

Proyecto Fin de Carrera
Ingeniería Electrónica, Robótica y Mecatrónica

Estudio de la Enfermedad de la Fibromialgia como
Sistema de Control

Autor: Andrés Pastor Alfaro

Tutor: Ascensión Zafra Cabeza

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

Sevilla, 2019



Proyecto Fin de Carrera
Ingeniería Electrónica, Robótica y Mecatrónica

Estudio de la Enfermedad de la Fibromialgia como Sistema de Control

Autor:

Andrés Pastor Alfaro

Tutor:

Asunción Zafra Cabeza

Profesora titular

Dpto. de Ingeniería de Sistemas y Automática

Escuela Técnica Superior de Ingeniería

Universidad de Sevilla

Sevilla, 2019

Proyecto Fin de Carrera: Estudio de la Enfermedad de la Fibromialgia como Sistema de Control

Autor: Andrés Pastor Alfaro

Tutor: Asunción Zafra Cabeza

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

Presidente:

Vocales:

Secretario:

Acuerdan otorgarle la calificación de:

El Secretario del Tribunal:

Fecha:

*A mi familia, porque aunque somos pocos
en comparación con todo el mundo, he
aprendido más del mundo con ellos que con
los siete mil millones de personas restantes.*

*A mis compañeros de clase y a mi hermano,
porque en la vida nadie pelea solo.*

*A mi tutora, Asunción, por darme la
oportunidad de aprender que los problemas
no comprenden de disciplinas académicas*

Resumen

La fibromialgia -caracterizada por la hipersensibilidad y por un fuerte dolor musculoesquelético crónico-, es un desorden corporal sin cura universal cuyo tratamiento resulta arduo debido a las causas pseudo-desconocidas de la enfermedad. El objetivo principal de este material es facilitar el tratamiento del desorden de la fibromialgia mediante procedimientos propios de la ingeniería de control, buscando localizar el método que mejor se adapte a las necesidades del paciente y del equipo médico asociado en su lucha contra los significativos síntomas de esta enfermedad.

Por otro lado, y como objetivo secundario, este trabajo presenta un desarrollo completo de varias metodologías de control con apoyo de la herramienta MATLAB[®], sirviendo como caso ejemplo de aplicación de dichos procedimientos y facilitando así la adaptación de las tecnologías aplicadas a otros proyectos de índole similar, apoyando por tanto el uso de nuevas técnicas en el área de la salud y otros campos complejos que pueden beneficiarse de las ventajas asociadas a la ingeniería de control.

PALABRAS CLAVE: Naltrexona, fibromialgia, control, PID, Predictivo, MPC, identificación, tratamiento.

Abstract

Fibromyalgia -characterized by hypersensitivity and a strong chronic musculoskeletal pain- is a body disorder with no universal cure whose treatment is difficult due to the pseudo-unknown causes of the disease. The main objective of this material is to provide the treatment of fibromyalgia using control engineering procedures, trying to locate the best method that meet the needs of the patient and his medical staff in their fight against the symptoms of this disease.

On the other hand, and as a secondary objective, this paper presents a complete development of several control methodologies with the support of the MATLAB[®] tool, serving as an example of the application of these procedures and facilitating the adaptation of technologies applied to other projects with a similar nature, supporting the use of new techniques in the health area and other complex fields that can benefit from the strenghts of the control engineering.

Índice

Resumen	3
Abstract	5
Índice	7
Índice de Figuras	9
Notación	12
1. Introducción	13
1.1 <i>Motivación del trabajo</i>	14
1.2 <i>Objetivos</i>	16
1.3 <i>Contenido y metodología</i>	16
2. Modelado de la fibromialgia	19
<i>Identificación del sistema</i>	19
<i>Relación entre las señales, modelo de la respuesta del paciente</i>	24
2.2.1 <i>La herramienta systemIdentification, modelos por mínimos cuadrados</i>	26
2.2.2 <i>Funciones de transferencia por métodos gráficos.</i>	34
<i>Restricciones de cada señal y consecuencias de las mismas</i>	38
3. Control del tratamiento	43
3.1 <i>Control del proceso mediante PID</i>	45
3.3 <i>Control del proceso mediante técnicas predictivas (MPC)</i>	53
4. Personalización y revisiones médicas	57
5. Referencias	65
6. Anexos	67
<i>Anexo 1: Entrevista a un especialista médico y a un enfermo de fibromialgia.</i>	67
<i>Anexo 2: Bibliografía recomendada sobre ingeniería de control.</i>	68
<i>Anexo 3: Códigos de dosificación y MPC.</i>	69

Índice de Figuras

<i>Ilustración 1: Tender Points Sites of Fibromialgia. Fuente: https://www.ssa.gov/OP_Home/rulings/di/01/SSR2012-02-di-01.html</i>	13
<i>Ilustración 2: MATLAB® & MathWorks logo. Fuente: https://es.mathworks.com/?s_tid=gn_logo</i>	18
<i>Ilustración 3: Comparación entre ejemplos básicos de de la ingeniería de control y del seguimiento de un paciente</i>	20
<i>Ilustración 4: Primary self-report variables associated with naltrexone intervention of fibromyalgia, part 1 [7]</i>	22
<i>Ilustración 5: Primary self-report variables associated with naltrexone intervention of fibromyalgia, part 2 [7]</i>	22
<i>Ilustración 6: Sistema controlado propuesto en nuestro trabajo</i>	23
<i>Ilustración 7: Datos experimentales, respuesta global del estudio, pacientes de fibromialgia frente a un tratamiento a la naltrexona.</i>	26
<i>Ilustración 8: Pantalla de inicio de la aplicación systemIdentification</i>	27
<i>Ilustración 9: Ventana de "Import", botón "more" pulsado.</i>	27
<i>Ilustración 10: Resultado del ajuste de un conjunto de datos a una función cuadrática; en rojo pueden verse los datos iniciales, en azul, la curva tendencia por mínimos cuadrado. Fuente: https://es.m.wikipedia.org/wiki/Archivo:Linear_least_squares2.png</i>	28
<i>Ilustración 11: Ilustración 7, con posibles zonas ruidosas remarcadas.</i>	29
<i>Ilustración 12: Algunos de los modelos creados y comprobados para la generación del modelo de respuesta del paciente.</i>	29
<i>Ilustración 13: Respuesta de los modelos $far3sin10$ y $far2sin10$ frente al experimento original.</i>	30
<i>Ilustración 14: Montaje en Simulink® para la comprobación del modelo $far3sin10$ en el entorno de simulación donde comprobaremos futuros módulos de control.</i>	31
<i>Ilustración 15: $Far3sin10$ frente a experimento, $K=1$.</i>	32
<i>Ilustración 16: $Far3sin10$ frente a experiento, $K=1.75$</i>	33

<i>Ilustración 17: Far3sin10 frente a experimento, $K=1.75$, datos de 200 días</i>	33
<i>Ilustración 18: Far3sin10 frente al experimento completo. Reacciona a lo que consideramos como “ruido inicial” al generar el modelo, a pesar de haberlo eliminado en el preprocesado.</i>	34
<i>Ilustración 19: Función de transferencia de primer orden y parámetros de la misma [17]</i>	35
<i>Ilustración 20: Respuesta del modelo Gs1 frente a experimento y a Far3Sin10 (teniendo este último $K=1.75$)</i>	35
<i>Ilustración 21: Respuesta del modelo Gs2 frente a experimento y a Gs1</i>	36
<i>Ilustración 22: Modelo Gs3 frente a Gs1, Gs2 y experimento</i>	37
<i>Ilustración 23: Prueba de modelo comparativa entre función continua (Gs3) y discreta (Gz), gráficas de datos.</i>	40
<i>Ilustración 24: Prueba de modelo comparativa entre función continua (Gs3) y discreta (Gz), montaje Simulink®.</i>	41
<i>Ilustración 25: Uso de la herramienta rltool para la creación de controladores PID</i>	45
<i>Ilustración 26: Uso de la ventana PID Tuning para la creación de controladores según especificaciones.</i>	46
<i>Ilustración 27: Control continuo ante seguimiento de referencia, comparado con datos del experimento inicial.</i>	47
<i>Ilustración 28: Montaje Simulink® para la implementación de un controlador PID discreta en el sistema diseñado.</i>	48
<i>Ilustración 29: Control discreto ante seguimiento de referencia por PID, comparado con datos del experimento inicial.</i>	49
<i>Ilustración 30: Montaje Simulink® para la implementación de un controlador PID discreta en el sistema diseñado, con saturación a la salida del PID</i>	50
<i>Ilustración 31: Control discreto ante seguimiento de referencia por PID, comparado con datos del experimento inicial, saturado según valores comerciales.</i>	50
<i>Ilustración 32: Montaje Simulink® para la implementación de un controlador PID discreta en el sistema diseñado, con dosificación a la salida del PID</i>	51
<i>Ilustración 33: Control discreto ante seguimiento de referencia por PID, comparado con datos del experimento inicial, dosificado según valores comerciales.</i>	52
<i>Ilustración 34: Control discreto ante seguimiento de referencia por MPC, comparado con datos del experimento inicial, sin dosificación.</i>	54
<i>Ilustración 35: Control discreto ante seguimiento de referencia por MPC, comparado con datos del experimento inicial y con el PID dosificado, introduciendo dosificación en el MPC.</i>	55
<i>Ilustración 36: Sistema de control completo sobre un paciente con fibromialgia, con revisiones médicas y personalización incluidas. Ejemplo con revisiones con</i>	

<i>valor cualitativo “igual que la estimación del sistema”, marcado con un valor cuantificado de 50.</i>	60
<i>Ilustración 37: Sistema de control completo sobre un paciente con fibromialgia, con revisiones médicas y personalización incluidas. Ejemplo con revisiones con valor cualitativo “mejor que la estimación del sistema”, marcado con un valor cuantificado de 0.</i>	60
<i>Ilustración 38: Sistema de control completo sobre un paciente con fibromialgia, con revisiones médicas y personalización incluidas. Ejemplo con revisiones con valor cualitativo “mejor que la estimación del sistema”, marcado con un valor cuantificado de 100</i>	61
<i>Ilustración 39: Sistema de control completo sobre un paciente con fibromialgia, con revisiones médicas y personalización incluidas. Ejemplo primero.</i>	62
<i>Ilustración 40: Sistema de control completo sobre un paciente con fibromialgia, con revisiones médicas y personalización incluidas. Ejemplo segundo</i>	62
<i>Ilustración 41: Sistema de control completo sobre un paciente con fibromialgia, con revisiones médicas y personalización incluidas. Ejemplo tercero</i>	63
<i>Ilustración 42: Sistema de control completo sobre un paciente con fibromialgia, con revisiones médicas y personalización incluidas. Referencia al 0% de síntomas</i>	63
<i>Ilustración 43: Sistema de control completo sobre un paciente con fibromialgia, con revisiones médicas y personalización incluidas. Tiempo de periodicidad de citas de 7 días.</i>	64
<i>Ilustración 44: Sistema de control completo sobre un paciente con fibromialgia, con revisiones médicas y personalización incluidas. Tiempo de periodicidad de citas de 7 días.</i>	64

Notación

X	Cantidad de días indeterminado entre cita y cita médica
n	Variación arbitraria de la señal de entrada de una función de transferencia
m	Variación de la señal de salida de una función de transferencia cuya señal de entrada ha variado “n”
Gs1	Función de transferencia de primer orden, continua, que modela la respuesta de un paciente al tratamiento de naltrexona contra la fibromialgia, extraída según datos experimentales de artículos previos a este estudio.
Gs2	Función de transferencia de segundo orden subamortiguada, continua, que modela la respuesta de un paciente al tratamiento de naltrexona contra la fibromialgia, extraída según datos experimentales de artículos previos a este estudio
Gs3	Función de transferencia de segundo orden sobreamortiguada, continua, que modela la respuesta de un paciente al tratamiento de naltrexona contra la fibromialgia, extraída según datos experimentales de artículos previos a este estudio
Gz	Función de transferencia de segundo orden que modela la respuesta de un paciente al tratamiento de naltrexona contra la fibromialgia, extraída según datos experimentales de artículos previos a este estudio, y que proviene de la discretización del modelo G_{s3}
Cs	Función de transferencia continua para el control de un sistema modelado a partir de la función G_{s3}
Cz	Función de transferencia para el control de un sistema modelad a partir de la función G_z , proveniente de la discretización de C_s .

1. Introducción

“Era demasiado orgullosa para oponer una resistencia inútil.”

*- Peter Pan y Wendy
J. M. Barrie*

La fibromialgia, enfermedad crónica reconocida por la OMS en 1992, es un desorden común que afecta aproximadamente al 5% de las mujeres y al 1.6% de los hombres en la sociedad actual [1], caracterizada por una gran hipersensibilidad en los característicos “puntos sensibles” (*tender points*) del cuerpo humano (véase *Ilustración 1*) donde pasan a producirse fuertes dolores al contacto, acompañada por otros tantos síntomas igualmente desagradables con los que muchos se han resignado a vivir.

Si bien los síntomas que acompañan a esta enfermedad son comunes en el día a día y el número de personas que la sufren es realmente elevado, hasta hace apenas unos años ni siquiera era considerada como enfermedad. La ausencia de este estatus se debía fundamentalmente a la controversia que durante siglos ha causado su identificación por parte de los expertos en los pacientes, ya que los síntomas asociados son ciertamente comunes en otros conjuntos de lesiones y/o trastornos crónicos o eventuales, llegándose a dar el caso de que, durante numerosos años (e incluso hoy en día), ciertos sectores de la medicina consideren el término “fibromialgia” como “un simple apelativo con el que ponerle nombre a cualquier conjunto de fenómenos de dolor físico con origen posiblemente neurológico que un equipo médico no sepa diagnosticar, con el fin de evadir la responsabilidad de ahondar más en el caso.” (más información en *Anexo I*).

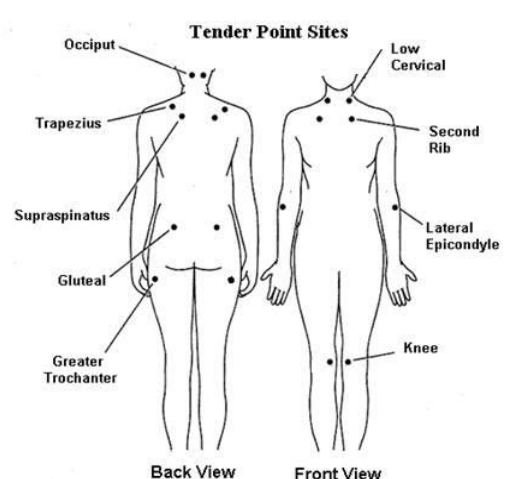


Ilustración 1: Tender Points Sites of Fibromialgia. Fuente: https://www.ssa.gov/OP_Home/rulings/di/01/SSR2012-02-di-01.html

Por suerte, no es así en todo el sector médico, donde desde un tiempo a esta parte han ido surgiendo

numerosos estudios y grupos de especialistas concienciados con esta enfermedad.

Realmente es en el año 1976 cuando, tras numerosos estudios del dolor muscular que se datan a partir de 1592, se hizo uso por primera vez del término “fibromialgia” [2], siendo el detonante de gran cantidad de proyectos que tratarían de acotar los síntomas, el diagnóstico y las relaciones y diferencias respecto a otros desórdenes crónicos parecidos [3]. Si bien en 1992 la OMS reconoce la fibromialgia como una enfermedad crónica, aún serían necesarias una gran cantidad de publicaciones antes de poder describir una forma bien definida de diagnosticar la fibromialgia, siendo las conclusiones de algunas de ellas aún dudosas para esos sectores de la medicina reacios a admitir la realidad de una enfermedad que atañe a tantas personas.

Gracias al mundo en el que vivimos, donde la tecnología y la investigación pueden darse la mano, muchos autores y equipos tratan a día de hoy de definir hasta los mínimos detalles de esta enfermedad y sus consecuencias, acercándonos cada vez más hacia la verdad de uno de los trastornos crónicos más comunes de la población. Si bien es cierto que los numerosos estudios con los que ya contamos han hecho posible diagnosticar innumerables veces esta enfermedad, aún queda un largo camino por recorrer antes de que la sociedad venza definitivamente a la fibromialgia.

Por suerte, es una meta al alcance de una humanidad que ha vencido y vencerá innumerables trastornos desde tiempos inmemoriales.

En un mundo actualizado como el nuestro, donde tenemos la oportunidad de identificar y tratar enfermedades que llevan disminuyendo la calidad de vida de la población durante siglos, un trastorno tan común como la fibromialgia merece los esfuerzos de la comunidad científica con el fin de aumentar los diagnósticos correctos, paliar sus síntomas y, en definitiva, hacer más felices a millones de personas.

Serán esa intención y ese espíritu los que nos apoyen a dar los primeros pasos de este trabajo, en una búsqueda de respaldar a esta sociedad en la lucha contra una adversidad que daña a tanta gente.

1.1 Motivación del trabajo

Pero, aunque tengamos esa intención y espíritu como apoyo, ¿qué necesita ser apoyado y qué motivos nos han llevado a crear este trabajo y no otro, y por qué tratar de estudiar la fibromialgia como sistema de control?

Quizá el motivo más simple por lo cual comenzar un trabajo como este lo hemos comentado varias veces en la introducción: la calidad de vida de una persona se ve enormemente reducida cuando sufre fibromialgia no diagnosticada o sin tratamiento eficaz. Esto se debe fundamentalmente a tres aspectos. El primero y ya nombrado son sus síntomas, que además de la hipersensibilidad en los *tender points* anteriormente remarcada (que se siente, en palabras textuales de enfermos, como “unos grilletos; hay veces que solo tienes un pequeñísimo hormigueo, como un calambre, y no te atreves ni a quedar para cenar con unos amigos porque igual hoy que te encuentras medio bien pero... ¿mañana podrás con tu cuerpo? ¿Podrás levantarte de la cama?”- más información en *Anexo 1*), la fibromialgia viene acompañada de otros efectos adversos tales como la fatiga, trastornos del sueño, anomalías intestinales, fuertes picos de ansiedad y desórdenes en el humor [4]. Si bien estos síntomas pueden aparecer en mayor o menor medida, de forma más o menos suave y con más o menos asiduidad en cada paciente, el segundo de los aspectos que reducen la calidad de vida de una persona se basa en que tratamos con una enfermedad que, actual y desgraciadamente, no tiene cura. Esto se relaciona directamente con el

tercero de los aspectos asociados: la dificultad que los equipos médicos tienen para diagnosticar la enfermedad y tratarla ya que, como destacamos anteriormente, los síntomas de la fibromialgia son comunes a otras enfermedades de afección musculoesquelética y los métodos más novedosos y eficaces para diagnosticarla aún no han llegado a todos los hospitales; sin contar además con la dificultad añadida de que las formas más sencillas de evaluar un daño musculoesquelético no tienen aplicación efectiva en la fibromialgia, debido a que, como muchas veces destacan los especialistas, uno de los problemas de este desorden es que “Aunque el dolor se produzca en un punto determinado del cuerpo, no hay lesiones ni daños visibles o palpables en la zona, debido a que la enfermedad se basa en un fallo neurológico y no en uno muscular o esquelético” (más información en *Anexo 1*).

Sin embargo, este motivo nos impulsaría más claramente a ahondar en los trastornos asociados a esta enfermedad, en cómo identificarlos y en cómo curar la afección.

¿Por qué entonces planteamos hacer de la fibromialgia un sistema de control?

Fundamentalmente, esto se sustenta en otros hechos tan destacables como los anteriores, comenzando por la curiosa realidad sobre los tratamientos eficaces contra la fibromialgia, ya que existen pacientes cuyos dolores menguan con tan solo hacer deporte, dejar de tomar gluten, pérdida de peso, estiramientos, yoga o imponiendo rutinas de sueño (más información en *Anexo 1*). No obstante, si bien estos tratamientos pueden o no producir mejoras en cada paciente en particular, con mayor o menor eficacia, numerosos estudios avalan el uso de tratamientos basados en la naltrexona (medicamento sobre el que hablaremos en futuros puntos de este trabajo) como medida eficaz contra la fibromialgia [1].

Además, debido a la naturaleza crónica de la enfermedad y las infinitas posibilidades de tratamiento (útil o inútil), establecer una forma eficaz de controlar la evolución de un paciente, reduciendo cuanto antes las molestias ocasionadas por la fibromialgia y buscando a la vez de forma rápida una terapia con grandes posibilidades de éxito se convierte en una plena necesidad. Si intentásemos apoyarnos en la práctica médica tradicional, basada en tratamientos que no necesariamente incluyen características individuales del paciente ni procesos de optimización exclusivos, podríamos generar desórdenes de otros tipos intentando paliar la enfermedad crónica (sin contar con el hecho de que no son necesariamente eficaces contra este tipo de enfermedades), mientras que el uso de tratamientos que se adapten al paciente, en los que las dosis se ajustan a una respuesta particular a través del tiempo, reducen el malgasto de medicamentos y las posibilidades de efectos secundarios indeseados, aumentando a su vez en gran cantidad las probabilidades de tener éxito en la mitigación de síntomas e incrementando la eficiencia de cada una de las consultas médicas (algo plenamente necesario en las asistencias médicas sobresaturadas de la sociedad actual, que deben hacer frente a presupuestos limitados), a la vez que hacen disminuir rápidamente los síntomas de unas enfermedades complejas que merman enormemente la calidad de vida de sus afectados [5] [6]. Es aquí donde se demuestra, como exhibiremos en este estudio y como otros estudios han relevado, la eficacia de los principios de la ingeniería de control en tratamientos de enfermedades crónicas como la fibromialgia [7], debido al claro e importante papel que el control ha demostrado ofrecer en cuanto a la eficacia en este tipo de aplicaciones (tratamientos complejos que precisen de mayor eficiencia, minimización de residuos y otros gastos, sin que por ello deje de aumentarse el número de objetivos cumplidos del tratamiento y sin dejar de mejorar la potencia de intervención) [8] [9] [10].

Buscaremos así, por tanto, realizar un estudio de la fibromialgia como sistema de control, apoyándonos en los estudios anteriormente mencionados y motivados por razones tan importantes como las que hemos enumerado anteriormente, tratando de alcanzar el método más eficaz y beneficioso para los grupos de especialistas y, sobre todo, para los afectados por la fibromialgia del presente y del futuro.

1.2 Objetivos

Para ello, estableceremos una lista de objetivos que este trabajo pretende cumplir y que, como podrá verse al final del mismo, se han conseguido alcanzar satisfactoriamente:

- 1) *Establecer una identificación clara del sistema.*
- 2) *Extraer una relación fiable entre las señales del sistema identificado.*
- 3) *Estudiar las restricciones básicas de cada señal y cómo afectan al conjunto del sistema.*
- 4) *Estudiar diversas metodologías de control que podrían ser aplicadas a nuestro caso.*
- 5) *Generar sistemas de control básicos basados en las metodologías anteriores.*
- 6) *Comparar los sistemas de control básicos con y sin restricciones.*
- 7) *Establecer un modelo más realista de nuestro sistema de control elegido, optimizando su respuesta y acercándolo a un entorno similar a la realidad.*
- 8) *Realizar pruebas de testeo y adaptabilidad de nuestro sistema a entornos realistas.*

1.3 Contenido y metodología

Con el fin de lograr estos objetivos, marcaremos un contenido más específico de cada uno de sus puntos, acompañado de una metodología que nos permita alcanzarlos de forma satisfactoria:

- 1) *Establecer una identificación clara del sistema:* exponiendo en primer lugar la necesidad de este paso, definiendo el sistema a determinar y sus elementos, diferenciando entre las diversas magnitudes que afectan a nuestro sistema, estableciendo las similitudes y disparidades existentes entre casos comunes de la ingeniería de control y un caso médico en el que se trate de aplicar los mismos conocimientos -refiriéndonos a los aspectos generales que deben tenerse en cuenta para adaptar la teoría general de la ingeniería de control a nuestra aplicación, y no aludiendo a las características particulares de nuestro sistema, cuestión que ahondaremos en el tercer punto del contenido-, para posteriormente realizar un análisis de identificación de las señales del mismo. Este punto se apoyará fundamentalmente en estudios, ensayos y artículos ya realizados por la comunidad científica, de entre cuyas propuestas elegiremos evaluaciones y resultados a partir de los cuáles erigir los cimientos de nuestro trabajo.
- 2) *Extraer una relación fiable entre las señales del sistema identificado:* definiendo y calculando posibles modelos matemáticos de relación entre las señales del sistema según diversos métodos de la ingeniería de control, estableciendo una comparación exhaustiva entre ellos con el fin de realizar una elección del más indicado para nuestro caso particular. Con el fin de alcanzar este punto, haremos uso de herramientas matemáticas de control, apoyándonos también en el programa MATLAB[®], cuyo conjunto de funciones y herramientas pueden resultar de ayuda para la definición de modelos y que será de gran utilidad para la comparación entre los mismos.
- 3) *Estudiar las restricciones básicas de cada señal y cómo afectan al conjunto del sistema:* tras encontrar el modelo más apropiado del sistema para nuestra aplicación, expondremos las limitaciones de las distintas señales, así como otros aspectos a tener en cuenta de nuestro sistema particular, que nos pueden llevar a modificar o no el conjunto de ecuaciones calculadas anteriormente para nuestro modelo, con la intención de acercar nuestras aproximaciones a un entorno más realista sin que por ello perdamos eficiencia u obtengamos resultados menos

óptimos. Usaremos para ello la información pertinente referida a las restricciones de nuestras señales, extraídas de los artículos seleccionados en el primer punto del contenido, evaluando los cambios y adaptando nuestros modelos mediante MATLAB[®] y otras herramientas matemáticas.

- 4) *Estudiar diversas metodologías de control que podrían ser aplicadas a nuestro caso:* analizar las numerosas opciones que nos da la ingeniería de control con el fin de establecer las mejores metodologías que este campo puede ofrecernos para nuestra aplicación, realizando una introducción a ellas y definiendo cuáles son los motivos que nos llevan a categorizarlas como preferibles para nuestro trabajo en particular. Para completar este objetivo, haremos uso de conocimientos que aluden al amplio campo de la ingeniería de control.
- 5) *Generar sistemas de control básicos basados en las metodologías anteriores:* una vez definidas, al término del objetivo anterior, las metodologías a partir de las cuales trabajaremos en nuestro estudio, el siguiente punto será la creación de diversos sistemas de control basados en dichas metodologías, susceptibles de constituir la mejor opción para nuestro caso, definiendo, al igual que en el apartado anterior, qué motivos nos harán clasificar los resultados de los diversos sistemas básicos de control de una metodología como preferibles o deficientes (pues no debemos olvidar que una metodología puede incluir diversos métodos de extracción de sistemas de control a evaluar). Para este paso, haremos uso de información relativa al campo de la ingeniería de control y de la herramienta MATLAB[®].
- 6) *Comparar los sistemas de control básicos, con y sin restricciones:* generando gráficas de los diferentes controles, con y sin restricciones, y analizando la información que pueda extraerse de ellas, detallando los últimos requisitos por los cuáles nos decantaremos por unos u otros métodos de control con el fin de alcanzar satisfactoriamente nuestros objetivos sin reducir por ello la cantidad de información necesaria que el sistema debe contemplar para que nuestro estudio sea un estudio adaptado a las necesidades de nuestro estudio. Para este punto, haremos uso de la herramienta MATLAB[®] y de información relativa al campo de la ingeniería de control.
- 7) *Establecer un modelo más realista de nuestro sistema de control elegido, optimizando su respuesta y acercándolo a un entorno similar a la realidad:* tras hacer selección en el punto anterior de un sistema de control que se adapte a nuestro caso de la forma más eficaz, útil y respetuosa con el paciente, trataremos de realizar una adaptación de dicho sistema a un entorno de aplicación más realista, teniendo en cuenta las interacciones médico-paciente, las citas médicas e, incluso, la relatividad de la percepción del dolor, estableciendo un ejemplo simplificado de las posibilidades que dichas opciones. Para este punto, haremos uso de la herramienta MATLAB[®] y de información relativa al campo de la ingeniería de control.
- 8) *Realizar pruebas de testeo y adaptabilidad de nuestro sistema a entornos realistas:* observando la versatilidad de nuestro trabajo, como un único código sistemático y autoadaptable, a un mayor número de posibilidades de las opciones realistas descritas en el anterior punto y no solo a un ejemplo simplificado (percepción de dolor relativa, citas médicas cíclicas y percepciones de dolor muy diferentes de las establecidas en los artículos médicos cuyos test sirven de base para la creación de nuestro trabajo). Para esto, se hará uso de la herramienta MATLAB[®].

Por supuesto, y como añadido en este trabajo, se mostrará claramente cómo realizar, paso por paso, cada uno de los métodos de control que creamos necesarios para el estudio, así como mostraremos también formas de identificar modelos relativos a conjuntos de datos reales, apoyados en el uso de la herramienta MATLAB[®].



Ilustración 2: MATLAB® & MathWorks logo. Fuente:
https://es.mathworks.com/?s_tid=gn_logo

MATLAB® (abreviatura de MATrix LABoratory, "laboratorio de matrices") es un sistema de cómputo numérico que ofrece un entorno de desarrollo integrado (IDE) con un lenguaje de programación propio (lenguaje M) [11]. En las propias palabras de los creadores de la herramienta: "Millones de ingenieros y científicos de todo el mundo utilizan MATLAB® para analizar y diseñar los sistemas y productos que transforman nuestro mundo. El lenguaje basado en matrices de MATLAB® [...] y las herramientas gráficas internas hacen que sea fácil de visualizar y obtener percepciones de los datos y conjuntos matemáticos. [...] Pudiéndose ejecutar análisis de grandes conjuntos de datos de inmensa escala. Además, el código MATLAB® se puede integrar con otros idiomas, lo que le permite implementar algoritmos y aplicaciones dentro de webs, empresas y sistemas de producción." [12]

Sin embargo, los conocimientos básicos de MATLAB® no serán explicados en esta memoria, sino solo aplicados. Existen centenares de manuales, foros y videotutoriales gratuitos y de fácil acceso que se han encargado ya de hacer y de llevar al público, de forma brillante, un trabajo como ese.

2. Modelado de la fibromialgia

*“Los cuentos de hadas son más que la verdad.
No porque nos digan que los dragones existen,
sino porque nos enseñan que los dragones se
pueden derrotar.”*

*- Coraline
Neil Gaiman*

Con la intención de alcanzar los objetivos anteriormente marcados, en búsqueda de lograr realizar un estudio de la fibromialgia como sistema de control, subdividiremos el contenido de esta memoria en dos grandes apartados, uno de modelado y otro de control.

El primer bloque que ahora presentamos se compondrá de tres puntos, siendo estos *“Identificación del sistema”* (donde definiremos completamente el sistema que pretendemos controlar), *“Relación entre las señales, modelo de la respuesta del paciente”* (donde encontraremos la relación matemática entre el tratamiento y los síntomas) y *“Restricciones de cada señal y consecuencias de las mismas”* (donde analizaremos más en profundidad los límites que debemos imponer en nuestra aplicación para alcanzar todos los objetivos de forma efectiva y útil para la sociedad).

2.1 Identificación del sistema

En nuestro esfuerzo por establecer un sistema de control a partir de una enfermedad crónica como es la fibromialgia, pasaremos por gran cantidad de puntos referentes a la ingeniería de control. Todos ellos se sustentarán en el pilar fundamental del modelo del sistema, pero antes de poder alcanzar el momento de la creación de dicho modelo será de gran importancia comprender cuál es nuestro sistema y de qué elementos se compone, puesto que no podemos pretender aplicar la ingeniería de control a algo que no podemos comprender (ya que sería similar a intentar ponerse a los mandos de un coche e intentar controlarlo sin saber la función del volante o de los pedales).

Para ello, procederemos a exponer en la siguiente figura (véase *Ilustración 3*) sendos ejemplos simplificados de la ingeniería de control (superior) y del seguimiento de un paciente crónico por su equipo médico (inferior).

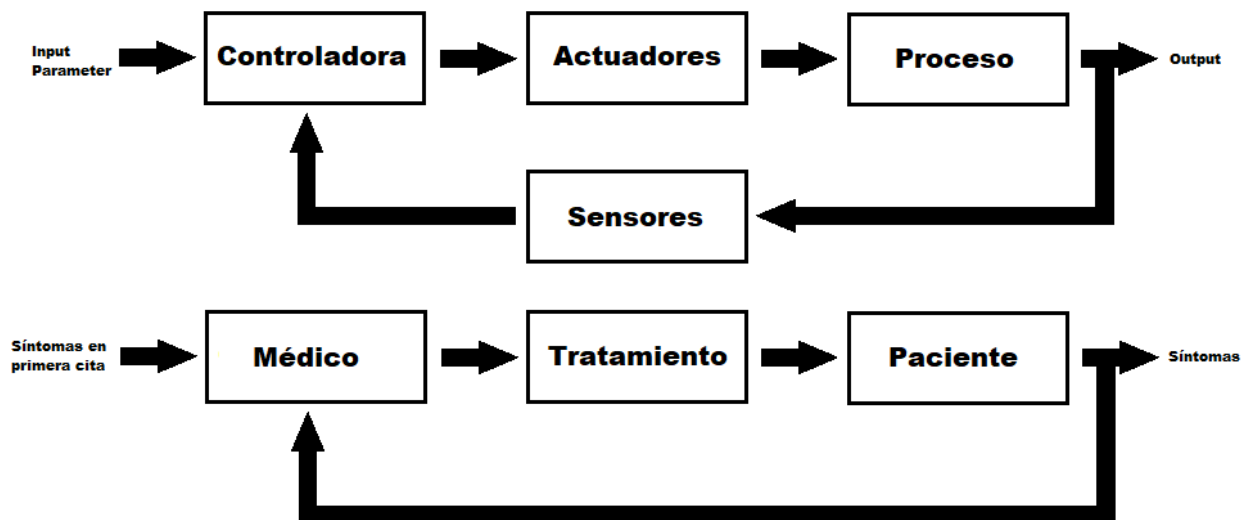


Ilustración 3: Comparación entre ejemplos básicos de la ingeniería de control y del seguimiento de un paciente

No es difícil percibir la similitud existente entre los dos esquemas, lo que deja expuesto de forma simple por qué es posible tratar de establecer un sistema de control automático en una enfermedad. Y es que es claramente sencillo ver que, en ambos casos, se precisa de unos actuadores que actúan en el proceso (o un tratamiento que afecta al paciente), de forma que las salidas del proceso (o los síntomas del paciente) se ven modificados. Esas salidas son evaluadas por un controlador mediante los sensores (al igual que un médico mide y valora los síntomas del paciente), que modificará la actuación (o el tratamiento) según convenga.

En el caso de este estudio, la intención es aunar la ingeniería de control y los procesos médicos tradicionales, con la intención de dar una atención completa al paciente que sufre fibromialgia. Eso podría suponer, como primera intención, la eliminación de la necesidad de un médico, sustituyéndolo por un conjunto de sensores y un módulo de control automático (o “controlador”).

Sin embargo, no creemos oportuno llevar a cabo esta sustitución. En primer lugar, los procesos médicos crónicos son muy sensibles y, si bien un controlador bien diseñada puede tomar un amplio abanico de decisiones, la experiencia y saberes de un especialista en medicina permiten ejecutar actuaciones diversas y, por encima de todo, permite aumentar considerablemente la capacidad de improvisación con respecto a un módulo de control automático (sin contar con que buscamos el bienestar del paciente y, actualmente, la mayoría de la población no confía en la automatización total de procesos tan sensibles). En segundo lugar, nuestra intención no es eliminar el trato del especialista con el paciente, sino apoyar al especialista en su trabajo para brindar beneficios al paciente.

Con esta segunda idea en mente, podemos plantear varias formas de abordar la parte del sistema referida a la toma de datos sobre los síntomas, la evaluación de ellos y el planteamiento de los cambios en el tratamiento.

Una primera forma de realizar este paso sería seguir usando el método médico tradicional: el especialista evalúa al paciente y modifica el tratamiento. Sin embargo, esta práctica no es de nuestro interés, puesto que queremos que el sistema ayude al médico y, por tanto, creemos en la necesidad de la introducción de un módulo de control supervisado por un especialista, de forma que se puedan aprovechar los beneficios de esta sinergia.

¿Y a qué beneficios nos referimos? Son simples de evaluar si seguimos planteando formas de definir esta parte del sistema.

Por ejemplo, otra posible propuesta podría basarse en la evaluación continua del estado del paciente. Si tenemos en cuenta que el tratamiento será de índole diario (como expondremos más adelante), ya fuese a través del equipo médico, de una aplicación con la que el paciente actualizase su estado o de una monitorización remota, podríamos lograr la obtención del nivel de los síntomas de forma diaria, siendo estos los datos necesarios para el control automático. Por tanto, una vez recibida la información, se cargaría en el sistema y este, actualizando sus datos internos, propondría los cambios pertinentes en el tratamiento para la mejora del estado del paciente.

Si bien esta opción es plausible, no será la que abordaremos en nuestro trabajo. Esta decisión se debe, fundamentalmente, a que el planteamiento anterior necesita una observación médica constante (es decir, que el enfermo acuda al especialista de manera diaria o viceversa, con el consiguiente gasto de recursos), un paciente crítico con el nivel de sus síntomas (situación claramente difícil, puesto que lo llevaría a auto-diagnosticarse, siendo sumamente complicado que lo haga de forma totalmente crítica) o la programación de un dispositivo de monitorización remota (fiable y capaz de analizar cada uno de los síntomas, tan diferentes entre sí, que atañen a la fibromialgia). Ya que no confiamos, actualmente, en el desarrollo o la posibilidad de llevar a cabo de forma fiable y rentable alguna de estas tres necesidades (además del hecho de que una actualización diaria del tratamiento puede resultar en cambios diarios en el mismo y, por tanto, en confusión para el paciente) descartamos utilizar esta opción de una carga de datos diaria para el controlador.

El siguiente planteamiento, por tanto, ya sí nos mostrará los beneficios de utilizar un módulo de control supervisado por un especialista médico, opción que tomaremos para abordar la parte del sistema referida a la toma de datos sobre los síntomas, la evaluación de ellos y el planteamiento de los cambios en el tratamiento.

Así pues, nos basaremos en el uso de dos fuentes de información. En primer lugar, tendremos en cuenta que un especialista médico es la mejor opción para la evaluación de los síntomas de la fibromialgia. Sin embargo, la evaluación médica no estará, en la mayoría de los casos, destinada a ser diaria en los pacientes de fibromialgia, sino que se dará cada cierto tiempo. Será en esos momentos, tras las citas médicas, cuando el sistema se actualice según los datos que provea el especialista, modificándose internamente la información del controlador y generándose un nuevo tratamiento adaptado a esos cambios (si esto último fuese considerado necesario por el controlador y el médico). ¿De qué servirá por tanto el apoyo del módulo de control automático? El controlador se encargará de proveer un tratamiento -aprobado por el médico en consulta el día de la cita con el paciente- que se adapte a las necesidades del paciente en los períodos entre las citas, buscando siempre la evolución más rápida y eficaz de los síntomas hacia valores que garanticen una alta calidad de vida. Para proponer este tratamiento, el controlador realizará cálculos según datos históricos, ya que no habrá una evaluación diaria del paciente. Así, el paciente se beneficiará de un tratamiento diario y eficaz basado en estudios de casos de fibromialgia (adaptado a sus necesidades particulares para mayor velocidad en la mitigación y eficacia contra los síntomas) y del asesoramiento y evaluación experta de un profesional médico durante las citas, capaz de poner en funcionamiento el módulo de control automático y valorar el tratamiento propuesto por el mismo, transcribiéndole simplemente al paciente su tratamiento, que no tendrá cambios hasta la próxima cita médica pero que contará ya con grandes garantías históricas de éxito.

Con el fin de que esto sea posible, partiremos desde, como ya hemos dicho, estudios realizados sobre la fibromialgia y sus posibles tratamientos.

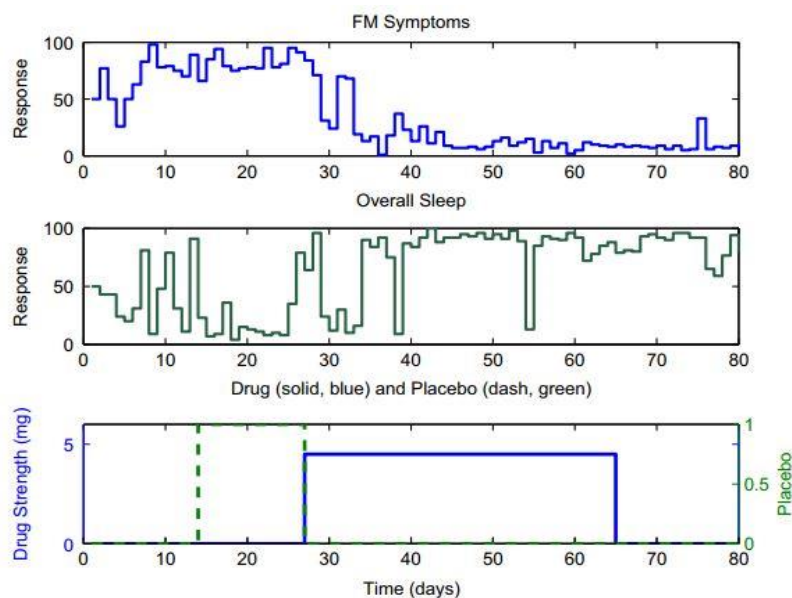


Ilustración 4: Primary self-report variables associated with naltrexone intervention of fibromyalgia, part 1 [7]

Concretamente, en nuestro trabajo analizaremos el estudio reflejado en el artículo “Optimized treatment of fibromyalgia using system identification and hybrid model predictive control” [7].

En dicho estudio, se realizan experimentos donde se evalúa el uso del medicamento “naltrexona” para tratar los síntomas de la fibromialgia, cuyos principales resultados se muestran en las siguientes ilustraciones (véase Ilustración 4 e Ilustración 5).

La naltrexona es un medicamento antagonista no selectivo de los opioides, administrado por vía oral, históricamente utilizado para la disminución de la drogodependencia a otras sustancias y con un fuerte papel neuro protector, que ha demostrado su papel como un tratamiento potencialmente efectivo contra la fibromialgia [1] [7] [13].

Ya que en el artículo anterior se demuestra la eficacia de la naltrexona, en nuestro trabajo hemos decidido adoptar el uso de este medicamento como base para el tratamiento que evaluaremos, cuya dosis será controlada mediante el modelado de los síntomas de la fibromialgia basado en los datos físicos extraídos de la ilustración anterior (cuestión que abordaremos en el próximo apartado de este trabajo), obteniendo así los datos con los que nuestro controlador podrá generar las propuestas de dosis diarias de naltrexona para los periodos entre citas.

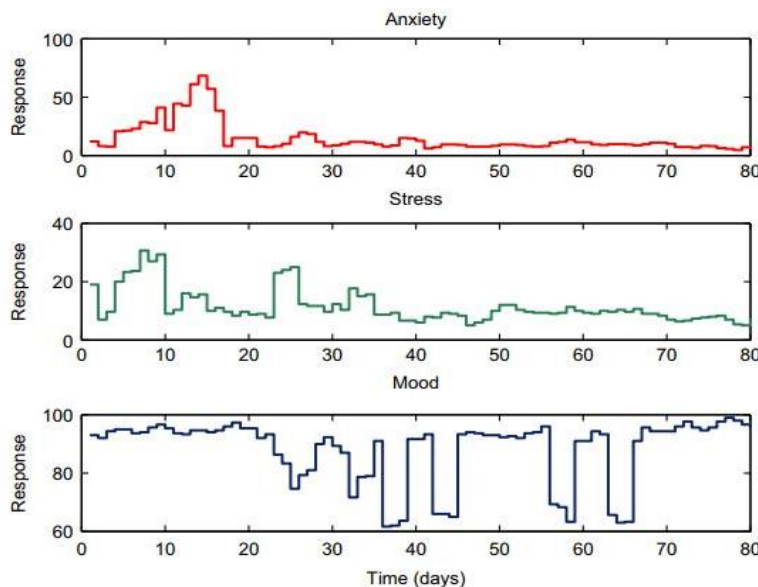


Ilustración 5: Primary self-report variables associated with naltrexone intervention of fibromyalgia, part 2 [7]

Por tanto, el sistema controlado propuesto por nuestro trabajo quedará definido con la información anterior (véase Ilustración 6). De una forma simple, podemos ver que lo que se pretende es que el médico, el día de la cita con el paciente, introduzca el estado realista del paciente en ese momento en el sistema y deje al controlador realizar cálculos. En esos cálculos tendrá en cuenta la evolución posible del paciente según el tratamiento que el programa vaya proponiendo (es decir, supondrá cómo evolucionará el paciente según datos

probabilísticos históricos cargados con anterioridad). Tras calcularse el tratamiento más óptimo, este será presentado al médico, que podrá aceptarlo o modificarlo.

Por supuesto, cabe destacar que las salidas del proceso (“Modelo del paciente de fibromialgia”) serán los síntomas probables de la fibromialgia según la simulación, que las entradas del mismo serán las dosis diarias de naltrexona generadas por el control, y que el papel del especialista médico y del módulo de control automático ya han sido definidos en este apartado.

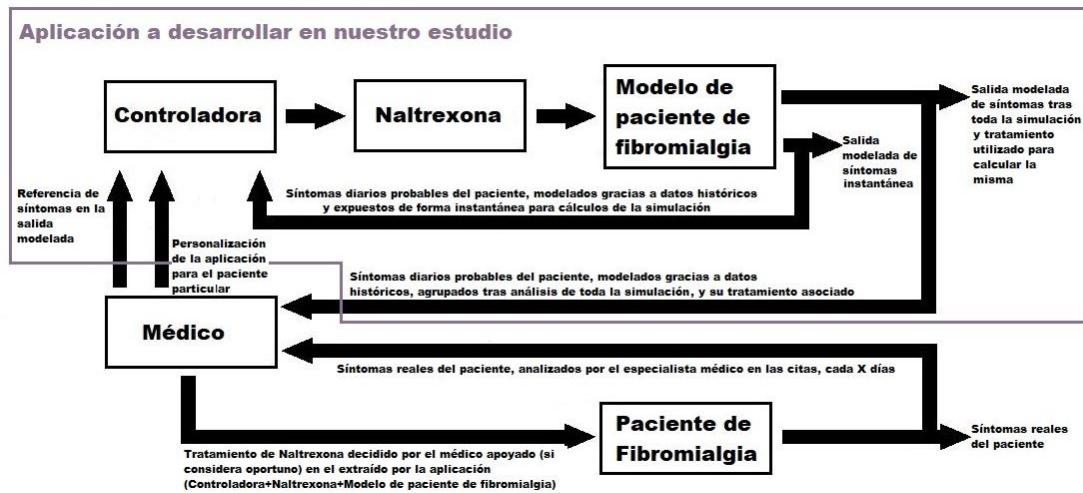


Ilustración 6: Sistema controlado propuesto en nuestro trabajo

Por último, y antes de pasar al apartado “*Relación entre las señales, modelo del sistema*”, hablaremos de los síntomas de la fibromialgia y de cómo son expuestos en el artículo anterior (y en nuestro trabajo) como un conjunto que acaba generando un único dato porcentual.

Como ya hemos comentado anteriormente, además de la hipersensibilidad en los *tender points* y el fuerte dolor musculo-esquelético en las zonas cercanas a ellos, la fibromialgia viene acompañada de otros efectos adversos tales como la fatiga, trastornos del sueño, anomalías intestinales, fuertes picos de ansiedad y desórdenes en el humor [4]. De entre estos síntomas, el menos común es la anomalía intestinal, que, aunque se ve paliada también gracias al efecto neuro protector de la naltrexona, no usaremos para modelar el nivel de síntomas de la fibromialgia en nuestro trabajo. En cambio, y como se ve en la *Ilustración 4* y en la *Ilustración 5*, sí tendremos muy en cuenta el resto de síntomas.

Curiosamente, debido a que la fibromialgia es una enfermedad que, como ya hemos comentado, aunque genera mayormente dolores musculo-esqueléticos, se debe a una afección neurológica y no por defectos en los propios músculos o el esqueleto, es aún más sensible a la percepción del dolor de cada paciente. Por ello, niveles como la ansiedad -“anxiety”-, el estrés -“stress”-, calidad de sueño -“overall sleep”- y estado anímico -“mood”- no solo se relacionan en los pacientes de fibromialgia como síntomas de la enfermedad (véase *Ilustración 4* e *Ilustración 5*), sino que también provocan que la sensibilidad hacia otros síntomas (como el dolor) varíe. Debido a esto, no solo son consecuencia de este desorden crónico, sino que también sus niveles son realimentados a la propia enfermedad y modifican las sensaciones, procesos y síntomas respecto a esta.

Así pues, para modelar esta retroalimentación, así como el dolor y la percepción del mismo, los estudios consultados sugieren reagruparlos en un único valor compuesto por la respuesta global del paciente a todos los síntomas, conocido generalmente como “Síntomas de la Fibromialgia” (véase *Ilustración 4*). Adoptar esta forma de agrupación de los síntomas permitirá generar un modelado de la respuesta del paciente a la fibromialgia de una forma más sencilla, como veremos en el próximo apartado.

2.2 Relación entre las señales, modelo de la respuesta del paciente

Una vez hemos expuesto el sistema con el cuál pretendemos estudiar la fibromialgia como un sistema controlado, podemos comenzar a recorrer los diversos pasos que la ingeniería de control necesita para apoyarnos en el éxito de nuestros objetivos.

Para ello, cada uno de los bloques expuestos en la ilustración anterior precisa de una evaluación profunda y una toma de decisiones clara, con el fin de poder ser evaluados y estudiados desde el punto de vista de la ingeniería (véase *Ilustración 6*).

Si bien las decisiones tomadas sobre el bloque designado como “Médico” se verán en el último apartado de esta memoria y el bloque “Controladora” (designado en femenino por ser “la sección de control” o “sección controladora” del sistema) se verá modificado a través de cada uno de los puntos dentro del tercer bloque, los primeros pasos de nuestro trabajo constarán de la definición de los bloques “Naltrexona” y “Modelo de paciente de Fibromialgia” en este punto 2.2. *Relación entre las señales, modelo de la respuesta del paciente* y en el sucesivo 2.3. *Restricciones de cada señal y consecuencias de las mismas*.

Como bien se dijo en el apartado anterior, plantearemos el caso de tratar a nuestro hipotético paciente de fibromialgia con dosis de naltrexona, evaluando el conjunto de síntomas que surgen de aplicar dichas dosis, buscando siempre reducir sus niveles a valores pequeños cuando sea posible, por la menor cantidad de efectos secundarios asociados. Para poder crear un sistema controlado capaz de evaluar todo el proceso, necesitaremos generar modelos matemáticos de ciertos comportamientos internos al mismo, con el fin de que el módulo de control automático sea capaz de evaluar el estado del sistema según las relaciones que conozca entre las magnitudes del mismo.

Para que pueda comprenderse de forma sencilla, incluso si el lector no ha tenido jamás contacto con la ingeniería de control, un ejemplo simple con el que poder entender esta necesidad es el de una persona en la ducha.

Supongamos que pasamos a la ducha y abrimos la manija del agua a cierta temperatura. La notamos fría, así que giramos la manija un cierto número de grados hasta que el agua se vuelve caliente, pudiendo reajustarla luego como necesitemos.

Si este proceso no lo hiciese un ser racional, sino que la manija estuviera automatizada y el plato de la ducha tuviera un sensor de temperatura, al encender el circuito y pedir cierta temperatura, el agua saldría fría, el sensor le diría al controlador la temperatura del agua que está cayendo al plato de ducha, y él giraría la manija el número de grados necesario para que el agua saliese a la temperatura requerida, reajustándolo como fuese necesario.

En el caso de una persona, reajustar la manija a un ángulo requerido es bastante sencillo; pero para un controlador, encontrar el ángulo que dota al agua de la temperatura ideal requiere de una serie de cálculos, y debe tener información en su interior sobre cómo se relacionan la temperatura del agua y la cantidad de grados del ángulo de la manija.

Esa relación es lo que se conoce como “modelo”, y se define en la ingeniería de control como el conjunto de ecuaciones matemáticas que representan el comportamiento de un proceso físico. Por supuesto, siempre existen diferencias entre el comportamiento del proceso real y el del modelo (denominadas “incertidumbres”), por lo que, cuanto más sensible sea la aplicación -en nuestro caso, una aplicación médica es normalmente muy sensible- más debe tratar el ingeniero de hacer que su modelo responda de forma más parecida al proceso real.

Concretamente, y en lo que a nosotros atañe, la cantidad de naltrexona administrada es equivalente al ángulo de giro de la manija del agua, y los síntomas que muestra el paciente al suministrarle dicha dosis responden como lo hacía la temperatura del agua en el proceso anterior.

Por ello, necesitaremos establecer un *modelo de la respuesta del paciente según la dosis de naltrexona*, con la intención de que nuestro futuro módulo de control pueda conocer la relación matemática entre el tratamiento y los síntomas del paciente.

Además (y debido a lo que ya explicamos en el punto anterior de esta memoria), como el acceso a los síntomas reales del paciente solo va a ser posible en las citas médicas, que se darán cada X días, este modelado cumplirá también la función de dejar conocer al sistema las relaciones históricas de los pacientes de fibromialgia ante el tratamiento de naltrexona. Debido a esta característica, y para poder definir el modelo de respuesta del paciente, haremos uso de la información extraída del artículo "*Optimized treatment of fibromyalgia using system identification and hybrid model predictive control*" [7], donde se relaciona la magnitud "Síntoma de la fibromialgia" con la dosis de naltrexona tomada por los pacientes del primer experimento de dicho estudio (véase *Ilustración 4*).

Para llevar a cabo el modelado de dicha relación, haremos uso de las herramientas matemáticas de MATLAB[®], y generaremos varios modelos hasta encontrar aquel que mejor se ajuste a nuestro caso.

Antes de continuar, definiremos el término "función de transferencia", ya que será un concepto que usaremos en numerosas ocasiones como término con el que referirnos a ese modelo que intentamos crear.

Una función de transferencia es un modelo matemático que a través de un cociente relaciona la respuesta de un sistema con una señal de entrada o excitación [14]. Por consiguiente, el modelo que tratamos de hallar, con forma de cociente y que relaciona la respuesta del paciente (es decir, sus síntomas) con el tratamiento que se le suministra, es una función de transferencia.

El único motivo por el cual el modelo que usaremos será un cociente se deberá a la herramienta matemática con la cual crearemos ese modelo, que será la "Transformada de Laplace". Existen muchos tipos de modelos en la ingeniería de control (otro, por ejemplo, es el de mínimos cuadrados), siempre pudiendo convertirse, gracias a MATLAB[®], de un tipo de modelos a otros. Por su simplicidad a la hora de comprensión, extracción e implementación, trataremos siempre de hallar funciones de transferencia o, en su defecto, realizar transformaciones a modelos del tipo de mínimos cuadrados a funciones de transferencia de la Transformada de Laplace.

Según la ingeniería de control, la función de transferencia de un sistema lineal e invariante en el tiempo (LTI) es el cociente entre la transformada de Laplace de la salida y la transformada de Laplace de la entrada, siempre y cuando las condiciones iniciales sean nulas. En nuestro caso concreto, aunque el sistema del experimento que analizaremos a continuación no sea un sistema lineal ni invariante en el tiempo, lo trataremos como tal por diversos motivos. En primer lugar, si bien los síntomas de una afección crónica no afectan durante todos los días del año de la misma forma a un enfermo, llegando incluso a padecerse agudos picos sintomáticos y al día siguiente no presentar apenas ninguno, incluso sin tratamiento o con tratamiento constante, siempre y cuando nos posicionemos en un margen de seguridad capaz de tratar los peores niveles medios de síntomas en cada momento y tengamos en cuenta que nuestro sistema contará dentro de unos apartados con un método dinámico de supervisión médica, podremos cometer la imprecisión de tratarlo como un sistema lineal. Por otro lado, y de igual manera que con el motivo anterior, si bien es verdad que el paciente puede demostrar una mejoría constante gracias al tratamiento, incluso pasando grandes temporadas sin él sin un crecimiento necesariamente elevado de síntomas, por la naturaleza de la fibromialgia, crónica y sin cura, la enfermedad tenderá a volver a su punto inicial tras un tiempo sin medicación; no obstante, sí puede variar en el tiempo, pero si tratamos la sintomatología con los datos medios de la población teniendo en cuenta, igual que antes, que contaremos con un sistema de supervisión por parte del especialista que podrá adaptar de forma sencilla estas varianzas temporales a posteriori, es posible aproximar el modelo de respuesta del paciente al tratamiento como invariante en el tiempo y, debido al motivo anterior, como lineal.

Con todo esto, podemos tratar ya de hallar nuestro modelo.

Para ello, comenzaremos mostrando los datos a partir de los cuales extraeremos la información necesaria. Como ya se vio en ilustraciones anteriores, nuestra intención es relacionar el tratamiento de naltrexona al que se ajusta un paciente con los niveles sintomáticos de fibromialgia que se miden durante el tiempo que dura el tratamiento. Aislando, por claridad, los datos anteriormente presentados, mostraremos la siguiente gráfica (véase *Ilustración 4* e *Ilustración 7*) donde se ve de forma clara la relación entre el tratamiento y los síntomas. Este proceso, que se debe a la respuesta global que los pacientes mostraron al tratamiento descrito, será el experimento por el cual nos guiaremos en la búsqueda de nuestro modelo.

Para crear dichas gráficas, hemos cargado los datos obtenidos en dos matrices de MATLAB®, copiando con cuidado cada uno de los valores de las mismas. En nuestro caso, generamos las matrices “Naltrexona” y “Sintoma_origen”, que posteriormente utilizamos para crear dichas gráficas con el comando “plot” aplicado de la siguiente forma:

```
subplot(2,1,1); plot(tiempo, Sintoma_origen, 'LineWidth', 2); grid;
xlabel('t(dias)'); ylabel('Sintomas (%)'); title('Sintomas FM'); legend('Experimento', 'Location', 'Best');
```

```
subplot(2,1,2); plot(t, Naltrexona, 'LineWidth', 2); grid; xlabel('t(dias)');
ylabel('Naltrexona (mg)'); title('Dosis de naltrexona'); legend('Naltrexona', 'Location', 'Best');
```

De igual manera, las matrices “Naltrexona” y “Sintoma_origen”, ya introducidas en el sistema, nos servirán para extraer las relaciones que necesitamos de forma analítica.

Para ello, comenzaremos apoyándonos en la herramienta “systemIdentification” de MATLAB®.

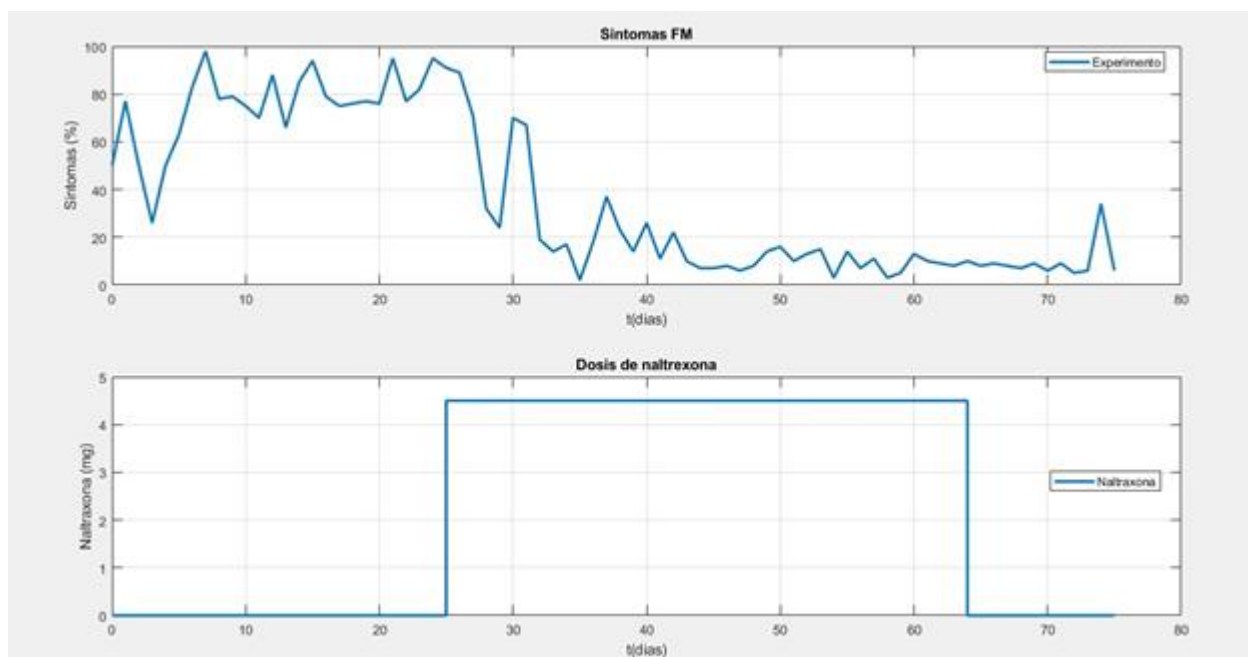


Ilustración 7: Datos experimentales, respuesta global del estudio, pacientes de fibromialgia frente a un tratamiento a la naltrexona.

2.2.1 La herramienta systemIdentification, modelos por mínimos cuadrados

La System Identification Toolbox™ es una herramienta que provee funciones de MATLAB®, bloques Simulink® y una aplicación para la construcción matemática de modelos basados en sistemas dinámicos a partir de datos entrada-salida [15].

Para hacer uso de la aplicación que nos trae la herramienta, con la cual trataremos de hallar el modelo que buscamos, solo necesitamos introducir en comando “systemIdentification” en la ventana de comandos.

Una vez introducido, se nos abrirá la pantalla inicial de la aplicación (véase *Ilustración 8*)

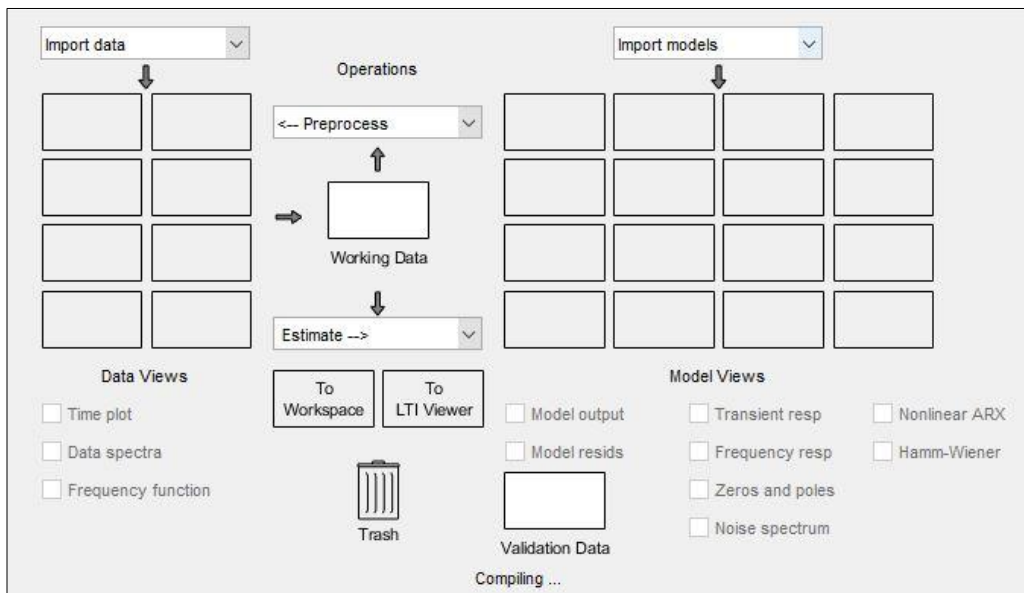


Ilustración 8: Pantalla de inicio de la aplicación systemIdentification

En dicha pantalla, pinchando en el desplegable “Import Data” y eligiendo “Time domain data”, se nos abrirá una ventana donde podremos escribir el nombre de nuestras matrices dato, siendo “Naltrexona” el “Input” y “Síntoma_ origen” el “Output”. Pinchando en “More” podremos introducir las magnitudes de nuestras variables, así como decidir nombres para los “canales” donde se cargarán las variables introducidas.

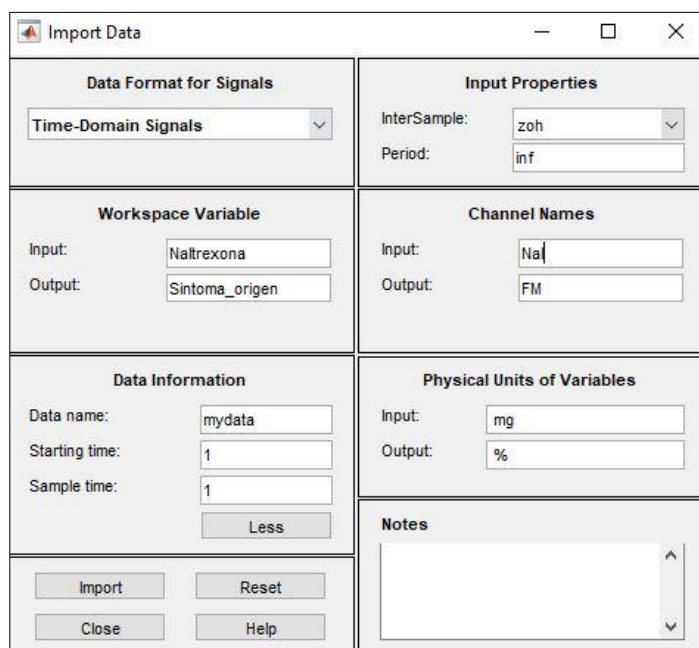


Ilustración 9: Ventana de "Import", botón "more" pulsado.

Será importante decidir adecuadamente el nombre del canal y las magnitudes (o podemos dejar dichos campos vacíos, con lo cual la aplicación se encargará de asociarles valores predeterminados) debido a que, una vez generemos modelos a partir de dichos datos, solo podrán contrastarse con aquellos grupos de datos con los mismos canales y magnitudes.

El resto de campos los dejaremos en blanco y pulsaremos “Import” (véase *Ilustración 9*).

Tras esto, aparecerán nuestros datos bajo el desplegable “Import data” de la ventana principal en forma de gráfica con el nombre que hayamos marcado en el cuadro “Data name” de la ventana “Import data” (véase *Ilustración 9*).

Pinchando ahora en la gráfica que se ha creado tras la importación, podremos llevar a cabo varios procesos. En el desplegable “Preprocess” que aparece en la columna titulada con el nombre “Operations” en la ventana principal (véase *Ilustración 8*), podremos llevar a cabo varios preprocesados antes de estimar nuestro modelo. En el desplegable “Estimate” que aparece en la columna “Operations” en la ventana principal (véase *Ilustración 8*) podremos elegir el tipo de estimación que usaremos para que la aplicación halle el modelo que queramos a partir de los datos introducidos.

En nuestro caso, en dicho desplegable seleccionaremos “Polynomial models”, ya que pretendemos hacer una estimación por mínimos cuadrados, y se nos abrirá una ventana con dicho nombre.

“Mínimos cuadrados” es una técnica de análisis numérico en la que, dados un conjunto de parejas de datos (variable independiente y variable dependiente relacionadas entre sí, como es nuestro caso) y una familia de funciones, se intenta encontrar la función continua, dentro de dicha familia, que mejor se aproxime a los datos (un "mejor ajuste"), de acuerdo con el criterio de mínimo error cuadrático - o mínimo valor del estimador que mide el promedio de los errores al cuadrado (es decir, la diferencia entre el estimador y lo estimado) [16].

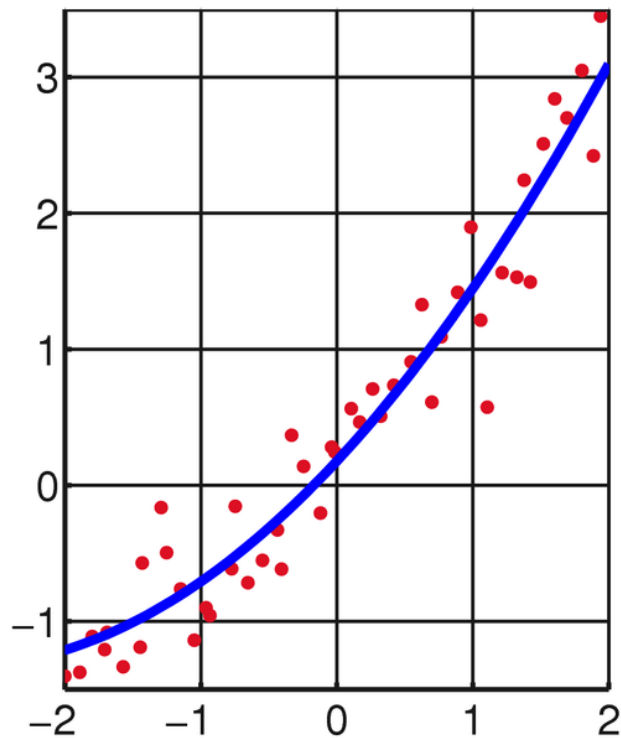


Ilustración 10: Resultado del ajuste de un conjunto de datos a una función cuadrática; en rojo pueden verse los datos iniciales, en azul, la curva tendencia por mínimos cuadrados. Fuente: https://es.m.wikipedia.org/wiki/Archivo:Linear_least_squares2.png

Para exponerlo de forma simple, trataremos de reproducir, para nuestro experimento, lo que se muestra en la imagen explicativa (véase *Ilustración 10*).

Tras seleccionar el orden que queramos, ([2 2 1] o [4 4 1] estarán bien), podremos seleccionar “Estimate”.

Una vez hecho esto, aparecerá en la ventana principal, bajo el desplegable “Import model”, el modelo estimado que acabamos de crear (véase *Ilustración 8*).

La herramienta permite, además, una vez estimado el modelo, comprobar su correspondencia con el conjunto de datos inicial. Para ello, arrastraremos la gráfica de datos sobre la casilla “Validation data”, y pinchando sobre la gráfica de nuestro modelo que se encuentra bajo “Import model”, así como en la casilla “Model output”, nos aparecerá una gráfica comparativa donde nos dirá el porcentaje de cercanía entre el modelo y el experimento original.

En el caso que concierne a este proyecto, no resultó en un porcentaje alto (apenas un 45%), debido a ello, creímos conveniente preprocesar la señal antes de generar el modelo, así como probar todas las opciones que la pestaña “Polynomial models”. Tras muchos intentos que no mostraremos, puesto que terminaríamos generando una memoria tediosa y lo consideramos innecesario (ya que se basa en un proceso repetitivo de preprocesar, probar, modificar, probar y volver a preprocesar y así seguir cíclicamente), apenas nos vimos con un porcentaje del 55%.

Para intentar mejorar dicho valor y tratar de extraer un modelo más fiable (es decir, más parecido al experimento real y, por tanto, con menor cantidad de desviación frente al estado real de un posible

paciente), realizamos un remodelado analítico de los datos.

Debido a que la herramienta estima el modelo a partir de los datos, ciertas partes del experimento, que podrían considerarse como “ruidosas”, podrían estar afectando negativamente a la estimación del modelo.

Concretamente, en nuestro experimento, en los primeros días del mismo, pueden verse ciertas bajadas bruscas que no parecen corresponderse a un periodo de no-tratamiento (véase *Ilustración 11*). Como bien dijimos en apartados anteriores, debido a la no-linealidad del modelo y a que el modelado de una enfermedad no trata de establecer una buena estimación en aquellos extraños días donde estos sucesos ocurren (días donde, afortunadamente, el paciente se encuentra mejor aún sin tratamiento), podríamos extraer estos primeros días de los datos, para tratar de evitar que la aplicación modele teniendo en cuenta este carácter dinámico que posteriormente solventaremos con el método para las supervisiones médicas que implementaremos en apartados posteriores.

De igual manera, el pico que se da en torno al día 75 también podría ser categorizado como un suceso del mismo tipo que el anterior, por lo que podría ser tratado de la misma forma.

No obstante, no trataremos de eliminar directamente dichas partes, sino que estudiaremos los casos según estén o no incluidas para tenerlas en cuenta en el modelado mediante un proceso igual que el anterior (preprocesar, probar, modificar, probar y así cíclicamente).

Con ello, conseguimos extraer dos posibles candidatos con un porcentaje de más del 60% (véase *Ilustración 13*), de entre los cuáles elegiremos “far3sin10” como el mejor de ambos, por tener un poco más de valor en la correspondencia y, por tanto, menos desviación respecto del experimento original (como vemos en la *Ilustración 13*, no hemos eliminado para hallar dichos modelos la zona experimental que aparece en verde en la *Ilustración 11*).

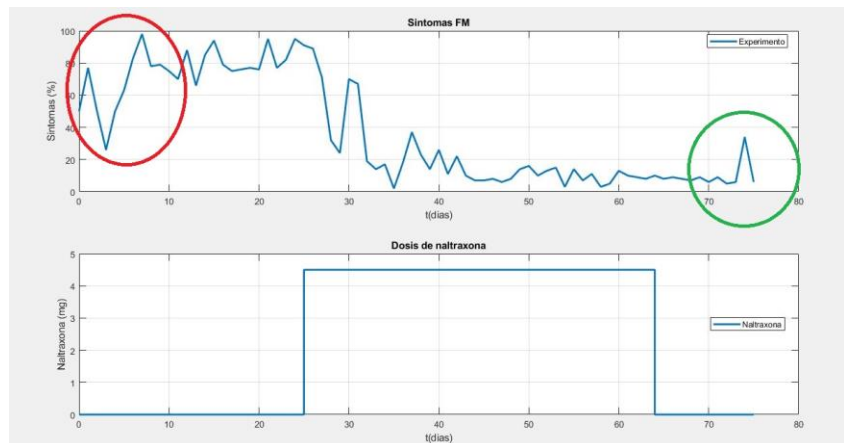


Ilustración 11: Ilustración 7, con posibles zonas ruidosas remarcadas.

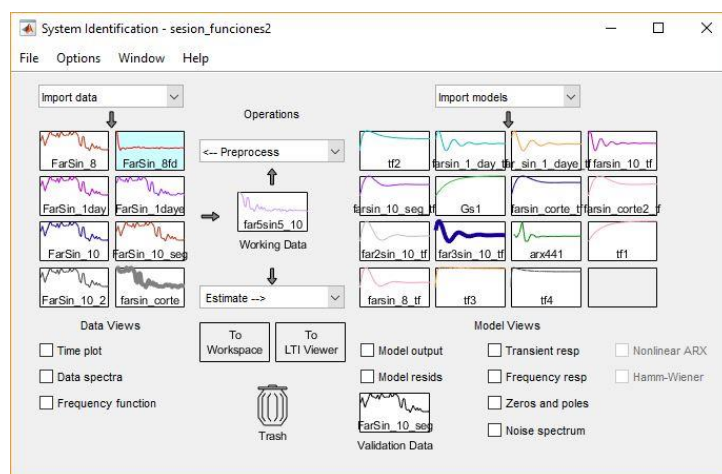


Ilustración 12: Algunos de los modelos creados y comprobados para la generación del modelo de respuesta del paciente.

Una vez elegido nuestro modelo, nos quedará comprobar su buen funcionamiento en el entorno de simulación donde hallaremos los controladores pertinentes.

Para ello, utilizando la propia herramienta System Identification Toolbox™, transformaremos el modelo arx en una función de transferencia en la aplicación, para luego seleccionar el modelo en forma de función de transferencia, pinchando dos veces en su gráfica de la columna “Import model”, gracias a lo cual se nos abrirá una pestaña con los detalles del modelo, en la cual

pincharemos “Export” en la zona inferior. Con ello, la función de transferencia pasará a estar en el Workspace de MATLAB® junto con “Naltrexona” y “Síntoma_origen”.

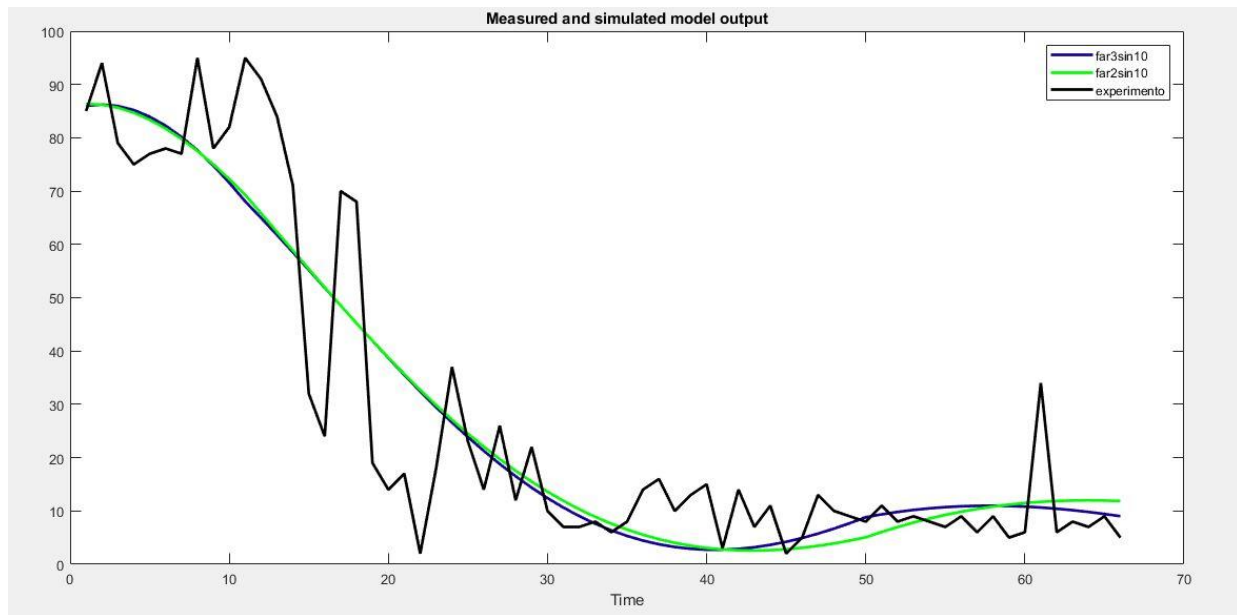


Ilustración 13: Respuesta de los modelos far3sin10 y far2sin10 frente al experimento original.

Así, la función de transferencia hallada habrá sido:

$$Far3sin10 = \frac{-(0.1317 \cdot s + 0.03938)}{s^2 + 0.03947 \cdot s + 0.006045}$$

Recordemos, por supuesto, que dicha “s” nos indica que estamos usando la Transformada de Laplace como herramienta matemática de apoyo para hallar e implementar nuestros modelos.

Con el fin de probar si, efectivamente, el modelo es apto para la puesta en marcha del control que queremos crear, haremos uso de la herramienta Simulink®.

La herramienta Simulink® es un entorno de programación visual, ejecutado sobre la base de MATLAB® y su lenguaje, que permite la simulación de multitud de procesos de ámbitos muy diferentes de forma gráfica e intuitiva mediante la unión de bloques, y en el cual se pueden introducir multitud de herramientas (que pueden ser tanto creadas por el propio equipo de MATLAB®, por empresas o usuarios particulares) para el apoyo a la simulación de procesos específicos.

En nuestro caso, para probar el modelo “far3sin10”, usaremos estos "bloques" de lenguaje Simulink® para introducir en el modelado de respuesta del paciente la misma entrada que tenía el experimento (es decir, la dosis diaria de naltrexona de la Ilustración 7) y comprobar si a su salida vemos unos síntomas de fibromialgia que se asemejen a los del experimento que hemos modelado.

Para ello, crearemos el siguiente montaje en Simulink® (véase Ilustración 14).

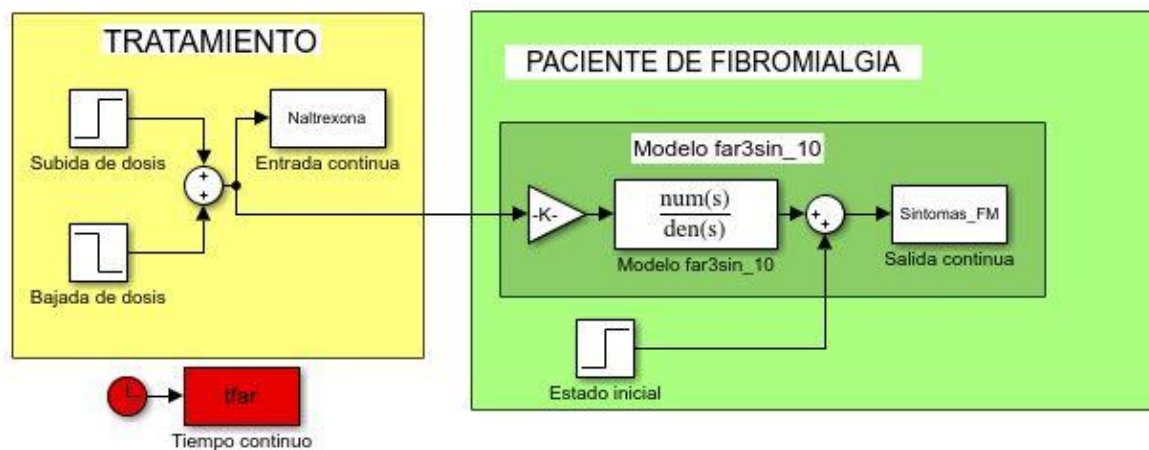


Ilustración 14: Montaje en Simulink® para la comprobación del modelo far3sin10 en el entorno de simulación donde comprobaremos futuros módulos de control.

En primer lugar, explicaremos cómo hemos diseñado el "Tratamiento" que se ve expuesto en la imagen superior. Hemos creado dos señales de escalón, una positiva (de 0 a 4.5) y otra negativa (de 4.5 a 0) que se suman; la positiva se activa (es decir, hace el escalón positivo) el día 25, de forma que el tratamiento de 4.5 mg comienza ese día (ya que es cuando comenzó el tratamiento en el experimento) y la negativa se activa el día 64, con lo que el tratamiento se anula a partir de ese momento (al igual que los datos a partir de los cuáles hemos hallado el modelo).

Estas señales, al sumarse y conformar el tratamiento, se almacenan como una única señal sobre "Naltrexona", la matriz de datos que teníamos anteriormente; para ello, hemos usado un bloque "To Workspace" marcado internamente (pinchando dos veces en él) como "Array". Ya que no hemos tocado nada sobre el "Sample Time" de este bloque, se almacenarán los datos como señales continuas.

Igual que hemos usado un bloque "To Workspace" para almacenar "Naltrexona", se hará lo mismo para almacenar el tiempo de simulación ("tfar", que irá conectado a un "clock" con sus parámetros predeterminados) y la salida de los síntomas ("Sintomas_FM").

Para comprobar la respuesta del paciente según nuestro modelo calculado, usaremos varios bloques que, en conjunto, nos darán a su salida los síntomas equivalentes al tratamiento entrante.

Con este fin, se ha utilizado un bloque "Gain" (marcado con una "K") o "ganancia", que multiplica a la señal que pasa a través de él por cierto valor. Esta ganancia puede ser útil para ajustar mejor la respuesta del permanente del modelo frente a la realidad (es decir, reducir el error constante que puede darse en un proceso modelado frente al proceso real en momentos de estabilidad -en -este caso, equivalente a los días a partir del día, aproximadamente, 53, tras lo cual el sistema ha atravesado un estado transitorio y se mantiene constante en un valor medio de bajo error en el experimento, concretamente, en torno al 8% de síntomas presentados).

En principio, el valor de esta "K" será 1, de forma que la señal que entra en dicho bloque será la misma que sale, y que entrará de forma pura en el modelo calculado, el cual se espera que nos dé una respuesta aproximada parecida al experimento y que, en el permanente, alcance ese 8% tras los 28 días (aproximados) que dura el periodo transitorio del experimento, al cual también debería ajustarse con el menor error posible.

Para terminar de implementar el modelo que hemos creado, usaremos un bloque "Transfer function", el cual renombraremos como "Modelo far3sin10" y en el que introduciremos la función de transferencia con el mismo nombre. A este le añadimos el efecto de un bloque escalón (pudiera ser un

constante en vez de un escalón, ya que se encontrará activo desde el día cero). Este bloque se encargará de situar la función de transferencia en el punto de trabajo que se crea necesario.

Para explicar la necesidad de este bloque, recordemos que la función de transferencia, según su propia definición, precisa de valores iniciales nulos. Esto es, que las funciones de transferencia deben sus valores a relaciones entre parejas de datos y, por tanto, miden la evolución de los datos y las variaciones de los mismos. Es decir, que, a efectos de la propia función de transferencia, si se varía a la señal de entrada un valor arbitrario n , a la salida del modelo se variaría un valor m de síntomas relacionado con n , independientemente de si, inicialmente, el paciente tiene un, por ejemplo, 80% de síntomas o, por ejemplo, un 40%. Por ello, necesitamos un bloque que se encargue de especificar dicho valor inicial. En nuestro caso, será el que acabamos de añadir, y su valor pondrá al sistema en un valor inicial del 84%, el valor medio al cuál se encuentra los síntomas del paciente del experimento en el inicio del mismo.

La intención, por tanto, será comprobar si el modelo se ajusta a los valores medios del experimento, dándole como el tratamiento el mismo que se aplicó en dicho estudio (de 0 a 4.5mg), comprobando que se llega a un 8% de síntomas en aproximadamente 28 días.

Con ello, el Simulink[®], estableceremos el tiempo de simulación en 75 días, pulsaremos el botón "Run" y representaremos los datos que se obtienen tras la simulación mediante:

```
subplot(2,1,1); plot(tfar, SintomasFM, tiempo, Sintomaorigen, 'LineWidth', 2); grid;
xlabel('t(dias)'); ylabel('Sintomas (%)'); title('Sintomas FM');
legend('Modelo far3sin10', 'Experimento', 'Location', 'Best');
```

```
subplot(2,1,2); plot(tfar, Naltrexona, 'LineWidth', 2); grid;
xlabel('t(dias)'); ylabel('Naltrexona (mg)'); title('Dosis de naltrexona');
legend('Naltrexona', 'Location', 'Best');
```

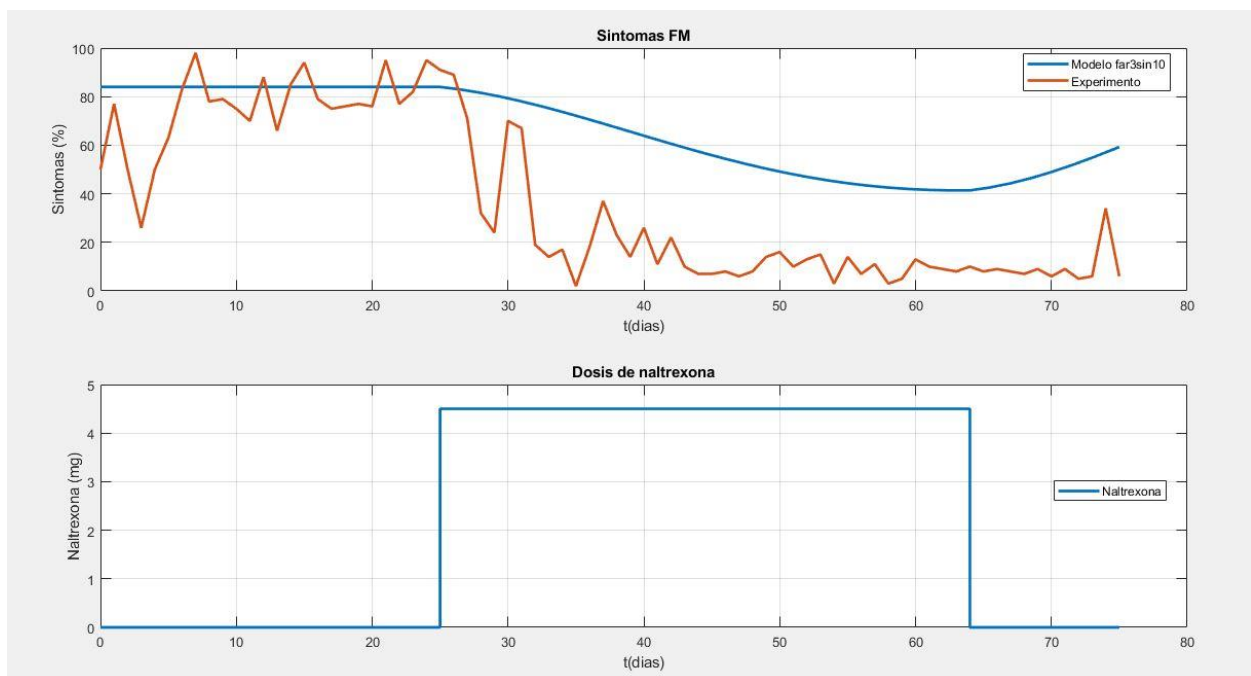


Ilustración 15: Far3sin10 frente a experimento, $K=1$.

Como podemos comprobar en la ilustración anterior (véase *Ilustración 15*), a pesar de que en la herramienta System Identification Toolbox[™] nos indicaba que se alcanzaría el mismo valor, de produce una gran diferencia.

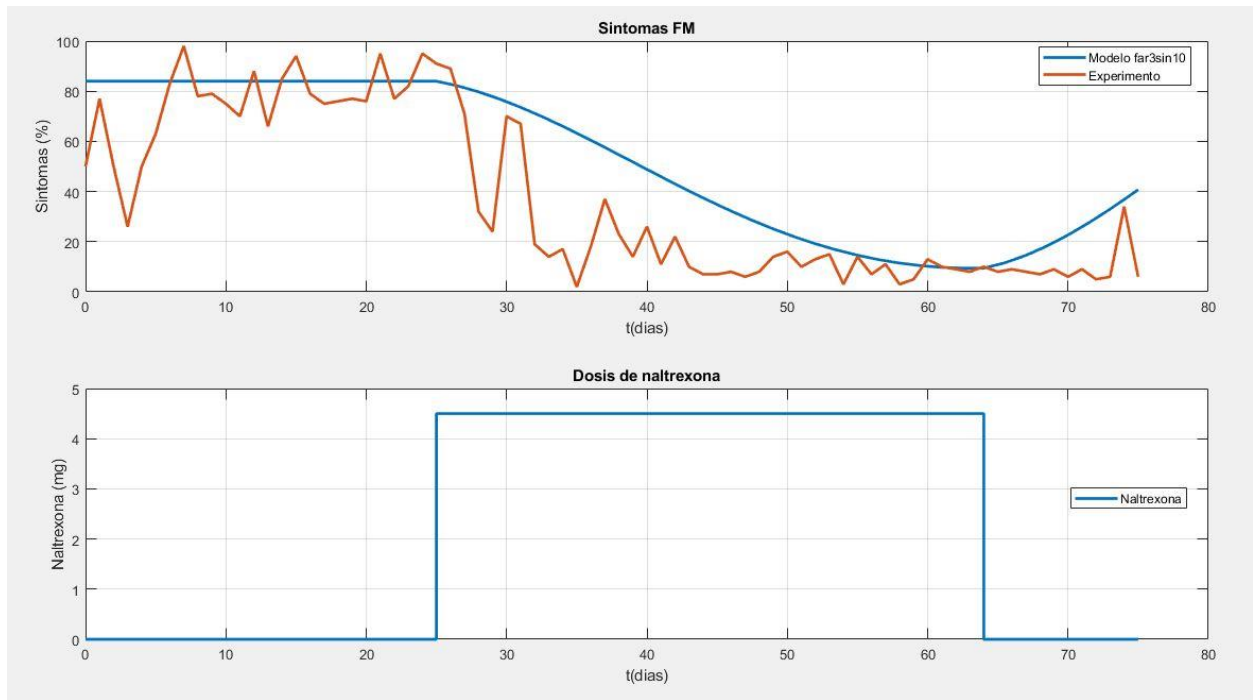


Ilustración 16: Far3sin10 frente a experimento, $K=1.75$

Para disminuirla, podemos aumentar la ganancia del modelo (valor de K). Con un cálculo sencillo se ve que, si la aumentamos a 1.75, al suministrar de forma constante 4.5mg, se alcanzará el 8% requerido.

No obstante, si bien este cálculo puede hacernos creer que el sistema va a funcionar correctamente, aumentar la ganancia del modelo de respuesta del paciente puede provocar efectos indeseados (véase *Ilustración 17*), como un modelado falso de efecto rebote tras dejar repentinamente el tratamiento.

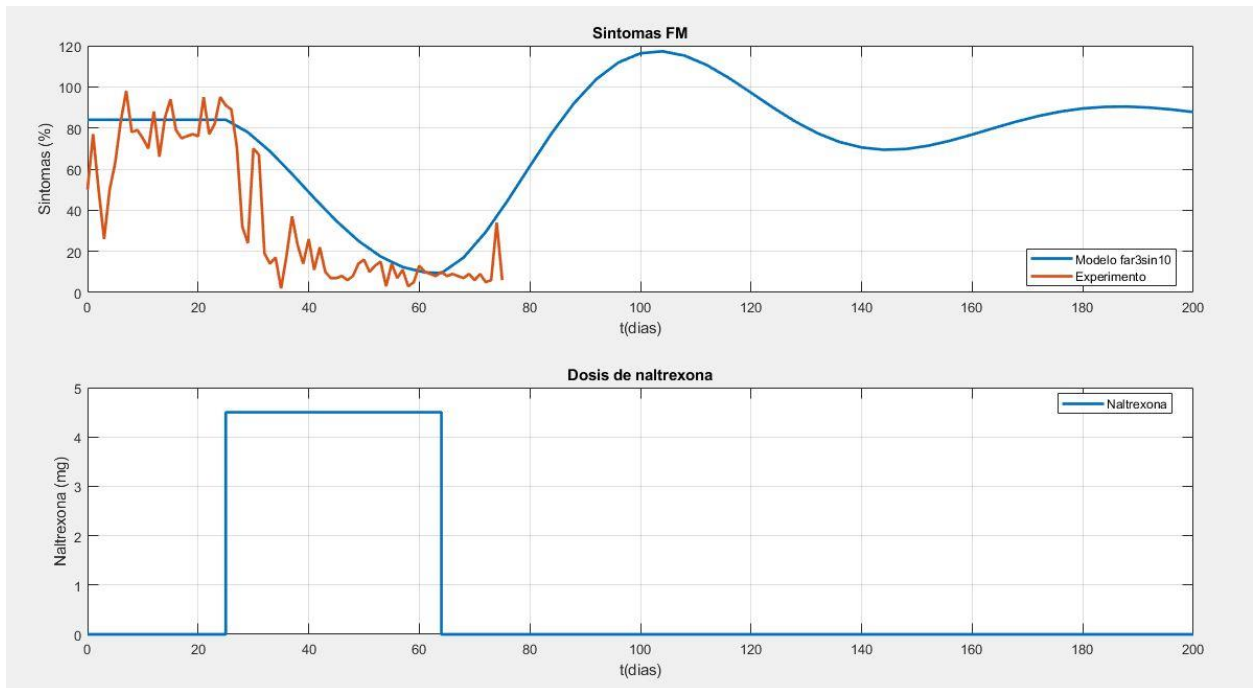


Ilustración 17: Far3sin10 frente a experimento, $K=1.75$, datos de 200 días

Esto podría deberse a que el modelado no estuviera correctamente hecho.

Volviendo a la aplicación de System Identification Toolbox™, podríamos extraer más información sobre por qué ocurre esto.

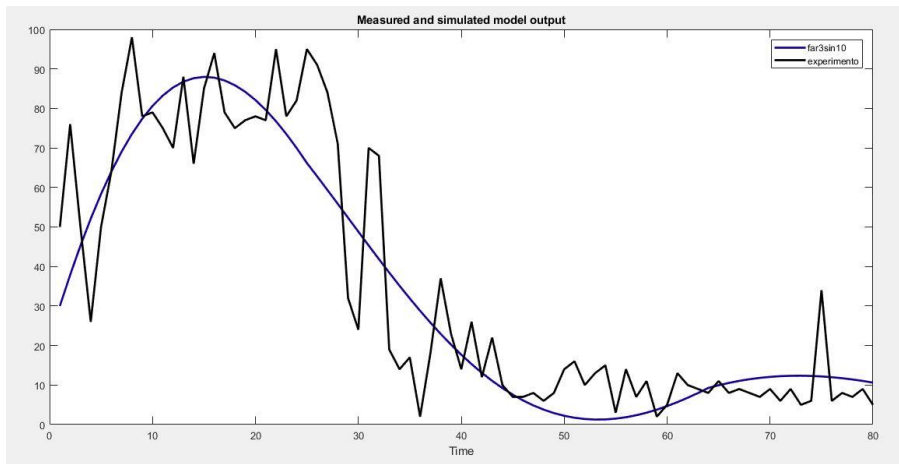


Ilustración 18: Far3sin10 frente al experimento completo. Reacciona a lo que consideramos como “ruido inicial” al generar el modelo, a pesar de haberlo eliminado en el preprocesado.

A pesar de tratar el experimento con la intención de mejorar la identificación automática del modelo, no hemos conseguido eliminar efectos indeseados que pudieran llevar a la creación de modelos muy dispares respecto a la realidad.

Prueba de ello es que, debido a la alta desviación que hay en el experimento del que partimos entre los valores

medios del mismo y los reales en cada día, la aplicación genera una función curva que no se aproxima a lo que buscamos, ya que necesitamos que tenga un carácter más lineal (ya que, como dijimos, la no-linealidad la implementaremos más tarde).

Debido a ello, y tratando de conseguir un modelo más con mayor fiabilidad para nuestra aplicación, pasaremos a generar funciones de transferencia mediante métodos gráficos.

2.2.2 Funciones de transferencia por métodos gráficos.

Para crear funciones de transferencia mediante métodos gráficos, solo necesitamos comprender ciertos datos que necesitamos extraer de las gráficas que muestran el experimento real.

Con este fin, lo primero será seleccionar un tipo de función de transferencia. Existen multitud de tipos de funciones de transferencia que podríamos aplicar, aunque por su simplicidad y buen resultado en multitud de casos, buscaremos hallar de primer y segundo orden.

2.2.2.1 Función de transferencia de primer orden

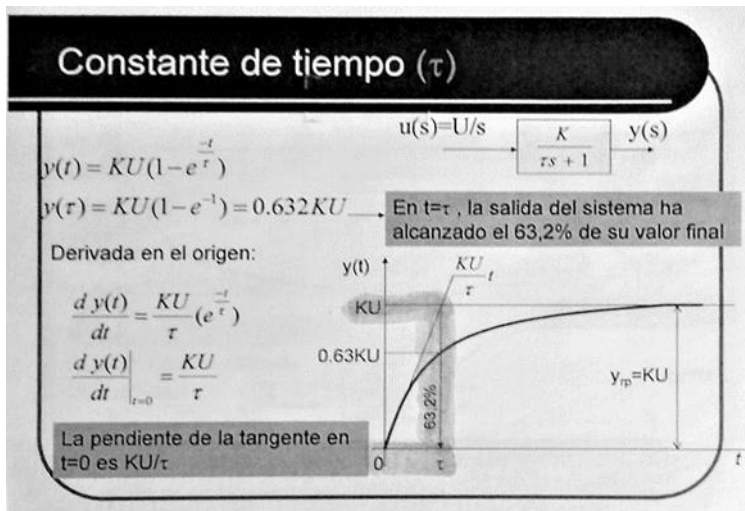
Una función de transferencia de primer orden corresponde a la siguiente formulación:

$$G_{s1} = \frac{K}{(\tau \cdot s + 1)}$$

Cuyos parámetros pueden extraerse aplicando los conocimientos extraídos de la siguiente imagen sobre nuestros datos experimentales (véase imagen *Ilustración 19* -más detalles pueden ser encontrados en cualquier manual sobre ingeniería de control que trate funciones de transferencia continuas, al igual que puede conocerse más sobre la Transformada de Laplace en cualquiera de ellos, ya que es usada para tratar este tipo de funciones).

Y de dónde podemos saber que:

$$K = \frac{\Delta y}{\Delta u} = \frac{\text{Variación de la señal de salida}}{\text{Variación de la señal de entrada}} = \frac{\text{Variación de los síntomas}}{\text{Variación del tratamiento}}$$



Una vez extraídos los parámetros, usaremos un montaje Simulink® igual que el anterior (véase *Ilustración 19*, donde debemos cerciorarnos de volver a poner el bloque “Gain” a valor 1 (no debemos confundir dicho valor K con la K de la función de transferencia de primer orden; ambas son ganancias, pero son diferentes) y cambiar el modelo far3sin10 por Gs1.

Es común que, tras mostrar los datos, el valor de τ sea ligeramente modificado para un mejor ajuste.

Ilustración 19: Función de transferencia de primer orden y parámetros de la misma [17]

En nuestro caso concreto, el modelo quedará definitivamente como:

$$Gs1 = - \frac{16.888}{(7 \cdot s + 1)}$$

Modelo del cuál hemos comprobado un correcto funcionamiento, y que se ajusta a las especificaciones requeridas por nuestro sistema, sirviendo como aproximación del valor medio de los valores del experimento (véase *Ilustración 20*).

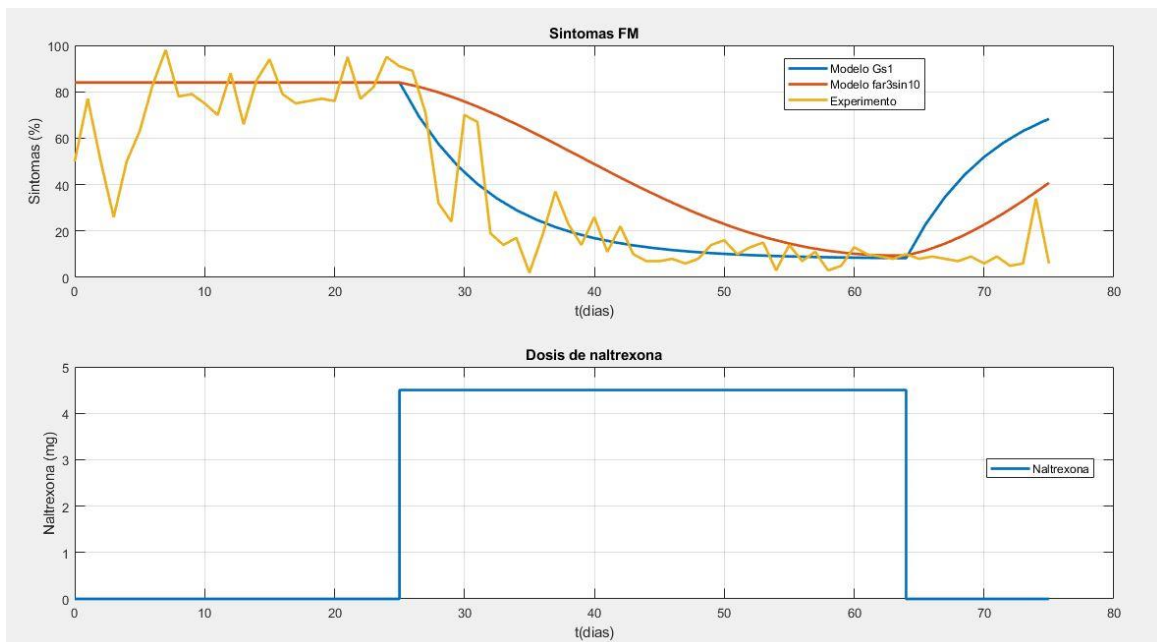


Ilustración 20: Respuesta del modelo Gs1 frente a experimento y a Far3Sin10 (teniendo este último $K=1.75$)

Comparándolo con el modelo anterior, notamos una gran mejoría en las zonas de mayor interés para nuestro modelado (del día 12 al 64, cuando más lineal es el sistema, puesto que responde con evoluciones proporcionales a las variaciones de tratamiento de igual valor -véase *Ilustración 20*).

No obstante, y debido a lo necesario que es conseguir que nuestro modelo de respuesta del paciente sea lo más parecido al valor medio del experimento posible en la parte cuyo comportamiento es más lineal (entre los días 12 y 72), propondremos otras funciones de transferencia y estudiaremos si se ajustan mejor a nuestras necesidades.

2.2.2.2 Función de transferencia de segundo orden

Siguiendo el mismo método de análisis que antes, trataremos de crear una función de transferencia de segundo orden para modelar la respuesta del paciente al tratamiento. Comenzaremos con un modelo subamortiguado, que se corresponde con la siguiente función:

$$G_{s2} = \frac{K \cdot \omega_n^2}{s^2 + 2\delta\omega_n \cdot s + \omega_n^2}$$

Donde:

$$K = \frac{\Delta y}{\Delta u} \qquad \omega_n = \frac{\pi}{tp\sqrt{1 - \delta^2}}$$

$$SO(\%) = 100 * e^{\frac{\pi \cdot \delta}{\sqrt{1 - \delta^2}}}$$

Extrayendo los datos pertinentes del experimento inicial (del que podemos saber que *tp* -tiempo de establecimiento del permanente, o tiempo que dura el periodo transitorio- es de 28 días, y que *SO* –“sobreoscilación”, o máximo valor al que se le permite al modelo sobrepasar la Δy durante el periodo del transitorio- debe ser cercana al 0%, y la tomaremos como 1%), y sin olvidarnos que no debemos confundir las ganancias de bloque y de la función, igual que antes, y seteando la del bloque a 1), el modelo queda de la siguiente forma:

$$G_{s2} = \frac{-0.6756}{s^2 + 0.3304 \cdot s + 0.04}$$

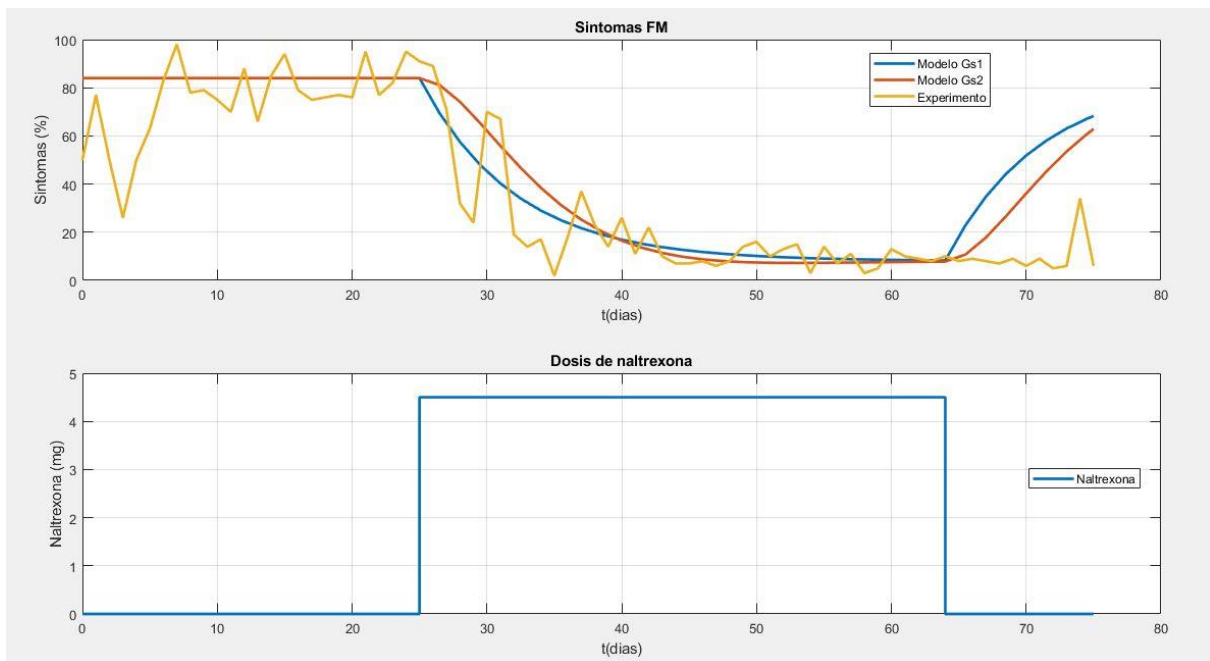


Ilustración 21: Respuesta del modelo Gs2 frente a experimento y a Gs1

No obstante, el comportamiento de este modelo de respuesta del paciente no se considera adecuado para nuestro sistema, debido a que en la zona transitoria se aleja demasiado del valor medio de los puntos experimentales.

Por tanto, y debido a que el modelo $Gs1$ de primer orden presenta menos desviación que $Gs2$, el siguiente paso constaría de probar una función de transferencia de segundo orden sobreamortiguada.

Dicho tipo de función de transferencia tiene la siguiente definición:

$$Gs3 = \frac{K}{(\tau_1 \cdot s + 1)(\tau_2 \cdot s + 1)}$$

Y si bien en el permanente responderá igual que $Gs1$, sin la necesidad por tanto de medir oscilaciones o efectos de sobreoscilación, la respuesta en el transitorio será lo que podamos modelar gracias al valor de τ_1 y τ_2 .

Los valores de τ_1 y τ_2 se ajustan experimentalmente, aunque es útil saber que la respuesta del sistema de segundo orden de este tipo puede aproximarse por la de uno de primer orden con constante de tiempo igual a la suma de constantes de tiempo de cada polo [17].

Por tanto, podríamos formular la siguiente hipótesis:

$$\tau \approx \tau_1 + \tau_2$$

Con esto, y tras el ajuste experimental, podemos constatar que el valor de $Gs3$ cuyos resultados consideramos más satisfactorios es:

$$Gs3 = -\frac{16.888}{(6.5 \cdot s + 1)(0.8 \cdot s + 1)}$$

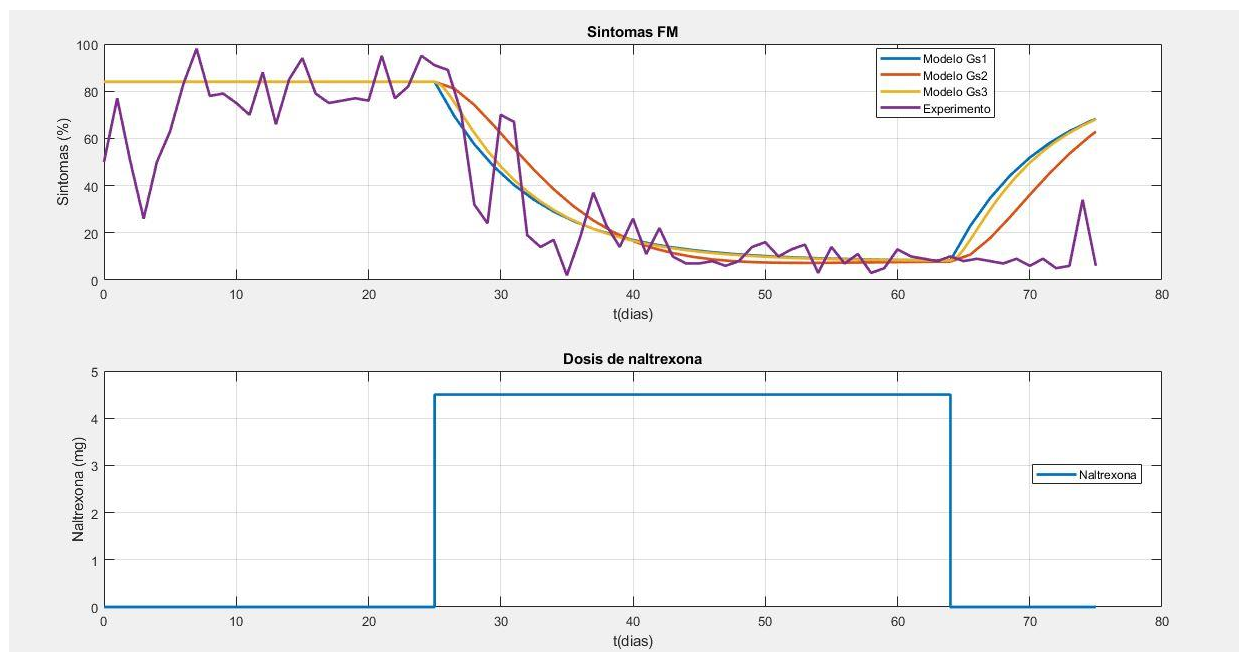


Ilustración 22: Modelo $Gs3$ frente a $Gs1$, $Gs2$ y experimento

El valor de $Gs3$ elegido consigue modelar con mayor precisión (véase *Ilustración 22*) que $Gs1$ y que $Gs2$ la respuesta del paciente durante el régimen transitorio, siendo igual de efectivo que los otros dos

en las zonas permanentes (del día 12 al 25 y del 53 al 64).

Por ello, será el modelo $Gs3$ el que conformará la base con la cual trabajaremos a partir de este momento, y que adoptaremos como función de transferencia continua de respuesta de nuestro paciente a la enfermedad de la fibromialgia y su tratamiento.

2.3 Restricciones de cada señal y consecuencias de las mismas

Debido a la naturaleza de las señales que se relacionan en el apartado anterior, se realizarán una serie de cambios en el modelo para acercarlo aún más al comportamiento del proceso real.

Respecto a los síntomas que surgen debido a la fibromialgia, y que son salida del modelo de la respuesta del paciente, cabe destacar que la percepción del dolor va ligada a ciertas cotas de subjetividad por parte del enfermo. Debido a ello, los niveles que marcan el porcentaje de síntomas de la fibromialgia que hasta ahora hemos estado utilizando para generar nuestro modelo matemático de respuesta son subjetivos (al fin y al cabo, un “50% de dolor” no es la misma cantidad de dolor para una persona A que para una persona B). Por ello, y para apoyar los esfuerzos del equipo médico por poder cuantificar el nivel de los síntomas, el sistema podrá adaptarse a los niveles de percepción de síntomas de cada paciente. Para ello, el especialista podrá cambiar el significado de los límites “0%” y “100%” de síntomas de fibromialgia en cada cita médica, adaptándolo al rango de percepción del paciente según su criterio como experto; siendo así cómo el médico supervisará las acciones de control además de su papel en la toma de decisiones respecto a la aprobación o no del tratamiento propuesto por el sistema durante las citas médicas.

No obstante, este cambio se realizará en apartados futuros, puesto que no consideraremos una influencia directa en la sección del sistema que atañe al punto actual, donde evaluamos el modelo de respuesta del paciente frente a las dosis de naltrexona; sino que influye en cómo el sistema debe interpretar los niveles de síntomas de fibromialgia que salen de este modelo de respuesta.

Por otro lado, otro cambio que debe tenerse en cuenta se debe a la naturaleza del medicamento que utilizamos en nuestro estudio: la naltrexona. Dicho medicamento, como muchos otros, se vende en farmacias con unas dosis específicas. Eso quiere decir que el sistema debe contemplar la posibilidad de que la dosis exacta que él proponga en su tratamiento puede no existir, siendo por tanto necesario que sea capaz de adaptar a dosis comerciales existentes del medicamento.

Al igual que antes, este cambio no atañe directamente al modelo que hemos creado, ya que no modifica la forma en la cual se relacionan la entrada del mismo (naltrexona) y la salida (síntomas), sino al módulo de control. Por lo que se aplicarán las modificaciones pertinentes más adelante.

El cambio que sí atañe a nuestro modelo y que debemos tener en cuenta es la naturaleza intrínseca de cualquier tratamiento dosificado con una naturaleza parecida a la nuestra: no corresponden a lo que en ingeniería de control se conoce como un sistema continuo.

En la ingeniería de control, los sistemas continuos trabajan con señales continuas en el tiempo, y se modelan con funciones continuas (como $Gs3$, la cual habíamos hallado como satisfactoria para nuestro proyecto). Sin embargo, el tratamiento general de una enfermedad supone que el paciente tome medicamentos cada cierto tiempo, es decir, de forma no continua (lo que, en términos propios de la ingeniería de control, viene a ser definido como una señal “discreta”). Para que lo comprendamos de forma fácil, el paciente no va a estar continuamente tomando el medicamento, sino solo (en nuestro caso concreto) una vez al día; por tanto, en la relación existente entre la respuesta del paciente y el

tratamiento debe tenerse en cuenta que no se le suministra de forma ininterrumpida, sino de forma intermitente, cuestión que cambiará completamente la relación matemática entre las señales de entrada y salida, ya que se deberá modelar también el tiempo de no-suministro del medicamento que se da entre día y día. Es decir, debemos tener en cuenta que la medicación es diaria y sus efectos se provocan solo en el instante en el que se toman (por lo que se disparan solo una vez al día, lo que no quiere decir que los efectos provocados no puedan permanecer en el tiempo).

Hasta ahora, en el modelado que nos atañe, hemos estado utilizando siempre la herramienta matemática conocida como “Transformada de Laplace”. Ahora, pasaremos a utilizar la llamada “Transformada en Z” capaz de tener en cuenta que las señales son discretas y que, por tanto, la evolución del sistema, así como la toma de datos del mismo, también son discretas.

Por suerte, aunque la transformada en Z solo puede relacionarse con el tiempo y con la transformada de Laplace cuando los datos se obtienen de una señal temporal muestreada, gracias a que los datos en los cuales hemos basado nuestro modelado tienen estas características (véase *Ilustración 4*) la conversión de nuestro modelo desde el uso de la transformada de Laplace al uso de la herramienta de la transformada en Z es sencillo, y mucho más si utilizamos la herramienta MATLAB®.

Para ello, simplemente debemos llevar a cabo los siguientes pasos:

- 1) Introducimos en la ventana de comandos el modelo continuo que extrajimos en el apartado anterior en MATLAB® mediante la función “tf”:

$$Gs3 = tf(-16.8888, conv([6.5 \ 1], [0.8 \ 1]))$$

- 2) MATLAB® nos devolverá $Gs3$ por la ventana de comandos en un formato amigable para que podamos interpretar si la hemos introducido bien, diciéndonos el tipo de función matemática que hemos introducido (continua, en este caso):

```
Gs3 =
      -16.89
-----
 5.2 s^2 + 7.3 s + 1
Continuous-time transfer function.
```

- 3) Convertiremos nuestra función de transferencia continua en una discreta mediante la transformada en Z, para ello, usaremos la función “c2d” (“continuous-to-discrete”):

$$Gz = c2d(Gs3,1)$$

- 4) Con esto, MATLAB® nos devolverá Gz por la ventana de comandos, igual que antes nos mostró $Gs3$, diciéndonos el tipo de función matemática resultante (discreta, en este caso):

```
Gz =
 -1.055 z - 0.6633
-----
 z^2 - 1.144 z + 0.2457
Sample time: 1
Discrete-time transfer function.
```

Es muy importante que nos fijemos en los dos últimos pasos. En el tercero, dentro de la función “c2d”, se ve que hemos introducido un “1”, como queda reflejado en el resultado que se nos da en el cuarto paso.

Este “1” es el “Sample time”, o “tiempo de muestreo”.

Se define el “tiempo de muestreo” como la cantidad de tiempo por unidad de muestra que se tarda en evaluar una señal continua para producir una señal discreta. De forma simple, en nuestro caso es un tiempo de muestreo de 1 día, puesto que es lo que se tarda entre ingesta e ingesta (es decir, muestreo y muestreo) de una dosis del tratamiento. Por tanto, introducimos ese “1” en el sistema para que en el modelo figure cuánto tiempo de “inactividad” hay entre dosis y dosis de naltrexona.

Con ello, ya hemos logrado incluir en el modelo de respuesta del paciente la discretización intrínseca de nuestra aplicación, y podemos mostrar a continuación los resultados de este paso, comparando en una gráfica el modelo continuo del sistema, G_{s3} , con su tratamiento de naltrexona como señal continua, frente al modelo discreto del sistema, G_z , con su tratamiento de naltrexona como señal discreta, que como vemos concuerdan a la perfección (véase *Ilustración 23*).

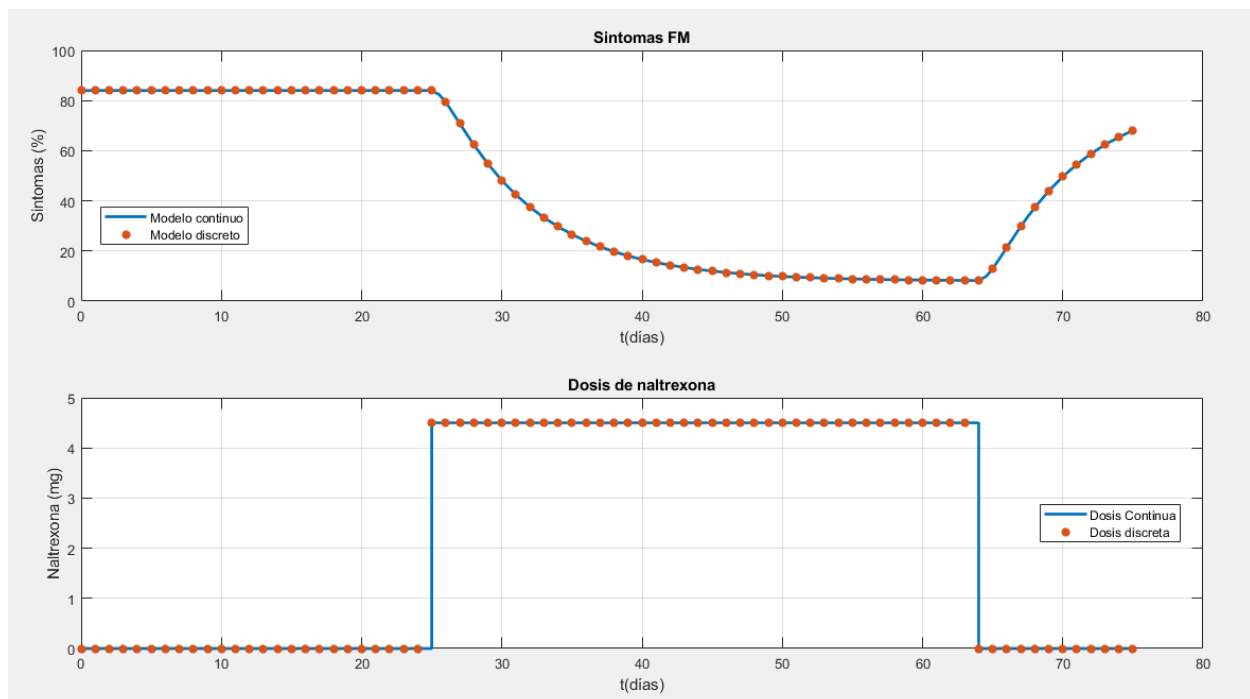


Ilustración 23: Prueba de modelo comparativa entre función continua (G_{s3}) y discreta (G_z), gráficas de datos.

Para generar estas gráficas, hemos usado las siguientes líneas de código en la ventana de comandos:

```
subplot(2,1,1);plot(t3,Sintomas_FM3,t,Sintomas_FMD,'*','LineWidth',2);grid;
xlabel('t(días)');ylabel('Sintomas (%)');title('Síntomas FM');
legend('Modelo continuo','Modelo discreto','Location','Best');

subplot(2,1,2);plot(t3,Naltraxona,t,Naltraxona1,'*','LineWidth',2);grid;
xlabel('t(días)');ylabel('Naltraxona (mg)');title('Dosis de naltraxona');
legend('Dosis Continua','Dosis discreta','Location','Best');
```

Hemos creado el pequeño montaje en MATLAB[®] que se refleja a continuación para extraer los datos antes expuestos (véase *Ilustración 24*). En él, reproducimos parte por parte el montaje creado en ocasiones anteriores, con la función de transferencia continua G_{s3} , introduciendo además una sección similar, pero para G_z .

Para poder incluir de forma exitosa G_z y poder ver su comportamiento, hemos introducido unos bloques tipo escalón y suma copia de los de la sección de G_{s3} . Sin embargo, el bloque en el cual hemos puesto la función de transferencia G_z es un bloque del tipo “Discrete Transfer Fcn” y no del tipo “Transfer Fcn” como ocurría con aquel en el que introdujimos G_{s3} . Además, internamente en dicho bloque se define, en la opción “Sample Time” (a la que se accede clicando dos veces en el bloque, con lo que se abre la ventana de valores del bloque) un “1”, debido a nuestro tiempo de muestreo y a que necesita saber su valor para tomar los datos de entrada (es decir, de dosis de naltrexona) cada tiempo de muestreo. Esto se ha debido hacer también en los bloques “t”, “Naltrexona1” y “Síntomas_FMD”, bloques del tipo “To Workspace”, donde también hemos puesto un “1” en la opción “Sample Time” para que se guarden los datos de forma discreta.

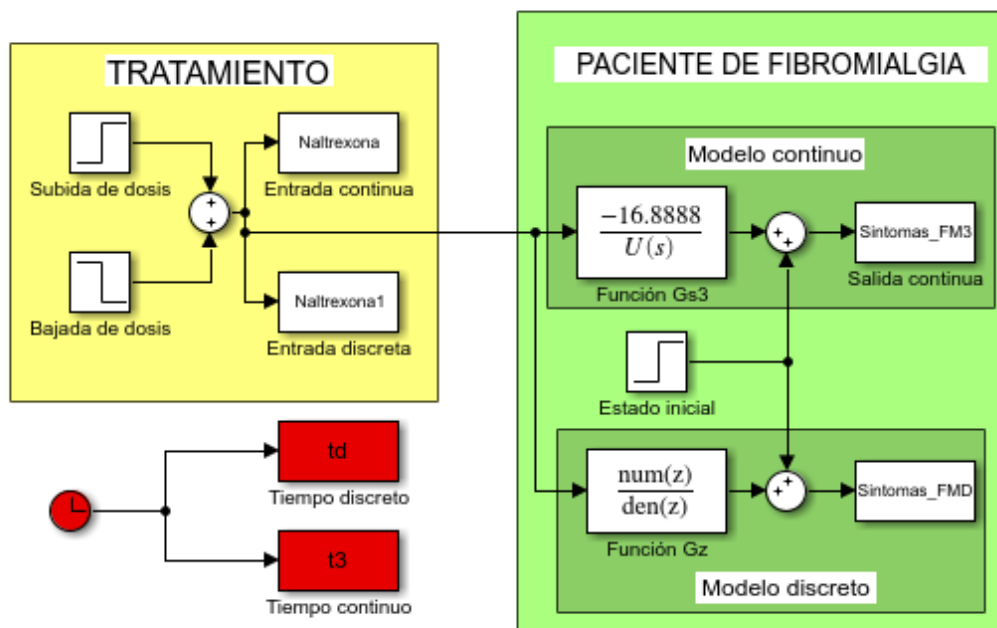


Ilustración 24: Prueba de modelo comparativa entre función continua (G_{s3}) y discreta (G_z), montaje Simulink®.

Tras esto, ya hemos definido completamente el modelo de la respuesta del paciente ante la dosis de naltrexona con el cuál trabajaremos a partir de ahora, G_z , y al cuál trataremos de asociarle un control que cumpla con todos los objetivos especificados por nuestro trabajo.

3. Control del tratamiento

*"Será necesario que soporte dos o tres orugas
si quiero conocer las mariposas"*

*- El Principito
Antoine de Saint-Exupéry.*

Tal y como adelantamos en el esquema de nuestra aplicación bajo desarrollo (véase *Ilustración 6*), el siguiente punto de este trabajo contará con el desarrollo de un módulo de control, capaz de estimar un tratamiento que se adecúe eficazmente a las necesidades del paciente, modelado gracias a la realimentación de la información que puede extraerse del modelo de respuesta del paciente de fibromialgia ante un tratamiento de naltrexona desarrollado en el apartado anterior de esta memoria.

Aunque ya durante la introducción definimos cuál era la función del módulo de control, no nos costará resumirlo ahora en apenas una frase: la intención es dar al sistema una referencia (es decir, un nivel de síntomas al cuál debe conseguir que el paciente evolucione) y que el módulo de control sea capaz de estimar el tratamiento más eficaz para llegar a dicha referencia según unas especificaciones de velocidad, sobreoscilación, oscilación, agresividad...

Para la creación de un controlador capaz de cumplir con estas y otras necesidades ya descritas a lo largo de la memoria, podemos proponer multitud de metodologías de control que podrían ser eficaces en nuestro caso particular. No obstante, nosotros nos centraremos en dos propuestas que, históricamente, han demostrado su alta eficacia en multitud de problemas a los cuáles ha debido enfrentarse la ingeniería de control: el *Control PID* y el *Control Predictivo*.

El *Control PID*, basado en un módulo de control (o, simplemente, "controlador") con el mismo nombre, es una técnica de control simultáneo por realimentación basada en el cálculo de la desviación o error entre un valor medido y un valor deseado y que actúa gracias a los efectos provocados por tres submódulos (*P*, *I* y *D*).

El submódulo *P* (o "proporcional") trata de llevar a cabo la evolución del sistema hacia la referencia en, fundamentalmente, los momentos de amplios cambios (es decir, de alto nivel de error entre la referencia y la situación actual del sistema); es el encargado principal de la velocidad y de las aproximaciones a la referencia, y depende del valor actual del error, pero es sensible al ruido y, en la mayoría de las ocasiones, no es capaz, por sí solo, de hacer que el sistema alcance cualquier referencia con error nulo o próximo a cero.

Para poder corregir estos errores y minimizar su valor, existe el segundo submódulo, el *I* (o "integral"), sensible a los valores del error y capaz de corregir las desviaciones provocadas por ruido, perturbaciones o errores acarreados por aproximaciones matemáticas, gracias a que actúa no según el

valor del error, sino de su integral (es decir, del sumatorio de los errores a lo largo del tiempo) siendo, por tanto, más sensible ante pequeñas desviaciones, aunque no es efectivo en la mayoría de los casos si no se acompaña de un submódulo *P*.

Por último, el submódulo *D* (o “derivativo”) trata de mantener el error al mínimo corrigiéndolo proporcionalmente a la misma velocidad a la que se produce; evitándose por tanto un incremento del error que pudiese llevar a tomar medidas agresivas por parte del controlador para solventarlo (es decir, pretende que las variaciones en la “variable de control” -nombre que recibe nuestra señal “dosis de naltrexona diaria”- no tengan un valor exagerado).

Dichos submódulos, juntos, conformando el controlador *PID*, generan uno de los módulos más completos, eficaces y sencillos de programar e implementar del mundo (siempre y cuando se tenga información previa de un modelo de proceso a controlar).

Por ello, será una de las dos técnicas de control que implementaremos y compararemos.

La otra técnica que nos queda por explicar es, por tanto, el *Control Predictivo*.

Dicho método de control se basa en la “predicción” de la evolución del sistema. Concretamente, existen múltiples formas de *Control Predictivo*; en nuestro caso, aplicaremos el *MPC* (“Control Predictivo por Modelo”)

El Control Predictivo por Modelo (*CPM*, más conocido como *MPC* por sus siglas en inglés) es un método avanzado de control de procesos que depende, igual que el *PID*, de tener un modelo del proceso que pretendamos controlar. Su ventaja principal frente a otros controladores es que el *MPC* permite optimizar los rangos de tiempo. Si bien el *PID* solo trata de alcanzar la referencia a una velocidad específica, el *MPC* tiene la capacidad de estimar los acontecimientos futuros y tomar acciones de control en consecuencia a dichas estimaciones. No obstante, y como contra, el *MPC* tiene una implementación más costosa y compleja, precisa de una mayor necesidad de cómputo y se implementa casi universalmente como un sistema de control digital programado (a diferencia de otros controladores como el *PID* que pueden ser implementados además de forma analógica o por electrónica digital no programada).

No obstante, el *Control Predictivo* ha demostrado numerosas veces ser una herramienta altamente eficaz en aplicaciones de control para sistemas sensibles como enfermedades, y sus ventajas en cuanto a adaptación ante cada paciente particular, imprevistos y situaciones complejas (como sucesivas variaciones de referencia), que podrían suponer complicaciones para numerosas formas de control, conforman para el *Control Predictivo* simples retos ante los cuáles demostrar su alta eficacia [10] [18].

Como puede comprobarse por nuestra introducción a este punto de control, nuestra intención no es explicar letra a letra la ingeniería de control ni los controladores que vamos a aplicar sino, simplemente, su implementación, así como sus definiciones y pasos necesarios. No obstante, la completa comprensión, así como mucha más información sobre la ingeniería de control y el funcionamiento teórico/práctico del controlador *PID* y del *Predictivo* puede ser alcanzada haciendo uso de la bibliografía del *Anexo 2*.

3.1 Control del proceso mediante PID

Como ya hemos dicho anteriormente, el control del proceso mediante *PID* es sencillo de diseñar e implementar, y más aún si hacemos uso de las herramientas que MATLAB® nos ofrece.

En primer lugar, y para analizar el sistema base, realizaremos el control *PID* sobre nuestro proceso continuo, para posteriormente realizar el pequeño salto (gracias a MATLAB®) hacia el modelo discreto con el que terminamos de definir el modelo de respuesta del paciente.

Para ello, comenzaremos introduciendo el modelo *G_s* en la ventana de comandos, para luego cargar dicha función de transferencia en la herramienta “*rltool*” de MATLAB®.

```
>> Gs=tf(-16.888,conv([6.5 1],[0.8 1]))

Gs =

      -16.89
-----
 5.2 s^2 + 7.3 s + 1

Continuous-time transfer function.

>> rltool(Gs)
```

El comando “*rltool*” abre la herramienta de diseño SISO (“Simple-Input/Single-Output”), que nos ayuda en el diseño interactivo de controladores de una sola entrada/salida (SISO) usando la técnica del lugar de las raíces (el lugar geométrico de los polos y ceros de una función de transferencia en el plano de los ejes real e irreal).

Desde un punto de vista práctico, simplemente necesitamos abrir el panel “Tuning Methods” y, dentro de él, pulsar en “PID Tuning” (véase *Ilustración 25*).

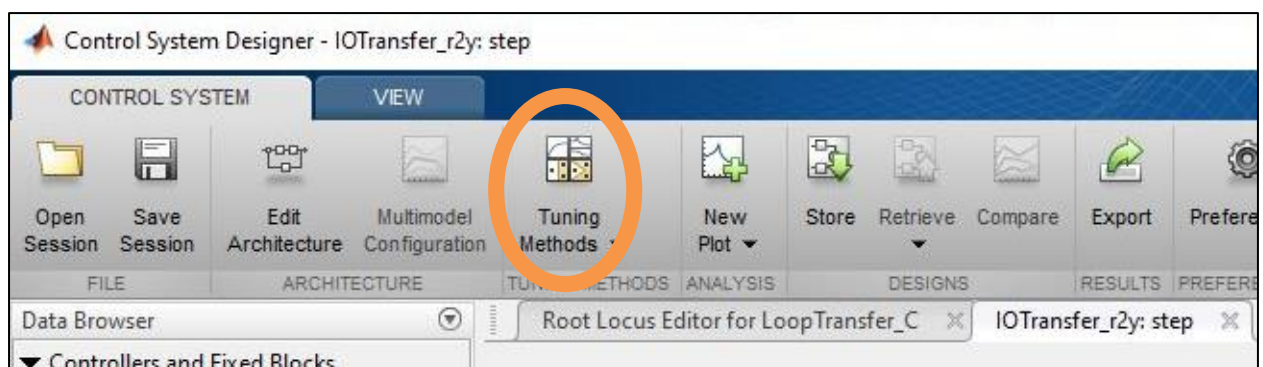


Ilustración 25: Uso de la herramienta rltool para la creación de controladores PID

Una vez abierta dicha herramienta, se nos desplegará una ventana donde solo necesitaremos pulsar en “PID” y, posteriormente, en “Update Compensator” (véase *Ilustración 26*). Al hacerlo así, sin modificar las especificaciones de “Response Time” o “Transient Behaviour”, se creará un controlador con las especificaciones predeterminadas de la aplicación (en nuestro caso, el controlador tratará de llegar a la referencia en 7.26 días y se encontrará a la media entre “agresivo” -controles rápidos pero que llevan a cambios bruscos de la señal de control, tendiéndose a grandes niveles de sobreoscilación y oscilación en la salida del proceso- y “robusto” -controles lentos pero que se encargan de evitar sobreoscilaciones y oscilaciones en la señal de salida).

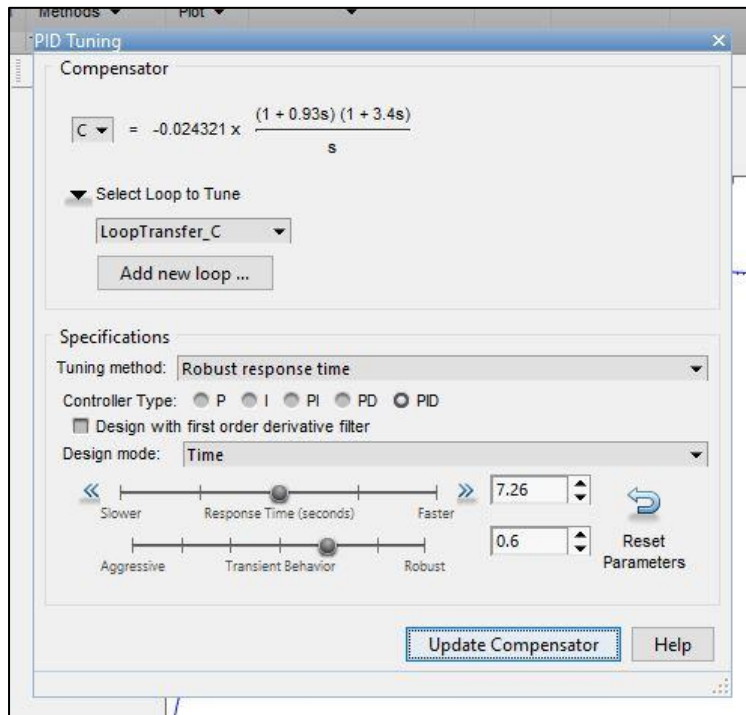


Ilustración 26: Uso de la ventana PID Tuning para la creación de controladores según especificaciones.

Al haber pulsado el anterior botón, nos percataremos de que hace aparición un valor a la derecha de “C”. Esa será la función de transferencia asociada a nuestro controlador, y la cual ya podremos aplicar a nuestro proceso.

Para ello, solo nos bastará introducirla en Simulink® mediante un sencillo montaje realimentado basado en el que hemos ido usando hasta ahora. Concretamente, será igual que el de la Ilustración 28, pero sin el bloque llamado “Perturbación” y con bloques continuos en vez de discretos (esto es, que los “To Workspace” tengan en “Sample Time” el valor “-1” y que los bloques de funciones de transferencia –“Modelo del sistema” y “PID”- sean del tipo continuo, es decir, que sean simplemente “Transfer Function”).

Es importante destacar que el escalón de “Referencia” se dará en el día 25, y

que caerá de 84% a 8%, de forma que el controlador tenga que adaptar el tratamiento para que los síntomas alcancen dicho valor en 7.26 días aproximadamente. Este bloque será el mismo que usaremos en todos los montajes Simulink® donde apliquemos control PID.

Además, debido a imposición por parte de Simulink®, en un bloque de función de transferencia no podremos introducir una función con más ceros que polos (esto es, con mayor orden de transformada -de Laplace o Z- en el numerador que en el denominador). Por supuesto, nuestro controlador “C”, función que necesitamos introducir en un bloque de función de transferencia, tiene un orden mayor de ceros que de polos (concretamente, tiene un cero más que el número de polos). Para evitar este problema y aprovecharlo en nuestro favor, crearemos un polo de alta frecuencia. Los polos de alta frecuencia solo actúan como filtro para eliminar componentes de ruido de alta frecuencia a la salida (en caso de que puedan existir). En un caso como el nuestro no es común que existan dichos ruidos, pero añadir este polo no va a ocasionar otra cosa sino beneficios.

Para ello, nos basta con escoger una frecuencia adecuada e implementarla, cumpliendo lo siguiente:

- 1) En primer lugar, se definirá como “polo de alta frecuencia” a aquel que, siendo el polo:

$$\text{Polo: } \frac{1}{\frac{1}{p} * s + 1}$$

cumpla que $p \geq 10 * \omega$ donde ω es, en nuestro caso, el valor del término independiente del denominador de $Gs3$ (es decir, $\omega=1$). Por tanto, el polo de alta frecuencia queda como:

$$\text{Polo de alta freq: } \frac{1}{0.1 * s + 1}$$

- 2) Convolucionar ese Polo de alta freq con el denominador de “C”, dando lugar a “Cs”:

$$Cs = \frac{-0.0769 * s^2 - 0.1053 * s - 0.02432}{0.1 * s^2 + s}$$

- 3) Introducirlo Cs en Simulink[®] mediante el bloque “Transfer Function” nombrado antes, adaptando el montaje de la *Ilustración 28*.

Una vez llevado esto a cabo, con elegir un tiempo de simulación nos bastará para obtener resultados, que luego podremos mostrar por pantalla mediante los comandos:

```
subplot(2,1,1); plot(t3, SintomasFMRef, t3, SintomasFM3, tiempo, Sintomaorigen, 'LineWidth', 2); grid;
xlabel('t(días)'); ylabel('Sintomas (%)'); title('Sintomas FM');
legend('Referencia', 'Control continuo', 'Experimento', 'Location', 'Best');
```

```
subplot(2,1,2); plot(t3, Naltrexona, 'LineWidth', 2); grid; xlabel('t(días)'); ylabel('Naltrexona (mg)');
title('Dosis de naltrexona'); legend('Dosis', 'Location', 'Best');
```

Por tanto, ya podemos analizar los resultados de aplicar este control *PID* (véase *Ilustración 27*).

Como podemos observar, el controlador consigue su objetivo, y permite que el sistema alcance su referencia en aproximadamente siete días. Debido a que no es del todo robusto, se da una sobreoscilación (es decir, la señal de salida atraviesa el valor de la referencia). Cabe destacar que este controlador sería también capaz de evitar los efectos negativos de las perturbaciones, así como queremos clarificar que el tratamiento no tiene ninguna restricción aún (de ahí que puedan darse dosis tan altas de forma esporádica -véase *Ilustración 27*).

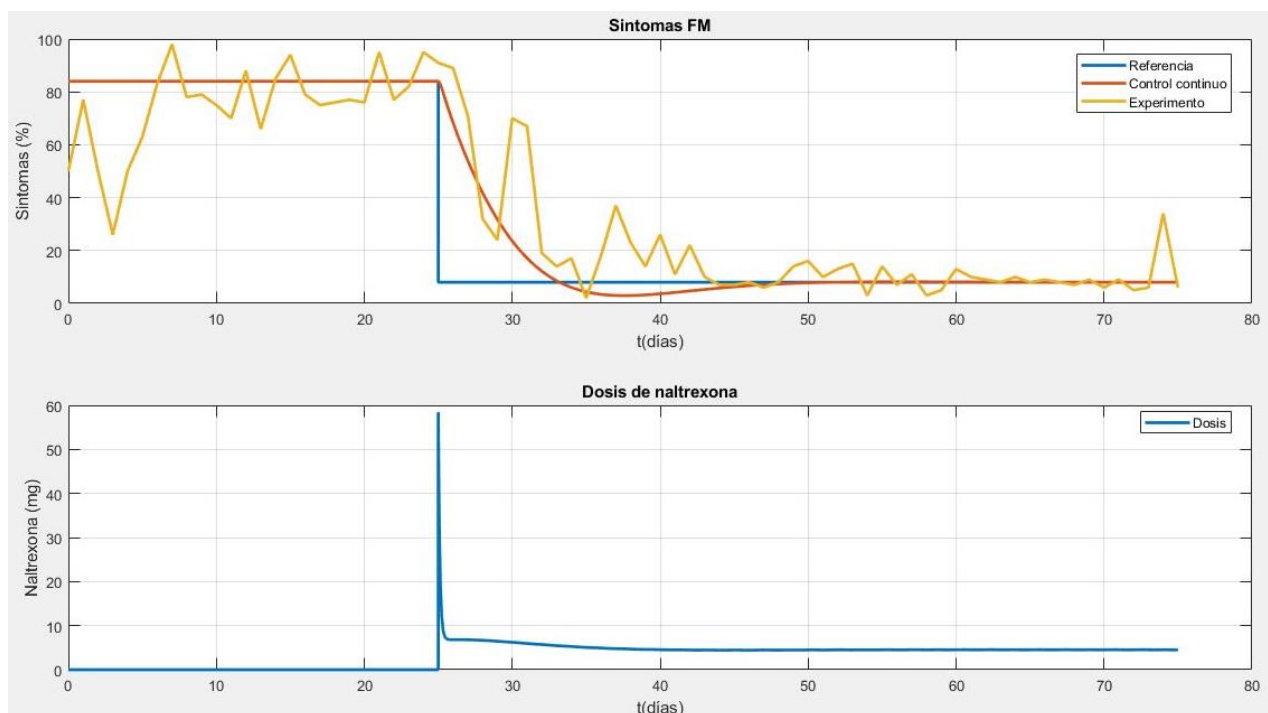


Ilustración 27: Control continuo ante seguimiento de referencia, comparado con datos del experimento inicial.

Sin embargo, como ya destacamos anteriormente, el sistema impone algunas restricciones que permiten su acercamiento con el caso real.

El primero de todos era que la ingesta de la dosis era diaria y que, por tanto, el sistema debía ser discreto (como lo era el modelo de respuesta del paciente). Por tanto, será necesario que el módulo de control

sea, igualmente, discreto.

Con este fin, llevaremos a cabo pasos similares a los aplicados para la discretización de G_{s3} a G_z .

```
>> Cs=tf(-0.02432*conv([0.93 1],[3.4 1]),conv([1 0],[0.1 1]))

Cs =

    -0.0769 s^2 - 0.1053 s - 0.02432
    -----
           0.1 s^2 + s

Continuous-time transfer function.

>> Cz=c2d(Cs,1)

Cz =

    -0.769 z^2 + 1.411 z - 0.6661
    -----
           z^2 - z + 4.54e-05

Sample time: 1 seconds
Discrete-time transfer function.
```

Una vez hallado el controlador discreto (C_z), solo nos faltará implementarla en Simulink® (fijémonos en que en C_s ya estaba introducido el polo de alta frecuencia y que, debido a ello, ahora no necesitaremos introducirlo).

Para esto, y como hicimos antes, desarrollaremos el montaje realizado para probar el modelo discreto de respuesta del paciente (véase *Ilustración 24*), añadiendo pues un módulo de control realimentado *PID* (véase *Ilustración 28*). Por ello, debemos cerciorarnos de que en dicho montaje los bloques “PID” y “Modelo del sistema” sean “Discrete Transfer Function” (donde hemos introducido G_z y C_z), con un “Sample Time” de valor uno. Cabe destacar que los bloques del tipo “To Workspace” pueden mantener el valor “-1” en su sample time, de forma que guarden valores continuamente y, a pesar de

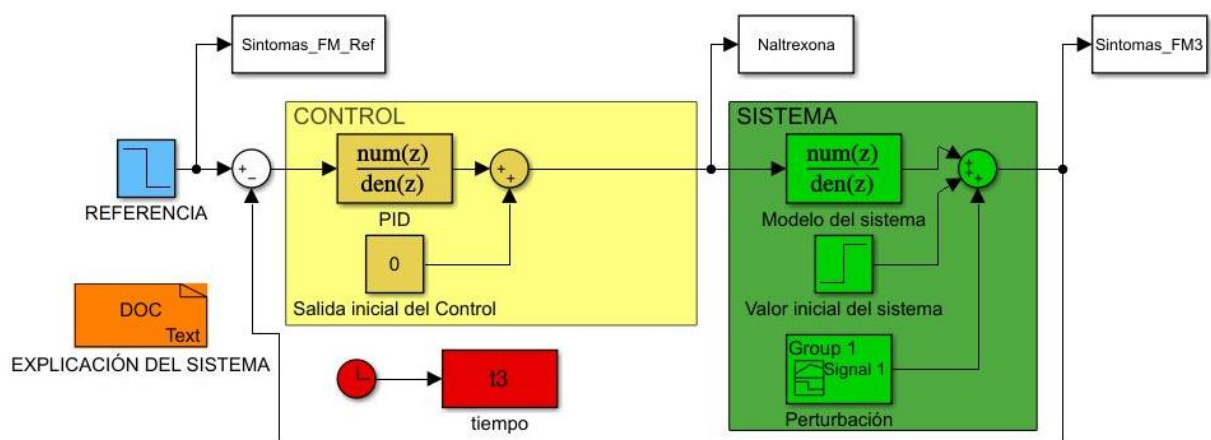


Ilustración 28: Montaje Simulink® para la implementación de un controlador PID discreta en el sistema diseñado.

que el sistema reaccionará y evolucionará de manera discreta, podremos monitorizar, simplemente por representar mediante gráficas de sencilla comprensión, los tiempos entre cambios de valores (es decir,

a pesar de que los cambios son discretos, mediremos los datos, solo para representarlos, de forma continua).

Además, se ha añadido una señal de “Perturbación”. Dicha señal es una señal cuadrada de valor 10 entre los días 83 y 84, y viene a simular cambios ruidosos no contemplados por el modelo linealizado de respuesta del paciente (por ejemplo, días sueltos que el paciente sea más sensible al dolor sin motivo aparente). Se simulará este montaje con dicha señal para comprobar que, como ya comentamos antes, el control *PID* es capaz de anular eficazmente las perturbaciones.

Así pues, representando los datos obtenidos tras simulación mediante los siguientes comandos:

```
subplot(2,1,1);plot(t3,SintomasFMRef,t3,SintomasFM3,tiempo,Sintomaorigen,'LineWidth',2);
grid;xlabel('t(días)');ylabel('Sintomas (%)');title('Sintomas FM');
legend('Referencia','Control discreto','Experimento','Location','Best');
```

```
subplot(2,1,2);plot(t3,Naltrexona,'LineWidth',2);grid;xlabel('t(días)');ylabel('Naltrexona (mg)');
title('Dosis de naltrexona');legend('Dosis','Location','Best');
```

Podremos observar (véase *Ilustración 29*) que el sistema (aunque tardando más de los 7.23 días especificados en el controlador continuo) consigue alcanzar su objetivo de referencia en un tiempo razonable, a la vez que es capaz de solventar el problema de la perturbación de igual forma. No obstante, con un simple vistazo al tratamiento podemos percatarnos de que este sistema controlado se aleja enormemente de la realidad, puesto que propone dar dosis negativas al paciente.

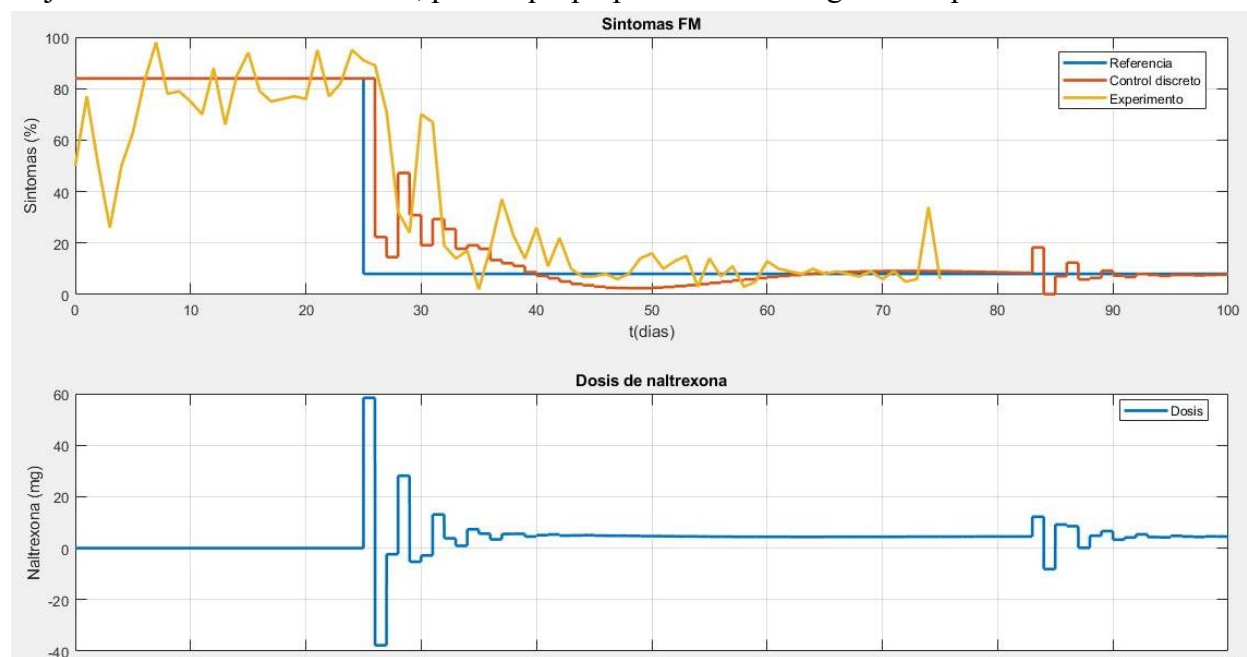


Ilustración 29: Control discreto ante seguimiento de referencia por PID, comparado con datos del experimento inicial.

Sin embargo, esto se debe al desconocimiento por parte del módulo de control de la naturaleza de la señal de control (“dosis de naltrexona diaria”). Para poder brindar al controlador de este conocimiento, proponemos introducir una saturación a la salida del control, capaz de filtrar las dosis a valores razonables. Es importante destacar que, gracias a que usamos un bucle cerrado y realimentado de control, la introducción de una saturación no influirá a un mal funcionamiento del módulo de control, sino que el controlador será capaz de adaptar sus respuestas a partir de cómo se comporta el sistema condicionado por dicha saturación.

Para esto, introduciremos en Simulink® un bloque del tipo “Saturation” donde impondremos como valor mínimo 0 mg (puesto que no puede tomarse menos dosis que “nada”) y como valor máximo 13.5 mg (debido a que el máximo valor comercial común de dosis de naltrexona es de dicha cantidad), y volveremos a representar los datos obtenidos (véase *Ilustración 30* e *Ilustración 31*).

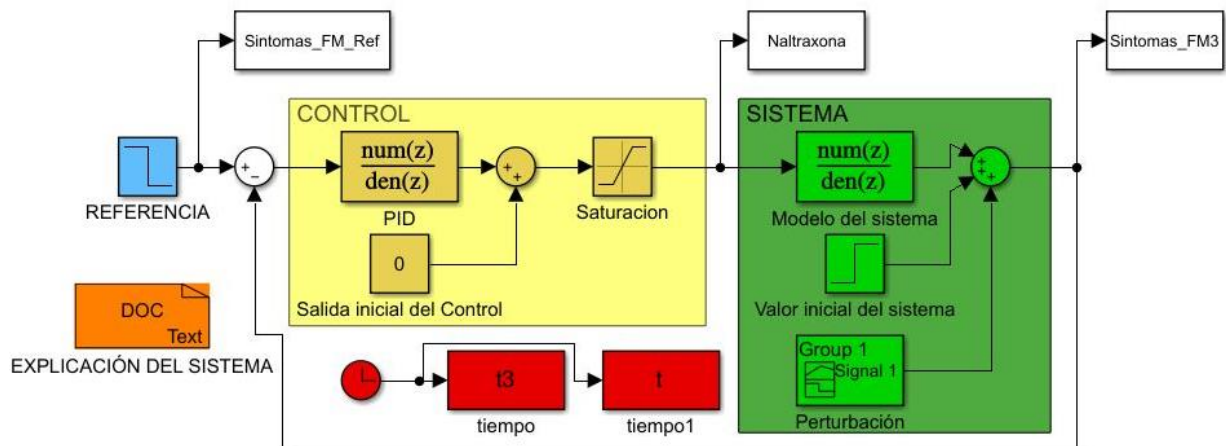


Ilustración 30: Montaje Simulink® para la implementación de un controlador PID discreta en el sistema diseñado, con saturación a la salida del PID

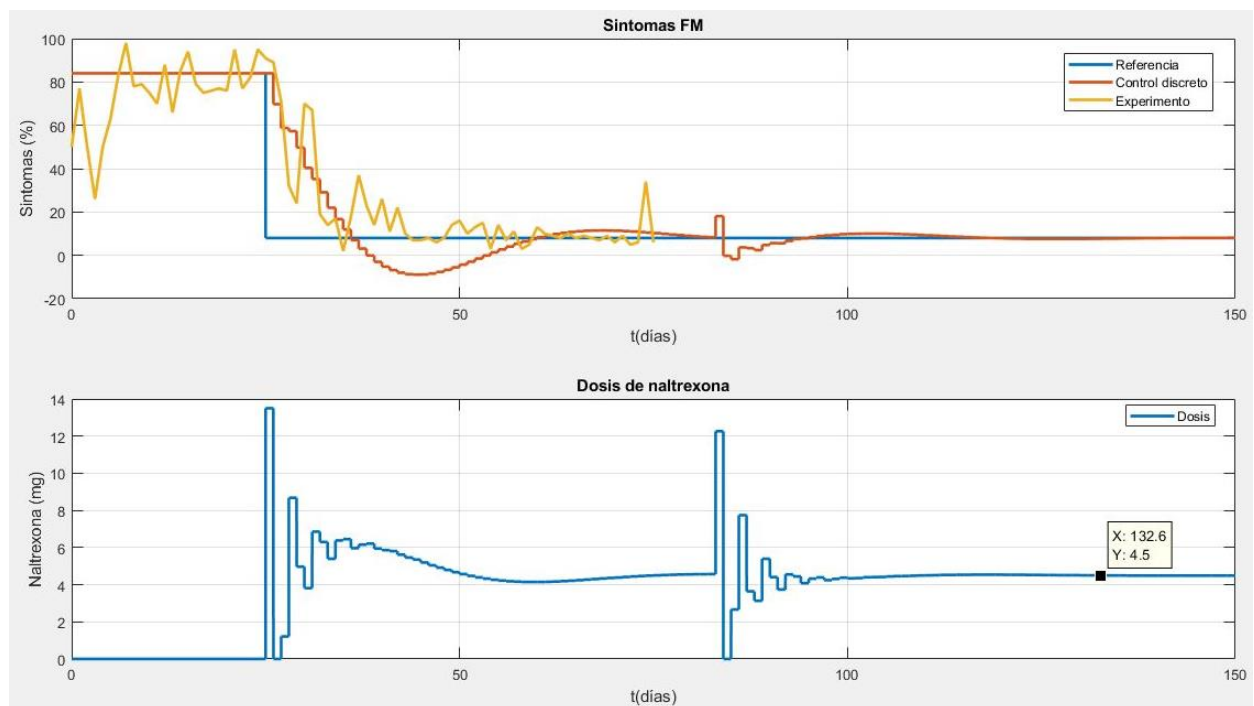


Ilustración 31: Control discreto ante seguimiento de referencia por PID, comparado con datos del experimento inicial, saturado según valores comerciales.

Como podremos comprobar en los datos extraídos del experimento, veremos que, efectivamente, el sistema ya no contempla la posibilidad de tener dosis negativas de naltrexona. No obstante, en la estimación de la respuesta del paciente mediante el control discreto, podemos observar que se alcanzan niveles negativos de síntomas.

Antes de poder hacer propuestas para solucionar este problema, nos faltará tener en cuenta una

restricción más de nuestra señal de control, la naltrexona, relacionada con las dosis comerciales que pueden adquirirse en farmacias.

Según ciertos artículos, las dosis de naltrexona de adquisición comercial que, por tanto, pueden recetarse a pacientes no-ingresados (como es nuestro caso de pacientes con afección fibromiálgica) son: 0 mg, 1.92 mg, 3.85 mg, 5.76 mg, 7.68 mg, 9.6 mg, 11.58 mg, 13.5mg. [4] [7]. Además, también se advierte en dichos artículos que no debe incrementarse o decrementarse la dosis más de dos valores superiores o inferiores respectivamente en días seguidos (esto es, que si el paciente está tomando 1.92 mg, las únicas dosis que podría tomar el día siguiente serían 0 mg, 1.92 mg, 3.85 mg y 5.76 mg) [4].

Por todas estas restricciones, hemos decidido introducir en nuestro sistema un módulo “dosificador”, que se encargue de convertir la salida del controlador en una salida apta, teniendo en cuenta cuáles son los valores de dosis posibles y las restricciones de máximas variaciones de dosis en días consecutivos. El código de dicho módulo puede encontrarse en el *Anexo 3*. Para introducirlo en el montaje de simulación, solo necesitaremos hacer uso de un bloque tipo “Interpreted MATLAB Fcn” y darle como entrada los datos que necesite el código. En nuestro caso, el control propondrá una dosis para el día actual de simulación; dicha dosis se introducirá en el módulo de dosificación y será evaluada según el valor anterior de la dosificación y el valor de entrada (así sabremos el rango en el cual puede variar), aproximando el valor de salida del controlador a un valor comercial inocuo para el paciente (puesto que, al fin y al cabo, no solo pretendemos bajar los síntomas de fibromialgia que sufre sino, por supuesto, evitar que su calidad de vida no mejore debido a efectos secundarios).

Tras realizar simulaciones (véase *Ilustración 32* e *Ilustración 33*) podremos comprobar de manera sencilla cómo las restricciones de la señal de control se cumplen a la perfección, y se consigue una salida lógica e inocua para el paciente.

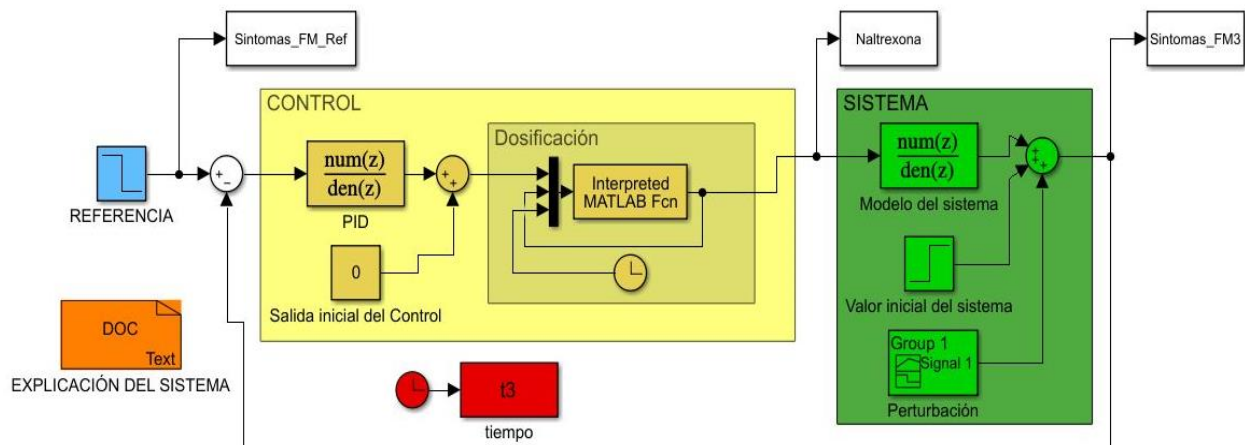


Ilustración 32: Montaje Simulink® para la implementación de un controlador PID discreto en el sistema diseñado, con dosificación a la salida del PID

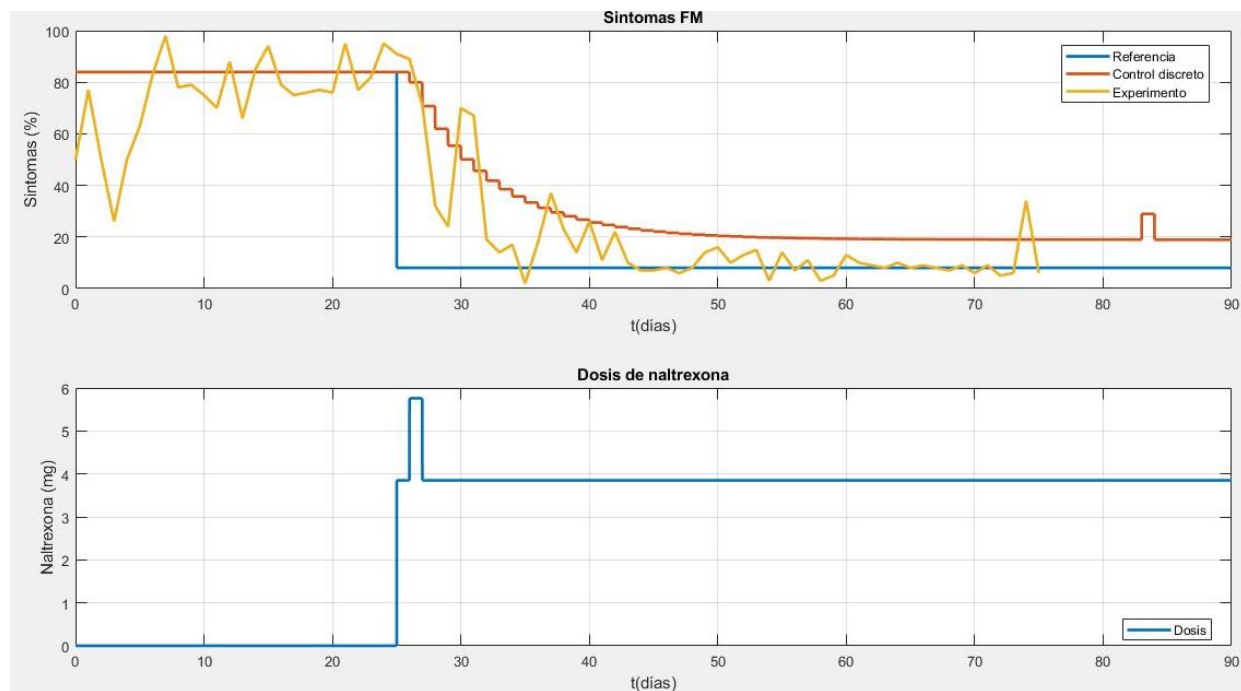


Ilustración 33: Control discreto ante seguimiento de referencia por PID, comparado con datos del experimento inicial, dosificado según valores comerciales.

No obstante, el sistema controlado jamás será capaz de alcanzar la referencia del 8% de síntomas.

Este efecto se debe a que, para que el *PID* mantenga el valor de la salida del sistema en el 8%, necesita que el valor de la dosis de naltrexona sea de 4.5mg (como se puede corroborar en cualquiera de las respuestas de las simulaciones anteriores). Sin embargo, entre las dosis comerciales posibles no existe la de 4.5mg, siendo las más cercanas la de 3.85 mg y la de 5.76 mg, por lo que el sistema, que intenta constantemente dar a la salida 4.5mg, caerá también de forma constante a darnos la dosis comercial más cercana, la de 3.85mg.

Este problema podría solucionarse de múltiples formas, puesto que combinaciones alternas de dosis de 3.85mg con 5.76mg podría provocar que el sistema oscilase de forma leve en torno al 8% de referencia. No obstante, esta característica no es común en el *PID* y debería ser forzado a ello (por ejemplo, incrementando artificialmente su error en situaciones como esta para que el sistema crea estar más alejado de la referencia de lo que realmente está e incremente la dosis en consecuencia).

Por otro lado, esto podría ocasionar que surja la necesidad de estudiar múltiples situaciones de referencia, pudiéndose generar mayores niveles de error por modelado en un sistema que, de por sí, tenía como dificultad ser muy sensible (puesto que no estamos dejando de hablar de la calidad de vida de un ser humano en un ámbito tan sensible como es la salud).

Una forma más sencilla de solventar este problema es consiguiendo que el sistema tenga conciencia de lo que está provocándose en la salida (con algo más que con la realimentación que ya está implementada, por supuesto), previendo cómo va a evolucionar y actuando en consecuencia según eso. Así sabría, por ejemplo, que jamás alcanzará la referencia de 8% usando dosis constantes de 3.85mg.

No obstante, pretender que el sistema “vea el futuro”, analice el “pasado” y actúe según un análisis de ellos no es algo inherente al *PID*.

Por ello, propondremos ahora el uso de un *MPC*, un control predictivo que estimará la evolución futura

del sistema y la analizará, junto con un registro de la evolución en instantes pasados, para actuar en situaciones como esta y evitar dificultades debido a la dosificación (además de permitiendo, como ocurre con el *MPC*, una personalización más sencilla y directa del problema para cada paciente particular).

3.3 Control del proceso mediante técnicas predictivas (MPC)

En numerosas ocasiones a lo largo de esta memoria hemos apoyado el hecho de que un control del tipo predictivo es, por muchos aspectos, una elección óptima para nuestra aplicación de la ingeniería de control a la estimación de un tratamiento contra la fibromialgia. Sin embargo, será este apartado el que nos permita comprobar si el control predictivo (en nuestro caso concreto, su vertiente *MPC*) es realmente eficaz para un caso como el nuestro, donde se precisa niveles de personalización, eficacia y poca cota de error.

Es conocido que, debido a sus características como control del tipo avanzado, el *MPC* consume mayor cantidad de recursos computacionales que otros controladores menos exigentes. No obstante, y para eliminar ya la posibilidad de desechar el *MPC* por el gasto de recursos, no propondremos utilizar un modelo Simulink[®], sino que directamente generaremos un código que ejecute el *MPC* sobre nuestro sistema, lo que hará que se usen una gran cantidad de recursos adicionales menos (puesto que Simulink[®], que facilita las creaciones de aplicaciones de simulación, hace uso de gran cantidad de recursos debido a las constantes llamadas a funciones externas).

Igual que ocurrió con el *PID*, no pretendemos explicar toda la teoría que sustentará nuestro código, puesto que existe gran cantidad de material en la cual puede aprenderse más sobre el *MPC* de lo que aquí pudiésemos explicar (más información en *Anexo 2* y *Anexo 3*). Por otro lado, somos conscientes de que la base del *MPC* ya ha sido explicada en esta memoria, y creemos que con eso es suficiente.

Lo que sí expondremos es, igual que ocurrió en el apartado 3.2, cómo vamos a pasar de los datos que ya tenemos a poder aplicar un control predictivo.

Para ello, partiremos de la base de que el controlador *MPC* necesitará unos parámetros que adecúen el control a nuestro sistema, y serán dichos parámetros los que expondremos a continuación:

- 1) Primero, necesitaremos el modelo de respuesta del paciente, que puede ser una función de transferencia discreta como la nuestra; así que usaremos la función de transferencia Gz ya calculada.
- 2) Definimos valores de ventana, horizontes y parámetros varios (como el de ponderación). Estos parámetros son característicos del control predictivo (algunos, concretamente, del *MPC*) y permiten ajustar detalles del funcionamiento (por ejemplo, cuánto queremos que prediga el control -lo que se conoce como la ventana de predicción).
- 3) Aplicamos el *MPC* y su álgebra asociada (funciones de costo, ecuaciones diofánticas, vectores de controles pasados y futuros, bucle de control...)
- 4) En dicho bucle, y para ajustar la aplicación a parámetros parecidos a los del experimento inicial, tocará definir la duración de la simulación en días, las referencias, las perturbaciones y los valores iniciales de salida y entrada de las señales que interactúan con Gz (recordemos que Gz actúa según variaciones de señales; de ahí que tengamos definir desde qué punto se miden las variaciones, que serán los puntos iniciales).

5) Al finalizar el bucle, ya se ha llevado a cabo la simulación del sistema con *MPC*.

Como podemos comprobar, no hemos introducido saturaciones ni restricciones al tratamiento aún, puesto que pretendemos primero analizar el sistema comparándolo con el experimento inicial.

Para ello, usaremos comandos de MATLAB[®] de la misma naturaleza que los ya expresados para crear gráficas (más información en *Anexo 3*).

Al hacer eso, observaremos que el control predictivo podría ajustarse perfectamente al experimento si así lo necesitásemos (mediante escalones de referencia que, gracias a la predicción, se superpone la respuesta a ellas, puesto que el control predictivo no solo busca llegar a la referencia rápidamente, sino que ante cambios de referencia dentro del horizonte de predicción, trata de generar una única salida compuesta que pueda ajustarse a las dos referencias de la forma más eficaz, a diferencia del *PID*, que trataría de llegar a la primera antes de llegar a la segunda). Podemos asegurar también una mayor velocidad por parte del *MPC* en comparación con el *PID*, puesto que hemos necesitado hacer una bajada escalonada hasta llegar al 8% de referencia para que no cayesen los síntomas tan rápido que no se pudiese comparar con el experimento inicial. Además, queremos destacar que rechaza perturbaciones de forma eficaz y que, aunque no ha sido saturado ni dosificado, el *MPC* permite en su llamada “función de costo” (función central del algoritmo de control) exponer una máxima variación posible de la señal de entrada (dosis de tratamiento), que hemos definido en 15mg, ya que el ajuste no es trivial, para que pueda considerarla en sus iteraciones de control de la forma más eficaz posible. (Véase *Ilustración 34*).

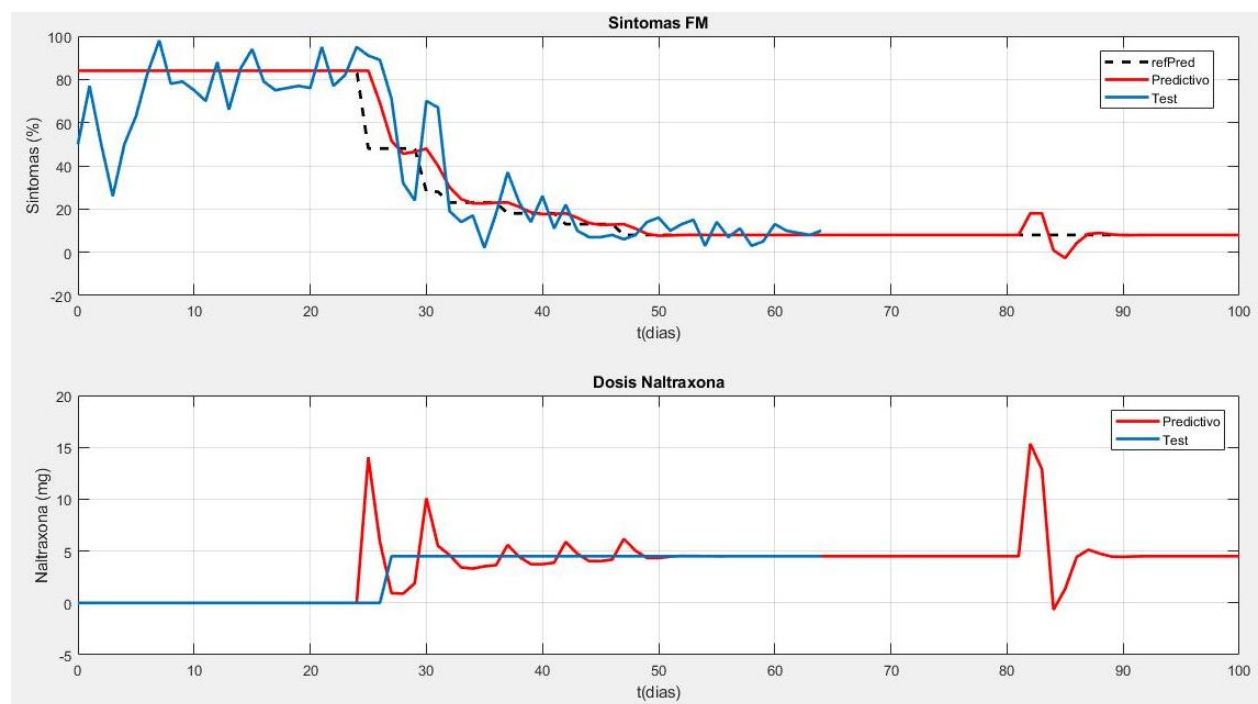


Ilustración 34: Control discreto ante seguimiento de referencia por MPC, comparado con datos del experimento inicial, sin dosificación.

Tras extraer conclusiones de la simulación anterior, pasaremos a introducir la dosificación y saturación a nuestra aplicación controlada según métodos predictivos.

Para ello, será tan sencillo como extraer la función de dosificación usada en los sistemas controlados por *PID* e introducirla en nuestro sistema en el bucle de control, justo cuando se actualiza la salida del controlador.

Ciertamente, existen métodos internos al control predictivo que permiten definir un valor de variación en la señal de control. Si hiciésemos esto, imponiéndole un valor cualquiera, la señal de control (la

dosis de naltrexona) solo sería expuesta a la salida del controlador con magnitudes que fuesen múltiplos de dicho valor de variación. Esto provocaría, en un principio, que las respuestas del controlador tuvieran un factor de rizado menor que si no se usase esta opción (es decir, oscilaría menos veces al tratar de mantener un valor constante de salida no alcanzable por uno de los múltiplos de la variación definida en la entrada). No obstante, la dosificación impuesta comercialmente en la naltrexona no se trata de una señal de este tipo, puesto que sus valores discretos no son múltiplos de factor constante frente a un mínimo común múltiplo, por lo que, aunque usásemos las técnicas anteriores para forzar una discretización a la salida del controlador, seguiríamos necesitando el módulo de dosificación usado en el control por *PID*.

No obstante, es cierto que aplicar ambas opciones a la vez suaviza el rizado en valores de dosificación cercanos a valores medios (dosis 7.68 mg) en algunos puntos de trabajo. Sin embargo, también provoca que, en dichos puntos, la aplicación de dosis más extremas (0, 1.92, 11.58, 13.5 mg) se vea condicionada por mayores rizados que en el método inicial de solo utilizar la dosificación.

Por tanto, aunque hay mejoría en ciertos puntos de trabajo para ciertos valores de dosificación, también hay mayores rizados para otros valores diferentes, sin olvidar el hecho de que se da un mayor costo computacional que usando solo la dosificación ya diseñada y comprobada para el *PID*.

Además de esto, y aunque estamos adelantando información de apartados siguientes de esta memoria, la mejoría de rizado es claramente dependiente del punto de trabajo y, debido a la necesidad de personalización para cada paciente de nuestra aplicación y a la implementación de revisiones médicas, el punto de trabajo se verá modificado en numerosas ocasiones, atravesando valores en los cuales la aplicación dosificada mediante solamente el código aplicado en el control *PID* tendrá rizados despreciables.

Con esto, queremos corroborar el hecho de que es mayormente un costo a cambio de un beneficio ínfimo en nuestro caso particular.

Por lo tanto, y una vez aplicada la dosificación ya usada para el *PID* a nuestro control por *MPC*, podremos ejecutar la aplicación y analizar resultados.

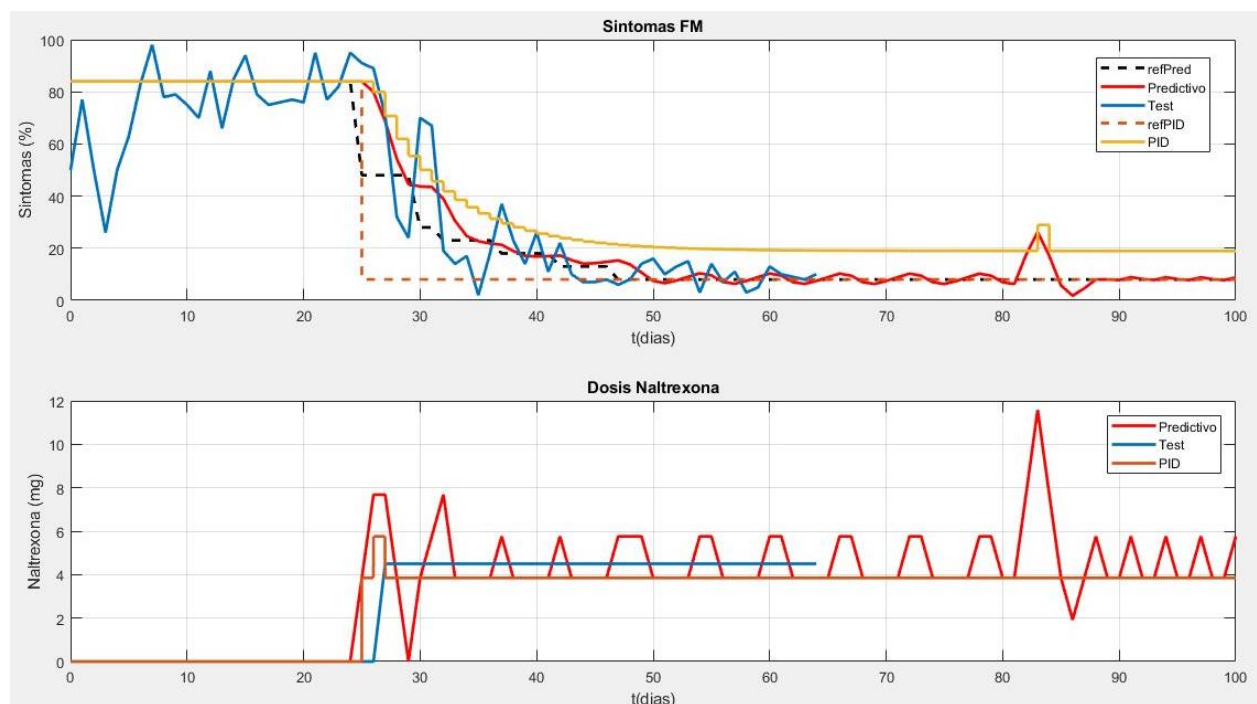


Ilustración 35: Control discreto ante seguimiento de referencia por MPC, comparado con datos del experimento inicial y con el PID dosificado, introduciendo dosificación en el MPC.

Al igual que antes, el tiempo que tarda en llevarse a cabo la simulación es menor que el de PID por montaje Simulink[®]. Y podremos apreciar que, además de tener que forzar la lentitud con referencias escalonadas para que las respuestas transitorias ocurran a la vez (esto, al igual que antes, se lleva a cabo solo por realizar un análisis comparativo; por supuesto, nos interesa que el sistema sea lo más rápido posible, siempre y cuando cumpla con las restricciones asociadas que evitan cualquier tipo de complicaciones de salud para el paciente), el sistema sí consigue llegar a la referencia final del 8%, a diferencia de lo que ocurría con el PID, mediante un rizado en el tratamiento que alterna dosis de 3.85 y 5.76 mg (véase *Ilustración 35*).

Si tomásemos el cruce de datos (tiempo-tratamiento), podríamos generar una tabla clara de dosis diarias que debiera tomar el paciente para que los síntomas de su enfermedad se redujesen de forma drástica en apenas unos días (recordemos que hemos forzado una caída lenta, pero no será nuestra intención hacerlo en un caso real, como se comprobará en el siguiente apartado de esta memoria).

Por tanto, tras comparar la aplicación apoyada sobre un *PID* y la aplicación apoyada sobre un *MPC*, la resolución definitiva por parte de este trabajo será mantener el control *MPC* para nuestra aplicación de análisis de la fibromialgia como sistema de control en beneficio de sus pacientes.

Somos conscientes de que el tratamiento generado por nuestra aplicación no puede considerarse como un tratamiento "simple". En numerosas ocasiones, cuando un paciente acude a consulta con un especialista de la salud, dicho especialista le receta un tratamiento basado en cambios leves de dosis (la mayoría de los tratamientos cuentan con las mismas dosis a lo largo de grandes períodos de tiempo o con diferentes dosis a lo largo de los días que tienen repercusiones cíclicas de corto período -por ejemplo, todos los lunes y jueves la misma dosis, que es diferente a la de los martes y miércoles, pero que supone un ciclo de repetición de una semana). Nuestra aplicación, sin embargo, propone dosis diarias de tratamiento para el paciente, pero lo hace a cambio de una gran eficacia y (como veremos en el siguiente apartado) una personalización mucho más directa que permite un ratio de efectividad considerablemente mayor que un tratamiento tradicional médico. El paciente solo necesitaría marcar un calendario de dosis que debe seguir cuidadosamente entre una cita y la siguiente cita con el especialista.

¿Y por qué hablamos de "entre una cita y la siguiente"?

Como bien dijimos al comienzo de esta memoria, la aplicación debe funcionar bajo el apoyo de supervisión médica que permita la personalización y aceptación del tratamiento de las fechas entre citas. Por ello, pasaremos ahora a implementar esta capacidad de supervisión y personalización del sistema por parte del especialista, para una adaptación más cercana a la realidad de cada paciente particular.

4. Personalización y revisiones médicas

"El hombre es la única criatura que se niega a ser lo que es"

Albert Camus.

Tras conseguir generar un módulo de control capaz de crear tratamientos eficaces contra la fibromialgia, que ayuden al paciente disminuyendo posibles efectos secundarios y que permita aumentar su calidad de vida hasta altos niveles, nos quedará apenas un paso para dar por terminado este estudio.

Este punto final consistirá en introducir la última fracción de nuestro sistema planteado (véase *Ilustración 4*). La finalidad será modificar la aplicación para dar facilidades al equipo médico para la personalización y los cambios de tratamiento según las evaluaciones por parte de los especialistas a los pacientes en cada una de las citas concertadas con ellos.

Para este fin, crearemos una opción en el sistema con el cual el médico pueda introducir, tras cada cita médica, si el paciente se encuentra mejor, igual o peor de lo esperado.

Con el fin de comprenderlo de forma sencilla, clarificaremos que, para evaluar los niveles de “mejor”, “igual” o “peor”, el especialista necesitará evaluar al paciente en su estado actual el día de la cita, así como interesarse por conocer su parecer sobre el paso de los días entre cita y cita.

Sabemos que, hasta ahora, hemos supuesto que, entre cita y cita, el paciente tomaría el tratamiento que el sistema ha creado (más allá de cualquier aprobación o desaprobación médica que pudiera dar el especialista). Por ello, será en el momento de la cita (y mucho más, durante las primeras), que el equipo médico podrá personalizar el tratamiento. La finalidad de esta personalización es que, si tras un tiempo de tratamiento el paciente se encuentra mejor de lo estimado por nuestra aplicación, se pueda bajar gradualmente su tratamiento (volviéndolo a subir cuando sea necesario), de igual forma que podrá subirse su dosis si su percepción de los síntomas es más sensible que la de los pacientes del test en el cual hemos basado nuestra aplicación. Al fin y al cabo, nosotros hemos propuesto que, de media, un paciente sin tratamiento sufre un 84% de síntomas; por supuesto, este valor es variable según lo que el especialista quiera introducir en la primera cita, pero, ¿qué es un 84% de síntomas cuando algunos de ellos son tan íntimos como el dolor? ¿Qué marca el 100%? ¿Cómo hará el especialista para saber si mi 100% es igual que el de otro paciente? Desgraciadamente, esto último depende, claramente, de prueba y error, así como provocar la necesidad de personalizar los límites en cada paciente. Debido a esto, y tratando de conseguir que ese proceso de “prueba y error” sea el más eficaz y corto posible, desde

nuestro trabajo queremos proponer la modificación del punto de trabajo de una forma tan simple como la que hemos explicado.

Si se había estimado que el paciente sufriría un 20% de síntomas antes de venir a la cita actual, pero sufre más, significaría que el nivel “20%” de nuestro modelo de respuesta del paciente no es el mismo que el del paciente actual (quizá porque la dosis le afecta menos, o porque sus umbrales son mayores), así que subiríamos el punto de trabajo, de forma que el control proponga una dosis mayor para alcanzar el susodicho 20%, aumentando por tanto el significado del nivel “20%” de nuestro modelo de respuesta del paciente. De igual forma, podría hacer que, si el paciente lleva un tiempo sintiéndose bien y sin síntomas de fibromialgia dentro de unos márgenes aceptables, se pudiese ir bajando su dosis de forma gradual indicando que el paciente se encuentra “mejor”; es decir, mejor de lo esperado.

Para llevar esta modificación a cabo, haremos una serie de modificaciones leves en el código que permitan la modificación del valor del punto de trabajo (más información en *Anexo 3*), con el fin de generar cambios en el significado de los niveles de fibromialgia y, por tanto, adaptarlos personalmente al paciente actual.

Fundamentalmente, el cambio se basará en que, tras cada cita, el médico introduzca la valoración comparativa (“peor” se caracterizará con un valor “100” en cierta variable, “igual” con un valor “50” y “mejor” con un “0”) y se incrementará o decrementará el valor del punto de trabajo actual una cantidad porcentual según dicha valoración médica, proponiendo a continuación un tratamiento consecuente con dicha modificación, de forma que el sistema vuelva a alcanzar el valor de la referencia en una cantidad de tiempo mínima, habiéndose dado el cambio interno de significado de los porcentajes de síntomas (así pues, tras una cita médica, si no se ha marcado “igual”, el significado de un nivel no será el mismo que el de antes de la cita, aunque en ambos casos tengan asociado el mismo número porcentual).

A continuación, lo plasmaremos con un ejemplo que nos permitirá analizar todas estas posibilidades de forma simple.

Para poder crear este ejemplo y cualquiera de los posteriores, modificamos la aplicación para que el especialista pudiera introducir los datos equivalentes a las citas médicas y sus valoraciones periódicas. Para ello, en nuestro código cuyo esquema básico se definió en el apartado anterior, introduciremos, antes de nada, los días totales que durará el estudio (sería equivalente a los días que han pasado hasta ahora desde que el paciente llegó a la cita inicial, sumándole también los días que quedarán hasta la siguiente cita -la finalidad de esto será analizar el recorrido histórico del paciente y permitir a la aplicación generar un tratamiento para los días entre la cita actual y la siguiente).

Lo siguiente se basará en imponer cómo ha sido, hasta ahora, la respuesta del paciente durante las revisiones (si se ha encontrado “mejor”, “igual” o “peor” durante el periodo entre la cita actual y la anterior, de lo que había estimado la aplicación). Por simplificación, en nuestros ejemplos hemos expuesto las citas periódicas y las respuestas aleatorias, pero sería tan sencillo como hacer una matriz de dos columnas, que relacione el día de la cita con la respuesta durante la revisión.

Tras esto, solo debemos irnos al momento del código donde se modifica el Offset de la salida. Modificar dicho Offset de forma dinámica, según los resultados de cada una de las citas, será lo que nos permitirá ajustar el punto de trabajo a uno más adecuado para nuestro paciente (es decir, modificar el significado de los límites porcentuales para nuestra aplicación). Por tanto, tras cada revisión, se observa el resultado de la misma (“mejor”, “peor” o “igual”) y se modifica el punto de trabajo de inmediato de ser necesario.

Cabe destacar que esta modificación del punto de trabajo ha sido ejecutada de forma arbitraria en nuestra aplicación, ya que queríamos crear un ejemplo visualmente representativo y no tenemos datos sobre errores de sensibilidad porcentual del experimento inicial. Este valor, sin embargo, debería ser

un número pequeño o variable según la consideración del equipo médico, para realizar un ajuste más personalizado al paciente. En nuestro caso, decidimos que esta variación tendría un valor arbitrario del 12% (como argumentamos, visualmente permite una mayor comprensión visual).

Por supuesto, debemos eliminar la perturbación provocada en simulaciones anteriores para medir efectos de las mismas sobre el sistema, por el simple hecho de que no tenemos información sobre cuándo podrán producirse dichas perturbaciones, así que nuestro sistema de control contra la fibromialgia, a pesar de poder rechazarlas, no podrá estimar cuándo o de qué intensidad podrían ser futuras perturbaciones; será trabajo del equipo médico, durante las revisiones, analizar si un malestar en el periodo entre citas es una perturbación o necesita ser catalogado como “peor” respecto a lo esperado, para que se modifique el punto de trabajo (generalmente, las perturbaciones se caracterizan por ser puntuales).

Con todo esto, acabaríamos de generar la aplicación final de nuestro trabajo, a falta de analizar sus resultados (más información en *Anexo 3*).

Para comenzar, probaremos una simulación de 200 días con revisiones cada 15 días y setearemos los síntomas inicial del paciente (analizados). Como dijimos, la cita actual es la penúltima cita cuyos datos se muestren (equivalente al día 180), y más allá de la siguiente cita (el día 195) debe ser desechado, ya que la respuesta en esta última cita es arbitraria y solo sirve para evitar que el sistema no tenga un buen funcionamiento para algunos tiempos de simulación. Por tanto, nos pondremos en la piel del equipo médico que, en el día 180, decide usar de nuevo la aplicación. Para todos los futuros casos, habremos definido la referencia como 8% (por el simple hecho de tratar de mantener la referencia anterior; lo ideal sería darle de valor el 0%, para que se palien todos los síntomas del paciente frente a la fibromialgia padecida). Y también tomaremos que, durante la primera cita, habíamos observado un histórico de síntomas entorno al 84%, lo que nos daría el valor inicial de la salida.

Supongamos que, en un primer caso hipotético, hemos comprobado que los síntomas de nuestro paciente se ajustan perfectamente a las estimaciones de la aplicación tanto a día de hoy como en los días entre una cita y la anterior. Por ello, en cada una de las revisiones habremos marcado “igual” como el análisis del paciente durante la misma, y por ello, las revisiones tendrán un valor constante de “50” en las gráficas que la aplicación nos mostrará por pantalla.

Correremos pues el sistema y comprobaremos el tratamiento que nos propone. Si, como equipo médico, no estuviéramos de acuerdo con el mismo (porque, por ejemplo, tras 180 días con unos síntomas del 8%, queremos ir reduciendo la dosis de naltrexona que esté tomando el paciente), solo necesitaríamos modificar la resolución de la cita actual (en el caso planteado, sería marcar un “mejor”, que dará un valor de “0” y reducirá la dosis administrada al paciente, sin dejar de tratar que los síntomas sigan encontrándose entorno al 8%). Sin embargo, si estuviéramos de acuerdo, solo tendremos que indicarle al paciente que siga con el tratamiento que ya tenía impuesto (véase *Ilustración 36*).

Desde nuestro estudio, queremos remarcar un hecho ya comentado y que ahora se corrobora, y es que el *MPC* se encargará de que el paciente tenga un tiempo de transición mínimo entre el valor actual de los síntomas y la referencia puntual requerida sin que por ello se dejen de respetar las restricciones de seguridad de la naltrexona (de ahí que, en la *Ilustración 36* se observe que se pasa del 84% de síntomas al 8% con completa estabilidad en tan solo 8 días, a diferencia del *PID* con dosificación, que tardaba aproximadamente 28 días en estabilizarse sin llegar a la referencia nunca).

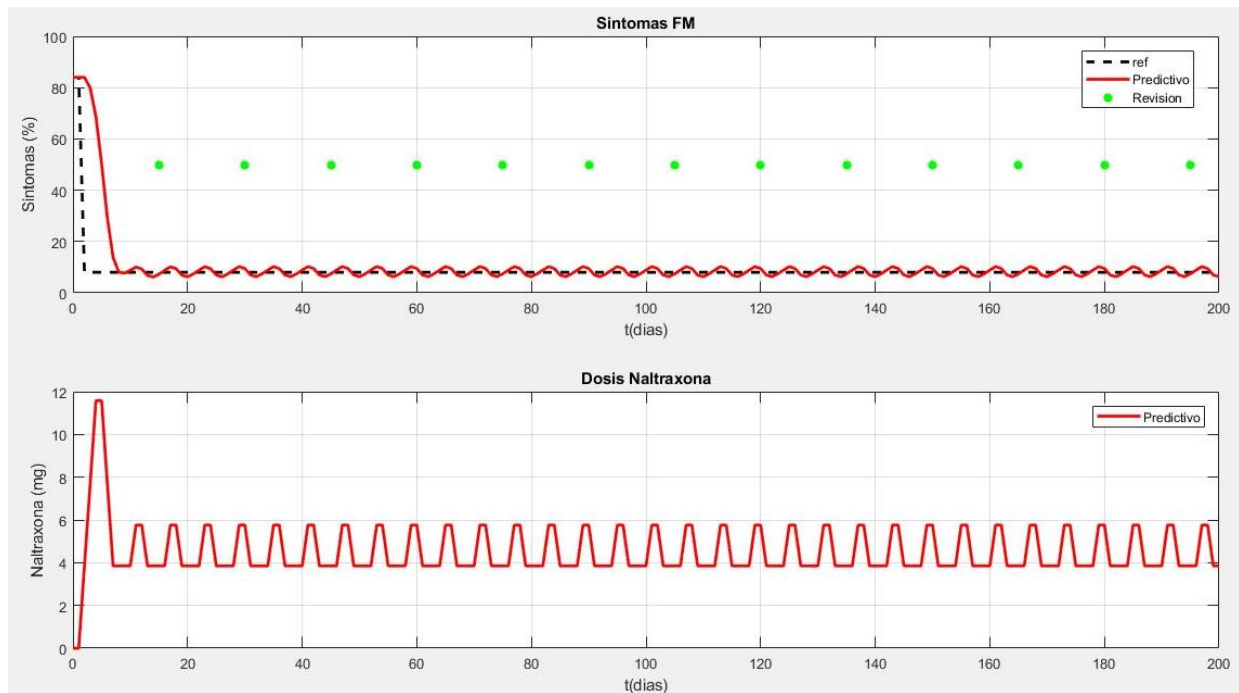


Ilustración 36: Sistema de control completo sobre un paciente con fibromialgia, con revisiones médicas y personalización incluidas. Ejemplo con revisiones con valor cualitativo “igual que la estimación del sistema”, marcado con un valor cuantificado de 50.

Otros dos ejemplos sencillos que expondremos a continuación se basan en qué ocurriría si, durante las revisiones, hemos observado repetidamente que el punto de trabajo no se ajusta al paciente (ya sea porque se siente continuamente “mejor” de lo estimado por la aplicación o “peor” que lo estimado por la aplicación).

En el caso de que el paciente se sienta constantemente mejor de lo esperado, el punto de trabajo irá reduciendo su valor hasta llegarse al caso en el que se estima que el paciente no necesita más medicamento para no sufrir ningún síntoma (por supuesto, en el futuro podría encontrarse “peor” durante una revisión y volver a necesitarlo). (Véase *Ilustración 37*).

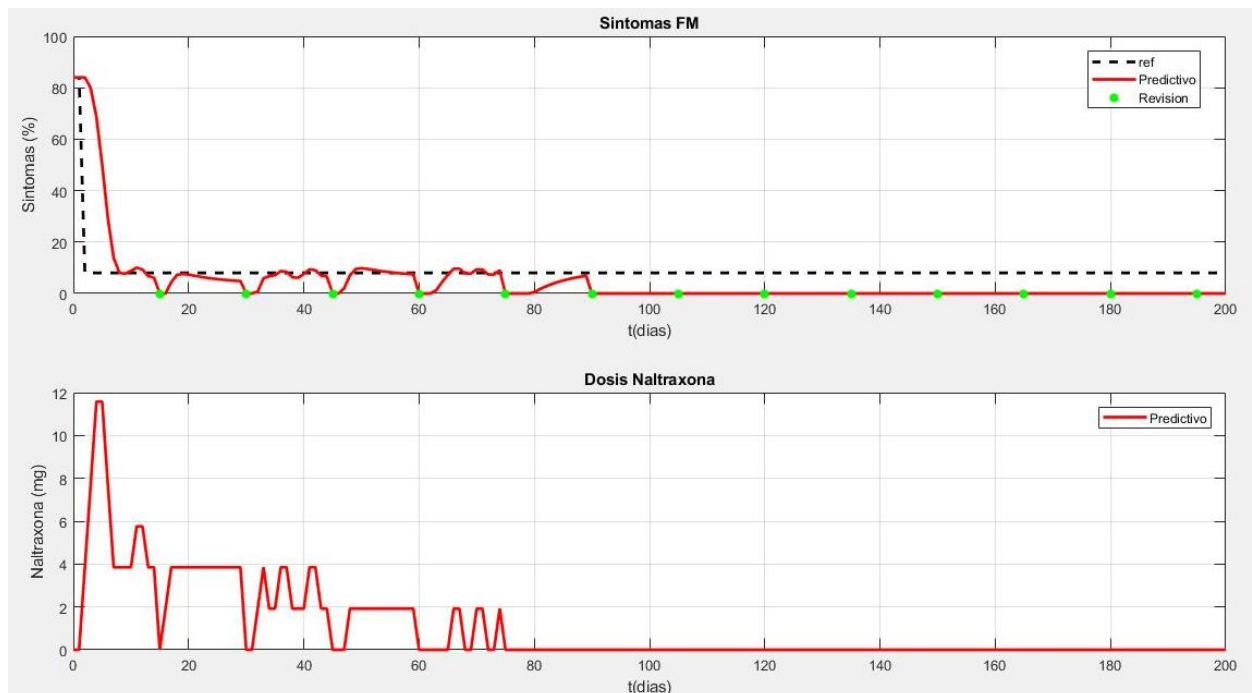


Ilustración 37: Sistema de control completo sobre un paciente con fibromialgia, con revisiones médicas y personalización incluidas. Ejemplo con revisiones con valor cualitativo “mejor que la estimación del sistema”, marcado con un valor cuantificado de 0.

Por otra parte, en caso de que se sienta constantemente peor de lo esperado, el punto de trabajo irá aumentando su valor hasta llegar a un punto donde se puede comprobar que ni siquiera la mayor dosis de naltrexona es efectiva en el paciente. (Véase *Ilustración 38*).

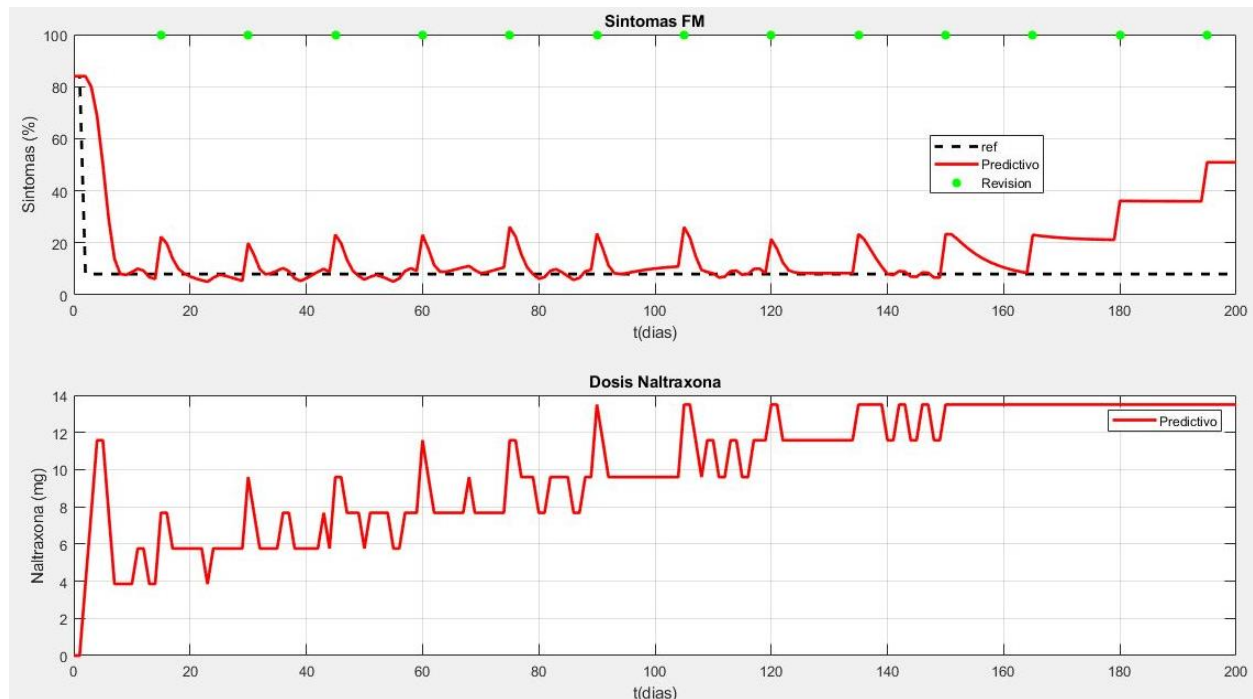


Ilustración 38: Sistema de control completo sobre un paciente con fibromialgia, con revisiones médicas y personalización incluidas. Ejemplo con revisiones con valor cualitativo “mejor que la estimación del sistema”, marcado con un valor cuantificado de 100

Cabe destacar que siempre que se dé un “mejor” o “peor” en una revisión, el sistema nos dejará entrever un pico mínimo o uno máximo, respectivamente, en el día siguiente a la revisión, debido a que el sistema ha recalculado el punto de trabajo.

Por supuesto, estos casos anteriores no reflejan el funcionamiento real del sistema, debido a que hemos forzado a las revisiones a tener valores constantes para mostrar evoluciones.

A continuación, realizaremos varios experimentos donde se podrá comprobar el correcto funcionamiento de nuestra aplicación con diversos pacientes hipotéticos a los cuáles se les ha hecho un seguimiento por revisión cada quince días (véase *Ilustración 39*, *Ilustración 40* e *Ilustración 41*).

También existirá la posibilidad, como ya dijimos, de usar otras referencias (véase *Ilustración 42*) o de cambiar la periodicidad de las citas (*Ilustración 43* e *Ilustración 44*).

Con esto, se dan por cumplidos todos los objetivos de este estudio, y se da por cerrado el mismo, tras haber demostrado la viabilidad de una aplicación capaz de estimar tratamientos eficaces contra la fibromialgia capaces de mejorar, como ayuda para los especialistas, la calidad de vida de los enfermos de este desagradable desorden crónico.

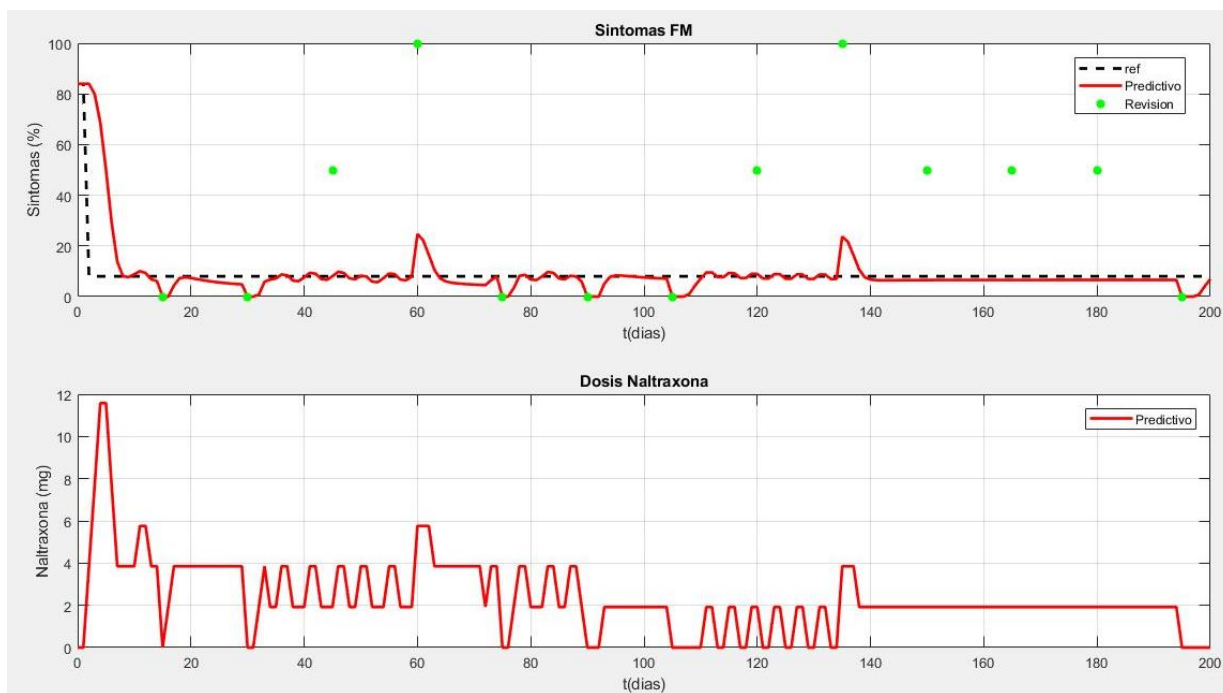


Ilustración 39: Sistema de control completo sobre un paciente con fibromialgia, con revisiones médicas y personalización incluidas. Ejemplo primero.

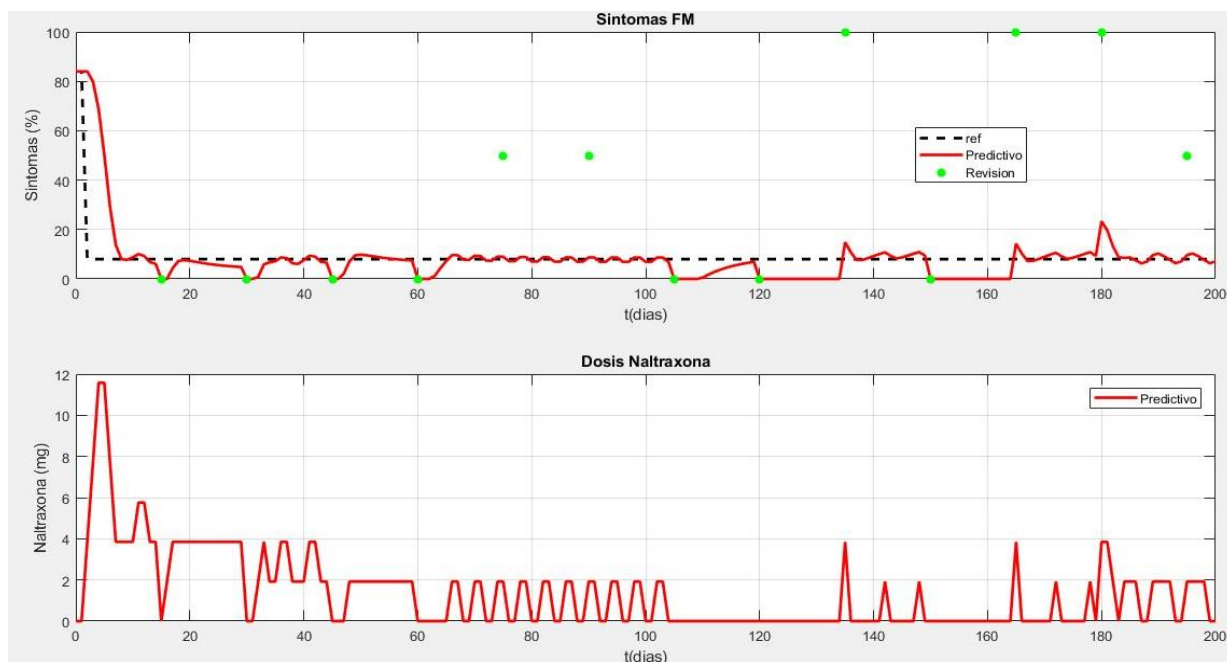


Ilustración 40: Sistema de control completo sobre un paciente con fibromialgia, con revisiones médicas y personalización incluidas. Ejemplo segundo

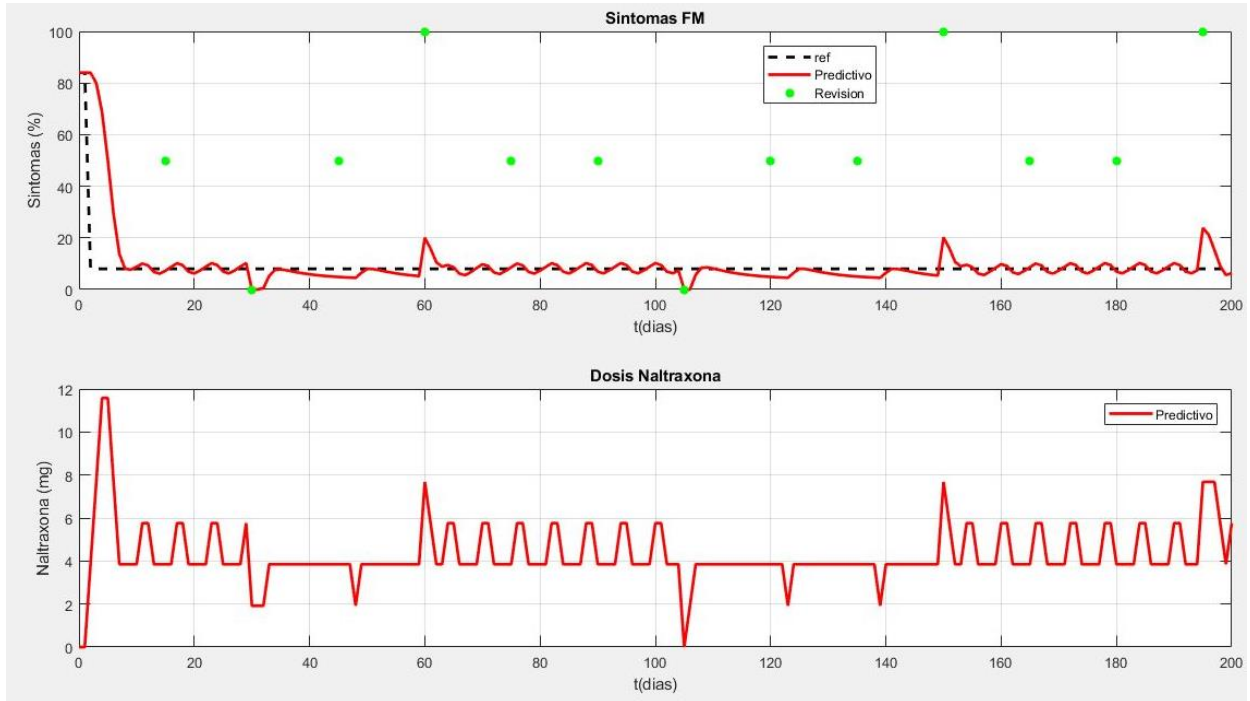


Ilustración 41: Sistema de control completo sobre un paciente con fibromialgia, con revisiones médicas y personalización incluidas. Ejemplo tercero

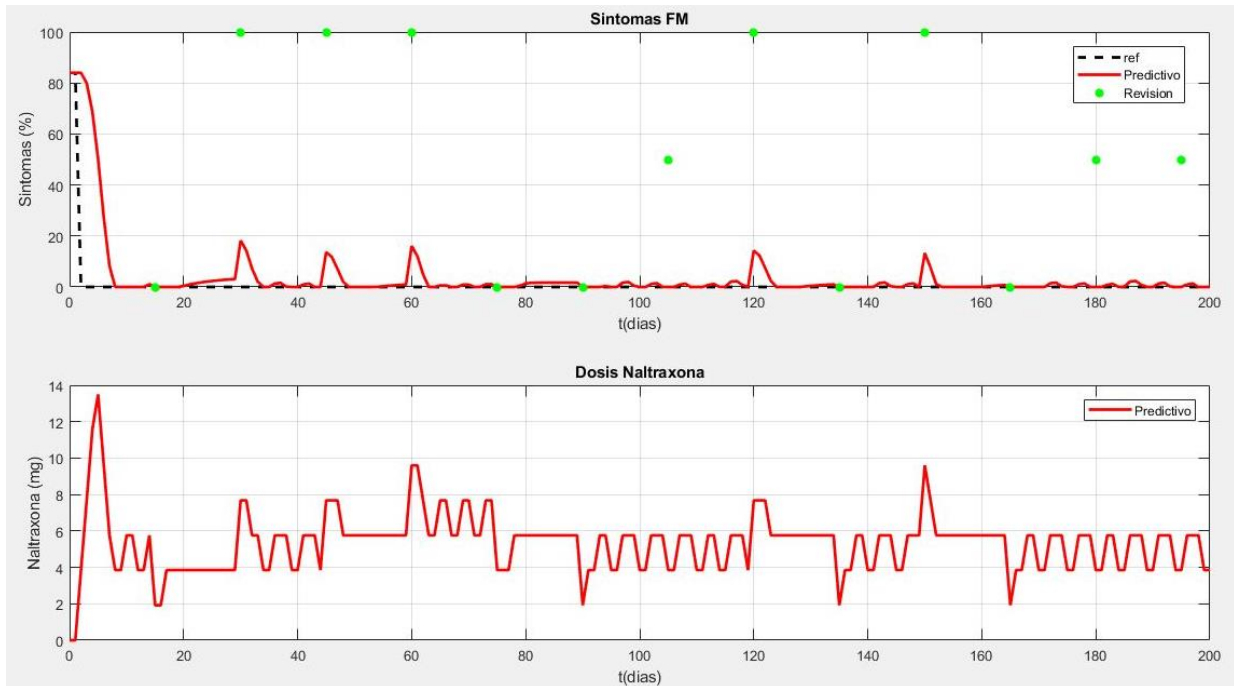


Ilustración 42: Sistema de control completo sobre un paciente con fibromialgia, con revisiones médicas y personalización incluidas. Referencia al 0% de síntomas

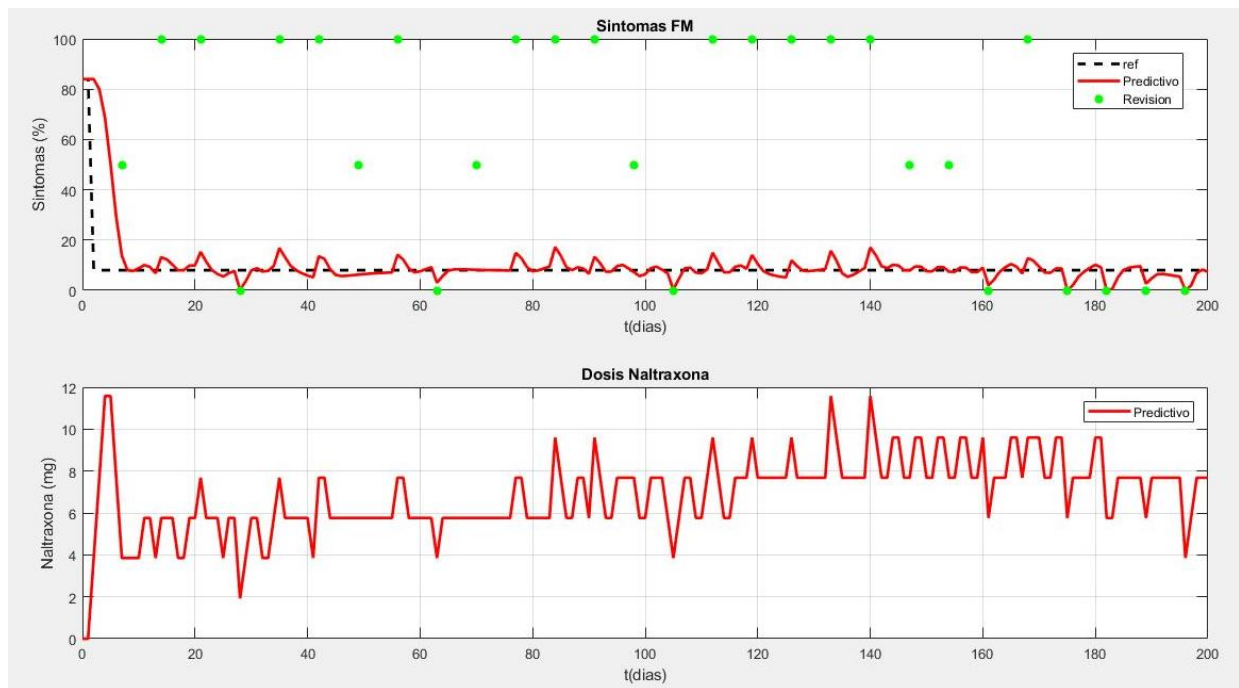


Ilustración 43: Sistema de control completo sobre un paciente con fibromialgia, con revisiones médicas y personalización incluidas. Tiempo de periodicidad de citas de 7 días.

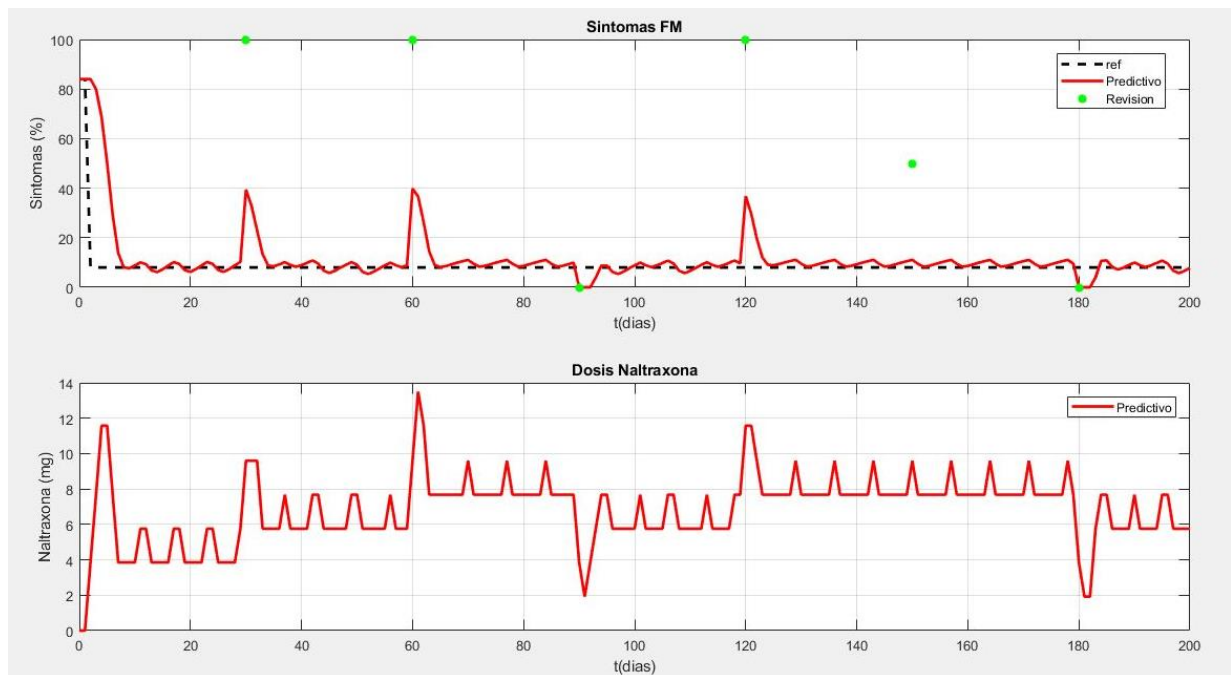


Ilustración 44: Sistema de control completo sobre un paciente con fibromialgia, con revisiones médicas y personalización incluidas. Tiempo de periodicidad de citas de 30 días.

5. Referencias

- [1] Younger, J. et. al., «Fibromyalgia Symptoms Are Reduced by Low-Dose Naltrexone: A Pilot Study,» *Pain Medicine*, vol. 10, n° 4, pp. 663-672, 2009.
- [2] H. PK, «Nonarticular rheumatism, 22nd rheumatism review: review of the American and English literature for the years 1973 and 1974,» *Arthritis Rheum*, vol. 19, n° (suppl), pp. 1081-1089, 1976.
- [3] I. Fatma y Y. Muhammad B, «History of Fibromyalgia: Past to Present,» *Current Pain and Headache Reports*, vol. 8, pp. 369-378, 2004.
- [4] D. Sunil, N. N. Nandola, D. E. Rivera y J. Younger, «A Control Engineering Approach for Designing an Optimized Treatment Plan for Fibromyalgia,» de *Proc Am Control Conf.*, 2011 June 29.
- [5] A. Bemporad y M. Morari, «Toward an integrated understandig of fibromialgia syndrome,» *Pain*, vol. I Medical and pathophysiological aspects, n° 45, pp. 227-238, 1991.
- [6] A. Bemporad y M. Morari, «Toward an integrated understandig of fibromialgia syndrome,» *Pain*, vol. II Psychological and phenomenological aspects, n° 45, pp. 239-248, 1991.
- [7] D. Sunil, N. N. Nandola, D. E. Rivera y J. Younger, «Optimized treatment of fibromyalgia using system identification and hybrid model predictive control,» *Control Engineering Practice*, n° 33, pp. 161-173, 2014.
- [8] W. Riley, D. E. Rivera, A. Atienza, W. Nilsen, S. Allison y R. Mermelstein, «Health behavior models in the age of mobile interventions: Are our theories up to the task?,» *Translational Behavioral Medicine*, n° 1, pp. 53-71, 2011.
- [9] D. E. Rivera, M. D. Pew y L. M. Collins, «Using engineering control principles to inform the design of adaptive interventions: A conceptual introduction.,» *Drug and Alcohol Dependence.*,

nº 88, p. S31–S40, 2007.

- [10] A. Zafra-Cabeza, D. Rivera, L. Collins, M. Ridaou y E. Camacho, «A risk-based model predictive control approach to adaptive interventions in behavioral health.,» *IEEE Transactions on Control Systems Technology*, nº 19, p. 891–901, 2011.
- [11] Wikipedia, «MATLAB - Wikipedia, la enciclopedia libre.,» Wikipedia Foundation., 2 Abril 2018. [En línea]. Available: <https://es.wikipedia.org/wiki/MATLAB>. [Último acceso: 16 Mayo 2019].
- [12] The MathWorks, Inc., «MATLAB,» MathWorks, [En línea]. Available: <https://es.mathworks.com/products/matlab.html>. [Último acceso: 16 Mayo 2019].
- [13] T. M. Mattiloi, B. Milne y C. Cahill, «Ultra-low dose naltrexone attenuates chronic morphine-induced gliosis in rats.,» *Molecular Pain*, nº 6, pp. 1-11, 2010.
- [14] Wikipedia, «Función de transferencia - Wikipedia, la enciclopedia libre.,» Wikipedia Foundation, 2014. [En línea]. Available: https://es.wikipedia.org/wiki/Funci%C3%B3n_de_transferencia. [Último acceso: 5 Mayo 2019].
- [15] The MathWorks, Inc., «System Identification Toolbox,» [En línea]. Available: <https://www.mathworks.com/products/sysid.html>. [Último acceso: 5 Mayo 2019].
- [16] Wikipedia, «Mínimos cuadrados - Wikipedia, la enciclopedia libre.,» Wikipedia Foundation, [En línea]. Available: https://es.m.wikipedia.org/wiki/M%C3%ADnimos_cuadrados. [Último acceso: 25 Mayo 2019].
- [17] M. Gil Aguilares, «Sistema de primer orden, constante de tiempo,» de *Análisis de la respuesta temporal, Asignatura de Fundamentos de Control*.
- [18] N. N. Nandola y D. E. Rivera, «An improved formulation of hybrid model predictive control with application to production-inventory systems.,» *IEEE Transactions on Control Systems Technology*, nº 21, pp. 121-135, 2013.

6. Anexos

Anexo 1: Entrevista a un especialista médico y a un enfermo de fibromialgia.

Entrevistador: Dígame, usted lleva sufriendo esta enfermedad, la fibromialgia, desde hace mucho tiempo, ¿no es así?

Paciente: Realmente no puedo decir cuándo empezó la enfermedad; fue algo muy gradual. Al principio solo fue cansancio, luego además empecé a notar hormigueos y, con el tiempo, se volvió algo insoportable.

E: Pero, ¿esa sensación de que sea insoportable ocurre todos los días?

P: No, por suerte, no; aunque, tristemente, la mayoría de los días sí.

E: ¿Y cómo son esos días donde se encuentra mejor?

P: ¿Aparte de escasos, quieres decir? Realmente he probado muchas cosas contra la fibromialgia, pero, aunque con algunas he conseguido algún día bueno esporádico, no han hecho mucho efecto en mí. Hay pocos días donde pueda decir que me siento “realmente bien”; aunque es verdad que los síntomas y, sobre todo, el dolor, no molesta en ocasiones. Pero, ¿sabes lo peor? Incluso los días donde los síntomas no son tan graves, los síntomas más leves te hacen sentirse aprisionada, como unos grilletes; hay veces que solo tienes un pequeñísimo hormigueo, como un calambre, y no te atreves ni a quedar para cenar con unos amigos porque igual hoy que te encuentras medio bien, pero... ¿mañana podrás con tu cuerpo? ¿Podrás levantarte de la cama?”

E: ¿Se tiene en cuenta este componente psicológico entre los síntomas de la fibromialgia, doctor?

Médico: Ciertamente, muchos artículos lo tienen en cuenta; la psique humana es muy importante en cuestiones de salud, y más aún en desórdenes crónicos como la fibromialgia. Además, hay que partir de la base de que, sobre la fibromialgia, aún hay mucho que investigar, tristemente, para algunos especialistas, la “fibromialgia” es un simple apelativo con el que ponerle nombre a cualquier conjunto de fenómenos de dolor físico con origen posiblemente neurológico que un equipo médico no sepa diagnosticar, con el fin de evadir la responsabilidad de ahondar más en el caso.

E: ¿Y por qué cree que pueda ser esto?

M: Aunque el dolor se produzca en un punto determinado del cuerpo, no hay lesiones ni daños visibles o palpables en la zona, debido a que la enfermedad se basa en un fallo neurológico y no en uno muscular o esquelético; esto llevó, tradicionalmente (y hablo de tiempos donde la neurología no estaba

tan avanzada como ahora) a que, como no había signos físicos aparentes por los cuáles tuvieran explicación esas molestias, se tuviera a la fibromialgia como un mito y se introdujese cualquier síntoma parecido bajo dicho nombre, aunque no se corresponda realmente a la fibromialgia. a primera vista por los cuáles los síntomas tuvieran “sentido”, así que se mitificó la fibromialgia y se decidió popularmente que muchas enfermedades

E: ¿Eso podría ser considerado como una negligencia por parte de los especialistas?

M: Ese es un término muy grave, y más en un caso como este. No trato de decir que no lo sea, simplemente hay que entender que los artículos y estudios más actuales sobre la fibromialgia, que despejan estas dudas y clarifican en mayor medida qué es esta enfermedad y cómo identificarla, así como proponen tratamientos más eficaces, aún no se han extendido a todos los especialistas. Por ello, no creo que sea justo hablar de una negligencia por parte de los especialistas, sino que deberíamos tratarlo como un argumento de suma importancia por el cual deberían extenderse más estos estudios por hospitales y demás sectores médicos (con formaciones, por ejemplo).

P: El problema es que los que más sufrimos esto somos los enfermos.

M: Sí, sin duda; tristemente, todos conocemos lo difícil que es desmitificar un hecho, sea del tipo que sea, sobre el cuál se han extendido cientos de rumores falsos. Y, por supuesto, es completamente injusto que por unos cuentos sean los pacientes los que sufran, cuando ya tienen suficiente con sufrir un desorden crónico como este.

E: Ya les hemos hablado anteriormente de lo que trata de conseguir nuestro estudio, y de cómo planea ayudar a los especialistas a calcular un tratamiento altamente eficaz contra la fibromialgia que pueda adaptarse a cada paciente concreto. ¿Cuáles son sus opiniones al respecto?

P: Conozco historias de otros enfermos de fibromialgia que han mejorado haciendo deporte (como el yoga o estiramientos), dejando de tomar gluten o perdiendo peso, o incluso imponiéndose unas horas para dormir y despertarse bien; pero a mí esto nunca me ha terminado de funcionar; así que cualquier avance que pueda ayudar a otro enfermos como yo, o que lleve a que otros estudios puedan aparecer y avanzar, siempre será de agradecer.

M: Sí, sin duda, cualquier avance en un tema como este será de agradecer, no solo por los pacientes, sino también por los equipos médicos, que tendrán más capacidad de mejorar el nivel de vida de los enfermos de esta enfermedad crónica.

Anexo 2: Bibliografía recomendada sobre ingeniería de control.

Bolzern, Paolo et al. *Fundamentos de control automático*. 3a ed. Madrid: McGraw Hill, 2009. Print.

Weyrick, Robert C. *Introducción al control automático*. Barcelona: Gustavo Gili, 1977. Print.

Pinto Bermúdez, Enrique, and Matía Espada, Fernando. *Fundamentos de control con MATLAB*. Madrid: Pearson Education, 2010. Print.

Crowe, J. et al. *PID Control New Identification and Design Methods*. London: Springer London, 2005. Web.

Dukkipati, Rao V. *Analysis and Design of Control Systems Using MATLAB*. New Delhi: New Age International P Ltd., Publishers, 2006. Print.

Aström, Karl Johan et al. *Control PID avanzado*. Madrid: Pearson-Prentice Hall, 2009. Print

Fernández Camacho, Eduardo, and Bordóns Alba, Carlos. *Model Predictive Control in the Process Industry*. London: Springer-Verlag, 1995. Print.

Camacho, E. F., and Bordons, C. *Model Predictive Control*. 2nd ed. London: Springer London, 2007. Web.

Borrelli, Francesco, Bemporad, Alberto, and Morari, Manfred. *Predictive Control for Linear and Hybrid Systems*. New York: Cambridge University Press, 2017. Print.

Wang, Liuping. *Model Predictive Control System Design and Implementation Using MATLAB®*. 1st ed. London: Springer London, 2009. Web.

Anexo 3: Códigos de dosificación y MPC.

gráficas_control.m

Código basado en líneas de comando para mostrar por pantalla los datos obtenidos de las simulaciones de los montajes para control PID e identificación del modelo de respuesta del paciente.

```
%Funcion para la muestra de graficas de analisis de controles PID

tiempo=0:1:75;

Sintoma_origen=[50;77;50;26;50;63;83;98;78;79;75;70;88;66;85;94;79;75;76;77;76;95
;77;82;95;91;89;71;32;24;70;67;19;14;17;2;18;37;23;14;26;11;22;10;7;7;8;6;8;14;16
;10;13;15;3;14;7;11;3;5;13;10;9;8;10;8;9;8;7;9;6;9;5;6;34;6]';

%FAR3SIN_10 VS Experimento

%subplot(2,1,1); plot(tfar, Sintomas_FM, tiempo, Sintoma_origen, 'LineWidth',2);
grid; xlabel('t(dias)'); ylabel('Sintomas (%)'); title('Sintomas FM');
legend('Modelo far3sin10', 'Experimento', 'Location', 'Best');

%subplot(2,1,2);plot(tfar,Naltrexona, 'LineWidth',2);grid;xlabel('t(dias)');ylabel
('Naltrexona (mg)');title('Dosis de naltrexona');legend('Naltrexona','Location',
'Best');

%EXPERIMENTO

%subplot(2,1,1); plot(tiempo, Sintoma_origen, 'LineWidth', 2); grid;
xlabel('t(dias)'); ylabel('Sintomas(%)'); title('Sintomas FM');
legend('Experimento','Location', 'Best');

%subplot(2,1,2);plot(t,Naltrexona, 'LineWidth',2);grid;xlabel('t(dias)');ylabel('N
altrexona (mg)');title('Dosis de naltrexona');legend('Naltrexona','Location',
```

```

'Best');

%GS1 VS FAR3SIN_10 VS EXPERIMENTO

%subplot(2,1,1); plot(t1,Sintomas_FM1, tfar,Sintomas_FM, tiempo,Sintoma_origen,
'LineWidth',2); grid; xlabel('t(días)'); ylabel('Sintomas (%)'); title('Sintomas
FM'); legend('Modelo Gs1','Modelo far3sin10','Experimento','Location','Best');

%subplot(2,1,2);plot(t1,Naltrexona,'LineWidth',2);grid;xlabel('t(días)');ylabel('
Naltrexona (mg)');title('Dosis de naltrexona');legend('Naltrexona','Location',
'Best');

%GS1 VS GS2 VS EXPERIMENTO

%subplot(2,1,1); plot(t1,Sintomas_FM1, t2,Sintomas_FM2, tiempo,Sintoma_origen,
'LineWidth',2); grid; xlabel('t(días)'); ylabel('Sintomas (%)'); title('Sintomas
FM'); legend('Modelo Gs1','Modelo Gs2','Experimento','Location','Best');

%subplot(2,1,2);plot(t1,Naltrexona,'LineWidth',2);grid;xlabel('t(días)');ylabel('
Naltrexona (mg)');title('Dosis de naltrexona');legend('Naltrexona','Location',
'Best');

%GS1 VS GS2 GS3 VS EXPERIMENTO

%subplot(2,1,1); plot(t1, Sintomas_FM1, t2, Sintomas_FM2, t3, Sintomas_FM3, tiempo,
Sintoma_origen, 'LineWidth',2); grid; xlabel('t(días)'); ylabel('Sintomas (%)');
title('Sintomas FM'); legend('Modelo Gs1','Modelo Gs2','Modelo
Gs3','Experimento','Location','Best');

%subplot(2,1,2); plot(t1,Naltrexona,'LineWidth',2); grid; xlabel('t(días)');
ylabel('Naltrexona (mg)'); title('Dosis de naltrexona');
legend('Naltrexona','Location','Best');

% GS3 VS GZ

%subplot(2,1,1); plot(t3,Sintomas_FM3,t,Sintomas_FMD,'*', 'LineWidth',2); grid;
xlabel('t(días)'); ylabel('Sintomas (%)'); title('Sintomas FM'); legend('Modelo
continuo','Modelo discreto','Location','Best');

%subplot(2,1,2);plot(t3,Naltrexona,t,Naltrexona1,'*', 'LineWidth',2);grid;xlabel('
t(días)');ylabel('Naltrexona (mg)');title('Dosis de naltrexona');legend('Dosis
Continua','Dosis discreta','Location','Best');

% CONTROL GS3 VS EXPERIMENTO

%subplot(2,1,1);plot(t3,Sintomas_FM_Ref,t3,Sintomas_FM3,tiempo,Sintoma_origen,'Li
neWidth',2;grid; xlabel('t(días)'); ylabel('Sintomas (%)'); title('Sintomas
FM');legend('Referencia','Control continuo','Experimento','Location','Best');

%subplot(2,1,2);plot(t3,Naltrexona,'LineWidth',2);grid;xlabel('t(días)');ylabel('
Naltrexona(mg)'); title('Dosis de naltrexona'); legend('Dosis','Location','Best');

% CONTROL GZ - VS EXPERIMENTO

subplot(2,1,1);
plot(t3,Sintomas_FM_Ref,t3,Sintomas_FM3,tiempo,Sintoma_origen,'LineWidth',2);
grid; xlabel('t(días)'); ylabel('Sintomas (%)'); title('Sintomas FM');
legend('Referencia','Control discreto','Experimento','Location','Best');

subplot(2,1,2); plot(t3,Naltrexona,'LineWidth',2); grid; xlabel('t(días)');
ylabel('Naltrexona(mg)'); title('Dosis de naltrexona');
legend('Dosis','Location','Best');

```

Extraccion_dosis.m

Función para la extracción de una dosis no dañina para el usuario para montajes de Simulink[®], dependiente del valor pedido por el controlador para el tratamiento y las definiciones de cambios de dosis suaves así como de las dosis comerciales existentes.

```

%Funcion para la extraccion de una dosis no dañina para el usuario,
%dependiente de la petición del sistema y un cambio de dosis suave.

function dosis = Extraccion_dosis(dosis_in)

%Siendo "dosis_in(1)" la salida del controlador, que contiene la dosis que el
%controlador estima oportuna para el sistema, siendo "dosis_n", "dosis_continua"
y
%"dosis_nueva" variables de apoyo, y siendo "dosis_in(2)" la realimentación
%de la salida del sistema, siendo por tanto la salida del instante
%anterior en cada momento de la simulación.
dosis_a=dosis_in(2);

%% Saturamos la entrada,
%Haciendo uso de la entrada y de una de las variables de apoyo
if(dosis_in(1) < 0 )
    dosis_continua=0;
elseif (dosis_in(1) > 13.5)
    dosis_continua=13.5;
else
    dosis_continua=dosis_in(1);
end

%% Discretizamos la dosis de entrada
%Entre las dosis comerciales posibles mediante un redondeo simple,
%teniéndose en cuenta por tanto las dosis posibles, siendo cada una de
%ellas un "escalón" de dosis: [0 1.58 3.85 5.76 7.68 9.6 11.58 13.5]
if(dosis_continua > 12.54 && dosis_continua <= 13.5)
    dosis_n=13.5;
elseif(dosis_continua > 11.58 && dosis_continua <= 12.54)
    dosis_n=11.58;
elseif(dosis_continua > 10.59 && dosis_continua <= 11.58)
    dosis_n=11.58;
elseif(dosis_continua > 9.6 && dosis_continua <= 10.59)
    dosis_n=9.6;
elseif(dosis_continua > 8.64 && dosis_continua <= 9.6)
    dosis_n=9.6;
elseif(dosis_continua > 7.68 && dosis_continua <= 8.64)
    dosis_n=7.68;
elseif(dosis_continua > 6.72 && dosis_continua <= 7.68)
    dosis_n=7.68;
elseif(dosis_continua > 5.76 && dosis_continua <= 6.72)
    dosis_n=5.76;
elseif(dosis_continua > 4.805 && dosis_continua <= 5.76)
    dosis_n=5.76;
elseif(dosis_continua > 3.85 && dosis_continua <= 4.805)
    dosis_n=3.85;
elseif(dosis_continua > 2.885 && dosis_continua <= 3.85)
    dosis_n=3.85;
elseif(dosis_continua > 1.92 && dosis_continua <= 2.885)
    dosis_n=1.92;
elseif(dosis_continua > 0.96 && dosis_continua <= 1.92)
    dosis_n=1.92;
elseif(dosis_continua >= 0 && dosis_continua <= 0.96)
    dosis_n=0;
end

```

```
%% Cambio suave de dosis
%Programamos un cambio de dosis controlado, que sea lo suficientemente
%suave como para reducir los posibles efectos secundarios por el cambio de
%dosis, teniendo en cuenta que no debe superarse un cambio de más de 2
%escalones de dosis.
switch dosis_n
  case 13.5
    if(dosis_a>=9.6)
      dosis_nueva=13.5;
    elseif(dosis_a==7.68)
      dosis_nueva=11.58;
    elseif(dosis_a==5.76)
      dosis_nueva=9.6;
    elseif(dosis_a==3.85)
      dosis_nueva=7.68;
    elseif(dosis_a==1.92)
      dosis_nueva=5.76;
    else
      dosis_nueva=3.85;
    end

  case 11.58
    if(dosis_a>=7.68)
      dosis_nueva=11.58;
    elseif(dosis_a==5.76)
      dosis_nueva=9.6;
    elseif(dosis_a==3.85)
      dosis_nueva=7.68;
    elseif(dosis_a==1.92)
      dosis_nueva=5.76;
    else
      dosis_nueva=3.85;
    end

  case 9.6
    if(dosis_a>=5.76)
      dosis_nueva=9.6;
    elseif(dosis_a==3.85)
      dosis_nueva=7.68;
    elseif(dosis_a==1.92)
      dosis_nueva=5.76;
    else
      dosis_nueva=3.85;
    end

  case 7.68
    if(dosis_a==13.5)
      dosis_nueva=9.6;
    elseif(dosis_a>=3.85 && dosis_a<=11.58)
      dosis_nueva=7.98;
    elseif(dosis_a==1.92)
      dosis_nueva=5.76;
    else
      dosis_nueva=3.85;
    end

  case 5.76
    if(dosis_a==13.5)
      dosis_nueva=9.6;
    elseif(dosis_a==11.58)
      dosis_nueva=7.68;
    elseif(dosis_a>=1.92 && dosis_a<=9.6)
      dosis_nueva=5.76;
    else
      dosis_nueva=3.85;
    end

  case 3.85
    if(dosis_a==13.5)
```

```

        dosis_nueva=9.6;
    elseif(dosis_a==11.58)
        dosis_nueva=7.68;
    elseif(dosis_a==9.6)
        dosis_nueva=5.76;
    elseif(dosis_a<=7.68)
        dosis_nueva=3.85;
    end

    case 1.92
        if(dosis_a==13.5)
            dosis_nueva=9.6;
        elseif(dosis_a==11.58)
            dosis_nueva=7.68;
        elseif(dosis_a==9.6)
            dosis_nueva=5.76;
        elseif(dosis_a==7.68)
            dosis_nueva=3.85;
        elseif(dosis_a<=5.76)
            dosis_nueva=1.92;
        end

    case 0
        if(dosis_a==13.5)
            dosis_nueva=9.6;
        elseif(dosis_a==11.58)
            dosis_nueva=7.68;
        elseif(dosis_a==9.6)
            dosis_nueva=5.76;
        elseif(dosis_a==7.68)
            dosis_nueva=3.85;
        elseif(dosis_a==5.76)
            dosis_nueva=1.92;
        elseif(dosis_a<=3.85)
            dosis_nueva=0;
        end

end

%% Definimos la salida
%%Una vez dosificada la dosis que debe darse al usuario, se define como
%%salida de este bloque de dosis definida y, por tanto, como entrada del
%%sistema
dosis= dosis_nueva;

```

graficas_predictivo.m

Código basado en líneas de comando para mostrar por pantalla los datos obtenidos de las simulaciones de los sistemas controlados por MPC sin dosificación y con dosificación (ambos sin interacción médica).

```

%% Representamos los resultados.
%%Esta sección solo muestra los resultados a solas
nm=nit; %numero de iteraciones de la simulación MPC
t = 0:T:(nm-1)*T;
%figure
subplot(2,1,1),plot(t, (r+pi), '--k', t, (ym+pi), 'r', 'Linewidth',2);
xlabel('t(dias)');ylabel('Sintomas (%)');title('Sintomas FM');
legend('ref','Predictivo','Location','SouthEast'); grid on; hold on;
subplot(2,1,2),plot(t,u, 'r', 'Linewidth',2);
xlabel('t(dias)');ylabel('Naltrexona (mg)'); title('Dosis
Naltrexona');
legend('Predictivo'); grid on; hold on;

```

```

%% Con el test
tiempo=0:1:64;
Sintoma_origen=[50 77 50 26 50 63 83 98 78 79 75 70 88 66 85 94 79
75 76 77 76 95 77 82 95 91 89 71 32 24 70 67 19 14 17 2 18 37 23 14
26 11 22 10 7 7 8 6 8 14 16 10 13 15 3 14 7 11 3 5 13 10 9 8 10];
Farmaco_origen=[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 4.5 4.5 4.5 4.5 4.5 4.5 4.5 4.5 4.5 4.5 4.5 4.5 4.5 4.5 4.5
4.5 4.5 4.5 4.5 4.5 4.5 4.5 4.5 4.5 4.5 4.5 4.5 4.5 4.5 4.5
4.5 4.5 4.5 4.5 4.5];

subplot(2,1,1);plot(tiempo, Sintoma_origen, 'Linewidth',2);
legend('refPred', 'Predictivo', 'Test', 'Location', 'Best');
hold on;
subplot(2,1,2),plot(tiempo,Farmaco_origen, 'Linewidth',2);
legend('Predictivo', 'Test');
hold on;

%% Con PID
subplot(2,1,1);plot(t3,Sintomas_FM_Ref, '--
',t3,Sintomas_FM3, 'Linewidth',2);
legend('refPred', 'Predictivo', 'Test', 'refPID', 'PID');
subplot(2,1,2);plot(t3,Naltrexona, 'Linewidth',2);
legend('Predictivo', 'Test', 'PID');

```

Predictivo_SinDosi.m

Código de simulación del sistema, controlado por MPC, sin restricciones de dosis y sin interacción médica.

```

% SISTEMA BASADO EN UN CONTROL PREDICTIVO DISCRETO SIN DOSIFICACIÓN
% (solo saturado)
% ANDRES PASTOR ALFARO
% Extracción del programa apoyada en el libro:
% Control of Dead-time Processes, Springer-Verlag, 2007
% Y apoyada en el trabajo de Julio Normey-Rico, Eduardo F. Camacho y
Sergio Andres Castaño G.
%
%----- %
% Solution of the Diophantine equation %
%----- %
%[E,F] = diophantine(A,N,d)
% A = Denominator
% N = Prediction Horizon
% d = Transport delay

%%
%----- %
% CONTROL GPC %
%----- %
clearvars -except t3 Sintomas_FM3 Sintomas_FM_Ref Naltrexona
close all

%% Parametros del sistema
T=1; %Tiempo de muestreo en días

```

```

% Modelo del sistema
%      -1.055 z^-1 - 0.6633 z^-2
% Gz=  -----
%      1 - 1.144 z^-1 + 0.2457 z^-2

B=[0 -1.055 -0.6633]; %Numerador del modelo
A=[1 -1.144 0.2457]; %Denominador del modelo
d=0; %Retardo de la planta

%% Parametros de sintonia del GPC
%Ventana de prediccion
N=11; %Valor de la ventana
N1=d+1; %Horizonte Inicial
N2=d+N; %Horizonte Final de salida
Nu=N; %Horizonte de entrada (o de control)
lambda=1; %Parametro de ponderacion, no se pondera nada si son 1
delta=1;

%Hallamos ahora las matrices de la función de costo, son diagonales
Q1=eye(Nu)*lambda;
Qd=eye(N)*delta;

%% Calculo de la ecuacion Diofantina
[En,F] = diophantine(A,N2,0); %Calculo de la funcion Diofantina
E=En(end,:); %El polinomio E seria la ultima fila de En arrojada por la
funcion diophantine
F=F(N1:N1+N-1,1:end); %El polinomio F es la respuesta libre, y cogemos
de él
%la parte correspondiente a nuestro horizonte en el primer sector, y
toda
%la información en el resto de F

%% Determina los coeficientes de control pasados

%Elimino el cero de la primera posicion de B en caso de tenerlo, como
%tenemos nosotros, antes de comenzar los cálculos.
if B(1)==0
    B=B(2:end);
end

uG=zeros(N1,N)'; %Vector de controles pasados (Pertenece a la respuesta
libre)

j=2;
for i=N1:N2
    aux=conv(En(i,1:end),B);
    if length(aux) < N1+j-1 %Si la longitud del auxiliar es menor que j
        aux=[aux zeros(1,(N1+j-1)-length(aux))]; %Completar con ceros
    end
    uG(j-1,1:N1)=aux(j:N1+j-1);
    j=j+1;
end

g=conv(E,B); %Calcula polinomio g, producto de E y B, es por tanto los
%coeficientes de la planta frente al escalón.

```



```

G=zeros(N,Nu); % Inicializa la matriz G, matriz de respuesta de los
%coeficientes del escalón

for k=1:Nu
    G(k:end,k)=g(1:N-k+1); % Forma la matriz G a partir del polinomio g
end

Mn=inv(G'*Qd*G+Ql)*G'*Qd; %Calculo de la Funcion de Costo sin
Restriccion

%Calculo del controlador K1 (Primera fila de Mn)
K1=Mn(1,:);

%% Bucle de control
%Inicializamos los parámetros de simulación
drs=100; %, duracion de la simulación= NIT-1
nit=drs+1; %Número de interacciones
inc u=0; %Incremento de control predictivo
Y=10; %COMIENZO DEL CONTROL
u_ant(1:Y) = 0; %Inicializamos valores de salida de controles anteriores
u(1:Y+d) = 0; ym(1:Y+d) = 0; r(1:Y+d) = 0;
%Inicializamos la señal de control, la salida y la referencia, que
durante
%un tiempo serán 0 para mejor visualización en la simulación. Cabe
destacar
%que, realmente ym=84, pero al ser esto un offset constante, lo
añadiremos
%más adelante, tratando los cálculos como si fuera de valor 0 durante el
%resto. Lo mismo ocurre con la referencia; inicialmente debería valer
igual
%que la salida ym. Además, como veremos más adelante, no empezaremos a
%controlar hasta el día N.

%Offsets de referencia y salida
pi=84;

%Referencia del usuario (referida respecto del offset)
ref=8;

%Transformación de la referencia con offset a sin offset
%REFERENCIA COMPLEJA DE TEST
r(26:30)=- (pi-ref)+40; r(31:32)=- (pi-ref)+20;
r(33:37)=- (pi-ref)+15;r(38:42)=- (pi-ref)+10;r(43:47)=- (pi-ref)+5;
r(48:nit) = - (pi-ref);%El índice 26 indica un t=25, cuando comienza el
test
% %REFERENCIA SIMPLE
% r((Y+1):nit)=- (pi-ref);
% %REFERENCIA SIMPLE DE TEST
% r((25+1):nit)=- (pi-ref);

%Perturbación
do(1:82)= 0;do(83:84)=10; do(85:nit) = 0;
%Como antes, el índice 86 indica t=85

lduf=size(F); %Averiguo la longitud del polinomio F
lduf=lduf(1,2); %Será el incremento de control de la respuesta libre

luG=size(uG); %Averiguo la longitud de uG

```

```

luG=luG(1,2);
duf=zeros(1,luG); %Vector de incrementos de control pasados de la
respuesta libre

%% Comenzamos a controlar
for k=(Y+1)+d:nit
    % Salida del proceso
    ym(k)=B(1)*u(k-1-d)+B(2)*u(k-2-d)-A(2)*(ym(k-1))-A(3)*(ym(k-
2))+do(k);

    %Cálculo de la respuesta libre teniendo en cuenta el incremento
    %de control de la respuesta libre
    free=0;
    for i=1:lduf
        free=free+ym(k-i+1)*F(:,i);
    end

    if luG==1 %Si uG es un vector columna
        free=free+duf*uG; %Invierto el producto, para que sea realizable
    else %Si uG es una matriz
        free=free+uG*duf';
    end

    %Cálculo de la variable de control por incrementos
    inc_u=K1*(r(k)-free);

    %En nuestro caso, k!=1 siempre, pero habría que tener en cuenta este
    %detalle, que la acción de control se basa en un sumatorio.
    if k==1
        u(k)=inc_u;
    else
        u(k)=u(k-1)+ inc_u;
    end

    %Actualizamos el vector de valores de control pasados
    aux_u=u_ant(1:length(B)-1);
    u_ant=[u(k) aux_u];

    %Y actualizaremos los incrementos de control pasados, que se usan en
    la
    %respuesta libre.
    aux_2=duf(1:end-1);
    duf=[inc_u aux_2];

end

%% Para representar, solo debemos usar la funcion "graficas"

%% FUNCION DIOFANTINA
%-----
function[E,F] = diophantine(A,N,d)

delta = [1 -1];

AD = conv(A,delta);

nAD = size(AD);
nAD = nAD(2);

```

```

N1 = d +1;
N2 = d + N;

f(1,:) = [1 zeros(1,nAD-2)];

for j = 1: N2
    for i = 1:nAD-2
        f(j+1,i) = f(j,i+1)-f(j,1)*AD(i+1);
    end
    f(j+1,nAD-1) = -f(j,1)*AD(nAD);
end

F = f(1+N1:1+N2,:);

% Computes E(z^-1)

E = zeros(N2);
e(1) = 1; % for the special case 1/~A

E(1,1) = e(1);

for i = 2: N2
    e(i) = f(i,1);
    E(i,1:i)=e;
end

E = E(N1:N2,:);
end

```

Predictivo_ConDosi.m

Código de simulación del sistema, controlado por MPC, con restricciones de dosis y sin interacción médica.

```

% SISTEMA BASADO EN UN CONTROL PREDICTIVO DISCRETO CON DOSIFICACIÓN
% ANDRES PASTOR ALFARO
% Extracción del programa apoyada en el libro:
% Control of Dead-time Processes, Springer-Verlag, 2007
% Y apoyada en el trabajo de Julio Normey-Rico, Eduardo F. Camacho y
Sergio Andres Castaño G.
%
%----- %
% Solution of the Diophantine equation %
%----- %
%[E,F] = diophantine(A,N,d)
% A = Denominator
% N = Prediction Horizon
% d = Transport delay

%%
%----- %
% CONTROL GPC %
%----- %
clearvars -except t3 Sintomas_FM3 Sintomas_FM_Ref Naltrexona
close all
%% Parametros del sistema
T=1; %Tiempo de muestreo en días

```

```

% Modelo del sistema
%      -1.055 z^-1 - 0.6633 z^-2
% Gz=  -----
%      1 - 1.144 z^-1 + 0.2457 z^-2

B=[0 -1.055 -0.6633]; %Numerador del modelo
A=[1 -1.144 0.2457]; %Denominador del modelo
d=0; %Retardo de la planta

%% Parametros de sintonia del GPC
%Ventana de prediccion
N=11; %Valor de la ventana
N1=d+1; %Horizonte Inicial
N2=d+N; %Horizonte Final de salida
Nu=N; %Horizonte de entrada (o de control)
lambda=1; %Parametro de ponderacion, no se pondera nada si son 1
delta=1;

%Hallamos ahora las matrices de la función de costo, son diagonales
Ql=eye(Nu)*lambda;
Qd=eye(N)*delta;

%% Calculo de la ecuacion Diofantina
[En,F] = diophantine(A,N2,0); %Calculo de la funcion Diofantina
E=En(end,:); %El polinomio E seria la ultima fila de En arrojada por
la funcion diophantine
F=F(N1:N1+N-1,1:end); %El polinomio F es la respuesta libre, y cogemos
de él
%la parte correspondiente a nuestro horizonte en el primer sector, y
toda
%la información en el resto de F

%% Determina los coeficientes de control pasados

%Elimino el cero de la primera posicion de B en caso de tenerlo, como
%tenemos nosotros, antes de comenzar los cálculos.
if B(1)==0
    B=B(2:end);
end

uG=zeros(N1,N)'; %Vector de controles pasados (Pertenece a la
respuesta libre)

j=2;
for i=N1:N2
    aux=conv(En(i,1:end),B);
    if length(aux) < N1+j-1 %Si la longitud del auxiliar es menor que
j
        aux=[aux zeros(1,(N1+j-1)-length(aux))]; %Completar con ceros
    end
    uG(j-1,1:N1)=aux(j:N1+j-1);
    j=j+1;
end

```

```

g=conv(E,B); %Calcula polinomio g, producto de E y B, es por tanto los
%coeficientes de la planta frente al escalón.

G=zeros(N,Nu); % Inicializa la matriz G, matriz de respuesta de los
%coeficientes del escalón

for k=1:Nu
    G(k:end,k)=g(1:N-k+1); % Forma la matriz G a partir del polinomio
g
end

Mn=inv(G'*Qd*G+Ql)*G'*Qd; %Calculo de la Funcion de Costo sin
Restriccion

%Calculo del controlador K1 (Primera fila de Mn)
K1=Mn(1,:);

%% Bucle de control
%Inicializamos los parámetros de simulación
drs=100; %, duracion de la simulación= NIT-1
nit=drs+1; %Número de interacciones
inc_u=0; %Incremento de control predictivo
Y=10; %COMIENZO DEL CONTROL
u_ant(1:Y) = 0; %Inicializamos valores de salida de controles
anteriores
u(1:Y+d) = 0; ym(1:Y+d) = 0; r(1:Y+d) = 0;
%Inicializamos la señal de control, la salida y la referencia, que
durante
%un tiempo serán 0 para mejor visualización en la simulación. Cabe
destacar
%que, realmente ym=84, pero al ser esto un offset constante, lo
añadiremos
%más adelante, tratando los cálculos como si fuera de valor 0 durante
el
%resto. Lo mismo ocurre con la referencia; inicialmente debería valer
igual
%que la salida ym. Además, como veremos más adelante, no empezaremos a
%controlar hasta el día N.

%Offsets de referencia y salida
pi=84;

%Referencia del usuario (referida respecto del offset)
ref=8;

%Transformación de la referencia con offset a sin offset
%REFERENCIA COMPLEJA DE TEST
r(26:30)=- (pi-ref)+40; r(31:32)=- (pi-ref)+20;
r(33:37)=- (pi-ref)+15;r(38:42)=- (pi-ref)+10;r(43:47)=- (pi-ref)+5;
r(48:nit) = - (pi-ref);%El indice 26 indica un t=25, cuando comienza el
test
%REFERENCIA SIMPLE
% r((Y+1):nit)=- (pi-ref);
%REFERENCIA SIMPLE DE TEST
% r((25+1):nit)=- (pi-ref);

%Perturbación
do(1:82) = 0;do(83:84)=10; do(85:nit) = 0;
%Como antes, el índice 86 indica t=85

```

```

lduf=size(F); %Averiguo la longitud del polinomio F
lduf=lduf(1,2); %Será el incremento de control de la respuesta libre

luG=size(uG); %Averiguo la longitud de uG
luG=luG(1,2);
duf=zeros(1,luG); %Vector de incrementos de control pasados de la
respuesta libre

%% Comenzamos a controlar
for k=(Y+1)+d:nit % Salida del proceso
    ym(k)=B(1)*u(k-1-d)+B(2)*u(k-2-d)-A(2)*(ym(k-1))-A(3)*(ym(k-
2))+do(k);

    %Cálculo de la respuesta libre teniendo en cuenta el incremento
    %de control de la respuesta libre
    free=0;
    for i=1:lduf
        free=free+ym(k-i+1)*F(:,i);
    end

    if luG==1 %Si uG es un vector columna
        free=free+duf*uG; %Invierto el producto, para que sea
realizable
    else %Si uG es una matriz
        free=free+uG*duf';
    end

    %Cálculo de la variable de control por incrementos
    inc_u=K1*(r(k)-free);

    %En nuestro caso, k!=1 siempre, pero habría que tener en cuenta
este
    %detalle, que la acción de control se basa en un sumatorio.
    if k==1
        u(k)=inc_u;
    else
        u(k)=u(k-1)+ inc_u;
    end

    dosis_a=u(k-1);

    %Realizamos una pequeña situación de la variable de control
    if u(k)>13.5
        dosis_continua=13.5;
    elseif u(k)<0
        dosis_continua=0;
    else
        dosis_continua=u(k);
    end

    %Discretizamos las dosis
    if(dosis_continua > 12.54 && dosis_continua <= 13.5)
        dosis_n=13.5;
    elseif(dosis_continua > 11.58 && dosis_continua <= 12.54)
        dosis_n=11.58;
    elseif(dosis_continua > 10.59 && dosis_continua <= 11.58)
        dosis_n=11.58;
    elseif(dosis_continua > 9.6 && dosis_continua <= 10.59)
        dosis_n=9.6;

```

```

elseif(dosis_continua > 8.64 && dosis_continua <= 9.6)
    dosis_n=9.6;
elseif(dosis_continua > 7.68 && dosis_continua <= 8.64)
    dosis_n=7.68;
elseif(dosis_continua > 6.72 && dosis_continua <= 7.68)
    dosis_n=7.68;
elseif(dosis_continua > 5.76 && dosis_continua <= 6.72)
    dosis_n=5.76;
elseif(dosis_continua > 4.805 && dosis_continua <= 5.76)
    dosis_n=5.76;
elseif(dosis_continua > 3.85 && dosis_continua <= 4.805)
    dosis_n=3.85;
elseif(dosis_continua > 2.885 && dosis_continua <= 3.85)
    dosis_n=3.85;
elseif(dosis_continua > 1.92 && dosis_continua <= 2.885)
    dosis_n=1.92;
elseif(dosis_continua > 0.96 && dosis_continua <= 1.92)
    dosis_n=1.92;
elseif(dosis_continua >= 0 && dosis_continua <= 0.96)
    dosis_n=0;
end

%Programamos un cambio de dosis controlado
switch dosis_n
    case 13.5
        if(dosis_a>=9.6)
            dosis_nueva=13.5;
        elseif(dosis_a==7.68)
            dosis_nueva=11.58;
        elseif(dosis_a==5.76)
            dosis_nueva=9.6;
        elseif(dosis_a==3.85)
            dosis_nueva=7.68;
        elseif(dosis_a==1.92)
            dosis_nueva=5.76;
        else
            dosis_nueva=3.85;
        end

    case 11.58
        if(dosis_a>=7.68)
            dosis_nueva=11.58;
        elseif(dosis_a==5.76)
            dosis_nueva=9.6;
        elseif(dosis_a==3.85)
            dosis_nueva=7.68;
        elseif(dosis_a==1.92)
            dosis_nueva=5.76;
        else
            dosis_nueva=3.85;
        end

    case 9.6
        if(dosis_a>=5.76)
            dosis_nueva=9.6;
        elseif(dosis_a==3.85)
            dosis_nueva=7.68;
        elseif(dosis_a==1.92)
            dosis_nueva=5.76;
        else

```

```
        dosis_nueva=3.85;
    end

    case 7.68
        if(dosis_a==13.5)
            dosis_nueva=9.6;
        elseif(dosis_a>=3.85 && dosis_a<=11.58)
            dosis_nueva=7.68;
        elseif(dosis_a==1.92)
            dosis_nueva=5.76;
        else
            dosis_nueva=3.85;
        end

    case 5.76
        if(dosis_a==13.5)
            dosis_nueva=9.6;
        elseif(dosis_a==11.58)
            dosis_nueva=7.68;
        elseif(dosis_a>=1.92 && dosis_a<=9.6)
            dosis_nueva=5.76;
        else
            dosis_nueva=3.85;
        end

    case 3.85
        if(dosis_a==13.5)
            dosis_nueva=9.6;
        elseif(dosis_a==11.58)
            dosis_nueva=7.68;
        elseif(dosis_a==9.6)
            dosis_nueva=5.76;
        elseif(dosis_a<=7.68)
            dosis_nueva=3.85;
        end

    case 1.92
        if(dosis_a==13.5)
            dosis_nueva=9.6;
        elseif(dosis_a==11.58)
            dosis_nueva=7.68;
        elseif(dosis_a==9.6)
            dosis_nueva=5.76;
        elseif(dosis_a==7.68)
            dosis_nueva=3.85;
        elseif(dosis_a<=5.76)
            dosis_nueva=1.92;
        end

    case 0
        if(dosis_a==13.5)
            dosis_nueva=9.6;
        elseif(dosis_a==11.58)
            dosis_nueva=7.68;
        elseif(dosis_a==9.6)
            dosis_nueva=5.76;
        elseif(dosis_a==7.68)
            dosis_nueva=3.85;
        elseif(dosis_a==5.76)
            dosis_nueva=1.92;
        elseif(dosis_a<=3.85)
```



```

        dosis_nueva=0;
    end
end

u(k)= dosis_nueva;

%Actualizamos el vector de valores de control pasados
aux_u=u_ant(1:length(B)-1);
u_ant=[u(k) aux_u];

%Y actualizaremos los incrementos de control pasados, que se usan
en la
%respuesta libre.
aux_2=duf(1:end-1);
duf=[inc_u aux_2];

end

%% Para representar, solo debemos usar la funcion "graficas"

%% FUNCION DIOFANTINA
%-----
function[E,F] = diophantine(A,N,d)

delta = [1 -1];

AD = conv(A,delta);

nAD = size(AD);
nAD = nAD(2);

N1 = d +1;
N2 = d + N;

f(1,:) = [1 zeros(1,nAD-2)];

for j = 1: N2
    for i = 1:nAD-2
        f(j+1,i) = f(j,i+1)-f(j,1)*AD(i+1);
    end
    f(j+1,nAD-1) = -f(j,1)*AD(nAD);
end

F = f(1+N1:1+N2, :);

% Computes E(z^-1)

E = zeros(N2);
e(1) = 1; % for the special case 1/~A

E(1,1) = e(1);

for i = 2: N2
    e(i) = f(i,1);
    E(i,1:i)=e;
end

E = E(N1:N2, :);

```

```
end
```

Predictivo_Revisiones.m

Código de simulación del sistema, controlado por MPC, con restricciones de dosis y con interacción médica (y, por tanto, personalización a cada paciente).

```
% SISTEMA BASADO EN UN CONTROL PREDICTIVO DISCRETO CON DOSIFICACIÓN Y
% REVISIONES MÉDICAS
% ANDRÉS PASTOR ALFARO
% Extracción del programa apoyada en el libro:
% Control of Dead-time Processes, Springer-Verlag, 2007
% Y apoyada en el trabajo de Julio Normey-Rico, Eduardo F. Camacho y
% Sergio Andres Castaño G.
%
%----- %
% Solution of the Diophantine equation %
%----- %
%[E,F] = diophantine(A,N,d)
% A = Denominator
% N = Prediction Horizon
% d = Transport delay
%%
%----- %
% CONTROL GPC %
%----- %
clearvars -except t3 Sintomas_FM3 Sintomas_FM_Ref Naltrexona sal
close all

%% Simulación de respuesta del paciente:
%Días totales del estudio:
estudio=200;
%Días entre revisiones:
der=15;

%Respuesta experimental del paciente:
for L=1:1:round(estudio/der)
    azar=2*rand-1;
    if azar>0.35
        rpa(L)=100; %El paciente se encuentra peor de lo estimado
    elseif azar<-0.35
        rpa(L)=0; %El paciente se encuentra mejor que lo estimado
    else
        rpa(L)=50; %El paciente se encuentra igual que lo estimado (que
lo estimado, no que la reerencia)
    end
    % rpa(L)=0; %Descomentar si quiero ejemplo de dejar de medicar
    % rpa(L)=100;%Descomentar si quiero ejemplo de hipersensibilidad al
dolor
    % rpa(L)=50; %Descomentar si quiero ejemplo de modelado perfecto
end

%% Parametros del sistema
T=1; %Tiempo de muestreo en días

% Modelo del sistema
% -1.055 z^-1 - 0.6633 z^-2
```

```

% Gz= -----
%      1 - 1.144 z^-1 + 0.2457 z^-2

B=[0 -1.055 -0.6633]; %Numerador del modelo
A=[1 -1.144 0.2457]; %Denominador del modelo
d=0; %Retardo de la planta

%% Parametros de sintonia del GPC
%Ventana de prediccion
N=3; %Valor de la ventana
N1=d+1; %Horizonte Inicial
N2=d+N; %Horizonte Final de salida
Nu=N; %Horizonte de entrada (o de control)
lambda=1; %Parametro de ponderacion, no se pondera nada si son 1
delta=1;

%Hallamos ahora las matrices de la función de costo, son diagonales
Ql=eye(Nu)*lambda;
Qd=eye(N)*delta;

%% Calculo de la ecuacion Diofantina
[En,F] = diophantine(A,N2,0); %Calculo de la funcion Diofantina
E=En(end,:); %El polinomio E seria la ultima fila de En arrojada por la
funcion diophantine
F=F(N1:N1+N-1,1:end); %El polinomio F es la respuesta libre, y cogemos de
él
%la parte correspondiente a nuestro horizonte en el primer sector, y toda
%la información en el resto de F

%% Determina los coeficientes de control pasados

%Elimino el cero de la primera posicion de B en caso de tenerlo, como
%tenemos nosotros, antes de comenzar los cálculos.
if B(1)==0
    B=B(2:end);
end

uG=zeros(N1,N)'; %Vector de controles pasados (Pertenece a la respuesta
libre)

j=2;
for i=N1:N2
    aux=conv(En(i,1:end),B);
    if length(aux) < N1+j-1 %Si la longitud del auxiliar es menor que j
        aux=[aux zeros(1,(N1+j-1)-length(aux))]; %Completar con ceros
    end
    uG(j-1,1:N1)=aux(j:N1+j-1);
    j=j+1;
end

g=conv(E,B); %Calcula polinomio g, producto de E y B, es por tanto los
%coeficientes de la planta frente al escalón.

G=zeros(N,Nu); % Inicializa la matriz G, matriz de respuesta de los
%coeficientes del escalón

```

```

for k=1:Nu
    G(k:end,k)=g(1:N-k+1); % Forma la matriz G a partir del polinomio g
end

Mn=inv(G'*Qd*G+Ql)*G'*Qd; %Calculo de la Funcion de Costo sin Restriccion

%Calculo del controlador K1 (Primera fila de Mn)
K1=Mn(1,:);

%% Bucle de control
%Inicializamos los parámetros de simulación
drs=estudio; %, duracion de la simulación= NIT-1
nit=drs+1; %Número de interacciones
inc_u=0; %Incremento de control predictivo
Y=2; %COMIENZO DEL CONTROL
u_ant(1:Y) = 0; %Inicializamos valores de salida de controles anteriores
u(1:Y+d) = 0; ym(1:Y+d) = 0; r(1:Y+d) = 0;
%Inicializamos la señal de control, la salida y la referencia, que
durante
%un tiempo serán 0 para mejor visualización en la simulación. Cabe
destacar
%que, realmente ym=84, pero al ser esto un offset constante, lo
añadiremos
%más adelante, tratando los cálculos como si fuera de valor 0 durante el
%resto. Lo mismo ocurre con la referencia; inicialmente debería valer
igual
%que la salida ym. Además, como veremos más adelante, no empezaremos a
%controlar hasta el día N.

%Offsets de referencia y salida
pi(1:nit)=84;

for L=1:l:round(drs/der)
    if(rpa(L)>50)
        for H=((der*L)+1):1:(der*(L+1)+1)
            pi(H)=pi((der*L))+der;
            %         if(pi(H)>100)
            %             pi(H)=100;
            %         end
        end
    elseif(rpa(L)==50)
        for H=((der*L)+1):1:(der*(L+1)+1)
            pi(H)=pi((der*L));
        end
    else
        for H=((der*L)+1):1:(der*(L+1)+1)
            pi(H)=pi((der*L))-der;
            if(pi(H)<0)
                pi(H)=0;
            end %La saturación solo es por abajo; no te puedes sentir mejor
que "perfecto", pero sí peor que el máximo de la media
        end
    end
end

pi=pi(1:nit);

%Referencia del usuario (referida respecto del offset)
ref=8;

```

```

%Transformación de la referencia con offset a sin offset
%REFERENCIA COMPLEJA
% r(26:30)=- (pi-ref)+40; r(31:32)=- (pi-ref)+20;
% r(33:37)=- (pi-ref)+15;r(38:42)=- (pi-ref)+10;r(43:47)=- (pi-ref)+5;
% r(48:nit) = -(pi-ref);%El índice 26 indica un t=25, cuando comienza el
test
%REFERENCIA SIMPLE
r((Y+1):nit)=- (pi((Y+1):nit)-ref);

%Perturbación
do(1:nit)= 0;
%Como antes, el índice 86 indica t=85

lduf=size(F); %Averiguo la longitud del polinomio F
lduf=lduf(1,2); %Será el incremento de control de la respuesta libre

luG=size(uG); %Averiguo la longitud de uG
luG=luG(1,2);
duf=zeros(1,luG); %Vector de incrementos de control pasados de la
respuesta libre

%% Comenzamos a controlar
for k=(Y+1)+d:nit
    % Salida del proceso
    ym(k)=B(1)*u(k-1-d)+B(2)*u(k-2-d)-A(2)*(ym(k-1))-A(3)*(ym(k-
2))+do(k);

    %Cálculo de la respuesta libre teniendo en cuenta el incremento
    %de control de la respuesta libre
    free=0;
    for i=1:lduf
        free=free+ym(k-i+1)*F(:,i);
    end

    if luG==1 %Si uG es un vector columna
        free=free+duf*uG; %Invierto el producto, para que sea realizable
    else %Si uG es una matriz
        free=free+uG*duf';
    end

    %Cálculo de la variable de control por incrementos
    inc_u=K1*(r(k)-free);

    %En nuestro caso, k!=1 siempre, pero habría que tener en cuenta este
    %detalle, que la acción de control se basa en un sumatorio.
    if k==1
        u(k)=inc_u;
    else
        u(k)=u(k-1)+ inc_u;
    end

    dosis_a=u(k-1);

    %Realizamos una pequeña saturación de la variable de control
    if u(k)>13.5
        dosis_continua=13.5;
    elseif u(k)<0

```

```

        dosis_continua=0;
    else
        dosis_continua=u(k);
    end

%Discretizamos las dosis
if(dosis_continua > 12.54 && dosis_continua <= 13.5)
    dosis_n=13.5;
elseif(dosis_continua > 11.58 && dosis_continua <= 12.54)
    dosis_n=11.58;
elseif(dosis_continua > 10.59 && dosis_continua <= 11.58)
    dosis_n=11.58;
elseif(dosis_continua > 9.6 && dosis_continua <= 10.59)
    dosis_n=9.6;
elseif(dosis_continua > 8.64 && dosis_continua <= 9.6)
    dosis_n=9.6;
elseif(dosis_continua > 7.68 && dosis_continua <= 8.64)
    dosis_n=7.68;
elseif(dosis_continua > 6.72 && dosis_continua <= 7.68)
    dosis_n=7.68;
elseif(dosis_continua > 5.76 && dosis_continua <= 6.72)
    dosis_n=5.76;
elseif(dosis_continua > 4.805 && dosis_continua <= 5.76)
    dosis_n=5.76;
elseif(dosis_continua > 3.85 && dosis_continua <= 4.805)
    dosis_n=3.85;
elseif(dosis_continua > 2.885 && dosis_continua <= 3.85)
    dosis_n=3.85;
elseif(dosis_continua > 1.92 && dosis_continua <= 2.885)
    dosis_n=1.92;
elseif(dosis_continua > 0.96 && dosis_continua <= 1.92)
    dosis_n=1.92;
elseif(dosis_continua >= 0 && dosis_continua <= 0.96)
    dosis_n=0;
end

%Programamos un cambio de dosis controlado
switch dosis_n
    case 13.5
        if(dosis_a>=9.6)
            dosis_nueva=13.5;
        elseif(dosis_a==7.68)
            dosis_nueva=11.58;
        elseif(dosis_a==5.76)
            dosis_nueva=9.6;
        elseif(dosis_a==3.85)
            dosis_nueva=7.68;
        elseif(dosis_a==1.92)
            dosis_nueva=5.76;
        else
            dosis_nueva=3.85;
        end

    case 11.58
        if(dosis_a>=7.68)
            dosis_nueva=11.58;
        elseif(dosis_a==5.76)
            dosis_nueva=9.6;
        elseif(dosis_a==3.85)
            dosis_nueva=7.68;

```

```
elseif(dosis_a==1.92)
    dosis_nueva=5.76;
else
    dosis_nueva=3.85;
end

case 9.6
if(dosis_a>=5.76)
    dosis_nueva=9.6;
elseif(dosis_a==3.85)
    dosis_nueva=7.68;
elseif(dosis_a==1.92)
    dosis_nueva=5.76;
else
    dosis_nueva=3.85;
end

case 7.68
if(dosis_a==13.5)
    dosis_nueva=9.6;
elseif(dosis_a>=3.85 && dosis_a<=11.58)
    dosis_nueva=7.68;
elseif(dosis_a==1.92)
    dosis_nueva=5.76;
else
    dosis_nueva=3.85;
end

case 5.76
if(dosis_a==13.5)
    dosis_nueva=9.6;
elseif(dosis_a==11.58)
    dosis_nueva=7.68;
elseif(dosis_a>=1.92 && dosis_a<=9.6)
    dosis_nueva=5.76;
else
    dosis_nueva=3.85;
end

case 3.85
if(dosis_a==13.5)
    dosis_nueva=9.6;
elseif(dosis_a==11.58)
    dosis_nueva=7.68;
elseif(dosis_a==9.6)
    dosis_nueva=5.76;
elseif(dosis_a<=7.68)
    dosis_nueva=3.85;
end

case 1.92
if(dosis_a==13.5)
    dosis_nueva=9.6;
elseif(dosis_a==11.58)
    dosis_nueva=7.68;
elseif(dosis_a==9.6)
    dosis_nueva=5.76;
elseif(dosis_a==7.68)
    dosis_nueva=3.85;
elseif(dosis_a<=5.76)
    dosis_nueva=1.92;
```

```

        end

        case 0
            if(dosis_a==13.5)
                dosis_nueva=9.6;
            elseif(dosis_a==11.58)
                dosis_nueva=7.68;
            elseif(dosis_a==9.6)
                dosis_nueva=5.76;
            elseif(dosis_a==7.68)
                dosis_nueva=3.85;
            elseif(dosis_a==5.76)
                dosis_nueva=1.92;
            elseif(dosis_a<=3.85)
                dosis_nueva=0;
            end
        end

        u(k)= dosis_nueva;

        %Actualizamos el vector de valores de control pasados
        aux_u=u_ant(1:length(B)-1);
        u_ant=[u(k) aux_u];

        %Y actualizaremos los incrementos de control pasados, que se usan en
        la
        %respuesta libre.
        aux_2=duf(1:end-1);
        duf=[inc_u aux_2];

    end

    %% Para representar, solo debemos usar la funcion "graficas"
    %Esta sección solo muestra los resultados a solas
    nm=nit;
    if((drs/round(drs/der))>=der)
        rreal=rpa(1:round(drs/der));
    else
        rreal=rpa(1:(round(drs/der)-1));
    end
    tiem=der:der:drs;
    t = 0:T:(nm-1)*T;
    sal=ym+pi;
    for U=0:T:(nm-1)*T
        if(sal(U+1)<0)
            sal(U+1)=0;
        end
    end
    %figure
    subplot(2,1,1),plot(t, (r+pi), '--k', t, sal, 'r',
        tiem, rreal, '*g', 'Linewidth', 2)
    xlabel('t(dias)');ylabel('Síntomas (%)');title('Síntomas FM');
    legend('ref', 'Predictivo', 'Revision', 'Location', 'Best');
    grid on;
    hold on;
    subplot(2,1,2),plot(t, u, 'r', 'Linewidth', 2)
    xlabel('t(dias)');ylabel('Naltrexona (mg)'); title('Dosis Naltrexona');
    legend('Predictivo');
    grid on;
    hold on;

```



```

%% FUNCION DIOFANTINA
%-----
function[E,F] = diophantine(A,N,d)

delta = [1 -1];

AD = conv(A,delta);

nAD = size(AD);
nAD = nAD(2);

N1 = d + 1;
N2 = d + N;

f(1,:) = [1 zeros(1,nAD-2)];

for j = 1: N2
    for i = 1:nAD-2
        f(j+1,i) = f(j,i+1)-f(j,1)*AD(i+1);
    end
    f(j+1,nAD-1) = -f(j,1)*AD(nAD);
end

F = f(1+N1:1+N2,:);

% Computes E(z^-1)

E = zeros(N2);
e(1) = 1; % for the special case 1/~A

E(1,1) = e(1);

for i = 2: N2
    e(i) = f(i,1);
    E(i,1:i)=e;
end

E = E(N1:N2,:);
end

```

