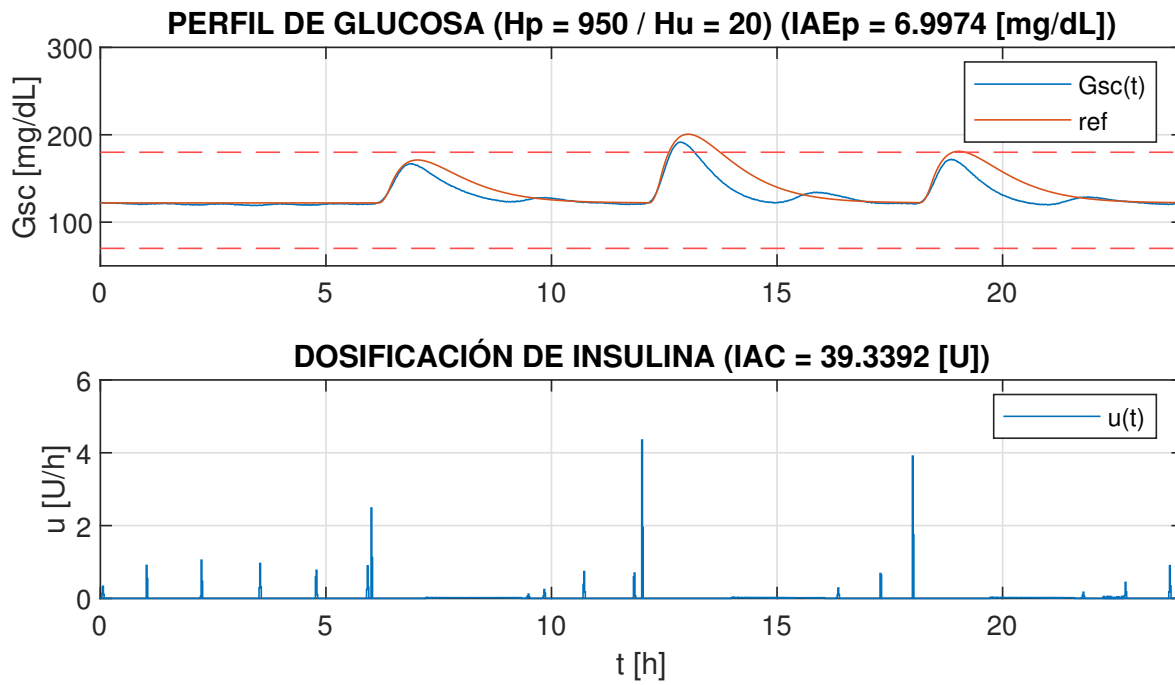


Figura 29. Gráfica generada por el algoritmo de optimización para la clase MPC_QDMC2 simulando una ingesta de alimentos de 50, 80 y 60 [g] de carbohidratos a las 6 a.m., 12 m. y 6 p.m. respectivamente para el paciente *adult_001*.

6.2.1. Pruebas con modelo nominal. Con los controladores sintonizados se realizaron las pruebas de validación de estabilidad y comportamiento nominal sobre la cohorte de 10 pacientes adultos. El desempeño de un controlador para diabetes mellitus tipo I generalmente se mide en relación con los límites de hiperglucemia e hipoglucemia. Para la hiperglucemia se espera que el controlador mantenga la glucosa por debajo de 180 [mg/dL], que es un límite conservador, ya que por lo general es difícil de cumplir para ingestas superiores a 50 [g] de carbohidratos. Por esta razón, lo que se midió respecto a hiperglucemia es que tanto se supera el límite de 180 [mg/dL] y durante cuanto tiempo. Por otra parte, que el controlador permita que la glucosa del paciente este por debajo de 70 [mg/dL] que es límite de hipoglucemia, se considera insatisfactorio.

Control mediante bolo óptimo. Se tomaron como referencia los resultados producidos por el control mediante bolo óptimo del simulador T1DMS, esto se debe a que si esta estrategia se aplica en condiciones ideales, presenta resultados al nivel de las mejores estrategias para el control de la diabetes mellitus tipo I; para sostener esta afirmación se tomaron como referencia los resultados de estrategias de control avanzado de algunos de los artículos más relevantes respecto al control de glucosa de pacientes in silico del simulador T1DMS Chakrabarty et al. (2017); El Hachimi et al. (2019); Ho (2019).

La figura 30 presenta los resultados del control mediante bolo óptimo para cada paciente adulto in silico. A pesar de que no se registran violaciones a nivel de hipoglucemia, la glucosa llega a estar por debajo de los 100 [mg/dL], concretamente su valor mínimo es 88.62 [mg/dL] que se toma como un punto de referencia para comparar la tendencia a hipoglucemia de cada controlador,

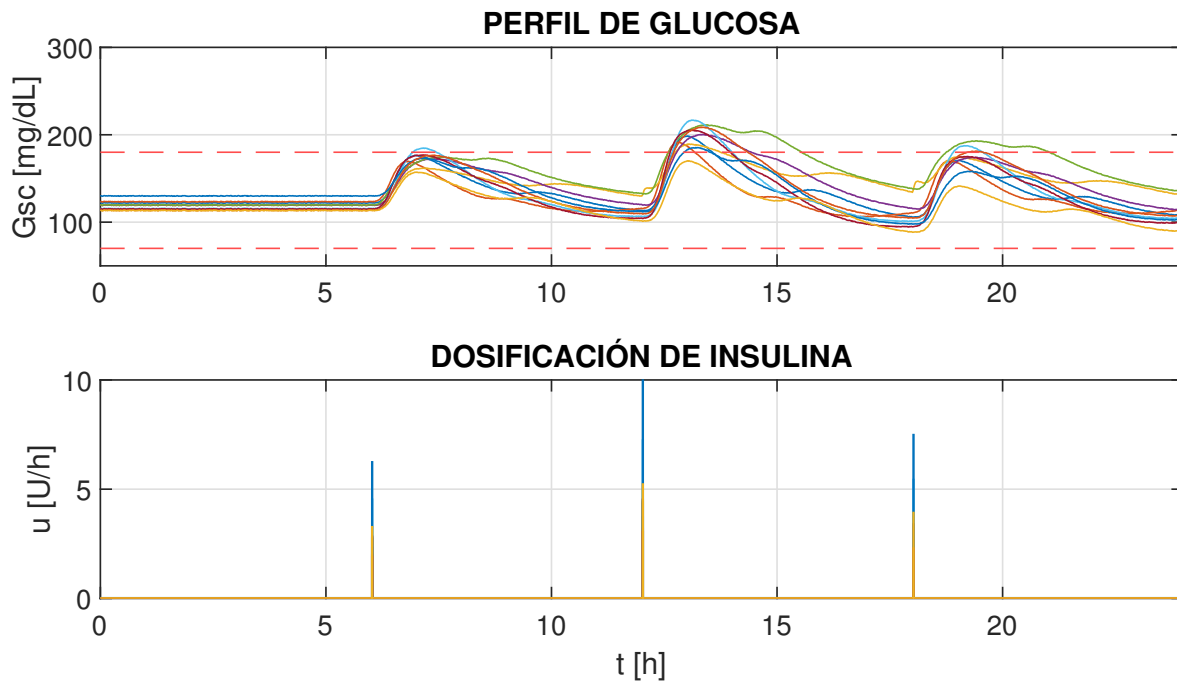


Figura 30. Resultados del control mediante bolo óptimo del simulador T1DMS ante una ingesta de alimentos de 50, 80 y 60 [g] de carbohidratos a las 6 a.m., 12 m. y 6 p.m. respectivamente para los pacientes adultos in silico.

debido a que se prefiere que la glucosa permanezca siempre por encima del valor basal del paciente. Por otro lado, el límite de hiperglucemia se supera varias veces, sin embargo, estos resultados del bolo óptimo sugieren un excelente desempeño ya que es similar al de los controladores de los artículos citados.

Tabla 8

Medidas de hiperglucemia aplicando control mediante bolo óptimo al banco de adultos in silico, comparación de pico de glucosa (PG) en [mg/dL] y tiempo del paciente en hiperglucemia en minutos.

Hiperglucemia						
Adultos	6am		12pm		6pm	
Paciente	PG	Tiempo	PG	Tiempo	PG	Tiempo
1	176.15	-	198.32	54	170.30	-
2	169.30	-	191.22	35	169.93	-
3	161.59	-	189.10	61	174.36	-
4	172.33	-	200.01	99	174.11	-
5	174.34	-	210.95	177	192.88	137
6	184.55	32	216.68	80	187.50	39
7	176.68	-	205.09	80	175.07	-
8	173.08	-	185.24	35	158.11	-
9	176.50	-	208.36	99	181.12	21
10	157.03	-	169.90	-	141.02	-

La tabla 8 presenta los resultados de hiperglucemia de los pacientes in silico. En la ingesta del desayuno (6 a.m, 50 [g] de carbohidratos) se registró una violación al límite de hiperglucemia con un pico máximo de glucosa (PMG) de 184.55 [mg/dL] que duró 32 minutos, el pico promedio de glucosa (PPG) fue de 172.15 [mg/dL]; para la ingesta del almuerzo (12 p.m, 80 [g] de carbohidratos) en contraste con la del desayuno, únicamente el adulto 10 no presentó hiperglucemia, sin

embargo, esta ingesta es considerablemente alta, por lo que el único resultado que se considera insatisfactorio fue el del adulto 5, que se mantuvo por 3 horas en estado de hiperglucemia. con niveles de glucosa sobre 200 [mg/dL]. El PMG y PPG registrados fueron 216.68 y 197.48 [mg/dL] respectivamente. Finalmente, en la ingesta de la comida (6 p.m, 60 [g] de carbohidratos) el adulto 5 nuevamente tiene una elevada duración en estado de hiperglucemia, demorando 2 horas 17 minutos, sin embargo, su PG llegó a 192.88 [mg/dL], por lo que no se considera insatisfactorio. Ya que el PPG fue de 172.44 [mg/dL] los todos resultados se consideran satisfactorios. A partir de este análisis de desempeño, se definen como resultados insatisfactorios a los pacientes que duren 2 horas o más en estado de hiperglucemia si su PG supera los 200 [mg/dL]. Como medida adicional no se permite superar en más de 30% el valor del límite de hiperglucemia, es decir 234 [mg/dL], este límite se estableció en base a que es aproximadamente el valor más elevando al que llega la glucosa aplicando las estrategias de control de los artículos citados Chakrabarty et al. (2017); El Hachimi et al. (2019); Ho (2019).

Controlador basado en la clase MPC_QDMC2. La figura 31 presenta los resultados de las validaciones in silico, usando el controlador basado en la clase *MPC_QDMC2* con modelo nominal para el banco de pacientes adultos. El nivel mínimo de glucosa registrado fue de 105 [mg/dL], esto representa un margen de 16.38 [mg/dL] a favor en comparación con el bolo óptimo, lo que asegura mayor prevención de hipoglucemia.

La tabla 9 presenta valores relevantes para el análisis de hiperglucemia de los pacientes. En la ingesta del desayuno no se detecta hiperglucemia, siendo 174.46 [mg/dL] el PMG y 165.46 [mg/dL] el PPG; En la ingesta del almuerzo, se detecta hiperglucemia en los mismos pacientes que

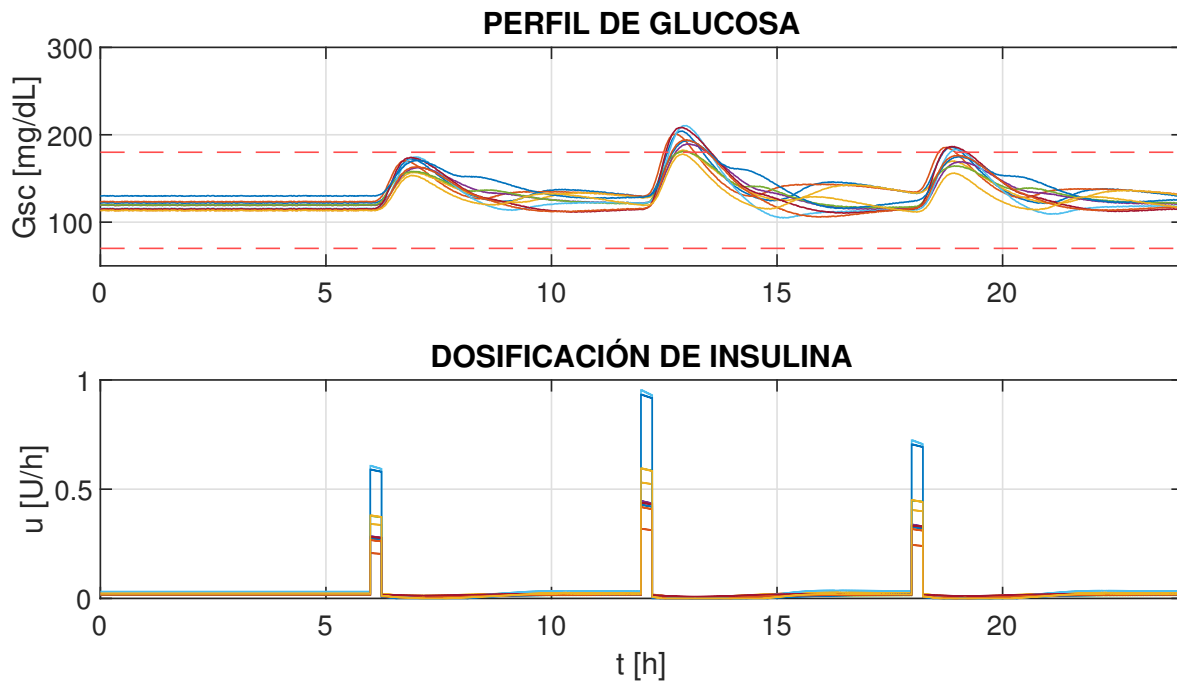


Figura 31. Gráfica generada por el algoritmo de optimización para la clase MPC_QDMC2 simulando una ingesta de alimentos de 50, 80 y 60 [g] de carbohidratos a las 6 a.m., 12 m. y 6 p.m. respectivamente para todo el banco de pacientes adultos el modelo nominal de cada uno.

Tabla 9

Medidas de hiperglucemia aplicando control basado en la clase MPC_QDMC2 al banco de adultos in silico usando modelo nominal, comparación de pico de glucosa en [mg/dL] y tiempo del paciente en hiperglucemia en minutos.

Hiperglucemia						
Adultos	6am		12pm		6pm	
Paciente	Pico	Tiempo	Pico	Tiempo	Pico	Tiempo
1	172.22	-	203.98	47	185.70	24
2	170.05	-	200.88	41	185.44	22
3	156.84	-	182.09	15	169.87	-
4	162.34	-	189.30	41	168.96	-
5	157.82	-	180.63	12	164.22	-
6	174.46	-	210.29	53	183.26	19
7	173.76	-	208.17	59	186.45	31
8	170.58	-	193.18	57	174.86	-
9	163.13	-	193.95	48	176.28	-
10	153.23	-	177.54	-	156.02	-

la presentaron usando el bolo óptimo, registrando un PMG y PPG de 210.29 y 194 [mg/dL] respectivamente. Sin embargo, los resultados del adulto 5 son considerablemente mejores, superando el límite por tan solo 0.63 [mg/dL] con una duración de apenas 12 minutos; en cuanto a la comida, se registran pocas condiciones de hiperglucemia, presentando un PMG y PPG de 186.45 y 175.1 [mg/dL] respectivamente. Los resultados se consideran satisfactorios y superiores a los presentados por el control mediante bolo óptimo.

Controlador basado en la clase MPC_{RP}. La figura 32 presenta los resultados de las validaciones in silico, usando el controlador basado en la clase MPC_{RP} con modelo nominal para el banco de pacientes adultos. El nivel mínimo de glucosa registrado fue de 110 [mg/dL]

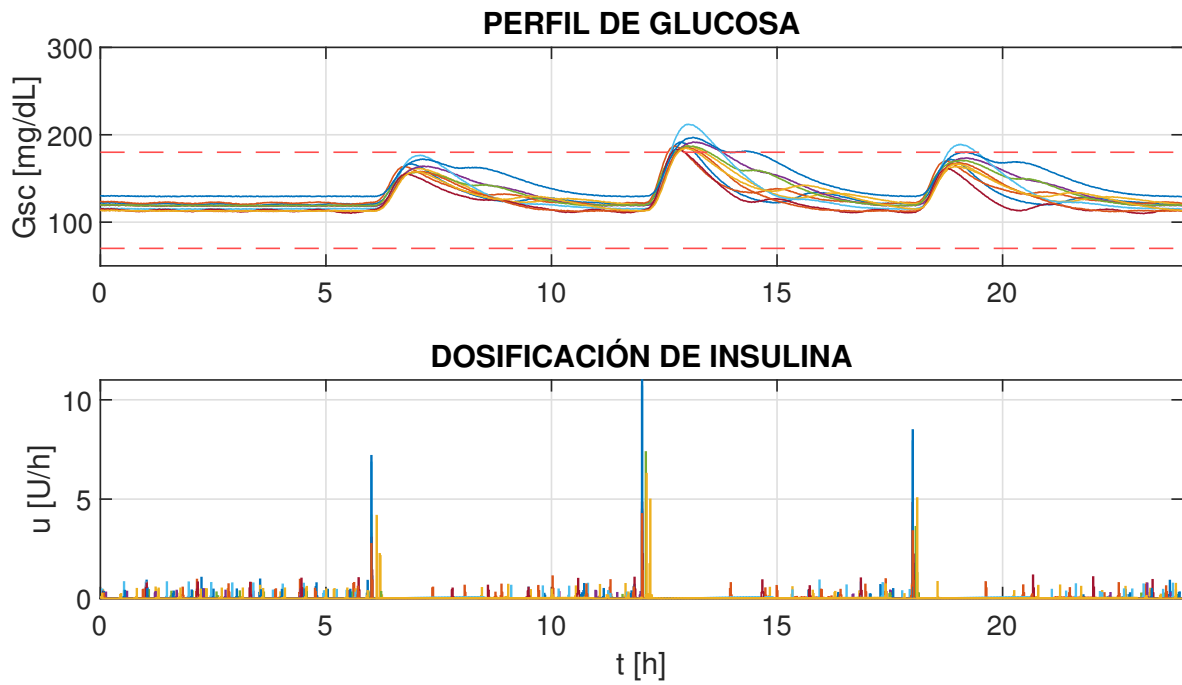


Figura 32. Gráfica generada por el algoritmo de optimización para la clase MPC_{RP} simulando una ingesta de alimentos de 50, 80 y 60 [g] de carbohidratos a las 6 a.m., 12 m. y 6 p.m. respectivamente para todo el banco de pacientes adultos el modelo nominal de cada uno.

dejando un margen de 21.48 y 5 [mg/dL] a favor en comparación con las dos estrategias de control anteriores, lo que asegura la mayor prevención de hipoglucemia de las 3 estrategias para modelo nominal con medición ideal de perturbación.

Tabla 10

Medidas de hiperglucemia aplicando control basado en la clase MPC_RP al banco de adultos in silico usando modelo nominal, comparación de pico de glucosa (PG) en [mg/dL] y tiempo del paciente en hiperglucemia en minutos.

Hiperglucemia						
Adultos	6am		12pm		6pm	
Paciente	PG	Tiempo	PG	Tiempo	PG	Tiempo
1	166.65	-	191.43	31	171.58	-
2	163.39	-	186.77	22	169.77	-
3	161.40	-	187.08	37	163.61	-
4	163.76	-	191.38	59	173.15	-
5	157.41	-	186.87	45	170.56	-
6	176.63	-	211.95	67	188.75	38
7	154.94	-	183.28	16	161.33	-
8	171.81	-	196.63	97	179.88	-
9	157.91	-	184.81	31	167.65	-
10	157.79	-	184.43	25	166.60	-

La tabla 10 presenta valores relevantes para el análisis de hiperglucemia de los pacientes. En la ingesta del desayuno no se detecta hiperglucemia, presentado un PMG de 176.63 [mg/dL] y un PPG de 163.14 [mg/dL]. Estos resultados se consideran similares a los del anterior controlador. Para la ingesta del almuerzo se detecta hiperglucemia en todos los pacientes, registrando un PMG y PPG de 211.95 y 190.46 [mg/dL] respectivamente. El adulto 5 también presenta un comportamiento satisfactorio, el paciente crítico es el adulto 8 que presenta un prologado estado

de hiperglucemia de 97 minutos. Sin embargo, su PG no supera los 200 [mg/dL], por lo tanto no se considera insatisfactorio. En la ingesta de la comida solo el adulto 6 presenta hiperglucemia, su PG fue de 188.75 [mg/dL] por 38 minutos de duración. El PPG registrado fue de 171.29 [mg/dL]. Los resultados se consideran satisfactorios y también superiores a los presentados por el control mediante bolo óptimo.

Comparaciones. Ahora se compara el desempeño de los controladores propuestos calculando los índices IAE (respecto a la referencia generada) e ICA. Para evitar valores elevados del índice IAE debidos simulaciones extensas, se divide su valor en la cantidad de muestras, para sacar un promedio que se denominó IAEP.

Tabla 11

Valores IAEP de los controladores usando modelo nominal

IAEP [mg/dL]	Adultos	
	Paciente	QDMC
1	8.781	6.997
2	9.435	7.466
3	9.018	4.705
4	3.641	3.894
5	4.671	3.372
6	6.448	2.536
7	4.269	7.096
8	6.114	5.443
9	3.376	3.111
10	9.114	5.174

La tabla 11 presenta las comparaciones del índice IAEP de los adultos para ambos controladores usando modelo nominal con medición de perturbación ideal. Promediando los resultados

de todos los paciente se obtuvo un registro de 6.286 [mg/dL] para la clase *MPC_QDMC2* y otro de 4.979 [mg/dL] para la clase *MPC_RP*. Esta última clase presenta un 20.8 % menos de error de seguimiento de referencia.

Tabla 12

Unidades de insulina total administrada al paciente durante el día

ICA [U]	Adultos		
Paciente	Máx	QDMC	RP
1	43.89	37.80	39.33
2	44.20	37.95	39.15
3	38.98	49.18	49.41
4	37.12	31.23	30.63
5	36.90	36.03	35.45
6	40.15	65.87	62.52
7	25.33	40.95	41.54
8	54.26	48.02	44.85
9	37.28	31.09	31.40
10	44.43	40.22	38.67

La tabla 12 presenta las comparaciones de la cantidad de unidades de insulina administradas al paciente in silico a lo largo del día por cada controlador. El controlador *MPC_QDMC2* dosificó en promedio 41.83 unidades y el controlador *MPC_RP* 41.29 unidades. Ambos presentaron el mismo desempeño en cuanto a la administración total de insulina. Sin embargo, si se tiene en cuenta la recomendación médica, a un paciente diabético se le debe suministrar como máximo una cantidad de unidades de insulina equivalente a su peso en Kg multiplicado por 0.55. Basándose en este criterio los adultos 3, 6 y 7 reciben más de la insulina diaria recomendada, este resultado puede deberse a la forma en que el modelo UVA PADOVA sintoniza los parámetros de los pacientes, ya

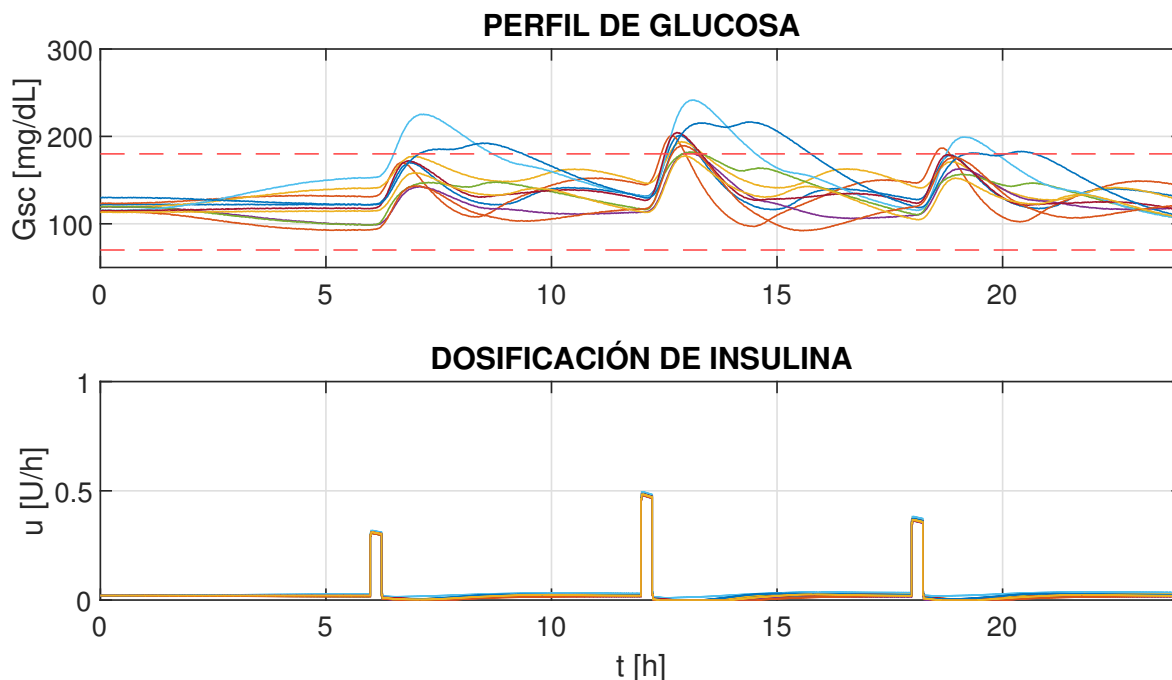


Figura 33. Gráfica generada por el algoritmo de optimización para la clase *MPC_QDMC2* simulando una ingesta de alimentos de 50, 80 y 60 [g] de carbohidratos a las 6 a.m., 12 m. y 6 p.m. respectivamente para todo el banco de pacientes adultos usando el modelo de predicción *adult_001* para todos.

que realmente solo caracterizaron un paciente, los demás pacientes resultan de variaciones en los parámetros de el primero.

6.2.2. Pruebas con modelo incierto. Para medir la robustez de los controladores, se realizan simulaciones usando como modelo de predicción los parámetros del adulto 1 (*adult_001*) y se prueban los controladores con medición ideal de perturbación en toda la cohorte. Esta prueba busca comparar los resultados de cada controlador propuesto con los resultados que obtuvo en las pruebas con modelo nominal.

Controlador basado en la clase *MPC_QDMC2*. La figura 33 presenta los resultados de las validaciones in silico, usando el controlador basado en la clase *MPC_QDMC2* con modelo

incierto para el banco de pacientes adultos. El nivel mínimo de glucosa registrado fue de 92.27 [mg/dL], que representa una reducción en el margen de hipoglucemia de 12.73 [mg/dL] respecto a las pruebas de este controlador con modelo nominal. Sin embargo, permanece 3.65 [mg/dL] por encima del nivel mínimo de glucosa registrado por el bolo óptimo.

Tabla 13

Medidas de hiperglucemia aplicando control basado en la clase MPC_QDMC2 al banco de adultos in silico con modelo incierto, comparación de pico de glucosa en [mg/dL] y tiempo del paciente en hiperglucemia en minutos.

Hiperglucemia						
Adultos	6am		12pm		6pm	
Paciente	Pico	Tiempo	Pico	Tiempo	Pico	Tiempo
1	169.85	-	200.99	41	178.41	-
2	170.08	-	200.67	33	186.56	19
3	177.14	-	193.69	45	171.02	-
4	142.27	-	181.52	16	162.85	-
5	147.47	-	181.91	25	156.48	-
6	225.27	131	241.45	119	199.24	67
7	171.72	-	204.14	48	178.85	-
8	192.06	138	216.30	185	182.57	58
9	143.30	-	189.32	34	175.76	-
10	158.27	-	177.52	-	152.10	-

La tabla 13 presenta los resultados de las pruebas de hiperglucemia. En la ingesta del desayuno 2 pacientes registran hiperglucemia durante un intervalo de 2 horas 15 minutos aproximadamente. El adulto 6 con un PG de 225.27 [mg/dL], que se considera insatisfactorio y el adulto 8 con un PG 192.06 [mg/dL] que se acepta. El PPG se mantuvo por debajo del límite de hiperglucemia en 169.74 [mg/dL]. En la ingesta del almuerzo se detecta hiperglucemia en los mismos

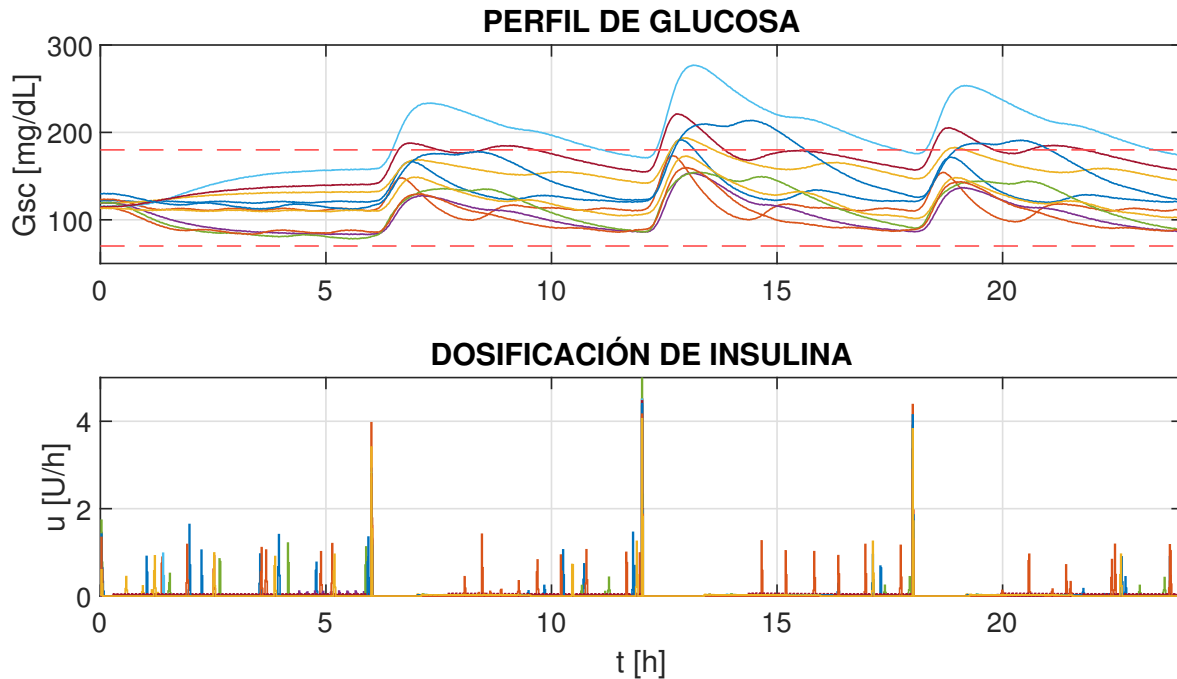


Figura 34. Gráfica generada por el algoritmo de optimización para la clase *MPC_QDMC2* simulando una ingesta de alimentos de 50, 80 y 60 [g] de carbohidratos a las 6 a.m., 12 m. y 6 p.m. respectivamente para todo el banco de pacientes adultos usando el modelo de predicción *adult_001* para todos.

pacientes que la presentaron usando el modelo nominal, donde el PPG subió a 198.75 [mg/dL]. Para esta prueba los resultados de los pacientes 6 y 8 se consideran insatisfactorios y el PMG superó el límite de desempeño establecido llegando a 241.45 [mg/dL], sin embargo, los resultados de los demás pacientes in silico se aprueban. En la ingesta de la comida nuevamente se registran pocas condiciones de hiperglucemia, presentado un PMG y un PPG de 199.24 y 174.38 [mg/dL] respectivamente. Los resultados se consideran satisfactorios y equivalentes a los presentados por el control mediante bolo óptimo, por lo tanto este controlador se considera robusto.

Controlador basado en la clase *MPC_RP*. La figura 34 presenta los resultados de las validaciones in silico usando el controlador basado en la clase *MPC_RP* para el banco de pacientes

adultos con modelo incierto. El controlador presenta error en estado estable respecto al valor basal de los pacientes. Sin embargo, ningún valor basal quedó por fuera de los límites de hiperglucemia e hipoglucemia. El nivel mínimo de glucosa registrado fue de 78.43 [mg/dL] estando por debajo de valor presentado por el bolo óptimo. Sin embargo, en 2 de los 3 controladores de los artículos citados el nivel mínimo llega a los 70 [mg/dL]. Por lo tanto, se considera que el controlador es bueno evitando estados de hipoglucemia.

Tabla 14

Medidas de hiperglucemia aplicando control basado en la clase MPC_{RP} al banco de adultos in silico con modelo incierto, comparación de pico de glucosa (PG) en mg/dL y tiempo del paciente en hiperglucemia en minutos.

Hiperglucemia						
Adultos	6am		12pm		6pm	
Paciente	PG	Tiempo	PG	Tiempo	PG	Tiempo
1	166.65	-	191.43	31	171.15	-
2	147.84	-	173.41	-	153.96	-
3	168.55	-	193.64	57	182.65	24
4	127.41	-	153.35	-	136.60	-
5	135.40	-	154.25	-	144.06	-
6	233.44	271	276.78	288	253.47	348
7	187.67	124	221.05	87	205.17	156
8	177.69	-	213.58	175	190.86	125
9	129.30	-	159.43	-	143.36	-
10	148.71	-	172.66	-	148.20	-

La tabla 14 presenta valores relevantes para el análisis de hiperglucemia de los pacientes. Se encontró insatisfactorio el control de glucosa del adulto 6 en toda la prueba, a su vez el adulto 8 presento resultados insatisfactorios para la ingesta del almuerzo y el adulto 7 en la ingesta

de la comida, esta muestra representa el 16.66% de los resultados de la prueba. Sin embargo, el PPG registrado en las 3 ingestas de carbohidratos fueron 162.27 [mg/dL] para el desayuno, 190.96 [mg/dL] para el almuerzo y 172.99 [mg/dL] para la comida. Estos resultados se encuentran dentro de los rangos de las demás pruebas presentadas, es decir, se puede considerar satisfactorios. Finalmente, este controlador demostró ser menos robusto en comparación con el *MPC_QDMC2*. Sin embargo, no lleva a ningún paciente a condiciones de hipoglucemia y obtuvo resultados satisfactorios en las medidas de hiperglucemia en la mayoría de pacientes. Por lo tanto, también se considera robusto.

Comparaciones. Se compara el desempeño como controlador de seguimiento de referencia a las clases *MPC_QDMC2* y *MPC_RP* con modelo incierto usando medición de perturbación ideal.

Tabla 15

Valores IAEP de los controladores usando modelo nominal

IAEP [mg/dL]	Adultos	
Paciente	QDMC	RP
1	10.354	7.843
2	20.418	21.877
3	15.942	17.523
4	12.631	29.637
5	10.846	22.686
6	24.833	61.963
7	8.804	34.980
8	16.685	16.118
9	12.641	26.168
10	8.432	8.899

La tabla 15 presenta las comparaciones del índice IA_{Ep} de los adultos para ambos controladores. El controlador *MPC_QDMC2* obtuvo un promedio de 14.158 [mg/dL], que representa un incremento del 118.75% respecto a los resultados usando modelo nominal. Mientras que el controlador *MPC_RP* obtuvo 24.769 [mg/dL], aumentando un 397.46% respecto a la prueba con modelo nominal. En este escenario la clase *MPC_RP* muestra un 75% más de error de seguimiento de referencia, lo que demuestra que es menos robusta en cuanto a seguimiento en comparación con la clase *MPC_QDMC2*.

Tabla 16

Unidades de insulina total administrada al paciente durante el día

ICA [U] Paciente	Adultos		
	Máx	QDMC	RP
1	43.89	38.48	38.88
2	44.20	39.79	44.89
3	38.98	47.14	44.78
4	37.12	32.77	37.00
5	36.90	37.06	40.78
6	40.15	55.82	43.43
7	25.33	41.30	38.46
8	54.26	44.46	43.80
9	37.28	32.75	37.29
10	44.43	39.73	41.31

La tabla 16 presenta las comparaciones de la cantidad de unidades de insulina administradas al paciente a lo largo del día. En este escenario, el controlador *MPC_QDMC2* inyectó en promedio 40.93 unidades, que representa una disminución de 2.15% respecto a la prueba con modelo nominal y el controlador *MPC_RP* inyectó 41.06 unidades, un 0.55% unidades menos que en

la prueba con modelo nominal. Debido a que los resultados de dosificación de insulina total, son prácticamente los mismos en ambas pruebas a pesar de las variaciones percibidas en las respuestas, se remarca la necesidad del control en lazo cerrado sobre el lazo abierto para tratar este problema, ya que se evidencia que el instante en que se inyecta la insulina es lo que diferencia los resultados.

6.3. Validación del esquema de control completo

6.3.1. Sintonización del estimador. Se entrenó un estimador para el controlador basado en la clase *MPC_QDMC*. Las matrices de covarianza del filtro de Kalman lineal se ajustaron para dar prioridad a la medición sobre la predicción. Ya que al no tener en cuenta los estados del tracto digestivo, la predicción no es fiable. Por lo tanto, se definió $Q_k = 1 \cdot 10^7$. Por otra parte, para la covarianza de la medición se calculó la covarianza del sensor de glucosa, donde se obtuvo $R_k = 0.01$.

La figura 35 presenta los resultados de estimación de perturbación, para el controlador basado en la clase *MPC_QDMC*. Donde en la primera gráfica se comparan la perturbación real y la perturbación real retardada 15 minutos con la estimación generada. La segunda gráfica compara las referencias generadas por la perturbación real y la estimada a lo largo de un día simulando al paciente *adult_001*. Se estimaron 56.66, 80.26 y 45.20 [g] de CHO respectivamente para cada ingesta, presentando errores en la cantidad de carbohidratos de 5.566%, 0.325% y 9.6% respectivamente. El error total en la cantidad de CHO fue de 2.87%. Para la referencia generada, el porcentaje de error en la forma de la señal estimada fue de 2.17% en promedio, por lo que los resultados se consideran satisfactorios.

La figura 36 presenta los resultados del controlador basado en la clase *MPC_QDMC2* cuan-

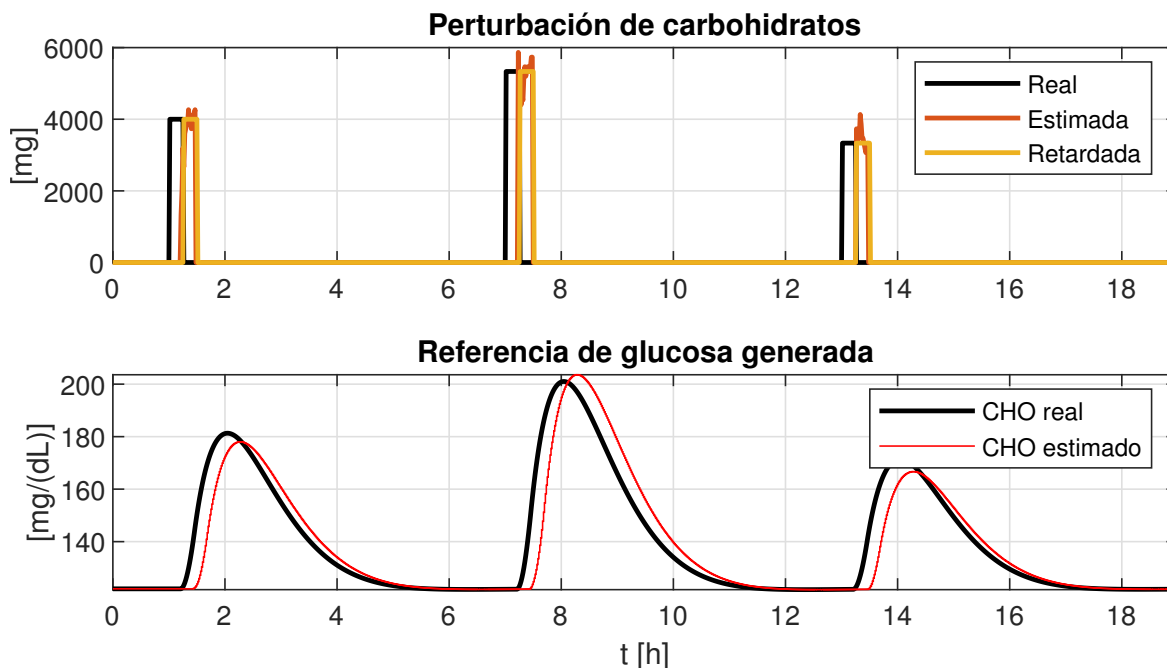


Figura 35. Resultados de estimación de perturbación y generación de referencia para una ingesta de 60, 80 y 50 [g] de carbohidratos a las 6 a.m., 12 m. y 6 p.m. respectivamente para el paciente *adult_001* usando un controlador basado en la clase *MPC_QDMC2*.

do recibe información de la perturbación a través del estimador. No se evidencia una variación representativa en el error IAEP a pesar de los efectos del retardo de estimación. Esto se debe a que el aumento de error al inicio del transitorio, se ve compensado al final del mismo. Por otra parte, es innegable que el desempeño del controlador se ve afectado, ya que el paciente presenta mayores picos de glucosa. Sin embargo, el resultado no deja de ser satisfactorio.

6.3.2. Pruebas con modelo nominal. Se realizaron mismas pruebas de validación de estabilidad y comportamiento nominal de cada controlador agregando el estimador de perturbaciones para la cohorte de 10 pacientes adultos.

Controlador basado en la clase MPC_QDMC2. La figura 37 presenta los resultados de las validaciones in silico, usando el controlador *MPC_QDMC2* con modelo nominal y estimador

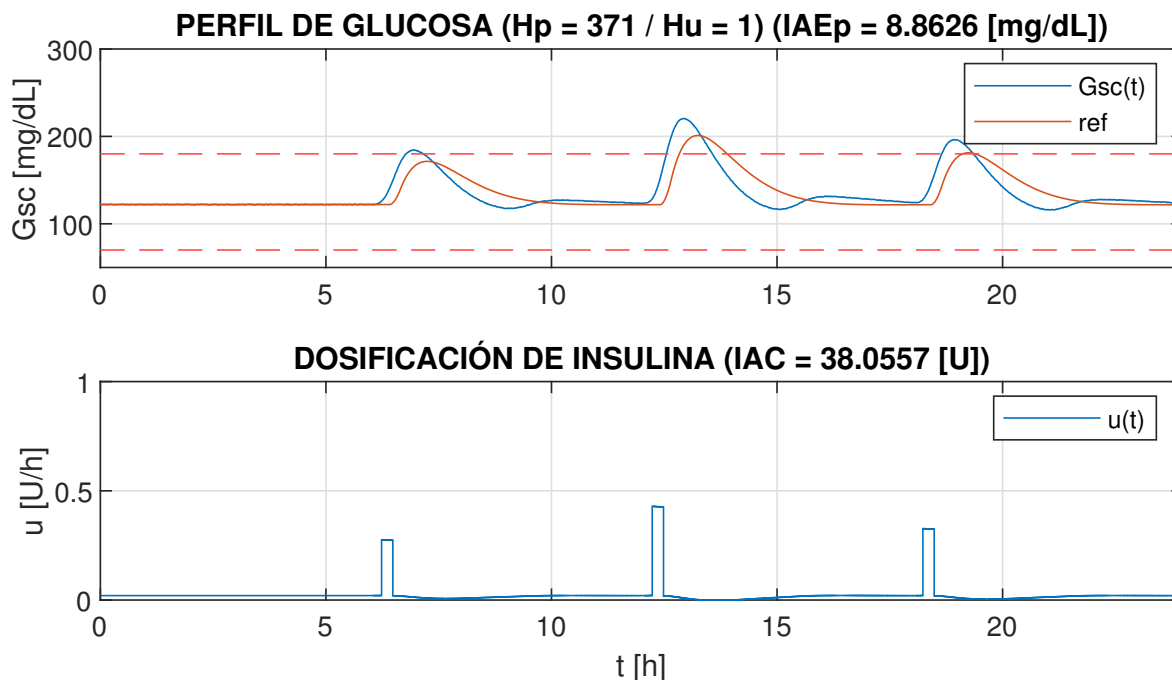


Figura 36. Resultados de estimación de perturbación y generación de referencia para una ingesta de 60, 80 y 50 [g] de carbohidratos a las 6 a.m., 12 m. y 6 p.m. respectivamente para el paciente *adult_001* usando un controlador basado en la clase *MPC_QDMC2* y el estimador de perturbaciones.

de perturbaciones para el banco de pacientes adultos. El nivel mínimo de glucosa registrado fue de 90.42 [mg/dL], dejando un margen de 1.8 [mg/dL] a favor en comparación con el bolo óptimo, por lo tanto la prevención de hipoglucemia es similar.

La tabla 17 presenta valores relevantes para el análisis de hiperglucemia de los pacientes. Se evidencia el efecto del retardo de estimación en las medidas de hiperglucemia. En la ingesta del desayuno, 4 pacientes presentan hiperglucemia, sin embargo, ninguno de los resultados se considera insatisfactorio. El PMG registrado fue 197.49 [mg/dL] y el PPG 178.74 [mg/dL] que aumento un 7.41 % respecto a la misma prueba con perturbación ideal. Para la ingesta del almuerzo todos los pacientes presentan hiperglucemia. Donde los resultados del adulto 7 se consideran insatisfacto-

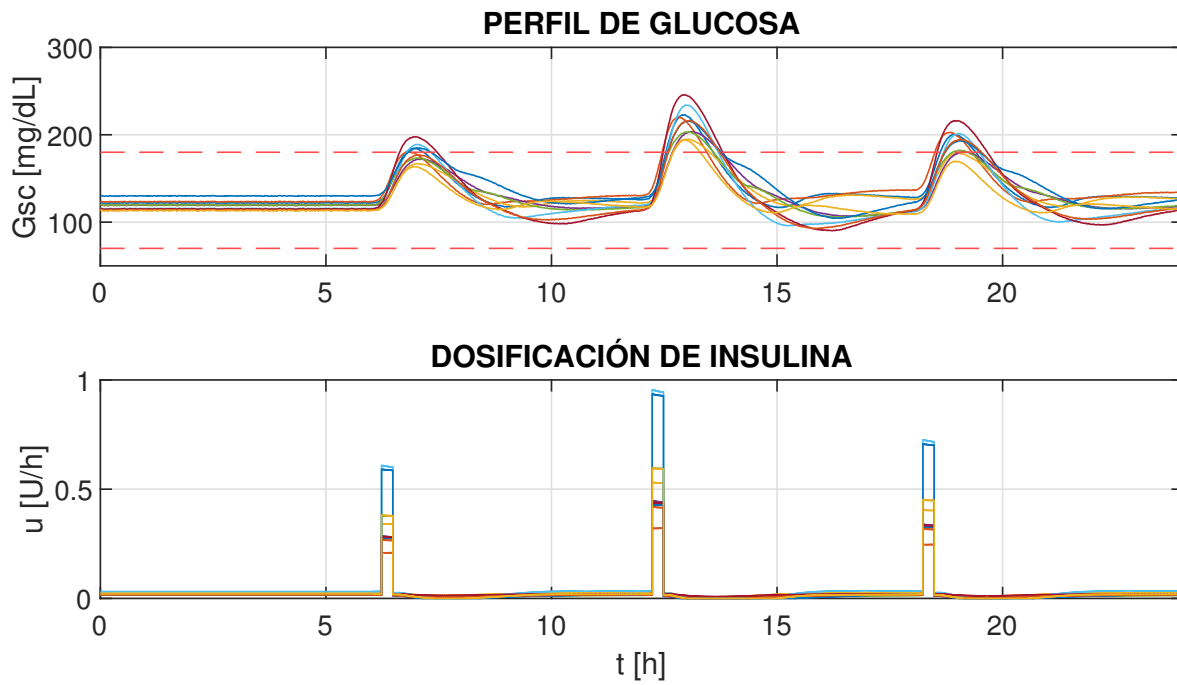


Figura 37. Gráfica generada por el algoritmo de optimización para la clase MPC_QDMC2 simulando una ingesta de alimentos de 50, 80 y 60 [g] de carbohidratos a las 6 a.m., 12 m. y 6 p.m. respectivamente para todo el banco de pacientes adultos el modelo nominal de cada uno.

Tabla 17

Medidas de hiperglucemia aplicando controlador basado en la clase MPC_QDMC2, comparación de pico de glucosa (PG) en [mg/dL] y tiempo del paciente en hiperglucemia en minutos.

Hiperglucemia						
Adultos	6am		12pm		6pm	
Paciente	PG	Tiempo	PG	Tiempo	PG	Tiempo
1	184.28	23	222.72	62	200.12	47
2	180.19	5	219.80	59	202.58	50
3	166.74	-	194.86	48	178.69	-
4	172.10	-	203.48	64	180.18	6
5	173.15	-	203.27	60	182.06	20
6	188.68	32	233.98	68	201.24	47
7	197.49	51	245.62	79	216.10	67
8	184.78	30	215.57	71	192.94	47
9	176.68	-	216.16	68	194.30	50
10	163.33	-	194.16	36	169.54	-

rios, ya que fue el PMG presentando un valor de 245.62. El PPG registrado fue de 214.96 [mg/dL] que aumentó 10.8% respecto a la prueba con medición de perturbación ideal. En la comida el único paciente que no registró hiperglucemia fue el adulto 10. Sin embargo, todos los resultados fueron satisfactorios, donde el PMG fue de 216.10 y el PPG de 191.77 [mg/dL], un 9.52% mayor en comparación por el presentado en la prueba con medición ideal de perturbación. Por lo tanto, los resultados generales se consideran satisfactorios. Se concluye que incluir el estimador aumenta el pico de glucosa aproximadamente un 10% para el controlador *MPC_QDMC2*.

Controlador basado en la clase MPC_{RP}. La figura 38 presenta los resultados de las validaciones in silico, usando el controlador basado en la clase *MPC_{RP}* con modelo nominal añadiendo el estimador de perturbaciones para el banco de pacientes adultos. El nivel mínimo de

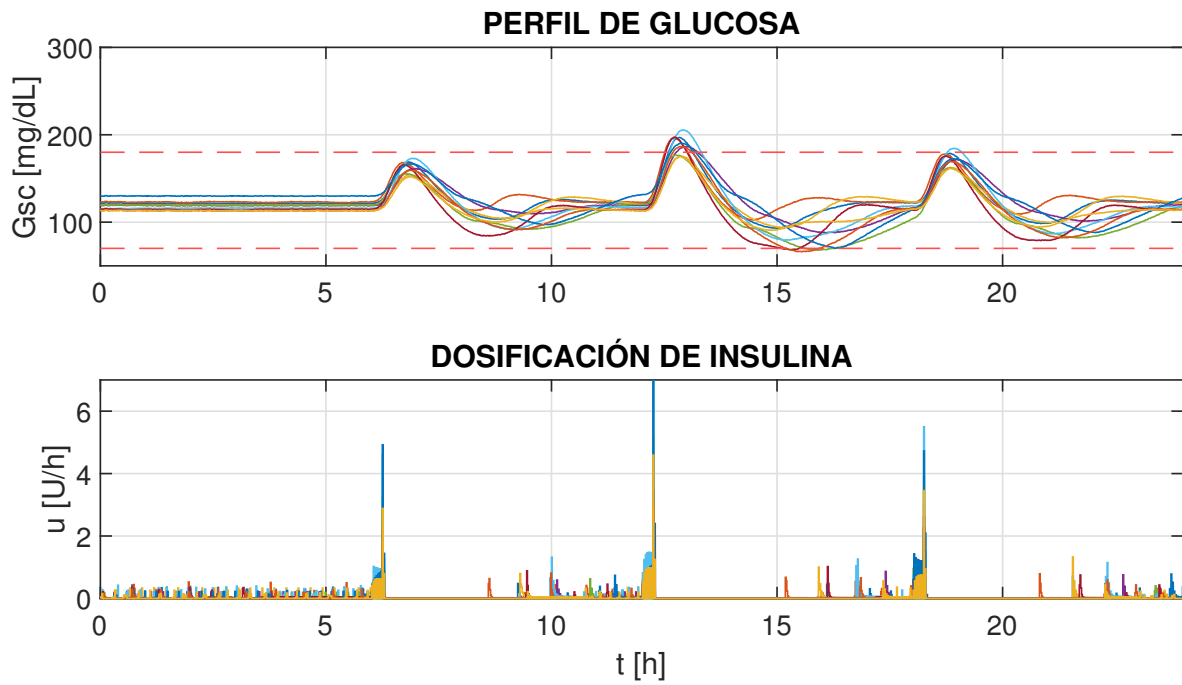


Figura 38. Gráfica generada por el algoritmo de optimización para la clase MPC_{RP} simulando una ingesta de alimentos de 50, 80 y 60 [g] de carbohidratos a las 6 a.m., 12 m. y 6 p.m. respectivamente para todo el banco de pacientes adultos el modelo nominal de cada uno.

glucosa registrado es de 66.30 [mg/dL], presentado por el adulto 9. Este resultado se encuentra por debajo del límite de hipoglucemia por 3.7 [mg/dL], lo que representa una disminución considerable con respecto a la prueba con estimador ideal, específicamente de 43.7 [mg/dL]. Por lo tanto, es evidente que el retardo de estimación perjudica en gran medida el desempeño de este controlador para la prevención de hipoglucemia.

Tabla 18

Medidas de hiperglucemia aplicando controlador basado en la clase MPC_QDMC2, comparación de pico de glucosa (PG) en [mg/dL] y tiempo del paciente en hiperglucemia en minutos.

Hiperglucemia						
Adultos	6am		12pm		6pm	
Paciente	PG	Tiempo	PG	Tiempo	PG	Tiempo
1	168.82	-	196.76	33	178.48	-
2	168.20	-	196.56	31	178.11	-
3	154.17	-	175.55	-	162.51	-
4	161.15	-	186.37	29	171.56	-
5	155.16	-	176.61	-	161.92	-
6	173.01	-	205.60	42	184.46	19
7	165.61	-	196.97	28	175.65	-
8	166.56	-	190.25	33	172.52	-
9	159.22	-	186.54	25	169.22	-
10	151.89	-	174.94	-	191.35	-

La tabla 18 presenta valores relevantes en el análisis de hiperglucemia de los pacientes. En la ingesta del desayuno, se registró un PMG de 173.01 [mg/dL] y un PPG de 162.36 [mg/dL]. En la ingesta del almuerzo, el PMG y PPG, 205.60 y 188.62 [mg/dL] respectivamente. Finalmente, en la ingesta de la comida comida, se registró un PG de 184.46 [mg/dL] y un PPG fue de 171.58 [mg/dL]. Estos resultados son ligeramente mejores comparados con los de la prueba usando medición de

perturbación ideal para este mismo controlador, siendo los mejores en términos de prevención de hiperglucemia. Sin embargo, no se puede ignorar que esto se debe a que el controlador lleva al paciente a niveles más cercanos a la hipoglucemia, llegando incluso a violar el límite de hipoglucemia. Los resultados de este controlador son similares al control que propone el Event-triggered model predictive control del autor Chakrabarty en su artículo Chakrabarty et al. (2017).

Comparaciones. Las comparaciones buscan verificar que tanto afecta el seguimiento de referencia y la dosificación de insulina total la inclusión del estimador de perturbaciones.

Tabla 19

Valores IAEP de los controladores usando modelo nominal

IAEP [mg/dL]	Adultos	
Paciente	QDMC	RP
1	10.406	16.011
2	12.278	14.267
3	7.658	15.446
4	6.414	14.389
5	8.521	25.885
6	13.625	17.645
7	15.847	21.610
8	9.550	23.817
9	11.130	19.077
10	9.002	15.218

La tabla 19 presenta las comparaciones del índice IAEP de los adultos para ambos controladores. En promedio el IAEP del controlador *MPC_QDMC2* fue 10.443 [mg/dL], que representa un aumento del 66.13 % respecto a los resultados usando medición de perturbación ideal. Para el controlador *MPC_RP* el promedio fue de 18.336 [mg/dL], que representa un aumento del 268.26 %.

La clase MPC_{RP} muestra un error de seguimiento un 75% mayor en comparación con la clase MPC_{QDMC2} cuando se incluye al estimador de perturbaciones.

Tabla 20

Unidades de insulina total administrada al paciente durante el día

ICA [U]	Adultos		
	Máx	QDMC	RP
1	43.89	37.79	41.29
2	44.20	37.92	40.33
3	38.98	49.16	52.66
4	37.12	31.13	33.06
5	36.90	35.81	41.26
6	40.15	66.46	72.91
7	25.33	41.05	45.22
8	54.26	47.87	57.99
9	37.28	31.13	34.33
10	44.43	40.16	43.48

La tabla 20 presenta las comparaciones de la cantidad de unidades de insulina administradas al paciente a lo largo del día. Donde el controlador MPC_{QDMC2} inyectó en promedio 41.84 unidades, siendo prácticamente el mismo resultado que presento al usar medición de perturbación ideal. Por otra parte, el controlador MPC_{RP} inyectó 46.25 unidades, lo que representa un aumento en la dosificación del 12.12%. Esto causó que además de los pacientes 3, 6 y 7 que siempre han superado el límite de insulina máxima recomendado, los pacientes 5 y 8 también lo superen. El controlador MPC_{RP} tiende a aplicar una acción de control más agresiva cuando se usa junto al estimador de perturbaciones, aumentando la tendencia a hipoglucemia y el consumo de insulina. Sin embargo, esto produce buen desempeño a la hora de evitar la hiperglucemia.

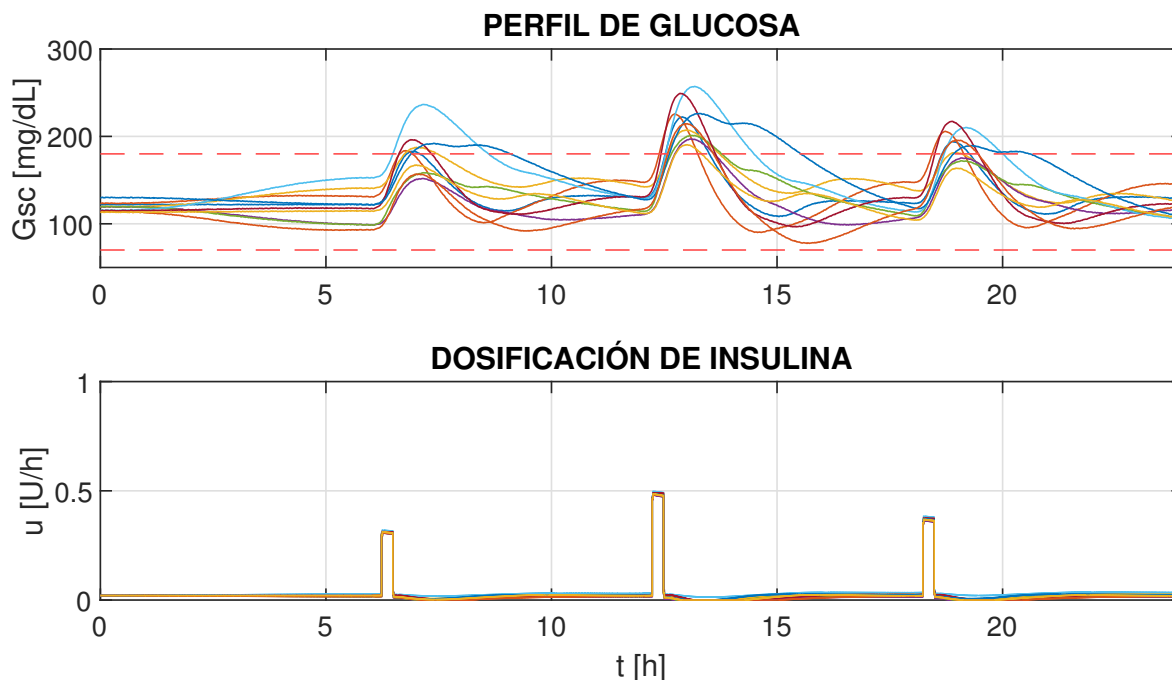


Figura 39. Gráfica generada por el algoritmo de optimización para la clase *MPC_QDMC2* simulando una ingesta de alimentos de 50, 80 y 60 [g] de carbohidratos a las 6 a.m., 12 m. y 6 p.m. respectivamente para todo el banco de pacientes adultos usando el modelo de predicción *adult_001* para todos.

6.3.3. Pruebas con modelo incierto. Para validar si el conjunto controlador-estimador es robusto, se realizan simulaciones usando como modelo de predicción los parámetros del adulto 1 y se prueban los controladores usando el estimador de perturbaciones en toda la cohorte. Se busca comparar los resultados de cada conjunto controlador-estimador con los resultados que obtuvo en las pruebas con modelo nominal.

Controlador basado en la clase *MPC_QDMC2*. La figura 39 presenta los resultados de las validaciones in silico, usando el controlador basado en la clase *MPC_QDMC2* con modelo incierto junto al estimador de perturbaciones para el banco de pacientes adultos. El nivel mínimo de glucosa

registrado fue de 77.82 [mg/dL], que representa una reducción en el margen de hipoglucemia de 14.45 [mg/dL] respecto a las pruebas usando medición de perturbación ideal. Por lo tanto, se verifica que este controlador en ninguno de los escenarios simulador lleva al paciente a condición de hipoglucemia, por lo que se considera robusto para evitar condiciones de hipoglucemia.

Tabla 21

Medidas de hiperglucemia aplicando controlador basado en la clase MPC_QDMC2, comparación de pico de glucosa (PG) en [mg/dL] y tiempo del paciente en hiperglucemia en minutos.

Hiperglucemia						
Adultos	6am		12pm		6pm	
Paciente	PG	Tiempo	PG	Tiempo	PG	Tiempo
1	182.78	17	222.23	58	193.86	36
2	183.54	17	225.40	51	205.61	42
3	187.20	44	207.25	69	191.38	15
4	151.43	-	196.90	52	175.19	-
5	158.06	-	201.05	67	171.90	-
6	236.13	127	256.95	118	210.06	78
7	196.08	43	249.07	73	216.89	58
8	191.67	132	226.01	175	189.18	101
9	156.68	-	214.49	61	195.63	47
10	166.94	-	190.43	34	163.58	-

La tabla 21 presenta valores relevantes en el análisis de hiperglucemia de los pacientes. En la ingesta del desayuno los adultos 6 y 8 presentan condición de hiperglucemia durante más de 2 horas, donde el adulto 6 registró un PG de 236.13 [mg/dL], por lo que sus resultados se consideran insatisfactorios. Por otra parte, el adulto 8 presentó un PG de 191.08 [mg/dL], por lo que se acepta. El PPG fue de 181.05 [mg/dL], justo sobre el límite de hiperglucemia. En la ingesta del almuerzo se detectó hiperglucemia todos los pacientes, donde se consideraron insatisfactorios los adultos 6,

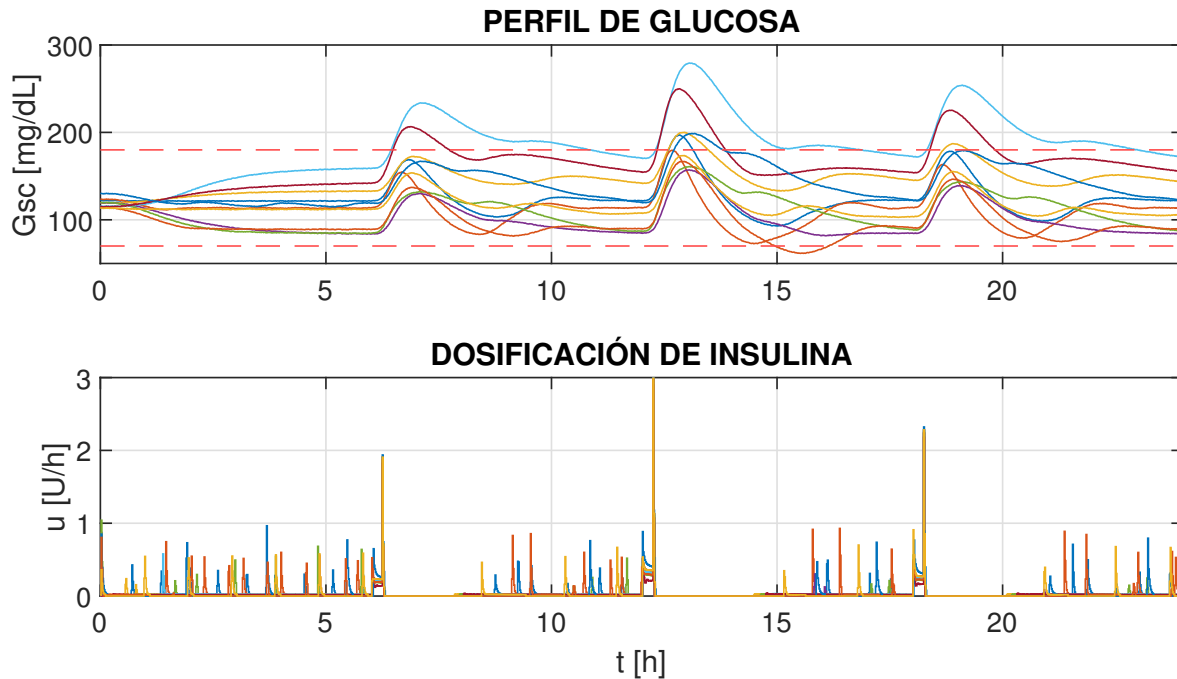


Figura 40. Gráfica generada por el algoritmo de optimización para la clase *MPC_QDMC2* simulando una ingesta de alimentos de 50, 80 y 60 [g] de carbohidratos a las 6 a.m., 12 m. y 6 p.m. respectivamente para todo el banco de pacientes adultos usando el modelo de predicción *adult_001* para todos.

7 y 8. El PPG registrado fue de 218.98 [mg/dL]. Finalmente, en la ingesta de la comida el comportamiento de la glucosa de todos los pacientes se considera satisfactorio. Al terminar el análisis se encontró que el controlador *MPC_QDMC2* con modelo incierto y estimador de perturbaciones obtuvo un 13.33% de resultados insatisfactorios. Sin embargo, este es el escenario más exigente y los resultados insatisfactorios no representan un mal desempeño en general, sino un desempeño inferior al esperado en comparación con el bolo óptimo y los controladores de los artículos citados. Por lo tanto, se considera que este sistema de control es robusto.

Controlador basado en la clase *MPC_RP*: La figura 40 presenta los resultados de las validaciones in silico, usando el controlador basado en la clase *MPC_RP* para el banco de pacientes

adultos con modelo incierto usando el estimador de perturbaciones. El nivel mínimo de glucosa registrado fue de 61.70 [mg/dL], donde se evidenció el peor desempeño para prevenir la hipoglucemia en comparación a las demás validaciones. Afortunadamente solo el adulto 9 esta por debajo de los 70 [mg/dL]. Sin embargo, en 2 de los 3 controladores de los artículos citados el nivel mínimo llega a estar ligeramente por debajo de los 70 [mg/dL] para algunos pacientes, por lo que este resultado no descarta al controlador *MPC_RP* con estimador de perturbaciones. Adicionalmente, el segundo nivel de glucosa más bajo registrado fue de 72.88 [mg/dL]. Por lo tanto, este esquema de control presentó un índice de 90% de éxito en la prevención de hipoglucemia. Sin embargo, se recomienda usar un algoritmo de estimación de perturbaciones que no presente un retardo tan prolongado.

Tabla 22

Medidas de hiperglucemia aplicando controlador basado en la clase MPC_QDMC2, comparación de pico de glucosa (PG) en [mg/dL] y tiempo del paciente en hiperglucemia en minutos.

Hiperglucemia						
Adultos	6am		12pm		6pm	
Paciente	PG	Tiempo	PG	Tiempo	PG	Tiempo
1	168.62	-	196.76	33	178.48	-
2	154.37	-	179.60	-	163.00	-
3	172.37	-	200.00	53	186.96	33
4	129.84	-	156.99	-	138.94	-
5	131.96	-	160.80	-	143.31	-
6	233.62	265	279.36	273	253.69	279
7	206.28	74	249.65	91	225.93	86
8	166.79	-	198.74	66	179.11	-
9	136.87	-	166.75	-	146.35	-
10	153.53	-	173.54	-	155.09	-

La tabla 22 presenta valores relevantes en el análisis de la hiperglucemia de los pacientes. Se registraron resultados insatisfactorios en el control de glucosa del adulto 6 durante toda la prueba y en el control de glucosa del adulto 7 durante la ingesta del almuerzo, lo que representa el 13.33 % de los resultados de la prueba. El PPG registrado en las 3 ingestas fue de 165.42 [mg/dL] para el desayuno, 196.22 [mg/dL] para el almuerzo y 177.03 [mg/dL] para la comida. Estos resultados están dentro de los rangos satisfactorios de las demás pruebas presentadas. Finalmente, este sistema de control presenta resultados satisfactorios en las pruebas de hipoglucemia e hiperglucemia para 7 de 10 pacientes adultos in silico, por lo que se puede considerar robusto.

Comparaciones. A continuación se compara el desempeño como controlador de seguimiento de referencia de las clases *MPC_QDMC2* y *MPC_RP* con modelo incierto usando estimador de perturbaciones.

Tabla 23

Valores IAEP de los controladores usando modelo nominal

IAEP [mg/dL]	Adultos	
	Paciente	QDMC
1	12.152	16.011
2	24.067	28.505
3	15.046	15.750
4	13.817	34.336
5	7.566	25.821
6	24.589	53.957
7	16.868	33.357
8	13.989	7.421
9	18.617	31.781
10	7.545	12.763

La tabla 23 presenta las comparaciones del índice IAEP de los adultos para ambos controladores. El controlador *MPC_QDMC2* obtuvo un promedio de 15.425 [mg/dL], que representa un incremento del 8.95 % respecto a los resultados con medición de perturbación ideal. Mientras que el controlador *MPC_RP* obtuvo 25.970 [mg/dL], aumentando en 4,85 %. La clase *MPC_RP* continúa presentando mayor índice de error, específicamente un 68,36 % más de error de seguimiento de referencia. Por lo tanto en este escenario, incluir el estimador de perturbaciones no produce un efecto significativo en el seguimiento de referencia.

Tabla 24

Unidades de insulina total administrada al paciente durante el día

ICA [U]	Adultos		
	Máx	QDMC	RP
1	43.89	38.51	41.29
2	44.20	40.20	47.89
3	38.98	46.99	45.95
4	37.12	32.56	38.23
5	36.90	36.61	41.48
6	40.15	55.84	45.01
7	25.33	41.14	38.31
8	54.26	44.07	47.36
9	37.28	32.81	38.52
10	44.43	39.69	43.25

La tabla 24 presenta las comparaciones de la cantidad de unidades de insulina inyectadas al paciente a lo largo del día. En este escenario el controlador *MPC_QDMC2* inyectó en promedio 40.84 unidades, que representa una disminución de 0.22 % respecto a la prueba con medición de perturbación ideal y el controlador *MPC_RP* inyectó 42.729 unidades, una dosificación 4.06 %

mayor que en la prueba con medición de perturbación ideal. Se evidencio un aumento menor en la dosificación de insulina respecto a las pruebas con medición de perturbación ideal.

7. Trabajo futuro

Se propone trabajar en el diseño del algoritmo de control de la clase *MPC_RPI*, ya que añadir un integrador de forma correcta al algoritmo de control predictivo de la clase *MPC_RP* podría producir una mejora significativa de robustez, reduciendo la mayor desventaja de esta clase que por otra parte, presento los mejores resultados con modelo nominal en las validaciones.

Se podría aprovechar mejor el calculo propuesto para la proyección de la perturbación en las predicciones si se aplica un esquema no lineal al algoritmo de control. Por ejemplo se puede plantear un controlador NMPC, ya que la perturbación produce una rata de aparición de glucosa significativamente diferente en el UVA PADOVA linealizado en contraste con el no lineal.

A pesar de los buenos resultados que presenta el estimador de perturbaciones, se evidencio que el retardo de 15 minutos asociado al mismo puede afectar significativamente el desempeño de los controladores. Por lo tanto, se propone buscar la manera de reducirlo.

Todos los algoritmos propuestos en este proyectos fueron pensados para realizar pruebas de validación "*hardware in the loop*". Por lo tanto, queda por programar el sistema de control propuesto en algún dispositivo de procesamiento y realizar estas pruebas de validación.

Se puede buscar la forma de recolectar datos de curvas de glucosa reales para entrenar y validar el estimador teniendo en cuenta que no depende de un modelo.

8. Conclusiones

En esta propuesta de control se pudo comprobar que usando el modelo UVA PADOVA, se evidenció una mejora del 11.7% en el error IAE al incluir las restricciones a la salida, en contraprestación de una carga computacional elevada del algoritmo de control, lo que demandaría hardware de mejores prestaciones y podría hacer inviable implementar el algoritmo. Sin embargo, al sintonizar los algoritmos de control predictivo, se encontró una mejora de más del 4000% en la velocidad de procesamiento al quitar las restricciones a la salida, sin perjudicar notablemente la respuesta del sistema. Por lo tanto, se recomienda no tenerlas en cuenta en el control del modelo UVA PADOVA.

Si se mantienen constantes los valores de perturbación y referencia medidos por el controlador a lo largo del horizonte de predicción, resulta contraproducente al aplicar una acción feed-forward en el algoritmo de control predictivo para pacientes diabéticos in silico. Por una parte, se predice un consumo de carbohidratos mayor al consumo real, lo que produce una alta dosificación de insulina y consecuentemente que al establecer horizontes de control mayores a 1 paso saturan la acción de control. Por otra parte, que el controlador incluya información dinámica de la referencia en sus predicciones, como por ejemplo que la tendencia de la referencia sea creciente, evita que dosifique insulina de forma innecesaria mejorando el seguimiento. Los controladores de la clase *MPC_{RP}* a los que se aplicaron estas modificaciones presentaron buenos resultados en condiciones nominales, pero demostraron ser menos robustos.

Los controladores de las clases *MPC_QDMC2* y *MPC_RP* demostraron un desempeño superior al del control mediante bolo óptimo, comparándolos en el mismo escenario. Es decir, con modelo nominal y medición ideal de perturbaciones, presentaron un mejor margen de prevención de hipoglucemia en comparación con el bolo óptimo (16.38 y 21.48 [mg/dL] respectivamente). Todos los resultados de hiperglucemia fueron satisfactorios para los controladores feed-forward propuestos, presentando menos pacientes en condición de hiperglucemia, picos de glucosa ligeramente menores y menos duración de tiempo en hiperglucemia. Adicionalmente, haciendo una comparación entre controladores, los mejores resultados en estas condiciones los presentó el controlador *MPC_RP*, registrando picos de glucosa ligeramente más bajos, mayor prevención de hipoglucemia y error IAE 20.8% menor.

Los controladores de las clases *MPC_QDMC2* y *MPC_RP* demostraron ser robustos, ya que usando como modelo de predicción los parámetros del paciente adulto 1, mantuvieron a toda la cohorte dentro del margen hipoglucemia. En cuanto a la hiperglucemia el controlador *MPC_QDMC2* presentó 27 de 30 resultados satisfactorios y el controlador *MPC_RP* 26 de 30 en comparación al bolo óptimo. Por lo tanto, ambos controladores demostraron ser robustos. Sin embargo, el controlador *MPC_QDMC2* superó al *MPC_RP* notablemente en las pruebas de modelo incierto.

El estimador propuesto presentó resultados satisfactorios, consiguiendo errores de estimación por debajo del 10% y demostrando ser robusto ante la presencia de ruido. Sin embargo, el retardo de 15 minutos afecta el desempeño de los controladores propuestos. La

clase *MPC_QDMC2* presentó picos de glucosa 10% mayores y un error de seguimiento IAE 66.13% mayor. A pesar de esto, los resultados generales del controlador fueron satisfactorios. Por otra parte, el desempeño de los controladores de la clase *MPC_RP* se vió afectado de forma diferente, haciendo que su tendencia a hipoglucemia aumente considerablemente. En ocasiones se registraron pacientes con niveles de glucosa por debajo de 70 [mg/dL], lo que se debe a que si los controladores generan una alta dosificación de insulina y esta se inyecta con un retardo respecto al momento en que se produce la perturbación, la acción de dicha insulina se produce en forma retardada haciendo que la glucosa del paciente llegue a niveles hipoglucémicos. Por lo tanto, se recomienda usar para los controladores de la clase *MPC_RP* un estimador de perturbaciones que presente menos retardo como por ejemplo un observador de perturbaciones.

El conjunto controlador-estimador usando el controlador *MPC_QDMC2* presentó resultados favorables, manteniendo a todos los pacientes fuera de condición de hipoglucemia en las pruebas con modelo incierto. Adicionalmente, presentó 26 de 30 resultados satisfactorios para evitar hiperglucemia en comparación con el bolo óptimo, demostrando ser robusto. Por otra parte, el controlador *MPC_RP* permitió que un paciente in silico presentara condición de hipoglucemia tanto en pruebas con modelo nominal, como en las pruebas de modelo incierto. Sin embargo, este controlador obtuvo los mejores resultados respecto a prevención de hiperglucemia y su comportamiento se asemeja al Event-triggered model predictive control del autor Chakrabarty, por lo tanto se considera robusto.

Referencias Bibliográficas

- Barrett, K. E. (2019). *Ganong's review of medical physiology*. Number 1. : McGraw Hill Education.
- Bequette, B. W. (2012). Challenges and recent progress in the development of a closed-loop artificial pancreas. *Annual reviews in control*, 36(2):255–266.
- Best, M. J. (2017). *Quadratic programming with computer programs*. CRC Press.
- Bishop, C. M. (2006). *Pattern recognition and machine learning*. springer.
- Chakrabarty, A., Zavitsanou, S., Doyle, F. J., and Dassau, E. (2017). Event-triggered model predictive control for embedded artificial pancreas systems. *IEEE Transactions on Biomedical Engineering*, 65(3):575–586.
- Cobelli, C., Dalla Man, C., Sparacino, G., Magni, L., De Nicolao, G., and Kovatchev, B. P. (2009). Diabetes: models, signals, and control. *IEEE reviews in biomedical engineering*, 2:54–96.
- Cobelli, C., Renard, E., and Kovatchev, B. (2011). Artificial pancreas: past, present, future. *Diabetes*, 60(11):2672–2682.
- Colmegna, P. and Peña, R. S. (2014). Analysis of three t1dm simulation models for evaluating robust closed-loop controllers. *Computer methods and programs in biomedicine*, 113(1):371–382.

- Dalla Man, C., Camilleri, M., and Cobelli, C. (2006). A system model of oral glucose absorption: validation on gold standard data. *IEEE Transactions on Biomedical Engineering*, 53(12):2472–2478.
- Dalla Man, C., Micheletto, F., Lv, D., Breton, M., Kovatchev, B., and Cobelli, C. (2014). The UVA/PADOVA type 1 diabetes simulator: New features. *Journal of Diabetes Science and Technology*, 8.
- Dormand, J. R. (1996). *Numerical methods for differential equations: a computational approach*, volume 3. CRC press.
- Dormand, J. R. and Prince, P. J. (1980). A family of embedded runge-kutta formulae. *Journal of computational and applied mathematics*, 6(1):19–26.
- El Hachimi, M., Ballouk, A., and Baghdad, A. (2019). Rapid model predictive controller for artificial pancreas. In *Recent Advances in Electrical and Information Technologies for Sustainable Development*, pages 67–74. Springer.
- Federation, I. D. (2019). Idf diabetes atlas ninth edition 2019.
- Ho, Y. (2019). Model predictive controller using interior point and ant algorithm. In *Patient-Specific Controller for an Implantable Artificial Pancreas*, pages 49–61. Springer.
- Kovatchev, B. P., Breton, M., Dalla Man, C., and Cobelli, C. (2009). In silico preclinical trials: a proof of concept in closed-loop control of type 1 diabetes.

Maciejowski, J. M. (2002). *Predictive control: with constraints*. Pearson education.

Molano-jiménez, A. and León-vargas, F. (2017). UVa / Padova T1DMS Dynamic Model Revision.

Reichhartinger, M., Koch, S., Niederwieser, H., and Spurgeon, S. K. (2018). The robust exact differentiator toolbox: Improved discrete-time realization. In *2018 15th International Workshop on Variable Structure Systems (VSS)*, pages 1–6. IEEE.

Rodriguez, R., Villamizar, R., and Padilla, D. (2018). Simulación en tiempo real de la dinámica de glucosa para diferentes pacientes con diabetes mellitus tipo i, orientada a la implementación "hardware in the loop".

Simon, D. (2006). *Optimal state estimation: Kalman, H infinity, and nonlinear approaches*. John Wiley & Sons.

Visentin, R., Campos-Náñez, E., Schiavon, M., Lv, D., Vettoretti, M., Breton, M., Kovatchev, B. P., Dalla Man, C., and Cobelli, C. (2018). The uva/padova type 1 diabetes simulator goes from single meal to single day. *Journal of diabetes science and technology*, pages 273–281.

Apéndices

Apéndice A. Programación orientada a objetos

Teniendo en cuenta que la finalidad de esta investigación era diseñar sistemas que pudieran ser programados en dispositivos de computo para hacer pruebas de “*hardware in the loop*”, así como usarlos en cualquier entorno de simulación, reutilizarlos y modificar sus parámetros con facilidad, se evitó usar al máximo las toolbox y funciones de de Matlab para implementar de estos sistemas, principalmente porque se genera una dependencia hacia la toolbox “*Coder*” para poder migrar los códigos hacia algún lenguaje de programación fuera de Matlab, y por otro lado, restringe cambios o adiciones que se pueden hacer en los propios algoritmos.

Por esta razón la mayoría de los sistemas de este proyecto: controladores, observadores, estimadores, modelos de pacientes consisten en códigos programados desde cero, basados en programación orientada a objetos (OOP “*Object-Oriented Programming*”). Se decidió programar de esta manera en Matlab debido a la serie de ventajas que presenta:

- La estructura de un algoritmo OOP es la misma ya sea en Matlab o en otro lenguaje, facilitando la migración del código a mano.
- Un objeto es una estructura sumamente versátil, que permite ser usada con facilidad en un script de Matlab o como un bloque en Simulink, permitiendo realizar cualquier tipo de prueba.

- A pesar de que un objeto es muy fácil de usar, es una estructura sumamente compleja, que permite tener propiedades, funciones internas, almacenar variables y procesarlas.
- Un objeto es una estructura reutilizable, permitiendo generar una serie de sistemas del mismo tipo con diferentes características, por ejemplo, muchos controladores sintonizados de diferente manera, muchos modelos de diferentes pacientes.

En el desarrollo de los algoritmos de control, únicamente se usaron dos funciones de Matlab, la función `c2d` para discretizar y la función `quadprog` para resolver el problema de optimización. Sin embargo, se sabe que existen funciones que pueden encontrarse en la red para diferentes lenguajes de programación que pueden reemplazar a las anteriormente mencionadas cuando se desee migrar el código.

Clase modelo

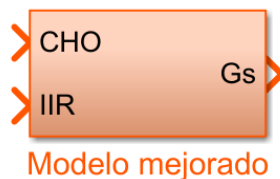


Figura 41. Subsistema de Simulink que contiene los sistemas de bloques descritos en la sección Modelo Diabético.

La figura 41 muestra un subsistema de Simulink que contiene el conjunto de sistemas de bloques que describen el modelo UVA PADOVA especificado en la sección 2.1.4. Se trata de un sistema continuo que será usado para realizar todas las validaciones de pacientes in silico, emulando a un paciente diabético de la misma forma que lo hace el software T1DMS, pero con una serie de ventajas. Se deben cargar los parámetros del paciente correspondientes en el Workspace de

Matlab antes de ejecutar el programa, para facilitar el proceso, se cuenta con un banco de archivos precargados de todos los pacientes disponibles.

El algoritmo desarrollado para el modelo diabético fue guardado como la clase *UVA_PADOVA_2.m* que se muestra en la figura 47. Una clase en Matlab se diferencia de otro tipo de script por llevar un icono con un cubo naranja como se muestra en la figura.

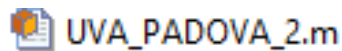


Figura 42. Archivo que contiene la clase *UVA_PADOVA_2* en Matlab.

A continuación se muestra brevemente como usar la clase *UVA_PADOVA_2* en un sistema de Simulink o en un script de MATLAB.



Figura 43. Subsistema de Simulink que contiene el Matlab System en el que se instancia el objeto del modelo del paciente diabético.

La figura 43 muestra el subsistema que contiene el modelo diabético representado por el objeto de la clase *UVA_PADOVA_2*. Para instanciar una clase en un objeto y usarlo en Simulink se usa el bloque Matlab System y se selecciona la clase que se desee utilizar.

La figura 44 muestra el bloque Matlab System ya instanciado con el objeto del modelo diabético, este proceso es similar a usar bloques de toolbox de Simulink, permitiendo trabajar de forma modular, configurar parámetros para generar fácilmente distintos sistemas de una misma clase, en este caso, distintos pacientes.

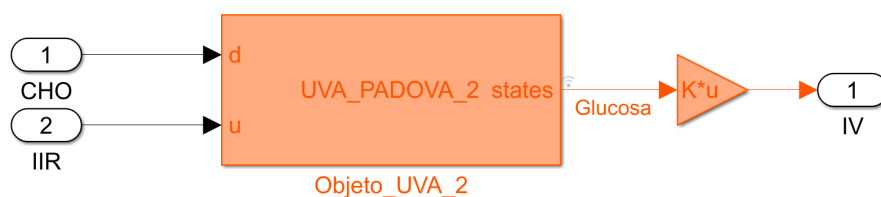


Figura 44. Clase UVA_PADOVA_2 instanciada en el bloque Matlab System de Simulink.

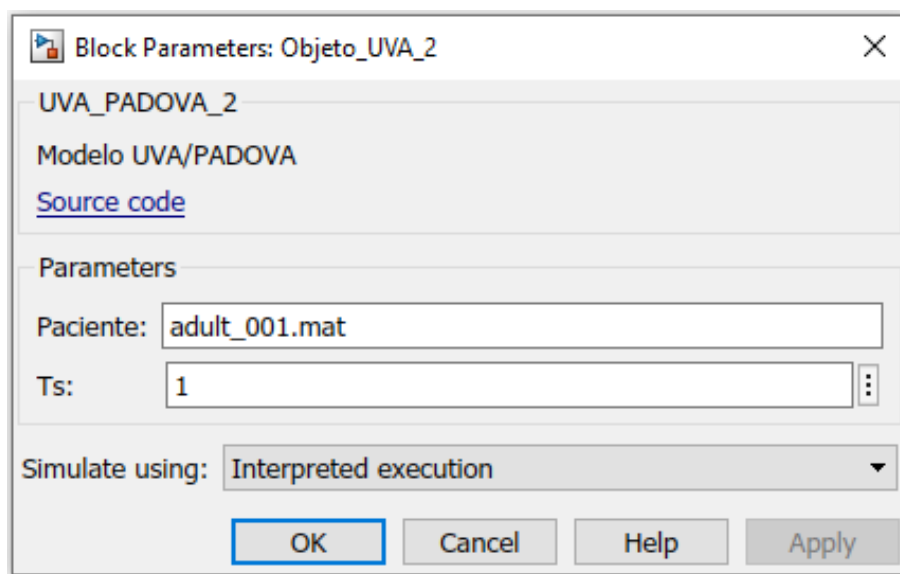


Figura 45. Parámetros configurables para el objeto del modelo diabético.

En la figura 45 se muestran los parámetros que se pueden modificar en el objeto del paciente diabético, que son el periodo de muestreo del objeto y el paciente que se desea simular. Todo esto facilita mucho la validación de controladores y sistemas en Simulink para distintos pacientes diabéticos. Sin embargo, en Matlab se puede aprovechar este tipo de programación en el desarrollo de scripts complejos valiéndose del concepto de sistemas modulares, donde se puede llamar el modelo como una función para usarse con facilidad en cualquier código, por ejemplo, en algoritmos de optimización.

La figura 46 muestra como se instancia la clase UVA_PADOVA_2 en el objeto planta, como

Instanciación de la clase en el objeto:

```
modelo = UVA_PADOVA_2();
```

Configuración de parámetros:

```
modelo.Ts = 1;  
modelo.Paciente = 'adult_001';
```

Uso del objeto como función:

```
states = modelo(CHO,u);
```

Figura 46. Instanciación, configuración de parámetros y uso como función de la clase UVA_PADOVA_2 en MATLAB.

se configuran sus parámetros y finalmente como se usaría como función en un script de MATLAB.

Clase controlador

El primer algoritmo de control predictivo desarrollado fue guardado como la clase *MPC_QDMC.m* que se muestra en la figura 47.



Figura 47. Archivo que contiene la clase MPC_QDMC en Matlab.

En la figura 48 se muestra la clase “*MPC_QDMC.m*” instanciada en un bloque Matlab System de Simulink para usarla en simulación.

En la figura 50 se muestra como instanciar la clase, configurar sus parámetros y usar el objeto resultante como función en un script de MATLAB.

El segundo algoritmo de control predictivo desarrollado fue guardado como la clase *MPC_RP.m* que se muestra en la figura 51.

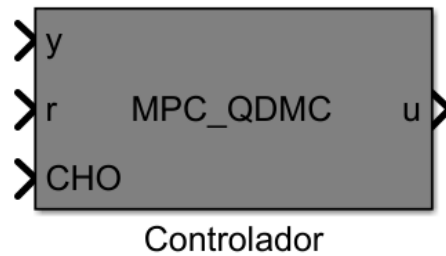


Figura 48. Clase MPC_QDMC instanciada en el bloque Matlab System de Simulink.

La clase MPC_RP tiene los mismos parámetros de configuración que la clase MPC_QDMC, por lo tanto, se usan de la misma forma en MATLAB y Simulink, teniendo en cuenta la diferencia que existe en las entradas de los objetos generados con estas clases.

El tercer algoritmo de control predictivo desarrollado fue guardado como la clase *MPC_RPI.m* que se muestra en la figura 52.

Clase observador

Se programó el algoritmo del filtro de Kalman extendido como la clase *EKF_UVA.m*. Esta clase recibe como entradas a la salida del proceso $y_p[k]$ (Glucosa subcutánea G_{sc} en [mg/dL]), la acción de control $u[k]$ (dosificación de insulina en [pmol/min]) y la perturbación de carbohidratos $CHO[k]$ en mg; como salida entrega los 16 estados estimados del modelo UVA PADOVA.

En la figura 54 se muestra la clase “*EKF_UVA.m*” instanciada en un bloque Matlab System de Simulink para usarla en simulación.

La figura 55 muestra la interfaz de configuración de parámetros de un objeto de la clase EKF_UVA en Simulink.

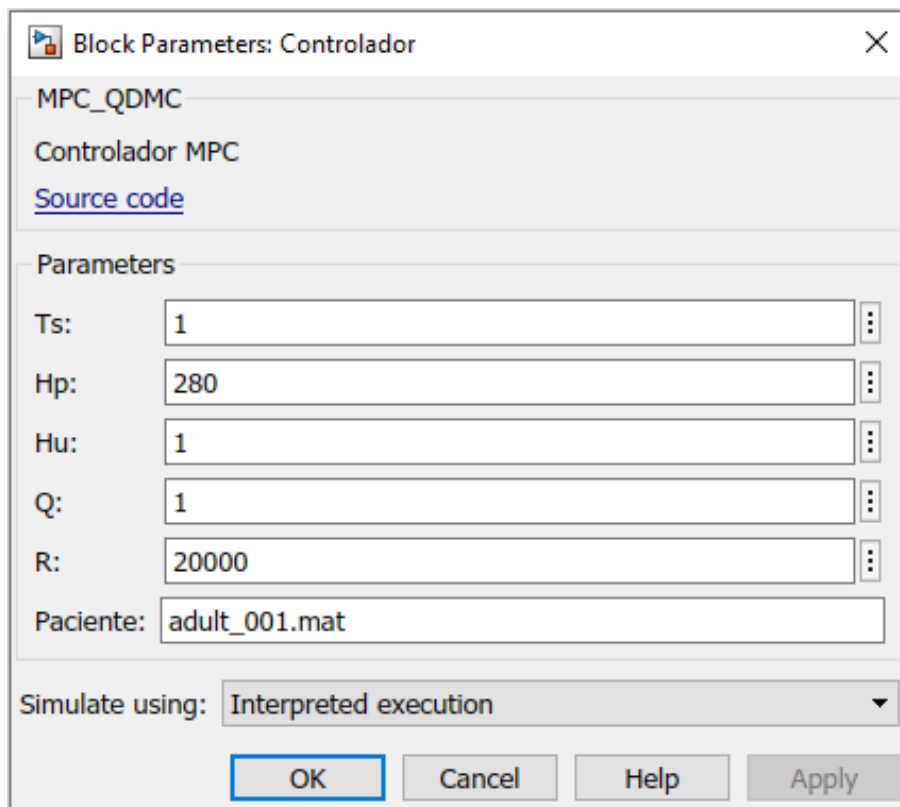


Figura 49. Configuración de parámetros de la clase MPC_QDMC.

Clase estimador

Se creó la clase ESTIMADOR_CHO para usar cualquier estimador con facilidad, esta clase tiene 3 entradas ($G_{sc}, \frac{dG_{sc}}{dt}, \frac{d^2G_{sc}}{dt^2}$), una salida (CHO) y como parámetros de configuración recibe el periodo de muestreo y el modelo del clasificador previamente entrenado. El algoritmo desarrollado para el estimador de perturbaciones se guardó como la clase *ESTIMADOR_CHO.m* que se muestra en la figura 56.

La figura 57 muestra el bloque Matlab System instanciado con el objeto del sistema de adecuación de señales.

Instanciación de la clase en el objeto:

```
mpc = MPC_QDMC();
```

Configuración de parámetros:

```
mpc.Ts = 1;  
mpc.Hp = 280;  
mpc.Hu  
mpc.Q  
mpc.R  
mpc.Paciente = 'adult_001';
```

Uso del objeto como función:

```
u = mpc(y,r,CHO);
```

Figura 50. Instanciación, configuración de parámetros y uso como función de la clase MPC_QDMC en MATLAB.



Figura 51. Archivo que contiene la clase MPC_RP en Matlab.

La figura 58 muestra la interfaz de configuración de parámetros de un objeto de la clase ESTIMADOR_CHO en Simulink.

La figura 59 muestra la instanciación, configuración de parámetros y uso como función de un objeto de la clase ESTIMADOR_CHO en MATLAB. Finalmente, se desarrollo el script de MATLAB *diseño_clasificador.m* que junto a un sistema de Simulink *sistema_clasificador.slx*, se encarga de generar generar señales de entrenamiento, ajustarlas, entrenar el clasificador y crear el



Figura 52. Archivo que contiene la clase MPC_RPI en Matlab.



Figura 53. Archivo que contiene la clase EKF_UVA en Matlab.

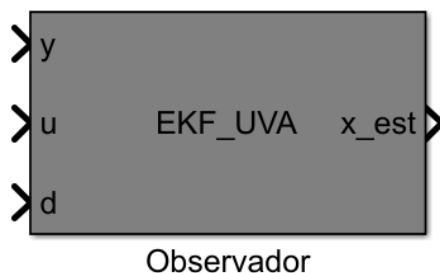


Figura 54. Clase *EKF_UVA* instanciada en el bloque Matlab System de Simulink.

modelo de clasificación para que el proceso de diseño de cada clasificador sea automático y solo requiera cambiar el controlador que usará el estimador en el sistema de Simulink.

Clase filtro de señales

El algoritmo desarrollado para el sistema de adecuación de señales se guardo como la clase *KALMAN_FMM.m* que se muestra en la figura 60. La clase *KALMAN_FMM* recibe como entradas la acción de control $u[k]$ y la salida de glucosa sensada $G_{sc}[k]$ para filtrar y generar los predictores $G_{sc}[k]$, $dG_{sc}[k]$ y $d^2G_{sc}[k]$.

La figura 61 muestra el bloque Matlab System instanciado con el objeto del sistema de adecuación de señales.

La figura 62 muestra la interfaz de configuración de parámetros de un objeto de la clase *KALMAN_FMM* en Simulink.

La figura 63 muestra la Instanciación, configuración de parámetros y uso como función de un objeto de la clase *KALMAN_FMM* en MATLAB.

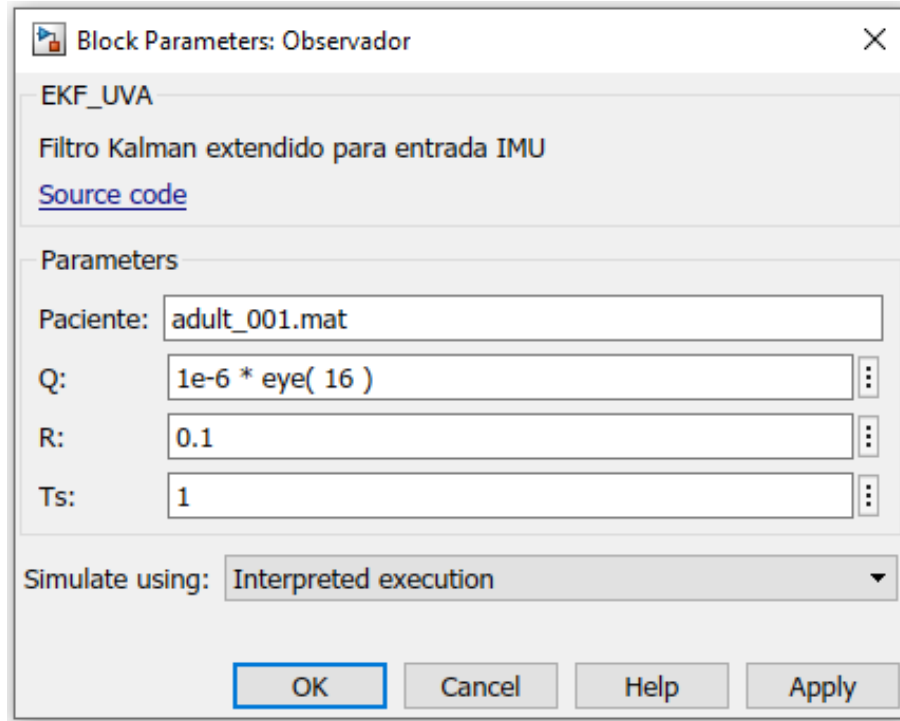


Figura 55. Configuración de parámetros de la clase EKF_UVA.

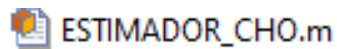


Figura 56. Archivo que contiene la clase ESTIMADOR_CHO en Matlab.



Figura 57. Clase ESTIMADOR_CHO instanciada en el bloque Matlab System de Simulink.

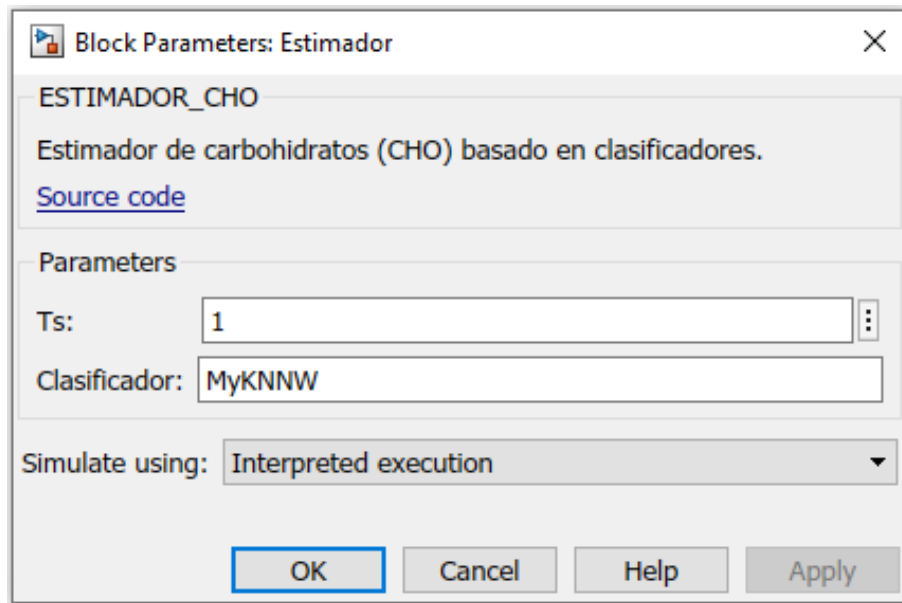


Figura 58. Parámetros configurables del objeto de la clase ESTIMADOR_CHO de señales en Simulink.

Instanciación de la clase en el objeto:

```
estimador = ESTIMADOR_CHO();
```

Configuración de parámetros:

```
estimador.Ts = 1;
estimador.Clasificador = 'MyKNNW';
```

Uso del objeto como función:

```
CHO = estimador(Gsc,dGsc,d2Gsc);
```

Figura 59. Instanciación, configuración de parámetros y uso como función de la clase ESTIMADOR_CHO en MATLAB.

 KALMAN_FMM.m

Figura 60. Archivo que contiene la clase KALMAN_FMM en Matlab.

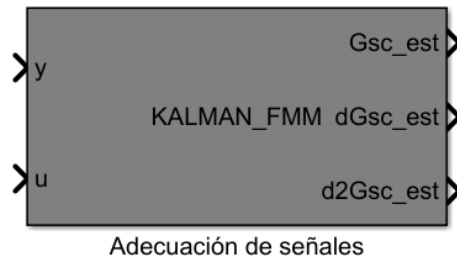


Figura 61. Clase KALMA_FMM instanciada en el bloque Matlab System de Simulink.

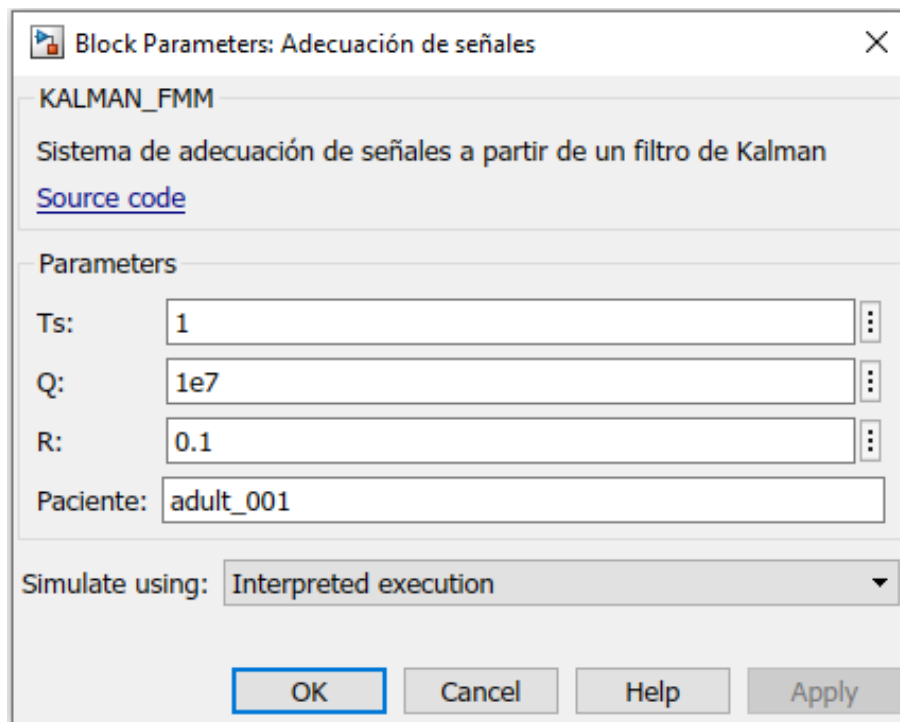


Figura 62. Parámetros configurables del objeto de la clase KALMAN_FMM de señales en Simulink.

Instanciación de la clase en el objeto:

```
adecuacion = KALMAN_FMM();
```

Configuración de parámetros:

```
adecuacion.Ts = 1;  
adecuacion.Paciente = 'adult_001';
```

Uso del objeto como función:

```
[Gsc, dGsc, d2Gsc] = adecuacion(y,u);
```

Figura 63. Instanciación, configuración de parámetros y uso como función de la clase KALMAN_FMM en MATLAB.

Apéndice B. Hardware

Sensor de glucosa

El bloque sensor de glucosa agrega ruido de medición (media cero y 0.4 [mg/dL] de amplitud), cuantización (0.05 [mg/dL]) de la glucosa medida y periodo de muestreo del sensor (1 [min]), en la figura 64 se muestra el contenido del bloque.

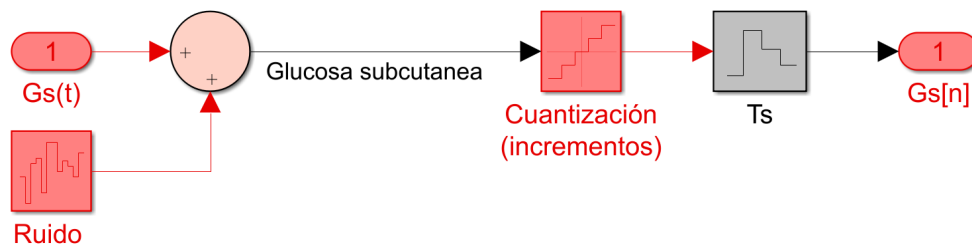


Figura 64. contenido del bloque sensor de glucosa.

Bomba de insulina

El bloque bomba de insulina agrega saturación en la inyección de insulina (0.0 a 35 [U/h]), cuantización en los incrementos de insulina (0.025 [U/h]), periodo de muestreo de la bomba (1 [min]) y ruido asociado a errores de inyección (0.01 [U/h]), en la figura 65 se muestra el contenido del bloque (Basado en la bomba del dispositivo MiniMed Veo de Medtronic).

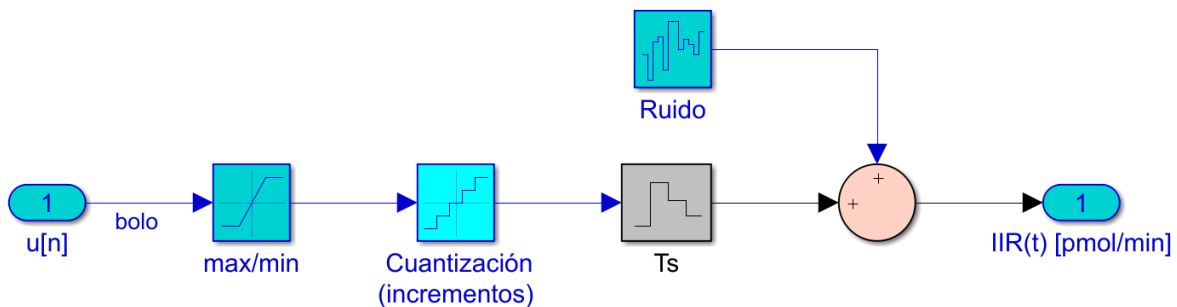


Figura 65. Contenido del bloque bomba de insulina.

Generación de perturbación

El bloque ingesta de alimento se diseñó para generar pulsos de ancho 15 minutos y altura variable, los pulsos representan la cantidad de carbohidratos en miligramos que consume el paciente durante la ingesta de alimentos. Se puede modificar fácilmente la cantidad de pulsos y el momento en el que ocurren. En la figura 66 se muestra un ejemplo en el que se simula la ingesta de 60, 80 y 50 gramos de carbohidratos a las 7 a.m, 12 m y 6p.m respectivamente.

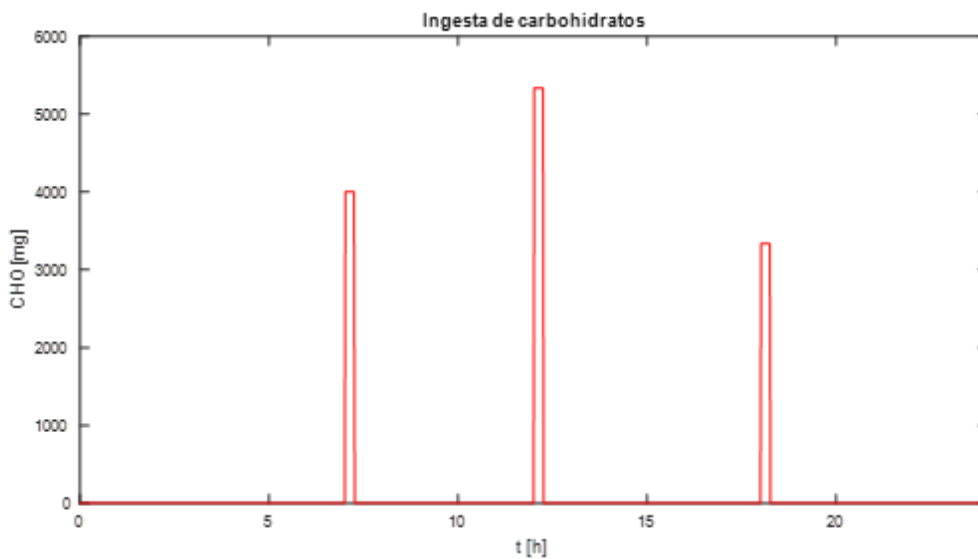


Figura 66. Ingesta de 60, 80 y 50 gramos de carbohidratos a las 7 a.m, 12 m y 6p.m respectivamente.

Los bloques modelo de paciente y sistema de control se explican en las secciones posteriores debido a que se aplicó programación orientada a objetos en su desarrollo y conceptualmente son de mayor complejidad.