

Trabajo Fin de Grado
Ingeniería Electrónica, Robótica y Mecatrónica

Modelo y control automático de la glucosa en sangre
en pacientes con Diabetes Mellitus tipo 1 (DM1)

Autor: Sofía Tarín Roldán

Tutor: Manuel Gil Ortega Linares

Dpto. Ingeniería de Sistemas y Automática
Escuela Técnica Superior de Ingeniería
Universidad de Sevilla

Sevilla, 2021



Ingeniería de Sistemas y Automática



Trabajo Fin de Grado
Ingeniería Electrónica, Robótica y Mecatrónica

Modelo y control automático de la glucosa en sangre en pacientes con Diabetes Mellitus tipo 1 (DM1)

Autor:

Sofía Tarín Roldán

Tutor:

Manuel Gil Ortega Linares

Catedrático de Universidad

Dpto. Ingeniería Automática y Sistemas

Escuela Técnica Superior de Ingeniería

Universidad de Sevilla

Sevilla, 2021

Trabajo Fin de Grado: Modelo y control automático de la glucosa en sangre en pacientes con Diabetes Mellitus tipo 1 (DM1)

Autor: Sofia Tarín Roldán

Tutor: Manuel Gil Ortega Linares

El tribunal nombrado para juzgar el Trabajo arriba indicado, compuesto por los siguientes miembros:

Presidente:

Vocales:

Secretario:

Acuerdan otorgarle la calificación de:

Sevilla, 2021

El Secretario del Tribunal

*A todos los que nunca dejan de
creer en sus sueños y luchan por
ellos cada día latido a latido*

Agradecimientos

Gracias a Manuel Gil, mi tutor durante este trabajo. Desde la primera clase consiguió transmitirme algo que creo que es imprescindible en el aprendizaje: la curiosidad. Y también desde la primera tutoría en la que le hablé de este trabajo para proponérselo me brindó todo su apoyo, un apoyo que se ha mantenido a lo largo del mismo hasta acabarlo. Sin su gran ayuda este trabajo no sería posible.

Gracias a mi familia, a mis padres, a mis tías y a mis abuelos, por todo lo que me han dado en la vida, y por todo lo que me han enseñado. Gracias por todo el esfuerzo que cada uno de ellos han hecho, de una manera u otra, para que a día de hoy yo pueda estar escribiendo estas líneas y acabando esta etapa de mi vida. Gracias por el apoyo y el cariño incondicional.

Gracias a Rafa, el mejor regalo que me ha podido dar esta aventura en la universidad. Gracias por estar siempre a mi lado para sacarme una sonrisa cuando más lo necesito, por querer compartir conmigo esta montaña rusa que es la vida y por no soltarme nunca de la mano. Estoy muy orgullosa de lo que hemos construido y seguimos construyendo día tras día juntos.

Gracias a todos mis amigos, especialmente a los que me ha dado el grado, tanto dentro de clase como fuera de ella. Me habéis acompañado y apoyado durante estos 4 años siendo mi segunda familia aquí en Sevilla, y espero que así sea toda la vida. En especial a Pablo, que tantos y tantos momentos hemos vivido juntos desde el comienzo de esta etapa, algunos mejores y otros peores, pero sabemos que siempre estamos ahí cerca para apoyarnos el uno al otro.

Gracias a los profesores y maestros que he tenido a lo largo de mi vida por todo lo que me han podido enseñar. Cada uno de ellos y ellas han puesto su granito de arena para que hoy sea la persona que soy. En especial, gracias a Francisco Javier León, un verdadero apasionado por el mundo de la tecnología que me transmitió su sana locura, especialmente por los robots, y con ello que hace cuatro años yo eligiera este grado y decidiera dedicarme a esto.

Por último, gracias a todas las personas que, de una forma u otra, me han apoyado y han hecho posible este trabajo y lo que ello ha conllevado.

Sofía Tarín Roldán

Sevilla, 2021

Resumen

La Diabetes Mellitus es una enfermedad que afecta al sistema metabólico, en concreto al páncreas y a la hormona insulina, y que es una de las principales causas de muerte debido al cada vez mayor aumento de casos en la población y las enfermedades y posibles complicaciones que puede generar en caso de no estar bien controlada.

En este trabajo me centraré en especial en la Diabetes Mellitus tipo 1, de origen autoinmune, para la que aún no se ha desarrollado una cura. Es por ello que en este tipo de diabetes un buen control de la glucosa en sangre es esencial, haciendo que esta se encuentre el máximo tiempo posible en un rango de valores normales mediante inyecciones externas de insulina.

A lo largo del trabajo se presentan diferentes técnicas de control aplicadas a un modelo matemático biológico que simula el sistema glucorregulatorio del cuerpo humano, usando para ello las herramientas MATLAB Y Simulink. Primero se estudia y se simula el modelo escogido, basado en el Modelo de Bergman, para analizar su comportamiento. Posteriormente, se diseña un primer controlador PID básico, tanto en tiempo continuo como discreto, al que más tarde se le añaden una serie de mejoras para desarrollar un control más realista. Por último, se diseña un controlador basado en MPC como técnica de control más avanzado. Se realizan pruebas con cada uno de los controladores llevados a cabo de las que posteriormente se podrán sacar unas comparaciones entre los controladores y unas conclusiones finales.

Abstract

Diabetes Mellitus is a chronic disease that affects the metabolic system, specifically the pancreas and the insulin hormone, and which is one of the main death causes due to the each-time-higher increase of cases in population, as well the different diseases and possible complications that can be caused in case the diabetes is not correctly controlled.

This project is focused on type 1 Diabetes Mellitus, with an autoimmune origin, that has no known cure yet. Hence, a good control of the glucose in the blood is essential in this type of diabetes, getting this glucose to be in the range of normal values as long as possible thanks to external insulin injections.

Throughout the project, different control techniques applied to a mathematical biologic model which simulates the glucoregulatory system are presented, using for these the MATLAB and Simulink tools. First of all, the model chosen, which is based on the Bergman's Model, is studied and simulated in order to analyze its performance. Afterwards, a first, basic, PID controller is designed, both in continuous and discrete time. A series of improvements are lately added to this PID controller in order to make it more realistic. Finally, a MPC controller is designed as a more advanced control technique. Different experiments are carried out with each one of the controllers that have been designed, so we can realize a comparison between them and draw some final conclusions.

Índice

Agradecimientos	ix
Resumen	xi
Abstract	xiii
Índice	xiv
Índice de Tablas	xvi
Índice de Figuras	xviii
1 Introducción	1
1.1 Tipos de diabetes	2
1.2 Monitorización de glucosa	3
1.3 Tratamiento con insulina en Diabetes Tipo	4
1.3.1 Principales tipos de insulina	4
1.3.2 Formas de administración de la insulina	5
2 Modelo del sistema gluco-regulatorio	7
2.1 Modelo Minimalista de Bergman	7
2.1.1 Ecuaciones del Modelo de Bergman	8
2.1.2 Modificaciones realizadas	9
2.2 Implementación y simulación del Modelo de Bergman	10
2.2.1 Bloque "Modelo_Bergman"	10
2.2.2 Bloques de perturbaciones. Protocolo de pruebas	11
2.3 Linealización del modelo	12
3 Control PID	15
3.1 Controlador en tiempo continuo	15
3.1.1 Especificaciones definidas para el controlador	15
3.1.2 Diseño del controlador PID en tiempo continuo	16
3.1.3 Implementación del controlador en tiempo continuo en Simulink	18
3.1.4 Pruebas con el controlador PID en tiempo continuo	19
3.2 Controlador PID en tiempo discreto	22
3.2.1 Diseño del controlador PID en tiempo discreto	22
3.2.2 Implementación del controlador PID en tiempo discreto en Simulink	23
3.2.3 Pruebas con el controlador PID en tiempo discreto	23
3.3 Modificaciones y mejoras realizadas en el controlador PID en tiempo discreto	25
3.3.1 Controlador PID discreto con algoritmo anti-windup	25
3.3.2 Relajación en el tiempo de muestreo en el PID	26
3.3.3 Control PID con inyección de insulina por lotes	28

3.3.4	Control PID Feedforward	31
4	Control Predictivo basado en el Modelo (MPC)	35
4.1.	<i>Introducción al Control Predictivo</i>	35
4.2.	<i>Implementación del controlador MPC en Matlab y Simulink</i>	36
5	Comparaciones y conclusiones finales	43
	Referencias	47
	Anexos	49

ÍNDICE DE TABLAS

Tabla 2-1. Parámetros del Modelo de Bergman	9
Tabla 2-2. Parámetros de la ecuación de perturbaciones del Modelo de Bergman	9
Tabla 2-3. Protocolo de perturbaciones de comida	12
Tabla 2-4. Protocolo de perturbaciones de ejercicio físico	12
Tabla 5-1. Comparación cuantitativa de los controladores	44

ÍNDICE DE FIGURAS

Figura 1-1. Poster informativo sobre la diabetes	1
Figura 1-2. Medición de la glucemia con un glucómetro	3
Figura 1-3. Medición de la glucemia con un dispositivo de monitorización continua	4
Figura 1-4. Plumas de distintos tipos de insulina y aguja desechable	5
Figura 1-5. Ejemplo de bomba de insulina	6
Figura 2-1. Esquema del Modelo de Bergman	8
Figura 2-2. Modelo de Bergman en Simulink	11
Figura 2-3. Interior del bloque "Modelo_Bergman"	11
Figura 3-1. Bode $G_m(s)$	16
Figura 3-2. Bode $G_m(s)*C(s)$ sin ajuste de K_p	17
Figura 3-3. Bode de $G_m(s)*C(s)$ final	18
Figura 3-4. Modelo con controlador PID continuo en Simulink	19
Figura 3-5. Comportamiento del modelo de Bergman sin perturbaciones	19
Figura 3-6. Comportamiento del modelo de Bergman con perturbaciones	20
Figura 3-7. Control PID continuo sin perturbaciones	21
Figura 3-8. Control PID continuo con perturbaciones	21
Figura 3-9. Modelo con controlador PID discreto en Simulink	23
Figura 3-10. Bloque del controlador PID en tiempo discreto en Simulink	23
Figura 3-11. PID discreto sin perturbaciones	24
Figura 3-12. PID discreto con perturbaciones	24
Figura 3-13. PID con algoritmo anti-windup	25
Figura 3-14. PID con tiempo de muestreo de 3 minutos	27
Figura 3-15. PID con tiempo de muestreo de 5 minutos	27
Figura 3-16. PID con tiempo de muestreo de 10 minutos	28
Figura 3-17. Bloques del control PID con inyección de insulina por lotes	29
Figura 3-18. PID con inyección de insulina por lotes	29
Figura 3-19. Zoom del PID con inyección de insulina por lotes	30
Figura 3-20. Tiempo de inyección de insulina	30
Figura 3-21. Modelo del controlador PID Feedforward	31
Figura 3-22. Esquema para la obtención de C_{ff}	32
Figura 3-23. Bloque del control PID Feedforward	33
Figura 3-24. Controlador PID Feedforward con inyección de insulina por lotes	33

Figura 4-1. Modelo del controlador MPC	37
Figura 4-2. Bloque del control MPC	37
Figura 4-3. Estructura del controlador MPC	38
Figura 4-4. Barra de herramientas para el diseño del controlador MPC	38
Figura 4-5. Controlador MPC HP: 300 s; Hc: 10 s; wu: 0.1; wg: 1	38
Figura 4-6. Controlador MPC HP: 300 s; Hc: 100 s; wu: 0.1; wg: 1	39
Figura 4-7. Controlador MPC HP: 300 s; Hc: 100 s; wu: 0.1; wg: 10	39
Figura 4-8. Controlador MPC HP: 500 s; Hc: 50 s; wu: 0.1; wg: 50	40
Figura 4-9. Controlador MPC HP: 300 s; Hc: 50 s; wu: 0.1; wg: 50	40

1 INTRODUCCIÓN

“Cuando el corazón y la mente están unidos, todo es posible”.

- Diego Pablo Simeone -

La diabetes mellitus (DM) es una enfermedad crónica del sistema metabólico que se produce cuando el páncreas deja de producir insulina, o bien la insulina que segrega tiene una calidad inferior y el organismo no es capaz de usarla de una manera eficaz. La insulina es una hormona cuya función es regular la glucosa (azúcar) en sangre. El resultado principal de la diabetes son las hiperglucemias (aumento de la glucosa en la sangre por encima de los valores normales), que a largo plazo pueden causar daños muy importantes en diferentes órganos, así como provocar el desarrollo de otras enfermedades.

Según la OMS (Organización Mundial de la Salud), en 2014 el número total de personas en el mundo que sufrían diabetes era de 422 millones, con un ascenso de 314 millones desde el año 1980. La prevalencia en adultos se sitúa entorno al 8.5%, y alrededor de 1.6 millones de muertes en 2016 fueron causadas a la diabetes. Asimismo, la diabetes es una de las principales causas de ceguera, infartos y amputación de los miembros inferiores. [1]



Figura 1-1. Poster informativo sobre la diabetes

Una persona no diabética mantiene siempre sus niveles de glucosa en un rango de 70 a 110 mg/dL, aunque lo que haya comido tenga una alta cantidad de azúcares o hidratos de carbono. Dichos valores de glucosa forman los llamados “valores normales”. Este mantenimiento constante dentro del rango es gracias a la secreción rápida de insulina. Cuando una persona ingiere alimentos, los azúcares presentes en los mismos se absorben y pasan a la sangre. Los niveles de glucosa se elevan, pero inmediatamente las células que se encargan de producir insulina detectan dicho aumento de glucosa y segregan de forma rápida esta hormona, haciendo que la glucosa entre en los tejidos y músculos donde posteriormente se metabolizará para producir energía, y de esta forma se disminuye los niveles de glucosa en sangre.

Sin embargo en una persona con diabetes, como ya hemos dicho antes, la producción de insulina es muy baja o incluso nula, por lo que cuando la glucosa aumenta en sangre no se produce la secreción instantánea de insulina y los niveles se mantienen altos, por encima de los valores altos, imposibilitando la conversión de dicha glucosa en energía para las células. [2]

1.1 Tipos de diabetes

Se puede diferenciar 3 tipos de diabetes:

- Diabetes tipo 1: también denominada diabetes insulino dependiente o juvenil. Su principal característica es la producción muy deficiente de insulina, debida a la destrucción de tipo autoinmune de las células beta del páncreas. No se conoce a día de hoy la causa que conlleva a dicha destrucción autoinmune y, por tanto, a la diabetes tipo 1 (DM1). Se suele desarrollar a una edad temprana, sobre todo en niños y jóvenes, y sus síntomas se conocen como las 3 P: polidipsia (excesiva sed), poliuria (excesivas ganas de orinar) y polifagia (hambre constante). Además, también conlleva una importante pérdida de peso, cansancio extremo y trastornos visuales, y en caso más extremos donde la diabetes no se diagnostica a tiempo, puede llegar a causar cetoacidosis (ocurre cuando el cuerpo descompone la grasa demasiado rápido y el hígado la convierte en cetonas, volviendo a la sangre ácida), coma diabético o incluso la muerte. Una vez se diagnostica, se pone en tratamiento con administración externa de insulina para controlar la glucosa e intentar que los niveles sean lo más cercanos posible a los valores normales. Dicho tratamiento es de por vida ya que no existe aún cura para la diabetes tipo 1, de ahí el nombre de insulino dependiente.
- Diabetes tipo 2: también conocida como no insulino dependiente o de edad adulta. Esta diabetes se caracteriza por el uso poco eficaz de la insulina por parte del organismo, debido principalmente a que su calidad es peor. Las causas de la diabetes tipo 2 (DM2) son el sobrepeso, el sedentarismo (muy poca actividad física) y una dieta desequilibrada y rica en azúcares y grasas; de ahí a que se suele dar con mayor prevalencia en personas con una edad avanzada, aunque cada vez más se da en personas más jóvenes por malos hábitos de salud. Los síntomas característicos de la DM2 son parecidos a los de la de tipo 1, si bien normalmente no son tan intensos, lo cual hace que sea más difícil una detección precoz y se pueden pasar años con la enfermedad sin que esta haya sido diagnosticada. Por este motivo, cuando se detecta, suele haber algún tipo de complicación más. El tratamiento dependerá de lo avanzada que esté la diabetes. Si está en una fase inicial y los niveles de glucosa son ligeramente mayores a los normales, bastará con una dieta saludable, baja en azúcares y grasas, y actividad física. También se recomendará determinados hábitos saludables como no fumar o no tomar alcohol, para evitar complicaciones. Podrá tratarse con medicamentos orales para controlar la glucemia. Si, por el contrario, la diabetes está en una fase más avanzada, donde los niveles de glucosa son demasiado altos, se tratará con inyección externa de insulina (al igual que en la DM1), aunque si los niveles se van corrigiendo con el tiempo se podrá dejar este tratamiento y aplicar el mencionado anteriormente para fases más leves.

- Diabetes Gestacional: se caracteriza principalmente por hiperglucemias que se dan en algunas mujeres embarazadas. Aunque no se llegan a tener niveles de glucosa tan altos como para diagnosticar diabetes, sí que estos superan ligeramente los valores normales y recomendados. Sin embargo, este tipo de diabetes es igualmente peligrosa, pues hace que tanto la madre como el hijo puedan sufrir más complicaciones durante el embarazo y el parto. El diagnóstico de este tipo de diabetes se realiza mediante pruebas durante el embarazo, y para evitarla se recomienda una dieta equilibrada y ejercicio físico. [1]

1.2 Monitorización de la glucemia

Actualmente existen dos tipos de dispositivos con los cuales podemos realizar una monitorización de la glucosa en sangre o glucemia:

- Glucómetros: siendo el más usado en las últimas décadas, este dispositivo es un instrumento de medida de la glucemia capilar de forma inmediata. Su funcionamiento consiste en insertar una tira reactiva en el glucómetro, pinchar la yema del dedo con una aguja, y poner una pequeña gota de sangre en la tira reactiva. Pasados unos 5 segundos, obtendremos en la pantalla del glucómetro el valor de la glucemia capilar. Si bien es relativamente rápido, la principal y obvia desventaja de este método “tradicional” es lo invasivo que resulta, ya que puesto que una persona diabética necesita controlar varias veces al día sus niveles de glucosa en sangre, esto se traduce en una gran cantidad de pinchazos. Por otra parte, nos da datos de medidas puntuales y discontinuas, sin poder saber cómo ha variado la glucemia entre una medida y otra.



Figura 1-2. Medición de la glucemia con un glucómetro

- Sistema de monitorización continua de glucosa: son los nuevos dispositivos que, cada vez con mayor frecuencia, se usan a día de hoy. Se diferencian de los glucómetros porque, como su nombre indica, miden de forma continua la glucosa en sangre, cada minuto aproximadamente, lo cual es mucho más beneficioso. Estos sistemas se componen de un sensor con un filamento flexible el cual se inserta debajo de la piel, un monitor para observar los valores medidos (este monitor puede ser el propio teléfono móvil del paciente con una aplicación específica para esta tarea), y un transmisor que permite comunicarse al sensor con el monitor. Estos dispositivos miden el nivel de glucosa en el líquido intersticial (el espacio entre las células), que si bien difiere ligeramente del valor capilar, al final resulta ser prácticamente idéntico. Además, los sistemas de monitorización continua de glucosa suelen incluir alarmas que avisan cuando detectan que la glucosa supera, bien por arriba o por abajo, el rango establecido como deseado, disminuyendo así la probabilidad de tener hiperglucemias e hipoglucemias severas. [3]



Figura 1-3. Medición de la glucemia con un dispositivo de monitorización continua

1.3 Tratamiento con insulina en la Diabetes tipo 1

Puesto que para este trabajo me centraré en el control específicamente de la diabetes tipo 1, realizaré a continuación un breve repaso de los distintos tipos de tratamientos o terapias que existen para este tipo de diabetes. Como ya he mencionado antes, la diabetes tipo 1 se caracteriza por la nula secreción de insulina en el páncreas y, por consecuencia, es imprescindible para el paciente la administración exógena de insulina. Este procedimiento se podrá llevar a cabo de dos formas: mediante inyecciones de jeringuillas o plumas de insulina, o bien mediante una bomba de insulina. Para entender mejor estos dos tipos de administración, primero explicaré los tipos de insulina que existen. Principalmente, podemos distinguir dos tipos de esta hormona: la basal, que es la que genera el páncreas en pequeñas cantidades de forma continuada a lo largo del día, y los llamados bolos, que se corresponde con la insulina que se genera en mayor cantidad cuando la persona come y hay que contrarrestar la cantidad de glucosa ingerida.

1.3.1 Formas de administración de la insulina

En las personas diabéticas se usan dos tipos de insulina exógena que imitan el comportamiento de las dos mencionadas anteriormente: la insulina de acción prolongada y la de acción rápida o ultrarrápida.

- Insulina de acción prolongada: este tipo de insulinas tienen una duración aproximada de 24 horas. El comienzo de su efecto es alrededor de 2 horas después de la inyección y su curva de acción o actividad es prácticamente plana, es decir, que hace el mismo efecto desde esas 2 horas después de inyectarse hasta que se cumplen las 24 horas, sin tener picos de acción. Se utiliza como sustitución de la insulina basal.
- Insulina de acción rápida o ultrarrápida: la duración de estas insulinas son de aproximadamente 3 o 4 horas, empezando a hacer efecto a los 15 minutos después de la inyección. Su curva de actividad

presenta un pico en la hora posterior a la administración, es decir, cuando hace su máximo efecto. Esta insulina sustituye a los llamados bolos, la insulina que genera el páncreas después de la ingesta de alimentos. Es por eso que la insulina de acción ultrarrápida se administra antes de cada comida (desayuno, almuerzo y cena, y meriendas si es que se hacen), así como para corregir la glucosa cuando esta se encuentra a valores demasiado altos después de comer, si no se ha calculado bien la cantidad de insulina administrada antes de comer. [3]

1.3.2 Formas de administración de la insulina

Actualmente existen dos formas principales de administrar la insulina, tanto basal como prandial (bolos):

- Plumas: también conocidos como “bolígrafos” de insulina por su forma, estas vienen cargadas con un cartucho de un tipo de insulina y sirven para varias inyecciones, que serán más o menos dependiendo de la cantidad de insulina que necesita el paciente. El pinchazo es fácil y preciso, y se realiza colocando una pequeña aguja desechable tras la administración de unos 5 mm en la punta. Una vez se carga la cantidad deseada, solo hay que pulsar un botón y la insulina saldrá de forma inmediata, en apenas unos segundos. Las zonas más frecuentes donde se administra la insulina con las plumas son la zona superior de los brazos, los muslos, la parte inferior del abdomen y los glúteos.



Figura 1-4. Plumitas de distintos tipos de insulina y aguja desechable

- Bombas de insulina: son unos pequeños dispositivos programables que contienen insulina de acción rápida o ultrarrápida que se va administrando de forma continua durante todo el día al paciente gracias a un catéter o cánula que se inserta normalmente en la zona inferior del abdomen. Las bombas son portátiles, lo que permite que el paciente vaya de una parte a otra con ella sin que esta le moleste en la mayoría de sus tareas cotidianas. Normalmente las bombas funcionan en bucle abierto, es decir, no reciben información de los valores de glucemia, y por tanto es el propio paciente quien tiene que programarla en función de la comida que vaya ingerir o de los valores de glucemia que tenga para que se administre más o menos insulina.



Figura 1-5. Ejemplo de bomba de insulina

Este trabajo precisamente se centrará en diseñar el control automático de un dispositivo que combine las funciones de la bomba de insulina (para la administración de la misma) y la de un sistema de monitorización continua de glucosa (para obtener los valores de la glucosa), para que de esta forma la bomba funcione en bucle cerrado sin necesidad de que el paciente la programe en ningún momento, haciendo además que se obtengan de manera más prolongada y uniforme unos valores de glucosa en sangre adecuados. En la siguiente figura podemos observar el esquema de control que se corresponde al control en bucle cerrado que se quiere diseñar.

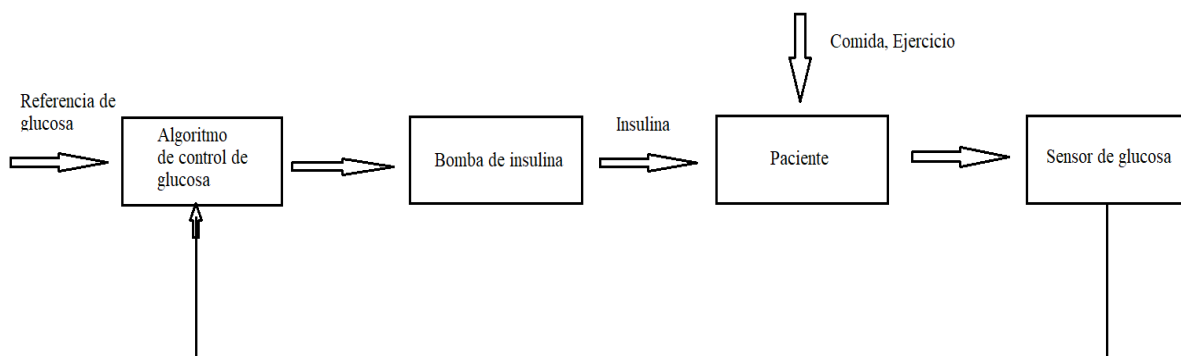


Figura 1-6. Esquema de control en bucle cerrado

2 MODELO DEL SISTEMA GLUCOREGULATORIO

*“La risa es atemporal. La imaginación no tiene edad.
Y los sueños son para siempre”.*

- Walt Disney-

Para desarrollar el control automático de la glucosa, primero necesitamos un modelo matemático-biológico del sistema gluco-regulatorio humano. El objetivo de este capítulo es elegir un modelo que sea útil para el objetivo de este trabajo, así como implementar dicho modelo en Simulink para observar y estudiar su comportamiento previamente a desarrollar el control.

2.1 Modelo Minimalista de Bergman

Desde la segunda mitad del siglo XX, distintos matemáticos e investigadores han propuesto diferentes modelos del sistema que regula la glucosa en el organismo tanto en animales como en humanos, para comprender mejor el comportamiento de este y también de la diabetes [4]. El elegido para este trabajo es uno de los más extendidos, el modelo desarrollado por Bergman et al. (1979) y conocido como “Modelo Minimalista de Bergman”. Es llamado así debido a que usa el mínimo número de depósitos o compartimientos para modelar el sistema gluco-regulatorio de la persona, que son 3. Por lo tanto, este modelo representa 3 dinámicas distintas en sus ecuaciones. Por una parte, la dinámica de la glucosa que se encuentra en el plasma sanguíneo representado en la ecuación (1). Por otra parte, la dinámica de la insulina que se encuentra en el plasma, en la ecuación (2), en cuyo compartimiento iría la insulina que se administra desde el exterior. Por último, la dinámica de la insulina efectiva que se encarga directamente de bajar los niveles de glucosa, en la ecuación (3) [5]. En la figura 2-1 se puede observar el esquema equivalente al Modelo de Bergman donde se representan los 3 depósitos mencionados anteriormente.

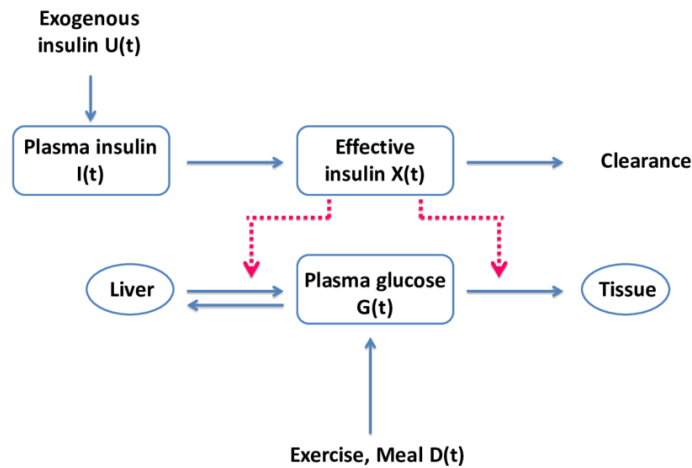


Figura 2-1. Esquema del Modelo de Bergman

2.1.1 Ecuaciones que describen el Modelo de Bergman

Las ecuaciones diferenciales no lineales son las que aparecen a continuación, tal y como las define Barragán et al.[2]:

$$\frac{dG(t)}{dt} = -P_1 G(t) - X(t)(G(t) + G_b) + D(t) \quad (1)$$

$$\frac{dI(t)}{dt} = -n(I(t) + I_b) + \frac{U(t)}{V_i} \quad (2)$$

$$\frac{dX(t)}{dt} = -P_2 X(t) + P_3 I(t) \quad (3)$$

Donde las variables que aparecen en las ecuaciones son las siguientes:

- $G(t)$ es la concentración de la glucosa en plasma (mmol/L) por encima del nivel basal G_b
- $I(t)$ es la concentración de la insulina en plasma (mU/L) por encima del nivel basal I_b
- $X(t)$ es la concentración de insulina efectiva (min^{-1})
- $U(t)$ es la insulina exógena que se administra (mU/min)
- $D(t)$ es la perturbación causada por la ingesta de comida (mmol/L min^{-1}), cuyos valores se calculan según la ecuación (4)
- G_b e I_b son los valores de glucosa e insulina basales, respectivamente
- n es el ratio de desaparición de insulina en plasma (min^{-1})
- V_i es el volumen de distribución de insulina (L)

En la siguiente tabla, Tabla 1, se definen los valores de las constantes en las ecuaciones:

Parámetro	Valor	Unidad
P ₁	0.028	min ⁻¹
P ₂	0.025	min ⁻¹
P ₃	0.000013	L/mU min ²
n	5/54	min ⁻¹
G _b	4.5	mmol/L
I _b	15	mU/L
V _i	12	L

Tabla 2-1. Parámetros del Modelo de Bergman

La ecuación que define la dinámica de las perturbaciones debidas a las comidas es la siguiente:

$$D(t) = \frac{D_g A_g t e^{-t/t_{max,i}}}{V_g t_{max,g}^2} \quad (4)$$

Donde D(t) tiene unidades de (mmol/L min⁻¹) como he dicho anteriormente y D_g es el valor en mmol de la cantidad de hidratos de carbono ingeridos en la comida.

En la Tabla 2 se definen los valores de las constantes que aparecen en la ecuación.

Parámetro	Valor	Unidad
A _g	0.8	Adimensional
t _{max,i}	55	min
t _{max,g}	40	min
V _g	13.79	L

Tabla 2-2. Parámetros de la ecuación de perturbaciones del Modelo de Bergman

2.1.2 Modificaciones realizadas

Puesto que en estas ecuaciones del Modelo de Bergman no se recoge el efecto del ejercicio físico que realice el paciente como otra perturbación posible a tener en cuenta aparte de la comida, en este trabajo se propone como modificación añadir una nueva variable en la ecuación diferencial de G(t), llamada *Ej(t)*, que refleje dicho comportamiento. Se sabe que el ejercicio físico ayuda a bajar el valor de glucosa en sangre, de forma que actuaría de forma contraria a la comida, que la eleva. Es decir, el ejercicio físico supondría un “plus” a la insulina efectiva

que se está encargando de bajar los valores de glucosa y cuya variable es $X(t)$. Por tanto se define una nueva ecuación para $G(t)$ basada en la ecuación (1) a la que le añado esta variable $Ej(t)$, quedando de la siguiente forma:

$$\frac{dG(t)}{dt} = -P_1 G(t) - (X(t) + Ej(t))(G(t) + G_b) + D(t) \quad (5)$$

La unidad en la que se mide $Ej(t)$ debe coincidir con la de $X(t)$ y es min^{-1} . La ecuación que rige a $Ej(t)$ es la siguiente:

- En caso de estar en reposo o ejercicio suave: $Ej(t) = 0.005 \quad (6)$
- En caso de ejercicio físico moderado o intenso: $Ej(t) = 0.005 * 2$

Es decir, suponemos que por el simple hecho de mantener las funciones vitales el organismo consume un mínimo de glucosa constante (caso de reposo o ejercicio físico suave). Sin embargo, cuando se realiza ejercicio físico más intenso, el organismo consume más glucosa y es por eso que el valor de esta constante se duplica, haciendo que baje más la glucosa.

2.2 Implementación y simulación del Modelo de Bergman

Una vez que se ha elegido el modelo, se pasa a implementar sus ecuaciones diferenciales en Matlab para posteriormente poder realizar una simulación del modelo en Simulink. Las ecuaciones escogidas para la implementación, tal y como he dicho en el apartado anterior, van a ser las ecuaciones (5), (2) y (3) para $G(t)$, $I(t)$ y $X(t)$, respectivamente; la ecuación (4) para $D(t)$; y por último la ecuación (6) para $Ej(t)$.

Puesto que para realizar en los siguientes capítulos el control resulta más fácil simular en unidades de segundos, previamente se pasan todos los parámetros de las ecuaciones del modelo a sus valores correspondientes en segundos (en lugar de minutos). Asimismo, ya que la unidad natural en la que normalmente se expresa la glucemia es mg/dL , se haya el factor de conversión de mmol/L (la unidad en la que se expresa la glucemia en las ecuaciones del modelo) a mg/dL , que es 18 puesto que 1 mol de glucosa ($\text{C}_6\text{H}_{12}\text{O}_6$) pesa 180 g.

El modelo de bloques que se construye en Simulink se puede observar en la figura 2-2. Se constituye de un subsistema llamado “Modelo_Bergman” dentro del cual se implementan las ecuaciones del modelo en sí; un subsistema llamado “Comida”, donde se implementa la ecuación de $D(t)$; un tercer subsistema llamado “Ejercicio”, donde se implementa $Ej(t)$; una constante que corresponde a $U(t)$ (en mU/s) y que será sustituida por el bloque de control en los siguientes capítulos; y por último el factor de conversión mencionado en el párrafo anterior para pasar $G(t)$ de mmol/L a mg/dL antes de mostrarla en gráficas.

2.2.1 Bloque Modelo_Bergman

El subsistema “Modelo_Bergman” es el bloque principal dentro del modelo, ya que dentro es donde se implementan sus ecuaciones. Tal y como podemos ver en la figura 2-3, este bloque recibe 3 entradas desde el exterior que se corresponden a la que será la señal de control $U(t)$ y a las dos perturbaciones que tenemos, $D(t)$ y $Ej(t)$. Dentro del bloque *Interpreted Matlab Fnc* llamamos a la función de Matlab “Modelo_Bergman.m”, en donde se definen los valores de los parámetros y se implementan las ecuaciones diferenciales. A esta función le llega como entrada las 3 entradas del exterior que he mencionado y además $G(t)$, $I(t)$ y $X(t)$. Para obtener estos tres últimos valores se deben integrar los valores de salida de la función, ya que estos valores de salida son las derivadas de las variables con respecto al tiempo.

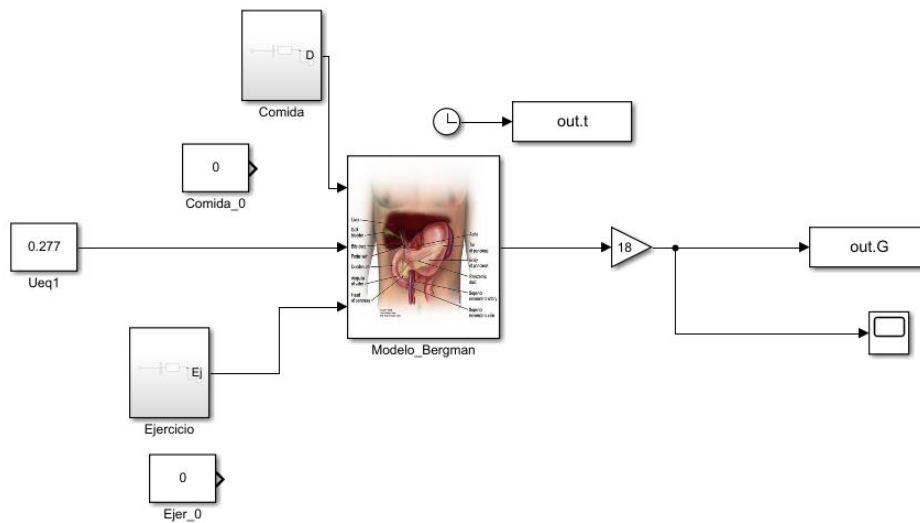


Figura 2-2. Modelo de Bergman en Simulink

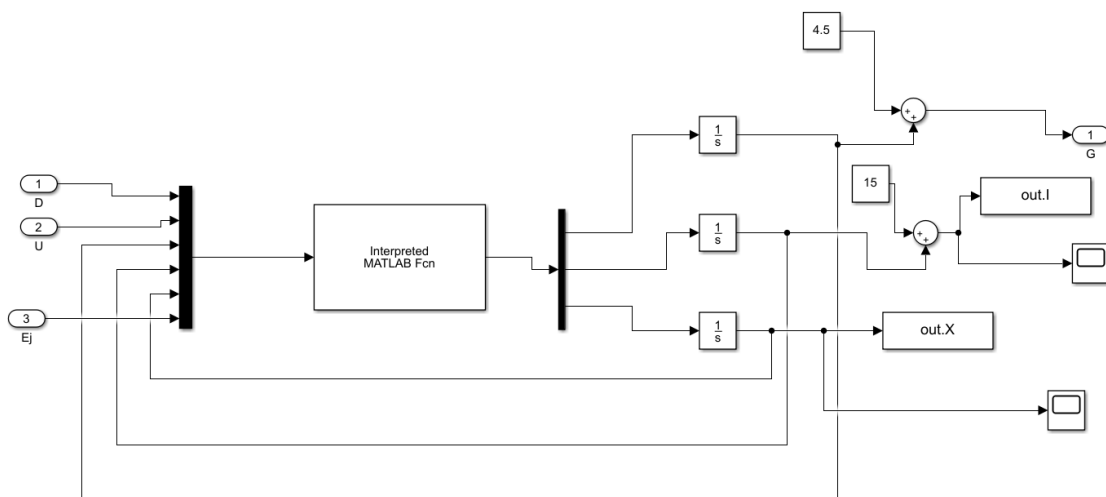


Figura 2-3. Interior del bloque “Modelo_Bergman”

Puesto que las ecuaciones del modelo son incrementales, es decir, que las salidas son incrementos respecto a un cierto valor constante, hay que sumarle estos valores de glucosa e insulina basales a la salida (G_b y I_b definidos anteriormente; X_b es 0).

2.2.2 Bloques de perturbaciones. Protocolo de pruebas

Tanto el subsistema de *Comida* como el de *Ejercicio* tienen por dentro una estructura muy similar: se llama a una función de Matlab que tiene como variable de entrada el tiempo de la simulación. Las funciones son “comida.m” y “ejercicio.m” respectivamente. Dentro de las funciones se implementan las ecuaciones correspondientes.

Se define un protocolo de pruebas que será el usado posteriormente a la hora de experimentar con los distintos controladores que se van a diseñar. Dicho protocolo consiste en las siguientes tablas:

Perturbaciones de Comida	
Hora de la ingesta	Gramos de $C_6H_{12}O_6$
8:00	55
14:00	90
21:00	45

Tabla 2-3. Protocolo de perturbaciones de comida

Perturbaciones de Ejercicio	
Hora	Tipo de ejercicio
0:00-18:00	Reposo/Ejercicio suave
18:00-19:00	Ejercicio intenso
19:00-0:00	Reposo/Ejercicio suave

Tabla 2-4. Protocolo de perturbaciones de ejercicio físico

Dentro de las funciones de Matlab descritas antes se define este protocolo dependiendo del tiempo de simulación. La simulación durará un total de 86400 segundos, los equivalentes de 1 día completo. Se ha elegido este protocolo por ser estándar, con horas habituales de desayuno, almuerzo y cena en un paciente medio que además realiza ejercicio intenso una hora al día. La referencia de glucosa, como se verá en el capítulo siguiente dedicado al control, se pondrá a 90 mg/dL, un valor adecuado y entorno al cual suelen encontrarse las personas no diabéticas.

2.3 Linealización del modelo

Como he dicho en apartados anteriores, el modelo de Bergman es un modelo no lineal. Por lo tanto, es necesario linealizarlo previamente a diseñar el control, y para ello se debe obtener las ecuaciones de estado que definen el modelo, decidir los puntos de equilibrio y obtener las matrices correspondientes a la linealización para poder sacar la función de transferencia del modelo $G(s)$.

Se quiere linealizar el modelo entorno a los siguientes puntos de equilibrio: $G_{eq}=0$ mmol/L, $I_{eq}=0$ mU/L, $X_{eq}=0$ min⁻¹, $D_{eq}=0$, $E_{j_{eq}}=0$, y $U_{eq}=16.667$ mU/min. Se obtienen las siguientes matrices:

$$A = \begin{bmatrix} -P_1 & 0 & -G_b \\ 0 & -n & 0 \\ 0 & P_3 & -P_2 \end{bmatrix} \quad B = \begin{bmatrix} 0 & 1 & -G_b \\ \frac{1}{v_i} & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad C = [1 \quad 0 \quad 0] \quad D = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \quad (7)$$

Por lo tanto, y sabiendo que:

$$\dot{\bar{x}} = A\bar{x} + B\bar{u}$$

$$\bar{y} = C\bar{x} + D\bar{u}$$

Siendo \bar{x} , \bar{u} y \bar{y} los valores de los incrementos que se le suman a los puntos de equilibrio, tendríamos que las ecuaciones del modelo serían las siguientes:

$$\begin{bmatrix} \dot{G} \\ \dot{I} \\ \dot{X} \end{bmatrix} = \begin{bmatrix} -P_1 & 0 & -G_b \\ 0 & -n & 0 \\ 0 & P_3 & -P_2 \end{bmatrix} \begin{bmatrix} G \\ I \\ X \end{bmatrix} + \begin{bmatrix} 0 & 1 & -G_b \\ \frac{1}{V_i} & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} U - U_{eq} \\ D \\ E_j \end{bmatrix} \quad (8)$$

$$y = [1 \quad 0 \quad 0] \begin{bmatrix} G \\ I \\ X \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \begin{bmatrix} U - U_{eq} \\ D \\ E_j \end{bmatrix} \quad (9)$$

Para calcular la función de transferencia del modelo, se tiene la siguiente ecuación:

$$G(s) = C(sI - A)^{-1}B + D \quad (10)$$

Dicha ecuación se calcula en el script de Matlab llamado “función_transferencia.m”, y se obtienen 3 funciones de transferencia: una para la señal de control $u(t)$ y otras dos para cada una de las perturbaciones, D y E_j . En este caso, me quedo con la de la señal de control, que es la siguiente:

$$G_m(s) = \frac{-31.59}{(15000s+7)(2400s+1)(648s+1)} \quad (11)$$

Ahora que ya está calculada la función de transferencia del modelo se puede empezar a diseñar el control en el siguiente capítulo.

3 CONTROL PID

“If you want to make the world a better place, take a look at yourself and then make a change”.

- Michael Jackson -

En este capítulo se desarrollará un control PID básico, primero en tiempo continuo y después en tiempo discreto, al que se le irán añadiendo distintas modificaciones para mejorarlo. Se realizarán diversos experimentos para observar si su comportamiento es el deseado.

3.1 Controlador PID en tiempo continuo

Primero se va a diseñar un controlador PID en tiempo continuo, es decir, con un tiempo de muestreo igual a 0. Este controlador no se puede usar en la realidad, ya que el algoritmo de control se implementa y calcula dentro de una computadora que tiene un cierto tiempo de muestreo, por pequeño que sea, y en donde por lo tanto tendremos un controlador en tiempo discreto. Sin embargo, este primer diseño en tiempo continuo servirá para definir especificaciones, así como la frecuencia de control y distintos parámetros del controlador. También se podrá observar al final si los experimentos con el controlador salen tal y como se esperaban y si es válido para el control de la glucosa.

3.1.1 Especificaciones definidas para el controlador

Se va a diseñar el controlador usando el dominio de las frecuencias. La frecuencia de control (a la que trabajará el controlador) se puede elegir según dos criterios, del cual se escogerá el valor más restrictivo. El primer criterio será el de la frecuencia de Nyquist. Siguiendo dicho criterio se marcará que la frecuencia de control ω_c sea al menos 10 veces superior que la frecuencia de Nyquist. Para calcular la frecuencia de Nyquist se tiene que:

$$\omega_N = \frac{\pi}{T_m} \quad (12)$$

Siendo T_m el tiempo de muestreo. Si bien en tiempo continuo el tiempo de muestreo como he dicho antes es 0, se va a definir un tiempo de muestreo inicial de 1 minuto (60 segundos) en vistas al controlador en tiempo discreto.

Por lo tanto, obtenemos una frecuencia de Nyquist:

$$\omega_N = \frac{\pi}{60} = 0.05 \text{ rad/s} \quad (13)$$

Sabiendo que la frecuencia de control debe ser al menos 10 veces mayor que la frecuencia de Nyquist:

$$\omega_c = 10 * 0.05 = 0.5 \text{ rad/s} \quad (14)$$

El segundo criterio que se va a usar es el de limitación de fase, ya que no es recomendable que el controlador trabaje a frecuencias que se encuentren por encima de la frecuencia que marca los -180° de fase en el bode la función de transferencia del modelo. Para ver cuál es esta frecuencia se dibuja el bode de $G_m(s)$ (de la ecuación (11)), que podemos ver en la figura 3-1.

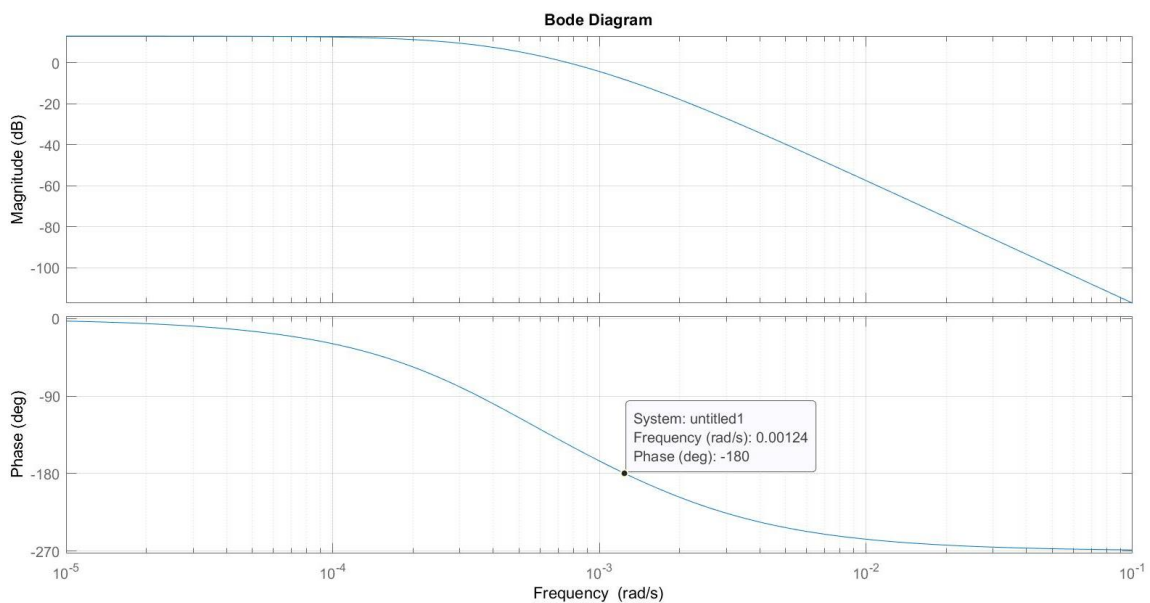


Figura 3-1. Bode de $G_m(s)$

Podemos observar que la frecuencia en la que el sistema tiene una fase de -180° es $\omega_c = 0.00124 \text{ rad/s}$.

Puesto que se debe escoger la frecuencia de control más restrictiva de los dos criterios usados, me quedaré en este caso la frecuencia de control determinada por el criterio de limitación de fase, ya que es la más pequeña de las dos. Por lo tanto, mi frecuencia de control a partir de este punto es:

$$\omega_c = 0.00124 \text{ rad/s} \quad (15)$$

3.1.2 Diseño del controlador PID en tiempo continuo

Se va a diseñar el controlador, en el dominio de la frecuencia tal y como he mencionado antes, optando para el diseño del mismo que sea sin cancelación de dinámica. Esto es debido a que nos enfrentamos a un problema de regulación principalmente, y no de seguimiento. Se va a definir un valor constante de referencia de glucosa y el controlador se encargará de regular la respuesta del sistema de forma que ante las distintas perturbaciones que sufra bien por comida o bien por ejercicio esta se mantenga en el valor de referencia. Por lo tanto, el principal problema serán las perturbaciones y no los cambios a lo largo del tiempo de valores de referencia que el controlador deba ir siguiendo (problema de seguimiento). Puesto que los controladores diseñados con

cancelación de dinámica (cancelando los polos de la función de transferencia del modelo) trabajan peor ante las perturbaciones, se elige el diseño sin cancelación de dinámica, para obtener mejores resultados.

El margen de fase actual que tenemos se calcula como:

$$M_f^{act} = 180 + (-180) = 0^\circ \quad (16)$$

El margen de fase deseado se fija en 75° , para que posteriormente la fase del $G_m(s)*C(s)$ sea aproximadamente -105° a la frecuencia de control. De esta forma, la fase que debe aportar el PID se calcula de la siguiente forma:

$$\varphi = M_f^{des} - M_f^{act} = 75 - 0 = 75^\circ \quad (17)$$

Se calcula la función de transferencia del controlador tipo PID:

$$C(s) = K_p \frac{(\tau_1 s + 1)^2}{\tau_1 s} \quad (18)$$

Para obtener el valor de τ_1 se tiene la siguiente ecuación:

$$\varphi = 2 * \arctg(\tau_1 * \omega_c) - 90 \quad (19)$$

De donde se despeja y se obtiene que:

$$\tau_1 = 6028.4 \text{ s} \quad (20)$$

Sustituyendo en la ecuación (18) con el valor obtenido tenemos que la función de transferencia del controlador es:

$$C(s) = K_p \frac{(6028.4s+1)^2}{6028.4s} \quad (21)$$

Se añade a esta función de transferencia un polo de alta frecuencia, que hace que la función del controlador sea bipropia y además ayuda a que la respuesta sea menos ruidosa. Para calcular este polo de alta frecuencia se tiene que:

$$\frac{1}{\tau_{af}} \cong 10 * \omega_c = 0.012 \text{ rad/s} \quad (22)$$

$$\tau_{af} = 83.33 \text{ s} \quad (23)$$

Con el polo de alta frecuencia añadido $C(s)$ queda de la forma:

$$C(s) = K_p \frac{(6028.4s+1)^2}{6028.4s} \frac{1}{83.33s+1} \quad (24)$$

Por tanto, solo faltaría hallar el valor de K_p para tener definida la función. Dicho valor se obtiene ajustando el

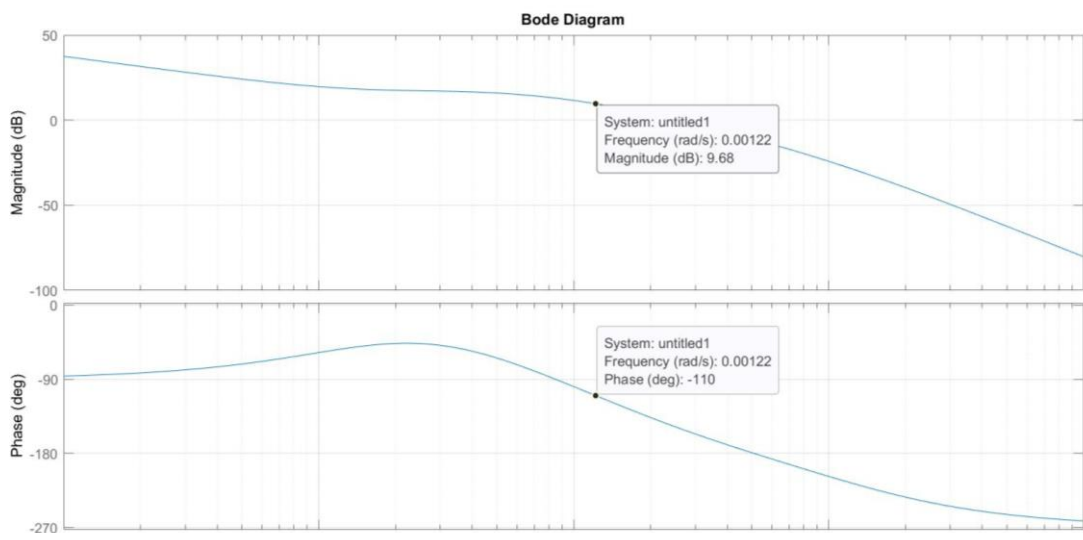


Figura 3-2. Bode de $G_m(s)*C(s)$ sin ajuste de K_p

bode para que la ganancia En la figura 3-2 se representa el bode resultante de multiplicar $G_m(s)*C(s)$, es decir, la función de transferencia del modelo por la del controlador:

Se observa que la fase a la frecuencia de control es -110° , 5 grados por debajo de lo que se esperaba según los cálculos de la ecuación (17). Estos 5 grados de pérdida de fase se deben al polo de alta frecuencia añadido, pero no afectan a la estabilidad del sistema en ningún caso y considero aceptable el resultado. Mirando el diagrama de ganancia, se ajusta la K_p de $C(s)$ de forma que la ganancia sea 0 a la frecuencia de control, obteniendo un valor de $K_p=-0.3385$. Por lo tanto y finalmente se tiene que:

$$C(s) = -0.3385 \frac{(6028.4s+1)^2}{6028.4s} \frac{1}{83.33s+1} \quad (25)$$

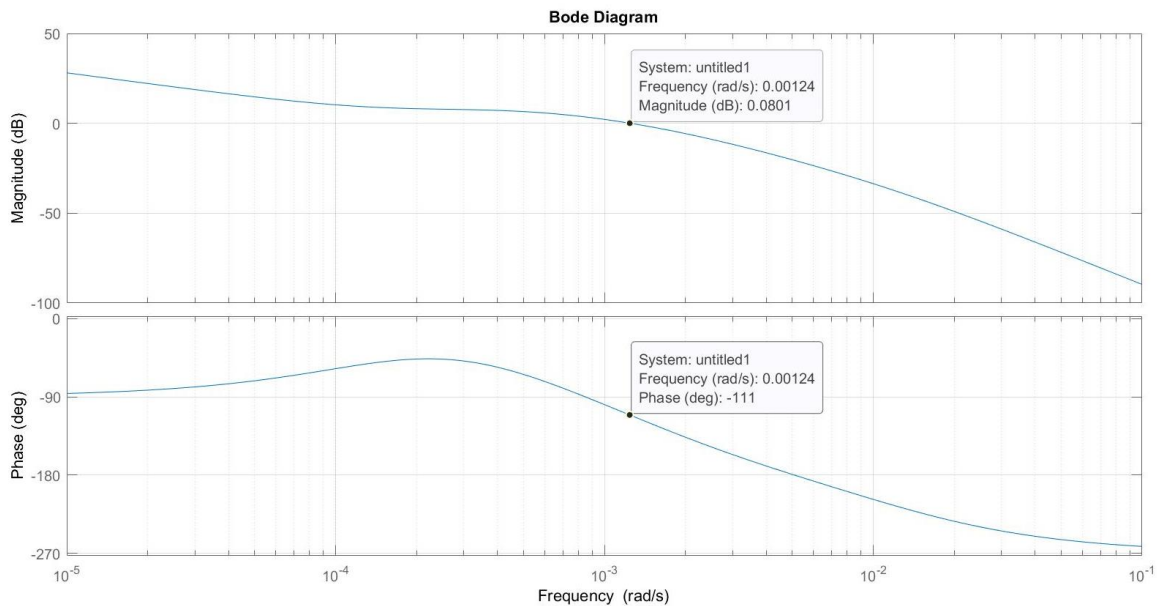


Figura 3-3. Bode de $G_m(s)*C(s)$ final

3.1.3 Implementación del controlador PID en tiempo continuo en Simulink

Una vez calculada $C(s)$, se implementa el controlador en Simulink. Para ello se sustituye el bloque constante para $U(t)$ que vemos en la figura 2-2 por un bloque de función de transferencia con la función de $C(s)$ calculada. A este bloque le llega el error que resulta de restar la referencia de glucosa con la glucosa real, que es la salida del sistema. Puesto que al hablar de función de transferencia hablamos necesariamente de incrementos, la salida del controlador es un incremento al que habrá que sumarle el punto de equilibrio, que en este caso es U_{eq} definido en el capítulo 2.3, donde se linealizaba el modelo. Ya que en Simulink estoy trabajando en segundos, tenemos que U_{eq} es igual a 0.277 mU/s. Se aplica a la señal de control un bloque de saturación, tanto por debajo ya que no se pueden inyectar valores negativos de insulina (sería el equivalente a sacar insulina del cuerpo del paciente, y obviamente esto no se puede hacer en la realidad), como por arriba, para no inyectar demasiada insulina. Para el valor límite superior usamos el escogido también por F. Barragán *et al.* [2] y P. Dua *et al.* [5], que es 100 mU/min. El rango de valores de 0 a 100 mU/min se considera adecuado para la administración de insulina usando bombas, ya que hay que tener en cuenta también el diámetro del catéter por el que se administra, el cual es de un tamaño reducido para ser lo menos invasivo posible. La referencia de glucosa será 90 mg/dL, y se mantendrá para los siguientes experimentos. En la figura 3-4 se puede observar este modelo de control PID continuo en Simulink.

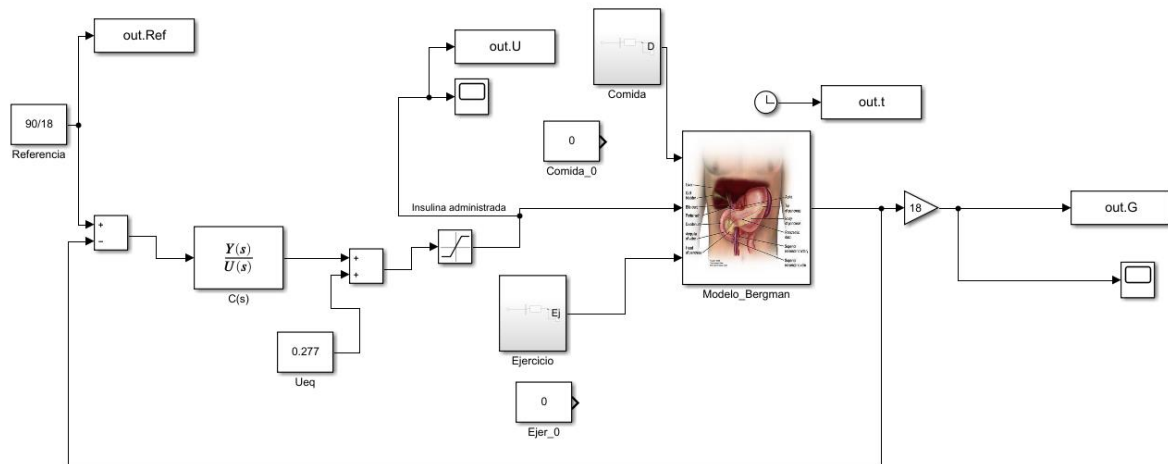


Figura 3-4. Modelo con controlador PID continuo en Simulink

3.1.4 Pruebas con el controlador PID en tiempo continuo

Antes de mostrar las gráficas obtenidas de la simulación del control PID en tiempo continuo, enseñaré primero las gráficas correspondientes a la simulación del modelo de la figura 2-2, donde no hay ningún tipo de control aplicado, solo el propio comportamiento del sistema en sí. Se realizan dos pruebas: sin perturbaciones y con las perturbaciones de comida y ejercicio. Esta primera prueba servirá para comparar con las posteriores en donde se aplica el control y comprobar que realmente este control mejora la respuesta del sistema según las especificaciones.

Aunque el tiempo de simulación en Simulink sea en segundos, en todas las gráficas de pruebas, tanto en este capítulo como en los siguientes, el tiempo en el eje X se muestra en minutos, que resulta una escala mucho más adecuada a la hora de observar los resultados.

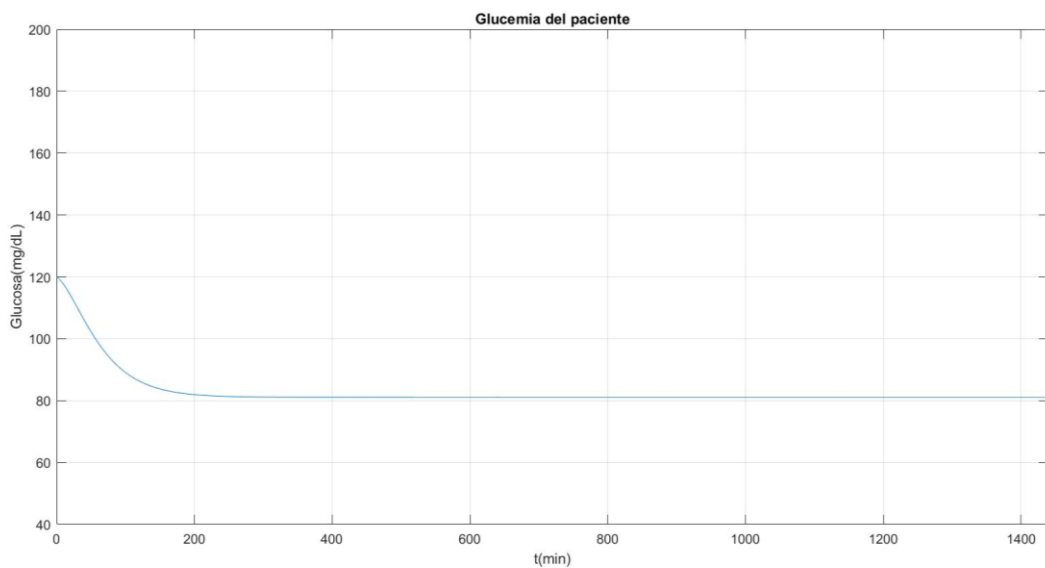


Figura 3-5. Comportamiento del modelo de Bergman sin perturbaciones

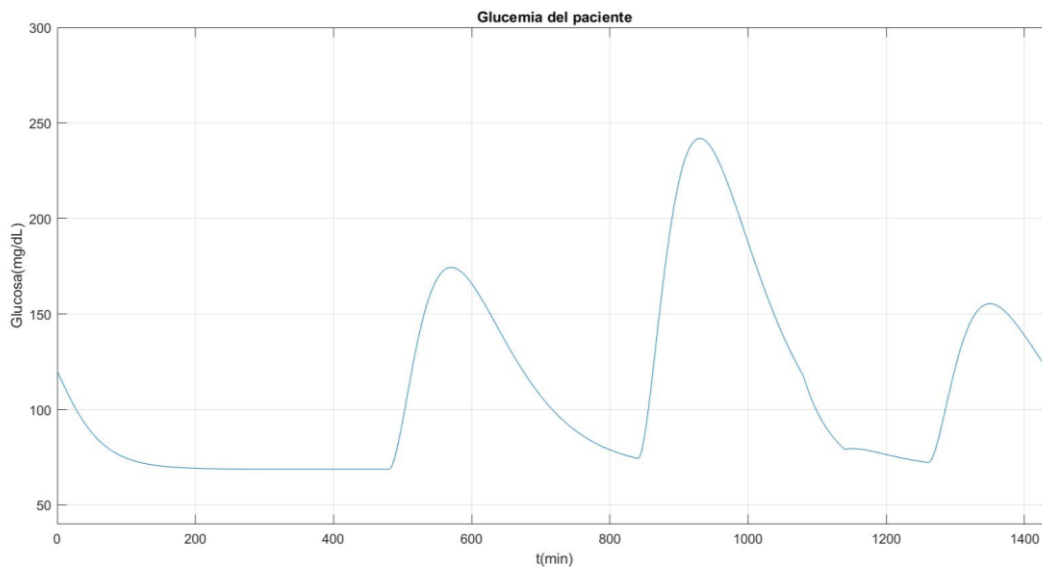


Figura 3-6. Comportamiento del modelo de Bergman con perturbaciones

En la figura 3-5 se observa el comportamiento del modelo de Bergman sin aplicar perturbaciones, es decir, con $D(t) = 0$ y $E_j = 0$. La señal de control $U(t)$ se aplica constante y es igual al valor de equilibrio, es decir, 0.277 mU/s. Si bien las simulaciones son pocos útiles ya que no son realistas (en la vida real el paciente tendrá distintos tipos de perturbaciones a lo largo del día, y es muy raro que no tenga alguna en algún momento), se realizan para comprobar posteriormente con el control que la salida alcanza rápidamente el valor de referencia y se mantiene constante alrededor de él, tal y como se espera. En este caso, el sistema definido por las ecuaciones del modelo de Bergman parte de un valor de glucosa cercano a 120 mg/dL y pasados unos 200 minutos alcanza un valor permanente de 80 mg/dL, que es un valor aceptable de glucosa. Esta curva estaría muy bien y no necesitaría control si en la vida real no hubiera perturbaciones, pero como he dicho antes, en la realidad el paciente tendrá constantemente perturbaciones de comida, ejercicio y demás.

Por su parte en la figura 3-6 vemos el comportamiento del modelo ahora sí con las perturbaciones, por lo tanto es mucho más realista. El tiempo de las perturbaciones, como en todas las gráficas siguientes, viene dado por el protocolo de pruebas definido en el capítulo 2.2.2. Se observa como al principio, cuando todavía no hay perturbación alguna, la glucosa tiende a irse a 80 mg/dL como en la gráfica anterior. En cuanto empieza a haber perturbaciones de comida, la glucosa tiende a subir mucho, alcanzando valores máximos de casi 250 mg/dL, que están muy por encima de los valores deseados. Por ejemplo, en el almuerzo, la glucosa no vuelve al valor deseado (entorno a 100 mg/dL) hasta pasados un poco más de 200 minutos, es decir, más de 3 horas. El paciente estaría todo ese periodo de tiempo a niveles de glucosa por encima de lo deseado, lo cual puede tener consecuencias negativas en su salud si esto se repite día tras día. Esta gráfica correspondería a una bomba de insulina trabajando en bucle abierto, con un valor constante de insulina administrándose durante todo el tiempo. Es más que evidente que no resulta muy efectiva, ya que si bien a lo largo de las horas hace que la glucosa acabe bajando, ese tiempo es demasiado grande y además los valores máximos de glucosa que se alcanzan son demasiado alejados de los valores deseados.

A continuación se realiza la simulación correspondiente al modelo de la figura 3-4, el del control PID continuo. En las figuras se mostrarán dos gráficas fundamentales en las pruebas. Por una parte, arriba está la correspondiente a la glucosa del paciente, donde se muestra la glucosa real (la salida $G(t)$ del sistema), la glucosa de referencia, y los valores límite aceptables en una persona con diabetes tipo 1, que serán 60 mg/dL por abajo (en amarillo, marcando la hipoglucemia) y 180 mg/dL por arriba (en morado, marcando la hiperglucemia), que ayudarán a ver con facilidad si la salida del sistema está dentro de ese rango. Por otra parte, abajo se encuentra la gráfica de la señal de control, es decir, la insulina que se administra y que ha sido calculada previamente por

el algoritmo de control. Para la obtención de las mismas se usa el script “graficas.m”.

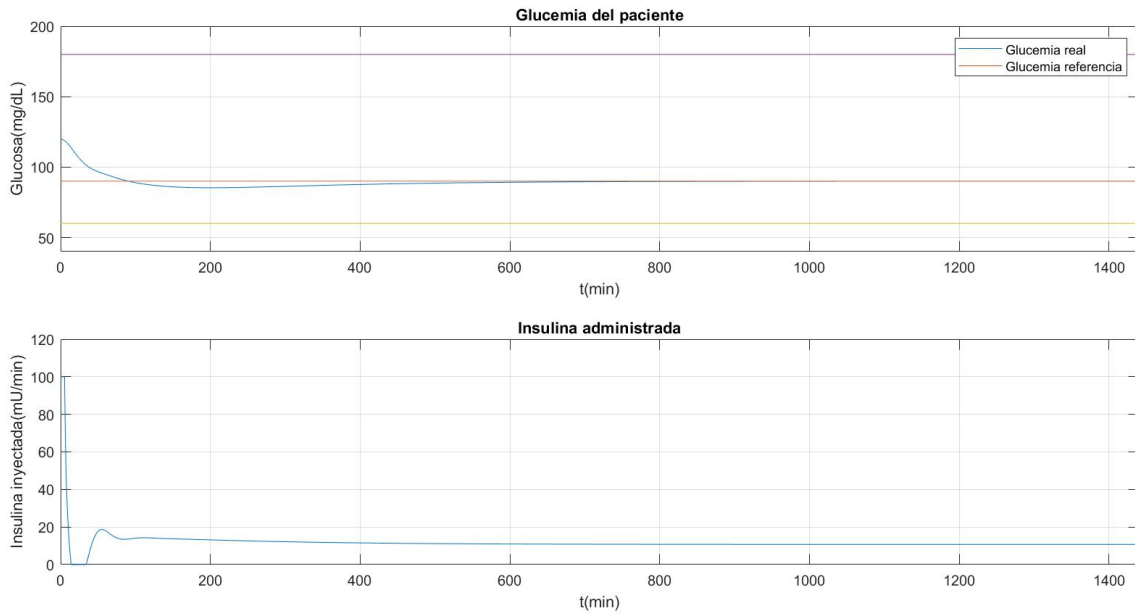


Figura 3-7. Control PID continuo sin perturbaciones

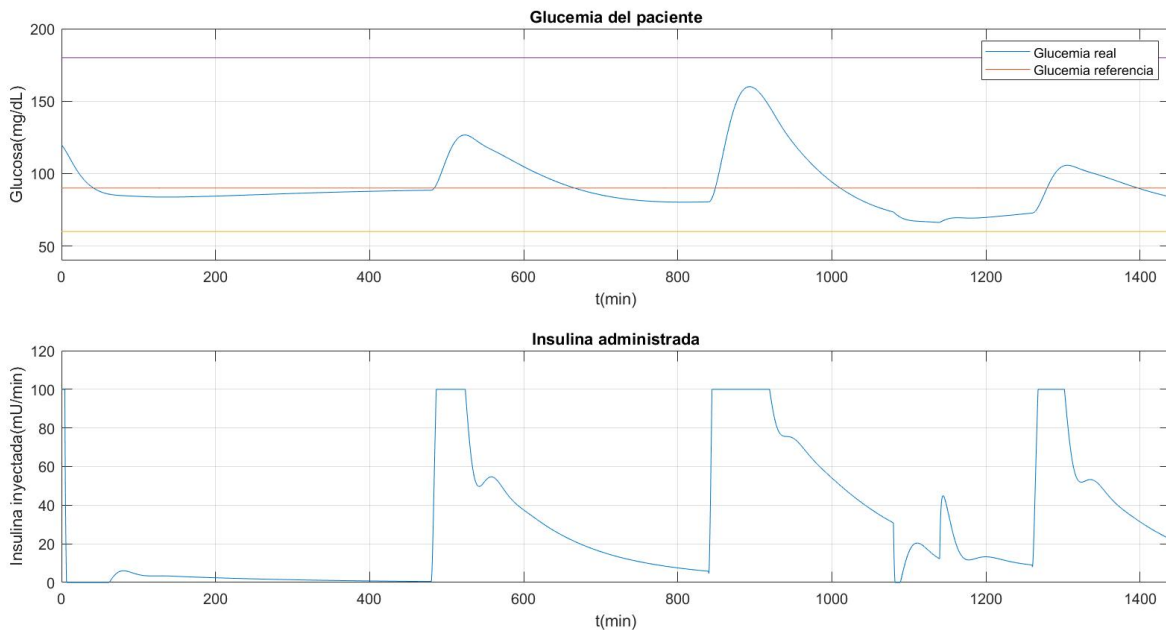


Figura 3-8. Control PID continuo con perturbaciones

En la figura 3-7 se puede observar cómo es la respuesta del sistema sin perturbaciones, y se comprueba que ésta es capaz de seguir la referencia de 90 mg/dL sin ningún problema, y para ello se administra un valor constante de insulina de alrededor de 10 mU/min una vez alcanzado el valor de referencia de glucosa para mantenerla.

En la figura 3-8 se ve cómo el sistema responde cuando sí hay perturbaciones, y podemos observar a simple vista una mejora importante con respecto a la figura 3-6. Para empezar, cuando se aplica el control, la glucosa del paciente mantiene unos valores aceptables durante toda la simulación, siendo el valor máximo que se alcanza 160 mg/dL después del almuerzo (que es la comida en donde más hidratos de carbono ingiere). Esto sin control no ocurría, y se alcanzaban casi los 250 mg/dL, es decir, se ha conseguido bajar hasta 100 mg/dL la glucosa del

paciente. La curva es mucho más suave, y aunque tiene los picos propios de las perturbaciones de comida, estos no son tan pronunciados y se consigue volver al valor de referencia en menos tiempo gracias a la acción del controlador. En la gráfica de la señal de control se observa que esta satura dando el máximo valor cuando hay las perturbaciones de comida, y posteriormente baja cuando esta deja de hacer efecto. Por otra parte, se ve que la perturbación del ejercicio hace que baje ligeramente más la glucosa, pero a su vez la insulina también baja, e incluso llega a 0, para evitar que se produzcan hipoglucemias. Por tanto, se comprueba tal y como se esperaba que empleando un control tipo PID y haciendo que la bomba de insulina trabaje en bucle cerrado, la glucosa del paciente se consigue mantener durante mucho más tiempo dentro de un rango de valores aceptables, y que cuando ésta suba debido a la ingesta de comida, se corrija rápidamente haciendo que vuelva a su valor de referencia.

3.2 Controlador PID en tiempo discreto

En este apartado se diseñará el controlador PID en tiempo discreto para que el algoritmo de control se pueda implementar posteriormente en una computadora o microprocesador que lleve la bomba de insulina, y se realizarán pruebas como se ha hecho con el controlador en tiempo continuo para comprobar que los resultados son válidos.

3.2.1 Diseño del controlador PID en tiempo discreto

Partimos de las mismas especificaciones del apartado anterior en tiempo continuo, es decir, con la misma frecuencia de control (de la ecuación (15)), y los mismos parámetros K_p y τ_1 (de las ecuaciones (25) y (20) respectivamente). El algoritmo de control se implementa en el fichero de Matlab llamado “control.m”. En él se define el tiempo de muestreo, que empezará siendo 1 minuto y posteriormente se realizarán pruebas para ver si se puede relajar y hacerlo mayor sin que afecte a la respuesta del sistema. Se define también los parámetros T_d y T_i , a partir de los cuales se calcularán las constantes derivativa e integral del controlador (la proporcional ya la tenemos y es K_p). Se calculan estos parámetros como:

$$T_i = 2 * \tau_1 = 12057 \text{ s} \quad (26)$$

$$T_d = \frac{\tau_1^2}{2 * \tau_1} = 3014.2 \text{ s} \quad (27)$$

Una vez tenemos estos parámetros podemos calcular K_d y K_i :

$$K_d = K_p * T_d = -1020.3 \quad (28)$$

$$K_i = \frac{K_p}{T_i} = -2.807 * 10^{-5} \quad (29)$$

La ecuación en la que se calcula el incremento de la señal de control es la correspondiente a un controlador PID:

$$u(t) = K_p * errG(t) + K_i * InterrG(t) + \frac{K_d}{T_m} * (errG(t) - errantG(t)) \quad (30)$$

Siendo $errG(t)$ el error de la glucosa que se define como la diferencia entre el valor de referencia de glucosa y el real; $InterrG(t)$ el error integral de la glucosa, calculado como el propio error integral más el error multiplicado por el tiempo de muestreo; y por último $errantG(t)$ el error del instante anterior, que se va actualizando en cada iteración.

Puesto que $u(t)$ es un valor incremental, hay que sumarle el valor del punto de equilibrio, 0.277 mU/s, al igual que se hacía en el controlador en tiempo continuo para obtener $U(t)$, que será la salida de la función. Por último se implementan las líneas correspondientes al saturador, para que los valores de la insulina administrada se

encuentren entre 0 y 100 mU/min.

3.2.2 Implementación del controlador PID en tiempo discreto en Simulink

Para implementar el controlador en Simulink, ahora se tiene el modelo que se puede observar en la figura 3-9. Se crea un subsistema llamado “Control”, que será el bloque correspondiente que se implementaría dentro de la computadora o microprocesador y al que le llega tanto la referencia de glucosa como el valor medido real de glucosa. Por su parte, la salida será la señal de control directa, es decir, la insulina exógena.

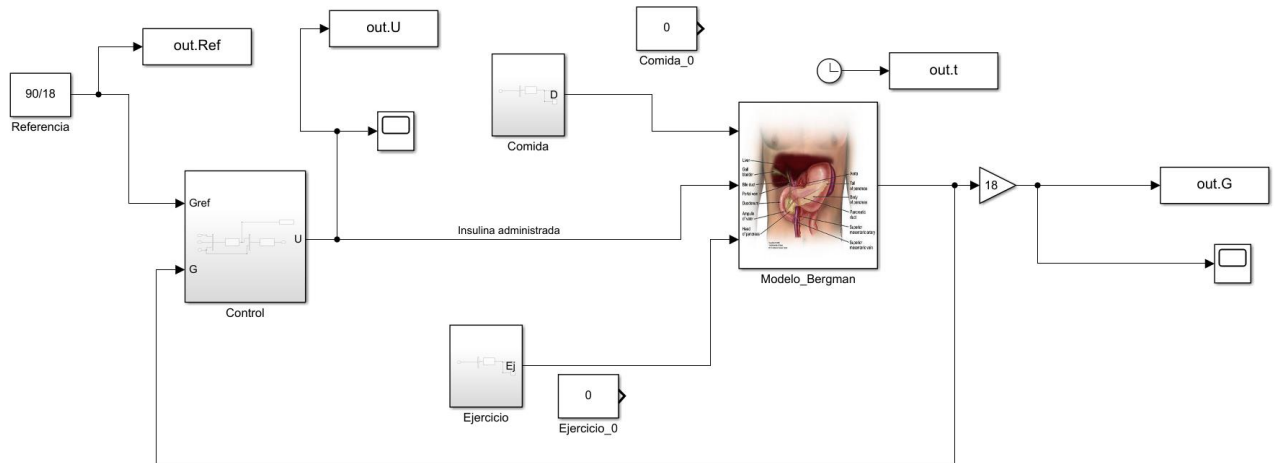


Figura 3-9. Modelo con controlador PID discreto en Simulink

Dentro del bloque de control tenemos un bloque que llama a la función de Matlab descrita en el apartado anterior, a la que le llega, aparte de las dos entradas del propio subsistema, el tiempo de simulación, necesario en el controlador para que el error integral sea 0 al principio de la simulación y hasta que pasen unos milisegundos. Podemos ver el interior del bloque en la siguiente figura 3-10:

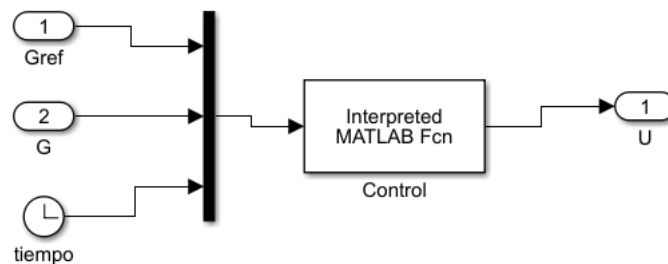


Figura 3-10. Bloque del controlador PID en tiempo discreto en Simulink

3.2.3 Pruebas con el controlador PID en tiempo discreto

Se realiza la simulación correspondiente al modelo de la figura 3-9, con el mismo protocolo de perturbaciones, referencia, tiempo de simulación y estilo de gráficas para poder comparar equitativamente. Se realizan dos simulaciones de nuevo: primero sin perturbaciones, y luego añadiendo estas.

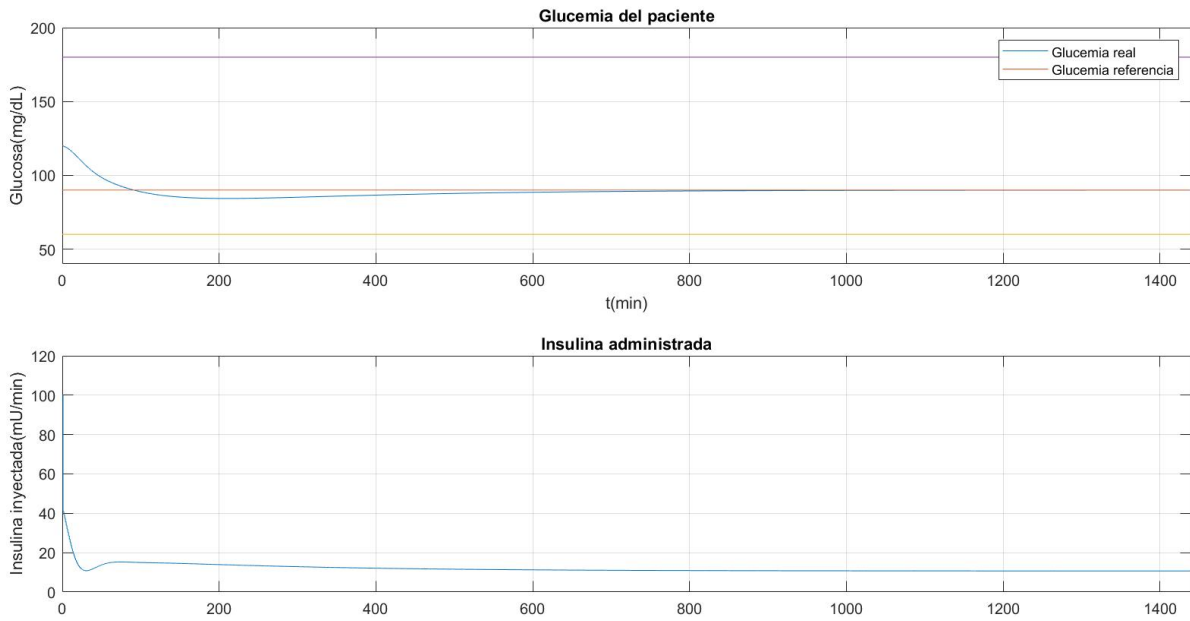


Figura 3-11. PID discreto sin perturbaciones

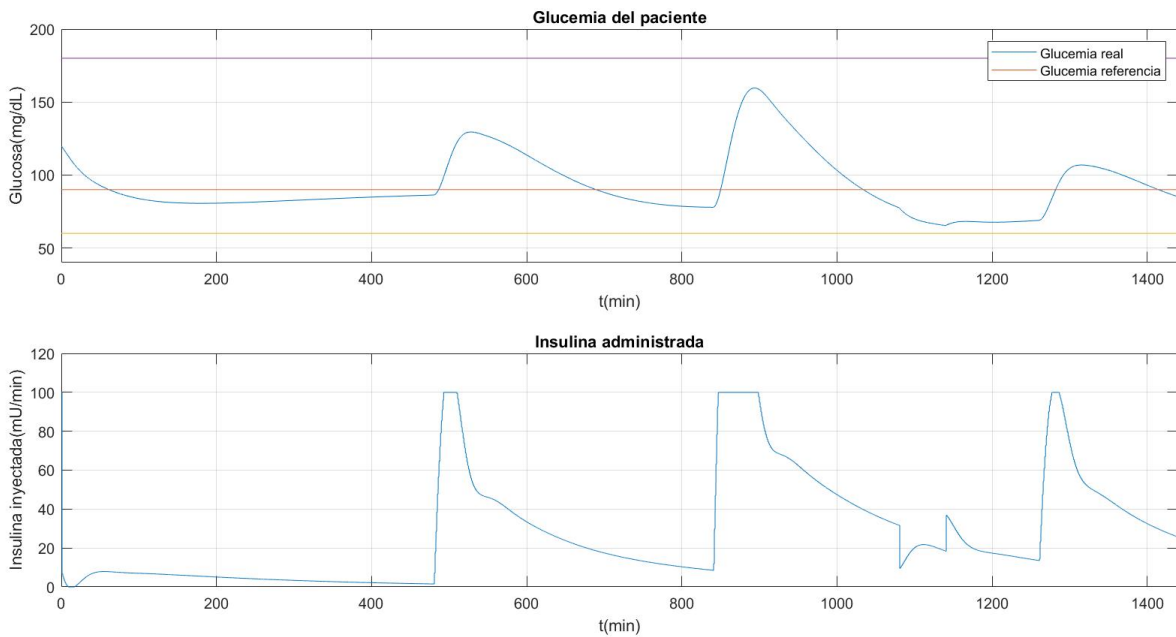


Figura 3-12. PID discreto con perturbaciones

Vemos que los resultados que se obtienen son prácticamente idénticos a los obtenidos con el PID continuo, tanto para el caso con perturbación como en el de sin perturbación, lo cual era de esperar ya que los parámetros de este controlador en discreto han sido calculados a partir de los calculados para el continuo. El único cambio levemente apreciable es en la señal de control, que si nos fijamos bien ahora ya no es continua, sino a pequeños escalones, producto del muestreo y el tiempo discreto. En cualquier caso, vemos un control más real y sobre todo, factible de realizar ya que este control sí se puede implementar en una computadora (no como el control en tiempo continuo que era “ideal”). La glucosa sigue estando todo el tiempo dentro del rango de valores aceptables, y las perturbaciones se corrigen en un tiempo adecuado para que la salida se mantenga el máximo tiempo posible entorno a la referencia. Por último, se observa que el tiempo de muestreo de 1 minuto se ha aplicado no es para nada muy restrictivo, y más adelante se estudiará si este se puede relajar con el fin de trabajar de forma más eficiente, sin necesidad estar administrando cada minuto insulina y tomando muestras.

3.3 Modificaciones y mejoras realizadas al controlador PID en tiempo discreto

Una vez tenemos el controlador PID en tiempo discreto básico se plantean una serie de modificaciones para hacer un controlador más real y con una mejor respuesta para nuestro sistema en concreto. Tales modificaciones serán añadirle un algoritmo *anti-windup*, relajar el tiempo de muestreo y diseñar un control “por lotes” para que la insulina se administre cada cierto tiempo como lo haría en la realidad. Se harán pruebas, ahora ya únicamente con el caso del modelo con perturbaciones de comida y ejercicio puesto que será el que luego se tendrá en la realidad, y se estudiarán los resultados para poder sacar unas conclusiones posteriores.

3.3.1 Controlador PID discreto con algoritmo anti-windup

El efecto *windup* se produce en el PID cuando la señal de control satura. Aunque esta no puede aumentar o disminuir el valor puesto que se fija al máximo o mínimo, el error integral sí que sigue aumentando durante todo el tiempo de la saturación y esto provoca grandes picos en la señal de control y, por tanto, en la salida del sistema, cuando la señal deja de saturar, porque se aplica de golpe todo el error que se llevaba acumulando pero que no se estaba corrigiendo. Como es obvio, este efecto tan común en los PIDs no es deseable porque puede provocar desestabilización y una sobreoscilación mayor en la respuesta, y particularmente en nuestro caso, picos peligrosos de los valores de glucosa que causen hipoglucemias y afecten al paciente. Es por eso que se añade este algoritmo al control PID para evitar que se produzca este efecto. El algoritmo consiste en cambiar la línea que actualiza el valor del error integral en cada iteración, que se define como el error integral anterior más el error de la glucosa multiplicado por el tiempo de muestreo. De esta forma, con el algoritmo *anti-windup* se añade la línea en la que se define el error integral en el caso de que la señal de control sature como el error integral anterior menos (se cambia el signo para que no se produzca el efecto de acumulación de error) el error de la glucosa por el tiempo de muestreo.

A continuación se realiza la simulación de nuevo para observar el resultado de haber añadido el algoritmo *anti-windup*.

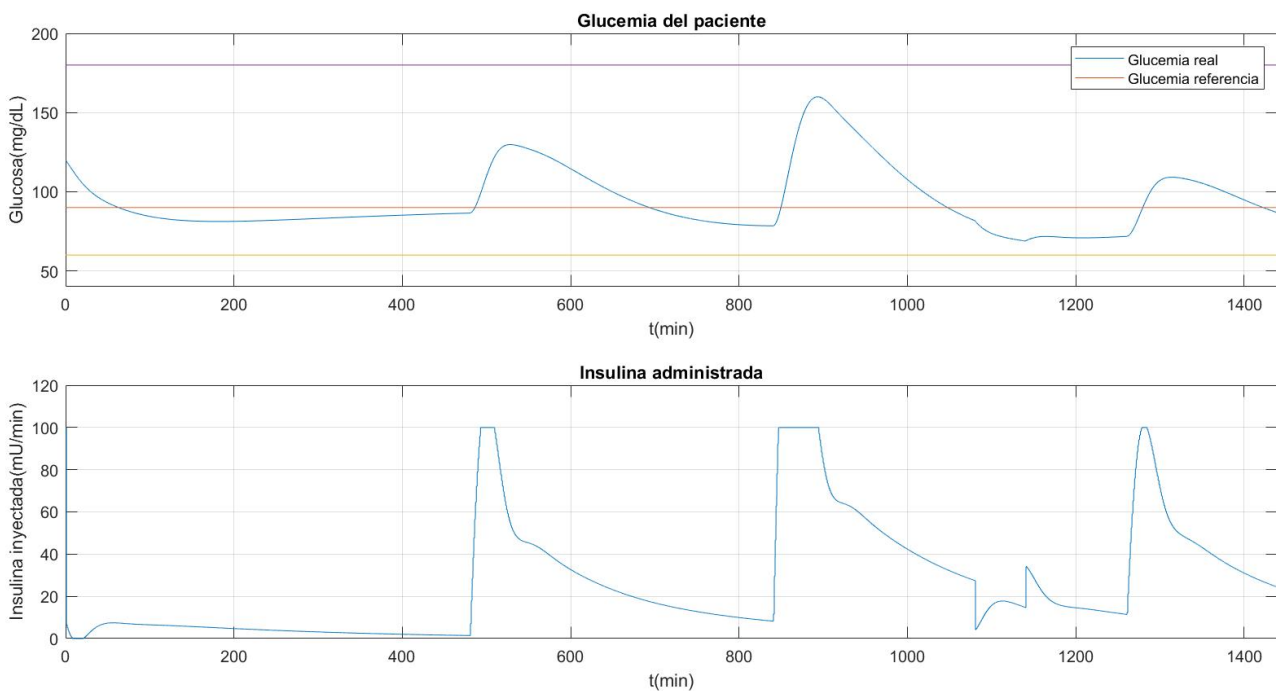


Figura 3-13. PID con algoritmo anti-windup

Comparando la figura anterior con la figura 3-12 podemos observar que ahora los valores mínimos de glucosa que se alcanzan después de haber corregido las perturbaciones con la insulina son ligeramente más altos. Tal y como se esperaba, con el algoritmo *anti-windup* se ha conseguido suavizar estos picos al no ir acumulando el error mientras la señal de control estaba saturando. Esto es especialmente notable después de la perturbación correspondiente al almuerzo, sobre el minuto 1150 (que equivaldría a las 19:15 horas), ya que es la mayor perturbación de comida (y por tanto donde más saturará la señal de control) y a la vez coincide con el ejercicio físico que hace bajar ligeramente la glucosa. En la figura 3-12, con el efecto *windup*, la glucosa llegaba a alcanzar valores de 65 mg/dL, que están muy cercanos a la hipoglucemia. En cambio en la figura 3-13, ya corrigiendo este efecto, este pico se iba a poco menos de 69 mg/dL, un valor algo más adecuado. Aunque aparentemente no es mucha la diferencia en valor numérico, sí que podemos garantizar que con este algoritmo mejoramos la respuesta del sistema y lo más importante: podemos evitar posibles accidentes a los pacientes causados por este tipo de efectos, sin que ello suponga un mayor coste computacional. Por ello, a partir de este apartado todo el control PID que se realice será con el algoritmo *anti-windup* implementado.

3.3.2 Relajación del tiempo de muestreo en el PID

El tiempo de muestreo se define, en este caso, como el intervalo de tiempo que pasa entre que el sensor toma una muestra del valor de glucosa en sangre y el siguiente. Puesto que la toma de la muestra y el cálculo por parte del algoritmo de control de la nueva señal (insulina) a aplicar es prácticamente inmediato, el tiempo de muestreo también define el tiempo que pasa entre que se calcula un nuevo valor para la señal de control y el siguiente. Es decir, cada cuánto tiempo se va actualizando el valor de insulina a administrar. Hasta ahora se empleado un tiempo de muestreo de 1 minuto en todos los controladores diseñados en tiempo discreto, y se ha obtenido una respuesta muy buena. Esto quiere decir que cada minuto se leía un valor nuevo y se calculaba el valor adecuado de insulina. Es fácil ver que cuanto más pequeño sea el tiempo de muestreo, más parecido será el control a un control en tiempo continuo y más exacto será. No obstante, un tiempo de muestreo pequeño también conlleva un mayor coste computacional, ya que se están tomando muestras y calculando el algoritmo de control constantemente.

En este caso específico del control de la glucosa, esta no sufre apenas cambios de valor en un periodo de tiempo de un minuto como para exigir que el tiempo de muestreo sea más pequeño, y por otra parte, tampoco es necesario estar administrando insulina en tan corto periodo de tiempo. Es por esto que se plantea relajar el tiempo de muestreo del sistema y hacerlo algo mayor, con el fin de hacerlo más realista y reducirle el coste computacional, además de evitar estar inyectando al paciente insulina cada minuto.

Se realizan las simulaciones para un tiempo de muestreo de 3 minutos (180 segundos), 5 minutos (300 segundos) y 10 minutos (600 segundos), y posteriormente se comparan con el resultado de la figura 3-13.

En la figura 3-14 podemos ver la respuesta del sistema antes un tiempo de muestreo de 3 minutos. Si bien en la gráfica de la señal de control podemos observar ahora más notablemente los escalones que esta va describiendo debido al aumento del tiempo de muestreo, ya que ahora tarda más en calcular un nuevo valor, si nos fijamos en la gráfica que muestra la glucosa en sangre del paciente vemos que la respuesta es prácticamente igual a la de la figura 3-13, con tiempo de muestreo de 1 minuto. Es decir, se puede aumentar a 3 minutos el tiempo de muestreo sin que esto afecte al sistema.

Se prueba ahora con un tiempo de muestreo de 5 minutos, tal y como se muestra en la figura 3-15. De nuevo podemos observar que los escalones de la señal correspondiente a la insulina administrada están ligeramente más pronunciados que en el caso de 3 minutos, y ahora si se amplía la gráfica de la glucosa podemos observar que el pico correspondiente a la subida de glucosa provocada por el almuerzo es ligeramente más alto, pero no más allá de 2 mg/dL. En cualquier caso, este valor es de 165 mg/dL, un valor totalmente normal después de la ingesta de comida en una persona diabética, y que se mantiene dentro del rango, por lo tanto no supone un problema.

Por último se realiza la misma prueba con un tiempo de muestreo de 10 minutos, cuyo resultado se puede ver en la figura 3-16. En este caso, tal y como se esperaba debido al aumento del tiempo de muestreo (ahora el doble que en el caso anterior), sí que podemos observar que los valores que se alcanzan en los picos de glucosa son más elevados, alcanzando los casi 171 mg/dL, muy cerca de superar el límite superior del rango de valores aceptables. Asimismo, la forma de los picos es más pronunciada, no como en los otros casos, sobre todo aquellos

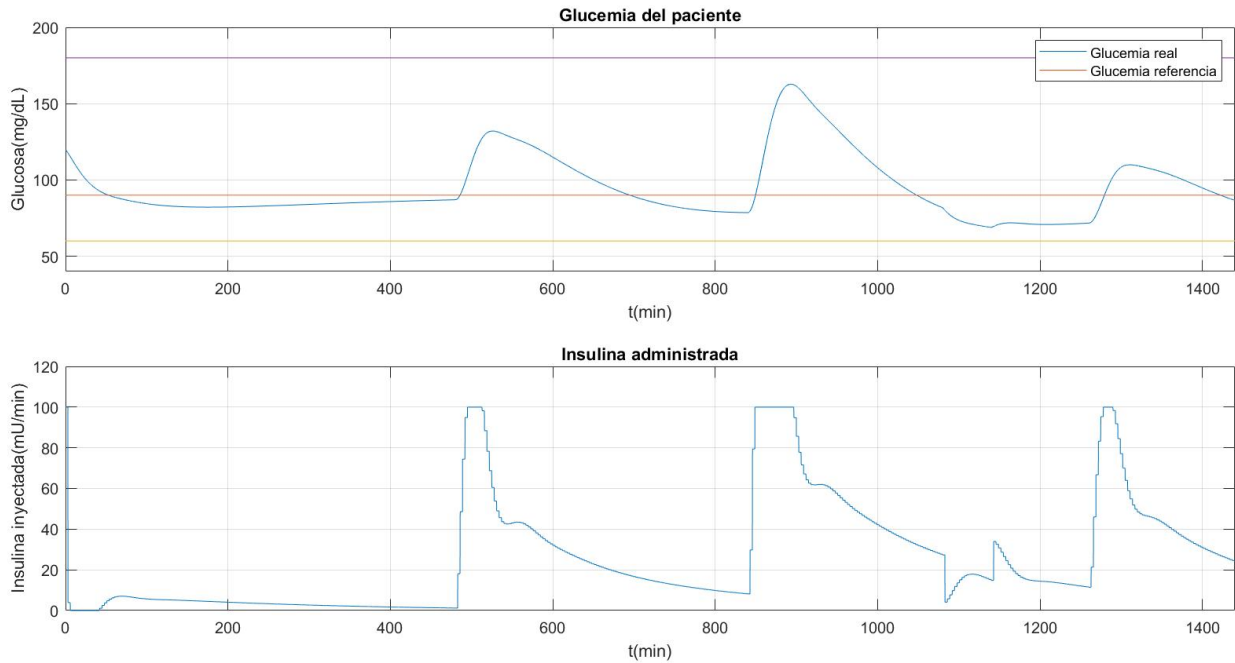


Figura 3-14. PID con tiempo de muestreo de 3 minutos

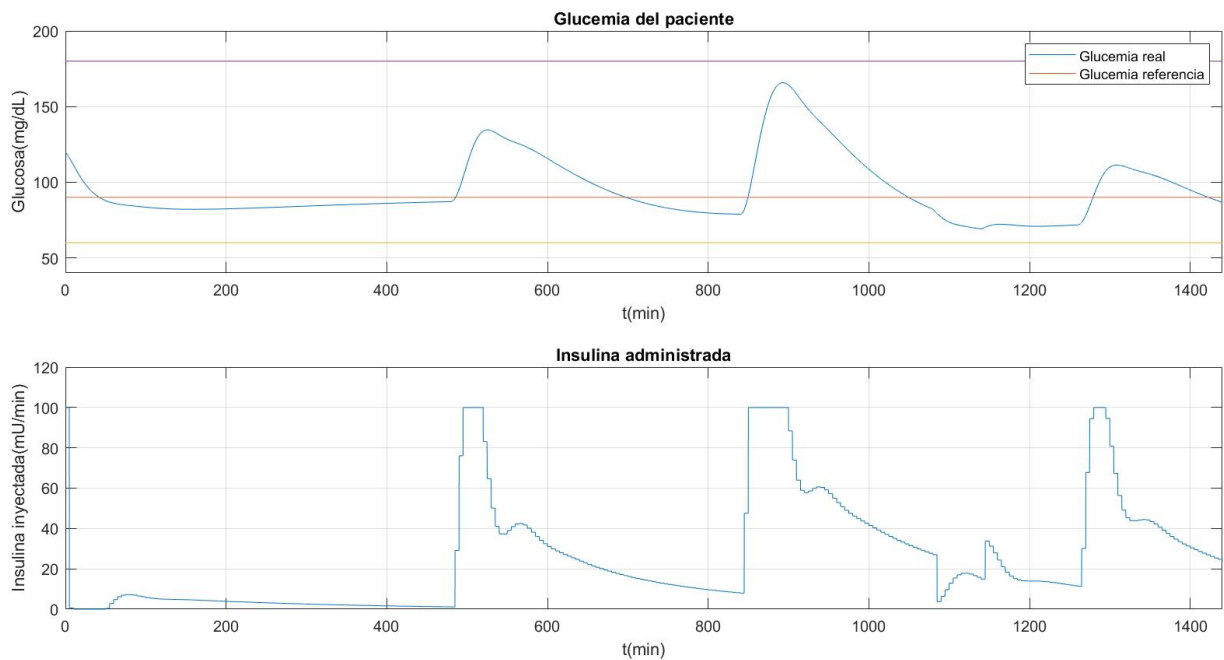


Figura 3-15. PID con tiempo de muestreo de 5 minutos

con el tiempo de muestreo más pequeño, que eran unos picos más redondeados. Por lo tanto, y porque en 10 minutos sí que los valores de glucosa pueden cambiar significativamente, se decide que 10 minutos es un tiempo de muestreo demasiado grande y que puede conllevar problemas a la hora de un control exacto de la glucosa.

Después de estas pruebas con diferentes tiempos de muestreo, finalmente se toma como mejor opción la del tiempo de muestreo de 5 minutos, ya que resulta un punto intermedio entre relajar este parámetro para que reducir el coste computacional y evitar estar administrando distintos valores de insulina constantemente, y por otra parte realizar un control adecuado y robusto priorizando en todo momento la salud del paciente. A partir de este punto, se tomará por consiguiente para el resto de pruebas el tiempo de muestreo descrito.

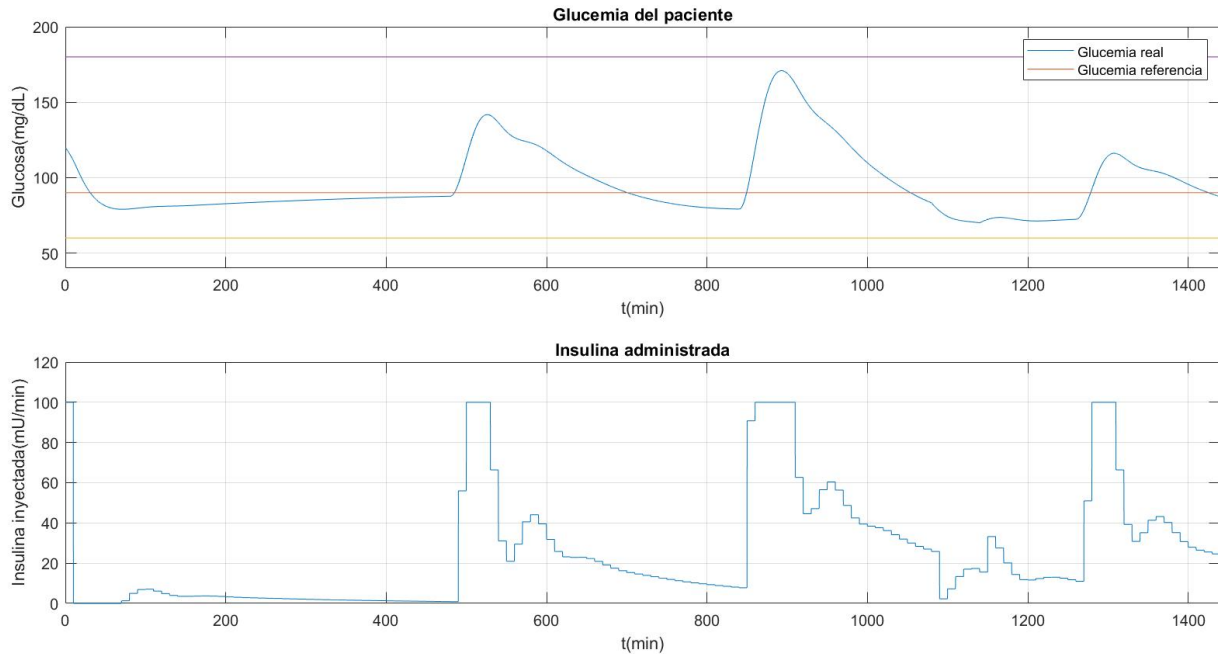


Figura 3-16. PID con tiempo de muestreo de 10 minutos

3.3.3 Control PID con inyección de insulina por lotes

Hasta ahora se había hablado de un controlador PID en tiempo discreto puesto que la señal de control no se calculaba de forma continua, es decir, con un tiempo de muestreo igual a 0, sino que esta se calculaba cada cierto tiempo cuando el sensor tomaba una muestra nueva y se actualizaba el error en el algoritmo de control implementado en la computadora. Sin embargo, aunque el valor de la insulina administrada se calcule cada 5 minutos como sucede en nuestro caso, sí que estamos suponiendo hasta este momento (tal y como se puede apreciar en todas las figuras correspondientes a las pruebas realizadas) que la forma de administrar esta insulina por parte de la bomba es continua. Es decir, que una vez se calcula el valor a inyectar, la bomba está administrando esa cantidad de insulina de forma continua durante los 5 minutos siguientes. Sin embargo, en la realidad esto no es lo que sucede. Lo normal es que la insulina se administre de forma “discreta”, esto es, por lotes. El procedimiento a seguir será el siguiente: cuando se calcule el nuevo valor de insulina, cada tiempo de muestreo, se calculará a su vez el valor total equivalente que se administraría de forma continua durante los 5 minutos. Esta cantidad total será la que se administre de una sola vez al comienzo del periodo de muestreo, y tardará en administrarse el tiempo que venga impuesto por el caudal de la bomba (la máxima cantidad de insulina que sea capaz de administrar en un instante de tiempo). De esta forma, la insulina no se inyectará de forma continua, sino en pequeños lotes cada 5 minutos, haciendo que el control sea más realista.

Para esta mejora del control PID se crean dos nuevos archivos nuevos en Matlab, uno para el control llamado “controlotes.m”, con estructura muy similar al usado anteriormente, y otro para mandar la señal de control según el tiempo de inyección calculado, llamado “inyectalotes.m”. En el primer archivo se calcula la señal de control tal y como se hacía en el archivo “control.m”, con un tiempo de muestreo de 5 minutos y *anti-windup*. La novedad viene en las últimas líneas, donde se calcula el tiempo durante el cual se debe inyectar insulina al caudal máximo para inyectar toda la cantidad total necesaria durante ese periodo de muestreo. El caudal máximo será $Q=100$ mU/min, el mismo definido anteriormente como valor máximo de la señal de control a la hora de saturar. Esta función, que se llama desde Simulink con un bloque de función igual que en los casos anteriores, devuelve el tiempo de inyección calculado que servirá de entrada, aparte del tiempo de simulación, a la segunda función. En “inyectalotes.m” primero se calcula el tiempo que ha pasado desde que comenzó el último periodo de muestreo, que irá entre 0 y 5 minutos, y que calcula como el resto del tiempo de simulación entre el tiempo de muestreo. A continuación, se tiene la condición siguiente: si el tiempo que se lleva de periodo es menor que el

tiempo calculado de inyección, entonces el valor de U_k (la señal de control y salida de la función) será 100 mU/min. En caso de que el tiempo sea mayor, entonces querrá decir que ya se ha administrado toda la insulina necesaria en ese periodo y no hay que inyectar más hasta el siguiente. Hay que decir que bloque de esta función en Simulink sí que tiene un tiempo de muestreo igual a 0, es decir, se ejecuta de forma continua, ya que es el propio algoritmo quien se encarga que la administración sea “discreta”, como ya se ha explicado. El modelo de Simulink se observa en la siguiente figura:

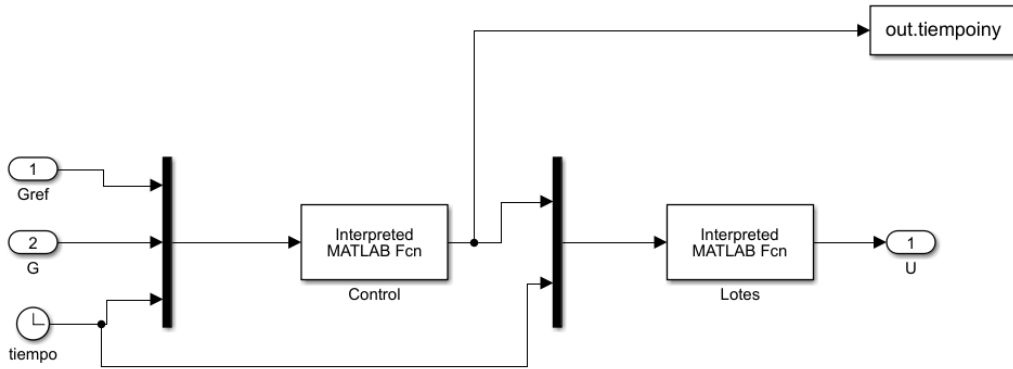


Figura 3-17. Bloques del control PID con inyección de insulina por lotes

Se realizan pruebas y se obtienen los resultados que se muestran a continuación.

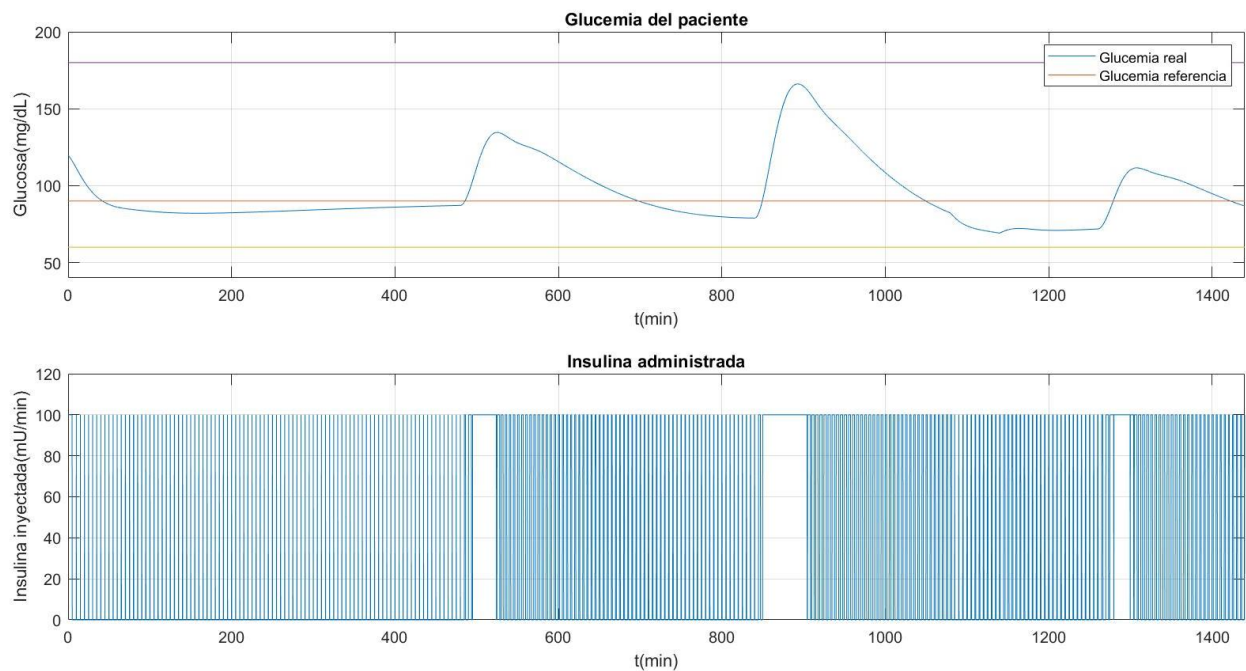


Figura 3-18. PID con inyección de insulina por lotes

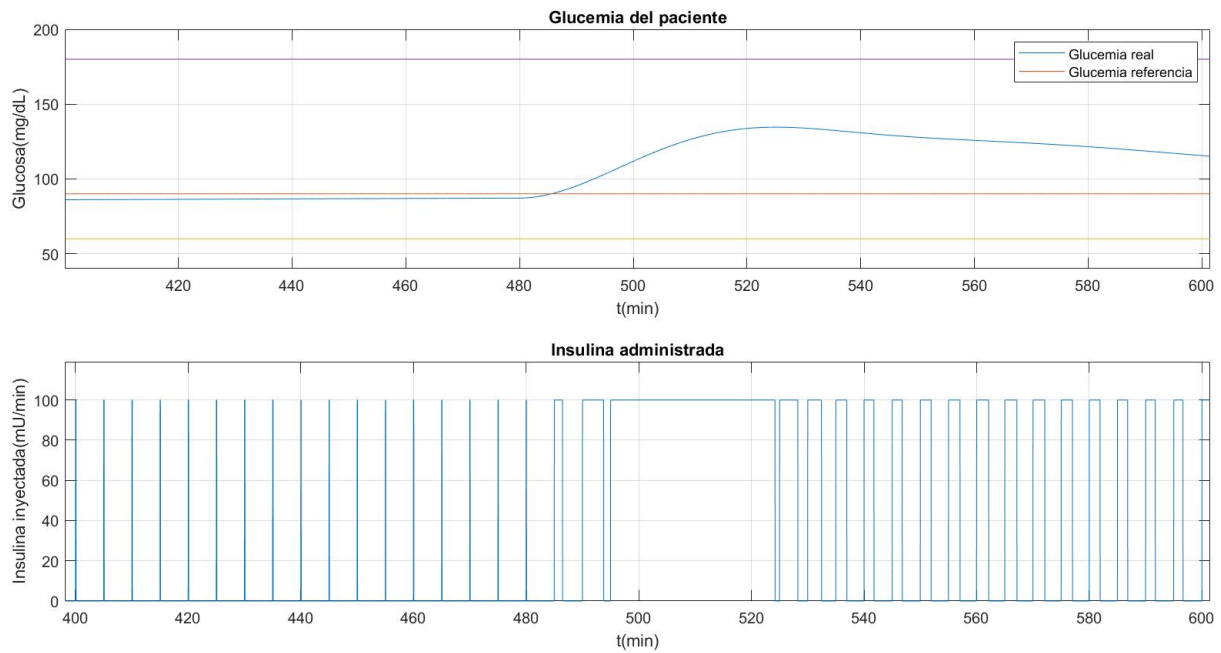


Figura 3-19. Zoom del PID con inyección de insulina por lotes

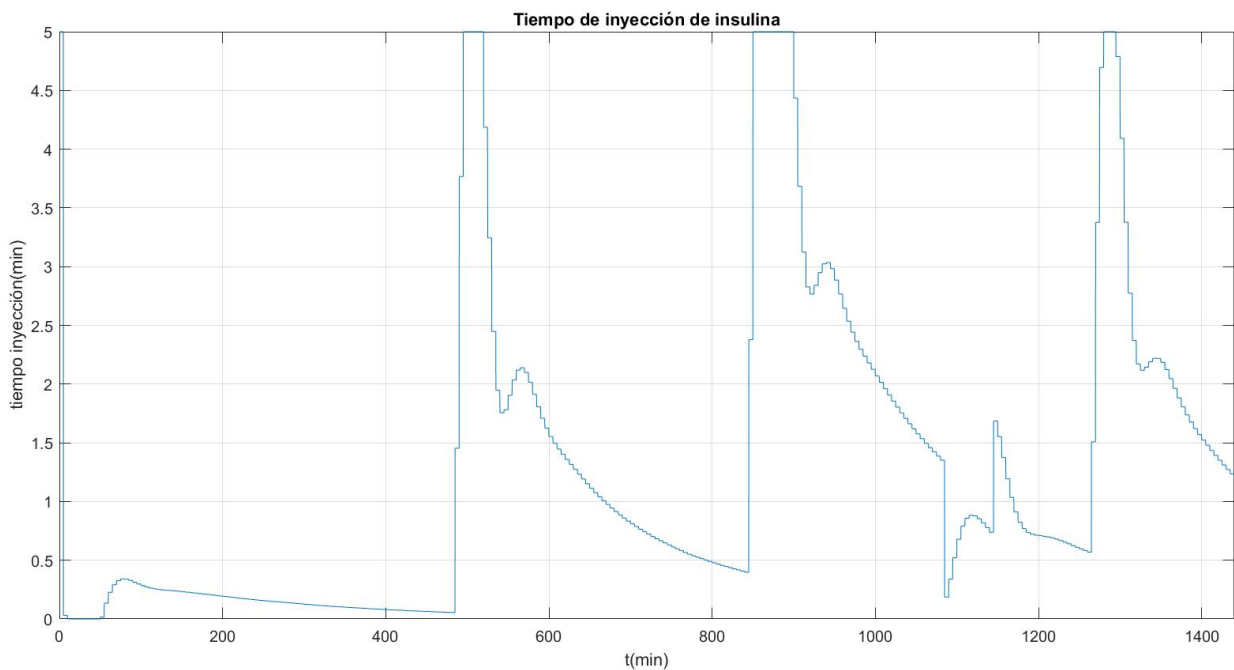


Figura 3-20. Tiempo de inyección de insulina

Tal y como podemos ver en la figura 3-18, la respuesta de la glucosa en sangre en el paciente con esta nueva versión más realista del control es prácticamente idéntica a la obtenida en el caso anterior, cuando la insulina se administraba de forma continua. Esto es excelente, porque podemos implementar el controlador PID con inyección de insulina por lotes con total seguridad ya que sabemos que no va a afectar a la glucosa del paciente, con la mejora de que se diseñando un control más realista. Puesto que en la figura no se puede apreciar del todo bien como es la señal de control y como varía esta cuando se produce una perturbación, se muestra en la figura 3-19 un zoom de la misma realizado en el intervalo de tiempo correspondiente a la primera perturbación de

comida (el desayuno). Se observa como justo antes de la perturbación, cuando el valor de glucosa es muy cercano a la referencia, la cantidad de insulina que se debe inyectar es baja y por lo tanto, el tiempo de inyección, y la señal de control tiene forma de pulso prácticamente. En cambio, cuando se produce la perturbación, se puede ver que se necesita una mayor cantidad de insulina y de esta manera el tiempo de inyección aumenta, teniendo ahora la señal de control forma de escalón. Conforme se va corrigiendo la perturbación, más estrecho es el escalón de la insulina administrada puesto que menos insulina se va necesitando. En cualquier caso, en la figura 3-20 se muestra el tiempo de inyección calculado en la primera función, que tal y como se esperaba nunca sobrepasa el valor del tiempo de muestreo, y que aumenta cuando se producen las perturbaciones y viceversa. Por lo tanto, y tras estudiar el resultado de las pruebas, se decide que la implementación de este control PID con inyección de insulina por lotes es totalmente válido para nuestro caso.

3.3.4 Controlador PID Feedforward

El control feed forward, también llamado control anticipativo de prealimentación, es un tipo de control que mide las perturbaciones que le entran al sistema a controlar de forma que puede anticiparse y compensarlas, mejorando la respuesta del sistema ante el problema de regulación. Este control PID feedforward sumaría la señal de control propia del PID que ya se ha obtenido en los apartados anteriores, con una señal de control calculada a partir de la perturbación medida (en nuestro caso dos señales de control, puesto que hay dos perturbaciones), obteniéndose así una señal de control total que se aplica al sistema. Hay que destacar un punto importante en nuestro sistema en particular, y es el hecho de medir las perturbaciones. Esto se podría traducir a la realidad como que el paciente, justo antes de comer o de hacer ejercicio (es decir, cuando se producen las perturbaciones), debería introducir en la bomba o el dispositivo que le administre la insulina la cantidad más o menos exacta de los hidratos de carbono que va a ingerir o el tiempo que va a hacer ejercicio, respectivamente. De esta forma, el algoritmo de control tendría la información necesaria para medir las perturbaciones que va a tener el sistema, ya que no se pueden medir directamente de otra forma. En la simulación que haré, sin embargo, y puesto que en Simulink sí es posible medir directamente las perturbaciones, lo que le llegue al bloque de control, además de los parámetros que ya conocemos, serán las salidas directas de cada uno de los bloques de perturbaciones, como podemos ver en la figura 3-21.

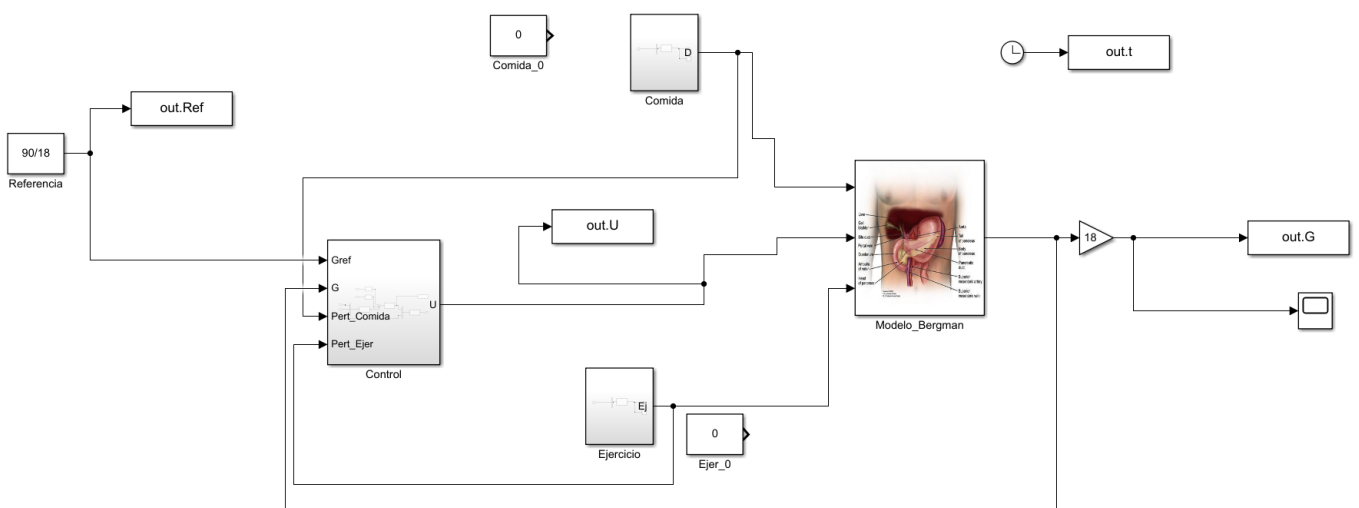


Figura 3-21. Modelo del controlador PID Feedforward

Para obtener las respectivas señales de control de cada perturbación se calcula una función de transferencia de forma que la entrada sea la perturbación medida y la salida sea la propia señal de control (de forma similar a como se hacía en el caso del controlador en tiempo continuo, aunque ahí la entrada era el error de glucosa). Para calcular Cff_{p1} y Cff_{p2} , las funciones de transferencia para el control feedforward para la perturbación 1 (comida) y 2 (ejercicio físico), partimos de la siguiente ecuación observando el siguiente esquema:

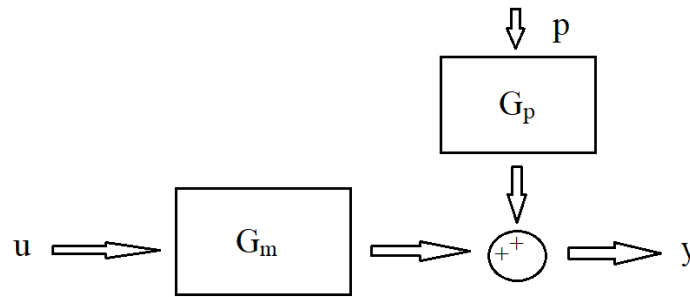


Figura 3-22. Esquema para la obtención de Cff

$$0 = y = u * G_m + p * G_p \quad (31)$$

Siendo u la señal de control aplicada al sistema, G_m la función de transferencia del modelo, p la perturbación medida, G_p la función de transferencia de la perturbación, e y la salida.

De esta forma en la ecuación se muestra el caso que queremos, es decir, que el incremento de la salida sea 0 y esté en equilibrio, y para esto la señal de control que le entra al sistema debe compensar la perturbación, tal que:

$$u = -p * \frac{G_p}{G_m} \quad (32)$$

Por tanto la función de transferencia, a la que en este caso general llamaré Cff, será igual a:

$$Cff(s) = \frac{-G_p}{G_m} \quad (33)$$

En nuestro caso particular, se tienen las siguientes funciones de transferencia para las perturbaciones, calculadas en el script “funcion_transferencia.m” ya mencionado en el capítulo 2:

$$G_d(s) = \frac{15000}{(15000s+7)} \quad (34)$$

$$G_{ej}(s) = \frac{-67500}{(15000s+7)} \quad (35)$$

En el script “feed_forward.m” se calculan las dos funciones de transferencia para el control feed forward tal como se ha visto en la ecuación (33), y se obtiene:

$$Cff_d(s) = \frac{3.49*10^{14}s^3+8.49*10^{11}s^2+5.45*10^8s+105000}{(473850s+221.1)} \quad (36)$$

$$Cff_{ej}(s) = \frac{-(1.575*10^{15}s^3+3.821*10^{12}s^2+2.453*10^9s+472500)}{(473850s+221.1)} \quad (37)$$

Una vez se tienen las funciones se puede implementar el controlador en Simulink, el cual podemos ver en la siguiente figura 3-23. Hay que decir que dentro de los bloques de función de transferencia para Cff_d y Cff_{ej} , puesto que estas no son bipropias (mismo número de polos que de ceros) y Simulink pide que lo sean, se añaden polos de alta frecuencia hasta que el denominador tiene tantos polos como ceros el numerador. Los polos de alta frecuencia que se añaden son, en este caso, los mismos que se añadieron en el controlador PID de tiempo continuo y que se pueden ver en la ecuación (23), y no afectan al comportamiento del sistema. De esta forma, y como ya se ha explicado antes, se tiene la señal de control propia del PID, que se obtiene en el bloque “ControlPID”, a la que se suman las señales de control de las perturbaciones calculadas. Puesto que en el apartado anterior se había estudiado que la inyección de insulina por lotes era válida e incluso favorable al hacer el control más realista, se opta por seguir manteniéndola en este apartado también (así como en el siguiente capítulo). Para esto, la señal de control total se pasa por el bloque “Control”, donde dentro se llama a la función “lotes.m” que primero satura la señal y después calcula el tiempo de inyección. Por último, se tiene de nuevo el bloque “Lotes” que a partir del tiempo de inyección calcula el valor de insulina a administrar.

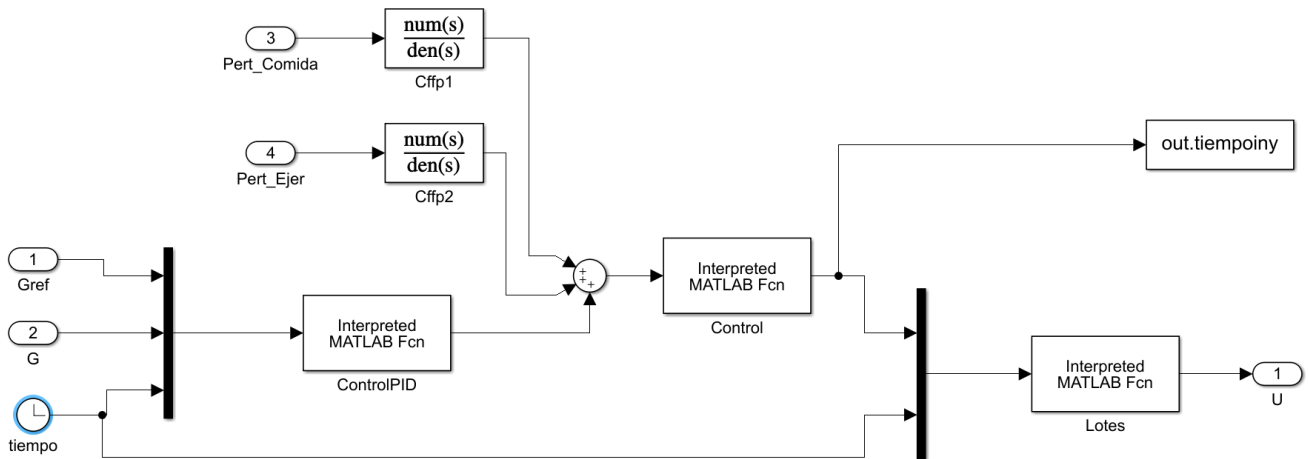


Figura 3-23. Bloque del control PID Feedforward

Cuando se simula se obtienen los siguientes resultados:

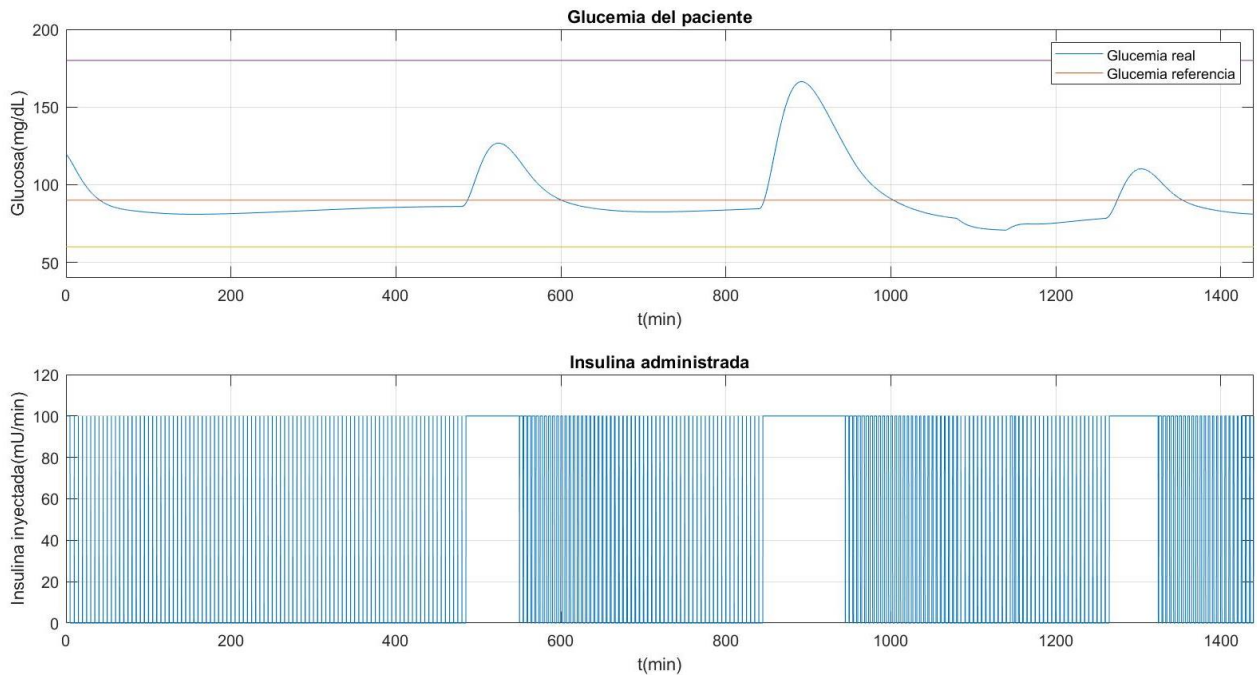


Figura 3-24. Controlador PID Feedforward con inyección de insulina por lotes

A primera vista, y comparando con la figura 3-18, observamos una clara diferencia en cuanto al tiempo en corregir las perturbaciones, especialmente la comida. En este caso el controlador es capaz de corregir las perturbaciones en menos tiempo de forma notable, reduciendo el tiempo por lo tanto en el que la glucemia se encuentra a niveles por encima del de referencia. En el caso del desayuno, por ejemplo, observamos que con el feed forward el pico se corrige y la glucosa en sangre vuelve al valor de referencia hacia el minuto 600, mientras que sin el feed forward este no se corregía hasta el minuto 700. Esto 100 minutos menos (casi 2 horas) son vitales y de gran importancia para la salud del paciente, ya que significan maximizar el tiempo con una glucosa a niveles de una persona no diabética, reduciendo las hiperglucemias y los problemas que ellas conllevan. En el caso del almuerzo y cena también se reduce de forma igualmente notable este tiempo. También podemos destacar que los picos (valores máximos y mínimos alcanzados) son similares a los obtenidos con el controlador sin feed forward, o en cualquier caso ligeramente menos pronunciados, lo cual también es bueno para nuestro sistema. Por estos motivos, y sin olvidar que el feedforward conllevaría que el paciente proporcione a la bomba los

valores que tomarán las perturbaciones como he mencionado anteriormente (sin que esto conlleve un problema en sí, ya que es factible), se determina que el controlador feedforward con inyección de insulina por lotes es una excelente opción para este sistema en concreto, ya que aporta unas mejoras muy importantes a la hora de mantener la glucemia del paciente en los valores correctos.

4 CONTROL PREDICTIVO BASADO EN EL MODELO (MPC)

“Si quelqu’un aime une fleur qui n’existe qu’à un exemplaire dans les millions et les millions d’étoiles, ça suffit pour qu’il soit heureux quand il les regarde”.

- “El Principito”, Antoine de Saint-Exupéry -

El Control Predictivo basado en el Modelo o MPC (del nombre inglés, *Model based Predictive Control*) es un tipo de control avanzado ampliamente usado, especialmente en procesos industriales, pero también en ámbitos científicos, que se basa en el modelo del sistema a controlar para predecir las salidas o respuestas de este para de esta forma optimizar la señal de control a aplicar en el futuro. En este capítulo se aplicará este control al sistema formado por el Modelo de Bergman usando para ello la herramienta que ofrece Matlab y Simulink llamada “*MPC Toolbox*”.

4.1 Introducción al Control Predictivo

Si bien en el anterior párrafo he dicho que el Control Predictivo basado en el Modelo (a partir de ahora se llamará por sus siglas MPC para resultar más corto) es un tipo de control, realmente esto no es así ya que el MPC se puede definir como un conjunto de distintos tipos de control, como el óptimo o el de procesos multivariables. Tal y como se ha mencionado antes, el MPC se basa en dos principios básicos: el uso del modelo para poder predecir las salidas en un futuro del sistema, y la optimización de la señal de control. S. Diego Fernando [6] apunta en su artículo que el objetivo principal actualmente en el mundo del control es tener la posibilidad de actuar sobre las variables manipuladas o señales de control, para que estas puedan adaptarse según los requisitos del proceso, que cada vez son más exigentes y complejos (ya sea por calidad del producto, causas económicas, ambientales...). El MPC es una gran opción a la hora de enfrentarse a este objetivo, puesto que aporta flexibilidad en cuanto al modelo y a las restricciones, haciendo que sea probablemente la estrategia de control basada en modelo más escogida a la hora de controlar este tipo de procesos [7]. Algunas de las ventajas que se pueden destacar del MPC son las siguientes: es multivariable, funciona tanto con modelos lineales como no lineales, al ser predictivo sirve para solucionar el problema de los tiempos muertos, ayuda a compensar perturbaciones medibles, se le pueden añadir restricciones rígidas... Como contrapunto, el MPC conlleva una implementación más compleja, sobre todo comparado con los controladores tipo PID, así como un mayor coste

computacional. Aunque como todo, será el proceso en sí a controlar y los requisitos que este conlleve los que harán que se termine optando por un tipo de control u otro [6].

En el control predictivo tenemos variables manipulables (equivalente a las señales de control) y variables controladas (las salidas del sistema), así como las perturbaciones. Puesto que es un control en tiempo discreto, se tiene también un tiempo de muestreo, que será cada cuanto tiempo se realice el algoritmo de control. Se define un horizonte de predicción, que se corresponde con el rango de tiempo (o el número de instantes en discreto) que el controlador es capaz de predecir la salida del sistema desde el instante actual. Es decir, si nos encontramos en el instante 200 segundos y el horizonte de predicción es de 300 segundos, esto quiere decir que el controlador MPC puede predecir la salida del sistema hasta el instante 500 segundos. Se debe encontrar un valor adecuado para este parámetro, pues si es demasiado bajo, el controlador no podrá controlar de forma adecuada si se encuentra una perturbación o un cambio de referencia para los cuales el sistema necesite más tiempo de adaptación o corrección que el propio horizonte de control; igualmente, si es muy alto, se pueden estar haciendo predicciones en vano si se produce una perturbación de repente y ya no sirven las predicciones hechas. En cada instante de muestreo, el controlador primero mide la salida real del sistema y la compara con el valor que había predicho en el instante anterior que ésta tendría (podrá o no ser igual, dependiendo de las perturbaciones que haya y que no se puedan medir). Una vez hecho esto, predice una serie de escenarios del comportamiento del sistema aplicando en cada uno de ellos distintas señales de control. Se busca que el error entre la referencia y la salida real del sistema sea el mínimo posible, pero también que se cumplan, si las hay, una serie de restricciones. De esto se encarga el optimizador, ya que mediante unas funciones de coste en donde se asignan unos pesos dependiendo de si queremos penalizar acciones en la señal de control o en la salida (por ejemplo, si no queremos que la señal de control realice cambios muy bruscos, o si preferimos que sí haya cambios bruscos a cambio de darle un mayor peso a que la salida se mantenga al valor de referencia el máximo tiempo posible) es capaz de elegir la secuencia de control más adecuada en cada caso, correspondiéndose esta a la función con un coste menor. El tiempo o el número de instantes de muestreo que dura la secuencia de control se llama horizonte de control, e igual que el horizonte de predicción se debe definir al principio. Cuanto mayor sea el horizonte de control, mayor será el coste computacional, pero tampoco se puede definir demasiado bajo ya que entonces el control podría ser menos efectivo. Una vez se tiene la secuencia a aplicar elegida por el optimizador, se aplica al sistema solo el primer valor de la señal de control calculada (no la secuencia entera). En el siguiente instante de muestreo se volverán a repetir todos los pasos, empezando de nuevo con la medición de la salida real. El horizonte de predicción se va deslizando, moviéndose cada vez a un instante de muestreo más [7].

4.2 Implementación del controlador MPC en Matlab y Simulink

Una vez hecha una breve introducción de este tipo de control, pasaré a explicar como se ha implementado usando para ello el *Toolbox* que ofrece Matlab y Simulink.

Siguiendo los modelos anteriores de Simulink para los otros controladores (y más en concreto, para el PID feedforward), ahora se cambia el interior del bloque de control. A este bloque le sigue llegando el valor de glucosa real, la glucosa de referencia y las perturbaciones, estas últimas como una señal multiplexada, como se puede ver en la figura 4-1, y saca la insulina que se administra siguiendo con la inyección por lotes. Por su parte, dentro el bloque de control, el bloque de la función de Matlab que llamaba a la función que implementaba el PID se cambia por un bloque de controlador MPC, perteneciente al *Toolbox* mencionado. A este bloque le llega como entradas el valor de salida del sistema, la referencia y las perturbaciones, y saca la señal de control que posteriormente se pasará por las funciones correspondientes a la inyección por lotes al igual que con el PID feedforward, como vemos en la figura 4-2.

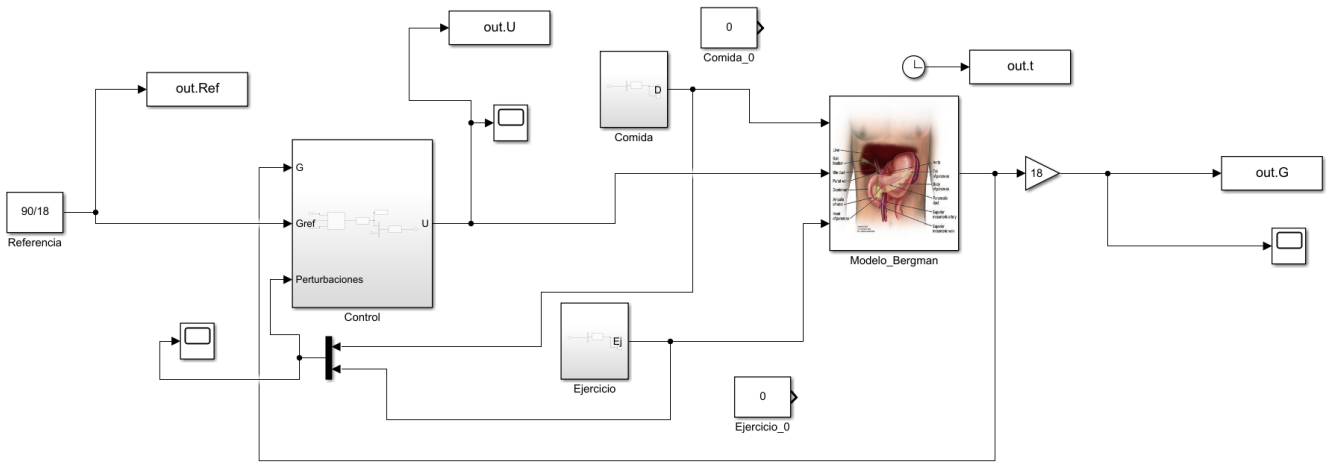


Figura 4-1. Modelo del controlador MPC

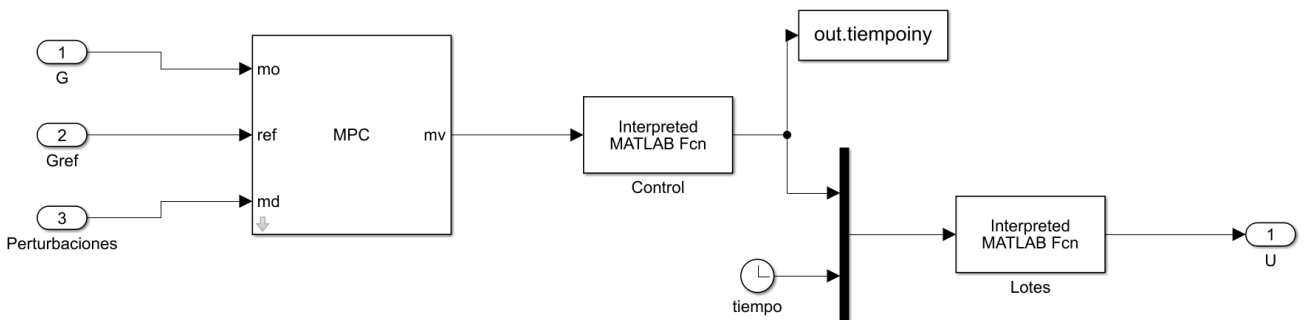


Figura 4-2. Bloque de control del MPC

El bloque MPC permite diseñar de forma intuitiva un controlador predictivo. Para ello, primero se introducen el número de variable manipulables, variables controladas y perturbaciones. En este caso, tal y como se muestra en la figura 4-3, se tienen una variable manipulable (la insulina), una variable controlada medida (la glucosa) y dos perturbaciones medidas (la comida y el ejercicio). Se escogen las señales del modelo que corresponden a cada una de las señales mencionadas anteriormente, y Simulink linealiza automáticamente el sistema. Una vez hecho esto, se abre una ventana donde se pueden realizar distintas acciones para modificar y diseñar el controlador, y dentro de la pestaña “Tuning” se puede ir modificando los horizontes de control y predicción, así como los pesos de la función de coste del optimizador y las restricciones en las variables. Esta barra de herramientas se puede observar en la figura 4-4.

Se pone como restricción que la variable manipulable solo pueda valer de 0 a 100 mU/min (tal y como estaba en el saturador del PID), y se define el tiempo de muestreo como 300 segundos. El horizonte de predicción se define como 300 segundos también al principio (para que prediga al menos hasta el siguiente tiempo de muestreo), mientras que el de control se deja el que viene por defecto, que es 10 segundos. Los pesos igualmente se dejan los que vienen por defecto en esta primera simulación. Se realizan distintas simulaciones, cambiando para cada una de ellas el valor de los parámetros anteriormente mencionados, para ir observando cómo afectan estos cambios y cuáles son los valores más adecuados. Se obtienen los siguientes resultados:

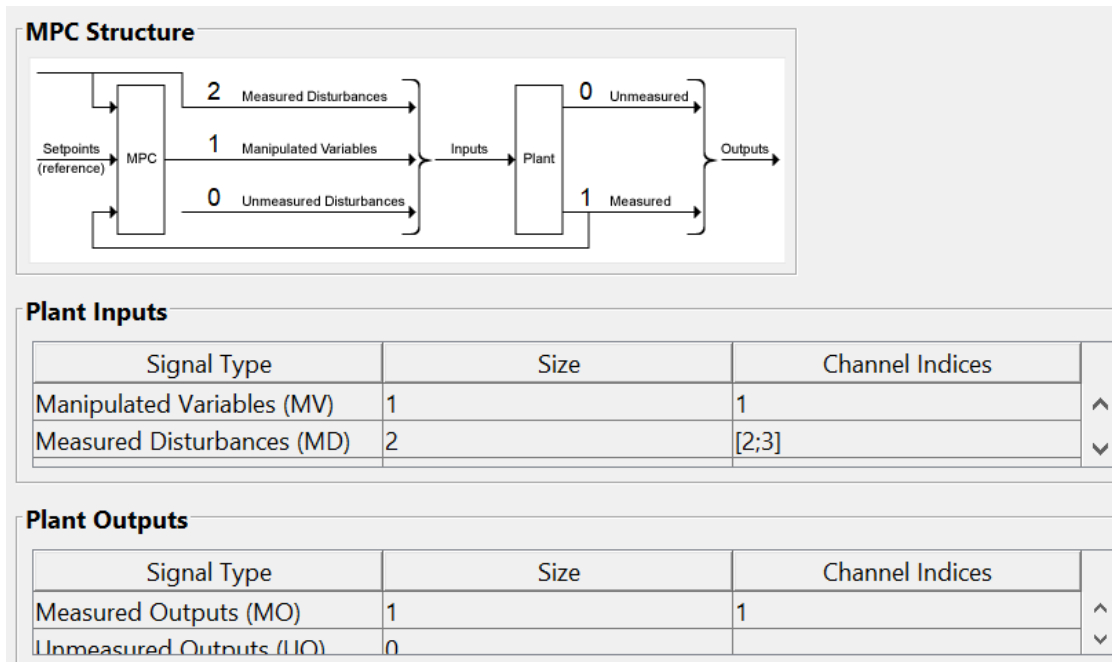


Figura 4-3. Estructura del controlador MPC

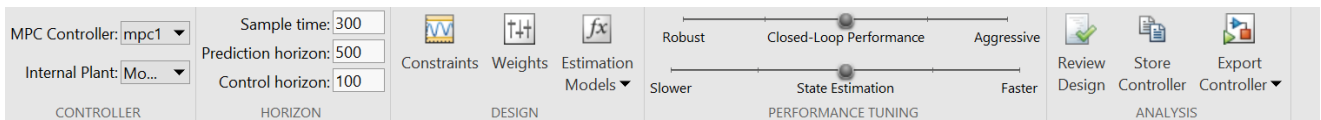


Figura 4-4. Barra de herramientas para el diseño del controlador MPC

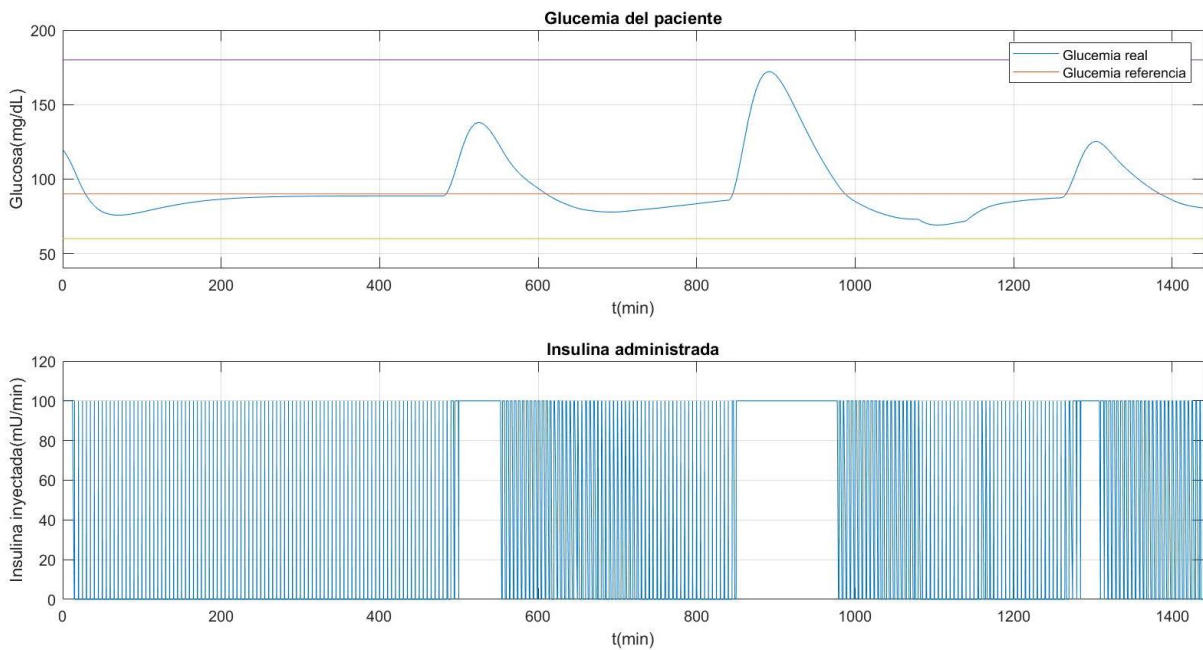


Figura 4-5. Controlador MPC HP: 300 s; HC: 10 s; wu: 0.1; wg: 1

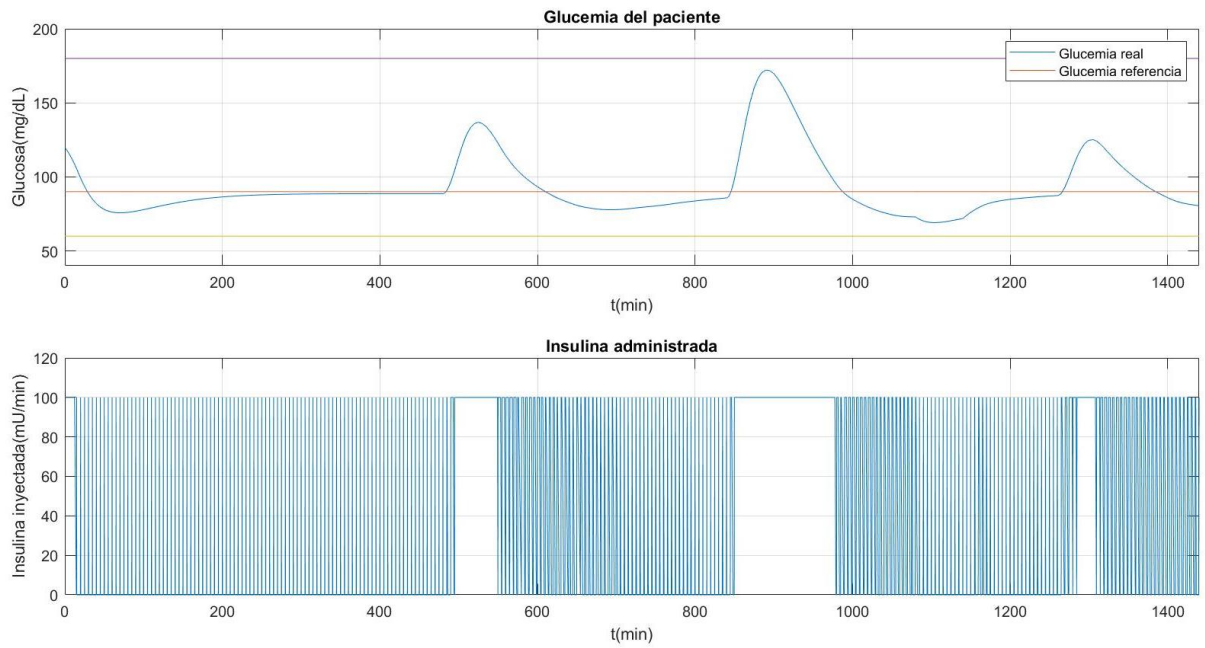


Figura 4-6. Controlador MPC HP: 300 s; HC: 100 s; wu: 0.1; wg: 1

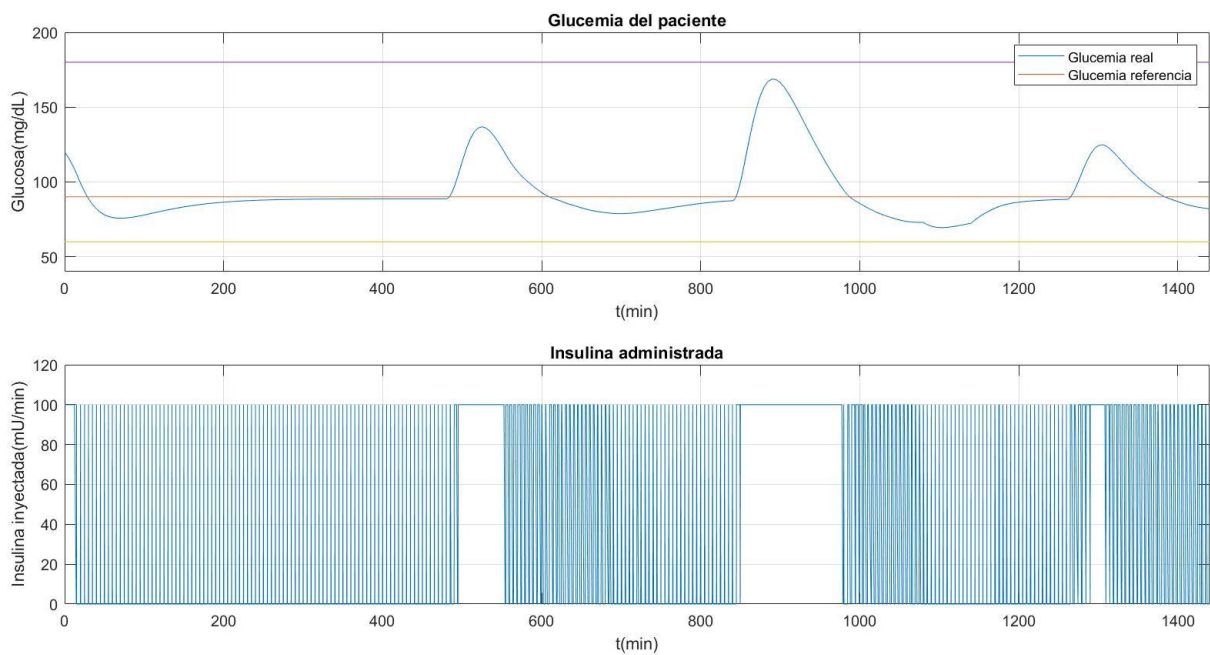


Figura 4-7. Controlador MPC HP: 300 s; HC: 100 s; wu: 0.1; wg: 10

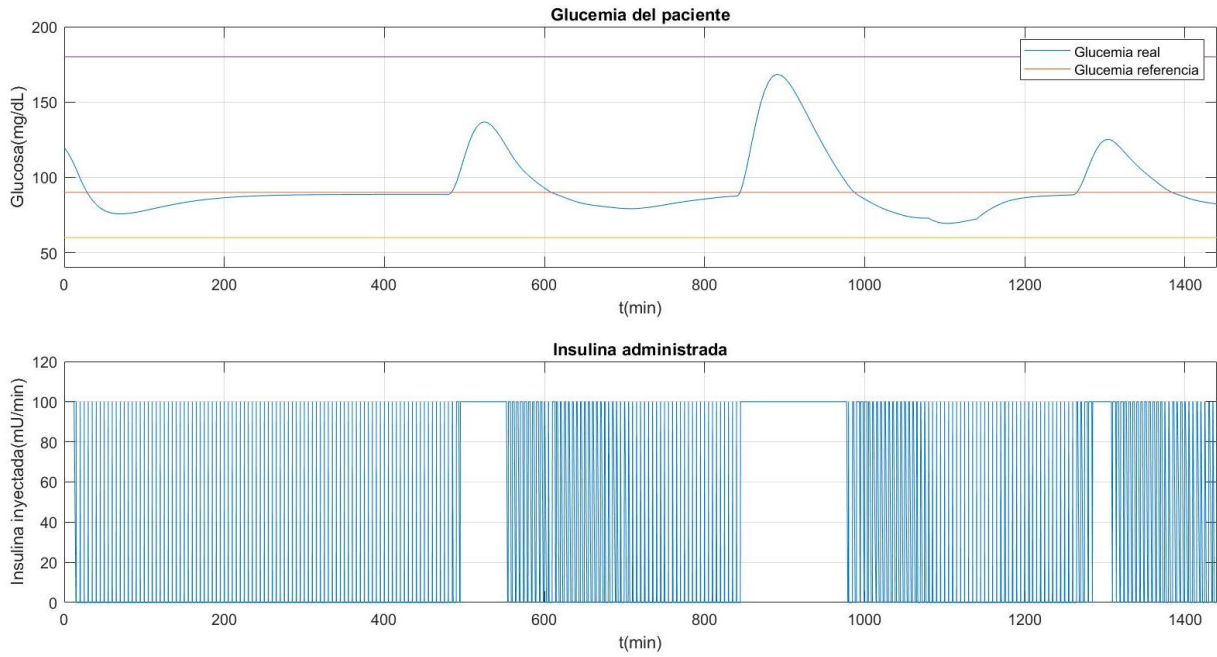


Figura 4-8. Controlador MPC HP: 500 s; HC: 50 s; wu: 0.1; wg: 50

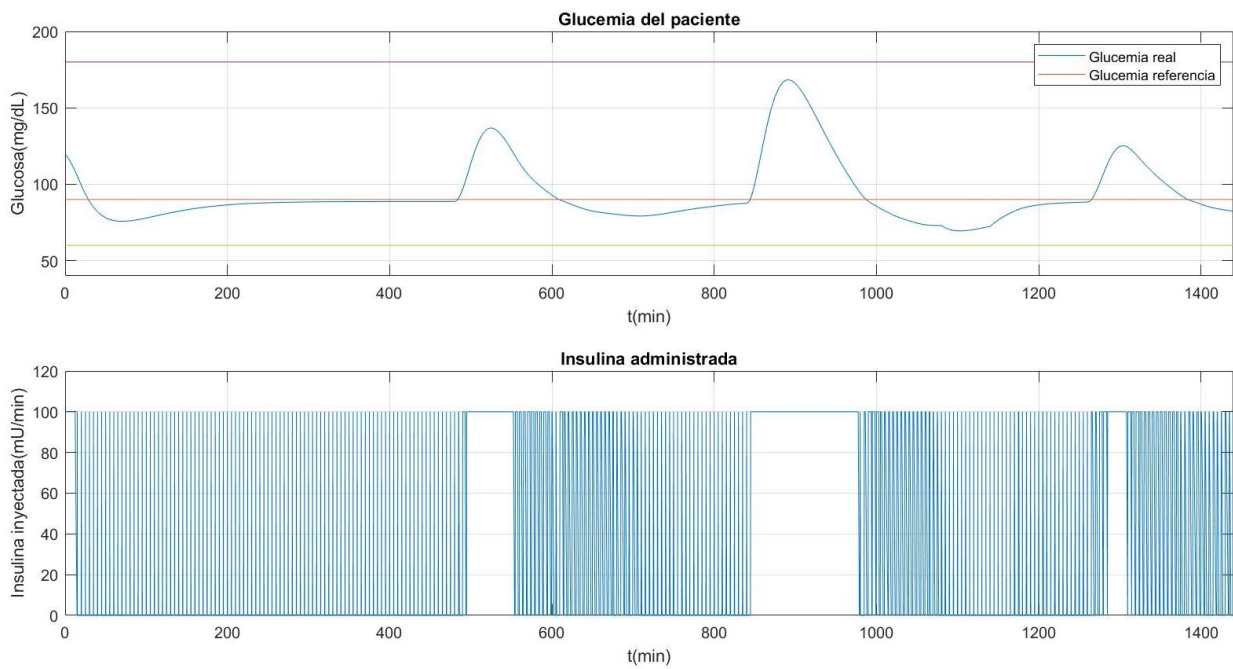


Figura 4-9. Controlador MPC HP: 300 s; HC: 50 s; wu: 0.1; wg: 50

En la figura 4-5, con los valores de los parámetros del MPC que se han definido al principio, podemos ver ya algunas diferencias notables respecto del controlador anterior, el PID feedforward, de las cuales hablaré con mayor detalle en el siguiente capítulo. La respuesta del sistema sigue cumpliendo el objetivo de mantenerse siempre dentro de los niveles deseados de glucosa, y ahora el tiempo de establecimiento, sobre todo al principio, cuando todavía no hay ninguna perturbación, es mucho menor. Por lo demás el comportamiento es bastante parecido, y los picos máximos y mínimos alcanzados tienen igualmente unos valores similares. Con el fin de intentar mejorar la acción de control, se aumenta en la siguiente simulación el horizonte de control a 100, mostrándose el resultado en la figura 4-6. Sin embargo, como se puede observar, la respuesta del sistema es idéntica a la anterior, sin ningún cambio destacable, y es que no hay que olvidar que se tiene la restricción de saturación de la señal de control que hace que, si bien el control podría ser mejor al modificar estos parámetros a priori, esta limitación de la señal de control hace que este no se pueda mejorar a partir de cierto punto. Por lo tanto, en este caso resulta poco útil aumentar tanto el horizonte de control y solo aumentaría el coste computacional.

A continuación se decide modificar el valor de los pesos. Tenemos dos: uno para la señal de control (w_u), que penaliza las acciones bruscas y “agresivas” (cuanto mayor sea, mayor es la penalización); y otro para la salida (w_g), que penaliza que ésta se desvíe de la referencia (igualmente cuanto mayor sea, mayor es la penalización). En nuestro caso, lo más importante es que la glucosa se mantenga lo máximo posible cerca de la referencia, y que cuando se aleje debido a las perturbaciones, se corrija lo más pronto posible. Es por ello que el peso que se aumenta para darle más importancia es el de la salida, poniéndolo con un valor de 10 (anteriormente a 1), mientras que el de la señal de control se mantiene a 0.1. En la figura 4-7 se puede observar el resultado, y se aprecia que, tal y como se esperaba, los picos debidos a las perturbaciones son ligeramente más suaves, puesto que ahora se penalizan más dichos picos en la función de coste dentro del optimizador. Esto es bueno para nuestro sistema, por supuesto. Aparte de esta suavización, el resto del comportamiento es prácticamente idéntico. A continuación, se aumenta w_g a 50, para observar si gracias a esto se pueden reducir aún más los picos, y también se aumenta a 500 segundos el horizonte de predicción, y se muestra el resultado en la figura 4-8. Sin embargo, se ve que la gráfica que se obtiene es prácticamente la misma que en el anterior caso, es decir, que el aumento del horizonte de predicción no hace ningún efecto (y sí que aumenta de nuevo el coste computacional), y el del w_g tampoco. Esto se debe de nuevo a la restricción de la señal de control que se ha comentado antes. Por lo tanto, y como punto intermedio para no evitar un exceso coste computacional y también tener el mejor control posible con este MPC, se termina optando por los siguientes valores como parámetros definitivos: horizonte de predicción 300 segundos; horizonte de control 50 segundos; w_u 0.1; w_g 50. El resultado se observa en la figura 4-9, y como se puede ver, aunque se han reducido algunos valores respecto de la figura anterior, el comportamiento del sistema sigue siendo el mismo. Como conclusión, podemos decir que el control MPC es bueno en nuestro caso y se consigue el objetivo, obteniendo unos resultados parecidos a los del controlador PID feedforward (esto es lógico, puesto que era el otro controlador al que también le llegaba la información de las perturbaciones). El MPC mantiene la glucosa en el rango, en la mayoría del tiempo a valores muy cercanos a la referencia, y corrige en poco tiempo las perturbaciones, lo cual garantiza un buen control de la glucosa para el paciente con diabetes.

5 COMPARACIONES Y CONCLUSIONES FINALES

“Recuerda siempre que eres más fuerte de lo que crees, más valiente de lo que pareces y más listo de lo que piensas”.

- “Winnie the Pooh”, A. A. Milne -

Después de haber diseñado y estudiado distintos controladores en los capítulos anteriores, en este último capítulo se realiza la comparación entre ellos. A partir de la misma, y también teniendo en cuenta otros aspectos igualmente importantes, como el económico y sobre todo la salud del paciente, se sacan unas conclusiones finales sobre este trabajo.

Al comienzo del trabajo se marcaron como objetivos, tanto por parte del tutor como por mi parte, los siguientes: la revisión de bibliografía sobre un tema concreto (en este caso, la diabetes tipo 1 y el control automático de la glucosa en este tipo de diabetes); el estudio y entendimiento de un modelo matemático-biológico complejo para el control (en este caso, el modelo de Bergman); el correcto manejo y entendimiento de los diferentes conceptos de control explicados en clase, así como de un programa matemático y de simulación como Matlab; la búsqueda y estudio de nuevos conceptos (no estudiados en clase) sobre control y aplicarlos al trabajo (en este caso, el control MPC); y por último, hacer un análisis crítico sobre los resultados obtenidos. Quitando este último objetivo, que es el que se desarrollará en este capítulo, en este punto del trabajo se puede afirmar que los objetivos marcados al comienzo se han cumplido satisfactoriamente.

De todos los controladores realizados, y puesto que algunos son mejoras de otros, me centraré en la comparación de 3 especialmente: el PID con inyección de insulina por lotes (sin feedforward), el PID feedforward, y el MPC. Si bien realmente los dos últimos son los que se pueden comparar de forma justa ya que son los dos a los que les llega la información de las perturbaciones medidas, mientras que al primero sin feedforward no, se mete también dentro de la comparación para estudiar si realmente resulta un gran cambio significativo a favor la inclusión de las perturbaciones dentro del control, o por el contrario la mejora no es tan grande y resulta más significativo el incremento en el coste. Si bien a lo largo de los capítulos se han podido ver las gráficas de los resultados y comparar unas con otras, para poder llevar a cabo una comparación cuantitativa se realiza una tabla a continuación con los valores pico máximo y mínimo alcanzados con cada controlador, así como el tiempo de establecimiento al principio de la simulación cuando aún no hay perturbaciones (correspondería desde las 00:00 horas hasta el desayuno) y también el tiempo que tarda la glucosa en volver al valor de referencia desde que comienza la segunda perturbación de comida (el almuerzo), para ver cuánto tarda en corregirse esa perturbación,

que es la mayor.

Comparación cuantitativa de los controladores				
Tipo de control	Valor máximo de pico	Valor mínimo de pico	Tiempo establecimiento antes de las perturbaciones	Tiempo de corrección después del almuerzo
Sin control (bucle abierto)	241.9 mg/dL	72.32 mg/dL	-	274 min
PID sin feedforward	166.1 mg/dL	69.14 mg/dL	346.5 min	210 min
PID feedforward	166.3 mg/dL	70.74 mg/dL	333 min	163 min
MPC	168.4 mg/dL	69.45 mg/dL	230 min	146 min

Tabla 5-1. Comparación cuantitativa de los controladores

Como se puede ver, en la tabla también se añade los valores correspondiente a la respuesta del modelo en bucle abierto, sin ningún tipo de control, para poder observar el cambio favorable y la mejora en la respuesta cuando se implementa un control en bucle cerrado en este sistema, que también era uno de los objetivos de este trabajo. Por una parte, en los 3 controladores se reduce el valor de pico máximo, casi 100 mg/dL respecto al bucle abierto, alcanzándose en los 3 un valor muy similar y que resulta adecuado pues no se llega ni si quiera a una hiperglucemia (que sería por encima de 180 mg/dL). Lo mismo se puede decir de los valores de pico mínimo, alcanzados cuando se realiza el ejercicio, también prácticamente iguales en los 3 controladores y con un valor cercano a la hipoglucemia, pero que tampoco la alcanza (ésta se marca a los 60 mg/dL). En cuanto al tiempo de establecimiento antes de las perturbaciones, aquí el MPC es el que destaca con claridad, tardando unos 100 minutos menos que los PID. Sin embargo, esto no quiere decir que estos tengan una respuesta muy alejada de la referencia, ya que como podemos ver en la figura 3-24 el valor de la glucosa durante este tiempo es cercano a la referencia, unos 5 o 10 mg/dL por debajo, siendo esto poco significativo ya que esos valores de glucosa son igualmente muy aceptables. En el MPC sí que llega al valor de referencia y se mantiene ahí hasta la perturbación. Por último para finalizar esta comparación cuantitativa, se estudia el tiempo de corrección de la segunda perturbación, un parámetro más importante si cabe que el anterior, ya que cuanto menor sea este tiempo, mejor será a la larga la salud del paciente. Es aquí donde se nota también que el control tiene información de las perturbaciones para poderse anticipar y compensarlas con una señal de control más alta. Mientras que en el PID sin feedforward se tarda 210 minutos (3 horas y media), en el PID feedforward y el MPC se tarda una hora menos, lo cual es bastante significativo y resulta muy beneficioso para la salud del paciente como acabo de decir. Desde luego, esta mejora hace que estos dos controladores sean mucho más efectivos que el PID sin feedforward, sin olvidarse que para ello el paciente debe proporcionar cuánto valen las perturbaciones.

Una vez realizada la comparación, toca sopesar distintos aspectos y elegir uno de los controladores. Antes de nada hay que destacar un aspecto importante, y es el hecho de poder mejorar los controladores diseñados. Como la mayoría de trabajos de ingeniería, este trabajo queda abierto a futuras mejoras con técnicas de control más complejas o con posibles innovaciones que surjan en este campo del control de la diabetes. De hecho, los controladores que se han diseñado a lo largo del trabajo podrían realizar un mejor control del sistema si no fuese por una restricción que se puso desde el principio y que se ha ido comentando a lo largo del mismo: la limitación de la señal de control. Desde el diseño del primer controlador se impuso que la insulina máxima que se podía administrar era 100 mU/min. Un valor más alto en esta limitación (o incluso no tener limitación) supondría un control mucho más exacto, sobre todo en el PID feedforward o el MPC, que reducirían al mínimo los picos que se provocan por las perturbaciones consiguiendo que la glucosa se mantuviera prácticamente todo el tiempo al nivel de referencia. Pero ahora bien, esto implicaría una bomba con un caudal más grande, y por lo tanto un tubo

o catéter más grande e invasivo para el paciente, lo cual no sería para nada cómodo y, quizás, tampoco sería viable si tuviera que ser demasiado ancho por ejemplo. Puesto que lo principal en este trabajo es la salud y el bienestar del paciente con diabetes tipo 1 (para ello se realiza un control automático de la glucosa, para que su salud pueda ser mejor), y puesto que igual de importante es el buen control como que el paciente se sienta bien y pueda realizar una vida normal sin tener que estar cargando con un dispositivo incómodo, invasivo o que le limite en sus actividades diarias, esta limitación en la señal de control es obligada, sabiendo, eso sí, que el control también tendrá un punto a partir del cual ya no se podrá mejorar debido a esto. Por la comparativa se puede deducir que las mejores opciones son el PID feedforward y el MPC, puesto que son los dos que corrigen más rápido las perturbaciones. Entre estos dos controladores, sabiendo que el MPC supone en la realidad una implementación más compleja y que conlleva un mayor coste computacional, yo optaría por quedarme con el PID feedforward para nuestro sistema. Este ofrece unos resultados muy similares a los del MPC y supondría un ahorro económico y más facilidad de implementación. Esto último es otro aspecto importante a destacar, puesto que si se quisiera llevar a la realidad este control automático e implementarlo en un dispositivo de inyección de insulina (ya sea bomba o dispositivos que se desarrollen a lo largo de estos años siguientes), es necesario que este conlleve un coste no muy alto para que se facilite esta implementación, y por consiguiente que el precio del producto no fuera demasiado alto y así poder llegar al máximo de pacientes con diabetes tipo 1 posible y que estos se pudieran beneficiar de este dispositivo, suponiendo una gran ayuda en su día a día. Puesto que esto último resulta mucho más importante que los 5 o 10 mg/dL de glucosa que hay de diferencia respecto a la referencia al comienzo de la simulación en el PID feedforward (y que hemos visto que no hay en el MPC), se añade otra razón de peso más para elegir este controlador.

Como conclusión, después de los estudios y comparaciones realizados a lo largo del trabajo, se termina optando como mejor opción de los controladores diseñados para el control automático de la glucosa en pacientes con diabetes tipo 1 el PID feedforward, tanto por sus buenos resultados a la hora de controlar, haciendo que la glucosa se mantenga en el rango adecuado y corrigiendo las perturbaciones de una forma eficaz, como por su facilidad de implementación y menor coste, posibilitando que así un mayor número de personas pudieran tener acceso a él si se llevara a cabo en la realidad. Para que este controlador funcionase de la forma más efectiva posible (la que hemos visto en este trabajo) haría falta que el paciente proporcionara la cantidad aproximada de hidratos de carbono que va a ingerir antes de comer y el ejercicio físico, sin que esto supusiera un esfuerzo extraordinario pues sería algo rápido y sencillo de llevar a cabo; si bien tampoco ocurriría nada grave si alguna vez por algún motivo no proporcionara esta información, ya que simplemente el controlador actuaría en este caso como el PID sin feedforward (que hemos visto que realiza un buen control también, solo tardaría algo más en corregir la perturbación correspondiente).

REFERENCIAS

- [1] Organización Mundial de la Salud. Diabetes. (13 de abril de 2021). [En línea]. Disponible: <https://www.who.int/es/news-room/fact-sheets/detail/diabetes> [Consulta: 16 de abril de 2021].
- [2] F. Barragán, M.G. Ortega Linares, y F. Rubio. "Técnicas de control robusto para la regulación de glucosa en sangre en pacientes con Diabetes Mellitus tipo 1".
- [3] A. Campos Carceller. "Diseño de un sistema de control multivariable del nivel de glucosa en sangre para pacientes con Diabetes Mellitus tipo 1". Trabajo de fin de grado en Ingeniería en Tecnologías Industriales. Escuela Superior de Ingenieros Industriales de Valencia, 2014.
- [4] A. Nath *et al.* "Physiological Models and Control for Type 1 Diabetes Mellitus: A Brief Review". IFAC PapersOnLine 51-1, 2018, pp. 289-294.
- [5] P. Dua, F. J. Doyle, E. N. Pistikopoulos. "Multi-objective Parametric Control of Blood Glucose Concentration for Type 1 Diabetes", en la 44ª Conferencia de IEEE sobre Decisión y Control, y la Conferencia Europea de Control, 2005.
- [6] S. Diego Fernando. "¿Qué es el Control Predictivo y hacia dónde se proyecta?". Revista Especializada en Ingeniería. Universidad Abierta y a Distancia. Publicaciones e Investigación ISSN: 1900-6608 Volumen 7, 2013.
- [7] J. M. Vargas Lara. "Control predictivo multivariable: evolución histórica y conceptos". TECSUP.

ANEXOS

A continuación se adjuntan como anexos los códigos de Matlab que se han realizado:

“Modelo Bergan.m”

```
function [salida] = Modelo_Bergman(in)

% Variables de entrada en la funcion
D      = in(1);
U      = in(2);
G      = in(3);
I      = in(4);
X      = in(5);
Ej     = in(6);

%Parametros del modelo
P1=0.028/60;    %s^-1
P2=0.025/60;    %s^-1
Gb=4.5;        %mmol/L
n=(5/54)/60;   %s^-1
Ib=15;        %mU/L
Vi=12;        %L
P3=0.000013/3600; %L/(mU s^2)
```

```
% Ecuaciones diferenciales
```

```
% Gd=(-P1*G) - X*(G+Gb) + D;
Gd=(-P1*G) - (X+Ej)*(G+Gb) + D;
Id=(-n*(I+Ib)) + (U/Vi);
Xd=(-P2*X) + (P3*I);
salida=[Gd;Id;Xd];
```

“comida.m”

```
function [D] = comida(in)

% Variables de entrada en la funcion
t      = in(1);

%Parametros del modelo
Ag=0.8;    %adimensional
Tmaxi=55*60; %s
Vg=13.79; %L
Tmaxg=40*60; %s
t1=0;
Dg=0;
f=1000/180; %factor de conversion de g a mmol

%Comidas (perturbaciones) a lo largo de un día
```

```

if t>=480*60 && t<840*60 %desayuno: 55 g de HC a las 8:00
    t1=480*60;
    Dg=55*f;
end
if t>=840*60 && t<1260*60 %almuerzo: 90 g de HC a las 14:00
    t1=840*60;
    Dg=90*f;
end
if t>=1260*60 %cena 45 g de HC a las 21:00
    t1=1260*60;
    Dg=45*f;
end

% Ecuacion
D= (Dg*Ag*(t-t1)*exp(-(t-t1)/Tmaxi))/(Vg*Tmaxg^2);

```

“ejercicio.m”

```

function [Ej] = ejercicio(in)

% Variables de entrada en la funcion
t = in(1);

%Parametros del modelo
cons = 0.005/60;

%Ejercicio (perturbaciones) a lo largo de un día
if t>=1080*60 && t<1140*60 %ejercicio de 18:00 a 19:00
    Ej=cons+ 0.005/60;
else
    Ej=cons;
end

```

“función transferencia.m”

```

%Modelo linealizado y funcion de transferencia
format short

%Parametros del modelo (en segundos)
P1=0.028/60;
P2=0.025/60;
Gb=4.5;
n=(5/54)/60;
Ib=15;
Vi=12;
P3=0.000013/3600;

syms s;

%Matrices del modelo linealizado
A=[-P1, 0, -Gb; 0, -n, 0; 0, P3, -P2];
B=[0, 1, -Gb; 1/Vi, 0, 0; 0, 0, 0];
C=[1, 0, 0];
D=[0; 0; 0];
I=eye(3);

%Funcion de transferencia entera
G=C*(inv((s*I)-A))*B;
G=vpa(G, 5);

```

```
%Gm=tf(-2.6325,conv(conv([250 7],[40 1]),[54 5])) en rad/min
Gm=tf(-31.59,conv(conv([15000 7],[2400 1]),[648 1])) %en rad/s
Gd= tf(15000,[15000 7])
Gej=tf(-67500, [15000 7])
```

“control.m”

```
function [U] = control(in)

% Variable de entrada en la funcion: Gref
Gref = in(1);
G = in(2);
tiempo = in(3);

% Tiempo de muestreo (revisar el utilizado en Simulink)
persistent Int_err;
Tm=300; %Cambiar en simulink también
persistent Err_G_ant;

if(tiempo < 1e-2) %al principio el error integral es 0
    Int_err= 0;
    Err_G_ant= 0;
end

%Implementacion PID
Kp=-0.3385; Td=3014.2; Ti=12057;

Err_G= Gref-G;

Kd=Kp*Td;
Ki=Kp/Ti;
u=Kp.*Err_G + Ki.*Int_err + Kd.*(Err_G-Err_G_ant)/Tm;
Err_G_ant=Err_G;
U=u + 0.277;

Int_err=Int_err + (Err_G*Tm);

%Anti windup
if (U<=0) || (U>=100/60)
    Int_err=Int_err - (Err_G*Tm);
end

%Saturacion
if U<0
    U=0;
end
if U>100/60 %mU/s
    U=100/60;
end
```

“controllotes.m”

```
function [tinyec] = controllotes(in)

% Variables de entrada en la funcion:
Gref = in(1);
G = in(2);
tiempo = in(3);

% Tiempo de muestreo (revisar el utilizado en Simulink)
persistent Int_err;
Tm=300; %Cambiar en simulink también
```

```

persistent Err_G_ant;

if(tiempo < 1e-2)
    Int_err= 0;
    Err_G_ant= 0;
end

%Implementacion PID
Kp=-0.3385; Td=3014.2; Ti=12057;

Err_G= Gref-G;

Kd=Kp*Td;
Ki=Kp/Ti;
u=Kp.*Err_G + Ki.*Int_err + Kd.*(Err_G-Err_G_ant)/Tm;
Err_G_ant=Err_G;
U=u + 0.277;

    Int_err=Int_err + (Err_G*Tm);
%Anti windup
if (U<=0) || (U>=100/60)
    Int_err=Int_err - (Err_G*Tm);
end

%Saturacion
if U<0
    U=0;
end
if U>100/60 %mU/s
    U=100/60;
end

% %Inyeccion de la insulina "por lotes"
Uq=Tm*U; %cantidad de insulina administrada durante 5 min (tm)
Q= 100/60; %el caudal maximo de inyeccion es de 100 mU/min
tinyec=Uq/Q; %tiempo en segundos que debe estar inyectando insulina para inyectar
toda la necesaria en ese minuto

```

“inyectalotes.m”

```

%funcion a la que le entra el tiempo real y el tiempo de control calculado
%durante el que se tiene que inyectar insulina
function [Uk] = inyectalotes(in)
tinyec    = in(1);
tiempo    = in(2);
Tm=300;

t=mod(tiempo,Tm); %tiempo que lleva en el periodo de muestreo

if t<=tinyec
    Uk=100/60;
else
    Uk=0;
end

```

“feed forward.m”

```

%Calculo de Cffd y Cffej para el control feed forward
Gm=tf(-31.59,conv(conv([15000 7],[2400 1]),[648 1])); %en rad/s
Gd= tf(15000,[15000 7]);
Gej=tf(-67500,[15000 7]);

```



```

Cffd=-Gd/Gm;
Cffej=-Gej/Gm;
Cffd=tf([3.49e14 8.49e11 5.45e8 105000],[473850 221.1])
Cffej=tf([-1.575e15 -3.821e12 -2.453e9 -472500],[473850 221.1])

```

“lotes.m”

```

function [tinyec] = lotes(in)

% Variables de entrada en la funcion:
U      = in(1);

% Tiempo de muestreo (revisar el utilizado en Simulink)
Tm=300; %Cambiar en simulink también

% Saturacion
if U<0
    U=0;
end
if U>100/60 %mU/s
    U=100/60;
end

% %Inyeccion de la insulina "por lotes"
Uq=Tm*U; %cantidad de insulina administrada durante 5 min (tm)
Q= 100/60; %el caudal maximo de inyeccion es de 100 mU/min
tinyec=Uq/Q; %tiempo en segundos que debe estar inyectando insulina para inyectar
toda la necesaria en ese minuto

```

“graficas.m”

```

%Para dibujar las gráficas obtenidas

t=out.t;
G=out.G;
I=out.I;
X=out.X;
U=out.U;
Ref=out.Ref*18;
tmin=t/60;
Umin=U*60;
Gmin=60;
Gmax=180;
% plot(tmin,(out.tiempoiny/60));grid;xlabel('t(min)');ylabel('tiempo
inyección(min)');title('Tiempo de inyección de insulina');axis([0 1440 0 10]);
subplot(2,1,1);plot(tmin,G,tmin,Ref,[0,1440],[Gmin,Gmin],[0,1440],[Gmax,Gmax]);grid;xl
abel('t(min)');ylabel('Glucosa(mg/dL)');axis([0 1440 40 200]);legend('Glucemia real',
'Glucemia referencia');title('Glucemia del paciente');
subplot(2,1,2);plot(tmin,Umin);grid;xlabel('t(min)');ylabel('Insulina
inyectada(mU/min)');axis([0 1440 0 120]);title('Insulina administrada');

```