

UNIVERSIDAD DEL VALLE DE GUATEMALA  
Facultad de Ingeniería



**Desarrollo de actuadores altamente dinámicos con  
aplicaciones en robótica bio-inspirada**

Trabajo de graduación presentado por Juan Diego Castillo Amaya para  
optar al grado académico de Licenciado en Ingeniería Mecatrónica

Guatemala,

2022







UNIVERSIDAD DEL VALLE DE GUATEMALA  
Facultad de Ingeniería



**Desarrollo de actuadores altamente dinámicos con  
aplicaciones en robótica bio-inspirada**

Trabajo de graduación presentado por Juan Diego Castillo Amaya para  
optar al grado académico de Licenciado en Ingeniería Mecatrónica

Guatemala,

2022




Vo.Bo.:

(f)   
M.Sc. Miguel Enrique Zea Arenales

Tribunal Examinador:

(f)   
M.Sc. Miguel Enrique Zea Arenales

(f)   
Dr. Luis Alberto Rivera Estrada

(f)   
M.Sc. Pedro Iván Castillo Rivera

Fecha de aprobación: Guatemala, 6 de enero de 2022.





La realización de esta investigación fue gracias al apoyo brindado por el Ing. Miguel Enrique Zea Arenales, por haber aceptado ser mi asesor y por toda su ayuda brindada a lo largo del desarrollo de la misma. También, por sus enseñanzas en el campo de sistemas de control.

Agradezco a la Universidad del Valle de Guatemala, por la educación recibida, la cual me otorgó el suficiente conocimiento para poder realizar este proyecto. También a los maestros que me impartieron clases a lo largo de la carrera

Por último, agradezco a mis padres, quienes me han apoyado desde niño para cumplir mis sueños y metas, brindándome la mejor educación posible y permitirme estudiar ingeniería mecatrónica.



<b>Prefacio</b>	v
<b>Lista de figuras</b>	x
<b>Lista de cuadros</b>	xi
<b>Resumen</b>	xiii
<b>Abstract</b>	xv
<b>1. Introducción</b>	1
<b>2. Antecedentes</b>	3
<b>3. Justificación</b>	5
<b>4. Objetivos</b>	7
4.1. Objetivo general	7
4.2. Objetivos específicos	7
<b>5. Alcance</b>	9
<b>6. Marco teórico</b>	11
6.1. Controlador PID	11
6.2. Control lineal cuadrático integral (LQI)	12
6.3. Tipos de <i>encoders</i>	14
6.4. <i>Encoder</i> magnético	15
6.5. Cálculo de posición angular empleando un <i>encoder</i> magnético	16
6.6. <i>Field oriented control (FOC)</i>	17
6.7. Modelado de motor DC	19
6.8. Control óptimo	23
6.9. Filtro pasa bajas IIR	23
6.10. Motor Pololu 20.4:1 Gearmotor 25Dx65Lmm 12V	24
6.11. 48 CPR <i>Encoder</i>	25

6.12. VNH5019 <i>motor driver</i> . . . . .	26
6.13. Resistencia Shunt . . . . .	27
6.14. Fabricación de placas PCB . . . . .	28
<b>7. Control tradicional PID</b>	<b>29</b>
7.1. Metodología . . . . .	29
7.1.1. Configuraciones de la <i>Tiva C</i> . . . . .	30
7.1.2. Procesamiento de datos del microcontrolador . . . . .	32
7.1.3. PID velocidad . . . . .	36
7.1.4. PID posición . . . . .	36
7.1.5. Comparación de controladores con <i>Matlab</i> . . . . .	36
7.1.6. Comparación PID velocidad . . . . .	37
7.1.7. Comparación PID posición . . . . .	38
7.1.8. PID de posición y velocidad en cascada . . . . .	38
7.1.9. Optimización . . . . .	38
7.1.10. Control LQI implementado en microcontrolador . . . . .	40
7.1.11. Fabricación de placa PCB y Carcasa para el actuador . . . . .	40
7.2. Resultados . . . . .	41
7.2.1. Filtro pasa bajas IIR de primer orden para señal de entrada del po- tenciómetro . . . . .	41
7.2.2. PID velocidad . . . . .	41
7.2.3. PID posición . . . . .	43
7.2.4. Comparación PID velocidad . . . . .	44
7.2.5. Comparación PID Posición . . . . .	44
7.2.6. PID de posición y velocidad en cascada . . . . .	45
7.2.7. Controlador PID soportando torque . . . . .	46
7.2.8. Primer planteamiento de restricciones adicionales . . . . .	47
7.2.9. Segundo planteamiento de restricciones adicionales . . . . .	48
7.2.10. Filtro IIR de primer orden de corriente . . . . .	50
7.2.11. Controlador LQI . . . . .	50
7.2.12. Fabricación de placa PCB y carcasa para el actuador . . . . .	53
<b>8. Conclusiones</b>	<b>57</b>
<b>9. Recomendaciones</b>	<b>61</b>
9.0.1. Recomendaciones para la realización de los controladores . . . . .	61
9.0.2. Recomendaciones para la parte física del sistema . . . . .	61
<b>10. Bibliografía</b>	<b>63</b>
<b>11. Anexos</b>	<b>67</b>
11.1. Control tradicional PID . . . . .	67
11.2. Control moderno LQI . . . . .	67
11.3. Esquemáticos y carcasa del actuador . . . . .	67
11.3.1. Esquemático circuito . . . . .	68
11.3.2. Planos carcasa actuador . . . . .	69
<b>12. Glosario</b>	<b>71</b>

---

## Lista de figuras

---

1. Diagrama de bloques controlador PID.	11
2. Diagrama de bloques controlador LQI.	13
3. Disco adentro de un <i>encoder</i> incremental [11].	14
4. Señal que emite un <i>encoder</i> [11].	14
5. Disco adentro de <i>encoder</i> de valor absoluto [12].	15
6. Representación de un <i>encoder</i> magnético [14].	16
7. Ciclos cuadrados según dirección de un <i>encoder</i> [16].	17
8. Diagrama de coordenadas de un rotor [17].	17
9. Diagrama de bloques controlador FOC [5].	18
10. Diagrama para modelado de motor DC con escobillas.	20
11. Diagrama de bode de un filtro pasa bajas [21].	24
12. Motor Pololu 20.4:1 Gearmotor 25Dx65Lmm 12V [22].	25
13. 48 CPR <i>Encoder</i> [22].	25
14. VNH5019 <i>motor driver</i> [23].	26
15. Esquemático de VNH5019 <i>motor driver</i> [23].	27
16. Diagrama de circuito con resistencia shunt.	27
17. Diagrama de bloques conexiones.	29
18. Diagrama de bloques procesamiento <i>Tiva C</i> .	32
19. Gráfica de recta potenciómetro vs posición.	33
20. Gráfica de recta potenciómetro vs velocidad.	34
21. Respuesta al impulso del motor.	37
22. Diagrama de bloques <i>Simulink</i> .	37
23. Diagrama de bloques PID en cascada.	38
24. Señal de potenciómetro vs salida del filtro EMA.	41
25. Resultado PID posición.	42
26. Resultado PID posición.	42
27. Resultado PID posición <i>overshoot</i> .	43
28. Resultado PID posición.	43
29. Resultado PID simulado y PID experimental.	44
30. Resultado PID simulado y PID experimental.	45
31. Resultado de la posición PID en cascada.	46
32. Resultado de la posición PID en cascada.	46

33. Posición del motor con controlador PID ejerciendo torque.	47
34. Corriente medida prueba controlador PID ejerciendo torque.	47
35. Respuesta controlador LQI con restricción del polinomio característico.	48
36. Respuesta controlador LQI con restricción de desigualdad.	49
37. Efecto del filtro IIR de primer orden sobre la corriente.	50
38. Posición del motor con controlador LQI ejerciendo torque.	51
39. Corriente medida prueba controlador LQI ejerciendo torque.	52
40. Posición del motor con controlador LQI sin ejercer torque.	53
41. Corriente medida prueba controlador LQI sin ejercer torque.	53
42. 2D placa PCB controlador.	54
43. 3D placa PCB controlador.	54
44. Propuesta carcasa actuador.	55
45. Esquemático circuito controlador.	68
46. Planos 1 de carcasa para actuador.	69
47. Planos 2 de carcasa para actuador.	70

---

Lista de cuadros

---

1. Efecto de las constantes en un controlador PID. . . . . 12





El presente trabajo tiene como objetivo la creación de actuadores altamente dinámicos para ser utilizados en robótica bio-inspirada por la Universidad del Valle de Guatemala. Se programaron controladores PID para controlar la posición y velocidad de un motor DC con escobillas. Se realizaron los controladores inicialmente por separado, para luego unirlos en cascada y realizar un control más completo del motor DC.

Se planteó un problema de optimización por mínimos cuadrados. Se plantearon dos restricciones adicionales, las cuales dieron paso a la validación de la creación de un controlador LQI para el control de la posición, velocidad y torque de un motor DC con escobillas. Por último, se implementó un controlador LQI para el control del motor y se comparó con el controlador PID realizado.

Una de las complicaciones que se llegó a tener durante la elaboración de este proyecto, fue la correcta selección de las constantes del controlador PID. Estas suelen ser distintas para cada actuador, debido a las características internas de estos. Esta se solucionó colocando un periodo de muestreo y realizando ajustes menores a las constantes hasta llegar al controlador adecuado. En el caso del controlador LQI, la mayor complicación que se tuvo durante la creación de este fue el planteamiento de las restricciones que validasen que el control pudiese ser implementable.



The present work aims to create highly dynamic actuators to be used in bio-inspired robotics by the Universidad del Valle de Guatemala. PID controllers were programmed to control the position and speed of a brushed DC motor. The controllers were initially made separately, to later join them in cascade and perform a more complete control of the DC motor.

A least squares optimization problem was posed. Two additional restrictions were raised, which led to the validation of the creation of an LQI controller for the control of the position, speed, and torque of a brushed DC motor. Finally, an LQI controller was implemented to control the motor, and it was compared with the PID controller made.

One of the complications that arose during the development of this project was the correct selection of the PID controller constants. These are usually different for each actuator, due to their internal characteristics. This was solved by setting a sampling period and making minor adjustments to the constants until reaching the appropriate controller. In the case of the LQI controller, the biggest complication that arose during its creation was the approach to the constraints that would validate that the control could be implementable.



Hoy en día, la robótica posee un papel muy importante en el desarrollo continuo de tecnología, tanto para hogares como para tecnología en la industria. Cuando se habla de robótica se piensa en los robots que se observan en videos o en las caricaturas. Todas esas ideas están en lo correcto, pero entrando en más detalle en los componentes más utilizados son los actuadores, estos siendo motores. Estos como se observan en los robots creados hoy en día, deben ser rápidos y precisos para cumplir con su funcionalidad adecuadamente. Sin embargo, conseguir uno de estos actuadores en el mercado hoy en día suele ser bastante costoso económicamente; esta es la causa del objetivo principal de investigación y creación de un actuador altamente dinámico que otorgue alta precisión de posición, velocidad y control del torque que utiliza.

Cuando se habla del control de estos actuadores, por lo general se habla del controlador tradicional PID. Este controlador utiliza como señales de entrada la referencia del sistema y el valor retornado por el sensor. Estas señales de entrada, al realizar una diferencia entre ellas, se pueden crear un controlador que le permita al sistema llegar a un estado estable.

Los controladores PID pueden no depender de las tres constantes al mismo tiempo, lo cual da paso a los distintos nombres que este puede llegar a tomar, como lo son controladores P, PI, PD, etc. A lo largo de este documento se detalla qué controladores se crearon para controlar cada una de las salidas del motor que se desea controlar. También se detalla el proceso de ajuste de las constantes realizado para cada uno de los controladores y cómo estos se compararon con datos obtenidos de simulaciones realizadas en [Matlab](#).

Los controladores PID, a pesar de estos ser control tradicional, cumplen la mayoría de veces con el funcionamiento deseado, pero hoy en día existen técnicas de control moderno que pudiesen mejorar el control de los sistemas. Se planteó la propuesta de la creación de uno de estos controladores para determinar si es mejor utilizar control tradicional o moderno en el caso de actuadores altamente dinámicos.

Los actuadores altamente dinámicos, a diferencia de los actuadores controlados por un controlador PID, poseen la gran ventaja de que otorgan un torque. En aplicaciones dinámicas

es necesaria la presencia del torque, ya que estos actuadores no solo realizan un control de posición sino que también deben soportar las cargas que se ejercen sobre estos actuadores. Un claro ejemplo de estos actuadores se encuentra en el robot *Atlas* de *Boston Dynamics*. En este caso, el torque es requerido para que el robot pueda ser capaz de imitar los movimientos humanos e incluso en ocasiones soportar su peso sobre una pierna.

Durante este trabajo se muestra un planteamiento de optimización, la cual validó el uso de un controlador LQI para realizar el controlador dinámico. Se muestran las pruebas que se realizaron y se describen las dificultades que se tuvieron al momento de realizar el controlador. Por último, se presenta una propuesta de manufactura para realizar un contenedor para este controlador y poder utilizarlo en distintas aplicaciones en un futuro.

La robótica es la rama de la ingeniería que integra distintas disciplinas tales como: ingeniería mecánica, ingeniería electrónica, ingeniería biomédica e ingeniería en ciencias de computación. Esta es encargada del diseño, análisis y construcción de los robots en el mundo de la industria y desarrollo.

Uno de los principales componentes que se utilizan en la robótica son los **actuadores**, los cuales son componentes que pueden convertir energía de entrada en energía mecánica de salida. Por lo general, los más usados para esto son los motores eléctricos; el fin de estos motores es poder generar un movimiento deseado a una velocidad específica, cierto torque para lograr su objetivo y poder llegar a una posición final específica.

Existen distintos motores con características específicas adecuadas para distintas situaciones. Los más comunes de hoy en día son: motores DC con escobillas, servomotores y motores *stepper*. A pesar de que estos son los motores utilizados con mayor frecuencia, actualmente se está implementando un "nuevo" tipo de motor, el cual es el motor DC sin Escobillas.

Como se explica en [1], [2] y [3], estos motores poseen una eficiencia mayor ya que no poseen pérdidas por fricción a causa de las escobillas. Estos motores pueden verse frecuentemente en drones de carrera, ya que estos son caracterizados por sus capacidades de otorgar grandes cantidades de velocidad en revoluciones por minuto (rpm). Pero estos motores poseen otras ventajas, ya que también otorgan grandes cantidades de torque. El torque en este caso dependerá de la cantidad de polos que este tenga: a mayor cantidad de polos, el torque otorgado será mayor, pero se tendrá una menor velocidad máxima y viceversa.

Al utilizar alguno de los distintos tipos de motores en la robótica, estos deben cumplir funciones determinadas dependiendo del uso que se les de. Para esto, se deben utilizar **drivers** que permitan el control de lo mismo. En [4] y [5], se describen distintos *drivers* dependiendo de lo que se desee realizar, además, muestran como crear uno de estos *drivers*.

En la actualidad, los motores DC sin escobillas se están utilizando cada vez más por las

ventajas respecto a otros motores anteriormente mencionados. Un ejemplo donde se pueden observar estos actuadores es en los robots creados por *Boston Dynamics*. Uno de estos robots es el *Spot*, el cual utiliza estos motores para lograr el movimiento de sus patas. Este robot tiene una gran precisión de posición, con velocidad y torque por parte de los motores gracias a los drivers que utilizan.

## Motores DC si escobillas (*brushless*)

En [1] se explica cómo se crea un motor DC sin escobillas y el porqué se desean utilizar estos. En este caso, el autor buscaba la creación de un motor mucho más eficiente y más liviano para ser utilizados en el *Cheetah MIT*. Esto con la idea de no solo ahorrar el consumo de energía, sino que también estos realizaran un movimiento más rápido, preciso y con mayor torque al que podían otorgar los motores utilizados anteriormente. En este trabajo se mencionan que la masa del motor se pudo reducir en un 23% a la inicial, el aumento del torque fue de un factor mayor de 1.6 y se logró disipar las pérdidas de energía en un 16%, lo que permitió una mayor eficiencia.

## Mjbots

En [6] se puede encontrar un proveedor que brinda servos realizados con motores DC sin escobillas. Estos describen cómo fue el proceso de la creación y fabricación no solo de los motores sino que también del controlador que se utiliza para estos, en este caso siendo el *Field Oriented Control (FOC)*. Como ejemplo, esta empresa creó un robot cuadrúpedo que permite observar el funcionamiento de estos actuadores y el buen desempeño que tienen.



En el campo de la robótica es constante la búsqueda de la optimización de los componentes utilizados. Destaca entre estos el optimizar el funcionamiento de los actuadores, los cuales tienen como función principal la conversión de un tipo de energía de entrada en otro tipo de energía de salida. En el caso de la robótica, los motores son los que permiten convertir la energía eléctrica en energía rotacional mecánica.

El presente proyecto busca la creación de sistemas de control que permitan controlar diversas características, tales como el torque otorgado, velocidad y posición de los motores con aplicaciones en robótica bio-inspirada. Esto con el objetivo de poder utilizar estos motores altamente dinámicos en la Universidad del Valle de Guatemala para futuros proyectos en el laboratorio de robótica y para proyectos dedicados al área de robótica bio-inspirada.

Como se indicó en los antecedentes, estos actuadores se fabrican hoy en día con motores DC *Brushless*. Se busca desarrollar estos mismos actuadores altamente dinámicos, pero con motores DC *Brushed*. Esto debido a que los motores DC *Brushless* no poseen un proveedor local dedicado a estos con aplicaciones distintas a drones, por lo que se deben manufacturar según la aplicación, lo que provoca que el costo de estos sea elevado y deban ser importados a Guatemala. Por otro lado, existen *Drivers* que permiten el control de estos motores, refiriéndose a los brushles, pero su costo también es elevado, por lo que se busca desarrollar estos a un menor costo.



### 4.1. Objetivo general

Diseñar e implementar un actuador rotacional controlado por torque que permita la creación de plataformas robóticas, mejorando la velocidad, posición y torque del mismo.

### 4.2. Objetivos específicos

- Diseñar y manufacturar un contenedor capaz de almacenar un actuador rotacional, controlador diseñado, protocolo de comunicación y conjunto de engranes para reducción de potencia adecuada a aplicaciones de alta velocidad, posición precisa y alto torque.
- Seleccionar e implementar diversos sensores necesarios para estimar las variables de estado de interés que permitan el control adecuado del actuador
- Diseñar, programar y manufacturar una placa electrónica que permita el control de torque, velocidad y posición del actuador, junto con protocolo de comunicación digital capaz de configurar y controlar el actuador.



Durante la realización de este proyecto se plantearon diversas metas para ir alcanzándolas una por una y que no se perdiera una línea de trabajo ordenada. Lo primero fue la realización de un controlador PID para control de velocidad. Esta fue la primera meta alcanzada exitosamente y se comparó con la respuesta obtenida de *Matlab*. La segunda meta que se alcanzó exitosamente fue la realización del controlador PID para posición. Este al igual que el segundo caso se logró alcanzar la meta y compararlo con la respuesta retornada por *Matlab*. La tercera meta fue la realización del controlador PID en cascada para controlar la posición y velocidad del sistema. Esta meta se alcanzó por completo terminando así la fase de control tradicional PID.

Luego se plantearon metas para la realización de un controlador moderno LQI. Lo primero fue el planteamiento de una optimización por mínimos cuadrados. En este caso la optimización faltó restringirla de mejor manera para poder utilizar el controlador LQI que se obtuviese con la ayuda de *Matlab*. Pero en este caso la optimización permitió validar el uso de un controlador LQI para aplicaciones dinámicas. A pesar de que no se alcanzó la meta de utilizar por completo la optimización, esta permitió poder plantear un controlador LQI en el microcontrolador. Por último se alcanzó la meta del desarrollo de un actuador altamente dinámico.

Por último se realizó una propuesta de manufactura. Para esta parte del desarrollo, se tuvo limitantes para la manufacturación del sistema embebido. Estas fueron la escasez de componentes electrónicos utilizados debido a la pandemia, el no poder utilizar un material accesible para la carcasa, ya que al soportar torque no puede ser impreso en 3D e incluso solo utilizar aluminio podría llegar a fallar y por último la placa PCB realizada consta de 4 capas. En este caso esta fue una limitación muy grande ya que solo se tiene acceso rápido a placas de dos capas y esta se tendría que mandar a manufacturar.

Este proyecto aun se puede mejorar en el planteamiento de las restricciones para la optimización, el replanteamiento del controlador LQI, mejorar los diseños de manufactura ya que existe oportunidad de realizar un sistema más compacto si se desea, ajustar las constantes del controlador PID y se podrían realizar pruebas más dirigidas a sistemas dinámicos.



## 6.1. Controlador PID

Uno de los controladores más comunes es el controlador PID, el cual consta de tres partes: una parte proporcional, una parte integradora y una parte derivativa. Juntas, estas permiten realizar control de sistemas SISO (sistemas con una sola entrada y salida), como se indica en [7] y [8]. En la Figura 1 se puede observar un diagrama de bloques que describe la forma de un controlador PID. Este consta de la entrada, la cual entra a un sumador en el cual se le resta la salida, y este resultado va hacia el bloque del controlador PID que va conectado en serie con la planta. La planta está representada por el bloque  $G(s)$ , el cual es el sistema que se desea controlar, que en este caso corresponde al motor DC con escobillas.

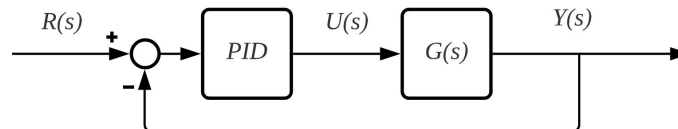


Figura 1: Diagrama de bloques controlador PID.

Como ya se mencionó el controlador PID está conformado por tres partes distintas:

- Proporcional:** Esta parte del controlador es el producto del error con una constante de proporcionalidad. Puede provocar sobre-oscilaciones, y mientras mayor sea la constante de proporcionalidad esta aumentará las oscilaciones lo que provocará que el controlador se vuelva inestable. La ecuación (1) describe la parte proporcional del controlador, donde  $K_p$  corresponde a la constante de proporcionalidad y  $e(t)$  corresponde a la señal de error.

$$C_p(t) = K_p e(t). \quad (1)$$

- Integral:** Esta parte del controlador busca la reducción y eliminación del error en

estado estacionario. Esta parte suele ser lenta ya que, como su nombre lo indica, integra el error respecto al tiempo, el cual luego se multiplica por una constante  $K_i$ . La ecuación (2) describe la parte integradora del controlador, donde  $K_i$  es la constante integradora y  $e(\tau)$  es el error en estado estable de la entrada.

$$C_i(t) = K_i \int e(\tau) d\tau. \quad (2)$$

- **Derivativa:** Esta parte del controlador busca la eliminación del error cuando existe un cambio abrupto en la señal de salida. La parte derivativa del control solo actúa si suceden estos cambios abruptos, por lo que la parte proporcional e integral son las únicas que actúan si no existe este cambio. Esta parte deriva el error con respecto al tiempo y luego la multiplica por una constante  $K_d$ . La ecuación (3) describe la parte derivadora del controlador donde  $K_d$  es la constante derivadora y  $e$  es el error.

$$C_d(t) = K_d \frac{de}{dt}. \quad (3)$$

Al sumar cada una de las partes del controlador se obtiene la ecuación (4).

$$u(t) = K_p e(t) + K_i \int e(\tau) d\tau + K_d \frac{de}{dt}. \quad (4)$$

La ecuación (5) es la ecuación del controlador PID en el dominio de la frecuencia.

$$U(s) = K_p + \frac{K_i}{s} + K_d s. \quad (5)$$

En el Cuadro 1, se pueden observar los distintos comportamientos y cómo afecta el controlador a la planta si se aumentan los valores de las variables del PID. Estos cambios no son proporcionales uno con el otro, no por aumentar dos valores la misma cantidad estos provocarán el mismo cambio en la señal.

Param.	Tiempo de subida	Sobre-elevación y oscilaciones	Tiempo de asentamiento	Error en estado estable	Estabilidad
$K_p$	Decrece	Crece	Cambio pequeño	Decrece	Degrada
$K_i$	Decrece	Crece	Crece	Elimina	Degrada
$K_d$	Cambio menor	Decrece	Decrece	No tiene efecto	Mejora si $K_d$ es pequeño

Cuadro 1: Efecto de las constantes en un controlador PID.

## 6.2. Control lineal cuadrático integral (LQI)

En (9) y (10) se detalla el funcionamiento del controlador Lineal Cuadrático Integral (LQI). Este busca la estabilización de la planta de un sistema mediante las matrices de



variables de estado aumentadas. Esto se debe a que este controlador agrega una parte integral al momento de realizar retroalimentación. Como se observa en la Figura 2, este controlador realiza una retroalimentación externa para estabilizar la retroalimentación interna.

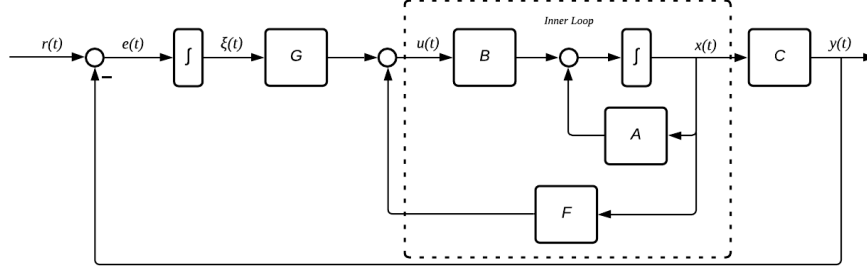


Figura 2: Diagrama de bloques controlador LQI.

Para este controlador se considera la planta LTI en variables de estado, en donde:

- $x$  son las variables de estado respecto al tiempo,  $x \in \mathbb{R}$ .
- $u$  es la entrada a la planta,  $u \in \mathbb{R}$ .
- $y$  es la salida del sistema,  $y \in \mathbb{R}$ .
- $A \in \mathbb{R}^{n \times n}$ ,  $B \in \mathbb{R}^{n \times m}$  y  $C \in \mathbb{R}^{p \times n}$  son matrices estrictamente propias con constantes del sistema.

$$\begin{aligned}\dot{x} &= Ax + Bu, \\ y &= Cx + Du.\end{aligned}$$

En un controlador LQI la salida de la planta LTI debe pasar multiplicando una matriz  $C_r$ . Esta matriz se encarga de definir la salida de control y seleccionar que variable de estado para la cual se desea realizar el control. La salida del sistema está descrita en la ecuación (6).

$$y_r = C_r x, \quad (6)$$

La ecuación (7) describe el error. Al derivarlo, se puede obtener la ecuación (8), siendo esta la nueva retroalimentación.

$$\sigma = \int_{t_0}^t (y_r - r) dt, \quad (7)$$

$$\dot{\sigma} = y_r - r. \quad (8)$$

### 6.3. Tipos de *encoders*

En [11] y [12] se puede encontrar información acerca de los *encoders* que existen. Estos dispositivos están destinados a la detección del movimiento rotativo. Por lo general, estos son utilizados en los rotores de los motores, ya que permiten la medición de la posición y la velocidad de rotación de estos. Existen dos tipos de *encoder*:

- **Encoder incremental:** Este consta de un foto-transistor, el cual genera un haz de luz que es recibido por un receptor el cual detecta interrupciones en un disco con ranuras pequeñas alrededor de su diámetro. El funcionamiento de este *encoder* es digital, ya que genera una señal cuadrada representando un 1 o un 0. Esto quiere decir que, al momento de girar el disco, la señal del foto-transistor se verá interrumpida por un momento ya que el disco no permitirá que la señal se reciba hasta que aparezca una ranura en el disco nuevamente. En la Figura 3, se puede observar cómo es el disco dentro de este *encoder* y en la Figura 4 se puede observar un ejemplo de la señal que devuelve.

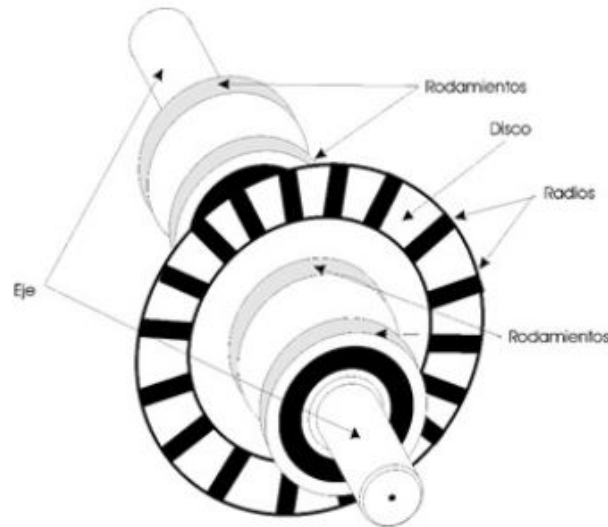


Figura 3: Disco adentro de un *encoder* incremental [11].



Figura 4: Señal que emite un *encoder* [11].

- **Encoder de Valor Absoluto:** Este tipo de *encoder*, a diferencia del incremental, transmite por código binario su información. Ambos poseen foto-transistores, los cuales generan la señal de 1 o 0, pero el de valor absoluto difiere en que, al enviar su

información de posición por código binario, este conserva la posición todo el tiempo en la que el rotor del motor esta. El disco de este, al igual que el del *encoder* incremental, posee ranuras que obstruyen la señal de los foto-transistores. En la Figura 5, se puede observar un disco de este *encoder*. Este posee ranuras en distintas posiciones de este, este caso, algunos foto-transistores están obstruidos por el disco y otros no, conservando la posición en código binario.

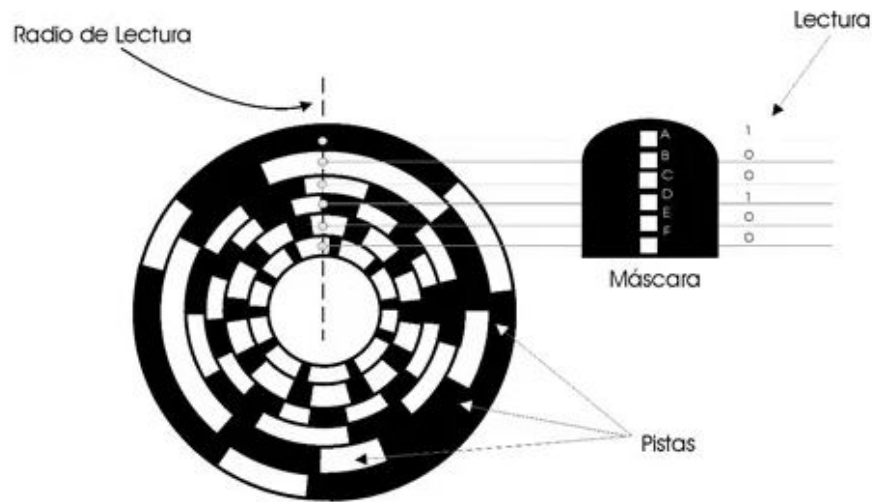


Figura 5: Disco adentro de *encoder* de valor absoluto [12].

#### 6.4. *Encoder* magnético

Dentro de estos *encoders* incrementales se encuentran otros tipos, uno de los cuales son los *encoders* magnéticos. Estos funcionan muy similar a los *encoders* con foto-transistores con la diferencia que estos trabajan con campos magnéticos. Como se indica en [13], estos *encoders* trabajan con efecto Hall. Como se indica en [14], el efecto Hall es el movimiento de cargas de un objeto atraídas por un campo magnético. Uno de los mejores ejemplos para explicar este efecto son los imanes. Los imanes constan de dos polos, una con cargas positivas representan como el polo norte y otra con cargas negativas representada como el polo sur. Los imanes suelen unirse debido a que las cargas negativas de uno se ven atraídas a las cargas positivas del otro debido al efecto Hall. En el caso que las cargas sean iguales estas se repelerán, lo que suele suceder con los imanes si no se colocan adecuadamente.

En la Figura 6, se puede observar el funcionamiento de un *encoder* magnético. Este consta de un disco el cual está separado por segmentos, unos siendo cargas positivas y otros siendo cargas negativas. El movimiento rotatorio de estas separaciones se detecta mediante un sensor de efecto Hall, el cual al detectar una carga positiva generando el flanco de subida y al detectar la carga negativa generando el flanco de bajara. Esto es lo que provoca la creación de la señal cuadrada en el *encoder* magnético. La cantidad de salidas de este tipo de *encoder* dependerá de cuantos sensores de efecto Hall tenga incorporados.

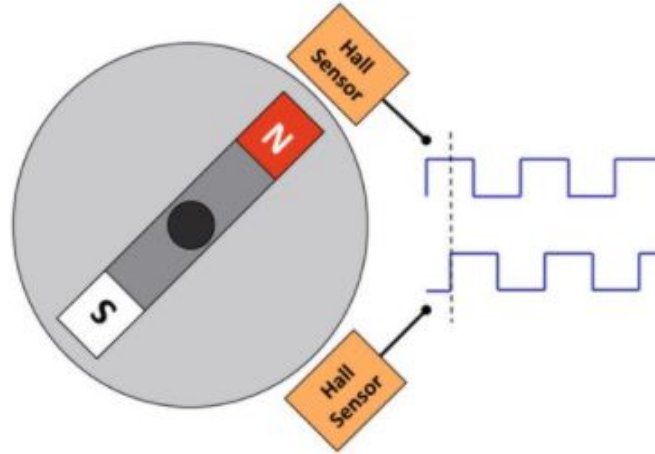


Figura 6: Representación de un *encoder* magnético [14].

## 6.5. Cálculo de posición angular empleando un *encoder* magnético

Como se mencionó anteriormente un *encoder* posee cierta cantidad de divisiones en el disco, las que permiten la generación de las señales cuadradas. En [15] y [16] se muestra cómo utilizando las señales generadas, se puede determinar la posición angular del motor. La ecuación (9) muestra cómo calcular la posición angular en radianes del motor, donde: *No.Pulso* es la cantidad de cambios que se han dado al momento de realizar un movimiento o giro y *N* es la cantidad de divisiones del disco que debe pasar para que el eje del motor realice una revolución completa. En caso que el motor tenga una caja reductora en la salida, el eje del motor no será el punto de interés si no que será la salida de la caja reductora. Para determinar este valor en dado caso se tenga la caja reductora, se debe de multiplicar la reducción que esta hace por la cantidad de divisiones que posee el disco del *encoder* y el número de pulsos contado deberá llegar hasta este valor antes de contar una revolución completa. Si esta se desease en grados, el valor de  $2\pi$  se sustituye por  $360^\circ$ .

$$\theta = 2\pi \frac{No.Pulso}{N}. \quad (9)$$

Sabiendo esto, se puede determinar hacia dónde gira el eje del motor. Para esto se deben de utilizar las dos señales cuadradas, ya que esto permite observar cuál cambio primero, lo que indica en que sentido gira el motor. En la Figura 7 se puede observar en el primer caso si la señal A del *encoder* realiza un cambio positivo primero. Este será el que gobernará, ya que la señal B esta desfasada e inicia después de la señal A. En este caso el motor gira en contra de las agujas del reloj. En el segundo caso la señal B es la que gobierna ya que esta realiza un cambio positivo primero, lo que indica que el motor gira a favor de las agujas del reloj.

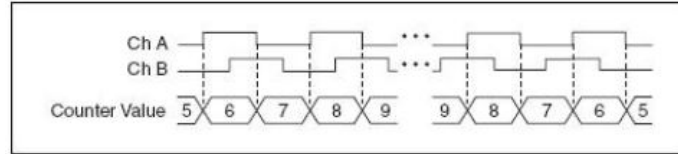


Figura 7: Ciclos cuadrados según dirección de un *encoder* [16].

## 6.6. *Field oriented control (FOC)*

El *Field Oriented Control (FOC)*, como se indica en [5] y [17], es una técnica para controlar el torque y el flujo magnético de motores AC por separado, con el propósito de tener un mejor control sobre el mismo y poder explotar en su totalidad el motor. Esta técnica, por lo general, se utiliza en motores trifásicos. En la Figura [8] se puede observar cómo se representa un diagrama de los ejes  $qd$ , los cuales están alineados con el flujo del rotor, observando cómo afectan las corrientes. En la Figura [9], se puede observar un diagrama de bloques de cómo se realiza este control.

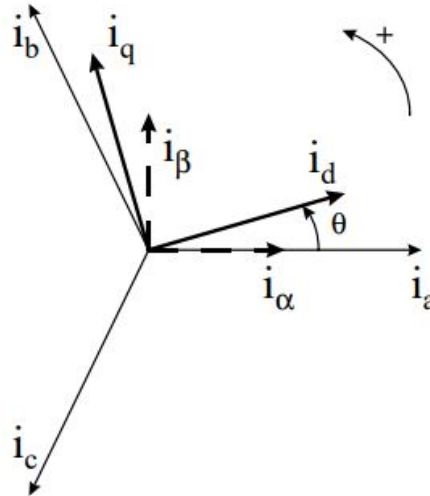


Figura 8: Diagrama de coordenadas de un rotor [17].

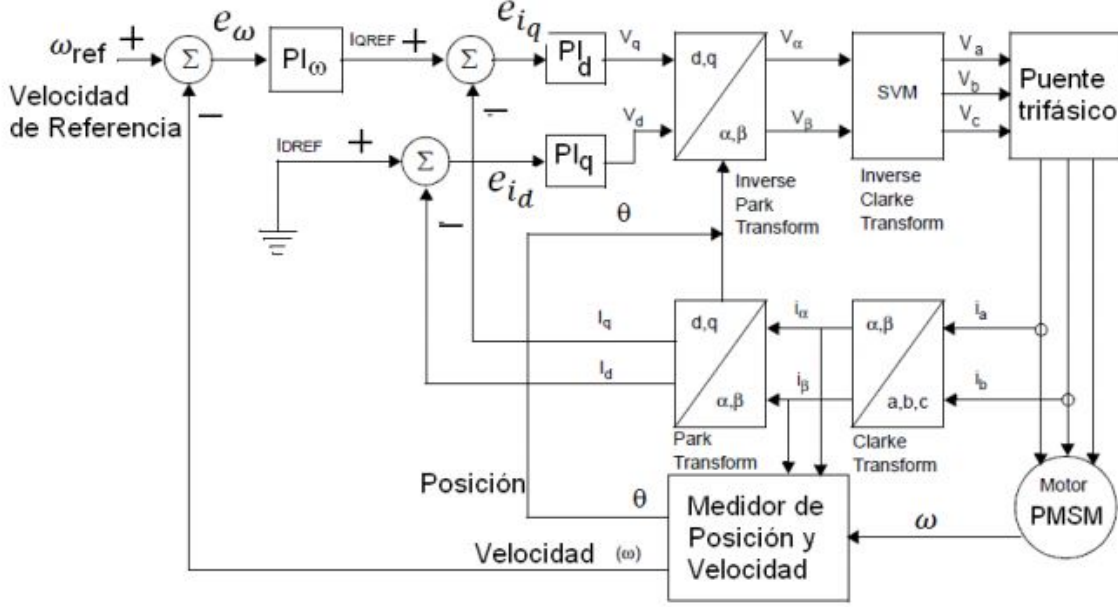


Figura 9: Diagrama de bloques controlador FOC [5].

En este caso se tiene dos controladores PI, los cuales se encargan de controlar el torque y el flujo magnético que se desea suministrar al motor AC. También se cuenta con un control PID el cual se encarga de hallar la constante de  $i_{qref}$  para el controlador del flujo magnético. En este caso, al ser un motor trifásico, se debe aplicar la transformación de *Clarke* y *Parker*

Como se indica en [18], el propósito de realizar esta transformación es convertir un sistema de 3 variables en un sistema de 2 variables, en este caso siendo las 3 corrientes  $i_a$ ,  $i_b$  e  $i_c$ , las cuales corresponden a las corrientes de cada fase. Esta transformación convertirá estas corrientes a  $i_\alpha$ ,  $i_\beta$  e  $i_0$  como se muestra en las ecuaciones (10), (11) y (12).

$$i_\alpha = \frac{1}{2}i_a - \frac{1}{3}(i_b - i_c), \quad (10)$$

$$i_\beta = \frac{2}{\sqrt{3}}(i_b - i_c), \quad (11)$$

$$i_0 = \frac{2}{3}(i_a + i_b + i_c). \quad (12)$$

En la mayoría de casos  $i_0$  tiende a cero, por lo que  $i_\alpha$  se iguala a  $i_a$  como se muestra en las ecuaciones (13), (14) y (15).

$$i_\alpha = i_a, \quad (13)$$

$$i_\beta = \frac{2}{\sqrt{3}}i_a + \frac{2}{\sqrt{3}}i_b, \quad (14)$$

$$i_0 = (i_a + i_b + i_c) = 0. \quad (15)$$

La transformación de *Parker* busca convertir las variables que están respecto a  $\alpha\beta$  al marco de referencia rotatorio  $qd$ , para así poder utilizar estas variables como retroalimentación en el sistema. En las ecuaciones (16) y (17), se muestra la transformación de *Parker*:

$$i_d = i_\alpha \cos \theta + i_\beta \sin \theta. \quad (16)$$

$$i_q = -i_\alpha \sin \theta + i_\beta \cos \theta. \quad (17)$$

Como se muestra en la Figura 9, se debe utilizar un *encoder* para obtener los datos de la posición y la velocidad del rotor de un motor. La posición es útil para poder realizar la transformación de *Parker*. La velocidad se utiliza en retroalimentación con la velocidad deseada ( $\omega_{ref}$ ), la cual luego entra a un controlador PID con el cual se obtiene  $i_{qref}$ , corriente de referencia para controlar la posición y velocidad del motor.

Luego de realizar el control tanto de la posición y torque, los controladores PI devuelven valores de voltajes  $V_q$  y  $V_d$ . A esto se les debe aplicar la transformada inversa de *Parker* y *Clarke* para poder obtener los voltajes  $V_a$ ,  $V_b$  y  $V_c$ . De la ecuación (18) a (22) se muestra la transformada inversa de ambos casos.

- **Parker:**

$$V_\alpha = V_d \cos \theta - V_q \sin \theta, \quad (18)$$

$$V_\beta = V_d \sin \theta + V_q \cos \theta. \quad (19)$$

- **Clarke:**

$$V_a = V_\alpha, \quad (20)$$

$$V_b = -\frac{1}{2}V_\alpha + \frac{\sqrt{3}}{2}V_\beta, \quad (21)$$

$$V_c = -\frac{1}{2}V_\alpha - \frac{\sqrt{3}}{2}V_\beta. \quad (22)$$

A pesar de que el controlador *Field Oriented Control (FOC)* está hecho para controlar motores trifásicos, otorga una muy buena idea de lo que se desea realizar para los actuadores que se desean realizar en el proyecto.

## 6.7. Modelado de motor DC

En la Figura 10 se encuentra el modelado de un motor DC con escobillas. Este es modelado por una parte electrónica, la cual consta de un inductor y una resistencia. Estas representan la inductancia que requiere el motor para su movimiento y la resistencia interna que este posee. Por otro lado, existe la parte mecánica la cual está representada por una masa, un resorte y un amortiguador. Estos representan la inercia del rotor, la elasticidad y qué tan rápido tiende a parar este.

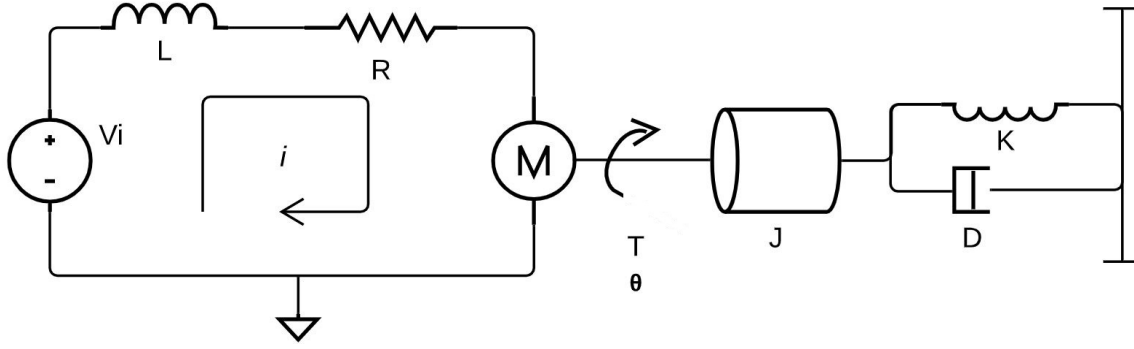


Figura 10: Diagrama para modelado de motor DC con escobillas.

Para obtener las ecuaciones que describen el comportamiento del motor se debe analizar la parte electrónica mediante análisis de voltajes en mallas en circuitos. La ecuación (23) representa el análisis de voltajes en mallas ya realizado, donde se muestran los voltajes que actúan dentro del motor, donde:

- $V_i$  es el voltaje de entrada.
- $V_L$  es el voltaje en el inductor.
- $V_R$  es el voltaje en la resistencia.
- $\varepsilon$  es el factor de *Back EMF*.

$$V_i - V_L - V_R - \varepsilon = 0. \quad (23)$$

La ecuación (23) solo representa el circuito en términos de voltajes. Estos se pueden representar como la resistencia por la corriente que pasa a través de cada uno de los componentes, gracias a la Ley de Ohm ( $V = iR$ ). En el caso del inductor este depende del tiempo, por lo que en este caso no se puede multiplicar el valor de la inductancia solo por la corriente pero si por la derivada de esta. Tomando todo esto en cuenta se obtiene la ecuación (24).

$$V_i - L \frac{di_L}{dt} - Ri_L - \varepsilon = 0. \quad (24)$$

En [19] se detalla qué es la constante *Back EMF* en un motor DC. Esta es variante ya que es la encargada de regular el giro del motor. Por ejemplo, si un motor gira a cierta velocidad y no se le ejerce un torque contrario esta, será casi igual al voltaje de entrada permitiendo así una velocidad rápida de giro con un torque bajo. Pero si el giro del motor se ve afectado esta constante disminuirá para aumentar el torque disminuyendo la velocidad, con el objetivo de permitir al motor regresar a su velocidad de giro inicial. Para este modelado se realizó la siguiente asunción  $\varepsilon = K_\varepsilon \dot{\theta}$ , obteniendo así la ecuación (25).

$$V_i - Li_L - Ri_L - K_\varepsilon \dot{\theta} = 0. \quad (25)$$

A continuación, se encuentra el comportamiento mecánico del motor. Este fue posible encontrarlo gracias a impedancias mecánicas rotativas de cada componente. La ecuación (26) describe la parte mecánica del motor DC. En donde:



- $J$  es la constante de inercia de la masa.
- $b$  es la constante de viscosidad del amortiguador.
- $K$  es la constante elástica del resorte.
- $\dot{\theta}$  es la primera derivada de la posición del rotor respecto al tiempo representando la velocidad de giro del roto.
- $\ddot{\theta}$  es la segunda derivada de la posición del rotor respecto al tiempo representando la aceleración de giro del rotor.
- $\tau$  es el torque del rotor.

$$J\ddot{\theta} + b\dot{\theta} + k\theta = \tau. \quad (26)$$

En el caso de los motores, el torque que suministran se puede controlar mediante la corriente que se inyecta a estos. Sabiendo esto se debe de representar el torque del motor de tal forma que este sea dependiente de la corriente suministrada. Por lo que  $\tau = K_\tau i$ , al sustituir esto en la ecuación (26), se obtiene la ecuación (27).

$$J\ddot{\theta} + b\dot{\theta} + k\theta = k_\tau i. \quad (27)$$

Las ecuaciones (25) y (27) describen por completo a un motor DC con escobillas. Pero si se observa la ecuación (25), esta está representada respecto a la corriente. Para poder manipular ambas ecuaciones simultáneamente y poder obtener las matrices de las variables de estado se debe multiplicar la ecuación (25) por la constante  $K_\tau$ . Esto permitirá que el torque no esté representado por la corriente, si no por la variable de estado correspondiente.

$$\left( \dot{i}_L = -\frac{R}{L}i_L - \frac{K_\varepsilon}{L}\dot{\theta} + \frac{1}{L}V_i \right) K_\tau. \quad (28)$$

La ecuación (29) muestra el resultado de la multiplicación de  $K_\tau$ .

$$\dot{\tau} = -\frac{R}{L}\tau - \frac{K_\varepsilon K_\tau}{L}\dot{\theta} + \frac{K_\tau}{L}V_i. \quad (29)$$

Se despejó de la ecuación (27) para  $\ddot{\theta}$

$$\ddot{\theta} = -\frac{K_\tau}{J}i_L - \frac{K}{J}\theta + \frac{b}{J}\dot{\theta}. \quad (30)$$

En la ecuación (30) se sustituyó  $K_\tau i = \tau$  para obtener así la ecuación (31)

$$\ddot{\theta} = -\frac{1}{J}\tau - \frac{K}{J}\theta + \frac{b}{J}\dot{\theta}. \quad (31)$$

Ahora que las ecuaciones (29) y (31) están en términos de la velocidad angular ( $\dot{\theta}$ ), aceleración angular ( $\ddot{\theta}$ ) y el torque ( $\tau$ ), se pueden definir las variables de estado para el controlador. Estas siendo las indicadas a continuación.

$$\begin{aligned} x_1 &= \tau, \\ x_2 &= \theta, \\ x_3 &= \dot{\theta}, \end{aligned} \quad u = V_i.$$

Una vez obtenidas las variables de estado, estas se deben derivar una vez para poder realizar  $\dot{x} = Ax + Bu$ , lo cual describe la entrada del sistema y su comportamiento.

$$\begin{aligned} \dot{x}_1 &= \dot{\tau}, \\ \dot{x}_2 &= \dot{\theta} = x_3, \\ \dot{x}_3 &= \ddot{\theta}. \end{aligned}$$

Estas variables de estado representan las variables de interés que se desean controlar del motor. Con estas definidas se pueden sustituir en las ecuaciones (29) y (31), obteniendo así las ecuaciones (32), (33) y (34).

$$\dot{x}_1 = -\frac{R}{L}x_1 - \frac{K_\varepsilon K_\tau}{L}x_3 + \frac{K_\tau}{L}u, \quad (32)$$

$$\dot{x}_2 = x_3, \quad (33)$$

$$\dot{x}_3 = -\frac{1}{J}x_1 - \frac{K}{J}x_2 - \frac{b}{J}x_3. \quad (34)$$

Ahora se debe encontrar la matriz A. En esta matriz se colocan las constantes que estén acompañando a  $x_1$ ,  $x_2$  y  $x_3$  (variables de estado). Estas van ordenados por columnas y las filas corresponden a  $\dot{x}_1$ ,  $\dot{x}_2$  y  $\dot{x}_3$ .

$$A = \begin{bmatrix} -\frac{R}{L} & 0 & -\frac{K_\varepsilon K_\tau}{L} \\ 0 & 0 & 1 \\ -\frac{1}{J} & -\frac{K}{J} & -\frac{b}{J} \end{bmatrix}. \quad (35)$$

También se puede encontrar la matriz B, correspondiente a las entradas al motor.

$$B = \begin{bmatrix} \frac{K_\tau}{L} \\ 0 \\ 0 \end{bmatrix}. \quad (36)$$

Para terminar de describir el sistema, se debe de especificar la matriz C. Esta corresponde a las salidas del sistema. En este caso  $\frac{\tau}{K_\tau}$ ,  $\theta$  y  $\dot{\theta}$ .

$$\begin{aligned} y_1 &= \frac{\tau}{K_\tau}y_2 = \theta y_3 = \dot{\theta}, \\ y &= Cx. \end{aligned}$$

La matriz C se especifica igual que la matriz A, solo que en este caso en vez de ser  $x_1$ ,  $x_2$  y  $x_3$ , ahora las columnas corresponden a  $y_1$ ,  $y_2$  y  $y_3$ .

$$C = \begin{bmatrix} -\frac{1}{K_\tau} & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}. \quad (37)$$

Por último se debe de especificar  $\sigma = y_r = x - r$  para poder implementar control LQI, donde  $r$  corresponde a la referencia del sistema (torque, velocidad angular y aceleración angular a la que se desea que el motor trabaje).

$$\sigma = \begin{bmatrix} \tau - \tau_{ref}(t) \\ \theta - \theta_{ref}(t) \\ \dot{\theta} - \dot{\theta}_{ref}(t) \end{bmatrix}. \quad (38)$$

## 6.8. Control óptimo

Un control óptimo utiliza las mediciones de las variables de estado del sistema para estabilizar estas en un punto de operación deseado. Como se indica en [20], estas variables son constantes medidas directamente del sistema a controlar, las cuales están relacionadas entre sí. Estas pueden ser, por ejemplo, los voltajes que pasan por capacitores, corrientes que pasan a través de resistencias o inductores e/o incluso las mediciones de los sensores. En este proyecto, las variables de estado son la corriente que demanda el motor, y la posición y velocidad del motor.

El problema de optimización busca una solución continua en el tiempo, ya que la estimación de los parámetros se da según la evolución del sistema en el tiempo. Esta está sujeta a una función objetivo y se puede solucionar por maximización o por minimización. En la ecuación (39) se muestra cómo la función objetivo está sujeta a ciertas restricciones, las cuales ayudan a determinar los parámetros a encontrar.

$$\begin{aligned} \underset{z}{\text{mín}} \quad & f(z) \\ \text{s.t.} \quad & g(z) \leq 0 \\ & h(z) = 0 \end{aligned} \quad (39)$$

Existen diversos métodos para plantear el control óptimo, como métodos lineales, no lineales, cuadráticos o convexos. En la ecuación (40), se muestra la forma que tiene la optimización no lineal por mínimos cuadrados, sujeta a restricciones de igualdad o de desigualdad.

$$\begin{aligned} \underset{z}{\text{mín}} \quad & \|f(z)\|_2^2 \\ \text{s.t.} \quad & g(z) \leq 0 \\ & h(z) = 0 \end{aligned} \quad (40)$$

## 6.9. Filtro pasa bajas IIR

Como se indica en [21], el filtro IIR tiene como propósito eliminar las frecuencias que se encuentran fuera de la frecuencia de corte que se especifique. Este filtro puede ser utilizado como un filtro pasa bajas, filtro pasa altas, filtro pasa bandas y filtro para bandas. En este caso, se estará trabajando con un filtro pasa bajas de primer orden. El propósito del filtro pasa bajas es eliminar todas las frecuencias que superen la frecuencia de corte determinado. Como se muestra en la Figura 11, las únicas frecuencias que podrán pasar a través del filtro son las que tengan frecuencias menores a las de corte, esta siendo en este caso, como se muestra en la Figura 11, frecuencias menores a  $0.77\pi Hz$ .

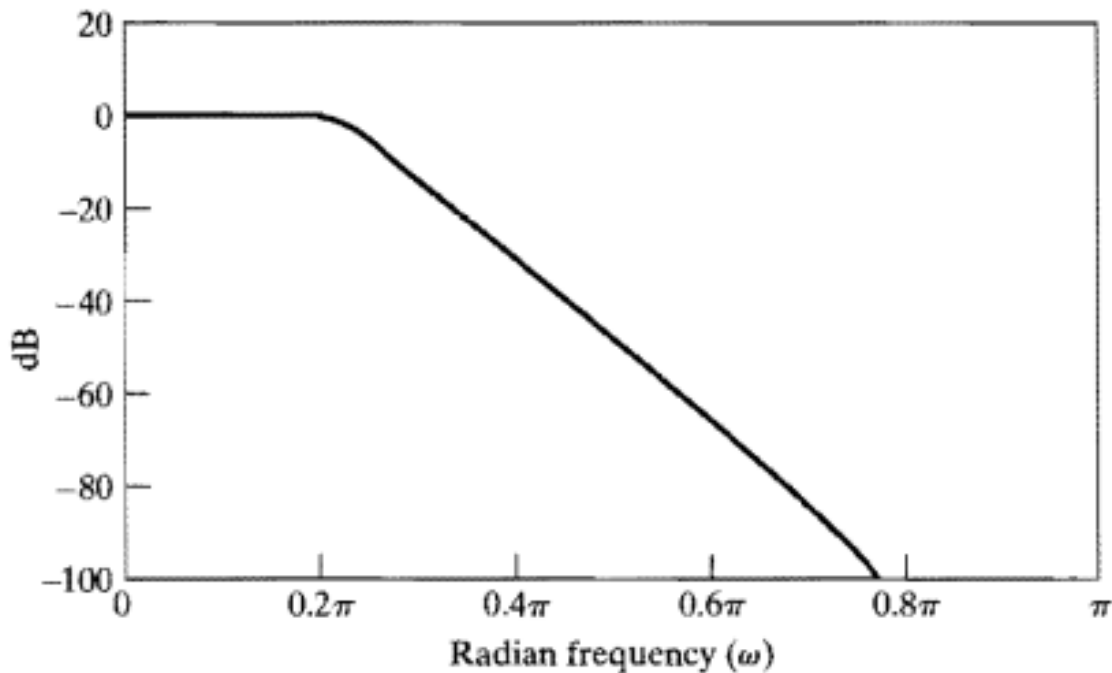


Figura 11: Diagrama de bode de un filtro pasa bajas [21].

En la ecuación (41), se muestra la ecuación del filtro pasa bajas IIR. En este caso, el factor  $\lambda$  es el encargado de indicar qué tan severo será el filtro. Los valores de esta constante se encuentran entre 0 y 1; si es más cercano a 0, el efecto del filtro será menor, mientras que si es más cercano a 1, el efecto del filtro será mayor. Esto quiere decir que este determina qué frecuencias serán capaces de pasar por el filtro y cuáles serán detenidas.

$$H(z) = \frac{1 - \lambda}{1 - \lambda z^{-1}} \quad (41)$$

La ecuación (42) muestra la ecuación de diferencias que se obtiene al momento de realizar la antitransformada  $Z$ . En este caso,  $y_n$  es la salida del filtro,  $y_{n-1}$  es el valor pasado del filtro y  $x_n$  es la señal a filtrar.

$$y_n = (1 - \lambda)x_n + \lambda y_{n-1} \quad (42)$$

## 6.10. Motor Pololu 20.4:1 Gearmotor 25Dx65Lmm 12V

En [22] se pueden encontrar los datos específicos del motor que se utilizará. Se detallan algunos a continuación.

- Trabaja a 12V.
- Tiene una caja reductora con una relación de 20.4:1.
- Posee un encoder 48 CPR en un extremo.

- Tiene una velocidad de giro de 500RPM sin carga aplicada.
- Sin carga aplicada, demanda una corriente de 300mA.



Figura 12: Motor Pololu 20.4:1 Gearmotor 25Dx65Lmm 12V [22].

### 6.11. 48 CPR *Encoder*

En [22] se pueden encontrar los datos específicos del *encoder* que el motor utiliza. Se detallan algunos a continuación.

- Posee 2 canales.
- Valor de conteo de 48 por vuelta.
- Cable rojo: entrada de voltaje para motor.
- Cable negro: referencia para el motor.
- Cable verde: la referencia del *encoder* (GND).
- Cable azul: entrada de voltaje de 3.3V - 20V para el *encoder*.
- Cable amarillo: para el primer canal del *encoder*.
- Cable blanco: para el segundo canal del *encoder*.



Figura 13: 48 CPR *Encoder* [22].

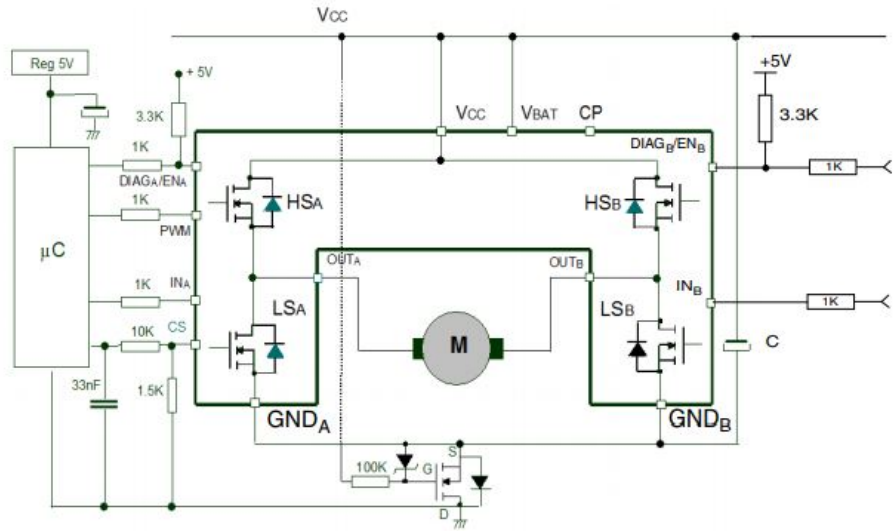
## 6.12. VNH5019 *motor driver*

En [23] se puede encontrar información acerca del *driver* que se utilizará, el cual será el VNH5019 *Motor Driver Carrier*. Este *driver* opera de 5.5V a 24V y puede entregar 12V. La forma en que este trabaja es un puente H que permite el giro del motor en distintas direcciones por una señal PWM. Este acepta una señal con una frecuencia de 20kHz máximo.



Figura 14: VNH5019 *motor driver* [23].

En la Figura [15] se muestra el circuito interno del *driver* VNH5019. El funcionamiento de este es el de un puente H. Se le suministra una señal PWM, dependiendo de esta es que el motor activará y desactivará ciertos transistores para que el motor gire en un sentido y viceversa. Las terminales  $DIAG_a$  y  $DIAG_b$  son las encargadas de activar cada mitad del puente H, si una de estas está desactivada solo se activará medio puente H. Las entradas  $IN_a$  e  $IN_b$ , permiten determinar la comparación deben tener los transistores para que estos cambien de estado.



Note:  
The value of the blocking capacitor (C) depends on the application conditions and defines voltage and current ripple onto supply line at PWM operation. Stored energy of the motor inductance may flyback into the blocking capacitor, if the bridge driver goes into 3-state. This causes a hazardous overvoltage if the capacitor is not big enough. As basic orientation, 500  $\mu$ F per 10 A load current is recommended.

Figura 15: Esquemático de VNH5019 motor driver [23].

### 6.13. Resistencia Shunt

En [24] describen que una resistencia Shunt es un dispositivo que se utiliza para medir la corriente. Este, como su nombre lo dice, es una resistencia, en este caso con un ohmeaje muy bajo ya que al conectarla al circuito utiliza parte de la potencia del mismo. Este gasto tiene que ser muy bajo ya que el único propósito de la resistencia Shunt es medir la corriente y no provocar cambios en el comportamiento del sistema. Para calcular la corriente se utiliza la Ley de Ohm.

En la Figura [16] se muestra un pequeño circuito, en el cual gracias a la resistencia shunt se puede medir el voltaje que sale de la resistencia R1.

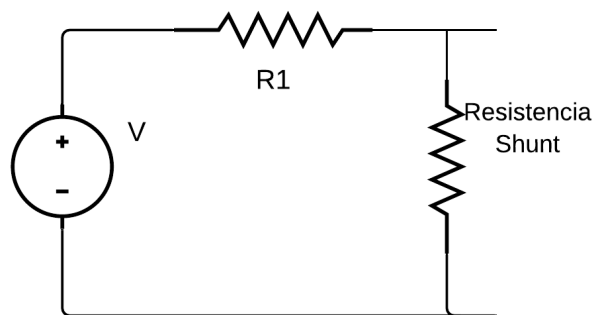


Figura 16: Diagrama de circuito con resistencia shunt.

## 6.14. Fabricación de placas PCB

El propósito de realizar una placa PCB es la reducción y la manufactura de sistemas embebidos. Lo que se busca no solo es realizar un circuito, sino reducir el tamaño de este tanto como sea posible para quedar empaquetado y fijo en un sistema.

Como se muestra en [25], Oshpark es una empresa de manufactura de placas PCB. Esta provee toda la información necesaria a tomar en consideración para la fabricación de sus PCB. Estas placas están hechas por una o varias capas de cobre separadas por un material dieléctrico, el propósito de este último siendo aislar todas las interferencias que se pudiesen llegar a dar entre capas.

Uno de los aspectos más importantes a tomar en consideración es la cantidad de cobre que se utiliza por cada una de las capas, ya que con esta información se puede calcular: el grosor que debe tener cada conexión de cobre que se realice dependiendo de la corriente máxima que soportara, la separación que debe de existir entre cada una y cuánto calor se logra disipar (en caso sea requerido para circuitos de alta potencias). Estas especificaciones se obtienen del fabricante que realizará la placa. En el caso de la empresa Oshpark [25] para la fabricación de una placa de cuatro capas indican que las capas de cobre externas contienen 1oz de cobre cada una y las dos capas internas contienen 0.5oz cada una. Otro detalle que puede brindar la empresa es la distancia mínima que se puede dejar entre líneas de cobre, ya que no poseen herramientas más pequeñas. En caso el cálculo que se realice sea menor a la distancia mínima que la empresa puede realizar, se debe escoger como distancia mínima la del fabricante.

Durante el diseño de las placas PCB, hay algunas consideraciones que deben tomarse en cuenta. Por ejemplo, si se trabaja con múltiples capas, se debe evitar colocar cables en paralelo entre capas o que recorran una distancia considerable unos sobre el otro, pues esto provocaría que la placa dejase de funcionar, ya que se comportaría como un capacitor. Otra consideración es el no colocar líneas de cobre con un cambio de dirección de  $90^\circ$ , dado que al haber menos material en estos cambios de dirección, el cobre podría sobrecalentarse, levantando el cobre de la placa y arruinando la misma. Por último, las capas están conectadas entre ellas mediante vías, las cuales pueden unir dos o todas las capas que se trabajen entre sí, también se debe de evitar usar en exceso las capas internas, ya que las capas externas pueden manejar corrientes mayores y permiten la disipación del calor.



## 7.1. Metodología

Se inició creando un diagrama de bloques, con el propósito de una mejor visualización del sistema y las configuraciones que se debiesen crear para poder controlar el motor correctamente. En la Figura 17 se muestra a gran escala como se realizarán las conexiones del circuito a utilizar.

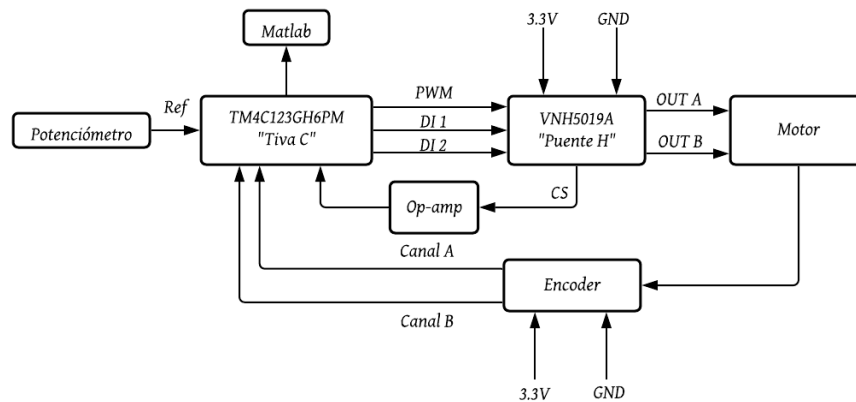


Figura 17: Diagrama de bloques conexiones.

Todo el control y el procesamiento de las señales se realizó con el microcontrolador TM4C123GH6PM, dentro de una tarjeta *Tiva C: Launchpad* [26]. Se tomó esta decisión basado en la velocidad de procesamiento que este otorga, la cual es lo suficientemente rápida

para realizar todo el procesamiento de las señales y realizar el control. Para realizar las configuraciones para la *Tiva C*, se utilizó la librería *TIVWARE* [27] y el programa *Code Composer Studio* de *Texas Instruments*.

### 7.1.1. Configuraciones de la *Tiva C*

El primer paso que se realizó para la configuración del microcontrolador fue establecer una frecuencia de  $16MHz$  y se establecieron dos pines como salidas digitales para controlar el cambio de giro del motor. Estos fueron los pines A5 y A6.

#### Configuración ADC lectura de la referencia de posición.

Siguiendo el diagrama de la Figura [17], lo primero por configurar fue la lectura de la referencia, en este caso un potenciómetro. Para esto se requirió utilizar una entrada ADC (*Analog-to-digital*), el cual tiene como propósito recopilar los datos analógicos entregados por el potenciómetro a una señal digital, la cual toma los valores de 0 a 4095. Se utilizó el pin B5 del microcontrolador para la recopilación de datos, cada vez que suceda un disparo (*trigger*) del procesador interno. En este microcontrolador se puede configurar que tome hasta 8 muestras cada vez que se active la lectura, pero en este caso como se desea realizar un procesamiento rápido se configuró para que solo se tome un dato, cada vez que se llega al final de la interrupción para la lectura de datos.

#### Configuración ADC lectura de la corriente consumida por el motor

Para la medición de la corriente demandada por el motor, la cual proviene del pin CS del VNH5019A se requirió primero pasar esta señal por un amplificador operacional. El propósito de este es amplificar la señal por un factor de 4.5232 para que la *Tiva C* pudiese tener una mejor lectura ya que esta se encuentra en la escala de  $mV$ .

Luego del amplificador esta se leía gracias a otra entrada ADC situada en el puerto E5. El pin ADC se configuró de la misma manera que el ADC para la lectura de la referencia de posición. Para obtener el valor de la corriente, debido a que una entrada ADC solamente puede realizar lecturas de voltaje, se realizó una conversión para obtener un valor de corriente en  $mA$ . Esta conversión se muestra en la ecuación (43)

$$i = \left( \frac{ADC}{4.5232} \right) \left( \frac{3.3}{4095} \right) \left( \frac{1000}{0.144} \right) \quad (43)$$

#### Configuración lectura *encoder*

El siguiente paso fue configurar los pines para la lectura del *encoder*. Para esto se utilizó la librería *Quadrature Encoder* (QEI) de *TIVWARE*. El primer paso fue definir los pines a utilizar para la lectura de los 2 canales del *encoder*, los cuales fueron D6 para el canal A y el pin D7 para el canal B del *encoder*. Estos pines al ser GPIO poseen un bloqueo por

defecto, el cual se debe desactivar si se desean utilizar. Se desactivaron las interrupciones de la librería QEI, para asegurar que estas interrupciones no se activaran en ningún momento, ya que pueden provocar que la *Tiva C* se detenga en alguna línea de código si estas no son desactivadas. Se debe de tomar en cuenta que para la configuración correcta del *encoder* se debe de configurar distintos aspecto como lo son; la cantidad de pulsos por revolución, si la lectura será utilizando ambos canales y que tipo de señal se recibe.

Para el cálculo de velocidad y posición, se configuró que se utilizaran ambos canales. Al haber desactivado las interrupciones, se debe de colocar que no se resetee la posición al ocurrir una de estas. El *encoder* que se utilizó entrega señales cuadradas, por lo que se configuró el conteo de estas señales por cada canal. Se configuró que las señales leídas de los canales del *encoder* no se cambiarán de posición entre ellas. Para la toma de datos de la velocidad, se debió especificar la frecuencia en la que los datos serán leídos, para especificar el período de muestreo. En este caso un período de  $10ms$ , que son equivalentes a una frecuencia de  $100Hz$ . Por último se calculó la cantidad de pulsos a contar para que se realice una revolución completa del motor. Para realizar esto se multiplicó la cantidad de pulsos que el *encoder* posee por la reducción de velocidad que provoca la caja reductora del motor ( $T_{pulsos} = (No.Pulsos)(Reduccion)$ ). Realizando este cálculo se encontró que se deben de contar 979 pulsos para completar una revolución.

### Configuración de la señal PWM

La salida del controlador hacia el motor fue mediante una señal PWM. Se utilizó una señal con una frecuencia de  $20kHz$  para el controlador PID y a  $40kHz$  para el controlador LQI, para especificar esta frecuencia se tomó en consideración el máximo que puede recibir el puente H utilizado. Se debe de configurar el reloj para generador PWM, el cual se seteo para que dividiese por uno el reloj global del microcontrolador. En este caso se utilizará el generador cero para la señal PWM del generador interno. La salida de este se encuentra en el pin B6. Se colocó que la generación de la señal se realizará en el flanco negativo del reloj y que este no sea síncrono. Por último se especificó el periodo de muestreo según la frecuencia utilizada. Para esto se utilizó la frecuencia global del microcontrolador, en este caso de  $16MHz$ , se dividió por uno ya que fue por lo que se especificó al setear el reloj del PWM y por último al resultado de esto se le dividió la frecuencia que se desea tener en la señal PWM.

Para el ancho de banda (*duty cycle*) que se desea colocar en el PWM se especificó una resolución de 4095. Cuando se trabaja con una señal PWM, se deben especificar límites ya que si se llega a los límites de resolución, la señal se saturará y solo enviará un valor continuo de voltaje. En este caso se utilizó como límites un valor de 10 y 3995. Para enviar el dato correctamente se debe de multiplicar el valor deseado por el cálculo realizado de la frecuencia y el resultado dividirlo dentro de la resolución deseada en este caso 4095.

### Configuración UART

Se realizó una comunicación UART con el propósito de enviar los datos a *Matlab* para graficar la entrada al controlador y la salida que se obtuviese de este. Para esta comunicación

se utilizó un *baud rate* de 115200.

En la *Tiva C* se utilizó el módulo UART0 debido a que este permite enviar los datos mediante USB hacia la computadora. Primero se configuraron los pines en los que se encuentra este módulo, siendo el pin A0 como RX (recibe los datos) y el pin A1 como TX (envía los datos). Se especificó que los datos enviados fuesen de 8 bits, el cual es el tamaño más grande que permite enviar la *Tiva C*, mediante UART. Se especificó un bit como paro y que no hubiese paridad.

Se configuró una interrupción para esta comunicación. Esta tiene la función de permitir que los valores se envíen de forma ordenada y que los datos no se estén enviando continuamente. Esta sucede siempre que se recibe un dato provocando que se active una bandera, la cual se encarga de iniciar la comunicación. Para configurar completamente esta interrupción se debe de especificar en el archivo `tm4c123gh6pm_startup_ccs.c`, donde se encuentran todas las interrupciones que posee el sistema.

En *Matlab* para la comunicación, únicamente se especificó el puerto 'COM' en el que se encuentra conectada la *Tiva C* y el *baud rate* de la comunicación.

## Configuración *TIMER0*

Se configuró el *TIMER0*, para poder controlar correctamente un periodo de muestreo, el cual es la velocidad con la que se procesarán los datos de entrada y salida del controlador. Para este se configuró que fuese un temporizador periódico, con una frecuencia de  $1kHz$  lo cual equivale a decir un periodo de muestreo de  $1ms$  para ambos casos de los controladores. Se configuró una interrupción que sucediese cada vez que el periodo de muestreo se cumpliera. Está activa una bandera, la cual es la encargada de indicar que se debe tomar los nuevos datos. Esto es similar a la interrupción para la comunicación UART. Por último también se especificó esta interrupción en el archivo `tm4c123gh6pm_startup_ccs.c` del proyecto.

### 7.1.2. Procesamiento de datos del microcontrolador

Luego de haber realizado todas las configuraciones requeridas para que el microcontrolador pudiese trabajar adecuadamente, se procedió a realizar la programación necesaria para cumplir con el funcionamiento adecuado de la Figura 17. En la Figura 18 se muestra un diagrama de bloque que permite visualizar de una forma resumida todo el procesamiento interno que realiza el microcontrolador *Tiva C*.

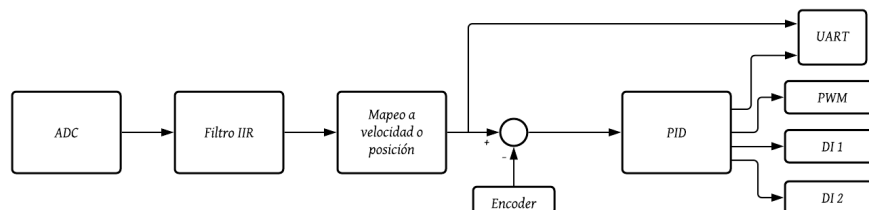


Figura 18: Diagrama de bloques procesamiento *Tiva C*.

El primer paso de este diagrama es la lectura del puerto ADC. Como se mencionó con anterioridad, esta señal corresponde a la de un potenciómetro. Cuando se trabaja con sensores analógicos, estos puede que presenten una pequeña perturbación al momento de transmitir su señal. Esto no es debido a que el sensor pueda estar malo o no, si no que se debe a las perturbaciones provocadas por el ambiente. Por esto se debe aplicar un filtro, para suavizar la señal y evitar que este ruido provoque datos erróneos no deseados. Para este caso se utilizó un filtro de media móvil exponencial.

## Mapeo de valores

Este mapeo se realizó para trabajar siempre en el mismo sistema de unidades tanto la referencia como la retroalimentación del potenciómetro. Para realizar este mapeo se utilizó la función de la recta. Gracias a esto, este corresponde a  $y = mx + b$ , en donde  $m$  es la pendiente de la función,  $x$  es el valor del potenciómetro y  $b$  es el desfase que existe en el eje  $y$ .

- Mapeo a posición:** En la Figura [19](#) se muestra la función de la recta para pasar los valores del potenciómetro a radianes. La ecuación a utilizar queda de la siguiente forma,  $y = \frac{2\pi}{4095} ADC$

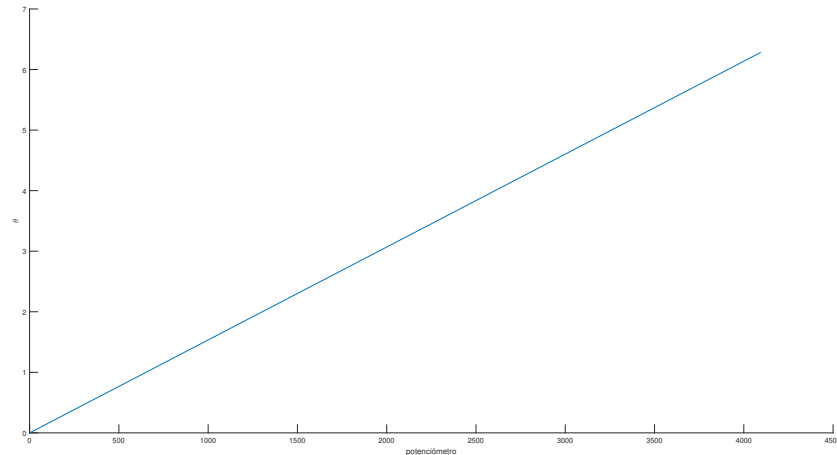


Figura 19: Gráfica de recta potenciómetro vs posición.

- Mapeo a velocidad:** En la Figura [20](#) se muestra la función de la recta para pasar los valores del potenciómetro a  $rad/s$ . Por lo que la ecuación a utilizar queda de la siguiente forma,  $y = \frac{500*2\pi}{4095*60} ADC$ .

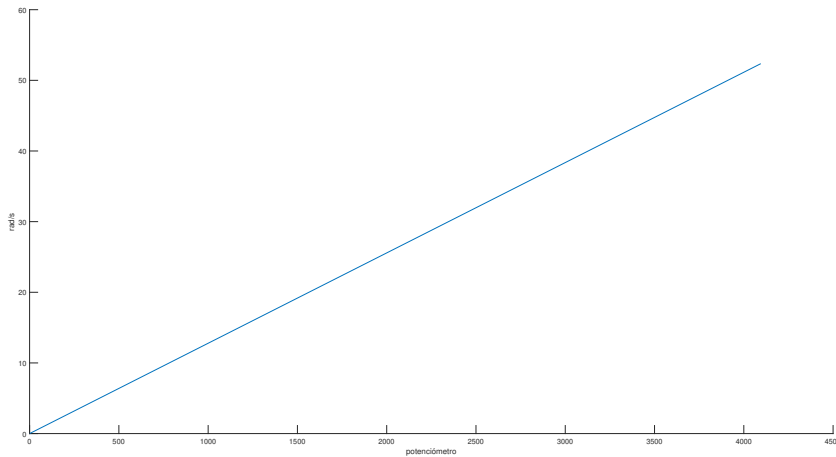


Figura 20: Gráfica de recta potenciómetro vs velocidad.

### Obtención de velocidad y posición *encoder*

Para obtener los datos del *encoder*, se utilizan funciones de la librería QEI con la que se configuró el sensor. Gracias a esta, este se vuelve un proceso mucho más sencillo de realizar.

- Obtención valor de posición** Al obtener el valor de lectura del sensor, este retorna la posición del *encoder* mediante el conteo general de los pulso retornados. Este valor se puede convertir a grados a radianes, dependiendo como se desea trabajar. En este caso se trabajó en grados, por lo que el resultado obtenido se multiplicó por  $360^\circ$  y se dividió por la cantidad de pulsos para completar una revolución del motor.

$$rad = \frac{Sensor * 2\pi}{No.Pulsos}$$

En el caso de la posición, los valores del *encoder*, solo representaban valores de 0 a  $2\pi$ , por lo que no era posible representar valores negativos ni mayores a  $2\pi$ . Por lo se realizó una programación para que los valores de este fuesen sumándose y restándose.

- Obtención valor de velocidad** Al obtener el valor de lectura del sensor, este retorna la cantidad de pulsos leídos en el periodo de tiempo configurado. Este valor se puede convertir a rpm, al multiplicar este valor por la frecuencia de muestreo, por 60 segundos de un minuto dividido la cantidad de pulso que equivalen a una revolución del motor.

$$rad/s = \frac{Sensor * Hz * 60 * 0.10472}{No.Pulsos}$$

### Creación controlador PID

Como se mencionó anteriormente, un controlador PID depende del error calculado por la diferencia de la referencia con la retroalimentación del sensor. Este error se utiliza para

calcular el error para la parte proporcional, integral y derivativa del PID. El error proporcional como se calculó, fue simplemente restar la señal de referencia con el valor retornado por el sensor, en este caso la señal mapeada del potenciómetro con la señal retornada por el *encoder*.

Para calcular el error de la parte integral del controlador, se debe de utilizar el error proporcional y el error integral pasado. Estos dos valores se suman para obtener el error integral actual. Por último se calculó el valor del error derivativo. Para calcular este valor se utilizó el valor pasado del error proporcional, para restárselo al valor del error proporcional actual.

Una vez calculados los tres errores se procedió a utilizar la ecuación (4). Para realizar este proceso, se creó una función que se llama dentro del periodo de muestreo del *TIMER0* cada vez que se requiriese el cálculo del nuevo valor de salida del PID y se almacenaron los valores pasados de cada error en un *struct*.

En este caso el valor absoluto de salida del controlador PID, se debe de retransformar, ya que estos se retornan en las escalas de grados sexagesimales o RPM dependiendo de cuando se use este controlador. Se debe de realizar esto ya que el PWM está configurado para recibir los valores en la escala de un ADC.

En el caso del control de posición, se debe de especificar un cambio de giro, debido a que se puede elegir una posición menor a la actual. Aquí entran en función los pines digitales configurados, ya que uno de estos se debe encender para que el motor gire en una dirección y si se desea el cambio de esta dirección se deben de invertir estos pines. Para realizar este cambio de giro se utilizó la salida del controlador PID. Esto es debido a que este cambia de signo dependiendo si se debe de aumentar la posición o disminuir.

## Comunicación UART

- **Comunicación *Tiva C*:** Como se mencionó en las configuraciones, el tamaño de dato más grande que se puede enviar desde la *Tiva C* es de 8 bits. Debido a que se está trabajando en este caso con valores de 32 bits, estos se deben de separar en 4 grupos de 8 bits y enviarlos individualmente. Para mantener ordenados estos valores, se almacenaron en un *array* para mantenerlos ordenados. No se utilizó un ciclo *for* para el envío de estos, ya que el ciclo *for* provoca un *delay* en el código y esto no se desea. Por esto se utilizó una interrupción, la cual al activarse envía valor por valor una vez se inició esta comunicación.
- **Comunicación *Matlab*:** Como al tener una interrupción del lado de la *Tiva C*, esta tiene que ser activada al enviar una valor cualquiera desde *Matlab*. Una de las grandes ventajas de la comunicación UART de *Matlab* es que al momento de recibir este se queda esperando hasta recibir un valor. Desde *Matlab*, se puede especificar cuántos valores se reciben y estos se almacenan en un *array*. Por último estos valores se convierten a valores binarios y se suman para poder obtener nuevamente el valor original de 32 bits y se grafican comparándolos entre ellos.

### 7.1.3. PID velocidad

Para tunear las constantes del controlador PID, primero se definió únicamente la constante  $K_p$ . Al realizar esto se obtuvo un controlador tipo P que permitía variar la velocidad del motor. A pesar de que esta constante permitió controlar la velocidad, el tiempo de asentamiento del controlador era muy lento. Para reducir este retraso en la reacción se pasó a ajustar la constante  $K_I$ . En ocasiones esta puede provocar que el controlador se vuelva inestable, pero si se aumenta poco a poco se puede lograr los ajustes necesarios.

### 7.1.4. PID posición

Para tunear las constantes del controlador PID, se inició de la misma forma que el controlador de velocidad definiendo primero la constante  $K_p$ . Al realizar esto se obtuvo un controlador tipo P que permitía que el motor llegase a la posición deseada, pero el problema que presentaba en este caso fue que existía una presencia de *overshoot*. En el caso de un motor esto es altamente negativo. Esto se debió a que provoca un un cambio de giro brusco en el momento de moverse a la posición nueva. En este caso para un motor el cambio brusco de giro puede provocar que este se dañe, ya que para llegar a su posición final ira cambiando de giro bruscamente provocando cortos en la fuente.

Para reducir este *overshoot* es útil la constante  $K_d$ . Pero esta se debe de cuidar ya que si esta se aumenta excesivamente, el controlador se volverá inestable. Se obtuvo muy buenos resultados al aumentar esta variable, pero surgió un nuevo problema. El controlador PID no poseía suficiente fuerza para llegar a la posición deseada. Por esto se agregó una constante  $K_i$ , la cual permitió que el controlador no solo llegase a la posición deseada sino que también permitió un movimiento fluido y sin *overshoot*.

### 7.1.5. Comparación de controladores con *Matlab*

Para realizar esta comparación se utilizó las funciones de transferencia que describen la posición y velocidad del motor. Para obtener estas funciones de transferencia, se utilizó la respuesta al impulso del motor. Esta respuesta se puede observar gráficamente en la Figura [21](#). Se utilizó la aplicación de *System Identification*, la cual permite relacionar la entrada del controlador con la salida de este. Se utilizó *Simulink* de *Matlab*, con el propósito de comparar el comportamiento del controlador PID con la función de transferencia obtenida y los datos recopilados desde la comunicación UART como se muestra en la Figura [22](#).

- **Función de transferencia de velocidad:**

$$G(s) = \frac{0.6661}{s + 5.3817}.$$

- **Función de transferencia de posición:**

$$G(s) = \frac{0.7489}{s^2 + 4.0946s + 0.1762}.$$



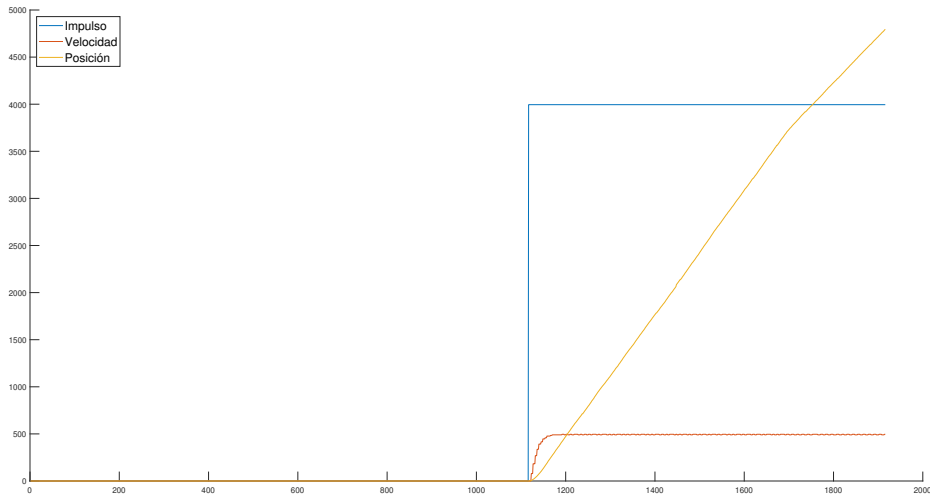


Figura 21: Respuesta al impulso del motor.

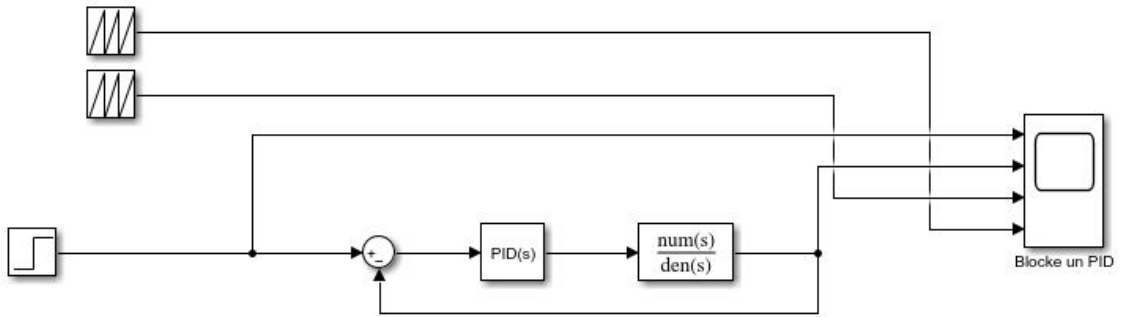


Figura 22: Diagrama de bloques *Simulink*.

### 7.1.6. Comparación PID velocidad

Para realizar la comparación de los controladores, se tuvo que convertir el valor de las variables del controlador PI a analógicas. Como se muestra en la ecuación (4), las constantes  $K_i$  y  $K_d$  analógicas, están acompañadas de un periodo de muestreo ( $\Delta t$ ). Como en este caso se desea pasar al valor analógico la variable  $K_i$ , esta se debe de dividir por el periodo de muestro de la interrupción del (TIMER0), esta siendo de  $0.01s$ . Esto sería suficiente para realizar la simulación de controlador PID, pero ya que se realizó un mapeo de la salida del PID, esta constante se debe de mapear para poder trabajarla. Por lo que en este caso se multiplicó el valor obtenido anteriormente por  $\frac{4095}{500}$ .

### 7.1.7. Comparación PID posición

Para el caso de la comparación del PID de posición, se realizó el mismo proceso de conversión de las constantes  $K_i$  y  $K_d$ . En este caso para la constante  $K_d$ , en vez de dividir el periodo de muestreo este se tuvo que multiplicar por la constante digital. También la transformación en esta caso cambió a ser  $\frac{4095}{360}$ .

### 7.1.8. PID de posición y velocidad en cascada

Una vez se logró ambos controladores por separado, se procedió a realizar el control de la posición y la velocidad al mismo tiempo. Para esto se realizó la configuración de controladores que se muestra en la Figura 23. Esta configuración se le denomina controladores PID en cascada y permite controlar tanto la posición como la velocidad del motor. La gran ventaja que tiene esto es que al ser la referencia del PID de velocidad la salida del de la posición, este permitirá que al acercarse a la posición final deseada el motor desacelere y tenga un paro suave.

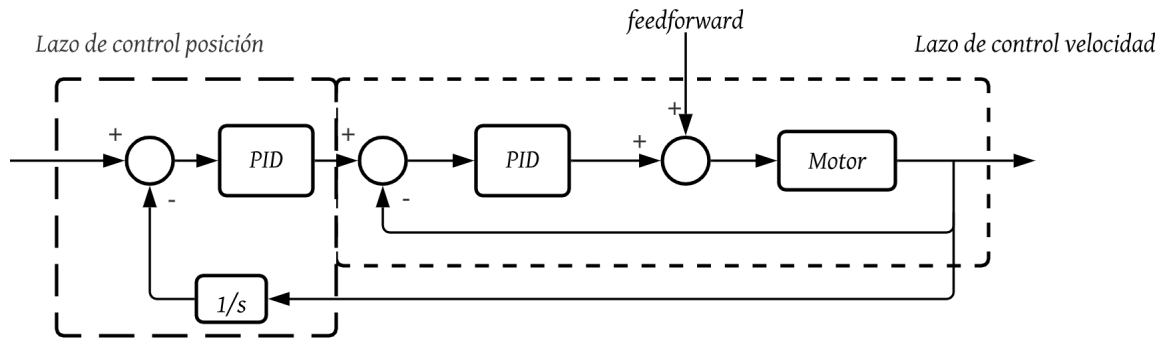


Figura 23: Diagrama de bloques PID en cascada.

Luego de haber unido los controladores en la programación se realizaron pruebas y ajustes mínimas a las constantes de los PID para un buen resultado.

### 7.1.9. Optimización

El propósito del planteamiento de la optimización es poder determinar los valores aproximados de las constantes especificadas en el modelado del motor. Una optimización por mínimos cuadrados posee la forma de la ecuación (40), pero en este caso, al plantear el problema, se obtiene la forma de la ecuación (44)

$$\begin{aligned}
 \min_z \quad & \int_0^T \|y_{real}(t) - Cx(t; z)\|_2^2 dt \\
 \text{s.t.} \quad & \dot{x} = A(z)x + B(z)u \\
 & lb \leq Z \leq ub
 \end{aligned} \tag{44}$$

La optimización de la ecuación (44) posee una integral, indicando que la optimización

a resolver puede llegar a obtener un resultado de infinitas dimensiones. Para evitar este problema, se realizó una discretización, que a la vez presenta la ventaja de simplificar el problema. Al discretizar el problema se obtiene el resultado de la ecuación (45).

$$\begin{aligned} \min_z \quad & \sum_{k=0}^K \|y_{real}(k\Delta t) - Cx(k\Delta t; z)\|_2^2 dt \\ \text{s.t.} \quad & x_k = A_d(z)x_{k-1} + B_d(z)u_k \\ & lb \leq Z \leq ub \end{aligned} \quad (45)$$

en donde  $Z$  son las constantes del motor a encontrar.

$$Z = \begin{bmatrix} Z_1 \\ Z_2 \\ Z_3 \\ Z_4 \\ Z_5 \\ Z_6 \\ Z_7 \end{bmatrix} = \begin{bmatrix} R \\ L \\ K_\varepsilon \\ K_\tau \\ k \\ b \\ J \end{bmatrix}$$

Durante el proceso de la estimación de los parámetros y el ajuste de las restricciones para la optimización, se plantearon dos tipos de restricciones adicionales por separado, las cuales fueron útiles para determinar si el sistema era controlable o no.

### Primer planteamiento de restricciones adicionales

El primer planteamiento de restricciones adicionales fue forzar que un polo del sistema estuviese en cero. Esto se planteó basándose en el modelado usual que se utiliza para los motores DC con escobillas. Se sabe que este modelado, al eliminar la dependencia del torque del motor, coloca un polo en cero. Por esta razón se utilizó un polinomio característico para hacer esto posible.

$$\chi_{A(z)}(\lambda) = \det(\lambda I - A(z)) = 0 \quad (46)$$

En la ecuación (46) se muestra un problema de eigenvalores, donde  $\lambda$  es el polo a evaluar. Para que un polo se encuentre en cero se debe de cumplir la siguiente condición:

$$\begin{aligned} \min_z \quad & \sum_{k=0}^K \|y_{real}(k\Delta t) - Cx(k\Delta t; z)\|_2^2 dt \\ \text{s.t.} \quad & x_k = A_d(z)x_{k-1} + B_d(z)u_k \\ & lb \leq Z \leq ub \\ & \chi_{A(z)}(0) = 0 \end{aligned} \quad (47)$$

El polinomio característico en este caso fue una relación entre  $Z_1$ ,  $Z_2$ ,  $Z_5$  y  $Z_7$ , obteniendo así:

$$Z_1 Z_5 - Z_2 Z_7 \varepsilon = 0$$

en donde  $\varepsilon$  es una nivel de tolerancia definido según la respuesta obtenida y si esta es la esperada. En este caso, se decidió utilizar una tolerancia de 0.01.

## Segundo planteamiento de restricciones adicionales

El segundo planteamiento de restricciones adicionales fue un planteamiento de desigualdades. Este se planteó con la constante de la inercia, fricción y la constante K de la parte mecánica del motor. Lo que se buscó con esta restricción fue que las constantes mecánicas del motor tuviesen una relación, la cual fue que la constante de fricción debe estar entre la constante K y la inercia según el problema de optimización de la ecuación (48). Esto garantiza que las constantes en cuestión no serán demasiado grandes, lo cual sucedía al momento de no poseer esta restricción.

$$\begin{aligned} \min_z \quad & \sum_{k=0}^K \|y_{real}(k\Delta t) - Cx(k\Delta t; z)\|_2^2 dt \\ \text{s.t.} \quad & x_k = A_d(z)x_{k-1} + B_d(z)u_k \\ & lb \leq Z \leq ub \\ & b - J \leq 0 \\ & k - b \leq 0 \end{aligned} \tag{48}$$

### 7.1.10. Control LQI implementado en microcontrolador

Una vez se verificó con una simulación en *Matlab* que el sistema era completamente controlable, se procedió a la implementación del controlador LQI en la *Tiva C*. El proceso para realizar este controlador fue ajustar las constantes del LQI manualmente, ya que las retornadas por *Matlab* provocaban que el sistema del mundo real fuese inestable.

### 7.1.11. Fabricación de placa PCB y Carcasa para el actuador

Para la realización del esquemático del controlador y placa PCB se utilizó el programa *Altium Designer* [28]. Se inició realizando las librerías de los componentes eléctricos a utilizar para la representación de cada uno de estos componentes y realizar su representación 3D para la placa PCB.

Posterior a la realización del esquemático para el correcto funcionamiento de la *Tiva C* y las conexiones necesarias para cada componente, se realizó la placa PCB. Se realizó una placa de cuatro capas, ya que lo que se busca es que el actuador sea compacto para ser utilizado en distintas aplicaciones.

Al terminar el diseño electrónico, se procedió a exportar la placa PCB como un archivo .step. Para la realización de la carcasa, se utilizó el programa *Autodesk Inventor* [29], del cual se exportaron los archivos .step proveídos por los fabricantes y la placa PCB realizada. Se organizó todo lo más compacto posible y se realizaron planos de la carcasa.

## 7.2. Resultados

### 7.2.1. Filtro pasa bajas IIR de primer orden para señal de entrada del potenciómetro

En la Figura 24, se muestra el resultado obtenido al aplicar el filtro utilizando un factor de  $\lambda = 0.95$ .

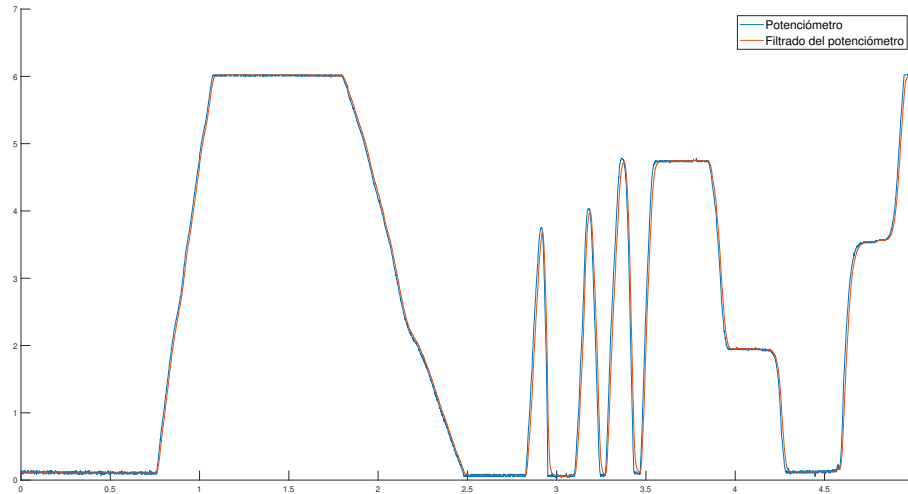


Figura 24: Señal de potenciómetro vs salida del filtro EMA.

Al observar el resultado se puede decir que este es bastante útil ya que las perturbaciones que podían afectar al sistema son eliminadas a cambio de introducir un pequeño lag a la referencia del sistema. Calculando el *group delay* que se presenta al utilizar este filtro, se obtuvo un tiempo de  $52.632\mu s$ . Al ser este tiempo pequeño, el *delay* que presenta a la señal del potenciómetro no es significativo para la realización del controlador PID.

### 7.2.2. PID velocidad

En la Figura 25, se puede observar el primer resultado obtenido solo al colocar una constante del controlador. Esto se logró con una constante  $K_p = 5$ .

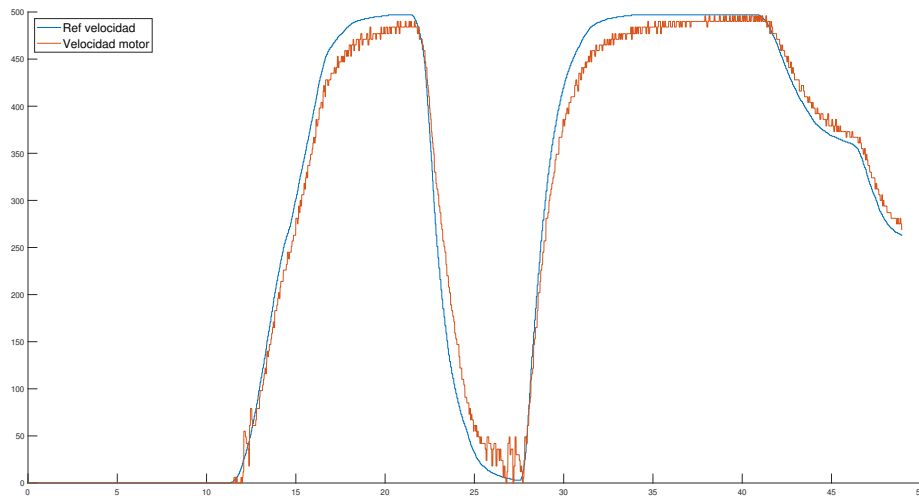


Figura 25: Resultado PID posición.

En la Figura 26 se utilizó un controlador PI para cumplir con el control de velocidad. Para lograr este resultado se utilizó una constante  $K_p = 5$  y  $K_i = 0.5$ .

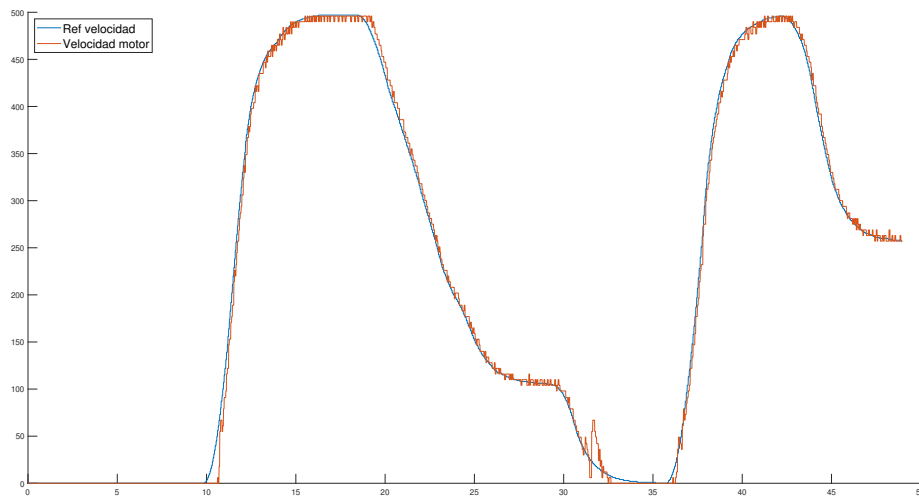


Figura 26: Resultado PID posición.

### 7.2.3. PID posición

En la Figura 27 se puede observar el controlador P generado con una constante  $K_p = 5$ .

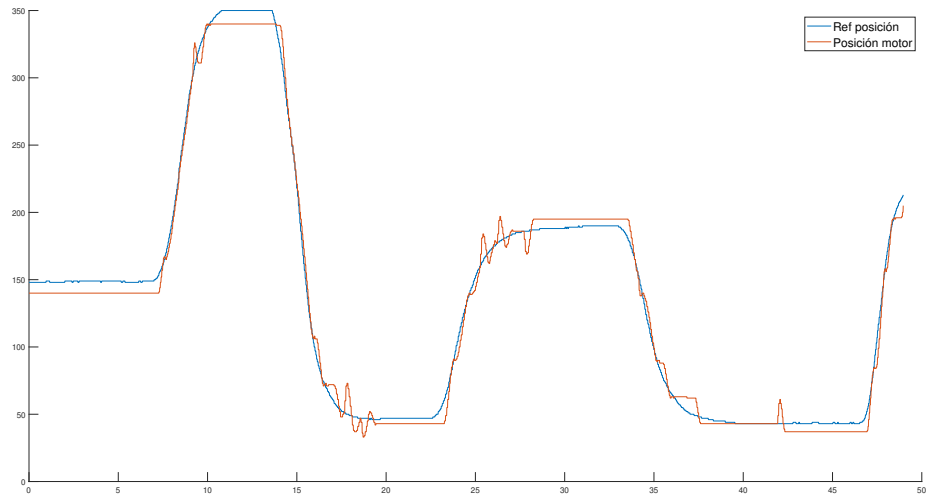


Figura 27: Resultado PID posición *overshoot*.

En la Figura 28 se muestra la respuesta del PID con constantes  $K_p = 5$ ,  $K_i = 0.5$  y  $K_d = 1.5$ .

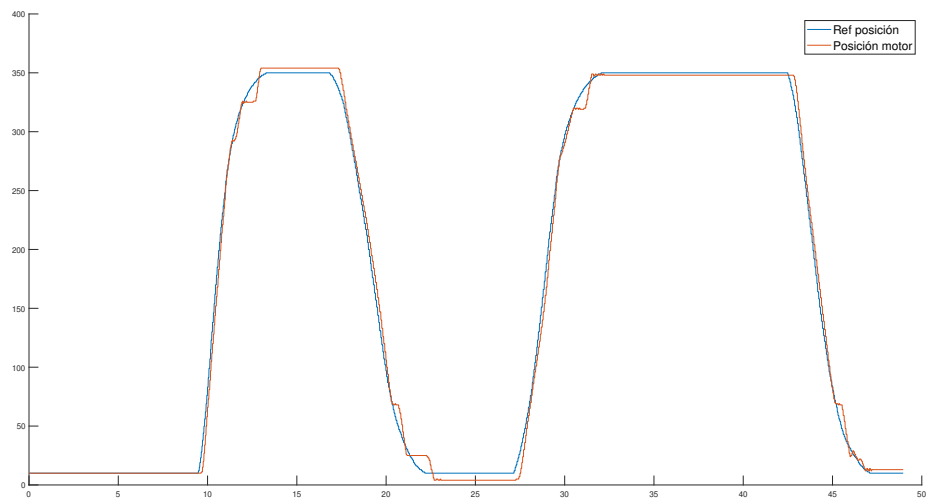


Figura 28: Resultado PID posición.

#### 7.2.4. Comparación PID velocidad

En la Figura 29 se observa la respuesta al impulso de la velocidad tanto simulada con bloque de *Simulink* utilizando las constantes  $K_p = 5$  y  $K_i = 50$  y los datos recopilados de la comunicación UART.

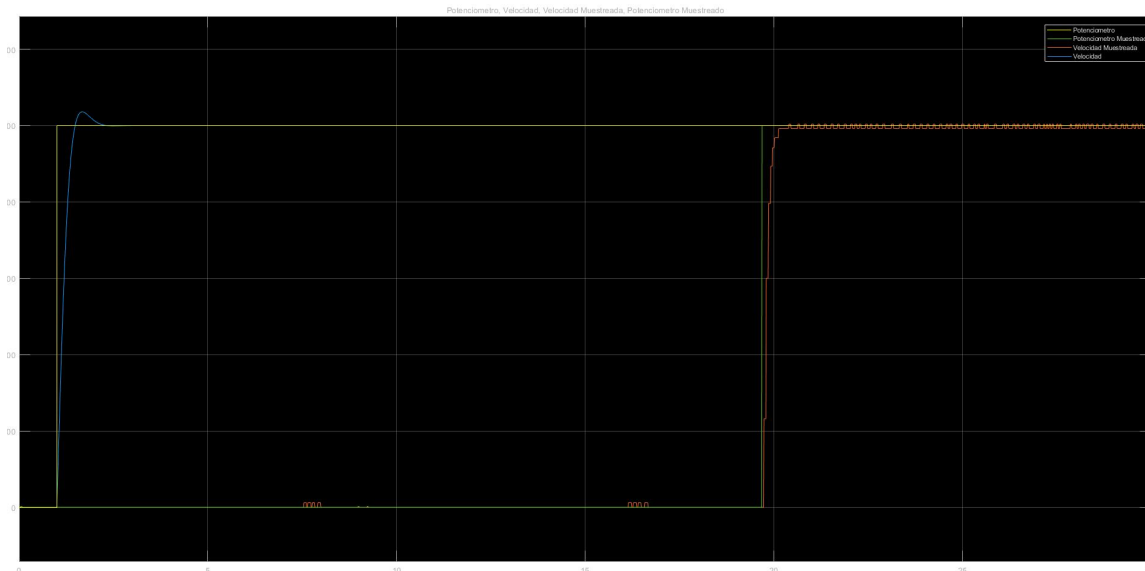


Figura 29: Resultado PID simulado y PID experimental.

Se compararon los valores del tiempo de subida, obtenido para los datos obtenidos por UART, un tiempo de subida de  $280.7ms$  y para la simulación de *Matlab* un tiempo de subida de  $379ms$ . Al compararlos se pudo concluir que el PID programado es más óptimo que el simulado, debido a que no presenta un *overshoot* y el tiempo de subida del experimental es menor.

#### 7.2.5. Comparación PID Posición

En el caso del PID de posición la respuesta no fue la esperada como se muestra en la Figura 30. En este caso el PID simulado no fue lo suficientemente estable para poder estabilizar el sistema a la respuesta del impulso.



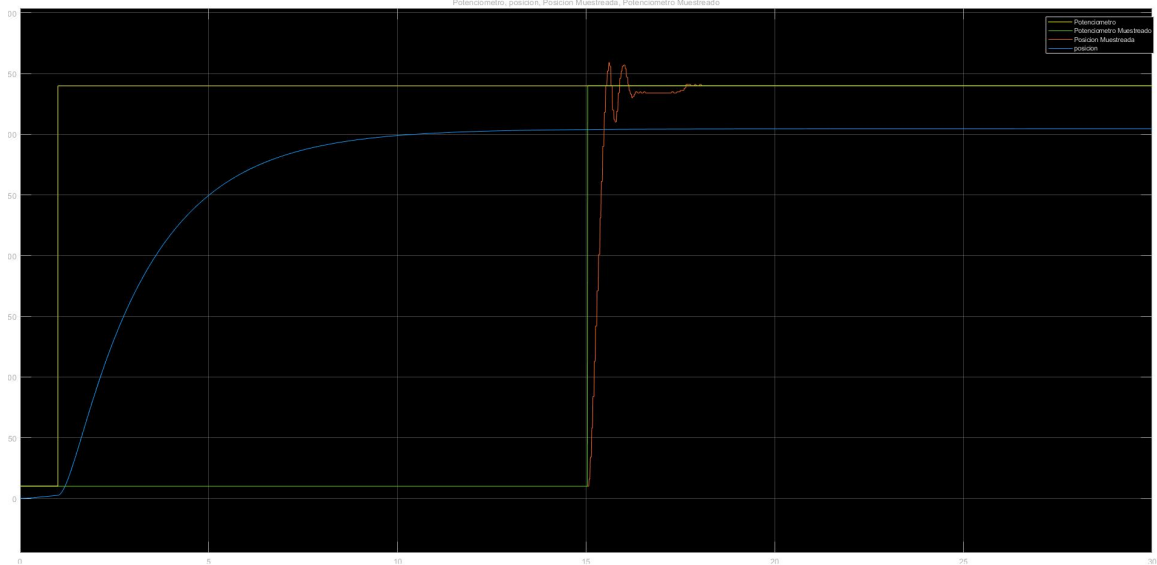


Figura 30: Resultado PID simulado y PID experimental.

Al observar este resultado, se procedió a reajustar el PID con la herramienta *pidTuner* de *Matlab*. Se obtuvieron como resultado las constantes  $K_p = 346.3$ ,  $K_i = 186.71$  y  $K_d = 102.73$ . Al probar esto en el controlador se concluyó que estas constantes a pesar de ser las más óptimas según *Matlab*, no funcionan para este controlador. Esto se debe a que el motor siempre está en constante cambio de giro brusco, provocando un corto en la fuente de alimentación. Esto no solo no funciona debido al corto que crea, si no que a corto plazo el motor se arruinará debido al constante golpeteo de este.

Debido a esto se decidió continuar con las constantes calculadas experimentalmente, ya que el tiempo de subida es de  $336ms$ , lo que es bastante bueno para este controlador.

### 7.2.6. PID de posición y velocidad en cascada

En las Figuras [31](#) y [32](#), se muestra el resultado al combinar ambos controladores. Al combinar estos, se puede observar que la posición posee un mejor movimiento este siendo más fluido, preciso y rápido. Las constantes que se utilizaron para el PID de posición fueron  $K_p = 3.5$ ,  $K_i = 0.000005$  y  $K_d = 1.75$  y para el del PID de velocidad, fueron  $K_p = 5$ ,  $K_i = 0.5$ .

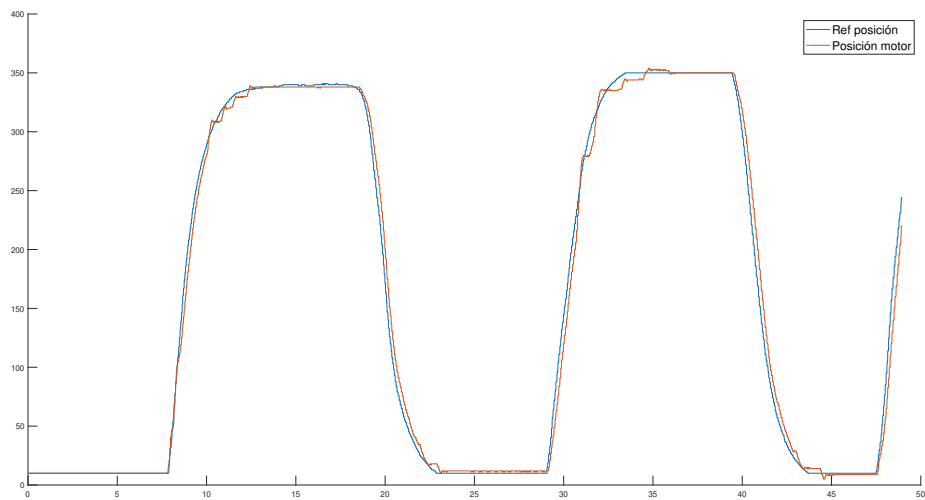


Figura 31: Resultado de la posición PID en cascada.

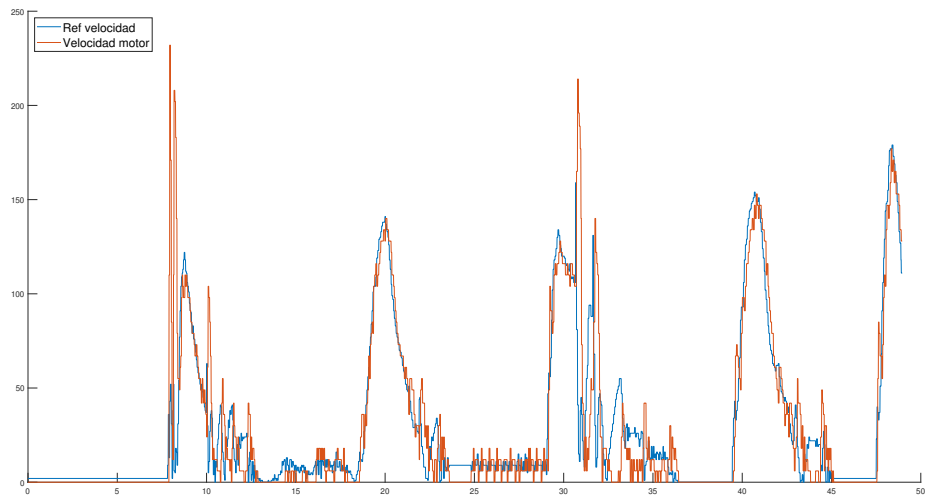


Figura 32: Resultado de la posición PID en cascada.

### 7.2.7. Controlador PID soportando torque

Se realizó una prueba de torque para el control PID utilizado, cuyo propósito fue el verificar que el control PID no es capaz de ejercer un torque que contrarreste la fuerza que se le aplica al motor. Como se observa en la Figura 33, la posición del motor al aplicarle un torque este se mueve fácilmente, ya que no ejerce un torque contrario para contrarrestar el ejercido. Al observar la Figura 34, la corriente posee un aumento significativo, lo cual es de esperarse dado que el motor, al querer provocar movimiento, va a demandar una mayor corriente.

Si bien el controlador PID es ideal para aplicaciones de robótica con aplicaciones cinemáticas, no es ideal para aplicaciones dinámicas que demanden el control de torque. En la sección de anexos se encuentra un enlace al vídeo donde se puede observar cómo el controlador PID no es suficiente para contrarrestar el torque.

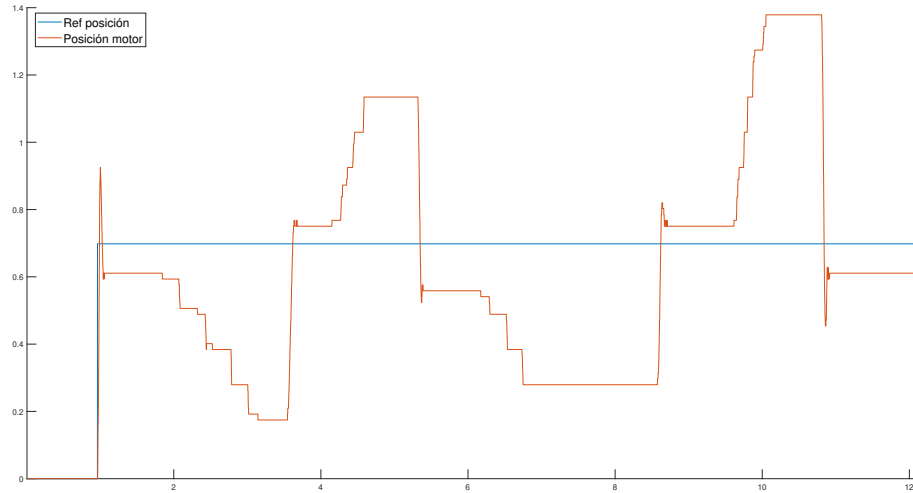


Figura 33: Posición del motor con controlador PID ejerciendo torque.

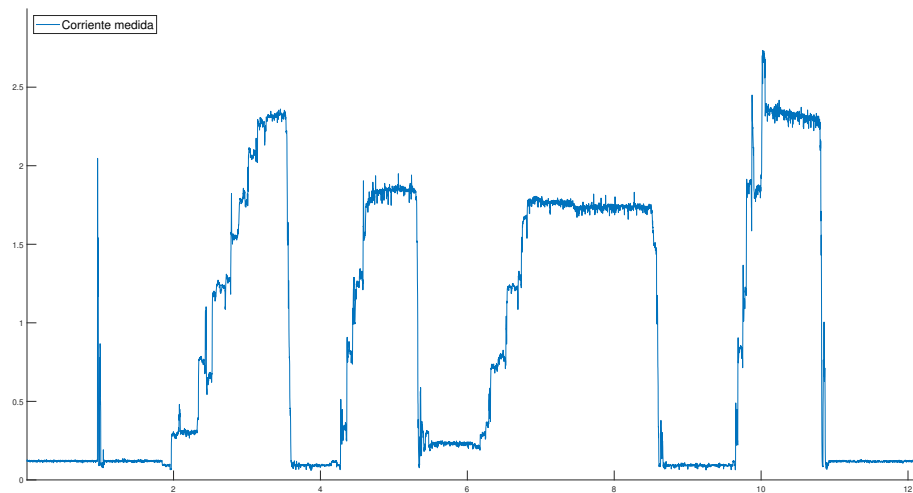


Figura 34: Corriente medida prueba controlador PID ejerciendo torque.

### 7.2.8. Primer planteamiento de restricciones adicionales

Para obtener las constantes del motor, se definieron los límites inferiores y superiores para las constantes. Los límites inferiores se definieron como cero, ya que los valores de estas

constantes no pueden ser negativas, mientras que los superiores variaron dependiendo de la constante en cuestión. Los límites se definieron según los valores por los que usualmente se encuentran. El único caso en el que se pudo obtener un valor fijo fue para la constante  $K_\tau$ , ya que esta se podía obtener de la gráfica del motor.

$$ub = \begin{bmatrix} R \\ L \\ K_\varepsilon \\ K_\tau \\ k \\ b \\ J \end{bmatrix} = \begin{bmatrix} 10 \\ 0.01 \\ \infty \\ 0.065 \\ \infty \\ 1 \\ 1 \end{bmatrix}$$

Al plantear el controlador LQI se obtuvo que el sistema era completamente controlable. Como se observa en la Figura 35, el controlador logra provocar que la posición (curva anaranjada) llegue a un punto de operación deseado, la velocidad (curva amarilla) aumenta y disminuye según el movimiento de la posición y el torque (curva azul) posee una variación y regresa a cero al momento de llegar a la posición final deseada. En este caso, este no es el comportamiento que se desea, ya que el torque debería de mantenerse en un valor de operación deseado, por lo que esta restricción adicional no es la adecuada ya que el comportamiento es el mismo que el del controlador PID en cascada.

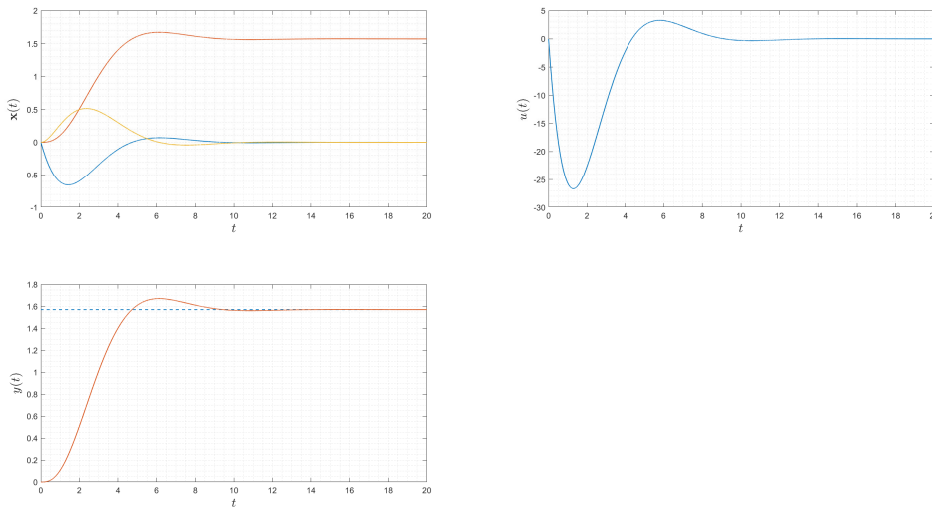


Figura 35: Respuesta controlador LQI con restricción del polinomio característico.

### 7.2.9. Segundo planteamiento de restricciones adicionales

Al igual que en el primer planteamiento de restricciones adicionales, se definieron los límites inferiores y superiores para realizar la optimización, los inferiores ya no fueron cero para la inercia y la fricción del motor (ya que estas no pueden ser cero), por lo que se

colocaron como límites inferiores:

$$lb = \begin{bmatrix} R \\ L \\ K_\varepsilon \\ K_\tau \\ k \\ b \\ J \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0.065 \\ 0 \\ 0 \\ 0.005 \end{bmatrix}$$

Se definieron como límites superiores los siguientes:

$$ub = \begin{bmatrix} R \\ L \\ K_\varepsilon \\ K_\tau \\ k \\ b \\ J \end{bmatrix} = \begin{bmatrix} 1000 \\ 0.07 \\ \infty \\ 0.065 \\ \infty \\ \infty \\ 0.007 \end{bmatrix}$$

Observando el comportamiento del controlador LQI de la Figura 36 pareciese que se obtuvo un resultado peor que el anterior, ya que el torque pareciese ser cero constantemente, sufriendo una variación muy pequeña. Esta se puede observar en la curva de la velocidad y el torque, ya que estas no están sobrepuestas, indicando que, a pesar de que el torque no es significativamente grande, se puede plantear el controlador dinámico capaz de controlar el torque y la posición de un motor DC con escobillas, lo cual valida el uso de un controlador LQI para este caso.

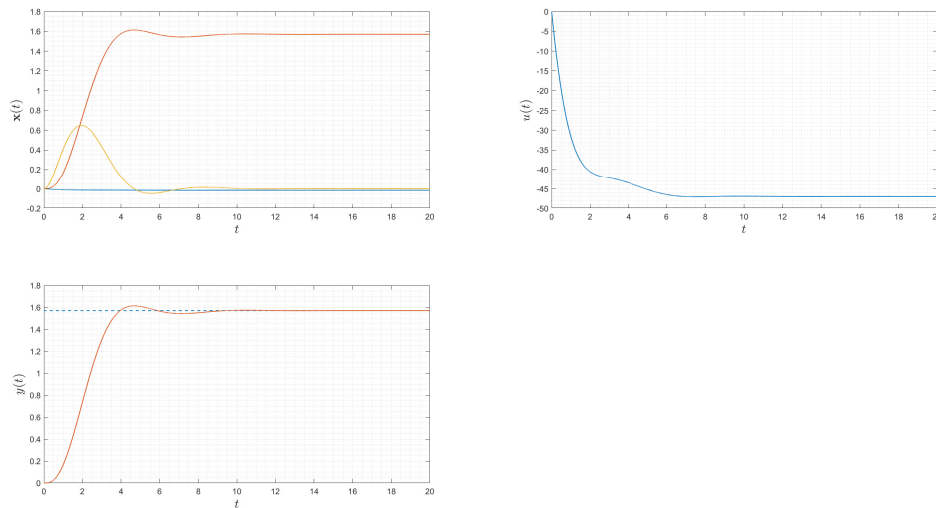


Figura 36: Respuesta controlador LQI con restricción de desigualdad.

### 7.2.10. Filtro IIR de primer orden de corriente

En la Figura 37 se muestra el resultado obtenido al aplicar el filtro IIR de primer orden a la lectura de la corriente con un factor de  $\lambda = 0.7$ .

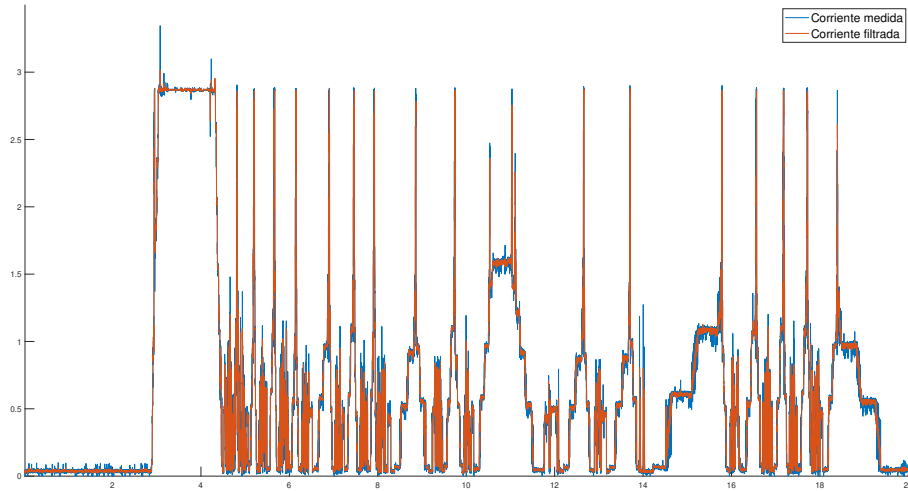


Figura 37: Efecto del filtro IIR de primer orden sobre la corriente.

Al observar este resultado, si bien parece que este filtro no funciona correctamente, se debe tomar en cuenta que el propósito de este filtro no es eliminar las perturbaciones repentinas de la señal de corriente en su totalidad, sino suavizar esta señal. Con el objetivo de que, si se producen perturbaciones demasiado grandes, estas aun puedan pasar por el filtro, pero si las perturbaciones son pequeñas estas sean suavizadas, se puede notar que este es el resultado obtenido.

Se calculó también un *group delay* para este filtro. Al tener un periodo de muestreo de  $1ms$ , se obtuvo un *group delay* de  $428.57\mu s$ , lo cual es un tiempo bastante pequeño, por lo que el retraso que provoca este filtro con respecto a la señal es muy pequeño y no interfiere al momento de efectuar el controlador.

### 7.2.11. Controlador LQI

Para realizar el control LQI se tuvieron que ajustar las constantes manualmente, ya que las obtenidas con la ayuda de *Matlab* provocaban que el sistema fuese inestable.

Para verificar el controlador LQI se realizaron dos pruebas distintas: la primera fue ejerciendo un toque que el motor debía ser capaz de mover y llegar a la posición deseada, mientras que la segunda prueba fue en la que no se ejerció un torque, por lo cual el motor debía oscilar intentando siempre llegar a la posición deseada.

## Control LQI aplicando torque al motor DC

El propósito de esta prueba fue evaluar que, al estar en la posición deseada, si el motor sufre una fuerza que deseara moverlo de su posición, este fuese capaz de mantener la posición al ejercer un torque contrario a la fuerza que desea modificar su posición.

En las Figuras 38 y 39 se muestra el resultado obtenido de la posición y la corriente del motor. En el caso de la posición, se observa que esta posee *overshoots* mientras intenta estabilizarse en un punto de operación. Esto se debe a que en ese momento, el torque que se le ejerció al motor no era suficiente para vencer la fuerza que el motor puede otorgar.

En el caso que se ejerce una fuerza para mover el motor, se puede observar que la corriente consumida por este inicia a aumentar considerablemente. Sin embargo, también se puede observar que, al momento de lograr vencer la fuerza contraria y superar el punto de operación, la corriente inicia a disminuir para que la misma fuerza provoque que el motor regrese al punto de operación deseado. En la sección de anexos se encuentra el enlace al vídeo del funcionamiento, donde se puede observar el funcionamiento correcto del motor y como este logra estabilizarse en la posición deseada.

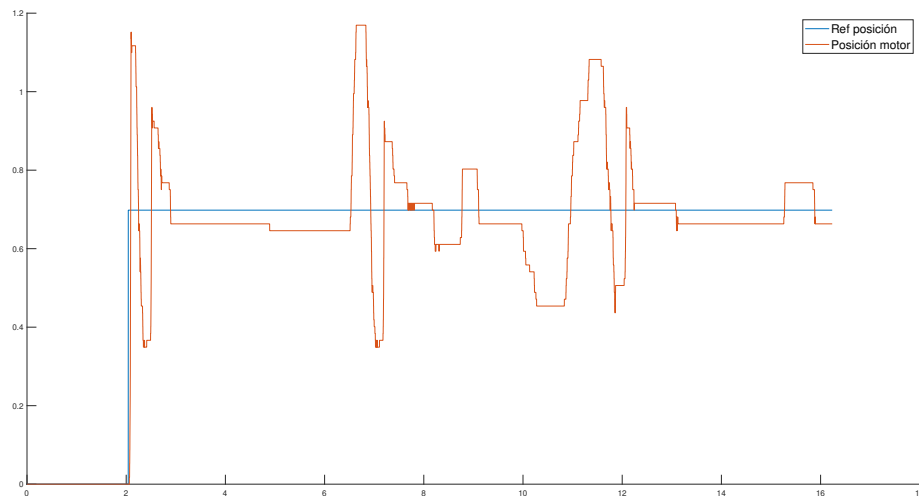


Figura 38: Posición del motor con controlador LQI ejerciendo torque.

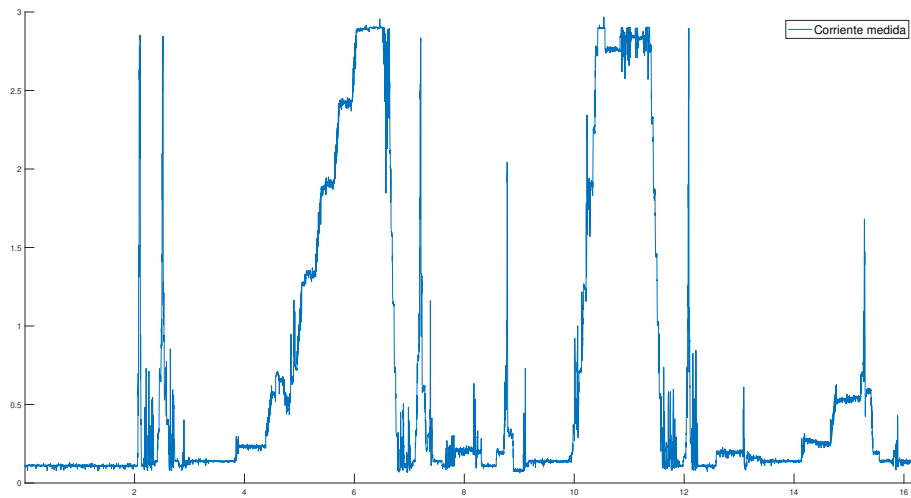


Figura 39: Corriente medida prueba controlador LQI ejerciendo torque.

Para lograr este comportamiento se utilizó el siguiente vector de constantes:

$$k_{lqi} = \begin{bmatrix} 10 \\ 0.5 \\ -0.8 \\ 50 \end{bmatrix}$$

### Control LQI sin aplicar torque al motor DC

El propósito de realizar una prueba sin carga fue el poder observar si el comportamiento del motor era el esperado o no. Al no tener torque, se esperaba que el motor presentase un *overshoot* continuo y que este fuese aumentando por la acumulación del error, siempre intentando regresar al punto de operación deseado. Como se observa en la Figura 40, la posición del motor presentó un *overshoot* creciente por el error acumulado, pero siempre retornaba al punto de operación, indicando que el controlador funciona correctamente. La prueba realizada se puede observar en el enlace que se encuentra en la sección de anexos.



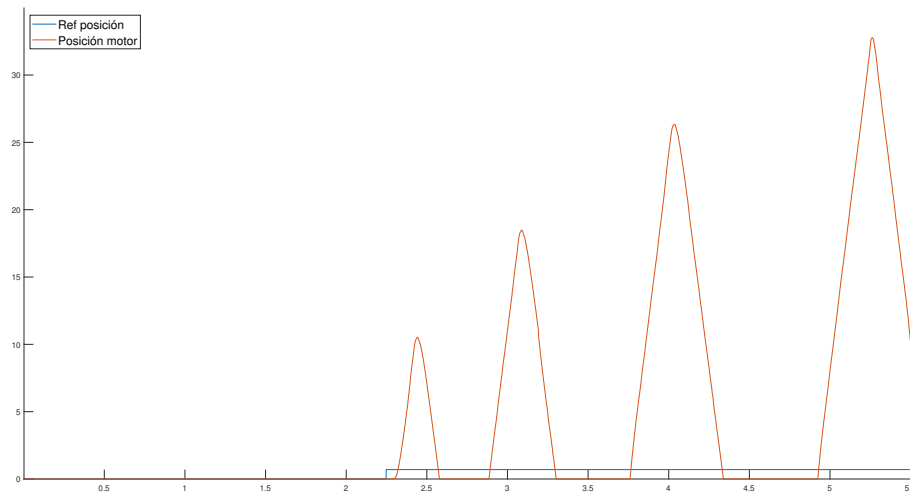


Figura 40: Posición del motor con controlador LQI sin ejercer torque.

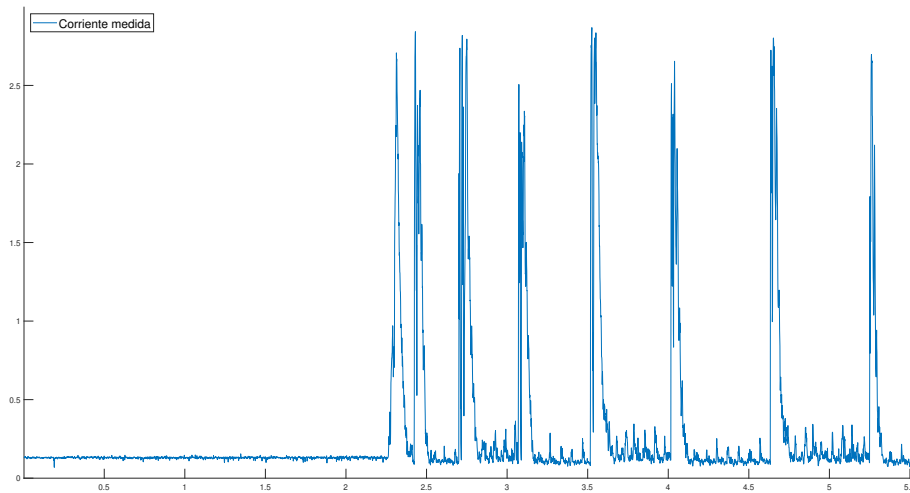


Figura 41: Corriente medida prueba controlador LQI sin ejercer torque.

### 7.2.12. Fabricación de placa PCB y carcasa para el actuador

En la Figura 42 se muestra la vista en 2D de la placa realizada. Esta mide  $28.381\text{mm}$ , largo y  $43.637\text{mm}$  de alto y contiene cuatro capas para realizar un diseño más compacto y poder facilitar las conexiones de los componentes. No se colocaron pines de entrada en cada uno de los agujeros, ya que la idea es soldar directamente a la placa los cables que se conectan a otras placas y los que se utilizan para la comunicación SPI.

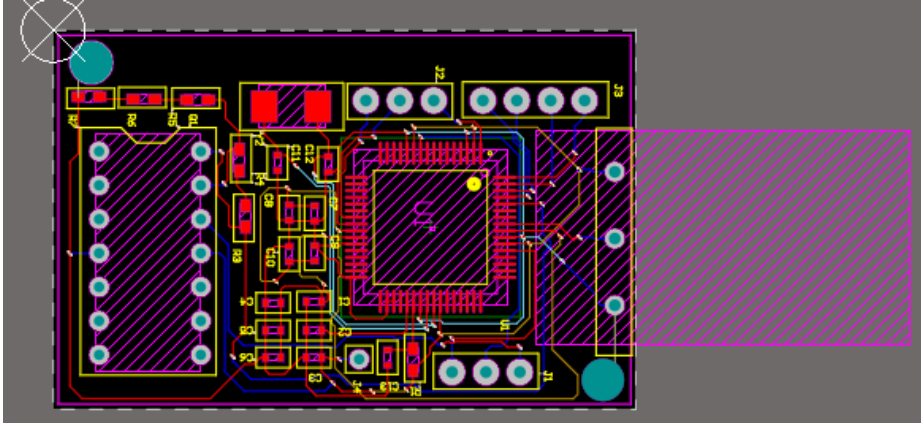


Figura 42: 2D placa PCB controlador.

En la Figura 43 se muestra como se vería la placa PCB con los componentes ya integrados.

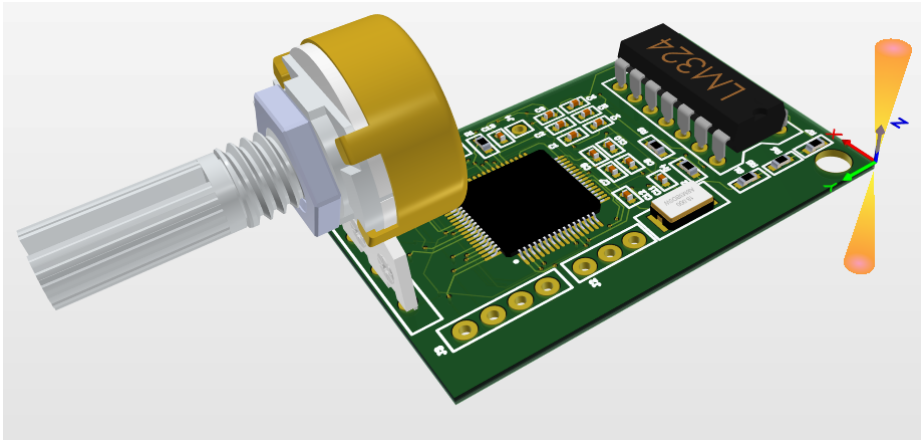


Figura 43: 3D placa PCB controlador.

En la Figura 44 se puede observar la propuesta de manufactura para el actuador. Esta cuenta con salida para comunicación SPI, potenciómetro para ajustar las constantes del controlador y agujeros para tornillos 3M.

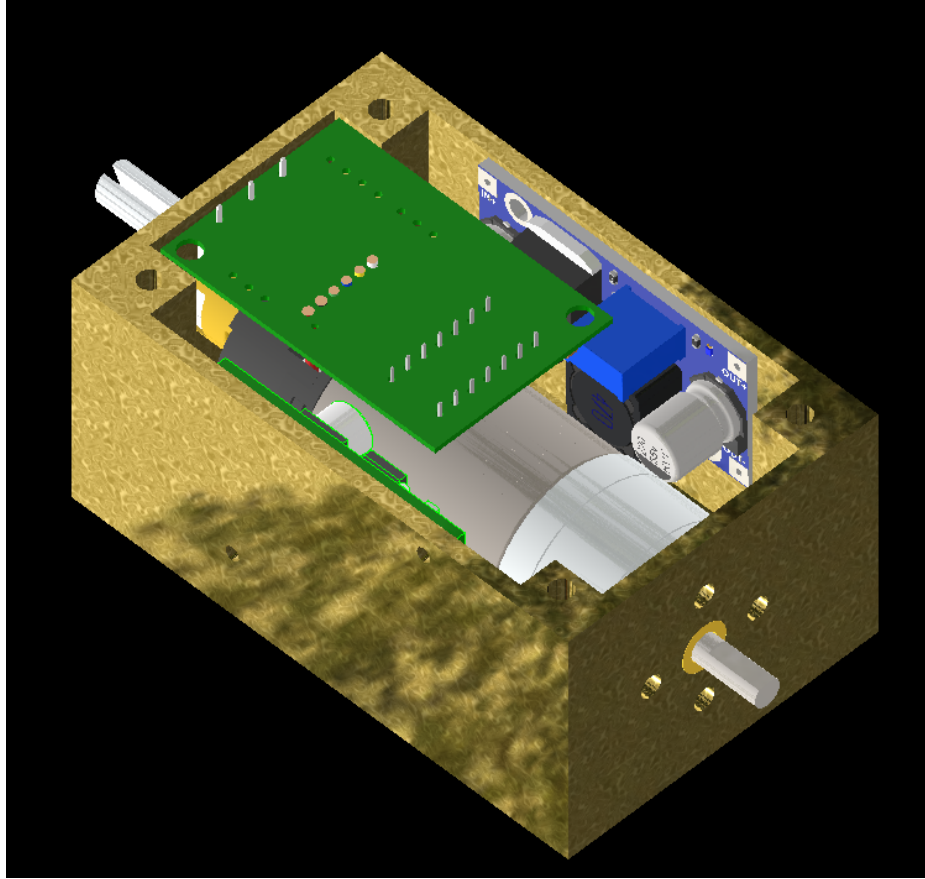


Figura 44: Propuesta carcasa actuador.

En la sección de anexos se encuentran los esquemáticos y planos de la propuesta de manufactura.



El principal objetivo de este proyecto era realizar y detallar un controlador que permitiese realizar un actuador altamente dinámico. Los controladores PID son muy útiles para poder controlar correctamente los motores DC. Siempre que se habla del control de un motor DC se suele hablar del control PID.

Por lo general, al realizar el control PID se toma en consideración que la entrada de la referencia será un valor estable y sin perturbaciones. En caso que la entrada del controlador no sea una señal digital, sino que una analógica, esta puede presentar un ruido que afecta la entrada del sistema. Existen métodos para eliminar este ruido con componentes pasivos o se puede optar por la opción, como lo fue en este caso, de realizar un filtro digital. Observando el resultado que se obtuvo del filtro IIR pasa bajas, este inyecta un *delay* en la señal de entrada del potenciómetro. El resultado obtenido fue bastante satisfactorio, ya que no presenta nada de ruido, la señal es estable, y el *delay* que presenta es de  $52.632\mu s$  lo cual es bastante pequeño para este caso. Por lo que el filtro fue útil para su aplicación y permitió que el sistema tuviese una entrada limpia sin ruido que lo perturbase.

La arquitectura más simple para realizar un sistema controlado por un controlador PID, es un controlador PID individual. Esto se realiza, por lo general, con el propósito de ajustar las constantes de los controladores por sí solos para que, al momento de unir estos, sea más sencillo el ajustes de las constantes. Al observar los resultados del controlador PID de velocidad, este posee una respuesta rápida y estable que permite el control de la misma. Es importante que no solo funcione sin cargas aplicada, sino que, al aplicarle una carga pequeña, este debe de forzar que el motor llegue a la velocidad de referencia.

El controlador PID de posición fue el más complicado de ajustar. Debido a que la posición debe ser capaz de cambiar el giro del motor. Esta fue la complicación más difícil de resolver, ya que no podía existir un *overshoot* en la salida del controlador. Al ajustar las constantes del controlador PID, en ocasiones, no se inyectaba suficiente voltaje al motor DC o en otras ocasiones se sobre inyectaba voltaje al motor DC. Observando los resultados de este controlador, no existe un *overshoot* significativo que pueda provocar que el motor realice cambios de giro bruscos que puedan dañar el motor.

Al comparar el comportamiento de los controladores programados con los resultados obtenidos en *Matlab*, se puede decir que los controladores programados funcionan mejor que los esperados, ya que el tiempo de subida es menor y el tiempo de asentamiento es más rápido. En el caso del PID de velocidad, el simulado en *Matlab* presenta un *overshoot*, el cual no existe en el PID programado. En el caso de la posición, el PID realizado con *Matlab* nunca llega a estabilizarse, a comparación del programado, que se estabiliza rápidamente.

Para optimizar los controladores realizados individualmente, se utilizó una configuración de controladores PID en cascada para que mejorara la posición con el control de la velocidad. Al observar los resultados obtenidos, la posición mejoró significativamente, ya que el motor DC sigue, la mayoría del tiempo, la señal de referencia de entrada. Al observar la velocidad, se puede ver que cumple con el funcionamiento esperado, ya que al agregar el término *feedforward* se eliminaron picos abruptos que podía llegar a tener la velocidad cuando el motor iniciaba el movimiento, lo que permitió así que el *overshoot* de la posición fuese eliminado.

Los problemas de control óptimo, tienden a ser sumamente complejos debido a la naturaleza de los sistemas. En estos problemas la mayor dificultad que se tiene son las decisiones de restricciones. Esto es debido a que se debe de realizar varias estimaciones de parámetros para la toma de decisión de las restricciones a colocar, ya que se conoce muy poco del sistema. En este caso la optimización no se logró realizar como se quería debido a las restricciones utilizadas, pero sin embargo al plantear la simulación del controlador LQI, se pudo demostrar que para el motor DC era posible el planteamiento de este. Esto ya que, al observar el resultado obtenido de la segunda propuesta de resultados, este validó que la posición logra llegar a una posición y quedarse en ella, la velocidad aumentó y disminuyó hasta cero y el torque no tuvo un gran cambio, pero este logró mantener un valor distinto en cero mientras la posición era estable.

Al momento de implementar el controlador LQI en el microcontrolador, pareció en un inicio que este no funcionaba como debía. Lo que se esperaba era que este se quedase en un punto fijo ejerciendo un torque. Pero al colocar una carga adecuada y observando los resultados obtenidos, se demostró que el controlador LQI cumplía su trabajo exitosamente, ya que otorgaba precisión en la posición y entregaba un torque que mantenía la carga en el lugar deseado. Este controlador funcionará correctamente, siempre y cuando se le coloque una carga proporcional al torque. Esto quiere decir que se debe de dimensionar correctamente no solo el controlador, si no que también el toque que puede entregar el motor.

Se deseaba proponer un diseño de manufactura el cual fuese un sistema embebido compacto del actuador desarrollado. En este caso se propuso un diseño suficientemente compacto para poder utilizar el actuador en distintas aplicaciones. La carcasa propuesta para el motor y los circuitos pareciese ser muy grande, pero en este caso lo que acapara la mayor parte de esta es el motor.

Este diseño no se manufacturó debido a diversas razones. La primera razón en este caso fue lo compacto que se deseaba el sistema. Al solo poseer acceso directo a manufacturar de placa PCB de solo dos capas, se debía contratar a una empresa que pusiese manufacturar mínimo 4 capas para realizar un sistema más compacto. La segunda razón fue que la carcasa no se podía realizar en impresión 3D. Esto es debido a que el motor está diseñado para ejercer cargas grandes y los detalles son tan pequeños que las herramientas que la Universidad

del Valle de Guatemala posee no satisfacen esta necesidad. Por último, está la limitante por la pandemia. Debido a esto existe escasez de componentes electrónicos como lo es el microcontrolador TM4C123G.





### **9.0.1. Recomendaciones para la realización de los controladores**

Durante la realización de este trabajo se tuvieron algunas complicaciones a la hora de implementar los controladores. Siempre que se realice un trabajo en el que se deban de realizar lecturas constantes de sensores es de suma importancia colocar un tiempo de muestreo. Esto es para evitar que el tiempo de lectura varíe y pueda perjudicar a los controladores. Si una de las referencias posee mucho ruido en ella, es útil realizar un filtrado de la señal para limpiar. Esto ya que si se requiere precisión, estas perturbaciones lo único que provocarán será ruido y que el sistema no se comporte como se desea. En el caso de los *encoders*, es muy útil que estos puedan aumentar su valor por cada revolución y también que puedan representar valores negativos. Si esto no es así, se debe realizar una programación que pueda calcular esto. Para el control del torque se recomiendo realizar un control de modelo predictivo para controlar mejor las variables de estado.

### **9.0.2. Recomendaciones para la parte física del sistema**

Una de las complicaciones más grandes que se tuvo, fue la realización del controlador LQI. Esta se vio afectada enormemente debido a la caja reductora del motor, ya que esta es 20.4:1, por lo que se recomienda altamente utilizar un motor con una caja reductora mucho más pequeña en la salida del motor. En el caso del sujetador del motor, se recomienda utilizar materiales que puedan soportar una gran cantidad de torque. Por último se recomienda utilizar un motor DC sin escobilla, ya que por lo regular se utilizan estos motores para aplicaciones dinámicas y con este se puede realizar el control de modelo predictivo.



- 
- [1] N. N. Farve, “Design of a Low-Mass High-Torque Brushless Motor for Application in Quadruped Robotics Professor of Electrical,” Massachusetts Institute of Technology, 2012.
  - [2] N. Gershenfeld, *Brushless Motors High Power Density with Control Overhead*, 2018. dirección: <https://fab.cba.mit.edu/classes/MAS.865/topics/motion/brushless/index.html>.
  - [3] J. L. K. Jr, “6.061 Class Notes, Chapter 12: Permanent Magnet "Brushless DC" Motors,” Massachusetts Institute of Technolog, 2003.
  - [4] W. D. Cuji y D. A. Avilés, “Control Directo al Par y Control de Campo Orientado para un Motor Síncrono de Imanes Permanentes,” *Ingeniería*, vol. 24, págs. 143-158, mayo de 2019.
  - [5] F. Ramírez-Leyva y E. Peralta-Sánchez, “Implementación del Control de Campo Orientado de un Motor de Imanes Permanentes con base en un Controlador Digital de Señales,” Universidad Popular Autónoma del Estado de Puebla, 2013.
  - [6] J. Pieper, *Walking in semi-rugged terrain | A Modicum of Fun*, oct. de 2020. dirección: <https://jpieper.com/2020/10/28/walking-in-semi-rugged-terrain/>.
  - [7] G. F. F. J. D. P. A. Emami-Naeini, *Feedback control of dynamic systems [Solutions manual]*, 6th, ép. Addison-Wesley series in electrical and computer engineering., Control engineering. Addison-Wesley, 1994.
  - [8] V. Mazzones, (*PDF*) *Controladores PID*, mar. de 2002.
  - [9] Z. Feng, R. Allen y J. Zhu, “Design of Continuous and Discrete LQI Control Systems with Stable Inner Loops,” Shanghai Jiaotong University, dic. de 2007, págs. 1-6.
  - [10] Z. Feng y R. Allen, “Design of continuous and discrete LQI control systems with stable inner loops,” *Journal of Shanghai Jiaotong University (Science)*, vol. 12, 6 dic. de 2007.
  - [11] Contaval, *Teoría encoders rotativos (vol#1)*, mar. de 2017. dirección: <https://www.contaval.es/teoria-encoders-rotativos-vol1/>.
  - [12] CONTAVAL, *Teoría encoders rotativos (vol#2)*, mar. de 2017. dirección: <https://www.contaval.es/teoria-encoders-rotativos-vol2/>.

- [13] W. A. López-Contreras, J. D. R. Antolines y H. Vacca, (PDF) *Diseño de un encoder magnético usando efecto Hall*, jul. de 2019. dirección: [https://www.researchgate.net/publication/343305136\\_Diseño\\_de\\_un\\_encoder\\_magnético\\_usando\\_efecto\\_Hall](https://www.researchgate.net/publication/343305136_Diseño_de_un_encoder_magnético_usando_efecto_Hall).
- [14] T. Nishad, (PDF) *Simple notes on Hall effect | NISHAD TN - Academia.edu*, 2019. dirección: [https://www.academia.edu/6923413/Simple\\_notes\\_on\\_Hall\\_effect](https://www.academia.edu/6923413/Simple_notes_on_Hall_effect).
- [15] M. D. Egerstedt, “Control of Mobile Robots,” Georgia Institute of Technology.
- [16] E. Ambitiously, *Encoder Measurements: How-To Guide - National Instruments*, sep. de 2020. dirección: <https://knowledge.ni.com/KnowledgeArticleDetails?id=kA03q000000x1riCAA&l=en-US>.
- [17] M. Bhardwaj y B. Akin, “Sensored Field Oriented Control of 3-Phase Induction Motors Bilal Akin C2000 Systems and Applications Team,” Texas Instruments, feb. de 2010, págs. 1-36.
- [18] “Clarke & Park Transforms on the TMS320C2xx Application Report Literature Number: BPRA048,” Texas Instruments Incorporated, 1997, págs. 1-46.
- [19] H. Fakham, M. Djemai y K. Busawon, (PDF) *Design and practical implementation of a back-EMF sliding-mode observer for a brushless DC motor*, 2008.
- [20] L. Mariano, S. Malagón, O. Andrés y A. León, “Introducción al Control Óptimo y aplicación del Principio de Pontryagin al comportamiento de un motor de Corriente Continua,” 2009.
- [21] A. V. 'Oppenheim y R. W. 'Schafer, *DISCRETE-TIME SIGNAL PROCESSIONG*, 3rd. PEARSON, 2010.
- [22] *Pololu - 20.4:1 Metal Gearmotor 25Dx65L mm HP 12V with 48 CPR Encoder*, 2021. dirección: <https://www.pololu.com/product/4843/pictures>.
- [23] *Pololu - VNH5019 Motor Driver Carrier*, 2021. dirección: <https://www.pololu.com/product/1451>.
- [24] Vishay, “Resistive Products,” Vishay intertechnology Inc., 2018. dirección: <http://www.vishay.com/doc?99914>.
- [25] OSHPARK, *OSH Park Docs Services 4 Layer Prototype Service*, 2017. dirección: <https://docs.oshpark.com/services/four-layer/>.
- [26] T. Instruments, “Tiva™ TM4C123GH6PM Microcontroller DATA SHEET TEXAS INSTRUMENTS-PRODUCTION DATA,” 2007. dirección: <http://www.ti.com/tm4c>.
- [27] —, “TivaWare™ Peripheral Driver Library,” 2013. dirección: [www.ti.com/tiva-c](http://www.ti.com/tiva-c).
- [28] Altium, *Altium Designer - PCB Design Software*, 2021. dirección: <https://www.altium.com/es/altium-designer/>.

[29] Autodesk, *Software Inventor | Consulta los precios y compra Inventor 2022 oficial*, 2021. dirección: [https://latinoamerica.autodesk.com/products/inventor/overview?mktvar002=4424517%7CSEM%7C11363024070%7C114341241594%7Ckwd-28960010&panel=buy&term=3-YEAR&ef\\_id=CjwKCAiA78aNBhAlEiwA7B76p\\_t3ihLu5txAsS-0GbgdIPL3CJ751oadugG0pwY8aSp\\_ljZ46QEL7hoCt3wQAvD\\_BwE:G:s&s\\_kwid=AL!11172!3!527555817297!e!!g!!autodesk%20inventor!11363024070!114341241594&mkwid=sfMXVyRAA%7Cpcrid%7C527555817297%7Cpkw%7Cautodesk%20inventor%7Cpmt%7Ce%7Cpdv%7Cc%7Cslid%7C%7Cpgrid%7C114341241594%7Cptaid%7Ckwd-28960010%7Cpid%7C&utm\\_medium=cpc&utm\\_source=google&utm\\_campaign=GGL\\_DEC\\_AutoCAD\\_AMER\\_MX\\_eComm\\_SEM\\_BR\\_New\\_EX\\_ADSK\\_3360535\\_&utm\\_term=autodesk%20inventor&utm\\_content=sfMXVyRAA%7Cpcrid%7C527555817297%7Cpkw%7Cautodesk%20inventor%7Cpmt%7Ce%7Cpdv%7Cc%7Cslid%7C%7Cpgrid%7C114341241594%7Cptaid%7Ckwd-28960010%7C&gclid=CjwKCAiA78aNBhAlEiwA7B76p\\_t3ihLu5txAsS-0GbgdIPL3CJ751oadugG0pwY8aSp\\_ljZ46QEL7hoCt3wQAvD\\_BwE](https://latinoamerica.autodesk.com/products/inventor/overview?mktvar002=4424517%7CSEM%7C11363024070%7C114341241594%7Ckwd-28960010&panel=buy&term=3-YEAR&ef_id=CjwKCAiA78aNBhAlEiwA7B76p_t3ihLu5txAsS-0GbgdIPL3CJ751oadugG0pwY8aSp_ljZ46QEL7hoCt3wQAvD_BwE:G:s&s_kwid=AL!11172!3!527555817297!e!!g!!autodesk%20inventor!11363024070!114341241594&mkwid=sfMXVyRAA%7Cpcrid%7C527555817297%7Cpkw%7Cautodesk%20inventor%7Cpmt%7Ce%7Cpdv%7Cc%7Cslid%7C%7Cpgrid%7C114341241594%7Cptaid%7Ckwd-28960010%7Cpid%7C&utm_medium=cpc&utm_source=google&utm_campaign=GGL_DEC_AutoCAD_AMER_MX_eComm_SEM_BR_New_EX_ADSK_3360535_&utm_term=autodesk%20inventor&utm_content=sfMXVyRAA%7Cpcrid%7C527555817297%7Cpkw%7Cautodesk%20inventor%7Cpmt%7Ce%7Cpdv%7Cc%7Cslid%7C%7Cpgrid%7C114341241594%7Cptaid%7Ckwd-28960010%7C&gclid=CjwKCAiA78aNBhAlEiwA7B76p_t3ihLu5txAsS-0GbgdIPL3CJ751oadugG0pwY8aSp_ljZ46QEL7hoCt3wQAvD_BwE).



## 11.1. Control tradicional PID

En el siguiente link, se puede encontrar el vídeo del funcionamiento de los 3 controladores PID realizados, control de posición, de velocidad, control en cascada y el peor de los caso si se ejerce un torque en la salida del motor. <https://youtu.be/Ci-5P6QPMmY>

## 11.2. Control moderno LQI

En el siguiente link, se puede encontrar un vídeo que muestra el funcionamiento del controlador LQI. Este muestra el comportamiento del controlador al ejercer una carga y al no ejercer una carga. <https://youtu.be/MiAXjcd90dY>

## 11.3. Esquemáticos y carcasa del actuador

En esta sección se muestra el esquemático del circuito realizado y los planos de la carcasa como propuesta de manufactura.

### 11.3.1. Esquemático circuito

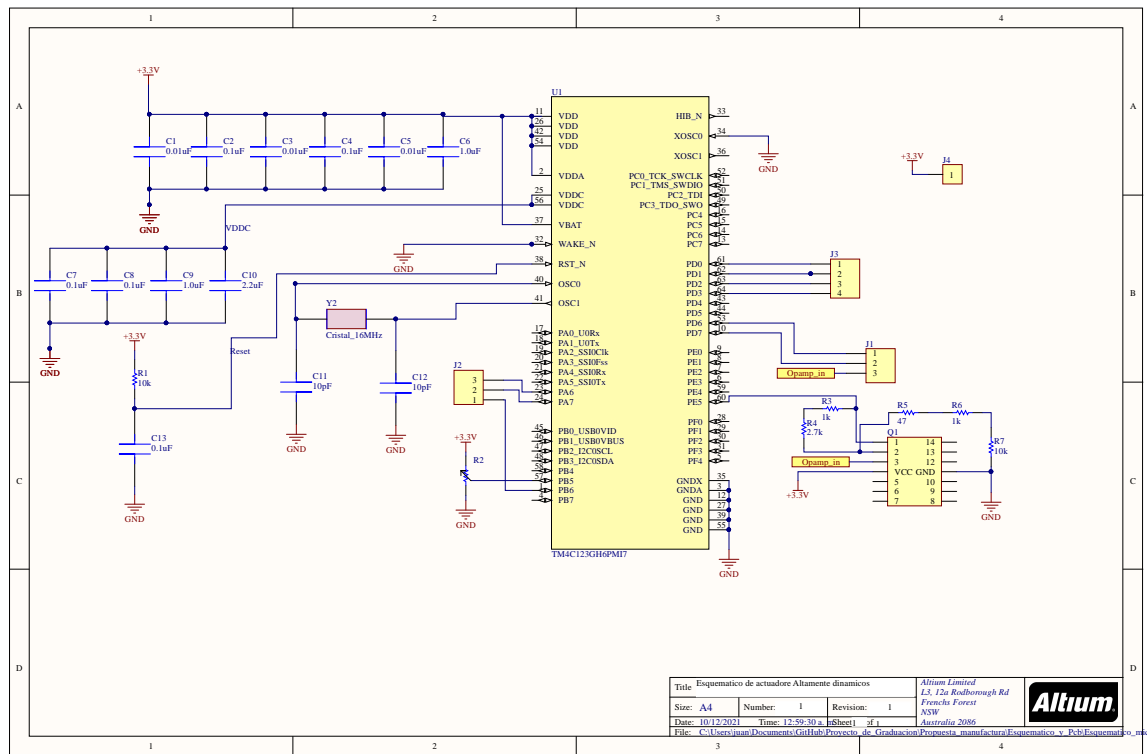


Figura 45: Esquemático circuito controlador.



### 11.3.2. Planos carcasa actuador

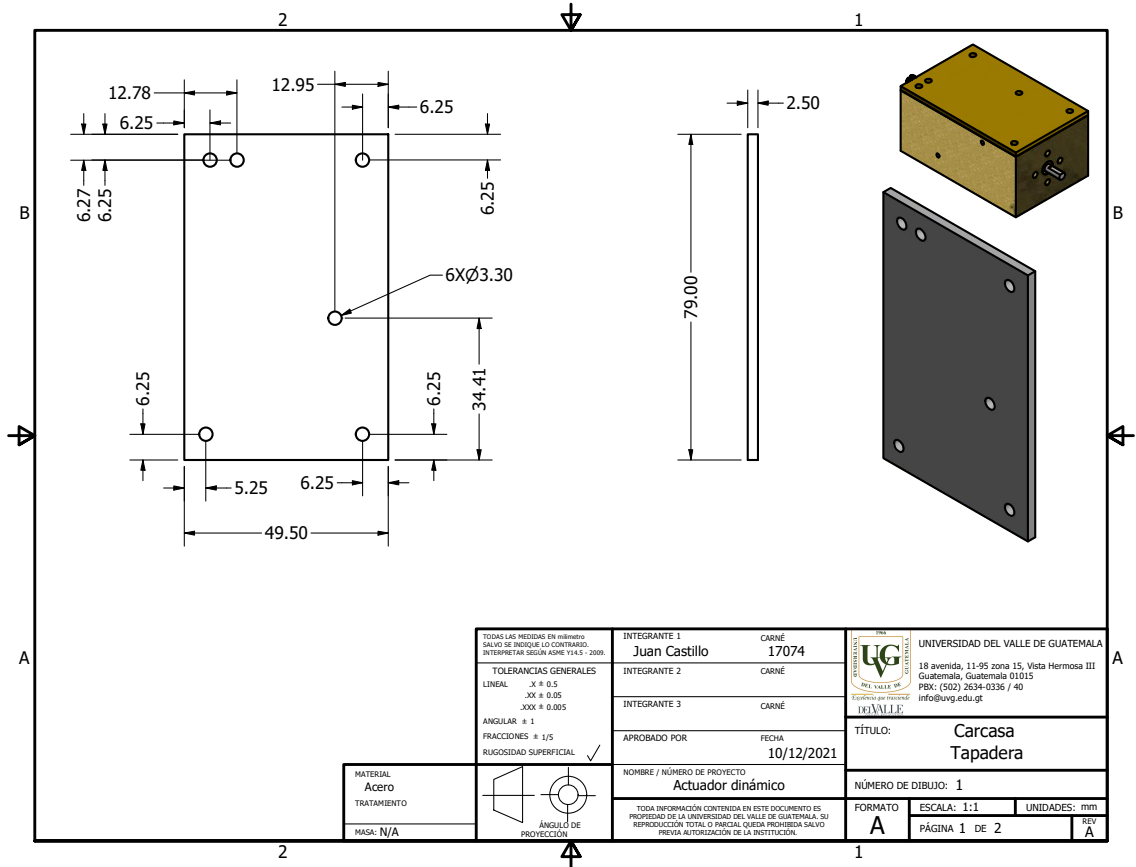


Figura 46: Planos 1 de carcasa para actuador.

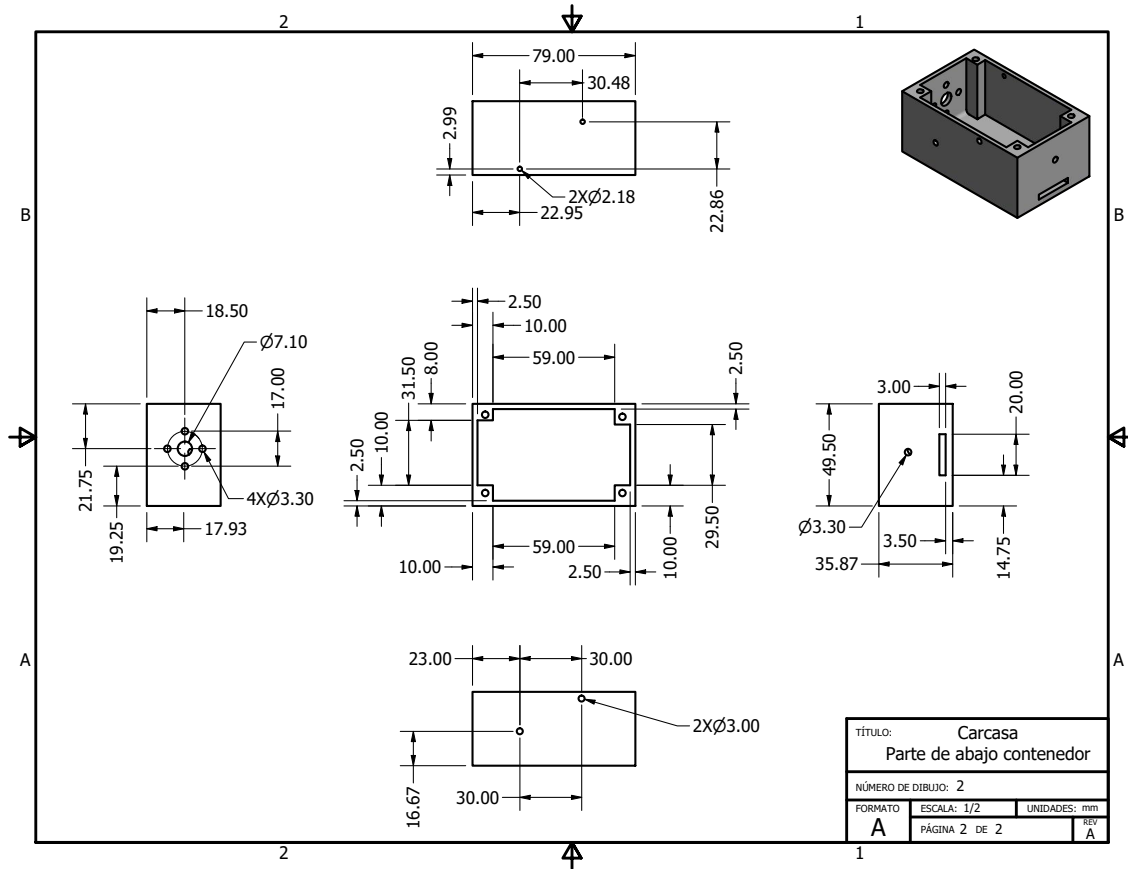


Figura 47: Planos 2 de carcasa para actuador.

**Actuadores:** Elementos que permite la conversión de un tipo de energía a otro tipo de energía. En este caso un motor DC. [3](#)

**Boston Dynamics:** Laboratorio de desarrollo en robótica. [4](#)

**Brushless:** Término utilizado para motores DC sin escobillas. [4](#)

**Code Composer Studio:** Es el software que se utilizó para la programación del microcontrolador. [30](#)

**Drivers:** Elementos electrónicos con circuitos internos para cumplir una función en específico. [3](#)

**Encoder:** Sensor utilizado para la toma de la posición y velocidad de un motor. [14](#)

**Field Oriented Control (FOC):** Controlador utilizado para realizar control en motores trifásicos. [4](#)

**Matlab:** Software utilizado para obtener las gráficas de velocidad y posición y la obtención de las funciones de transferencia. [1](#)

**Tiva C:** Microcontrolador utilizado para realizar el control del actuador. [29](#)