

**GRADO EN INGENIERIA ELECTRÓNICA INDUSTRIAL Y
AUTOMÁTICA**

TRABAJO DE FIN DE GRADO

**Modelado del estado analgésico de pacientes y control mediante
técnicas de Q-learning**

Alumno: Angie Garzón Nieto

Tutor: Santiago Torres Álvarez

Índice

1. Introducción	1
1.1. Introducción General	2
1.2. El proceso anestésico: conceptos básicos	2
2. Farmacodinámica y Farmacocinética	4
2.1. Farmacocinética	5
2.2. Modelo compartimental	5
2.3. Modelo Farmacocinético	8
2.4. Farmacodinámica	14
2.5. Modelo Farmacodinámico	15
3. Modelado del proceso analgésico	17
3.1. Modelo Directo	18
3.2. Modelado Inverso	19
3.3. Comparación de los resultados obtenidos	24
3.4. Estudio de la respuesta en pacientes	24
4. Controlador	28
4.1. Q-learning	29
4.2. Funcionamiento del algoritmo en el caso a estudiar	31
5. Algesiómetro	39
5.1. Introducción	40
5.2. Obtención de medidas	40
5.3. Implementación del controlador en el algesiómetro	42
6. Anexos	44
6.1. Modelado	45
6.2. Código	73

1. Introducción

1.1. Introducción General

Objetivos

Este trabajo se centra en el modelado y control del estado analgésico de pacientes durante operaciones quirúrgicas. Por un lado, se aborda el problema del modelado del proceso presentando resultados validados con pacientes reales y comparándolos con los resultados obtenidos en otros estudios científicos. Asimismo, se propone una estrategia de control avanzado para regular el estado analgésico. El algoritmo empleado es el Q-learning que pretende determinar cuál es la acción óptima a tomar según el estado en el que se encuentre el paciente. Con esta investigación, en definitiva, se busca desarrollar una técnica que permita la regulación del estado analgésico del paciente, y que se ajuste a un paciente modelo mediante un controlador, para el cual se hace uso de un paciente modelo obtenido a través de los resultados reales del comportamiento que presentan los pacientes sometidos a intervenciones quirúrgicas.

También se propone la implementación de este algoritmo en una plataforma software de control que se comunica con un algesiómetro, un monitor de dolor de paciente mediante el cual se obtienen variables de interés con las que se determina el estado analgésico del paciente.

Objectives

This final degree work focuses on modelling and control of the patients analgesic state for surgery operations. On the one hand, we try to solve the problem of modelling, using real patients with checked results, and comparing them with the results of other scientific studies. On the other hand, an advanced controller is proposed to regulate the analgesic state. The algorithm used is Q-learning, which aims to determine the best action depending on the state in which the patient is found.

With this research, we try to develop a technique that allows us to regulate the patient state by using the proposed model patient. Through current results, we obtained the behaviour of patients undergoing surgical procedures.

A final goal of this work is to implement the proposed control algorithm for being used by a software platform control which communicates with an algesiometer, a pain monitor to determine a patient analgesic state.

1.2. El proceso anestésico: conceptos básicos

Se puede definir la anestesia general como un estado de inconsciencia producido por la infusión de fármacos. Este estado implica que temporalmente el paciente se sume en un profundo sueño de tal manera que no siente dolor durante la cirugía. El proceso anestésico por tanto, abarca tres grandes campos: la hipnosis o inconsciencia, la analgesia o ausencia de dolor y la relajación muscular, propicia para conseguir la inmovilidad del paciente. A lo largo de los últimos tiempos se ha investigado la manera de controlar estas tres variables con el fin de optimizar el proceso anestésico, y crear un entorno más favorable tanto para los anestesiólogos como para el propio paciente, que con la optimización de este proceso verá reducido los plazos de recuperación.

Este proyecto sigue una dinámica continuista de varios trabajos que se han ido realizando previamente en una línea de investigación del departamento, basada en el control de la infusión de anestesia y analgesia en pacientes sometidos a intervenciones quirúrgicas. Podemos hacer mención en este ámbito, a los últimos artículos publicados por el departamento en esta área:

- A Marrero, JA Méndez, JA Rebozo, I Martín, JL Calvo - Journal of clinical monitoring and computing, (2016). Adaptive fuzzy modeling of the hypnotic process in anesthesia.

- Juan A Méndez, Ayoze Marrero, Jose A Reboso, Ana León-Control Engineering Practice, (2016). Adaptive fuzzy predictive controller for anesthesia delivery.
- José Luis Casteleiro-Roca, Juan Albino Méndez Pérez, Andrés José Piñón-Pazos, José Luis Calvo-Rolle, Emilio Corchado-10th International Conference on Soft Computing Models in Industrial and Environmental Applications, (2015). Modeling the Electromyogram (EMG) of Patients Undergoing Anesthesia During Surgery.
- Santiago Torres, Ana León, Héctor Reboso, José A Reboso, Juan A Méndez,(2012). Closed-Loop Control of Anaesthetic Effect.

Concretamente este trabajo se centra en intentar hallar una herramienta que nos permita monitorizar el estado analgésico en el que se encuentra un paciente sometido a una intervención quirúrgica, para ello se ha tomado como referencia las siguientes citas bibliográficas:

Referencias

- [1] C.L. Lim, R.J. Barry, E. Gordon, A. Sawant, C. Rennie, C. Yiannikas (1995). The relationship between quantified EEG and skin conductance level. *International Journal of Psychophysiology*, 21: 151-162.
- [2] Juan Albino Méndez Pérez, José Antonio Reboso, Flora Segura García, Héctor Reboso, Santiago Torres, Felipe González. (2010). *Estrategias de Control en la Práctica de Anestesia*. Capítulo 3: Modelado del proceso anésteico.
- [3] Minto, Charles F. MB, ChB; Schnider, Thomas W. MD; Egan, Talmage D. MD; Youngs, Elizabeth MD; Lemmens, Harry J. M. MD; Gambus, Pedro L. MD; Billard, Valerie MD; Hoke, John F. PhD; Moore, Katherine H. P. PharmD; Hermann, David J. PharmD; Muir, Keith T. PhD; Mandema, Jaap W. PhD; Shafer, Steven L. MD. Influence of Age and Gender on the Pharmacokinetics and Pharmacodynamics of Remifentanil: Part I. Model Development.
- [4] Minto, Charles F. MB, ChB; Schnider, Thomas W. MD; Shafer, Steven L. MD. Pharmacokinetics and Pharmacodynamics of Remifentanil: Part II. Model Application.
- [5] Artificial intelligence “Q-learning” . Disponible en http://artint.info/html/ArtInt_265.html
- [6] Juan Deltell Mendicute Alberto Lorente Sánchez Jesús Martínez Dotor: “Resolución de laberintos en StarCraft empleando aprendizaje por refuerzo”
- [7] “Q-learning (Model-free Value Iteration) Algorithm for Deterministic Cleaning Robot”. Disponible en: <http://.mathworks.com>
- [8] Manual del usuario del Monitor de Dolor “Med-Storm”.

2. Farmacodinámica y Farmacocinética

2.1. Farmacocinética

Introducción

Para encontrar un estado ideal en el que el paciente esté protegido de la agresión de una intervención quirúrgica, es necesario calcular la dosis y la velocidad adecuada a la que se administra el fármaco.

Se puede definir el proceso anestésico ideal como un estado farmacológico en el que el paciente pierde la conciencia debido a la administración de fármacos, por consiguiente, nos hemos apoyado en los avances en farmacocinética y farmacodinámica para tratar de conseguir un estado analgésico acorde a la situación en la que se pueda encontrar el paciente .

Farmacocinética

Estudia el curso temporal de las concentraciones de los fármacos en el organismo, desde su administración hasta su total eliminación. Es decir, estudia los procesos de absorción, distribución, metabolismo y eliminación. Permite, por ende, construir modelos para valorar la acción terapéutica o tóxica de un fármaco. Como es de esperar, la respuesta fluctúa dependiendo de parámetros tales como enfermedades, situaciones clínicas, edad o efectos de la droga in-situ. En una operación quirúrgica, por tanto, se procede de la siguiente manera: Se inyecta un bolo para generar un aumento de la concentración plasmática de manera rápida. Posteriormente se pasa a la distribución del fármaco por todo el cuerpo. Factores como la frecuencia cardíaca, el gradiente de concentración o la afinidad del fármaco por los tejidos, determinan la efectividad de la infusión. Se considera, por tanto, absorbida aquella cantidad de dosis administrada que es susceptible de llegar a la biofase y ejercer su acción. Después, se procede a la descomposición del fármaco en el organismo, proceso de metabolismo, para terminar escrutándolo por alguna vía por lo general, por medio del sistema renal. Esta distribución heterogénea hace que se genere la teoría de los modelos compartimentales.

2.2. Modelo compartimental

Simplifica el estudio de la circulación del fármaco empleando la representación de los modelos para describir y predecir las relaciones de concentración-efecto, a partir de una información limitada como lo son las observaciones clínicas. Se considera de esta forma, que el organismo está dividido en compartimentos que representan espacios teóricos. Surgen aquí tres modelos principales:

- Modelo monocompartmental
- Modelo Bicompartimental
- Modelo Tricompartimental

Para el fármaco Remifentanil, se emplea este último modelo. Esta suposición está cotejada a partir del trabajo de: Minto, Charles F. MB, ChB; Schnider, Thomas W. MD; Egan, Talmage D. MD; Youngs, Elizabeth MD; Lemmens, Harry J. M. MD; Gambus, Pedro L. MD; Billard, Valerie MD; Hoke, John F. PhD; Moore, Katherine H. P. PharmD; Hermann, David J. PharmD; Muir, Keith T. PhD; Mandema, Jaap W. PhD; Shafer, Steven L. MD:

“Influence of Age and Gender on the Pharmacokinetics and Pharmacodynamics of Remifentanil: Part I. Model Development”

En el que se establece el modelo tricompartimental como el que mejor que describe el comportamiento del paciente. Se representa de esta forma, al paciente como un conjunto de tres compartimentos. La distribución del fármaco, para este modelo, se puede dividir en tres fases:

1. Fase de distribución rápida (α), el fármaco pasa del compartimento central (V_1) a los tejidos peor irrigados, compartimento periférico rápido (V_2).

2. Fase de distribución lenta (β) se caracteriza por el paso del fármaco desde el compartimento central (V_1) hacia los tejidos pobremente perfundidos, compartimento periférico lento (V_3).
3. Fase de eliminación o terminal (γ) en el cual se produce el retorno desde (V_3) y (V_2) hacia (V_1).

A la suma de los tres compartimentos o volúmenes, se le conoce como “volumen en el estado estacionario o de equilibrio”. En este estado, la velocidad de entrada del fármaco al compartimento periférico desde el compartimento central, se realiza a una velocidad que es igual a la velocidad de salida desde el compartimento periférico hacia el compartimento central. Asumimos pues que los compartimentos están en equilibrio a la hora de realizar el estudio, para simplificar cálculos. Cada fase, posee implícitamente constantes de difusión que representan la velocidad a la que se administra el fármaco. Por ejemplo, para la distribución del fármaco entre el compartimento central y el rápido se emplean las constantes k_{12} y k_{21} y para representar la circulación entre el compartimento central y el lento se emplean las constantes k_{13} y k_{31} . La eliminación metabólica se representa mediante la constante de difusión k_{10} .

Caso aparte, es la constante de difusión del sitio-efecto, k_{e0} , que representa el ratio de eliminación definitiva de la droga fuera del sitio efecto. Entiéndase por sitio-efecto al lugar donde la concentración hace impacto en el paciente.

Se caracteriza por ser un valor inferido según el compartimento de la concentración central plasmática con el efecto farmacodinámico medido por el electroencefalograma (EEG), como en hipnóticos y opioides, o por la neuroestimulación muscular (en relajantes musculares).

Otra singularidad de esta constante es que tiene un compartimento asociado. Esto se debe a que al colocar un medicamento intravenoso, hay un tiempo de retraso llamado histéresis, el cual comprende entre la concentración plasmática alcanzada y la respuesta clínica observada (tiempo de latencia). El retraso provocado, es porque el sitio de acción del medicamento no se encuentra en el plasma (V_1), por lo cual el medicamento debe pasar del plasma (V_1) al sitio-efecto (V_e). Este nuevo compartimento, V_e al tener un volumen tan pequeño, desprecia la cantidad de droga eliminada del sitio efecto, y se considera, en consecuencia, que no regresa al compartimento central.

La constante de difusión del sitio-efecto, k_{e0} , por tanto, es el enlace entre la fase farmacocinética y la fase farmacodinámica y su importancia clínica radica en que una k_{e0} pequeña implica una gran histéresis y una gran $t_{1/2k_{e0}}$, que es el tiempo en que la concentración en el que el sitio del efecto alcanza el 50 % de la concentración plasmática cuando ésta se mantiene constante. Para el Remifentanil por ejemplo, se puede observar que hay una gran constante de difusión del sitio-efecto, k_{e0} y por ello una histéresis pequeña por consiguiente, hay un inicio de efecto rápido, lo cual permite ajustar la concentración plasmática dentro de rangos terapéuticos relativamente estrechos, dando flexibilidad a la estrategia del tratamiento.

Como se mencionó anteriormente, el volumen del compartimento sitio-efecto V_e está dentro del volumen del compartimento central, V_1 . Por lo tanto, al hacerse pequeño el volumen del compartimento central, disminuye el volumen del compartimento sitio-efecto y por consiguiente, se hace más pequeña la constante de difusión sitio-efecto, k_{e0} .

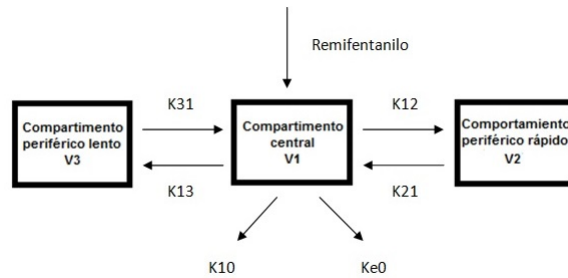


Figura 1: Modelo tricompartmental en donde se representa al paciente como un conjunto de compartimentos

Según el proceso que se trate se pueden presentar diversos modelos. Para el procedimiento de la analgesia, que es el que nos compete, se emplea el Modelo de Minto. Éste es un modelo que se caracteriza por ser capaz de aplicar técnicas de control en este campo. Para realizar este control se pueden emplear diversos fármacos pero el usado en las pruebas realizadas por el hospital, ha sido Remifentanil.

Esta elección se basa en que presenta, a diferencia de otros opioides, un rápido aclaramiento y una rápida constante de difusión sitio-efecto, k_{e0} , lo que conlleva un rápido inicio y desplazamiento del efecto del fármaco. En base al fármaco elegido, y mediante las siguientes referencias, comprobaremos la influencia de la edad y de la masa muscular magra:

Charles F. Minto, Thomas W. Schnider and Steve L. Shafer, "Pharmacokinetics and pharmacodynamics of Remifentanil, Anesthesiology", 1997, 86, 24-33

En este ensayo, se expone el efecto del Remifentanil en 65 voluntarios mediante la medición de variables extraídas del electroencefalograma (EEG). Estos voluntarios fueron escogidos de una población de 500 personas. Se escogió una muestra de manera que estuvieran presentes todos los rangos de edad, 20, 50 y 80 años. En este informe se observó que la concentración pico de sitio-efecto no depende de la edad, sin embargo este pico se produce más tarde en individuos de edad avanzada. Esto significa que, un anciano de 80 años requiere aproximadamente la mitad de dosis en bolo, en comparación con una persona de 20 años de edad. Asimismo, la velocidad de infusión requerida para mantener el 50% del efecto en una persona de 80 años, es aproximadamente un tercio de la que necesitaría una persona de 20 años. Como resultado las personas de avanzada edad pueden tener ocasionalmente una aparición más lenta de la anestesia de lo esperado.

Esto nos lleva a concluir que la edad es un aspecto importante a considerar en nuestro análisis. Seguidamente se puede hacer referencia a otro artículo en donde se pone de relieve la importancia de la edad y de la masa muscular magra:

Charles F. Minto, Thomas W. Schnider, Talmage D. Egan, Elizabeth Youngs, Harry J. M. Lemmens, Pedro L. Gambús, Valerie Billard, John F. Hoke, Katherine H. P. Moore, David J. Hermann, Keith T. Muir, Jaap W. Mandema and Steven L. Shafer, "Influence of Age and Gender on the Pharmacokinetics and Pharmacodynamics of Remifentanil, Anesthesiology", 1997, 86, 10-23.

El objetivo de este estudio fue determinar la influencia de la edad y el género, en la farmacocinética y la farmacodinámica mediante el uso del Remifentanil. Para ello se escogió una muestra significativa de 65 personas (38 hombres y 27 mujeres) con edades desde los 20 años hasta los 85 años. Se expone que para el análisis farmacocinético tricompartmental, la edad influye en los volúmenes y aclaramientos, además de asegurar que la masa muscular magra es una covariable de V_1 , V_2 y Cl_1 en el mejor de los modelos, entiéndase por covariable a una variable continua independiente que junto a una o más variables sirven para explicar una variable de respuesta continua.

Se estima por tanto, que se deben tener en cuenta como covariables a la edad y a la masa muscular magra a la hora del cálculo de los parámetros farmacocinéticos.

2.3. Modelo Farmacocinético

Teniendo en cuenta la covariable de la edad y la masa muscular magra, Modelo de Minto propone unas ecuaciones para estimar los valores.

Las ecuaciones propuestas son las siguientes:

$$A = \text{Age}-40$$

$$L = \text{LBM}-55$$

Volúmenes (l):

- Compartimento central
 - $V_1 = 5.1 - 0.0201 \cdot A + 0.072 \cdot L$
- Compartimento periférico rápido
 - $V_2 = 9.82 - 0.0811 \cdot A + 0.108 \cdot L$
- Compartimento periférico lento
 - $V_3 = 5.42$

Aclaramiento (l/min):

- Metabólico
 - $Cl_1 = 2.6 - 0.0162 \cdot A + 0.0191 \cdot L$
- Periférico rápido
 - $Cl_2 = 2.05 - 0.0301 \cdot A$
- Periférico lento
 - $Cl_3 = 0.076 - 0.00113 \cdot A$

Una vez obtenido estos valores se pueden obtener las constantes de difusión del Modelo de Minto:

- $k_{12} = Cl_2 / V_1$
- $k_{13} = Cl_3 / V_1$
- $k_{21} = Cl_2 / V_2$
- $k_{31} = Cl_3 / V_3$
- $k_{10} = Cl_1 / V_1$
- $k_{e0} = 0.595 - 0.007 \cdot A$

Aplicando las ecuaciones presentadas anteriormente que describen el modelo PK de Minto, se pueden obtener las matrices de estados que definen el modelo matemático utilizado para la administración de Remifentanil:

$$A = \begin{bmatrix} -(k_{10} + k_{12} + k_{13}) & k_{12} \cdot (V_2/V_1) & k_{31} \cdot (V_3/V_1) & 0 \\ k_{12} \cdot (V_1/V_2) & -k_{12} & 0 & 0 \\ k_{31} \cdot (V_1/V_3) & 0 & -k_{31} & 0 \\ k_{e0} & 0 & 0 & -k_{e0} \end{bmatrix}$$

$$B = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

$$C = [0 \ 0 \ 0 \ 1]$$

$$D = 0$$

Con la finalidad de comprobar lo propuesto en las investigaciones anteriormente citadas, se va a emplear los datos de pacientes reales y se va a comprobar si las hipótesis realizadas tienen fundamento práctico.

Comparación de resultados obtenidos con pacientes reales

En el primer artículo se sostenía que una persona con avanzada edad tendría el pico de concentración sitio-efecto más alto que una persona joven y que por ende, una persona con menos edad necesita menos concentración en el sitio-efecto para obtener la misma respuesta.

Para demostrar que esto es cierto, hemos escogido a dos pacientes con las siguientes características:

Paciente	Sexo (género)	Edad (años)	Altura (cm)	Peso (kg)
1ºPaciente	Mujer	63	170	90
2ºPaciente	Mujer	19	177	51

A estas dos pacientes se les va a someter a las mismas condiciones de operación, para ello se hace uso de los eventos registrados para la 1ºPaciente y se genera una gráfica donde se representa la variación de picos por segundo y la concentración sitio-efecto de Remifentanil con la edad, la altura y el peso de cada uno de los pacientes.

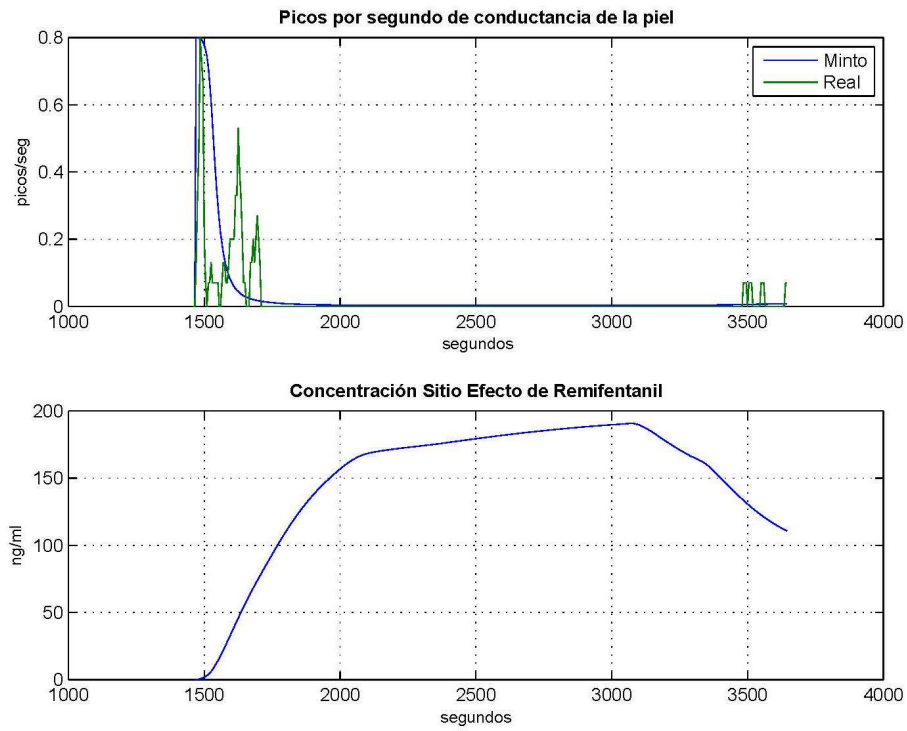


Figura 2: Representación del Modelo de Minto para la 1ª Paciente (mujer de avanzada edad) en donde en la primera gráfica se representa las variaciones de picos por segundos (*picos/seg*) con respecto al tiempo (*s*) y en la segunda gráfica se representa los cambios de concentración sitio-efecto de Remifentanil (*ng/ml*) con respecto al tiempo(*s*)

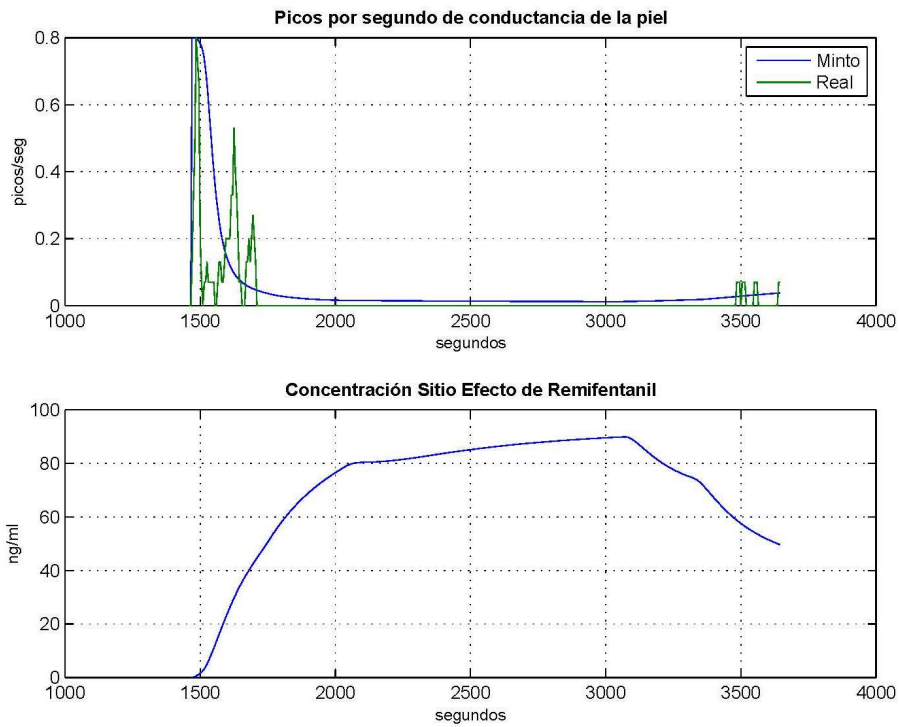


Figura 3: Representación del Modelo de Minto para el 2º Paciente (mujer joven) en donde en la primera gráfica se representa las variaciones de picos por segundos (*picos/seg*) con respecto al tiempo (*s*) y en la segunda gráfica se representa los cambios de concentración sitio-efecto de Remifentanil (*ng/ml*) con respecto al tiempo (*s*)

Como se puede comprobar la hipótesis planteada es correcta, ya que para la paciente de mayor edad, 1º Paciente, el pico de concentración sitio efecto es mucho mayor que para la paciente más joven (2º Paciente) y por consiguiente, para obtener la misma dinámica en los dos pacientes, es necesario variar la cantidad de concentración sitio-efecto.

En el segundo artículo se afirma que hay una fuerte dependencia de la edad y la masa muscular magra en el volumen acumulado en los compartimentos. Concretamente en el artículo se hace mención a que, la inclusión de masa muscular magra en el compartimento central, en el compartimento periférico rápido y en el aclaramiento del compartimento central, es necesaria para obtener un modelo que se ajusta más a las condiciones reales. Para dar crédito a esta afirmación se va a graficar a una serie de pacientes con diferentes edades y LBM con respecto a V_1 (Compartimento central).

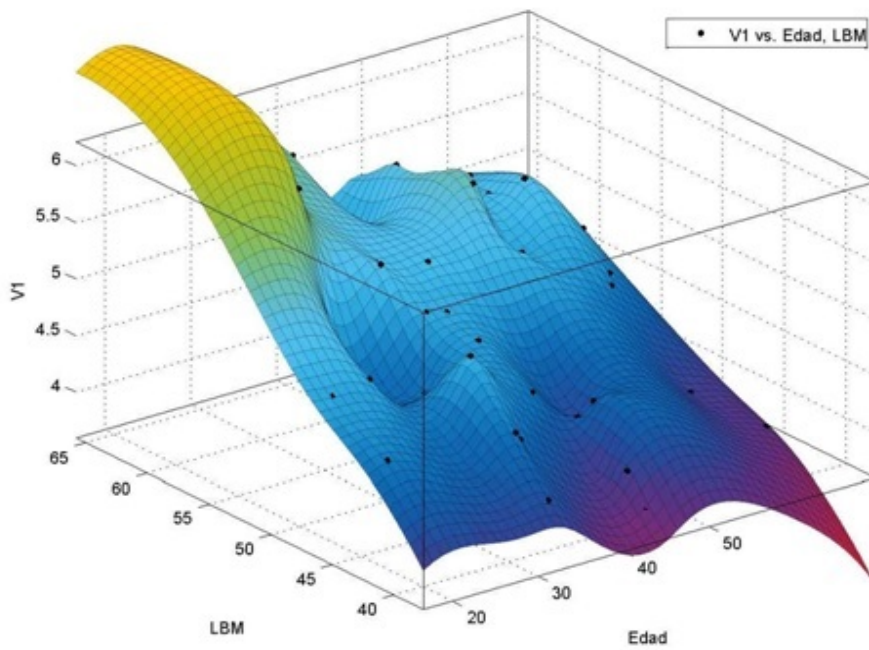


Figura 4: Representación de $V_1(l)$ con respecto a la Edad (Años) y a LBM en donde se manifiesta la dependencia entre estas variables

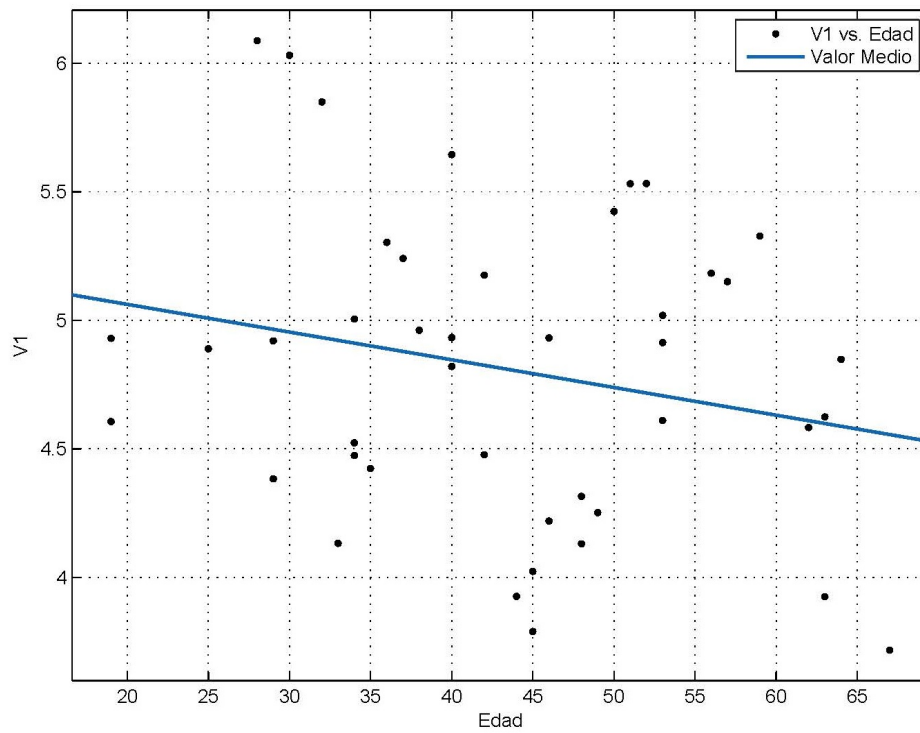


Figura 5: Representación de $V_1(l)$ con respecto a la Edad (Años) donde se pone de manifiesto la dependencia lineal entre estas variables

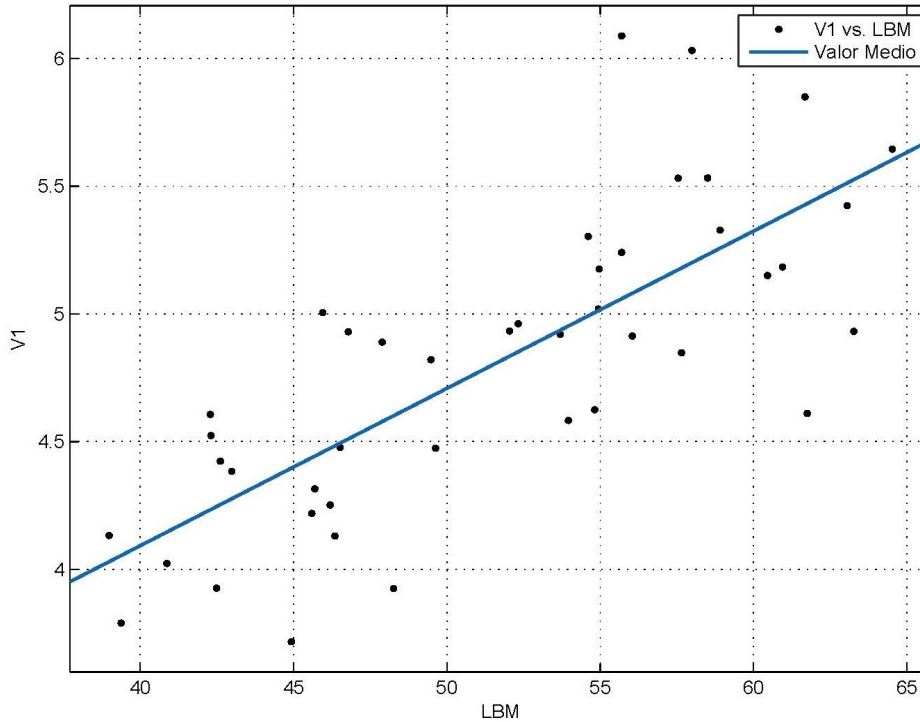


Figura 6: Representación de V_1 (l) con respecto a LBM donde se pone de manifiesto la dependencia lineal entre estas variables

Por un lado, hay que tener en cuenta que se produce una dispersión en los datos recogidos; pero podemos asegurar que igualmente los datos presentan una tendencia lineal. Al examinar con mayor detalle las ilustraciones anteriores, vemos que con esta estimación se puede considerar que hay una dependencia directa entre estos parámetros. En conclusión, se puede afirmar que estas variables son influyentes en el volumen que puedan tener los compartimentos.

Por otra parte también se estima que los parámetros de la farmacocinética tienen un valor concreto.

Escogiendo una muestra de 43 individuos de diferente género, edad y peso procedemos a comparar, haciendo una media aritmética de los resultados obtenidos en la investigación con las muestras recogidas del hospital:

Datos	V_1 (l)	V_2 (l)	Cl_1 (l/min)	Cl_2 (l/min)	Cl_3 (l/min)
Hospital	4.809	9.172	2.485	1.933	0.080
Estudio	4.98	9.01	2.46	1.69	0.065

Como podemos observar el compartimento periférico lento no se compara puesto que se estima que tiene un valor fijo, como en el Modelo de Minto se contempla. De esta forma, vemos que aproximadamente estos valores se ajustan a lo que en la investigación realizada se determina. Las conclusiones a las que se llegan, por consiguiente son:

- Tanto la edad como la masa muscular magra o LBM, son covariables de los parámetros farmacocinéticos.
- Concretamente se demuestra que es necesario la inclusión de la masa muscular magra o LBM a la hora de obtener el volumen compartimental central (V_1), el volumen compartimental periférico rápido (V_2) y el aclaramiento del compartimento central (Cl_1)

- Por otra parte, se llegó a la deducción de que el género no es influyente para el cálculo de los parámetros farmacocinéticos.

2.4. Farmacodinámica

Describe la relación entre la concentración de droga y el efecto real de dicha droga. Como el mecanismo de acción de las drogas se analiza a nivel molecular, esta rama comprende el estudio de cómo una molécula de una droga interactúa con otra molécula originando una respuesta (acción farmacológica).

En la relación concentración- efecto es importante estudiar cuatro parámetros que determinan la eficiencia del fármaco durante la anestesia. Estos son la potencia, la eficacia, la pendiente de la curva y el tiempo para alcanzar el máximo efecto.

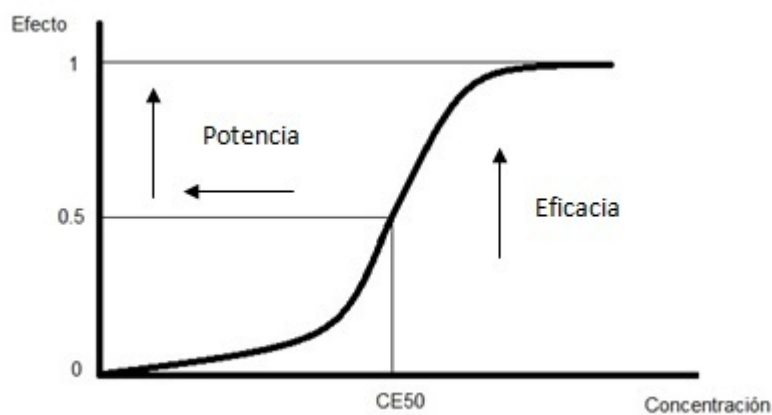


Figura 7: Curva sigmoidea donde en el eje de ordenadas se representa el grado de efecto conseguido (el valor de $1 = E_{m\acute{a}x}$), y en el eje de abscisas el logaritmo de la concentración alcanzada

Potencia

Determina la concentración de fármaco necesario para obtener un efecto preciso. Viene delimitado por la EC_{50} , cuánto menor sea ésta, más potente será el fármaco, ya que necesita una dosis menor para producir el efecto. Esto implica que cuánto más desplazada a la izquierda esté, más potente será.

Eficacia

Es la capacidad de un fármaco de producir un efecto. Cuanto mayor sea la eficacia de un fármaco mayor altura en el eje de ordenadas alcanzará la dosis-respuesta. Asimismo hay que tener presente, el techo terapéutico, por mucho que se aumente la dosis no se conseguirá una buena respuesta clínica, si éste ha alcanzado el máximo efecto farmacológico que puede tener.

Pendiente de la curva

Nos informa sobre cómo pueden influir pequeños cambios en la dosis sobre el efecto de fármaco. Por ejemplo, entre más pronunciada sea la pendiente, mayor será la influencia del bolo.

Tiempo para alcanzar el máximo efecto

Este concepto fue incorporado por Minto con el objetivo de predecir un mejor enlace entre la farmacocinética y la farmacodinámica.

El "t'peak" se refiere al tiempo de la máxima concentración en el sitio efecto después de un bolo intravenoso cuando no existe fármaco inicial en el sistema y permite predecir la respuesta clínica que presenta un paciente al conocerse de una manera más precisa, el inicio o el cese del efecto clínico deseado. Así pues, con el "t'peak" es posible calcular el valor que le corresponda a la k_{e0} dentro de un conjunto de parámetros farmacocinéticos que interesen en cada caso, describiendo así la evolución en el tiempo de la concentración en el compartimento efecto.

2.5. Modelo Farmacodinámico

Con el modelo de Minto se pretende dar cuenta de la sensibilidad de un paciente frente al dolor, empleando al Remifentanil como fármaco y a los picos por segundo como variable que describe la variación de la conductancia de la piel. Podemos, por tanto, con este modelo, se representan los picos por segundo de conductancia de piel a partir de la concentración sitio efecto de Remifentanil, como da cuenta de ello la siguiente función:

$$\frac{Picos}{seg} = f(C_e)$$

Empleando el modelo sigmoideal EMAX, la función anterior puede venir descrita por la siguiente ecuación:

$$Efecto = E_0 + (E_{m\acute{a}x} - E_0) \cdot \frac{C_e^\gamma}{C_e^\gamma + EC_{50}}$$

Donde E_0 es la medida basal del efecto en ausencia de droga, E_{max} es el máximo efecto medido, C_e es la concentración aparente de Remifentanilo en el sitio efecto, EC_{50} es la concentración en el sitio efecto para la que el efecto es la mitad del valor máximo y representa la sensibilidad del paciente a las pequeñas variaciones de concentración en el lugar efecto, se emplea, además, como índice que mide el grado de no linealidad del modelo.

Se hace otra vez alusión al segundo artículo *"Influence of Age and Gender on the Pharmacokinetics and Pharmacodynamics of Remifentanil"* con el fin de estimar el valor de los parámetros farmacodinámicos. Estos valores variarán según si se trata de un modelo simple farmacodinámico sin covariables o para un modelo complejo en el que se tiene en cuenta la covariable de la edad.

La estimación de los parámetros serían los siguientes:

Sin covariables (Modelo simple):

- $k_{e0}=0.516 \text{ l/min}$
- $E_0= 20 \text{ Hz}$
- $E_{m\acute{a}x}=11.20 \text{ Hz}$
- $\gamma=2.44$

Con covariables (Modelo complejo):

- $k_{e0}=0.595-0.007 \cdot (\text{Age}-40) \text{ l/min}$
- $E_0= 20 \text{ Hz}$
- $EC_{50}=13.1-0.148(\text{Age}-40) \text{ ng/ml}$

- $\gamma=2.44$

Con la consecución del modelado se determinarán estos parámetros para el objetivo que nos compete.

3. Modelado del proceso analgésico

3.1. Modelo Directo

Con el fin de hacer uso de la capacidad de predicción del comportamiento del sistema para mejorar el rendimiento de los controladores basados en señales, se empleará la técnica del modelado. Para la consecución de este proceso se ha modificado los valores pertenecientes al modelo de Minto, estos parámetros han sido EC_{50} y el parámetro γ . Para ello se ha estudiado la respuesta y según como variaba ésta gráficamente, se ha hallado unos valores que se ajustaban a un paciente modelo. Realizamos un estudio sobre cómo evoluciona el paciente con la variaciones de los parámetros EC_{50} y γ :

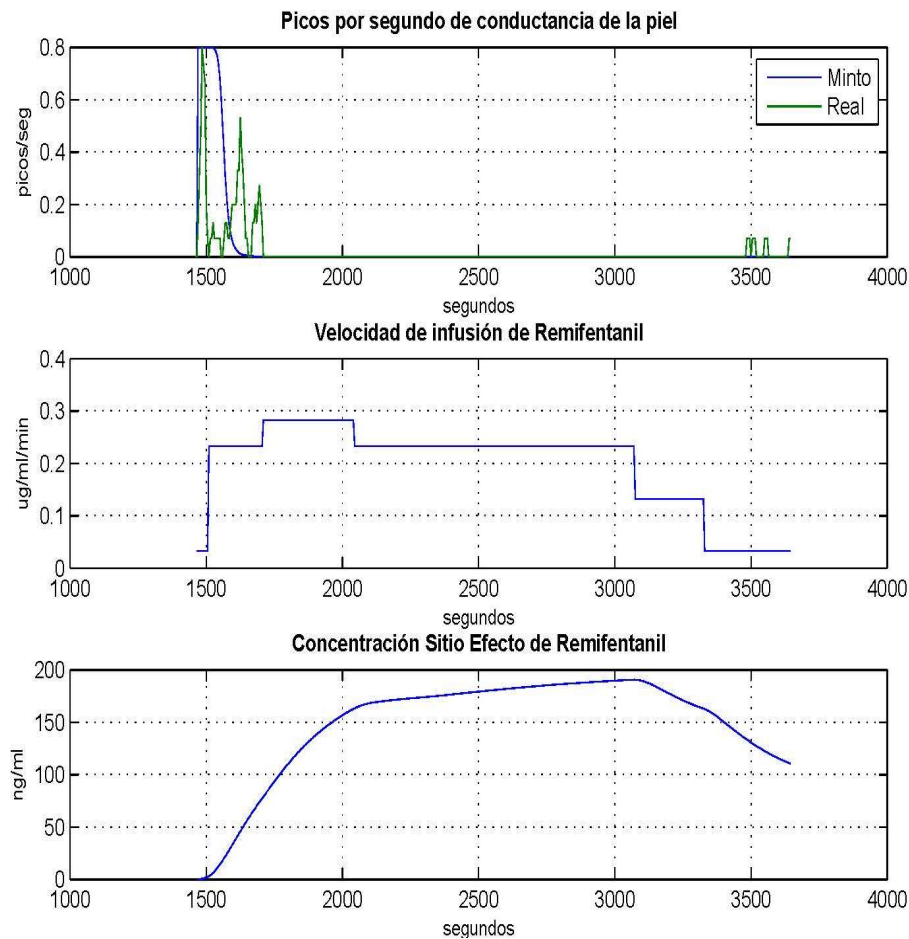


Figura 8: Modelado directo a partir de la aplicación del Modelo de Minto en un paciente para una $EC_{50}=20$ ng/ml y una $\gamma=5$ en donde se refleja los cambios de picos por segundo (*picos/seg*), las variaciones de velocidad ($\mu g/ml/min$) y las alteraciones de la concentración sitio-efecto de Remifentanil (ng/ml) con respecto al tiempo(s)

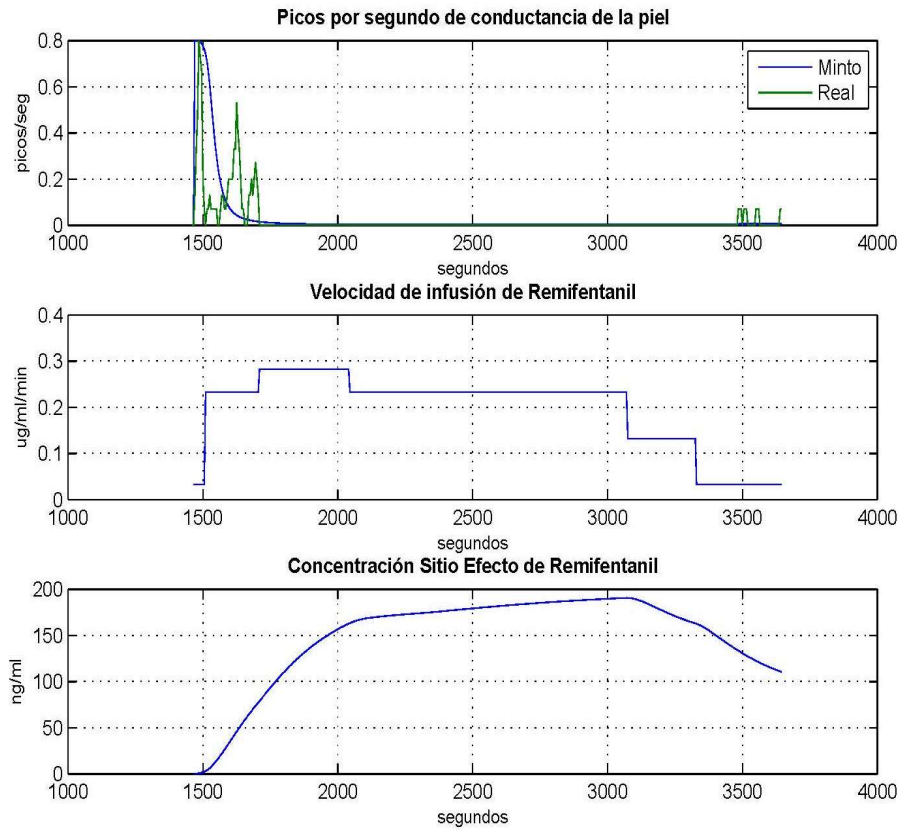


Figura 9: Modelado directo a partir de la aplicación del Modelo de Minto en un paciente para una $EC_{50} = 11 \text{ ng/ml}$ y $\gamma = 2$ en donde se refleja los cambios de picos por segundo (picos/seg), las variaciones de velocidad ($\mu\text{g/ml/min}$) y las alteraciones de la concentración sitio-efecto de Remifentanil (ng/ml) con respecto al tiempo (s)

Aun variando significativamente los valores de la EC_{50} y de γ vemos que prácticamente no hay una diferencia reseñable entre las dos hipótesis contempladas. Además se observa que el modelado para los dos casos no se ajusta bien a la dinámica que presenta el paciente. Por consiguiente, se ha de buscar una manera alternativa de poder aproximarnos al comportamiento deseado.

3.2. Modelado Inverso

Para ello se emplean las siguientes ecuaciones del modelo EMAX:

$$\alpha = \frac{P_s - P_{s0}}{P_{smax} - P_{s0}}$$

$$C_e = \exp\left(\frac{1}{\gamma}\right) \cdot \ln\left(EC_{50}^\gamma \cdot \frac{\alpha}{1 - \alpha} \right)$$

Como vemos es necesario, para el cálculo de la concentración equivalente de Remifantnil en el comportamiento efecto, fijar los valores de picos por segundo mínimos, máximos y deseados. En base a las variaciones que presentan los pacientes a modelar se ha establecido lo siguiente:

Picos por segundo mínimos (P_{s0}) = 0

Picos por segundo máximos ($P_{smáx}$) = 0.8

Picos por segundo deseados (P_s) = 0.60

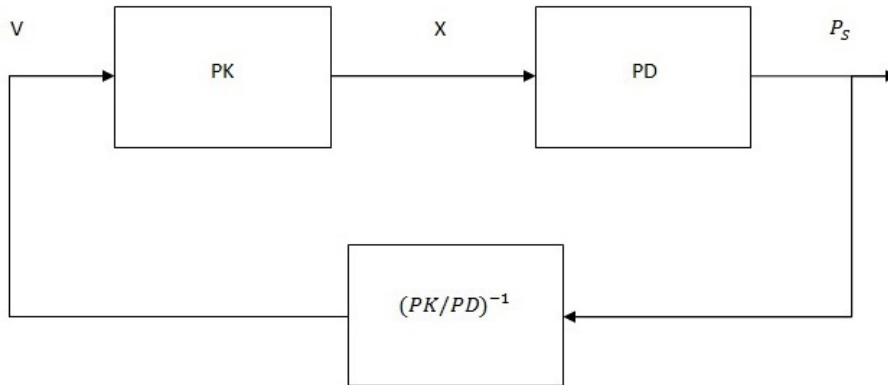


Figura 10: Representación del modelo inverso en forma de diagrama de bloques

El modelo se puede expresar de manera matricial como:

$$\dot{C}(t) = AC(t) + Bu(t)$$

Donde el estado del paciente viene dado por el valor de las concentraciones de Remifentilil en los distintos comportamientos:

$$C(t) = [C_1(t)C_2(t)C_3(t)C_e(t)]^t$$

Y por las matrices de estado:

$$A = \begin{bmatrix} -(k_{10} + k_{12} + k_{13}) & k_{12} \cdot (V_2/V_1) & k_{31} \cdot (V_3/V_1) & 0 \\ k_{12} \cdot (V_1/V_2) & -k_{12} & 0 & 0 \\ k_{31} \cdot (V_1/V_3) & 0 & -k_{31} & 0 \\ k_{e0} & 0 & 0 & -k_{e0} \end{bmatrix}$$

$$B = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Si se asume que nos encontramos en el estado estacionario es decir que:

$$AC_n + Bu_n = 0$$

O lo que es lo mismo:

$$A \cdot \begin{bmatrix} x_1^0 \\ x_2^0 \\ x_3^0 \\ x_4^0 \end{bmatrix} + B \cdot u^0 = 0$$

Podemos calcular las concentraciones de los compartimentos en el estado de equilibrio, para ello se plantea lo siguiente:

$$Mx' = N \text{ entonces } x' = M^{-1} N$$

$$Mx' = N$$

entonces:

$$x' = M^{-1} N$$

Donde x' , M y N están definidas como:

$$x' = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ u \end{bmatrix}$$

$$M = \begin{bmatrix} -(k_{10} + k_{12} + k_{13}) & k_{12} \cdot (V_2/V_1) & k_{31} \cdot (V_3/V_1) & 0 \\ k_{12} \cdot (V_1/V_2) & -k_{12} & 0 & 0 \\ k_{31} \cdot (V_1/V_3) & 0 & -k_{31} & 0 \\ k_{e0} & 0 & 0 & 0 \end{bmatrix}$$

$$N = \begin{bmatrix} 0 \\ 0 \\ 0 \\ -k_{e0} \cdot C_e \end{bmatrix}$$

Con la implementación de estas ecuaciones en el modelo de Minto podemos obtener la concentración equivalente de Remifentanil en el compartimento efecto, que produce el valor deseado, en nuestro caso este valor corresponde a 0.60 picos/seg. Con esta nueva entrada, se espera que la respuesta evolucione favorablemente.

Así pues, hecha esta mejora, procedemos a ver si se consigue mejorar la respuesta del paciente:

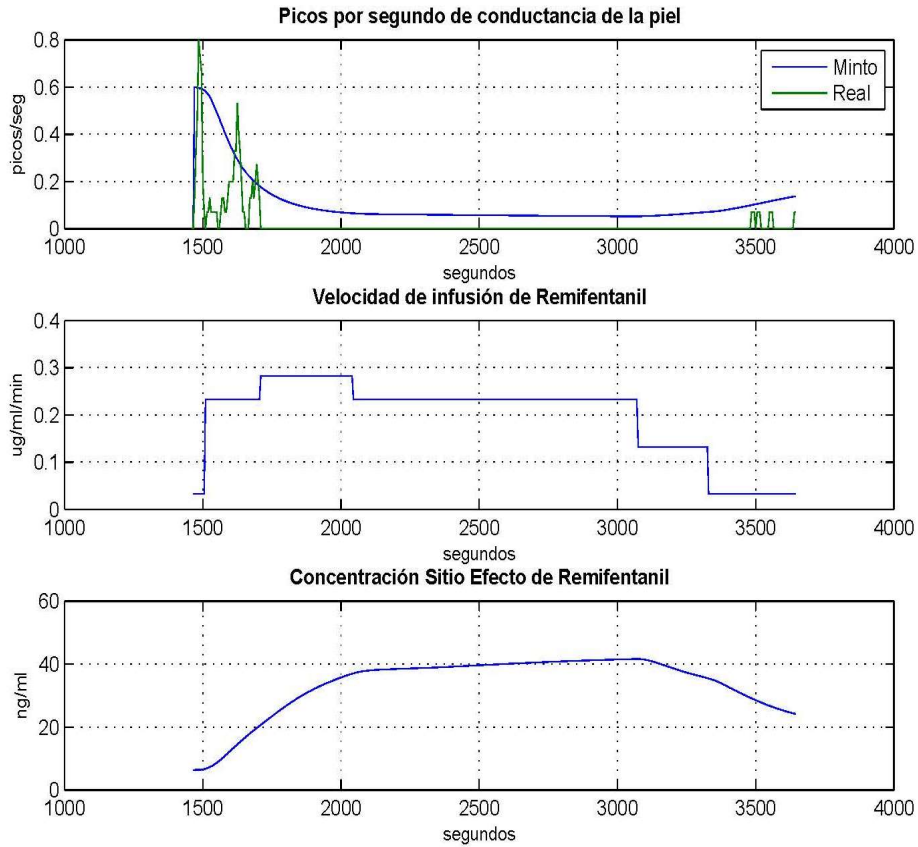


Figura 11: Modelado a partir de la aplicación de la dinámica inversa a un paciente con $EC_{50} = 11 \text{ ng/ml}$ y $\gamma = 2$ en donde se refleja los cambios de picos por segundo (*picos/seg*), las variaciones de velocidad ($\mu\text{g/ml/min}$) y las alteraciones de la concentración sitio-efecto de Remifentanil (ng/ml) con respecto al tiempo (s)

Vemos que sí se produce un mejor modelado, ya que ahora el modelo de Minto se ajusta mejor al modelo real del paciente; como es apreciable en la imagen anterior. A medida que se aumenta la concentración sitio-efecto, los picos por segundo disminuyen paulatinamente, sólo cuando se deja de infundir la dosis de Remifentanil es cuando éstos picos vuelven a aumentar.

Si, además, tuviéramos en cuenta un incremento de los parámetros γ y EC_{50} , como posible alternativa para un mejor modelado, obtendríamos la siguiente respuesta:

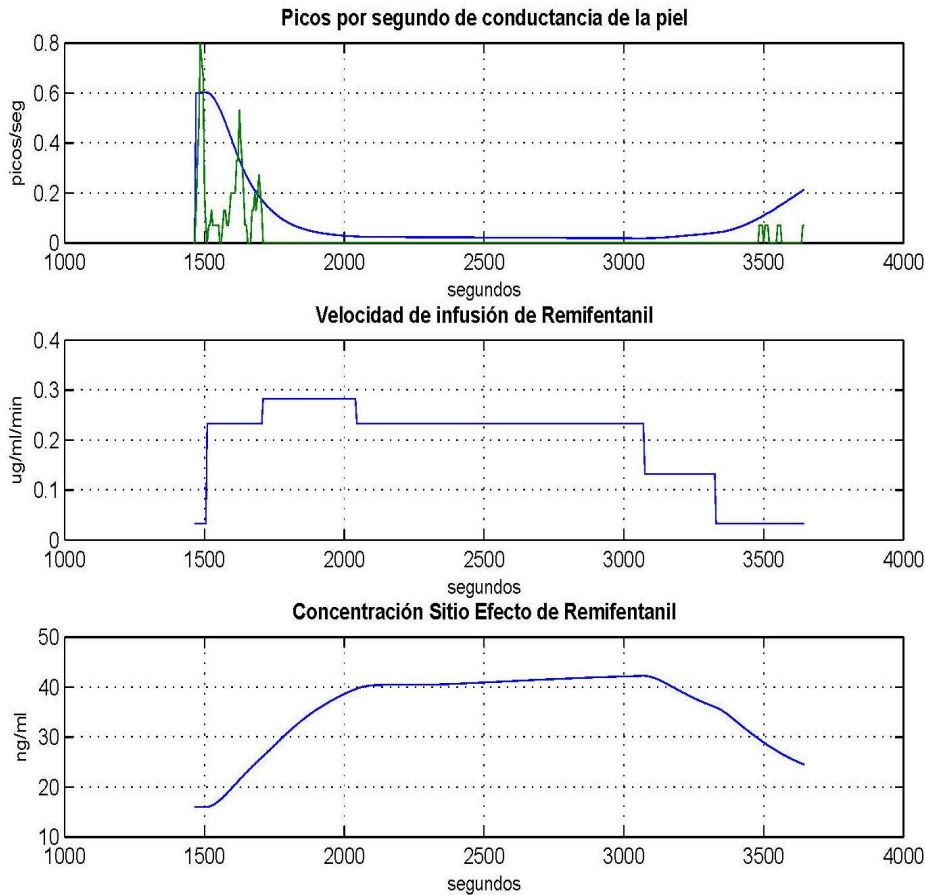


Figura 12: Modelado a partir de la aplicación de la dinámica inversa a un paciente con $EC_{50} = 20 \text{ ng/ml}$ y $\gamma = 5$ en donde se refleja los cambios de picos por segundo (*picos/seg*), las variaciones de velocidad ($\mu\text{g/ml/min}$) y las alteraciones de la concentración sitio-efecto de Remifentanil (ng/ml) con respecto al tiempo (s)

Como podemos ver la diferencia más reseñable, reside en que en la ilustración 12-con los parámetros de γ y EC_{50} bastantes altos- el modelo se ajusta un poco mejor a la respuesta real del paciente. Por el contrario, en la imagen 11 vemos como el modelo no tiene una respuesta tan fidedigna pero la variación de picos por segundo es mucho más gradual, así mismo, se observa que al colocar valores altos de γ y EC_{50} , podemos hacer al sistema no lineal y/o además de tener una respuesta efecto inicial mucho más lenta. En este aspecto, si lo comparamos con la dinámica que presenta este modelado con el que se hacía anteriormente, observamos que el valor máximo de concentración sitio-efecto era de casi 200 ng/ml mientras que ahora no llega a los 45 ng/ml , este decremento se debe a que ahora, con esta técnica, se ha conseguido que el comportamiento del modelo ante el mismo fármaco, se adapte de manera más óptima a la respuesta.

Estos factores nos hacen decantarnos por unos valores bajos de γ y EC_{50} .

Haciendo el análisis de cada uno de los pacientes, y teniendo en cuenta las consideraciones anteriores, hemos obtenido que los parámetros que mejor se ajustan al comportamiento medio del paciente sean los siguientes:

- $EC_{50} = 11 \text{ ng/ml}$
- $\gamma = 2$

3.3. Comparación de los resultados obtenidos

Si comparamos estos valores con los obtenidos en el modelo farmacodinámico de simple sigmoide y con el modelo complejo farmacodinámico, vemos que hay ligeras diferencias.

Esto es porque en el modelo de simple sigmoide no se contempla el decremento lineal que sufren k_{e0} y EC_{50} debido a la edad, sino que se estiman unos valores en función de la respuesta de la población muestreada. Se considera, por esto mismo, que el mejor modelo es el farmacodinámico complejo.

No se presenta una diferencia significativa con los datos obtenidos del muestreo realizado. Esto es observable en la siguiente tabla:

Modelo	k_{e0}	EC_{50}	γ
Simple	0.516	11.20	2.51
Complejo	0.596	13.1	2.44
Obtenido	0.569	11	2

3.4. Estudio de la respuesta en pacientes

Resultados de picos por segundo

Como podemos ver en el anexo adjunto, el modelado propuesto se ajusta a la dinámica del paciente. Sin embargo, se observa que los cambios bruscos y repentinos de picos por segundo que el paciente siente en una operación, no se pueden generar con el modelo. Esto es debido a que estos cambios en el comportamiento, no son provocados por lo general, por la disminución de fármaco en el paciente sino debido a los ruidos o artefactos que se pueden generar en la cirugía.

Seguidamente, es preciso comentar que los cambios en la concentración sitio-efecto provocan una variación de velocidad de infusión y repercuten asimismo en los cambios de picos por segundo. Por ejemplo, como vemos en la siguiente figura, al disminuir la velocidad de infusión de fármaco, disminuye la concentración sitio-efecto y se genera un aumento de los picos por segundo.

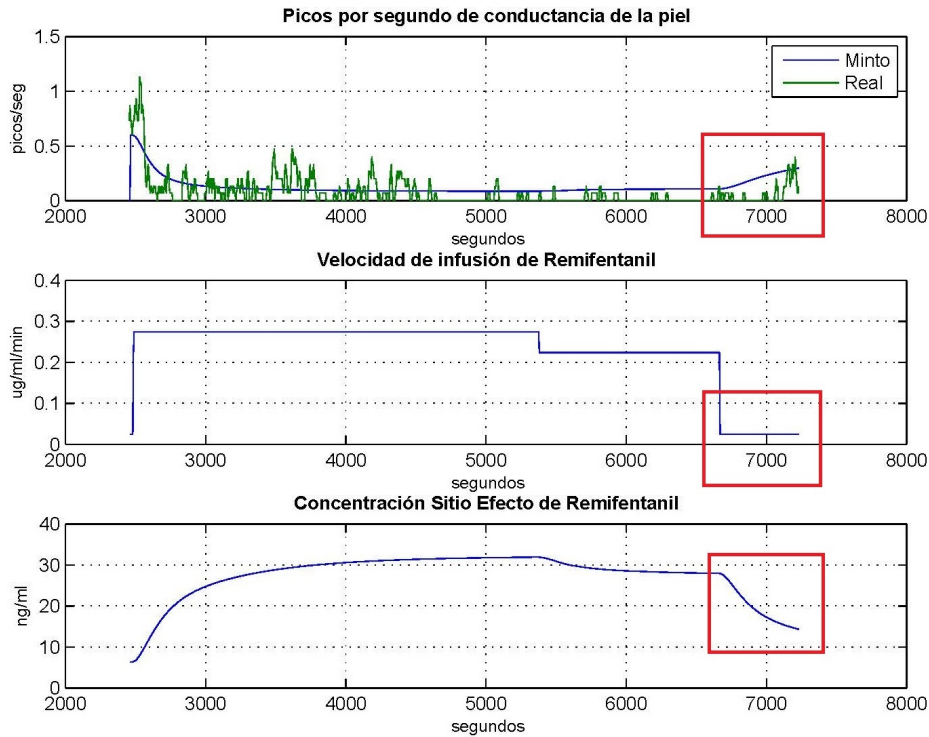


Figura 13: Representación de los cambios en el comportamiento del paciente a partir de la disminución de infusión de Remifentanil ($\mu\text{g}/\text{ml}/\text{min}$)

Conclusiones

Recapitulando lo anteriormente expuesto podemos decir que las inferencias son las siguientes:

- El modelo de Minto puede emplearse con el fin de lograr una reproducción de la variación de picos por segundo, es decir, es un modelo válido para conseguir una velocidad de infusión de Remifentanil apropiada para el paciente.
- El modelado inverso nos proporciona un modelado más fidedigno de la dinámica que presenta el paciente. Además, con este tipo de modelado podemos calcular una ley de control que nos ayude a obtener una respuesta que se amolde al paciente de manera eficaz.
- Se prevé una respuesta que se ajusta mejor al modelo de paciente si se coloca:
 - $EC_{50} = 11 \text{ ng/ml}$
 - $\gamma = 2$
- Los parámetros farmacodinámicos obtenidos gráficamente se asemejan a los hallados en otras investigaciones ya mencionadas.

Área bajo la curva

Otro aspecto importante en el estudio, dado que es calibrable mediante el empleo del algesiómetro, es el área bajo la curva, que se define como un parámetro capaz de indicarnos cuánto de próximo está el paciente a responder al estímulo.

En la ilustración 14, divisamos que el área bajo la curva correspondiente al generado por el modelo de Minto (“ABC de conductancia de la piel del modelo de Minto”) toma un valor de 2.4, esto nos induce a

pensar que el paciente necesita de más analgésicos e hipnóticos pero en una cuantía baja, ya que sigue estando bastante lejos de despertar.

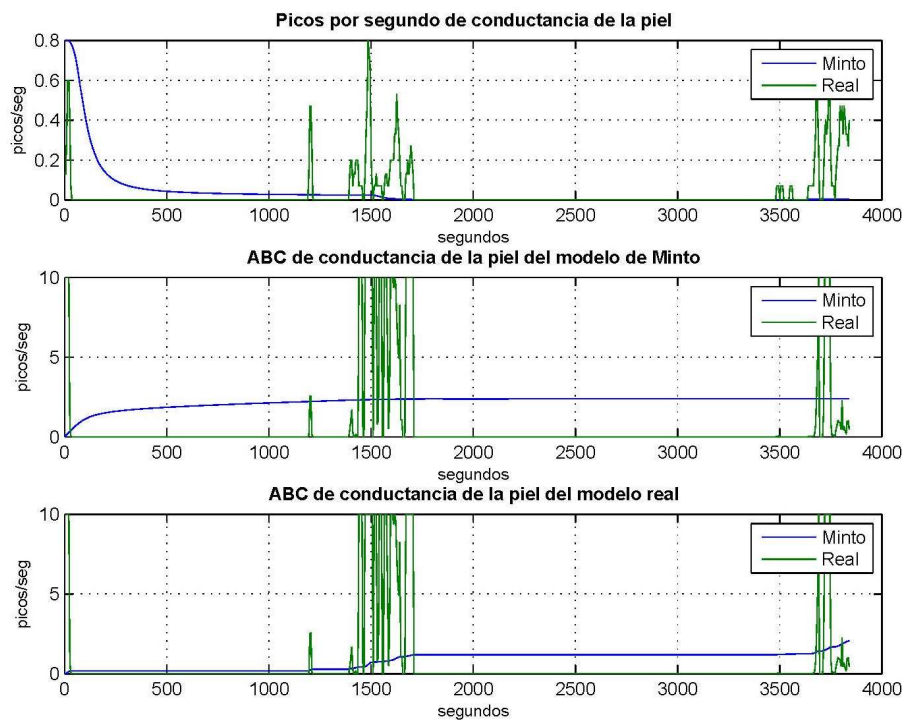


Figura 14: Variación de picos segundo y de área bajo la curva de un paciente real con respecto al tiempo

En este mismo orden, se ha generado el área bajo la curva de la variación de picos por segundo reales, dándonos como resultado el recuadro número tres.

En la siguiente imagen se puede observar con mejor detalle que efectivamente se obtiene el área bajo la curva del paciente real.

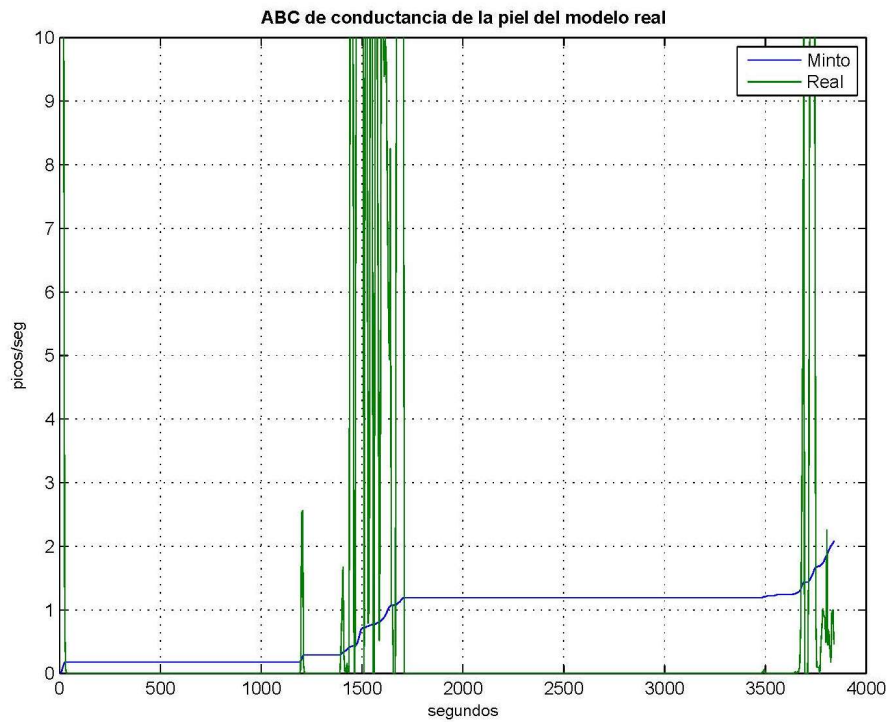


Figura 15: Área bajo la curva de conductancia de piel de un paciente con las variaciones de picos/seg reales en función del tiempo (s)

Por lo que se observa se puede decir que a partir del área bajo la curva podemos obtener la suma de las variaciones reales de picos por segundo con bastante precisión de un paciente, con lo cual puede emplearse como medio para obtener un modelado y/o control de la analgesia. Queda, de esta forma, como una línea abierta para un trabajo posterior.

4. Controlador

4.1. Q-learning

Finalizado el modelado procedemos a proponer un controlador que nos permita infundir fármaco según el comportamiento que predecimos que tendrá el paciente.

Para ello hemos de tener en cuenta dos factores:

1. La vida media efectiva del fármaco tras la administración de la dosis, que en este caso es de entre 3 y 10 minutos, aproximadamente.
2. La velocidad de infusión continua de Remifentanyl (junto con hipnótico o agente volátil), oscilará entre 0.25-0.4 $\mu\text{g}/\text{kg}/\text{min}$ para una persona adulta.

Teniendo esto en cuenta, procedemos a explicar el controlador escogido.

Es un algoritmo que aprende por refuerzo, es decir, consiste en probar a realizar diversos movimientos y observar el refuerzo que proporcionan. Para ello, el algoritmo memoriza los movimientos y las recompensas adyacentes a estos movimientos. Por lo tanto, con este controlador lo que se pretende es mezclar recompensas directas (a corto plazo y que implican llegar al objetivo) con recompensas a largo plazo (se pesan las acciones que nos acercan o nos alejan del estado deseado).

Definición formal del Q-learning

$$Q[s, a] \leftarrow \alpha \cdot (r + \gamma \max_{a'} Q[s', a']) - Q[s, a]$$

O equivalentemente:

$$Q[s, a] \leftarrow (1 - \alpha) \cdot Q[s, a] + \alpha(r + \gamma \max_{a'} Q[s', a'])$$

Donde:

$r(s, a)$ = recompensa directa o inmediata

γ = factor de descuento

α = velocidad de aprendizaje

s' = estado siguiente después de una acción a

a, a' = Acciones en los estados s y s' , respectivamente

A continuación explicaremos conceptos con profundidad que son clave a la hora de implementar este controlador para la finalidad que se pretende.

Recompensa a lo largo plazo

La recompensa a largo plazo es la recompensa que esperamos acumular a la larga si en cada estado realizamos la mejor acción posible.

Experiencia

Una experiencia incluye un estado, la acción realizada, la recompensa recibida, y como novedad, el estado siguiente tras realizar esta acción.

Se emplea el estado siguiente y la recompensa directa que se le otorga a ese nuevo estado como método para conocer si la acción tomada del estado del que se parte nos ayuda a acercarnos al objetivo, por ende

se puede decir que utilizaremos esa recompensa directa en el estado siguiente para aprender algo sobre la recompensa a largo plazo asociada al estado del que se parte y la acción tomada.

A medida que se aprende de las distintas experiencias se va otorgando recompensas más precisas.

Por ejemplo, vamos a suponer que intentamos implementar este algoritmo en el juego “tres en raya”.

Supongamos que conocemos los movimientos ganadores de una partida, como por ejemplo, este caso:

$$r\left(\begin{array}{|c|c|c|} \hline x & o & o \\ \hline x & & \\ \hline | & | & | \\ \hline \end{array}, \begin{array}{|c|c|c|} \hline x & o & o \\ \hline x & & \\ \hline | & | & x \\ \hline \end{array}\right) = 1$$

Figura 16: Recompensa directa para un movimiento ganador

Al ganar la partida, empezando desde un estado concreto del tablero y realizando una acción determinada, se produce un refuerzo positivo, como se observa en la ilustración 16. Además, para ese estado, coincide con que es la única acción que produce un refuerzo positivo, por tanto, es la mejor acción posible.

Ahora vamos a analizar la jugada anterior:

$$r\left(\begin{array}{|c|c|c|} \hline x & o & \\ \hline | & | & | \\ \hline | & | & | \\ \hline \end{array}, \begin{array}{|c|c|c|} \hline x & o & \\ \hline | & | & | \\ \hline | & | & x \\ \hline \end{array}\right) = 0$$

Figura 17: Recompensa directa a un estado intermedio

Si partimos del estado de la izquierda, realizando la acción en rojo, no recibimos recompensa directa pues se trata de un estado intermedio, pero sí recompensa a largo plazo ya que como hemos visto, esa acción nos acerca a ganar la partida.

Por consiguiente, ahora que tenemos dos estados consecutivos podemos comprender mejor el concepto y la relevancia, de la experiencia.

$$\left(\begin{array}{|c|c|c|} \hline x & o & \\ \hline | & | & | \\ \hline | & | & | \\ \hline \end{array}, \begin{array}{|c|c|c|} \hline x & o & \\ \hline | & | & | \\ \hline | & | & x \\ \hline \end{array}, 0, \begin{array}{|c|c|c|} \hline x & o & o \\ \hline x & & \\ \hline | & | & | \\ \hline \end{array}\right)$$

Figura 18: Experiencia (Estado, acción, recompensa, estado siguiente)

De igual modo, el algoritmo necesita dos parámetros que debemos ajustar en función del problema que estemos resolviendo:

Velocidad de aprendizaje(learning rate)

Es un valor entre cero y uno que indica cuánto podemos aprender de cada experiencia. Cero significa que no aprendemos nada de una nueva experiencia, premia mejor lo que ya conocemos, y uno significa que olvidamos todo lo que sabíamos hasta ahora y nos fiamos completamente de la nueva experiencia.

Factor de descuento(discount factor)

Es también un valor entre cero y uno e indica cuán importante es el largo plazo. Cero significa que sólo nos importan los refuerzos inmediatos, y uno significa que los refuerzos inmediatos no importan, sólo

importa el largo plazo. Este factor nos ayuda a mezclar recompensas directas con recompensas a largo plazo y producir la recompensa mixta.

En ambos parámetros los extremos son poco útiles. La velocidad de aprendizaje se puede ajustar en función de la incertidumbre respecto a los estados siguientes en las experiencias y el factor de descuento establece un balance entre el refuerzo inmediato y el refuerzo a largo plazo por lo que si se reciben refuerzos intermedios se debe decidir cómo ponderar.

En otro orden de cosas, es importante recalcar que se debe dejar de explorar cuando se estabilizan los valores de la matriz en vista a que ésta ya no aprende nada nuevo.

4.2. Funcionamiento del algoritmo en el caso a estudiar

Objetivo

El objetivo del algoritmo Q-Learning es intentar resolver el problema de cómo se debe actualizar la función de evaluación para que el agente pueda aprender cuál es la función óptima en un entorno no determinista.

Para nuestra investigación, el fin es obtener una velocidad de infusión de fármaco según el estado en el que se esté y al que nos lleve la acción escogida. En definitiva, se pretende enseñar al algoritmo cuál es la velocidad que mejor se adecua para cada posible estado en el que nos podamos encontrar, y que además nos aproxime al objetivo.

Para ello, el estudio se divide en dos fases:

Offline o entrenamiento previo para obtener en cada caso las velocidades de infusión de Remifentanil adecuadas, y Online, consecución del proceso ya una vez obtenidas las medidas de la evolución del paciente, en la fase anterior.

1ª Fase:

Offline

Con este proceso se quiere entrenar al algoritmo a partir de la observación de la respuesta durante un periodo de 2.5 minutos. En consecuencia se pretende tomar la mejor acción a raíz de la predicción del estado.

En primer lugar, se establecen los estados y las acciones de la matriz Q-learning. Las filas corresponderán a los estados y las acciones a las columnas. Cada elemento de esta matriz recibirá una recompensa para una combinación (estado, acción).

En nuestro caso particular, las acciones representarán a la velocidad de infusión y los estados a los picos por segundo. El rango para las acciones oscila entre 0 y 0.3 $\mu g/kg/min$ y para los estados fluctúa entre 0 y 0.7 picos/seg.

Posteriormente se delimitan los valores que tendrán γ y α :

Por una parte, el factor de descuento, γ , posee el valor de 0.3, cercano a 0, ya que nos interesa que se valore más que se ha llegado al objetivo que cómo se ha llegado a él, es decir, se premia mejor la recompensa a corto plazo que la recompensa a largo plazo. Esto se hace porque independientemente del estado en el que nos encontremos, nuestra principal preocupación es que el paciente sienta el mínimo grado de dolor posible.

Por otra parte, la velocidad de aprendizaje, α , se limitó al valor 0.2 ya que se pretende dar una gran importancia a la recompensa recién adquirida, esto implica que el aprendizaje será mucho más lento y

por consiguiente, tiene sentido que sea necesario realizar bastantes iteraciones puesto que hasta que este valor no es alto, no comienza a aprender.

Se fija por esto el número de iteraciones a 1000, ya que se pretende evaluar las suficientes veces todos los estados de manera que analizamos todos los posibles casos que se puedan dar a la hora de la exploración.

Podemos decir de esta forma, que la matriz generada será lenta pero fiable. Como veremos al final de este análisis será trascendental adecuar estos valores ya que repercute de manera vital en la evolución que presenta el paciente en una cirugía.

Comenzamos la exploración partiendo del máximo estado, que es el correspondiente a 0.7 picos/seg y de una matriz Q inicial, que como vemos posteriormente, es una matriz de ceros:

$$Q = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Empieza la exploración y aplicamos una acción que puede ser, o correspondiente al máximo valor de la fila, por ejemplo: si estamos en el estado seis (0.5 picos/seg) y el máximo valor se encuentra en la segunda columna, le aplicaremos la acción 0.05; o respectivo a una acción que se generara de manera aleatoria, este caso se dará cuando al recorrer la fila, el algoritmo se encuentre con que los valores son cero y no sepa qué acción otorgar.

Con lo cual, de a partir de un estado, en este caso el estado 0.7, se aplica una acción sopesando los valores que haya.

Para asegurarnos de que escoge una de las acciones anteriormente definidas, buscamos el identificador que corresponda a la acción escogida.

Por contraste, para conceder un valor de picos por segundo es necesario remitirnos al “Simula_bolo_Inverso” que es donde se encuentra la simulación de un paciente modelo.

Simulación de un bolo de Remifentanil

En este documento se calculan a partir de los datos del paciente, las ecuaciones de la farmacodinámica y la farmacocinética del modelo de Minto. Además a partir del “Inverse_Model” obtenemos la concentración sitio efecto, que comenzaremos a administrar para 0.60 picos/seg, como ya hemos mencionado con anterioridad. En este aspecto, es importante recordar que al suponer un estado de equilibrio la concentración en los compartimentos es la misma. En contraste se empleará como entrada, la concentración efecto calculada para los picos segundos supuestos.

Como podemos comprobar en la siguiente imagen (ilustración 19), a lo largo del tiempo los compartimentos tienden a tener las mismas concentraciones al suponerse en equilibrio.

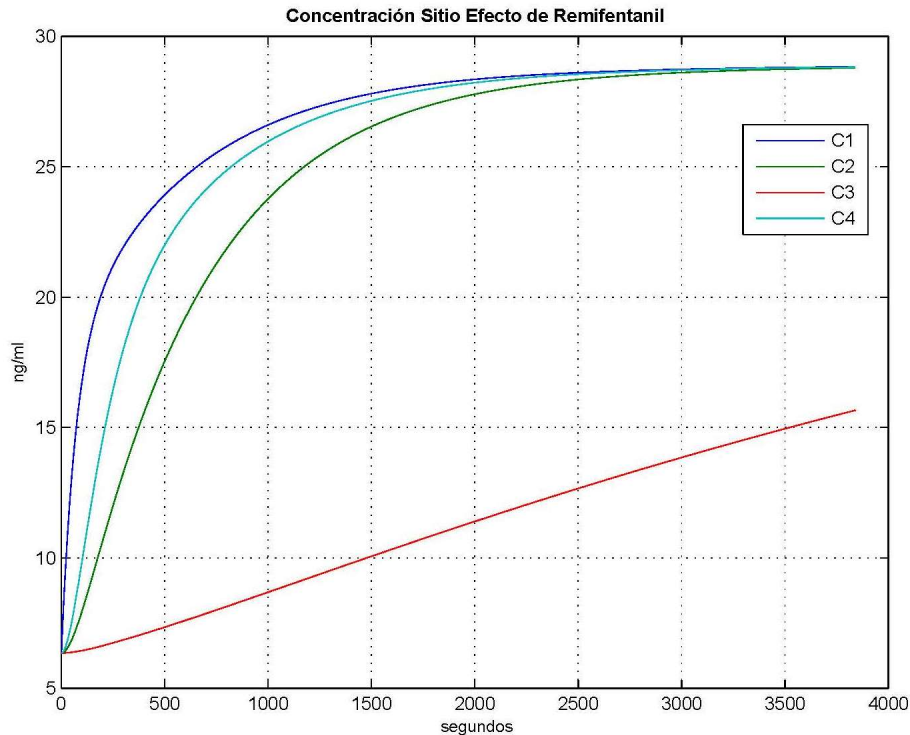


Figura 19: Dinámica de los compartimentos central (C1), periférico rápido (C2), periférico lento (C3) y compartimento efecto (Ce) con respecto al tiempo (s)

A partir de la concentración sitio efecto, impartida para 0.60 picos/seg, procedemos a enseñar al algoritmo, durante 2.5 minutos, la respuesta que desarrolla un paciente según el modelo de Minto.

Se halla la variación de picos por segundo, a raíz del cálculo de las concentraciones en cada uno de los compartimentos. Como lo que nos interesa es calcular el estado que va a poseer el paciente en el siguiente instante de manera fiable y que además nos permita observar un cambio en la dinámica, escogemos el último valor de picos por segundo.

Siguiente estado y recompensa

Obtenido este valor, volvemos a la función donde se calcula el siguiente estado y la recompensa. Es decir, nos dirigimos a la función interna que tiene Q-learning para hacer esto.

En esta función, hacemos un redondeo a la baja del pico por segundo hallado, esto es así porque si lo hiciéramos al alza no se contemplaría el estado cero por ejemplo, para 0.05 picos/segundos, los picos redondeados al alza serían de 0.1 en cambio que a la baja serían 0.

Una vez hecho esto, se asignará el pico por segundo siguiente y con él, la recompensa asociada.

Para otorgar la recompensa hemos considerado tener dos factores en cuenta. El primero de ellos es la variación de picos por segundo; aunque nuestro objetivo se centra en obtener un valor muy pequeño de picos por segundo, no se puede recompensar igual a aquella acción que nos aleje de un estado muy alto de picos por segundo, a pesar de que no se consiga llegar al objetivo, que a una acción que apenas cambie el estado actual.

El segundo factor a tener en cuenta es el propio valor que toma la variable picos por segundo, cuyo objetivo es que tome valores pequeños.

Por este motivo, se realiza una ponderación teniendo en cuenta que es más importante el cambio de estado que el estado siguiente en sí. De esta manera, multiplicamos por un factor de 0.8 y 0.2 a la variación y al estado siguiente, respectivamente.

Para limitar las recompensas, nos fijamos en la distribución de esta variable (Ilustración 20):

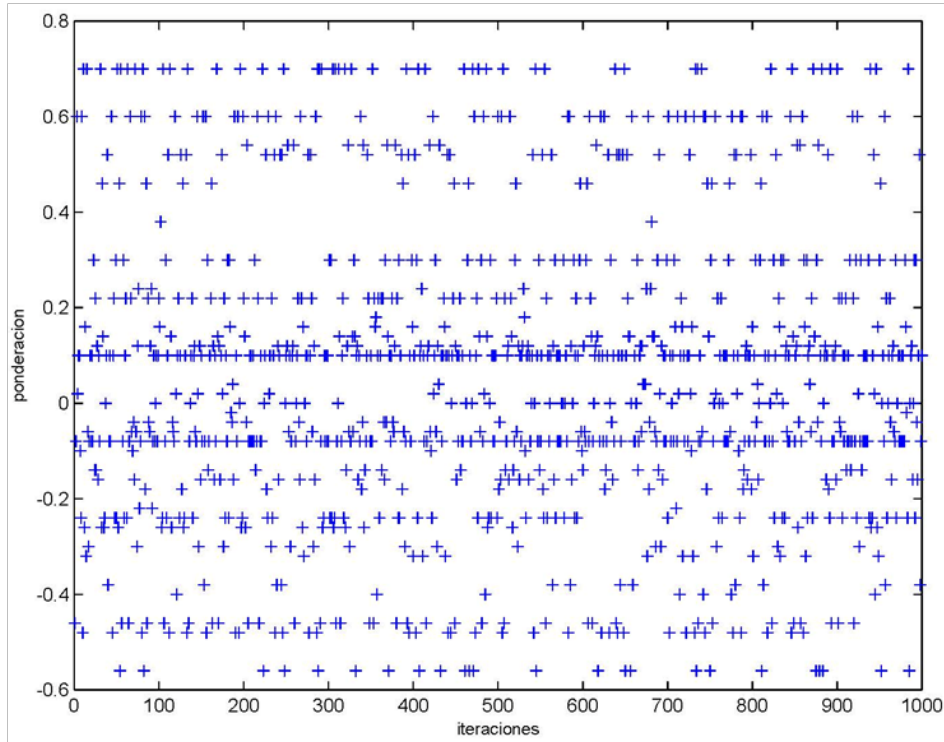


Figura 20: Distribución de los valores de distribución con respecto a las 1000 iteraciones

Hay que contemplar un caso particular a la hora de premiar. Se trata de cuando se llega al objetivo o se está muy próximo a él (valores de 0 o 0.1 picos/segundos) pues es un caso ambiguo a la hora de otorgar la recompensa. Esto es debido a que se debe premiar diferente, el llegar a este estado sea cual fuere la variación de picos por segundo, pues éste es el objetivo en sí. Se ha optado, por lo tanto por tomar una posición conservadora con respecto a esta cuestión dándole una recompensa media a este caso.

Finalmente, tenemos una ecuación Q-learning, $[Q[s, a] \leftarrow \alpha \cdot (r + \gamma \max'_a Q[s', a'] - Q[s, a])]$, en la cual está presente implícitamente tanto las variaciones de picos por segundo, como el pico por segundo conseguido una vez realizada la acción.

Función Q-learning

Ya conocemos la recompensa que se lo otorga por llegar a un estado siguiente concreto, pero no la manera matemática que tiene este controlador para calcular el valor que le corresponde a cada elemento de la matriz.

Por ello, se explicará de manera específica y mediante un ejemplo, cómo opera la matriz Q-learning:

Por definición, los valores que tome la matriz Q-learning vendrán dado por:

$$Q[s, a] \leftarrow (1 - \alpha) \cdot Q[s, a] + \alpha(r + \gamma \max'_a Q[s', a'])$$

Si, empezamos del estado 0.7 como anteriormente habíamos dicho, y se le aplica una acción 0.3, la recompensa valdrá 80 ya que el siguiente estado será 0.1, como vemos a continuación:

```

>> qlearning
current state : 0.7 next state : 0.1 taken action : 0.2
next reward : 80
  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   0   0   0   0   0
  0   0   0   0   0   0
  0   0   0   0   0   0
  0   0   0   16  0   0

```

Figura 21: Command Window de la primera iteración correspondiente a la matriz Q

Esto es, porque si aplicamos la fórmula anteriormente mencionada tenemos que:

$$Q[s, a] \leftarrow (1 - \alpha) \cdot Q[s, a] + \alpha(r + \gamma \max_{a'} Q[s', a'])$$

$$Q[s, a] = 0 + 0,3 \cdot (80 + 0,2 \cdot 0 - 0) = 16$$

Con las iteraciones el algoritmo evaluará todos los estados y las acciones de forma que los valores pertenecientes a la matriz irán cambiando con la exploración. Llegará a un punto que aunque le coloquemos un valor alto de iteración la matriz no cambie y llegue a valores estacionarios.

```

>> Q
Q =
 30.1699  30.1901  44.3321  71.4034  71.4104  71.4285
 29.5078  44.3238  40.0043  71.3431  71.4070  71.1265
 20.7022  21.9246  34.1230  67.1410  58.8106  72.6127
 29.2709  46.4859  37.4994  79.0809  81.3032  80.9919
 26.3786  15.2921  23.4135  37.4888  59.7306  93.7088
 15.7148  29.8910  60.2139  86.4899  68.1913  36.5133
 45.1491  28.4416  65.7520  79.4645  100.9662  97.6013
 48.2972  41.3923  83.1488  101.0082  100.6231  100.7515

```

Figura 22: Matriz Q-learning con 1000 iteraciones

Con la matriz Q formada teniendo en cuenta todos estos factores, elegimos los mayores valores de cada fila o estado, para poder analizarlo de manera Online. Estos valores representarán a la velocidad de infusión:

```

>> vel
vel =
 0.3000  0.2500  0.3000  0.2500  0.3000  0.2000  0.2500  0.2000
>>

```

Figura 23: Velocidad de infusión en $\mu g/kg/min$

2ª Fase:

Online

Para ajustar el modelo a un paciente particular, ya que cada paciente posee su propia respuesta a los fármacos, se realiza el procedimiento Online, que nos permite hallar un modelo en función de los datos que recibe del comportamiento real del paciente.

Las velocidades anteriormente expuestas corresponden a cada uno de los estados del camino óptimo que elige la matriz Q .

Es natural, cuestionarnos los valores de velocidad de infusión que elige el algoritmo para cada estado, por ejemplo, cuando los picos por segundo son cero se elige $0.3 \mu\text{g}/\text{kg}/\text{min}$ como acción apropiada y sin embargo, para cuando los picos por segundo son 0.7 se selecciona $0.2 \mu\text{g}/\text{kg}/\text{min}$ como acción óptima. Esto se explica porque al algoritmo le es más fácil otorgar una acción menor a un estado como el de 0.7 picos por segundo, y llevarnos a otro estado en donde sea menos costoso llegar al objetivo, que hacerlo directamente. Además elegir una acción tan alta para cuando se llega al objetivo ($0.3 \mu\text{g}/\text{kg}/\text{min}$ para 0 picos/seg) se debe a que es tan importante conseguir el objetivo como mantenerse en él.

Para asociar las velocidades con el estado correspondiente empleamos el identificador del estado, de tal forma que si los picos por segundo son 0.55 aplicamos una aproximación al alza y le otorgamos el identificador correspondiente. El motivo por el cual ha sido una aproximación al alza, se debe a que nos debemos colocar en la peor situación, pues es preferible suponer que el paciente está inmerso en un estado el cual le supone un mayor grado de dolor que lo contrario. Una vez hecho esto, se asocia al identificador del estado con la velocidad que le compete.

En la siguiente figura se muestra finalmente el diagrama de bloques que se propone:

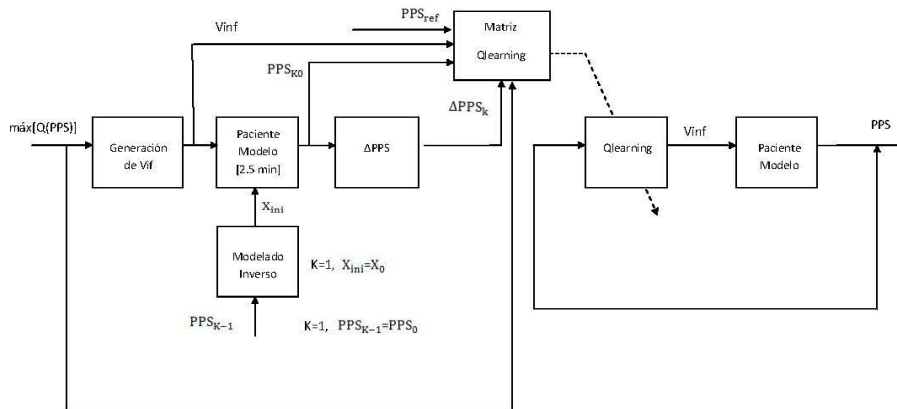


Figura 24: Estructura del controlador en forma de diagrama

Finalmente hemos simulado diversos cambios bruscos que perturban el estado en el que se sume el paciente y que se pueden dar en una cirugía real. A partir de la generación de estos picos por segundo, que alteran el estado del paciente, se calculan las concentraciones compartimentales y en última instancia, la concentración sitio-efecto que corresponderá a la entrada re-alimentada de nuestro sistema.

Los eventos han sido los siguientes:

```

% Bucle
for i=2:length(muestras)
    idx=min( ceil(picos_seg(i-1)*10) , 8);
    u=vel(idx);
    uvector(i-1)=u;
    x_r(:,i)=A*x_r(:,i-1)+B*(u)*WT/V1/60; % Le pasamos la velocidad de infusión total de remifentanil, en ug/ml/min
    picos_seg(i)=picos_seg_0+((picos_seg_min-picos_seg_0)*(x_r(4,i)^g))/(x_r(4,i)^g+(ec50/1000)^g);% Obtenemos como salida los picos/seg

    %Alteración del estado del paciente para instantes concretos
    if (i >= 400 && i<=450)
        picos_seg(i)=picos_seg(i)+0.4/51;
        [A,B,Uu,C,D,picos_seg_0,picos_seg_min,ec50,g,V1] = InverseModel(WT,HT,Age,Gender,T,picos_seg(i));
        x_r(:,i)=C/1000;
    end
    if (i >= 600 && i<=650)
        picos_seg(i)=picos_seg(i)+0.6/51;
        [A,B,Uu,C,D,picos_seg_0,picos_seg_min,ec50,g,V1] = InverseModel(WT,HT,Age,Gender,T,picos_seg(i));
        x_r(:,i)=C/1000;
    end
    if (i >= 150 && i<=200)
        picos_seg(i)=picos_seg(i)+0.3/51;
        [A,B,Uu,C,D,picos_seg_0,picos_seg_min,ec50,g,V1] = InverseModel(WT,HT,Age,Gender,T,picos_seg(i));
        x_r(:,i)=C/1000;
    end
end
end
end

```

Figura 25: Código para las perturbaciones provocadas

En la imagen siguiente podemos observar la funcionalidad práctica del controlador propuesto para un intervalo de tiempo concreto:

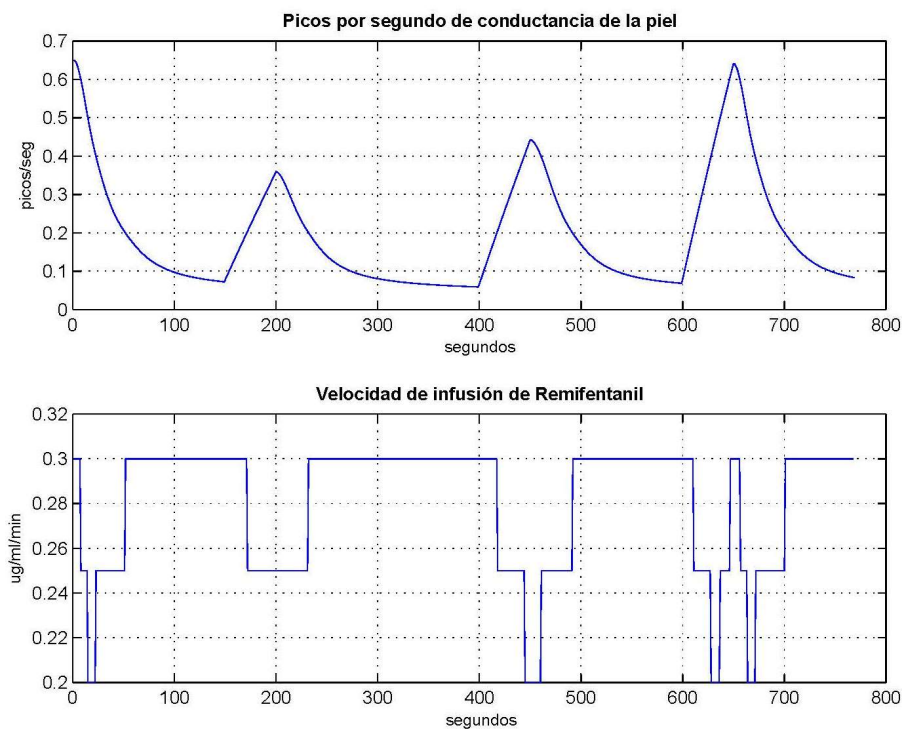


Figura 26: Funcionamiento del controlador ante variaciones de picos por segundo en 13.3 minutos

Podemos visualizar que se representan tres posibles situaciones que pueden acontecer en una cirugía. Todas ellas se caracterizan por poseer un intervalo pequeño de tiempo 50 segundos

1. La primera de ellas, comienza a los 2.5 minutos y se aumentan 0.3 picos por segundo durante 50 segundos. Vemos que para este tramo los picos por segundo disminuyen hasta estar cercanos a los 0.05 picos por segundo, es decir aproximándose al objetivo a lograr.

2. La segunda de ellas, comienza a los 6.7 minutos y se aumentan 0.4 picos por segundo durante el mismo periodo de tiempo anterior. En esta fase se consigue disminuir un poco menos que en el intervalo anterior, pero sigue estando en valores considerados como óptimos ya que está entorno a los 0.07 picos por segundo.
3. La última de ellas, comienza a los 10 minutos y se aumentan 0.6 picos por segundo durando el mismo tiempo que las anteriores etapas. Comprobamos gráficamente que este valor oscila entorno a 0.08 picos por segundo, dando veracidad a que cuánto mayor sean los picos por segundo generados, más difícil le resulta al algoritmo llegar al estado objetivo, para un mismo intervalo de muestreo.

Conclusiones

Finalmente podemos afirmar que ante las variaciones de picos por segundo, el controlador administra la velocidad que estima oportuna para cada caso manteniendo el nivel de picos por segundo entorno a cero, como era nuestro objetivo. De esta manera se consigue dotar al anestesista de una herramienta con la que puede regular mejor el grado de dolor que pueda tener un paciente en una cirugía.

5. Algesiómetro

5.1. Introducción

El algésiómetro “Med-Storm” tiene como objetivo determinar la sensibilidad de un paciente al dolor, analizando los cambios de conductancia que se producen en la piel.

La conductancia de la piel es una variable psicofisiológica que mide la actividad presecretora de las glándulas sudoríparas ecrinas, que principalmente, se localizan en las palmas de las manos y en las plantas de los pies. La importancia de estas glándulas radica en que éstas responden primariamente a la estimulación psíquica, es decir, ante los cambios de emociones. Por otro lado, es importante conocer que estas glándulas reflejan la actividad de la rama simpática del sistema nervioso autónomo, por lo que con ellas podemos contar con un valioso instrumento de medida de dicho sistema nervioso.

Una persona en un estado de relajación no es buena conductora de la electricidad. Sin embargo, cuando ésta está sometida a un estímulo externo o interno, libera acetilcolina incrementando la conductancia de la piel. Esto se debe a que, ante situaciones por ejemplo, de peligro o emergencia, la temperatura corporal aumenta, y ante esto el sistema termostático hipotalámico provoca un descenso de la temperatura emitiendo una respuesta termorreguladora, para ello, eso activa el sistema nervioso simpático que ayuda a que el cuerpo se refrigere, mediante la segregación de las glándulas sudoríparas. La resistencia de la piel disminuye de nuevo, reabsorbiendo éstas glándulas y generando un pico de conductancia, el tamaño del pico depende con cuanta fuerza está tirando el nervio simpático de la piel. El pico de conductancia es específico para el estímulo, el cual induce la respuesta y es evidente pasados 1 o 2 segundos después de la estimulación.

El algésiómetro mide principalmente cambios de la conductancia de la piel en tiempo real para determinar dolor en el paciente. A partir del pico de conductancia de la piel, nos encontramos con dos parámetros muy importantes, los picos por segundo y el área bajo la curva, que pueden ser calculados Online y utilizados para la evaluación del dolor, analizando una ventana de desplazamiento de 15 segundos, que se actualiza cada segundo.

5.2. Obtención de medidas

Metodología e instrumental empleado

La medida de la conductancia de la piel se realiza utilizando tres electrodos autoadhesivos, denominados C (corriente), R (referencia) y M (medida), pegados a la palma de la mano (en adultos) o a la planta del pie (en niños). La central de medida utiliza los electrodos C y R en una configuración en realimentación, aplicando una tensión alterna constante entre los electrodos R y M. La corriente de retorno del electrodo M es registrada, y su valor nos proporciona la información directa sobre la conductancia de la piel. El sistema puede medir valores de la conductancia de la piel dentro del rango 1-200 μs , con un nivel de ruido inferior a 0.002 μs .

La colocación correcta de los electrodos para obtener la medida de la conductancia de la piel se aprecia en la siguiente fotografía:

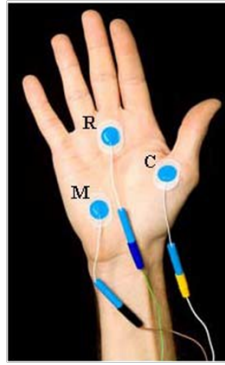


Figura 27: Colocación de los electrodos en la palma de la mano del paciente

Los elementos del sistema se pueden observar en el croquis que se presenta en la siguiente imagen:

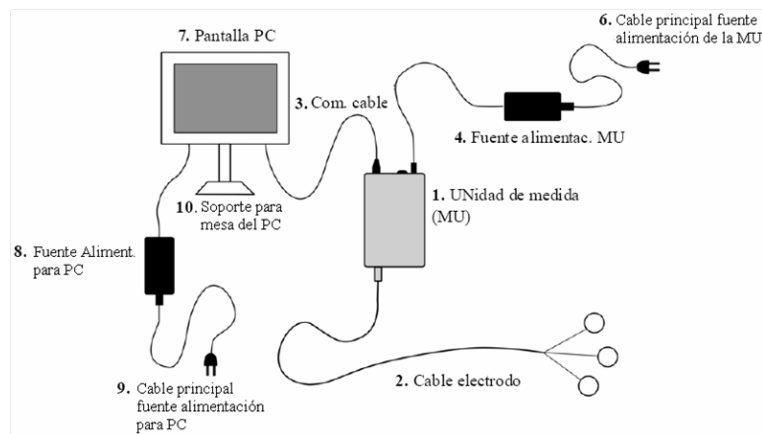


Figura 28: Elementos del monitor de dolor “Med-Storm”

Parámetros calculables

Los parámetros que calcula el programa informático son los siguientes:

- Picos por segundo[Hz]: Es el número de picos de la ventana dividido por el lapso de tiempo de la misma.
- Promedio de Picos[μs]: En un pico, la diferencia entre el valor de la conductancia de la piel de un máximo y un mínimo es el valor de pico. El promedio de picos se calcula haciendo la media aritmética de todos los valores de pico en la ventana de tiempo.
- Tiempo de subida[$\mu s/s$]: El tiempo de subida es la diferencia entre el valor de conductancia al final de la ventana de tiempo y el valor al comienzo de dicha ventana.
- Área grande de picos[$\mu s \cdot s$]: Este parámetro se calcula mediante el establecimiento de una línea base horizontal desde el mínimo del primer pico hasta el final de la ventana de tiempo. El área que se calcula es el sumatorio de la diferencia entre los valores de conductancia en la curva de registro y los de la línea base de referencia, siempre y cuando los primeros sean más grandes que la línea de base.
- Área pequeña de picos[$\mu s \cdot s$]: Este parámetro se calcula estableciendo una línea entre dos puntos mínimos de picos consecutivos. El área que se calcula es el sumatorio de la diferencia entre los

valores de conductancia en la curva de registro y los de la línea, siempre y cuando los valores de esa diferencia sean mayores que cero.

5.3. Implementación del controlador en el algosímetro

Para finalizar este proyecto se va a intentar implementar el controlador a partir de las variaciones en la conductancia de la piel provocadas de manera física por medio del uso del algosímetro “Med-Storm”.

Hacer esta tarea conlleva utilizar y modificar la interfaz hecha por un alumno de la Universidad de la Laguna en un trabajo de fin de grado anterior en la que se comunica el algosímetro con el programa Matlab.

Alteración de la interfaz gráfica

Como queremos representar las variaciones de velocidad que se generan a partir de provocar cambios en la conductancia en la piel, cogemos el último valor real de picos por segundo y calculamos la acción a tomar según el valor que adopte el estado. Como en el trabajo de fin de grado, ya nombrado, se almacenan los 30 primeros valores de conductancia de piel, se adapta el vector de velocidad para que genere la acción tantas veces como valores tengamos. Cabe hacer referencia a que el valor de velocidad que posee cada una de los estados es fruto del entrenamiento previo hecho mediante el controlador Q-learning.

En la siguiente imagen se muestra parte del código añadido a la interfaz:

```

% CÁLCULO Y REPRESENTACIÓN DE LOS PARÁMETROS CARACTERÍSTICOS EN TIEMPO REAL
% Llamamos a la función "parámetros" para obtener los parámetros de la señal de conductancia
[picos_seg, promedio_picos, tiempo_subida, area_grande, area_pequena, area_bajo_curva, picos] = Parametros (datos_muestras, ini_ven, tam_vent)

pics_seg = picos_seg(end);
idx = min(ceil(pics_seg*10), 7);
u = vel(idx+1);
uvector = [uvector; u*ones(30,1)];
% MOSTRARLO A TIEMPO REAL
% REPRESENTACIÓN VELOCIDAD
plot (grafica2, uvector, 'b')
title (grafica2, 'Velocidad de infusión de Remifentanil');
ylabel (grafica2, 'velocidad ng/ml/min');
marcas_tiempo (t, grafica2, uvector)

```

Figura 29: Parte del código añadido a la interfaz

En la siguiente fotografía podemos observar su implementación:

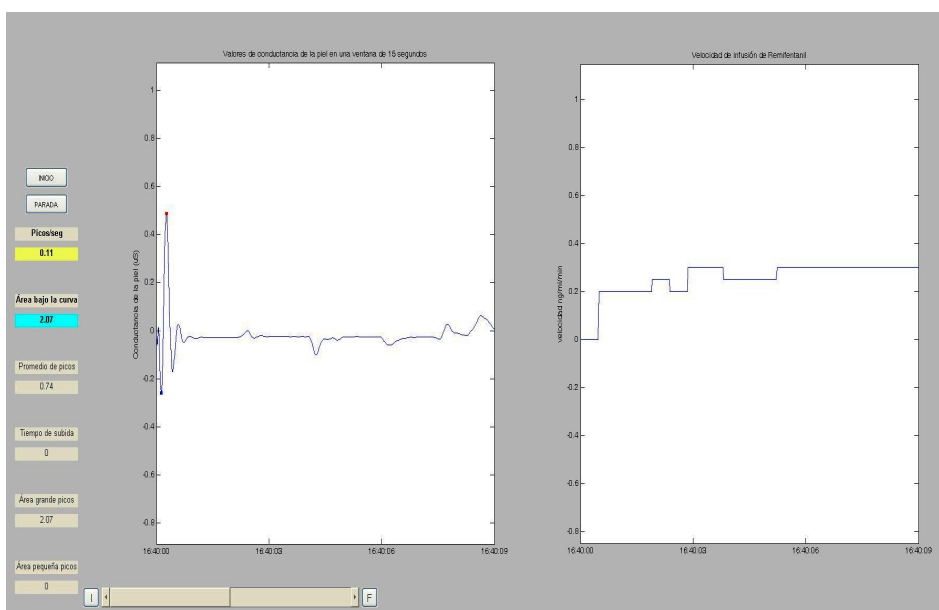


Figura 30: Implementación del controlador en la interfaz de comunicación

Conclusiones

En definitiva, se ha conseguido obtener un control de la dinámica física registrada acorde a los cambios de picos por segundo generados en tiempo real tal y como se muestra en la imagen. Si nos fijamos detenidamente podemos comprobar que para 0.1 picos/seg se aplica la acción $0.3 \mu\text{g}/\text{kg}/\text{min}$, lo cual da fiabilidad a que el algoritmo Q-learning está correctamente implementado en la interfaz que comunica el algesiómetro con el programa Matlab.

Podemos resumir, en los siguientes puntos los objetivos alcanzados con este trabajo:

- En primera instancia se ha conseguido demostrar mediante el empleo de datos de pacientes reales que tanto la edad como la masa muscular magra (LBM) son covariables de los parámetros farmacocinéticos y que concretamente se debe tener en cuenta la masa muscular magra a la hora de obtener el volumen compartimental central (V_1), el volumen compartimental periférico rápido (V_2) y el aclaramiento del compartimento central (Cl_1).
- Se ha caracterizado un modelo de paciente a partir de la obtención gráfica de los valores de la concentración sitio-efecto para la que el efecto es la mitad del valor máximo (EC_{50}) y del parámetro gamma (γ). En este aspecto se ha implementado la dinámica inversa del Modelo de Minto con la que se ha obtenido una respuesta que se ajusta mejor al comportamiento que presenta un paciente sometido a una intervención quirúrgica.
- Se ha logrado desarrollar una técnica que nos permite regular el estado analgésico del paciente a partir del hallazgo de una velocidad de infusión de fármaco que se adecua al estado analgésico en el que se sume el paciente en una cirugía.
- Por último se ha implementado el algoritmo en una plataforma software de control que se comunica con el algesiómetro, un monitor de dolor de paciente mediante el cual se obtienen variables de interés con las que se determina el estado analgésico del paciente, con la finalidad de demostrar su utilidad práctica.

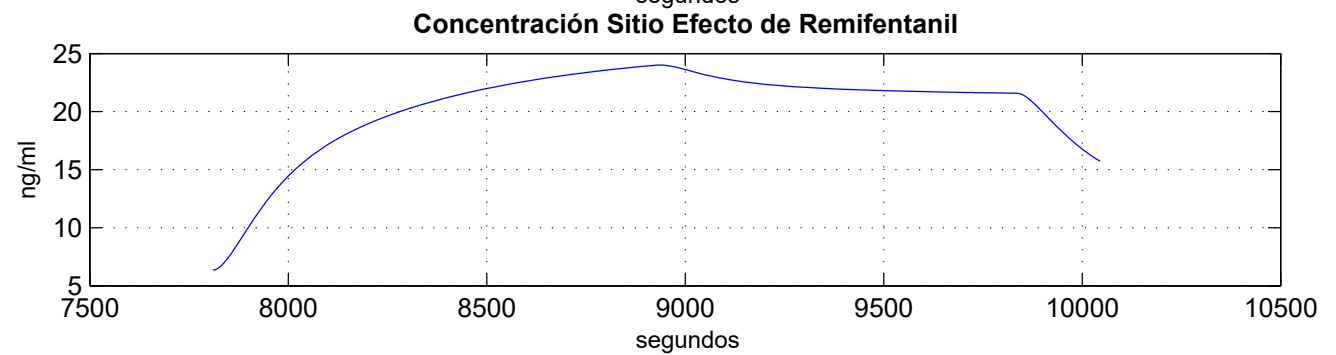
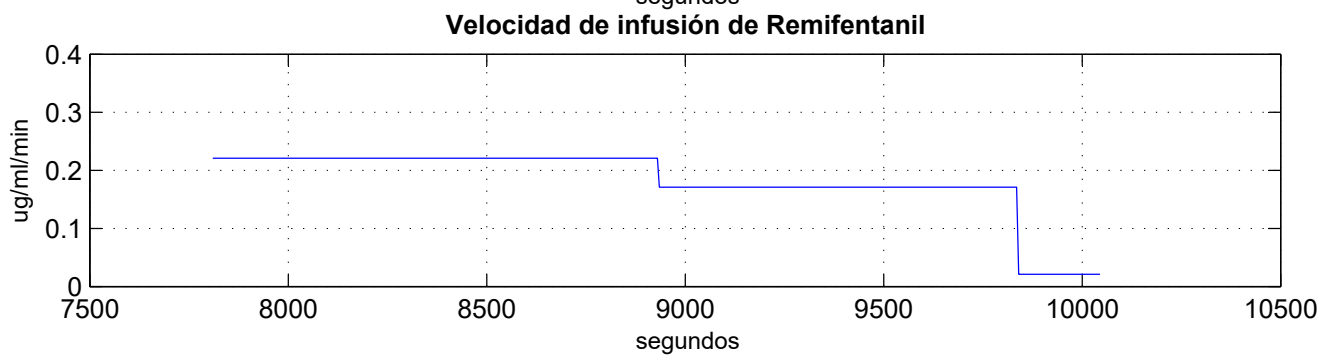
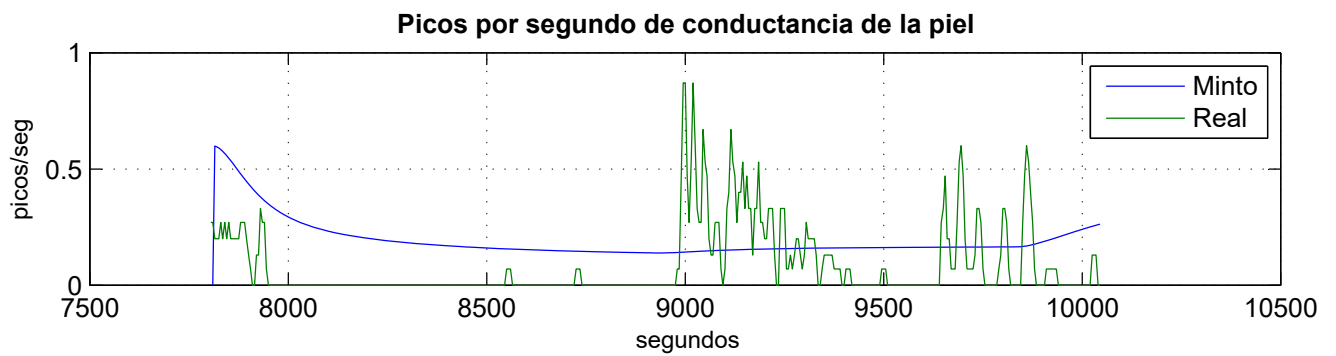
Conclusions

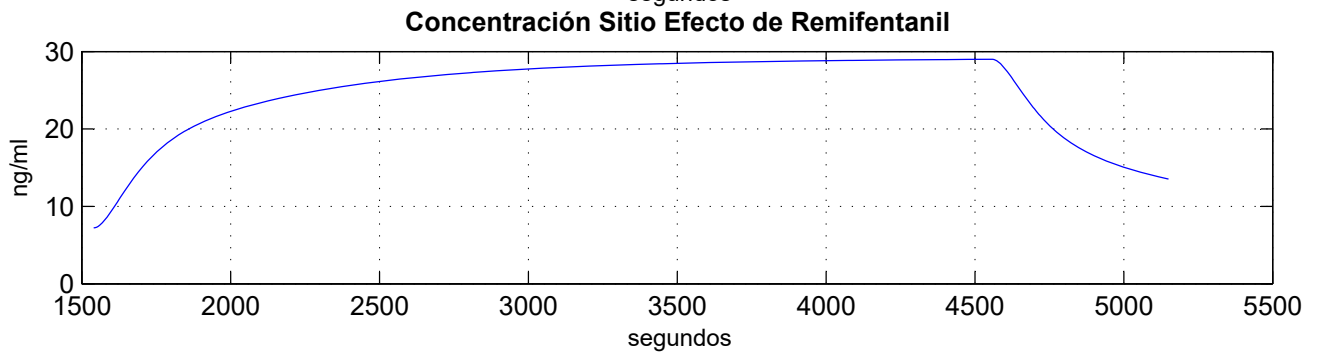
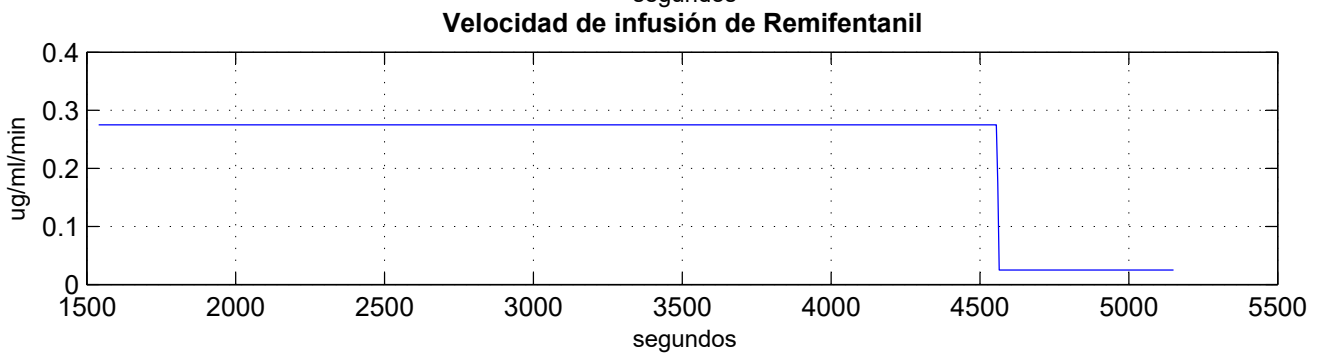
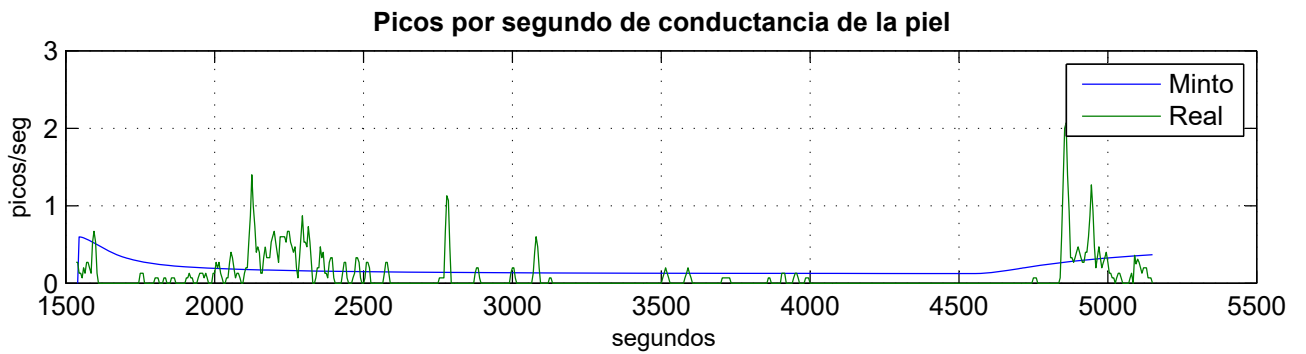
We can conclude with the following objectives achieved in this work:

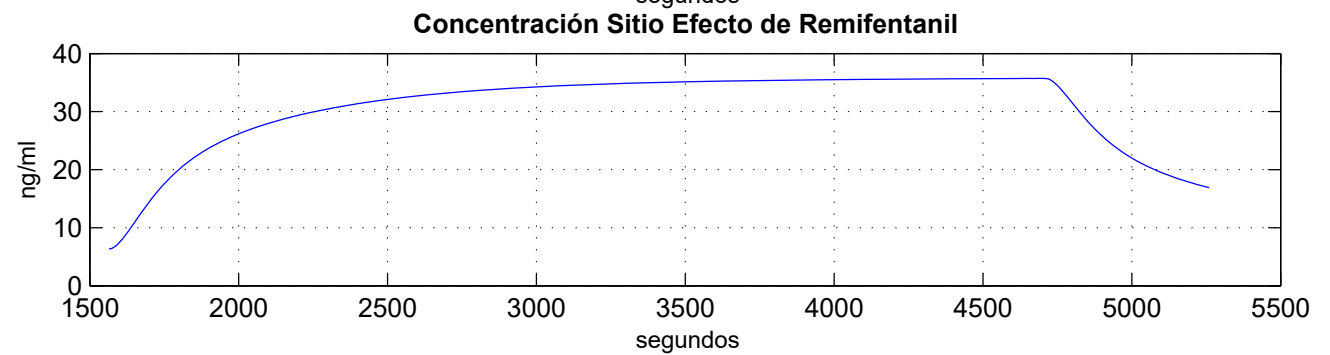
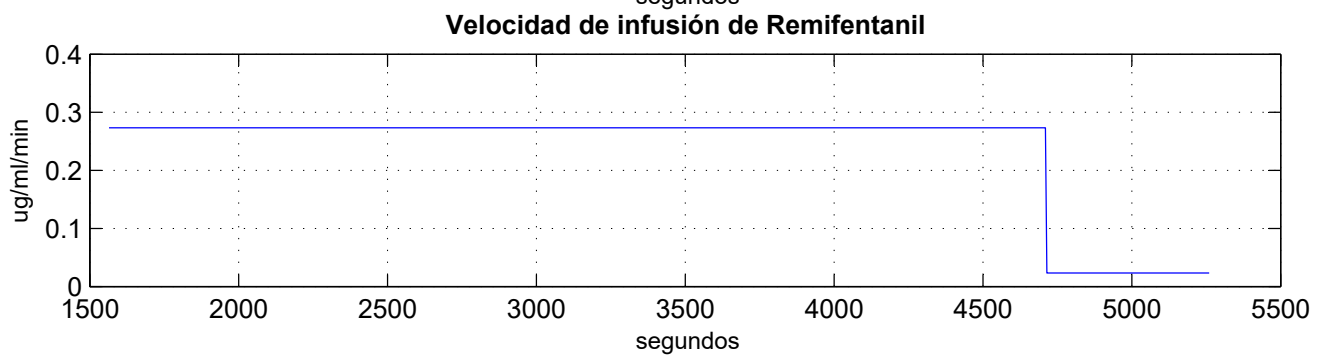
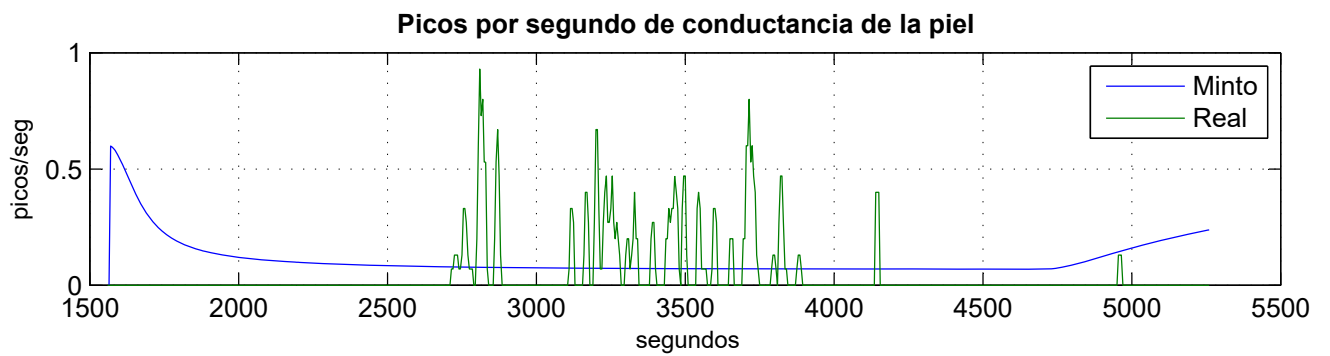
- In the first instance, it has been possible to demonstrate, by using data from real patients, both age and lean body mass (LBM) are covariates of the pharmacokinetic parameters. Specifically, LBM should be considered when obtaining the central compartment volume, the fast peripheral compartment and the clearance of the central compartment.
- It has been characterized a patient model obtaining the values of concentration and effect site with the effect of half the maximum value of concentration (EC_{50}) and the gamma parameter. In this aspect it has been implemented inverse dynamic of Minto Model. We obtained a response that is the best fit for behaviour that presents a patient during the surgery.
- It has developed a technique that allows us to regulate the analgesic state from finding an infusion rate of drug that fits the patient analgesic state during the surgery operation.
- Finally we have implemented the algorithm in a software control platform that communicates with the algesiometer in order to demonstrate its practical utility.

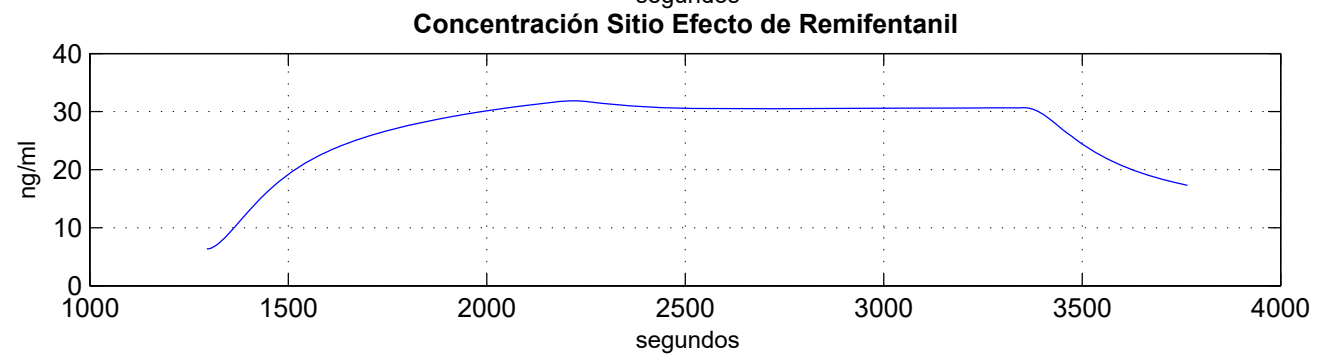
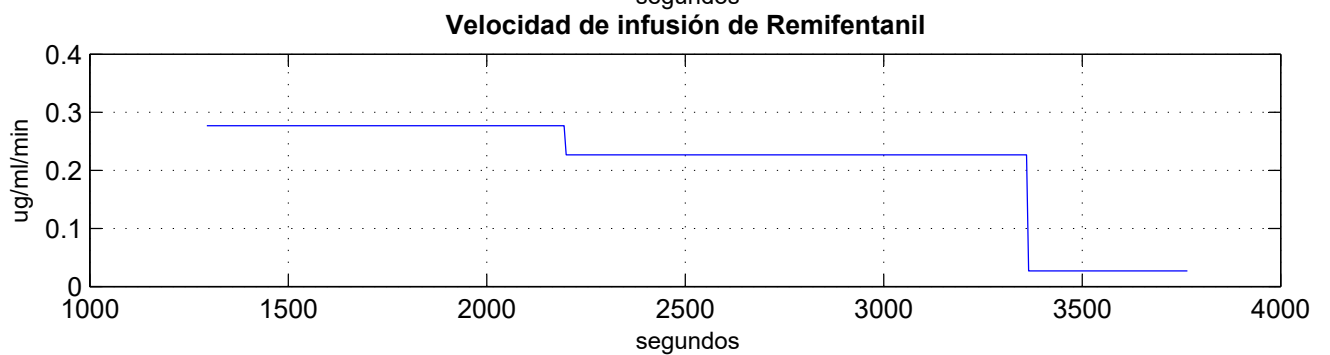
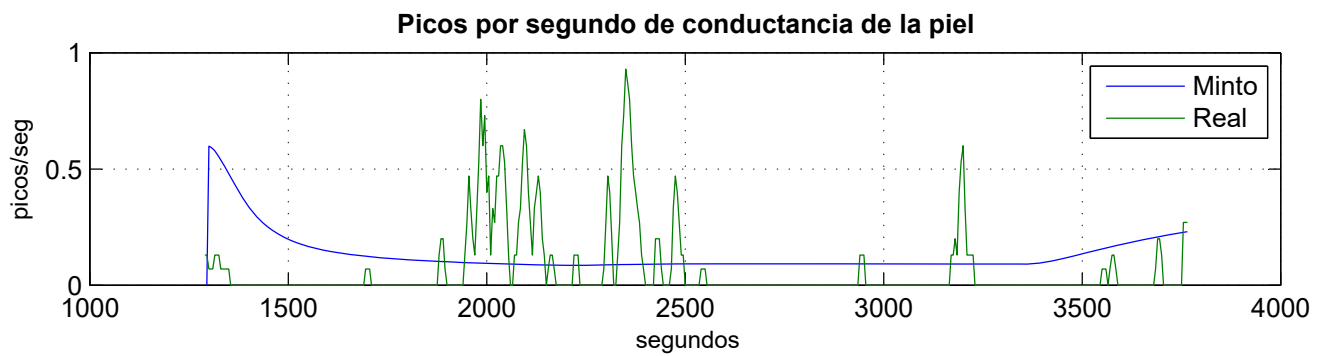
6. Anexos

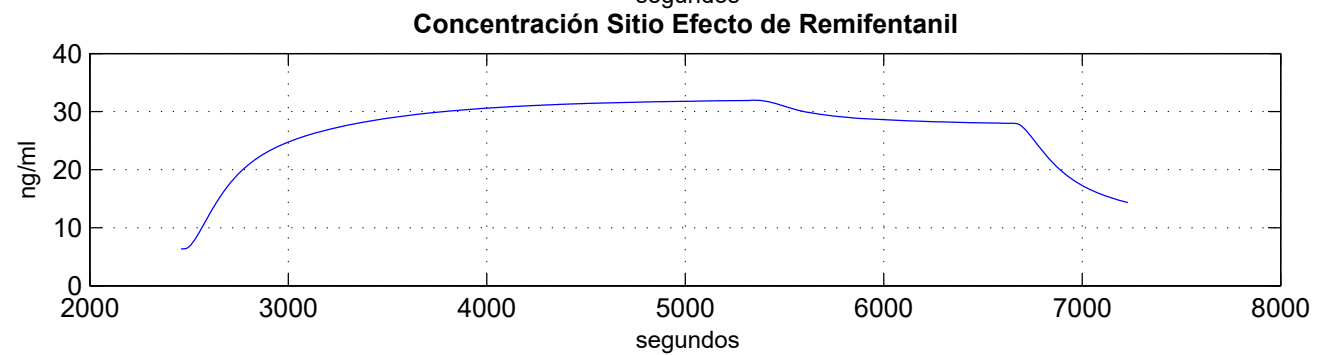
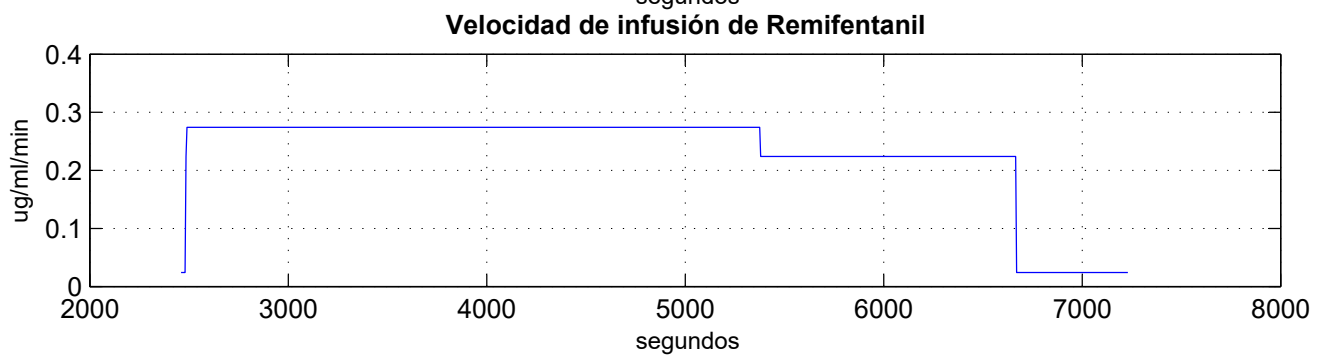
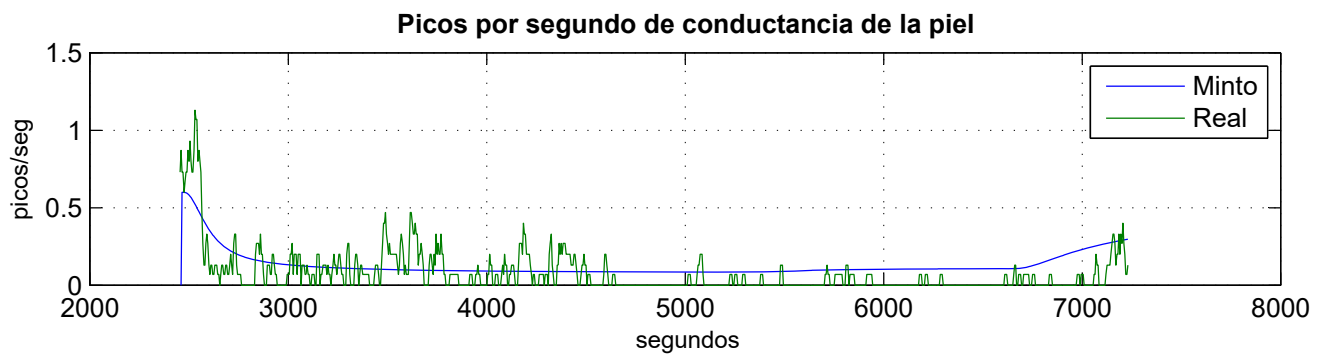
6.1. Modelado

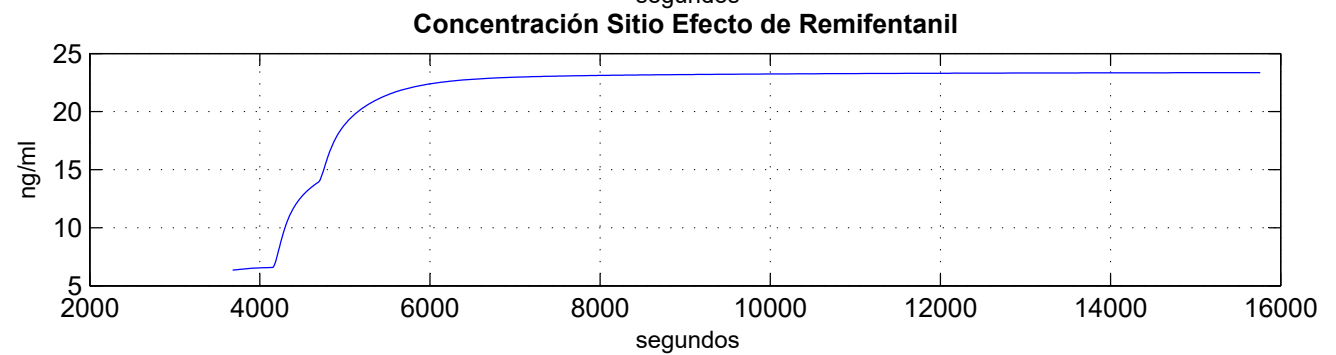
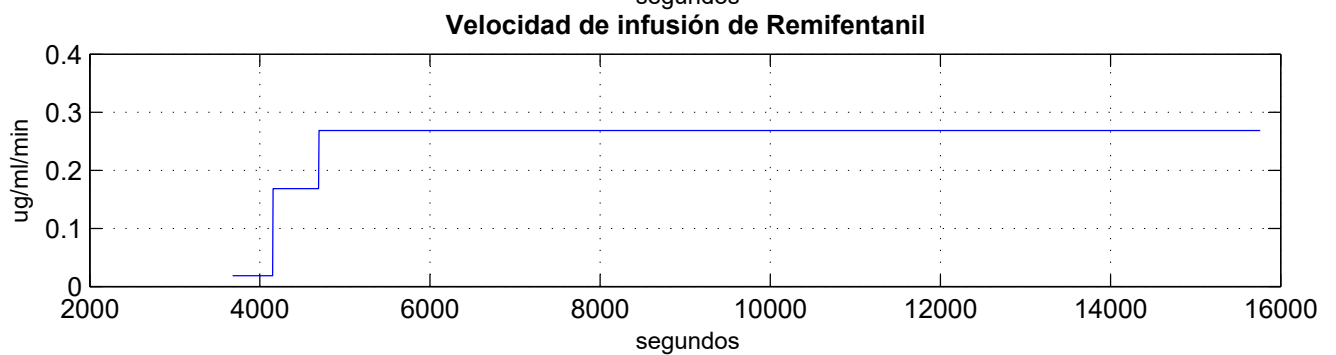
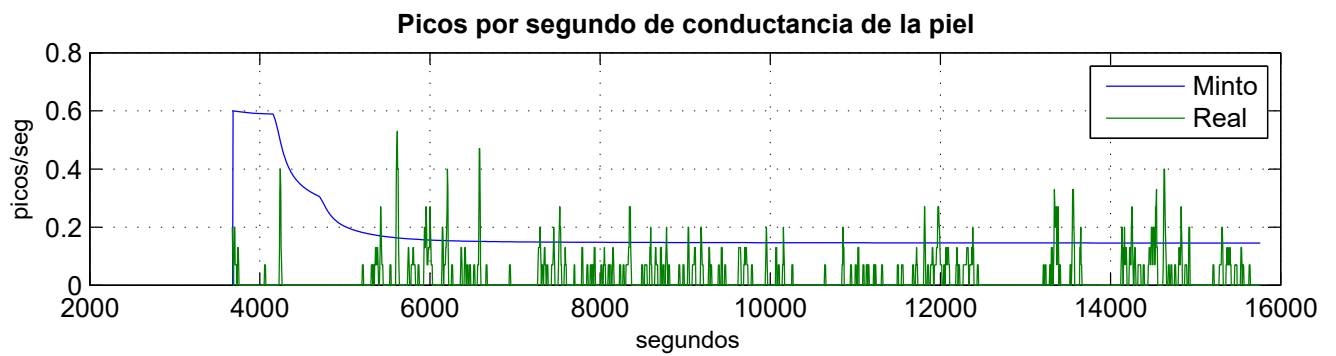


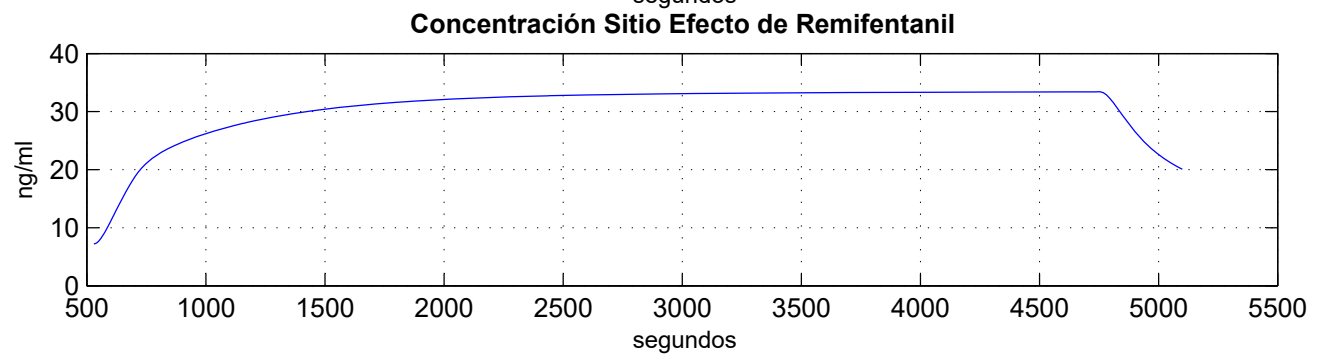
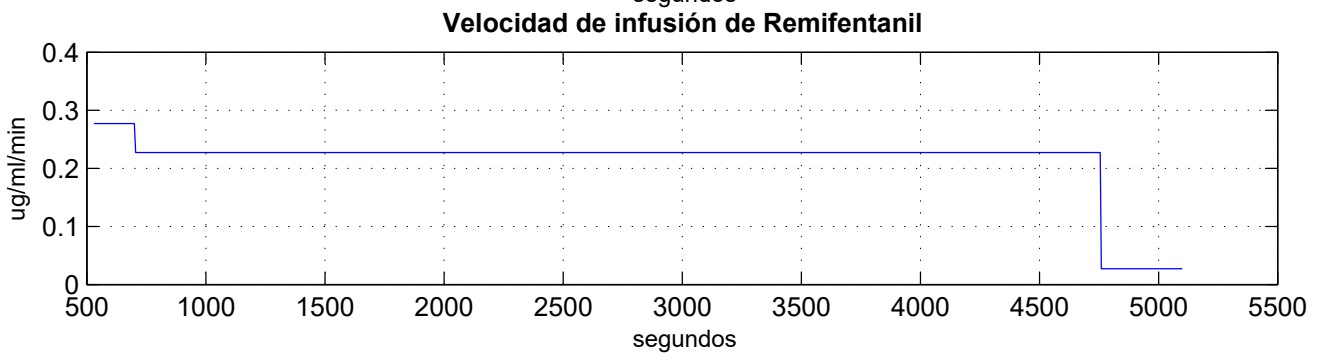
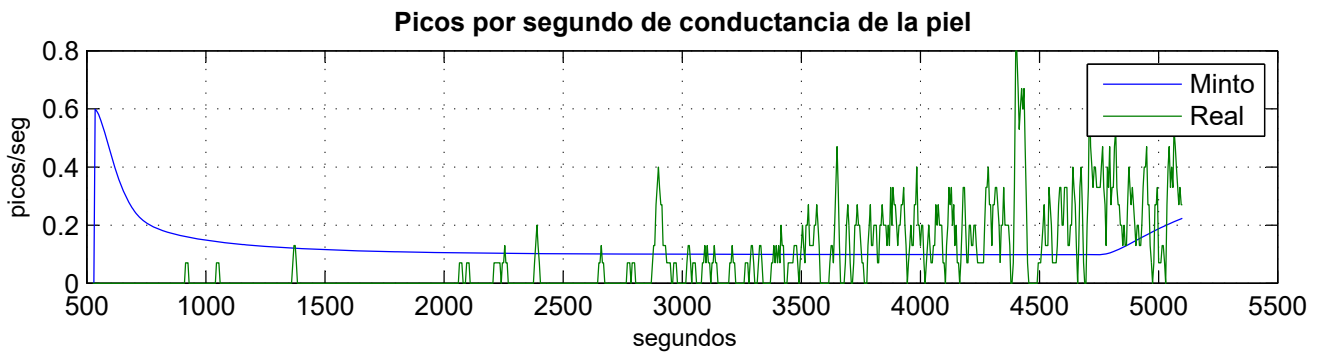


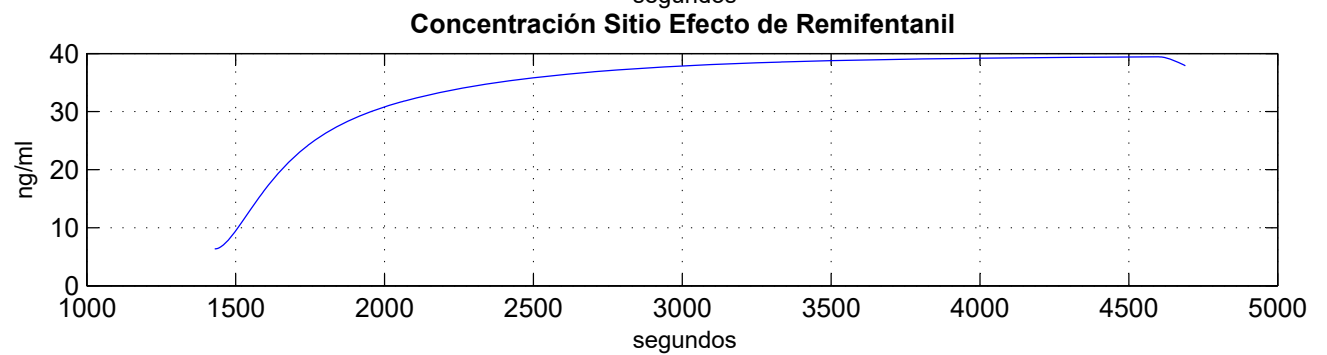
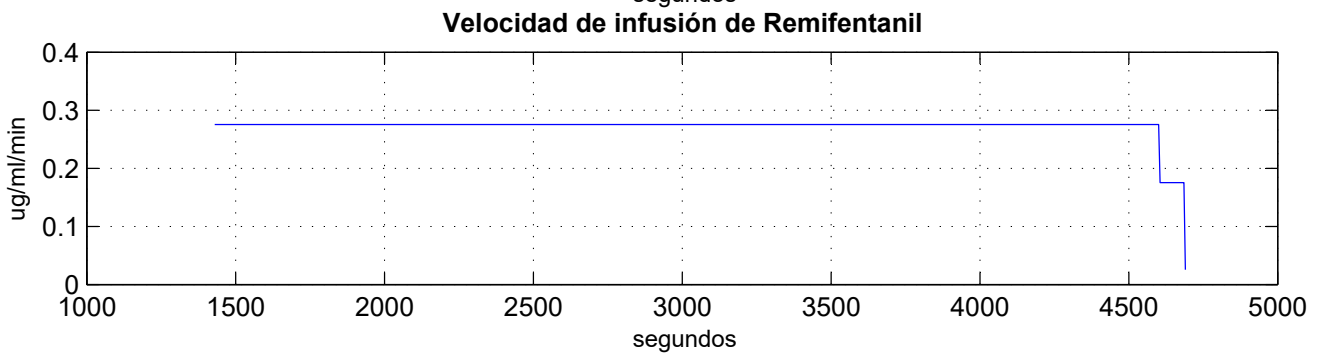
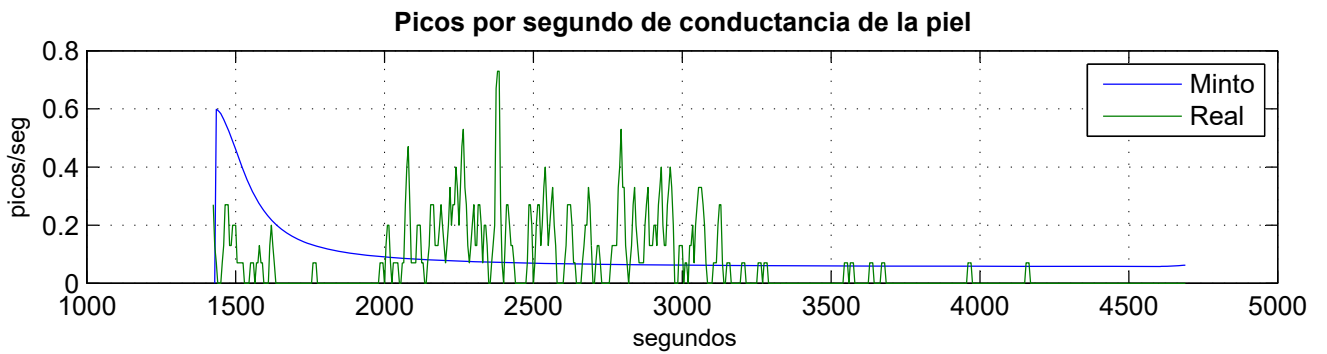


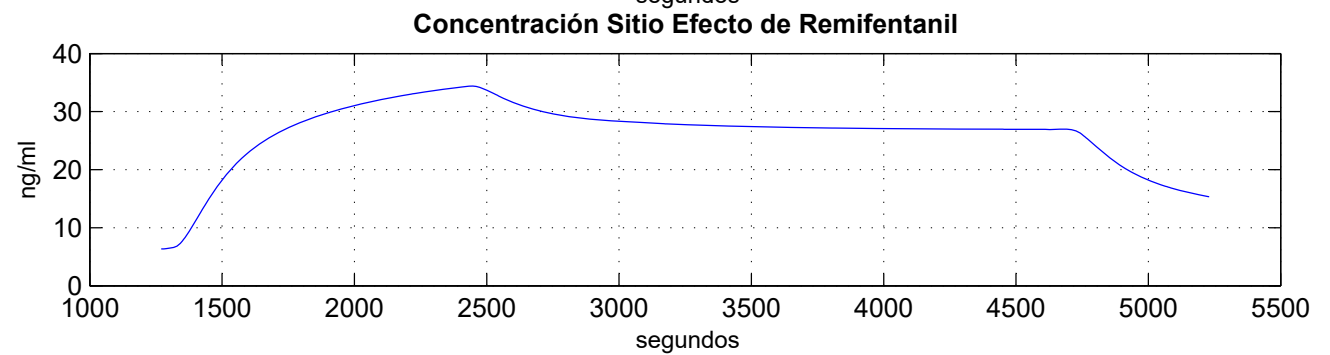
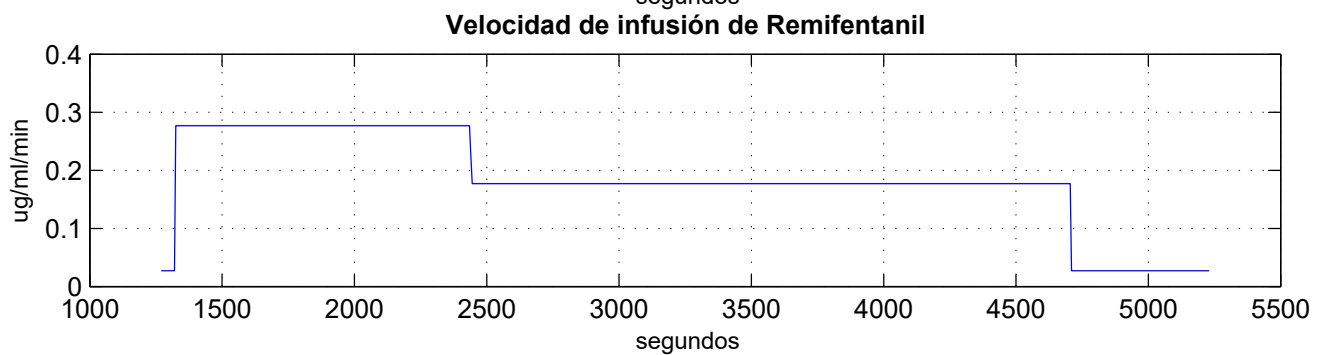
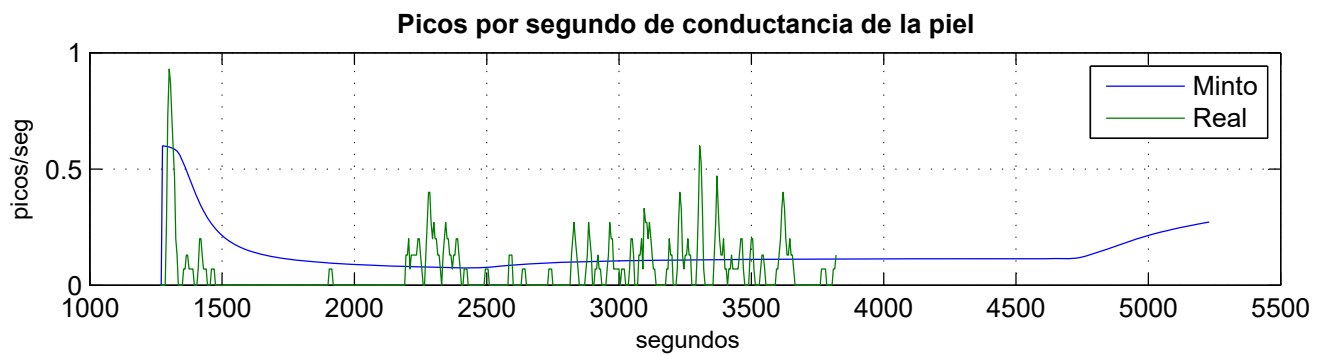


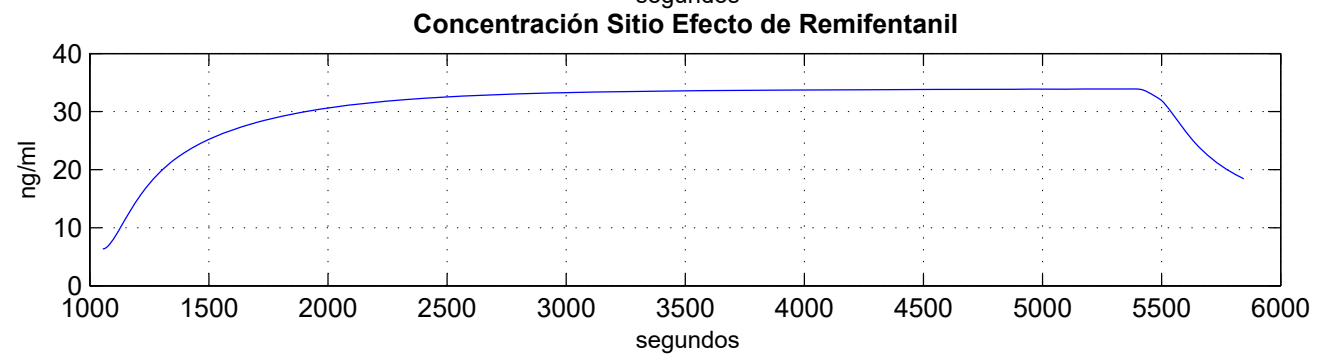
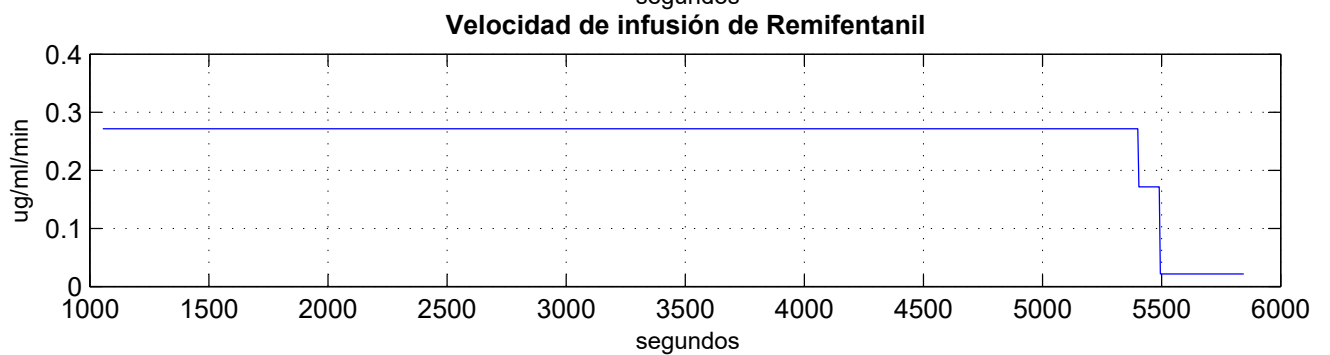
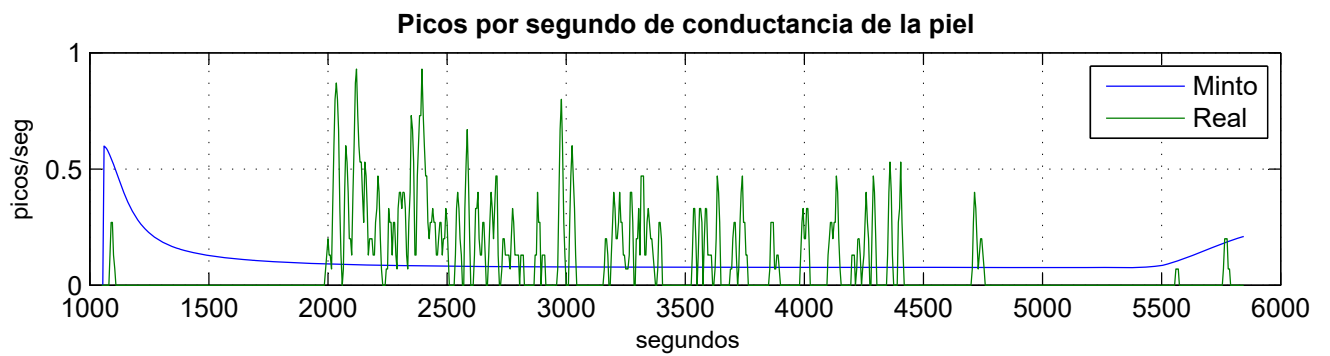


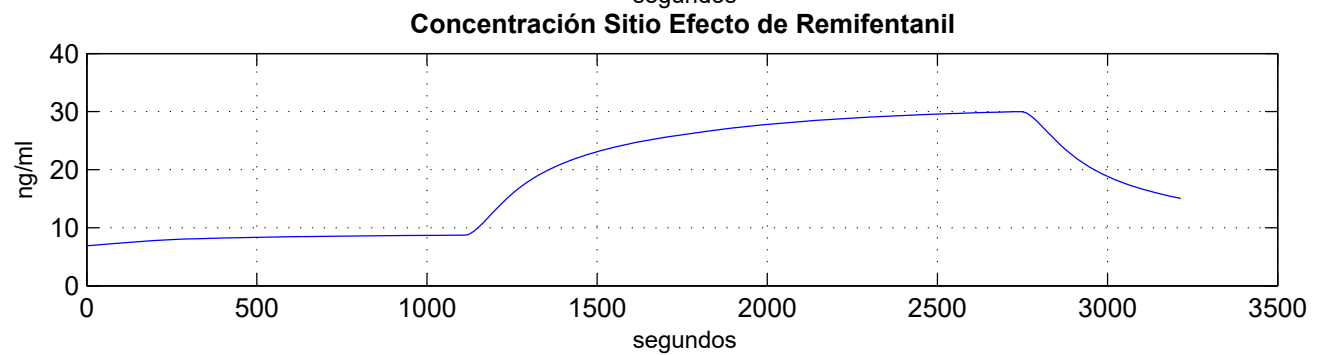
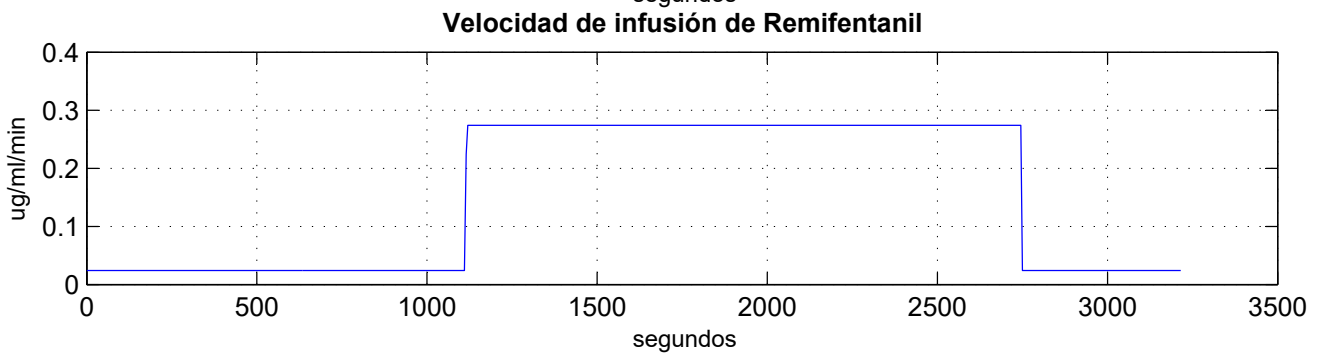
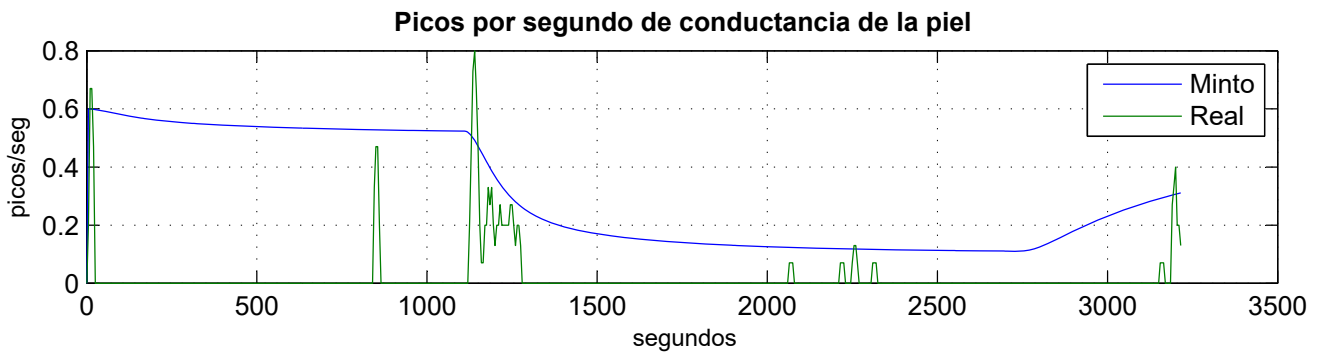


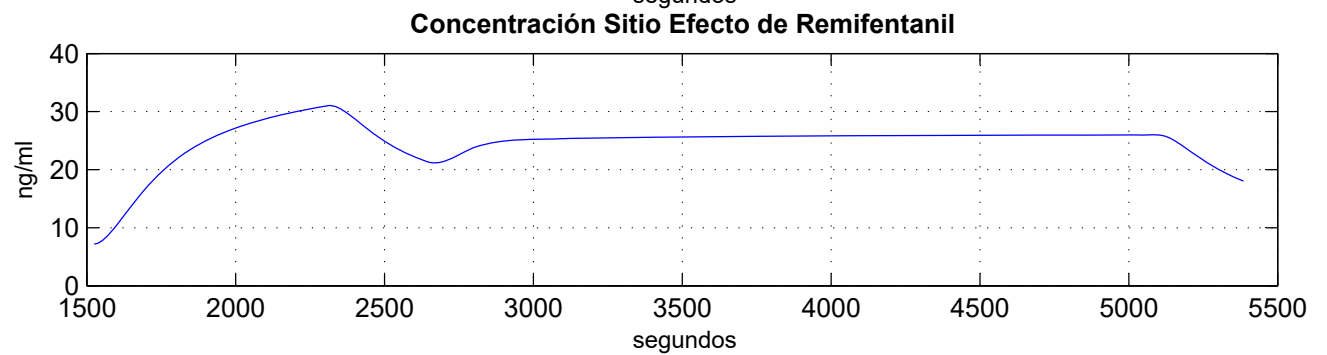
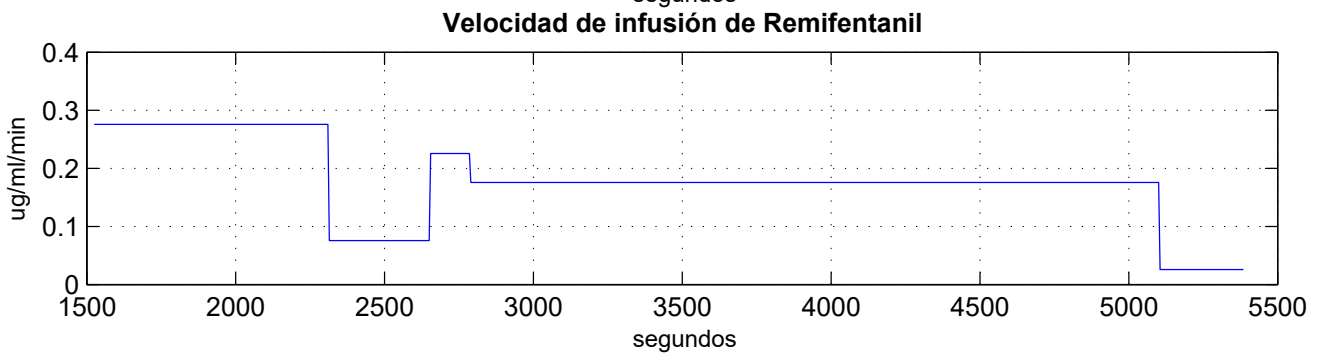
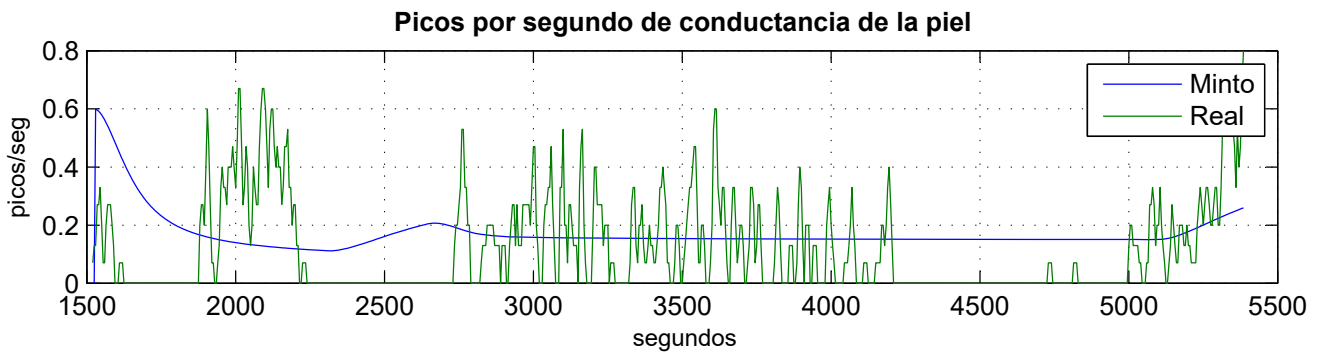


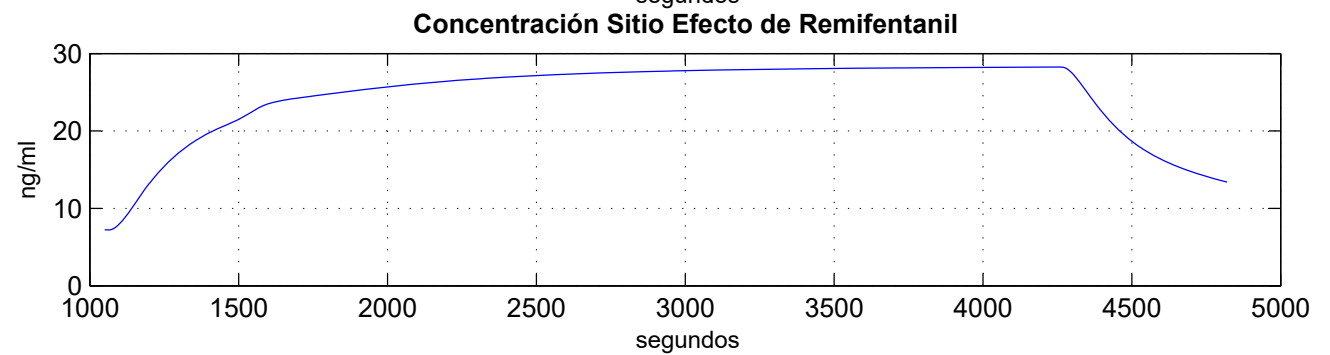
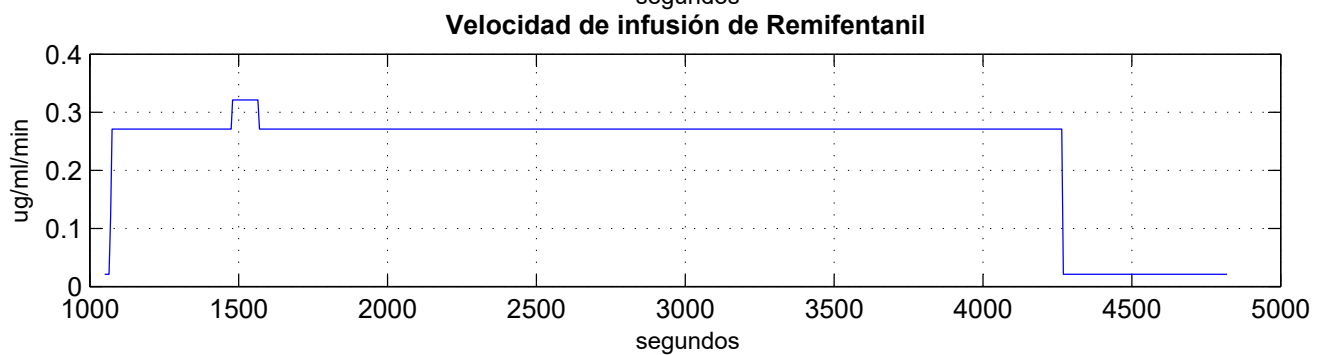
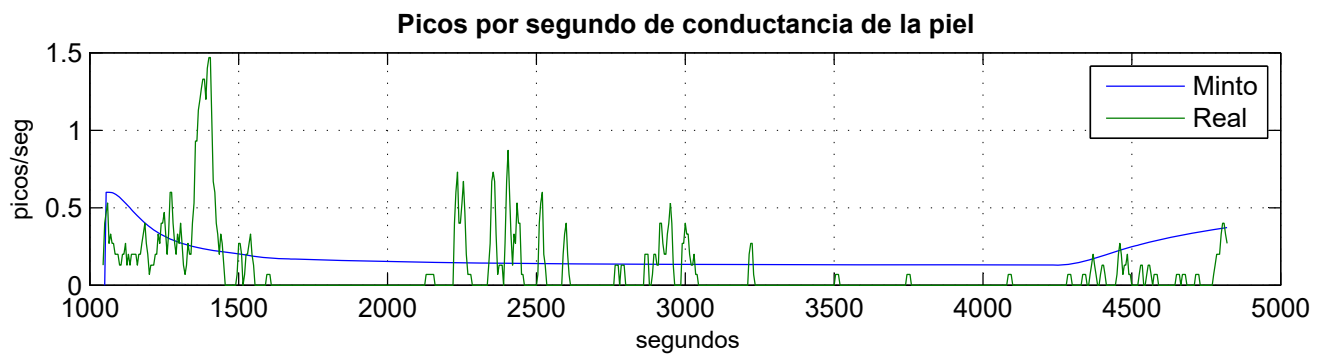


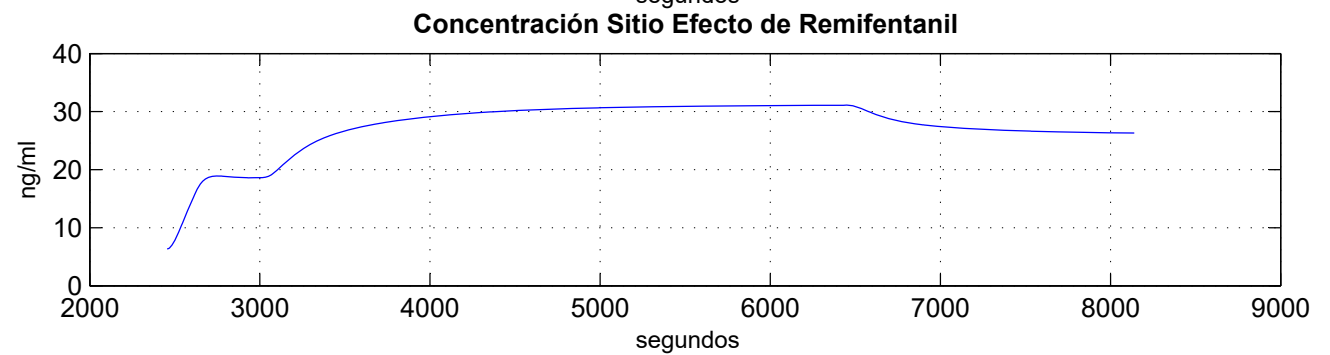
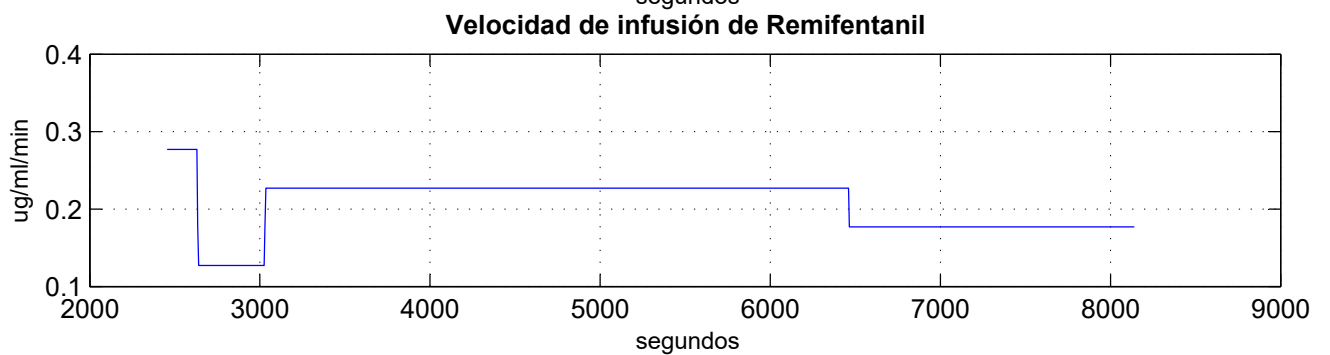
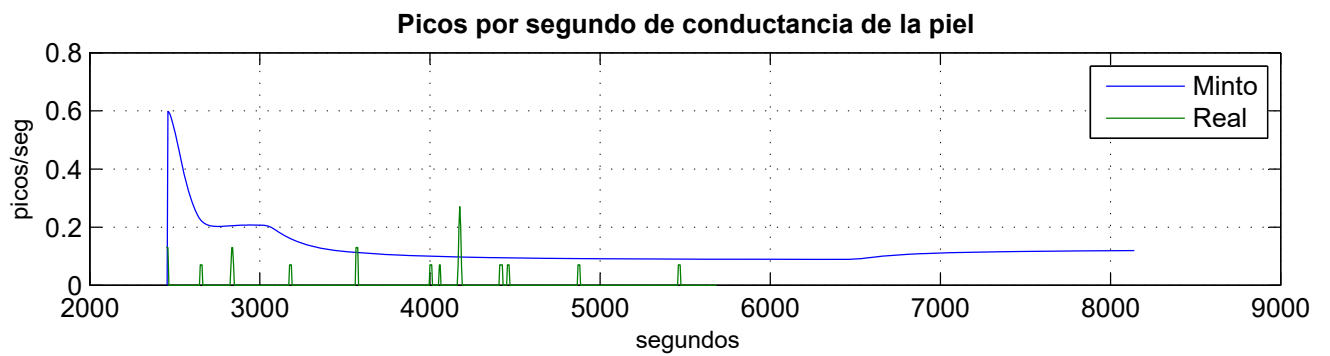


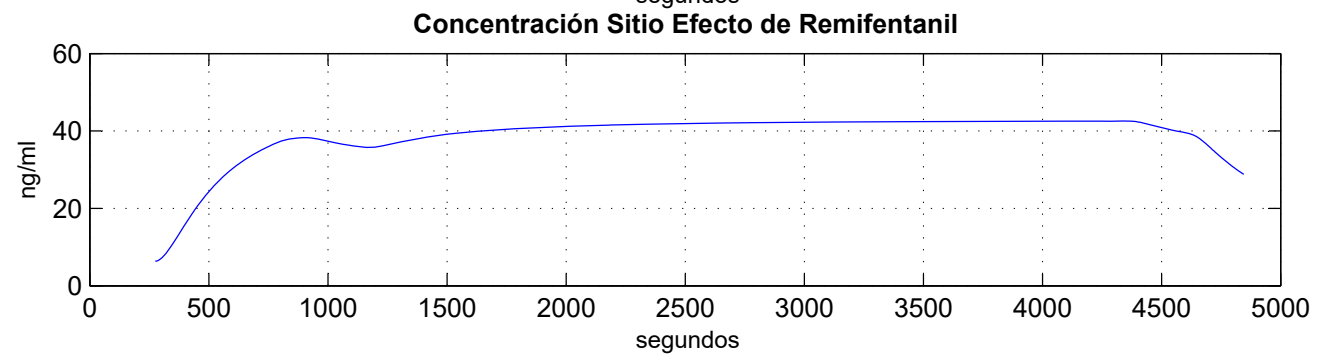
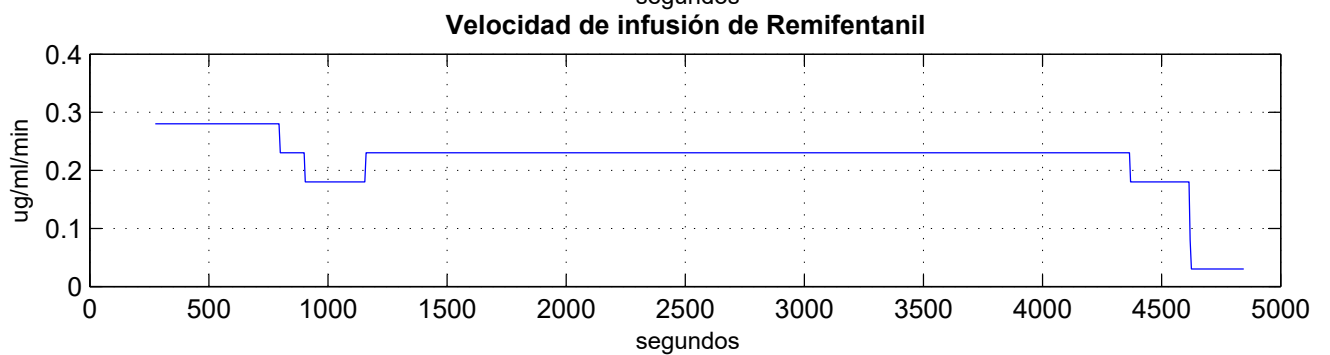
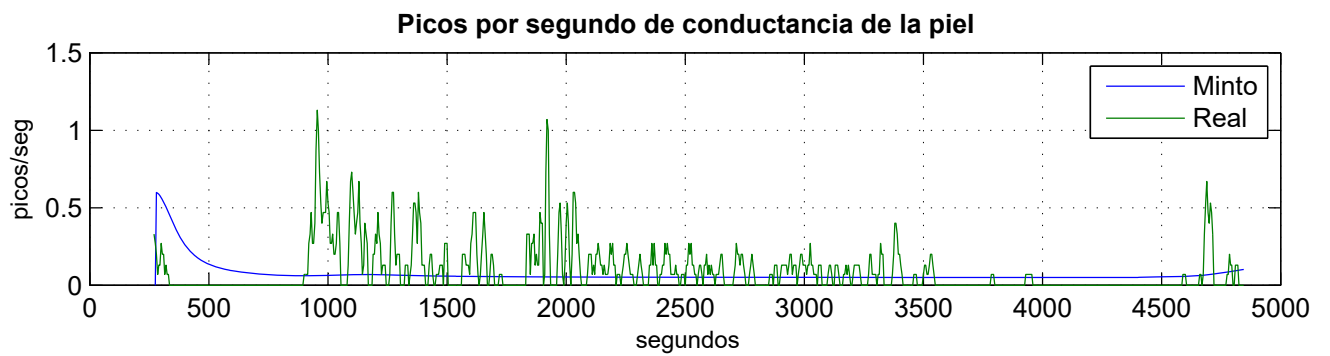


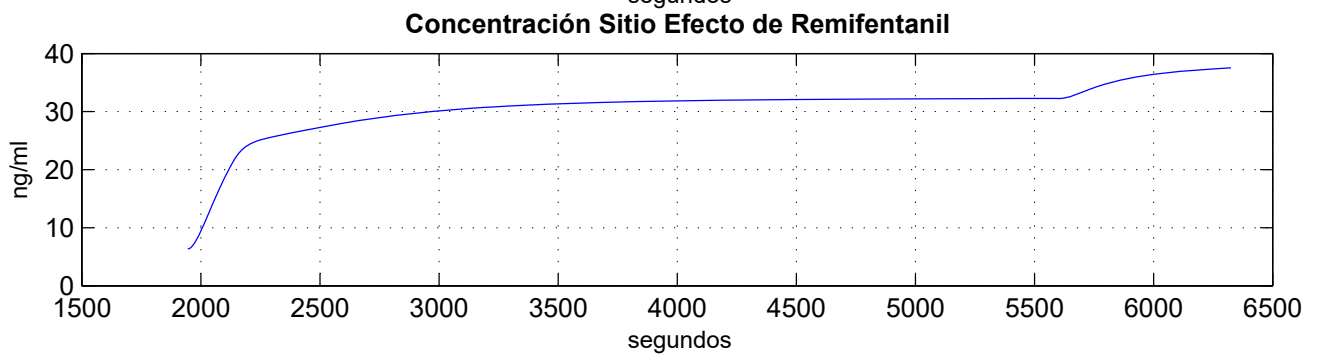
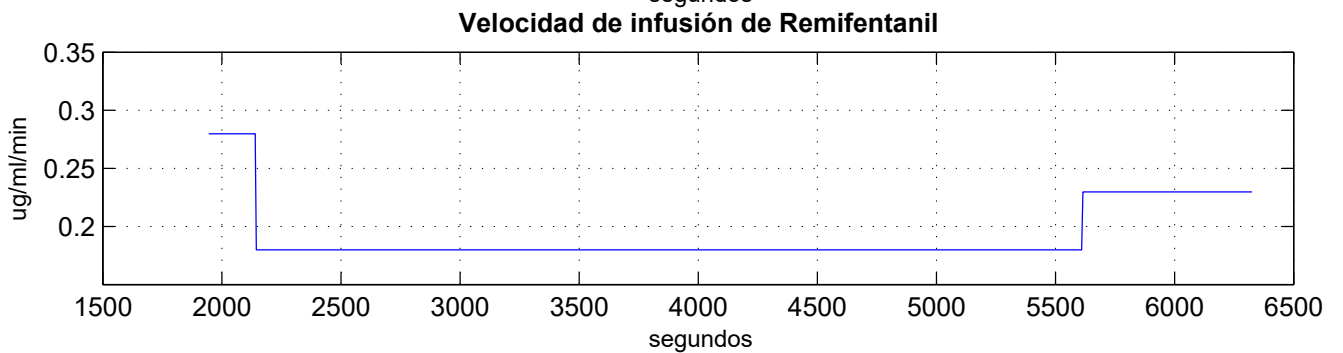
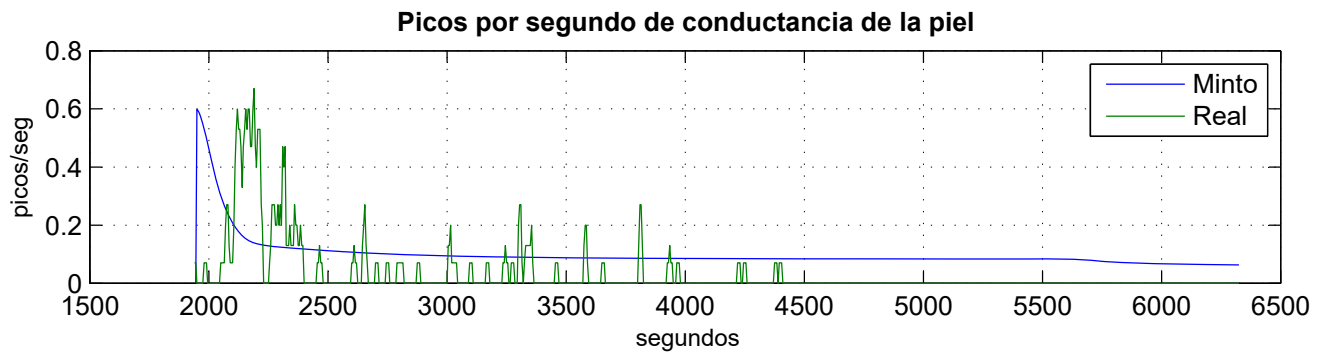


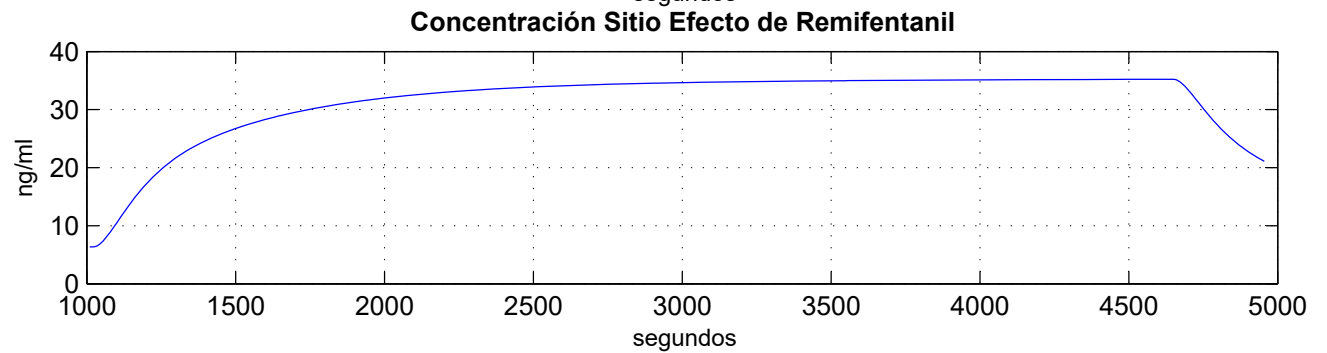
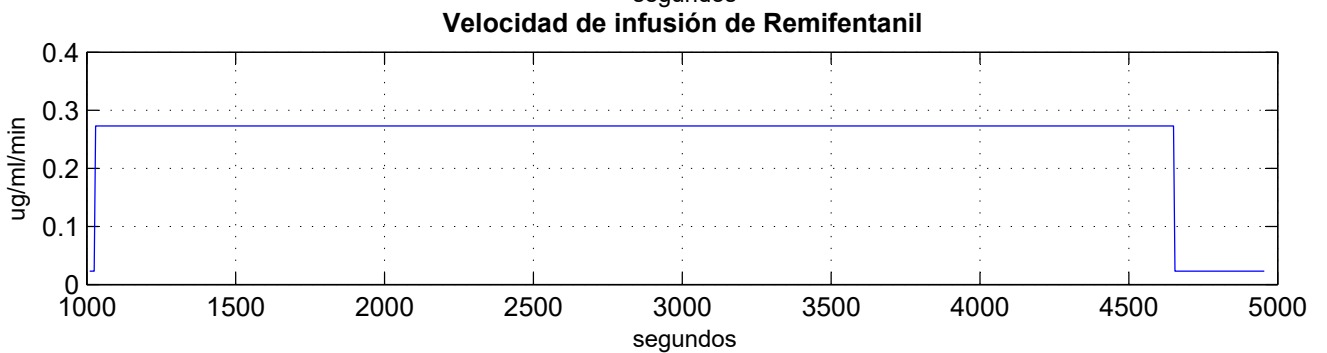
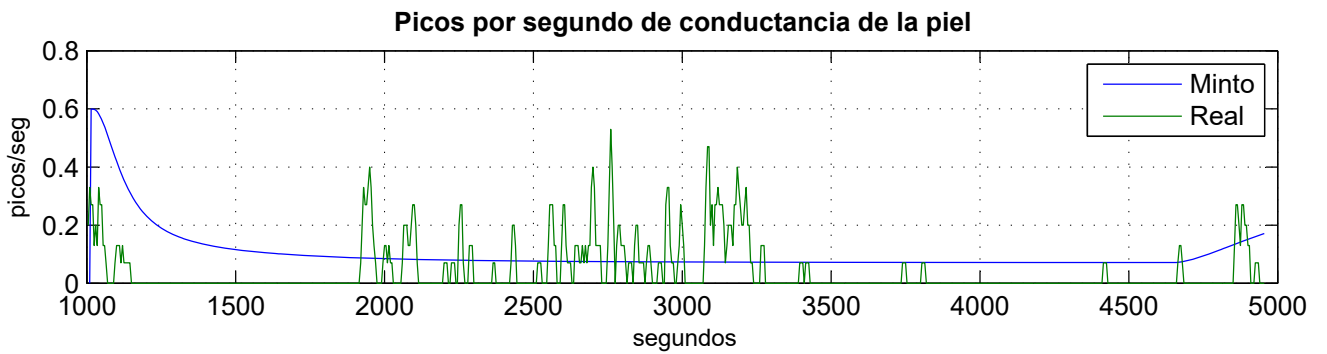


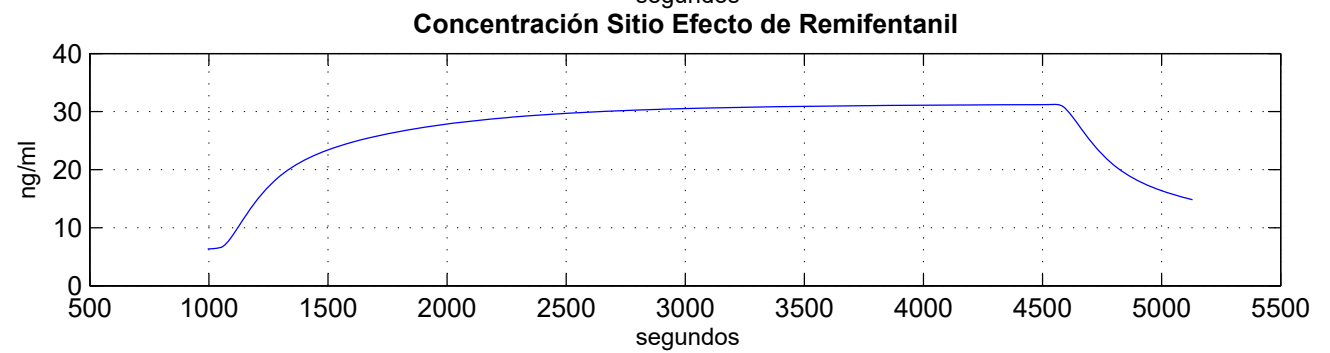
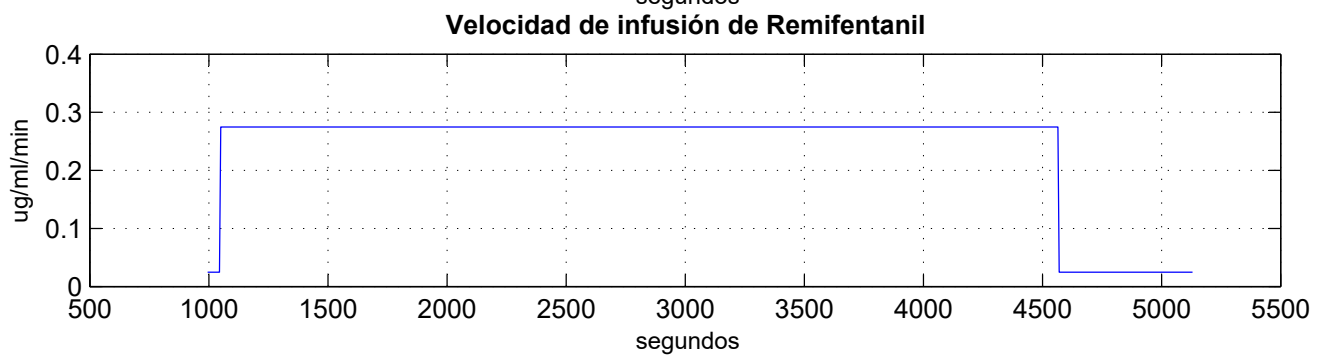
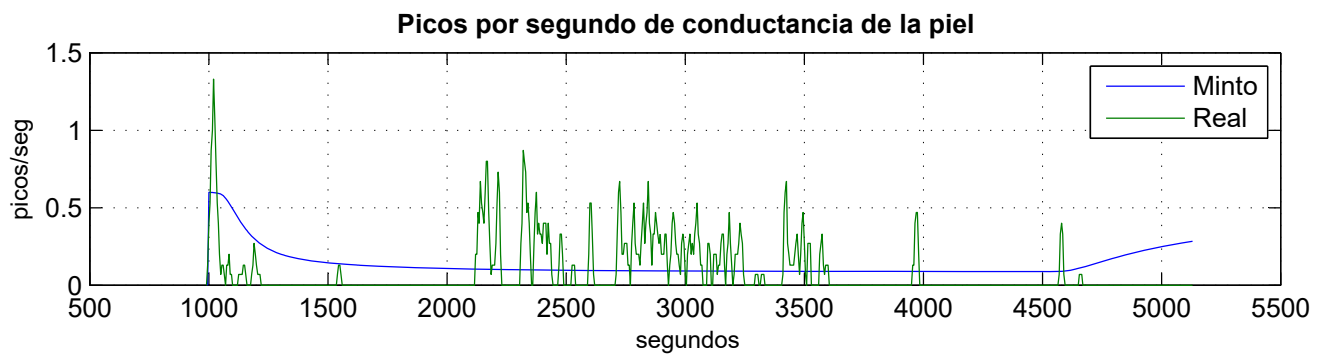


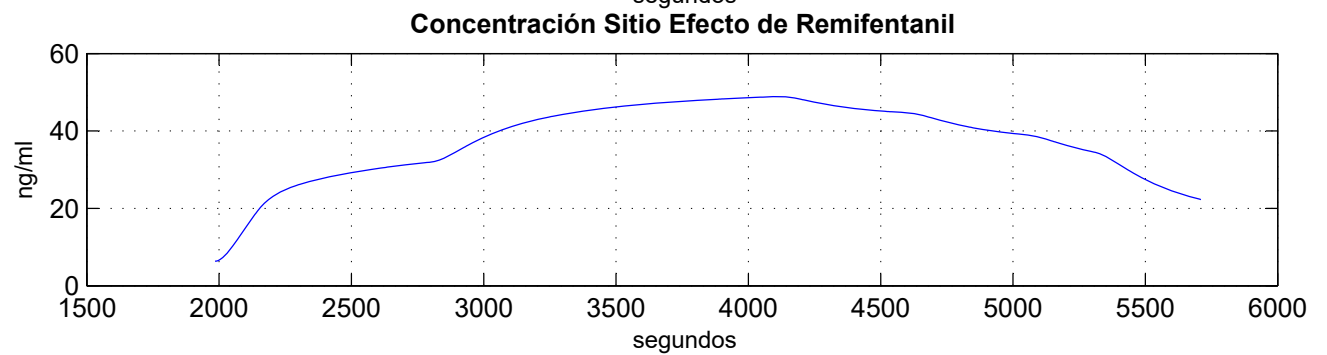
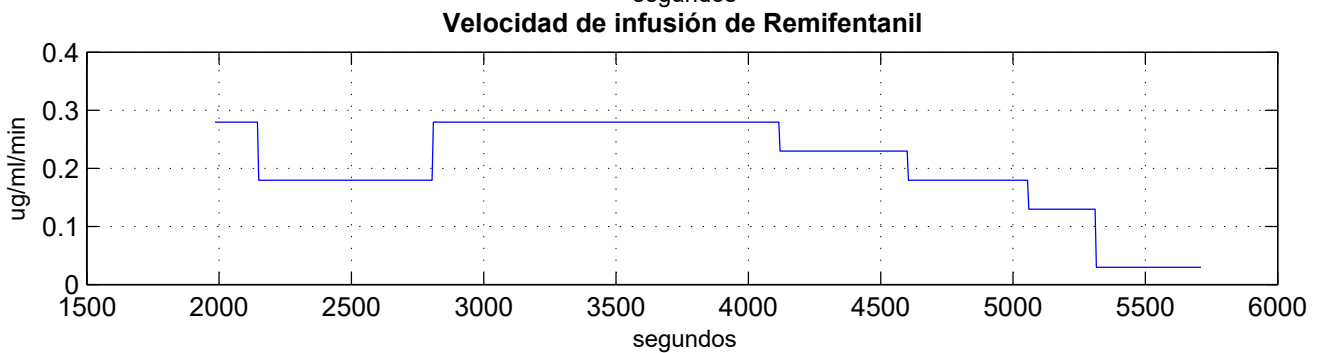
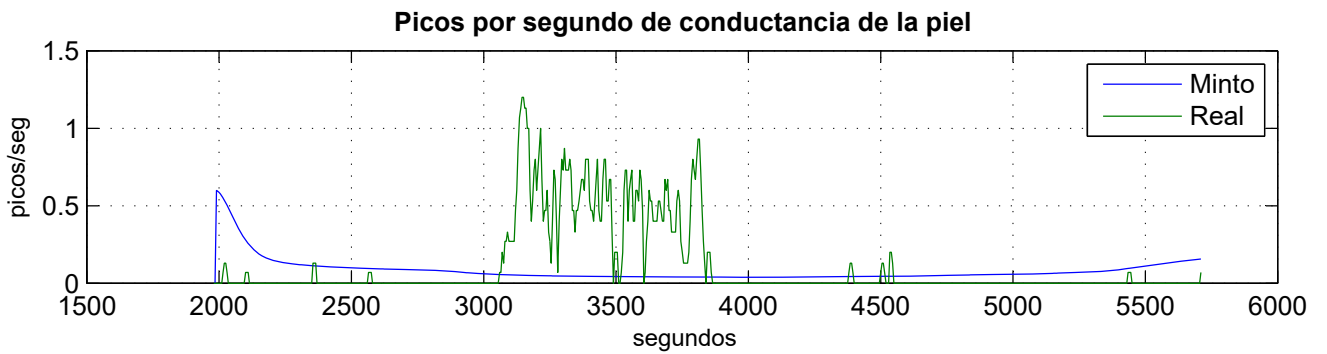


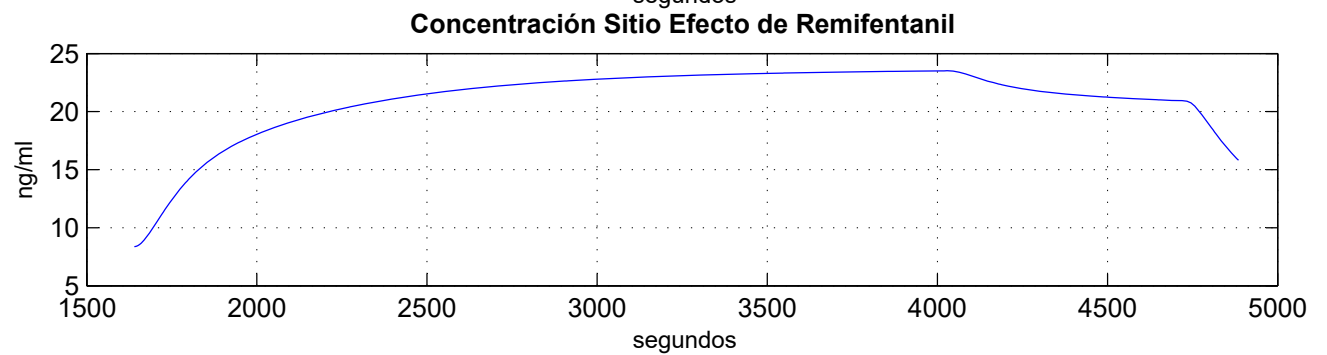
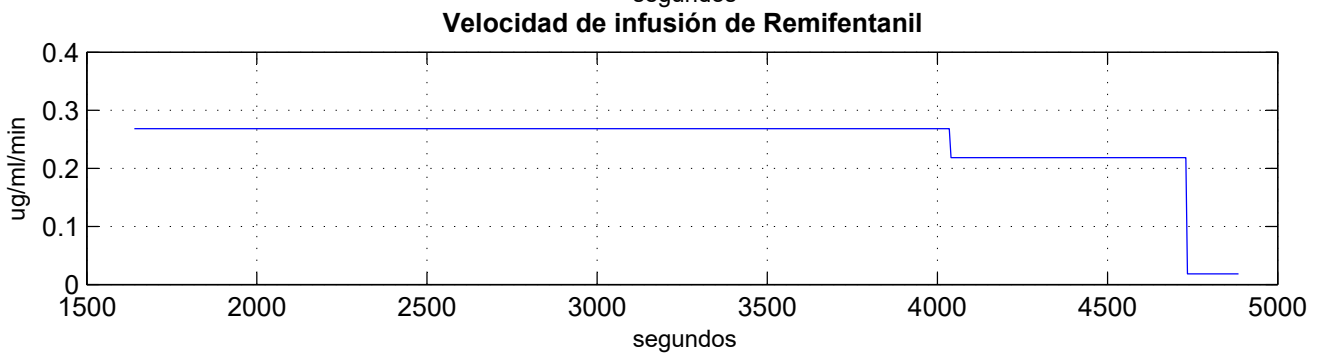
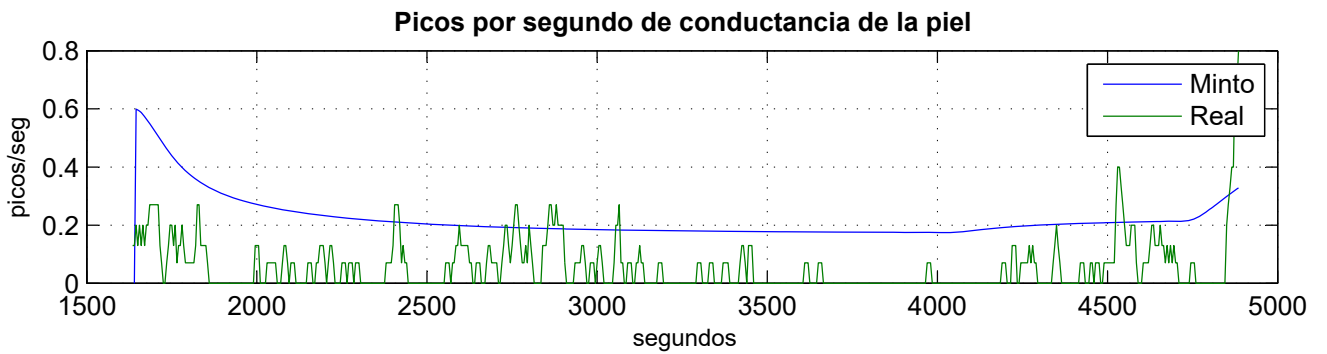


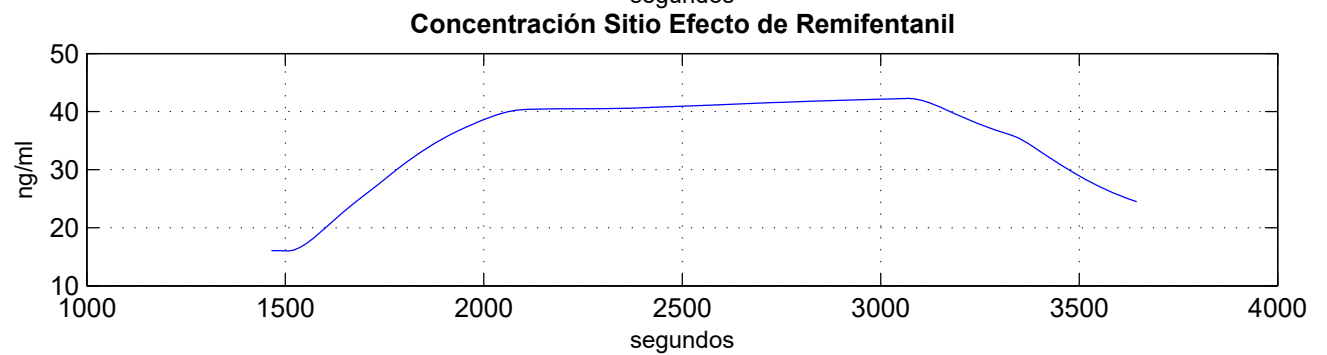
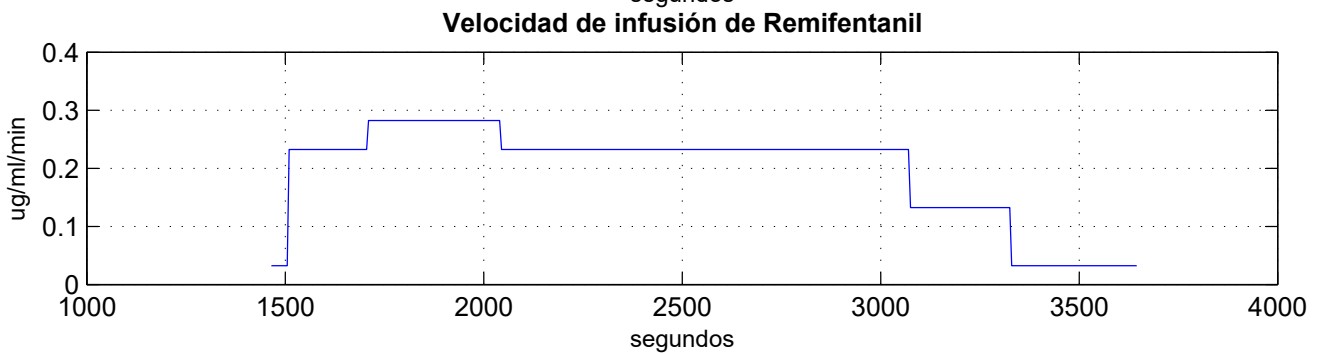
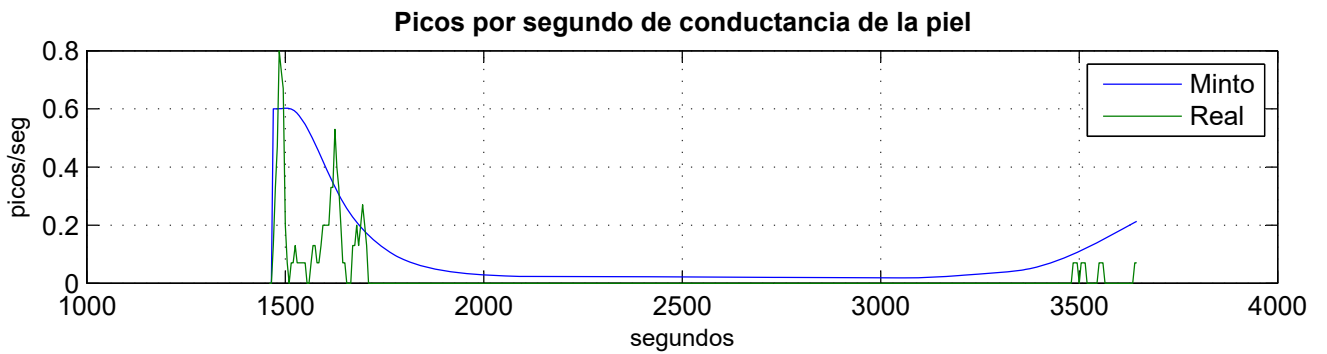


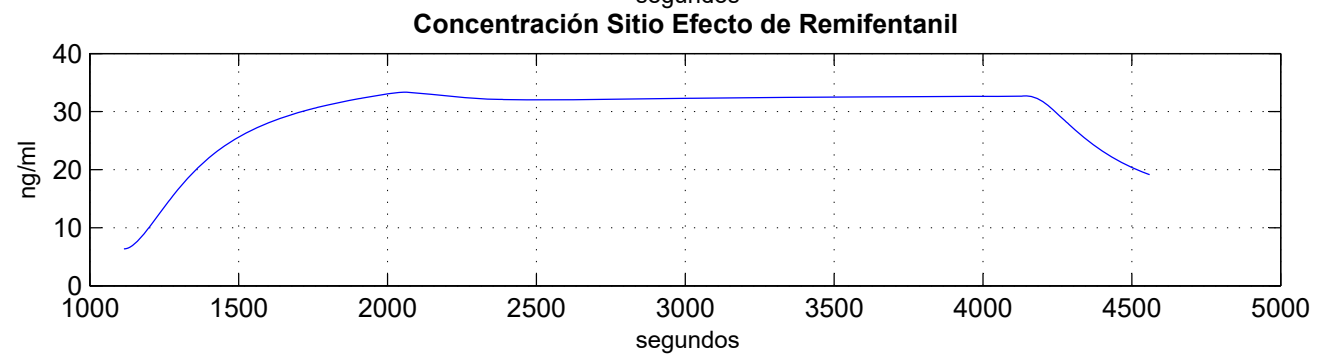
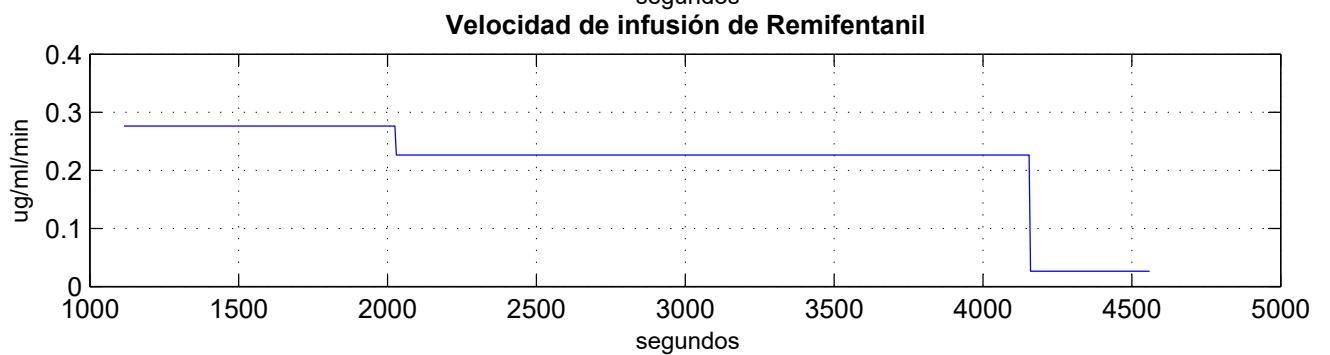
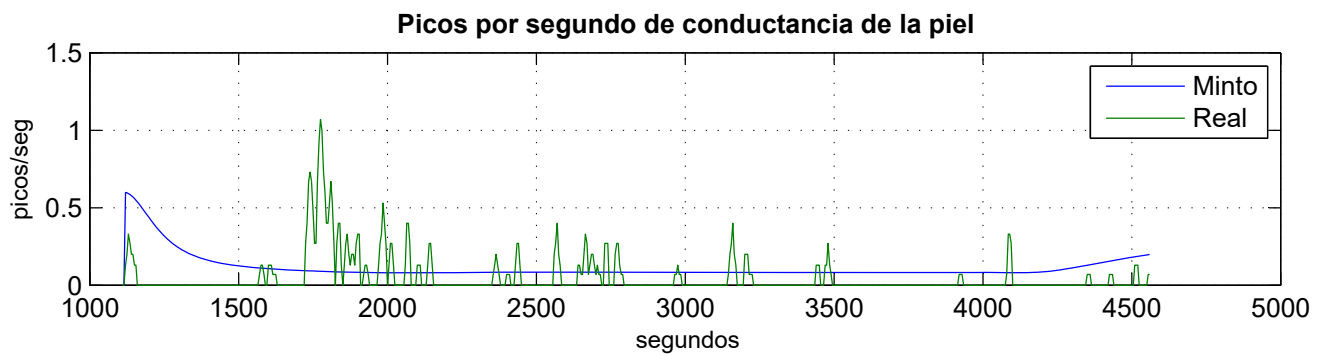


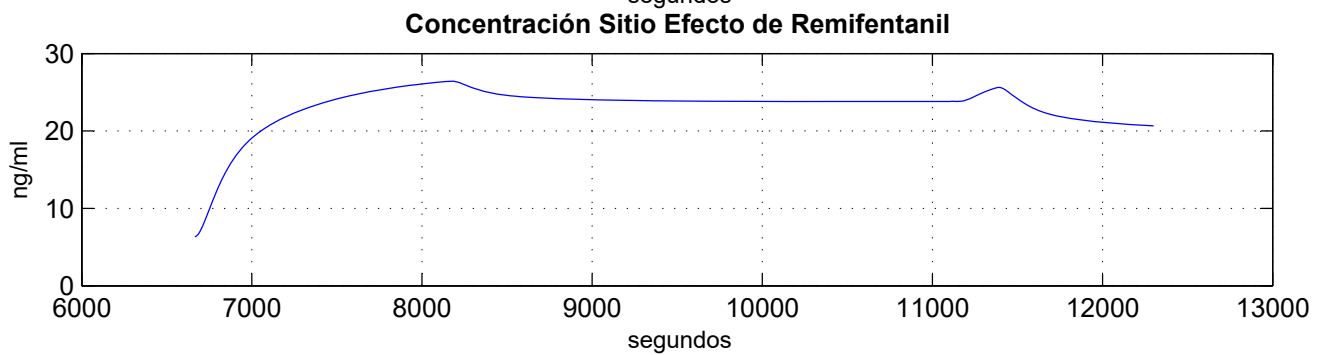
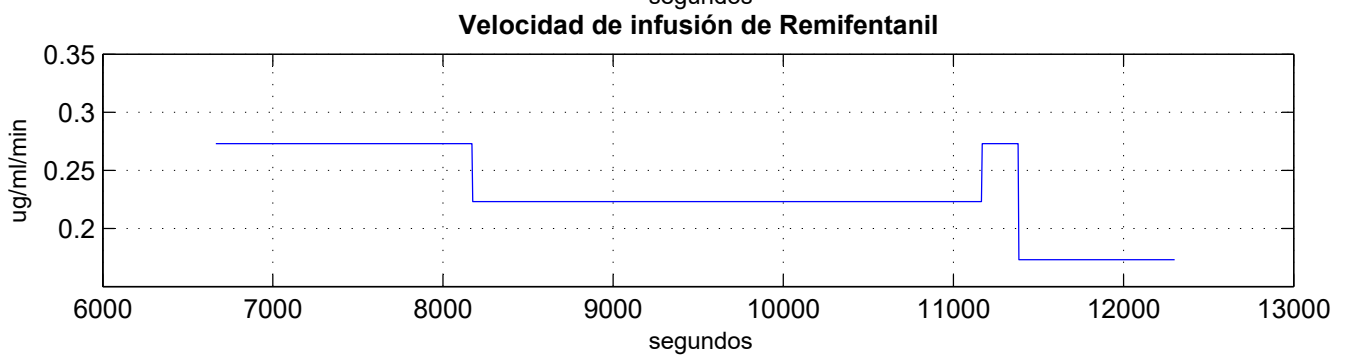
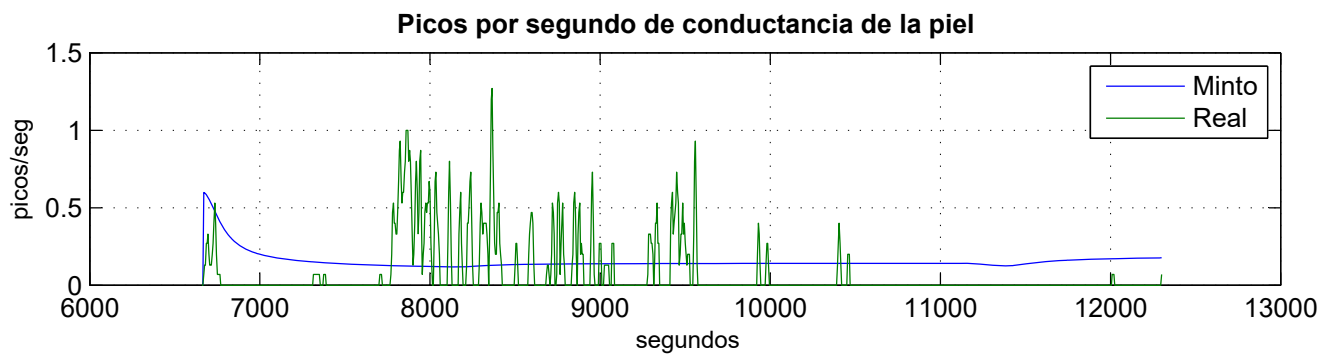


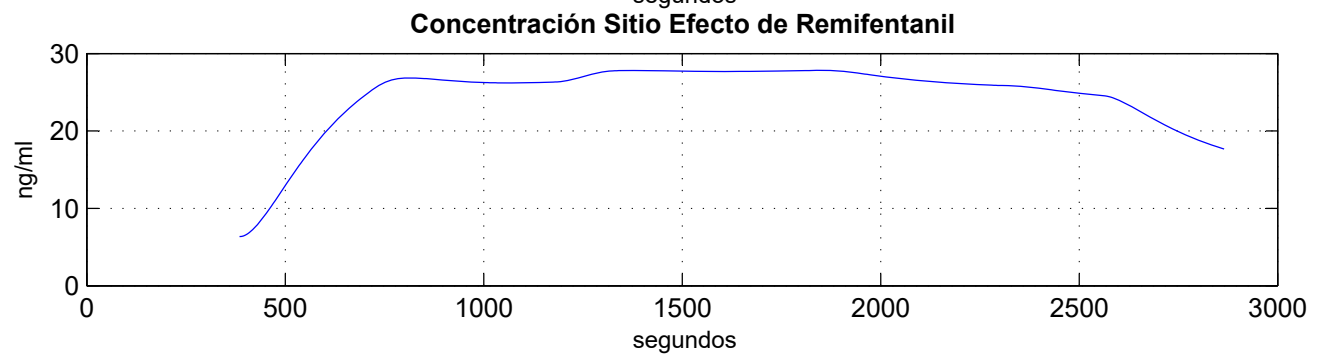
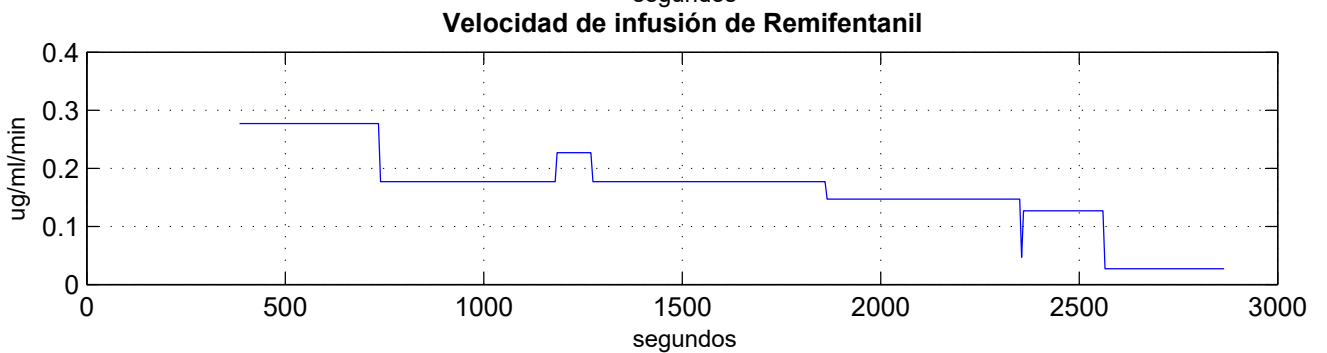
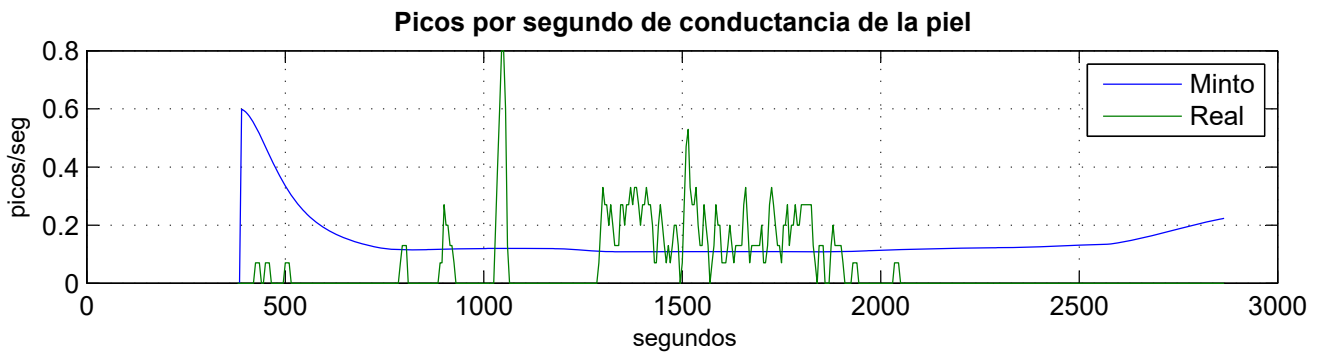


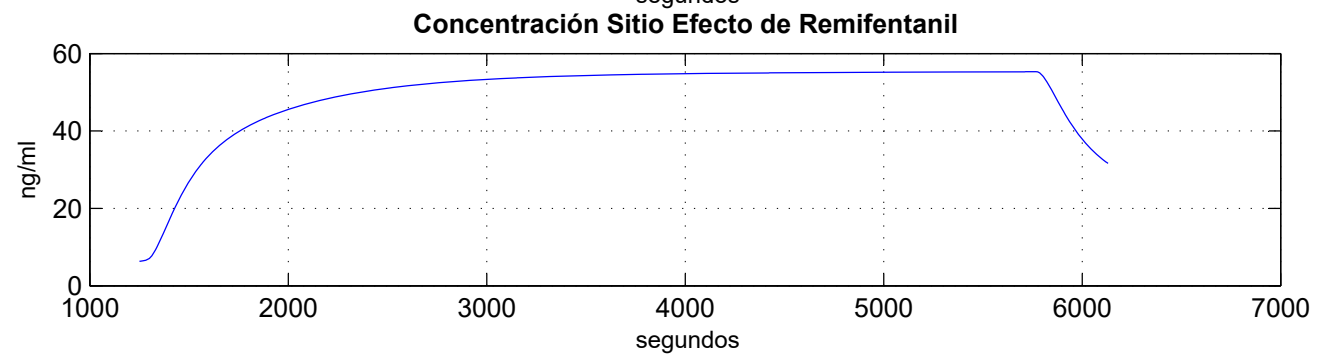
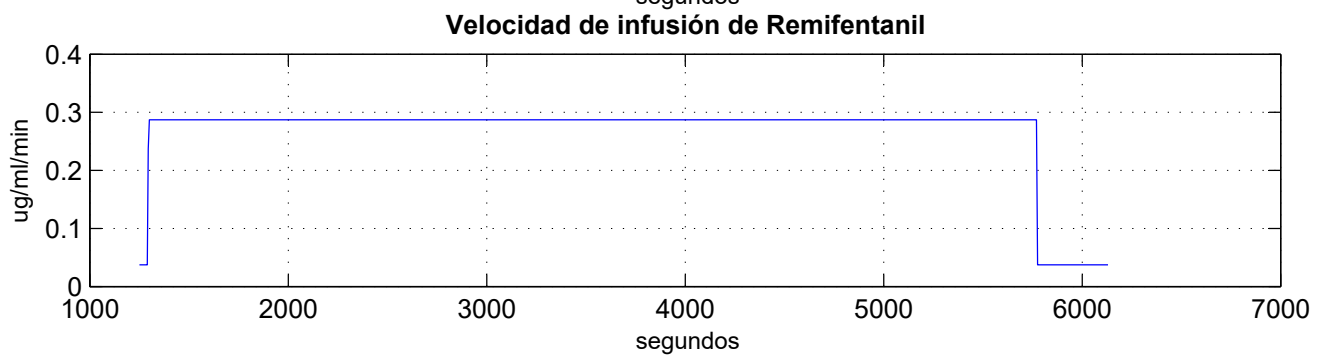
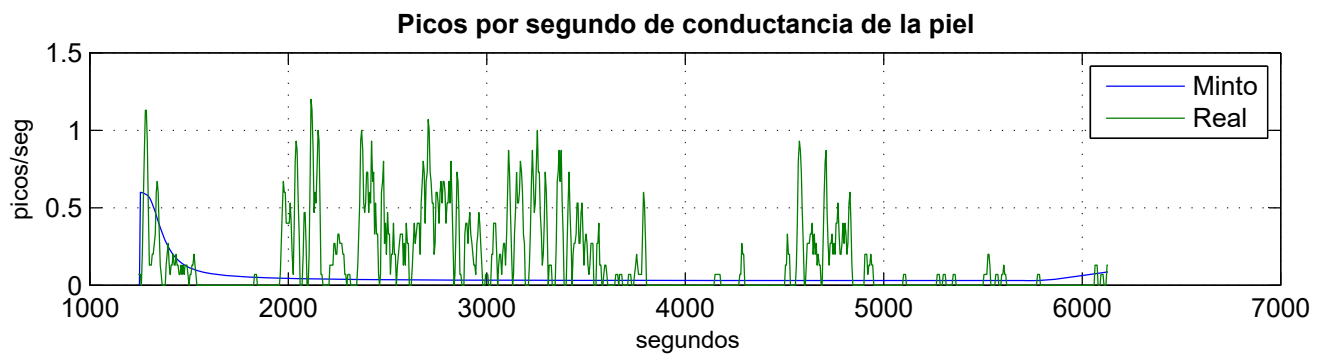


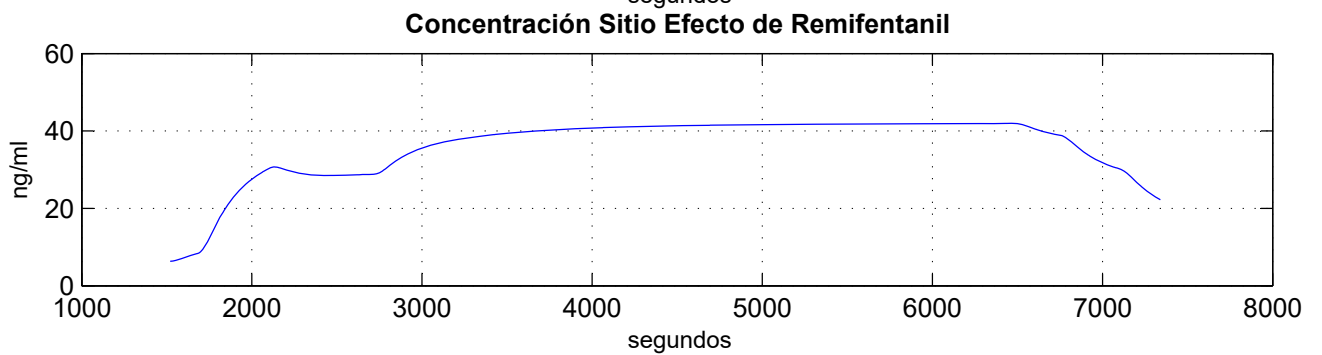
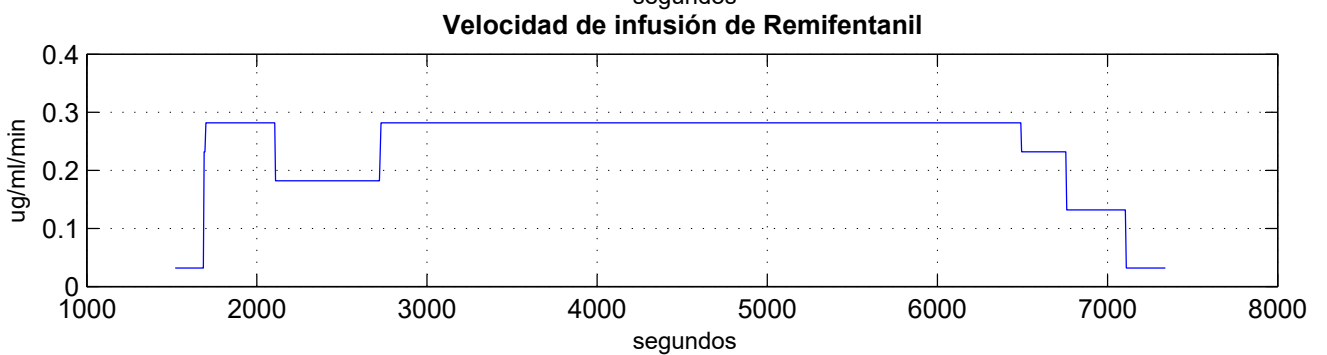
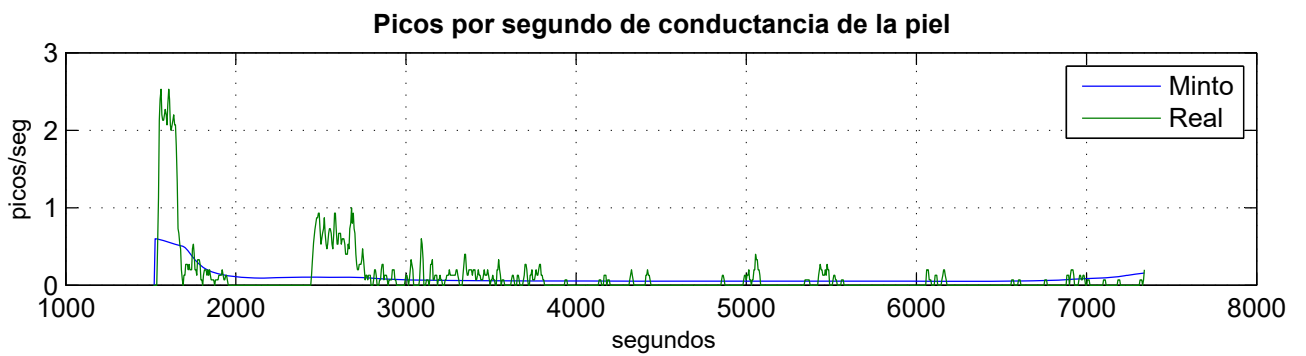


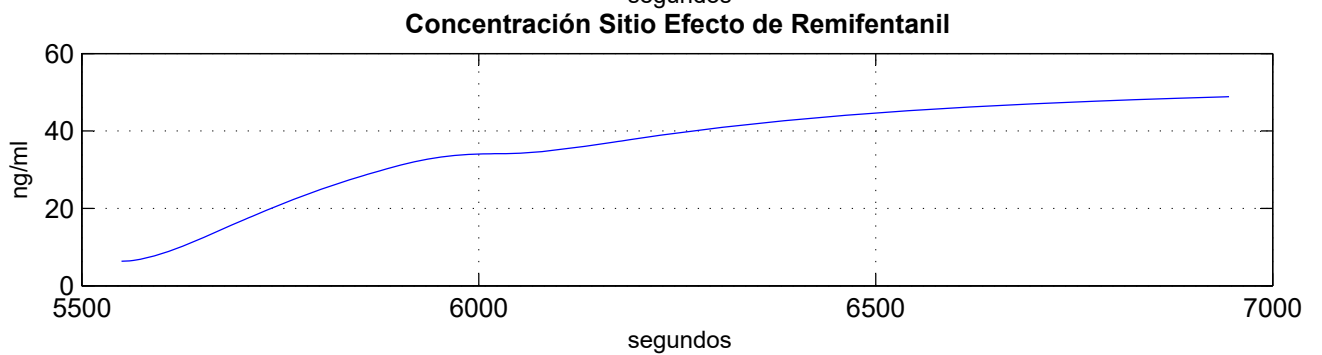
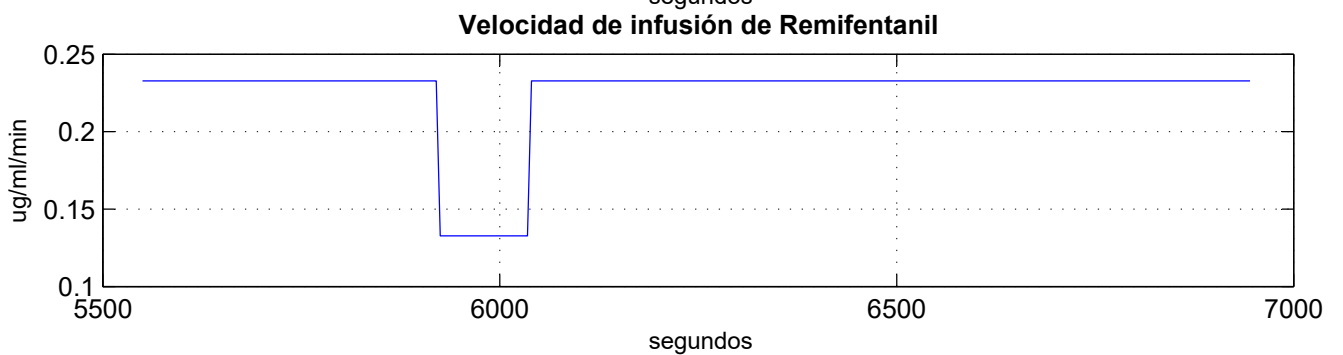
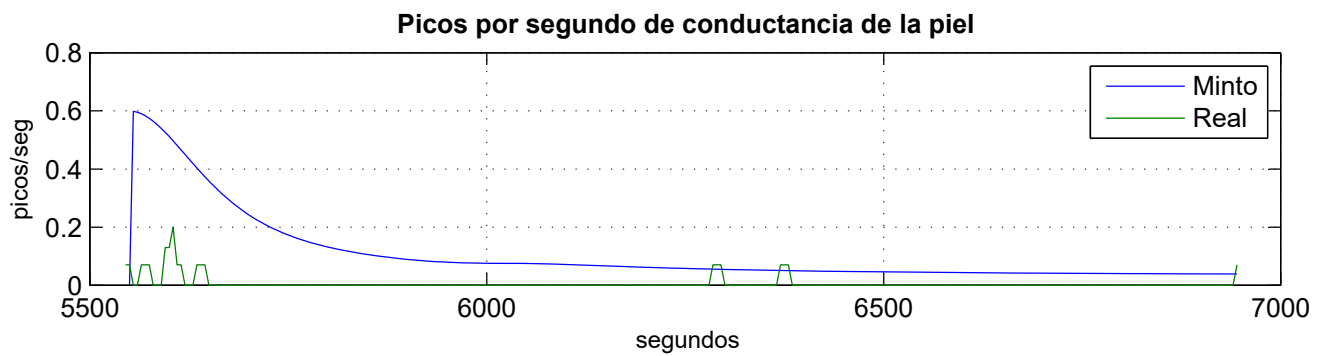












6.2. Código

```

1  function [A,B,C,D,picos_seg_0 ,picos_seg_min ,ec50 ,g,V1] = MintoModel(WT,HT,
    Age, Gender , ts)
2
3  %Función que genera parámetros farmacocinéticos y farmacodinámicos según
    el modelo de Minto.
4
5  %Parámetros de entrada:
6  % WT: peso del paciente (kg.)
7  % HT: altura del paciente (cm.)
8  % Age: edad del paccinte (años)
9  % Gender: género del paciente (1, si masculino, 0 si femenino)
10
11 %Parámetros de salida:
12 % Ac,Bc,Cc,Dc: Matrices de la farmacocinética (formulación continua)
13 % bis0 ,bismin ,ec50 ,g,ke0: Parámetros de la farmacodinámica (cálculo del
    BIS)
14 % V1: Volumen del compartimento central en modelo farmacocinético
15
16 %Calculo de LBM (Lean Body Mass)
17 if Gender==1,
18     LBM = 1.1*WT - 128 * (WT/HT) ^2;
19 elseif Gender==0,
20     LBM = 1.07*WT - 148 * (WT/HT) ^2;
21 else
22     disp('Gender debe ser 0 ó 1. Por defecto se toma Gender=1 (MALE)')
23     Gender = 1;
24     LBM = 1.1*WT - 128 * (WT/HT) ^2;
25 end
26
27 %Pharmacokinetic Model Parameters -----
28 A = Age - 40;
29 L = LBM - 55;
30
31 V1 = 5.1 - 0.0201 * A + 0.072 * L;           % Volumen de droga en el
    compartimento central
32 V2 = 9.82 - 0.0811 * A + 0.108 * L;       % Volumen de droga en el
    compartimento rápido
33 V3 = 5.42;                                 % Volumen de droga en el
    compartimento lento
34 CL1 = 2.6 - 0.0162 * A + 0.0191 * L;     % Concentración droga compartimento
    central
35 CL2 = 2.05 - 0.0301 * A;                 % Concentración droga compartimento
    rápido
36 CL3 = 0.076 - 0.00113 * A;              % Concentración droga compartimento
    lento
37
38 k12 = CL2 / V1;

```

```

39 k13 = CL3 / V1;
40 k21 = CL2 / V2;
41 k31 = CL3 / V3;
42 k10 = CL1 / V1;
43 ke0 = 0.595 - 0.007 * (Age - 40);
44
45
46 % Ecuaciones del modelo Farmacocinético
47 Ac=[-(k10+k12+k13)   k21*V2/V1   k31*V3/V1   0
48      k12*V1/V2       -k21         0         0
49      k13*V1/V3       0             -k31       0
50      ke0              0             0         -ke0];
51 Bc=[1
52     0
53     0
54     0];
55 Cc=[0 0 0 1];
56 Dc=0;
57
58 % Discretización del modelo en variables de estado
59 sysc = ss(Ac,Bc,Cc,Dc);
60 sysd = c2d(sysc,ts);
61
62 A = sysd.A;
63 B = sysd.B;
64 C = sysd.C;
65 D = sysd.D;
66
67 % Definimos los parámetros de la Farmacodinámica
68 picos_seg_0 = 0.8;
69 picos_seg_min = 0;
70 ec50 = 20; % (ug/l)
71 g = 2;

```

```

1
2   function [A,B,Un,C,D,picos_seg_0,picos_seg_min,ec50,g,V1] = InverseModel(
      WT,HT, Age, Gender, ts, picos_seg_wish)
3   %Definimos los parámetros de la farmacodinámica
4   picos_seg_0=0.8;
5   ec50= 11;
6   g=2;
7   picos_seg_min= 0;
8
9   %definimos alfa
10  alpha = ((picos_seg_wish - picos_seg_0)/ (picos_seg_min-picos_seg_0 ));
11  %concentración en el sitio efecto
12  if alpha == 1,
13      Ce = 0;
14  else
15      Ce= exp((1/g)*log((ec50^g)*alpha/(1-alpha)));
16  end
17  %Calculo de LBM (Lean Body Mass/Masa muscular magra)
18  if Gender==1,
19      LBM = 1.1*WT - 128 * (WT/HT)^2;
20  elseif Gender==0,
21      LBM = 1.07*WT - 148 * (WT/HT)^2;
22  else
23      disp('Gender debe ser 0 ó 1. Por defecto se toma Gender=1 (MALE)')
24      Gender = 1;
25      LBM = 1.1*WT - 128 * (WT/HT)^2;
26  end
27
28  %Parámetros del modelo farmacocinético
29  A = Age - 40;
30  L = LBM - 55;
31
32  V1 = 5.1 - 0.0201 * A + 0.072 * L ; %Volumen de droga en el compartimento
      central
33  V2 = 9.82 - 0.0811 * A + 0.108 * L ; %Volumen de droga en el compartimento
      rápido
34  V3 = 5.42; %Volumen de droga en el compartimento lento
35  CL1 = 2.6 - 0.0162 * A + 0.0191 * L; % Concentración droga compartimento
      central
36  CL2 = 2.05 - 0.0301 * A; % Concentración droga compartimento rápido
37  CL3 = 0.076 - 0.00113 * A ; % Concentración droga compartimento lento
38  %Constantes de difusión
39  k12 = CL2 / V1;
40  k13 = CL3 / V1;
41  k21 = CL2 / V2;
42  k31 = CL3 / V3;
43  k10 = CL1 / V1;

```

```

44 ke0 = 0.595 - 0.007 * (Age - 40);
45
46 % Ecuaciones del modelo Farmacocinético
47 Ac=[-(k10+k12+k13)   k21*V2/V1   k31*V3/V1   0
48      k12*V1/V2       -k21         0         0
49      k13*V1/V3       0           -k31        0
50      ke0             0           0          -ke0];
51 Bc=[1/V1
52     0
53     0
54     0];
55 M=[-(k10+k12+k13)   k21*V2/V1   k31*V3/V1   1
56      k12*V1/V2       -k21         0         0
57      k13*V1/V3       0           -k31        0
58      ke0             0           0         0 ];
59
60 N= -[0
61      0
62      0
63      -ke0*Ce      ];
64
65 C = (inv(M)*N);
66 Cc=[C(1:3); Ce ]';
67 Un = C(4);
68 Dc=0;
69
70 % Dcretización del modelo en variables de estado
71 sysc = ss(Ac,Bc,Cc,Dc);
72 sysd = c2d(sysc,ts);
73
74 A = sysd.A;
75 B = sysd.B;
76 C = sysd.C;
77 D = sysd.D;

```

```

1 function [pp_muestras_bolo]= simula_bolo_Inverso(pps,u)
2
3 file= 'sesion_manual_m_teresa_febles_dguez';
4
5 eval(file)
6 datossesion='m_teresa_feb';
7
8
9
10 WT = eval([datossesion '.paciente.peso']);
11 HT = eval([datossesion '.paciente.altura']);
12 Age = eval([datossesion '.paciente.edad']);
13 if eval([datossesion '.paciente.sexo'])=='m',
14     Gender = 1;
15 else
16     Gender = 0;
17 end
18
19     muestras = eval([datossesion '.muestras(:,1)']);
20     ps = eval([datossesion '.muestras(:,12)']);
21     ac = eval([datossesion '.muestras(:,13)']);
22     vel_remi = eval([datossesion '.muestras(:,15)']);
23
24 % fijamos periodo de muestreo
25 T = 5/60;
26 % FUNCION
27 [A,B,Un,C,D,picos_seg_0,picos_seg_min,ec50,g,V1] = InverseModel(WT,HT,Age,
    Gender,T,pps);
28 % VALORES PREDEFINIDOS
29 x_r(:,1)=C/1000; %ug/ml
30 % x_r(:,1)=[0 0 0 0];
31 % BUCLE
32 picos_seg(1)=pps;
33 for i=2:150
34     x_r(:,i)=A*x_r(:,i-1)+B*(u)*WT/V1/60; % Le pasamos la velocidad de
    infusión total de remifentanil, en ug/ml/min
35     picos_seg(i)=picos_seg_0+((picos_seg_min-picos_seg_0)*(x_r(4,i)^g))/(x_r
    (4,i)^g+(ec50/1000)^g); % Obtenemos como salida los picos/seg
36 end
37 pp_muestras_bolo=picos_seg(end);
38
39
40 % REPRESENTACIÓN GRÁFICA DEL PARÁMETRO PICOS POR SEGUNDO DE CONDUCTANCIA DE
    LA PIEL
41 figure(2);
42 subplot(3,1,1);
43 plot(muestras((length(muestras))),picos_seg(1,:),muestras,ps);title('

```



```

        Picos por segundo de conductancia de la piel', 'FontWeight', 'bold');
        xlabel('muestras', 'FontSize', 9); ylabel('picos/seg', 'FontSize', 9);
44 legend('Minto', 'Real')
45 grid;
46 % REPRESENTACIÓN GRÁFICA DEL LA VELOCIDAD DE INFUSIÓN DE REMIFENTANIL
47 subplot (3,1,2)
48 grid;
49 plot(muestras((length(muestras))), vel_remi_basal+vel_remi((mini+1:length(
        muestras)))); title('Velocidad de infusión de Remifentanil', 'FontWeight',
        'bold'); xlabel('muestras', 'FontSize', 9); ylabel('ug/ml/min', 'FontSize',
        9); grid;
50 % REPRESENTACIÓN GRÁFICA DEL LA CONCENTRACIÓN SITIO EFECTO DE REMIFENTANIL
51 xm_ngml=x_r*1000; % Concentración sitio efecto de remi, en ng/ml
52 subplot (3,1,3)
53 plot(muestras((mini+1:length(muestras))), xm_ngml(4, :)); title('
        Concentración Sitio Efecto de Remifentanil', 'FontWeight', 'bold'); xlabel
        ('muestras', 'FontSize', 9); ylabel('ng/ml', 'FontSize', 9);
54 grid;
55 end

```

```

1
2 function [Q,N,ponderacion,vel]= qlearning ()
3
4 gamma = 0.3;      %Factor de descuento
5 alpha = 0.2;     %Velocidad de aprendizaje
6 epsilon = 0.9;   %Probabilidad de exploración
7 % estados
8 state = [0,0.1,0.2,0.3,0.4,0.5,0.6,0.7];
9 % acciones
10 action = [0,0.05,0.1,0.2,0.25,0.3];
11 %matriz inicial
12 Q = zeros(length(state),length(action));
13 N = zeros(length(state),length(action));
14 K =1000;        %iteraciones
15 state_idx = 8;  %estado inicial
16
17 %Bucle principal del algoritmo
18 for k = 1:K
19     disp(['iteration: ' num2str(k)]); %genera un string de k
20     r=rand; %número aleatorio entre 0 y 1
21     x=sum(r>=cumsum([0, 1-epsilon, epsilon]));
22     if (x == 1)
23         [~,umax]=max(Q(state_idx,:));
24         current_action = action(umax); %obtención de la acción según el
           valor máximo de Q para la fila explorada
25     else
26         current_action=datasample(action,1); %elección de la acción de
           manera aleatoria
27     end
28
29     action_idx = find(action==current_action);%identificador de la acción
           escogida
30     %A partir del estado inicial y de la acción correspondiente se obtiene
           el estado siguiente y la recompensa asociada
31     [next_state,next_reward,ponderacion(k)] = model(state(state_idx),action
           (action_idx));
32     next_state_idx = find(state==next_state); %identificador del estado
33     %impresión de los resultados en cada iteración
34     disp(['current state : ' num2str(state(state_idx)) ' next state : '
           num2str(state(next_state_idx)) ' taken action : ' num2str(action(
           action_idx))]);
35     disp([' next reward : ' num2str(next_reward)]);
36     %Actualización de la matriz Q a partir de la ecuación que rige al Q-
           learning
37     Q(state_idx,action_idx) = Q(state_idx,action_idx) + alpha * (
           next_reward + gamma* max(Q(next_state_idx,:)) - Q(state_idx,
           action_idx));

```

```

38     %Número de veces que explora
39     N(state_idx,action_idx) = N(state_idx,action_idx) + 1;
40     %Con el estado siguiente definido, se elige una nueva acción
41     if ( next_state_idx <= 8 && next_state_idx >= 1)
42         state_idx = next_state_idx;
43         %Si los picos/seg son mayores que 0.7, se parte de un estado aleatorio
44     elseif (next_state_idx > 8)
45         state_idx = datasample(2:length(state),1);
46     end
47     disp(Q); %display Q in each level
48     %Se obtiene la velocidad correspondiente a los valores óptimos de la
49     matriz Q
50     for (i=1:length(state))
51         [~,umax]=max(Q(i,:));
52         u(i)=umax;
53         vel =action(u);
54     end
55 end
56 %Esta función es empleada con el propósito de obtener el estado siguiente,
57 la recompensa siguiente y la ponderación, usando el estado y la acción
58 asociada
59 function [pico_seg_siguiente,r,ponderacion] = model(pps,u)
60     [pp_muestras_bolo]= simula_bolo_Inverso(pps,u); %Hallazgo de los picos/
61     seg correspondientes a los picos/seg iniciales y a la acción tomada
62     pps_muestras= pp_muestras_bolo;
63     pps_muestras_redondeo =floor(pps_muestras*10)/10; %redondeo de los
64     picos/seg a la baja
65     %Logro del pico segundo siguiente
66     if (pps_muestras <= 0.7 && pps_muestras >= 0)
67         pico_seg_siguiente=pps_muestras_redondeo;
68     elseif (pps_muestras >0.7)
69         pico_seg_siguiente=0.7;
70     end
71     %variación de los picos/seg a partir del estado inicial y el estado
72     siguiente
73     var_pps = pico_seg_siguiente-pps;
74     %Constantes que multiplican a las variables asociadas a la ponderación
75     beta=0.8;
76     delta=0.2;
77     %Se considera más importante premiar la variación de picos/seg
78     ponderacion=(beta*var_pps+(delta*(pico_seg_siguiente)));
79     %Recompensas asociadas a la ponderación dada
80     if (( ponderacion <= -0.3))
81         r=80; %caso óptimo
82     elseif (( ponderacion > -0.3 && ponderacion <= -0.1))
83         r=60; %caso medio

```

```
79     %Caso particulara ha tener en cuenta:
80     elseif ( picos_seg_siguiete ==0 || picos_seg_siguiete <=0.1)
81         r=55; %Se recompensa de diferente manera el llegar al objetivo
82     elseif(( ponderacion> -0.1 && ponderacion <=0.4))
83         r=20; %peor caso
84     else
85         r=0; %no consecución del objetivo ,mínima recompensa
86     end
87
88
89 end
```

```

1
2 function Interfaz_comunicacion
3     % Creamos la comunicación serial para la recepción y transmisión – COM1
4     s1 = serial('COM1');
5
6     % Configuramos los parámetros de la comunicación serial y el callback a
7     la función "leerSerial"
8     set(s1, 'BaudRate',19200, 'DataBits',8, 'Parity', 'even', 'StopBits',1, '
9         TimerPeriod',0.1, 'TimerFcn', @leerSerial, 'ReadAsyncMode', 'continuous
10        ');
11
12     % Definimos el buffer de recepción
13     buffer_rec = [];
14
15     % Definimos el paquete de inicio de transmisión y el paquete base de
16     transmisión
17     paquete_inicio_trans=[85 170 1 119 1 0 0 0 3 0 1 0 87 95 102 153];
18     primer_paquete_trans=[85 170 1 119 1 0 0 0 5 0 2 0 157 45 102 153];
19
20     % Definimos las variables donde se almacenan los valores de
21     conductancia filtrados
22     datos_muestras=[];
23
24     % DEFINIMOS UN FILTRO BUTTERWORTH PASA BAJA, DE ORDEN 5 y Wn = 0.11
25     [B,A] = butter(5,0.11, 'low');
26     ent_filtro=[];
27     sal_filtro=[];
28
29     % Definimos una variable booleana para indicar que llega la el paquete
30     inicio transmisión
31     primera_muestra = true;
32
33     % Definimos una variable para guardar la fecha de llegada del primer
34     paquete
35     fecha_inicio = 0;
36
37     % Definimos el número de paquetes y el tamaño de datos de cada paquete
38     [num_paq long_paq] = size (datos_muestras);
39
40     % Acotamos la variable long_paq para quedarnos sólo con 30 muestras de
41     120
42     long_paq = fix (long_paq / 4);
43
44     % Definimos una variable para representar las muestras convertidas al
45     formato numérico
46     valores = zeros (num_paq * long_paq, 1);
47
48

```

```

39     % Definimos dos gráficas para representar los valores de conductancia
40     figure(1)
41     % Gráfica de valores de conductancia de la piel en una ventana de 15
42     segundos
43     subplot (1,2,1)
44     grafica = gca;
45     set(grafica, 'Position',[0.15 0.15 0.3347 0.7550])
46     title(grafica, 'Valores de conductancia de la piel en una ventana de 15
47     segundos');
48     ylabel(grafica, 'Conductancia de la piel (uS)')
49     % Gráfica de valores de conductancia de la piel desde el instante
50     inicial
51     subplot (1,2,2)
52     grafica2 = gca;
53     % title(grafica2, 'Valores de conductancia de la piel desde el instante
54     inicial ');
55     % ylabel(grafica2, 'Conductancia de la piel (uS)')
56     % Ajustamos el tamaño de la figura 1 al tamaño de la resolución de pantalla
57     set (1, 'Position', get(0, 'ScreenSize'));
58
59     % Definimos una variable booleana para indicar si la gráfica está fija
60     o se mueve
61     posicion_fija = false;
62     % Definimos una variable booleana para indicar el arranque del sistema
63     y su parada
64     orden_arranque = true;
65
66     % Creamos los cuadros de texto para indicar los nombres de los
67     parámetros a representar en tiempo real
68     text_picos = uicontrol('style','text','string','Picos/seg','position
69    ',[20 600 120 20], 'FontSize',10, 'FontWeight','bold');
70     uicontrol('style','text','string','Promedio de picos','position',[20
71     400 120 20], 'FontSize',10);
72     uicontrol('style','text','string','Tiempo de subida','position',[20
73     300 120 20], 'FontSize',10);
74     uicontrol('style','text','string','Área grande picos','position',[20
75     200 120 20], 'FontSize',10);
76     uicontrol('style','text','string','Área pequeña picos','position'
77    ',[20 100 120 20], 'FontSize',10);
78     text_area = uicontrol('style','text','string','Área bajo la curva','
79     position',[20 500 120 20], 'FontSize',10, 'FontWeight','bold');
80
81     % Creamos la barra de desplazamiento de la gráfica, así como los botones
82     para ir al final y al principio de la ejecución
83     barra_desp = uicontrol('style','slider','position',[185 50 490 30], 'Min
84     ',0, 'Max',1, 'SliderStep',[0 1], 'Callback',@pulsa_barra);
85     bot_inicio = uicontrol('style','pushbutton','string','I', 'FontSize',13,

```

```

    'position',[150 50 30 30], 'Callback', @ir_a_principio);
71 bot_fin = uicontrol('style','pushbutton','string','F','FontSize',13,'
    position',[680 50 30 30], 'Callback', @ir_a_final);
72
73 % Creamos los botones para el arranque y la parada de la ejecución del
    programa
74 bot_arranque = uicontrol('style','pushbutton','string','INICIO','
    position',[40 680 80 30], 'Callback', @arranque);
75 bot_parada = uicontrol('style','pushbutton','string','PARADA','position
   ',[40 640 80 30], 'Callback', @parada);
76
77 % Creamos los cuadros de texto donde se mostrarán los parámetros
    calculados en tiempo real
78 texto_picos = uicontrol('style','text','position',[20 570 120 20], '
    FontSize',10, 'FontWeight','bold');
79 texto_area = uicontrol('style','text','position',[20 470 120 20], '
    FontSize',10, 'FontWeight','bold');
80 texto_prom_picos = uicontrol('style','text','position',[20 370 120
    20], 'FontSize',10);
81 texto_tiempo_sub = uicontrol('style','text','position',[20 270 120
    20], 'FontSize',10);
82 texto_area_grande = uicontrol('style','text','position',[20 170 120
    20], 'FontSize',10);
83 texto_area_pequena = uicontrol('style','text','position',[20 70 120
    20], 'FontSize',10);
84
85 % Normalizamos las unidades de la fuente utilizada para representar los
    parámetros en tiempo real
86 set ([texto_picos texto_area barra_desp texto_picos texto_area
    texto_prom_picos texto_tiempo_sub texto_area_grande
    texto_area_pequena] , 'FontUnits', 'normalized');
87
88 % Abrimos el puerto de comunicación serial
89 fopen(s1);
90
91 % Q-learning
92 uvector = zeros(30,1);
93 vel=[0.3, 0.25, 0.3, 0.25, 0.3, 0.2, 0.25, 0.2];
94
95
96 % FUNCION PARA LEER PAQUETES DEL PUERTO SERIE, PROCESARLOS Y ENVIAR
    COMANDOS
97 function leerSerial (s1, e)
98     % Si se ha pulsado el botón que activa la variable para arrancar el
    sistema, ejecuto la función
99     if orden_arranque
100         tam = s1.BytesAvailable;

```

```

101 % Si hay bytes disponibles en la comunicacion
102 if tam > 0
103     % Leemos esos bytes recibidos por puerto serie
104     ecoSc = fread (s1, tam);
105     % Los vamos almacenando en el buffer de paquetes recibidos
106     buffer_rec = [buffer_rec ecoSc'];
107     % Llamamos a la función para identificar los paquetes
108     recibidos
109     paquetes = busca_paquetes_rec;
110
111 % BUCLE PARA OBTENER LOS PAQUETES RECIBIDOS EN TIEMPO REAL
112 % Y MANTENER LA COMUNICACIÓN
113 for i = 1:length(paquetes)
114     pa = paquetes{i}; % Metemos cada paquete recibido en
115     una variable
116
117     % ENVÍO DE COMANDOS AL EQUIPO PARA MANTENER LA
118     COMUNICACIÓN
119     switch pa(9)
120         % Si el byte N° 9 de un paquete recibido es '1' (
121         contador)
122         case 1
123             disp ('Comunicacion iniciada'); % Mostramos por
124             pantalla el inicio de la comunicación
125             % Enviamos el comando con el paquete de inicio
126             de la transmisión
127             fwrite (s1, paquete_inicio_trans, 'async');
128             % Si el byte N° 9 de un paquete recibido es
129             '2'(contador)
130         case 2
131             % Enviamos el comando con el paquete base de la
132             transmisión
133             fwrite (s1, primer_paquete_trans, 'async');
134         end
135
136     % PROCESADO DE CADA UNO DE LOS PAQUETES RECIBIDOS
137     % Si la longitud del paquete es mayor de 16 bytes
138     if length(pa) > 16
139         % Si recibimos el paquete de inicio de transmisión
140         if primera_muestra
141             % Guardamos la fecha de recepción del paquete
142             de inicio transimisión, en el formato "now"
143             fecha_inicio = now;
144             % Ponemos a cero la variable de primera
145             transmisión
146             primera_muestra = false;
147         end
148     end

```



```

137
138 % Llamamos a la función "extraer paquete" para
      obtener los valores numéricos de los datos
      contenidos en los paquetes largos
139 valores = extraer_paquete(pa(13:132)');
140
141 % FILTRADO DE LOS VALORES NUMÉRICOS DE CONDUCTANCIA
142 for i=1:length(valores)
143     valores_fil(i)=filtrado(B,A, valores(i)); %
      Aplicamos el filtro butterworth pasa-baja
144 end
145
146 % ALMACENAJE DE LAS MUESTRAS DE VALORES DE
      CONDUCTANCIA
147 % Guardamos en un array los 30 valores ya filtrados
      que llegan en cada paquete
148 datos_muestras = [datos_muestras ; valores_fil'];
149
150
151 % MODIFICACIÓN DEL INICIO DE LA VENTANA PARA EL
      CÁLCULO DE LOS PARÁMETROS EN TIEMPO REAL
152 % Fijo el tamaño de la ventana de tiempos en 15
      segundos, para cada 65 muestras/seg
153 tam_vent = 975;
154 % Condición para poder calcular los parámetros en
      los primeros 15 segundos
155 if tam_vent > length(datos_muestras)
156     tam_vent = length(datos_muestras);
157 end
158
159
160 % REPRESENTACIÓN GRÁFICA DE CONDUCTANCIA EN TIEMPO
      REAL
161 % REPRESENTACIÓN CON ESCALA DE TIEMPOS MÓVIL
162 % Si la longitud de "datos_muestras" es menor de 65
      muestras por 15 segundos
163 if length(datos_muestras) < 15*65
164     % El inicio de la ventana de tiempos será la
      posición final menos 15 seg
165     ini_ven = length(datos_muestras)-tam_vent+1;
166     % El vector de tiempos irá desde 0 hasta la
      muestra 65 del segundo 15
167     t = 0:(length(datos_muestras)-1);
168     datos_grafica = datos_muestras;
169     plot (grafica, t', datos_muestras, 'b')
170     title(grafica, 'Valores de conductancia de la
      piel en una ventana de 15 segundos');

```

```

171         ylabel(grafica, 'Conductancia de la piel (uS)');
172     % Si la longitud de "datos muestras" es mayor de 65
        muestras por 15 segundos
173     else
174         % Fijo el máximo de la barra desplazadora, el
            paso mínimo y el paso máximo
175     set (barra_desp, 'Max', length(datos_muestras)-1,
        'SliderStep', [2*65/length(datos_muestras),
            15*65/length(datos_muestras)]);
176
177     % Si se ha actuado sobre la barra desplazadora,
        el botón de ir a inicio, o el de ir a final
178     if posicion_fija
179         pos_barra = round(get(barra_desp, 'value'));
            % Calculo el valor de la posición de
                la barra desplazadora
180         % Fijamos el inicio de ventana en el valor
            de posición de la barra desplazadora
181         if pos_barra > 0
182             ini_ven = pos_barra;
183         else
184             ini_ven = pos_barra+1; % Si vamos al
                comienzo de representación (
                    pos_barra=0) hay que sumar una
                    muestra
185         end
186         final_barra=pos_barra+15*65-1;
            % La posición final de
                la barra será aquella pasada los 15
                segundos
187         % Comprobación para seguir ejecutando el
            programa aunque se avance a los últimos
            valores que se están mostrando
188         if final_barra > (length(datos_muestras)-1)
189             final_barra = length(datos_muestras)-1;
190             ini_ven = length(datos_muestras)-
                tam_vent+1; % En ese caso el inicio
                    de ventana no depende de la
                    posición de la barra
191         end
192         t = pos_barra : final_barra;
            % El vector de tiempos
                irá de la posición actual de la barra
                hasta la final
193     % Si no se ha actuado sobre la barra
        desplazadora, el botón de ir a inicio, o el
        de ir a final

```

```

194         else
195             ini_ven = length(datos_muestras)-tam_vent
                +1;    %El inicio de la ventana de
                tiempos será la posición final menos 15
                seg
196             t = (length(datos_muestras)-15*65):(length(
                datos_muestras)-1); %El vector de
                tiempo abarcará los últimos 15 segundos
197             set(barra_desp, 'value', t(1));
                % Colocamos la barra
                desplazadora al inicio de la ventana de
                tiempos
198         end
199
200         datos_grafica = datos_muestras(t+1);
201         plot(grafica, t, datos_muestras(t+1), 'b');
202         title(grafica, 'Valores de conductancia de la
                piel en una ventana de 15 segundos');
203         ylabel(grafica, 'Conductancia de la piel (uS)');
204
205     end
206
207     %Llamamos a la función para ajustar los ejes y
                definir la escala de tiempos en el eje de
                absisas
208     marcas_tiempo(t, grafica, datos_grafica)
209
210     %CÁLCULO Y REPRESENTACIÓN DE LOS PARÁMETROS
                CARACTERÍSTICOS EN TIEMPO REAL
211     %Llamamos a la función "parámetros" para obtener
                los parámetros de la señal de conductancia
212     [picos_seg, promedio_picos, tiempo_subida, area_grande
                , area_pequena, area_bajo_curva, picos]=Parametros(
                datos_muestras, ini_ven, tam_vent)
213
214     pps = picos_seg(end);
215     idx=min(ceil(pps*10), 7);
216     u=vel(idx+1);
217     uvector=[uvector ; u*ones(30,1)];
218
219     %MOSTRARLO A TIEMPO REAL
220     %REPRESENTACIÓN VELOCIDAD
221
222     plot (grafica2, uvector, 'b')
223     title (grafica2, 'Velocidad de infusión de
                Remifentanil');
224     ylabel (grafica2, 'velocidad ng/ml/min');

```

```

225         marcas_tiempo(t, grafica2, uvector) %comentar 271
           hasta 278 (MIRAR LINEA 166 para marcas_tiempo
           )
226
227 hold(grafica, 'on');
228 % Representación gráfica de los picos (máximos y
           mínimos), en la gráfica con escala de tiempos
           móvil
229 plot (grafica, picos(:,1)-1, datos_muestras(picos
           (:,1)), 'bs', 'MarkerFaceColor', 'b', '
           MarkerSize', 4); % Minimios
230 plot (grafica, picos(:,2)-1, datos_muestras(picos
           (:,2)), 'rs', 'MarkerFaceColor', 'r', '
           MarkerSize', 4); % Maximos
231
232 % Representación en tiempo real del parámetro "
           picos por segundo", variando su color de fondo
233 if picos_seg <= 0.07
234     set (texto_picos, 'string', picos_seg, '
           BackgroundColor', 'w');
235 elseif (picos_seg >= 0.07) && (picos_seg <= 0.21)
236     set (texto_picos, 'string', picos_seg, '
           BackgroundColor', [0.956,0.988,0.427]);
237 elseif (picos_seg >= 0.21) && (picos_seg <= 0.27)
238     set (texto_picos, 'string', picos_seg, '
           BackgroundColor', [1,0.9,0.1]);
239 elseif (picos_seg >= 0.27) && (picos_seg <= 0.4)
240     set (texto_picos, 'string', picos_seg, '
           BackgroundColor', [1,0.492,0.151]);
241 elseif picos_seg >= 0.4;
242     set (texto_picos, 'string', picos_seg, '
           BackgroundColor', 'r');
243 end
244
245 % Representación en tiempo real del parámetro "area
           bajo la curva", variando su color de fondo
246 if area_bajo_curva <= 2
247     set (texto_area, 'string', area_bajo_curva, '
           BackgroundColor', 'w');
248 elseif (area_bajo_curva >= 2) && (area_bajo_curva
           <= 5)
249     set (texto_area, 'string', area_bajo_curva, '
           BackgroundColor', 'c');
250 elseif area_bajo_curva >= 5;
251     set (texto_area, 'string', area_bajo_curva, '
           BackgroundColor', 'b');
252 end

```

```

253
254         % Representación del resto de parámetros
                característicos en tiempo real
255         set (texto_prom_picos, 'string', promedio_picos);
256         set (texto_tiempo_sub, 'string', tiempo_subida);
257         set (texto_area_grande, 'string', area_grande);
258         set (texto_area_pequena, 'string', area_pequena);
259
260         hold(grafica, 'off');
261
262         end
263     end
264 end
265 end
266 end
267
268 %FUNCION PARA IDENTIFICAR LOS PAQUETES RECIBIDOS (LARGOS Y CORTOS)
269 function paquetes = busca_paquetes_rec
270     % Creamos un "cell" para almacenar los paquetes recibidos
271     paquetes = {};
272     % Fijamos el inicio de un paquete buscando los cinco primeros
                caracteres del mismo
273     pos_inicial = strfind (buffer_rec, [85 170 1 119 0 1]);
274     % Inicializamos la posición final de un paquete
275     fin = 0;
276     % Cada vez que se reconozca un paquete por su secuencia inicial de
                caracteres
277     for ini = pos_inicial
278         % Si en el buffer de recepción hay un paquete (hay más de 6
                caracteres)
279         if length (buffer_rec) > (ini + 5)
280             lon = buffer_rec (ini+6) + 16; % Fijo la longitud de un
                paquete corto
281             if length (buffer_rec) >= (ini + lon - 1) % Si la longitud
                de un paquete del buffer es mayor que la de uno corto
282                 fin = ini + lon - 1; % Recalculo el
                valor del fin del paquete (paquete largo)
283                 paquetes{end+1} = buffer_rec (ini:fin); % Meto en el "
                cell" de paquetes los caracteres del paquete
                reconocido (de inicio a fin)
284             end
285         end
286     end
287     % Si se ha encontrado un paquete largo (su longitud es mayor que la
                de uno corto)
288     if fin > 0
289         buffer_rec = buffer_rec ((fin+1):end); % El buffer de recepción

```

```

                abarcará todos los caracteres de que consta un paquete
                largo
290     end
291 end
292
293 %FUNCION PARA EXTRAER LAS MUESTRAS DE UN PAQUETE Y CONVERTIRLAS AL
        VALOR NUMÉRICO
294 function valores = extraer_paquete (m)
295     % Multiplicamos el vector de 120 muestras (de 30 en 30) por los
        pesos de los bytes
296     valores = [m(1:4:120) m(2:4:120) m(3:4:120) m(4:4:120)] * [1 2^8
        2^16 2^24]';
297     % Realizamos el complemento a 2 de las muestras a procesar
298     neg = valores >= 2^31;
299     valores(neg) = valores(neg) - 2^32;
300     % Hacemos la conversión de las muestras al formato numérico
        escalado), según el rango (2285)
301     valores = -2285/2^32 * valores;
302 end
303
304 %FUNCION PARA AJUSTAR LA ESCALA EN EL EJE Y, Y PARA CREAR LAS
        MARCAS DE TIEMPO EN EL EJE X
305 function marcas_tiempo(t, grafica, datos)
306     % Calculamos el maximo y el minimo del conjunto de datos a
        representar
307     maximo = max(datos);
308     minimo = min(datos);
309
310     % Definimos la escala del eje x como un vector del minuto anterior
        hasta el próximo minuto, dividido en 8 intervalos
311     eje_x = t(1):(t(end)-t(1))/3:t(end);
312     set(grafica, 'xtick', eje_x);
313
314     % Definimos un "cell" para guardar las marcas de tiempo en formato
        'HH:MM:SS' para las 65 muestras por segundo
315     marcas = {};
316
317     % Un bucle le suma a la fecha inicial segundos en los que van
        llegando las 65 muestras
318     for i = 1:length(eje_x)
319         marcas{end+1} = datestr (datenum ([0 0 0 0 0 round(eje_x(i)/65)
        ]) + fecha_inicio, 'HH:MM:SS');
320     end
321
322     % Fijamos las marcas de tiempo a la gráfica en tiempo real
323     set (grafica, 'xticklabel', marcas)
324

```

```

325     % Capturamos la posición de los ejes de la gráfica
326     ejes = axis (grafica);
327
328     % Si el mínimo en el eje y es superior a -1, lo fijamos en ese
        valor
329     if ejes(3) > -1
330         ejes(3) = -1;
331     end
332
333     % Si el máximo en el eje y es inferior a 1, lo fijamos en ese valor
334     if ejes(4) < 1
335         ejes(4) = 1;
336     end
337
338     % Si el máximo en el eje y es superior a 1, recalculamos los ejes
        según el valor máximo y mínimo de conductancia
339     if ejes(4) >= 1
340         centro = (maximo+minimo)/2; % Calculamos valor central en la
            escala del eje y, según máxima y mínima conductancia
341         % Si la diferencia entre máximo y mínimo de conductancia es
            menor que 2, la escala en el eje y está entre el valor
            central y +-1
342         if maximo - minimo <= 2
343             ejes(3) = centro - 1;
344             ejes(4) = centro + 1;
345         % Si la diferencia entre máximo y mínimo de conductancia es
            mayor que 2, la escala en el eje se recalcula mutiplicando
            por un factor 2
346         else
347             ejes(4) = (maximo-centro)*2 + centro;
348             ejes(3) = (minimo-centro)*2 + centro;
349         end
350     end
351
352     % Fijamos la nueva escala del eje x y del eje y, según los valores
        que han sido recalculados
353     axis (grafica, [eje_x(1) eje_x(end) ejes(3:4)]);
354 end
355
356 % FUNCIÓN PARA FILTRAR LOS VALORES NUMÉRICOS DE CADA PAQUETE DE DATOS
    RECIBIDO
357 % Aplicando las ecuaciones que definen un filtro butterworth pasa-baja
358 function Y = filtrado (B,A,X)
359     lonb = length(B);
360     lona = length(A);
361     if (length(ent_filtro) >= lonb) && (length(sal_filtro) >= (lona-1))
362         ent_filtro = [X ; ent_filtro(1:(lonb-1))];

```

```

363         Y = B * ent_filtro - A(2:end)*sal_filtro;
364         sal_filtro = [Y ; sal_filtro(1:(lona-2))];
365     else
366         ent_filtro = [X ; ent_filtro];
367         if length(ent_filtro) > lonb
368             ent_filtro = ent_filtro(1:lonb);
369         end
370         Y = X;
371         sal_filtro = [Y ; sal_filtro];
372         if length(sal_filtro) > (lona-1)
373             sal_filtro = sal_filtro(1:(lona-1));
374         end
375     end
376 end
377
378 %FUNCIÓN PARA IR AL FINAL DE LA REPRESENTACIÓN GRÁFICA EN LA VENTANA
379 %MÓVIL
380 function ir_a_final(o,e)
381     posicion_fija = false; %Desactivamos la variable que perite
382     congelar la representación gráfica
383 end
384
385 %FUNCIÓN PARA IR AL INICIO DE LA REPRESENTACIÓN GRÁFICA EN LA VENTANA
386 %MÓVIL
387 function ir_a_principio(o,e)
388     posicion_fija = true; %Activamos la variable que perite
389     congelar la representación gráfica
390     set(barra_desp,'value',0); %Ponemos la barra de desplazamiento
391     en el inicio de la ventana de tiempos
392 end
393
394 %FUNCIÓN PARA PERMITIR EL DESPLAZAMIENTO DE LA GRÁFICA EN LA VENTANA
395 %MÓVIL
396 function pulsa_barra(o,e)
397     posicion_fija = true; %Activamos la variable que permite
398     congelar la representación gráfica
399 end
400
401 %FUNCIÓN PARA PERMITIR LA EJECUCIÓN DEL SISTEMA
402 function arranque(o,e)
403     orden_arranque = true; %Activamos la variable que permite
404     realizar la ejecución del sistema
405 end
406
407 %FUNCIÓN PARA REALIZAR LA PARADA DE LA EJECUCIÓN DEL SISTEMA
408 function parada(o,e)
409     orden_arranque = false; %Desactivamos la variable ejecución del

```



```

    sistema
402 stopasync (s1);
403 delete (instrfindall); %Borramos el objeto puerto serie
404 fclose ('all'); %Cerramos todos los ficheros que hayan
    abiertos
405 clear classes; %Borramos todas las clases
406 end
407
408
409 end
```

```

1  function [picos_seg , promedio_picos , tiempo_subida , area_grande , area_pequena ,
    area_bajo_curva , picos]=Parametros(num, ini_ven , tam_vent)
2
3  maximos = []; % Posiciones donde hay máximos locales
4  minimos = []; % Posiciones donde hay mínimos locales
5  min_max = []; % Tabla de parejas [posición mínimo, posición máximo]
6  dif_num = []; % Diferencia entre un mínimo y su correspondiente máximo
7
8  % Establezco el instante de final de la ventana de tiempos
9  fin_ven = ini_ven+tam_vent-1;
10
11 % Encontramos la posición de cada mínimo y cada maximo en el vector de
    valores de conductancia
12 menores = num(1:(end-1)) < num(2:end);
13 mayores = num(1:(end-1)) > num(2:end);
14 minimos = find (mayores(1:(end-1)) & menores(2:end)) + 1;
15 maximos = find (menores(1:(end-1)) & mayores(2:end)) + 1;
16
17 if ~isempty(minimos) && ~isempty(maximos) % Si se ha encontrado algún
    maximo o mínimo
18     ini_max = 1; % Variable para identificar el primer elemento máximo
        del vector
19     if maximos(1) < minimos(1)
20         ini_max = ini_max + 1; % Si el primer máximo está antes del
            primer minimo lo descarto (me quedo con el posterior)
21     end
22     fin_min = length(minimos); % Variable para identificar el último
        elemento mínimo del vector
23     if minimos(end) > maximos(end)
24         fin_min = fin_min - 1; % Si el ultimo minimo está despues del
            ultimo maximo lo descarto (me quedo con el anterior)
25     end
26     min_max = [minimos(1:fin_min) maximos(ini_max:end)]; % Creo la
        matriz de minimos y maximos emparejados (por columnas)
27     dif_num = num(min_max(:,2)) - num(min_max(:,1)); % Calculo la
        diferencia entre minimo y maximo emparejado
28 end
29
30 % Creo una variable para almacenar las parejas mínimo - máximo con
    diferencia mayor que 0.005 uS, así como otra para su diferencia
31 picos = min_max(dif_num >= 0.2 , :);
32 dif_picos = dif_num(dif_num >= 0.2);
33
34 % CALCULO DE PICOS POR SEGUNDO
35 % Seleccionamos los mínimos y máximos que están dentro de la ventana de
    tiempos
36 min_dentro = (picos(:,1)>= ini_ven) & (picos(:,1)<=fin_ven);

```

```

37 max_dentro = (picos(:,2)>= ini_ven) & (picos(:,2)<=fin_ven);
38 picos_dentro = find (min_dentro | max_dentro); % Buscamos las
           posiciones de los picos en la ventana de tiempo
39 num_picos_dentro = length(picos_dentro);           % Calculamos el número
           de picos en la ventana de tiempo
40 picos_seg = round((num_picos_dentro / (tam_vent/65))*100)/100; % Los
           picos por segundo se halla dividiendo el n° de picos por el tiempo
           de la ventana
41
42 %CALCULO DEL PROMEDIO DE PICOS
43 if ~isempty(picos_dentro)
44     promedio_picos=mean(dif_picos(picos_dentro)); % El promedio de
           picos se halla calculando la media de la diferencia entre maximo
           y minimo
45     promedio_picos = round(promedio_picos*100)/100; % Redondeamos el
           resultado a dos decimales
46 else
47     promedio_picos = 0;
48 end
49
50 %CALCULO DEL TIEMPO DE SUBIDA
51 tiempo_subida = (num(fin_ven)-num(ini_ven))/(tam_vent/65); % El area
           pequeña es la diferencia entre valor inicial y final de conductancia
           , divido por el tamaño de ventana
52 tiempo_subida = round(tiempo_subida*100)/100; %
           Redondeamos el resultado a dos decimales
53
54 %CALCULO DEL AREA GRANDE DE PICOS
55 if ~isempty(picos_dentro)
56     min_primer_pico = picos(picos_dentro(1),1); % Escogemos el
           primer mínimo de los picos que tenemos dentro de la ventana de
           tiempos
57     valor_primer_pico = num(min_primer_pico);
58     inicio_area_grande = min_primer_pico; % La posición de
           ese primer pico será el instante de inicio del cálculo del área
           grande
59
60     if inicio_area_grande < ini_ven
61         inicio_area_grande = ini_ven; % Si ese primer
           pico está antes del inicio de ventana, el cálculo empieza al
           inicio de ventana
62     end
63
64     valores_encima = num(inicio_area_grande:(ini_ven+tam_vent-1)); %
           Escogemos todos los valores de cond. desde el inicio de calculo
           hasta fin de ventana
65     dif_valores = valores_encima-valor_primer_pico; %

```

```

        Calculo la diferencia entre los valores de calculo y el valor
        del primer pico hallado
66     area_grande = sum(dif_valores(dif_valores > 0))/65;           %
        Hago el sumatorio del vector de diferencias que son mayores de
        cero
67     area_grande = round(area_grande*100)/100;                   %
        Redondeamos el resultado a dos decimales

68
69     else
70         area_grande = 0;
71     end
72
73     %CALCULO DEL AREA PEQUEÑA DE PICOS
74     if ~isempty(picos_dentro)
75         picos_calc = picos(picos_dentro,1);   %Escogemos todos los mínimos
        del vector de picos en la ventana de tiempos
76         valores_picos = num(picos_calc);
77         areas_pequenas=[];                   % Inicializamos una matriz
        para almacenar todas las áreas pequeñas a encontrar
78
79         %Bucle for que recorre todos los mínimos en la ventana de tiempos
80         for i=1:(size(picos_calc,1)-1)
81             recta=[];                         % Inicializo la variable
        para definir una recta
82             inicio_area_peq = picos_calc(i);   % Calculo la posición de
        inicio del area pequeña (un mínimo)
83             fin_area_peq = picos_calc(i+1);    % Calculo la posición de
        fin del area pequeña (el siguiente mínimo)
84             if inicio_area_peq < ini_ven
85                 inicio_area_peq = ini_ven;    % Si el mínimo está antes
        de inicio de ventana, el area comienza en el inicio
        ventana
86             end
87             if fin_area_peq > fin_ven
88                 fin_area_peq = fin_ven;       % Si el mínimo consecutivo
        está después del fin ventana, el area termina en el fin
        de ventana
89             end
90
91             m = (valores_picos(i+1)-valores_picos(i))/(picos_calc(i+1)-
        picos_calc(i));   % Calculo la pendiente de la recta entre
        dos mínimos seguidos
92             for j = inicio_area_peq:1:picos_calc(i+1)
93                 recta=[recta (m*(j-picos_calc(i))+valores_picos(i) ); %
        Determino la recta que unirá los dos mínimos seguidos
94             end
95

```

```

96     valores_encima_2= num(inicio_area_peq:fin_area_peq);    % Escojo
        los valores de conductancia comprendidos entre los dos
        mínimos
97     dif_valores = valores_encima_2-(recta');                %
        Calculo la diferencia entre los valores de conductancia y
        los valores de la recta trazada
98     area_pequena = sum(dif_valores(dif_valores > 0))/65;    % Hago
        el sumatorio del vector de diferencias que son mayores de
        cero
99     area_pequena = round(area_pequena*100)/100;            %
        Redondeo el resultado a dos decimales
100    areas_pequenas =[areas_pequenas area_pequena];         % Meto
        en un vector fila cada una de las areas pequeñas calculadas
101
102    end
103
104    area_pequena=sum(areas_pequenas);    % Sumamos todas las areas
        pequeñas calculadas en la ventana de tiempo
105
106    else
107        area_pequena = 0;
108    end
109
110    %CALCULO DEL AREA BAJO LA CURVA
111    area_bajo_curva = max(area_grande,area_pequena);    % El área bajo la
        curva es el máximo entre el área pequeña y el área grande de picos
112
113    end

```

```
1 function limpiar
2     delete (instrfindall);
3     fclose ('all');
4     % close all;
5     clear classes;
6 end
```