



UNIVERSIDAD
NACIONAL
DE COLOMBIA

Control óptimo de diabetes mellitus sobre un modelo no lineal del sistema glucosa-insulina

Edwar Fabián Panqueba Moreno

Universidad Nacional de Colombia
Facultad de Ciencias, Departamento de Matemáticas
Bogotá D.C., Colombia
2021

Control óptimo de diabetes mellitus sobre un modelo no lineal del sistema glucosa-insulina

Edwar Fabián Panqueba Moreno

Tesis de maestría presentada como requisito para optar al título de:
Magister en Ciencias – Matemática Aplicada

Director:
Jorge Mauricio Ruiz Vera, Ph.D.

Línea de Investigación:
Matemática Aplicada, Biología Matemática

Universidad Nacional de Colombia
Facultad de Ciencias, Departamento de Matemáticas
Bogotá D.C., Colombia
2021

Dedicado a Dios, por poner siempre en mis manos la manera de continuar adelante y superar las adversidades que van apareciendo en los proyectos que emprendo, especialmente en este.

“Ten siempre a Ítaca en la memoria, llegar allí es tu meta, más no apresures el viaje, mejor que se extienda largos años; y en tu vejez arribes a la isla con cuanto hayas ganado en el camino.”

Constantino Kavafis

Agradecimientos

A mi familia, especialmente a mis padres y mi hermana, por apoyarme moral y económicamente para poder cumplir mi sueño de realizar estudios posgraduales, y por todas sus palabras de ánimo en momentos en los que me costó continuar adelante. Al profesor Mauricio Ruiz, por creer en mis capacidades para sacar adelante este proyecto, que aunque tomó más tiempo del previsto, siempre estuvo dispuesto a brindarme su colaboración, por toda su paciencia, por compartir conmigo sus valiosos aportes y conocimientos para nutrir este trabajo, corregirme cuando fue necesario y seguirme inspirando y fascinando con las interesantes aplicaciones de las matemáticas a la solución de diferentes problemas que mostró en sus clases de análisis numérico y modelos matemáticos.

Agradezco a mis amigos quienes también compartieron conmigo una palabra de aliento y me animaron a no renunciar a mi sueño, por más difícil que se tornara el panorama. Finalmente, agradezco a la Universidad Nacional de Colombia por abrir para mí las puertas de su campus y así poder continuar con mi formación profesional, por el apoyo económico otorgado para cubrir parte de los costos de matrícula por algunos semestres, y de manera particular agradezco a la Facultad de Ingeniería por la adquisición de una licencia académica del software MATLAB[®], ya que este recurso fue de suma importancia para la programación de los algoritmos propuestos e implementados durante el desarrollo de este trabajo.

Resumen

Control óptimo de diabetes mellitus sobre un modelo no lineal del sistema glucosa-insulina

En este trabajo se plantea el problema del régimen de suministro de insulina en un paciente diabético como un problema de control óptimo, de tal manera que no se genere sobredosis o insuficiencia del medicamento hormonal. La interacción entre la glucosa e insulina se modela como un sistema no lineal de ecuaciones diferenciales ordinarias que involucra seis parámetros, cada uno de ellos asociado a un hecho biológico de la dinámica glucosa-insulina en el organismo, por ejemplo, el aumento de la insulina por presencia de glucosa o la reducción de glucosa por proceso metabólico del cuerpo. Se usan datos de niveles de glucosa e insulina para la estimación de los parámetros del modelo, por medio de la implementación computacional de un método numérico adaptado a las condiciones propias del sistema de ecuaciones.

Posteriormente, el problema de control óptimo se resuelve de manera directa empleando el método de programación cuadrática secuencial. Resultados numéricos muestran el potencial de esta propuesta en la regulación del azúcar en sangre de estos pacientes.

Palabras Clave: Control óptimo, programación no lineal, dinámica glucosa-insulina, estimación de parámetros, diabetes.

Abstract

Optimal control of diabetes mellitus on a nonlinear model of the glucose-insulin system

In this work, we formulate the supply regimen insulin problem for a diabetic patient as an optimal control problem, such that there is not overdose or insufficient medication. We model the glucose and insulin interaction as nonlinear ordinary differential equations system that involves six parameters, each one of these parameters are associated to a biological factor of glucose-insulin dynamic, for example, the growth of insulin level due to the presence of glucose, or reduction of glucose as a result of metabolic process in the body. We use levels of glucose and insulin data in order to estimate the model parameters, implementing numerical methods computationally.

In the last part of the work, we solve the optimal control problem numerically, employing Sequential Quadratic Programming (SQP) method. Numerical results show the potential of this proposal in sugar regulation in the blood of these patients.

Keywords *Optimal control, nonlinear programming, glucose-insulin dynamic, parameters estimation, diabetes.*

Contenido

Introducción	1
1 La diabetes: un problema de salud pública	4
1.1 Planteamiento del problema	4
1.2 Funcionamiento del sistema glucosa-insulina	5
1.3 Caracterización de la diabetes	7
1.4 Tratamiento de la diabetes	7
1.5 Hipoglucemia, hiperglicemia y otras complicaciones de la diabetes	8
1.6 Estado del arte de algunos modelos matemáticos de la dinámica glucosa-insulina	11
1.6.1 Modelo de Ackerman	11
1.6.2 Modelo no lineal de Huang	14
1.6.3 Modelo de Bergman	15
1.6.4 Otro modelo	17
1.7 Objetivos	17
1.7.1 Objetivo general	17
1.7.2 Objetivos específicos	17
2 Modelo matemático de la dinámica glucosa-insulina	19
2.1 Descripción del modelo matemático	19
2.2 Análisis matemático del sistema	21
2.3 Solución numérica del sistema de ecuaciones	24
2.3.1 Simulación computacional	24
3 Estimación de parámetros	28
3.1 Contextualización	28
3.2 Formulación del problema de mínimos cuadrados	29
3.3 Estimación lineal y no lineal	30
3.4 Método de Gauss-Newton	32
3.5 Método de Levenberg-Marquardt	34
3.6 Estimación de parámetros en ecuaciones diferenciales ordinarias [2]	35
3.7 Cálculo del valor inicial de los parámetros	38
3.8 Experimentación numérica	39
3.8.1 Experimento 1: paciente sano	43

3.8.2	Experimento 2: paciente diabético	45
4	Control óptimo	49
4.1	Formulación general del problema de control óptimo	49
4.2	Existencia de una solución para el problema de control óptimo	52
4.3	Solución numérica del problema de control óptimo	56
4.3.1	Método directo: formulación del problema PNL	57
5	Resultados numéricos	67
5.1	Tratamiento de la diabetes y las bombas de insulina	67
5.1.1	Tipos de insulina	67
5.1.2	Tratamientos insulínicos	69
5.2	Aspectos generales de las simulaciones	71
5.3	Simulación de los distintos escenarios	73
5.3.1	Escenario 1	74
5.3.2	Escenario 2	82
5.3.3	Escenario 3	89
5.3.4	Escenario 4	95
6	Conclusiones y Trabajo futuro	103
6.1	Conclusiones	103
6.2	Trabajo futuro	104
	Bibliografía	106
A	Método de Runge-Kutta de cuarto orden	110
B	Códigos de programación: métodos numéricos y simulaciones	113
B.1	Códigos del Capítulo 2	113
B.2	Códigos del Capítulo 3	116
B.3	Códigos del Capítulo 5	134

Lista de Figuras

- 1.1 Sistema Glucosa - Insulina. 6
- 1.2 Esquema funcionamiento y producción de insulina. Adaptado de [20]. 6
- 2.1 Simulación de la dinámica glucosa-insulina, a partir el modelo (2.3). 27
- 3.1 Diagrama de flujo del proceso de estimación de parámetros para el modelo de la dinámica glucosa-insulina (2.3). 42
- 3.2 Resultados gráficos obtenidos de la estimación de parámetros para la dinámica de la glucosa de un paciente sano. 43
- 3.3 Resultados gráficos obtenidos de la estimación de parámetros para la dinámica de la insulina de un paciente sano. 44
- 3.4 Gráfica del número de iteraciones vs. error de aproximación del método Gauss-Newton aplicado a la estimación de parámetros de la dinámica glucosa-insulina. 45
- 3.5 Resultados gráficos obtenidos de la estimación de parámetros para la dinámica de la glucosa de un paciente diabético. 46
- 3.6 Resultados gráficos obtenidos de la estimación de parámetros para la dinámica de la insulina de un paciente diabético. 46
- 3.7 Gráfica del número de iteraciones vs. error de aproximación del método Gauss-Newton aplicado a la estimación de parámetros de la dinámica glucosa-insulina de un paciente diabético. 48
- 4.1 Esquematización de los métodos directos para resolver problemas de control óptimo. 58
- 4.2 Ilustración del método directo secuencial, adaptada de [17]. En la gráfica, x_0 y x_T (puntos azules claro) representan las condiciones iniciales y finales de la variable de estado, las cuales deben satisfacerse. La línea continua azul aproxima la función de control $u(t)$ en cada uno de los subintervalos, y la curva naranja describe la dinámica del sistema asociado a las variables de estado y control. 60
- 4.3 Diagrama de flujo propuesto para la solución del problema de control óptimo, formulado en (4.8). 66
- 5.1 Diferentes tipos de insulina según su tiempo de acción. Tomada de [45]. 68
- 5.2 Tratamiento de la diabetes a través de inyecciones. Tomadas de [32]. 69
- 5.3 Tratamiento de la diabetes a través de una bomba de insulina. Tomadas de [32]. 71

5.4	Conjunto de datos para la estimación de parámetros del modelo ajustado a la dinámica de un paciente diabético tipo 1.	75
5.5	Gráficas de las dinámicas de glucosa e insulina, junto con su control óptimo para un paciente diabético tipo 1 que se alimenta según el escenario 1, con glucosa deseada según la opción 1.	75
5.6	Funcional objetivo para glucosa deseada según forma 1, en cada iteración. Nfe: 8101, Nitr: 27, Time: 62.4 s, Fval: 2.734×10^2	76
5.7	Gráficas de las dinámicas de glucosa e insulina, junto con su control óptimo para un paciente diabético tipo 1 que se alimenta según el escenario 1, con glucosa deseada según la opción 2.	76
5.8	Funcional objetivo para glucosa deseada según forma 2, en cada iteración. Nfe: 10137, Nitr: 34, Time: 71.8 s, Fval: 5.134×10^2	77
5.9	Conjunto de datos para la estimación de parámetros del modelo ajustado a la dinámica de un paciente diabético tipo 2.	78
5.10	Gráficas de las dinámicas de glucosa e insulina, junto con su control óptimo para un paciente diabético tipo 2 que se alimenta según el escenario 1, con glucosa deseada según la opción 1.	79
5.11	Funcional objetivo para glucosa deseada según forma 1, en cada iteración. Nfe: 6361, Nitr: 21, Time: 41.9 s, Fval: 4.957×10^2	80
5.12	Gráficas de las dinámicas de glucosa e insulina, junto con su control óptimo para un paciente diabético tipo 2 que se alimenta según el escenario 1, con glucosa deseada según la opción 2.	81
5.13	Funcional objetivo para glucosa deseada según forma 2, en cada iteración. Nfe: 8122, Nitr: 28, Time: 61.9 s, Fval: 9.618×10^2	81
5.14	Conjunto de datos para la estimación de parámetros del modelo ajustado a la dinámica de un paciente diabético tipo 1.	83
5.15	Gráficas de las dinámicas de glucosa e insulina, junto con su control óptimo para un paciente diabético tipo 1 que se alimenta según el escenario 2, con glucosa deseada según la opción 1.	83
5.16	Funcional objetivo para glucosa deseada según forma 1, en cada iteración. Nfe: 7809, Nitr: 26, Time: 24.1 s, Fval: 1.621×10^2	84
5.17	Gráficas de las dinámicas de glucosa e insulina, junto con su control óptimo para un paciente diabético tipo 1 que se alimenta según el escenario 2, con glucosa deseada según la opción 2.	84
5.18	Funcional objetivo para glucosa deseada según forma 2, en cada iteración. Nfe: 8410, Nitr: 28, Time: 31.5 s, Fval: 3.724×10^2	85
5.19	Conjunto de datos para la estimación de parámetros del modelo ajustado a la dinámica de un paciente diabético tipo 2.	85

5.20	Gráficas de las dinámicas de glucosa e insulina, junto con su control óptimo para un paciente diabético tipo 2 que se alimenta según el escenario 2, con glucosa deseada según la opción 1.	86
5.21	Funcional objetivo para glucosa deseada según forma 1, en cada iteración. Nfe: 6940, Nitr: 23, Time: 23.5 s, Fval: 1.971×10^2	86
5.22	Gráficas de las dinámicas de glucosa e insulina, junto con su control óptimo para un paciente diabético tipo 2 que se alimenta según el escenario 2, con glucosa deseada según la opción 2.	87
5.23	Funcional objetivo para glucosa deseada según forma 2, en cada iteración. Nfe: 8679, Nitr: 29, Time: 35.5 s, Fval: 4.292×10^2	87
5.24	Conjunto de datos para la estimación de parámetros del modelo ajustado a la dinámica de un paciente diabético tipo 1.	90
5.25	Gráficas de las dinámicas de glucosa e insulina, junto con su control óptimo para un paciente diabético tipo 1 que se alimenta según el escenario 3, con glucosa deseada según la opción 1.	90
5.26	Funcional objetivo para glucosa deseada según forma 1, en cada iteración. Nfe: 7812, Nitr: 26, Time: 28.14 s, Fval: 9.471×10^2	91
5.27	Gráficas de las dinámicas de glucosa e insulina, junto con su control óptimo para un paciente diabético tipo 1 que se alimenta según el escenario 3, con glucosa deseada según la opción 2.	91
5.28	Funcional objetivo para glucosa deseada según forma 2, en cada iteración. Nfe: 8971, Nitr: 31, Time: 35.4 s, Fval: 1.389×10^3	92
5.29	Conjunto de datos para la estimación de parámetros del modelo ajustado a la dinámica de un paciente diabético tipo 2.	92
5.30	Gráficas de las dinámicas de glucosa e insulina, junto con su control óptimo para un paciente diabético tipo 2 que se alimenta según el escenario 3, con glucosa deseada según la opción 1.	93
5.31	Funcional objetivo para glucosa deseada según forma 1, en cada iteración. Nfe: 7321, Nitr: 24, Time: 38.6 s, Fval: 8.744×10^2	93
5.32	Gráficas de las dinámicas de glucosa e insulina, junto con su control óptimo para un paciente diabético tipo 2 que se alimenta según el escenario 3, con glucosa deseada según la opción 2.	94
5.33	Funcional objetivo para glucosa deseada según forma 2, en cada iteración. Nfe: 11895, Nitr: 41, Time: 40.2 s, Fval: 1.293×10^3	94
5.34	Conjunto de datos para la estimación de parámetros del modelo ajustado a la dinámica de un paciente diabético tipo 1.	96
5.35	Gráficas de las dinámicas de glucosa e insulina, junto con su control óptimo para un paciente diabético tipo 1 que se alimenta según el escenario 4, con glucosa deseada según la opción 1.	97

5.36	Funcional objetivo para glucosa deseada según forma 1, en cada iteración. Nfe: 8104, Nitr: 27, Time: 25.2 s, Fval: 1.901×10^2	97
5.37	Gráficas de las dinámicas de glucosa e insulina, junto con su control óptimo para un paciente diabético tipo 1 que se alimenta según el escenario 4, con glucosa deseada según la opción 2.	98
5.38	Funcional objetivo para glucosa deseada según forma 2, en cada iteración. Nfe: 9855 Nitr: 34, Time: 30.8 s, Fval: 2.913×10^2	98
5.39	Conjunto de datos para la estimación de parámetros del modelo ajustado a la dinámica de un paciente diabético tipo 2.	99
5.40	Gráficas de las dinámicas de glucosa e insulina, junto con su control óptimo para un paciente diabético tipo 2 que se alimenta según el escenario 4, con glucosa deseada según la opción 1.	99
5.41	Funcional objetivo para glucosa deseada según forma 1, en cada iteración. Nfe: 8441, Nitr: 28, Time: 25.1 s, Fval: 2.154×10^2	100
5.42	Gráficas de las dinámicas de glucosa e insulina, junto con su control óptimo para un paciente diabético tipo 2 que se alimenta según el escenario 4, con glucosa deseada según la opción 2.	100
5.43	Funcional objetivo para glucosa deseada según forma 2, en cada iteración. Nfe: 8977, Nitr: 31, Time: 32.2 s, Fval: 3.802×10^2	101

Lista de Tablas

- 1.1 Concentraciones de glucosa normales para personas sanas y dabéticas [44]. 8
- 1.2 Concentraciones de insulina saludables, en diferentes momentos del día [14]. 9
- 2.1 Ecuaciones del método de Runge-Kutta para la solución numérica de (2.3). 25
- 3.1 Símbolo, descripción y unidades de los parámetros usados en el sistema de ecuaciones diferenciales (2.3). 29
- 3.2 Resultados de la estimación de parámetros para el sistema (2.3) por el método Gauss-Newton para la dinámica de un paciente sano. El vector β_0 hace referencia al vector que contiene los valores iniciales de los parámetros para resolver el problema de estimación. El vector $\hat{\beta}$ contiene los parámetros estimados una vez se logra la convergencia del algoritmo de optimización. 43
- 3.3 Resultados de la estimación de parámetros para el sistema (2.3) por el método Gauss-Newton para la dinámica de un paciente diabético. 45
- 5.1 Clasificación de los tipos de insulina según su tiempo de acción. 68
- 5.2 Resultados de la estimación de parámetros para el sistema (2.3) de un paciente diabético tipo 1 que se alimenta según el escenario 1. β_0 es el vector que contiene los valores iniciales de los parámetros, necesario para resolver el problema de estimación. $\hat{\beta}$ es el vector que contiene los parámetros estimados una vez se logra la convergencia de los algoritmos de estimación. 74
- 5.3 Resultados de la estimación de parámetros para el sistema (2.3) de un paciente diabético tipo 2 que se alimenta según el escenario 1. 78
- 5.4 Resultados de la estimación de parámetros para el sistema (2.3) de un paciente diabético tipo 1 que se alimenta según el escenario 2. 82
- 5.5 Resultados de la estimación de parámetros para el sistema (2.3) de un paciente diabético tipo 2 que se alimenta según el escenario 2. 85
- 5.6 Resultados de la estimación de parámetros para el sistema (2.3) de un paciente diabético tipo 1 que se alimenta según el escenario 3. 89
- 5.7 Resultados de la estimación de parámetros para el sistema (2.3) de un paciente diabético tipo 2 que se alimenta según el escenario 2. 92
- 5.8 Resultados de la estimación de parámetros para el sistema (2.3) de un paciente diabético tipo 1 que se alimenta según el escenario 4. 96

5.9	Resultados de la estimación de parámetros para el sistema (2.3) de un paciente diabético tipo 2 que se alimenta según el escenario 4.	99
------------	---	----

Introducción

La biología matemática o biomatemática, nace del descubrimiento en la biología de toda una serie de fenómenos novedosos que resultan susceptibles de ser tratados y descritos en el lenguaje de las matemáticas. Científicos como Rashevsky, Von Bertalanffy y Turing iniciaron en la construcción de una estrecha relación entre las ciencias de la vida y el formalismo matemático, y con el apoyo de los ordenadores, se consolidó el estudio cuantitativo de diversos problemas biológicos [31]. En la actualidad, son numerosos los problemas de biología y biomedicina que se benefician del análisis matemático, por lo que la biomatemática ha contribuido y sigue contribuyendo al avance de la biología como ciencia [31].

En el campo de la biomedicina, desde la década del 60 empezaron a desarrollarse los primeros modelos matemáticos que buscaban describir cuantitativamente la relación existente entre la glucosa y la insulina, y de esta manera entender el proceso metabólico de la glucosa en el cuerpo humano. Los primeros modelos propuestos se componen de ecuaciones diferenciales lineales y posteriormente se han propuesto otros que contemplan ecuaciones diferenciales no lineales y en derivadas parciales [10]. Todos estos esfuerzos redundan en el objetivo de crear y mejorar tratamientos médicos útiles en el manejo de una enfermedad que ha causado la muerte de aproximadamente 422 millones de personas en el mundo [38], conocida como diabetes.

La diabetes es una patología que se caracteriza por una elevada cantidad de glucosa que permanece circulando por el torrente sanguíneo, debido a que el páncreas deja de producir insulina o produce muy poca cantidad. El principal tratamiento de esta enfermedad consiste en la administración de insulina fabricada artificialmente en laboratorios farmacéuticos, intentando así imitar el funcionamiento del páncreas. Actualmente, la inyección de insulina en pacientes diabéticos puede hacerse a través de bombas de infusión continua, pero estas bombas aún dependen de la manipulación del paciente, que sigue siendo sobre quien recae la responsabilidad de realizar los cálculos de la cantidad de insulina a suministrar y aceptar o corregir las sugerencias hechas por el dispositivo. Errores en estos cálculos pueden ocasionar problemas de salud que afectan considerablemente la calidad de vida del paciente, por lo que se busca dotar a estos dispositivos de un algoritmo capaz de estimar la cantidad de insulina necesaria en cada momento para mantener la concentración de glucosa dentro de un rango objetivo, usando la menor cantidad de medicamento que sea posible [10]. Lo anterior se puede traducir en un planteamiento conocido en matemáticas como un problema de control óptimo, y en este trabajo se presenta una alternativa de solución a este problema.

Automatizar y mejorar el manejo de la diabetes es el principal objetivo de desarrollar lo que se denominaría un *páncreas artificial*.

Para la construcción de una propuesta de solución al problema de control óptimo, se siguió una ruta de trabajo a partir de la cual se producen cada uno de los seis capítulos en los que se organiza el presente escrito. El primero presenta una contextualización sobre la diabetes, haciendo énfasis en las características y particularidades de la enfermedad, además de establecer el planteamiento del problema de investigación, mostrar algunos antecedentes en los que también se han desarrollado propuestas de solución al problema de control óptimo y se establecen los objetivos de la investigación. En el segundo capítulo se introduce el modelo matemático, compuesto de un sistema de ecuaciones diferenciales ordinarias no lineal, que representa el proceso metabólico de la glucosa y la insulina, denominado el *sistema dinámico de glucosa-insulina*. Los modelos matemáticos del metabolismo de la glucosa juegan un papel fundamental en el planteamiento de los problemas de control óptimo de este tipo, como se verá en el estudio teórico que estructura este capítulo y los posteriores. Aquí se emplean elementos del análisis matemático para demostrar la existencia y unicidad del sistema de ecuaciones. Posteriormente se selecciona e implementa un método que permita el hallazgo de una solución numérica, ya que por su naturaleza no es posible el cálculo de una solución analítica.

En el tercer capítulo se plantea el problema de la estimación de parámetros para el sistema dinámico glucosa-insulina. Este aspecto resulta ser de gran importancia, debido a que posibilita el ajuste del modelo matemático al comportamiento de los niveles de glucosa e insulina que registre cada paciente particularmente, y además aporta información con la que se pueden predecir concentraciones de los momentos del día en los que por diversas situaciones no es posible medirla de forma directa, posibilitando así la optimización de la infusión de insulina con base a ellos. Aquí se presenta teóricamente el método utilizado para la estimación de parámetros, y se incluyen experimentos numéricos que muestran el potencial de la propuesta. Todos los códigos de programación empleados para la experimentación numérica son de autoría propia, se encuentra alojados y debidamente comentados en los anexos que se ponen a disposición del lector en este trabajo.

En el cuarto capítulo se plantea el problema de control óptimo, donde también se hace uso de elementos teóricos de las matemáticas para demostrar la existencia de la solución al problema. Posteriormente se presenta un estudio teórico del método numérico seleccionado para el hallazgo de una solución, que en el siguiente capítulo será implementado computacionalmente, con el fin de analizar su efectividad y alcance.

Finalmente, en el quinto capítulo se incluyen algunas simulaciones en las que se resuelve el problema de control óptimo en cuatro distintos escenarios de alimentación que eventualmente pueden llegar darse en algún paciente, sea este diabético tipo 1 o tipo 2, en los cuales se tiene en cuenta diferentes horarios de ingesta de los alimentos y la cantidad de carbohidratos que son consumidos, además de probar también con dos tipos distintos de selección de la

concentración de glucosa deseada, para posteriormente presentar, analizar y discutir los resultados numéricos obtenidos. La posibilidad de escoger la concentración de glucosa deseada resultará también un factor importante a tener en cuenta al momento del planteamiento del problema de control óptimo, por las ventajas que podría llegar a representar sobre la cantidad de insulina a ser suministrada por el dispositivo al paciente. Para llevar a cabo los experimentos que en este capítulo se produjeron, se hizo uso un paquete de optimización disponible en el software MATLAB[®] por la confianza que proporciona usar una herramienta que ha sido sometida a numerosas pruebas técnicas y que cuenta con ventajas que tomaría bastante tiempo programar de manera independiente. En el capítulo seis, se presentan algunas conclusiones y recomendaciones que podría ser útil tener en cuenta para el desarrollo de futuros trabajos.

1

La diabetes: un problema de salud pública

La diabetes es una patología que actualmente causa la muerte de millones de personas a nivel mundial. Cada año, el número de diagnósticos de la enfermedad se incrementa con gran preocupación, lo que enciende las alarmas de los organismos de salud buscando evitar que su crecimiento continúe, debido a las complicaciones de diversa índole que esta situación atañe a la sociedad. En este capítulo, se presentan con algún grado de detalle la problemática de la diabetes, según informes y cifras presentadas por entidades de investigación en medicina, encontradas en [20] y [38]. Se presenta de manera general una descripción del sistema fisiológico glucosa-insulina, que servirá de apoyo al sustento del modelo matemático seleccionado en este trabajo, además de algunos otros aspectos estrechamente ligados con la problemática objeto de estudio.

1.1. Planteamiento del problema

La diabetes es una enfermedad crónica, y ha sido considerada como una de las mayores emergencias mundiales del siglo XXI, debido a que sus cifras de diagnóstico en personas jóvenes ha aumentado con gran preocupación en los últimos años.

Según estadísticas reveladas por la Organización Mundial de la Salud, hasta el año 2016, esta enfermedad era padecida por 422 millones de personas entre los 20 y los 79 años de edad, cifra que aumentó en 314 millones comparada con la cantidad de personas enfermas que se registraron en 1980 [38].

En 2012, la diabetes fue la causa directa de 1.5 millones de muertes a nivel mundial, mientras que otras 2.2 millones fueron provocadas por enfermedades cardiovasculares y otras afecciones asociadas a la diabetes, como las fallas renales, alteraciones en la tiroides y la obesidad.

Para el caso de Colombia, 2.1 millones de personas padecían diabetes, 6050 muertes fueron causa directa de la enfermedad (que representa el 3% de todas las muertes registradas), y otras 12590 fueron atribuibles a patologías asociadas a la diabetes (cifra que representa el 28% del total de muertes) en el año 2016. Este país ocupa el segundo lugar con el mayor número de habitantes enfermos de diabetes, entre los países de Suramérica [46].

La diabetes y sus complicaciones conllevan a importantes pérdidas económicas para las personas que la padecen, sus familias y los sistemas de salud nacionales, por los costos médicos asociados al tratamiento de la enfermedad, situación atribuida a la necesidad de un mayor uso de los servicios de salud tanto hospitalarios como ambulatorios, los elevados costos en medicamentos para el tratamiento de la enfermedad, pérdida de la productividad económica del paciente y el apoyo constante durante mucho tiempo que este requiere para superar complicaciones asociadas con la diabetes, especialmente aquellas relacionadas con la pérdida de visión, ceguera y amputación de extremidades. Por lo anterior, los tratamientos de diabetes generan un gasto entre el 5 % y 20 % del total del gasto de la nación [38], y se considera un problema de salud pública que afecta de manera importante la esperanza de vida de cualquier persona.

1.2. Funcionamiento del sistema glucosa-insulina

Todas las células del cuerpo necesitan energía para estar activas y así aumentar las funciones vitales como el latido cardiaco, los movimientos digestivos y la respiración, por nombrar algunas, además de mantener la temperatura corporal y los movimientos musculares [15].

La glucosa es la principal fuente de la energía para el cuerpo humano, y esta ingresa al organismo con los alimentos. Con la digestión, se ponen en acción una cadena de procesos químicos que convierten los alimentos en nutrientes y estos a su vez en elementos más pequeños como la glucosa. Los alimentos transitan por el tubo digestivo y, al llegar al intestino delgado, la glucosa pasa a la sangre. La sangre se encarga de transportar la glucosa a los órganos como el hígado en el cual se almacena, al cerebro y a otras células del cuerpo donde se utiliza inmediatamente como energía para llevar a cabo distintas funciones [48].

Para que la glucosa pueda entrar en las células y ser utilizada como energía, se requiere de la mediación de la insulina. La insulina es una hormona producida por las células beta del páncreas, la cual es secretada una vez se detecta la presencia de glucosa en la sangre proveniente del proceso alimenticio. Su objetivo es actuar como una “llave que, encajada en la cerradura”, abre la puerta de la célula para el paso de la glucosa hacia su interior [48].

El uso de la glucosa hecho por las células, hace que sus niveles de concentración en la sangre se reduzcan hasta alcanzar una concentración normal o de equilibrio. Sin embargo, en situaciones donde el organismo exige un aporte considerable de energía, la concentración de glucosa en sangre puede llegar a ser inferior al nivel normal. En este caso, el páncreas recibe una señal para que las células alfa reaccionen secretando la hormona glucagón, cuya función es provocar la liberación de la glucosa reservada por el hígado hasta que el sistema alcance de nuevo sus niveles de concentración normal [48, 15].

El páncreas y el hígado trabajan conjuntamente utilizando un sistema de equilibrio y control para mantener los niveles de glucosa e insulina en niveles saludables. Así pues, el cuerpo

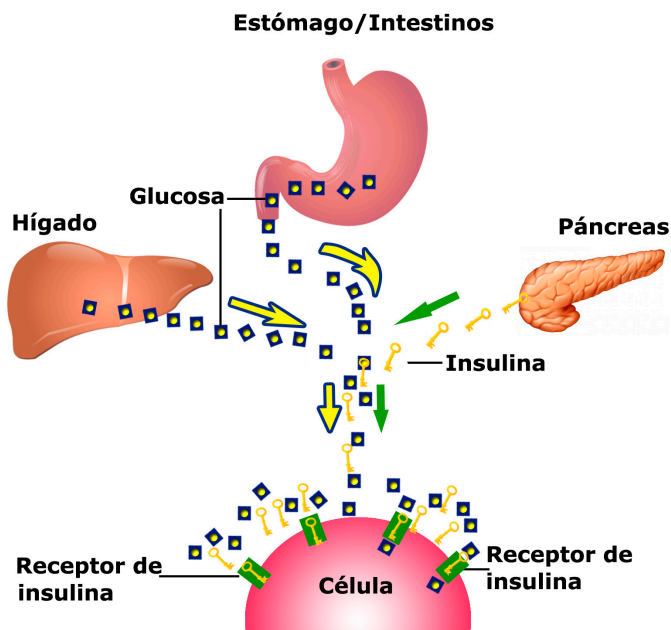


Figura 1.1: Sistema Glucosa - Insulina.

humano debe tener la capacidad de mantener un estricto control sobre la glucosa. Según la *Asociación Americana de Diabetes* (ADA, por sus siglas en inglés), los niveles de glucosa en el organismo, antes de comer, están entre 90 y 130 mg/dl, y después de comer deben ser menores a 180 mg/dl en personas sanas. Cuando se presenta un mal funcionamiento de este sistema, es muy difícil mantener estos niveles en el margen apropiado [44, 5], y este mal funcionamiento es básicamente lo que se conoce como diabetes.

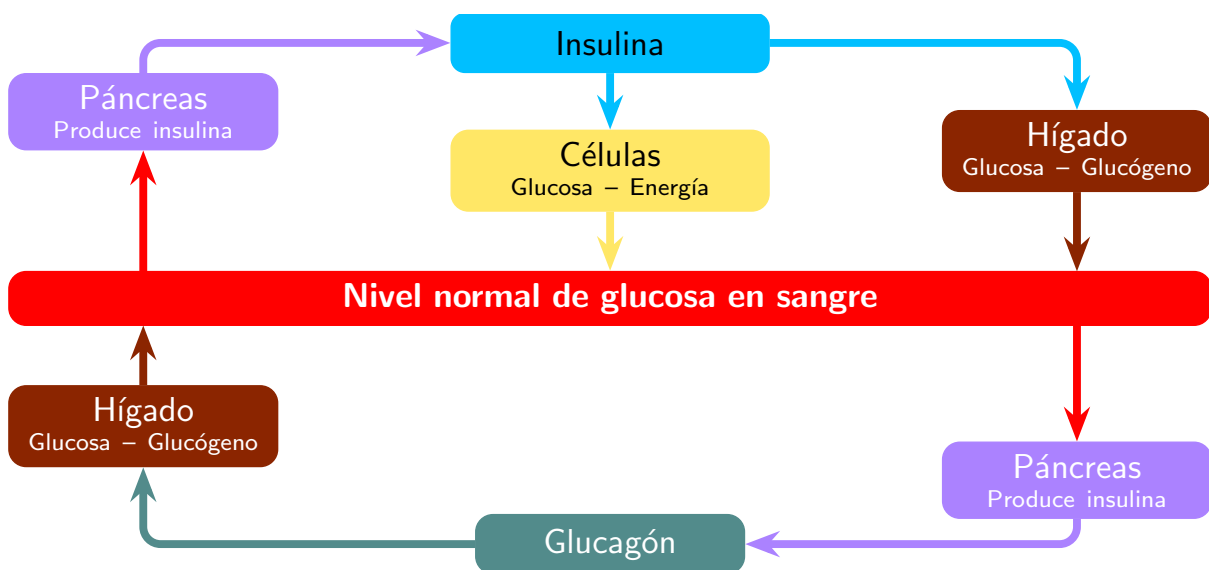


Figura 1.2: Esquema funcionamiento y producción de insulina. Adaptado de [20].

1.3. Caracterización de la diabetes

La diabetes es un trastorno endocrino-metabólico que se desencadena cuando el páncreas no produce suficiente insulina o no la produce, o cuando el organismo no puede utilizar con eficacia la insulina que produce, es decir, el sistema metabólico del cuerpo no puede mantener un equilibrio entre los niveles de glucosa e insulina de manera natural. La falta o ineficacia de la insulina en personas con diabetes, significa que la glucosa continúa circulando en la sangre, y con el tiempo, los altos niveles de glucosa (conocida como hiperglucemia) causan daño a muchos órganos y tejidos del cuerpo [5].

Los síntomas que presenta una persona que padece diabetes son: fatiga y aumento del apetito, sed constante, micción frecuente, náuseas, presencia de llagas que demoran en desaparecer, piel seca, visión borrosa, entre otros [38].

La OMS reconoce 3 tipos de diabetes, los cuales se describen brevemente a continuación [5, 38]:

- **Diabetes tipo 1:** Es una enfermedad autoinmune y metabólica, caracterizada por una destrucción selectiva de las células β del páncreas provocando que este no produzca insulina, lo que da lugar a un exceso de glucosa permanente en la sangre. Por lo general suele manifestarse en edades tempranas, antes de los 30 años. Solo 1 de cada 20 personas diabéticas tienen diabetes tipo 1.
- **Diabetes tipo 2:** Es el tipo más común de diabetes. Se caracteriza porque el organismo del paciente que la padece sí produce insulina, pero no se produce la suficiente, o no se puede aprovechar la que se produce (resistencia a la insulina), por lo que la glucosa no queda bien distribuida por los diferentes sistemas del organismo. Una de sus principales causas es el aumento en el peso corporal excesivo y la poca actividad física. Hasta hace poco tiempo esta patología solo se presentaba en adultos, pero actualmente también se está manifestando en niños, según la OMS.
- **Diabetes gestacional:** Se caracteriza por el aumento de los niveles de glucosa en sangre durante el embarazo. Las mujeres con diabetes gestacional corren mayor riesgo de sufrir complicaciones durante el embarazo y el parto. Además tanto ellas como sus hijos corren riesgo de padecer diabetes tipo 2 en un futuro.

1.4. Tratamiento de la diabetes

La deficiencia de insulina, así como la imposibilidad de aprovechar aquella que se produce naturalmente, requiere que la administración de insulina exógena sea necesaria. Así pues, el tratamiento de los pacientes que padecen de diabetes se realiza a través del ejercicio, una dieta sana y el aporte externo de insulina de manera artificial que se inyecta en la sangre [44]. Estas inyecciones se realizan con diferentes dispositivos, jeringas y bombas. De acuerdo con Alonso

[4, p.7], uno de los problemas que se presenta en todos los dispositivos de administración, es que estos deben ser controlados por el paciente, lo que implica que las dosis de la hormona no siempre se suministre cuando se necesita, o que las dosis sean muy altas o muy bajas para los procesos metabólicos que se llevan a cabo. Es por esta razón que solucionar este problema requiere la creación de un tipo de tratamiento, en el cual el dispositivo actúe de forma autoregulada y el paciente no necesite estar atento de proporcionar las cantidades de la hormona que considera correctas para el manejo de su enfermedad. En el capítulo 5 de este trabajo, se profundiza con más detalle en los aspectos relacionados con el tratamiento de la diabetes aquí mencionados.

1.5. Hipoglucemia, hiperglicemia y otras complicaciones de la diabetes

La glucemia es la medida de concentración de glucosa en sangre, suero o plasma sanguíneo. Esta medida se presentan usualmente en unidades de concentración molar (mmol/L), o en unidades de concentración de masa (mg/dL) [44].

El peso molecular de la glucosa ($C_6H_{12}O_6$), es de 180 g aproximadamente, lo que quiere decir que 1 mol de esta molécula tiene una masa de 180 g. Realizando las conversiones de mol a mmol y de g a mg, se obtiene que 1 mmol tiene una masa de 180 mg. Finalmente, tomando como concentración 1 mmol/L y realizando la conversión a unidades de mg/dL, se deduce que 1 mmol/L es equivalente a 180 mg/dL, lo cual establece el factor de conversión para presentar medidas de concentración de glucosa en cualquiera de las dos unidades. A continuación se presentan las concentraciones de glucosa que se consideran normales o saludables, tanto para personas sanas como para diabéticas, establecidas por la ADA.

	Paciente Sano		Paciente diabético	
	Concent. Masa	Concent. Molar	Concent. Masa	Concent. Molar
Antes de comer	70 - 100 mg/dL	3.89 - 5.6 mmol/L	70 - 130 mg/dL	3.89 - 7.2 mmol/L
2 horas después de comer	70 - 140 mg/dL	3.89 - 7.8 mmol/L	Menos de 180 mg/dL	Menos de 10 mmol/L

Tabla 1.1: Concentraciones de glucosa normales para personas sanas y dabéticas [44].

Por otra parte, las concentraciones de insulina pueden ser reportadas en unidades de concentración molar (pmol/L) o en número de unidades por unidad de volumen (μ U)/mL. En 1987, la OMS determinó que un gramo de insulina es equivalente a 26000 unidades de la hormona (esta equivalencia es llamada la potencia)[23]. Sabiendo que el peso molecular de la hormona es 5808 g/mol, se deduce que la masa de una unidad de la hormona es de $3,85 \times 10^{-5}$ g, y por lo tanto una unidad equivale a $6,62 \times 10^{-9}$ moles. Finalmente, haciendo la conversión entre unidades y microunidades (μ U) se tiene que 1 μ U es equivalente a $6,62 \times 10^{-15}$ moles, y de aquí se concluye que 1μ U/mL es equivalente a $6,62 \times 10^{-12}$ moles/L, o 6,62 pmol/L. Sin embargo, según [28], este cálculo puede ser incorrecto al tomar como base el valor de una

masa molar impreciso (la insulina puede estar impura al tener presencia de agua y sales), o por aplicación redondeos mal elaborados, por lo que se considera que el factor de conversión correcto, además de ser el más utilizado en el ámbito médico es $1 \mu\text{U}/\text{mL} = 6,00 \text{ pmol}/\text{L}$.

A continuación se presenta la tabla de valores saludables o normales para la concentración de insulina, en diferentes momentos del día (ver tabla 1.2).

	Nivel de Insulina ($\mu\text{U}/\text{mL}$)	Nivel de Insulina (pmol/L)
En ayunas	$< 25 \mu\text{U}/\text{mL}$	$< 174 \text{ pmol}/\text{L}$
30 minutos después de comer	30 - 230 $\mu\text{U}/\text{mL}$	208 - 1597 pmol/L
1 hora después de comer	18 - 276 $\mu\text{U}/\text{mL}$	125 - 1917 pmol/L
2 horas después de comer	16 - 166 $\mu\text{U}/\text{mL}$	111-1153 pmol/L
3 o más horas después de comer	$< 25 \mu\text{U}/\text{mL}$	$< 174 \text{ pmol}/\text{L}$

Tabla 1.2: Concentraciones de insulina saludables, en diferentes momentos del día [14].

La glucosa baja o hipoglucemia se produce cuando hay exceso de insulina en el organismo. El exceso de insulina traslada la mayor parte de la glucosa desde el torrente sanguíneo a las células, la cual se consume rápidamente y provoca que el nivel de glucosa sea de $55 \text{ mg}/\text{dL}$ o menos. Cuando la glucosa escasea, el cerebro y las células de tejido nervioso no pueden ser nutridas adecuadamente, y de no corregirse esta situación, el paciente puede experimentar síntomas como visión borrosa, dificultad en el habla, confusión mental, o incluso puede desencadenar un coma hipoglucémico (pérdida del conocimiento). Entre las causas de la hipoglucemia se encuentran, por nombrar algunas, la administración errónea de las cantidades de insulina en pacientes diabéticos, poca ingesta de carbohidratos en las comidas, malos hábitos con los horarios para el consumo de alimentos durante el día y excesos en las rutinas de actividad física [36].

Por el contrario, la hiperglucemia (glucosa alta) se presenta cuando el nivel de glucosa en sangre está por encima de $180 \text{ mg}/\text{dL}$, debido a que no hay presencia en el organismo de la cantidad de insulina necesaria para cubrir la cantidad de glucosa existente en ese momento. Los niveles de glucosa pueden elevarse demasiado por diferentes motivos, entre ellos se destacan el consumo de algunos alimentos que pueden producir un aumento significativo en los niveles de glucosa si no se recibe la cantidad de necesaria de insulina para compensarlo, algunas enfermedades o infecciones como los resfriados, virus estomacales, el estrés, o la administración de insulina que resulta ineficaz al perder sus propiedades por no ser conservada en condiciones apropiadas para su posterior uso [5, 32]. La hiperglucemia no tratada puede derivar en una complicación más seria conocida como cetoacidosis diabética (CAD)¹, que podría causar un coma diabético y la muerte del paciente [32].

¹La cetoacidosis diabética es una afección que ocurre cuando el organismo se ve obligado a quemar grasas para obtener energía, proceso del cual se producen gran cantidad de desechos (llamados cetonas), que se acumulan en la sangre y alteran su pH natural, volviéndola ácida y potencialmente tóxica para el cuerpo.

Uno de los objetivos principales del manejo adecuado de la diabetes es evitar la glucosa alta y tratarla adecuadamente tan pronto sea detectada, porque las personas con diabetes tienen mayor riesgo de desarrollar numerosos problemas de salud incapacitantes y potencialmente mortales. Niveles de glucosa en sangre considerablemente altos pueden desarrollar [20, p.29]:

- ❑ **Enfermedades en los ojos:** Muchas personas con diabetes padecen de rinopatía, que puede dañar la visión y provocar ceguera. La enfermedad consiste básicamente en el deterioro de los vasos sanguíneos que irrigan la retina del ojo, lo que genera pérdida paulatina de la visión. Los persistentes niveles altos de glucosa en sangre son la principal causa de rinopatía.
- ❑ **Enfermedades cardiovasculares:** Son la causa más común de muerte e incapacidad en personas con diabetes. Entre estas enfermedades se incluyen la angina de pecho, infarto al miocardio, accidentes cerebrovasculares, enfermedad arterial e insuficiencia cardíaca. Altos niveles de presión arterial, colesterol y azúcar en sangre, contribuyen al aumento de riesgo de estas complicaciones.
- ❑ **Pié diabético:** Las personas con diabetes pueden experimentar problemas de mala circulación en los pies, como resultado del daño en los vasos sanguíneos. Este problema aumenta el riesgo de ulceraciones, infecciones y amputaciones, sin embargo, con un buen manejo de la diabetes, gran parte de las situaciones que impliquen la amputación de dichas extremidades pueden ser evitadas.
- ❑ **Enfermedades renales:** La nefropatía es muy común entre los pacientes con diabetes, puesto que el daño causado por el azúcar alto a los vasos sanguíneos pequeños, provoca que los riñones sean menos eficientes, o fallen por completo. Mantener casi normales los niveles de glucosa en sangre reduce en gran medida el riesgo de la nefropatía.
- ❑ **Enfermedades en el sistema nervioso:** El daño en los nervios (neuropatía) también es el producto de niveles de glucosa altos por tiempos prolongados. Pueden afectar cualquier parte del cuerpo. El tipo más común es la neuropatía periférica, que afecta principalmente a los nervios de los pies, lo que puede producir dolor y pérdida de la sensibilidad. Esta última puede permitir que las lesiones pasen desapercibidas, generando ulceraciones, infecciones graves y en los casos más complejos amputaciones. La neuropatía también puede desarrollar disfunción eréctil, problemas digestivos y urinarios.

Los profesionales de la salud consideran de suma importancia un tratamiento eficaz y efectivo para la diabetes, haciendo necesaria la creación de terapias médicas que permitan alcanzar en el paciente diabético el control de su enfermedad, reduciendo de esta forma las complicaciones asociadas a la diabetes mal controlada, y disminuyendo los gastos monetarios para el paciente y su sistema de salud [38].

Mantener los niveles de glucosa dentro de los rangos establecidos como saludables por las organizaciones de salud a nivel mundial, en determinados intervalos de tiempo y con el apoyo de la cantidad suficiente de insulina, se convierte en un problema de control óptimo. El diseño de terapias médicas adecuadas depende de una buena comprensión y análisis de la dinámica fisiológica, y del uso de métodos de optimización utilizados en algunos procesos de control de similares características, convirtiéndolo así en un campo de investigación científico, bastante activo en la actualidad [3]. El desarrollo de esta investigación se orienta en construir un mecanismo de control óptimo a la cantidad de insulina suministrada a un paciente diabético, a través de un modelo matemático de ecuaciones diferenciales no lineales que describen la dinámica del sistema fisiológico glucosa-insulina.

1.6. Estado del arte de algunos modelos matemáticos de la dinámica glucosa-insulina

Desde la década del 60 se ha estudiado el problema de la diabetes. Gran número de investigadores y científicos han desarrollado numerosos modelos matemáticos, modelos estadísticos y algoritmos informáticos, que permitan comprender y describir matemáticamente la dinámica del sistema glucosa-insulina y, por supuesto, identificar características de la enfermedad en el paciente diabético, buscando así crear tratamientos más efectivos, evitar complicaciones y reducir costos asociados a su tratamiento.

Debido a la complejidad del organismo humano y de la propia fisiopatología que representa la diabetes, resulta ser prácticamente imposible el desarrollo de un modelo unificado que describa el sistema glucosa-insulina [4, p. 7], por lo que existe en la literatura especializada gran número de modelos para la representación de este sistema.

Por otro lado, en el presente trabajo, y para los consultados en la revisión bibliográfica realizada, el problema de control óptimo consiste en elegir el suministro externo de insulina, de tal manera que, dentro de un intervalo de tiempo determinado $[0, T]$ la concentración de glucosa se mantenga en un nivel considerado como saludable, y la cantidad de insulina inyectada sea pequeña [30].

A continuación se describen de manera muy general algunos modelos matemáticos desarrollados para el sistema glucosa-insulina, así como los métodos y estrategias utilizadas en la solución del problema de control óptimo planteado a partir de estos modelos.

1.6.1. Modelo de Ackerman

En 1960, Ackerman y su equipo de investigación, proponen un modelo matemático para describir la regulación de la glucosa en sangre, teniendo en cuenta las siguientes consideraciones biológicas [1, p.26]:

- Un aumento en la concentración de glucosa en sangre provoca que las células del hígado absorban la mayoría de la glucosa y la almacenen en forma de glucógeno. En caso de que la concentración de glucosa en la sangre caiga, el proceso se revierte.
- Un aumento de insulina en la sangre permite a la glucosa pasar a las células prontamente, ocasionando una gran absorción de glucosa de la sangre.
- Una disminución en la concentración de glucosa en la sangre, reduce la tasa de producción de insulina del páncreas. Mientras, un incremento en la concentración de glucosa en la sangre estimula la producción de insulina a tasas más rápidas.
- La insulina producida por el páncreas, es constantemente degradada por el hígado.

El modelo se compone del siguiente sistema de ecuaciones diferenciales ordinarias

$$\begin{aligned}\frac{dg}{dt} &= -m_1g(t) - m_2h(t) + p(t), \\ \frac{dh}{dt} &= -m_3h(t) + m_4g(t) + u(t),\end{aligned}\tag{1.1}$$

donde $g(t)$ denota la concentración de glucosa en sangre, $h(t)$ la concentración de insulina en sangre, $p(t)$ la tasa de aporte de glucosa externa, y $u(t)$ la tasa de inyección de insulina externa por unidad de volumen en sangre. Los multiplicadores m_1, m_2, m_3, m_4 son constantes positivas.

En seguida se presentan algunos de los detalles de la solución propuesta por Krabs [30], al estudiar el problema de control óptimo sobre el modelo lineal de Ackerman para el caso de la diabetes tipo 1. El desarrollo completo se puede consultar en [30, p. 119].

Considere el sistema (1.1). En el caso de la diabetes tipo 1, se supone que no existe producción de insulina, situación que solo es posible si $m_4 = 0$. Por esta razón, la influencia de $m_2h(t)$ en la primera ecuación de (1.1) para la reducción de la glucosa es bastante deficiente y la influencia del suministro externo de glucosa descrito por $p(t)$ se convierte en dominante.

Con el fin de evitar un exceso de glucosa en sangre, se anula el suministro externo de glucosa, es decir, $p(t) = 0$, y el cuerpo recibe un suministro externo de insulina en una concentración dependiente del tiempo, representado por $u = u(t)$. Haciendo el cambio de variable $\chi_1 = g$, $\chi_2 = h$, se obtienen las ecuaciones (escrita la derivada con la notación de Newton)

$$\begin{aligned}\dot{\chi}_1(t) &= -m_1\chi_1(t) - m_2\chi_2(t) \\ \dot{\chi}_2(t) &= -m_3\chi_2(t) + u(t), \quad t > 0,\end{aligned}\tag{1.2}$$

con condiciones iniciales

$$\chi_1(0) = g_0, \quad \chi_2(0) = h_0.\tag{1.3}$$

El objetivo ahora consiste en elegir el suministro externo de insulina de tal manera que dentro de un intervalo de tiempo determinado $[0, T]$ la concentración de glucosa se mantenga casi a un nivel g_d prescrito en la sangre, y la cantidad de insulina $u(t)$ inyectada sea lo más pequeña posible para el logro del objetivo. Lo anterior conduce a la minimización de la integral

$$J(u) = \int_0^T (\chi_1(t, u) - g_d)^2 + \zeta u(t)^2 dt, \quad (1.4)$$

con ζ parámetro de penalización que evita el aumento de la cantidad de insulina exógena a suministrar.

Solucionando el sistema de ecuaciones (1.2) por el método del factor integrante, y asumiendo que $g_d = 0$, el problema de control óptimo consiste en encontrar algún $u \in C[0, T]$ que minimice

$$J(u) = \int_0^T \left(\chi_1^0(t) - \int_0^t K(t-s)u(s)ds \right)^2 + \zeta u(t)^2 dt, \quad (1.5)$$

donde

$$\chi_1^0(t) = e^{-m_1 t} \left[g_0 - \frac{m_2 h_0}{m_1 - m_3} (e^{(m_1 - m_3)t} - 1) \right], \quad y$$

$$K(t-s) = \frac{m_2}{m_1 - m_3} (e^{(m_1 - m_3)(t-s)} - 1) e^{-m_1(t-s)}.$$

Para resolver el problema, aproximadamente, se reemplaza $C[0, T]$ por un subespacio U_n n -dimensional apropiado, generado por las funciones $u_1, \dots, u_n \in C[0, T]$. El subespacio U_n consiste de todas las combinaciones lineales de la forma

$$u(t) = \sum_{j=1}^n \alpha_j u_j(t), \quad t \in [0, T]. \quad (1.6)$$

De reemplazar (1.6) en (1.5) se obtiene

$$f(\alpha_1, \dots, \alpha_n) = J \left(\sum_{j=1}^n \alpha_j u_j \right) = \int_0^T \left(\chi_1^0(t) - \sum_{j=1}^n w_j \alpha_j \right)^2 + \zeta \left(\sum_{j=1}^n u_j(t) \alpha_j \right)^2 dt, \quad (1.7)$$

con

$$w_j(t) = \int_0^t K(t-s)u_j(s)ds \quad \text{Para } t \in [0, T] \quad y \quad j = 1, \dots, n.$$

En lugar de encontrar algún $\hat{u} \in C[0, T]$ tal que

$$J(\hat{u}) \leq J(u) \quad \text{para todo } u \in C[0, T],$$

se busca un vector $\hat{\alpha} = (\hat{\alpha}_1, \dots, \hat{\alpha}_n)^T \in \mathbb{R}^n$ tal que

$$f(\hat{\alpha}) \leq f(\alpha) \quad \text{para todo } \alpha \in \mathbb{R}^n. \quad (1.8)$$

Calculando el gradiente de (1.8) e igualando a cero, se obtiene el sistema lineal

$$\sum_{k=1}^n \left[\int_0^T \zeta u_j(t) u_k(t) + w_j(t) w_k(t) dt \right] \hat{\alpha}_k = \int_0^T \chi_1^0(t) w_j(t) dt, \quad \text{para } j = 1, \dots, n.$$

El sistema lineal tiene única solución $(\hat{\alpha}_1, \dots, \hat{\alpha}_n)$, por ser una base canónica del subespacio, obteniendo así

$$\chi_1(t) = \chi_1^0 - \sum_{j=1}^n \hat{\alpha}_j w_j(t) = x_1^0 - \int_0^T K(t-s) \hat{u}(s) ds,$$

donde

$$\hat{u}(t) = \sum_{j=1}^n \hat{\alpha}_j u_j(t),$$

por lo cual

$$J(\hat{u}) = \int_0^T \left(\chi_1^0 - \sum_{k=1}^n w_k(t) \hat{\alpha}_k \right) \chi_1^0(t) dt. \quad (1.9)$$

1.6.2. Modelo no lineal de Huang

La obtención de este modelo se realiza a partir de la ley de conservación de masa aplicada al sistema glucosa-insulina. La estructura del modelo es

$$\begin{aligned} G'(t) &= -f_2(G(t)) - f_3(G(t))f_4(I(t)) + f_5(I(t)) + f(t), \\ I'(t) &= f_1(G(t)) - d_i I(t), \end{aligned} \quad (1.10)$$

con condiciones iniciales $G(0) = G_0 > 0$, $I(0) = I_0 > 0$, donde $G(t)$ e $I(t)$ representan la cantidad de glucosa e insulina en la sangre en el instante t respectivamente. La función f_1 representa matemáticamente la producción de insulina estimulada por la glucemia, f_5 representa una producción de glucosa controlada por la concentración de insulina, f_2 , f_3 y f_4 representan la glucosa utilizada principalmente por las células del cerebro, los nervios, músculos y tejidos, y d_i es una constante que cuantifica la tasa de eliminación y degradación de la insulina por procesos metabólicos generados en el hígado y los riñones. La deducción del modelo junto con la especificación de las funciones f_1 , f_2 , f_3 , f_4 y f_5 puede leerse con más detalle en [4, pp. 31-33], [24].

En el trabajo desarrollado en [4], se incluye una linealización de (1.10) alrededor del punto de equilibrio del sistema, que se utiliza para realizar simulaciones y comparalas con las realizadas a partir del modelo no lineal.

Al igual que en el apartado anterior, resolver el problema de control óptimo consiste en

minimizar el funcional

$$J(u) = \int_0^T [(G(t) - G_B)^2 + \zeta u^2(t)] dt, \quad (1.11)$$

para el cual proponen una resolución numérica utilizando el software GPOPS [40]. Este software transforma el problema de control óptimo en un problema de programación no lineal finito dimensional, mediante la aproximación del control y del estado con polinomios de Lagrange, realizando un proceso de discretización del sistema de ecuaciones diferenciales ordinarias.

Se presentan varias simulaciones asignando distintos valores a las condiciones iniciales del problema, y al parámetro de penalización ζ tanto para el modelo no lineal como para el modelo linealizado, concluyendo que para ciertos valores del parámetro de penalización y condiciones iniciales, el modelo no lineal modela mejor el problema de control óptimo, puesto que se requieren administrar cantidades menores de insulina, alcanzando el nivel de glucosa basal en tiempos más cortos comparado con el modelo linealizado, lo que justifica de esta manera la utilidad del modelo no lineal [4, pp. 45-48].

Cabe mencionar que las simulaciones presentadas en [4], solo consideran intervalos de tiempo entre 0 y 240 minutos, es decir, se simuló el comportamiento de la glucosa durante la aplicación de la prueba oral OGTT (*Oral Glucose Tolerance Test*), sin embargo, es importante que se considere realizar simulaciones para las 24 horas del día, puesto que este aspecto se asemeja más a la realidad del contexto de la situación problema que se pretende estudiar, además de posibilitar la identificación y análisis de otros detalles y particularidades que pueden pasar desapercibidas en una simulación donde el intervalo de tiempo no sea lo suficientemente amplio (por ejemplo los diferentes estilos de alimentación adoptados por un paciente diabético), aspecto que se pretende tener en cuenta en el desarrollo de este trabajo. Adicionalmente, los valores de los parámetros utilizados en la aplicación de este modelo, fueron estimados a través de experimentos y simulaciones de pruebas orales como la IVGTT (*Intravenous Glucose Tolerance Test*) [24, p. 1546], pero no se menciona qué técnica o método matemático y/o estadístico se usó para la deducción de estos valores. En este trabajo, se presenta en el capítulo 3, un método numérico que permitirá encontrar los valores de los parámetros a utilizar en el modelo de la dinámica glucosa-insulina incorporado en el estudio del problema de control óptimo, a partir de una tabla de concentraciones de glucosa e insulina de un paciente diabético.

1.6.3. Modelo de Bergman

El modelo mínimo de Bergman es un modelo unicompartmental, es decir, que el cuerpo es descrito como un solo compartimiento o tanque con una concentración basal de insulina [3].

Matemáticamente, el modelo está descrito por las ecuaciones diferenciales

$$\begin{aligned}\frac{dg}{dt} &= -p_1g(t) - X(t)g(t), \\ \frac{dX}{dt} &= -p_2X(t) + p_3h(t), \\ \frac{dh}{dt} &= p_4[g(t) - p_5]t - p_6h(t),\end{aligned}\tag{1.12}$$

con condiciones iniciales $g(0) = G_0$, $h(0) = H_0$ y $X(0) = 0$, donde $p_i > 0$ para todo i , $g(t)$ y $h(t)$ representan la concentración de glucosa e insulina respectivamente, $X(t)$ hace referencia a una variable llamada efecto de insulina activa, que está relacionada con la concentración de insulina en sangre en la medida en que su actuación afecta la captación de glucosa por parte de los tejidos, así como la absorción y producción por parte del hígado. La deducción de este modelo también se hace, al igual que en el caso anterior, a partir de un balance de masas al compartimiento considerado. El desarrollo detallado del modelo puede consultarse en [3, p. 15-18] y [11].

En [9] se realiza un estudio al problema de control óptimo para la diabetes tipo 1, simplificando el modelo no lineal mediante una linealización de éste entorno a un punto de equilibrio. El objetivo de este trabajo es aplicar una técnica de control de sistemas a través de redes de comunicaciones, con el fin de controlar las dosis de insulina por medio de controladores digitales conectados a bombas que suministran el medicamento al paciente diabético.

Se concluye que el modelo linealizado representa de forma muy fiable al modelo no lineal alrededor del punto de equilibrio, por lo que se decide utilizar el modelo linealizado para el diseño del controlador de suministro de insulina, por la simplicidad que su uso representa. Los controladores son diseñados aplicando la técnica H2/H, que permite diseñar un controlador estabilizante el cual garantiza que el sistema controlado presente un comportamiento óptimo y, al mismo tiempo, tenga un margen de estabilidad definido con respecto a incertidumbres, es decir, robustez [9].

El modelo no lineal y el controlador robusto son programados con la herramienta Simulink de MATLAB[®] y probados en distintas simulaciones de sistemas metabólicos bajo condiciones de diabetes tipo 1. Las simulaciones permitieron comprobar que se logró controlar la glucosa de manera eficiente, evitando que se obtuviesen altas concentraciones y que no se genere sobredosis del medicamento, puesto que esta situación podría provocar la muerte del paciente [9, p. 129-131]. Aquí, tampoco se menciona la forma o método de obtención de los valores usados en los parámetros del modelo simulado como restricción del problema de control óptimo.

1.6.4. Otro modelo

El trabajo desarrollado en [47] propone una estrategia de control sobre inyecciones de insulina suministradas a pacientes con diabetes tipo 1, utilizando el método de control en lazo cerrado. Este método consiste en un sistema de control con realimentación, en el cual una cierta proporción de la salida del sistema se redirige a la entrada con el objeto de controlar su comportamiento [18]. El objetivo principal de un método de control en lazo cerrado es monitorear continuamente el nivel de glucosa en sangre y, en función de éste nivel, poder determinar las dosis de insulina que ha de suministrar una bomba de regulación [47].

El sistema de lazo cerrado diseñado incluye un regulador PID (Proporcional-Integral-Derivativo) con el fin de modelar el sistema metabólico glucosa-insulina utilizando un conjunto de reglas que no necesariamente se rigen explícitamente por un modelo del sistema a controlar, así que las reglas del regulador están sujetas a un modelo no lineal que consta de otros tres subsistemas, el cual no es explícito.

El esquema general del modelo y las simulaciones fueron elaboradas con la herramienta Simulink de MATLAB®. Se concluye que el modelo permite hacer un buen control de las variaciones de la glucosa en pacientes con diabetes tipo 1, a pesar de que la información ingresada al controlador presenta incertidumbres. El modelo desarrollado podría mejorarse si contempla en el modelado incertidumbres en la alimentación, asegurando así la obtención de un control más seguro.

1.7. Objetivos

El objetivo general y los objetivos específicos que se plantean en esta investigación son los siguientes.

1.7.1. Objetivo general

Diseñar un mecanismo de control óptimo para el suministro de insulina a un paciente diabético, teniendo en cuenta particularidades como el tipo de diabetes, o algunas características de los hábitos alimenticios.

1.7.2. Objetivos específicos

- ❑ Explorar las técnicas de control óptimo, con el fin de seleccionar la más apropiada para resolver el problema, aplicado al sistema glucosa-insulina escogido.
- ❑ Plantear y analizar matemáticamente el problema del control óptimo que emerge de la búsqueda de un manejo adecuado de la diabetes, en personas que la padecen.

- ❑ Adaptar y aplicar un método numérico para la estimación del valor de los parámetros que componen el sistema de ecuaciones escogido como descriptor de la dinámica glucosa-insulina, a partir de un conjunto de datos experimentales.
- ❑ Implementar computacionalmente el mecanismo de control óptimo propuesto, con el fin de analizar los resultados obtenidos.

2

Modelo matemático de la dinámica glucosa-insulina

Los modelos matemáticos biológicos se configuran a partir de las características de los procesos biológicos que se pretenden estudiar. Es así como se han propuesto gran cantidad de modelos que describen, por ejemplo, el comportamiento macroscópico de la propagación de una enfermedad infecciosa o viral, como el caso del modelo SIR y sus variaciones, o los modelos que se han usado en problemas de dinámica de poblaciones (Malthus, Lotka-Volterra, entre otros). Para el caso de modelos fisiológicos como el de la regulación glucosa-insulina, son varios los modelos que hasta la fecha se han propuesto y desarrollado, estos incorporan las diferentes interacciones dadas a partir del proceso de metabolización de la glucosa en el cuerpo humano, tal como se mencionó en el capítulo anterior.

En este capítulo se presentan con detalle los aspectos considerados en la construcción del modelo que se ha seleccionado como representación matemática del proceso fisiológico de interés de este trabajo, así como la demostración de la existencia y unicidad del sistema de ecuaciones diferenciales que lo componen. Seguidamente, se presentan los resultados obtenidos al implementar computacionalmente un método numérico, escogido entre otros disponibles en la literatura por las características de su error de aproximación, que permita aproximarse a la solución del sistema de ecuaciones debido a que por su naturaleza no es posible el hallazgo de una solución analítica, teniendo en cuenta algunas consideraciones particulares sobre el modelo. Finalmente, estos resultados son analizados y discutidos.

2.1. Descripción del modelo matemático

La dinámica de la glucosa y la insulina puede ser interpretada y analizada de forma similar como se hace con un modelo del tipo depredador-presa, en el que la glucosa resulta ser la presa, y la insulina el depredador. Como es conocido, en estos modelos se relacionan las tasas de cambio de las poblaciones que intervienen, con las tasas de crecimiento y las tasas de interacción entre las especies, esta última, como resultado de la ley de conservación de masas usada en el contexto de la química, en la que se establece una proporcionalidad entre la tasa a la cual cambian las poblaciones que interactúan con los encuentros entre ellas [6, p. 11].

En modelos sencillos como el presentado en (1.1), puede verse que no se incluyen los términos que permitan cuantificar dicha interacción entre la glucosa y la insulina, pensadas análoga-

mente como las especies que intervienen en un problema de dinámica de poblaciones, por lo que puede concluirse que este modelo deja de lado un aspecto que debería considerarse importante, si lo que se desea es la modelación de un proceso fisiológico lo más fiel y parecida a la realidad, aunque debe mencionarse que esta propuesta fue pionera [3, 4] en el intento por realizar un proceso de modelación que sirviera como método para el diagnóstico de la diabetes, buscando ser lo más sencilla posible y con pocos parámetros.

Por otro lado, en modelos como los planteados en (1.10) o (1.12), ya se incluye el factor de interacción entre la glucosa y la insulina, junto con otros elementos que describen algunos subprocesos que también se llevan a cabo como parte del proceso de metabolización de la glucosa. Los trabajos alrededor del problema de control óptimo se han desarrollado, en su mayoría, a partir de los modelos mencionados anteriormente, tal como se puede apreciar en [4] y [9] respectivamente, y que se describen brevemente en el capítulo 1 del presente trabajo. Las ecuaciones que componen estos dos últimos modelos, pueden dificultar cualquier tipo de análisis que se pretenda realizar sobre el sistema, ya sea porque poseen muchos parámetros o por el número de ecuaciones que lo componen, sin embargo, en muchos casos un modelo simple podría ser suficiente para desarrollar un buen análisis [3, p. 15].

Sobre el modelo escogido para el estudio del problema de control óptimo que se desarrollará más adelante, no se encontraron trabajos previos, y por lo tanto, los análisis y conclusiones resultado de esta investigación pueden considerarse como una innovación en este campo. Cabe mencionar que a diferencia de los sistemas (1.10) (1.12), el número de ecuaciones y de parámetros es menor, y se incluye dentro de la dinámica del sistema, al igual que en los modelos mencionados anteriormente, el término de interacción entre la glucosa y la insulina, el rol que desempeñan algunos órganos como el hígado, la ingesta de comida y aportes externos de insulina en el caso de pacientes diabéticos.

El modelo que se presenta a continuación es tomado de [27, p. 70]. Sean $g(t)$, $h(t)$ los niveles de glucosa e insulina en el flujo sanguíneo en un tiempo t . La tasa de cambio dg/dt de la concentración de glucosa es proporcional a:

- (i) El producto gh , puesto que cuanto mayor sea el nivel de glucosa e insulina, mayor es el metabolismo de la glucosa, es decir, la presencia de insulina permite el uso de la glucosa como energía en diferentes tejidos y órganos del cuerpo, lo que genera su disminución con tendencia a desaparecer.
- (ii) El término $(g_0 - g)$, porque si el nivel de glucosa decae por debajo del nivel basal o de ayuno, entonces la glucosa debe ser liberada desde las reservas en el hígado donde se encuentra almacenada, para elevarse al nivel basal.
- (iii) El término $(g - g_0)$, debido a que si $g > g_0$, entonces el nivel de glucosa decae naturalmente proporcional a su exceso sobre el nivel de ayuno.
- (iv) La función $(t - \tau)$ donde τ es el tiempo en el cual es consumido el alimento.

De lo anterior, se obtiene que:

$$\frac{dg}{dt} = -a_1gh + a_2(g_0 - g)^+ - a_3(g - g_0)^+ + a_4p(t), \quad (2.1)$$

donde $(\cdot)^+$ representa la parte positiva de una función y la expresión $a_2(g_0 - g)^+$ describe la acción del hígado que secreta la glucosa almacenada cuando su concentración está por debajo del nivel basal, proceso que se activa por liberación de la hormona glucógeno desde el páncreas.

Ahora, la tasa de cambio dh/dt del nivel de insulina es proporcional a:

- (i) El exceso $g - g_0$ de glucosa en la sangre por encima del nivel de ayuno (el nivel basal), ya que este exceso hace que el páncreas secrete insulina en el torrente sanguíneo.
- (ii) La concentración $h(t)$ de insulina, ya que la concentración de esta hormona tiende a decaer a una tasa proporcional a su concentración.

Así pues, se obtiene la ecuación diferencial:

$$\frac{dh}{dt} = b_1(g - g_0)^+ - b_2h, \quad (2.2)$$

y de manera análoga, el término $(g - g_0)^+$ puede interpretarse como una función que representa la reacción del páncreas, órgano que libera insulina a medida que el nivel de glucosa se eleva por encima del nivel basal, como se mencionó anteriormente.

Finalmente, $p(t)$ representa el ingreso de glucosa externa al sistema, que comúnmente se obtiene a través de la ingesta de alimentos, con $\tau_i = 1, 2, \dots, n$ que indica la hora del día en la que una persona consume el alimento y n el número de comidas que la persona consume. En la literatura, por ejemplo [4] o [9], pueden encontrarse varias propuestas para describir adecuadamente esta función, sin embargo, en [27] se incluye también una propuesta de función para modelar esta entrada de glucosa al organismo, la cual será descrita en el apartado 2.3.1 y sobre la cual se realizan diferentes modificaciones en las simulaciones de los distintos escenarios de alimentación de un paciente diabético considerados en el capítulo 5 del presente trabajo.

2.2. Análisis matemático del sistema

El modelo matemático que representa el sistema glucosa-insulina, viene dado por el problema de valor inicial

$$\begin{cases} \frac{dg}{dt} = -a_1gh + a_2(g_0 - g)^+ - a_3(g - g_0)^+ + a_4p(t), & g(t_0) = g_0, \\ \frac{dh}{dt} = b_1(g - g_0)^+ - b_2h, & h(t_0) = h_0. \end{cases} \quad (2.3)$$

Antes de introducir cualquier resultado sobre este sistema de ecuaciones, es necesario garantizar la existencia de una solución única, definida para t en el intervalo $[0, T]$, lapso de tiempo en el cual interesa estudiar la dinámica del sistema fisiológico a modelar. Además teniendo en cuenta el contexto fisiológico, se considera que $0 \leq g(t) \leq g_{m\acute{a}x}$ y $0 \leq h(t) \leq h_{m\acute{a}x}$ donde $g_{m\acute{a}x}$, $h_{m\acute{a}x}$ son los niveles de concentración de glucosa e insulina más altos que el cuerpo humano tolera. Para esto, se empleará el teorema de Picard-Lindelöf.

Teorema 1. Sea $\mathbf{D} = \{(t, g(t), h(t)) \mid 0 \leq t \leq T, 0 \leq g(t) \leq g_{m\acute{a}x}, 0 \leq h(t) \leq h_{m\acute{a}x}\}$ un subconjunto abierto de $[0, T] \times \mathbb{R}^2$ sobre el cual está definido el PVI (2.3) con $(t_0, g_0, h_0) \in \mathbf{D}$. Entonces para algún $r > 0$ existe una única solución $(g(t), h(t))$ definida en el intervalo $I = [t_0 - r, t_0 + r]$.

Demostración:

En primer lugar, se define

$$\vec{f}(t, \vec{x}) = (-a_1gh + a_2(g_0 - g)^+ - a_3(g - g_0)^+ + a_4p(t), b_1(g - g_0)^+ - b_2h),$$

que por la forma en que está definida, puede verse claramente su continuidad en el abierto \mathbf{D} . Para comprobar que la función es Lipschitz respecto de \vec{x} en \mathbf{D} , se aplica primero la definición sobre f_1 , respecto de g y h , y posteriormente sobre f_2 .

Por lo tanto, tomando a f_1 , se tiene que para todo $t \in [0, T]$, toda $h \in [0, h_{m\acute{a}x}]$ y toda $g_1, g_2 \in [0, g_{m\acute{a}x}]$:

$$\begin{aligned} |f_1(t, g_1, h) - f_1(t, g_2, h)| &= |(-a_1g_1h + a_2(g_0 - g_1)^+ - a_3(g_1 - g_0)^+ + a_4p(t)) \\ &\quad - (-a_1g_2h + a_2(g_0 - g_2)^+ - a_3(g_2 - g_0)^+ + a_4p(t))| \\ &= | - a_1g_1h + a_2(g_0 - g_1)^+ - a_3(g_1 - g_0)^+ + a_4p(t) + a_1g_2h \\ &\quad - a_2(g_0 - g_2)^+ + a_3(g_2 - g_0)^+ - a_4p(t) | \\ &= | - a_1h(g_1 - g_2) + a_2((g_0 - g_1)^+ - (g_0 - g_2)^+) \\ &\quad - a_3((g_1 - g_0)^+ - (g_2 - g_0)^+) |. \end{aligned}$$

Por la forma en la que están definidas las funciones $(g - g_0)^+$ y $(g_0 - g)^+$, deben estudiarse los siguientes casos

Caso 1. Si $g_1, g_2 < g_0$

$$\begin{aligned} |f_1(t, g_1, h) - f_1(t, g_2, h)| &= | - a_1h(g_1 - g_2) + a_2(g_0 - g_1 - g_0 + g_2) | \\ &= | - a_1h(g_1 - g_2) - a_2(g_1 - g_2) | \\ &= |(-a_1h - a_2)(g_1 - g_2)| \\ &= |a_1h + a_2||g_1 - g_2| \\ &\leq L|g_1 - g_2|, \end{aligned}$$

donde la constante de Lipschitz es $L = |a_1h + a_2|$, con $h = h_{m\acute{a}x}$, llegando así a la igualad

que se quería demostrar.

Caso 2. Si $g_0 < g_1 < g_2$

$$\begin{aligned}
 |f_1(t, g_1, h) - f_1(t, g_2, h)| &= | -a_1h(g_1 - g_2) - a_3(g_1 - g_0 - g_2 + g_0) | \\
 &= | -a_1h(g_1 - g_2) - a_3(g_1 - g_2) | \\
 &= |(-a_1h - a_3)(g_1 - g_2)| \\
 &= |a_1h + a_3||g_1 - g_2| \\
 &\leq L|g_1 - g_2|,
 \end{aligned}$$

donde la constante de Lipschitz es $L = |a_1h + a_3|$, con $h = h_{m\acute{a}x}$, llegando así a la igualdad que se quería demostrar.

Caso 3. Si $g_1 < g_0 < g_2$

$$\begin{aligned}
 |f_1(t, g_1, h) - f_1(t, g_2, h)| &= | -a_1h(g_1 - g_2) + a_2(g_0 - g_1) + a_3(g_2 - g_0) | \\
 &\leq | -a_1h(g_1 - g_2) | + |a_2(g_0 - g_1)| + |a_3(g_2 - g_0)| \\
 &\leq |a_1h||g_1 - g_2| + |a_2||g_0 - g_1| + |a_3||g_2 - g_0| \\
 &\leq (a_1h + a_2 + a_3)|g_1 - g_2| \\
 &\leq L|g_1 - g_2|,
 \end{aligned}$$

donde la constante de Lipschitz es $L = a_1h + a_2 + a_3$, con $h = h_{m\acute{a}x}$, llegando así a la desigualdad que se quería demostrar. El caso donde $g_0 < g_2 < g_1$ se demuestra análogamente al caso 2 presentado anteriormente.

Ahora, tomando a f_2 se tiene que para todo $t \in [0, T]$, toda $h \in [0, h_{m\acute{a}x}]$ y toda $g_1, g_2 \in [0, g_{m\acute{a}x}]$

$$\begin{aligned}
 |f_2(t, g_1, h) - f_2(t, g_2, h)| &= |(b_1(g_1 - g_0)^+ - b_2h) - (b_1(g_2 - g_0)^+ - b_2h)| \\
 &= |b_1(g_1 - g_0)^+ - b_2h - b_1(g_2 - g_0)^+ + b_2h| \\
 &= |b_1(g_1 - g_0)^+ - b_1(g_2 - g_0)^+|.
 \end{aligned}$$

Caso 1. Si $g_1, g_2 < g_0$, entonces $L = 1$ y se puede concluir que la función es Lipschitz.

Caso 2. Si $g_0 < g_1 < g_2$, se tiene

$$\begin{aligned}
 |f_2(t, g_1, h) - f_2(t, g_2, h)| &= |b_1(g_1 - g_0) - b_1(g_2 - g_0)| \\
 &= |(b_1g_1 - b_1g_0 - b_1g_2 + b_1g_0)| \\
 &= |b_1g_1 - b_1g_2| \\
 &= |b_1(g_1 - g_2)| \\
 &= b_1|g_1 - g_2| \\
 &\leq L|g_1 - g_2|,
 \end{aligned}$$

donde la constante de Lipschitz es $L = b_1$, llegando así a la igualdad que se quería demostrar.

Caso 3. Si $g_1 < g_0 < g_2$

$$\begin{aligned} |f_2(t, g_1, h) - f_2(t, g_2, h)| &= |-b_1(g_2 - g_0)| \\ &= |-b_1||g_2 - g_0| \\ &\leq |-b_1||g_1 - g_2| \\ &\leq L|g_1 - g_2|, \end{aligned}$$

donde la constante de Lipschitz es $L = |-b_1|$, llegando así a la desigualdad que se quería demostrar.

Finalmente, para ver que f_1 y f_2 son Lipschitz con respecto a h , es fácil comprobar que las derivadas parciales de estas funciones respecto de h existen, son continuas y además están acotadas en \mathbf{D} , lo que permite garantizar la existencia de la constante de Lipschitz L , de acuerdo con la definición.

Así pues, por el teorema de Picard-Lindelöf se puede concluir que existe una única solución del sistema de ecuaciones diferenciales (2.3), con $t_0 = 0$, $\vec{x}_0 = (g_0, h_0)$ en un intervalo de la forma $I = [0, r]$ para algún $r > 0$, que es solución del problema de valor inicial propuesto. ■

2.3. Solución numérica del sistema de ecuaciones

Una vez demostrada la existencia y unicidad de la solución del modelo, se aplica un método numérico que permita aproximarse a dicha solución, y así obtener representaciones gráficas a partir de las cuales sea posible interpretar la situación problema que se intenta modelar, puesto que este sistema de ecuaciones diferenciales no posee una solución analítica general.

2.3.1. Simulación computacional

Dada la exactitud que ofrece el método de Runge-Kutta (ver apéndice A), frente a otros métodos numéricos utilizados en la solución de ecuaciones diferenciales, se opta por seleccionarlo como algoritmo para aproximar la solución de (2.3).

Como el sistema en mención involucra dos ecuaciones diferenciales que deben resolverse simultáneamente, se requiere entonces construir el esquema iterativo teniendo en cuenta esta particularidad, para posteriormente ser programado en un computador usando algún lenguaje de programación (en el desarrollo de este trabajo, todas las simulaciones serán construidas en MATLAB®), por lo que el método estará determinado por las ecuaciones que se muestran en la tabla 2.1.

En el algoritmo 1 se presenta el pseudocódigo de la simulación empleado para obtener la solución numérica del sistema (2.3) que describe la dinámica glucosa-insulina, escogido para el desarrollo del presente estudio. Se utilizó un intervalo de tiempo comprendido entre las 0 y las 24 horas del día ($t \in [0, 24]$), con paso de tiempo $\Delta t = 0.01$, escogido arbitrariamente.

$\frac{dg}{dt} = f_1(t, g, h)$	$\frac{dh}{dt} = f_2(t, g, h)$
$k_1 = \Delta t f_1(t_i, g_i, h_i)$	$l_1 = \Delta t f_2(t_i, g_i, h_i)$
$k_2 = \Delta t f_1\left(t_i + \frac{\Delta t}{2}, g_i + \frac{k_1}{2}, h_i + \frac{l_1}{2}\right)$	$l_2 = \Delta t f_2\left(t_i + \frac{\Delta t}{2}, g_i + \frac{k_1}{2}, h_i + \frac{l_1}{2}\right)$
$k_3 = \Delta t f_1\left(t_i + \frac{\Delta t}{2}, g_i + \frac{k_2}{2}, h_i + \frac{l_2}{2}\right)$	$l_3 = \Delta t f_2\left(t_i + \frac{\Delta t}{2}, g_i + \frac{k_2}{2}, h_i + \frac{l_2}{2}\right)$
$k_4 = \Delta t f_1(t_i + \Delta t, g_i + k_3, h_i + l_3)$	$l_4 = \Delta t f_2(t_i + \Delta t, g_i + k_3, h_i + l_3)$
$g_{i+1} = g_i + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4)$	$h_{i+1} = h_i + \frac{1}{6}(l_1 + 2l_2 + 2l_3 + l_4)$

Tabla 2.1: Ecuaciones del método de Runge-Kutta para la solución numérica de (2.3).

Algoritmo 1: MÉTODO DE RUNGE-KUTTA CUARTO ORDEN.

$p := (a_1, a_2, a_3, a_4, b_1, b_2)$ Valores de los parámetros.

$g_0, h_0 :=$ Condiciones iniciales.

$t := [0, T]$ Intervalo de tiempo.

$\Delta t :=$ Tamaño del paso.

$N := T/\Delta t$ Número de iteraciones.

$f_1(t, g, h) = -a_1gh + a_2[g_0 - g]^* - a_3[g - g_0]^* + a_4z(t)$

$f_2(t, g, h) = b_1[g - g_0]^* - b_2h$

Entrada: $p, g_0, h_0, t, \Delta t, N, f_1, f_2$.

Salida: g, h .

Inicio

Para $i \leftarrow 1$ **hasta** N **hacer**

$k_1 \leftarrow \Delta t f_1(t_i, g_i, h_i)$

$l_1 \leftarrow \Delta t f_2(t_i, g_i, h_i)$

$k_2 \leftarrow \Delta t f_1(t_i + \Delta t/2, g_i + k_1/2, h_i + l_1/2)$

$l_2 \leftarrow \Delta t f_2(t_i + \Delta t/2, g_i + k_1/2, h_i + l_1/2)$

$k_3 \leftarrow \Delta t f_1(t_i + \Delta t/2, g_i + k_2/2, h_i + l_2/2)$

$l_3 \leftarrow \Delta t f_2(t_i + \Delta t/2, g_i + k_2/2, h_i + l_2/2)$

$k_4 \leftarrow \Delta t f_1(t_i + \Delta t, g_i + k_3, h_i + l_3)$

$l_4 \leftarrow \Delta t f_2(t_i + \Delta t, g_i + k_3, h_i + l_3)$

$g_{i+1} \leftarrow g_i + (k_1 + 2k_2 + 2k_3 + k_4)/6$

$h_{i+1} \leftarrow h_i + (l_1 + 2l_2 + 2l_3 + l_4)/6$

Fin

Devolver g_{i+1}, h_{i+1}

Fin

Para los parámetros de las ecuaciones se seleccionaron los valores $a_1 = 0.05$, $a_2 = 1.0$, $a_3 = 2$, $a_4 = 4$, $b_1 = 0.5$, $b_2 = 2$, y como condiciones iniciales del problema se tiene $g_0 = 100$ mg/dL, $h_0 = 0$ μ U/mL.

Por último, para simular el ingreso de glucosa externa al sistema, dado por la ingesta de alimentos de la persona n veces al día, se tomó la función $p(t)$ la cual se adapta de [27] como sigue

$$p(t) := \sum z(t - \tau_i), \quad (2.4)$$

con

$$z(t - \tau_i) = \begin{cases} 0, & \text{si } t < \tau_i, \\ Q e^{-k(t-\tau_i)}, & \text{si } t \geq \tau_i. \end{cases} \quad (2.5)$$

El término Q , introducido en (2.5), representa la cantidad de carbohidratos (g CHO) ingeridos por comida durante las 24 horas, ya que son los carbohidratos responsables del aporte de glucosa, una vez el alimento es descompuesto por el sistema digestivo. Por lo anterior, este término toma diferentes valores dependiendo de la hora del día a la que se consumen los alimentos, y τ_i , $i = 1, 2, \dots, n$ indica la hora del día en la que se lleva a cabo el consumo de los alimentos. Para el caso de la simulación que se presenta a continuación, el parámetro Q_i toma los siguientes valores:

$$Q = \begin{cases} 50 \text{ g CHO}, & \tau_1 = 8, \text{ (8:00 am, hora del desayuno).} \\ 20 \text{ g CHO}, & \tau_2 = 10.5, \text{ (10:30 am, hora de la merienda).} \\ 100 \text{ g CHO}, & \tau_3 = 12, \text{ (12:00 am, hora del almuerzo).} \\ 50 \text{ g CHO}, & \tau_4 = 18.5, \text{ (6:30 pm, hora de la cena).} \end{cases}$$

Observe que las concentraciones de glucosa e insulina se mantienen en sus niveles basales (condiciones iniciales del PVI definido en (2.3)), entre las 0:00 y las 8:00, debido a que durante este lapso de tiempo no hay ingesta de comida que perturbe el nivel basal de glucosa, y por lo tanto el término $(g_0 - g)^+$ debe ser diferente de cero, lo que indica secreción de la hormona glucagón por parte del páncreas para que la insulina almacenada en el hígado sea liberada en cantidades suficientes, de tal manera que sea posible mantener elevados los niveles de glucosa. A las 8:00 la persona consume su primera comida del día (el desayuno), y los alimentos que lo componen aportan 50 g de carbohidratos (por lo tanto, $Q = 50$ en (2.5)), los cuales a medida que son convertidos en glucosa por acción del sistema digestivo hacen que los niveles de glucosa en sangre se eleven paulatinamente, tal como puede apreciarse en la curva azul de la figura 2.1, hasta alcanzar el pico sobre las 9:20.

Por otro lado, como el nivel de glucosa empieza a elevarse, entonces el término $(g_0 - g)^+$ es cero, y por tanto el término $(g - g_0)^+$ es diferente de cero, lo que indica el inicio de la producción de insulina por el páncreas tal como puede apreciarse al seguir visualmente la tendencia de la curva roja en la figura 2.1, sin embargo, nótese que la insulina se eleva con un tiempo de retraso con respecto al momento en el que empieza a elevarse la glucosa (apro-

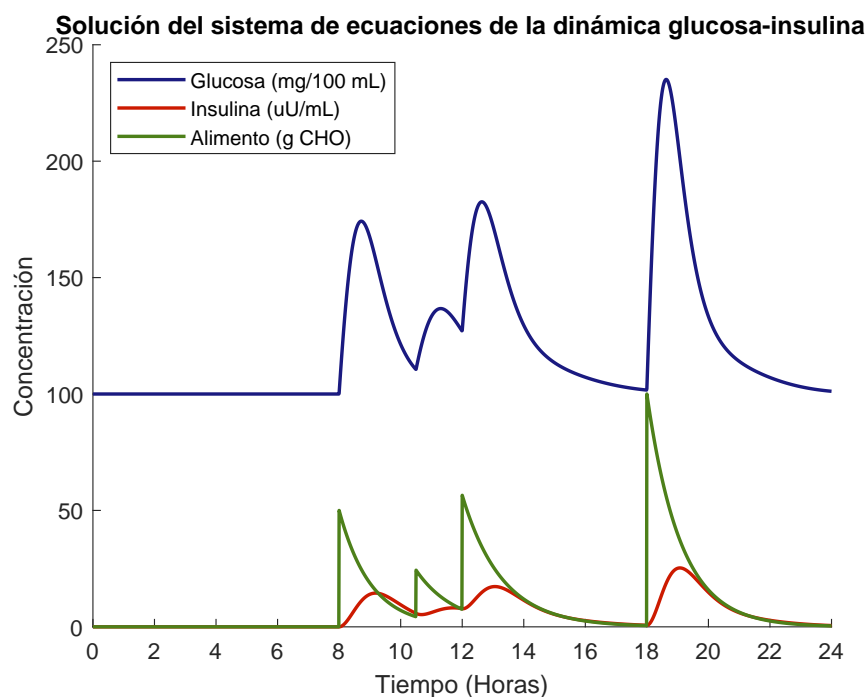


Figura 2.1: Simulación de la dinámica glucosa-insulina, a partir el modelo (2.3).

ximadamente 20 minutos después), y esto se debe a que en situaciones reales, la insulina es secretada una vez es detectado el aumento de los niveles de glucosa sobre el nivel basal, lo cual ocurre algunos minutos o incluso horas después de haberse consumido alguna comida, dependiendo del tiempo que le tome al sistema digestivo ir procesando y descomponiendo en glucosa los alimentos que la persona va ingiriendo. Este hecho como se puede ver, es simulado de manera adecuada por el modelo del sistema glucosa-insulina descrito en la sección 2.1.

Después de las 9:20, la concentración de glucosa en sangre disminuye, ya que por acción de la insulina ha podido ser utilizada por los diferentes órganos y tejidos del cuerpo como energía para realizar sus funciones, y de la misma manera, los niveles de insulina también disminuyen a medida que la hormona se va degradando naturalmente por su uso en este proceso metabólico, tendiendo ambos niveles a regresar al nivel basal, sin embargo, esto no alcanza a suceder porque a las 10:30 hay ingesta nuevamente de alimentos (la merienda), lo cual hace que los niveles de glucosa e insulina se eleven nuevamente. La dinámica descrita para el caso del desayuno se repite para las otras tres comidas que la persona consume durante el día. No obstante, se resalta que los niveles de glucosa, para todas las comidas, se encuentran por debajo de los 140 mg/dL pasadas dos horas o menos (en el caso de la merienda) después de haber sido consumidas, y además a las 24:00, se observa los niveles tanto de glucosa como de insulina han regresado a sus concentraciones basales, con lo cual se puede concluir, de acuerdo con la tabla 1.1, que la dinámica glucosa-insulina mostrada en la figura 2.1 con los parámetros y características considerados en esta simulación particular, corresponden al proceso metabólico de un paciente sano.

3

Estimación de parámetros

El problema de identificación o estimación de parámetros en modelos descritos por sistemas de ecuaciones diferenciales ordinarias ha sido tema del interés de numerosos matemáticos desde inicios de la década de los sesenta. Este problema tiene lugar en varias situaciones o fenómenos de la ciencia, la naturaleza y la tecnología, de los cuales es posible disponer un conjunto de datos obtenidos de manera experimental o a partir de test de laboratorio, sobre los cuales se desean ajustar los parámetros involucrados en estos modelos de ecuaciones [2]. Este capítulo presenta la formulación del problema de identificación de parámetros como un problema de mínimos cuadrados, además de una descripción del método utilizado para resolver problemas de este tipo, especialmente cuando son no lineales. Finalmente se presentan algunas simulaciones y resultados encontrados en la aplicación de la técnica de optimización a la estimación de los parámetros del modelo de la dinámica glucosa-insulina introducido en el capítulo 2.

3.1. Contextualización

Para el caso de la diabetes, es posible utilizar técnicas de identificación de parámetros para diseñar nuevas pruebas de diagnóstico que permitan discriminar, de manera más efectiva entre personas sanas y aquellas que sufren las diversas formas y complicaciones de esta enfermedad [26]. Por ejemplo, de la experimentación con la prueba OGTT (*Oral Glucose Tolerance Test*) por sus siglas en inglés, Ackerman y su equipo de investigación no solo propusieron el modelo de ecuaciones diferenciales para la dinámica glucosa-insulina presentado en (1.1), sino que además crearon un criterio bastante útil para distinguir entre pacientes sanos, casos leves de diabetes o propensos a padecer la enfermedad, tal como se puede revisar en su reporte de resultados de investigación publicado en 1965 (se sugiere revisar [1]).

Por otro lado, según [3, p. 10] el test o prueba conocida como IVGTT, junto con un modelo matemático como el presentado en (1.12), se puede utilizar para estimar la sensibilidad a la insulina, la eficacia de la glucosa y los parámetros de respuesta o sensibilidad del páncreas de un paciente diabético, lo que aportará elementos e información importante para el diseño de tratamientos más efectivos de esta enfermedad.

Como se mencionó en el capítulo 2, el modelo descrito en (2.3) involucra 6 parámetros que cuantifican la proporcionalidad entre los diferentes factores que actúan sobre la dinámica glucosa-insulina. En la tabla 3.1 se presentan nuevamente estos parámetros con sus respec-

tivos símbolo, descripción y unidades.

Prmt.	Descripción	Unidades
a_1	Tasa de eliminación de glucosa por su interacción con la insulina.	$1/(\mu\text{U}/\text{mL})/\text{hora}$
a_2	Tasa de liberación de glucosa desde reservas del hígado.	1/hora
a_3	Tasa de eliminación natural de la glucosa.	1/hora
a_4	Tasa de liberación de glucosa por ingesta de alimentos.	10 mg/mL/hora
b_1	Tasa de liberación de insulina por el páncreas después de comer.	100 $\mu\text{U}/\text{mg}/\text{hora}$
b_2	Tasa de eliminación natural de la insulina.	1/hora

Tabla 3.1: Símbolo, descripción y unidades de los parámetros usados en el sistema de ecuaciones diferenciales (2.3).

En este trabajo, para resolver el problema de control óptimo a partir de un conjunto de datos que representan las concentraciones de glucosa e insulina de un paciente a lo largo del día, se hace necesario contar con los valores de estos parámetros, puesto que son precisamente hechos biológicos estrechamente ligados a los parámetros como el factor alimenticio y la efectividad de la insulina por el páncreas los que permitirán encontrar la cantidad apropiada de insulina que debe ser suministrada de manera exógena a través de una bomba de dosificación o del dispositivo de administración del medicamento elegido para el tratamiento de esta patología.

3.2. Formulación del problema de mínimos cuadrados

El método de los mínimos cuadrados es frecuentemente usado para la estimación de un vector $\beta \in \Omega$, $\Theta \subseteq \mathbb{R}^p$, de parámetros desconocidos, que compone una función $f : [t_0, T] \times \Omega \rightarrow \mathbb{R}^n$ y al menos una vez diferenciable, la cual determina un modelo con variable de respuesta y [26, p. 165], es decir,

$$y = f(t, \beta), \quad (t, \beta) \in [t_0, T] \times \Omega. \quad (3.1)$$

Suponga que se requiere estimar las componentes b_1, b_2, \dots, b_p del vector de parámetros β , tal que la función f se ajuste de la mejor manera posible a una colección de observaciones dadas por la tabla

$$\mathbf{T} = \{(t_i, z_i) \in [t_0, T] \times \mathbb{R}^n \mid z_i = y_i + e_i, i = 1, \dots, L\}, \quad (3.2)$$

donde $e_i \in \mathbb{R}^n$ son los errores de las estimaciones con respecto al valor observado, y por lo tanto e es un vector columna de dimensión L , cuyas componentes son de la forma

$$e_i = z_i - y_i, \quad i = 1, \dots, L. \quad (3.3)$$

El problema de estimación de parámetros por mínimos cuadrados consiste en determinar un

vector $\hat{\beta}$ de tal manera que la función objetivo definida como

$$\phi(\beta) = \sum_{i=1}^L (z_i - f(t_i, \beta))^2 = \sum_{i=1}^L e_i^2, \quad (3.4)$$

sea mínima.

3.3. Estimación lineal y no lineal

En algunos casos, la función $y = f(t, \beta)$ puede expresarse en forma lineal con respecto a los parámetros, es decir,

$$y = \sum_{i=1}^p b_i \varphi_i(t), b_i \in \beta \quad (3.5)$$

con $\varphi_i : [t_0, T] \rightarrow \mathbb{R}^n$, razón por la que es conocida como estimación lineal de parámetros.

De la ecuación (3.5), se observa que los valores generados por la función definida en (3.1) pueden obtenerse a través del producto matricial

$$H\beta := \begin{pmatrix} \varphi_1(t_1) & \varphi_2(t_1) & \cdots & \varphi_p(t_1) \\ \varphi_1(t_2) & \varphi_2(t_2) & \cdots & \varphi_p(t_2) \\ \vdots & \vdots & \ddots & \vdots \\ \varphi_1(t_L) & \varphi_2(t_L) & \cdots & \varphi_p(t_L) \end{pmatrix} \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_p \end{pmatrix}, \quad (3.6)$$

donde H es una matriz por bloques $nL \times p$, con bloques de tamaño $n \times 1$.

Reemplazando (3.6) en (3.4) y tomando $z = (z_1, z_2, \dots, z_L)^T$, se obtiene finalmente:

$$\phi(\beta) = (z - H\beta)^T (z - H\beta) = \|z - H\beta\|_2^2. \quad (3.7)$$

Como se desea minimizar la función objetivo ϕ , desde el punto de vista algebraico el problema se centra en buscar un vector $\hat{\beta}$ en \mathbb{R}^p , tal que $H\hat{\beta}$ sea lo más cercano a z como sea posible. Suponiendo que W es el espacio generado por las columnas de H , entonces del álgebra lineal es conocido que el vector en W más cercano a z es $\text{Proy}_W z$, esto es, $\|z - w\|$, para w en W se minimiza cuando $w = \text{proy}_W z$. En consecuencia, al encontrar $\hat{\beta}$ tal que $H\hat{\beta} = \text{Proy}_W z$, con seguridad se tiene que $\|z - H\hat{\beta}\|_2^2$ será lo más pequeña posible. Finalmente, en términos de una ecuación matricial

$$H^T(H\hat{\beta} - z) = 0,$$

o de manera equivalente,

$$H^T H \hat{\beta} = H^T z, \quad (3.8)$$

conocidas como las ecuaciones normales de Gauss. Despejando $\hat{\beta}$ de la ecuación (3.8), se

obtiene

$$\hat{\beta} = (H^T H)^{-1} H^T z. \quad (3.9)$$

Teorema 2. Sea $H \in \mathbb{R}^{nL \times p}$, $z \in \mathbb{R}^{nL}$ con $Ln \geq p$.

$$\|z - H\hat{\beta}\|_2^2 = \min_{\beta \in \Omega} \|z - H\beta\|_2^2,$$

si y solo si $\hat{\beta}$ soluciona las ecuaciones normales de Gauss $H^T H\hat{\beta} = H^T z$.

Demostración:

Sea $\phi(\beta) := \|z - H\beta\|_2^2$ y $\hat{\beta}$ un minimizador de la función $\phi(\beta)$. Se escoge $\alpha \in \mathbb{R}^{nL}$ y $\varepsilon \neq 0$, por lo tanto

$$\phi(\hat{\beta} + \varepsilon\alpha) \geq \phi(\hat{\beta}) \quad (3.10)$$

$$\begin{aligned} \phi(\hat{\beta} + \varepsilon\alpha) &= \|z - H(\hat{\beta} + \varepsilon\alpha)\|_2^2 \\ &= (z - H(\hat{\beta} + \varepsilon\alpha))^T (z - H(\hat{\beta} + \varepsilon\alpha)) \\ &= ((z - H\hat{\beta}) - \varepsilon H\alpha)^T ((z - H\hat{\beta}) - \varepsilon H\alpha) \\ &= (z - H\hat{\beta})^T (z - H\hat{\beta}) - \varepsilon (z - H\hat{\beta})^T H\alpha - \varepsilon (H\alpha)^T (z - H\hat{\beta}) + \varepsilon^2 (H\alpha)^T H\alpha \\ &= \|z - H\hat{\beta}\|_2^2 - 2\varepsilon (z - H\hat{\beta})^T H\alpha + \varepsilon^2 \|H\alpha\|_2^2. \end{aligned}$$

Como $\hat{\beta}$ es un minimizador de ϕ , es necesario que $(z - H\hat{\beta})^T H\alpha = 0$, para todo $\alpha \in \mathbb{R}^{nL}$, con el objetivo de mantener el sentido de la desigualdad planeada en (3.10).

Por lo tanto

$$\begin{aligned} [(z - H\hat{\beta})^T H\alpha]^T &= 0, \quad \text{para todo } \alpha \in \mathbb{R}^{nL}, \\ \alpha^T H^T (z - H\hat{\beta}) &= 0, \quad \text{para todo } \alpha \in \mathbb{R}^{nL}, \\ \alpha^T (H^T z - H^T H\hat{\beta}) &= 0, \quad \text{para todo } \alpha \in \mathbb{R}^{nL}. \end{aligned}$$

Luego

$$H^T z - H^T H\hat{\beta} = 0, \quad \text{y} \quad H^T z = H^T H\hat{\beta},$$

y se concluye que $\hat{\beta}$ soluciona las ecuaciones normales de Gauss.

Ahora, supóngase que $\hat{\beta}$ soluciona las ecuaciones normales de Gauss, así que:

$$H^T z - H^T H\hat{\beta} = 0 \quad \text{y} \quad H^T (z - H\hat{\beta}) = 0. \quad (3.11)$$

De la igualdad (3.11) se puede concluir que $H\hat{\beta} - z$ es ortogonal a todo vector H_j^T de W espacio columna de H . Por otro lado, sea $w = \text{proy}_W z$, entonces $z - w$ también es ortogonal a todo vector en W . Así, igualando esta última expresión con (3.11)

$$(z - w) = z - H\hat{\beta}. \quad (3.12)$$

De (3.12) se concluye que $w = H\hat{\beta}$ y por lo tanto $H\hat{\beta} = \text{Proy}_w z$, y como $\|z - w\|$ se minimiza cuando $w = \text{Proy}_w z$, entonces $\|z - H\hat{\beta}\|$ es lo más pequeña posible, por lo cual $\hat{\beta}$ es un minimizador de $\|z - H\beta\|$. ■

Cuando la función $y = f(t, \beta)$ no es lineal con respecto a los parámetros, se tiene un problema de estimación no lineal. Estos problemas no lineales se abordan numéricamente mediante la aplicación de métodos iterativos. De acuerdo con [2], un método iterativo para aproximar $\hat{\beta}$ donde se minimiza la función objetivo $\phi(\beta)$ dada en (3.7), consiste en generar una sucesión de vectores $\{\beta_k\}_{k \geq 0}$ a partir de un vector β_0 llamado vector inicial de parámetros, la cual se espera que converja al vector de parámetros $\hat{\beta}$ buscado. Una sucesión converge después de un número finito de N iteraciones, al satisfacer un criterio de parada impuesto y se acepta $\hat{\beta} = \beta_N$. El vector $\sigma_k = \beta_{k+1} - \beta_k$ se le llama el **k -ésimo paso**, y este se dice aceptable si satisface que $\phi(\beta_{k+1}) < \phi(\beta_k)$.

3.4. Método de Gauss–Newton

La idea central de este método es aproximar un problema de mínimos cuadrados no lineales a un problema de mínimos cuadrados lineales, alrededor de una aproximación β_k conocida de la solución. Inicialmente, dada una función $f(\beta)$ con $f : [t_0, T] \times \mathbb{R}^n \rightarrow \mathbb{R}^n$ y al menos una vez diferenciable, se construye una linealización de esta usando una expansión en serie de Taylor de primer orden

$$f(\beta_k + \Delta\beta) = f(\beta_k) + \frac{\partial f}{\partial \beta^T} \Delta\beta = f(\beta_k) + J(\beta_k)\Delta\beta, \quad (3.13)$$

donde β_k es el vector columna, de tamaño p , de parámetros en la iteración k y J es la matriz jacobiana de $f(\beta_k)$ (con tamaño $nL \times p$, si $f : [0, T] \times \mathbb{R}^p \rightarrow \mathbb{R}^n$), es decir, cada uno de los elementos de J son calculados por

$$[J]_{i,j} = \frac{\partial f_i}{\partial b_j}, \quad \begin{array}{l} i = 1, \dots, n \\ j = 1, \dots, p. \end{array}$$

La estimación $k + 1$ del vector de parámetros desconocidos β_{k+1} , es obtenida a partir de la relación

$$\beta_{k+1} = \beta_k + \Delta\beta, \quad (3.14)$$

donde β_k es la estimación del vector de parámetros desconocidos en la iteración k y $\Delta\beta$ es la estimación por mínimos cuadrados necesaria para la actualización del vector β de la iteración k a la iteración $k + 1$, la cual se obtiene como sigue a continuación.

Sea z el vector de datos observados, el error sobre estas observaciones es dado por el siguiente sistema lineal de ecuaciones de tamaño $nL \times p$

$$z - f(\beta_k + \Delta\beta) = z - f(\beta_k) - J(\beta_k)\Delta\beta = e_k - J(\beta_k)\Delta\beta.$$

Como se quiere minimizar este error, aplicando la estimación por mínimos cuadrados lineal introducida en las ecuaciones (3.4) a (3.9), se tiene que

$$\Delta\beta_k = (J(\beta_k)^T J(\beta_k))^{-1} J(\beta_k)^T e_k. \quad (3.15)$$

En resumen, el método de Gauss–Newton usa la ecuación de iteración (3.14) donde $\Delta\beta_k$ es una estimación por mínimos cuadrados obtenida a partir de la ecuación (3.15).

Del razonamiento presentado anteriormente, se puede deducir el esquema iterativo del Algoritmo 2.

Algoritmo 2: MÉTODO DE GAUSS-NEWTON

$\beta_0 = (b_1^0, b_2^0, \dots, b_p^0)$ Vector inicial de parámetros.

$z = (z_1, z_2, \dots, z_L)$ Vector de valores observados.

$t = [t_0, T]$ Intervalo de tiempo.

$y = f(t, \beta)$

Entrada: $\beta_0, z, f(t, \beta)$.

Salida: β_{k+1} .

Inicio

Mientras $\|\Delta\beta_k\| > \text{Criterio de parada}$ **hacer**

 Calcular $[J]_{i,j} \leftarrow \frac{\partial f_i}{\partial b_j}$

 Calcular $e_k \leftarrow z - f(t, \beta_k)$

 Resolver $\Delta\beta_k \leftarrow (J(\beta_k)^T J(\beta_k))^{-1} J(\beta_k)^T e_k$

 Actualizar $\beta_{k+1} \leftarrow \beta_k + \Delta\beta_k$

 Calcular $\Delta\beta_k$

Fin

Devolver β_{k+1}

Fin

Los teoremas y demostraciones que garantizan la convergencia del método de Gauss–Newton pueden ser consultadas en [16, p. 221] o en [37, p. 255].

El método de Gauss–Newton es la técnica más popular para resolver numéricamente problemas de mínimos cuadrados no lineales, sin embargo, para su aplicación es necesario tener en cuenta los siguientes aspectos que pueden representar algunas dificultades o desventajas:

- El método falla si la matriz jacobiana $J(\beta_k^*)$ es singular o casi singular.
- Para garantizar la convergencia del método a un mínimo local, es necesario proporcionar al algoritmo una primera aproximación a $\hat{\beta}$, notada como β_0 , lo más cercana que sea posible.

La primera dificultad es posible superarla con el método de Levenberg-Marquardt [2], mientras que para la segunda se propone una alternativa que permita el cálculo de las componentes del vector β_0 . En los siguientes dos apartados, serán desarrolladas con más detalle estas ideas.

3.5. Método de Levenberg-Marquardt

Este algoritmo consiste en variar adaptativamente las actualizaciones del vector de parámetros, por medio de la inclusión de una componente $\mu_k B$ en la dirección de descenso, es decir

$$d_k = [J^T(\beta_k)J(\beta_k) + \mu_k B]^{-1} J^T(\beta_k) e_k, \quad (3.16)$$

donde $\mu_k \geq 0$ y $B \in \mathbb{R}^{p \times p}$ es una matriz simétrica y definida positiva, que en la mayoría de los casos es una matriz identidad I de orden $p \times p$.

El término μ es llamado parámetro de amortiguación, y este se inicializa en un valor cercano a cero que incrementa o disminuye en cada paso. Si $\|\Delta\beta_k\| \leq \|\Delta\beta_{k+1}\|$, entonces el valor de μ se multiplica por un factor de $\gamma \gg 1$, y en caso contrario, el parámetro μ se divide por γ [22] y su valor disminuye, por lo que el método de Levenberg-Marquardt se aproxima al método de Gauss-Newton y la solución se acelera al mínimo local buscado. El pseudocódigo que ilustra el esquema iterativo de este método numérico puede consultarse en el Algoritmo 3.

El comportamiento de su convergencia local es similar al del método de Gauss-Newton [2], y los teoremas y demostraciones que lo garantizan pueden consultarse en [37, p. 285].

Algoritmo 3: MÉTODO DE LEVENBERG-MARQUARDT.

$\beta_0 = (b_1^0, b_2^0, \dots, b_p^0)$ Vector inicial de parámetros.

$z = (z_1, z_2, \dots, z_L)$ Vector de valores observados.

$t = [t_0, T]$ Intervalo de tiempo.

$y = f(t, \beta)$

Entrada: $\beta_0, z, f(t, \beta)$.

Salida: β_{k+1} .

Inicio

$e1 \leftarrow 1$

Mientras $e1 > \text{Criterio de parada}$ **hacer**

 Calcular $[J]_{i,j} \leftarrow \frac{\partial f_i}{\partial b_j}$

 Calcular $e_k \leftarrow z - f(t, \beta_k)$

$e2 \leftarrow 1$

Mientras $e2 \geq e1$ **hacer**

 Resolver

$\Delta\beta_k \leftarrow (J(\beta_k)^T J(\beta_k) + \mu_k B)^{-1} J(\beta_k)^T e_k$

El método más utilizado para resolver numéricamente problemas de mínimos cuadrados no lineales, es el método de Gauss-Newton combinado con el método de Levenberg-Marquardt. Sin embargo, en el caso de la estimación de parámetros de ecuaciones diferenciales, para realizar la evaluación de la función objetivo definida en (3.19), así como el cálculo del jacobiano $J(\phi)$ que se utiliza en la aplicación del método de Gauss-Newton a la minimización de la función objetivo, es necesario integrar el problema de valores iniciales (3.18) desde t_0 hasta $t_L = T$. De esta manera se determinan las componentes del vector $y(t_i)$ para $i = 1, \dots, L$.

A continuación se describe con detalle cada uno de los pasos requeridos para la aplicación del método de Gauss-Newton a la estimación de parámetros de ecuaciones diferenciales:

1. Dado $\beta_0 = (b_1^0, b_2^0, \dots, b_p^0)$ se resuelve numéricamente el problema de valor inicial

$$y' = f(t, y, \beta_0), \quad y(t_0) = y_0, \quad (3.20)$$

en el intervalo $[0, T]$, donde $t_1, t_2, \dots, t_{L-1} \in [t_0, T]$, con el fin de obtener los valores $y(t_i) = \rho(t_i, \beta_0)$, $i = 1, \dots, L$.

2. Si $\phi(\beta_0)$ es el mínimo o satisface algún criterio de parada, el proceso termina. En caso contrario, se calcula $\Delta\beta_0$ como solución, según (3.15), de:

$$J^T(\beta_0)J(\beta_0)\Delta\beta_0 = J^T(\beta_0)e_0,$$

donde el jacobiano $J(\beta_0)$ está dado por

$$J(\beta_0) = \begin{bmatrix} \frac{\partial \rho}{\partial \beta}(t_1, \beta_0) \\ \frac{\partial \rho}{\partial \beta}(t_2, \beta_0) \\ \vdots \\ \frac{\partial \rho}{\partial \beta}(t_L, \beta_0) \end{bmatrix} \in \mathbb{R}^{nL \times p}. \quad (3.21)$$

Si ρ es una solución al problema de valores iniciales (3.18), con $y(t_0) = y_0$, entonces por el Teorema Fundamental del Cálculo,

$$\rho(t, \beta) = y_0 + \int_{t_0}^t f(v, \rho(v, \beta), \beta) dv. \quad (3.22)$$

Derivando (3.22) con respecto a β se obtiene

$$\frac{\partial \rho}{\partial \beta} = \int_{t_0}^t \left(\frac{\partial f}{\partial y} \frac{\partial \rho}{\partial \beta} + \frac{\partial f}{\partial \beta} \right) dv, \quad (3.23)$$

donde el valor de $\frac{\partial \rho}{\partial \beta}(t_0) = 0$ siempre que el valor inicial $y(t_0) = y_0$ sea independiente

del vector de parámetros β , como sucede en este caso. Finalmente, si se deriva (3.23) con respecto a t , se obtiene un sistema de ecuaciones diferenciales lineales de primer orden con valores iniciales, dado por

$$\rho'_\beta = f_y \rho_\beta + f_\beta, \quad \rho_\beta(t_0) = 0, \quad (3.24)$$

$$\text{con } \rho'_\beta = \frac{d}{dt} \left[\frac{\partial \rho}{\partial \beta} \right]; f_y = \frac{\partial f}{\partial y}; \rho_\beta = \frac{\partial \rho}{\partial \beta} \text{ y } f_\beta = \frac{\partial f}{\partial \beta}.$$

La incógnita ρ_β de (3.24) es una matriz que se compone justamente de las funciones que son necesarias determinar en (3.21). El sistema de ecuaciones (3.24) se conoce también como el sistema de ecuaciones auxiliares para estimación de parámetros [26]. Si $y = (y_1, y_2, \dots, y_n)^T$, $f = (f_1, f_2, \dots, f_n)^T$ y $\beta = (b_1, b_2, \dots, b_p)^T$ entonces el sistema (3.24) puede descomponerse en p sub-sistemas que es posible escribir de forma matricial como

$$\frac{d}{dt} \begin{bmatrix} \frac{\partial \rho_1}{\partial b_i} \\ \frac{\partial \rho_2}{\partial b_i} \\ \vdots \\ \frac{\partial \rho_n}{\partial b_i} \end{bmatrix} = \begin{bmatrix} \frac{\partial f_1}{\partial y_1} & \frac{\partial f_1}{\partial y_2} & \dots & \frac{\partial f_1}{\partial y_n} \\ \frac{\partial f_2}{\partial y_1} & \frac{\partial f_2}{\partial y_2} & \dots & \frac{\partial f_2}{\partial y_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_n}{\partial y_1} & \frac{\partial f_n}{\partial y_2} & \dots & \frac{\partial f_n}{\partial y_n} \end{bmatrix} \begin{bmatrix} \frac{\partial \rho_1}{\partial b_i} \\ \frac{\partial \rho_2}{\partial b_i} \\ \vdots \\ \frac{\partial \rho_n}{\partial b_i} \end{bmatrix} + \begin{bmatrix} \frac{\partial f_1}{\partial b_i} \\ \frac{\partial f_2}{\partial b_i} \\ \vdots \\ \frac{\partial f_n}{\partial b_i} \end{bmatrix}, \quad i = 1, \dots, p, \quad (3.25)$$

con valores iniciales

$$\begin{bmatrix} \frac{\partial \rho_1}{\partial b_i}(t_0) \\ \frac{\partial \rho_2}{\partial b_i}(t_0) \\ \vdots \\ \frac{\partial \rho_n}{\partial b_i}(t_0) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}. \quad (3.26)$$

3. Se resuelven simultáneamente los problemas de valor inicial (3.18) y (3.24), lo cual implica resolver numéricamente $n(p+1)$ ecuaciones para cada $t_i = 1, \dots, L$.
4. Una vez determinado $\beta_1 = (b_1^1, b_2^1, \dots, b_p^1)$ se resuelve el nuevo problema de valores iniciales

$$y' = f(t, y, \beta_1), \quad y(t_0) = y_0, \quad (3.27)$$

sobre el intervalo $[t_0, T]$ donde $t_1, t_2, \dots, t_{L-1} \in [t_0, T]$, con el fin de obtener los valores de $y(t_i) = \rho(t_i, \beta_1)$, $i = 1, \dots, L$.

5. Si $\phi(\beta_1)$ es el mínimo o satisface algún criterio de parada el proceso termina, y si no es así, se repite todo el proceso anteriormente descrito para determinar un nuevo valor

$\Delta\beta_1 = \beta_0 + \Delta\beta_0$ y así encontrar $\beta_2 = \beta_1 + \Delta\beta_1$.

6. El proceso continúa hasta lograr la convergencia.

3.7. Cálculo del valor inicial de los parámetros

Como ya se había mencionado, los métodos usados en la solución de problemas de mínimos cuadrados no lineales requiere de un vector inicial β_0 a partir del cual se generan las aproximaciones β_k que sucesivamente convergen al vector $\hat{\beta}$ buscado. Para que el método converja, se requiere además la selección de β_0 lo más cercana a $\hat{\beta}$ que sea posible. Así pues, se propone una estrategia adaptada de la presentada en [2, p. 34] para calcular las componentes del vector inicial β_0 , que consiste en:

1. Ajustar los datos observados con un spline cúbico, es decir, dado el conjunto de los datos observados $\{(t_0, z_0), (t_1, z_1), \dots, (t_L, z_L)\}$ se seleccionan r elementos, de los cuales sus primeras componentes, denotadas ahora como $\xi_1, \xi_2, \dots, \xi_r$ se usan como nodos para la construcción de un spline cúbico de la forma:

$$s(t) := \begin{cases} q_1(t), & \text{si } \xi_1 \leq t < \xi_2, \\ q_2(t), & \text{si } \xi_2 \leq t < \xi_3, \\ \vdots & \\ q_{r-1}(t), & \text{si } \xi_{r-1} \leq t \leq \xi_r. \end{cases}$$

Los nodos deben escogerse cuidadosamente, procurando que el spline describa de la mejor manera posible la tendencia de los datos.

2. Una vez construido el spline cúbico, se determina un β_0 tal que minimice la función objetivo

$$\phi(\beta) = \left(\sum_{i=1}^{r-1} (s'_i(\hat{t}_i) - f(\hat{t}_i, s(\hat{t}_i), \beta)) \right)^2, \quad (3.28)$$

para un conjunto de datos muestrales $\hat{t}_1, \hat{t}_2, \dots, \hat{t}_{L-1}$ seleccionados al azar de los intervalos $(t_0, t_1), (t_1, t_2), \dots, (t_{L-1}, t_L)$ respectivamente.

Si el modelo de ecuaciones diferenciales (3.18) es lineal con respecto al vector de parámetros β , es decir, puede ser escrito de manera semejante a la expresión (3.5), entonces la función objetivo (3.28) se puede minimizar al plantear un problema de mínimos cuadrados lineal y usando la ecuación (3.8). Por lo tanto, para cada ecuación

$$y'_k = \sum_{i=1}^{M_j} b_i \varphi_i,$$

que hace parte del sistema que componen el PVI (3.18), entonces

$$B_i = (H_i^T H_i)^{-1} H_i w,$$

donde

$$B_i = \begin{bmatrix} \varphi_1(t_1) & \varphi_2(t_1) & \cdots & \varphi_{M_j}(t_1) \\ \varphi_1(t_2) & \varphi_2(t_2) & \cdots & \varphi_{M_j}(t_2) \\ \vdots & \vdots & \ddots & \vdots \\ \varphi_1(t_L) & \varphi_2(t_L) & \cdots & \varphi_{M_j}(t_L) \end{bmatrix}; \quad w_i = \begin{bmatrix} s'(t_1) \\ s'(t_2) \\ \vdots \\ s'(t_L) \end{bmatrix},$$

para $i = 1, \dots, n$, de tal manera que $\sum M_j = p$ y $\beta_0 = (B_1, B_2, \dots, B_n)^T$. En la sección que sigue a continuación, se aplicará la idea aquí presentada para el hallazgo de vector inicial de parámetros β_0 al modelo (2.3) que describe la dinámica glucosa-insulina.

3.8. Experimentación numérica

En esta sección se realiza el proceso de la estimación de parámetros para el modelo de ecuaciones diferenciales que describe la dinámica glucosa-insulina, presentado en (2.3), usando el método de Gauss-Newton combinado con el método de Levenberg-Marquardt.

Los datos experimentales se obtienen al resolver (2.3) numéricamente, dando un valor que se considera como “*el valor real*” para cada uno de los parámetros del sistema de ecuaciones en mención, en el intervalo $[t_0, T]$ con $t_0 = 0$ y $T = 24$ horas. Posteriormente, estos datos son perturbados al agregar un término generado aleatoriamente que suma o resta con los datos obtenidos inicialmente, de tal manera que si $g(t)$ representa una de las concentraciones de glucosa obtenida de la integración numérica de (2.3) para un instante de tiempo $t \in [0, 24]$ y ε corresponde a la concentración de glucosa para ese mismo instante de tiempo pero perturbada, entonces $|g(t) - \varepsilon| \leq \pm 10^{-3}$. Los datos perturbados se consideran como los datos observados para cada uno de los tiempos evaluados en la integración numérica, y sus valores serán guardados en los vectores columna z_1 para la glucosa y z_2 para la insulina. En la sección B.2 del apéndice que incluye este trabajo, el lector podrá consultar los códigos de programación que resuelven el sistema de ecuaciones y construyen los vectores de datos perturbados, nombrados como `Sol_SistGH.m` y `randomData.m`.

El vector de parámetros a estimar es $\beta = (a_1, a_2, a_3, a_4, b_1, b_2)^T$ y el vector de los errores en los datos observados está dado por

$$e = \begin{bmatrix} z_1(t_1) - g(t_1, \beta) \\ z_2(t_2) - h(t_2, \beta) \\ \vdots \\ z_1(t_L) - g(t_L, \beta) \\ z_2(t_L) - h(t_L, \beta) \end{bmatrix}_{2L \times 1}. \quad (3.29)$$

Para aplicar el método de Gauss-Newton siguiendo el algoritmo 3 presentado en la sección 3.4, se calcula además la matriz jacobiana J de las funciones que componen el sistema (2.3), cuyas entradas son:

$$J(\beta) = \begin{bmatrix} \frac{\partial g(t_1)}{\partial a_1} & \frac{\partial g(t_1)}{\partial a_2} & \frac{\partial g(t_1)}{\partial a_3} & \frac{\partial g(t_1)}{\partial a_4} & \frac{\partial g(t_1)}{\partial b_1} & \frac{\partial g(t_1)}{\partial b_2} \\ \frac{\partial h(t_1)}{\partial a_1} & \frac{\partial h(t_1)}{\partial a_2} & \frac{\partial h(t_1)}{\partial a_3} & \frac{\partial h(t_1)}{\partial a_4} & \frac{\partial h(t_1)}{\partial b_1} & \frac{\partial h(t_1)}{\partial b_2} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \frac{\partial g(t_L)}{\partial a_1} & \frac{\partial g(t_L)}{\partial a_2} & \frac{\partial g(t_L)}{\partial a_3} & \frac{\partial g(t_L)}{\partial a_4} & \frac{\partial g(t_L)}{\partial b_1} & \frac{\partial g(t_L)}{\partial b_2} \\ \frac{\partial h(t_L)}{\partial a_1} & \frac{\partial h(t_L)}{\partial a_2} & \frac{\partial h(t_L)}{\partial a_3} & \frac{\partial h(t_L)}{\partial a_4} & \frac{\partial h(t_L)}{\partial b_1} & \frac{\partial h(t_L)}{\partial b_2} \end{bmatrix}_{2L \times 6}. \quad (3.30)$$

Los elementos de esta matriz jacobiana son obtenidos al resolver numéricamente el sistema auxiliar de ecuaciones (3.24), donde para este caso particular

$$f_y = \begin{bmatrix} -a_1 h + a_2 \delta(-\bar{g}) - a_3 \delta(\bar{g}) & -a_1 g \\ b_1 \delta(\bar{g}) & -b_2 \end{bmatrix}, \quad (3.31)$$

$$\rho_\beta = \begin{bmatrix} \frac{\partial g}{\partial a_1} & \frac{\partial g}{\partial a_2} & \frac{\partial g}{\partial a_3} & \frac{\partial g}{\partial a_4} & \frac{\partial g}{\partial b_1} & \frac{\partial g}{\partial b_2} \\ \frac{\partial h}{\partial a_1} & \frac{\partial h}{\partial a_2} & \frac{\partial h}{\partial a_3} & \frac{\partial h}{\partial a_4} & \frac{\partial h}{\partial b_1} & \frac{\partial h}{\partial b_2} \end{bmatrix},$$

$$f_\beta = \begin{bmatrix} -gh & (g_0 - g)^+ & -(g + g_0)^+ & z(t - \tau_i) & 0 & 0 \\ 0 & 0 & 0 & 0 & (g - g_0)^+ & -h \end{bmatrix}, \quad (3.32)$$

con $\bar{g} = (g - g_0)$ y

$$\delta(\bar{g}) = \begin{cases} 1, & \text{si } \bar{g} > 0, \\ 0, & \text{si } \bar{g} \leq 0. \end{cases}$$

Como ya se comentó anteriormente, el sistema de ecuaciones auxiliares puede ser reescribirse como puede observarse en (3.25), donde para este caso, resultan 6 sub-sistemas de ecuaciones diferenciales que deben ser resueltos junto con el sistema (2.3). Uno de estos sistemas es

$$\frac{d}{dt} \begin{bmatrix} \frac{\partial g}{\partial a_1} \\ \frac{\partial h}{\partial a_1} \end{bmatrix} = \begin{bmatrix} -a_1 h + a_2 \delta(-\bar{g}) - a_3 \delta(\bar{g}) & -a_1 g \\ b_1 \delta(\bar{b}) & -b_2 \end{bmatrix} \begin{bmatrix} \frac{\partial g}{\partial a_1} \\ \frac{\partial h}{\partial a_1} \end{bmatrix} + \begin{bmatrix} -gh \\ 0 \end{bmatrix}, \quad (3.33)$$

con condiciones iniciales

$$\begin{bmatrix} \frac{\partial g}{\partial a_1}(t_0) \\ \frac{\partial h}{\partial a_1}(t_0) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}.$$

De manera análoga pueden escribirse las 8 sub-ecuaciones restantes.

Usando la aproximación inicial del vector de parámetros β_0 , se integran los sistemas (2.3) y (3.31) por el método de Runge-Kutta desde $t_0 = 0$ hasta $t_L = T$, y los valores resultantes del proceso se utilizan en la construcción de la matriz jacobiana definida en (3.30). El vector de errores (3.29) junto con la matriz jacobiana, se utilizan para calcular la estimación de $\Delta\beta_k$ tal como se definió en (3.15) y se calcula una nueva aproximación del vector de parámetros usando la expresión dada en (3.14). El ciclo se repite hasta cumplir el criterio de parada, empezando nuevamente por la integración de los sistemas de ecuaciones (2.3) y (3.24), utilizando ahora como vector de parámetros la aproximación encontrada en la iteración anterior.

Para el cálculo del vector inicial de parámetros β_0 se emplea el método descrito en la sección 3.7, usando como parámetros exactos los utilizados en la simulación presentada en la sección 2.3.1. Aquí $\beta_0 = (B_1, B_2)^T$, donde $B_1 = (a_1, a_2, a_3, a_4)^T$ y $B_2 = (b_1, b_2)^T$. En la figura 3.1 se presenta un diagrama de flujo que resume cada una de las etapas que se ejecutan para la estimación de los parámetros del sistema (2.3).

En la sección B.2 del apéndice que se incluye en este trabajo, se puede consultar el script que contiene el sistema de ecuaciones auxiliares (3.31) nombrado como `AUX_SYSTEM.m`, además de las funciones `RK4_solAuxSystem.m`, `ParametersEstimate.m` y `EstPrmtInitial.m` que resuelven el sistema de ecuaciones (3.24), implementa el método de Gauss-Newton para la estimación del vector de parámetros $\hat{\beta}$, y calcula el vector de parámetros inicial β_0 respectivamente. El script nombrado como `PrincipalCap3` contiene las instrucciones que son necesarias para invocar en el lenguaje de programación a cada una de las funciones mencionadas anteriormente, además de establecer los valores de los parámetros que deben ser pasados a cada una de estas funciones una vez son invocadas. En esta simulación se realiza la estimación de parámetros para la dinámica glucosa-insulina de un paciente sano y de un paciente diabético tipo 2, se presentan y discuten los resultados encontrados en el siguiente apartado del presente capítulo. Estos experimentos fueron programados y ejecutados en MATLAB® versión R2019a, instalado en un computador con procesador AMD® Ryzen™ 5 2500 U 3.6 GHz y una memoria RAM de 8 GB.

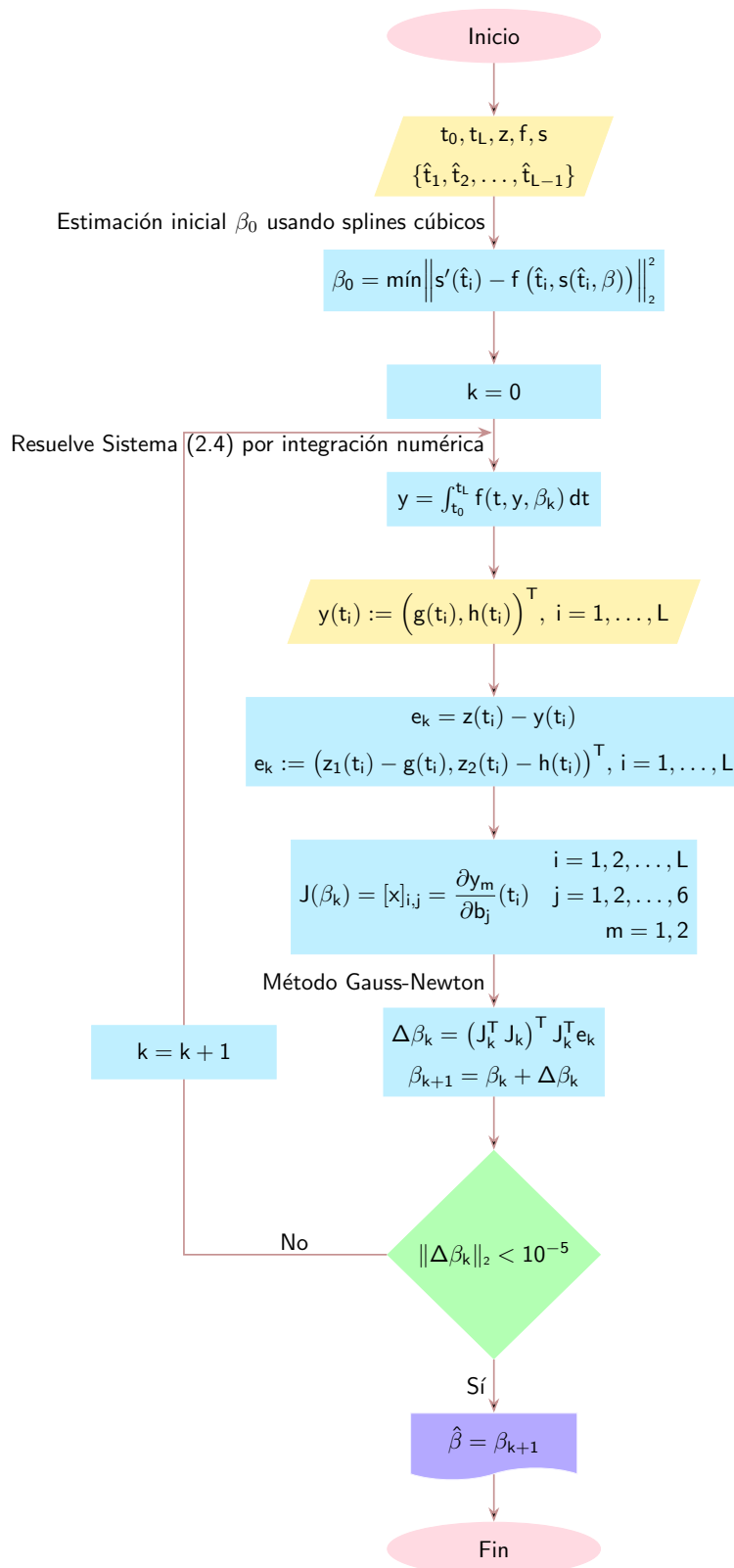


Figura 3.1: Diagrama de flujo del proceso de estimación de parámetros para el modelo de la dinámica glucosa-insulina (2.3).

3.8.1. Experimento 1: paciente sano

Parámetro	Valor exacto	β Inicial (β_0)	β estimado ($\hat{\beta}$)	Error Relativo
a_1	0.05	0.0482	0.0504	0.8 %
a_2	1	1.5953	1.0953	9.53 %
a_3	0.03	0.1618	0.0295	2 %
a_4	4	4.0947	4.0117	0.29 %
b_1	0.5	0.4961	0.5003	0.06 %
b_2	2	2.0091	2.0019	0.095 %

Tabla 3.2: Resultados de la estimación de parámetros para el sistema (2.3) por el método Gauss-Newton para la dinámica de un paciente sano. El vector β_0 hace referencia al vector que contiene los valores iniciales de los parámetros para resolver el problema de estimación. El vector $\hat{\beta}$ contiene los parámetros estimados una vez se logra la convergencia del algoritmo de optimización.

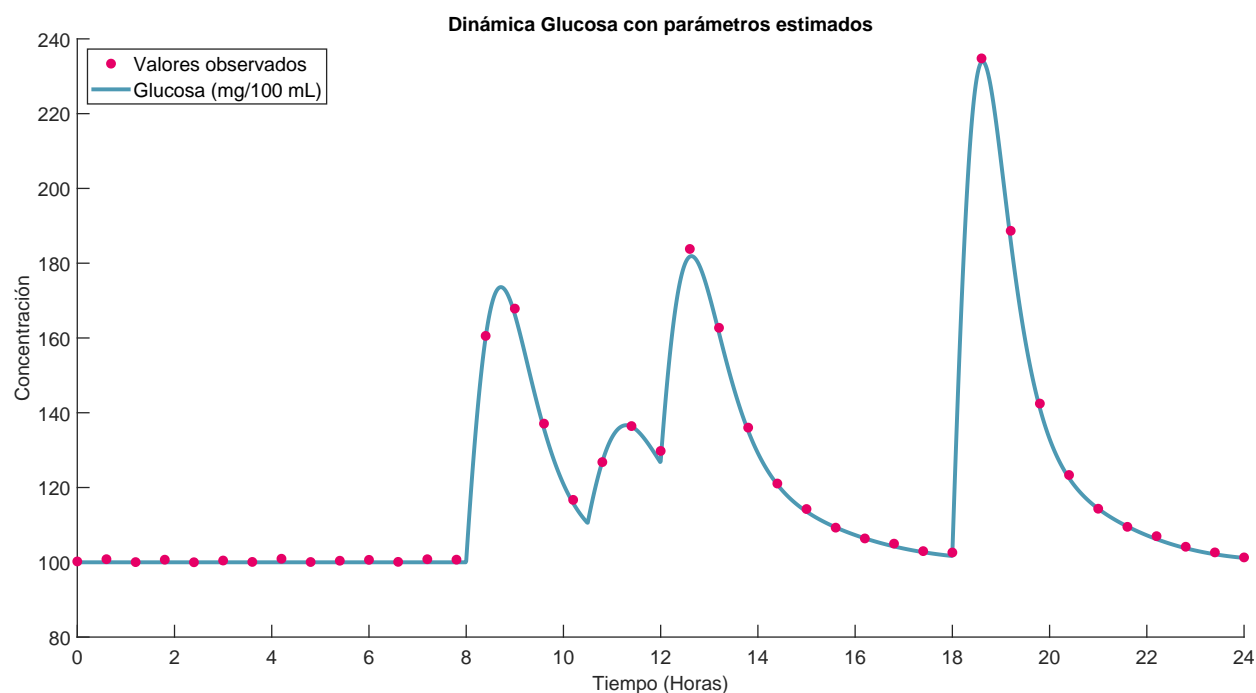


Figura 3.2: Resultados gráficos obtenidos de la estimación de parámetros para la dinámica de la glucosa de un paciente sano.

Por la dinámica del comportamiento de los niveles de glucosa e insulina en sangre, según los aspectos mencionados en la sección 2.3.1, y por el valor que toma en este caso el parámetro b_1 , que corresponde a la tasa de liberación de insulina por el páncreas (ver tabla 3.1), es posible afirmar que los parámetros estimados anteriormente se encuentran asociados al sistema de ecuaciones (2.3) los cuales permiten describir la dinámica de un paciente sano.

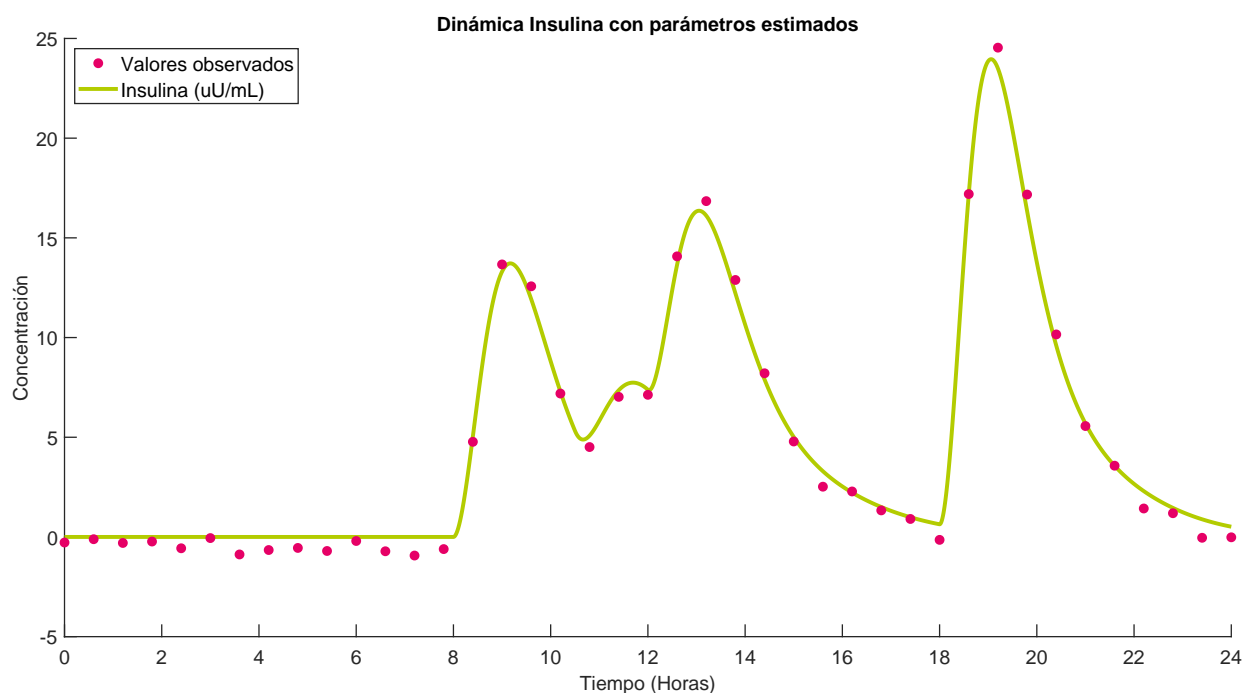


Figura 3.3: Resultados gráficos obtenidos de la estimación de parámetros para la dinámica de la insulina de un paciente sano.

En la tabla 3.2 se puede observar que el porcentaje del error relativo, para cada uno de los parámetros, es pequeño en la mayoría de los casos, e incluso se puede apreciar que el error más grande no supera el 10%. La combinación con el método de Levenberg-Marquardt resulta ser bastante útil para la estimación de los parámetros, puesto que el término que acompaña al parámetro a_2 en el modelo de ecuaciones diferenciales (2.3) provoca que para este caso particular la mayoría de componentes de la columna que corresponde a la derivada de la glucosa con respecto al parámetro a_2 $\left(\frac{\partial g}{\partial a_2}\right)$ en la matriz jacobiana sean cero, puesto que son pocas las ocasiones en las que se requiere la liberación de la glucosa reservada en el hígado al ser este el caso de una persona que consume tres comidas durante el día. El hecho de que las componentes de una de las columnas de la matriz jacobiana sean en su mayoría o todas cero, trae problemas con la implementación del método Gauss-Newton, debido a que la matriz jacobiana será singular o casi singular y por lo tanto no podría invertirse, problema que es superado, como se mencionó en la sección 3.5, al incorporar el término $\mu_k B$ en el esquema iterativo del método Gauss-Newton, generando así finalmente la convergencia al satisfacer el criterio de parada impuesto.

De igual manera, la estrategia propuesta para el hallazgo de los valores iniciales o vector inicial de parámetros β_0 , al componerse de valores muy cercanos a los valores “reales” de los parámetros, permitió no solo que el error relativo para cada uno de ellos fuese pequeño y que el criterio de paro se cumpliera en tan solo 12 iteraciones, sino además una convergencia bastante rápida en términos del tiempo que empleó el computador en realizar todos los cálculos necesarios en el algoritmo, que fue de tan solo 4.38 segundos.

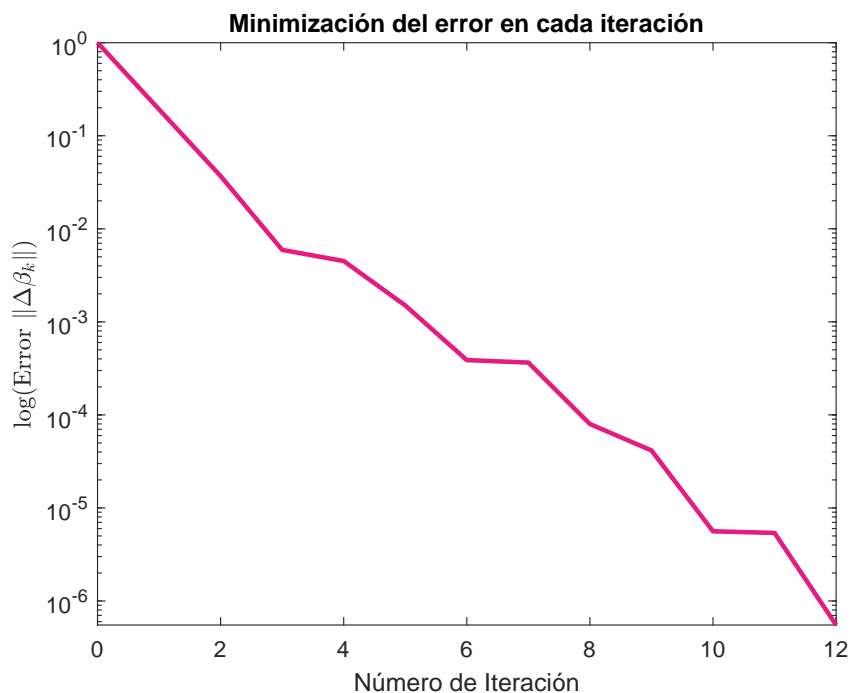


Figura 3.4: Gráfica del número de iteraciones vs. error de aproximación del método Gauss-Newton aplicado a la estimación de parámetros de la dinámica glucosa-insulina.

3.8.2. Experimento 2: paciente diabético

Parámetro	Valor exacto	β Inicial (β_0)	β estimado $\hat{\beta}$	Error Relativo
a_1	0.05	0.0861	0.0519	1.8 %
a_2	1	1.3829	0.9471	5.29 %
a_3	0.03	0.0703	0.0328	9.33 %
a_4	4	3.7752	4.0284	0.71 %
b_1	0.01	0.0112	0.0109	9 %
b_2	2	1.9801	2.3822	19.17 %

Tabla 3.3: Resultados de la estimación de parámetros para el sistema (2.3) por el método Gauss-Newton para la dinámica de un paciente diabético.

Como se puede apreciar en la tabla 3.3, el parámetro que corresponde a la tasa de liberación de insulina por el páncreas es pequeño, cercano a cero, característica que se relaciona con el hecho de que los paciente que padecen diabetes tipo 2 producen una cantidad de insulina insuficiente para la metabolización de la cantidad de glucosa que ingresa a través de la ingesta de alimentos. La figura 3.5 muestra la gráfica de la dinámica de la glucosa de una persona que consume tres comidas con los respectivos horarios de ingesta que se consideraron en la simulación presentada en la sección 2.3.1.

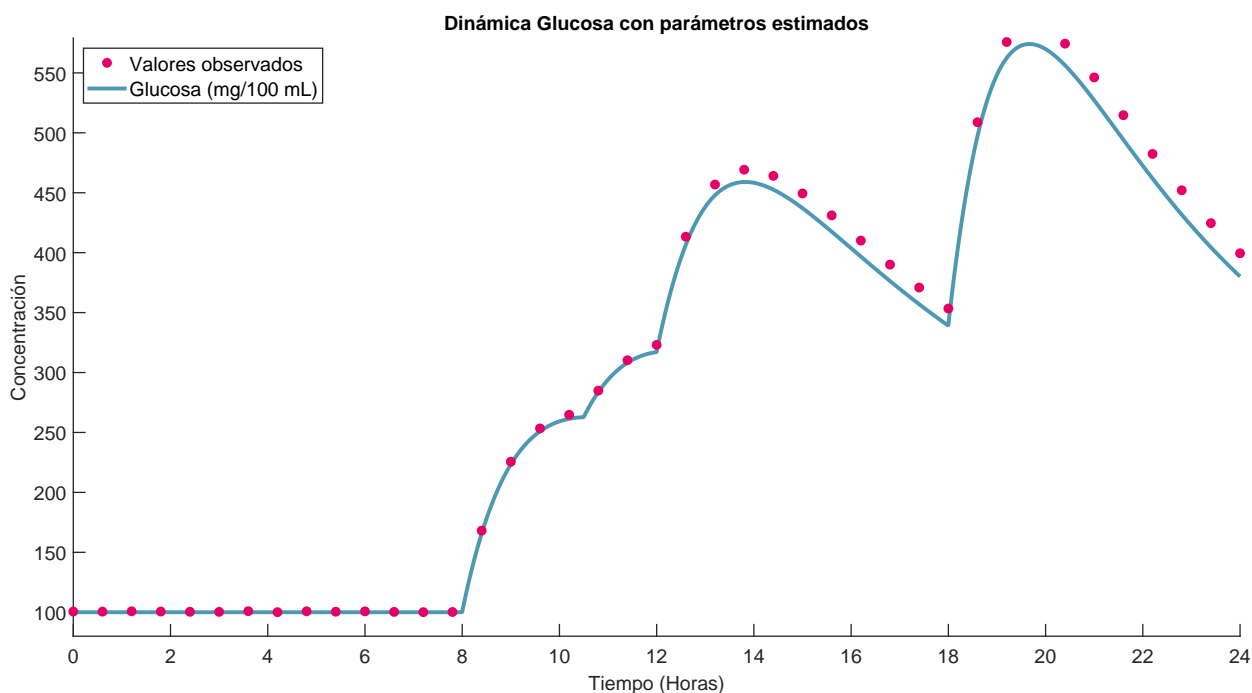


Figura 3.5: Resultados gráficos obtenidos de la estimación de parámetros para la dinámica de la glucosa de un paciente diabético.

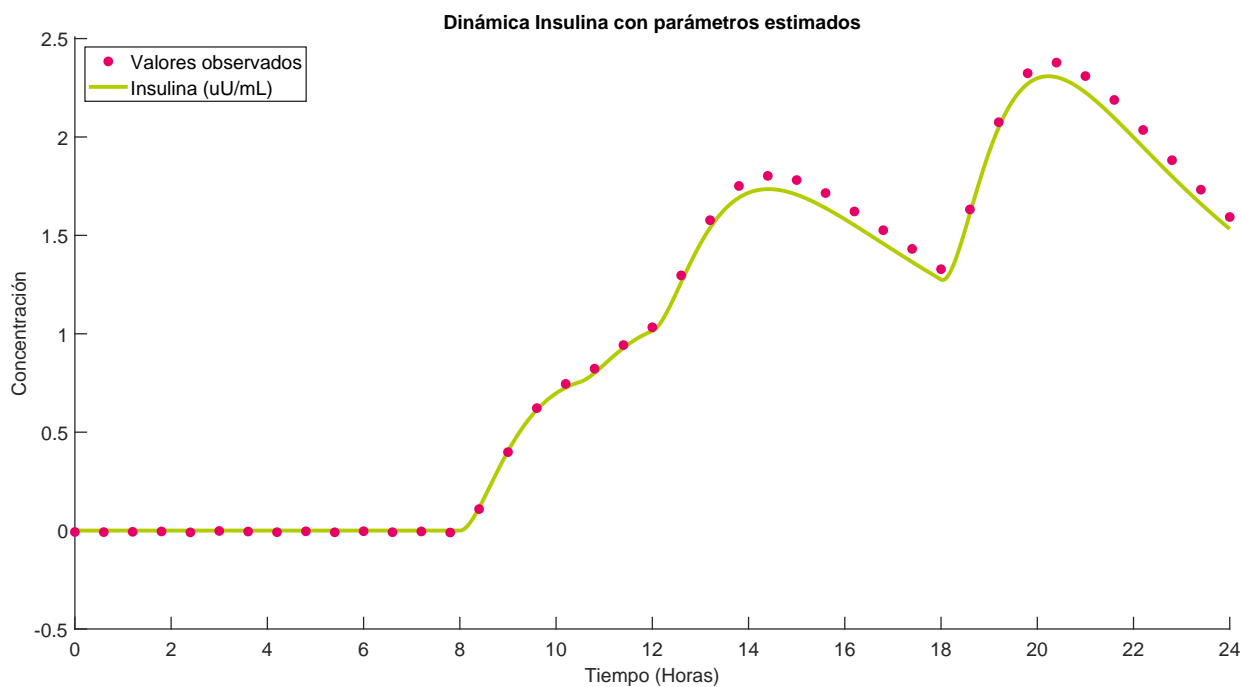


Figura 3.6: Resultados gráficos obtenidos de la estimación de parámetros para la dinámica de la insulina de un paciente diabético.

Nótese que a partir las 8:00 am, hora en la que se realiza el primer consumo de comida del día, el nivel de glucosa en sangre comienza a ascender, pero debido a la poca producción de insulina, la glucosa no alcanza a descender antes de que se consuma la siguiente comida a las 10:30 am, por lo que el nivel de glucosa sigue ascendiendo a lo largo del día, e incluso su concentración aumenta mucho más después de que la persona consume el almuerzo a las 12:30 pm. Después de las 2:00 pm, se observa el primer descendimiento del nivel de glucosa en sangre en lo que va corrido del día hasta el momento, sin embargo, nuevamente como la producción de insulina por parte del paciente sigue siendo muy poca y existe además la presencia de glucosa rezagada de otras comidas ingeridas anteriormente que no ha podido ser metabolizada, entonces es muy poca la disminución en la concentración de glucosa que logra obtenerse, siendo imposible alcanzar el nivel basal (que sería lo que debería ocurrir si la persona estuviese sana) antes de ingerir la última comida del día a las 6:30 pm, donde el nivel de glucosa vuelve a elevarse hasta alcanzar un pico alrededor de las 8:00 pm. Al final del día el paciente termina con una concentración de glucosa alrededor de 360 mg/dL, un valor muy por encima del valor basal en el cual debería estar.

Por otro lado, observe también que tanto las concentraciones de glucosa como las de insulina están durante casi todo el día por encima de 350 mg/dL y por debajo de 18 $\mu\text{U}/\text{mL}$ respectivamente. Finalmente, teniendo en cuenta las características descritas anteriormente y con el apoyo de la información presentada en las tablas 1.1 y 1.2, se concluye que la dinámica de la glucosa-insulina que en este experimento se considera, corresponde con certeza a la de un paciente que padece diabetes tipo 2.

Con respecto a la estimación de parámetros, nótese que los errores relativos de la estimación de cada parámetro resultan ser más grandes comparados con los obtenidos en el experimento anterior (ver tabla 3.3), aunque pueden considerarse como errores aceptables. Esta situación puede deberse a que los valores iniciales que componen el vector de parámetros (β_0) están más alejados de los valores “reales”, hecho que afecta el proceso de estimación, ya que como se había mencionado anteriormente, el método de Gauss-Newton requiere como datos de entrada un valor inicial para cada parámetro lo más cercano que sea posible al valor real para que se obtenga la convergencia. El tamaño de estos errores provoca, como se puede apreciar en las figuras 3.5 y 3.6, que las curvas que bosquejan los niveles de concentración de glucosa e insulina respectivamente durante el intervalo de tiempo $[0, 24]$ horas, una vez los parámetros han sido estimados, estén más alejadas de los puntos que representan los datos observados comparando esta misma característica con el experimento anterior. Aún así, las curvas obtenidas describen de manera bastante aceptable la tendencia que siguen los datos observados.

El tiempo de cómputo para este nuevo experimento fue de 4.54 segundos, y se logra el cumplimiento del criterio de parada del algoritmo en tan solo 22 iteraciones (ver la figura 3.7). Al comparar estos dos resultados con los correspondientes obtenidos en el experimento anterior, puede decirse que resulta ser más costosa computacionalmente la estimación de los parámetros para la dinámica glucosa-insulina que se describe por el modelo (2.3) de un paciente diabético que la de un paciente sano, aspecto que podría tenerse en cuenta en el

desarrollo de posibles dispositivos que a futuro incorporen métodos y rutinas que realicen esta tarea.

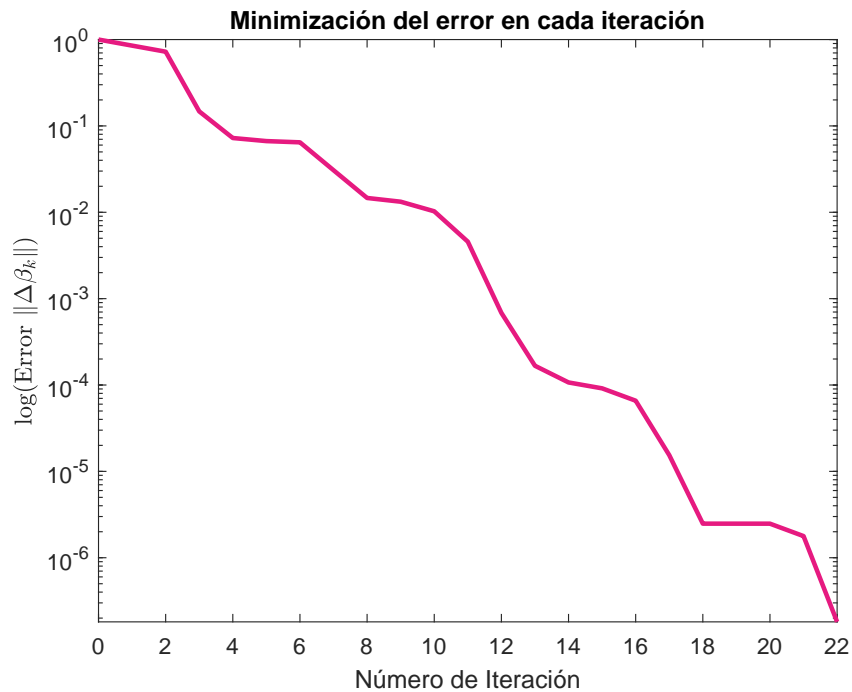


Figura 3.7: Gráfica del número de iteraciones vs. error de aproximación del método Gauss-Newton aplicado a la estimación de parámetros de la dinámica glucosa-insulina de un paciente diabético.

Finalmente, es importante destacar que los resultados encontrados muestran el potencial que pueden llegar a tener la combinación de métodos y técnicas presentados e introducidos en este capítulo, especialmente para el ajuste de los parámetros del modelo (2.3) a un conjunto de datos obtenidos de mediciones reales a los niveles de glucosa e insulina en sangre de cualquier paciente, bien sea sano o diabético.

4

Control óptimo

El control óptimo es un campo muy importante de la teoría de la optimización, con muchas aplicaciones en diferentes áreas. Desde 1960 [41], muchos artículos en economía, ingeniería, medicina entre otras, han reportado su uso para resolver los diferentes problemas que resultan ser del interés de cada una de estas ciencias. En este capítulo, se presenta la formulación del problema de control óptimo para el sistema de la dinámica glucosa-insulina definido en el capítulo 2, junto con algunas justificaciones que sustentan teóricamente el hallazgo de una posible solución para el problema que se intenta resolver en el desarrollo de este trabajo. Finalmente, se presentan los aspectos teóricos de los métodos numéricos a implementar computacionalmente con el objetivo de averiguar la efectividad y el potencial de la propuesta elaborada.

4.1. Formulación general del problema de control óptimo

Se puede definir el control de un objeto o proceso como la influencia que se ejerce sobre su comportamiento, de manera que pueda alcanzar un fin deseado. Se entiende por proceso algún movimiento o acción que tenga lugar a medida que transcurre el tiempo [43].

Considere un sistema cuyo estado en el tiempo está descrito por $x = (x_1, x_2, \dots, x_n)$ conocido como *variable o vector de estado*, con $x \in \mathbb{R}^n$. Sobre ese sistema se formula una estructura que permita determinar el curso posterior del proceso o su evolución a partir del conocimiento del estado previo del sistema. Dicha estructura se llama *dinámica* y por lo general está dada a partir de una ecuación o sistema de ecuaciones diferenciales de la forma

$$\frac{dx}{dt} = f(t, x(t), u(t)), \quad t \in [t_0, T], \quad (4.1)$$

donde $f(t, x(t), u(t)) = (f_1(t, x, u), \dots, f_n(t, x, u))$ y $u = u_1, \dots, u_n$. Un control es un conjunto de funciones del tiempo, que aplicadas sobre un sistema, modifican su dinámica con la finalidad de alcanzar un estado objetivo [43], y que para caso de las ecuaciones introducidas en (4.1) es $u(t)$ quien representa el control ejercido sobre el sistema. $u(t)$ recibe el nombre de función o vector de control y usualmente debe satisfacer varios tipos de restricciones, dependiendo de la naturaleza del problema. Para los propósitos de este trabajo, el control está sujeto a restricciones de la forma $u(t) \in U(t)$, para todo t , con $t \in [t_0, T]$ y $U \subset \mathbb{R}^m$.

La dinámica o ecuaciones de estado están sujetas también a unas condiciones iniciales y/o

finales

$$x(t_0) = x_0 \quad x(T) = x_T.$$

Es necesario contar además con un funcional de costo que mida que tan bueno es un control dado u , puesto que un proceso puede ser controlable en varias formas, y dentro del conjunto de controles que satisfacen un objetivo, es deseable la elección “del mejor” con respecto a un criterio de comportamiento. Por lo tanto, un problema de control óptimo pretende buscar y seleccionar un par (x^*, u^*) el cual minimiza cierto costo funcional sobre el conjunto de todas las parejas admisibles de funciones (x, u) que pertenecen al espacio de todas las funciones continuas de $[t_0, T]$ a \mathbb{R}^n , satisfaciendo además el sistema (4.1). La pareja $(x, u) : [t_0, T] \times [t_0, T] \rightarrow \mathbb{R}^n \times \mathbb{R}^m$ se dice admisible o factible si $u \in U$ y x es una función que satisface (4.1) y las condiciones impuestas en los puntos extremos.

La forma del funcional objetivo a optimizar es

$$J(x, u) = \int_{t_0}^T L(t, x(t), u(t)) dt + \ell(x(t_0), x(T)), \quad (4.2)$$

donde $L : [0, T] \times \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}$ y $\ell : \mathbb{R}^n \times \mathbb{R}^n \rightarrow \overline{\mathbb{R}}$ son funciones dadas. Esta última representa una función de costo inicial o final y se añade, por ejemplo, a la función objetivo de algunos problemas de optimización económica, en los cuales se debe considerar alguna función que represente una utilidad o los costos y pérdidas de una producción, que se asocia únicamente al valor inicial o final de la variable de estado [41]. Por esta razón, su aparición en la formulación del funcional objetivo, depende de la naturaleza del problema a resolver.

En resumen, un problema de control óptimo puede ser escrito de la siguiente forma general [7]:

Problema P

Minimizar el funcional J sobre el conjunto de todas las parejas factibles (x, u) que satisfacen el sistema (4.1), es decir:

$$\min_{x(t), u(t)} = \int_{t_0}^T L(t, x(t), u(t)) dt + \ell(x(t_0), x(T)) \quad (4.3)$$

$$\text{Sujeto a:} \quad \frac{dx}{dt} = f(t, x(t), u(t)) \quad (4.4)$$

$$x(t_0) = x_0; \quad x(T) = x_T \quad (4.5)$$

$$r(x(t), u(t)) = 0 \quad (4.6)$$

$$s(x(t), u(t)) \leq 0, \quad (4.7)$$

para algún $t \in [t_0, T]$.

Las expresiones (4.6) y (4.7) representan restricciones de igualdad y desigualdad respectivamente, que se imponen sobre las variables de control o de estado, por ejemplo, los límites superiores o inferiores entre los cuales estas variables pueden tomar valores, además de otras restricciones que se pueden incorporar, dependiendo también del contexto del problema de control óptimo a resolver. Si se tiene un problema de p restricciones de igualdad y q restricciones de desigualdad, entonces se definen las funciones vectoriales r y s como $r(x, u) := (r_1(x, u), r_2(x, u), \dots, r_p(x, u))^T$ y $s(x, u) := (s_1(x, u), s_2(x, u), \dots, s_q(x, u))^T$.

Un par (x^*, u^*) que resuelve el problema \mathbf{P} es llamado par óptimo. La función u^* es llamada control óptimo y x^* estado óptimo del problema \mathbf{P} .

Problema de control óptimo para el sistema glucosa-insulina

Es claro que la dinámica entre la glucosa y la insulina es diferente entre pacientes diabéticos y no diabéticos, por lo cual el problema de control óptimo para este sistema consiste en influenciar la dinámica de un paciente diabético, para que esta se comporte de manera similar a la de un paciente no diabético o sano.

Así pues, se estudiará el caso de un paciente enfermo de diabetes que no produce de manera natural la cantidad de insulina necesaria para aprovechar la glucosa adecuadamente, es decir, la sensibilidad del metabolismo a la presencia de glucosa es muy pequeña i.e., $b_1 \approx 0$, razón por la que el paciente requerirá del aporte externo de insulina en el tiempo, representado por la función $u(t)$. Lo anterior implica que la segunda ecuación del sistema (2.3) se transforme en:

$$\frac{dh}{dt} = b_1(g - g_0)^+ - b_2h + u(t).$$

Como se desea que este suministro de insulina externa sea lo más pequeño posible, de tal manera que la concentración de glucosa se mantenga muy cercana a un nivel de concentración deseado, que podría ser por ejemplo los niveles de glucosa de un paciente sano (denotado por g_d), el problema \mathbf{P} de control óptimo para este caso particular consiste en minimizar el funcional:

$$\min_{u(t)} J = \min_{u(t)} \int_0^T (g(t) - g_d)^2 + (u(t) - \bar{u})^2 dt. \quad (4.8)$$

Sujeto a:

$$\begin{cases} \frac{dg}{dt} = -a_1gh + a_2(g_0 - g)^+ - a_3(g - g_0)^+ + a_4p(t), \\ \frac{dh}{dt} = b_1(g - g_0)^+ - b_2h + u(t), \\ g(0) = g_0, \quad h(0) = h_0, \quad g(T) = g_0 \\ u_{\min} \leq u(t) \leq u_{\max}, \quad \underline{g} \leq g(t) \leq \bar{g}, \quad t \in [0, T] \end{cases} \quad (4.9)$$

donde \bar{u} es la concentración promedio de insulina en sangre de un paciente sano durante un tiempo de 24 horas, \underline{g} y \bar{g} son las concentraciones mínima y máxima de glucosa consideradas saludables en el organismo de una persona respectivamente (de acuerdo con la tabla 1.1), y $u_{\text{mín}}, u_{\text{máx}}$ son las restricciones del control, que en el contexto del problema, corresponden a los valores máximos y mínimos de insulina exógena que pueden ser suministrados a un paciente diabético.

4.2. Existencia de una solución para el problema de control óptimo

Existe una extensa literatura sobre la existencia de soluciones a problemas de optimización de la forma **P**. A continuación se presenta una versión adaptada a las condiciones del problema de control óptimo para el sistema glucosa-insulina presentado en la sección anterior, del teorema de existencia de control óptimo que se encuentra en [7, p. 36-38].

Teorema 3. *Si el problema de control óptimo (4.8) y (4.9) satisface las siguientes condiciones:*

- a) *La función $L : [0, T] \times \mathbb{R}^2 \times \mathbb{R} \rightarrow \mathbb{R}$ es medible en t , continua en (g, u) y $u \rightarrow L(t, g, u)$ es convexa para todo $(t, g) \in [0, T] \times [0, g_{\text{máx}}]$; $U(t) \equiv U \subseteq \mathbb{R}$ es cerrado y convexo. La función hamiltoniana $H(t, g, p) = \sup \{p \cdot v - L(t, g, v); v \in U\}$, satisface la condición de crecimiento:*

$$H(t, g, p) \leq \mu(t, p) + \|g\| (\sigma(t) + \gamma(t) \|g\|) \\ \forall t \in [0, T], g \in \mathbb{R}, p \in \mathbb{R}.$$

Donde $\sigma, \gamma \in L^1(0, T)$, $\mu(\cdot, p) \in L^1(0, T)$ para todo $p \in \mathbb{R}$ y $\sup \{\mu(t, p); \|p\| \leq \delta\} \in L^1(0, T)$ para cada $\delta > 0$.

- b) *$f(t, x, u) = f_0(t, x) + f_1(t, x)u$, con $x = (g, h)$, $\forall (t, x, u) \in [0, T] \times \mathbb{R}^2 \times \mathbb{R}$, donde f_0, f_1 son funciones continuas en x , medibles en t y además:*

$$f_0(t, x) \cdot x \leq (\alpha_0(t) \|x\| + \beta_0(t)) \|x\|, \quad t \in (0, T), x \in \mathbb{R}^2,$$

$$\|f_1(t, x)\| \leq \alpha_1(t), \quad t \in (0, T), x \in \mathbb{R}^2,$$

$$\|f_0(t, x)\| \leq \alpha_\delta(t), \quad t \in (0, T), \|x\| \leq \delta,$$

donde $\alpha_0, \beta_0 \in L^1(0, T)$, $\alpha_1 \in L^\infty(0, T)$ y $\alpha_\delta \in L^1(0, T)$ para todo $\delta > 0$,

entonces tiene al menos una solución (g^, h^*, u^*) en el espacio de todas las funciones continuas en $([0, T] \times \mathbb{R}^2) \times L^1(0, T; \mathbb{R})$.*

Demostración:

a) Sea $L(t, g, u) = (g(t) - g_d)^2 + (u(t) - \bar{u})^2$ y $U = [u_{\min}, u_{\max}]$. La función $L(t, \cdot, \cdot)$ es medible puesto que L es composición de funciones continuas y toda función continua es Lebesgue-medible.

Por otro lado, para comprobar que la función $u \rightarrow L(t, g, u)$ es convexa, esta debe cumplir que:

i) El dominio de $u \rightarrow L(t, g, u)$ es convexo.

Como el dominio de $u \rightarrow L(t, g, u)$ es el intervalo $[u_{\min}, u_{\max}]$, y se sabe que todo intervalo cerrado de números reales es un conjunto cerrado y convexo, entonces el dominio de la función es convexo, y se corrobora la primera condición.

ii) Para todo $u_1, u_2 \in \text{dom } L$, y $0 \leq \theta \leq 1$, se tiene que

$$L(t, g, \theta u_1 + (1 - \theta)u_2) \leq \theta L(t, g, u_1) + (1 - \theta)L(t, g, u_2).$$

Comprobación:

Para efectos de la simplificación de escritura, sea $\phi = g(t) - g_d$, por lo tanto

$$L(t, g, u) = (u - \bar{u})^2 + \phi^2 = u^2 - 2u\bar{u} + \bar{u}^2 + \phi^2.$$

Sea $u_1, u_2 \in [u_{\min}, u_{\max}]$ y $\theta \in [0, 1]$, luego

$$\begin{aligned} L(t, g, \theta u_1 + (1 - \theta)u_2) &= (\theta u_1 + (1 - \theta)u_2)^2 - 2(\theta u_1 + (1 - \theta)u_2)\bar{u} + \bar{u}^2 + \phi^2 \\ &= \theta^2 u_1^2 + 2\theta(1 - \theta)u_1 u_2 + (1 - \theta)^2 u_2^2 - 2(\theta u_1 + (1 - \theta)u_2)\bar{u} + \bar{u}^2 + \phi^2. \end{aligned}$$

Si $u_1 \neq u_2$, entonces $(u_1 - u_2)^2 > 0$, por lo tanto $u_1^2 - 2u_1 u_2 + u_2^2 > 0$, lo que implica $(u_1^2 + u_2^2) > 2u_1 u_2$, y así

$$2\theta(1 - \theta)u_1 u_2 \leq \theta(1 - \theta)(u_1^2 + u_2^2). \quad (4.10)$$

Usando la desigualdad (4.10) y algunas propiedades de los números reales, se tiene que

$$\begin{aligned} &\theta^2 u_1^2 + 2\theta(1 - \theta)u_1 u_2 + (1 - \theta)^2 u_2^2 - 2(\theta u_1 + (1 - \theta)u_2)\bar{u} + \bar{u}^2 + \phi^2 \\ &\leq \theta^2 u_1^2 + (1 - \theta)^2 u_2^2 - 2(\theta u_1 + (1 - \theta)u_2)\bar{u} + \bar{u}^2 + \phi^2 + (1 - \theta)\theta(u_1^2 + u_2^2) \\ &= u_2^2 - \theta u_2^2 - 2\theta u_1 \bar{u} - 2u_2 \bar{u} + 2\theta u_2 \bar{u} + \bar{u}^2 + \phi^2 + \theta u_1^2 + \theta \bar{u}^2 - \theta \bar{u}^2 + \theta \phi^2 - \theta \phi^2 \\ &= \theta(u_1^2 - 2u_1 \bar{u} + \bar{u}^2 + \phi^2) - \theta(u_2^2 - 2u_2 \bar{u} + \bar{u}^2 + \phi^2) + (u_2^2 - 2u_2 \bar{u} + \bar{u}^2 + \phi^2) \\ &= \theta((u_1^2 - 2u_1 \bar{u} + \bar{u}^2) + \phi^2) + (1 - \theta)((u_2^2 - 2u_2 \bar{u} + \bar{u}^2) + \phi^2) \\ &= \theta((u_1 - \bar{u})^2 + \phi^2) + (1 - \theta)((u_2 - \bar{u})^2 + \phi^2) \\ &= \theta((u_1 - \bar{u})^2 + (g(t) - g_d)^2) + (1 - \theta)((u_2 - \bar{u})^2 + (g(t) - g_d)^2) \\ &= \theta L(t, g, u_1) + (1 - \theta)L(t, g, u_2), \end{aligned}$$

luego la función $u \rightarrow L(t, g, u)$ es convexa.

La función hamiltoniana $H(t, g, p)$ para el caso del problema de control óptimo sobre el modelo de la dinámica glucosa-insulina es

$$H(t, g, p) = \sup \{ p \cdot v - (g(t) - g_d)^2 - (v(t) - \bar{u})^2; v \in U \} \quad (4.11)$$

$$g \in \mathbb{R}, p \in \mathbb{R}, t \in [0, T].$$

Como $v(t)$ es una función continua para todo $t \in [0, T]$, entonces existe $M > 0$ tal que

$$|v(t)| \leq M \iff -M \leq v(t) \leq M. \quad (4.12)$$

Por otro lado, si $(g(t) \neq g_d)$ y $(v(t) \neq \bar{u})$, entonces se tienen las siguientes desigualdades

$$2gg_d < g^2 + g_d^2 \quad (4.13)$$

$$2v\bar{u} < v^2 + \bar{u}^2. \quad (4.14)$$

De (4.12) se puede deducir que

$$pv \leq M |p|, \quad \text{para todo } p \in \mathbb{R} \quad (4.15)$$

Sumando (4.13), (4.14) y (4.15), se tiene

$$pv + 2gg_d + 2v\bar{u} \leq M |p| + g^2 + g_d^2 + v^2 + \bar{u}^2$$

$$pv + 2gg_d + 2v\bar{u} - g^2 - g_d^2 - v^2 - \bar{u}^2 \leq M |p|$$

$$pv - (g - g_d)^2 - (v - \bar{u})^2 \leq M |p|,$$

luego $\mu(t, p) = M |p|$, $\sigma(t) = 0$, $\gamma(t) = 0$. Como las funciones μ , σ y γ son continuas en $[0, T]$, entonces también son L^1 , y por lo tanto $\sup \{ \mu(t, p); \|p\| \leq \delta \} \in L^1(0, T)$ para cada $\delta > 0$.

b) Sean

$$f_0(t, \overbrace{(g, h)}^{:=x}) := \begin{bmatrix} -a_1gh + a_2[g_0 - g]^+ - a_3[g - g_0]^+ + a_4z(t - \tau) \\ b_1[g - g_0]^+ - b_2h \end{bmatrix} \quad (4.16)$$

$$f_1(t, g, h) := \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \quad (4.17)$$

tal que $f(t, x, u) = f_0(t, x(t)) + f_1(t, x(t)) u(t)$.

Nótese que $f_0(t, (g, h))$ y $f_1(t, (g, h))$ son funciones continuas en (g, h) (como se observó en la demostración del teorema 1, además, son medibles en t porque el sistema es autónomo, y así, al ser f_0 y f_1 continuas en g y h también lo son para t y por lo tanto son medibles.

Ahora, para encontrar las cotas necesarias en la prueba de las desigualdades, se debe tener presente que:

- $f_0(t, (g, h)) : \mathbb{R} \times \mathbb{R}^2 \rightarrow \mathbb{R}^2$.
- En \mathbb{R}^n , la norma euclídea y la norma del máximo son equivalentes, es decir:

$$\|x\|_\infty \leq \|x\|_2 \leq \sqrt{n} \|x\|_\infty$$

- **Desigualdad de Cauchy-Schwarz:** Si $x, y \in \mathbb{R}^n$ entonces $x^T y \leq \|x\|_2 \|y\|_2$.

De esta desigualdad, junto con la mostrada en el ítem anterior, se deduce que $x^T y \leq n \|x\|_\infty \|y\|_\infty$.

Por definición de la norma del máximo

$$\|x\|_\infty := \max\{\|g\|_\infty, \|h\|_\infty\}, \quad (4.18)$$

se puede deducir que

$$\|gh\|_\infty \leq \|g\|_\infty \|h\|_\infty \leq \|g\|_\infty \|x\|_\infty. \quad (4.19)$$

Luego,

$$\begin{aligned} f_0(t, x) \cdot x &\leq 2\|f_0(t, x)\|_\infty \|x\|_\infty && \text{(por desigualdad de Cauchy-Schwartz)} \\ &\leq 2\| -a_1gh + a_2(g_0 - g)^+ - a_3(g - g_0)^+ + a_4 z(t - \tau) \|_\infty \|x\|_\infty \\ &\leq 2(a_1\|gh\|_\infty + a_2\|(g_0 - g)^+\|_\infty && \text{(por desigualdad triangular)} \\ &\quad - a_3\|(g - g_0)^+\|_\infty + a_4\|z(t - \tau)\|_\infty) \|x\|_\infty \\ &\leq 2(a_1\|g\|_\infty \|x\|_\infty + a_2\|x\|_\infty - a_3\|x\|_\infty + a_4\|z(t - \tau)\|_\infty) \|x\|_\infty \\ &\quad \text{(por (4.18) y (4.19))} \\ &\leq \underbrace{2(a_1\|g\|_\infty \|x\|_\infty + a_2 - a_3)}_{\alpha_0(t)} \|x\|_\infty + \underbrace{2a_4\|z(t - \tau)\|_\infty}_{\beta_0} \|x\|_\infty \end{aligned}$$

Ahora, se probará que $\|f_1(t, x)\| \leq \alpha_1(t)$, $t \in (0, T)$, $x \in \mathbb{R}^2$. Como $f_1(t, x) = [0, 1]^T$, entonces, tomando la norma del máximo:

$$\|f_1(t, x)\|_\infty = \left\| \begin{bmatrix} 0 \\ 1 \end{bmatrix} \right\|_\infty = 1 = \alpha_1(t),$$

con lo cual se cumple la desigualdad.

Ahora, se probará que $\|f_0(t, x)\| \leq \alpha_\delta(t)$, $t \in (0, T)$, $\|x\| \leq \delta$. Por la desigualdad triangular y por las desigualdades (4.18) y (4.19), se tiene que:

$$\|f_0(t, x)\| \leq (a_1\|g\|_\infty + a_2 - a_3) \|x\|_\infty + a_4\|z(t - \tau)\|_\infty$$

y como $\|x\|_\infty \leq \delta$, entonces

$$\|f_0(t, x)\| \leq \underbrace{(a_1 \|g\|_\infty + a_2 - a_3) \delta + a_4 \|z(t - \tau)\|_\infty}_{\alpha_\delta(t)}.$$

Finalmente, las funciones α_0, β_0 y α_δ pertenecen a $L^1(0, T)$, puesto que son funciones constantes. La función α_1 pertenece a $L^\infty(0, T)$ ya que también es una función constante, y la función $z(t - \tau)$ es continua en $[0, T]$, por ser la función de alimento.

Como las condiciones **a)** y **b)** del teorema se satisfacen para el problema de control óptimo formulado en (4.8) y (4.9), entonces existe una solución para este problema. ■

4.3. Solución numérica del problema de control óptimo

Como ya se mencionó en el capítulo 2, el sistema de ecuaciones (2.3) escogido para la representación de la dinámica glucosa-insulina no posee una solución analítica, situación que conlleva a que tampoco sea posible el hallazgo de una solución analítica del problema de control óptimo formulado en (4.8). Por esta razón se hace necesario el uso de métodos numéricos que permitan obtener aproximaciones a lo que se podría considerar como una solución al problema.

Existen en la literatura diversas técnicas para la solución numérica de un problema de control óptimo. Estos métodos de solución puede clasificarse en tres tipos o familias de enfoques [12], [25], los cuales se describen brevemente a continuación:

- **Métodos directos:** consisten en la discretización de las variables de estado, de control y el funcional, de tal manera que el problema se reduce a un problema de *programación no lineal* (PNL), que se puede resolver con ayuda de algún algoritmo de optimización como el de *programación cuadrática secuencial* (SQP por sus siglas en inglés), o el método de punto interior. En general, el método se puede esbozar como “primero discretizar, luego optimizar” [17].
- **Métodos indirectos:** consisten en la formulación de un problema de valor en la frontera, obtenido de la aplicación del *principio del mínimo de Pontryagin*, que posteriormente requiere la discretización de los sistemas resultantes a resolver para la búsqueda de la solución numérica. Este principio fue desarrollado en la década de los años 50 por el matemático ruso Lev Pontriagyn, junto con un grupo de otros matemáticos [41], [33]. Este principio es uno de los más importantes resultados teóricos en el análisis de problemas de control óptimo.
- **Programación dinámica:** la programación dinámica es una técnica matemática también desarrollada durante los años 50, principalmente por el matemático estadounidense Richard Bellman. Aquí, la idea principal es la aplicación del *principio de*

optimalidad de Bellman, que consiste en descomponer el problema en etapas que pueden ser completados de una o varias formas (estados), y enlazando cada etapa a través de cálculos recursivos. Este método plantea el problema de optimización a partir de una ecuación diferencial parcial no lineal, llamada ecuación de programación dinámica o ecuación de Hamilton-Jacobi-Bellman. La solución de esta ecuación es la función de valor o de costo óptimo, de la cual se obtiene el costo mínimo para el control del sistema dinámico dado, donde para cada subproblema o etapas, se han seleccionado los estados cuyas trayectorias tienen costo mínimo [26].

La mayoría de software desarrollados y utilizados en la solución numérica de problemas de control óptimo están basados en esquemas propuestos a partir de los métodos directos [12]. Una de las razones por las que existe esta preferencia, tiene que ver con la disponibilidad de algoritmos de optimización bastante robustos y confiables con los cuales es posible resolver el problema PNL que resulta de aplicar esta estrategia, y por lo tanto es muy fácil el manejo de las diferentes restricciones de igualdad y desigualdad asociadas al problema de control óptimo, además de que no requiere el procesamiento de cálculos directos de las ecuaciones de *co-estado* o *adjuntas* que resultan de la aplicación del principio del mínimo de Pontriagyn al implementar el método indirecto.

Por otro lado, un gran número de artículos que abordan los métodos numéricos para resolver problemas de control óptimo, reportan que el método indirecto y algunos esquemas de resolución originados a partir de esta estrategia, no son computacionalmente los más eficientes, debido precisamente a la necesidad de manipular y resolver la cantidad de ecuaciones de co-estado o adjuntas, además de las asociadas a las restricciones, lo que amplía la complejidad computacional considerablemente [12, p. 154]. Sin embargo, en la actualidad es posible encontrar algunos trabajos en los que se realiza la implementación computacional de esquemas basados en métodos indirectos, en los cuales se reportan conclusiones muy favorables en cuanto a tiempo de cómputo, por lo que podría ser también una alternativa muy competitiva a los métodos que se vienen implementando tradicionalmente [12, p. 164].

Por las razones expuestas anteriormente, la solución numérica por métodos directos fue la escogida como estrategia para resolver el problema de control óptimo formulado en (4.8). Aún así valdría la pena, en un trabajo futuro, intentar resolver el problema de control óptimo utilizando algún esquema de resolución basado en otro de los métodos presentados anteriormente, con el fin de reportar posibles ventajas que se puedan encontrar, con respecto al implementado en este trabajo. En las siguientes secciones del presente capítulo, se desarrollarán con más detalle las particularidades de cada etapa del método directo usado.

4.3.1. Método directo: formulación del problema PNL

Los métodos directos para la resolución numérica de problemas de control óptimo, requieren de la ejecución de un conjunto de tres etapas cuya finalidad es la conversión del problema de control óptimo a un problema de optimización, o mejor, de programación no lineal (PNL).

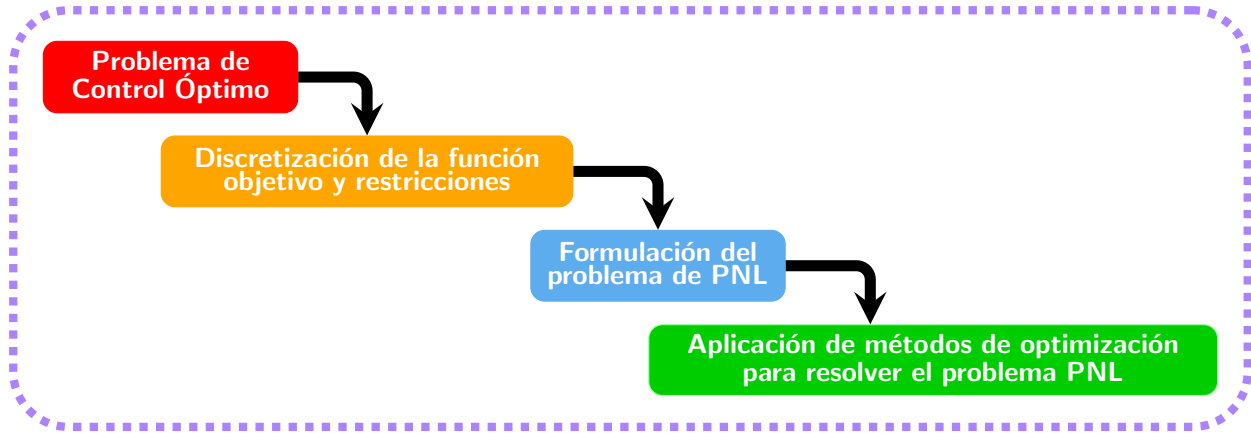


Figura 4.1: Esquemización de los métodos directos para resolver problemas de control óptimo.

Estas etapas son: *discretización*, *formulación del problema PNL* y *aplicación de métodos de optimización para resolver el problema PNL*.

Etapa 1: Discretización del problema de control óptimo

Para plantear la formulación de un problema de control óptimo como un problema de programación no lineal, se comienza por la discretización de la variable de control. Así pues, el intervalo $[0, T]$ se divide en N subintervalos $[t_0, t_1, \dots, t_N]$ de tamaño $\Delta t = T/N$, por definición $t_k = k \Delta t$, $k = 0, 1, \dots, N$. Sobre esta grilla de puntos se representa el control $u(t)$ como un spline o función polinomial a trozos que se define sobre el intervalo $[t_k, t_{k+1}]$, por simplicidad y por la cantidad de puntos disponibles sobre el intervalo, el control se escoge como una función de interpolación lineal entre $u(t_k)$ y $u(t_{k+1})$ para $t_k \leq t \leq t_{k+1}$. Así pues, se considera

$$u(t) \approx u_{k+1} \left(\frac{t - t_k}{t_{k+1} - t_k} \right) - u_k \left(\frac{t - t_{k+1}}{t_{k+1} - t_k} \right)$$

Ahora, como $\Delta t = t_{k+1} - t_k$, entonces:

$$\begin{aligned} u(t) &\approx u_{k+1} \left(\frac{t - t_k}{\Delta t} \right) - u_k \left(\frac{t - (t_k + \Delta t)}{\Delta t} \right) \\ &= \frac{u_{k+1}(t - t_k) - u_k(t - t_k - \Delta t)}{\Delta t} \\ &= \frac{u_{k+1}t - u_{k+1}t_k - u_kt + u_k\Delta t}{\Delta t} \\ &= u_k + \frac{(u_{k+1} - u_k)t}{\Delta t} - \frac{(u_{k+1} - u_k)t_k}{\Delta t} \end{aligned}$$

$$\boxed{= u_k + \frac{u_{k+1} - u_k}{\Delta t} (t - t_k).} \quad (4.20)$$

Usando una rutina de simulación numérica se resuelve el sistema de ecuaciones (4.4) asociado a la dinámica de la variable de estado, que en este caso, se realiza a través del esquema iterativo del método de Runge-Kutta de cuarto orden presentado en el anexo A. Sin embargo, este esquema iterativo debe modificarse con el fin de hacer la inclusión de la variable de control, para lo cual se hace uso de la expresión (4.20) con el fin de estimar el valor que toma la función de control u en $(t_k + \frac{\Delta t}{2})$, quedando de forma general como

$$\begin{aligned} K_1 &= f(t_k, x_k, u_k) \\ K_2 &= f\left(t_k + \frac{\Delta t}{2}, x_k + \frac{\Delta t}{2} K_1, \frac{1}{2}(u_k + u_{k+1})\right) \\ K_3 &= f\left(t_k + \frac{\Delta t}{2}, x_k + \frac{\Delta t}{2} K_2, \frac{1}{2}(u_k + u_{k+1})\right) \\ K_4 &= f(t_k + \Delta t, x_k + \Delta t K_3, u_{k+1}) \\ x_{k+1} &= F_k = x_k + \frac{\Delta t}{6} (K_1 + 2K_2 + 2K_3 + K_4). \end{aligned}$$

Este esquema se denomina *método directo secuencial* o *método de disparo simple*, y considera únicamente a la variable de control $u(t)$ como la variable a optimizar, es decir, el método minimiza con respecto a la variable de control y resuelve (4.4) sobre el intervalo $[0, T]$ en cada nuevo punto generado por el algoritmo de optimización. Lo anterior sugiere entonces que las ecuaciones diferenciales que describen la dinámica del sistema se usan para eliminar las variables de estado del nuevo problema de optimización a formular [8].

Finalmente, definiendo la variable $z = (x_N, u_0, u_1, \dots, u_{N-1})$, y usando una regla de cuadratura rectangular que toman el extremo izquierdo de cada subintervalo con el propósito de resolver la integral definida en (4.8), el **problema P** se puede reescribir como

$$\min_z W(z) = \min_{\substack{x_N \\ u_0, u_1, \dots, u_{N-1}}} \Delta t \sum_{k=0}^{N-1} L(x_k, u_k) \quad (4.21)$$

Sujeto a:

$$\begin{cases} x(k+1) = F(x_k, u_k), & x_0 = x(t_0), \\ x_{\min} \leq x_k \leq x_{\max}, \\ u_{\min} \leq u_k \leq u_{\max}, \\ x_N = x(T). \end{cases} \quad (4.22)$$

Estas restricciones pueden reescribirse de forma compacta, igualadas, o mayores iguales a cero así

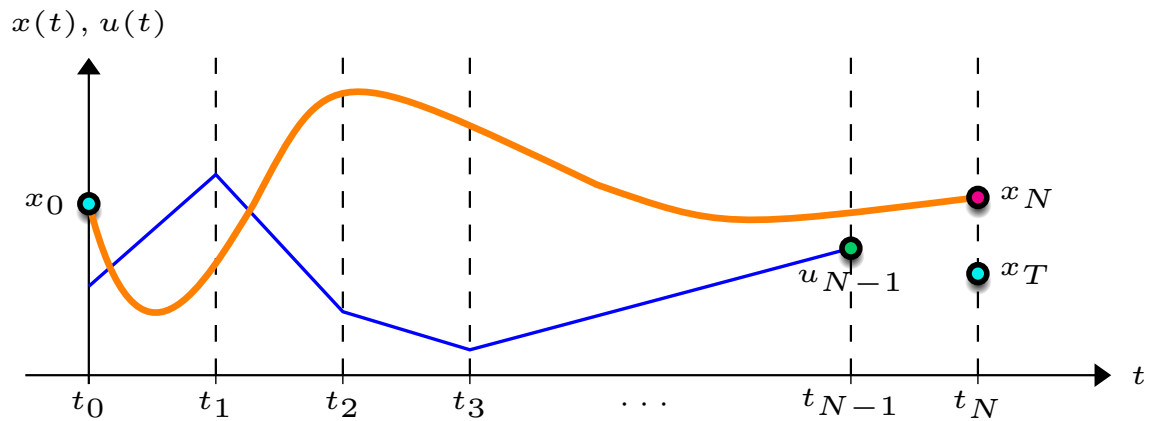


Figura 4.2: Ilustración del método directo secuencial, adaptada de [17]. En la gráfica, x_0 y x_T (puntos azules claro) representan las condiciones iniciales y finales de la variable de estado, las cuales deben satisfacerse. La línea continua azul aproxima la función de control $u(t)$ en cada uno de los subintervalos, y la curva naranja describe la dinámica del sistema asociado a las variables de estado y control.

$$r(x_k, u_k) = [x_0 - x(t_0), x_N - x_T]^T = 0, \quad (4.23)$$

$$s(x_k, u_k) = \begin{bmatrix} x_{\text{mín}} - x_k \\ x_k - x_{\text{máx}} \\ u_{\text{mín}} - u_k \\ u_k - u_{\text{máx}} \end{bmatrix} \leq \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad k = 0, 1, \dots, N-1. \quad (4.24)$$

El problema definido en (4.21), junto con sus restricciones (4.23) y (4.24) ya posee la estructura de un problema de programación no lineal, del cual se comentarán algunas particularidades en las siguientes secciones del presente capítulo.

Algoritmo 4: MÉTODO DIRECTO SECUENCIAL O DISPARO SIMPLE

Dados: x_0, u_0, \dots, u_{N-1} ;

$W = 0$;

$r = []$; $s = []$;

Para $k := 0$ **hasta** $N - 1$ **hacer**

$x_{k+1} := \int f(x_k, u_k) dt$;
 $l_k := L(x_k, u_k)$;
 $W := W + l_k$;
 $s_k := ([x_k - x_{\text{mín}}; x_{\text{máx}} - x_k])$;
 $s_k := ([u_k - u_{\text{mín}}; u_{\text{máx}} - u_k])$;

Fin

$r := [x_0 - x(t_0); x_N - x_T]$;

Resolver PNL (F, r, s);

Por último, el problema de control óptimo formulado en (4.8) y (4.9) para la dinámica del sistema glucosa-insulina, se reemplaza por un problema de programación no lineal, el cual se formula como sigue a continuación

$$\min_{\substack{g_N \\ u_0, u_1, \dots, u_{N-1}}} \Delta t \sum_{k=0}^{N-1} (g_k - g_d)^2 + (u_k - \bar{u})^2 \quad (4.25)$$

Sujeto a:

$$\begin{aligned} x_{k+1} &= (g_{k+1}, h_{k+1}) = (F_1(t_k, x_k, u_k), F_2(t_k, x_k, u_k)) \\ x_0 &= (g_0, h_0) \\ r(x_T) &= [g_N - g_0] = 0 \\ s(x_k, u_k) &= \begin{bmatrix} g_k - g_{\text{mín}} \\ g_{\text{máx}} - g_k \\ u_k - u_{\text{mín}} \\ u_{\text{máx}} - u_k \end{bmatrix} \leq \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad k = 0, 1, \dots, N-1, \end{aligned} \quad (4.26)$$

donde $x_k = (g_k, h_k)$, y F_1, F_2 representan las trayectorias de la variable de estado que resultan de la solución por integración numérica del sistema de ecuaciones diferenciales definido en (2.3), usando el esquema de Runge-Kutta cuarto orden.

Etapa 2: Formulación del problema de PNL

Un problema de programación no lineal es un problema de optimización, definido a partir de una función a optimizar, sujeta a un conjunto de restricciones de igualdad y desigualdad, en el que algunas de estas restricciones, o incluso la función objetivo, son no lineales. De manera general

Definición: *Un problema de programación no lineal consiste en*

$$\min_z W(z) \quad (4.27)$$

sujeto a:

$$r_i(z) = 0, \quad i = 1, \dots, p \quad (4.28)$$

$$s_i(z) \leq 0, \quad i = 1, \dots, q, \quad (4.29)$$

donde z es la variable de optimización, $W \in C^2$ y $W : \mathbb{R}^n \rightarrow \mathbb{R}$ es la función objetivo. Para todo $i \in \{1, \dots, p\}$, $r_i \in C^2$ y $r_i : \mathbb{R}^2 \rightarrow \mathbb{R}$ son las funciones correspondientes a las restricciones de igualdad, y para todo $j \in \{1, \dots, q\}$, $s_j \in C^2$, $s_j : \mathbb{R}^n \rightarrow \mathbb{R}$ son las funciones correspondientes a las restricciones de desigualdad.

Por comodidad en la escritura, se define $r(z) := (r_1(z), \dots, r_p(z))^T$, $s(z) = (s_1(z), \dots, s_q(z))^T$, $\nabla g(z) := (\nabla r_1(z), \dots, \nabla r_p(z))^T$ y $\nabla s(z) := (\nabla s_1(z), \dots, \nabla s_q(z))^T$.

El problema definido anteriormente, lleva al planteamiento del siguiente interrogante [39]:

¿qué condiciones deben satisfacer los puntos elegibles a soluciones óptimas para un problema de optimización, en particular uno de programación no lineal? La respuesta a esta pregunta implica pensar en las condiciones de optimalidad conocidas como las condiciones *Karush-Kuhn-Tucker* (KKT). La aplicación de estas condiciones permitirá encontrar soluciones óptimas a un problema de programación no lineal [39], y servirán además como punto de partida para el desarrollo de los métodos numéricos que se pueden implementar en el hallazgo de las soluciones a este tipo de problemas.

Teorema 4 (Condiciones de optimalidad de Karush-Kuhn-Tucker). *Si z^* es una solución local al problema (4.27), entonces existen los multiplicadores de Lagrange $\lambda \in \mathbb{R}^p$ y $\mu \in \mathbb{R}^q$ tal que:*

$$\nabla_z \mathcal{L}(z^*, \lambda, \mu) = \nabla W(z^*) + \nabla r(z^*) \lambda + \nabla s(z^*) \mu = 0 \quad (4.30)$$

$$r_i(z^*) = 0, \quad i = 1, \dots, p \quad (4.31)$$

$$s_j(z^*) \leq 0, \quad j = 1, \dots, q \quad (4.32)$$

$$s_j(z^*) \mu_j = 0, \quad j = 1, \dots, q \quad (4.33)$$

donde \mathcal{L} se denomina la función lagrangiana, y se define como

$$\mathcal{L}(z, \lambda, \mu) = W(z) + \lambda^T r(z) + \mu^T s(z). \quad (4.34)$$

Demostración:

La demostración puede consultarse en [13] y en [37]. ■

Etapa 3: Aplicación de métodos de optimización para resolver el problema PNL

Una vez planteado el problema de programación no lineal, un camino para poder resolverlo es emplear el método de *programación cuadrática secuencial*, el cual se desarrolla a partir de las condiciones de optimalidad del teorema 4. El uso de este método fue reportado por primera vez, de acuerdo con [29], en el artículo “*Algorithms for nonlinear constraints that use Lagrangian functions*” del año 1978. A continuación se presenta en detalle la deducción del método.

Método de Programación Cuadrática Secuencial (SQP)

El método de programación cuadrática secuencial (*SQP por sus siglas en inglés*) es una clase de método de optimización que consiste en la descomposición del problema original (4.27) en subproblemas de programación cuadrática (*QP*), que por lo general son más fáciles de resolver. La estructura básica de un *SQP* implica la ejecución de una iteración principal y otra secundaria [8]. La iteración principal genera la secuencia de una tripla (z_k, λ_k, μ_k) , a partir de unas estimaciones iniciales (z_0, λ_0, μ_0) , que se espera que converjan a los valores óptimos (z^*, λ^*, μ^*) . En la iteración secundaria se genera otra tripla de valores, que permiten actualizar la tripla de la primera iteración, hasta garantizar un criterio de parada. Las

actualizaciones que se generan durante las iteraciones secundarias, resultan de resolver un problema de programación cuadrática (QP), formulado como sigue:

En la iteración k , se aproxima la función \mathcal{L} de la expresión (4.34), usando una expansión en serie de Taylor de segundo grado alrededor del punto (z_k, λ_k, μ_k) , por lo tanto

$$\mathcal{L}(z_k + \Delta z, \lambda_k + \Delta \lambda, \mu_k + \Delta \mu) \approx \mathcal{L}(z_k, \lambda_k, \mu_k) + \nabla_{(z,\lambda,\mu)} \mathcal{L}^T \begin{pmatrix} \Delta z \\ \Delta \lambda \\ \Delta \mu \end{pmatrix} + \frac{1}{2} \begin{pmatrix} \Delta z \\ \Delta \lambda \\ \Delta \mu \end{pmatrix} \nabla_{(z,\lambda,\mu)}^2 \mathcal{L} \begin{pmatrix} \Delta z \\ \Delta \lambda \\ \Delta \mu \end{pmatrix}$$

Por simplicidad en escritura, $\mathcal{L}_k(\Delta z, \Delta \lambda, \Delta \mu) = \mathcal{L}(z_k + \Delta z, \lambda_k + \Delta \lambda, \mu_k + \Delta \mu)$, y \mathcal{L} siempre hará referencia a $\mathcal{L}(z_k, \lambda_k, \mu_k)$.

$$\mathcal{L}_k(\Delta z, \Delta \lambda, \Delta \mu) = \frac{1}{2} \begin{pmatrix} \Delta z \\ \Delta \lambda \\ \Delta \mu \end{pmatrix}^T \begin{pmatrix} \nabla_{zz}^2 \mathcal{L} & \nabla_{\lambda z}^2 \mathcal{L} & \nabla_{\mu z}^2 \mathcal{L} \\ \nabla_{z\lambda}^2 \mathcal{L} & 0 & 0 \\ \nabla_{z\mu}^2 \mathcal{L} & 0 & 0 \end{pmatrix} \begin{pmatrix} \Delta z \\ \Delta \lambda \\ \Delta \mu \end{pmatrix} + \begin{pmatrix} \nabla_z \mathcal{L} \\ \nabla_\lambda \mathcal{L} \\ \nabla_\mu \mathcal{L} \end{pmatrix}^T \begin{pmatrix} \Delta z \\ \Delta \lambda \\ \Delta \mu \end{pmatrix} + \mathcal{L}.$$

Calculando $\nabla_\lambda \mathcal{L}$ y $\nabla_\mu \mathcal{L}$ de la expresión (4.34)

$$\begin{aligned} \mathcal{L}_k(\Delta z, \Delta \lambda, \Delta \mu) &= \frac{1}{2} \begin{pmatrix} \nabla_{zz}^2 \mathcal{L} \Delta z + \nabla_{\lambda z}^2 \mathcal{L} \Delta \lambda + \nabla_{\mu z}^2 \mathcal{L} \Delta \mu \\ \nabla_{z\lambda}^2 \mathcal{L} \Delta z \\ \nabla_{z\mu}^2 \mathcal{L} \Delta z \end{pmatrix} + \nabla_z \mathcal{L}^T \Delta z \\ &\quad + r(z_k)^T \Delta \lambda + s(z_k)^T \Delta \mu + \mathcal{L} \\ &= \frac{1}{2} (\Delta z^T \nabla_{zz}^2 \mathcal{L} \Delta z + \Delta z^T \nabla_{\lambda z}^2 \mathcal{L} \Delta \lambda + \Delta z^T \nabla_{\mu z}^2 \mathcal{L} \Delta \mu + \Delta \lambda^T \nabla_{z\lambda}^2 \mathcal{L} \Delta z \\ &\quad + \Delta \mu^T \nabla_{z\mu}^2 \mathcal{L} \Delta z) + \nabla_z \mathcal{L}^T \Delta z + r(z_k)^T \Delta \lambda + s(z_k)^T \Delta \mu + \mathcal{L} \\ &= \frac{1}{2} \Delta z^T \nabla_{zz}^2 \mathcal{L} \Delta z + \nabla_z \mathcal{L}^T \Delta z + \mathcal{L} + \Delta \lambda^T \nabla_{z\lambda}^2 \mathcal{L} \Delta z \\ &\quad + \Delta \mu^T \nabla_{z\mu}^2 \mathcal{L} \Delta z + \Delta \lambda^T r(z_k) + \Delta \mu^T s(z_k). \end{aligned} \tag{4.35}$$

La expresión (4.35) tiene la forma de una función lagrangiana con multiplicadores $\Delta \lambda$ y $\Delta \mu$. A partir de aquí se puede formular el siguiente problema de optimización

$$\min_{\Delta z} \frac{1}{2} \Delta z^T \nabla_{zz}^2 \mathcal{L} \Delta z + \nabla_z \mathcal{L}^T \Delta z + \mathcal{L} \tag{4.36}$$

sujeto a:

$$\nabla_{z\lambda}^2 \mathcal{L} \Delta z + r(z_k) = 0 \tag{4.37}$$

$$\nabla_{z\mu}^2 \mathcal{L} \Delta z + s(k) \leq 0. \tag{4.38}$$

Por (4.31), (4.32) y (4.34):

$$\nabla_{z\lambda}^2 \mathcal{L}(z, \lambda, \mu) \Delta z = \left(\nabla r_i(z)^T \Delta z \right)_{i=1}^p; \quad \nabla_{z\mu}^2 \mathcal{L}(z, \lambda, \mu) \Delta z = \left(\nabla s_j(z)^T \Delta z \right)_{j=1}^q,$$

las restricciones del problema de programación cuadrática (4.36), pueden reescribirse como

$$\begin{aligned} r_i(z_k) + \nabla r_i(z_k)^T \Delta z &= 0, \quad i = 1, \dots, p \\ s_j(z_k) + \nabla s_j(z_k)^T \Delta z &\leq 0, \quad j = 1, \dots, q \end{aligned}$$

y finalmente por simplificación en la notación, el problema (4.36) puede plantearse de la siguiente manera

$$\min_{\Delta z} \frac{1}{2} \Delta z^T A \Delta z + b^T \Delta z + c \quad (4.39)$$

sujeto a:

$$\begin{cases} R \Delta z + r = 0 \\ S \Delta z + s \leq 0, \end{cases} \quad (4.40)$$

donde

$$R = \left(\nabla r_i(z)^T \right)_{i=1}^p; \quad S = \left(\nabla s_j(z)^T \right)_{j=1}^q; \quad r = r(z_k); \quad s = s(z_k).$$

Este problema ya posee la estructura de un problema de programación cuadrática QP , con restricciones de igualdad y desigualdad. Los problemas QP se resuelven aplicando las condiciones de optimalidad KKT del teorema 4, del cual se obtiene el siguiente sistema de ecuaciones e inecuaciones [39].

La función lagrangiana del problema (4.39) será

$$\mathcal{L}_k(\Delta z, \Delta \lambda, \Delta \mu) = \frac{1}{2} \Delta z^T A \Delta z + b^T \Delta z + c + \underbrace{\Delta \lambda^T (R \Delta z + r)}_{(R^T \Delta \lambda)^T \Delta z + \Delta \lambda^T r} + \underbrace{\Delta \mu^T (S \Delta z + s)}_{(S^T \mu)^T \Delta z + \Delta \mu^T s},$$

y por lo tanto, de esta función se obtiene el sistema KKT

$$\begin{cases} A \Delta z + b + R^T \Delta \lambda + S^T \Delta \mu = 0 \\ R \Delta z + r = 0 \\ S \Delta z + s \leq 0 \\ \Delta \mu^T (S \Delta z + s) = 0. \end{cases} \quad (4.41)$$

De la solución sistema (4.41), se obtienen los valores de Δz , $\Delta \lambda$ y $\Delta \mu$ los cuales son utilizados para la actualización de los valores de z_{k+1} , λ_{k+1} y μ_{k+1} calculados como parte de la iteración principal del método SQP . Por lo general, el sistema KKT asociado al problema

QP se resuelve por medio del método del *conjunto activo*, el cual ya viene implementado en los diferentes paquetes de software especializado en resolución de problemas de optimización. Sin embargo, puede consultarse los capítulos 16 al 18 de [37], para revisar con detalle los aspectos teóricos referentes al método del *conjunto activo*.

Por otro lado, en la iteración principal del método SQP , se calcula los valores actualizados de la variable a optimizar, junto con los respectivos multiplicadores, de la siguiente manera:

$$(z_{k+1}, \lambda_{k+1}, \mu_{k+1}) = (z_k, \lambda_k, \mu_k) + \rho_k(\Delta z, \Delta \lambda, \Delta \mu), \quad (4.42)$$

donde el término $\rho_k > 0$ se conoce como el parámetro del *tamaño de paso*, que en este caso cumple el mismo rol que en el caso de los métodos de búsqueda lineal. Estos últimos pueden no converger cuando se utiliza un *tamaño de paso* ρ fijo para todas las iteraciones, y por esta razón, es conveniente seleccionar el *tamaño de paso* de forma que se garantice la convergencia. Existen varios criterios para seleccionar este parámetro ρ , entre ellos se pueden mencionar la *condición de Armijo*, las *condiciones débiles y fuertes de Wolfe*, la *condición de Goldstein*, entre muchos otros [37], y cada paquete de software desarrollado para la solución de este tipo de problemas, selecciona alguno de estos, o diseña sus propios algoritmos donde construye combinaciones con algunos de los criterios.

Las evaluaciones de la función lagrangiana y de la matriz A , que resulta ser la matriz Hessiana de la función lagrangiana, que como ya se mostró, son necesarias para la implementación de los algoritmos presentados anteriormente, pueden obtenerse también como aproximaciones a partir de diferentes métodos como el *BFGS*, *Gauss-Newton*, entre otros, dependiendo nuevamente del solucionador de problemas de optimización que se seleccione. Para el caso de este trabajo, se utilizará la función *fmincon* de la herramienta *Optimization Toolbox* de MATLAB®. Para más detalles sobre la función *fmincon*, y los métodos utilizados en su desarrollo, puede consultarse [42].

Algoritmo 5: MÉTODO PROGRAMACIÓN CUADRÁTICA SECUENCIAL (SQP)

Dados: $(z_0, \lambda_0, \mu_0) \in \mathbb{R}^n \times \mathbb{R}^p \times \mathbb{R}^q$, W , r , s , $tol = 10^{-6}$;

Repetir $k = 0, 1, \dots$

Aproximar el problema original como un problema cuadrático, usando una expansión en serie de Taylor para el lagrangiano, alrededor de (z_k, λ_k, μ_k) ;

Resolver el problema QP para obtener $(\Delta z, \Delta \lambda, \Delta \mu)$;

Escoger un tamaño de paso ρ_k apropiado;

Actualizar $(z_{k+1}, \lambda_{k+1}, \mu_{k+1}) = (z_k, \lambda_k, \mu_k) + \rho_k(\Delta z, \Delta \lambda, \Delta \mu)$;

Actualizar la matriz Hessiana A , usando el esquema *BFGS*;

$k = k + 1$;

Hasta que $\|z_{k+1} - z_k\| < tol$;

Para finalizar, se presenta a continuación un diagrama de flujo que resume toda la propuesta de solución numérica para el problema de control óptimo abordado en este trabajo, el cual se pretende implementar computacionalmente, con el fin de realizar diferentes simulaciones y experimentos numéricos que permitan esbozar algunas soluciones al problema del control de suministro de insulina en pacientes diabéticos.

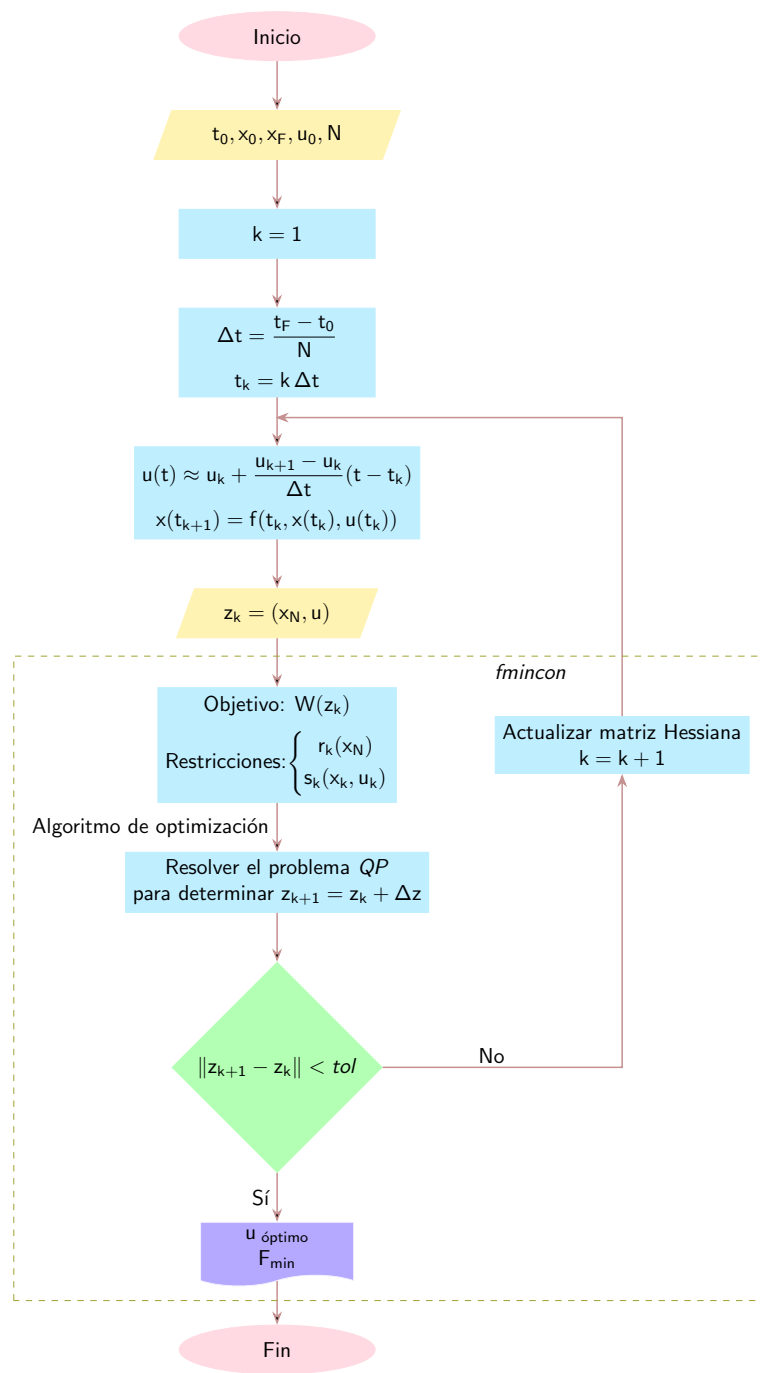


Figura 4.3: Diagrama de flujo propuesto para la solución del problema de control óptimo, formulado en (4.8).

5

Resultados numéricos

En este capítulo se presentan los resultados de cuatro posibles escenarios de alimentación durante un día de un paciente diabético de tipo 1 y tipo 2. Se emplea un conjunto de datos que representa los niveles reales de glucosa e insulina en sangre para diferentes instantes de tiempo t en un día ($t \in [0, 24]$) para inicialmente ajustar los parámetros del modelo (2.3) y posteriormente se busca la cantidad mínima de insulina que debe ser suministrada para que las concentraciones de glucosa se mantengan en los niveles considerados como saludables, es decir, se resuelve el problema de control óptimo. Suponiendo que tanto el modelo como el método de control óptimo podrían ser utilizados para el desarrollo de bombas de insulina totalmente automatizadas en el futuro, es necesario presentar algunas particularidades sobre el funcionamiento de estos dispositivos para el tratamiento de la diabetes.

5.1. Tratamiento de la diabetes y las bombas de insulina

Como se mencionó en el capítulo 1, todas las personas que padecen diabetes tipo 1 y gran parte de pacientes con diabetes tipo 2 requieren de tratamiento farmacológico para el manejo de la enfermedad, debido a las graves complicaciones de salud que se pueden desencadenar si no se realiza un manejo adecuado de la misma. Este tratamiento consiste en la aplicación de inyecciones subcutáneas de insulina o la infusión continua de la hormona a través de una bomba.

5.1.1. Tipos de insulina

Los diferentes tipos de insulina están caracterizados por tres factores farmacológicos: instante de tiempo de actuación, instante de máxima actuación y duración del efecto hipoglucimante [34]. Las combinaciones de estos factores producen tres tipos de insulina: de acción rápida, de acción corta y de acción retardada.

- **Insulina de acción rápida:**

Comienza a surtir efecto 10 o 15 minutos después de haber sido suministrada, tiene un efecto máximo al cabo de una hora y es eficaz durante un intervalo de dos a cuatro horas.

- **Insulina de acción corta:**

Comienza a actuar 30 minutos después de inyectada, que es el tiempo aproximado

que tarda en llegar al torrente sanguíneo, tiene su efecto máximo de dos a tres horas después de su aplicación y es eficaz de tres a seis horas.

■ **Insulina de acción retardada:**

Este tipo de insulinas poseen tiempos mayores de absorción y de efecto que las insulinas de acción rápida. Estos tiempos se hacen más extensos debido a la acción de una proteína básica o zinc. Así, en función de la concentración de zinc utilizada en su fabricación, se distinguen dos tipos:

- *Insulina de acción intermedia:* Actúan alrededor de dos o tres horas después de haber sido aplicadas, tienen su máximo efecto de cuatro a doce horas después de la inyección y su eficacia es de aproximadamente 12 a 18 horas.
- *Insulina de acción lenta:* Su tiempo de inicio de acción es de aproximadamente cuatro a seis horas y su efecto se prolonga durante periodos de tiempo relativamente largos que van desde las 24 horas hasta las 36 horas.

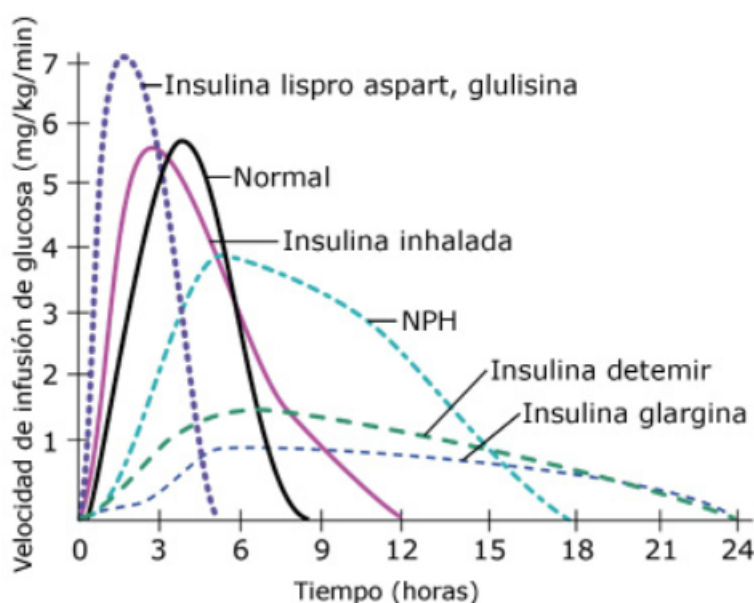


Figura 5.1: Diferentes tipos de insulina según su tiempo de acción. Tomada de [45].

Tipo de Insulina		Inicio	Pico	Duración
Acción rápida	Lyspro / Aspart / Glulisina	< 15 min	1 - 2 h	4 - 6 h
Acción corta	Regular / Normal	30 - 60 min	2 - 4 h	6 - 8 h
Acción intermedia	NPH	1 - 3 h	4 - 12 h	12 - 18 h
Acción lenta	Detemir	4 - 6 h	1 - 2 h	24 - 36 h
	Glargina	4-6 h	1 h	24-36 h

Tabla 5.1: Clasificación de los tipos de insulina según su tiempo de acción.

5.1.2. Tratamientos insulínicos

El tratamiento con suministro exógeno de insulina más común actualmente es de las inyecciones subcutáneas. Este tipo de terapia se lleva a cabo mediante la administración de tres o cuatro inyecciones de insulina diaria (lo que se conoce como múltiples inyecciones de insulina [34]). En el tratamiento con inyecciones se utiliza insulina de acción intermedia o lenta para cubrir las necesidades de insulina basal (es la que se produce entre comidas y durante el sueño para que la glucosa almacenada en el hígado pueda ingresar y ser absorbida por las células), esta se acumula bajo la piel y es absorbida por el torrente circulatorio paulatinamente hasta desaparecer. Sin embargo, no es claro cuánta cantidad de insulina es la que está siendo absorbida en determinada hora del día, así como tampoco su disponibilidad para llevar a cabo procesos metabólicos. La cantidad y el momento pueden variar de un día a otro.

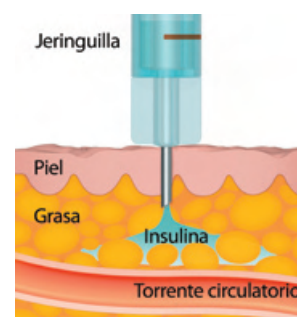
La variabilidad y la absorción de la insulina de acción intermedia o lenta no permite mantener los niveles de glucosa dentro de unos valores razonables y frecuentemente contribuyen a generar gran parte de las hiper o hipoglucemias que se producen sin razón alguna [32]. Luego de las inyecciones de insulina de acción intermedia o lenta:

- No es posible controlar cuándo será absorbida por el organismo.
- No se puede aumentar o disminuir la cantidad suministrada para que coincida con las diversas necesidades del organismo.
- No es posible ajustar la dosis de acuerdo con los cambios repentinos de rutina del paciente.

Finalmente, debe mencionarse que este tipo de tratamiento también requiere de una gran responsabilidad por parte del paciente, quien debe estar atento a inyectarse las veces que sean indicadas en su tratamiento, además de tener que realizar manualmente el cálculo de la dosis de insulina a aplicar con cada inyección, en función de la cantidad de carbohidratos que consume con cada comida, lo cual supone que un olvido en la aplicación de la inyección en algunos de los momentos del día en los que debe ser suministrado el medicamento, o errores en los cálculos de la dosis, pueden conllevar a que se presenten las complicaciones de salud ya mencionadas.



(a) Aplicación de insulina a través de inyección



(b) Acumulación de insulina debajo de la piel

Figura 5.2: Tratamiento de la diabetes a través de inyecciones. Tomadas de [32].

Otro tratamiento de tipo farmacológico con tendencia a ser muy usado durante los últimos años, es el que se realiza a través de una bomba de insulina. Estas bombas son dispositivos electrónicos programables que administran un ritmo “continuo” de insulina, por medio de un cánula que se conecta al abdomen del paciente, cerca al ombligo. Las bombas de insulina utilizan solamente insulina de acción rápida, ya que resulta ser más confiable por la forma en la que es absorbida y utilizada por el organismo [32], administrando pequeñas dosis cada hora, de manera similar a como sería suministrada la insulina basal por el páncreas.

De igual manera, estas bombas pueden administrar dosis bolus (cantidad grande de insulina que produce el páncreas para cubrir el aumento en la cantidad de glucosa que ingresa al torrente sanguíneo cuando se consumen los alimentos), sin embargo, en la mayoría de los dispositivos que se manejan en el mercado actualmente, es el paciente quien con apoyo del médico tratante debe programar la cantidad de insulina basal, llevando el registro de los niveles de glucosa tomados durante varios momentos del día y la noche, así como la cantidad de insulina en bolus a través del conteo de carbohidratos que el paciente desea comer. Si bien es cierto que este método de tratamiento ofrece un mejor control para el cuidado de la diabetes, puesto que las bombas imitan como ningún otro método la función del páncreas, debe mencionarse que en la mayor parte de estas siguen dependiendo de la manipulación por parte del paciente, pero lo ideal sería poder avanzar en la construcción y diseño de alternativas cada vez más automáticas e independientes, lo cual se logra al dotar a estos dispositivos de un algoritmo de control capaz de estimar la cantidad de insulina necesaria en cada momento para mantener el nivel de glucosa en sangre dentro del rango objetivo [21]. Actualmente, científicos e ingenieros trabajan en el desarrollo de dispositivos que se enlacen de manera inalámbrica con glucómetros que puedan enviar a la bomba el dato del nivel de glucosa que el paciente registra en ese instante de tiempo, para que este calcule de manera instantánea la cantidad de insulina a suministrar, es la esencia del trabajo a realizar por parte del “páncreas artificial”, no obstante, el único algoritmo de control probado hasta el momento es PID (de este método se mencionó algunas generalidades en la sección 1.6.4 del capítulo 1), a partir del cuál se han construido algunos dispositivos híbridos, es decir no son totalmente automáticos, es el paciente quién, al igual que en la bombas convencionales, debe ingresar manualmente la cantidad de carbohidratos, aceptar o no las sugerencias en la cantidad de insulina a suministrar por parte de la bomba, ingresar manualmente los niveles de glucosa cuando la comunicación entre la bomba y el glucómetro se rompa, y además solo se ha probado y utilizado en pacientes diabéticos tipo 1, por lo que el control óptimo de la diabetes mellitus sigue siendo todo un reto [10].

En las simulaciones desarrolladas en este trabajo se depende únicamente de la cantidad de carbohidratos que el paciente consume y las horas del día en las que toma los alimentos, ya que la cantidad de insulina a suministrar se realiza en periodos de tiempo más cortos buscando imitar mejor la función del páncreas, y es calculada en su totalidad por el algoritmo que soluciona el problema de control óptimo implementado sobre el modelo con parámetros ajustados que describe la dinámica de la glucosa-insulina de cada paciente, sin la necesidad de una programación previa del dispositivo. La estimación de los parámetros permite predecir

las concentraciones de glucosa y con esta información se optimiza la infusión de insulina, lo cual podría considerarse como una nueva alternativa en la búsqueda de producir dispositivos cada vez más automáticos.



(a) Aplicación de insulina a través de una bomba

(b) Ingreso de la insulina de acción rápida

Figura 5.3: Tratamiento de la diabetes a través de una bomba de insulina. Tomadas de [32].

5.2. Aspectos generales de las simulaciones

Es conveniente recordar aquí nuevamente la definición del problema de control óptimo se resolverá en cada una de las simulaciones consideradas:

$$\min_{u_0, u_1, \dots, u_{N-1}} \Delta t \sum_{k=0}^{N-1} (g_k - g_d)^2 + (u_k - \bar{u})^2 \quad (5.1)$$

Sujeto a:

$$\begin{aligned} g_{k+1} &= F_1(g_k, h_k, u_k) \\ h_{k+1} &= F_2(g_k, h_k, u_k) \\ r(g_T) &= [g_N - g_0] = 0 \end{aligned} \quad (5.2)$$

$$s(g_k, u_k) = \begin{bmatrix} g_k - \bar{g} \\ \bar{g} - g_k \\ u_k - u_{\text{mín}} \\ u_{\text{máx}} - u_k \end{bmatrix} \leq \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad k = 0, 1, \dots, N-1,$$

A continuación se mencionan algunos aspectos generales que se tienen en cuenta para la construcción de las simulaciones en los diferentes escenarios considerados:

1. **Punto de equilibrio basal:** De acuerdo con la tabla 1.1 mostrada en el capítulo 1, en todas las simulaciones tanto de pacientes diabéticos tipo 1 como tipo 2, se toma el valor de 100 mg/dL (5,56 mmol/L) como el punto de equilibrio alrededor del cual se intenta estabilizar el sistema (2.3) y que corresponde con su valor inicial, es decir, $g_0 = 100$ mg/dL y $h_0 = 2.7$ μ .U/mL. Estos valores se usan en el esquema numérico que permite encontrar las trayectorias de las variables de estado definidas por las dos primeras ecuaciones en (4.26).

2. **Límites de glucosa en sangre:** De igual manera, de acuerdo con la tabla 1.1 se utilizarán como concentraciones saludables de glucosa en sangre (o límites de normoglucesmia) los valores 60 y 180 mg/dL. Por lo tanto, $\underline{g} = 60$ mg/dL y $\bar{g} = 180$ mg/dL en (5.2)
3. **Límites de la insulina suministrada:** El control óptimo (dosis de insulina suministrada por la bomba) debe estar restringido, de acuerdo con la formulación del problema presentada en (4.8) y (4.9). Esto debido a que en el contexto del problema que se pretende resolver en este trabajo, los valores de insulina exógena no tienen sentido si son negativos, ni tampoco podrá tomar valores superiores a $100\mu\text{U/mL}$, puesto que podría causar una hipoglucesmia al paciente. Así pues,

$$0 \text{ } (\mu\text{U/mL}) \leq u(t) \leq 100 \text{ } (\mu\text{U/mL}),$$

donde $u_{\min} = 0$ ($\mu\text{U/mL}$) y $u_{\max} = 100$ ($\mu\text{U/mL}$) en (5.2).

4. **Tiempo de toma de muestras:** En la actualidad, ya se dispone de sensores de glucosa que permiten conocer los niveles de glucosa en sangre cada 5 minutos. Así pues, el problema se resuelve para los diferentes escenarios simulados en un intervalo de tiempo de $[0, 24]$ horas, con un paso de discretización $\Delta t = 1/12$, donde 12 es el número de subintervalos en los que se puede partir 1 hora en periodos de tiempo de 5 minutos, y $N = 24/\Delta t$.
5. **Glucosa deseada (g_d):** El término g_d en (5.1) representa la concentración de glucosa que se desea tuviese el paciente en cada instante del día que se simula. Aquí se usarán dos tipos de glucosa deseada, con el fin de considerar algunas alternativas, que permitan comparar y discutir los resultados obtenidos. En el primer caso, los niveles de glucosa deseada están dados por la solución numérica del sistema (2.3) con parámetros de un paciente sano, es decir, con los parámetros usados en la simulación presentada en la sección 2.3.1 ($a_1 = 0.05, a_2 = 1, a_3 = 0.03, a_4 = 4, b_1 = 0.5, b_2 = 2$). En el segundo caso, (g_d) toma un valor constante que se obtiene del promedio de los niveles de glucosa para un paciente sano durante las 24 horas.
6. **Comidas:** La ingesta de comidas se modela por medio de la función (2.5), donde el parámetro Q representa la cantidad de carbohidratos que se consumen en cada comida (desayuno, almuerzo, cena, onces y/o merienda), y el parámetro τ_i , $i = 1, 2, \dots, 5$ indica la hora del día en la que es consumido el alimento. En cada uno de los escenarios simulados se consideran diferentes cantidades de carbohidratos ingeridas, así como diferentes horas del día en las que son consumidas, esto con el fin de evaluar la propuesta presentada en este trabajo aplicada a varios estilos de alimentación que podría llegar a adoptar un paciente diabético. En cada uno de los escenarios simulados se indicará con detalle el menú junto con la cantidad total de carbohidratos y la hora del día en la que se consumen.

7. **Datos observados para la estimación de parámetros:** Los datos observados usados para la estimación de los parámetros en cada uno de los escenarios de alimentación, se construyen de la misma manera que los construidos para los experimentos numéricos que se presentan en la sección 3.8 del capítulo 3. Se asignan valores a cada uno de los parámetros del sistema (2.3) siendo (2.5) la función de alimento modificada para cada escenario de alimentación, y este sistema se resuelve para obtener una solución numérica. Posteriormente, a cada dato obtenido se le agrega un error generado aleatoriamente, de tal manera que $\|g_k \pm \varepsilon\| \leq 10^{-3}$ y $\|h_k \pm \varepsilon\| \leq 10^{-3}$ para $k = 1, 2, \dots, N$. Los datos perturbados se consideran ahora los datos observados para cada uno de los tiempos evaluados en la integración numérica del sistema (2.3). Finalmente, se aplican los algoritmos mostrados en el capítulo 3 para obtener la estimación de los parámetros, que se espera sea muy cercana a los parámetros que se usaron para la construcción de los datos observados, y esta estimación es utilizada para la solución del problema de control óptimo.
8. **Características del cómputo:** Las simulaciones presentadas en este capítulo fueron elaboradas usando el software MATLAB[®] versión 2019a, instalado en un computador con procesador AMD[®] Ryzen[™] 5 2500 U 3.6 GHz y una memoria RAM de 8 GB.

5.3. Simulación de los distintos escenarios

La simulación de los distintos escenarios contempla inicialmente un conjunto de datos que representan los niveles de glucosa e insulina de pacientes diabéticos tipo 1 y tipo 2 para un posible estilo de alimentación, sin considerar inicialmente el ingreso de insulina exógena. Posteriormente se ajustan los parámetros del modelo (2.3) y una vez ajustados se resuelve el problema de control óptimo que indicará la cantidad de insulina exógena a ser suministrada por la bomba o el dispositivo electrónico que cumpla estas funciones. Los menús contemplados en cada escenario de alimentación son de invención propia, complementados con información disponible en [32] y con ayuda de calculadoras de dieta disponibles en la web, como las que se encuentran alojadas en [35] y [19]. En las tablas que se incluyen en cada escenario, se referencia solamente la cantidad de carbohidratos (en gramos) que son consumidos por la persona en cada porción consumida, totalizando al final la cantidad de carbohidratos a consumir por comida. No se referencia el peso total por porción, porque esta información no resulta relevante y no es necesaria en los análisis y estudios del presente trabajo.

En seguida de la gráfica que presenta el comportamiento de la dinámica glucosa-insulina, junto con la gráfica del alimento y la cantidad de insulina a ser inyectada, se presentará una gráfica que describe el comportamiento del funcional durante las iteraciones que realice el algoritmo, y en su leyenda se incluye la siguiente información:

- N_{fe} : Número de evaluaciones de la función objetivo.
- N_{itr} : Número de iteraciones realizadas por el algoritmo hasta converger.

- *Time*: Tiempo (en segundos) utilizados por el algoritmo hasta converger.
- *Fval*: Evaluación de la función objetivo en el óptimo estimado por el algoritmo.

5.3.1. Escenario 1

Aquí se contempla la ingesta de 4 comidas (desayuno, almuerzo, cena y onces) para el caso de un paciente que no es muy riguroso con el conteo de la cantidad de carbohidratos que consume, pero procura que no sea muy alta. El menú considerado se presenta en la siguiente tabla.

Desayuno (8:00 am)		Onces (10:30 am)		Almuerzo (12:00 pm)		Cena (6:00 pm)	
2 tostadas	30 g	1 paquete de galletas saladas	35 g	1 sándwich de pavo	30 g	4 onzas de pechuga de pollo	0g
1 huevo hervido	0 g	1 gaseosa dietética	0 g	1 ensalada pequeña	5 g	1 papa pequeña la horno	30 g
1/2 vaso de jugo de naranja	15 g			1 taza de sopa de vegetales	15 g	1 taza de habichuelas verdes	5 g
1 tajada de tocino	0 g			1 manzana pequeña	15 g	1 pan	15 g
2 cucharadas de mermelada	30 g					1/2 vaso de jugo	15 g
Gramos totales	75 g	Gramos totales	35 g	Gramos totales	50 g	Gramos totales	65 g

Paciente Diabético tipo 1

Parámetro	β Inicial (β_0)	β Estimado ($\hat{\beta}$)
a_1	0.2937	0.2837
a_2	1.6237	1.5663
a_3	0.0145	0.0159
a_4	4.3796	4.0715
b_1	0.0001	0
b_2	1.8592	2.0351

Tabla 5.2: Resultados de la estimación de parámetros para el sistema (2.3) de un paciente diabético tipo 1 que se alimenta según el escenario 1. β_0 es el vector que contiene los valores iniciales de los parámetros, necesario para resolver el problema de estimación. $\hat{\beta}$ es el vector que contiene los parámetros estimados una vez se logra la convergencia de los algoritmos de estimación.

Observe que sin suministro de insulina exógena a un paciente diabético tipo 1, la concentración de glucosa en sangre puede alcanzar niveles cercanos a los 1000 mg/dL, tal como se aprecia en la figura 5.4, lo que podría resultar bastante peligroso para el paciente, no solo por el daño que niveles tan elevados de glucosa circulando por el torrente sanguíneo pueda causar a otros órganos del cuerpo, sino que además poner en riesgo al paciente de sufrir una cetoacidosis diabética que lo enviaría a una sala de urgencias, si no se logran disminuir los niveles de glucosa en cuestión de minutos.

En la figura 5.5 se observar que al suministrarse la insulina exógena, esta actúa y logra mantener la concentración de glucosa del paciente en niveles iguales o inferiores a 180 mg/dL

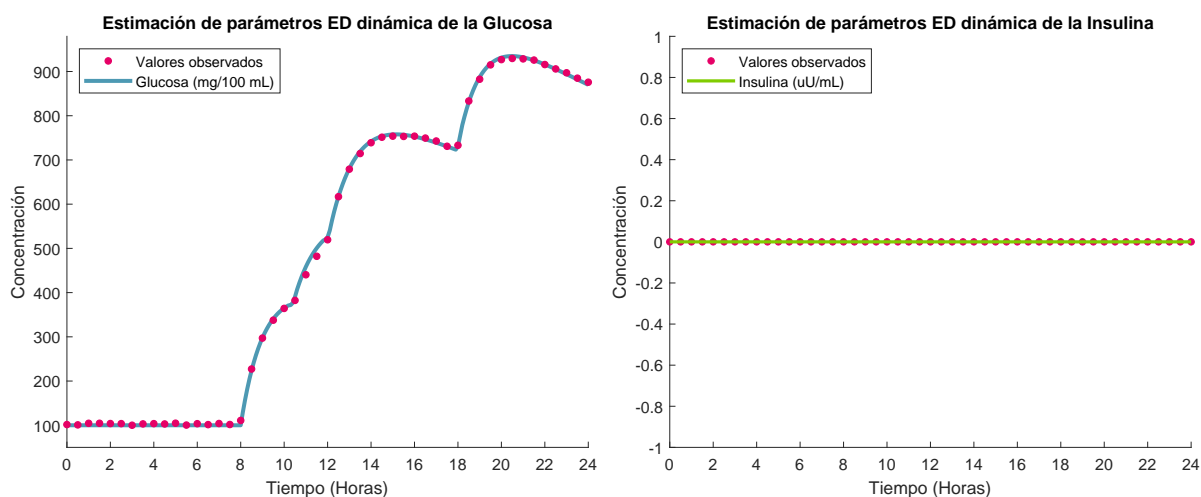


Figura 5.4: Conjunto de datos para la estimación de parámetros del modelo ajustado a la dinámica de un paciente diabético tipo 1.

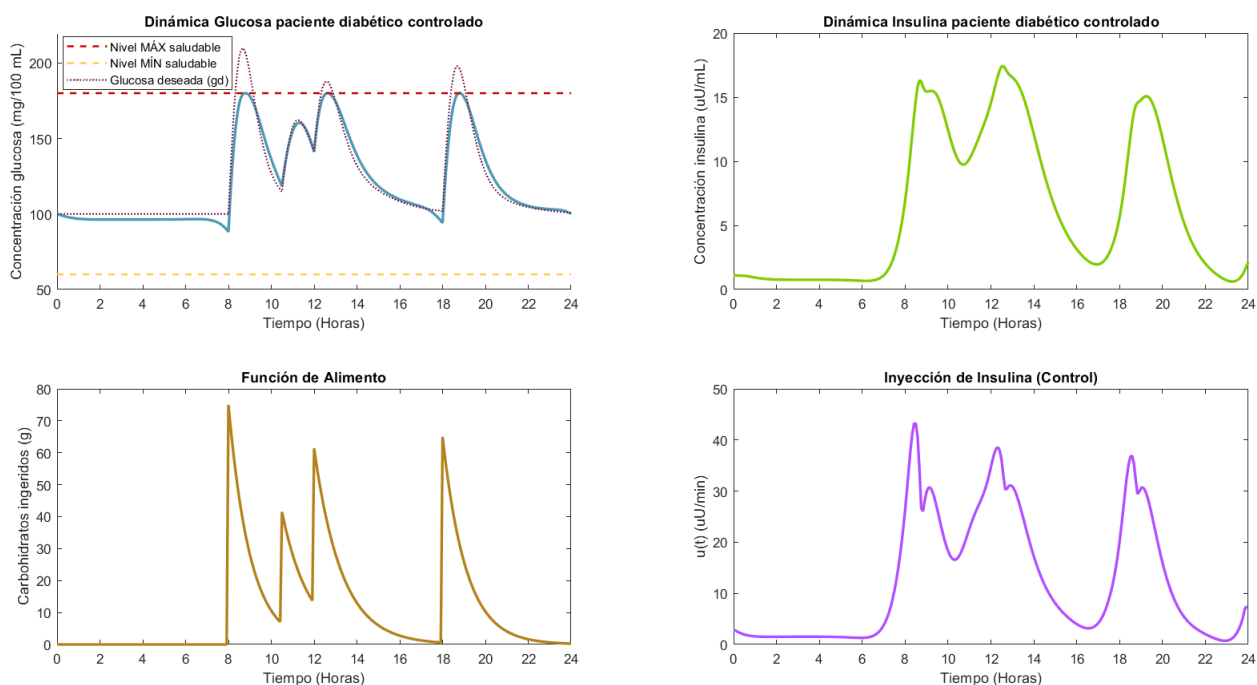


Figura 5.5: Gráficas de las dinámicas de glucosa e insulina, junto con su control óptimo para un paciente diabético tipo 1 que se alimenta según el escenario 1, con glucosa deseada según la opción 1.

pero superiores a 60 mg/dL que son los niveles de normogluemia establecidos, evitando así que el paciente presente algún episodio de hiper o hipogluemia durante todo el día. Sin embargo, nótese que a diferencia de lo que sucede en el caso de un paciente sano, donde ocurre que la insulina actúa con cierto retraso sobre la dinámica de la glucosa, para el

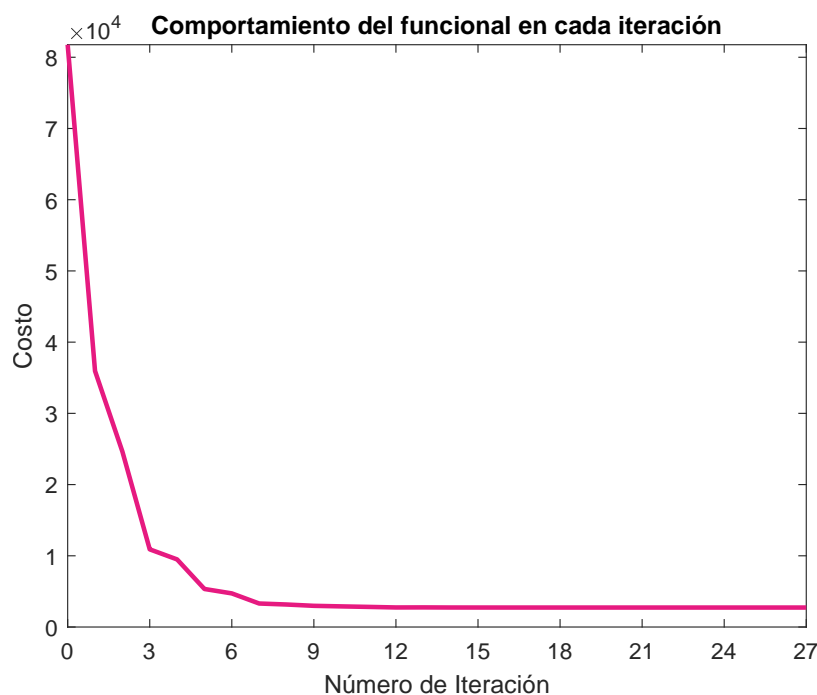


Figura 5.6: Funcional objetivo para glucosa deseada según forma 1, en cada iteración.
Nfe: 8101, *Nitr:* 27, *Time:* 62.4 s, *Fval:* 2.734×10^2 .

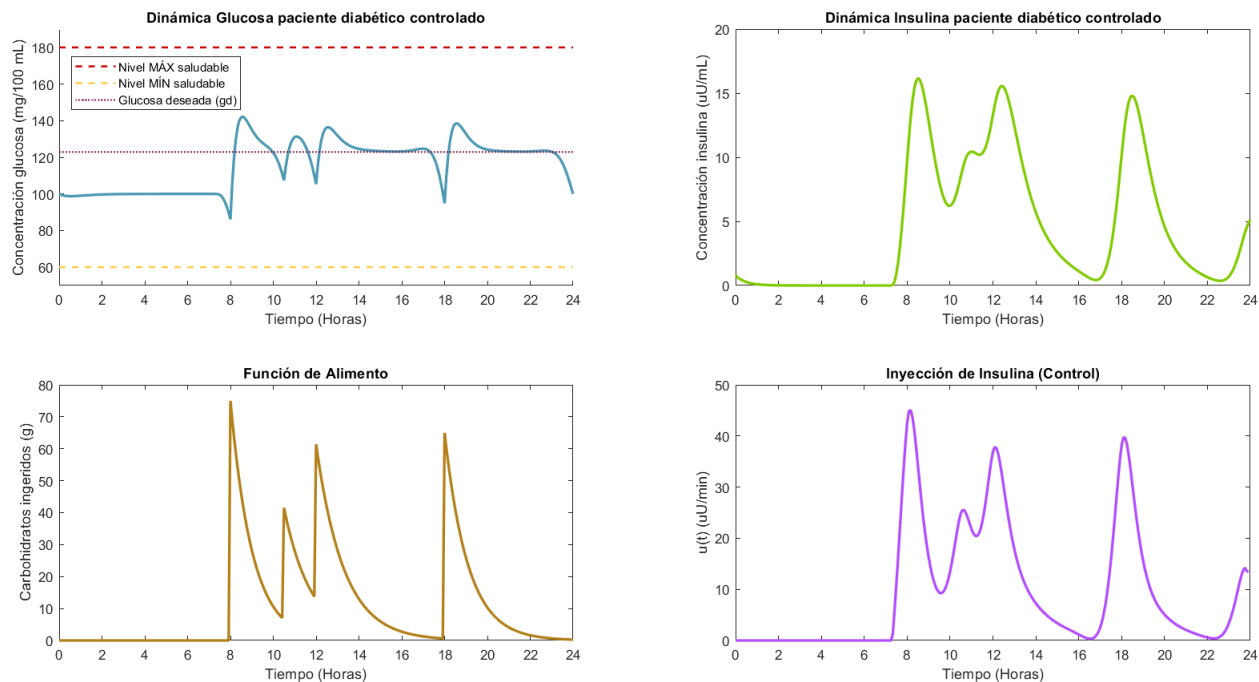


Figura 5.7: Gráficas de las dinámicas de glucosa e insulina, junto con su control óptimo para un paciente diabético tipo 1 que se alimenta según el escenario 1, con glucosa deseada según la opción 2.

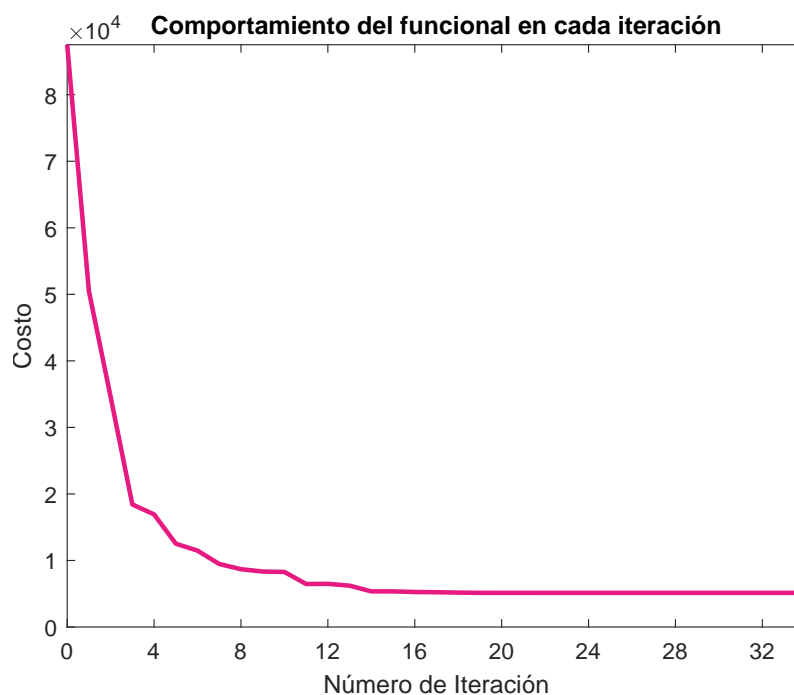


Figura 5.8: Funcional objetivo para glucosa deseada según forma 2, en cada iteración.
Nfe: 10137, *Nitr:* 34, *Time:* 71.8 s, *Fval:* 5.134×10^2 .

caso del paciente diabético debe empezar a actuar unos minutos antes de que el nivel de glucosa empiece a elevarse, para mantener la concentración por debajo del nivel máximo permitido. La cantidad máxima de insulina exógena suministrada durante todo el día es de $44.35 \mu\text{U}/\text{mL}$, lo que permite concluir que el problema de control óptimo usando PNL ha sido resuelto cumpliendo con las restricciones impuestas sobre la formulación del problema.

Note además que, cuando la glucosa deseada es dada por la dinámica del sistema (2.3) para un paciente sano, la concentración de glucosa es más alta y permanecen más cercana al límite máximo permitido como se observa en la gráfica superior izquierda de la figura 5.5, sin embargo, esta forma de seleccionar la glucosa deseada requiere mucha menos cantidad de insulina a ser suministrada para mantener los niveles de glucosa dentro los límites saludables, y aunque requiere que pequeñas dosis de insulina sean suministradas durante las primeras 7 horas y media del día, en total se requirieron $2422 \mu\text{U}/\text{mL}$ distribuidas a lo largo de las 24 horas del día para mantener la glucosa totalmente controlada. Por su parte, cuando la glucosa deseada se selecciona constante, la concentración en el torrente sanguíneo del paciente permanece muchísimo más baja como se aprecia en la gráfica superior izquierda de la figura 5.7, aquí el algoritmo de optimización calcula las cantidades de insulina que se requieren para que los niveles de glucosa se mantengan lo más cercanos que sea posible a la concentración de glucosa deseada, representada con la línea punteada, y es por esta razón que entre las 2 y las 4 pm, la dinámica de la glucosa tiende a mantenerse constante decayendo solo cuando se aumenta la cantidad de insulina que se suministra unos minutos antes de

ingerir el alimento. No obstante, como la glucosa deseada puede llegar a ser muy restrictiva, la cantidad de insulina que debe suministrarse para mantener concentraciones de glucosa cercanas a la deseada es mayor en este caso, donde es necesario suministrar $2727 \mu\text{U}/\text{mL}$ distribuidas a lo largo de las 24 horas del día. Finalmente, cabe también mencionar que la primera forma de selección de insulina implica un costo computacional mucho más bajo, en términos de menor cantidad de iteraciones, menor cantidad de evaluaciones de la función objetivo, menor tiempo de cómputo y un valor más pequeño del funcional, comparada con la segunda forma de selección de la glucosa deseada, como se observa en la gráfica de las figura 5.6 y 5.8 respectivamente.

Paciente Diabético tipo 2

Parámetro	β Inicial (β_0)	β Estimado ($\hat{\beta}$)
a_1	0.0659	0.0503
a_2	1.1627	1.0680
a_3	-0.0073	0.0162
a_4	4.3102	4.1762
b_1	0.0093	0.0098
b_2	1.8951	2.0561

Tabla 5.3: Resultados de la estimación de parámetros para el sistema (2.3) de un paciente diabético tipo 2 que se alimenta según el escenario 1.

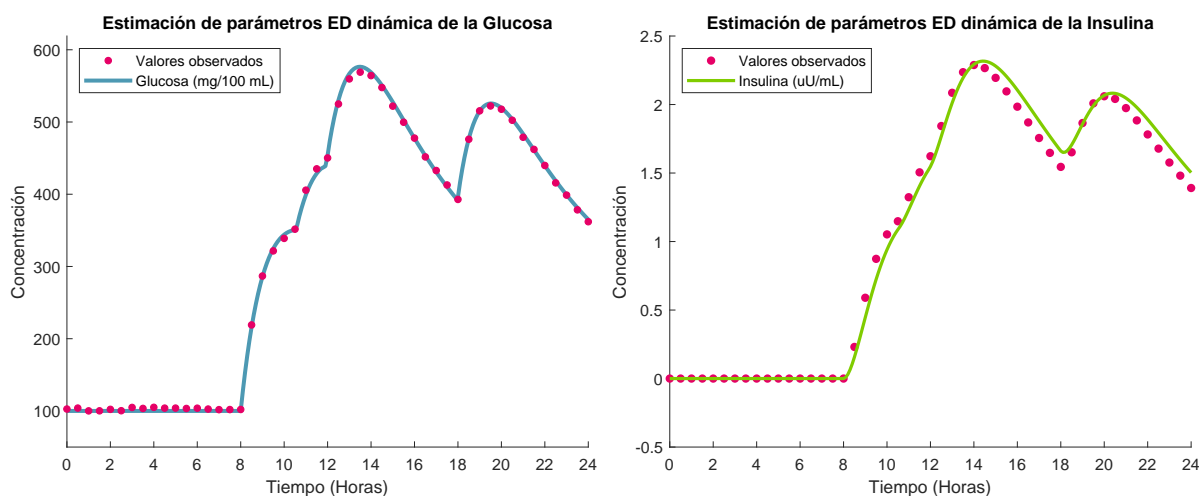


Figura 5.9: Conjunto de datos para la estimación de parámetros del modelo ajustado a la dinámica de un paciente diabético tipo 2.

Observe que en este caso, a diferencia del paciente diabético tipo 1, los niveles de glucosa alcanzados sin el suministro de insulina exógena no superan el valor de $600 \text{ mg}/\text{dL}$, y esto es debido a que las personas que padecen de este tipo de diabetes sí producen insulina (por lo que el valor de parámetro b_1 es mayor a cero) como se aprecia en la tabla 5.3 pero la cantidad producida es insuficiente o no puede aprovecharse eficientemente.

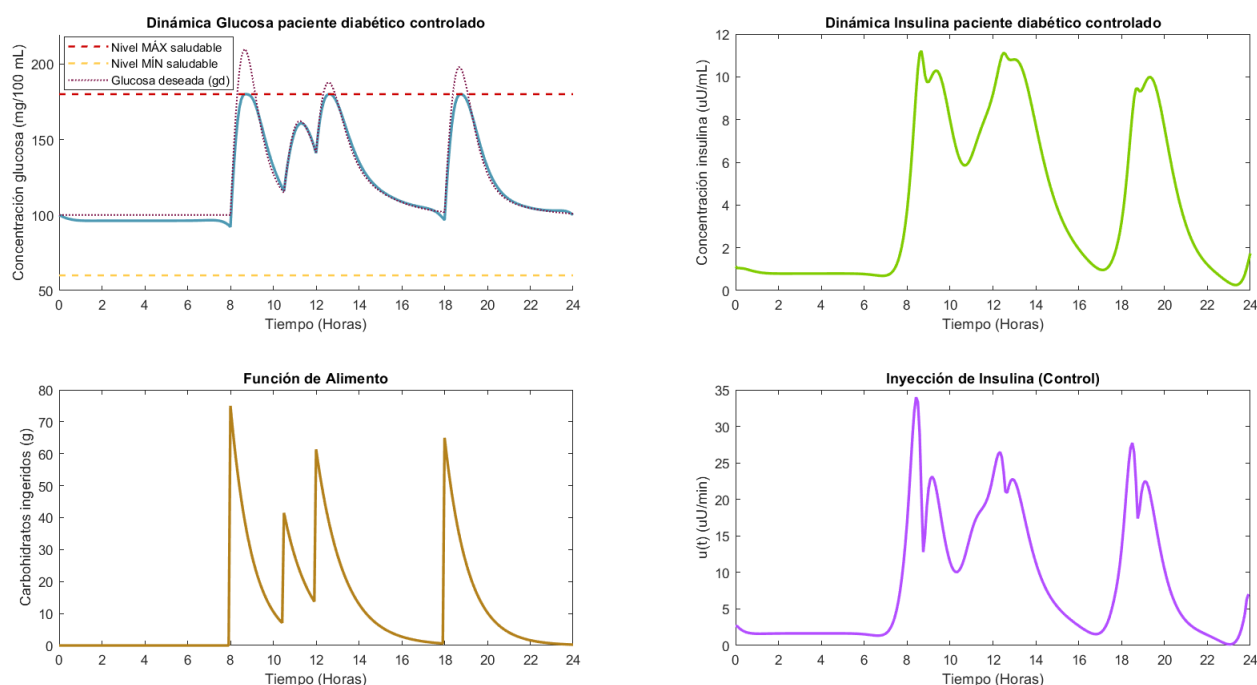


Figura 5.10: Gráficas de las dinámicas de glucosa e insulina, junto con su control óptimo para un paciente diabético tipo 2 que se alimenta según el escenario 1, con glucosa deseada según la opción 1.

En el gráfico de la figura 5.9, se observa que este paciente sí produce insulina pero es muy poca para la cantidad de glucosa que ingresa al organismo a través del proceso de alimentación (solo alcanza un nivel máximo de $2.63 \mu\text{U}/\text{mL}$).

Al ser suministrada la insulina exógena que proporciona la bomba a lo largo del día, se observa tanto en la figura 5.10 como en la figura 5.12 que los niveles de glucosa han bajado quedando entre los límites permitidos, tendiendo siempre a estar muy cercanos a las concentraciones de glucosa deseada. Nótese también que la cantidad de insulina inyectada es más baja que la cantidad suministrada para el paciente diabético tipo 1, y esto se debe a que la insulina que produce el paciente de manera natural también es utilizada en su proceso metabólico, lo que implica que requerirá un menor suministro del fármaco por parte de la bomba de insulina. Por otro lado, la solución al problema de control óptimo para el paciente diabético tipo 1, como era de esperarse, requirió un número mayor de iteraciones, de número de evaluaciones de la función objetivo a minimizar y de tiempo de ejecución del algoritmo de optimización, que para el caso del paciente diabético tipo 2, ya que la diabetes tipo 1 suele ser más complicada de tratar debido a la total dependencia al medicamento a la que están sujetos estos pacientes, lo cual implica para este caso, un mayor tiempo de ejecución o cómputo de los algoritmos y rutinas que determinan la cantidad de insulina a suministrar.

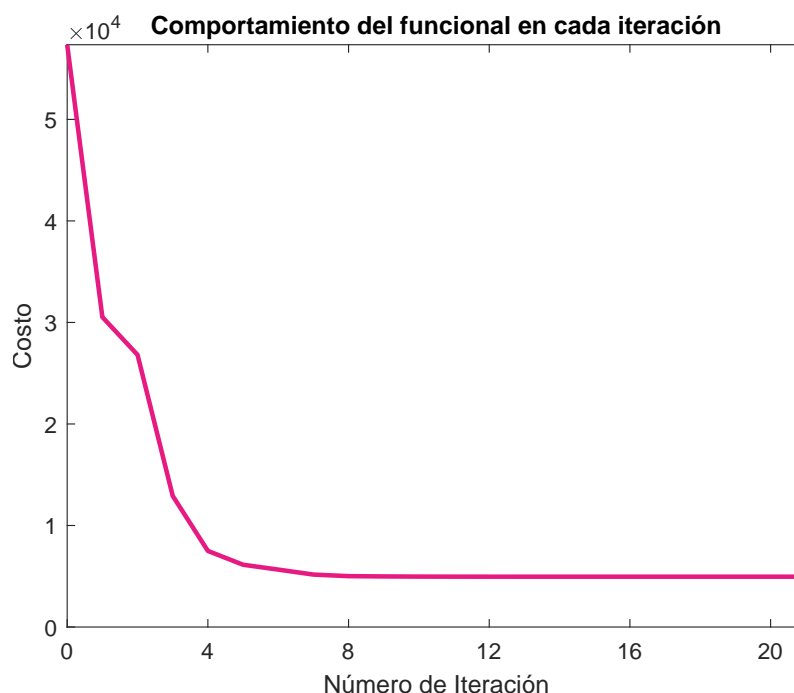


Figura 5.11: Funcional objetivo para glucosa deseada según forma 1, en cada iteración.
Nfe: 6361, *Nitr:* 21, *Time:* 41.9 s, *Fval:* 4.957×10^2 .

Con respecto a las dos diferentes formas de selección de la glucosa deseada, en este caso se observan comportamientos muy similares a los observados para el caso del paciente diabético tipo 1. Para la primera forma de selección de la glucosa deseada la concentración del paciente se mantiene más alta pero dentro de los niveles máximo y mínimo permitidos, tendiendo a estar siempre muy cercanas, y para la segunda forma de selección de glucosa deseada, la concentración del paciente se mantiene en niveles mucho más bajos que en la primera forma, esto debido a que el problema de control óptimo busca minimizar la distancia entre la glucosa deseada y la concentración de glucosa del paciente, y como en este caso es un valor constante cercano al nivel basal, de ahí que se genere este comportamiento. Al igual que para el paciente diabético tipo 1, aquí también la primera forma de selección de glucosa deseada implica menor cantidad de insulina a ser suministrada durante todo el día, y menor costo computacional, como se observa en la gráfica inferior derecha y la gráfica de las figuras 5.10 y 5.11 respectivamente, al ser comparadas con las gráficas de las figuras 5.12 y 5.13 de la segunda forma de selección de glucosa deseada.

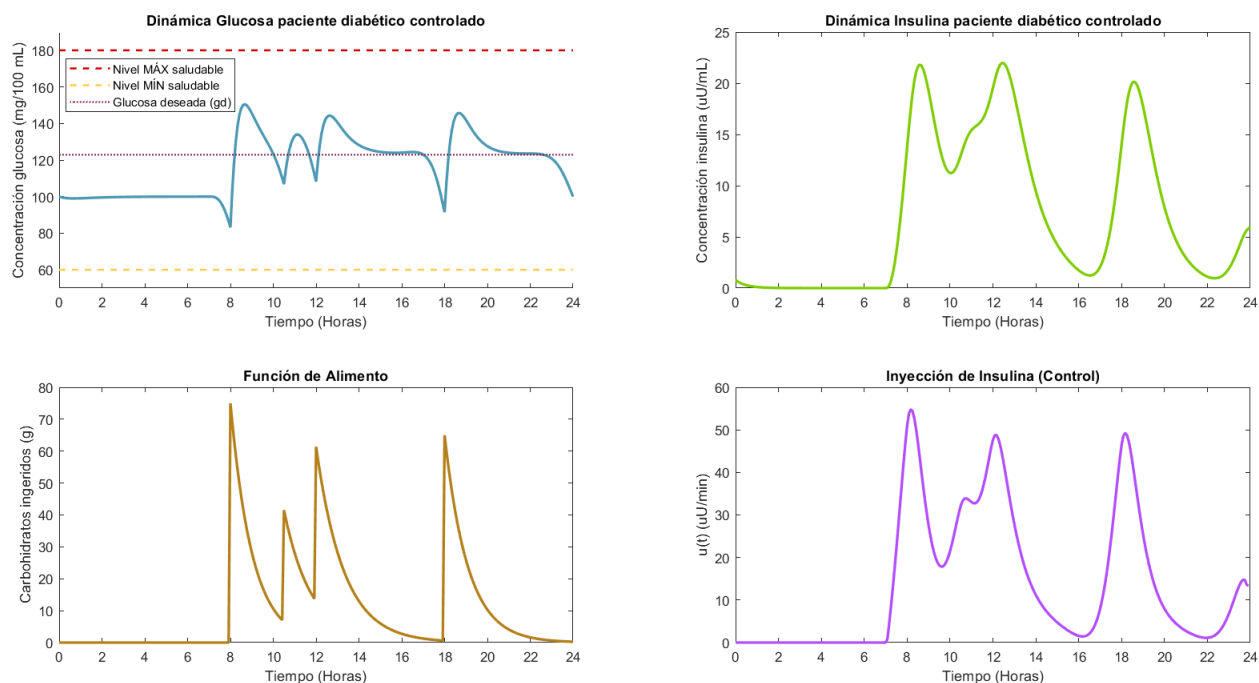


Figura 5.12: Gráficas de las dinámicas de glucosa e insulina, junto con su control óptimo para un paciente diabético tipo 2 que se alimenta según el escenario 1, con glucosa deseada según la opción 2.

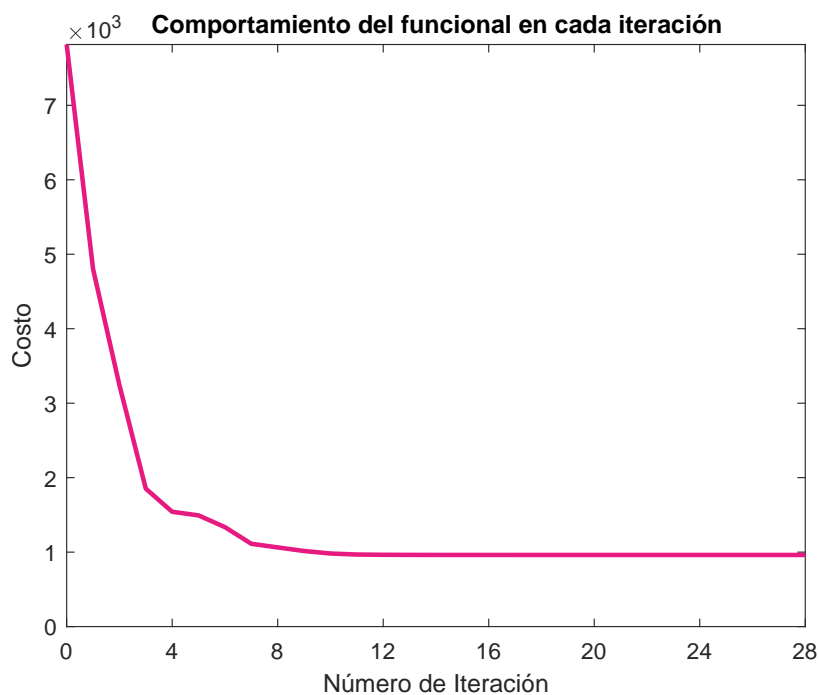


Figura 5.13: Funcional objetivo para glucosa deseada según forma 2, en cada iteración.
Nfe: 8122, *Nitr:* 28, *Time:* 61.9 s, *Fval:* 9.618×10^2 .

5.3.2. Escenario 2

En la mayoría de los casos, muchos pacientes diabéticos deben complementar su tratamiento para el manejo de la diabetes con dietas bajas en carbohidratos, puesto que como ya se mencionó, son los carbohidratos los responsables del aumento de los niveles de glucosa una vez digeridos por el sistema digestivo, y de ahí la importancia de mantener su consumo al mínimo. Por esta razón, para este escenario se contempla un estilo de alimentación que considera 5 comidas, todas estas muy bajas en carbohidratos, lo que no significa que los pacientes deban comer pequeñas porciones o diseñar menús que contengan solo verduras, como se verá a continuación.

Desayuno (8:00 am)		Onces (10:30 am)		Almuerzo (12:30 pm)	
1 taza de café sin azúcar	0 g	1 manzana	15 g	4 cucharadas de arroz integral	10 g
2 tajadas de pan integral	20 g	10 unidades de marañón	5 g	1/2 taza de frijoles negros	15 g
2 tajadas de jamón	4 g			120 g de pollo al horno	0,12 g
1 tajada de melón	8 g			1 taza de vegetales salteados	17 g
Gramos totales	32 g	Gramos totales	20 g	Gramos totales	42 g
Merienda (3:30 pm)		Cena (7:00 pm)			
1 yogurt bajo en azúcar	10 g	120 g de salmón o atún	0 g		
2 tajadas de pan integral	20 g	4 cucharadas de arroz integral	10 g		
1 tajada de queso blanco	1 g	1 taza de vegetales al vapor	0 g		
Gramos totales	31 g	Gramos totales	10 g		

Paciente Diabético tipo 1

Parámetro	β Inicial (β_0)	β Estimado ($\hat{\beta}$)
a_1	0.0392	0.0579
a_2	1.2479	1.0104
a_3	0.0139	0.0135
a_4	4.4500	4.0324
b_1	0.0001	0
b_2	1.9564	2.0886

Tabla 5.4: Resultados de la estimación de parámetros para el sistema (2.3) de un paciente diabético tipo 1 que se alimenta según el escenario 2.

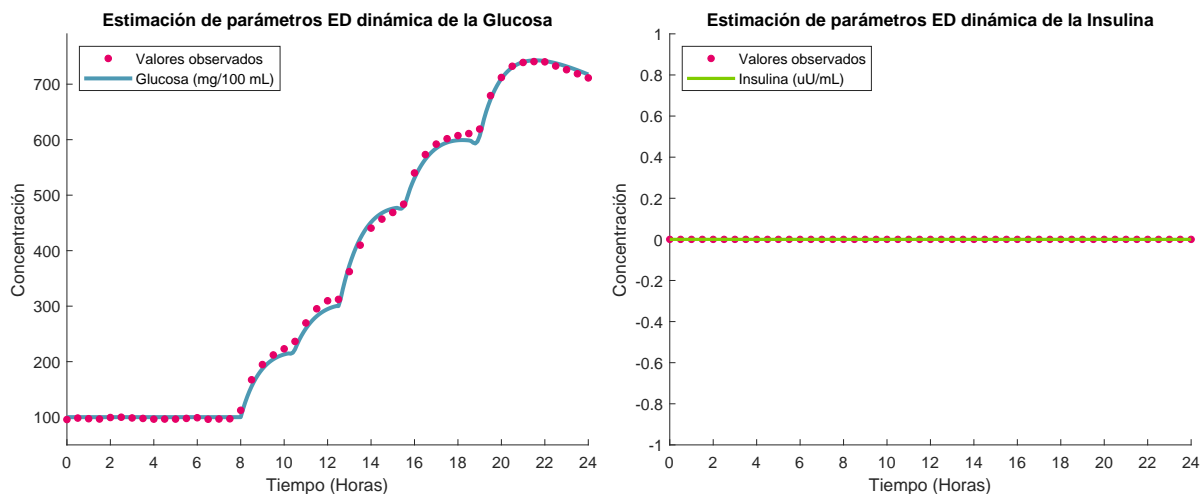


Figura 5.14: Conjunto de datos para la estimación de parámetros del modelo ajustado a la dinámica de un paciente diabético tipo 1.

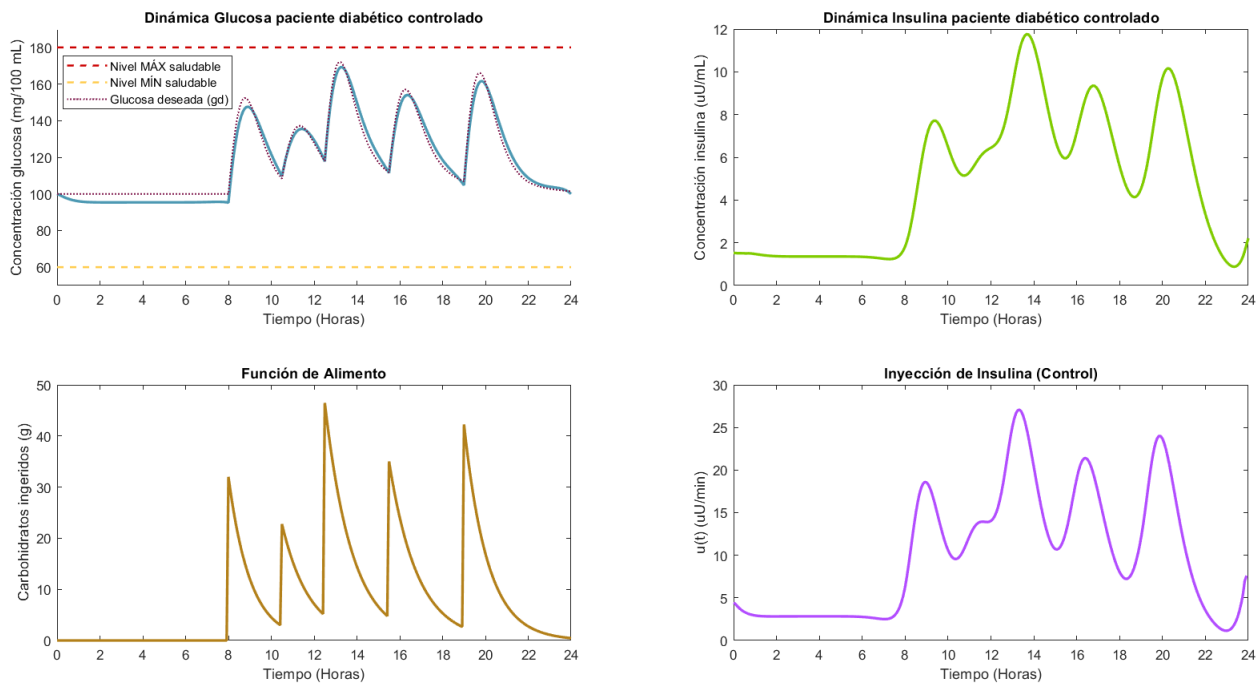


Figura 5.15: Gráficas de las dinámicas de glucosa e insulina, junto con su control óptimo para un paciente diabético tipo 1 que se alimenta según el escenario 2, con glucosa deseada según la opción 1.

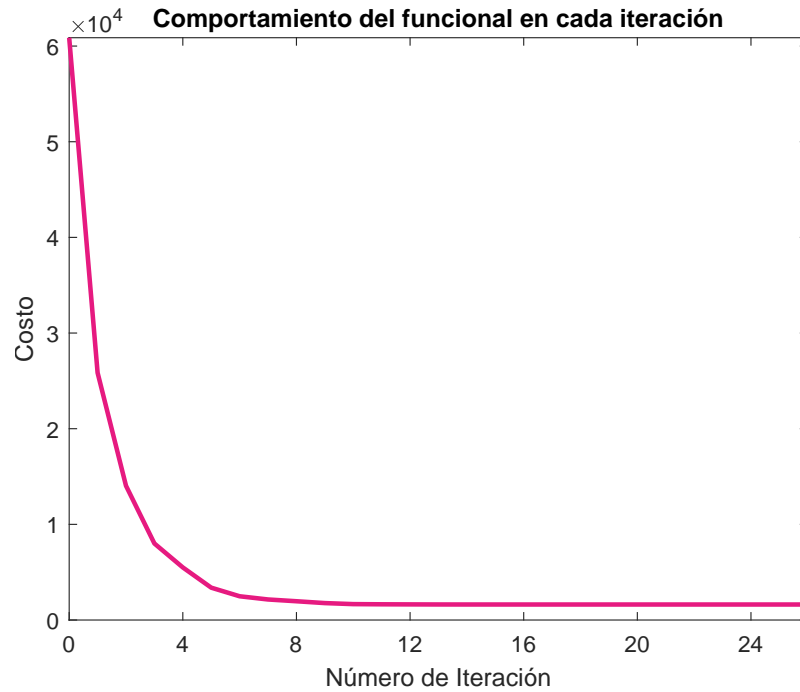


Figura 5.16: Funcional objetivo para glucosa deseada según forma 1, en cada iteración.
Nfe: 7809, *Nitr:* 26, *Time:* 24.1 s, *Fval:* 1.621×10^2 .

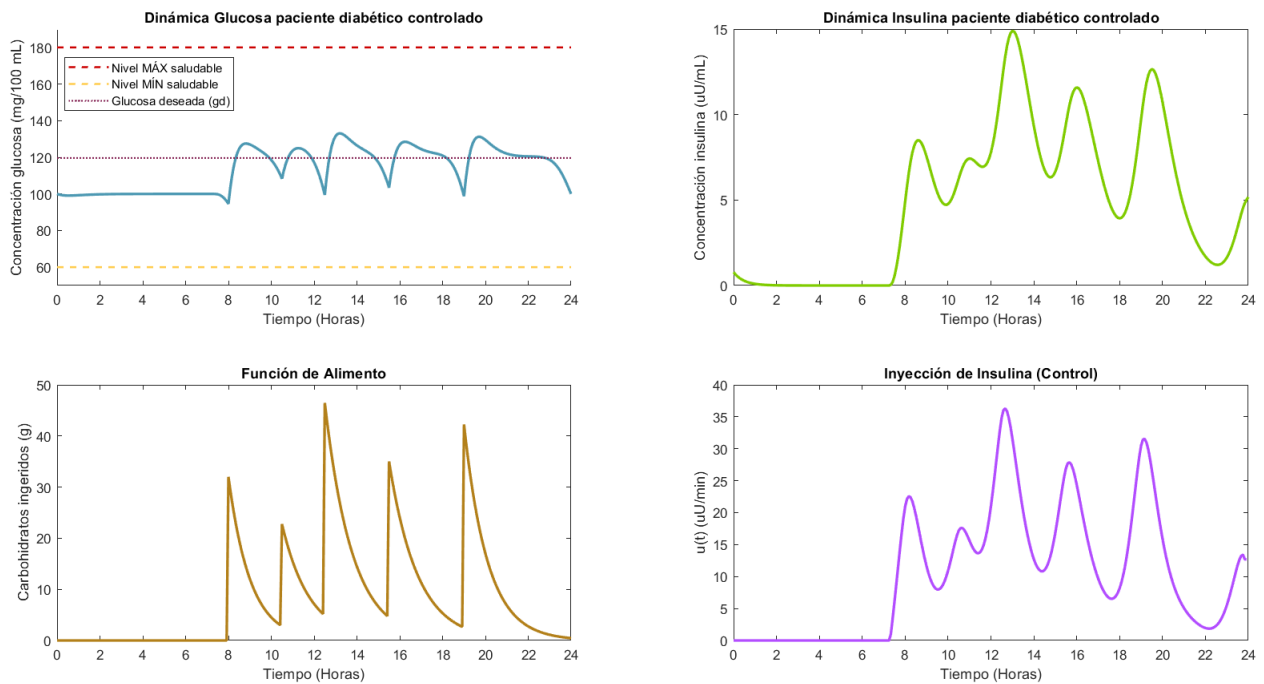


Figura 5.17: Gráficas de las dinámicas de glucosa e insulina, junto con su control óptimo para un paciente diabético tipo 1 que se alimenta según el escenario 2, con glucosa deseada según la opción 2.

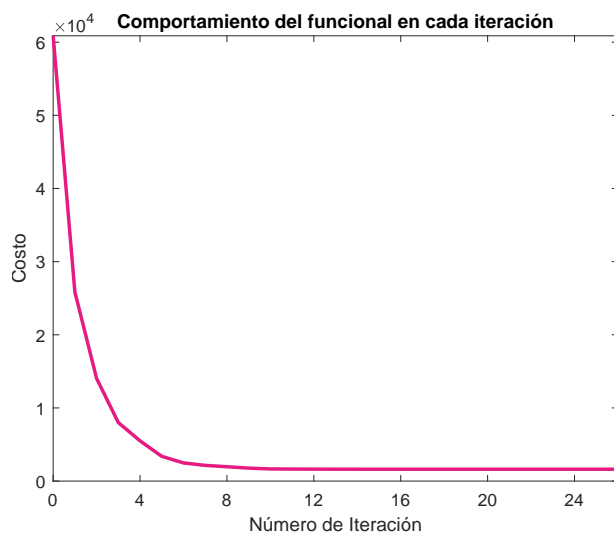


Figura 5.18: Funcional objetivo para glucosa deseada según forma 2, en cada iteración.
Nfe: 8410, *Nitr:* 28, *Time:* 31.5 s, *Fval:* 3.724×10^2 .

Paciente Diabético tipo 2

Parámetro	β Inicial (β_0)	β Estimado ($\hat{\beta}$)
a_1	0.077	0.0525
a_2	1.2447	1.0734
a_3	0.2970	0.0153
a_4	4.4955	4.0592
b_1	0.0078	0.0099
b_2	1.9040	2.0232

Tabla 5.5: Resultados de la estimación de parámetros para el sistema (2.3) de un paciente diabético tipo 2 que se alimenta según el escenario 2.

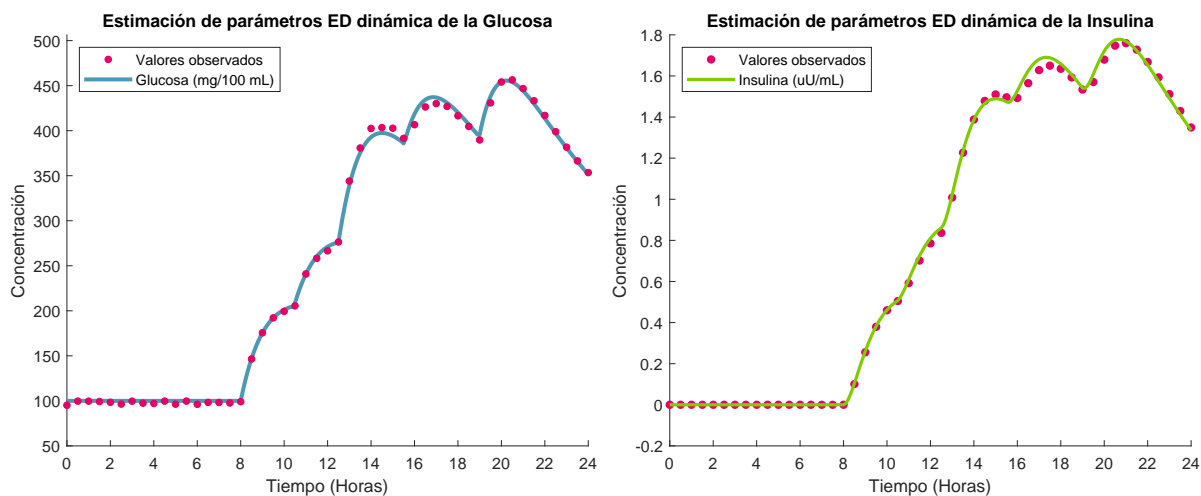


Figura 5.19: Conjunto de datos para la estimación de parámetros del modelo ajustado a la dinámica de un paciente diabético tipo 2.

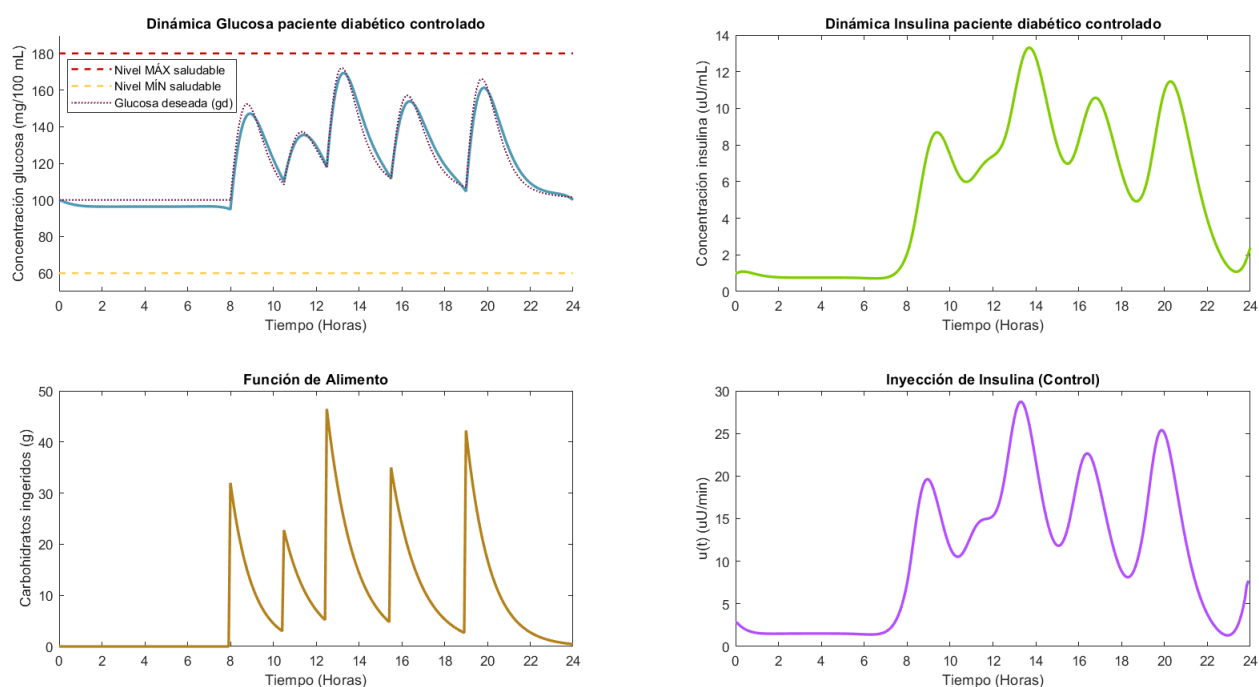


Figura 5.20: Gráficas de las dinámicas de glucosa e insulina, junto con su control óptimo para un paciente diabético tipo 2 que se alimenta según el escenario 2, con glucosa deseada según la opción 1.

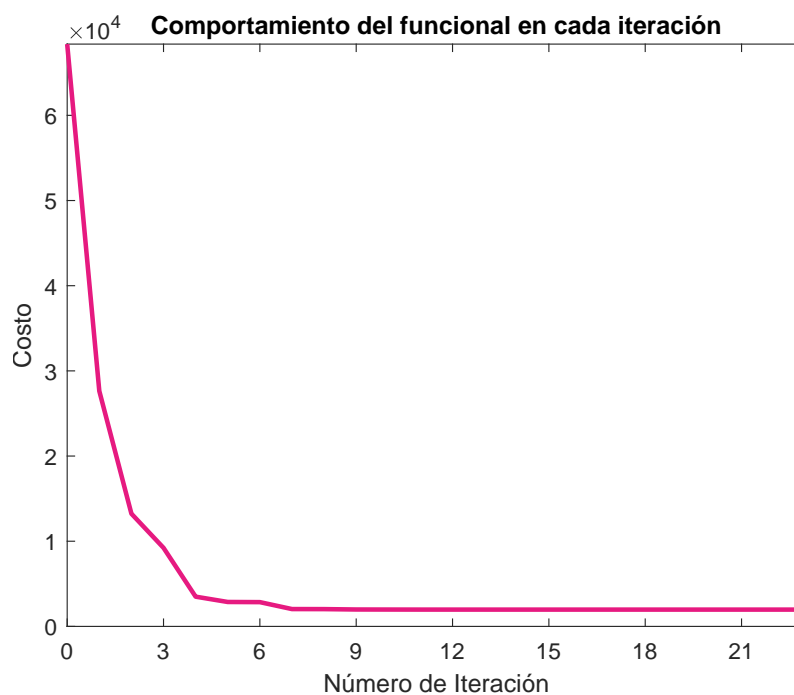


Figura 5.21: Funcional objetivo para glucosa deseada según forma 1, en cada iteración.

Nfe: 6940, *Nitr:* 23, *Time:* 23.5 s, *Fval:* 1.971×10^2 .

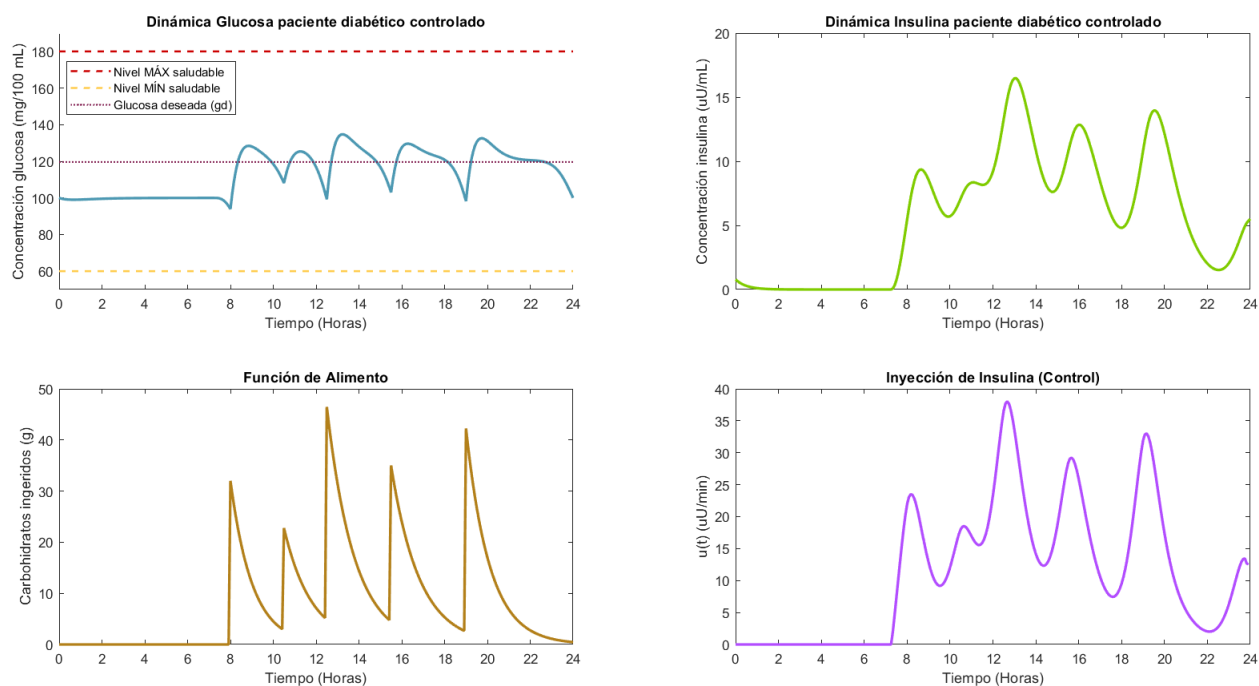


Figura 5.22: Gráficas de las dinámicas de glucosa e insulina, junto con su control óptimo para un paciente diabético tipo 2 que se alimenta según el escenario 2, con glucosa deseada según la opción 2.

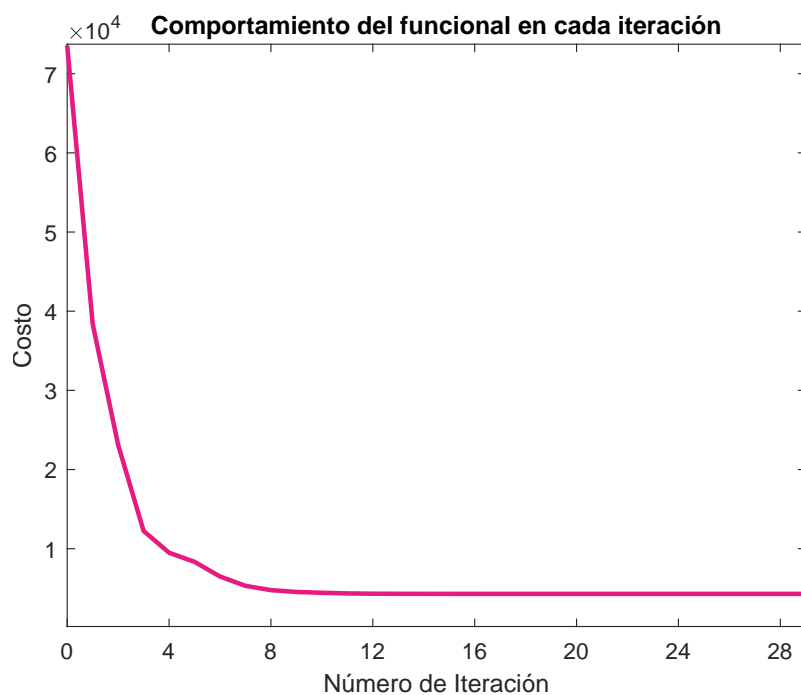


Figura 5.23: Funcional objetivo para glucosa deseada según forma 2, en cada iteración.
Nfe: 8679, *Nitr:* 29, *Time:* 35.5 s, *Fval:* 4.292×10^2 .

Como se puede observar en las figuras 5.14 y 5.19, los niveles de glucosa tanto para el paciente diabético tipo 1 como para el tipo 2 se encuentran por debajo de 1000 mg/dL sin el suministro exógeno de insulina, a diferencia de lo que se observó en el escenario 1, a pesar de que en el menú considerado en este escenario se contempla el consumo de una comida adicional con respecto al escenario anterior, y esto se debe a la baja cantidad de carbohidratos que contienen los alimentos que consumen los pacientes que siguen una cuidadosa dieta. Aún así, los niveles de glucosa siguen siendo muy altos y están supremamente alejados de los niveles superior e inferior de glucosa permitidos, por lo que debe aplicarse entonces el suministro de insulina exógena para reducirlos y llevarlos a los niveles de normoglucemia.

En las gráficas ubicadas en la esquina superior izquierda de las figuras 5.15, 5.17, 5.20 y 5.22, se aprecia que efectivamente se ha logrado reducir los niveles de glucosa en sangre a partir del suministro de insulina exógena calculado como solución del problema de control óptimo, de tal manera que incluso las concentraciones de glucosa se mantienen muy por debajo de 180 mg/dL, además de que la cantidad de insulina requerida es mucho menor que la requerida para en reducción de los niveles de glucosa de pacientes con estilos de alimentación similares al contemplado en el escenario 1. Además, nótese que los costos de cómputo asociados a la simulación de este escenario (ver figuras 5.16, 5.18, 5.21 y 5.23) son menores que los encontrados en el escenario 1. Estos resultados permiten validar el hecho de que una dieta baja en carbohidratos funciona muy bien como parte del tratamiento de la enfermedad de un paciente diabético, lo que reduciría costos tanto del medicamento, lo cual resultaría ser muy beneficioso para los sistemas de salud, como en los requerimientos que deban cumplir los dispositivos que sean fabricados como bombas de insulina.

En cuanto a la comparación entre los dos tipos de pacientes diabéticos considerados en la simulación de este escenario, como era de esperarse, el paciente diabético tipo 1 requerirá mayor cantidad de suministro de insulina durante todo el día, aunque el comportamiento de la dinámica de la glucosa como el de la insulina resultaron ser muy parecidos en ambos tipos de paciente. Esto sugiere entonces que el parámetro b_1 del modelo (2.3) no es sensible a los pequeños cambios que se realice sobre los valores que toma.

Para terminar, al centrar la atención en la selección de la forma de la glucosa deseada, en las gráficas ubicadas en la esquina superior izquierda de las figuras 5.15 y 5.20 se observa nuevamente que las concentraciones de glucosa resultantes son más altas que obtenidas en las respectivas gráficas de las figuras 5.17 y 5.22, esto debido a la libertad que ofrece la forma 1 de selección de glucosa deseada con respecto a la forma 2 que resulta ser bastante restrictiva en la medida en que se obliga a la concentración de glucosa del paciente a estar cercana a un valor constante durante todo el día sin la posibilidad de considerar niveles un poco más elevados, pero que siguen estando al interior de los límites permitidos. Aquí también se aprecia una ventaja en cuanto a las cantidades de insulina suministrada para el cumplimiento del objetivo al seleccionar la forma 1 de la glucosa deseada, que es más baja que las necesarias al seleccionar la forma 2, según lo muestran las gráficas ubicadas en la esquina

inferior derecha de las figuras 5.15 y 5.20 al ser comparadas con las correspondientes gráficas en las figuras 5.17 y 5.22, aunque en la segunda forma de selección de glucosa deseada, se presenta nuevamente el hecho de que no es necesario el suministro de insulina durante las primeras 7 horas y medias del día. En cuanto a costos computacionales, la selección de la forma 1 de glucosa deseada presenta un mejor rendimiento (en tiempo de cómputo, cantidad de iteraciones, cantidad de evaluaciones de la función objetivo y valor del funcional) para pacientes diabéticos tanto tipo 1 como tipo 2, que el ofrecido por la selección de la forma 2, al comparar las gráficas y la información suministrada en las figuras 5.16 y 5.21 con los resultados obtenidos en las figuras 5.18 y 5.23.

5.3.3. Escenario 3

Así como existen pacientes muy responsables con el tratamiento de su enfermedad, que llevan una dieta saludable y son muy cuidadosos con el suministro de las dosis de insulina según las recomendaciones médicas, existen otros pacientes que no son conscientes de su enfermedad y de las graves consecuencias que pueden contraerse de una diabetes mal tratada al no tener en cuenta los cuidados que deberían seguirse. En este escenario se simula el estilo de alimentación de un paciente que no restringe en absoluto la cantidad de carbohidratos que ingiere, al tener una marcada preferencia por el consumo de comida conocida como “chatarra”. El tipo de alimentos que ingiere en cada comida se muestra en la siguiente tabla, así como la estimación de los parámetros del modelo ajustados al conjunto de datos que se obtiene de la medición de la glucosa sin suministro de insulina y la solución al problema de control óptimo, para ambos pacientes (diabético tipo 1 y tipo 2), tal como se ha hecho en los escenarios considerados anteriormente.

Desayuno (9:00 am)		Almuerzo (1:00 pm)		Cena (6:00 pm)	
1 taza de cereal	25 g	1 hamburguesa 1/4 de libra	37 g	2 porciones de pizza	64 g
1 tajada de pan	25 g	1 porción de papas a la francesa	21 g	1 vaso (200 mL) de gaseosa	22 g
1 vaso de jugo de naranja	26 g	1 vaso (200 mL) de gaseosa	22 g		
Gramos totales	76 g	Gramos totales	80 g	Gramos totales	86 g

Paciente Diabético tipo 1

Parámetro	β Inicial (β_0)	β Estimado ($\hat{\beta}$)
a_1	0.1532	0.0486
a_2	1.2894	1.1453
a_3	0.0094	0.0108
a_4	4.3880	3.9952
b_1	0.0001	0
b_2	2.3670	2.0157

Tabla 5.6: Resultados de la estimación de parámetros para el sistema (2.3) de un paciente diabético tipo 1 que se alimenta según el escenario 3.

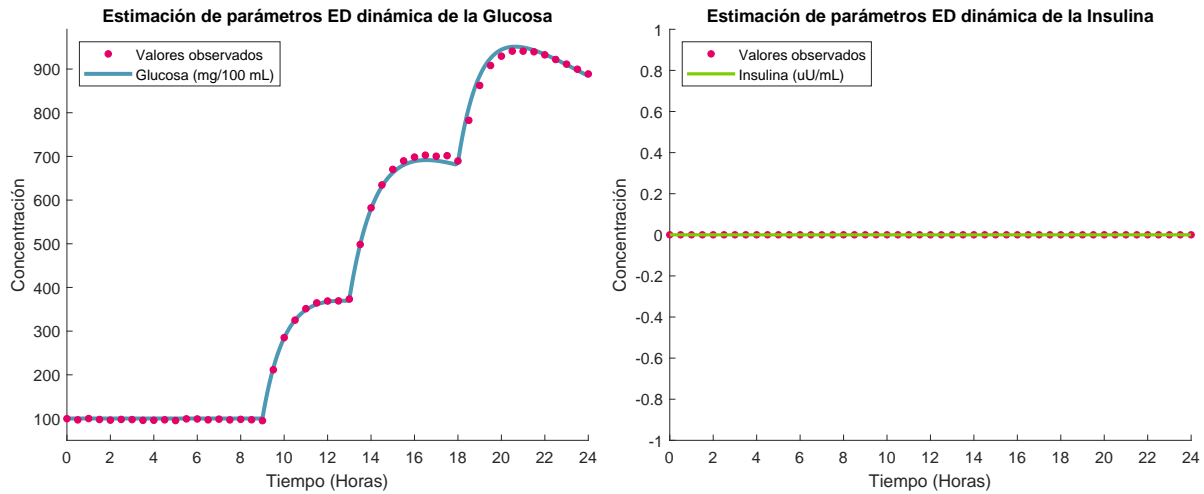


Figura 5.24: Conjunto de datos para la estimación de parámetros del modelo ajustado a la dinámica de un paciente diabético tipo 1.

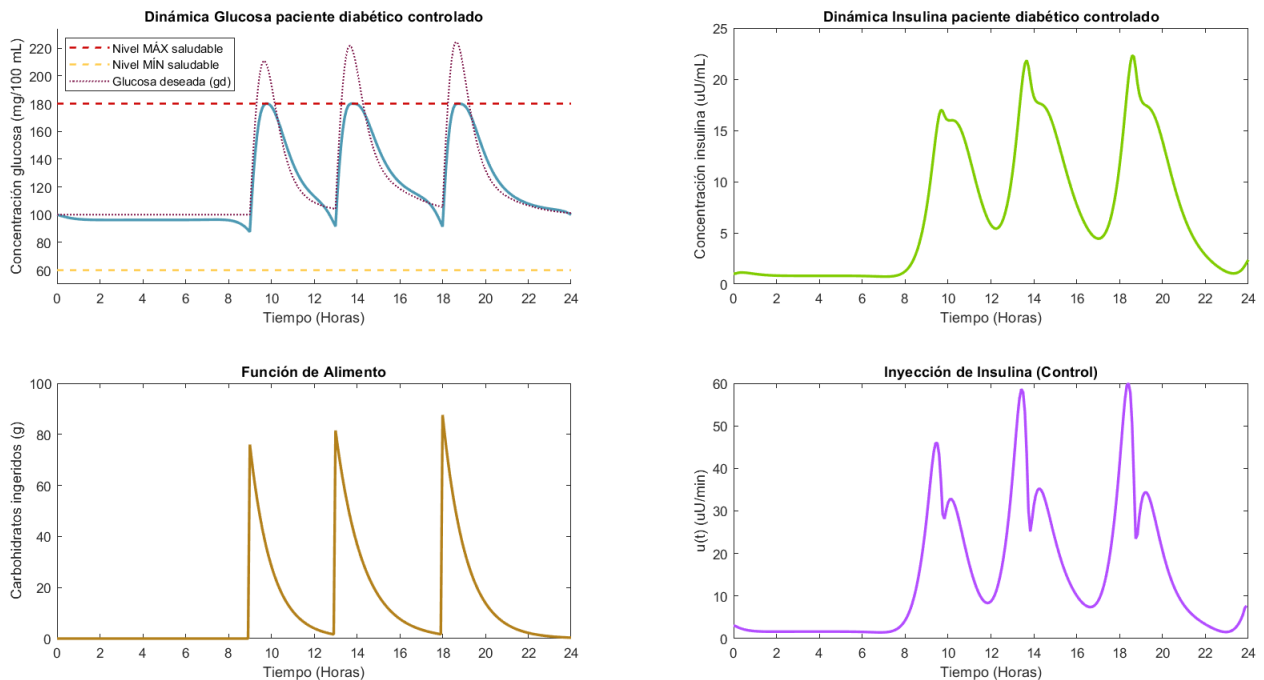


Figura 5.25: Gráficas de las dinámicas de glucosa e insulina, junto con su control óptimo para un paciente diabético tipo 1 que se alimenta según el escenario 3, con glucosa deseada según la opción 1.

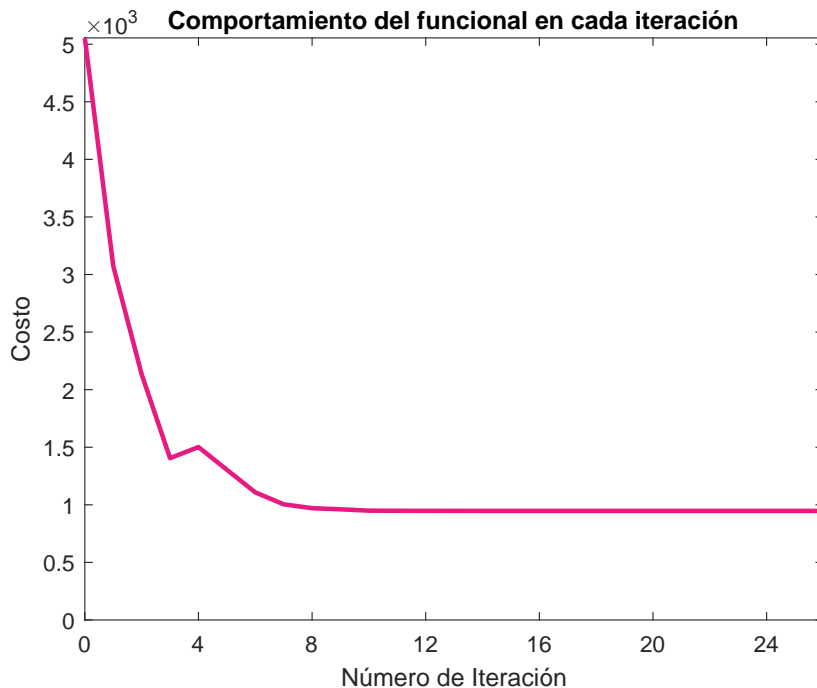


Figura 5.26: Funcional objetivo para glucosa deseada según forma 1, en cada iteración. *Nfe:* 7812, *Nitr:* 26, *Time:* 28.14 s, *Fval:* 9.471×10^2 .

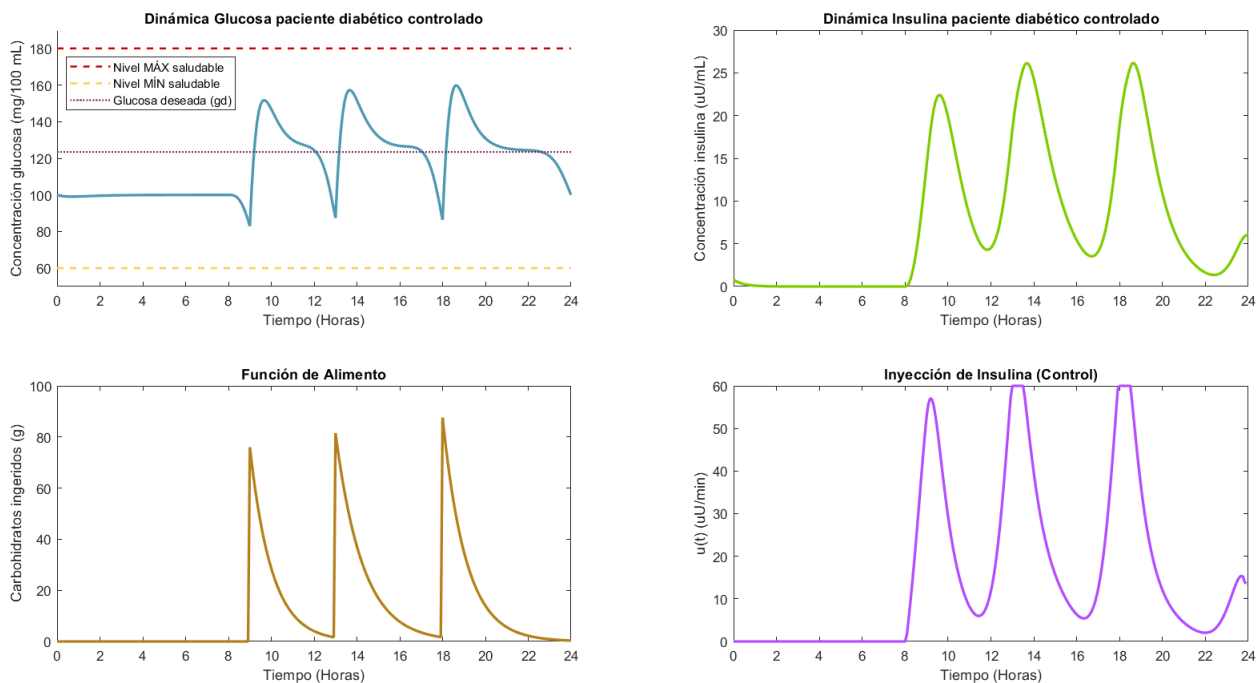


Figura 5.27: Gráficas de las dinámicas de glucosa e insulina, junto con su control óptimo para un paciente diabético tipo 1 que se alimenta según el escenario 3, con glucosa deseada según la opción 2.

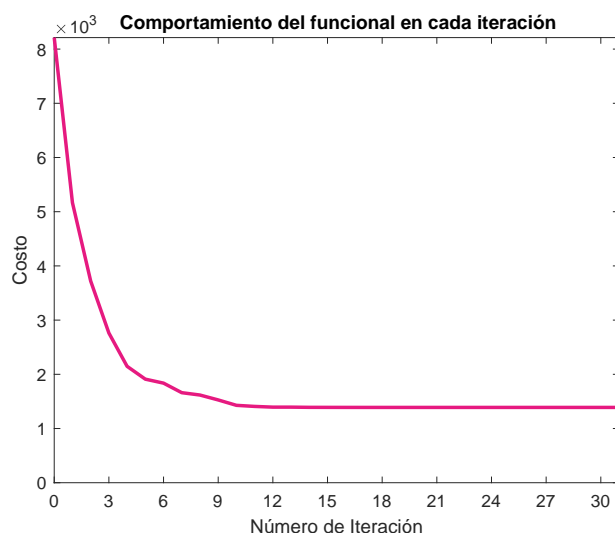


Figura 5.28: Funcional objetivo para glucosa deseada según forma 2, en cada iteración.
Nfe: 8971, *Nitr:* 31, *Time:* 35.4 s, *Fval:* 1.389×10^3 .

Paciente Diabético tipo 2

Parámetro	β Inicial (β_0)	β Estimado ($\hat{\beta}$)
a_1	0.0188	0.0530
a_2	0.9564	1.0256
a_3	0.0525	0.0032
a_4	3.4011	4.1008
b_1	0.0097	0.0123
b_2	1.8774	2.1497

Tabla 5.7: Resultados de la estimación de parámetros para el sistema (2.3) de un paciente diabético tipo 2 que se alimenta según el escenario 2.

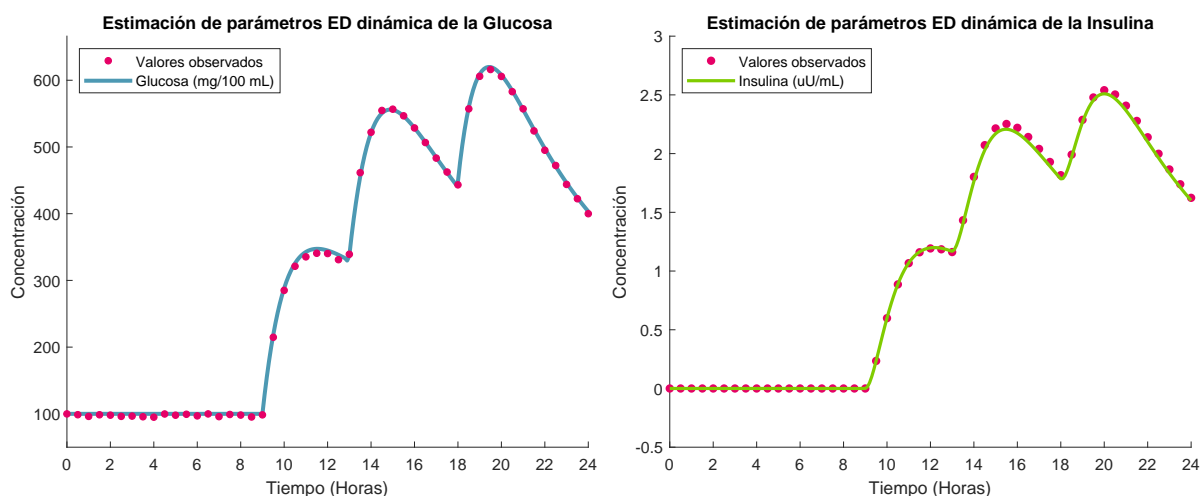


Figura 5.29: Conjunto de datos para la estimación de parámetros del modelo ajustado a la dinámica de un paciente diabético tipo 2.

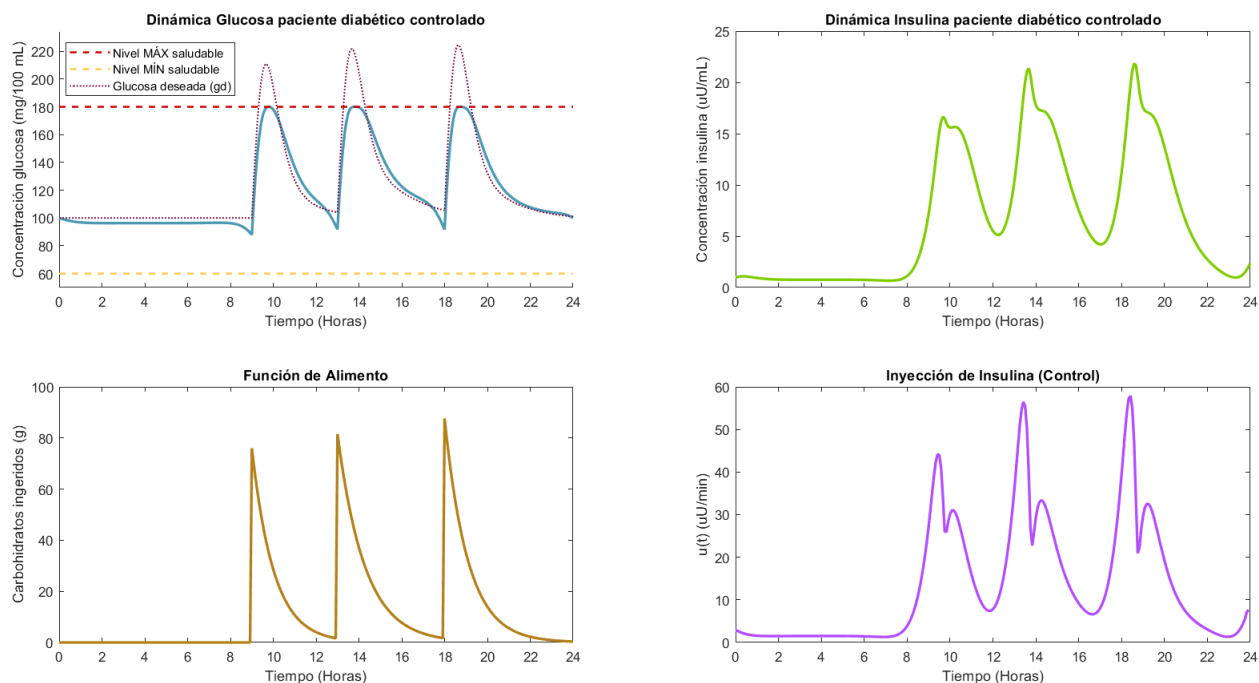


Figura 5.30: Gráficas de las dinámicas de glucosa e insulina, junto con su control óptimo para un paciente diabético tipo 2 que se alimenta según el escenario 3, con glucosa deseada según la opción 1.

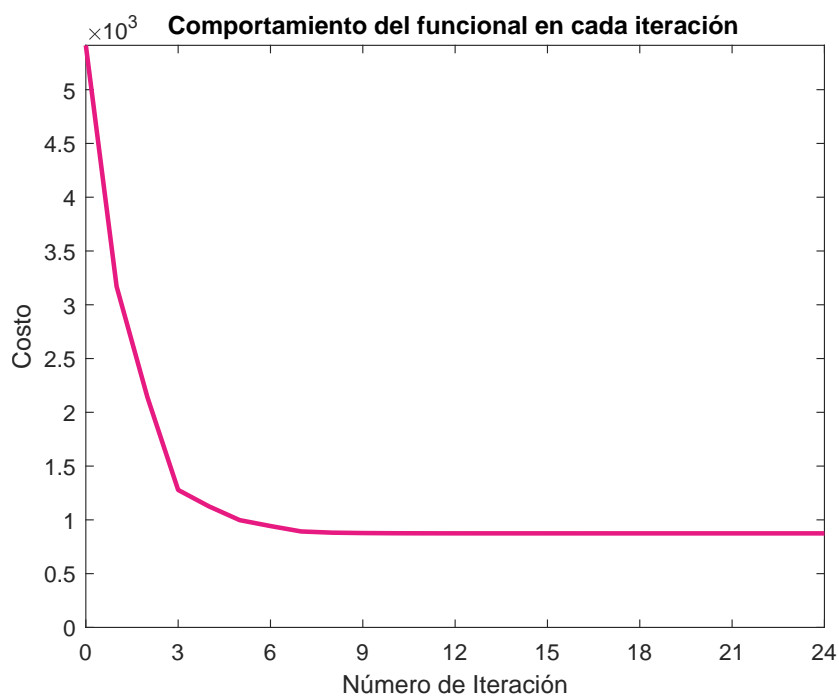


Figura 5.31: Funcional objetivo para glucosa deseada según forma 1, en cada iteración.

Nfe: 7321, *Nitr:* 24, *Time:* 38.6 s, *Fval:* 8.744×10^2 .

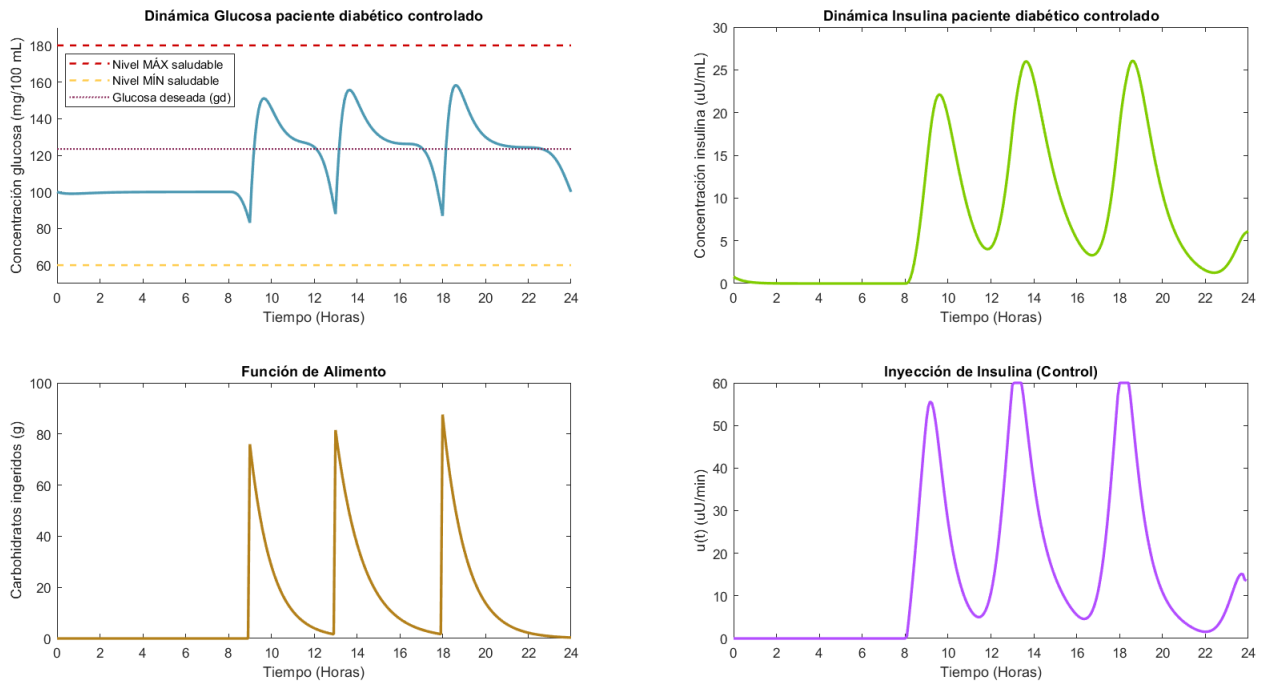


Figura 5.32: Gráficas de las dinámicas de glucosa e insulina, junto con su control óptimo para un paciente diabético tipo 2 que se alimenta según el escenario 3, con glucosa deseada según la opción 2.

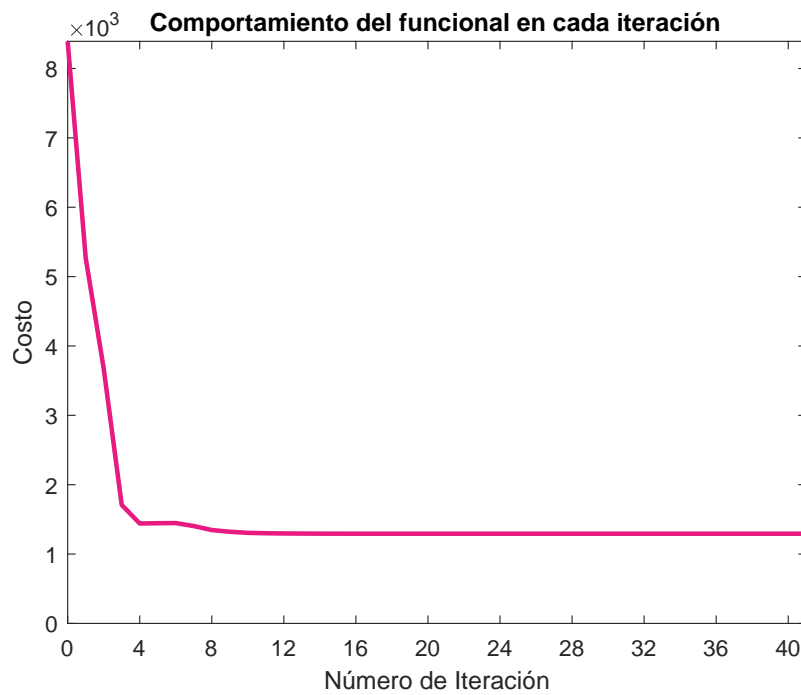


Figura 5.33: Funcional objetivo para glucosa deseada según forma 2, en cada iteración.

Nfe: 11895, ***Nitr:*** 41, ***Time:*** 40.2 s, ***Fval:*** 1.293×10^3 .

Nótese que en las figuras 5.24 y 5.29 los niveles de glucosa alcanzados sin el suministro de insulina casi alcanzan valores 900 mg/dL y 600 mg/dL para el paciente diabético tipo 1 y tipo 2 respectivamente, siendo estos los valores más altos registrados hasta el momento, situación que era de esperarse por el alto contenido de carbohidratos presentes en las denominadas comidas “chatarra” que componen el menú contemplado para este escenario en particular. Como se observa en la gráfica ubicada en la esquina superior izquierda de las figuras 5.25 y 5.30, una vez aplicado el control óptimo sobre la dinámica glucosa-insulina, se logra que los niveles de glucosa descieran y se mantengan dentro de los límites de normoglucemia, aunque estos puedan llegar a alcanzar valores de 180 mg/dL hora y media o dos horas después de comer. Sin embargo, la cantidad de insulina a administrar por la bomba es la más grande (alrededor de 60 $\mu\text{U}/\text{mL}$ para el paciente diabético tipo 1 y de 58.7 $\mu\text{U}/\text{mL}$ para el paciente diabético tipo 2) comparada con la cantidad administrada para los escenarios considerados anteriormente, siendo esta la suministrada para mantener dentro de los límites la cantidad de glucosa rezagada que es aportada por la comida ingerida durante el almuerzo y la que ingresa por el consumo de la cena.

Por su parte, en las gráficas de la esquina superior izquierda de las figuras 5.27 y 5.32, en las que se seleccionó la forma 2 de glucosa deseada, las concentraciones de glucosa de los pacientes se mantienen mucho más bajas, quedando alrededor de los 140-150 mg/dL, pero la cantidad de insulina que se requiere ha sido la más alta (llegando a los 58.7 $\mu\text{U}/\text{mL}$ y manteniendo esta cantidad por periodos de tiempo más largos), comparada no solo con las cantidades de insulina necesarias en este escenario al seleccionar la forma 1 de glucosa deseada, sino también con todos los demás escenarios analizados hasta el momento, como se aprecia en las gráficas ubicadas en la esquina inferior derecha de las figuras que se mencionaron anteriormente. Las simulaciones de este escenario registraron los costos de cómputo más altos hasta el momento, comparados con los de anteriores escenarios. Fueron necesarios mayores tiempos de cómputo, ejecutar más iteraciones y por supuesto más evaluaciones de la función objetivo, y aún así, los valores del funcional son los más grandes. Las gráficas de las figuras 5.26, 5.28, 5.31 y 5.28 presentan el comportamiento del funcional en cada iteración registrada, y la información sobre el número de iteraciones, tiempos y número de evaluaciones de la función objetivo para cada uno de los cuatro ensayos realizados en este escenario.

Es también importante mencionar, que el valor de insulina máximo que se presentó para este escenario, es menor que la cantidad de insulina exógena máxima suministrada que es reportada en [9], el cual oscila alrededor de 80 $\mu\text{U}/\text{mL}$, teniendo en cuenta que las funciones que modelan la ingesta de alimento y la cantidad de carbohidratos ingeridos en el único escenario considerado, son similares, lo que podría representar una ventaja del método presentado en este trabajo, con respecto a otros reportes realizados previamente en otras investigaciones.

5.3.4. Escenario 4

Para este último escenario, se simulará el estilo de alimentación de un paciente que, aunque las comidas que consume son bajas en carbohidratos, solamente consume dos comidas durante

todo el día, como suele ser el caso de personas que permanecen muy ocupadas en todo el día y no prestan la atención que merece a su alimentación. El menú que se considera, así como los resultados obtenidos de la simulación se presentan a continuación.

Desayuno (10:00 am)		Almuerzo (4:00 pm)	
1 taza de café sin azúcar	0 g	200 g de pescado al horno	0 g
2 tortillas de trigo integral	26 g	3 papas medianas	40 g
2 rebanadas de queso blanco	2 g	1½ taza de vegetales cocidos	0 g
1 pera	2 g	1 manzana	11 g
Gramos totales	30 g	Gramos totales	51 g

Paciente Diabético tipo 1

Parámetro	β Inicial (β_0)	β Estimado ($\hat{\beta}$)
a_1	0.0875	0.0682
a_2	1.5663	1.0154
a_3	0.0747	0.0139
a_4	4.1345	3.9271
b_1	0.00026	0
b_2	1.9307	2.0364

Tabla 5.8: Resultados de la estimación de parámetros para el sistema (2.3) de un paciente diabético tipo 1 que se alimenta según el escenario 4.

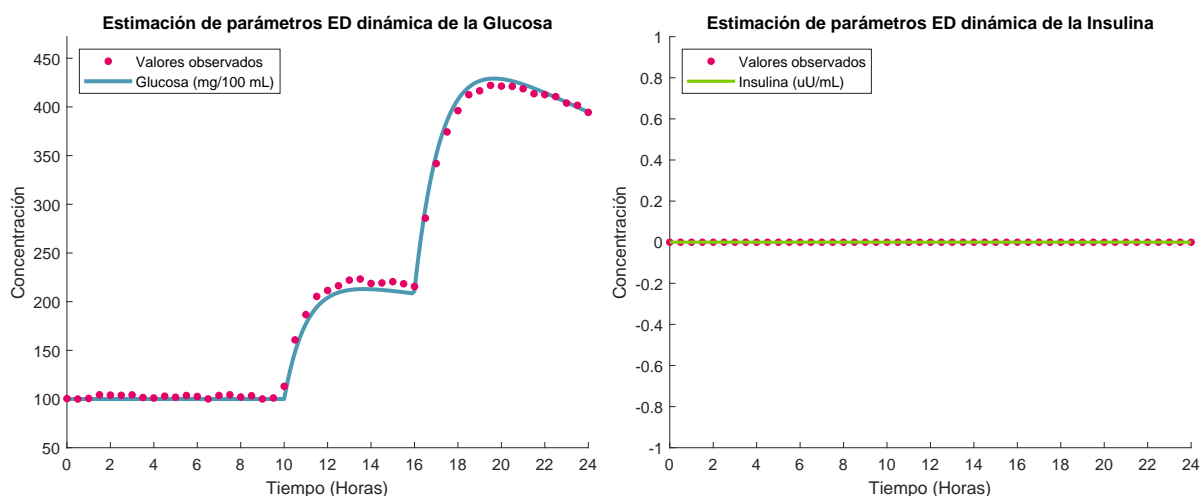


Figura 5.34: Conjunto de datos para la estimación de parámetros del modelo ajustado a la dinámica de un paciente diabético tipo 1.

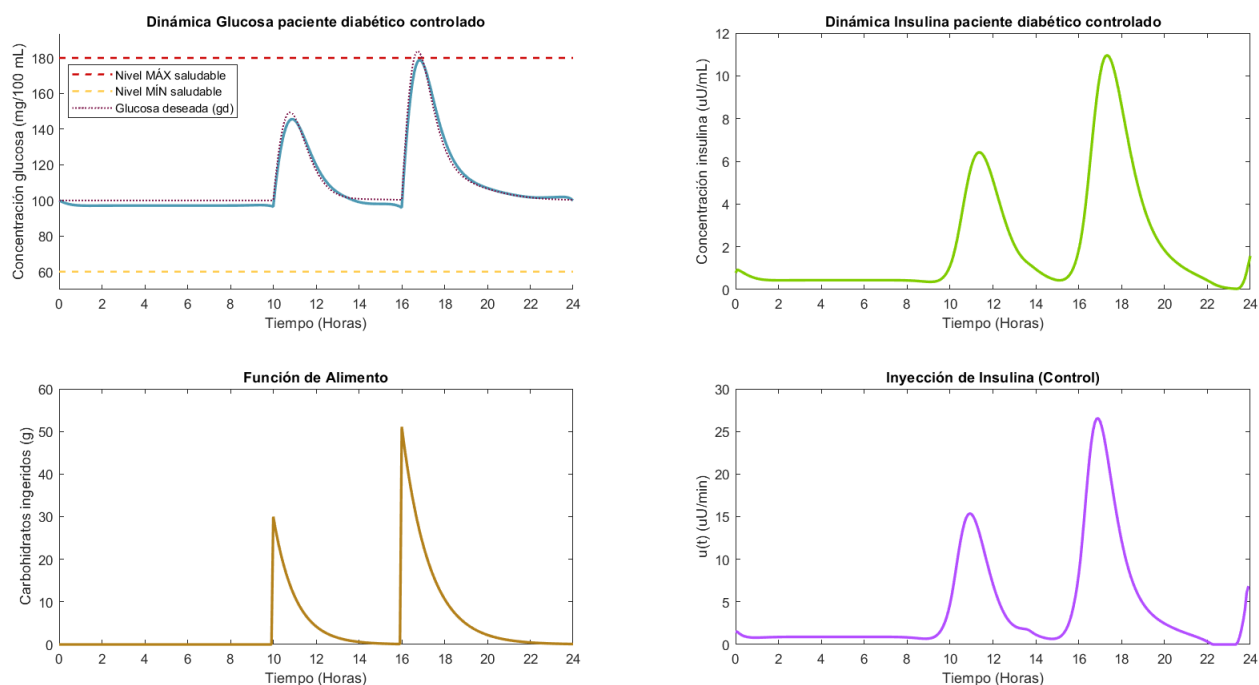


Figura 5.35: Gráficas de las dinámicas de glucosa e insulina, junto con su control óptimo para un paciente diabético tipo 1 que se alimenta según el escenario 4, con glucosa deseada según la opción 1.

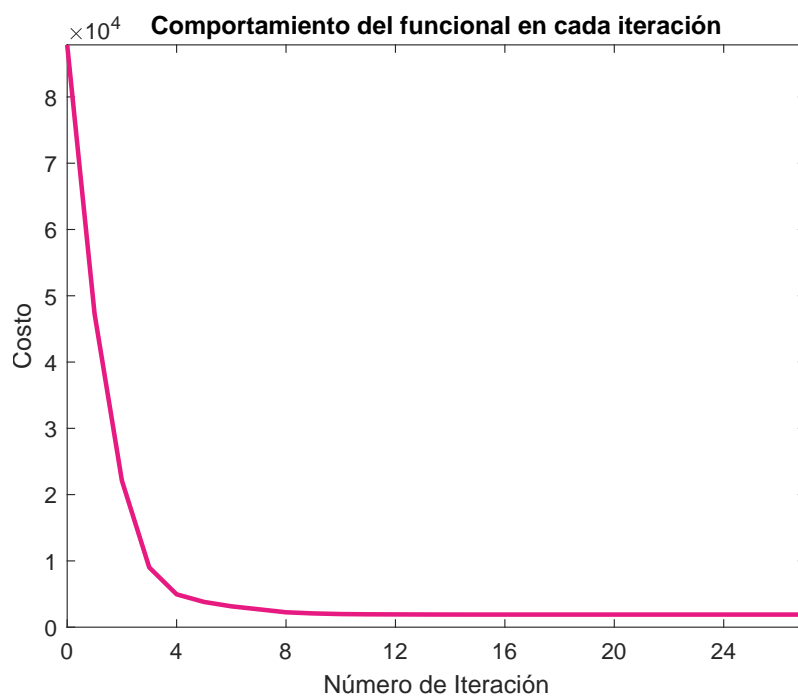


Figura 5.36: Funcional objetivo para glucosa deseada según forma 1, en cada iteración.

Nfe: 8104, *Nitr:* 27, *Time:* 25.2 s, *Fval:* 1.901×10^2 .

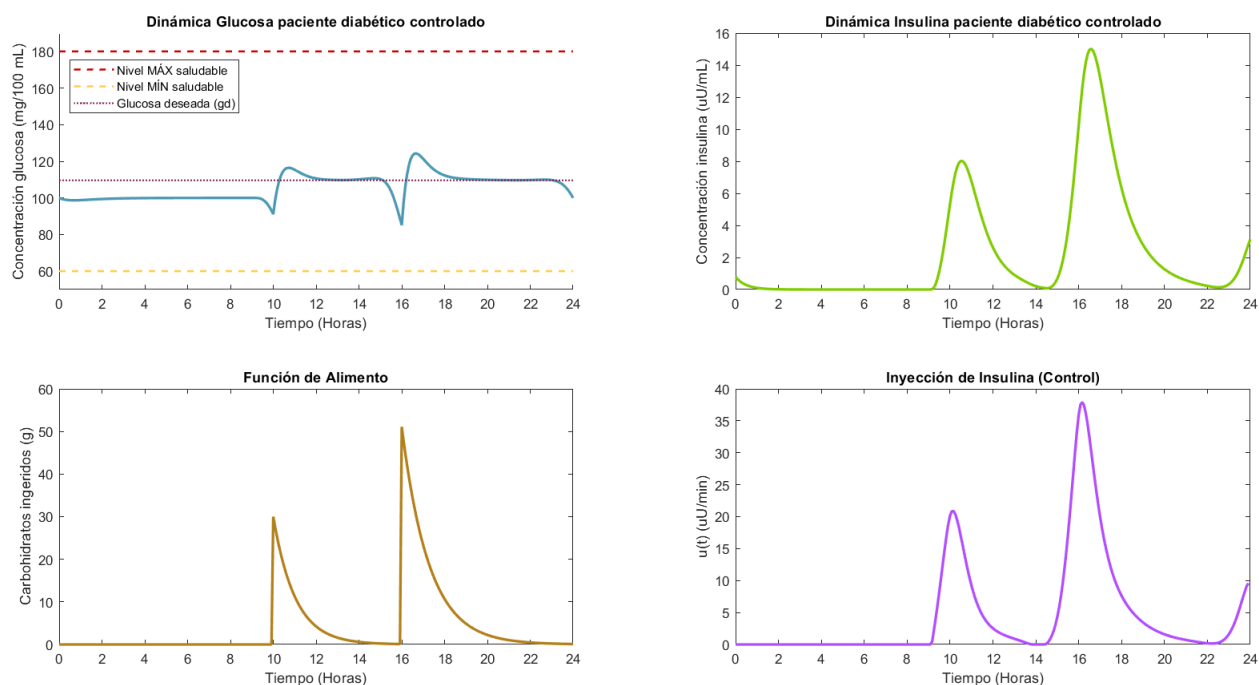


Figura 5.37: Gráficas de las dinámicas de glucosa e insulina, junto con su control óptimo para un paciente diabético tipo 1 que se alimenta según el escenario 4, con glucosa deseada según la opción 2.

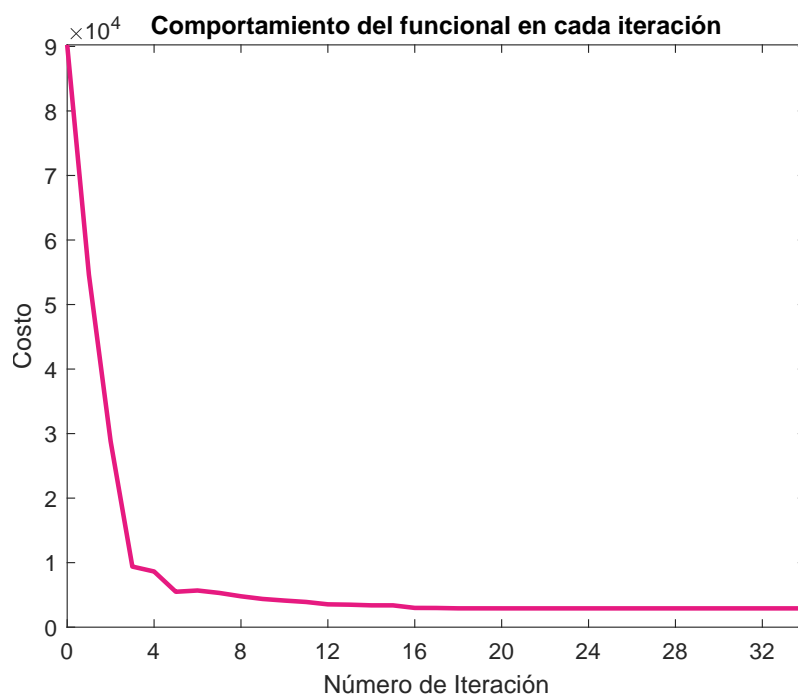


Figura 5.38: Funcional objetivo para glucosa deseada según forma 2, en cada iteración.

Nfe: 9855 ***Nitr:*** 34, ***Time:*** 30.8 s, ***Fval:*** 2.913×10^2 .

Paciente Diabético tipo 2

Parámetro	β Inicial (β_0)	β Estimado ($\hat{\beta}$)
a_1	0.0599	0.0577
a_2	1.8859	1.0032
a_3	0.0043	0.0120
a_4	4.3412	4.0251
b_1	0.0129	0.0101
b_2	2.5955	2.0197

Tabla 5.9: Resultados de la estimación de parámetros para el sistema (2.3) de un paciente diabético tipo 2 que se alimenta según el escenario 4.

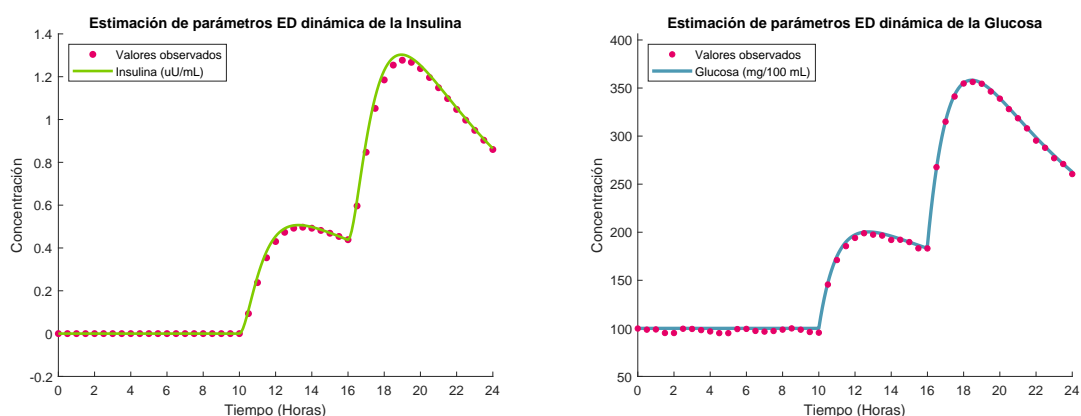


Figura 5.39: Conjunto de datos para la estimación de parámetros del modelo ajustado a la dinámica de un paciente diabético tipo 2.

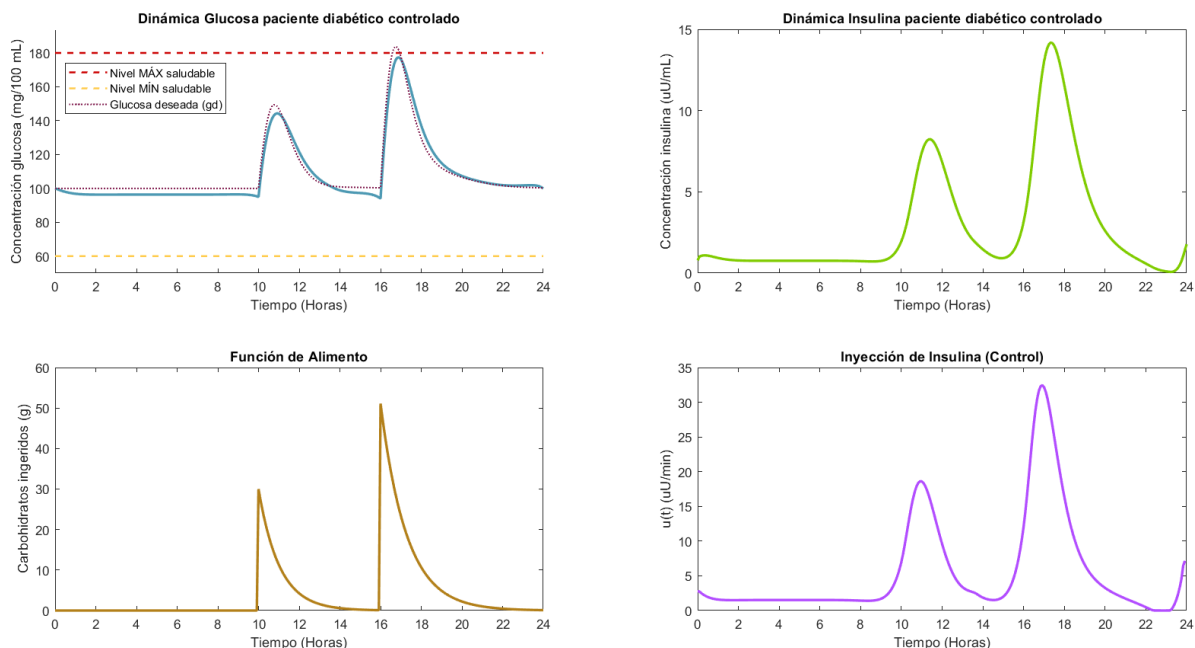


Figura 5.40: Gráficas de las dinámicas de glucosa e insulina, junto con su control óptimo para un paciente diabético tipo 2 que se alimenta según el escenario 4, con glucosa deseada según la opción 1.

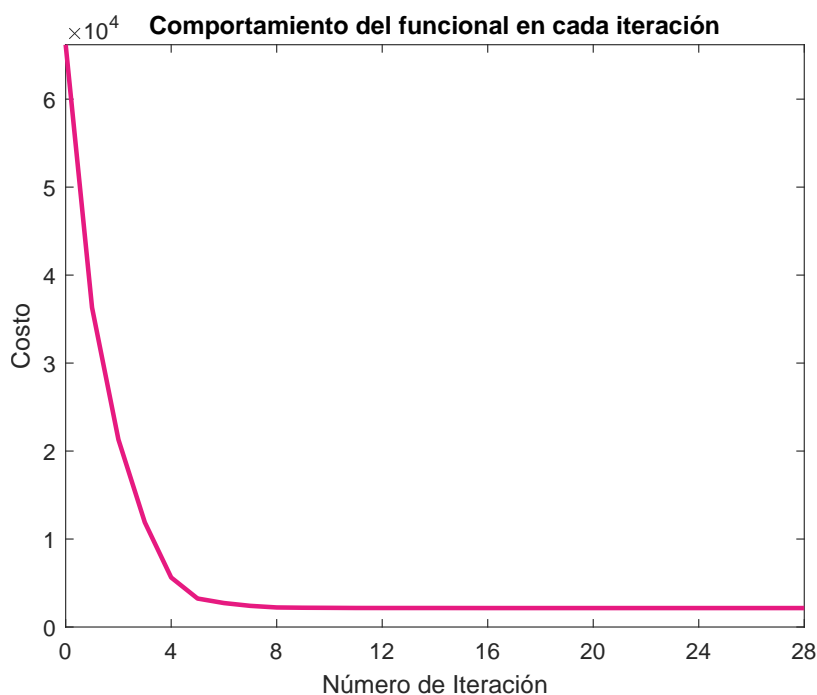


Figura 5.41: Funcional objetivo para glucosa deseada según forma 1, en cada iteración.
Nfe: 8441, *Nitr:* 28, *Time:* 25.1 s, *Fval:* 2.154×10^2 .

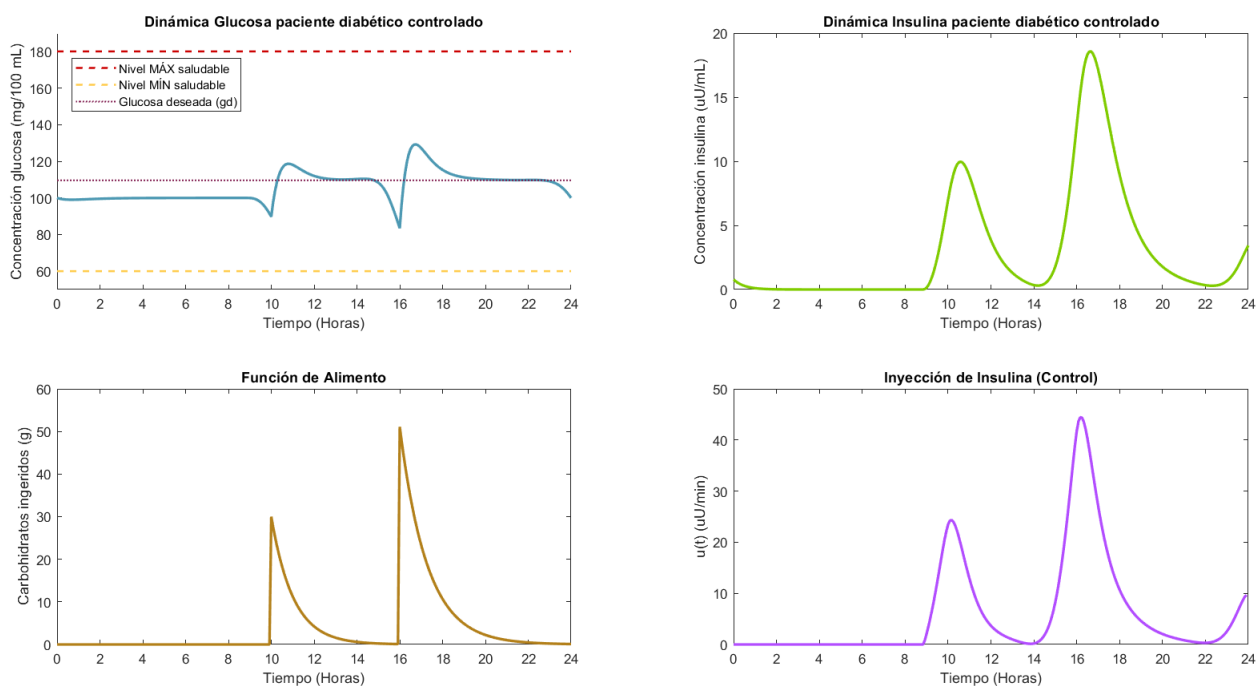


Figura 5.42: Gráficas de las dinámicas de glucosa e insulina, junto con su control óptimo para un paciente diabético tipo 2 que se alimenta según el escenario 4, con glucosa deseada según la opción 2.

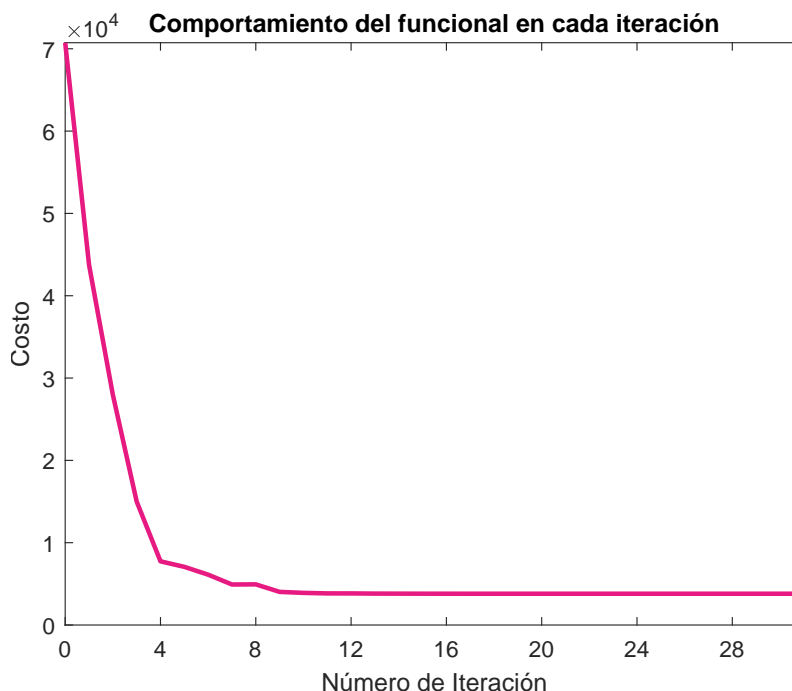


Figura 5.43: Funcional objetivo para glucosa deseada según forma 2, en cada iteración.
Nfe: 8977, *Nitr:* 31, *Time:* 32.2 s, *Fval:* 3.802×10^2 .

Debido a la mala alimentación de los pacientes considerados en este escenario, nótese que los niveles de glucosa alcanzados sin el suministro de insulina exógena son los más bajos comparados con las observaciones hechas a los escenarios anteriormente considerados, como se puede apreciar en las figuras 5.34 y 5.39. En las figuras 5.35 y 5.40, nótese también que la cantidad de insulina suministrada por el dispositivo debe mantenerse muy baja durante un periodo de tiempo más prolongado, para que el hígado pueda secretar las reservas de glucosa que conserva mientras la persona toma su primer alimento del día, ya que al existir una presencia elevada de concentración de insulina, el hígado lo detecta y por lo tanto inhibe el proceso de secreción de sus reservas, lo que puede provocar que la poca glucosa que ya circula por el torrente sanguíneo se consuman muy rápidamente y el paciente sufra de un episodio de hipoglucemia. De igual manera como se puede apreciar en las gráficas de las figuras ya mencionadas, el nivel de glucosa sigue una tendencia muy semejante a la marcada por la forma 1 de glucosa deseada, sin presentarse ningún peligro de hipoglucemia, suponiendo que la cantidad de glucosa que se conservan en las reservas del hígado alcanzan para proveer al organismo hasta que se lleve a cabo el proceso de ingesta de alimentos, sin embargo, esto puede que no sea así y aquí se tendría la necesidad de incluir en el modelo (2.3) un término que permita cuantificar la cantidad de glucosa que reserva el hígado y que puede ser utilizada cuando no hay ingreso de glucosa al organismo a través de los alimentos, debido a que esta seguramente no es infinita. El dispositivo solo secreta la cantidad de insulina estrictamente necesaria para consumir la glucosa aportada por el hígado, incluso llegando a suspenderse el suministro por un lapso de tiempo (entre las 22:00 y las 24:00), para garantizar que al final del día el nivel de glucosa esté en su valor basal (es

una restricción de igualdad establecida en (5.1)), situación que definitivamente no podría manejarse de ser el paciente quien suministre el medicamento a través de inyecciones, corriendo un riesgo aún mayor de aplicar una sobredosis y caer en un cuadro hipoglucémico, lo que pone de manifiesto en este escenario las dificultades que acarrea el tratamiento con inyecciones, tal como fue mencionado en los primeros apartados del presente capítulo.

En las gráficas ubicadas en la esquina superior izquierda de las figuras 5.37 y 5.42, debido a que la glucosa deseada seleccionada es la forma 2, las concentraciones de glucosa de los pacientes permanecen más bajas, por la cercanía de estas al valor constante que toma el término g_d . Se aprecia en las gráficas que estas concentraciones tienden a acomodarse a dicha función constante, entre las 12 y las 2pm y también entre las 7 y las 11 pm y solo se desvían de esta tendencia cuando se suministra la insulina necesaria 30 minutos antes de que la persona consuma el alimento. Aquí, al igual que en los escenarios anteriores, no hay presencia de insulina suministrada por el dispositivo en las primeras 9 horas del día, sin embargo, durante el resto del día se necesita del suministro de mayores cantidades de insulina para poder cumplir con la minimización de la distancia entre la concentración de glucosa del paciente y una glucosa deseada tan restrictiva como lo es la selección de un valor constante, situación que se repite aquí, tal cual como se ha venido presentando en los escenarios anteriores.

Con relación al tiempo de cómputo, la simulación de este escenario requirió un número de iteraciones y de evaluaciones de la función objetivo semejantes a los registrados en la simulación del escenario 3, a pesar de que en este cuarto escenario solo se consideraron dos comidas. Esta situación puede darse debido a que el control necesita calcularse teniendo en cuenta un estilo de alimentación tan limitado con la ingesta de glucosa, y además satisfacerse todas las restricciones impuestas sobre su formulación.

Finalmente, de los cuatros escenarios simulados, puede decirse que el de mejores resultados presentó, en cuanto a cantidad de insulina suministrada por el dispositivo (es decir, el control), niveles de glucosa en sangre más saludables, tiempo de cómputo y estilo de alimentación es el escenario número 2, con lo que se comprueba que un tratamiento efectivo y eficaz de la diabetes, sea tipo 1 o tipo 2, debe contemplar el uso de insulina exógena disponible para la reducción de la concentración de glucosa en sangre durante el día cuando sea necesaria, pero además debe estar acompañado por una dieta baja en consumo de carbohidratos y la ingesta de comidas debe darse varias veces al día. Esto permitirá que el paciente tenga una mejor calidad de vida, pueda tener la posibilidad de tener su enfermedad bajo control y así evite el desarrollo de otras enfermedades y complicaciones de salud que pueda desarrollar a futuro por una diabetes mal controlada.

6

Conclusiones y Trabajo futuro

En el presente trabajo se plantea un problema de control óptimo que consiste en encontrar la cantidad adecuada de insulina que debe ser suministrada a un paciente diabético para el tratamiento de la enfermedad, de tal manera que no se genere sobredosis ni escases del medicamento, los niveles de concentración de glucosa en sangre se mantengan siempre dentro de unos límites considerados como saludables y la cantidad de fármaco suministrado sea la más pequeña posible. Este problema de control está sujeto a un sistema de ecuaciones diferenciales ordinarias que modelan muy bien, y de una manera más sencilla, la dinámica glucosa-insulina en el cuerpo humano en comparación con modelos propuestos por otros autores. En este último capítulo se presentan las conclusiones al trabajo realizado, así como algunas consideraciones a tener en cuenta para posibles futuros trabajos.

6.1. Conclusiones

- ❑ Se logró la adaptación y aplicación de un método numérico para la estimación de los parámetros que componen el sistema de ecuaciones diferenciales con el que se describe la dinámica de la glucosa-insulina, cuyo desarrollo teórico se muestra en el capítulo 3, y se implementa de manera exitosa en la simulación de los diferentes escenarios considerados en el capítulo 5, con los cuales se pudo comprobar su eficacia y efectividad, además de que permite el trabajo con un modelo mucho más cercano a la realidad, en la medida en que estos parámetros son ajustados a las particularidades y condiciones de cada paciente. La estimación de los parámetros es un aspecto novedoso que se ha incorporado en el desarrollo de este trabajo, ya que, en los trabajos consultados como estado del arte, no se ha hecho un estudio similar y en la mayoría de estos, los parámetros han sido obtenidos de revisiones bibliográficas a otros trabajos e investigaciones, o a partir de técnicas experimentales, de las cuales no se mencionan mayores detalles sobre el proceso de su obtención.
- ❑ Se exploró y se probó el método de programación no lineal (PNL) como estrategia de solución al problema de control óptimo planteado, cumpliendo con todas las restricciones a las que fue sujeta la formulación del problema, de tal manera que fue posible mantener siempre los niveles de glucosa entre los límites deseados en todos los diferentes tipos de escenarios que tuvieron en cuenta durante el proceso de simulación, sin poner en riesgo, en ningún caso, al paciente de sufrir una hipo o hiperglucemia, tal como puede verse en los reportes que componen el capítulo 5. Estos resultados permiten concluir que el método propuesto es una alternativa efectiva, que podría ser

considerada frente a la que actualmente ya se encuentra disponible en el mercado [10] (algoritmo de control PID, usado en [47]) para el futuro desarrollo y construcción de bombas de administración de insulina cada vez más automáticas en las que se disminuya considerablemente la necesidad de la manipulación del dispositivo por parte del paciente, y así poder avanzar en el diseño y construcción de los llamados “páncreas artificiales”.

- Se realizaron experimentos con dos tipos de selección de glucosa deseada, donde al seleccionarla de acuerdo a la dinámica de glucosa de un paciente sano a lo largo del día, se obtienen resultados mucho más favorables (en cantidad de insulina a ser suministrada y costos computacionales) que cuando se selecciona como un valor constante, debido a que este último tiende a ser bastante restrictivo sobre la tendencia que debe seguir la concentración de glucosa del paciente, lo cual por supuesto exige mayores cantidades de insulina que deben ser suministradas para el cumplimiento del objetivo impuesto. Lo anterior sugiere entonces, que podría seleccionarse una glucosa deseada para cada estilo de alimentación del paciente, donde una similar a la forma 1 sería ideal para una persona rigurosa con sus horarios de comida y conteo de carbohidratos ya que siempre se podrá predecir la tendencia del comportamiento de sus niveles de glucosa dependiendo de la cantidad de carbohidratos que desea comer, mientras que la forma 2 de selección de glucosa deseada tal vez resulte mejor en pacientes que comen todos los días a diferentes horas del día y no son cuidadosos con el conteo de carbohidratos, puesto que como se evidenció en el capítulo anterior, el problema de control óptimo logra resolverse manteniendo siempre las concentraciones de glucosa cercanas al valor constante que se tome como glucosa deseada, sin importar el tipo de comida ni la hora en que se consuma y sin poner el riesgo al paciente de llegar a estar fuera de los límites de glucosa permitidos, pero por supuesto, esto requiere que la cantidad de insulina a suministrar sea más alta.
- Es importante en todo caso, que los pacientes diabéticos mantengan un estilo de vida saludable, que incluya el consumo de alimentos bajos en carbohidratos, puesto que son los responsables en el aumento de los niveles de glucosa en sangre debido a su consumo excesivo. Seguir las dietas y recomendaciones dadas por los profesionales de la salud, conllevará a que la diabetes sea mejor tratada y la cantidad de medicamento que el paciente deba consumir se reduzca considerablemente, lo que representaría un beneficio económico en la adquisición y compra del fármaco por parte de los sistemas de salud.

6.2. Trabajo futuro

- Aunque el tiempo de ejecución y el consumo de recursos computacionales en la propuesta aquí presentada es baja, la solución al problema de control óptimo también podría ser buscada cambiando el método de PNL por el del principio del mínimo de Pontryagin, con el fin de poder comparar sus resultados con los encontrados en este

trabajo y así obtener más alternativas de posibles estrategias que puedan ser utilizadas el desarrollo de nuevos dispositivos para el tratamiento de la diabetes.

- Además de llevar estilos de alimentación saludable, muchos pacientes combinan sus tratamientos con rutinas de ejercicio físico, y en muchos casos, algunos pacientes diagnosticados con diabetes tipo 2 no llegan a necesitar del suministro exógeno de insulina, y su tratamiento consiste solo en estos dos aspectos mencionados inicialmente. Resultaría muy positivo considerar, en futuros desarrollos, la ampliación del modelo matemático presentado en (2.3) de tal manera que en él se incorporen la actividad física que pueda ejecutar el paciente como parte del tratamiento a su enfermedad, y algunos otros factores como su estado de ánimo, de tal manera que el modelo resulte ser mucho más versátil, y pueda ser utilizado en pacientes con más variedad de estilos de vida. Estos aspectos que podrían ser tenidos en cuenta para la predicción de un perfil glucémico mucho más ajustado a la realidad del paciente. También podría considerarse la posibilidad de incorporar al modelo algún término que permita cuantificar de manera más fidedigna la cantidad de glucosa que es almacenada por el hígado, para así poder simular de mejor manera escenarios donde los pacientes no consuman más de dos comidas al día, como se ha considerado en algunos otros modelos elaborados para la descripción matemática de la dinámica glucosa-insulina, y que fueron presentados en el capítulo 1 del presente trabajo.
- Pueden realizarse más experimentos con otros tipos de glucosa deseada diferentes a los trabajados en los resultados numéricos aquí presentados, con el fin de tener más alternativas, y así seleccionar aquella que permita el cumplimiento de las restricciones impuestas sobre la dinámica de la glucosa, pero también que implique una menor cantidad a suministrar de insulina. Al disponer de variadas alternativas, podría ser posible seleccionar una que se adecúe de la mejor manera al perfil glucémico de cada paciente.
- Si bien, en este trabajo se logró la simulación de los diferentes escenarios considerados en un periodo de tiempo de un día (24 horas), que resulta ser mucho más extenso que la mayoría de periodos de tiempo considerados en los trabajos anteriores a este y que fueron consultados para la construcción del estado del arte (donde solo se consideran periodos de 200 a 400 minutos), es necesario también llegar a considerar periodos de tiempo más extensos (2, 3 o más días), de tal manera que sea posible apreciar con más detalle el comportamiento de la dinámica glucosa-insulina y así ajustar más a la realidad tanto el modelo matemático, como sus parámetros y por supuesto el perfil glucémico particular de cada uno de los pacientes.

Bibliografía

- [1] E. ACKERMAN, L. C. GATEWOOD, J. W. ROSEVEAR, AND G. D. MOLNAR, *Model studies of blood-glucose regulation*, Bulletin of Mathematical Biology, 27 (1965), pp. 21–37.
- [2] J. ALAVEZ-RAMÍREZ, *Estimación de Parámetros en Ecuaciones Diferenciales Ordinarias: Identificabilidad y Aplicaciones a Medicina*, PhD thesis, National University of Mexico (UNAM), 2007.
- [3] I. ALONSO CISNEROS, *Modelos matemáticos para la diabetes*, tech. rep., Universidad de Cantabria, 2014.
- [4] I. ALONSO CISNEROS, *Técnicas de control en modelos matemáticos para la diabetes*, Master’s thesis, Universidad de Cantabria, 2015.
- [5] AMERICAN DIABETES ASSOCIATION, *Diagnosis and Classification of Diabetes Mellitus*, Diabetes Care, 37 (2013), pp. 81–90.
- [6] D. ARBELÁEZ, *Modelo matemático para el transporte de toxinas en una red trófica marina.*, Master’s thesis, Universidad Nacional de Colombia, 2011.
- [7] V. BARBU, *Mathematical methods in optimization of differential systems*, vol. 310, Springer Science & Business Media, 2012.
- [8] A. BARCLAY, P. E. GILL, AND J. B. ROSEN, *SQP methods and their application to numerical optimal control*, in Variational calculus, optimal control and applications, Springer, 1998, pp. 207–222.
- [9] F. BARRAGÁN, *Técnicas de control robusto h_2/h_∞ para sistemas controlados a través de redes de comunicación*, Master’s thesis, Universidad de Sevilla, 2012.
- [10] P. BEATO, *Sistemas de páncreas artificial disponibles en el mercado español: ¿qué aportan al paciente?*, Revista Diabetes, (2019).
- [11] R. BERGMAN, Y. IDER, C. BOWDEN, AND C. COBELLI, *Quantitative estimation of insulin sensitivity.*, American Journal of Physiology-Endocrinology And Metabolism, 236 (1979), p. E667.
- [12] F. BIRAL, E. BERTOLAZZI, AND P. BOSETTI, *Notes on numerical methods for solving optimal control problems*, IEEJ Journal of Industry Applications, 5 (2016), pp. 154–166.

- [13] S. BOYD AND L. VANDENBERGHE, *Convex optimization*, Cambridge University Press, 2004.
- [14] S. BUPPAJARNTHAM, P. JUNPAPARP, R. SALAMEH, AND C. ANASTASOPOULOU, *Insulin*. <https://emedicine.medscape.com/article/2089224-overview>, Dec. 2021.
- [15] R. CERVANTES-VILLAGRANA AND J. PRESNO-BELTRÁN, *Fisiopatología de la diabetes y los mecanismos de muerte de las células beta pancreáticas*, *Revista de Endocrinología y Nutrición*, 21 (2013), pp. 98–106.
- [16] J. DENNIS AND R. SCHNABEL, *Numerical methods for unconstrained optimization and nonlinear equations*, vol. 16, Siam, 1996.
- [17] M. DIEHL, H. G. BOCK, H. DIEDAM, AND P.-B. WIEBER, *Fast direct multiple shooting algorithms for optimal robot control*, in *Fast motions in biomechanics and robotics*, Springer, 2006, pp. 65–93.
- [18] R. C. DORF AND R. BISHOP, *Sistemas de control moderno*, Pearson Prentice Hall, 2005.
- [19] FATSECRET CHILE, *Calculadora conteo de calorías y carbohidratos por porción de alimento*. <https://www.fatsecret.cl/calorias-nutricion/>, May 2021.
- [20] FEDERACIÓN INTERNACIONAL DE DIABETES (FID) AND OTHERS, *Atlas de la DIABETES de la FID*, tech. rep., IDF, 2015.
- [21] M. Á. GARCÍA-FERRERO, *Cómo usar matemáticas para desarrollar un páncreas artificial*, *El País*, (2020).
- [22] H. GARVIN, *The Levenberg-Marquardt algorithm for nonlinear least squares curve-fitting problems.*, tech. rep., Duke University, 2019.
- [23] L. HEINEMANN, *Insulin assay standardization: leading to measures of insulin sensitivity and secretion for practical clinical care.*, *Diabetes Care*, 33 (2010), p. 83.
- [24] M. HUANG, J. LI, X. SONG, AND H. GUO, *Modeling impulsive injections of insulin: towards artificial pancreas*, *SIAM Journal on Applied Mathematics*, 72 (2012), pp. 1524–1548.
- [25] A. IBAÑEZ, *Optimal control and turnpike properties of the Lotka Volterra model*, Master’s thesis, Universidad del País Vasco, 2013.
- [26] R. KALABA AND K. SPINGARN, *Control, identification, and input optimization*, Springer Science & Business Media, 2012.
- [27] J. N. KAPUR, *Mathematical modelling*, New Age International, 1988.

- [28] J. KNOPP, L. HOLDER-PEARSON, AND J. G. CHASE, *Insulin units and conversion factors: A story of truth, boots, and faster half-truths*, Journal of diabetes science and technology, 13 (2019), pp. 597–600.
- [29] A. KOZMA, *Distributed Optimization Methods for Large Scale Optimal Control*, PhD thesis, KU Leuven, 2013.
- [30] W. KRABS, *Modelling, analysis and optimization of biosystems*, Springer Science & Business Media, 2007.
- [31] R. LAHOZ-BELTRA, *Las matemáticas de la vida: Modelos numéricos para la biología y la ecología*, RBA Coleccionables S.A, 2010.
- [32] MEDTRONIC MINMED INC, *Información básica sobre el tratamiento con bomba de insulina*, 2015.
- [33] R. MENEU, *La teoría del control óptimo en tiempo discreto. Modelos financieros aplicados a las pensiones de jubilación.*, PhD thesis, Universidad de Valencia, 1995.
- [34] V. MORALES, *Ampliación de un modelo matemático incluyendo la actividad física para la predicción de los niveles de glucosa en sangre de forma continua en pacientes diabéticos.*, tech. rep., Universidad Politécnica de Cartagena, 2017.
- [35] MYFITNESSPAL, *Calculadora conteo de calorías y carbohidratos por porción de alimento*. <https://www.myfitnesspal.com/es/food/search>, May 2021.
- [36] NATIONAL INSTITUTE OF DIABETES AND DIGESTIVE AND KIDNEY DISEASES, *Low blood glucose (Hypoglycemia)*. <https://www.niddk.nih.gov/health-information/diabetes/overview/preventing-problems/low-blood-glucose-hypoglycemia>, July 2021.
- [37] J. NOCEDAL AND S. WRIGHT, *Numerical optimization*, Springer Science & Business Media, 2006.
- [38] ORGANIZACIÓN MUNDIAL DE LA SALUD, *Informe mundial sobre la diabetes: Resumen de orientación*. <http://www.who.int/diabetes/global-report>, 2016.
- [39] P. PEDREGAL, *Introduction to Optimization*, vol. 46, Springer Science & Business Media, 2006.
- [40] RP OPTIMIZATION RESEARCH, LLC, *GPOPS-II-Next-Generation Optimal Control Software*. <http://https://www.gpops2.com/index.html>, 2015.
- [41] K. SYDSÆTER, P. HAMMOND, A. SEIERSTAD, AND A. STROM, *Further mathematics for economic analysis*, Pearson Education, 2008.
- [42] THE MATHWORKS, INC., *Optimization Toolbox User's Guide*, 2020.

- [43] A. TOCANCIPA, *Teoría del control*, Boletín de Matemáticas, 10 (1976), pp. 28–47.
- [44] N. TOUCHETTE, *Complete guide to diabetes*, American Diabetes Association, fourth ed., 2005.
- [45] UNIVERSITY OF CALIFORNIA, *Types of Insulin*. Diabetes Education Online. URL: <https://dtc.ucsf.edu/types-of-diabetes/type2/treatment-of-type-2-diabetes/medications-and-therapies/type-2-insulin-rx/types-of-insulin/>.
- [46] H. VARGAS-URICOECHEA AND L. Á. CASAS-FIGUEROA, *Epidemiología de la diabetes mellitus en Sudamérica: la experiencia de Colombia*, Clínica e Investigación en Arteriosclerosis, 28 (2016), pp. 245–256.
- [47] S. VELÁSQUEZ, R. VELÁSQUEZ, M. LEYTON, J. BORJAS, AND Á. CUSTODIO, *Modelado del control de la regulación de glucosa*, Universidad, Ciencia y Tecnología, 17 (2013).
- [48] E. WIDMAIER, H. RAFF, AND K. STRANG, *Human Physiology: The mechanisms of body function*, Mc Graw Hill, fifteenth ed., 2018.

A

Método de Runge-Kutta de cuarto orden

El método de Runge-Kutta es una técnica usada para la aproximación numérica a la solución de ecuaciones diferenciales, desarrollada alrededor del año 1900 por los matemáticos *Carl Runge* y *Martín Kutta*. La idea del método es mejorar la aproximación de la solución (suponga que sea $y(t_{i+1})$), usando información que se tiene en el intervalo $[t_i, t_{i+1}]$. El método se puede deducir como se muestra a continuación:

Considere el problema de valor inicial

$$\begin{cases} y' = f(t, y(t)) \\ y(t_0) = y_0. \end{cases} \quad (\text{A.1})$$

Integrando la primera igualdad de (A.1) en el intervalo $[t_i, t_{i+1}]$

$$\int_{t_i}^{t_{i+1}} y'(t) dt = \int_{t_i}^{t_{i+1}} f(t, y(t)) dt. \quad (\text{A.2})$$

Por TFC aplicado a la integral del lado izquierdo de la ecuación (A.2)

$$y(t_{i+1}) - y(t_i) = \int_{t_i}^{t_{i+1}} f(t, y(t)) dt, \quad (\text{A.3})$$

aplicando la regla de integración de Simpson al lado derecho de la igualdad (A.2)

$$\int_{t_i}^{t_{i+1}} f(t, y(t)) dt \approx \frac{t_{i+1} - t_i}{6} \left[f(t_i, y(t_i)) + 4f\left(\frac{t_i + t_{i+1}}{2}, y\left(\frac{t_i + t_{i+1}}{2}\right)\right) + f(t_{i+1}, y(t_{i+1})) \right]. \quad (\text{A.4})$$

Sustituyendo $\Delta t = t_{i+1} - t_i$, $y(t_i) = y_i$ y $y(t_{i+1}) = y_{i+1}$ en (A.4)

$$\int_{t_i}^{t_{i+1}} f(t, y(t)) dt \approx \frac{\Delta t}{6} \left[f(t_i, y_i) + 4f\left(t_i + \frac{\Delta t}{2}, y\left(t_i + \frac{\Delta t}{2}\right)\right) + f(t_{i+1}, y_{i+1}) \right].$$

$$= \frac{\Delta t}{6} \left[f(t_i, y_i) + 2f\left(t_i + \frac{\Delta t}{2}, y\left(t_i + \frac{\Delta t}{2}\right)\right) + 2f\left(t_i + \frac{\Delta t}{2}, y\left(t_i + \frac{\Delta t}{2}\right)\right) + f(t_{i+1}, y_{i+1}) \right]. \quad (\text{A.5})$$

Como y_{i+1} y $y\left(t_i + \frac{\Delta t}{2}\right)$ son términos desconocidos, se hace necesario aproximarlos por el método de Euler de la siguiente manera:

$$y_{i+1} \approx y_i + \Delta t f(t_i, y_i),$$

$$y_{i+\frac{1}{2}} \approx y_i + \frac{\Delta t}{2} f(t_i, y_i).$$

Reemplazando estas igualdades en (A.4)

$$\approx \frac{\Delta t}{6} \left[f(t_i, y_i) + 2f\left(t_i + \frac{\Delta t}{2}, y_i + \frac{\Delta t}{2} f(t_i, y_i)\right) + 2f\left(t_i + \frac{\Delta t}{2}, y\left(t_i + \frac{\Delta t}{2}\right)\right) + f(t_{i+1}, y_i + \Delta t f(t_i, y_i)) \right]. \quad (\text{A.6})$$

Sea $k_1 = \Delta t f(t_i, y_i)$, entonces

$$\approx \frac{1}{6} \left[k_1 + 2\Delta t f\left(t_i + \frac{\Delta t}{2}, y_i + \frac{k_1}{2}\right) + 2\Delta t f\left(t_i + \frac{\Delta t}{2}, y\left(t_i + \frac{\Delta t}{2}\right)\right) + \Delta t f(t_{i+1}, y_i + k_1) \right]. \quad (\text{A.7})$$

Sea $k_2 = \Delta t f\left(t_i + \frac{\Delta t}{2}, y_i + \frac{k_1}{2}\right)$ y sustituyendo en (A.7)

$$\int_{t_i}^{t_{i+1}} f(t, y(t)) dt \approx \frac{1}{6} \left[k_1 + 2k_2 + 2\Delta t f\left(t_i + \frac{\Delta t}{2}, y\left(t_i + \frac{\Delta t}{2}\right)\right) + \Delta t f(t_{i+1}, y_i + k_1) \right]. \quad (\text{A.8})$$

Ahora, para aproximar la segunda evaluación del punto medio faltante de la ecuación (A.8), se construye nuevamente una aproximación por el método de Euler, pero esta vez se utiliza la aproximación encontrada en la primera evaluación del punto medio. Por lo tanto:

$$y_{i+\frac{1}{2}} \approx y_i + \frac{\Delta t}{2} f\left(t_i + \frac{\Delta t}{2}, y_i + \frac{\Delta t}{2} f(t_i, y_i)\right),$$

$$y_{i+\frac{1}{2}} \approx y_i + \frac{\Delta t}{2} f\left(t_i + \frac{\Delta t}{2}, y_i + \frac{k_1}{2}\right), \quad (\text{A.9})$$

$$y_{i+\frac{1}{2}} = y_i + \frac{k_2}{2}.$$

Reemplazando esta última igualdad en la ecuación (A.8):

$$\int_{t_i}^{t_{i+1}} f(t, y(t)) dt \approx \frac{1}{6} \left[k_1 + 2k_2 + 2\Delta t f \left(t_i + \frac{\Delta t}{2}, y_i + \frac{k_2}{2} \right) + \Delta t f(t_{i+1}, y_i + k_1) \right]. \quad (\text{A.10})$$

Sustituyendo $k_3 = \Delta t f \left(t_i + \frac{\Delta t}{2}, y_i + \frac{k_2}{2} \right)$ en (A.10):

$$\int_{t_i}^{t_{i+1}} f(t, y(t)) dt \approx \frac{1}{6} \left[k_1 + 2k_2 + 2k_3 + \Delta t f(t_{i+1}, y_i + k_1) \right]. \quad (\text{A.11})$$

Finalmente, sustituyendo $k_4 = \Delta t f(t_{i+1}, y_i + k_1)$ en (A.11):

$$\int_{t_i}^{t_{i+1}} f(t, y(t)) dt \approx \frac{1}{6} \left[k_1 + 2k_2 + 2k_3 + k_4 \right].$$

Se puede resumir entonces el método de cuarto orden, como la solución iterativa de las ecuaciones:

$$\begin{aligned} k_1 &= \Delta t f(t_i, y_i) \\ k_2 &= \Delta t f \left(t_i + \frac{\Delta t}{2}, y_i + \frac{k_1}{2} \right) \\ k_3 &= \Delta t f \left(t_i + \frac{\Delta t}{2}, y_i + \frac{k_2}{2} \right) \\ k_4 &= \Delta t f(t_i + \Delta t, y_i + k_3) \\ y_{i+1} &= y_i + \frac{1}{6} (k_1 + 2k_2 + 2k_3 + k_4). \end{aligned} \quad (\text{A.12})$$

Un análisis sobre los errores local y global de este método, mostrará su proporcionalidad a $\mathcal{O}(\Delta t^4)$ y $\mathcal{O}(\Delta t^5)$ respectivamente.

B

Códigos de programación: métodos numéricos y simulaciones

Este apéndice contiene los códigos de los métodos numéricos usados y de las simulaciones desarrolladas en los capítulos 2, 3, y 5 del presente trabajo.

Los comentarios colocados en los códigos indican el método numérico o simulación a la que corresponden, así como los datos de entrada y de salida de cada uno, haciendo además referencia a alguno de los pseudocódigos o diagramas de flujo de los algoritmos introducidos a lo largo de cada capítulo mencionado anteriormente, con el fin hacer evidente al lector la forma en la que han sido programados para su implementación computacional.

B.1. Códigos del Capítulo 2

Archivo: rk4.m

```
function [t,x,y] = rk4(f1,f2,tspan,x0,y0,h)

%RK4 Integra un sistema de ecuaciones diferenciales ordinarias usando
% el método de Runge–Kutta de cuarto orden mostrado en el Algoritmo 1.
%
% [t,g,h] = rk4(gprima,hprima,tspan,g0,h0,N) Integra el sistema de
% ecuaciones diferenciales compuesto de las funciones gprima y hprima en el
% intervalo tspan = [t0, tfinal] y usando las condiciones iniciales x0, y0,
% donde 'gprima' y 'hprima' son las funciones que componen el sistema
% glucosa–insulina.
%
% ENTRADAS:
% f1    – función derivada de la glucosa con respecto al tiempo. f1 = dg/dt.
% f2    – función derivada de la insulina con respecto al tiempo. f2 = dh/dt.
% tspan = [t0,tfinal], donde t0 es el valor inicial de t, y tfinal, es el
%         valor final de t.
% x0,y0 – Condiciones iniciales.
% h     – Tamaño del paso de integración.

% SALIDAS:
% t     – Retorna los puntos del tiempo usados para la integración.
% x,y   – Retorna vectores columna con las soluciones al sistema glucosa–
```

```

%      insulina.

% Autor: Edwar Fabián Panqueba Moreno
% Fecha: 04/06/2021.
% Inicio del algoritmo

t0 = tspan(1);
tfinal = tspan(2);
N = tfinal/h;
t = t0:h:tfinal;

x = zeros(N+1,1);
y = zeros(N+1,1);
x(1) = x0; y(1) = y0;

for i = 1:N
    k1 = h*f1(t(i),x(i),y(i));
    l1 = h*f2(t(i),x(i),y(i));
    k2 = h*f1(t(i)+h/2,x(i)+k1/2,y(i)+l1/2);
    l2 = h*f2(t(i)+h/2,x(i)+k1/2,y(i)+l1/2);
    k3 = h*f1(t(i)+h/2,x(i)+k2/2,y(i)+l2/2);
    l3 = h*f2(t(i)+h/2,x(i)+k2/2,y(i)+l2/2);
    k4 = h*f1(t(i)+h,x(i)+k3,y(i)+l3);
    l4 = h*f2(t(i)+h,x(i)+k3,y(i)+l3);

    x(i+1) = x(i)+(k1+2*k2+2*k3+k4)/6;
    y(i+1) = y(i)+(l1+2*l2+2*l3+l4)/6;
end

```

Archivo: Sol_SistGH.m

```

function [t,g,h,eat] = Sol_SistGH(p,t0,tfinal,g0,h0,ssize)

% Sol_SistGH soluciona el sistema de ecuaciones diferenciales que describen
% la dinámica de la glucosa–insulina definido en el capítulo 2 a través de
% la función RK4 que contiene la programación del método Runge–Kutta de
% cuarto orden.

% ENTRADAS:
% p      – Vector de parámetros del sistema de ecuaciones.
% t0     – Valor inicial de t.
% tfinal – Valor final de t.
% g0,h0  – Condiciones iniciales del sistema de ecuaciones.
% ssize  – Tamaño de paso de integración para el método RK cuarto orden.

% SALIDAS:
% t      – Retorna los puntos del tiempo usados para la integración.
% g,h    – Retorna vectores columna con las soluciones al sistema glucosa–
%         insulina.

```

```

% Los resultados pueden ser graficados plot(t,g)

% Autor: Edwar Fabián Panqueba Moreno
% Fecha: 04/06/2021.

% Parámetros del sistema de ecuaciones.
a1 = p(1); a2 = p(2); a3 = p(3); a4 = p(4);
b1 = p(5); b2 = p(6);
% Definición del sistema glucosa-insulina de acuerdo con (2.3) (ver tesis)
gprima = @(t,g,h) -a1*g*h + a2*(g0-g)*Heaviside(g0-g) -
    a3*(g-g0)*Heaviside(g-g0) + a4*alimento3(t);
%
hprima = @(t,g,h) b1*(g-g0)*Heaviside(g-g0) - b2*h;

% Integración del sistema de ecuaciones
[t,g,h] = rk4(gprima,hprima,[t0,tfinal],g0,h0,ssize);

%
eat = zeros(length(t),1);
for i = 1:length(t)
eat(i) = alimento3(t(i));
end

```

Archivo: SimulationGraphChtp2.m

```

% En este script se soluciona el sistema de ecuaciones (2.4), dado los
% parámetros del sistema, las condiciones iniciales, el punto inicial y
% final del intervalo de tiempo y el tamaño de paso de integración.
% Estos datos pueden ser modificados por el usuario dentro de este
% script para obtener diferentes simulaciones. Como salida se obtienen
% las gráficas de la solución numérica del sistema de ecuaciones.
%
% Autor: Edwar Fabián Panqueba Moreno
% Fecha: 04/06/2021

% Parámetros del sistema glucosa insulina, [a1,a2,a3,a4,b1,b2]
prm = [0.05,1,0.03,4,0.5,2];

% Tiempo inicial y final del intervalo en el que se resuelve el sistema
t0 = 0; tfinal = 24;

% Condiciones iniciales del sistema de ecuaciones
g0 = 100; h0 = 0;

% Tamaño del paso de integración
delta_t = 0.01;

[t,g,h,food] = Sol_SistGH(prm,t0,tfinal,g0,h0,delta_t);

```

```

% Ploteo de la solución
figure
hold on
plot(t,g,'color',[0.1 0.1 0.5],'LineWidth',1.5)
plot(t,h,'color',[0.8 0.1 0],'LineWidth',1.5)
plot(t,food,'color',[0.3 0.5 0.1],'LineWidth',1.5)
xlim([0,24])
xticks(0:2:24)
ylabel('Concentración')
xlabel('Tiempo (Horas)')
title('Solución del sistema de ecuaciones de la dinámica Glucosa-Insulina')
legend('Glucosa (mg/100 mL)','Insulina (uU/mL)','Alimento (g
      CHO)','Location','northwest')
hold off

```

B.2. Códigos del Capítulo 3

Archivo: randomData.m

```

function [ND,ND1] = randomData(X,Y)
%randomData perturba los vectores de la dinámica glucosa e insulina
%obtenidos de la integración del sistema (2.4) con el vector de parámetros
%"real". Los datos obtenidos de esta rutina se usarán como los datos
%observados para construir la estimación tanto inicial como final del
%vector de parámetros. El error agregado es de alrededor  $10^{-3}$ ,  $10^{-4}$ 
%generados aleatoriamente. El usuario puede hacer este error más pequeño o
%más grande, de acuerdo a sus necesidades, modificando la condición del
%ciclo while.
%
%ENTRADAS:
%X,Y: Vectores que contiene las concentraciones de la glucosa y la insulina
%      para cada tiempo considerado en el intervalo [0,24], obtenidos de la
%      integración del sistema (2.4) con el vector de parámetros real.
%
%SALIDAS:
%ND,ND1: Vectores que contiene los datos perturbados de las
%         concentraciones de glucosa e insulina.
%
% Autor: Edwar Fabián Panqueba Moreno.
% Fecha: 04/06/2021.
%

R = (-1)^(ceil(10*rand))*rand(length(X),1);
R1 = (-1)^(ceil(10*rand))*rand(length(X),1);

ND = X + R;
ND1 = Y + R1;

```

```

for i = 1:length(X)
    while abs(X(i)-ND(i)) > 0.001
        aux = R(i)/10;
        R(i) = aux;
        ND(i) = X(i) + R(i);
    end

    while abs(Y(i)-ND1(i)) > 0.001
        aux = R1(i)/10;
        R1(i) = aux;
        ND1(i) = Y(i) + R1(i);
    end
end
end

```

Archivo: AUX_SYSTEM.m

```

function ddt = AUX_SYSTEM(t, ci_aux, p0)

% AUX_SYSTEM contiene el sistema de ecuaciones diferenciales auxiliares que
% debe ser resuelto para la construcción de la matriz jacobiana, necesaria
% para la implementación del método Gauss_Newton. Se evalúan las derivadas
% parciales de cada una de las ecuaciones del sistema (2.3) con respecto a
% cada uno de los parámetros del sistema, y evaluadas en el instante de
% tiempo t.

% ENTRADAS:
% t: tiempo en el que se evalúa cada derivada parcial.
% ci_aux: Condición inicial de cada ecuación que compone el sistema.
% p0: Vector que contiene los valores de los parámetros.
%
% SALIDAS:
% ddt: Matriz que contiene las soluciones del sistema de ecuaciones en cada
% tiempo t.
%
% Autor: Edwar Fabián Panqueba Moreno.
% Fecha: 04/06/2021.

% Asignación de los parámetros entrados como argumento a la función.
a1 = p0(1,1); a2 = p0(2,1); a3 = p0(3,1); a4 = p0(4,1);
b1 = p0(5,1); b2 = p0(6,1);

% Término del valor basal de la glucosa.
g0 = 100;

% Iniciación de cada una de las variables soluciones del sistema de
% ecuaciones, a partir de las condiciones iniciales.
g = ci_aux(1);

```

```

h = ci_aux(2);
dgda1 = ci_aux(3);
dhda1 = ci_aux(4);
dgda2 = ci_aux(5);
dhda2 = ci_aux(6);
dgda3 = ci_aux(7);
dhda3 = ci_aux(8);
dgda4 = ci_aux(9);
dhda4 = ci_aux(10);
dgdb1 = ci_aux(11);
dhdb1 = ci_aux(12);
dgdb2 = ci_aux(13);
dhdb2 = ci_aux(14);

% Función del alimento
z = alimento(t,0);

% Define el sistema de ecuaciones diferenciales auxiliares. Su solución
% proporciona las entradas de la matriz jacobiana necesaria para el cálculo
% de los parámetros por el método de Gauss–Newton.
dgdtdt = -a1*g*h + a2*max(0,(g0 - g)) - a3*max(0,(g - g0)) + a4*z;
dhdt = b1*max(0,(g - g0)) - b2*h;

df1da1 = -a1*((dgda1*h) + (dhda1*g)) - g*h + a2*(-dgda1*Heaviside(g0 - g)) -
    a3*(dgda1*Heaviside(g - g0));
df2da1 = b1*(dgda1*Heaviside(g - g0)) - b2*dhda1;
df1da2 = -a1*((dgda2*h) + (dhda2*g)) + a2*(-dgda2*Heaviside(g0 - g)) +
    max(0,(g0 - g)) - a3*(dgda2*Heaviside(g - g0));
df2da2 = b1*(dgda2*Heaviside(g - g0)) - b2*dhda2;
df1da3 = -a1*((dgda3*h) + (dhda3*g)) + a2*(-dgda3*Heaviside(g0 - g)) -
    a3*(dgda3*Heaviside(g - g0)) - max(0,(g - g0));
df2da3 = b1*(dgda3*Heaviside(g - g0)) - b2*dhda3;
df1da4 = -a1*((dgda4*h) + (dhda4*g)) + a2*(-dgda4*Heaviside(g0 - g)) -
    a3*(dgda4*Heaviside(g - g0)) + z;
df2da4 = b1*(dgda4*Heaviside(g - g0)) - b2*dhda4;

df1db1 = -a1*((dgdb1*h) + (dhdb1*g)) + a2*(-dgdb1*Heaviside(g0 - g)) -
    a3*(dgdb1*Heaviside(g - g0));
df2db1 = b1*(dgdb1*Heaviside(g - g0)) + max(0,(g - g0)) - b2*dhdb1;
df1db2 = -a1*((dgdb2*h) + (dhdb2*g)) + a2*(-dgdb2*Heaviside(g0 - g)) -
    a3*(dgdb2*Heaviside(g - g0));
df2db2 = b1*(dgdb2*Heaviside(g - g0)) - b2*dhdb2 - h;

ddt=[dgdtdt; dhdt; df1da1; df2da1; df1da2; df2da2; df1da3; df2da3; df1da4; df2da4; df1db1;
df2db1; df1db2; df2db2];

```

Archivo: RK4_solAuxSystem.m

```
function [tout, yout] = RK4_solAuxSystem(FunFcn, tspan, y0, ssize)
```

```

%RK4   Resuelve un sistema de ecuaciones diferenciales ordinarias a partir
% del método de cuarto orden de Runge Kutta. Se usa para solución del
% sistema de ecuaciones diferenciales auxiliares que debe ser resuelto como
% parte del proceso de estimación de parámetros. Esta función simplifica en
% tiempo y recursos de cómputo la integración de las 14 ecuaciones que
% contiene el sistema auxiliar a ser resuelto.
%
% [t,y] = rk4('AUX_SYSTEM', tspan, y0) integra el sistema de ecuaciones
% diferenciales auxiliares colocado en el Script AUX_SYSTEM sobre el
% intervalo de tiempo tspan = [t0,tfinal], y usando el vector de
% condiciones iniciales y0.
%
% [t, y] = rk4('AUX_SYSTEM', tspan, y0, ssize) usa el paso de integración
% dado por ssize.
%
% ENTRADAS:
% 'AUX_SYSTEM'   – Contiene el sistema de 14 ecuaciones diferenciales a
%                  resolver de acuerdo con el método Gauss–Newton
%      t         – Tiempo (scalar).
%      y         – Vector columna de la solución.
% tspan = [t0, tfinal], tiempo inicial y final del intervalo de integración.
% y0       – Vector de condiciones iniciales para cada ecuación del sistema.
%
% SALIDAS:
% t        – Retorna los puntos de tiempo usados para la integración (vector
%            columna).
% y        – Retorna la solución numérica del sistema en cada uno de los puntos
%            (vector columna).
%
% Autor: Adaptado por Edwar Fabián Panqueba M, del encontrado en:
% https://www.math.purdue.edu/academic/files/courses/2009summer/MA26600/rk4.m
%
% Fecha: 04/06/2021.

% Asigna a variables independientes los valores dados por el vector tspan
t0 = tspan(1);
tfinal = tspan(2);

pm = sign(tfinal - t0);

% Si el usuario no ingresa tamaño de paso, el algoritmo lo calcula por
% defecto.
if nargin < 4, ssize = (tfinal - t0)/100; end

%
h = pm*ssize; %Tamaño de paso asignado a la variable h.
t = t0; % Inicialización del primer punto de tiempo a usar para la

```

```

    integración.
y = y0(:); % Inicialización de las condiciones iniciales de cada ecuación.

% Calcula el número de subintervalos en los que es dividido el intervalo de
% tiempo de integración [t0,tfinal], a partir del tamaño de paso.
dt = abs(tfinal - t0);
N = floor(dt/ssize) + 1;
if (N-1)*ssize < dt
N = N + 1;
end

% Asignación de tamaño a los vectores de salida.
tout = zeros(N,1);
tout(1) = t;
yout = zeros(N, size(y,1));
yout(1,:) = y.';
k = 1; % Inicialización del contador.

% Ejecución del bucle para la integración numérica de la ecuación.
while (k < N)
    % Condicional para verificar que no se ha llegado al último punto de
    % tiempo usado en la integración. Si es así, no se continúa incrementando
    % el tiempo de evaluación en la función.
    if pm*(t + h - tfinal) > 0
        h = tfinal - t;
        tout(k+1) = tfinal;
    else
        % Puntos del intervalo de tiempo en los que se integra el sistema de
        % ecuaciones.
        tout(k+1) = t0 + k*h;
    end
    k = k + 1;

    % Cálculo de los pasos intermedios del algoritmo RK de cuarto orden.
    k1 = feval(FunFcn, t, y);
    k2 = feval(FunFcn, t + h/2, y + h*k1/2);
    k3 = feval(FunFcn, t + h/2, y + h*k2/2);
    k4 = feval(FunFcn, t + h, y + h*k3);
    y = y + h*(k1 + 2*k2 + 2*k3 + k4)/6;
    t = tout(k);
    yout(k,:) = y.';
end

```

Archivo: ParametersEstimate.m

```
function [BetaEst, Nitr, ErrorArray] = ParametersEstimate(t, z, b0, ssize)
```

```

% ParametersEstimate estima el vector de parámetros del sistema de
% ecuaciones (2.3) que describe la dinámica Glucosa–Insulina, aplicando el

```



```

    método
% de Gauss–Newton, combinado con el método de Levenberg–Marquardt. La
% combinación de ambos métodos garantiza la convergencia a un vector de
% parámetros estimados, aún cuando la matriz jacobiana sea casi singular.
% En el capítulo 3 de este trabajo, se abordan todos los aspectos teóricos que
% se han tenido en cuenta para la implementación de este algoritmo, se
% recomienda su lectura para una mejor comprensión.

% ENTRADAS:
% t: Vector que contiene los puntos del tiempo a usar para la integración
%   numéricas de los sistemas de ecuaciones.
%
% z: Vector de los datos observados, obtenidos de experimentación o por
%   perturbación a los datos obtenidos de resolver el sistema (2.3) con unos
%   parámetros considerados como "reales".
%
% b0: Estimación inicial de los parámetros, obtenida por el método
%   propuesto en la sección xxx de este trabajo.
%
% ssize: Tamaño del paso de integración.
%
% SALIDAS:
% BetaEst: Vector con la estimación final de los parámetros, al alcanzar la
%   convergencia del método, de acuerdo al criterio de parada
%   establecido.
%
% Nitr:   Número de iteraciones necesarias para la convergencia del método.
%
% ErrorArray: Vector que contiene los errores de estimación de parámetros
%   calculados para cada iteración.

% Autor: Edwar Fabián Panqueba Moreno.
% Fecha: 04/06/2021.
%

NumbPrmt = 6; % Número de parámetros del sistema de ecuaciones a estimar.
ErrorArray = zeros(1,1); Nitr = 0;

xy0 = [100,0,0,0,0,0,0,0,0,0,0,0,0,0,0]; % Condiciones iniciales del sistema de
    ecuaciones auxiliar.

% Asignación del tamaño de la matriz Jacobiana del sistema con respecto a
% cada parámetro. (Ec. 3.22). De tamaño nL X p.
XM = zeros(length(t)*2,NumbPrmt);
e = zeros(length(t)*2,1); % Variable para el cálculo del vector de errores.

% Asigna al vector de parámetros la estimación inicial de la cual se parte
% parte el método G–N.
p = abs(b0);

```

```

if abs(p(2,1) - 1) > 1
p(2,1) = 0.9471;
end

error = 1; % Inicializa la variable que calcula el tamaño del error.
% Asigna un valor inicial al parámetro de dirección de descenso del método
% Levenberg–Marquardt
miu = 0.01;
% Reduce o amplía el parámetro de dirección de descenso, de acuerdo con
% el criterio de convergencia del método Levenberg–Marquardt
gamma = 10;

% Inicia el bucle del algoritmo Gauss–Newton. El criterio de parada es
% el tamaño del error de estimación, que debe ser menor al impuesto por el
% usuario.
% Encuentra una actualización del vector de parámetros, a partir de la
% anterior.
while error > 0.000001
    % Resuelve el sistema de ecuaciones auxiliares.
    [T,X] = RK4_solAuxSystem(@AUX_SYSTEM(t,x,p), [t(1), t(end)], xy0, ssize)
    %
    % Arma la matriz Jacobiana definida en (3.22)
    XM = armadoMatrizAux_Prueba(X(:,3:end), 2, NumbPrmt);

    k = 1;
    % Arma el vector de los errores, es decir, (z_i - y_i)
    for i = 1:(2*length(X))
        if mod(i,2)~=0
            e(i,1) = z(k,1) - X(k,1);
        end

        if mod(i,2)==0
            e(i,1) = z(k,2) - X(k,2);
            k = k+1;
        end
    end
end

error1 = 1; % Inicialización del error para el cálculo de la dir.
descenso.

% Este bucle calcula la dirección de descenso del método
% Levenberg–Marquardt.
while error1 >= error
    % Cálculo del incremento en el vector de parámetros para la iteración
    k.
    pk = (XM'*XM + miu*eye(6)) \ (XM'*e);
    error1 = norm(pk);
end

```

```

        % Si el error actual es mayor que el menor previo , aumenta el
        % parámetro de dirección de descenso.
        if error1 >= error
            miu = miu*gamma;
        end
    end

    % Si el error actual es menor que el error previo , disminuye el
    % parámetro de dirección de descenso.
    miu = miu/gamma;
    % Actualización del error.
    error = error1;

    BetaEst = p + pk; % Actualización del vector de parámetros
    p = BetaEst; % Vector actualizado con los valores de los parámetros.
    Nitr = Nitr + 1;
    ErrorArray(Nitr) = error;
end

```

Archivo: armadoMatrizAux.m

```

function [R] = armadoMatrizAux(A, NumEquations, cantParameters)
% armadoMatrizAux construye la matriz jacobiana siguiendo la
% estructura definida en (3.22), requerida para la implementación del
% método Gauss–Newton, a partir de la matriz que contiene las soluciones al
% sistema auxiliar de ecuaciones diferenciales definido en (3.31).
%
% ENTRADAS:
% A: Matriz de soluciones numéricas al sistema auxiliar de ecuaciones
%     diferenciales.
%
% NumEquations: Número de ecuaciones que contiene el sistema, determina la
%     cantidad de filas de la matriz jacobiana. En este caso toma el valor
%     de 2.
%
% CantParameters: Cantidad de parámetros que contiene el sistema de
%     ecuaciones diferenciales (2.4), en este caso toma el valor de 6.
%
% SALIDAD:
% R: Matriz jacobiana de las derivadas de las ecuaciones que componen el
%     sistema (2.4) con respecto a cada uno de los parámetros, siguiendo la
%     estructura definida en (3.22)
%
% Autor: Edwar Fabián Panqueba Moreno
% Fecha: 04/06/2021.
%
% Determina tamaño de la matriz de ingreso
sizeA = size(A);
% Inicializa variables a ser utilizadas dentro del algoritmo.

```

```

Z = zeros(sizeA(1)*NumEquations,1);
R = zeros(sizeA(1)*NumEquations,cantParameters);

s = 1;
for r = 1:NumEquations:sizeA(2)
    Aux = A(:,r:r+(NumEquations-1));
    k = 1;
    for i = 1:sizeA(1)
        for j = 1:NumEquations
            Z(k,1) = Aux(i,j);
            k = k + 1;
        end
    end
    R(:,s) = Z;
    s = s + 1;
end

```

Archivo: `EstPrmtInitial.m`

```

function [b0] = EstPrmtInitial(T,X,Y,delta_t)
% EstPrmtInitial Construye un vector de parámetros iniciales que será
% usado posteriormente como dato en entrada en la función que implementa el
% método Gauss–Newton.
%
% La estimación inicial consiste en la interpolación por spline cúbicos de
% los datos observados, y posteriormente utilizar su derivada para la
% estimación de los parámetros de las funciones f1 y f2 que componen el
% sistema de ecuaciones diferenciales (2.4), al realizar su cálculo a
% través de la formulación del problema de mínimos cuadrados lineal a
% resolver por medio de las ecuaciones normales de Gauss.
%
% ENTRADAS:
% T: Vector de los puntos de tiempo en los que se estudia la dinámica del
% problema considerado.
%
% X: Vector que contiene las concentraciones de glucosa que componen el
% conjunto de datos observados en el intervalo de tiempo [0,24].
%
% Y: Vector que contiene las concentraciones de insulina que componen el
% conjunto de datos observados en el intervalo de tiempo [0,24].
%
% delta_t: Tamaño de paso para la integración numérica.
%
% SALIDAS:
% b0: Vector que contiene la estimación de parámetros inicial, después de
% resolver el problema de mínimos cuadrados lineales asociado.
%
% Autor: Edwar Fabián Panqueba Moreno
% Fecha: 04/06/2021.

```

```
g0 = 100; % Valor basal de la glucosa.
% Contadores
k = 1;
i = 1;

% Número de nodos a usar para la interpolación. Puede ser modificado por el
% usuario, dependiendo de la cantidad que necesite para representar la
% tendencia de los datos.
NumNodos = 61;

% Vector que contiene los nodos a usar para la interpolación.
x = zeros(NumNodos,1);

% Vectores que contienen las imágenes de los nodos para glucosa e insulina.
y = zeros(NumNodos,1);
yy = zeros(NumNodos,1);

% Matriz de entradas correspondientes a la evaluación de las funciones f1 y
% f2 del Sistema Glucosa-Insulina, en los tiempos muestrales seleccionados.
A = zeros(NumNodos-1,4); % El 4 indica el número de parámetros de f1.
B = zeros(NumNodos-1,2); % El 2 indica el número de parámetros de f2.
% NúmParámetros f1 + NumParámetros f2 = p.

% Vector que contiene la función de interpolación de la glucosa, evaluada
% en los tiempos muestrales.
b = zeros(NumNodos-1,1);

% Vector que contiene la función de interpolación de la glucosa, evaluada
% en los tiempos muestrales.
b2 = zeros(NumNodos-1,1);

% Vector que contiene los puntos muestrales seleccionados del conjunto de
% datos.
tp = zeros(NumNodos-1,1);

% Bucle para seleccionar los nodos de la interpolación.
ni1 = (7*1/delta_t) + 1;
ni2 = (9*1/delta_t) + 1;
while k <= length(X)
    x(i) = T(k);
    y(i) = X(k);
    yy(i) = Y(k);
    % La condición ayuda a seleccionar puntos más seguidos, en algunos
    % intervalos donde se requiera, para mejorar el polinomio obtenido.
    if (k >= ni1 && k < ni2) || (k >= (10/delta_t) + ni1 && k < (10/delta_t)
+ ni2)
        k = k + 1/(delta_t*10);
    end
end
```

```

    else
        k = k + 1/delta_t;
    end
    i = i + 1;
end

% Construye el spline para la función de la glucosa.
Coef_ix = Spline_Natural(x,y);
% Construye el spline para la función de la insulina.
Coef_iy = Spline_Natural(x,yy);

% Selecciona los puntos muestrales del intervalo de tiempo [t_0,T].
for i = 1:NumNodos-1
    tp(i) = x(i) + (x(i+1) - x(i))*rand(1,1);
end

% El bucle construye las matrices señaladas en (3.6), a partir de las
    funciones
% f1 y f2 que componen el sistema de ecuaciones (2.4) evaluadas en cada uno
% de los puntos muestrales. Estas matrices permitirán encontrar la estimación
% de parámetros lineal a usar como estimacion inicial.
for i = 1:NumNodos-1
    for j = 1:4
        if j == 1
            A(i,j) =
-1*polyval(Coef_ix(i,:),tp(i))*polyval(Coef_iy(i,:),tp(i));
            B(i,j) = max(0.5, (polyval(Coef_ix(i,:),tp(i))- g0));
        end

        if j == 2
            A(i,j) = max(0.5,(g0 - polyval(Coef_ix(i,:),tp(i))));
            B(i,j) = -1*polyval(Coef_iy(i,:),tp(i));
        end

        if j == 3
            A(i,j) = -max(0.5, (polyval(Coef_ix(i,:),tp(i))- g0));
        end

        if j == 4
            A(i,j) = alimento3(tp(i));
        end
    end
end
%
% Evalúa la derivada de la función de interpolación en los ptos
% muestrales
b(i) = polyval(Coef_ix(i,1:3).*[3 2 1],tp(i));
b2(i) = polyval(Coef_iy(i,1:3).*[3 2 1],tp(i));
end

```

```

% Resuelve la ecuación normal de Gauss del planteamiento de un problema de
% mínimos cuadrados lineal. Se obtienen dos vectores, uno con la estimación
% de los parámetros que intervienen en f1, y otro con los parámetros de f2.
% La concatenación de ambos parámetros conforman el vector inicial de la
% estimación de parámetros.
p = (A'*A)\(A'*b);
p2 = (B'*B)\(B'*b2);

b0 = [p;p2];

```

Archivo: Spline_Natural.m

```

function [C_Pol] = Spline_Natural(x,y)
% Spline Natural construye un spline interpolador cúbico para un conjunto
% de datos que corresponden con los datos observados del problema de
% estimación de parámetros. La función devuelve los coeficientes de un
% polinomio de la forma  $ax^3 + bx^2 + cx + d$  entre cada par de nodos
% consecutivos, con condiciones de frontera natural.
%
% ENTRADAS:
% x:      Vector de nodos x_1, x_2, ... , x_n.
% y:      Vector de evaluaciones f(x_1), f(x_2), ... f(x_n).
%
% SALIDAS:
% C_POL:  Matriz que contiene los coeficientes del polinomio cúbico que
%         interpola
%         cada par de nodos consecutivos.
%
% Autor:  Edwar Fabián Panqueba Moreno.
% Fecha: 25/02/2019.

% Inicialización de las variables necesarias para la construcción del
% polinomio interpolador.
n = length(x);
h = zeros (1,length(x) - 1);
M = zeros (length(x),length(x));
b = zeros (n,1);
M(1,1) = 1;
M(n,n) = 1;
Coef_M = zeros(n,4);
C_Pol = zeros(n-1,4);

for i = 1:length(x) - 1
    h(i) = x(i + 1) - x(i);
end

for j = 2:length(x) - 1
    M(j,j-1) = h(j-1);

```

```

M(j,j) = 2*(h(j - 1) + h(j));
M(j,j+1) = h(j);
b(j) = (3/h(j)*(y(j+1) - y(j))) - (3/h(j-1)*(y(j)-y(j-1)));
end

Coef_M(:,1) = y;
Coef_M(:,3) = Gauss_Tridiagonal(M,b); % Resuelve el sistema Ax = b con A
    matriz tridiagonal.

for i = 1:n-1
    Coef_M(i,2) = (1/h(i)*(y(i + 1) - y(i))) - (h(i)/3*(2*Coef_M(i,3) +
    Coef_M(i + 1,3)));
    Coef_M(i,4) = (Coef_M(i + 1,3) - Coef_M(i,3))/(3*h(i));
end

% Calcula los coeficientes del polinomio que interpola cada par de nodos
% consecutivos del conjunto de datos ingresados.
k = 1;
while k < n
    Pol_Newton = Coef_M(k,1);
    q = 1;
    for i = 2:4
        monomio = [1, -x(k)];
        q = conv(q,monomio);
        Pol_Newton = [0,Pol_Newton] + Coef_M(k,i)*q;
    end
    C_Pol(k,:) = Pol_Newton;
    k = k + 1;
end

```

Archivo: Gauss_Tridiagonal.m

```

function [x] = Gauss_Tridiagonal(A,b)
% Gauss_Tridiagonal resuelve un sistema de ecuaciones Ax = b, con A matriz
% tridiagonal, usando factorización LU.
%
% ENTRADAS:
% A = Matriz asociada al sistema de ecuaciones, es tridiagonal.
% b = Vector con los términos del lado derecho de las ecuaciones que
%     compone el sistema.
%
% SALIDAS:
% x = Vector que contiene los valores que resuelven el sistema de
%     ecuaciones.
%
% Autor: Edwar Fabián Panqueba Moreno.
% Fecha: 25/02/2019.
%
n = length(A);

```



```

L = eye(n);
for i = 2:n
    m = A(i,i-1)/A(i-1,i-1);
    for j = i-1:n
        A(i,j) = A(i,j) - m*A(i-1,j);
    end
    L(i,i-1) = m;
end

y = SustProgr_Tridiag(L,b,n);
x = SustRegr_Tridiag(A,y,n);

```

Archivo: SustProgr_Tridiag.m

```

function [y] = SustProgr_Tridiag(L,b,n)
% Función de apoyo para la resolución de un sistema de ecuaciones Ax = b,
% con A matriz tridiagonal, usando factorización LU.
%
% ENTRADAS:
% L = Matriz triangular inferior.
% b = Vector que contiene los términos del lado derecho del sistema Ax = b.
%
% SALIDAS:
% y = Vector que contiene términos necesarios para la implementación del
% algoritmo de Gauss para solucionar el sistema.
%
% Autor: Edwar Fabián Panqueba Moreno.
% Fecha: 25/02/2019.
%
y = zeros(n,1);
for i = 1:n
    if L(i,i) == 0
        disp('La Matriz es Singular');
        break
    end
    if i == 1
        y(i) = b(i)/L(i,i);
    else
        y(i) = (b(i) - L(i,i-1)*y(i-1))/L(i,i);
    end
end
end

```

Archivo: SustRegr_Tridiag.m

```

function [x] = SustRegr_Tridiag(U,y,n)
% Función de apoyo para la resolución de un sistema de ecuaciones Ax = b,
% con A matriz tridiagonal, usando factorización LU.
%
% ENTRADAS:

```

```

%L = Matriz triangular superior.
%b = Vector que contiene la solución del sistema Ax = b.
%
%SALIDAS:
%y = Vector que contiene términos necesarios para la implementación del
% algoritmo de Gauss para solucionar el sistema.
%
% Autor: Edwar Fabián Panqueba Moreno.
% Fecha: 25/02/2019.
%
x = zeros(n,1);
for i = n:-1:1
    if U(i,i) == 0
        disp('La Matriz es Singular');
        break
    end
    if i == n
        x(i) = y(i)/U(i,i);
    else
        x(i) = (y(i) - U(i,i+1)*x(i+1))/U(i,i);
    end
end
end

```

Archivo: PrincipalCap3.m

```

% Script principal para la ejecución de la simulación que genera la
% estimación de parámetros para los dos casos de pacientes (sano y
% diabético tipo 2) que se consideran en el capítulo 3 de este trabajo.
% Para iniciar la simulación, el usuario debe digitar "1" si la simulación
% se realiza para un paciente sano, o debe digitar "2" si se hará para el
% un paciente diabético tipo 2. La simulación consiste en resolver el
% sistema de ecuaciones que describe la dinámica de la glucosa-insulina
% (2.4) usando el método Runge Kutta explicada en el capítulo 2, con un
% vector de parámetros que se consideran los valores "reales", de
% aquí se obtienen dos vectores que contiene los valores de la dinámica de
% la glucosa e insulina para un intervalo de tiempo de 0 a 24 horas con un
% paso de integración de 0.01, el usuario puede cambiar el tamaño del paso
% en este script, modificando el valor de la variable delta_t.
%
% Posteriormente, los datos son perturbados con un error de más o menos
%  $10^{-3}$  o  $10^{-4}$  usando la función randomData.m, y estos datos perturbados
% serán los considerados como los "datos observados" de acuerdo con (3.2).
% Con los datos perturbados, se realiza una estimación inicial del vector
% de parámetros, construyendo un spline cúbico que describa la tendencia de
% los datos observados al hacer la selección de los nodos convenientemente,
% a través de la función EstPrmtInital, programada teniendo en cuenta los
% aspectos teóricos abordados en las secciones 3.1 a 3.3 y la sección 3.7
% de mismo capítulo 3.
%

```

```
% La estimación inicial se ingresa como dato de entrada a la función
% ParametersEstimate.m, en la cual se encuentra implementado el método
% Gauss–Newton combinado con el método de Levenberg–Marquardt. La función
% devuelve un vector con la estimación final de los parámetros, cuyas
% componentes se esperan sean muy cercanas a las componentes del vector de
% parámetros reales.
%
% ENTRADAS:
% opc = Guarda un valor de "1" o "2" para determinar si la simulación se
% realiza para un paciente sano o diabético. Algunos de los parámetros como
% el paso de integración de los métodos de solución de las ecuaciones
% diferenciales y el vector de parámetros reales pueden ser modificados por
% el usuario (delta_t y prm).
%
% SALIDAS:
% Gráficos de las dinámicas de la glucosa, la insulina con sus datos
% observados, en el commando Windows se visualiza el vector de estimación
% inicial y final de parámetros (Beta_0 y Beta_Final), y un gráfico con la
% curva del error de estimación vs el número de iteraciones realizada hasta
% la convergencia del método Gauss_Newton a la estimación final de los
% parámetros.
%
% Autor: Edwar Fabián Panqueba Moreno.
% Fecha: 04/06/2021.

clear all
clc
% Solicita al usuario ingresar información para determinar con qué tipo de
% paciente se hará la simulación.
opc = input('Para estimación de parámetros paciente sano, digite 1. Para
    estimación de paciente diabético, digite 2: ');
fprintf('\n')

% Si el dato de entrada es 1, ingresa los siguientes valores como
% parámetros reales del sistema de ecuaciones (2.4) para un paciente sano.
% Los valores que aparecen en el vector corresponderán a los valores de
% los parámetros [a1,a2, a3, a4, b1, b2] en este respectivo orden.
if opc == 1
prm = [0.05,1,0.03,4,0.5,2];
end

% Si el dato de entrada es 1, ingresa los siguientes valores como
% parámetros reales del sistema de ecuaciones (2.4) para un paciente sano.
% Los valores que aparecen en el vector corresponderán a los valores de
% los parámetros [a1,a2, a3, a4, b1, b2] en este respectivo orden.
if opc == 2
prm = [0.05,1,0.03,4,0.01,2];
end
```

```

%Se asingan los extremos inicial y final del intervalo de tiempo, los
%valores iniciales para la solución del sistema (2.4) y el paso de
%integración.

tic
t0 = 0; tfinal = 24;
g0 = 100; h0 = 0;
delta_t = 0.01;
%Se resuelve numéricamente el sistema (2.4) con los datos ingresados
[T,G,H,Food] = Sol_SistGH(prm, t0, tfinal ,g0, h0, delta_t);

%Se inicializan las variables que contendrán solo algunos de los datos
%observados, para posteriormente construir las gráficas.
MinSubint = round((1440/length(T)),1);
if MinSubint < 1
    MinSubint = MinSubint/delta_t;
end
T_Odata = zeros(round(length(T)/MinSubint) + 1,1);
G_Odata = zeros(round(length(T)/MinSubint) + 1,1);
H_Odata = zeros(round(length(T)/MinSubint) + 1,1);
StepGraph = MinSubint;

%Se perturban los datos obtenidos de la integración numérica de (2.4) con
%el vector de parámetros real.
[X,Y] = randomData(G,H);

%Se obtiene una estimación inicial del vector de parámetros, ingresando
%el conjunto de dtos observados.
Beta_0 = EstPrmtInitial(T',G,H,delta_t);
fprintf('El vector de parámetros iniciales Beta_0 es: \n')
disp(Beta_0);

%Se obtiene la estimación final del vector de parámetros, ingresando el
%vector de estimación inicial y el conjunto de datos observados. Se obtiene
%también el número de iteraciones y el error de estimación hasta la
%convergencia del método.
[Beta_Final, NumIterc, Errors] = ParametersEstimate(T',[X,Y],Beta_0);
fprintf('El vector de parámetros estimados Beta es: \n')
disp(abs(Beta_Final));

j = 1;
n = 1;
while j <= length(T_Odata)
    T_Odata(j) = T(n);
    G_Odata(j) = X(n);
    H_Odata(j) = Y(n);
    j = j + 1;
    n = n + StepGraph;
end

```

```
end
```

```
% Soluciona el sistema (2.4) con el vector final de parámetros estimados.
```

```
[T,G,H,Food] = Sol_SistGH(abs(Beta_Final),t0,tfinal,g0,h0,delta_t);
```

```
% Dibuja la gráfica de la glucosa y los puntos observados tomados.
```

```
figure
```

```
hold on
```

```
p2 = plot(T,G,'color',[0.3 0.6 0.7],'LineWidth',3);
```

```
p1 = plot(T_Odata,G_Odata,'o','MarkerSize',4,'MarkerEdgeColor',[0.9,0,0.4],...  
    'MarkerFaceColor',[0.9,0,0.4],'LineWidth',3.5);
```

```
xlim([0,24])
```

```
ylim([80,max(G)+5])
```

```
xticks([0 2 4 6 8 10 12 14 16 18 20 22 24]);
```

```
ylabel('Concentración')
```

```
xlabel('Tiempo (Horas)')
```

```
title('Dinámica Glucosa con parámetros estimados')
```

```
legend([p1 p2],{'Valores observados','Glucosa (mg/100  
    mL')},'Location','northwest')
```

```
hold off
```

```
% Dibuja la gráfica de la insulina y algunos de los puntos observados.
```

```
figure
```

```
hold on
```

```
p4 = plot(T,H,'color',[0.7 0.8 0],'LineWidth',3);
```

```
p3 = plot(T_Odata,H_Odata,'o','MarkerSize',4,'MarkerEdgeColor',[0.9,0,0.4],...  
    'MarkerFaceColor',[0.9,0,0.4],'LineWidth',3.5);
```

```
xlim([0,24])
```

```
ylim([-0.5,max(H)+0.2])
```

```
xticks([0 2 4 6 8 10 12 14 16 18 20 22 24]);
```

```
ylabel('Concentración')
```

```
xlabel('Tiempo (Horas)')
```

```
title('Dinámica Insulina con parámetros estimados')
```

```
legend([p3 p4],{'Valores observados','Insulina  
    (uU/mL')},'Location','northwest')
```

```
hold off
```

```
% Gráfico del número de iteraciones Vs. error de estimación
```

```
ArrayIterc = 1:1:NumIterc + 1;
```

```
Errors1 = zeros(1,length(Errors)+1);
```

```
ArrayIterc(1) = 0;
```

```
Errors1(1) = 1;
```

```
Errors1(2:end) = Errors(:);
```

```
figure
```

```
semilogy(ArrayIterc,Errors1,'color',[0.9 0.1 0.5],'LineWidth',2)
```

```
xlim([0,NumIterc+1])
```

```
ylim([Errors1(end),1])
```

```
ylabel('Error  $\|\Delta \beta_k\|$ ','Interpreter','latex')
```

```
xlabel('Número de Iteración')
```

```
title('Minimización del error en cada iteración')
toc
```

B.3. Códigos del Capítulo 5

Archivo: obj_fun_diab.m

```
function f = obj_fun_diab(X,h, Tinicial , Tfinal ,gd, prmt ,mm)
% obj_fun_diab contiene la función objetivo a ser minimizada en el problema
% de control óptimo definido en el capítulo 4 de este trabajo.
%
% ENTRADAS:
% X: Vector columna que contiene los valores que toma el control u_k, en
% cada uno de los tiempos en los que se divide el intervalo [t_0,T].
% h: Tamaño de paso de integración.
% Tinicial: Tiempo inicial del intervalo de integración [t_0,T].
% Tfinal: Tiempo final del intervalo de integración [t_0,T].
% gd: Vector que contiene las concentraciones de glucosa deseada.
%
% SALIDAS:
% f: Valor del funcional en cada iteración del proceso de optimización.
%
% Autor: Edwar Fabián Panqueba Moreno.
% Fecha: 15/06/2021.

% Valores iniciales para las dos ecuaciones del sistema glucosa-insulina
g0 = 100; h0 = 0.78;

% Cantidad de subintervalos en los que es dividido el intervalo de
% integración.
N = Tfinal/h;

% Resuelve el sistema de ecuaciones de la dinámica glucosa-insulina
% modificado para la inclusión de la variable de control, la cual influye
% en la trayectoria de la variable de estado (en este caso la glucosa) para
% el cumplimiento del objetivo de la solución del problema.
[t,x,y] = rk_diab(prmt, Tinicial , Tfinal ,g0,h0,N,X,mm);

% Variables auxiliares para el cálculo de la suma que involucra la función
% objetivo, sin necesidad de usar ciclos, ya que su uso implica más tiempo
% de cómputo.
w = ones(1,length(x(:,1)));
D = diag(w);

% Evaluación de la función objetivo.
f = h*((x(:,1) - gd)')'*D*(x(:,1) - gd) + (X - 10)'*(X - 10));
```

 Archivo: con_fun_diab.m

```

function [s,r] = con_fun_diab(X,h, Tinicial ,Tfinal)
%con_fun_diab contiene las restricciones de igualdad y desigualdad
% definidas sobre la formulación del problema de control óptimo.
%
%ENTRADAS:
%X: Vector columna que contiene los valores que toma el control u_k, en
% cada uno de los tiempos en los que se divide el intervalo [t_0,T].
%h: Tamaño de paso de integración.
% Tinicial: Tiempo inicial del intervalo de integración [t_0,T].
% Tfinal: Tiempo final del intervalo de integración [t_0,T].
%
% SALIDAS:
%r: Vector columna con los valores de las restricciones de desigualdad para
% cada uno de los tiempos t_k en los que se resuelve el problema de
% control.
%s: vector columna con los valores de las restricciones de igualdad para
% cada uno de los tiempos t_k en los que se resuelve el problema de
% control.
%
% Autor: Edwar Fabián Panqueba Moreno
% Fecha: 15/06/2021.

% Valores iniciales del sistema de ecuaciones (2.3)
g0 = 100; h0 = 0.78;
% Cantidad de subintervalos en los que es dividido el intervalo de
% integración.
N = Tfinal/h;

% Resuelve el sistema de ecuaciones de la dinámica glucosa-insulina
% modificado para la inclusión de la variable de control, la cual influye
% en la trayectoria de la variable de estado (en este caso la glucosa) para
% el cumplimiento del objetivo de la solución del problema.
[t,x,y] = rk_diab(Tinicial ,Tfinal ,g0,h0,N,X);

% Evalúa las condiciones de desigualdad definidas en (5.2)
s = [60 - x; x - 180];
% Evalúa las condiciones de desigualdad definidas en (5.2)
r = [x(length(x)) - 100];

```

 Archivo: alimento.m

```

function [z] = alimento(t,mn)
% alimento devuelve la cantidad de carbohidratos que se van descomponiendo
% a lo largo del día por el proceso de digestión del organismo.
%
%ENTRADAS:
%t: Momento del día (tiempo) en el que se desea conocer la cantidad de
% carbohidratos que quedan sin descomponer en glucosa durante el día.

```

```
%mn: Tipo de menú alimenticio , dependiendo del escenario que se desee
%   simular.
%
```

```
%SALIDAS:
```

```
%z: Cantidad de carbohidratos en el instante de tiempo t.
```

```
% Autor: Edwar Fabián Panqueba Moreno.
```

```
% Fecha: 15/06/2021.
```

```
switch mn
```

```
  case 0 %Reservado para las simulaciones del capítulo 2 y 3.
```

```
    Qd = 50; %desayuno
```

```
    Kd = -0.25*log(1/50);
```

```
    td =8; % hora del desayuno
```

```
    Qo = 20; %onces
```

```
    Ko = -0.25*log(1/20);
```

```
    td = 10.5;
```

```
    Qa = 50; %almuerzo
```

```
    Ka = -1/5*log(1/50);
```

```
    ta = 13;
```

```
    Qc = 100; %cena
```

```
    Kc = -1/5*log(1/100);
```

```
    tc = 20;
```

```
    if t<8
```

```
        z = 0;
```

```
    end
```

```
    if 8<=t & t<10.5
```

```
        z = Qd*exp(-Kd*(t-8));
```

```
    end
```

```
    if 10.5<=t & t<12
```

```
        z = Qd*exp(-Kd*(t-8))+Qo*exp(-Ko*(t-10.5));
```

```
    end
```

```
    if 12<= t & t<18
```

```
        z = Qo*exp(-Ko*(t-10.5)) + Qa*exp(-Ka*(t-12)) ;
```

```
    end
```

```
    if 18<= t & t<25
```

```
        z = Qc*exp(-Kc*(t-18));
```

```
    end
```

```
  case 1
```

```
    % Escenario 1 de alimentación capítulo 5
```

```
    Qd = 75; %desayuno
```

```
    Kd = -0.25*log(1/50);
```

```
    td = 8; % hora del desayuno
```



```

Qo = 35; %onces
Ko = -0.25*log(1/20);
td = 10.5;

Qa = 50; %almuerzo
Ka = -1/5*log(1/50);
ta = 13;

Qc = 65; %cena
Kc = -1/5*log(1/100);
tc = 20;

if t<8
    z = 0;
end
if 8<=t & t<10.5
    z = Qd*exp(-Kd*(t-8));
end
if 10.5<=t & t<12
    z = Qd*exp(-Kd*(t-8))+Qo*exp(-Ko*(t-10.5));
end
if 12<= t & t<18
    z = Qo*exp(-Ko*(t-10.5)) + Qa*exp(-Ka*(t-12)) ;
end

if 18<= t & t<25
    z = Qc*exp(-Kc*(t-18));
end

case 2 % Escenario 2
Qd = 32; % desayuno
Kd = -0.25*log(1/50);
td = 8; % hora del desayuno

Qm = 20; % merienda
Km = -0.25*log(1/20);
tm = 10.5;

Qa = 42; % almuerzo
Ka = -1/5*log(1/50);
ta = 12.5;

Qo = 31;
Ko = -0.25*log(1/20);
to = 15.5;

Qc = 40; %cena
Kc = -1/5*log(1/100);
tc = 19;

```

```

    if t<8
        z=0;
    end
    if 8<=t & t<10.5
        z = Qd*exp(-Kd*(t-8));
    end
    if 10.5<=t & t<12.5
        z = Qd*exp(-Kd*(t-8))+Qm*exp(-Km*(t-10.5));
    end
    if 12.5<= t & t<15.5
        z = Qm*exp(-Km*(t-10.5)) + Qa*exp(-Ka*(t-12.5)) ;
    end
    if 15.5<= t & t<19
        z = Qa*exp(-Ka*(t-12.5)) + Qo*exp(-Ko*(t-15.5)) ;
    end
    if 19<= t & t<25
        z = Qo*exp(-Ko*(t-15.5)) + Qc*exp(-Kc*(t-19));
    end

case 3 % Escenario 3
    Qd = 76; % desayuno
    Kd = -0.25*log(1/50);
    td = 9; % hora del desayuno

    Qa = 80; % almuerzo
    Ka = -1/5*log(1/50);
    ta = 13;

    Qc = 86; % cena
    Kc = -1/5*log(1/100);
    tc = 18;

    if t<9
        z = 0;
    end
    if 9<=t & t<13
        z = Qd*exp(-Kd*(t-9));
    end
    if 13<=t & t<18
        z = Qd*exp(-Kd*(t-9))+Qa*exp(-Ka*(t-13));
    end
    if 18<= t & t<25
        z = Qa*exp(-Ka*(t-13)) + Qc*exp(-Kc*(t-18)) ;
    end

case 4 % Escenario 4
    Qd = 30; % desayuno

```

```

Kd = -0.25*log(1/50);
td = 10;    % hora del desayuno

Qa = 51;    % almuerzo
Ka = -1/5*log(1/50);
ta = 16;

if t<10
    z = 0;
end
if 10<=t & t<16
    z = Qd*exp(-Kd*(t-10));
end
if 16<=t & t<25
    z = Qd*exp(-Kd*(t-10))+Qa*exp(-Ka*(t-16));
end
end
end

```

Archivo: rk_diab.m

```

function [t,g1,h1] = rk_diab(p,t0,tf,g0,h0,n,u,mn)
%rk_diab resuelve el sistema de ecuaciones (2.3) que describe la dinámica
%de la glucosa-insulina, influida por los valores que toma el control
%u(t), para evaluar las condiciones impuestas sobre la formulación del
%problema de control óptimo (5.1).
%
%ENTRADAS:
%p: Parámetros del sistema de ecuaciones diferenciales (2.3), estimados a
% partir de los datos observados disponibles.
%t0: Extremo izquierdo del intervalo de integración.
%tf: Extremo derecho del intervalo de integración.
%g0, h0: Condiciones iniciales del sistema (2.3).
%n: Número de subintervalos en los que es dividido el intervalo [t_0,T].
%u: Vector que contiene los valores que toma el control u_k.
%mn: Indica el tipo de menú alimenticio de acuerdo con el escenario que se
% contemple para la simulación.
%
%SALIDAS:
%g1: Vector que contiene las concentraciones de glucosa en sangre para el
% intervalo de tiempo [t_0,T].
%h1: Vector que contiene las concentraciones de insulina en sangre para el
% intervalo de tiempo [t_0,T].
%t: Vector que contiene los tiempos en los que ha sido evaluada la
% solución del sistema de ecuaciones (2.3)
%
% Autor: Edwar Fabián Panqueba Moreno.
% Fecha: 15/06/2021,

```

```

%
% Tamaño de paso
h = (tf-t0)/n;
t = t0:h:tf;

% Asignación de los valores de los parámetros del sistema (2.3)
a1 = p(1); a2 = p(2); a3 = p(3); a4 = p(4); b1 = p(5); b2 = p(6);

% Reserva de memoria para los valores del control
u_k = zeros(length(t),1);
u_k(1:end-1,1) = u;

% Reserva memoria para n+1 elementos del vectores g1 y h1
g1 = zeros(n+1,1);
h1 = zeros(n+1,1);
g1(1) = g0; h1(1) = h0;

% Interpolación lineal del control para u_k y u_k/2
t1 = t0:h:tf-h;
tp = tf;
uh2 = interp1(t1,u',tp,'linear','extrap');
u_k(end) = uh2;

tp1 = 1.5:288.5;
u_middle = interp1(u_k,tp1);

for i=1:n
    k1 = -a1*g1(i)*h1(i) + a2*(g0-g1(i))*Heaviside(g0-g1(i)) - ...
        a3*(g1(i)-g0)*Heaviside(g1(i)-g0) + a4*alimento(t(i),mm);

    l1 = b1*(g1(i)-g0)*Heaviside(g1(i)-g0) - b2*h1(i) + u_k(i);

    k2 = -a1*(g1(i)+((h/2)*k1))*(h1(i)+((h/2)*l1)) + ...
        a2*(g0-(g1(i)+((h/2)*k1)))*Heaviside(g0-(g1(i)+((h/2)*k1))) - ...
        a3*((g1(i)+((h/2)*k1))-g0)*Heaviside((g1(i)+((h/2)*k1))-g0) + ...
        a4*alimento(t(i),mm);

    l2 = b1*((g1(i)+((h/2)*k1))-g0)*Heaviside((g1(i)+((h/2)*k1))-g0) - ...
        b2*(h1(i)+((h/2)*l1))+ u_middle(i);

    k3 = -a1*(g1(i)+((h/2)*k2))*(h1(i)+((h/2)*l2)) + ...
        a2*(g0-(g1(i)+((h/2)*k2)))*Heaviside(g0-(g1(i)+((h/2)*k2))) - ...
        a3*((g1(i)+((h/2)*k2))-g0)*Heaviside((g1(i)+((h/2)*k2))-g0) + ...
        a4*alimento(t(i),mm);

    l3 = b1*((g1(i)+((h/2)*k2))-g0)*Heaviside((g1(i)+((h/2)*k2))-g0) - ...
        b2*(h1(i)+((h/2)*l2)) + u_middle(i);

```

```

k4 = -a1*(g1(i)+((h)*k3))*(h1(i)+((h)*l3))+...
      a2*(g0-(g1(i)+((h)*k3))*Heaviside(g0-(h1(i)+((h)*k3))) -...
      a3*((g1(i)+((h)*k3))-g0)*Heaviside((h1(i)+((h)*k3))-g0) +...
      a4*alimento(t(i),mm);

l4 = b1*((g1(i)+((h)*k3))-g0)*Heaviside((g1(i)+((h)*k3))-g0) -...
      b2*(g1(i)+((h)*l3)) + u_k(i+1);

g1(i+1) = g1(i) + h*(k1+2*k2+2*k3+k4)/6;
h1(i+1) = h1(i) + h*(l1+2*l2+2*l3+l4)/6;
end

```

Archivo: Heaviside.m

```

function h=Heaviside(d)
if d<0
    h=0;
else
    h=1;
end

```

Archivo: PrincipalCap5.m

```

% Script principal para la ejecución de los 4 escenarios de alimentación
% contemplados en el capítulo 5.
%
% ENTRADAS:
% eT: Indica el escenario de alimentación a considerar para la simulación
%     a realizar. El usuario digita "1" para el escenario 1, "2" para simular
%     el escenario de alimentación 2, "3" para el escenario 2, "3" para simular
%     el escenario 3 y "4" para simular el escenario 4.
%
% TypePatient: Selecciona el tipo de paciente sobre el cual se desea
%              realizar la simulación. Digita "1" si desea simular los escenarios
%              alimenticios para un paciente diabético tipo 1, o "2" para paciente
%              diabético tipo 2.
%
% gl_d: Selecciona la forma de la glucosa deseada. Digita "1" si la forma
%       de la glucosa deseada resulta de la solución del sistema (2.4) con
%       parámetros de acuerdo a la dinámica glucosa-insulina de un paciente
%       sano,
%       o "2", si es un valor constante resultate del promedio de las
%       concentraciones de glucosa de un paciente sano durante 24 horas.
%
% SALIDAS:
% Gráficas de las dinámicas de la glucosa, insulina, alimento e inyecciones
% de insulina (control) para el escenario y tipo de paciente seleccionado.
%

```

```

% Autor: Edwar Fabián Panqueba Moreno.
% Fecha: 15/06/2021.

clear vars
clc
% Valores iniciales y finales del intervalo de integración
Tinicial = 0; Tfinal = 24;
% Tamaño de paso de integración
delta_t = 1/12;
% Número de subintervalos en los que se divide el intervalo [t_0,T]
N = Tfinal/delta_t;
% Condiciones iniciales del sistema (2.4)
g0 = 100; h0 = 0.78;
% Vector que contiene los valores iniciales de prueba para el control en
% cada tiempo u_k
x0 = 35*ones(N,1);
% Reserva memoria para la construcción de la gráfica del alimento.
eat = zeros(N+1,0);

% Restricciones de desigualdad sobre el control u(t)
Xmax = 60*ones(N,1);
Xmin = zeros(N,1);

eT = input('Indique el escenario de alimentación sobre el cual hacer la
simulación: ');
TypePatient = ('Indique el tipo de paciente a simular, "1" si es diabético
tipo 1 o "2" si diabético tipo 2: ');
gl_d = ('Indique el tipo de glucosa deseada a seleccionar. "1" para la forma
1, o "2" para la forma 2: ');

% Obtención de los datos observados y estimación de parámetros para cada
% escenario considerado.
switch typePatient
case 1
prmt = [0.05,1,0.03,4,0,2];
[t,g,h] = Sol_SistGH(prmt,t0,tfinal,g0,h0,ssize,delta_t,eT);
[gO,hO] = randomData(g,h);
Beta_0 = EstPrmtInitial(t,gO,hO,delta_t);
Beta_Est = ParametersEstimate(t,[gO,hO],Beta_0,delta_t);

case 2
prmt = [0.05,1,0.03,4,0.01,2];
[t,g,h] = Sol_SistGH(prmt,t0,tfinal,g0,h0,ssize,delta_t,eT);
[gO,hO] = randomData(g,h);
Beta_0 = EstPrmtInitial(t,gO,hO,delta_t);
Beta_Est = ParametersEstimate(t,[gO,hO],Beta_0,delta_t);
end

% Construcción de vector que contiene valores de la glucosa deseada

```

```

switch gl_d
    case 1
        % Valores de los parámetros para la dinámica del sistema (2.3) de
        % un paciente sano.
        p = [0.05,1,0.03,4,0.5,2];
        [gd,hd] = Sol_SistGH(p, Tinicial, Tfinal, g0, h0, h);
    case 2
        p = [0.05,1,0.03,4,0.5,2];
        [gd,hd] = Sol_SistGH(p, Tinicial, Tfinal, g0, h0, h);
        gd1 = mean(gd);
        gd = gd1*ones(1, length(hd));
end

tic
% Definición de las opciones de optimización
options =
    optimset('LargeScale','off','Display','iter','TolX',0.0001,'TolFun'...
,0.0001,'MaxIter',50000,'MaxFunEvals',50000,'Algorithm','sqp');

% Invocación de la rutina de Matlab para la solución del problema de
% control óptimo
[X,fval] = fmincon('obj_fun_diab',x0,[],[],[],[],Xmin,Xmax,'con_fun_diab',...
options,h,Tinicial,Tfinal,Beta_Est,gd,eT);
toc

% Instrucciones para la construcción de las gráficas.
T = [0:h:Tfinal-h];
[t,x,y] = rk_diab(Beta_Est, Tinicial, Tfinal, g0, h0, N, X, eT);

for i=1:length(t)
    eat(i) = alimento(t(i),eT);
end

% Gráfica concentración de la glucosa durante las 24 horas de día.
ax1 = subplot(2,2,1);
hold on
p1 = plot(t,x,'color',[0.3 0.6 0.7],'LineWidth',2);
p2 = plot([0,24],[180,180],'—','color',[0.8 0 0],'LineWidth',1.5);
p3 = plot([0,24],[60,60],'—','color',[1 0.8 0.3],'LineWidth',1.5);
p4 = plot(t,gd,':','color',[0.5 0.1 0.3],'LineWidth',1.2);
xlim([0,24])
ylim([50,max(gd)+10])
ax1.XTick = 0:2:24;
ylabel('Concentración glucosa (mg/100 mL)')
xlabel('Tiempo (Horas)')
title('Dinámica Glucosa paciente diabético controlado')
legend([p2 p3 p4],{'Nivel MÁX saludable','Nivel MÍN saludable','Glucosa
deseada (gd)'},'Location','southeast')

```

```
hold off

% Gráfica concentración de la insulina durante las 24 horas de día.
ax2 = subplot(2,2,2);
plot(t,y, 'color', [0.5 0.8 0], 'LineWidth', 2)
xlim([0,24])
ax2.XTick = 0:2:24;
ylabel('Concentración insulina (uU/mL)')
xlabel('Tiempo (Horas)')
title('Dinámica Insulina paciente diabético controlado')

% Gráfica función de alimento de acuerdo al escenario de simulación
% escogido.
ax3 = subplot(2,2,3);
plot(t,eat, 'color', [0.7 0.5 0.1], 'LineWidth', 2)
xlim([0,24])
ax3.XTick = 0:2:24;
xlabel('Tiempo (Horas)')
ylabel('Carbohidratos ingeridos (g)')
title('Función de Alimento')

% Gráfica de la inyección de insulina (control) durante las 24 horas del
% día.
ax4 = subplot(2,2,4);
plot(T,X, 'color', [0.7 0.3 1], 'LineWidth', 2)
xlim([0,24])
ax4.XTick = 0:2:24;
xlabel('Tiempo (Horas)')
ylabel('u(t) (uU/min)')
title('Inyección de Insulina (Control)')
```
