

Desarrollo e implementación de algoritmo de control para bomba de insulina
Alumno: Tedesco Victor Emmanuel
Carrera: Ingeniería Electrónica

Universidad Nacional de General San Martín
Escuela de Ciencia y Tecnología
Campus Miguelete - Martín de Irigoyen 3100
San Martín – Pcia. De Buenos Aires - Argentina
www.unsam.edu.ar
EscuelaCyT@unsam.edu.ar



UNSAM
UNIVERSIDAD
NACIONAL DE
SAN MARTÍN

ESCUELA DE CIENCIA Y TECNOLOGÍA

Carrera: Ingeniería Electrónica

Proyecto Final Integrador

Alumno: Tedesco Victor Emmanuel

Legajo: CYT-2723

**Tema: DESARROLLO E IMPLEMENTACIÓN DE ALGORITMO DE CONTROL
PARA BOMBA DE INSULINA**

Fecha de presentación: Jueves 5 de marzo de 2019

Tedesco Victor Emmanuel

DNI 33.339.757

Contenido

Descripción del proyecto	4
Requerimientos	4
Alcances y Objetivos	4
Gestión de Proyecto	5
WBS (Work Breakdown Structure)	5
Diagrama de Gantt	6
Evaluación de Riesgos	8
Desarrollo de las actividades realizadas	10
Primeros pasos.....	10
Microcontrolador	11
Servomotor	12
Sensor de final de carrera por exceso de corriente	14
Interfaz mecánica de adaptación	15
Simulación del cuerpo humano (sensor)	20
Diagramas circuitales	22
Pruebas de control.....	24
Validación.....	25
Primera validación:	27
Segunda validación:	30
Conclusiones.....	32
Posibles mejoras	32
Anexos	33
Anexo 1- Diabetes tipo 1.....	33
Anexo 2 - Control digital PID.....	33
Anexo 3 - Algoritmos PID discretos	34
Anexo 4 - Caracterización de la respuesta temporal al escalón de un sistema subamortiguado.....	36
Anexo 5 - Mbed	37
Anexo 6 - JavaScript.....	39
Anexo 7 - Control de potencia motor CC – PWM	40
Anexo 8 - Puente H	40
Anexo 9 - Amplificador operacional	42
Referencias bibliográficas	45

Descripción del proyecto

El propósito del proyecto es fabricar un equipo fiable que permita controlar el desplazamiento de una jeringa la cual contenga insulina. Dicho desplazamiento funcionará como bomba que suministre el medicamento. El desplazamiento debe respetar una dosificación específica controlada por un modelo matemático preestablecido.

Requerimientos

Este proyecto por desarrollar deberá contemplar los siguientes requerimientos:

- El control se basará en un algoritmo matemático de control desarrollado por el Dr. Carlos Amorena y el Dr. Carlos D'Attellis. (BSPC-D-17-00864).
- El control del equipo debe ser preciso y repetible.
- Las entradas de dicho algoritmo de control serán simuladas dejando lugar, en el desarrollo, para incluir la entrada de un sensor de glucosa en sangre en proyectos futuros.
- La salida de dicho sistema debe controlar la dosificación de insulina (en unidades de mililitros) a través de una jeringa. El movimiento mecánico de dicha jeringa queda abierto a posibles soluciones futuras, tomando como referencia la rotación de un motor de corriente continua que accione el desplazamiento de la jeringa.
- Se necesita un ciclo continuo de trabajo con lecturas del sensor (en este proyecto esta lectura será simulada) y accionamiento periódico con intervalos del orden de minutos.

Alcances y Objetivos

La prestación del proyecto incluye la adaptación de las entradas simuladas al hardware a utilizar, el diseño del hardware del sistema de control basado en el desarrollo de los solicitantes y la implementación de un prototipo que permita verificar el correcto funcionamiento del algoritmo desarrollado.

Cabe aclarar que el proyecto forma parte de un proyecto mayor que integra varias disciplinas y distintos equipos de trabajo, como así también varias etapas a las cuales este proyecto contempla adaptarse.

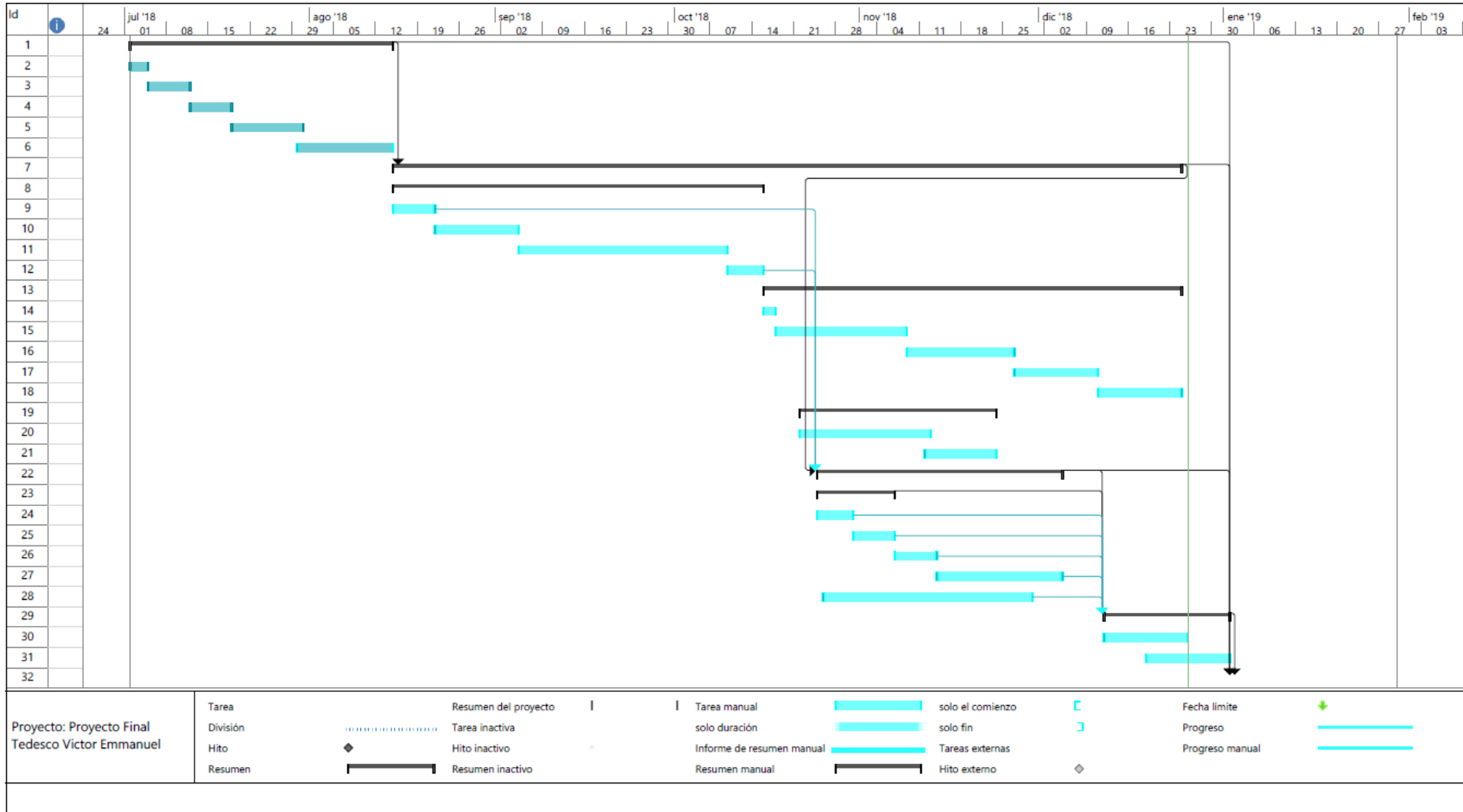
Gestión de Proyecto

WBS (Work Breakdown Structure)

1. Documentación e investigación
 - 1.1. Desarrollo y análisis de riesgos
 - 1.2. Investigación sobre proyectos pasados
 - 1.3. Desarrollo del plan de acción
 - 1.4. Investigación sobre programación de sistemas de control en sistemas embebidos (Mbed)
 - 1.5. Preparación de presentación y proyecto escrito
2. Hardware
 - 2.1. Microcontrolador
 - 2.1.1. Definición de microcontrolador a utilizar
 - 2.1.2. Investigación sobre metodologías de control
 - 2.1.3. Desarrollo del algoritmo en lenguaje de programación
 - 2.1.4. Implementación de primer prototipo
 - 2.1.4.1. Elección de componentes
 - 2.1.4.2. Fabricación de placas y test
 - 2.1.5. Mediciones sobre niveles lógicos o analógicos (sin actuador)
 - 2.2. Control de salida
 - 2.2.1. Análisis de factores de riesgo
 - 2.2.2. Desarrollo de etapas de salida
 - 2.2.3. Elección de actuador mecánico
 - 2.2.4. Pruebas de movimiento y velocidad
 - 2.2.5. Pruebas de posición efectiva
 - 2.3. Pruebas de funcionamiento
 - 2.3.1. Pruebas de funcionamiento del motor
 - 2.3.2. Pruebas de funcionamiento del algoritmo
 - 2.3.3. Pruebas en INTI sobre dispositivos de comprobación (a definir)
 - 2.4. Prototipado Final en PCB
 - 2.4.1. Armado, integración y pruebas finales
3. Validación
 - 3.1. Discretización
 - 3.2. Repetitividad
 - 3.3. Control oscilatorio

Diagrama de Gantt

Id	Modo de tarea	Nombre de tarea	Duración	Comienzo	Fin	Predecesoras
1		Documentación e investigación	32 días	lun 02/07/18	mar 14/08/18	
2		Desarrollo y análisis de riesgos	3 días	lun 02/07/18	mié 04/07/18	
3		Investigación sobre proyectos similares previos	5 días	jue 05/07/18	mié 11/07/18	
4		Desarrollo de plan de acción	5 días	jue 12/07/18	mié 18/07/18	
5		Investigación sobre programación de sistemas de control en sistemas embebidos (Mbe	8 días	jue 19/07/18	lun 30/07/18	
6		Preparación de presentación y documentación de ante proyecto	12 días	lun 30/07/18	mar 14/08/18	
7		Desarrollo de hardware & software	94 días	mié 15/08/18	lun 24/12/18	1
8		Microcontrolador	44 días	mié 15/08/18	lun 15/10/18	
9		Investigación de microntroladores y definición del micontrolador a utilizar	5 días	mié 15/08/18	mar 21/08/18	
10		Investigación de metodologías de control	10 días	mié 22/08/18	mar 04/09/18	
11		Desarrollo del algoritmo en el lenguaje de programación	25 días	mié 05/09/18	mar 09/10/18	
12		Pruebas y mediciones sobre niveles lógicos y analógicos	4 días	mié 10/10/18	lun 15/10/18	
13		Control de salida	50 días	mar 16/10/18	lun 24/12/18	
14		Análisis de factores de riesgo	2 días	mar 16/10/18	mié 17/10/18	
15		Desarrollo de etapas de salida	16 días	jue 18/10/18	jue 08/11/18	
16		Elección de actuador mecánico	12 días	vie 09/11/18	lun 26/11/18	
17		Pruebas de movimiento y velocidad	10 días	mar 27/11/18	lun 10/12/18	
18		Pruebas de posición efectiva	10 días	mar 11/12/18	lun 24/12/18	
19		Software de sensado	25 días	lun 22/10/18	vie 23/11/18	
20		Desarrollo se sensor de insulina	16 días	lun 22/10/18	lun 12/11/18	
21		Implementación de programa (JAVA)	10 días	lun 12/11/18	vie 23/11/18	
22		Prototipado mecánico y electrónico	29 días	jue 25/10/18	mar 04/12/18	7,9,12
23		Implementación del primer prototipo	9 días	jue 25/10/18	mar 06/11/18	
24		Elección de componente	4 días	jue 25/10/18	mar 30/10/18	
25		Fabricación de placas y test iniciales	5 días	mié 31/10/18	mar 06/11/18	
26		Diseño de componentes mecánicos	5 días	mié 07/11/18	mar 13/11/18	
27		Fabricación de elementos (impresión 3D)	15 días	mié 14/11/18	mar 04/12/18	
28		Implementación de prototipo definitivo	25 días	vie 26/10/18	jue 29/11/18	
29		Validación	15 días	mié 12/12/18	mar 01/01/19	22,23,24,25,26,27,28
30		Preparación de elemento adquisidor (Balanza de precisión)	10 días	mié 12/12/18	mar 25/12/18	
31		Pruebas de simulaciones (INTI)	10 días	mié 19/12/18	mar 01/01/19	
32		Entrega de informe	20 días	mié 02/01/19	mar 29/01/19	1,7,22,29



Evaluación de Riesgos

EVALUACIÓN DEL RIESGO		VALORACIÓN DEL RIESGO										MEDIDAS A TOMAR					
nº	Riesgo identificado	Probabilidad			Consecuencias			Estimación del riesgo				Medida de control	Información extra	¿Riesgo controlado?		Acción requerida	
		PB	PM	PA	LD	D	ED	T	TO	M	I			IN	SI		NO
BI1	Perdida del interés en el proyecto	X				X		X					Fomentar el trabajo en equipo, la organización de las partes involucradas e integrar a los participantes a colaborar con el desarrollo continuo	La desidia de los participantes del proyectos podrían causar un aumento del riesgo a largo plazo.	X		Fomentación del trabajo en equipo
BI2	Cambios en los requisitos del proyecto		X			X						X	Comunicación continua con el Director y/o Codirector del proyecto para solucionar inconvenientes de descoordinación.	Dada la complejidad del proyecto y la gran cantidad de equipos interdisciplinarios que se requieren, este riesgo debe controlarse de cerca para que el proyecto concluya satisfactoriamente.	X		Información constante del Director del proyecto y los integrantes de las partes que articulan.
BI3	Mala planificación del proyecto	X				X			X				Control de las partes integrantes del proyecto y revisión de los entregables.	Contemplando la baja experiencia del desarrollador del proyecto se implementa una auto-capacitación basada en la revisión de proyectos de índole similar	X		Capacitación basado en proyectos anteriores.
BI4	Subestimación de los tiempos necesarios para la finalización del proyecto		X		X				X				Revisión del diagrama de tiempos necesario para la finalización del proyecto.		X		Re planificación y revisión del diagrama de Gantt
BI5	Falta de documentación de las partes integrantes del proyecto global		X			X						X	Comunicación con las partes que integran de los sub-proyectos involucrados tanto anteriores como posteriores.	En caso de no disponer respuesta de las partes puede aumentar el riesgo basado en la incapacidad de reproducción la información requerida.	X		Construcción de un canal de comunicación constante con todos los integrantes.
BI6	Falta de equipamiento y/o partes necesarias para el desarrollo del proyecto	X				X			X				Revisión del equipamiento necesario.	La falta de disponibilidad de una bomba de insulina comercial puede volver la finalización del proyecto incierta.	X		Búsqueda de alternativas para las partes faltantes
BI7	Fallas del equipamiento en etapa de Test o prototipado	X				X			X				Implementación de una revisión del programa para dejar abierta las funcionalidades a proyectos futuros	Las buenas practicas de programación pueden disminuir la probabilidad de éste riesgo	X		Reprogramación
BI8	Daño colateral	X			X			X					Imposibilidad de controlar el riesgo.	Todo riesgo basado en una probabilidad baja de daño tanto de las personas como del equipamiento.		X	Monitoreo continuo del sistema de control de riesgos

		Consecuencias		
		Ligeramente Dañino	Dañino	Extremadamente Dañino
		LD	D	ED
Probabilidad	Baja B	Riesgo trivial T	Riesgo tolerable TO	Riesgo moderado MO
	Media M	Riesgo tolerable TO	Riesgo moderado MO	Riesgo Importante I
	Alta	Riesgo moderado M	Riesgo importante I	Riesgo Intolerable IN

Riesgo	Acción y temporización
Trivial (T)	No se requiere acción específica
Tolerable (TO)	No se necesita mejorar la acción preventiva. Sin embargo se deben considerar soluciones más rentables o mejoras que no supongan una carga económica importante. Se requieren comprobaciones periódicas para asegurar que se mantiene la eficacia de las medidas de control.
Moderado (M)	Se deben hacer esfuerzos para reducir el riesgo, determinando las inversiones precisas. Las medidas para reducir el riesgo deben implantarse en un período determinado. Cuando el riesgo moderado esta asociado con consecuencias extremadamente dañinas, se precisará una acción posterior para establecer, con más precisión, la probabilidad de daño como base para determinar la necesidad de mejora de las medidas de control.
Importante (I)	No debe comenzarse el trabajo hasta que se haya reducido el riesgo. Puede que se precisen recursos considerables para controlar el riesgo. Cuando el riesgo corresponda a un trabajo que se está realizando, debe remediarse el problema en un tiempo inferior al de los riesgos moderados.
Intolerable (IN)	No debe comenzar ni continuar el trabajo hasta que se reduzca el riesgo. Si no es posible reducir el riesgo, incluso con recursos ilimitados, debe prohibirse el trabajo.

Desarrollo de las actividades realizadas

Primeros pasos

Al comienzo del proyecto se demarcaron los objetivos del proyecto y las condiciones iniciales para llegar a dichos objetivos.

A continuación, detallaremos los primeros pasos que se presentan en este proyecto:

- La necesidad de contar con un microcontrolador para poder realizar el algoritmo de control es fácil de ver. En los requerimientos del proyecto se demarcó que el proceso de control no debe ser de muy alta velocidad, por ello la velocidad de procesamiento de un DSP o un FPGA no es necesaria en este tipo de aplicación.
- Se propone desarrollar un control de desplazamiento a través de un servomotor, el cual deba ser controlado con PWM. Esto nos indica que es necesaria, por lo menos, una salida de ese tipo en nuestro hardware.
- El motor a elegir debe ser pequeño, dado que se intenta implementar a futuro en dispositivos portátiles, y de bajas revoluciones por minuto ya que no es velocidad lo que se busca sino fuerza para desplazar la jeringa de inyección de insulina.
- Un paso que contemplar es el consumo de corriente y la posibilidad de detectar que el embolo de la jeringa ya fue desplazado hasta su máximo recorrido. Esto se intenta solucionar con un solo elemento de sensado de corriente dándole al dispositivo final menor cantidad de componentes mecánicos que puedan verse afectados en su funcionamiento.
- Toda la mecánica de desplazamiento y soporte de la jeringa nos obliga a hacer un desarrollo de un soporte para todos los componentes como prototipo. Para ellos se implementa un diseño por computadora (CAD) y se elabora dicho prototipo en una impresora 3D.
- El algoritmo matemático en el cual se basa este proyecto es altamente complejo y fue desarrollado como simulación de Matlab bajo la plataforma de SimuLink. Dicha simulación debió ser modificada para poder extraer los parámetros de validación como así también el algoritmo de control a implementar (pseudo ingeniería inversa).
- Uno de los principales impedimentos evaluados como un riesgo en este proyecto es la falta de un sensor funcional para la medición de glucosa en sangre. Este proyecto contempla la fabricación de una simulación del sensado de glucosa en sangre de un cuerpo humano por computadora.

Microcontrolador

La elección de un microcontrolador simple de utilizar y de programar fue esencial para el desarrollo del proyecto. Los parámetros a contemplar son la rapidez en el desarrollo de la interfaz, la simplicidad circuital y el bajo consumo de energía.

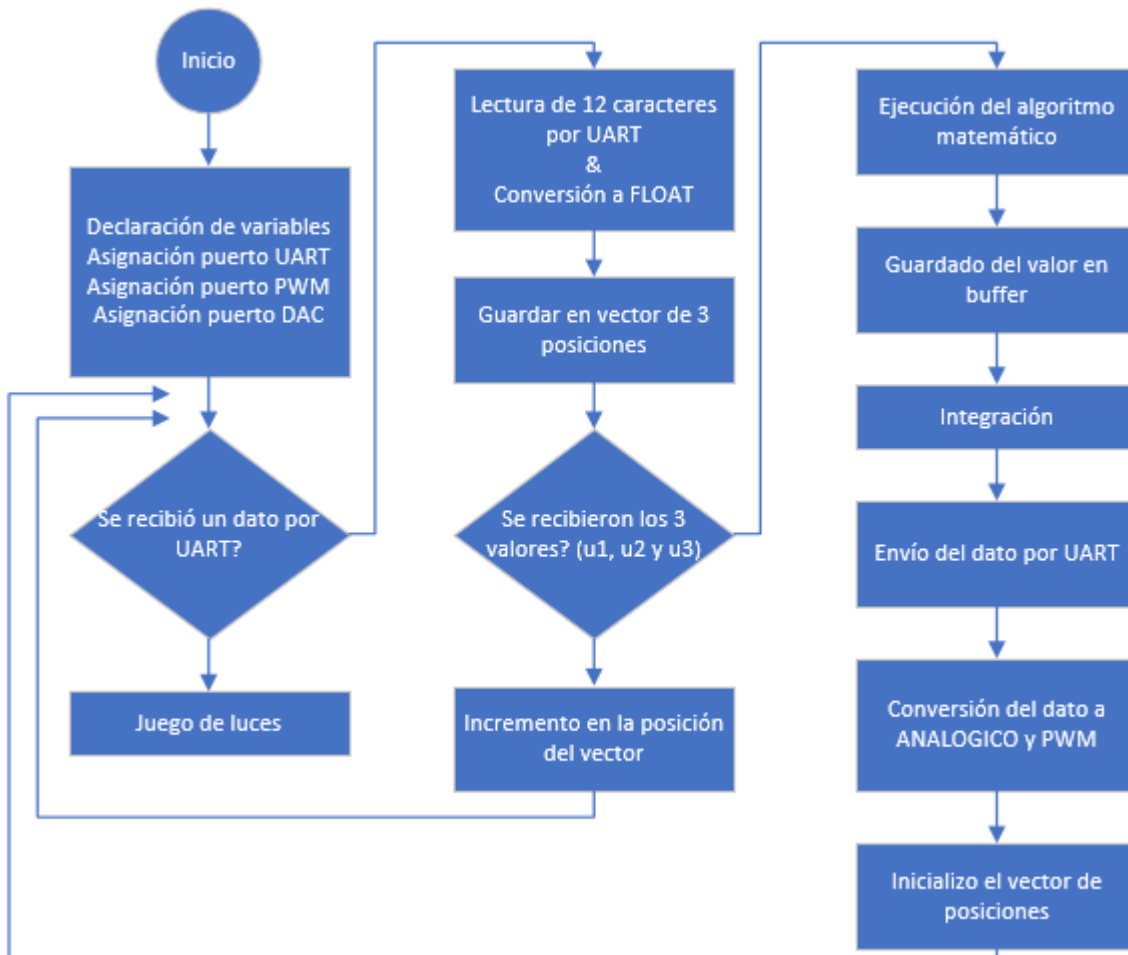
También cabe destacar que la velocidad de procesamiento no es un factor que contemplar para el proyecto ya que los tiempos de control son muy lentos.

Como se explica en el marco teórico, los dispositivos Mbed son útiles para este tipo de aplicación.

El board Mbed LPC1768 posee una velocidad de procesamiento de 100MHz (más que suficiente para nuestro desarrollo) y bibliotecas de Mbed OS V5 hacen fácil el manejo del motor por PWM.

Las bibliotecas de control de UART también nos ayudarán mucho ya que se pretende simular un sensor por computadora y podemos enviar y recibir estos datos por puerto serie.

A continuación, explicamos en un diagrama de flujo el funcionamiento básico del programa implementado.



Servomotor

Para poder generar el desplazamiento lineal haremos una conversión entre desplazamiento rotacional a lineal por medio de un eje de motor roscado.

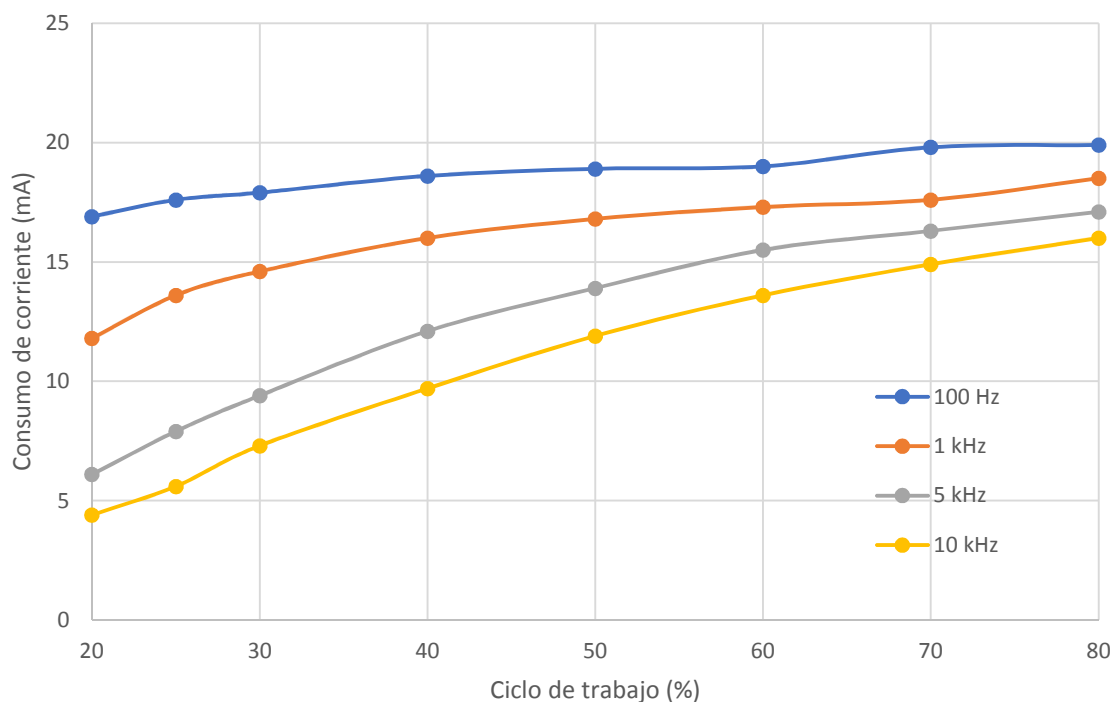
Esto lo haremos gracias a un mini motor de rodamientos metálicos (Mini Metal Gear Motor) ideal para hacer robots. Posee un peso ligero, alto torque y bajas revoluciones por minuto. Su fabricación es de precisión milimétrica y duradera. A este mini motor se le puede montar fácilmente una rueda en el eje de salida.

(Es ampliamente utilizado en embarcaciones, modelos de automóviles, robóticos, electrodomésticos, control de movimiento lineal, etc)

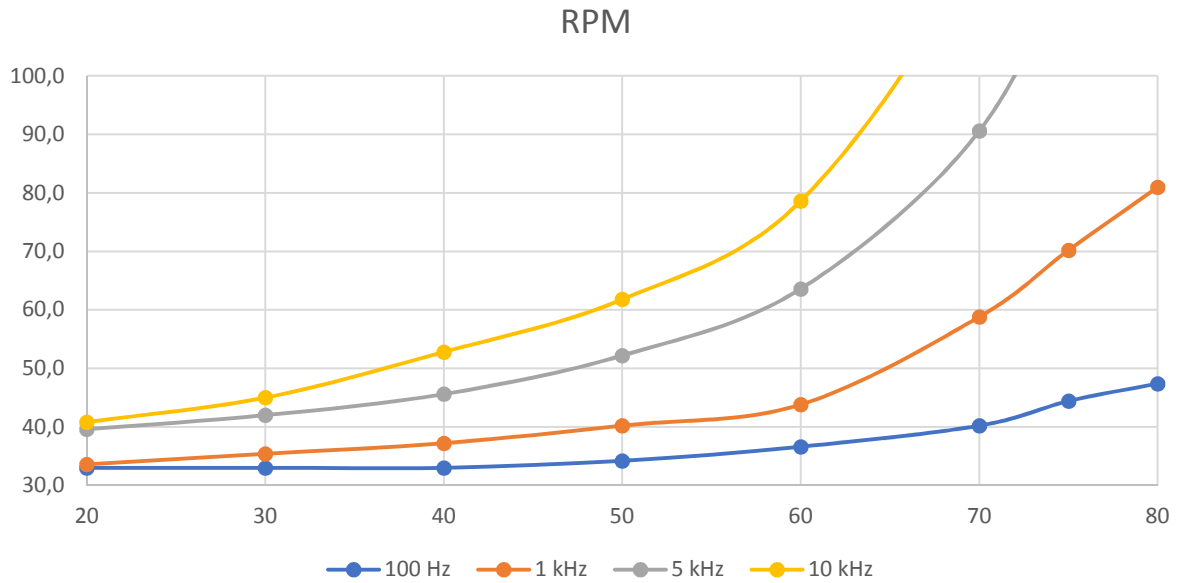


Sus elementos de reducción lo hacen un elemento útil para generar mucho torque en poco espacio. Su rotación en eje va desde 50 RPM a 200 RPM según la tensión de DC aplicada en sus bornes. Esto puede ser controlado y variado con el ciclo de trabajo aplicado.

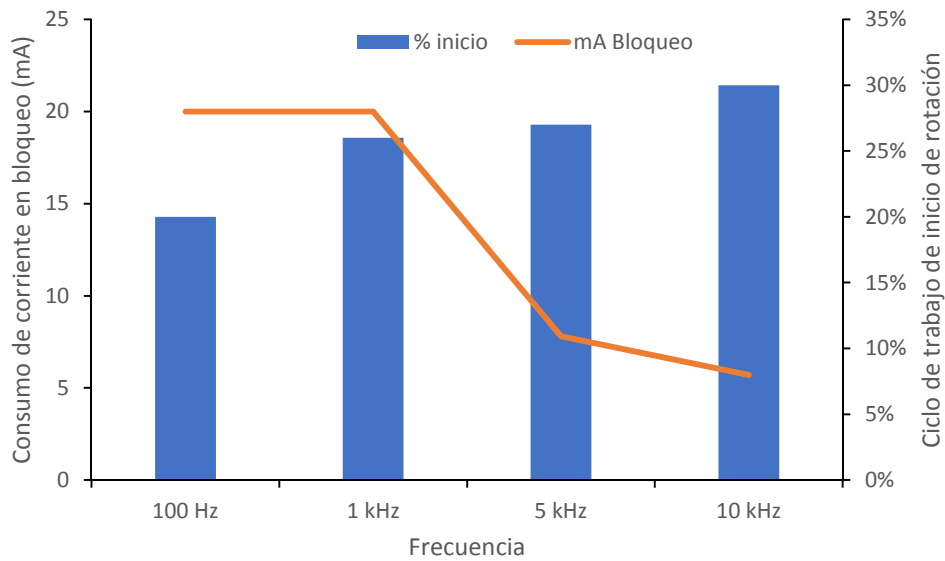
Para evaluar el rendimiento y el consumo de corriente relacionado tanto con el ciclo de trabajo como el bloqueo del motor por sobreesfuerzo se realizaron varias mediciones de prueba presentadas a continuación:



Se observa de las mediciones realizadas que al aumentar el ciclo de trabajo el consumo de corriente continua se eleva. Coincidente con la teoría de este tipo de control.



Al aumentar el ciclo de trabajo, las revoluciones por minuto en este servomotor aumentan lo que nos da una idea de a qué proporción podemos variar la velocidad del motor en función del ciclo de trabajo. Estos datos nos permitieron trabajar sobre la ecuación de salida del controlador oscilatorio.



De todas estas mediciones se sacaron valor de cómo debe ser controlado el servomotor y cuáles son los parámetros máximos y mínimos que debemos utilizar. También se evalúa como gran experiencia dado que el fabricante del componente no provee este tipo de información.

Sensor de final de carrera por exceso de corriente

La idea principal es sensar la corriente continua que atraviesa al bobinado del motor y así detectar su atascamiento o su llegada al final del recorrido de la jeringa.

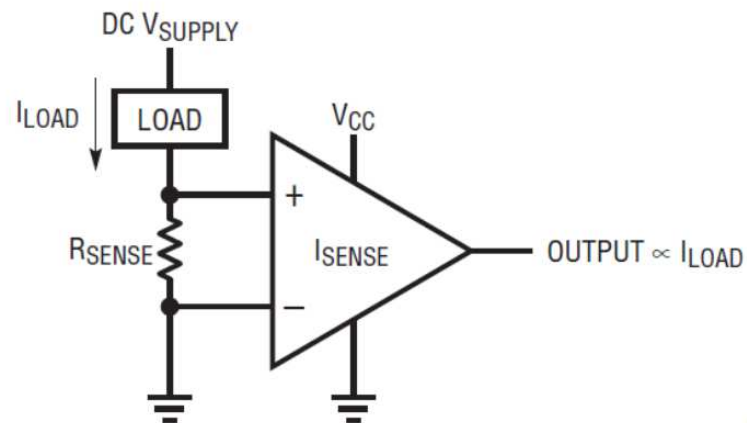
Para ello utilizamos un circuito de sensado por niveles de tensión entre dos extremos de un sensor.

Utilizamos un elemento de sensado de corriente como lo es una resistencia (shunt). Para evitar la pérdida de energía en forma de calor disipada por dicha resistencia se opta por una de muy bajo valor.

Los extremos de este sensor serán comparados por un amplificador operacional.

Las distintas técnicas de sensado varían dependiendo de la colocación del elemento resistivo (shunt) con respecto a la carga. Principalmente evaluamos tres situaciones:

- Sensado de corriente en la parte alta o sensado "High-Side": el elemento de sensado (shunt) aparece en el camino desde el positivo de la fuente de alimentación hacia la carga.
- Sensado de corriente en la parte baja o "Low-Side": la resistencia de shunt aparece en el camino de retorno desde la carga hacia la tierra o masa del circuito. Este es el que utilizaremos en nuestro caso.
- Sensado de rango completo (bidireccional): permite la medición de la corriente que fluye en ambos sentidos a través del resistor shunt, es común encontrar estos circuitos por ejemplo en la salida de un puente H.

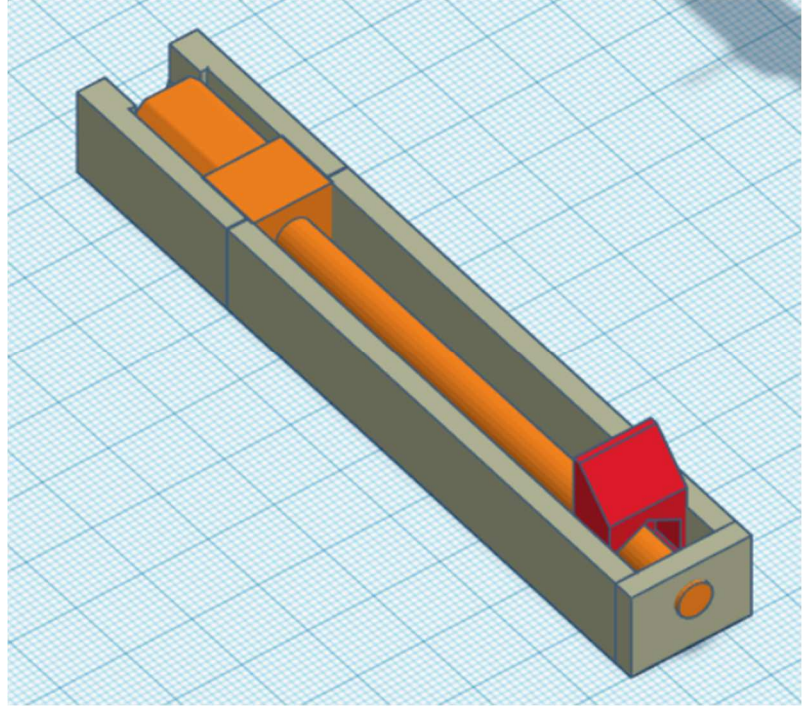


Como modificación el circuito de la imagen se dispondrá de un nivel de comparación más alto que el de GND a través de un divisor resistivo ya que nos interesa únicamente detectar cuando el bobinado del motor sobrepasa el nivel de bloqueo o de sobre exigencia.

Interfaz mecánica de adaptación

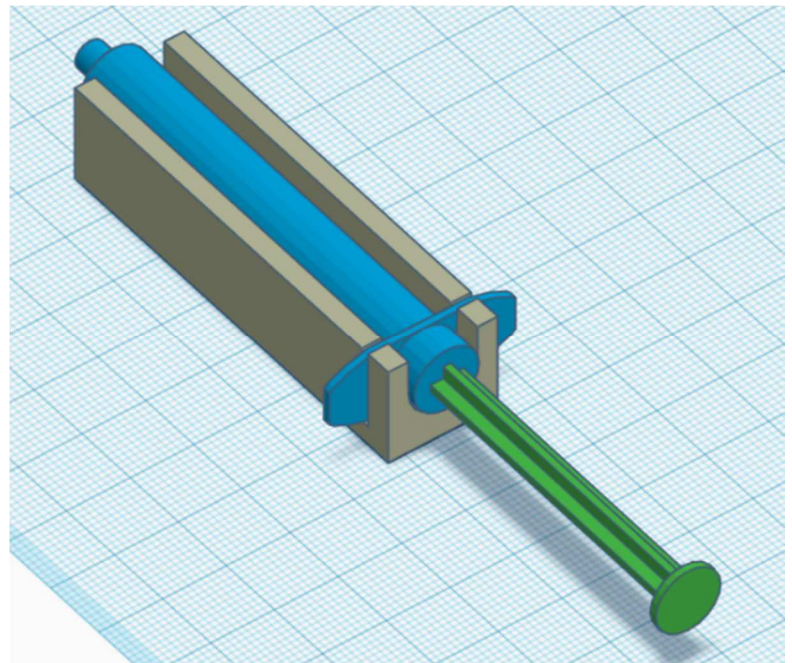
El primer desafío fuera del campo de la ingeniería electrónica fue realizar un prototipo que permita el desplazamiento mecánico de la jeringa con la precisión y repetitividad que el proyecto requiere.

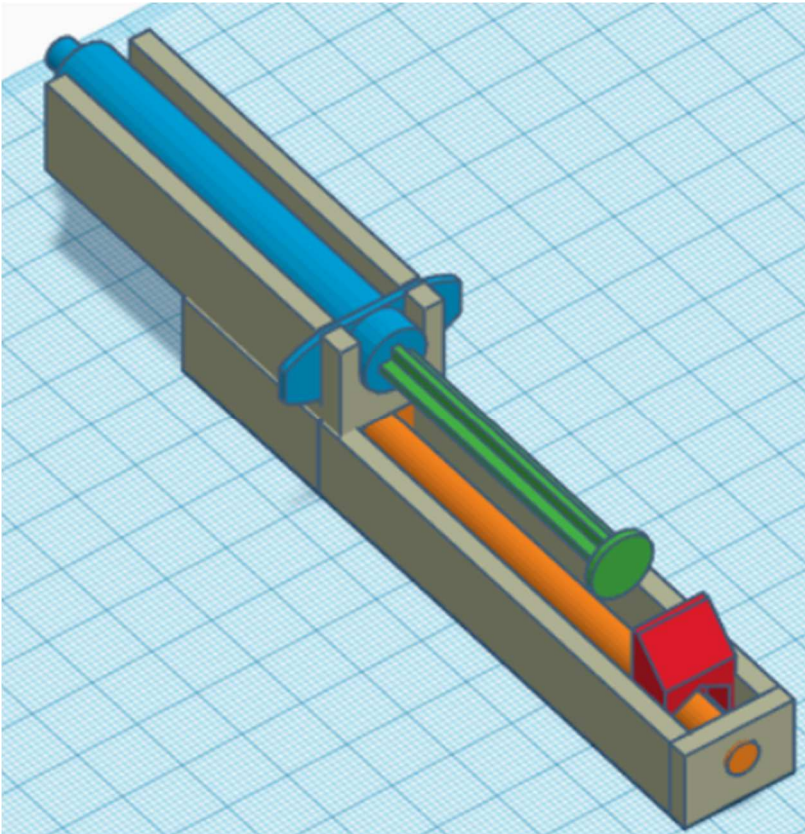
Como primera característica nos basamos en el soporte del motor de corriente continua y su sujeción a un eje sin fin que permitiera convertir su rotación en desplazamiento lineal por medio de una tuerca. Los parámetros de paso de la rosca y el diámetro del eje sin fin fueron definidos en base al eje del motor y la precisión necesaria, es decir, el eje del sin fin debería ser como mínimo 1,5 mm más grueso que el eje del motor (lo que nos deja unas paredes mínimas de 0,75 mm). El paso del eje sin fin determina la precisión del sistema ya que un paso más fino nos dará menor desplazamiento lineal por unidad de vueltas como así también más fuerza para empujar la jeringa con fluido incluido.



Dicho paso define cual es el desplazamiento lineal por unidad de rotación: en este proyecto defino que el paso a utilizar es M6 (métrica 6 x 1) lo cual nos dice que el diámetro del eje es de 6mm, suficiente para que el eje del motor quede incrustado dentro del mismo y un paso de 1mm lo cual nos dice que por cada rotación el desplazamiento lineal será de 1mm.

Como segunda tarea, el soporte de la jeringa debe ser paralelo al eje de desplazamiento, por ello se diseñó el soporte sobre el eje espiralado.





Junto a una “chaveta” que permita la sujeción de una tuerca podemos hacer presión sobre el apoyo del émbolo.

Todo este prototipo pudo ser desarrollado gracias a una impresora 3D ya que mecanizar todos los elementos será muy laborioso.

Como producto final obtuvimos un prototipo que obtiene como entrada tensión aplicada a un servomotor y como salida fluido inyectado.

Azul: Jeringa de 3ml

Verde: apoyo de émbolo

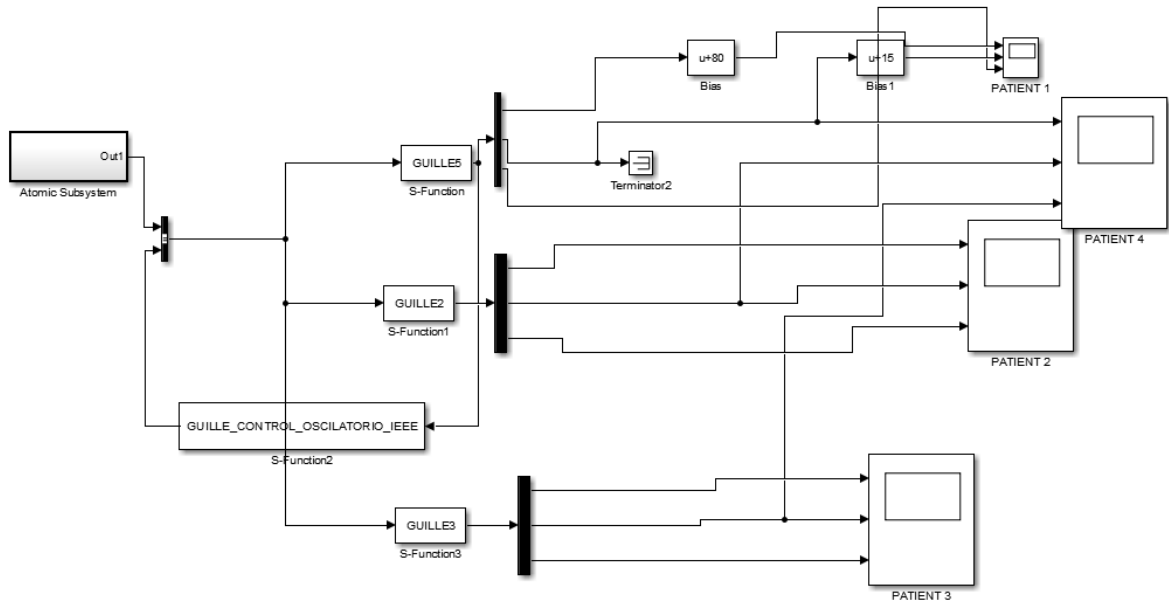
Rojo: Soporte de tuerca

Naranja: Motor y eje

Control oscilatorio MatLab (SimuLink)

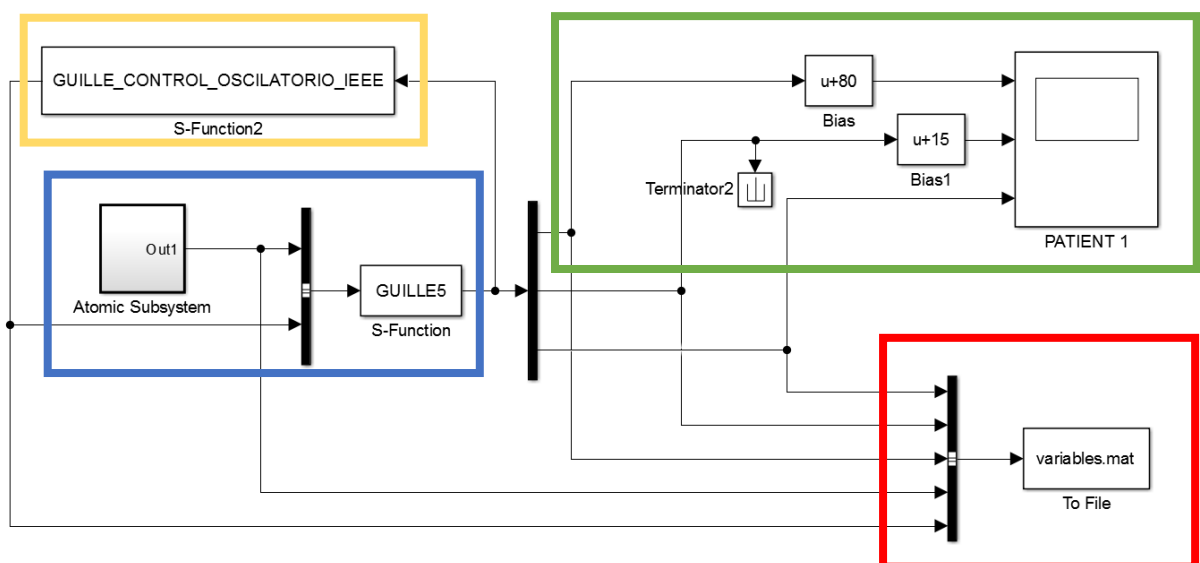
Uno de los elementos más importantes (o se podría decir el principal elemento) de este proyecto es la simulación bajo la plataforma de SimuLink de Matlab. Esta simulación fue desarrollada por el Dr. Guillermo Cocha basándose en el algoritmo matemático de control desarrollado por el Dr. Carlos Amorena y el Dr. Carlos D'Attellis. (BSPC-D-17-00864).

Dicha simulación se torna compleja y extensa (varias simulaciones simultaneas y varios pacientes) para ser implementada en este proyecto.



Por ello se decidió focalizarse en un solo paciente y una simulación puntual para comprobar el correcto funcionamiento.

El diagrama reducido de la simulación de convirtió al siguiente:



Este nuevo diagrama reducido se resume en cuatro bloques:

- El control oscilatorio (amarillo) es el encargado de ejecutar el algoritmo de control. Este bloque contiene tanto las ecuaciones del modelo mínimo de Bergman (ver BSPC-D-17-00864) como así también el control PID utilizado.
- El modelo del cuerpo humano (azul) el cual simula tanto el aporte de insulina por ingesta de alimento como el aporte de insulina del control a implementar.
- La visualización de los valores inyectados y la glucosa en sangre (verde)
- Y por último un bloque (rojo) utilizado para exportar los datos de dicha simulación a un archivo manejable a nivel computacional. Dichos datos son utilizados durante el proyecto para la validación de las simulaciones “reales” obtenidas.

Simulación del sistema oscilatorio por computadora

De los datos obtenidos de la simulación de Matlab, se representó la ecuación del sistema en computadora (controlador oscilatorio encargado de la inyección de insulina) y se realizó una simulación para verificar su funcionamiento.

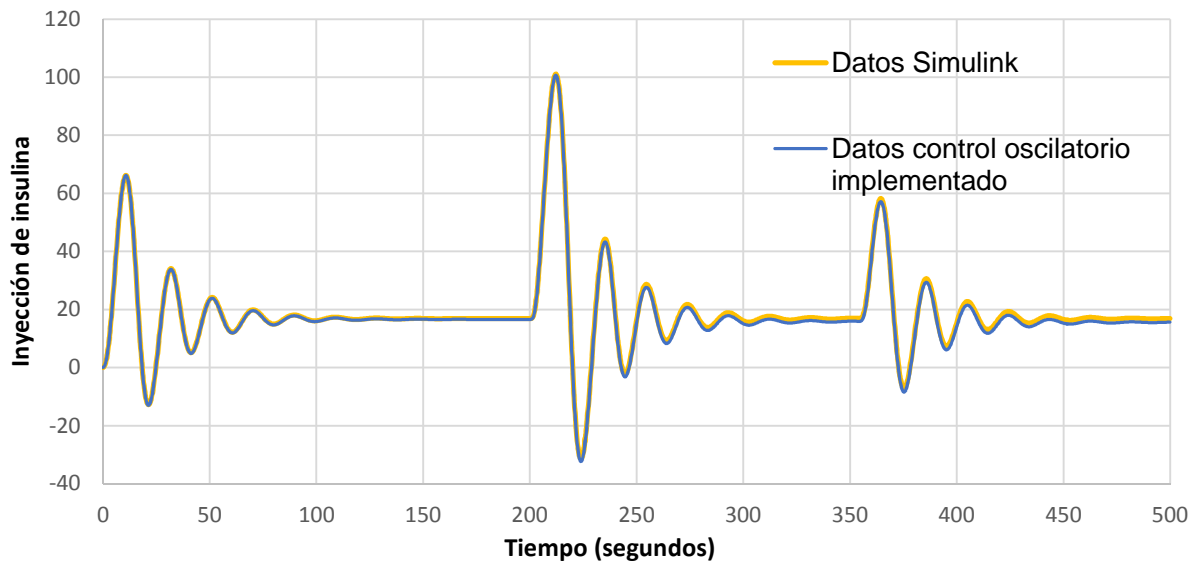
La ecuación de control oscilatorio a implementar es la siguiente:

$$\frac{-V * P3}{(u1 + Gb)} * \left\{ \begin{aligned} & \left[1000 * \frac{u1 - Gb}{u1 + 10 * Gb} \right] \\ & + [K1 * (u1 + Gb)] \\ & + [K2 * (-P1 * u1 - u2 * (u1 + Gb))] \\ & - K3 \{ [P1^2 * u1 + P2 * u2 * (u1 + Gb) - u1 * P2 * u2 - Gb * P3 * u3] \\ & + [(P1^2 + P1 * u2 - P2 * u2 - P3 * u3) * (-P1 * u1 - u2 * (u1 + Gb))] \\ & + [(P1 * (u1 + Gb) - P2 * u2 - P2 * Gb) * (-P2 * u2 + P3 * u3)] \\ & + [(-P3 * u3 - Gb * P3) * (N * (u3 + Ib))] \} \end{aligned} \right\}$$

Donde los parámetros son:

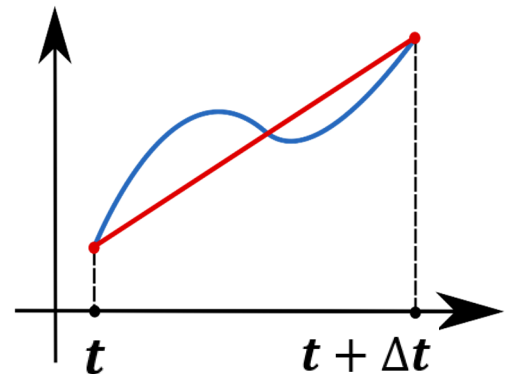
- u1, u2 y u3 son las variables temporales de la infusión de insulina o las variables de entrada del sensor.
- P1, P2 y P3 son parámetros dependientes del paciente
- K1, K2 y K3 son los parámetros del controlador PID implementado
- Gb es la concentración basal de glucosa en sangre
- Ib es la concentración basal de insulina en sangre
- V es el volumen plasmático
- N es la relación de transferencia de insulina

Una vez obtenida y entendida la ecuación que controla la inyección de insulina nos encontramos que todo el sistema debe ser integrado, por lo tanto, se realizó un ajuste de esta ecuación haciendo algunos cambios en dichas integraciones. El resultado de estas simulaciones es el siguiente:



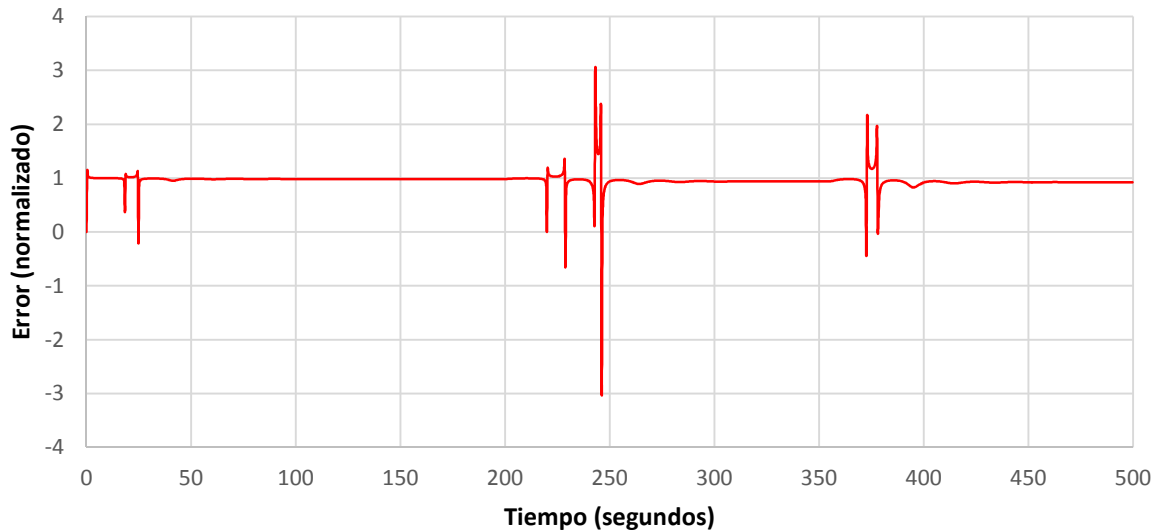
Una principal diferencia entre el controlador implementado en Matlab Simulink y la simulación utilizada para representar el controlador oscilatorio a implementar en hardware es la integración de la ecuación. El controlador oscilatorio entrega, a su salida, la integral de la ecuación antes mostradas, esto nos obliga a realizar una aproximación de la integración de manera discreta para poder implementarlo.

Dicha aproximación se realiza con el método de integración por trapecios:



$$\int_t^{t+\Delta t} f(x) dx \approx \Delta t \cdot \frac{f(t) + f(t + \Delta t)}{2}$$

Esta aproximación trae aparejado un error el cual se puede deducir de la comparación de la salida simulada con la implementada en nuestro controlador.



Del análisis de este gráfico de error podemos concluir algunas cosas, la primera es que la aproximación por trapecios es bastante aceptable para ser utilizada en nuestra implementación y la segunda es que el error crece considerablemente cuando la salida del controlador cambia su signo. Esto no es un factor para tener en cuenta ya que la representación de los valores negativos no debe ser interpretada como “una quita de insulina” sino como una anulación de la salida del controlador, es decir que no se debe inyectar insulina en ese tiempo.

Simulación del cuerpo humano (sensor)

La falta de un sensor que mida glucosa en sangre es uno de los problemas con los que debemos lidiar, pero no debe ser un impedimento para el desarrollo de este proyecto.

Por tal motivo se decide “simular” este sensor por medio de una computadora y una comunicación UART con el microcontrolador.

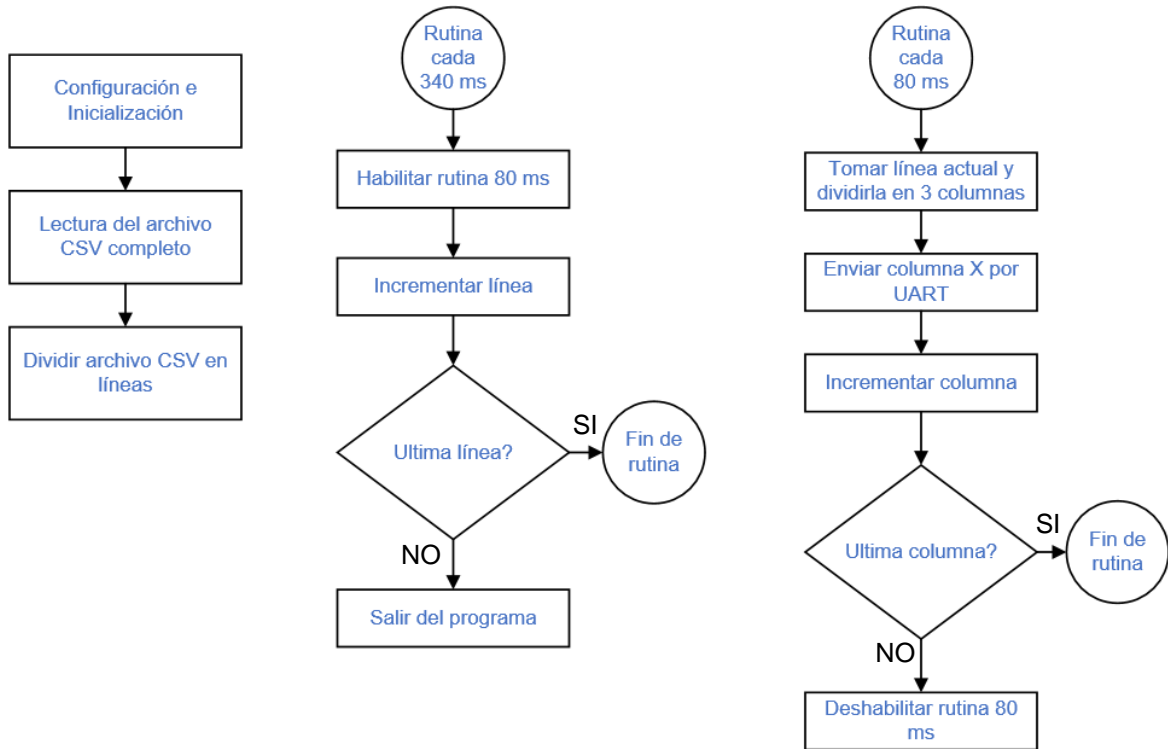
Esta simulación del sensor debe darnos tres variables como entrada de nuestro controlador: u1, u2 y u3.

De la simulación por SimuLink obtuvimos estas variables y se representaron en un archivo CSV (el inglés comma-separated values) el cual es leído por un software implementado bajo la plataforma de JavaScript.

1	+000.0000000;	+000.0000100;	+000.0000000
2	+000.0031892;	+000.0000989;	-000.4652151
3	+000.0235894;	+000.0000970;	-000.9087192
4	+000.0720193;	+000.0000943;	-001.3250075
5	+000.1543493;	+000.0000910;	-001.7081260
6	+000.2730840;	+000.0000870;	-002.0516584
7	+000.4284764;	+000.0000824;	-002.3488965
8	+000.6193102;	+000.0000774;	-002.5931014
9	+000.8434457;	+000.0000720;	-002.7777833
10	+001.0982004;	+000.0000665;	-002.8969473
11	+001.3806094;	+000.0000607;	-002.9452825
12	+001.6868969;	+000.0000551;	-002.9182837
13	+002.0036660;	+000.0000495;	-002.8123434

Cada fila es separada por un salto de línea identificado como “\n” y cada columna de datos es separada por el símbolo “;”. Esto nos permite saber dónde comienza y termina cada dato.

A continuación, se detalla un diagrama de flujo explicativo de la rutina de simulación del sensor:



Al utilizar una comunicación UART asincrónica debemos implementar un protocolo de comunicación para que el controlador pueda reconocer cada valor.

Esto implica que el dato debe ser enviado en un formato preestablecido tanto su magnitud como su signo ya que ambos son esenciales para el correcto funcionamiento.

La trama establecida para transmitir los datos es la siguiente:

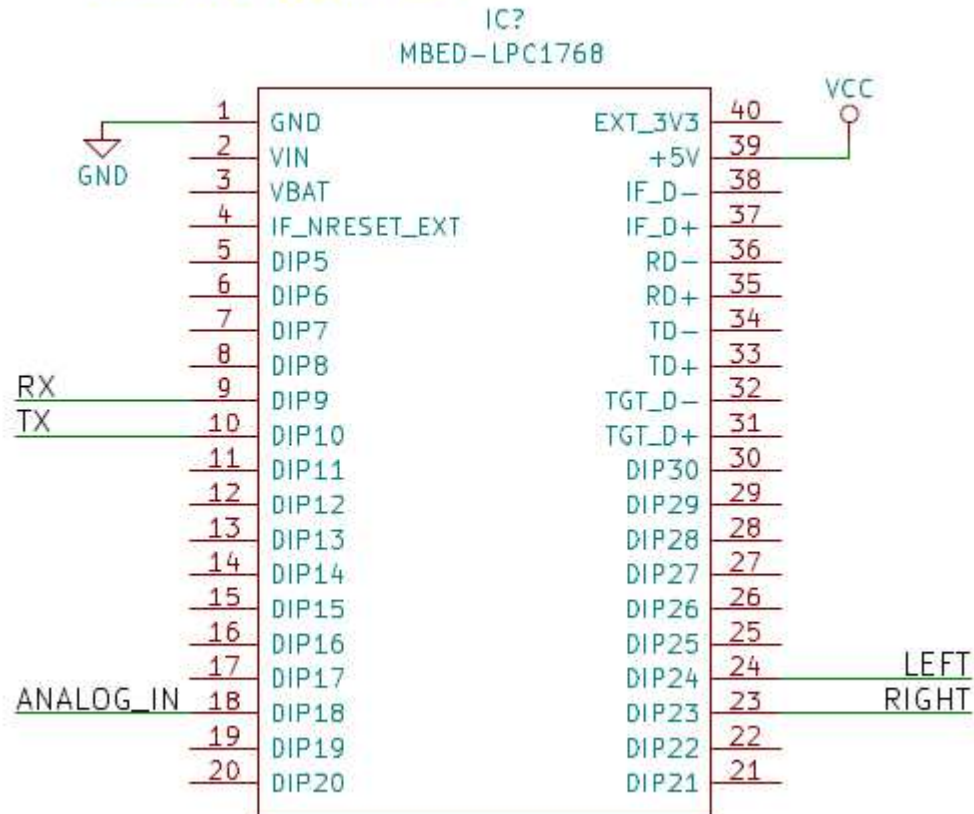
- El primer carácter es el de signo
- Desde el carácter dos hasta el cuatro se compone el valor entero del dato
- El quinto carácter es el separador de decimales
- Desde el carácter seis al doce se componen los decimales del dato

Todos ellos son convertidos en el microcontrolador a una variable del tipo "float".

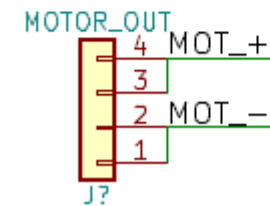
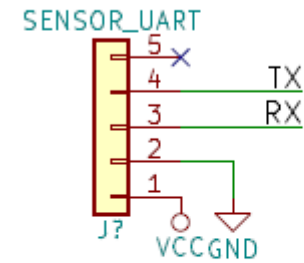
Signo	Entero			Punto	Decimales						
+/-	#	#	#	.	#	#	#	#	#	#	#

Diagramas circuitales

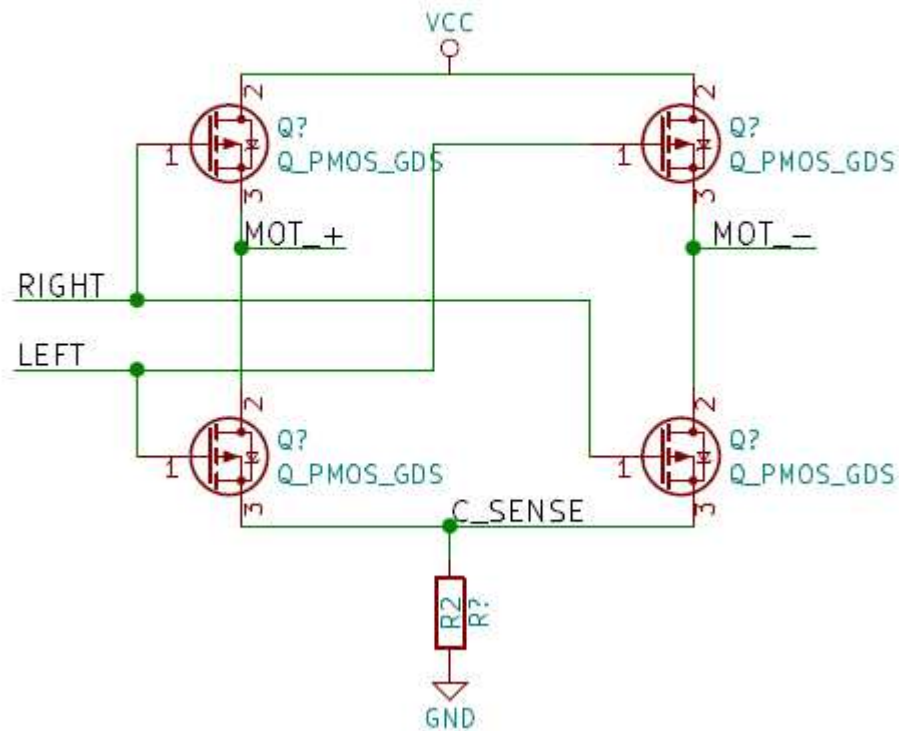
PLACA DE DESARROLLO



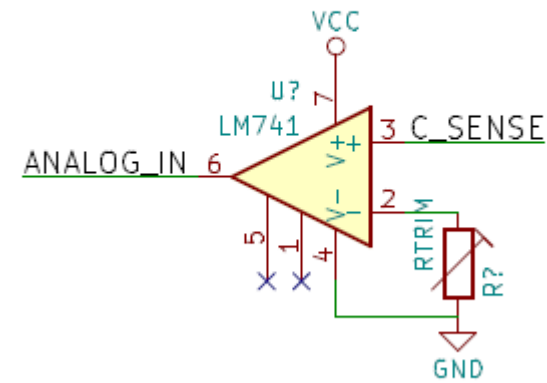
CONEXIONES EXTERNAS



PUENTE H

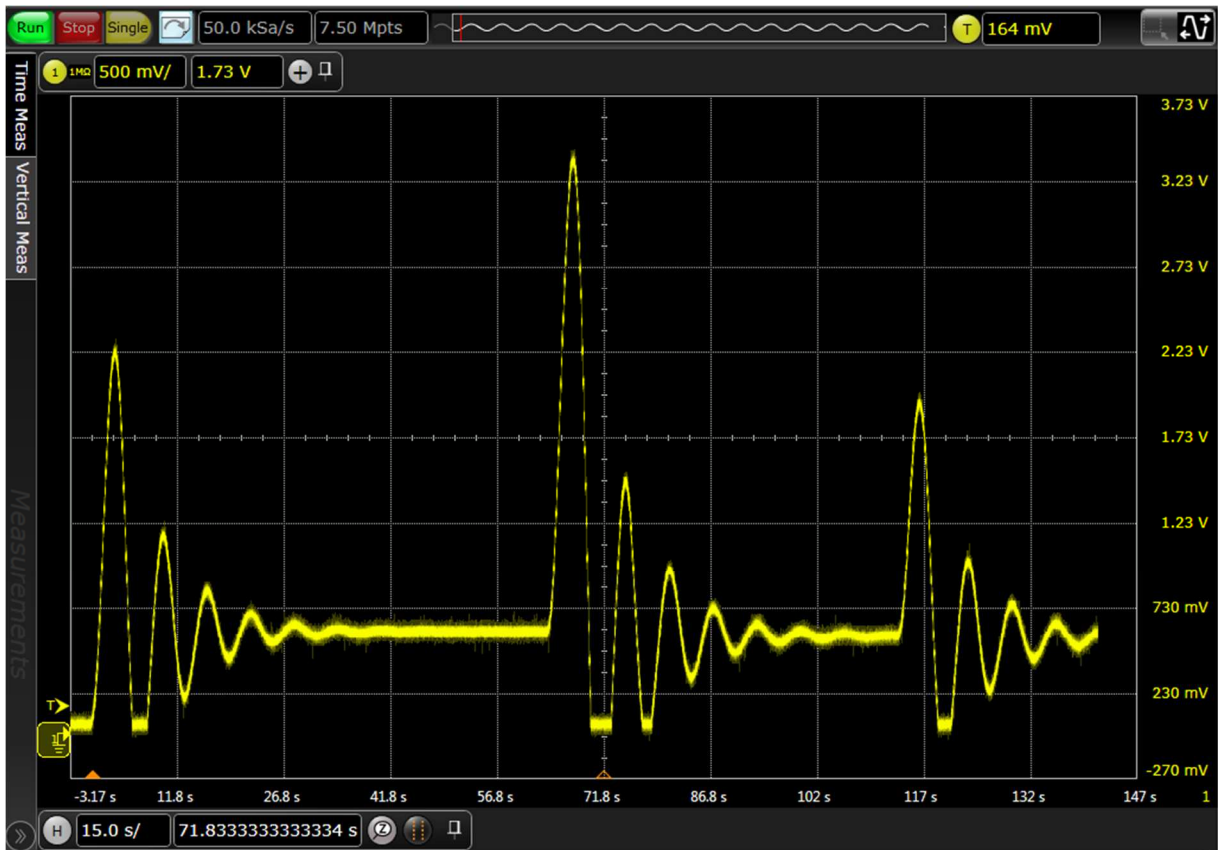


SENSOR DE CORRIENTE DEL MOTOR

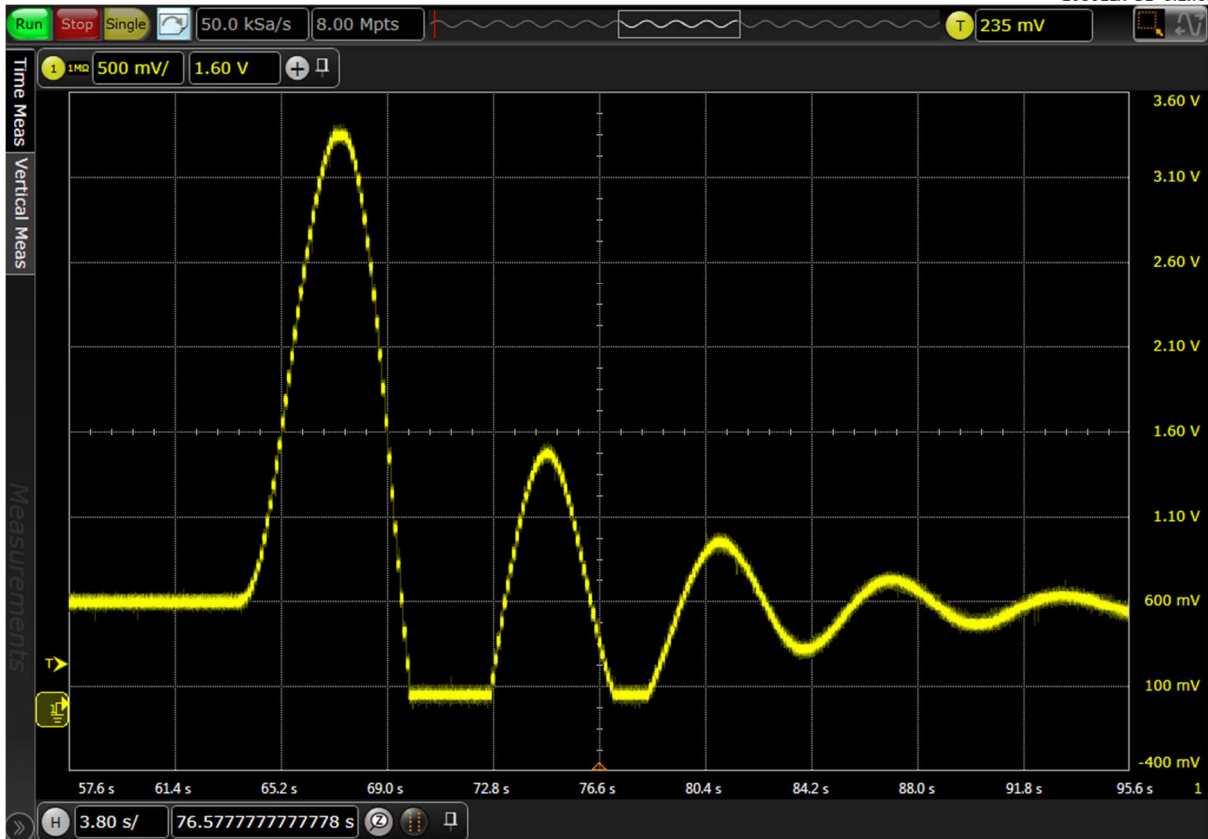


Pruebas de control

Una de las primeras pruebas que se realizó fué la representación discreta de la salida del controlador (nivel de inyección de insulina en sangre) como una salida analógica de nuestro microcontrolador. Esto nos permitió verificar en una primera aproximación que nuestro controlador respetará “la forma” de la oscilación resultante. Los resultados obtenidos fueron los siguientes:



Se observa que la forma de la onda coincide con lo simulado en MatLab. No se puede comparar, en estas validaciones, que los órdenes de magnitud coinciden. Para esto se implementa la validación por pesaje lo cual nos da una correlación directa con el volumen de insulina inyectado. También se observa el recorte de la componente negativa de la oscilación ya que no es admisible la “quita de insulina en sangre”.



Haciendo una visualización de una única oscilación se puede apreciar con más facilidad la discretización de los niveles que se convertirán en unidades de insulina en sangre o unidades de volumen de insulina inyectado como así también la parte donde el controlador indica que debe “quitarse” insulina o que deba entregar valores negativos de la misma. Esto último es recortado a valor cero o apagado dadas las indicaciones del sistema a desarrollar.

Validación

La validación del correcto funcionamiento del sistema es una parte importante de este proyecto ya que determina si el trabajo realizado para su desarrollo fue suficiente para lograr los objetivos previstos.

Dicha validación del sistema implica medir la cantidad de insulina inyectada luego de que el sistema sea estimulado.

Esta medición se realiza por medio de una balanza de la serie Timbeline de Denver Instruments la cual tiene un fondo de escala de 220 g y una resolución de 0,1 mg la cual lo hace más que suficiente para nuestras mediciones (toda la jeringa pesa 3 g lo cual nos da 30000 niveles de medición).

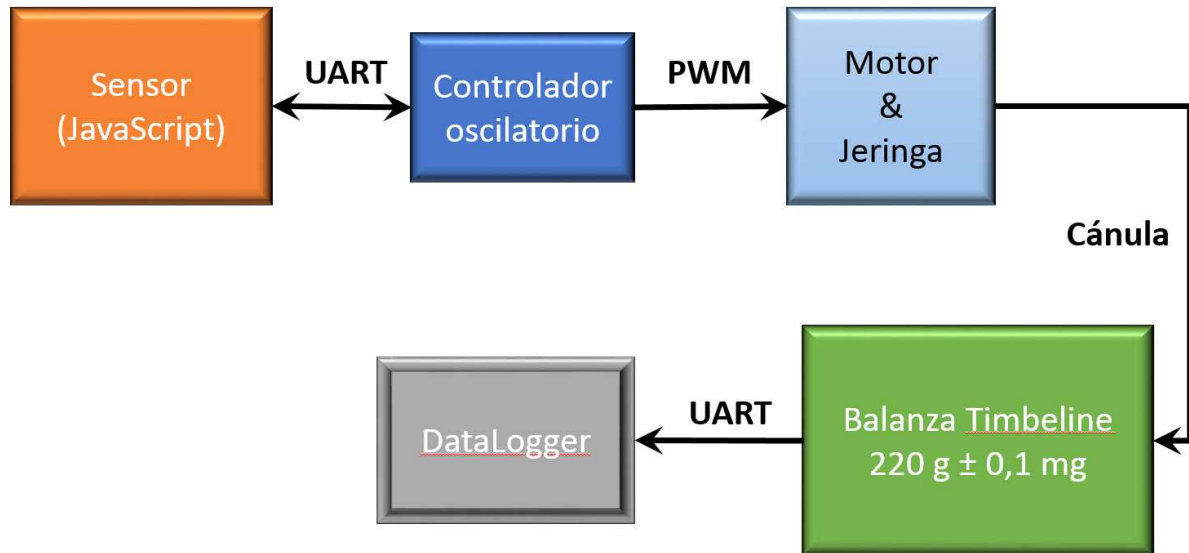
La simulación del sensor se realiza con una computadora portátil y se conecta al controlador por medio de una UART.

El controlador y el motor son conectados por una interfaz cableada que lleva la modulación PWM. Esta conexión fue mallada para evitar interferencias con la balanza de precisión.

Entre la jeringa y la balanza se utilizó una cánula (tubo de goma delgado) lo que nos permitió no tener influencia de capilaridad sobre el volumen a pesar.

El Datalogger fue implementado con otra computadora portátil con su conexión UART, independiente de la implementada en el sensor para evitar problemas de velocidad en el protocolo de comunicación.

El conexionado de las partes es como se muestra en la siguiente figura:

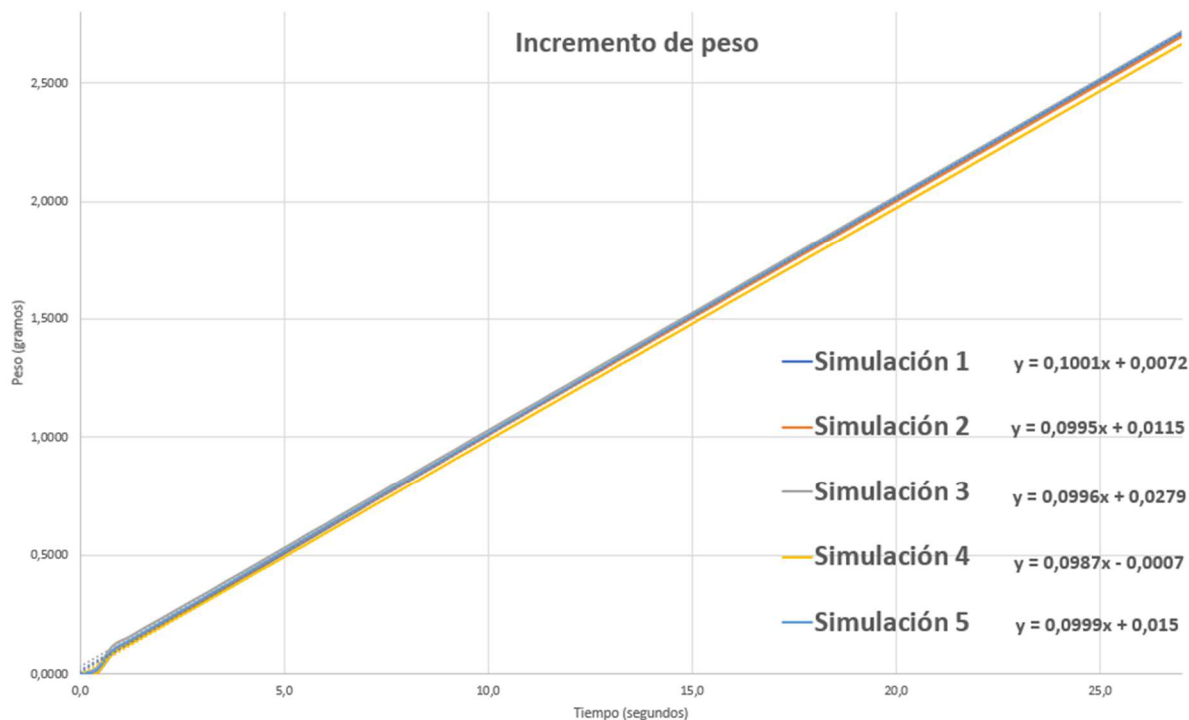


La validación implementada se compondrá de dos grupos:

- El primer grupo de datos de validación corresponde al desplazamiento continuo de toda la jeringa. Esto se realizará a velocidad constante con un escalón como entrada del sistema. Con esta validación esperamos obtener una respuesta continua en todo el trayecto de desplazamiento y también verificar la respuesta del sistema cuando pasa del estado de reposo al estado estacionario. Esta prueba también nos permite verificar que la mecánica del sistema no tiene variaciones a lo largo de todo el trayecto del embolo de la jeringa.
- El segundo grupo de datos se realiza tomando como entrada una simulación computarizada del cuerpo humano (implementada en JavaScript). Dicha entrada simulada nos dará una respuesta temporal de la cual ya tenemos la forma esperada. Estimamos comparar la respuesta medida con la respuesta matemáticamente calculada y esperada para validar, en este caso, el correcto funcionamiento del controlador oscilatorio implementado.

Primera validación:

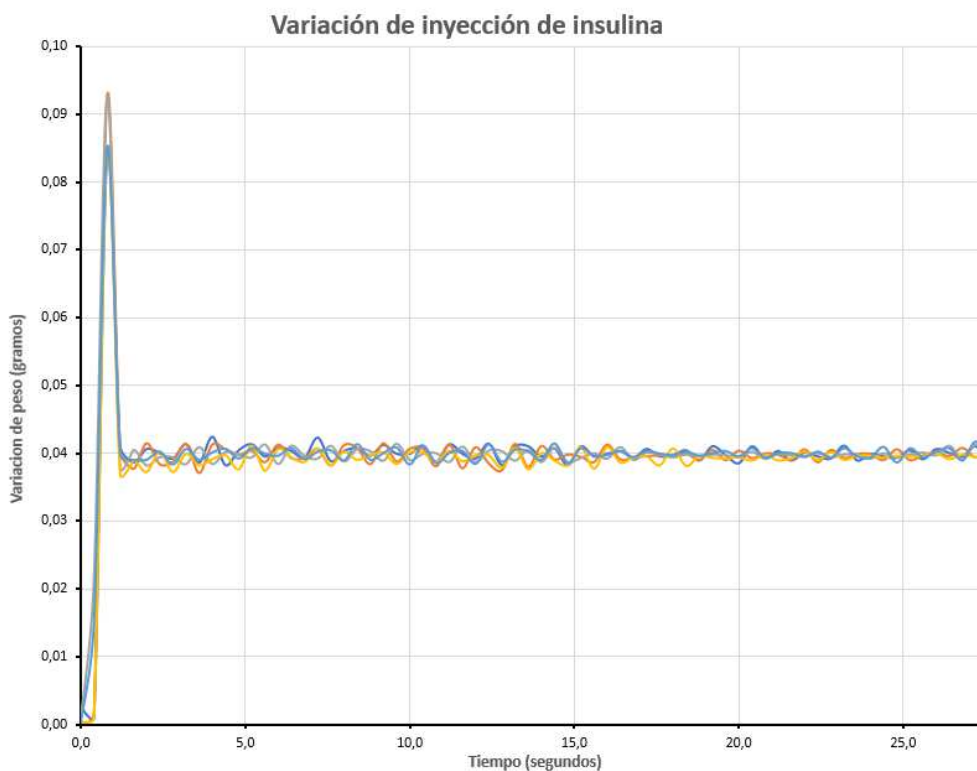
En esta validación del sistema se verifica el desplazamiento de la jeringa en respuesta a un escalón como entrada al sistema. Se realizan 5 simulaciones iguales y continuas con una velocidad de inyección constante. Los resultados son los siguientes:



Para cada una de las simulaciones se graficó una línea de tendencia y se calculó la pendiente de incremento del peso de la insulina inyectada. Cada uno da valores de las pendientes de las líneas de tendencia son muy próximos a 0,1 lo que nos da un incremento de 0,04 gramos por unidad de tiempo (el tiempo de muestreo de la simulación fueron 400 mS). De este grafico podemos intentar deducir dos cosas:

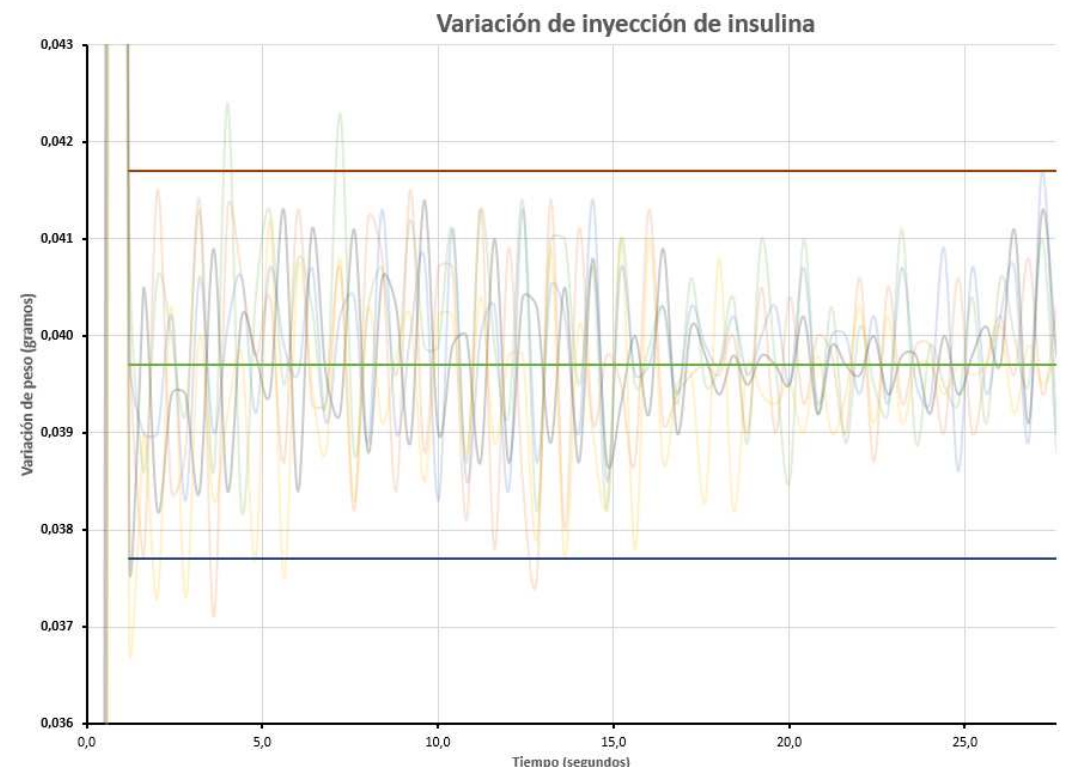
- La variación o diferencia de insulina inyectada por unidad de tiempo es constante a lo largo de todo el trayecto.
- La respuesta inicial del sistema, es decir desde que la jeringa está en reposo y comienza su movimiento constante, tiene forma oscilatoria

Para la primera deducción podemos graficar la variación del peso en función del tiempo y observar su forma:

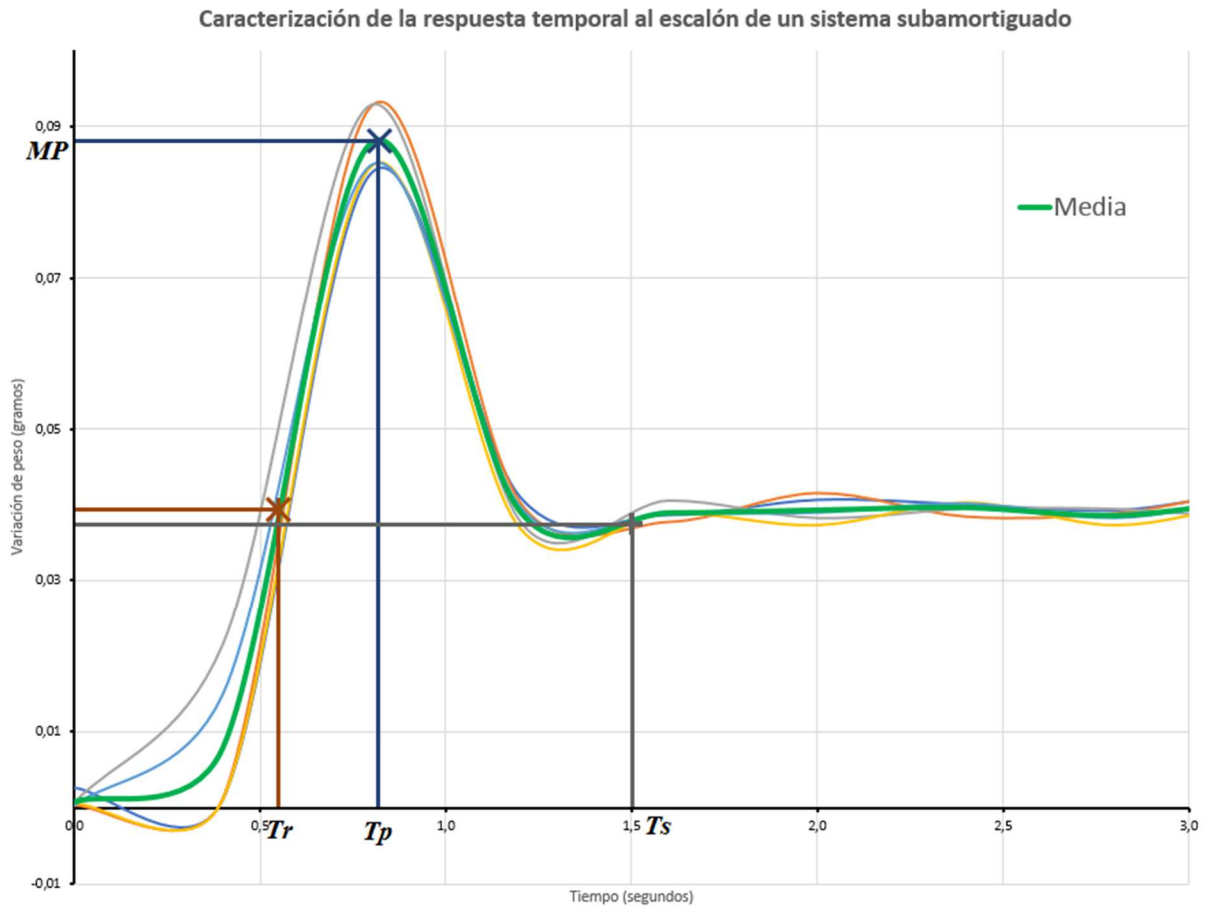


De esta medición se aprecia el desplazamiento constante de 40 miligramos por unidad de tiempo lo que se traduce a 40 microlitros por unidad de tiempo. Este valor es sumamente aceptable para el desarrollo en cuestión.

Haciendo un análisis más profundo podemos calcular el error del valor medio de estas mediciones:



Para la segunda deducción de los datos obtenidos veremos la respuesta inicial del sistema, es decir, desde el estado de reposo hasta el estado estacionario:



Para no tener cinco unidades a representar de cada uno de los valores que caracterizan esta respuesta al escalón, se realizó el análisis sobre una curva de la media de las cinco simulaciones.

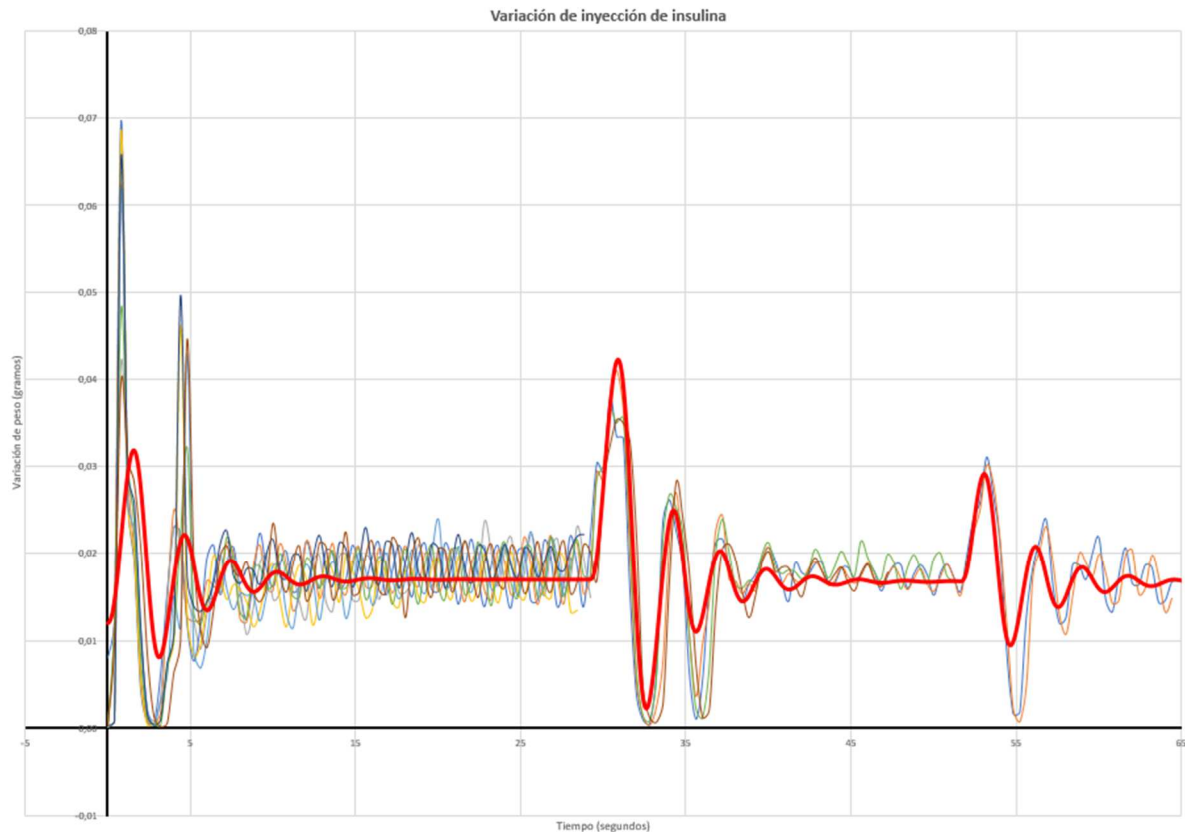
Esta caracterización nos da los valores de:

- $T_p = 0,82$ segundos
- $T_r = 0,55$ segundos
- $MP = 0,088$ gramos/segundos
- $T_s = 1,5$ segundos

Segunda validación:

En esta segunda prueba de validación de funcionamiento se realizaron varias mediciones utilizando el controlador para lo que fue implementado.

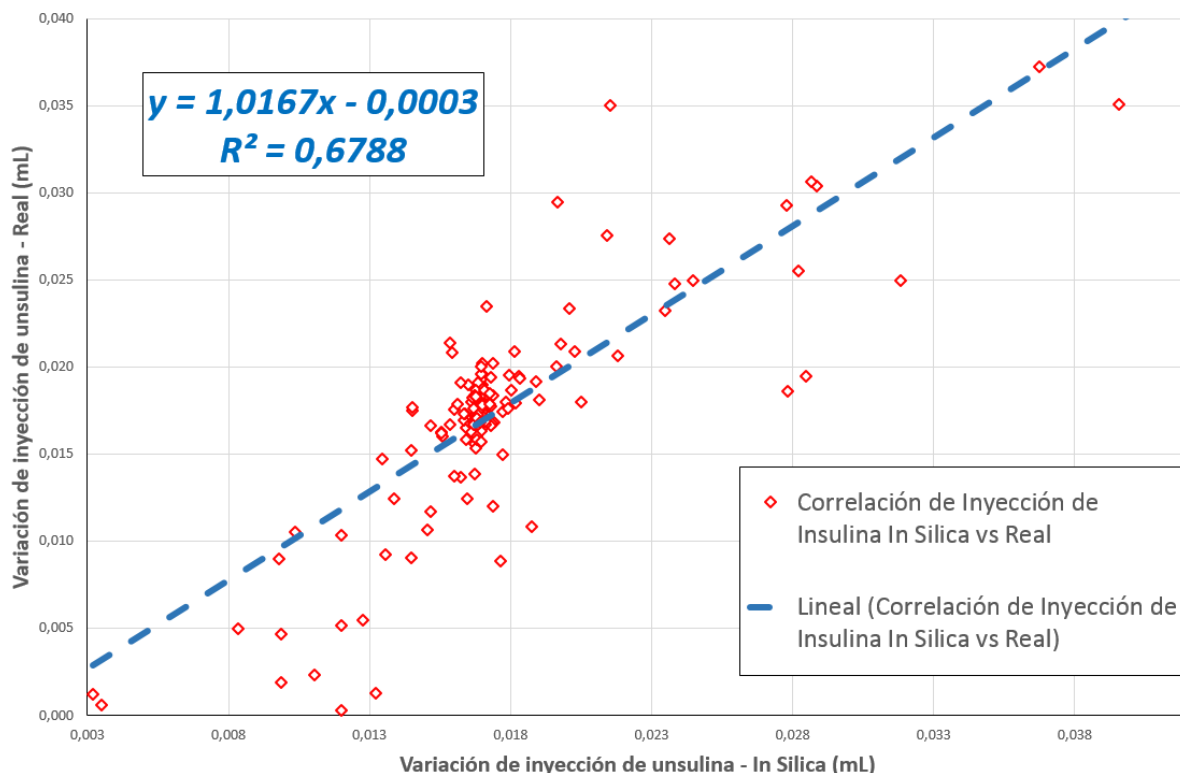
Los datos recopilados por el datalogger fueron representados en la siguiente gráfica:



Estas mediciones fueron muy satisfactorias y validan el funcionamiento del controlador implementado.

Cada una de ellas siguió la forma predicha con los valores matemáticamente calculados en la simulación en MatLab.

Al inicio de la gráfica se observa una gran diferencia en la amplitud de los valores obtenidos y los calculados. Esto se debe a que todo el sistema se encuentra en reposo al inicio de las pruebas y el controlador está ajustado para ajustarse a las variaciones de este, y esta variación inicial es muy grande ya que el valor de inyección constante se encuentra lejos del cero. Esto puede ser medido con el valor MP calculado en la primera validación que era de más de 120% del valor nominal.



También se realizó la correlación entre las variaciones medidas de inyección de insulina (el promedio de todas las pruebas) y la simulación “in silica” (en computadora). Podemos observar que se comporta de manera lineal, respondiendo a una recta de tendencia con una pendiente de aproximadamente 1 y ordenada al origen casi 0. Esto nos dice que el prototipo se comporta en gran medida igual al modelo simulado como era lo esperado.

Conclusiones

Como conclusiones de los datos adquiridos del servomotor y de la mecánica implementada, nos permitió calcular y proyectar la inyección de insulina a partir del valor matemático que entregaba el controlador oscilatorio. Este valor pudo ser convertido a una modulación por ancho de pulso, que luego fue convertida a velocidad, que luego se convierte a desplazamiento y por último a volumen inyectado. Sin todos estos datos solo nos basábamos en “predicciones” del comportamiento del sistema. De las validaciones del sistema a velocidad constante podemos concluir que el desplazamiento de la jeringa es constante, invariante a lo largo de todo el recorrido y repetible. Los valores de los tiempos de respuesta y de estabilización son muy aceptables para nuestro desarrollo inicial.

Los valores de sobrepico en respuesta a una entrada de escalón fueron mayores a los esperados (se esperaba que fueran menores al 40%). Esto se puede mejorar ajustando los valores de las constantes del controlador PID sin afectar el sistema de control de insulina.

Podemos también concluir que la forma de la respuesta oscilatoria es fiable y representa el comportamiento del algoritmo matemático de control desarrollado por el Dr. Carlos Amorena y el Dr. Carlos D´Attellis. (BSPC-D-17-00864).

Como conclusión principal podemos decir que el prototipo implementado responde de igual manera de las simulaciones matemáticas en las que nos basamos para todo el desarrollo.

Posibles mejoras

- Ajuste del controlador PID para evitar el sobrepico excesivo cuando el sistema sale del reposo.
- Cambiar el sistema mecánico por una bomba presostática. Esto debe ser evaluado medicamentosa para saber si es viable la implementación en un ser humano.
- El desarrollo de un sensor real es una gran mejora al proyecto global lo que requerirá otras pruebas sobre seres vivos.

Anexos

Anexo 1- Diabetes tipo 1

La diabetes tipo 1 es causada por una pérdida o disfunción de las células productoras de insulina, llamadas células beta pancreáticas.

El daño a las células beta da como resultado una ausencia o producción insuficiente de la insulina producida por el cuerpo. La mayoría de los casos de diabetes tipo 1 tienen un origen autoinmune, y el sistema inmunitario ataca y destruye por error a las células beta. Como la insulina es necesaria para mantener la vida, hay que reemplazar la insulina faltante. La insulina de reemplazo se administra mediante inyección utilizando una jeringa o una bomba de insulina, que entrega la insulina debajo de la piel.



La insulina actúa como una llave que abre las puertas de las células para que pueda entrar la glucosa en su interior. Sin la insulina, la glucosa no puede acceder al interior de las células (las puertas están "cerradas" y no se dispone de una llave para abrirlas); por lo tanto, la glucosa permanece en el torrente sanguíneo. Consecuentemente, la concentración de azúcar en sangre es más alta de lo normal. Una concentración elevada de azúcar en sangre es un problema porque puede provocar diversos problemas de salud.

La diabetes tipo 1 no se puede prevenir, y no hay forma alguna de predecir quién la va a desarrollar. No hay nada que un padre o un niño hayan podido hacer que pueda causar esta enfermedad.

Una vez que una persona contrae una diabetes tipo 1, esta enfermedad no desaparece y requiere tratamiento de por vida. Los niños y adolescentes con una diabetes tipo 1 dependen de inyecciones de insulina diarias o de una bomba de insulina para controlar las concentraciones de glucosa en sangre.

Anexo 2 - Control digital PID

Un controlador PID (Controlador Proporcional-Integral-Derivativo) es un mecanismo de control por realimentación utilizado en sistemas de control. Este calcula la desviación o error entre un valor medido y un valor deseado.

El algoritmo del control PID consiste en tres parámetros distintos:

- El valor Proporcional depende del error actual.
- El Integral depende de los errores pasados.
- El Derivativo es una predicción de los errores futuros.

La suma de estas tres acciones es usada para ajustar al proceso por medio de un elemento de control como la posición de un servomotor o la potencia suministrada a una resistencia calefactora.

El peso de la influencia que cada una de estas partes tiene en la suma final, viene dado por la constante proporcional, el tiempo integral y el tiempo derivativo, respectivamente.

Cuando no se tiene conocimiento del proceso se considera que el controlador PID es el controlador más adecuado. Ajustando estas tres variables en un algoritmo de control, el controlador puede proveer una acción de control diseñado para los requerimientos del proceso en específico. La respuesta del controlador puede describirse en términos de la respuesta del control ante un error, el grado el cual el controlador sobrepasa el punto de ajuste, y el grado de oscilación del sistema.

La utilización de un controlador PID no garantiza control óptimo del sistema o la estabilidad de este.

Para el correcto funcionamiento de un controlador PID que regule un proceso o sistema se necesita, al menos:

- Un sensor, que determine el estado actual del sistema.
- Un controlador, que genere la señal que maneje al actuador.
- Un actuador, que modifique al sistema de la manera que se desea.

El sensor proporciona una señal analógica o digital al controlador, la cual representa el punto actual en el que se encuentra el proceso o sistema. La señal puede representar ese valor en tensión eléctrica, intensidad de corriente eléctrica o frecuencia. En este último caso la señal es de corriente alterna, a diferencia de los dos anteriores, que también pueden ser con corriente continua.

El controlador recibe una señal externa que representa el valor que se desea alcanzar. Esta señal recibe el nombre de punto de referencia, valor deseado o set point, la cual es de la misma naturaleza y tiene el mismo rango de valores que la señal que proporciona el sensor.

El controlador resta la señal de punto actual a la señal de punto de referencia, obteniendo así la señal de error, que determina en cada instante la diferencia que hay entre el valor deseado (referencia) y el valor medido (sensor). La señal de error es utilizada por cada uno de los tres componentes del controlador PID. Las tres señales sumadas, componen la señal de salida que el controlador va a utilizar para manejar al actuador. La señal resultante de la suma de estas tres se llama variable manipulada y no se aplica directamente sobre el actuador, sino que debe ser transformada para ser compatible con el actuador utilizado.

Anexo 3 - Algoritmos PID discretos

El PID, en sus distintas versiones constituye la solución mas aceptada en los problemas de control en la industria. Se calcula que el 95% de controladores industriales analógicos o digitales, constituyen el algoritmo del PID.

Se puede decir que el PID es el regulador natural por excelencia. Comportándose como un ser racional ante una toma de decisión: tomando en cuenta el estado actual (P - proporcional), la historia pasada (I - integral) y un pronóstico del futuro (D - derivativo) del error o desviación del comportamiento deseado.

Se pretenderá lograr que el bucle de control corrija eficazmente y en el mínimo tiempo posible los efectos de las perturbaciones externas.

El PID discreto tiene dos formas principales conocidas como posicional y de velocidad.

La siguiente ecuación ofrece la forma posicional del algoritmo PID discreto:

$$u(t) = Kp \left[e(t) + \frac{Tc}{Ti} \sum_{i=1}^t e(i) + \frac{Td}{Tc} [e(t) - e(t-1)] \right] + u(0)$$

Donde:

- $u(t)$ -variable de control, generalmente posición.
- $u(0)$ es la posición inicial.
- Kp , Ti y Td - Ganancia, tiempo integral y tiempo derivativo.
- Tc - Período de control.

La forma de velocidad del PID discreto se obtiene fácilmente a partir de la ecuación anterior, calculando $u(t)$ menos $u(t-1)$, con lo que se llega a:

$$\Delta u(t) = Kp \left[(e(t) - e(t-1)) + \frac{Tc}{Ti} e(t) + \frac{Td}{Tc} (e(t) - 2e(t-1) + e(t-2)) \right]$$

Donde

$$\Delta u(t) = u(t) - u(t-1)$$

O

$$u(t) = u(t-1) + \Delta u(t)$$

El algoritmo posicional requiere la verificación de la sumatoria correspondiente al modo integral para evitar saturación. En el caso de la velocidad no se requiere esta verificación, pero por otra parte hay que establecer los límites a los incrementos de la variable de control.

En el algoritmo posicional se requiere el conocimiento de la posición inicial del actuador. En el de velocidad, si el actuador es de tipo incremental, como por ejemplo un motor de pasos o cuando se tiene un estado inicial del actuador, no se requiere conocer la posición inicial. Si, por otra parte, se utiliza un algoritmo de velocidad asociado a un actuador posicional, entonces de todas formas se requiere la posición inicial para poder efectuar el cálculo.

En el algoritmo posicional se puede intuir el significado de los parámetros Kp , Ti y Td semejante al de los reguladores PID analógicos, conocidos tradicionalmente en la industria. En el caso del algoritmo de velocidad, el significado de dichos parámetros se desvirtúa.

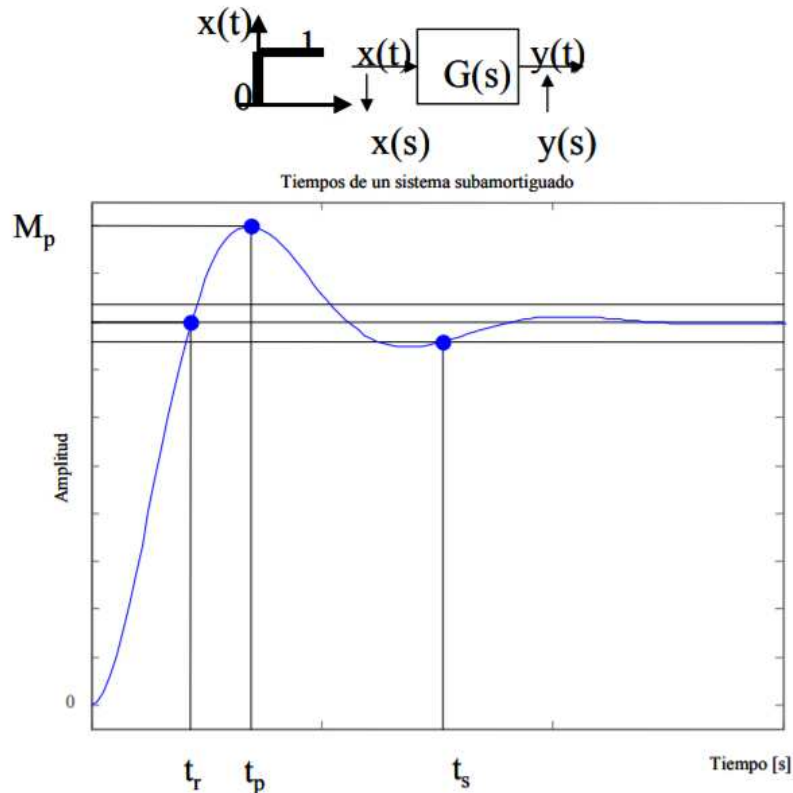
El algoritmo de velocidad trabaja con la segunda derivada del error, lo cual es inconveniente en presencia de ruido y, por otra parte, puede dar lugar a un comportamiento imprevisto e inadmisibles como, por ejemplo, que se mande a cerrar una válvula cuando el error está aumentando, requiriéndose lo contrario.

En resumen, se recomienda el algoritmo de posición, aun cuando requiere un cuidado especial con la sumatoria, para evitarse lo que se conoce como desborde (wind up)

del nodo integral. El algoritmo de velocidad debe utilizarse cuando los actuadores son de tipo incremental, por ejemplo, motores de pasos.

Anexo 4 - Caracterización de la respuesta temporal al escalón de un sistema subamortiguado

Dado que nuestro sistema de control oscilatorio predice (basado en las simulaciones por computadora presentadas en BSPC-D-17-00864) una respuesta oscilatoria, es de suma importancia parámetros temporales de un sistema subamortiguado cuando es excitado a su entrada con un escalón. Esto se debe a que, en muchas plantas o procesos físicos, sus dinámicas son aproximadas a esta FDT. Por lo que este modelado simplificado permite conocer características de la estabilidad y de la naturaleza de la respuesta del régimen transitorio. Adicionalmente, no sólo permite analizar o predecir el comportamiento temporal, sino que, a veces, los requisitos de diseño de los reguladores de control emplean definiciones dadas por estos parámetros, como por ejemplo el valor de sobreoscilación. Por todas estas razones, se trata de caracterizar mediante medidas de tiempo y de valor de pico, la salida de un sistema subamortiguado ante una excitación de escalón unitario.



Tiempo de establecimiento, t_s

El tiempo que necesita el sistema para alcanzar el régimen permanente con un error del 5% o del 2% del valor final, depende básicamente de la componente envolvente de la señal de salida. Simplificando y no considerando el efecto senoidal, el 95% de la señal se alcanzará cuando la envolvente valga 0.05 ó 0.02, según criterio del 5% o el 2% del valor final.

Tiempo de pico, t_p

Este valor se dará cuando se alcance el valor máximo de la amplitud de la señal de salida. Tomando la primera derivada de la expresión analítica de la respuesta al escalón unitario e igualando a cero, permitirá definir los tiempos de máximos y mínimos de amplitud de la señal

Sobreoscilación, M_p

Existe sobreoscilación si el factor de amortiguamiento está entre 0.707 y 0. La sobreoscilación se dará en el tiempo de pico (t_p) y corresponderá con el valor ponderado entre la máxima amplitud de salida respecto al valor de la misma en el régimen permanente. Al tener un sistema de ganancia estática k , la respuesta al escalón unitario, después de finalizado el régimen transitorio, también será k veces de amplitud.

Tiempo de subida, t_r

Muy empleado en los catálogos de componentes electrónicos en el apartado de las características dinámicas. El tiempo de subida es el intervalo que tarda el sistema o el dispositivo en pasar del 10% al 90% en una de sus señales. Para el tratamiento matemático y con el objeto de simplificar la expresión, se considerará el paso de tiempo entre el 0% al 100% del valor final, esto es, la primera vez la señal alcanza su valor final.

Anexo 5 - Mbed

Mbed es una plataforma y un OS para dispositivos “conectados a Internet” basados en microcontroladores ARM Cortex-M de 32 bits. Tales dispositivos también se conocen como dispositivos de IoT (Internet of Things - Internet de las cosas). El proyecto es desarrollado en colaboración con ARM y sus socios tecnológicos.

Las aplicaciones para las distintas plataformas Mbed se pueden desarrollar utilizando el IDE (Integrated Development Environment o Entorno de Desarrollo Integrado) en línea de Mbed, un editor y compilador de código en línea gratuito (<https://os.mbed.com/compiler>). Solo es necesario tener instalado un navegador web, ya que el proyecto se compila en la “nube”, es decir, en un servidor remoto, utilizando el compilador ARMCC C/C ++. El IDE de Mbed proporciona espacios de trabajo privados con la capacidad de importar, exportar y compartir código con el control de versión de Mercurial distribuido, y puede usarse también para la generación de la documentación del código. Las aplicaciones también se pueden desarrollar con otros entornos de desarrollo como Keil μ Vision, IAR Embedded Workbench y Eclipse con sus herramientas integradas GCC ARM.

Mbed OS proporciona la plataforma de software Mbed y las herramientas para crear el firmware del microcontrolador que se ejecuta en dispositivos IoT. Consiste en las bibliotecas centrales que proporcionan los controladores periféricos de microcontroladores, redes, RTOS y el entorno de ejecución, herramientas de compilación y scripts de prueba y depuración. Este OS es catalogado como Open Source bajo una licencia Apache 2.0, lo cual indica que se puede utilizar Mbed OS en proyectos comerciales y personales.



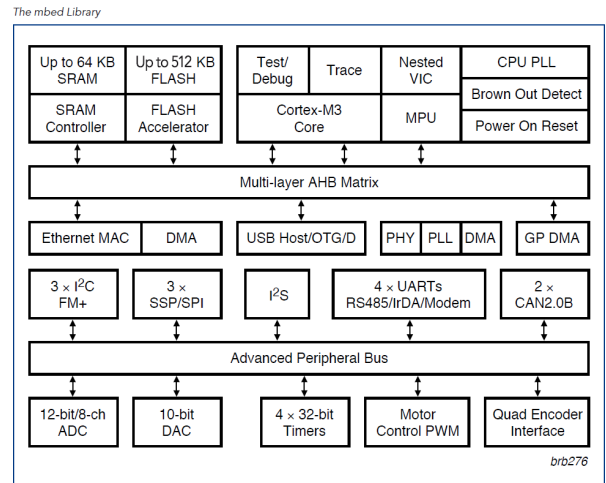
Una base de datos proporciona bibliotecas de controladores para componentes y servicios que se pueden conectar a los microcontroladores para crear un producto final. Dichas bibliotecas pueden ser generadas por otros usuarios del compilador online y también pueden disponer de revisiones que permite la colaboración.

Al ser un OS diseñado para IoT posee mayor compatibilidad y bibliotecas para su uso para los distintos tipos de tecnologías de conectividad como los son BlueTooth LE, NFC, RFID, WiFi, Ethernet, GSM & GPRS Celular, LoRa LPWAN y 6LoWPAN.

Como todo OS, Mbed OS debe ser cargado en una placa de desarrollo o demostración para la plataforma Mbed. La placa de microcontroladores Mbed (comercializada como "mbed NXP LPC1768") es una board de demostración basada en un microcontrolador NXP. El Mbed OS recibió el primer premio en la categoría anual de Software / Herramientas Integradas de los Premios a la Innovación EDN en 2010.

Características de Mbed LPC1768

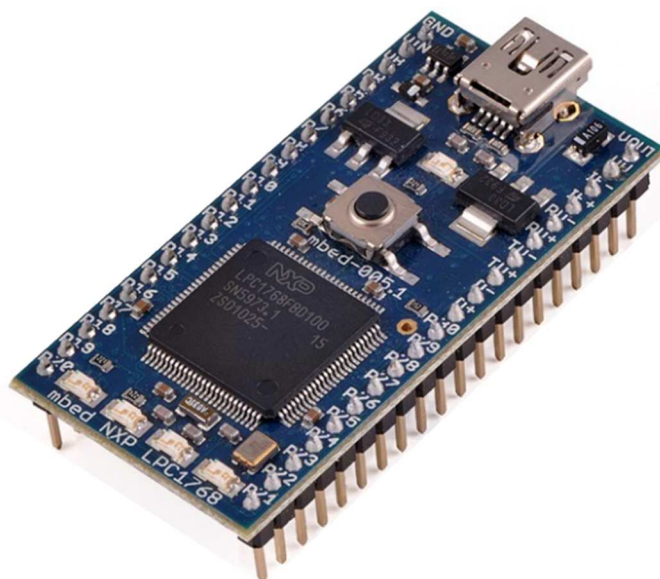
- Conveniente Form Factor: DIP de 40 pines con paso de 0,1"
- Programación de Drag&Drop, con el Board simulando una unidad de disco USB
- Un procesador Cortex-M3 (el mejor en su clase)
- ARM de 100 MHz con 64 KB de SRAM, 512 KB de Flash
- Ethernet, USB OTG
- SPI, I2C, UART, CAN
- GPIO, PWM, ADC, DAC
- Herramientas Online fáciles de usar
- Entorno de programación C/C++ basado en web
- Utiliza el motor de compilación ARM RealView
- Desarrollo basado en API utilizando bibliotecas con interfaces intuitivas
- Ayuda integral y comunidad online



Beneficios

- Se puede comenzar de inmediato, sin nada que instalar
- Podemos programar rápido usando APIs de alto nivel online
- Escritura de código limpio y compacto que es fácil de modificar
- La posibilidad de iniciar sesión desde cualquier lugar, en Windows, Mac o Linux

Se lanzaron varias versiones de la misma placa, como NXP LPC2368 (ARM7TDMI-S), NXP LPC1768 (Cortex-M3) y NXP LPC1114 (Cortex-M0).



Anexo 6 - JavaScript

JavaScript es un lenguaje de programación que permite realizar actividades complejas en una página web o aplicación. A cada minuto una página web hace más cosas que sólo mostrar información estática ya sea mostrar actualizaciones de contenido en el momento, interactuar con mapas, animaciones gráficas 2D/3D, etc. Es la tercera "capa del pastel" de los estándares en las tecnologías web, dos de las cuales son (HTML y CSS). HTML es un lenguaje de marcado o etiquetado (no es un lenguaje de programación) que usa la estructura para dar un sentido al contenido web, por ejemplo, define párrafos, cabeceras, tablas, imágenes y vídeos en la página. CSS es un lenguaje de reglas en cascada que usamos para aplicar un estilo a nuestro contenido en HTML, por ejemplo, colocando colores de fondo, fuentes y marginando nuestro contenido en múltiples columnas.

Las 3 capas se complementan una con la otra.

El núcleo de JavaScript consiste en características comunes de programación que permite hacer cosas como:

- Almacenar valores útiles dentro de variables.
- Operaciones escritas en formato de texto (Conocidas como "Strings" en lenguaje de programación).
- Hacer funcionar el código en respuesta a algunos eventos que están ocurriendo.
- Sin embargo, la funcionalidad construida por encima del núcleo del lenguaje de JavaScript es lo más interesante. Llamada Interfaz de Programación en Aplicaciones (Application Programming Interfaces - API's) otorgando muchas funciones extra a nuestro código JavaScript.

Los APIS son inserciones de líneas, incluso bloques gigantes de código listos para usar que permiten a un desarrollador implementarlo en programas que de cualquier otra forma podría ser difícil o incluso imposible de terminar. Así como las herramientas para construir una casa, es lo mismo para las cosas de programación Es mucho más fácil tomar los paneles que ya estén cortados y atornillarlos para hacer una biblioteca, ya que es más extenso y trabajoso diseñarlo por uno mismo, ir y encontrar la madera correcta, cortarla del tamaño correcto y lijlarla, buscar los tornillos del tamaño correcto y ensamblarla para hacer la biblioteca.

Anexo 7 - Control de potencia motor CC – PWM

Para la generación de la señal PWM (Pulse Width Modulated en inglés) es necesario conocer el periodo de modulación constante y el ancho del pulso. El ancho del pulso se calcula a partir de la señal generada por el controlador, de tal manera que la entrada al motor o señal PWM satisfaga ciertas condiciones para que este pueda ser excitado.

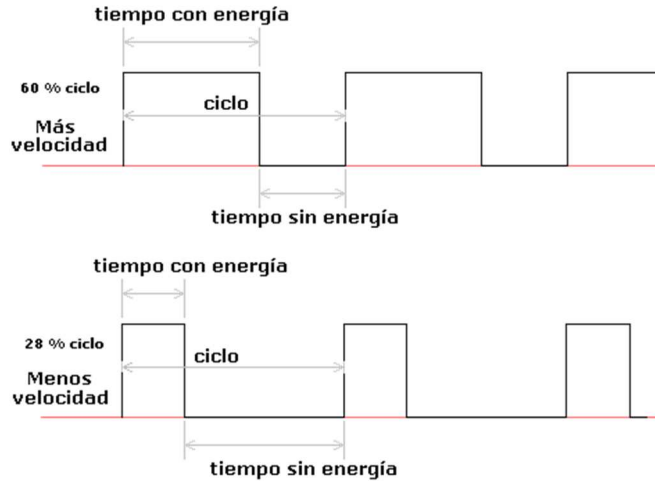
El ciclo de trabajo de una señal periódica es el ancho relativo de su parte positiva en relación con el período.

Expresado matemáticamente:

$$D = \frac{\tau}{T}$$

Siendo:

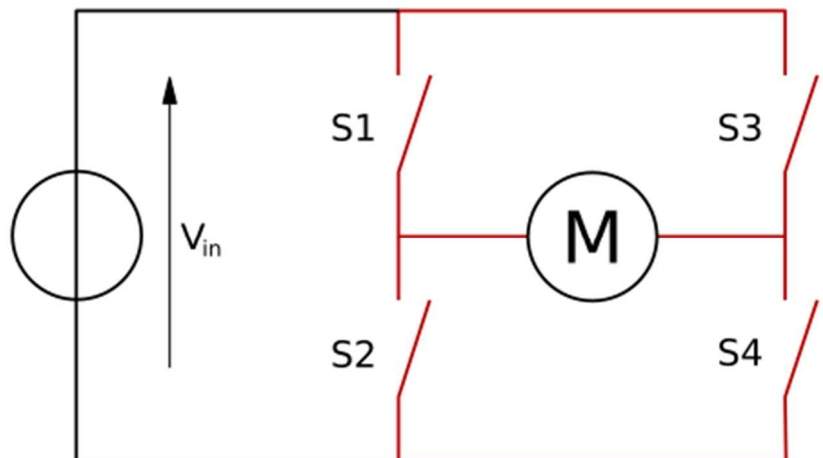
- D: el ciclo de trabajo (relación entre el tiempo con energía y la duración del ciclo).
- tau: el tiempo en que la función es positiva (ancho del pulso o tiempo con energía).
- T: el período de la función (tiempo con energía más tiempo sin energía).



La regulación por ancho de pulso o PWM de un motor de CC está basada en el hecho de que, si se recorta la CC de alimentación en forma de una onda cuadrada, la energía que recibe el motor disminuirá de manera proporcional a la relación entre la parte alta (con suministro de corriente) y baja (sin suministro o cero corriente) del ciclo de la onda cuadrada. Controlando esta relación se logra variar la velocidad del motor de una manera bastante aceptable.

Anexo 8 - Puente H

Un Puente en H es un circuito electrónico que generalmente se usa para permitir a un motor eléctrico DC girar en ambos sentidos. Son ampliamente usados en robótica y como convertidores de potencia. Los puentes H están disponibles como circuitos integrados, pero también pueden construirse a partir de componentes discretos.



El término "puente H" proviene de la típica representación gráfica del circuito. Un puente H se construye con 4 interruptores (mecánicos o mediante transistores). Cuando los interruptores S1 y S4 (ver primera figura) están cerrados (y S2 y S3 abiertos) se aplica una tensión positiva en el motor, haciéndolo girar en un sentido. Abriendo los interruptores S1 y S4 (y cerrando S2 y S3), la tensión se invierte, permitiendo el giro en sentido inverso del motor.

Con la nomenclatura que estamos usando, los interruptores S1 y S2 nunca podrán estar cerrados al mismo tiempo, porque esto cortocircuitaría la fuente de tensión. Lo mismo sucede con S3 y S4.

S1	S2	S3	S4	Función
ON	OFF	OFF	ON	Giro hacia adelante
OFF	ON	ON	OFF	Giro hacia atrás
OFF	OFF	OFF	OFF	Movimiento según inercia
ON	OFF	ON	OFF	Freno
OFF	ON	OFF	ON	Freno
ON	ON	OFF	OFF	Corto Circuito
OFF	OFF	ON	ON	Corto Circuito
ON	ON	ON	ON	Corto Circuito

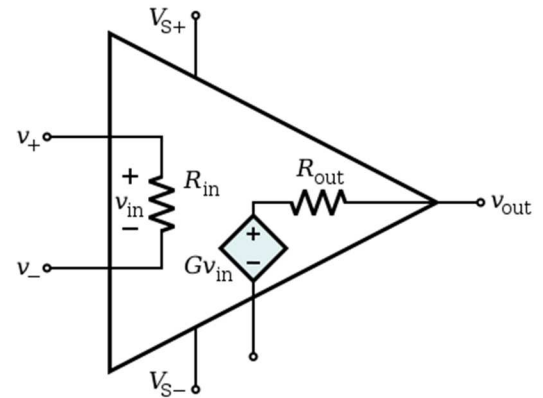
El puente H se usa para invertir el giro de un motor, pero también puede usarse para detener su rotación (de manera brusca), al hacer un corto circuito entre los bornes del motor, o incluso puede usarse para permitir que el motor frene bajo su propia inercia, cuando desconectamos el motor de la fuente que lo alimenta. En el siguiente cuadro se resumen las diferentes acciones.

Lo más habitual en este tipo de circuitos es emplear interruptores de estado sólido por su tiempo de vida prolongado y frecuencias de conmutación altas (del orden de los kHz). En convertidores de potencia es impensable usar interruptores mecánicos, dado su bajo número de conmutaciones dentro de su vida útil y las altas frecuencias que se suelen emplear.

Además, los interruptores se acompañan de diodos (conectados a ellos en paralelo) que permitan a las corrientes circular en sentido inverso al previsto cada vez que se conmute la tensión, puesto que el motor está compuesto por bobinados que durante breves períodos se opondrán a que la corriente varíe.

Anexo 9 - Amplificador operacional

Un amplificador operacional (OA por sus siglas en inglés Operational Amplifier) es un dispositivo amplificador electrónico de alta ganancia acoplado en corriente continua que tiene dos entradas y una salida. En esta configuración, la salida del dispositivo es, generalmente, de cientos de miles de veces mayor que la diferencia de potencial entre sus entradas.



Parámetros de los amplificadores operacionales:

- Impedancia de entrada: Es la resistencia entre las entradas del amplificador.
- Impedancia de salida: Es la que se observa a la salida del amplificador.
- Ganancia en lazo abierto: Indica la ganancia de tensión en ausencia de realimentación. Se puede expresar en unidades naturales (V/V, V/mV) o logarítmicas (dB). Son valores habituales de 100.000 a 1.000.000 V/V. Algunos fabricantes denominan a este parámetro Large-signal differential voltage amplification (Amplificación de tensión diferencial para gran señal).
- Tensión en modo común: Es el valor promedio de tensión aplicado a ambas entradas del amplificador operacional.
- Tensión de desequilibrio (offset) de entrada: Es la diferencia de tensión, entre las entradas de un amplificador operacional que hace que su salida sea cero voltios.
- Corriente de desequilibrio de entrada: Es la diferencia de corriente entre las dos entradas del amplificador operacional, que hace que su salida tome el valor cero.
- Tensión de entrada diferencial: Es la mayor diferencia de tensión entre las entradas del operacional que mantienen el dispositivo dentro de las especificaciones.
- Corriente de polarización de entrada: Corriente media que circula por las entradas del operacional en ausencia de señal.
- Rapidez de variación de tensión (slew rate, en idioma inglés): Es la máxima variación de la tensión de salida respecto de la variación del tiempo, como respuesta a una entrada del tipo escalón. Se mide en V/ μ s, kV/ μ s o unidades similares. Este parámetro está limitado por la compensación en frecuencia de la mayoría de los amplificadores operacionales.
- Relación de Rechazo en Modo Común (RRMC, o CMRR en sus siglas en inglés): Es la capacidad de un amplificador de rechazar señales en modo común.

Características de un OA ideal:

- Infinita ganancia en lazo abierto.
- Infinita resistencia de entrada.
- Corriente de entrada cero.
- Tensión de desequilibrio de entrada igual a cero.
- Infinito rango de tensión disponible en la salida.

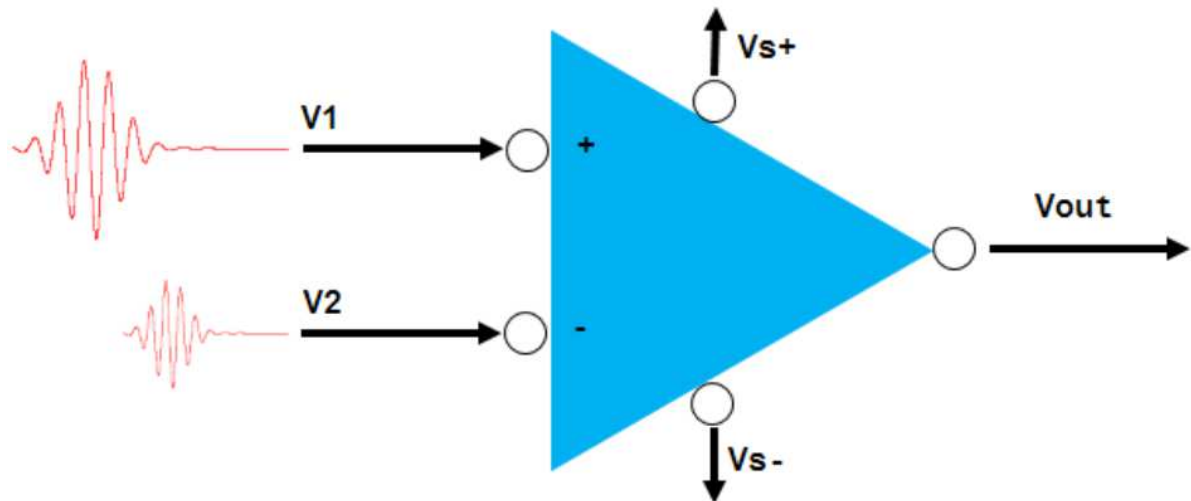
- Infinito ancho de banda con desplazamiento de fase cero.
- Infinita velocidad de variación de tensión
- Resistencia de salida cero.
- Ruido cero.
- Rechazo de modo común infinito (CMRR)
- Factor de rechazo a fuente de alimentación infinito (PSRR).

Características de un OA real:

- Ganancia en lazo abierto, para corriente continua, desde 100.000 hasta más de 1.000.000.
- Resistencia de entrada finita, desde 0,3 MΩ en adelante.
- Resistencia de salida no cero.
- Corriente de entrada no cero, generalmente de 10 nA en circuitos de tecnología bipolar.
- Tensión de desequilibrio de entrada no cero, en ciertos dispositivos es de ± 15 mV
- Rechazo de modo común no infinito, aunque grande, en algunos casos, de 80 a 95 dB.
- Rechazo a fuente de alimentación no infinito.
- Características afectadas por la temperatura de operación.
- Deriva de las características, debido al envejecimiento del dispositivo.
- Ancho de banda finito, limitado a propósito por el diseño o por características de los materiales.
- Presencia de ruido térmico.
- Presencia de efectos capacitivos en la entrada por la cercanía de los terminales entre sí.
- Corriente de salida limitada.
- Potencia disipada limitada.

Podemos utilizar un amplificador operacional para determinar cuál de las dos señales de entrada es mayor. Con que una de las dos señales sea ligeramente superior para que se produzca la salida máxima en el amplificador, sea positiva (+V_{sat}) o negativa (-V_{sat}).

Al utilizar el amplificador operacional en lazo abierto, la ganancia en la salida será siempre muy grande, aproximadamente del orden de 100.000 veces o más. Una pequeña variación en las tensiones de entrada V_{s+} y V_{s-} produce que, a la salida del amplificador, tengamos un valor cercano o igual a la tensión de alimentación.



Si $V1$ es mayor que $V2$, la tensión a la salida del comparador será la alimentación positiva de la fuente $Vs+$.

Si $V2$ es mayor que $V1$, la tensión a la salida del comparador será la alimentación negativa de la fuente $Vs-$.

Un amplificador operacional, trabajando como comparador, puede ser configurado en modo inversor o no inversor, dependiendo a que entrada se aplique la señal a detectar y la señal de referencia.

OA no inversor

En este tipo de comparador, la tensión de referencia es aplicada a la entrada inversora y la señal a detectar se aplica a la entrada no inversora, vale la pena comentar que la tensión de referencia puede ser tanto positiva como negativa.

En el caso de que la señal que tenemos que detectar, sea superior a la señal de referencia, a la salida vamos a tener una señal igual a $Vs+$.

Si la tensión de referencia es mayor a la señal a detectar, en el terminal de salida tendremos como resultado una tensión igual a $Vs-$.

OA inversor

En este tipo de comparador, la tensión de referencia es aplicada a la entrada no inversora y la señal a detectar se aplica a la entrada inversora, al igual que dijimos antes tanto tensión de referencia puede ser tanto positiva como negativa.

En el caso de que la señal que tenemos que detectar, sea superior a la señal de referencia, a la salida vamos a tener una señal igual a $Vs-$.

Si la tensión de referencia es mayor a la señal a detectar, en el terminal de salida tendremos como resultado una tensión igual a $Vs+$.

Referencias bibliográficas

- <https://os.mbed.com/cookbook/PID>
- <https://github.com/ARMmbed/mbed-os/blob/master/features/filesystem/FileSystem.h#L157>
- <https://www.electronicwings.com/mbed/uart-in-arm-mbed-lpc1768>
- <http://playground.arduino.cc/Code/PIDLibrary>
- <https://www.codeproject.com/Articles/43607/Solving-ordinary-differential-equations-in-C>
- <https://www.tinkercad.com>
- <http://www.medtronicdiabeteslatino.com/productos/bombas-de-insulina/que-es-una-bomba-de-insulina>
- <https://www.geekfactory.mx/circuitos-io/sensado-de-corriente-con-op-amp/>
- <http://platea.pntic.mec.es/~lmarti2/circuitos/celsolar/diodorecircul.htm>
- Modeling and Simulation of Glucose-Insulin Metabolism (Kongens Lyngby 2007 IMM-Bachelor-2007)
- User Guide 508 insuline pump (Medtronic MiniMed)
- Motor DC, etapa de potencia y PWM (Felix Monasterio – Huelin – 8 de febrero de 2016)
- Towards an autonomous fully-implantable artificial pancreas (V. Iacovacci, I. Tamadon, C. Perri, P. Dario, L. Ricotti, and A. Menciassi – GNB2018)
- Regulación automática de Glucosa en pacientes diabéticos insulino-dependientes (Guillermo R. Cocha, Melisa Podestá, Carlos E. D'Attellis – RPIC2009)
- The performance of nonlinear blood glucose controllers for type 1 diabetes improves by introducing an oscillatory behaviour in the model (Guillermo R. Cocha, Omar Rodriguez, Dr. Carlos Amorena, Dr. Carlos D'Attellis - BSPC-D-17-00864)
- Análisis en el dominio del tiempo de sistemas de primer y segundo orden - Derecho de Autor © 2008 Carlos Platero Dueñas