



Diseño de una plataforma de prueba de sensores virtuales para el sistema glucosa-insulina de pacientes UCI usando la técnica HIL

Design of a virtual sensor test platform for ICU patient glucose-insulin system using the HIL technique

Cristian Hernández González ^a, Dahiana Vélez Castaño ^b, & Jhon Alexander Isaza ^c

^a Instituto Tecnológico Metropolitano, Medellín, Colombia, cristianhernandez181700@correo.itm.edu.co

^b Instituto Tecnológico Metropolitano, Medellín, Colombia, dahianavelez178843@correo.itm.edu.co

^c Instituto Tecnológico Metropolitano, Medellín, Colombia, jhonisaza@itm.edu.co

Recibido: noviembre 9, 2018. Aceptado: diciembre 19, 2018.

Resumen

La simulación tradicional In Silico (simulación computacional) no permite recrear ambientes realistas. Herramientas como hardware-in-the-loop (HIL) permiten simular en tiempo real la respuesta de un sistema ante diferentes perturbaciones y situaciones en la regulación de glucosa e insulina en las que podría encontrarse un paciente en Unidad de Cuidados Intensivos (UCI), tales como hiperglucemias o hipoglucemias. En esta investigación, se pretende desarrollar una metodología para la implementación de la técnica de simulación HIL para desarrollar una plataforma de prueba de sensores virtuales para el sistema glucosa-insulina de pacientes en UCI usando la metodología HIL y estimadores de estado, para lo cual se estableció una comunicación que permitió someter el sistema a perturbaciones en tiempo real, además de desarrollar una interfaz que permite la manipulación de las principales características y parámetros tanto del modelo como de los sensores virtuales.

Palabras Clave: medición de glucosa, unidad de cuidados intensivos, hardware in the loop, sistemas embebidos..

Abstract

The traditional simulation In Silico (computational simulation) does not allow to recreate realistic environments. Tools like hardware-in-the-loop (HIL) allow the real-time simulation of the response of a system to disturbances. These tools come in handy for the study of the variations in glucose regulation for a patient in the Intensive Care Unit, such as hyperglycemia or hypoglycemia. In this investigation, a methodology was developed using HIL simulation technique to create a test platform for virtual sensors for the system patients' glucose – insulin in UCI. Also, state estimation techniques were employed, requiring the use of a communication structure to submit the system to real-time disturbances. Moreover, the work involved the development of an interface for the manipulation of the principal characteristics and parameters of both the model and the virtual sensors.

Keywords: Glucose measurement, unit of intensive care, hardware in the loop, embedded systems.

1 INTRODUCCIÓN

La descompensación en los niveles de glucosa de los pacientes en Unidad de Cuidados Intensivos (UCI), está relacionada con un aumento de la mortalidad durante la enfermedad crítica. Además de estar estrechamente vinculada con el aumento de la probabilidad de desarrollar otras enfermedades como infecciones nosocomiales, polineuropatía, falla renal aguda, falla multi-orgánica, convulsiones, entre otras [1]. Por lo anterior, el control de los niveles de glucosa de pacientes en UCI está estrechamente relacionado con la mejora del estado de salud y disminución de la mortalidad. Por otra parte, para poder ejercer control es necesario primero conocer o medir la concentración de glucosa en plasma. Lo cual no es viable debido a que su medición es invasiva; por lo tanto, una forma de estimar la concentración de glucosa en plasma es usar un modelo matemático del sistema, una estructura de estimación de estado y la adquisición de algunas variables de mayor facilidad de medición [2]. Estos tres últimos elementos (modelo, estructura de estimación y datos), en conjunto con el hardware adecuado, pueden consolidarse como un sensor virtual.

Para la transferencia tecnológica de dichos sensores virtuales en seres vivos en la práctica, es primero necesario evaluar y garantizar su buen funcionamiento en ambientes simulados. La simulación tradicional In Silico (simulación computacional), no permite recrear ambientes realistas. Herramientas como *Hardware-In-the-Loop*

Citar como:

C. Hernández, D. Vélez, J. Isaza. "Diseño de una plataforma de prueba de sensores virtuales para el sistema glucosa-insulina de pacientes UCI usando la técnica HIL" Revista CINTEX, Vol. 23(2), pp. 61-75. 2018.

(HIL) permiten simular en tiempo real la respuesta de un sistema ante diferentes perturbaciones y situaciones en las que podría encontrarse un paciente en UCI (hiperglucemias, hipoglucemias). Este tipo de técnicas ha sido utilizado ampliamente en el desarrollo de dispositivos médicos y bioprocesos. En el 2007 la Universidad de Leeds, en conjunto con otras instituciones, desarrolló una prueba dentro de la cual se evaluó un dispositivo para asistencia cardiaca. Dicha prueba constaba de un modelo matemático realista del sistema cardiovascular y un prototipo del dispositivo de asistencia cardiaca [3]. El uso de este tipo de técnicas ha ido tomando fuerza en el área cardiaca debido a los limitantes que representa probar la interacción de los dispositivos médicos en vivo. En el 2009 se desarrolló un entorno de simulación para sistemas fisiológicos cardiovasculares, el cual relacionaba un modelo físico del corazón con un modelo matemático del sistema cardiovascular, que permitía evaluar el comportamiento realista de un corazón latiente a diferentes condiciones fisiológicas, evaluando así el dispositivo de asistencia cardiaca en una etapa previa a las evaluaciones clínicas y en animales [4]. Durante el 2015 se implementó HIL con el fin de evaluar la energía consumida por un marcapaso cardiaco cuando este se enfrenta a diferentes patologías [5]. Por lo anterior se demuestra que HIL constituye una herramienta de simulación que permite evaluar el rendimiento de sensores virtuales en ambientes más realistas para el sistema glucosa-insulina en pacientes en UCI. En esta investigación, se propone una metodología para implementación de la técnica HIL mediante una plataforma de prueba de sensores virtuales para el sistema glucosa-insulina de pacientes en cuidados intensivos. Dicha metodología se plantea a partir de un protocolo de cinco pasos para establecer un lazo de comunicación entre modelo y sensor virtual.

Este documento estará organizado de la siguiente manera: En la sección 2 se explica la metodología para la implementación de HIL; posteriormente, en la sección 3 se presenta el caso de estudio donde se aplicó la metodología propuesta. Luego, en la sección 4 se presenta el resultado de la validación de la metodología para HIL. Finalmente, en la sección 5 se presentan las conclusiones y trabajo futuro.

2 METODOLOGÍA PARA LA IMPLEMENTACIÓN DE HIL

HIL se ha constituido como una herramienta ampliamente usada para la simulación de diferentes sistemas. Este tipo de herramientas generan un acercamiento por medio de la simulación a un entorno realista debido a que permite evaluar el sistema bajo prueba a diferentes situaciones a las que podría verse sometido en un entorno real, permitiendo la detección de fallas en etapas tempranas de desarrollo [6]. Lo anterior posibilita tanto la evaluación de los algoritmos de control como de los instrumentos y partes del sistema (sensores, actuadores, tarjeta de control, entre otros) de una manera más realista que las simulaciones tradicionales ya que permite tener en cuenta aspectos como pérdidas de datos, mediante la recepción y envío de datos entre dos componentes (computador –computador, computador –tarjeta de control, computador –hardware real), como se describe en la Figura 1.

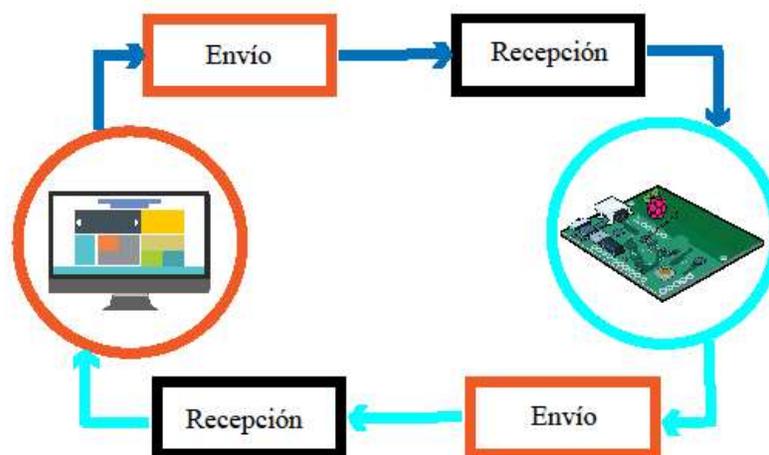


Figura 26. Envío y recepción de datos para la implementación HIL.

La Figura 2 presenta la metodología propuesta para validar sensores virtuales a partir de Hardware in the Loop (HIL). Dicha metodología se compone de 5 pasos. El primero corresponde a la elección del modelo y el estimador apropiado; la elección de ambos elementos (modelo y estimador) está relacionada con el tipo de aplicación. El paso siguiente es la simulación tradicional tanto del modelo como del estimador, en el tercer paso se realiza la selección del hardware con el cual se va a realizar la comunicación. En el cuarto paso se establece

comunicación entre el modelo y el estimador, y por último, se construye un ambiente dinámico que facilite las pruebas y su análisis.

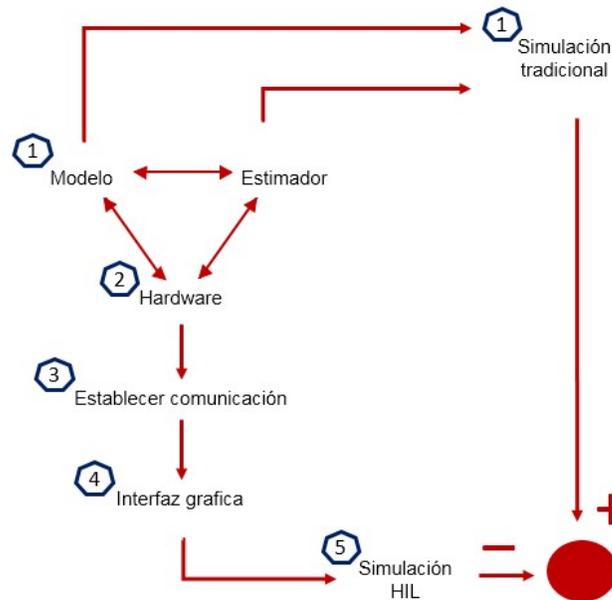


Figura 27. Metodología propuesta para validar sensores virtuales a partir de Hardware in the Loop (HIL)

2.1. Modelo

Los modelos dinámicos son una representación simplificada del sistema, a partir del cual es posible realizar análisis que en el sistema real están limitados, ya sea por el costo, el riesgo, o la incapacidad humana de experimentar. El modelo de un sistema es, en resumen, una representación matemática o simbólica de un proceso o un fenómeno, el cual permite responder interrogantes sin tener que recurrir a la intervención del sistema real [7]. Por lo anterior, contar con un modelo que represente el sistema que se desea intervenir, facilita su análisis y el desarrollo tecnológico. Sin embargo, existen en la literatura diferentes modelos que describen un mismo sistema; es por esta razón que para la elección del modelo adecuado es necesario conocer el sistema real, tener claro cuáles características de este son relevantes. Además, tener en cuenta el nivel de detalle respecto al sistema real que se desea en el modelo elegido, y en qué aplicación se va a usar el modelo seleccionado. En el caso de HIL para la validación de sensores virtuales, es necesario un modelo dinámico que permita evidenciar la evolución temporal del sistema. Además, el modelo debe ser controlable y observable para garantizar un buen diseño en los estimadores de estado, por otra parte, el modelo seleccionado debe ser un modelo validado que garantice una representación aproximada de la realidad.

Con el fin de verificar el cumplimiento de las características antes descritas es necesario realizar una simulación tradicional del modelo seleccionado, la cual debe ser comparada con otras dinámicas de simulaciones del mismo modelo presentes en la literatura, este tipo de simulaciones permiten un mejor reconocimiento del sistema, el cual será útil a la hora del diseño de un estimador de estados o un sensor virtual.

2.2. Estimador de estado

Los estimadores de estado constituyen una herramienta matemática que mediante el uso de un modelo matemático, la información disponible del sistema (mediciones) y una estructura de estimación, proporciona información en tiempo real del estado del sistema, lo cual generalmente es requerido para labores de monitoreo, supervisión y control de sistemas dinámicos [8], [9]. Los estimadores de estado pueden ser usados como suavizadores, filtros o predictores dependiendo de cómo se use la información disponible en el instante de tiempo actual; cuando un estimador de estado es implementado en una aplicación real como predictor, este pasa a denominarse sensor virtual, con la finalidad de proporcionar información no disponible y reducir el número de sensores físicos en un proceso determinado para así reducir costos [10].

Teniendo en cuenta lo anterior, para la implementación y posterior validación de un sensor virtual utilizando la herramienta HIL es necesario un conocimiento del modelo y del sistema real, debe conocerse que tipo de proceso representa, si es un proceso químico, bioquímico, físico; si el sistema es lineal o no lineal, la información del proceso que se encuentra disponible y de qué manera, entre otras características del modelo. Un conocimiento adecuado de la planta permitirá una mejor elección del estimador o sensor virtual. Para esta elección existen

metodologías como la propuesta por Denis Dochain [11]. En dicha investigación se realiza una comparación entre los diferentes estimadores, sus características y los tipos de sistemas o procesos en los cuales pueden ser utilizados. Luego de hacer la elección del estimador se debe hacer su respectiva programación, y evaluación de su funcionamiento, en una simulación clásica, esto permite verificar si la elección del estimador fue la adecuada.

2.3. Selección del hardware y software de simulación

La técnica de simulación HIL comprende, entre sus elementos principales, la necesidad de establecer una comunicación entre pares de elementos de hardware, con el fin de separar las partes de un sistema (en este caso el modelo o la planta real) del estimador o sensor virtual, para de esta manera recrear un ambiente realista. Para la elección del hardware se debe tener en cuenta el software en el cual se comprobó el funcionamiento del modelo y el estimador y la compatibilidad de los sistemas operativos. Debido a que para la implementación de HIL se debe hacer uso de hardware de bajo costo, este también se debe seleccionar de acuerdo con la complejidad del proceso, ya que dicho hardware debe tener la capacidad de procesar de forma rápida y fiable el algoritmo que representa el sistema bajo prueba. Por esta última razón, es de gran importancia tener presente características técnicas tales como la velocidad, el procesador, la memoria RAM, la cantidad de puertos analógicos y digitales (A/D) y los protocolos de comunicación que el hardware considerado puede soportar.

2.4. Establecimiento de la comunicación

Para recrear un ambiente realista mediante HIL, es necesario establecer un protocolo de comunicación entre dos elementos de hardware diferentes, donde se evidencie un envío y recepción de datos sincrónico. El protocolo de comunicación se debe establecer de acuerdo con las características de los dispositivos y a su capacidad de procesamiento, ya que estos pueden operar con diferentes protocolos de comunicación como Ethernet, USB o UART. Una vez se verifica la compatibilidad y se ha definido el protocolo de comunicación entre hardware y software, se debe hacer una prueba enviando y recibiendo un dato sencillo, para de esta manera comprobar la forma en que se está enviando y cómo se está recibiendo la información.

2.5. Construcción de la interfaz grafica

La interfaz gráfica recrear un ambiente dinámico que permite realizar pruebas de forma más interactiva y amigable ya que facilita el análisis y la validación del sistema bajo prueba. Para lo anterior la interfaz debe permitir variar las entradas del sistema generar perturbaciones, manipular variables de sintonía, además de facilitar la visualización en tiempo real de las variables de interés ya que HIL se plantea como una técnica de simulación entiempro real.

3 CASO DE ESTUDIO: HIL PARA ESTIMACIÓN DE GLUCOSA PLASMÁTICA DE PACIENTES EN UCI

3.1 Modelo de Chase para pacientes en UCI

Para este caso de estudio se hizo uso de un modelo matemático que representa el mecanismo homeostático de la glucosa de pacientes en UCI, basado en el modelo mínimo de Bergman [12]. El modelo conformado por las ecuaciones (1)-(6) se compone de seis ecuaciones diferenciales ordinarias que describen la evolución temporal de la glucosa en sangre (G_B en mmol/L), la evolución temporal de la glucosa en el intersticio abdominal (G_i en mmol/L), la concentración de insulina en el intersticio abdominal (Q en mU/L), la concentración de insulina en la sangre (I en mU/L), la cantidad de glucosa en el estómago (P_1 en mmol) (donde los carbohidratos se descomponen en moléculas más pequeñas) y finalmente la cantidad de glucosa en el intestino (P_2 en mmol) (donde las moléculas de glucosa son absorbidas y transportadas a la sangre), respectivamente [12].

$$\frac{dG_B}{dt} = -pG * G_B - Si * \frac{GB * Q}{1 + aG * q} + \frac{P + EGP_b - CNS}{V_G} \quad (1)$$

$$\frac{dG_i}{dt} = \beta_1 * G_B - \beta_2 * G_i \quad (2)$$

$$\frac{dQ}{dt} = nl(I - Q) - n_c * \frac{Q}{1 + aG * Q} \quad (3)$$

$$\frac{dI}{dt} = -n_k * I - n_L * \frac{I}{1 + aI * I} - n_I(I - Q) + \frac{u_{ex}}{V_I} + (1 - X_L) * \frac{u_{en}}{V_I} \quad (4)$$

$$\frac{dP_1}{dt} = -d_1 * P_1 + D \quad (5)$$

$$\frac{dP_2}{dt} = -\min(P_1, P_2, P_{max}) - d_1 * P_1 \quad (6)$$

Donde la variable P_1 que se encuentra definida en (7), corresponde a la aparición de glucosa en sangre a partir de la alimentación enteral, y u_{en} corresponde a la producción endógena de insulina por parte del páncreas.

$$P(t) = \min(d_2, P_2, P_{max}) + PN \quad (7)$$

$$u_{en} = K_1 * e^{-I(t)^{\frac{K_2}{K_3}}} \quad (8)$$

El modelo total presentado en las Ecuaciones (1-8) permitió representar por medio del software Matlab-Simulink la medición de glucosa de un paciente en UCI. El modelo tiene, además de los seis estados descritos en la Tabla 1, tres entradas U_{ex}, D, P que corresponden a la insulina exógena, la alimentación enteral y la alimentación parenteral, respectivamente.

Tabla 5. Estados y entradas del modelo matemático de Chase

Variable	Tipo	Unidades
B_G	Estado	mmol/L
G_I	Estado	mmol/L
Q	Estado	mmol/L
I	Estado	mmol/L
P_1	Estado	mmol/L
P_2	Estado	mmol/L
U_{ex}	Entrada	mU/min
D	Entrada	mmol/L
PN	Entrada	mmol/L

3.2 Filtro de Kalman y filtro de Kalman extendido

Partiendo de la metodología expuesta por Denis Dochain [11] para la elección adecuada del estimador, se eligió para este caso en específico un filtro de Kalman y un filtro de Kalman lineal debido a que permiten una estimación rápida basada en la predicción-corrección, además de ser un estimador versátil ya que es posible su implementación tanto en sistemas lineales como en sistemas no lineales. Además, ha sido ampliamente utilizado en bioprocesos [9]. Tanto el filtro de Kalman como el filtro de Kalman lineal fueron utilizados para la reconstrucción de la glucosa plasmática (B_G) a partir de la medición de la glucosa intersticial (G_I). En la literatura han sido utilizados diferentes estimadores con el fin de obtener una medición más cercana de la variable a controlar [8], [13]. Considerando el siguiente sistema no lineal en tiempo discreto como una representación del modelo anteriormente descrito,

$$x_{k+1} = f_k(x_k, u_k, w_k, d_k) \quad (9)$$

$$y_k = h_k(x_k, v_k) \quad (10)$$

Se pueden considerar como una estructura para el filtro de Kalman, las siguientes ecuaciones dentro de la cual se representan dos etapas:

1. Etapa de predicción: donde se calcula el estado estimado a priori (P_k^-) y la estimación de la covarianza del error (\hat{x}_k^-)

$$P_k^- = F_{k-1} P_k^+ F_{k-1}^T + Q_{e,k-1} \quad (11)$$

$$\hat{x}_k^- = f_{k-1}(\hat{x}_{k-1}^+, u_{k-1}) \quad (12)$$

2. Etapa de corrección: donde se calcula la ganancia de Kalman (K_k), la estimación a posteriori (P_k^+) y la covarianza del error asociada a la estimación posteriori (\hat{x}_k^+).

$$K = P_k H_k^T (H_k P_k^- H_k^T + R_{e,k})^{-1} \quad (13)$$

$$\hat{x}_k^+ = \hat{x}_k^- + K_k [y_k - h_k(\hat{x}_k^-)] \quad (14)$$

$$P_k^+ = (I - K_k H_k) P_k^- \quad (15)$$

Para aplicar el filtro de Kalman, se debe linealizar cada una de las ecuaciones del modelo descrito en las ecuaciones de (1) a (6) en el punto de operación, encontrando el jacobiano del sistema de la forma mostrada en la ecuación (16), dado que el filtro de Kalman está diseñado para sistemas lineales. El filtro de Kalman extendido también debe hacer una linealización del sistema, pero en este caso se debe hacer en cada punto de operación o en cada tiempo de muestreo, lo que permitirá tener más información de sistema y adaptarse mejor a sistemas no lineales; este desempeño se debe comprobar en una simulación de ambos filtros, comparándolos con la medición del modelo y evaluándolos por medio de indicadores de desempeño. Un diagrama del esquema de estimación de estados se presenta en la figura 3.

$$F_k = \frac{\partial f_k}{\partial x} \Big|_{\hat{x}_k^-, u_k} \quad H_k = \frac{\partial h_k}{\partial x} \Big|_{\hat{x}_k^-, u_k} \quad (16)$$

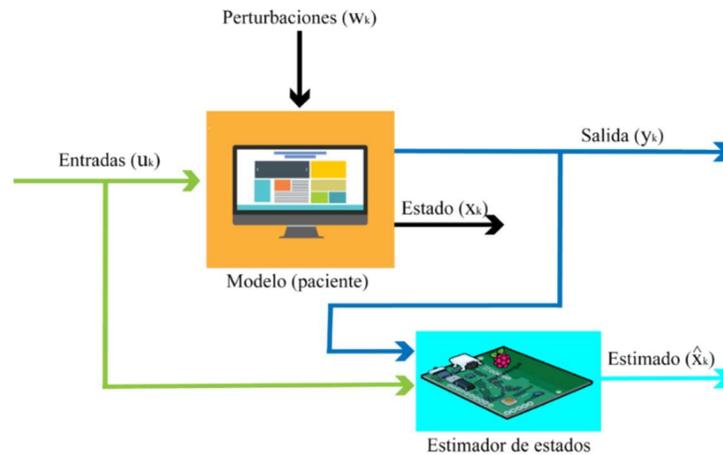


Figura 28. Esquema de estimación de estados

3.3 Selección del hardware

Para la selección del hardware en el que se programó el estimador y el modelo o paciente virtual, se tuvieron en cuenta las simulaciones clásicas antes realizadas, en las cuales se comprobó tanto el comportamiento del modelo como el rendimiento del estimador. El modelo del estimador fue programado en un PC por medio del software Matlab-Simulink, ya que es un lenguaje de programación potente para este tipo de aplicaciones. Para el segundo dispositivo de hardware, el cual debe ser de bajo costo, se hizo una comparación entre 3 tarjetas de desarrollo electrónico disponibles en el mercado, como se muestra en la Tabla 2.

Además de la tabla comparativa, se tomaron en cuenta diferentes ventajas y desventajas de los diferentes dispositivos.

3.3.1 Arduino:

- **Ventajas:** relativamente barato frente a los otros dos dispositivos, disponen de muchas variaciones de modelos de Arduino para elegir. Como es de bajo consumo, es ideal para aplicaciones de usos de larga

duración o incluso para uso con baterías. Debido a su popularidad conlleva a una gran facilidad de encontrar apoyo, documentación sobre proyectos particulares, tutoriales, además es compatible con el hardware y el software anteriormente implementados.

- **Desventajas:** el Arduino no tiene la capacidad de manejar diferentes procesos al mismo tiempo y tiene limitaciones en su programación ya que su interfaz es algo limitada. En cuanto al protocolo de comunicación solo cuenta la opción serial, aunque es una comunicación sincrónica no es muy estable, por lo generalmente es utilizado para proyectos de baja complejidad, debido a su procesador.

Tabla 6. Comparación entre hardware de bajo costo

Hardware	Arduino uno	Raspberry pi 3 B	BeagleBone Black
Procesador	AT Mega 328	ARMv8	AM335x
Velocidad	16 MHz	1,2 GHz	1 GHz
RAM	2 KB	512 MB	512 MB
USB	n/a	4	1
Audio	n/a	HDMI análogo	HDMI
Video	n/a	HDMI análogo	Mini-HDMI
ETHERNET	n/a	10/100	10/100
PIN A/D	14 GPIO	40 GPIO, wifi, bluetooth.	69 GPIO
Sistema operativo	n/a	Linux	Android, Linux, Windows
Entorno	Arduino IDE	Linux, Python, Scratchbox, Eclipse	Python, Scratch, Linux
Compatibilidad con matlab y simulink	Support Package	Support Package	Support Package
Costo	29.95 USD	35 USD	45 USD

3.3.2 Raspberry Pi 3 B:

- **Ventajas:** debido a que es catalogada como una minicomputadora, esta atrjeta de desarrollo electrónico tiene más versatilidad y capacidad frente al Arduino. La Raspberry Pi cuenta con amplia documentación debido a que últimamente ha tomado gran popularidad por su fácil manejo, versatilidad y compatibilidad con diferente software, además de su alta productividad, ya que cuenta con un sistema operativo y un procesador de alta resolución. Igualmente, cuenta con un Support Package que facilita la programación, integración y comunicación con Matlab-Simulink.
- **Desventajas:** a diferencia del Arduino y el BeagleBone, no tiene tantas opciones para interfaces con sensores externos o botones, es decir, orientación a hardware, lo que no lo hace una buena opción para proyectos meramente electrónicos.

3.3.3 BeagleBone Black:

- **Ventajas:** La gran ventaja es que tiene un buen conjunto de entradas/salidas: 69 pins GPIO5, para interfaz exterior de forma fácil con dispositivos electrónicos, que son parte de los, en total, 92 posibles puntos de conexiones, incluyendo además de las GPIO, SPI bus, temporizadores, puertos seriales, salidas PWM y entradas analógicas. En cuanto a procesadores, tiene un gran soporte para expansiones, conexiones a un monitor DVI, a uno VGA, a una conexión HDMI, agregar una pantalla LCD demás cuenta con compatibilidad con Matlab-Simulink [14].
- **Desventajas:** No se tienen tantos puertos USB como el Pi, ni se tiene codificación de video, por lo que en realidad no es tan buena como computadora independiente o sistema de entretenimiento. Tampoco tiene el mismo *feedback* de usuarios como la Raspberry. Por lo que no hay tanta información disponible acerca de su uso. Puede ser muy útil para programas que son muy complicados para el Arduino, pero no tan útil con sistemas complejos como lo que puede resolver Raspberry Pi [15].

Teniendo en cuenta todo lo anteriormente mencionado, para este trabajo se optó por implementar las Raspberry Pi, gracias a que su sistema operativo y procesador de 4 núcleos son ideales para aplicaciones donde es necesario resolver ecuaciones diferenciales y donde se requieran periodos de funcionamiento muy extensos; además el support package es mucho más potente en cuanto a decodificación y aplicaciones que los paquetes disponibles para los demás dispositivos considerados.

3.4 Envío y recepción de datos

Para establecer la comunicación se hizo uso del protocolo del UDP (User Datagram Protocol) debido a que el software Simulink de Matlab que se utilizó para la programación tiene un toolbox o support package que facilita establecer comunicación entre ambos entes por medio de dos bloques de programación, especificando la

dirección IP tanto del PC como de la Raspberry Pi, además el tipo de dato y el tamaño de este. En este caso específico se debe hacer una sincronización en el envío de los datos, ya que existe una diferencia en el tiempo de muestreo y el procesamiento de la información.

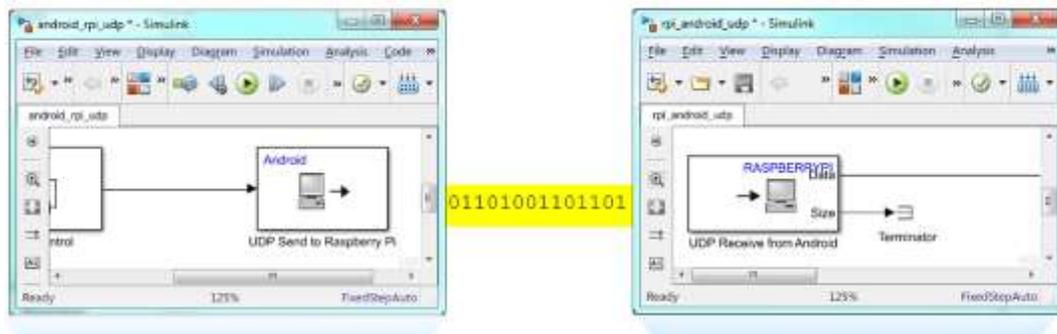


Figura 29. Bloques de comunicación del toolbox de Raspberry pi

3.5 Interfaz gráfica

Para la interfaz mostrada en la Figura 5, se generaron una entrada de insulina exógena y dos perturbaciones que corresponden a la alimentación enteral y parenteral del paciente, tanto en las entradas como en la perturbación es posible variar la duración de cada una. También se creó un panel que facilita la manipulación de algunos parámetros del modelo como lo son S_I , que corresponde a la sensibilidad de la insulina la cual se ve afectada en los pacientes que se encuentran en cuidado intensivos; β_1 , que representa la tasa de difusión de la insulina desde el plasma hasta el intersticio; y EGP_b , que representa la producción de insulina endógena del paciente. Además, contiene un panel que permite visualizar la variación en el tiempo de la glucosa en el intersticio G_i que corresponde a la medición realizada por un dispositivo de MCG (monitoreo continuo de glucosa).

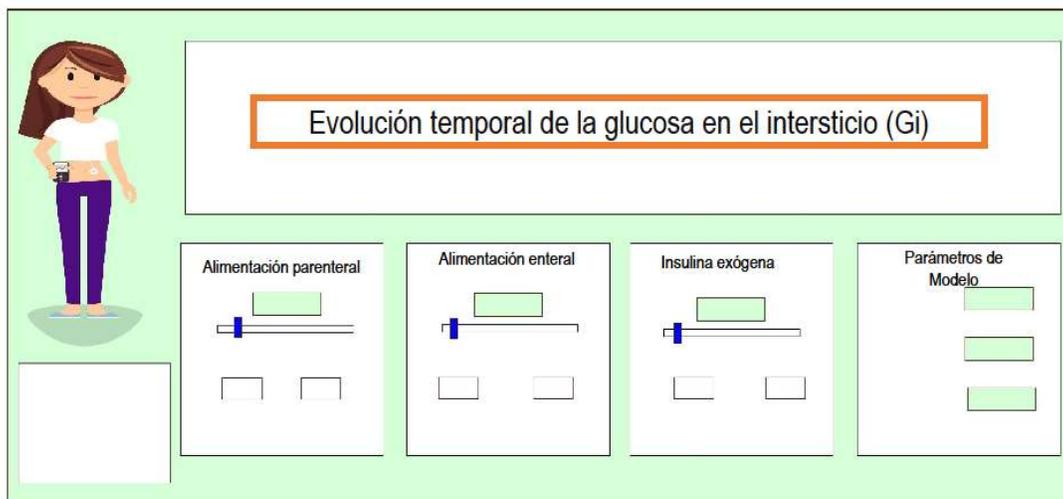


Figura 30. Interfaz para el modelo de Chase para pacientes en UCI

La Figura 6 muestra la interfaz desarrollada para los sensores virtuales (KF, UKF) dentro de la cual se pueden visualizar la variación en el tiempo tanto de la variable medida G_i como de la variable estimada B_G , además permite variar las condiciones iniciales del sensor virtual y sintonizar el sensor por medio de los parámetros R y Q . La interfaz contiene índices de desempeños $ITAE$, IAE , los cuales son calculados y visualizados en línea.

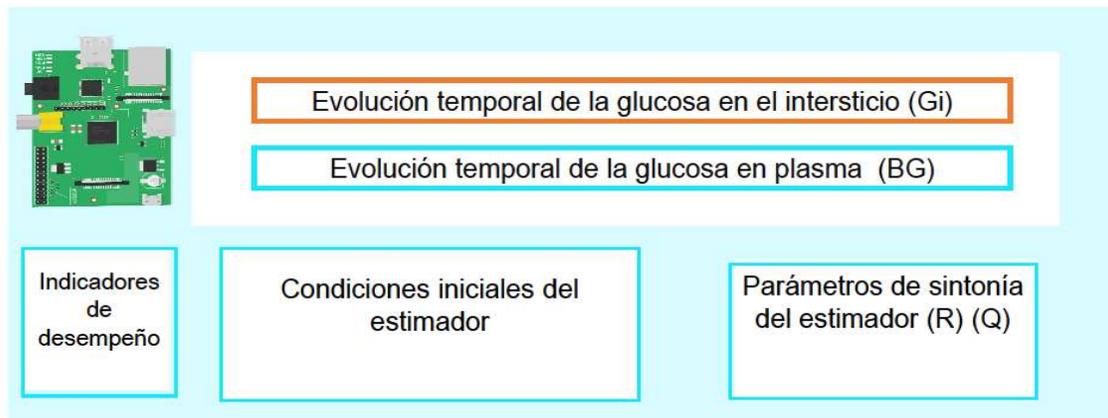


Figura 31. Interfaz para el sensor virtual

4 RESULTADOS

El sistema real (modelo de Chase), emula una medición del monitor continuo de glucosa, fue programado y simulado en el software Simulink. En el código de programación, se solucionan las ecuaciones diferenciales y se discretizan por el método de Euler con un tiempo de muestreo de un segundo. En esta simulación se muestran las entradas (alimentación enteral (D), alimentación parenteral (PN) e insulina exógena (U_{ex})); los estados glucosa plasmática (B_G) e intersticial (G_i), insulina plasmática (I) e intersticial (Q), glucosa en el estómago (P_1) y glucosa en el intestino (P_2) y su evolución en el tiempo de acuerdo a la variación en las entradas.

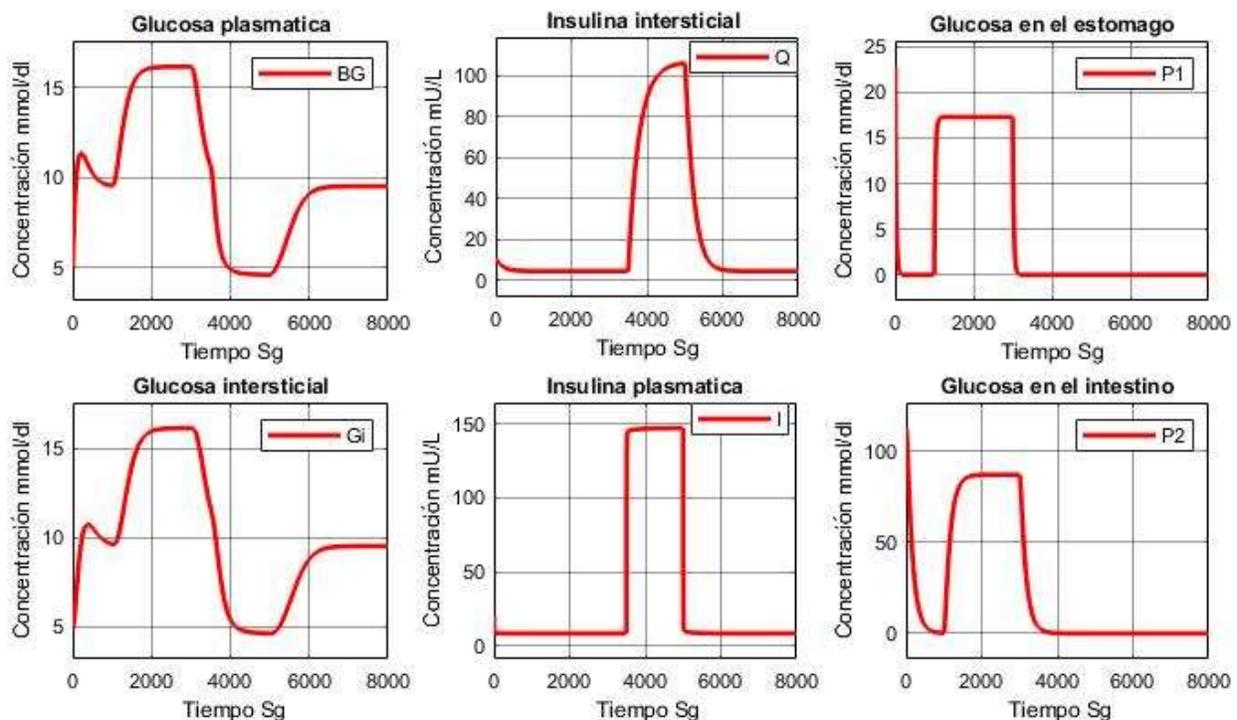


Figura 32. Simulación de los estados (B_G, Q, P_1, G_i, I, P_2) del modelo de Chase para pacientes en UCI.

En las Figuras 7 y 8 se puede visualizar el comportamiento de los estados y las entradas durante una simulación de 8000 segundos. Con una entrada desde el segundo 1000 al 2000 de alimentación enteral, seguida de una entrada de insulina exógena desde el segundo 3500 al segundo 5000.

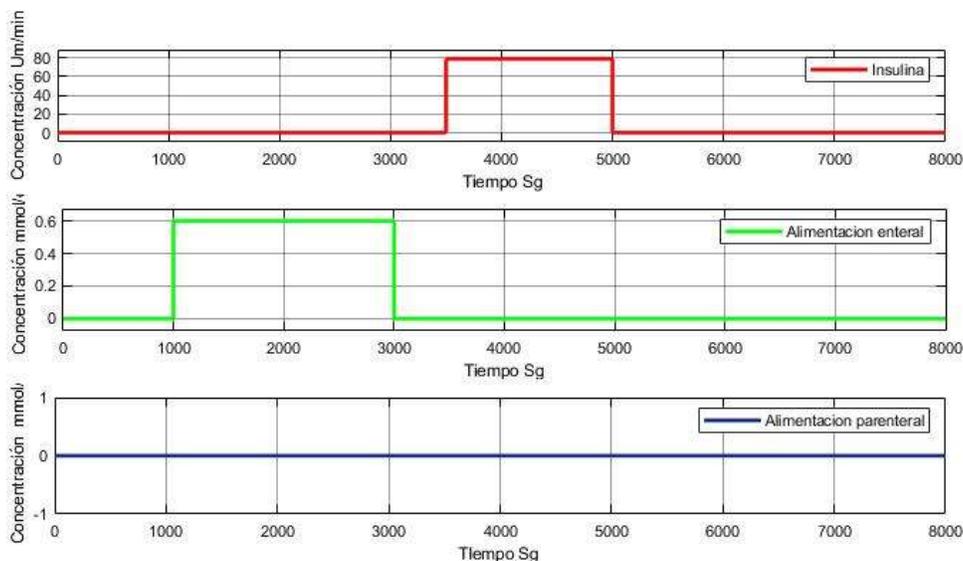


Figura 33. Simulación de las entradas del Sistema (U_{ex} , D , PN)

Partiendo del paciente virtual (modelo de Chase), se construyó un filtro de Kalman. El modelo fue linealizado por expansión en series de Taylor alrededor de los puntos de operación descritos en la Tabla 3.

Tabla 7. Puntos de operación para el modelo de Chase para pacientes en UCI.

Variable	Unidad	Punto de operación
B_G	[mmol/dl]	5
G_i	[mmol/dl]	5
Q	[mU/L]	10.86
I	[mU/L]	20.16
P_1	[mmol/dl]	22.33
P_2	[mmol/dl]	112.32
U_{ex}	[Um/min]	58.9
D	[mmol/dl]	0.5
PN	[mmol/dl]	0

Una vez linealizado el sistema para los puntos de operación mostrados en la Tabla (3), se implementó un filtro de Kalman y un filtro de Kalman extendido a través de una función de Matlab, donde fueron programadas las ecuaciones de predicción y corrección de los filtros que permitieron la reconstrucción de la glucosa plasmática (estimada), por medio de la glucosa intersticial (Sistema), como se puede observar en la Figura (4-5). Las condiciones de simulación se describen en las Tablas 4 y 5.

Tabla 8. Condiciones de simulación del modelo de Chase y el Filtro de Kalman.

Variable	Unidad	Condiciones de simulación del modelo	Condiciones de simulación del KF
B_G	[mmol/dl]	5	15
G_i	[mmol/dl]	5	15
Q	[mU/L]	10.86	8
I	[mU/L]	20.16	18
P_1	[mmol/dl]	22.33	20
P_2	[mmol/dl]	112.32	100
U_{ex}	[Um/min]	58.9	73.5
D	[mmol/dl]	0.5	0.6
PN	[mmol/dl]	0	0

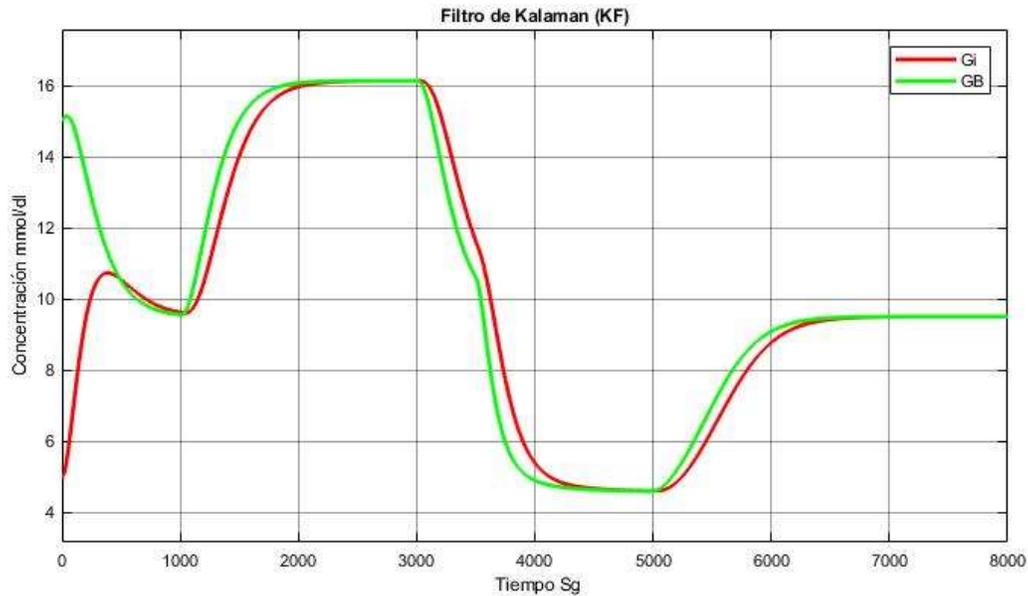


Figura 34. Simulación del filtro de Kalman y el modelo Chase.

Las Figuras 9 y 10 hacen parte de las pruebas realizadas en simulación tradicional. Estas pruebas permitieron evidenciar la respuesta del modelo y los estimadores a diferentes perturbaciones, asegurando de esta manera el funcionamiento del modelo y de los estimadores.

Por medio del toolbox de Raspberry Pi en Simulink, se embebió el estimador en las unidades de desarrollo Raspberry Pi; además, se estableció comunicación por medio del protocolo UDP disponible en el toolbox antes mencionado, lo que facilitó la comunicación ya que esta se realizó por medio de dos bloques de Simulink configurando la dirección IP de envío, el tamaño del dato, el puerto IP y el tipo de dato a recibir. Además, una vez descargados o codificados los estimadores a la Raspberry Pi, no es posible recibir información, solo permite enviarla. Es por esta razón que se cambiaron los roles, de esta manera el sensor virtual fue ejecutado en el computador y el modelo de Chase o paciente en la Raspberry Pi. También fueron ajustados algunos parámetros de la simulación como “pacing options”, lo que permitió una sincronización de ambas simulaciones (paciente y sensor virtual), de esta manera se recreó un ambiente realista para someter a pruebas los sensores virtuales.

Tabla 9. Condiciones de simulación del modelo de Chase y EKF

Variable	Condiciones de simulación del modelo	Condiciones de simulación del KF
B_G	5	15
G_i	5	15
Q	10.86	8
I	20.16	18
P_1	22.33	20
P_2	112.32	100
U_{ex}	58.9	73.5
D	0.5	0.6
PN	0	0

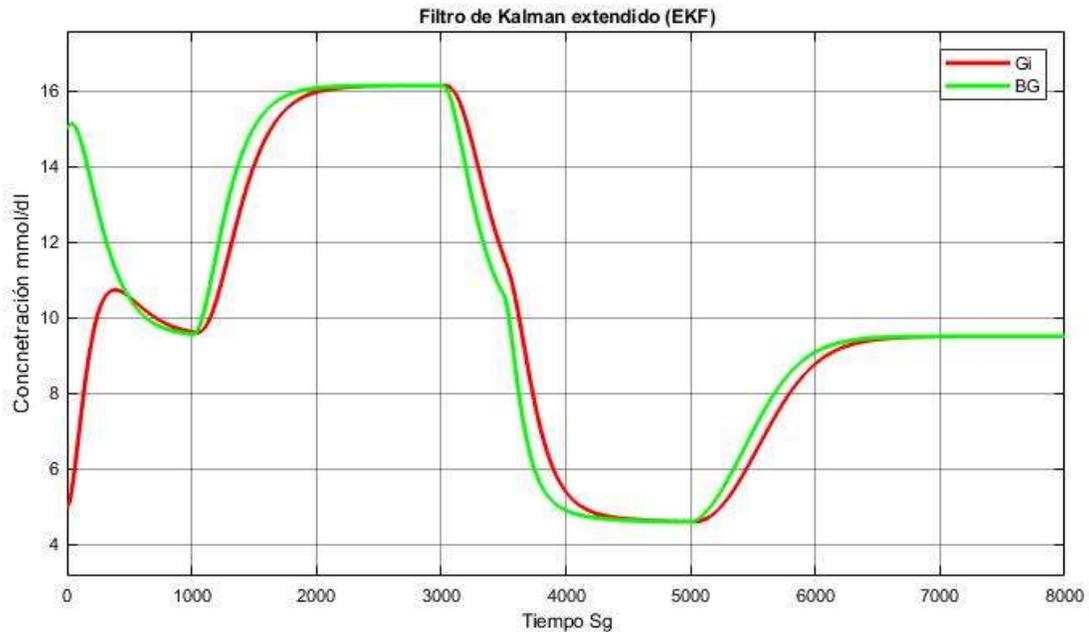


Figura 35. Simulación del filtro de Kalman extendido y el sistema.

Se diseñó una interfaz gráfica que se ilustra en las Figuras 11, 12 y 13, la cual permitió la manipulación de los parámetros de entrada del modelo y de las condiciones iniciales del estimador, además, se visualizó el cambio del modelo en el tiempo y la reconstrucción de la glucosa plasmática frente a la medición de glucosa intersticial. De igual forma, la interfaz permite someter el sensor virtual a incertidumbre paramétrica para evaluar la robustez de este y ajustar los parámetros de sintonía del sensor, recreando un ambiente dinámico.

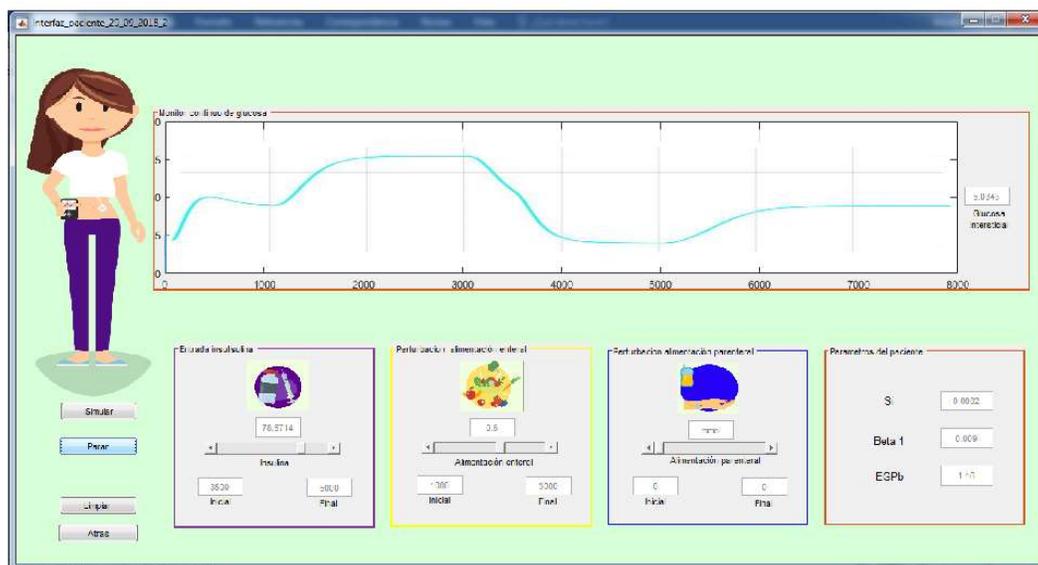


Figura 36. Simulación del modelo de Chase para pacientes en UCI utilizando la técnica HIL.



Figura 37. Simulación del KF utilizando la técnica HIL.



Figura 38. Simulación del EKF utilizando la técnica HIL.

El desempeño de los sistemas propuestos fue evaluado utilizando como indicadores la Integral Del Valor Absoluto Del Error (IAE) [15] y la Integral Del Valor Absoluto Del Error en el Tiempo (ITAE) [16]. La tabla 6 presenta los valores obtenidos, mostrando cómo el sensor virtual con el Filtro de Kalman Extendido alcanza mejor desempeño que el Filtro de Kalman Lineal.

Tabla 10. Indicadores de desempeño para los sensores virtuales implementados

Sensor / Rango de muestra en sg	Filtro de Kalman (KF)		Filtro de Kalman Extendido (EKF)	
	IAE	ITAE	IAE	ITAE
0 a 500 sg	0.0125	0.2353	0.0038	0.1570
501 a 1000 sg	0.0213	0.1265	0.0046	0.0132
2000 a 3000sg	0.0387	0.1560	0.0263	0.0137
4000 a 5000sg	0.0052	0.1594	0.0037	0.0116
6000 a 8000sg	0.0164	0.8957	0.0055	0.0365

5 CONCLUSIONES Y TRABAJO FUTURO

El protocolo desarrollado permitió establecer una comunicación bidireccional y un desempeño en tiempo real, para de esta manera reflejar así la técnica de simulación HIL, recreando un ambiente más realista en comparación a la simulación tradicional.

El filtro de Kalman extendido presenta mejor rendimiento que el filtro de Kalman evidenciado en los indicadores de desempeño. Sin embargo, puede decirse que tanto el KF y como el EKF presentan buen desempeño en tiempo real siempre y cuando la comunicación sea sincrónica, mostrando así que la metodología propuesta para implementar HIL podría ser usada en un sistema real para hacer pruebas pre-clínicas.

La interfaz de usuario facilita la interpretación y manipulación de los sistemas bajo prueba, recreando así un ambiente dinámico y realista de fácil manejo. La comunicación generada por el toolbox de Matlab para Raspberry Pi presenta algunas limitaciones en la codificación de los algoritmos haciendo que la Raspberry pi se comporte como una caja negra, dificultando la sincronía de los datos y el análisis de estos, además de que el protocolo de comunicación no permite la recepción de datos una vez codificado el programa.

Como trabajo futuro se pretende realizar un análisis cuantitativo del funcionamiento del estimador frente a condiciones como pérdida de datos, sincronización de sensores, entre otros haciendo una comparación directa frente a la simulación tradicional donde no se establece comunicación, también comparando los indicadores desempeño en tiempo real y los tradicionales. Con el fin de poder poner a prueba otros tipos de estimadores de estados y validar su funcionamiento en un ambiente realista, además se plantea desarrollar esta metodología usando otros tipos de software que facilite aún más la comunicación.

REFERENCIAS

- [1] S. Jereb, N. Asus, M. Blumtritt, and Y. Kreff, "Hiperglucemia en el paciente Neurocrítico Hyperglycemia in Neurocritical patients," *Diaeta*, vol. 33, no. 150, pp. 7–11, 2015.
- [2] J. C.-C. E. Aguirre-Zapata, "Glycemic Monitoring in Critical Care using Nonlinear State Estimators." cartagena, 2017.
- [3] B. M. Hanson, M. C. Levesley, K. Watterson, and P. G. Walker, "Hardware-in-the-loop-simulation of the cardiovascular system, with assist device testing application," *Med. Eng. Phys.*, vol. 29, no. 3, pp. 367–374, 2007.
- [4] A. Alazmani *et al.*, "Introducing a hardware-in-the-loop simulation of the cardiovascular system," *Proc. IEEE RAS EMBS Int. Conf. Biomed. Robot. Biomechatronics*, pp. 153–158, 2012.
- [5] C. Barker, M. Kwiatkowska, A. Mereacre, N. Paoletti, and A. Patan, "Hardware-in-the-loop simulation and energy optimization of cardiac pacemakers," pp. 7188–7191, 2015.
- [6] M. Bacic, "On hardware-in-the-loop simulation," *Proc. 44th IEEE Conf. Decis. Control*, pp. 3194–3198, 2005.
- [7] K. H. Ian T. Cameron, *Process Modelling and Model Analysis*, vol. 4, no. 2. 2001.
- [8] C. Eberle and C. Ament, "Real-time state estimation and long-term model adaptation: A two-sided approach toward personalized diagnosis of glucose and insulin levels," *J. Diabetes Sci. Technol.*, vol. 6, no. 5, pp. 1148–1158, 2012.
- [9] R. Gillis, C. C. Palerm, H. Zisser, L. Jovanovic, F. Doyle III, and D. E. Seborg, "Glucose Estimation and Prediction through Meal Responses Using Ambulatory Subject Data for Advisory Mode Model Predictive Control," *J. diabetes Sci. Technol.*, vol. 1, no. 6, pp. 825–33, 2007.
- [10] R. Sanchis, I. Peñarrocha, and P. Albertos, "Design of robust output predictors under scarce measurements with time-varying delays," *Automatica*, vol. 43, no. 2, pp. 281–289, 2007.
- [11] J. Mohd, N. H. Hoang, M. A. Hussain, and D. Dochain, "Review and classification of recent observers applied in chemical process systems," *Comput. Chem. Eng.*, vol. 76, pp. 27–41, 2015.
- [12] J. G. Chase, G. M. Shaw, X. W. Wong, T. Lotz, J. Lin, and C. E. Hann, "Model-based glycaemic control in critical care-A review of the state of the possible," *Biomed. Signal Process. Control*, vol. 1, no. 1, pp. 3–21, 2006.
- [13] C. Eberle and C. Ament, "The Unscented Kalman Filter estimates the plasma insulin from glucose measurement," *BioSystems*, vol. 103, no. 1, pp. 67–72, 2011.
- [14] S. Casco, "Raspberry Pi , Arduino y Beaglebone Black Comparación y Aplicaciones," vol. 1, pp. 4–8, 2014.
- [15] E. D. Aguirre Mesa, D. A. Garcia Arango, y C. F. Henao Villa, "Diseño y simulación de un amplificador 'Instrumental Operacional' como apoyo en la lectura y transmisión de señales mioeléctricas por un solo canal con bajo ruido", Revista CINTEX, vol. 22, núm. 1, pp. 11–23, 2017.
- [16] J. E. Castano, J. Patino, y J. J. Espinosa, "Model identification for control of a distillation column", en Robotics Symposium, 2011 IEEE IX Latin American and IEEE Colombian Conference on Automatic

Control and Industry Applications (LARC), Bogotá, Colombia, 2011, pp. 1–5. DOI:
10.1109/LARC.2011.6086832

- [17] V. Srdanovic et al., “Diseño y simulación de un sistema de medición de temperatura orientado hacia el aprendizaje aplicativo”, *Revista CINTEX*, vol. 19, pp. 9–22, 2014.