

UNIVERSIDAD DE CUENCA



FACULTAD DE INGENIERÍA

ESCUELA DE INGENIERÍA ELECTRÓNICA Y TELECOMUNICACIONES

“DISEÑO Y CONSTRUCCIÓN DE UNA INTERFAZ HÁPTICA DE UN  
GRADO DE LIBERTAD CON INTERFAZ HOMBRE-MÁQUINA”

UNIVERSIDAD DE CUENCA  
desde 1867

Tesis previa a la obtención del  
título de Ingeniero en Electrónica y Telecomunicaciones

Autor:

Oswaldo Felipe Guñansaca Soria

Director:

Ing. Luis Ismael Minchala Ávila, PhD.

Cuenca - Ecuador  
2016

---

## Resumen

---

**Palabras clave:** control, háptica, entornos, fuerza, sistemas, aplicaciones.

Las interfaces hápticas permiten la interacción *hombre-máquina* en entornos virtuales que pueden ser simulados o remotos. Los avances tecnológicos tanto de la electrónica como de herramientas informáticas han permitido un importante desarrollo de estos dispositivos. Este trabajo de investigación presenta resultados de la implementación de una interfaz háptica de un grado de libertad utilizando controladores clásicos, como el PID, y controladores modernos diseñados en espacio de estados que requieren la identificación paramétrica del modelo del sistema. También, se plantea el diseño de un controlador PID neuronal que sintoniza sus parámetros de acuerdo al punto de operación del sistema, como lo haría un control adaptativo. Finalmente, se proponen un par de aplicaciones: una en realidad virtual de un sistema dinámico, específicamente, el control de un péndulo invertido virtual, y un seguidor de movimiento que reproduce el movimiento de la articulación de una rodilla humana.

Oswaldo Guiñansaca Soria

---

## Abstract

---

**Keywords:** control, haptics, force, environments, systems, applications.

Haptic interfaces allow human-machine interaction in virtual environments that can be both simulated or remote. Technological advances, both in electronics and in informatic tools have allowed important developments of these devices. This research presents the results of the implementation of a haptic interface of one degree of freedom using classic controllers, such as PID, and modern controllers designed in state-space that require a parametric identification of the system. The design of a neural PID controller, which tunes their parameters depending on the operating point of the system as it would do an adaptive control, is also proposed. Finally, a pair of applications are presented: one in virtual reality of a dynamic system, which simulates a virtual pendulum as well as its controller and its closed-loop behavior; and a motion follower that reproduces the joint movement of a human knee.

Oswaldo Guiñansaca Soria

---

# Índice general

---

Resumen . . . . .	2
Abstract . . . . .	3
Índice general . . . . .	4
Índice de figuras . . . . .	7
Índice de tablas . . . . .	11
<b>1. Introducción</b>	<b>17</b>
1.1. Antecedentes . . . . .	17
1.2. Objetivos . . . . .	18
1.2.1. Objetivo general . . . . .	18
1.2.2. Objetivos específicos . . . . .	18
1.3. Descripción del problema . . . . .	18
1.4. Estado del arte . . . . .	19
1.5. Contribuciones de la tesis . . . . .	20
<b>2. Fundamentos teóricos</b>	<b>21</b>
2.1. Interfaces hápticas . . . . .	21
2.1.1. Impedancia vs admitancia . . . . .	22
2.1.2. Características generales . . . . .	23
2.2. Ambientes virtuales multimodales . . . . .	25
2.2.1. Influencia del sentido de la visión . . . . .	26
2.2.2. Ambientes virtuales simulados y remotos . . . . .	27
2.3. Tipos de interfaces . . . . .	27
2.4. Evaluación de las interfaces hápticas . . . . .	28
2.5. Definiciones en control clásico . . . . .	30
2.5.1. Sintonización del controlador PID . . . . .	33
2.6. Definiciones en control moderno . . . . .	35
2.6.1. Representación de sistemas en tiempo discreto en espacio de estados . . . . .	38



2.6.2.	Controlabilidad . . . . .	40
2.6.3.	Observabilidad . . . . .	41
2.6.4.	Realimentación de estados . . . . .	43
2.6.5.	Fórmula de Ackermann . . . . .	44
2.6.6.	Observadores de estados . . . . .	47
2.6.7.	Sistemas de seguimiento . . . . .	50
2.7.	Control PID neuronal . . . . .	54
2.7.1.	ADALINE . . . . .	55
2.7.2.	Controlador utilizando la red neuronal ADALINE	56
<b>3.</b>	<b>Construcción de una pala háptica</b>	<b>58</b>
3.1.	Interfaz háptica, componentes y montaje . . . . .	58
3.1.1.	Arduino . . . . .	58
3.1.2.	Ardumoto Shield . . . . .	59
3.1.3.	Sensor Magnetoresistivo KMA-210 . . . . .	61
3.1.4.	Actuador . . . . .	62
3.1.5.	Módulos de la interfaz háptica . . . . .	62
3.1.6.	Montaje . . . . .	65
<b>4.</b>	<b>Control de la interfaz háptica</b>	<b>70</b>
4.1.	Procesamiento de la señal de posición de la pala háptica	70
4.1.1.	Factor de conversión cinemático, <i>bits-posición</i> .	71
4.1.2.	Factor de conversión <i>fuerza-torque</i> . . . . .	73
4.2.	Determinación de parámetros del sistema . . . . .	74
4.2.1.	Constante de amortiguamiento del motor . . . . .	74
4.2.2.	Torque del motor . . . . .	81
4.2.3.	Fricción del motor . . . . .	83
4.2.4.	Inercia de la pala háptica . . . . .	84
4.3.	Control del interfaz háptico mediante PID . . . . .	87
4.3.1.	Control retroalimentado . . . . .	87
4.3.2.	Sintonización del PID . . . . .	88
4.4.	Control del interfaz háptico utilizando técnicas de control moderno . . . . .	90
4.4.1.	Identificación del sistema . . . . .	90
4.4.2.	Identificación de la planta para diseño de controladores . . . . .	90
4.4.3.	Controlabilidad y observabilidad del sistema . .	96



4.4.4.	Control por reubicación de polos . . . . .	96
4.4.5.	Seguidor de referencias constantes . . . . .	99
4.4.6.	Sistema de seguimiento con integrador y reali- mentación de estado observado . . . . .	100
4.5.	Control PID neuronal del interfaz háptico . . . . .	103
<b>5.</b>	<b>Aplicaciones con la interfaz háptica</b>	<b>107</b>
5.1.	Péndulo invertido virtual . . . . .	107
5.1.1.	Control del péndulo invertido en lazo abierto .	109
5.1.2.	Control del péndulo invertido en lazo cerrado .	111
5.2.	Seguidor del movimiento de una rodilla humana . . . .	112
5.3.	Evaluación de la interfaz háptica . . . . .	115
<b>6.</b>	<b>Conclusiones</b>	<b>117</b>
	<b>Anexos</b>	<b>119</b>
<b>A.</b>	<b>Hoja de especificaciones del driver L298</b>	<b>120</b>
<b>B.</b>	<b>Hoja de especificaciones del motor Maxon 353144</b>	<b>124</b>
<b>C.</b>	<b>Raíces de los polinomios normalizados de Bessel</b>	<b>126</b>
	<b>Bibliografía</b>	<b>129</b>

---

## Índice de figuras

---

2.1.	Diferencia entre las interfaces hápticas (der.) y dispositivos unidireccionales (izq.) . . . . .	22
2.2.	Dispositivo tipo Impedancia . . . . .	23
2.3.	Dispositivo tipo Admitancia . . . . .	23
2.4.	Influencia del sentido de la visión en los ambientes virtuales . . . . .	26
2.5.	Clasificación de las técnicas de evaluación de las interfaces hápticas . . . . .	29
2.6.	Función de transferencia . . . . .	30
2.7.	Diagrama de bloques de un sistema en lazo cerrado . . . . .	31
2.8.	Diagrama de bloques de un sistema en lazo abierto . . . . .	31
2.9.	Diagrama de bloques del PID . . . . .	32
2.10.	Curva experimental para determinar los parámetros del PID . . . . .	34
2.11.	Determinación de $P_u$ . . . . .	35
2.12.	Sistema de control en tiempo discreto, lineal e invariante en el tiempo . . . . .	38
2.13.	Sistema de control en lazo abierto . . . . .	44
2.14.	Sistema de control con realimentación de estado . . . . .	44
2.15.	Observador de estado . . . . .	47
2.16.	Sistema de regulación con un observador de estados . . . . .	48
2.17.	Sistema de control con realimentación de estado observado . . . . .	49
2.18.	Sistema de seguimiento con integrador . . . . .	51
3.1.	Plataforma Arduino . . . . .	59
3.2.	Ardumoto Shield . . . . .	59
3.3.	Distribución de pines del Ardumoto Shield . . . . .	60



3.4.	Ángulo medido por el sensor magnetoresistivo KMA-210	62
3.5.	Motor utilizado en la interfaz háptica . . . . .	62
3.6.	Base de la interfaz háptica . . . . .	63
3.7.	Módulo frontal de la interfaz háptica . . . . .	63
3.8.	Soportes para la estabilización del módulo frontal sobre la base de la interfaz háptica . . . . .	64
3.9.	Soporte para sujetar la pala háptica . . . . .	64
3.10.	Rueda motriz de la interfaz háptica . . . . .	65
3.11.	Pala de la interfaz háptica . . . . .	65
3.12.	Montaje interfaz háptica, pasos 1 y 2 . . . . .	66
3.13.	Montaje interfaz háptica, paso 3 . . . . .	66
3.14.	Montaje interfaz háptica, paso 4 . . . . .	67
3.15.	Montaje interfaz háptica, paso 5 . . . . .	67
3.16.	Montaje interfaz háptica, paso 6 . . . . .	68
3.17.	Montaje interfaz háptica, paso 7 . . . . .	68
3.18.	Montaje interfaz háptica, paso 8 . . . . .	69
3.19.	Montaje interfaz háptica, paso 9 . . . . .	69
4.1.	Señal a la salida del sensor magnetoresistivo . . . . .	70
4.2.	Diagrama de flujo utilizado en Arduino . . . . .	71
4.3.	Parámetros de la pala háptica y la rueda motriz . . . . .	71
4.4.	Diagrama de bloques, factor de conversión <i>cinemático</i>	72
4.5.	Diagrama de bloques, factor de conversión <i>fuerza-torque</i>	74
4.6.	Parámetros de la rueda motriz de la interfaz háptica .	75
4.7.	Conexión utilizada para determinar la constante de amortiguamiento . . . . .	77
4.8.	Respuesta de la fuerza electromotriz del motor . . . . .	77
4.9.	Datos válidos de la fuerza electromotriz del motor . . .	78
4.10.	Diagrama de bloques del modelo del motor - Simulink	79
4.11.	Aproximación del modelo teórico y real mediante una función de optimización . . . . .	80
4.12.	Señal obtenida en Simulink del modelo del motor . . .	80
4.13.	Experimento para determinar el torque del motor . . .	82
4.14.	Gráfico del torque en función de la corriente para de- terminar la constante $k_m$ . . . . .	83
4.15.	Conexión utilizada para determinar la fricción del motor	84



4.16. Montaje utilizado para determinar la frecuencia natural de la pala háptica, paso 1 . . . . .	85
4.17. Montaje utilizado para determinar la frecuencia natural de la pala háptica, paso 2 . . . . .	86
4.18. Control clásico del interfaz háptico . . . . .	87
4.19. Controlador PID para el interfaz háptico - Simulink . . . . .	87
4.20. Respuesta del controlador PID para una entrada escalón a) positivo y b) negativo . . . . .	88
4.21. Respuesta del controlador PID para una entrada escalón a) positivo y b) negativo . . . . .	89
4.22. Respuesta del controlador PID para una entrada escalón a) positivo y b) negativo . . . . .	89
4.23. Diagrama de bloques para la identificación de los sistemas dinámicos . . . . .	90
4.24. Diagrama de bloques para la identificación del sistema - Simulink . . . . .	91
4.25. Señal PRBS . . . . .	92
4.26. Respuesta de la planta . . . . .	92
4.27. Señal PRBS vs salida de la planta . . . . .	93
4.28. Modelos estimados vs varianza de salida . . . . .	94
4.29. Salida medida vs salida simulada de los modelos que mejor rendimiento presentan . . . . .	94
4.30. Señal de salida de la planta vs aproximación $arx232$ . . . . .	95
4.31. Control con estado observado para la interfaz háptica - Simulink . . . . .	97
4.32. Respuesta de la interfaz háptica con regulador implementado en Simulink . . . . .	98
4.33. Sistema de seguimiento de referencias constantes para el interfaz háptico en Simulink . . . . .	99
4.34. Respuesta de la interfaz háptica ante un escalón . . . . .	100
4.35. Respuesta de la interfaz háptica ante dos cambios de referencia . . . . .	100
4.36. Seguimiento con integrador y estado observado para la interfaz háptica - Simulink . . . . .	101
4.37. Respuesta del seguidor con integrador y estado observado ante una referencia cero - Simulink . . . . .	102



4.38. Respuesta del seguidor con integrador y estado observado ante una entrada escalón - Simulink . . . . .	103
4.39. Controlador PID neuronal del interfaz háptico - Simulink	103
4.40. Respuesta del controlador PID neuronal del interfaz háptico - Simulink . . . . .	105
4.41. Respuesta del controlador con redes neuronales ante una referencia escalón a) positiva y b) negativa . . . . .	106
5.1. Base del péndulo invertido virtual - Solidworks . . . . .	107
5.2. Péndulo virtual - Solidworks . . . . .	108
5.3. Ensamble final del péndulo invertido virtual - Solidworks	108
5.4. Diagrama de bloques importado a Simulink desde Solidworks . . . . .	109
5.5. Interfaz gráfica en Simmechanics del péndulo virtual .	109
5.6. Control en lazo abierto del péndulo invertido virtual - Simulink . . . . .	110
5.7. Interfaz de las propiedades del bloque <i>Revolute</i> - Simulink	110
5.8. Control en lazo cerrado del péndulo invertido virtual - Simulink . . . . .	111
5.9. Respuesta del controlador de la interfaz háptica en lazo cerrado para la aplicación del péndulo invertido - Simulink . . . . .	112
5.10. Respuesta del controlador del péndulo invertido virtual en lazo cerrado - Simulink . . . . .	112
5.11. Interfaz del programa Kinovea . . . . .	113
5.12. Ángulo medido en la recolección de datos . . . . .	113
5.13. Señal obtenida de los datos del movimiento de la rodilla, utilizada como referencia en el seguidor . . . . .	114
5.14. Diagrama de bloques para el seguidor de movimiento de la rodilla - Simulink . . . . .	114
5.15. Respuesta del seguidor ante el movimiento de la rodilla	115

---

## Índice de tablas

---

2.1.	VARIABLES PARA LA SINTONIZACIÓN, PRIMER MÉTODO . . . . .	34
2.2.	VARIABLES PARA LA SINTONIZACIÓN, SEGUNDO MÉTODO . . . . .	35
3.1.	DISTRIBUCIÓN DE PINES DE CONTROL DEL ARDUMOTO SHIELD .	61
3.2.	DISTRIBUCIÓN DE PINES DEL SENSOR MAGENTORESISTIVO KMA- 210 . . . . .	61
4.1.	DATOS DE CORRIENTE Y TORQUE . . . . .	82
4.2.	MEDIDAS DE CORRIENTE PARA ENCONTRAR LA FRICCIÓN DEL MOTOR	84
4.3.	PARÁMETROS ÚTILES PARA ENCONTRAR LA INERCIA DE LA PALA .	86
5.1.	VALORES DEL PID PARA EL CONTROL EN LAZO ABIERTO DEL PÉN- DULO INVERTIDO VIRTUAL . . . . .	111
5.2.	VALORES DEL PID PARA EL CONTROL EN LAZO CERRADO DEL PÉN- DULO INVERTIDO VIRTUAL . . . . .	112
5.3.	VALORES DEL PID PARA EL SEGUIDOR DE MOVIMIENTO DE LA RODILLA . . . . .	115
C.1.	RAÍCES DE LOS POLINOMIOS NORMALIZADOS DE BESSEL . . . . .	127



Yo, *Oswaldo Felipe Guiñansaca Soria*, autor de la tesis "Diseño y construcción de una interfaz háptica de un grado de libertad con interfaz hombre-máquina", reconozco y acepto el derecho de la Universidad de Cuenca, en base al Art. 5 literal c) de su Reglamento de Propiedad Intelectual, de publicar este trabajo por cualquier medio conocido o por conocer, al ser este requisito para la obtención de mi título de Ingeniero en Electrónica y Telecomunicaciones. El uso que la Universidad de Cuenca hiciere de este trabajo, no implicará afección alguna de mis derechos morales o patrimoniales como autor.

Cuenca, 10 de Julio de 2016.

A handwritten signature in blue ink, which appears to read "Oswaldo Felipe Guiñansaca Soria", written over a horizontal line.

Oswaldo Felipe Guiñansaca Soria

C.I: 0104457387



Yo, *Oswaldo Felipe Guiñansaca Soria*, autor de la tesis "Diseño y construcción de una interfaz háptica de un grado de libertad con interfaz hombre-máquina", certifico que todas las ideas, opiniones y contenidos expuestos en la presente investigación son de exclusiva responsabilidad de su autor.

Cuenca, 10 de Julio de 2016.

A handwritten signature in blue ink, reading "Oswaldo Felipe Guiñansaca Soria", written over a horizontal line.

Oswaldo Felipe Guiñansaca Soria

C.I: 0104457387

## CERTIFICO

Que el presente proyecto de tesis: "*Diseño y construcción de una interfaz háptica de un grado de libertad con interfaz hombre-máquina*" fue dirigido por mi persona.

---

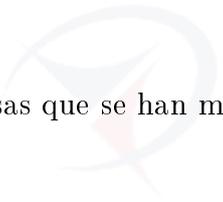
**Ing. Ismael Minchala Ávila, PhD.**  
**Director de Tesis**

---

## Agradecimiento

---

A todos los seres y cosas que se han movido y han aportado para que haya  
podido llegar a este momento.



UNIVERSIDAD DE CUENCA  
desde 1867

---

## Dedicatoria

---

A mi madre y a mi padre, María y Humberto, por todo su amor y ser la llave de  
mi existencia.

A Thalía, por ser y estar... que sonría siempre.

UNIVERSIDAD DE CUENCA  
desde 1867

Felipe

# Capítulo 1

---

## Introducción

---

### 1.1. Antecedentes

El cuerpo humano puede desenvolverse en cualquier entorno y adaptarse al mismo debido a la información sensorial que es capaz de reunir y procesar. La obtención de información -percepción sensorial- la realiza mediante sus cinco sentidos: visión, audición, tacto, gusto y olfato. En el cerebro se procesan estas señales transportadas hasta ahí por el sistema nervioso.

El sentido del tacto se encuentra localizado por todo el cuerpo humano, ya que diferentes órganos contienen células dedicadas a este tipo de percepción sensorial, pero el más grande es la piel, que aproximadamente tiene una superficie de  $1,8m^2$ , constituyendo el 16-18 % del peso del cuerpo humano [1]. La piel, además, funciona como una interfaz entre el ambiente externo y el organismo interno. A menudo se pasa por alto que el sentido del tacto también es un canal de comunicación, así como lo son la visión y audición. La háptica -normalmente considerada como el sentido unitario del tacto [1]-, en realidad, se compone de un número de diferentes clases de receptores sensoriales que responden al tacto, la presión, la temperatura, el dolor, la posición de las articulaciones y movimiento, siendo el cerebro el que diferencia todos estos estados y los une automáticamente en un único sentido. La háptica también está relacionada con la kinestesia, que estudia los movimientos del cuerpo.

Un dispositivo háptico, entonces, es un sistema, tanto de hardware y software, que permite la interacción con ambientes que pueden ser remotos o simulados, posibilitando, así, tener una experiencia sensorial háptica en dichos ambientes [2], [3]. Permite una retroalimentación activa, real e instantánea hacia el usuario mediante actuadores. Las interfaces hápticas brindan al usuario mayor información del entorno con el que se encuentra interactuando, complementándose con la



información visual y auditiva que se pueda tener del mismo. El auge y desarrollo de estas interfaces tienen diferentes campos de estudio e investigación como son la robótica y el control [4]; y han permitido logros como la telecirugía y demás aplicaciones.

Existe una gran variedad de dispositivos con interfaces hápticas, cada uno de ellos, cuenta con características específicas que mejoran su desempeño en el entorno donde se los vaya a utilizar. En esta tesis se realiza la construcción de una pala háptica *-joystick-* de un grado de libertad y se proponen diferentes tipos de control para la misma; posteriormente, se realizan aplicaciones prácticas para el interfaz háptico.

## 1.2. Objetivos

### 1.2.1. Objetivo general

Diseñar y construir un interfaz háptico de un grado de libertad, que permita interacción con el usuario a través de un interfaz hombre-máquina.

### 1.2.2. Objetivos específicos

- Identificar plenamente todos los conceptos relacionados con las interfaces hápticas.
- Construir una pala háptica.
- Generar diversos modelos para el control de la pala háptica.
- Validar, probar y mejorar los modelos planteados.
- Proponer diferentes aplicaciones a partir de la investigación actual.

## 1.3. Descripción del problema

Las comunicaciones han evolucionado a lo largo de la historia, desde los telegramas hasta las videoconferencias. Actualmente, las interfaces hápticas brindan una experiencia más interactiva a la persona que lo utiliza dentro de la comunicación, acoplándose con otros tipos de respuestas que dependen del entorno de



trabajo. Por tanto, estos dispositivos pueden estar ligados a diferentes profesiones que necesitan nuevas formas de comunicación –interacción y exploración-, o complementarse con las mismas.

Debido a la importancia que posee este campo de investigación y desarrollo, y el hecho de que las interfaces hápticas no han tenido un espacio dentro de los trabajos de investigación planteados en la Universidad de Cuenca, en este trabajo de tesis se plantea la construcción de una interfaz háptica, con objeto de abrir nuevos campos de investigación enfocados particularmente a la ingeniería de control.

## 1.4. Estado del arte

Los dispositivos hápticos se han desarrollado desde la década de los noventa, aunque, en años anteriores a esa década se presentó el primer avance en la simulación de un ambiente visual-háptico para edificaciones virtuales [5]. Desde entonces, esta tecnología ofrece una mejor interacción e inmersión con el usuario, presentando mayor cantidad de información de los ambientes de trabajo –remotos o virtuales- como se detalla en [6]. Las aplicaciones son diversas y ganan terreno en diferentes campos profesionales como la medicina, pasando por las ciencias e ingeniería, hasta el entretenimiento, tal como se menciona en [4].

En [3] se indican diferentes tipos de dispositivos hápticos, como guantes, plataformas de movimiento, exoesqueletos, *joysticks*, entre otros. Diferentes atributos o características deben ser considerados para el diseño de las interfaces hápticas. En [2] se detallan algunos parámetros que deben cumplir los dispositivos, los cuales tienen que ver con características físicas y también la experiencia que brinda al usuario, dicha experiencia es uno de los parámetros más importantes al momento de evaluar la calidad de las interfaces.

Todos estos dispositivos se rigen a diferentes modelos matemáticos, propios de cada planta, donde sus ecuaciones dinámicas involucran movimientos y fuerzas necesarias para la comunicación, que se realiza gracias a los sensores y actuadores involucrados. De la calidad y rendimiento de los sensores y actuadores, así como del número que se utilizan en la interfaz, dependerá también el rendimiento y eficiencia de la planta, como se explica en [7].

Diferentes dispositivos y aplicaciones se han sumado a esta nueva forma de comunicación e interacción, que se complementa con otras respuestas –como las audiovisuales- de los sistemas en estudio, mejorando la experiencia de comunica-



ción del usuario con los ambientes de trabajo.

## 1.5. Contribuciones de la tesis

Este trabajo de tesis propone un nuevo campo de investigación para los docentes y alumnos de la Universidad de Cuenca en el que se podrán aplicar conceptos teóricos de sistemas dinámicos directamente en la interfaz háptica.

Las principales contribuciones de este trabajo son:

- Interfaz háptica para el estudio de diferentes sistemas físicos, y también para la implementación de nuevas aplicaciones.
- Comparación de desempeño de algoritmos de control clásicos con algoritmos de control modernos en la interfaz háptica.
- Metodología de desarrollo del interfaz háptica para fácil reproducción de resultados en hardware y software.

UNIVERSIDAD DE CUENCA  
desde 1867

## Capítulo 2

---

# Fundamentos teóricos

---

### 2.1. Interfaces hápticas

La principal característica de las interfaces hápticas es la retroalimentación que permite a un usuario realizar una tarea simulada o remota. Entonces, el principio de funcionamiento de estas interfaces depende de la trayectoria y el nivel de la fuerza aplicada hacia el cuerpo del operador permitiendo que éste, a su vez, obtenga información relevante del entorno de trabajo con el que está interactuando [2] [8].

El funcionamiento de las interfaces hápticas está supeditado también al desarrollo y/o evolución que el ambiente de trabajo experimenta a medida que el operador -limitado por la naturaleza propia del ambiente y la interfaz- interactúa con éste [9].

La interacción que el usuario se ve obligado a realizar en el ambiente simulado es lo que diferencia a las interfaces hápticas con las respuestas audiovisuales que los mismos ambientes proporcionan, ya que no solo envía o recibe información de manera unidireccional, sino que puede recibir, por el mismo medio, una respuesta instantánea ante algún gesto que se realice. Por ejemplo, en una computadora existen canales –periféricos- de entrada o salida, pero estos solo pueden enviar o recibir datos de manera unidireccional, en cambio, si un *mouse* tuviera un canal bidireccional, podría proveer de información relevante al usuario del entorno donde éste interactúa, lo que se traduce en un intercambio de energía bidireccional a través de una interfaz única [10] [4] -Figura 2.1-.

Una de las condiciones, en el comportamiento de la interfaz, es que ésta no debe interferir con su dinámica en la interacción entre el usuario y el ambiente de trabajo. También, el rendimiento de la interfaz se ve afectado por la presencia de diferentes causas que impiden un comportamiento aceptable en la dinámica del

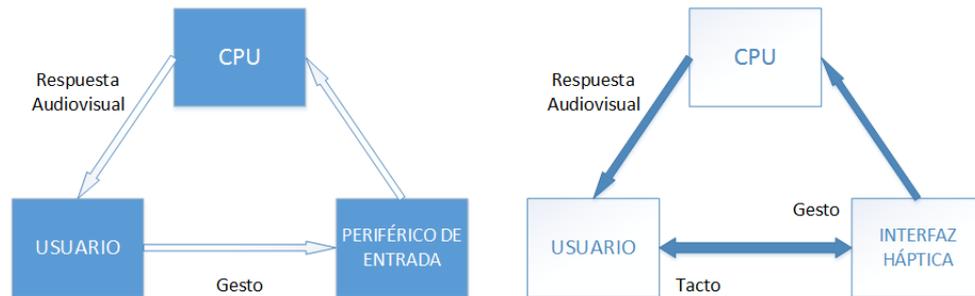


Figura 2.1: Diferencia entre las interfaces hápticas (der.) y dispositivos unidireccionales (izq.)

sistema como son la calidad de los actuadores y sensores, la rigidez y fricción del sistema, el tiempo y velocidad computacional, entre otras.

Dos características definen un buen rendimiento de la interfaz en general [9]:

- **Transparencia:** es una medida de la distorsión que existe entre la fuerza que se aplica al sistema y la retroalimentación que ésta produce, es decir, mide la fuerza producida por la dinámica de la interfaz.
- **Estabilidad:** se refiere a todo el rango de fuerzas que puede producir la interfaz para la comunicación con el usuario.

### 2.1.1. Impedancia vs admitancia

El acople de la interfaz con el ambiente de trabajo es un punto clave para la interacción debido a que limita el máximo o mínimo de impedancia que presenta el entorno de trabajo y que a su vez garantiza la estabilidad. Para dicho acople, se tienen dos tipos de dispositivos: de impedancia y admitancia, los cuales utilizan la fuerza o velocidad en la entrada y el complemento en la salida. En el primer caso, impedancia, ver Figura 2.2, se tiene una entrada referente al movimiento y una salida de fuerza; mientras, en el segundo caso, ver Figura 2.3, se aplica una fuerza en la entrada y se obtiene una salida referente al movimiento [10].

La mayoría de dispositivos hápticos tienen un control de impedancia, mientras que los dispositivos en los que se controla la admitancia, no son muy comunes.

### 2.1.2. Características generales

Varios son los parámetros técnicos y propiedades que poseen las interfaces hápticas y de estos dependerá el cumplimiento de las condiciones de transparencia



Figura 2.2: Dispositivo tipo Impedancia

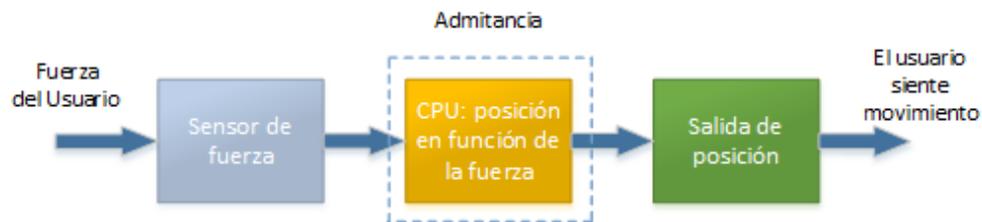


Figura 2.3: Dispositivo tipo Admitancia

y estabilidad. Al optimizar dichos parámetros, el rendimiento y calidad de la interfaz también se verán afectados de manera positiva.

- **Grados de libertad:** Una de las principales características que definen la interfaz es el número de grados de libertad que posee el dispositivo en el extremo o extremos activos. Nos indica el número de traslaciones y rotaciones que puede realizar el dispositivo [3].
- **Interfaz hombre-máquina:** Se diferencian dos tipos de interfaces que dependen de si el dispositivo se acopla al cuerpo del usuario -por ejemplo un guante- ó si la interfaz se encuentra en un punto fijo externo al usuario -por ejemplo un *joystick*-[11]. Esto da una idea clara, en la etapa de diseño, del tipo de dispositivo que se necesita y se va a construir.
- **Rango de movimiento:** Depende principalmente del número de grados de libertad que posee la interfaz y la orientación que puede seguir ésta. Por ende, a mayor número de rotaciones y traslaciones que pueda seguir el dispositivo, mayor complejidad en determinar el rango de movimiento. Además, pueden existir plataformas manejadas paralelamente, con lo cual, la complejidad aumenta debido a que el rango de movimiento no solo depende de los grados de libertad y la orientación del dispositivo sino que tiene variables ambiguas de forma y tamaño [12].
- **Fricción:** La transparencia del dispositivo está supeditada, entre otras, a la fuerza que percibe el usuario como retroalimentación o respuesta por



parte del sistema, pero, si esa fuerza es menor a las pérdidas por fricción, la interfaz pierde transparencia, esto conlleva a que las fuerzas producidas por fricción deben ser menores a la mínima fuerza que se percibe del sistema [13].

- **Inercia aparente y rigidez:** La inercia aparente es la masa mínima que percibe el usuario al momento de interactuar con la interfaz a través del espacio libre. Esta masa es proporcional al nivel de fatiga del operador dependiendo del tiempo de interacción con el entorno [13]. La rigidez, en cambio, se refiere a la inercia que obtienen los objetos virtuales permitiendo a la interfaz determinar las características físicas y geométricas de los mismos mediante restricciones físicas al movimiento de la interfaz.
- **Resolución:** Es uno de los aspectos fundamentales para la estabilidad de la interfaz; representa la desviación mínima que puede ser detectada por los sensores, y está considerada desde la salida, por lo que, la calibración del sensor debe ser óptima para conseguir estabilidad del sistema [12]. En el control mediante el computador la resolución se limita por medio del convertidor analógico digital.
- **Precisión:** Es la diferencia o el error existente entre la coordenada de referencia del dispositivo y la coordenada en la curva de posicionamiento perteneciente a la posición del objeto virtual [12]. La función de la curva se puede construir a partir de valores experimentales consiguiendo un alto grado de confiabilidad en la respuesta del sistema.
- **Ancho de banda:** Hace referencia al rango o escala de fuerzas con las que puede trabajar la interfaz. El actuador está representado por la fuerza de entrada  $F$  en la Ecuación (2.1), donde  $k$  es una constante,  $F_d$  es la fuerza deseada y  $F_s$  es la fuerza de salida, y se asume que el actuador responde idealmente. Pero suponiendo que el actuador no puede responder a alguna componente de la señal de entrada, el sistema puede variar de un sistema estable a uno inestable [14].

$$F = k(F_d - F_s) \quad (2.1)$$

Entonces, el sistema se ve restringido por el ancho de banda, en el cual, el actuador es capaz de responder de manera óptima y sin retardos.



## 2.2. Ambientes virtuales multimodales

Las interfaces pueden interactuar tanto con ambientes simulados o remotos, lo que a su vez trae consigo diferentes tipos de aplicaciones -las teleoperaciones y la realidad virtual son las que mayor auge han tenido- en el campo profesional y de investigación. Para tener una idea más clara de los espacios donde las interfaces se desenvuelven es necesario hablar de los ambientes virtuales multimodales y sus implicaciones efectivas a la hora de crear un sentido de presencia dentro de sus entornos.

Los sistemas de ambientes virtuales multimodales pueden realizar una comunicación con un usuario mediante presentación y combinación de diferentes canales de información: visión, háptica y audición.

Cabe mencionar que el sentido de presencia dentro de un ambiente con estas características aumenta, y la sensación de inmersión del usuario se enriquece debido a la mayor presencia de estímulos y señales que recibe, que se combinan y llenan vacíos que dejarían si estos canales se utilizaran por separado, además, la velocidad de procesamiento cognitivo del observador aumenta cuando se encuentra en ambientes multimodales a diferencia de si interactuara con ambientes unimodales o bimodales, que solo pueden presentar uno y dos canales de información, respectivamente [15].

El sentido de presencia también depende de la atención del observador, y esto es lo que logran, precisamente, este tipo de ambientes, que mediante mayor cantidad de estímulos sensoriales logran obtener mayor atención por parte del observador, y una mayor atención se traduce en un mayor sentido de presencia dentro del ambiente virtual, y como se dijo anteriormente esta presencia se enriquece.

### 2.2.1. Influencia del sentido de la visión

El complemento de las señales visuales con las señales hápticas en la exploración e identificación de objetos es uno de los varios temas que se pueden abarcar con los interfaces hápticos.

Diferentes experimentos y estudios [16] [17] se han realizado en este ámbito, que trata la psicofísica, donde se muestran resultados claros, principalmente, en la acción de reconocer el tamaño y rigidez de los objetos.

La visión y el tacto se compensan para que el usuario tenga una idea más transparente de las características del objeto que manipula.

Si se trabaja solo con una señal visual, por ejemplo, el tamaño del objeto no es transparente hacia el observador como se desearía, ya que por más que se trabaje con ambientes 3-D, la percepción de profundidad no es tan real; aquí es donde entran las interfaces hápticas, haciendo que la percepción de tamaño que se tenía anteriormente, reduzca su error.

Lo mismo sucede con la sensación de rigidez en un objeto. Mientras se trabaje solo con una señal háptica, no se tendrá una medida real en la percepción de esta característica, pero si se complementa con la señal visual se puede lograr que los objetos parezcan más y/o menos rígidos. Por ejemplo, al momento que un objeto, el cual manipula un usuario, intenta atravesar una superficie, la cual se desea que parezca más rígida que el objeto mencionado, se limita la posición del primer objeto, y más que limitar, mostrar visualmente que el objeto no avanza más allá de ese punto en la simulación, así el objeto la atraviere en realidad, con esto se logra que la superficie parezca más rígida de lo que en realidad es, ver Figura 2.4.

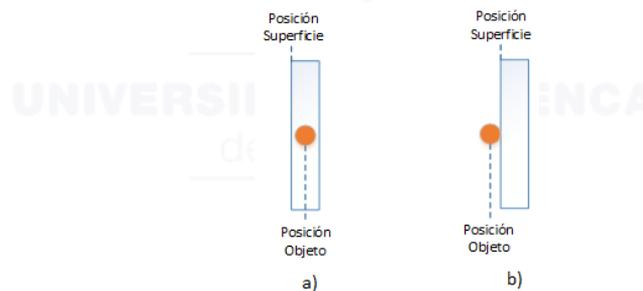


Figura 2.4: Influencia del sentido de la visión en los ambientes virtuales

### 2.2.2. Ambientes virtuales simulados y remotos

Dentro de los ambientes virtuales se distinguen dos tipos: simulados y remotos, cada uno diseñado específicamente para algún tipo de aplicación en particular.

Un ambiente simulado es aquel algoritmo que el CPU procesa produciendo una fuerza -en el caso de la interfaz con *feedback* de fuerza- como una reacción propia de la interfaz ante una entrada de posición o velocidad.

Una de las causas del desarrollo de las interfaces hápticas ha sido el progreso en materia de *hardware* que ha tenido la industria gráfica, permitiendo mejor calidad de imágenes, animaciones más realistas y simulaciones en tiempo real. Mientras, en materia de *software* se ha tenido un gran avance en las últimas décadas con las herramientas y técnicas utilizadas, así como la optimización de algoritmos.



Diferentes ambientes pueden ser simulados dependiendo del campo profesional en el que se quiera utilizar la interfaz háptica.

El control de los ambientes remotos es una tarea complicada debido a que se requiere demasiada información del ambiente para coordinar una acción o tarea. De hecho, obliga a que la comunicación deba tener el menor retardo posible y un gran ancho de banda para la transmisión de información.

La calidad de los actuadores frente a la percepción humana a menudo causa deficiencias en el control del actuador al otro lado de la interfaz, y es que, no es solo la calidad, sino las limitaciones propias de los dispositivos frente a lo que en realidad una persona puede percibir sensorialmente. Por ejemplo, al momento de recibir imágenes, así éstas sean de gran resolución, el observador no puede percibir la profundidad del entorno; lo mismo sucede en el caso de los motores, los cuales no pueden imprimir todo el rango de fuerza que el cuerpo humano está capacitado para sentir, un ejemplo claro de este tipo de ambientes son las telecirugías.

### 2.3. Tipos de interfaces

Una de las características importantes de estos dispositivos es que permiten restricciones naturales al movimiento de objetos. Al ser la comunicación bidireccional y en tiempo real, permite al usuario una sensación de inmersión.

Existen diferentes tipos de interfaces hápticas que difieren principalmente del tipo de respuesta que se desea obtener a la salida, y del tipo de características que se controla en el sistema. Por tanto, existen interfaces con *feedback* de fuerza, *feedback* táctil o *feedback* propioceptivo [13].

- **Interfaz con feedback de fuerza o movimiento (kinestésico):** proporciona información acerca del peso, inercia y dureza de un objeto virtual. Este tipo de retroalimentación es capaz de administrar fuerzas grandes capaces de detener el movimiento en su punto de interacción [18]. Por ejemplo, si un objeto virtual choca con una pared, también virtual, debería detenerse en ese punto sin poder continuar, deteniendo por completo el movimiento del hardware que el usuario controla en dirección de la pared. El empleo de la fuerza se lo realiza mediante alguna herramienta. Estos dispositivos, por lo general, están sujetos a alguna superficie [19] [3].
- **Interfaz con feedback táctil:** entrega datos del objeto virtual relacionados con la temperatura, vibraciones, presión, rugosidad que son detectados



por los receptores de la piel. La interfaz excita directamente la piel, lo que hace que estos dispositivos sean portables [20].

- **Interfaz con feedback propioceptivo:** brindan información acerca de la posición del objeto además de su estado, por ejemplo: la postura.

El uso de las interfaces antes mencionadas dependerá también del tipo de campo profesional en el que se aplique: medicina, educación, industria, entretenimiento, arte y demás [21].

## 2.4. Evaluación de las interfaces hápticas

La evaluación de las interfaces, como todo sistema en general, es importante en el sentido de tener medidas reales, tanto cualitativas y cuantitativas, del rendimiento y realismo que presente el sistema [22].

Muchas características son puestas a prueba en la identificación del rendimiento y calidad de una interfaz háptica. Se consideran las habilidades y limitaciones del usuario cuando utiliza el dispositivo, complementado con las características técnicas y físicas que poseen las interfaces para realizar una evaluación completa de las mismas [23].

La usabilidad se define como la evaluación de la interfaz de usuario de una aplicación específica, una técnica de interacción -algoritmo- o un dispositivo de entrada/salida, con el fin de determinar su facilidad de uso real o probable [22] [24].

Dentro de las características técnicas evaluables en una interfaz están parámetros como: grados de libertad, aceleración, rango de movimiento, inercia, fuerza máxima, resolución, precisión y ancho de banda que se presentan como alternativas para la evaluación cuantitativa de las interfaces [12].

En la Figura 2.5 se observan, resumidamente, las partes y parámetros en los cuales se basa la evaluación de las interfaces hápticas.

El dispositivo debe cumplir con, al menos, las siguientes especificaciones:

- Baja inercia y fricción en la retroalimentación, en el caso de los joysticks; y movimiento libre del actuador, sin restricciones al movimiento.
- Diferenciación entre objetos virtuales. El usuario debe diferenciar la fuerza de reacción que ejerce cada objeto debido a su densidad, por supuesto, esto depende del rango de fuerzas que puede producir la interfaz.

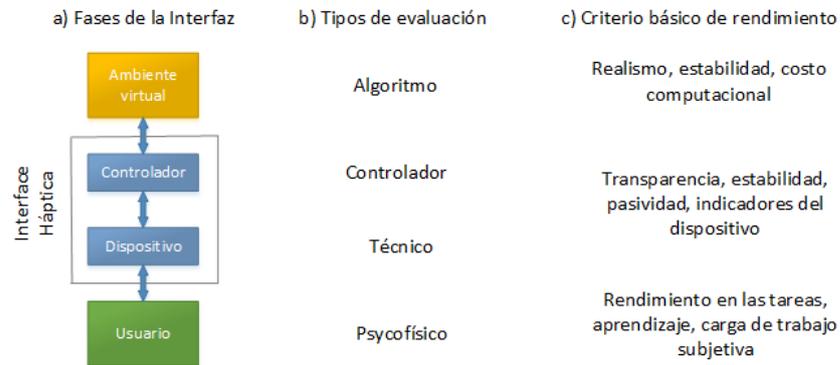


Figura 2.5: Clasificación de las técnicas de evaluación de las interfaces hápticas

- El dispositivo debe ser capaz de filtrar fenómenos involuntarios como errores de cuantización o pobre respuesta del sistema.
- Los actuadores no deben influir en el comportamiento del sistema con su dinámica propia, logrando mayor transparencia.
- Debido a que el dolor e incomodidad supera la mayoría de las sensaciones, la ergonomía y confort son parte importante del diseño de la interfaz háptica, así el rendimiento del usuario se prolonga por más tiempo ya que no existe demasiada fatiga cuando utiliza el dispositivo.

## 2.5. Definiciones en control clásico

La ingeniería de control busca controlar diferentes tipos de variables de un determinado sistema en busca del beneficio energético, además, de tener total control sobre procesos y así obtener el mayor margen de beneficios de producción y, por lo tanto, económicos.

El control clásico se basa únicamente en el estudio de la estabilidad de sistemas mediante la reducción del error entre la salida y la entrada. La limitante de este tipo de sistemas es que solo se pueden aplicar a plantas cuyos comportamientos son invariantes en el tiempo, así como, la relación entre su entrada y salida debe ser lineal.

El análisis de estabilidad se realiza mediante la obtención de modelos matemáticos del funcionamiento de la planta. El modelo matemático relaciona la salida y entrada del sistema, denominado función de transferencia. La Figura 2.6 muestra el diagrama de bloques de este tipo de modelo.



Figura 2.6: Función de transferencia

Matemáticamente, la función de transferencia, en el dominio de Laplace, de acuerdo a la Figura 2.6 es

$$FT(s) = \frac{Y(s)}{X(s)} \quad (2.2)$$

Un sistema donde la salida del sistema tiene influencia sobre la entrada del mismo es un sistema retroalimentado o realimentado, y se denomina sistema en lazo cerrado, ver Figura 2.7. Mientras que, cuando el sistema o proceso solo tiene una entrada independiente de la salida se denomina sistema en lazo abierto, ver Figura 2.8.

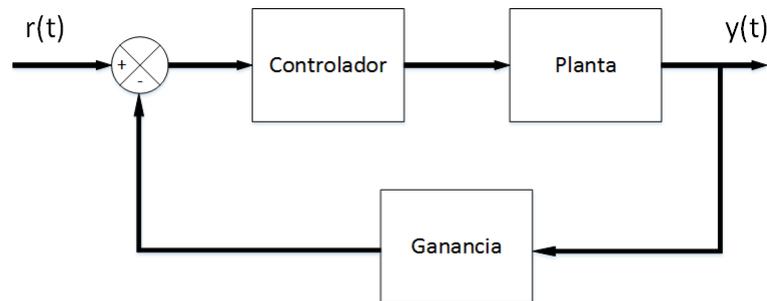


Figura 2.7: Diagrama de bloques de un sistema en lazo cerrado

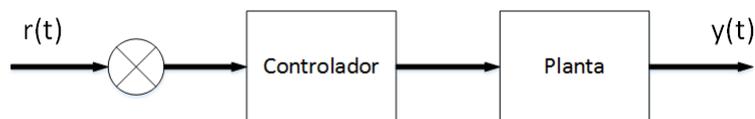


Figura 2.8: Diagrama de bloques de un sistema en lazo abierto

La realimentación consiste en disminuir el error de respuesta del sistema, haciendo que éste permanezca estable ante cualquier perturbación a la que pueda estar expuesto.

- Estabilidad.** *“Un sistema que se puede describir por medio de ecuaciones integrodiferenciales lineales, con coeficientes constantes, es estable si todas sus raíces características están a la izquierda del eje imaginario del plano*



complejo” [25]. Mientras que los sistemas cuyos polos se encuentren en el semiplano derecho del eje imaginario del plano complejo son inestables.

La estabilidad de la planta se la analiza mediante la obtención de polos que corresponden a las raíces características de la función de transferencia, Ecuación (2.2) pertenecientes al denominador de la misma,  $\mathbf{X}(s)$ .

Si el sistema es estable, pero la respuesta del sistema no es viable a partir de una simple realimentación de la salida, se puede proceder a aplicar controladores que agreguen polos al sistema de lazo abierto, provocando una respuesta más rápida de la planta, optimizando el rendimiento de la misma.

Uno de estos controladores es el PID, estudiado en la siguiente sección.

### 2.5.0.1. Controlador proporcional-integral-derivativo (PID)

Los controladores PID han sido ampliamente usados en el control industrial, debido a su sencillez de ajuste de parámetros sin la necesidad de conocer el modelo del sistema que se controla [26].

Del mismo modo, este tipo de controlador ha sido utilizado como alternativa en el ámbito educacional, donde las aplicaciones no requieren un control estricto pero si necesitan eficiencia y precisión en la respuesta.

La acción P actúa de manera proporcional al error producido entre la señal de referencia y la salida actual, y ajusta la salida del controlador de acuerdo al tamaño del error. La acción I elimina el error de estado estacionario inducida por el efecto proporcional; esta acción, por el contrario, provoca una respuesta más lenta del controlador generando un aumento en el tiempo de establecimiento. La acción D se anticipa ante una futura tendencia de la señal de referencia, que resulta en una disminución del exceso de sobreoscilaciones [27].

El controlador PID se ajusta mediante los parámetros  $k_c$ ,  $T_i$  y  $T_d$  (correspondientes a las acciones Proporcional-Integral-Derivativa, respectivamente), todos estos agrupados en la ecuación característica que modela un controlador PID, Ecuación (2.3), y cuyo diagrama de bloques se observa en la Figura 2.9.

$$u(t) = k_c \left[ e(t) + \frac{1}{T_i} \int e(t) dt + T_d \frac{d}{dt} [e(t)] \right] \quad (2.3)$$

Debido al control digital de la interfaz, el modelo del PID del dominio continuo se lo traslada al dominio discreto. El desarrollo para la discretización de un controlador PID se muestra a continuación.

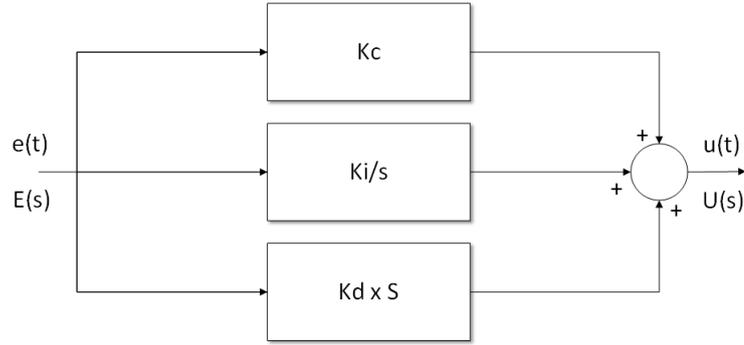


Figura 2.9: Diagrama de bloques del PID

Se transforma la Ecuación (2.3) al dominio de Laplace.

$$U(s) = k_c \left[ E(s) + \frac{1}{sT_i} E(s) + sT_d E(s) \right] \quad (2.4)$$

Se utiliza un retenedor de orden cero -ZOH: *zero order hold*- para la transformación al dominio Z, el cual contiene un periodo de muestreo, característico de los sistemas discretos.

$$s = \frac{1 - z^{-1}}{T} \quad (2.5)$$

Reemplazando tenemos,

$$U(z) = k_c \left[ E(z) + \frac{1}{T_i \frac{1-z^{-1}}{T}} [E(z)] + T_d \frac{1-z^{-1}}{T} E(z) \right]$$

$$U(z) = k_c \left[ E(z) + \frac{T}{T_i} \frac{1}{1-z^{-1}} E(z) + \frac{T_d}{T} (1 - z^{-1}) E(z) \right]$$

$$U(z) = k_c \left[ \frac{E(z) - z^{-1}E(z) + \frac{T}{T_i} E(z) + \frac{T_d}{T} (1 - z^{-1})^2 E(z)}{1 - z^{-1}} \right]$$

agrupando términos,

$$U(z) - z^{-1}U(z) = k_c \left[ E(z) - z^{-1}E(z) + \frac{T}{T_i} E(z) + \frac{T_d}{T} E(z) + \frac{T_d}{T} (1 - 2z^{-1} + z^{-2}) E(z) \right]$$

$$U(z) - z^{-1}U(z) = k_c \left[ \left(1 + \frac{T}{T_i} + \frac{T_d}{T}\right) E(z) - \left(1 + 2\frac{T_d}{T}\right) z^{-1}E(z) + \frac{T_d}{T} z^{-2}E(z) \right]$$

finalmente, a la salida del controlador discretizado se tiene

$$u_k = u_{k-1} + k_c \left[ \left(1 + \frac{T}{T_i} + \frac{T_d}{T}\right) e_k - \left(1 + 2\frac{T_d}{T}\right) e_{k-1} + \frac{T_d}{T} e_{k-2} \right] \quad (2.6)$$

De la Ecuación (2.6) se observa que el tiempo de muestreo juega un papel importante en el control, del cual dependerá el tiempo de respuesta del sistema.

### 2.5.1. Sintonización del controlador PID

La sintonización de los controladores PID depende del tipo de variables que se encuentren involucrados en el control de la planta, por lo tanto, no existe un método mejor que el experimental para la determinación o sintonización de los valores óptimos de  $k_c$ ,  $T_d$  y  $T_i$ . Uno de estos métodos de sintonización es el de Ziegler – Nichols, que es uno de los más utilizados y conocidos en la práctica.

#### 2.5.1.1. Método de Ziegler – Nichols

Los parámetros del PID se encuentran expresados de forma explícita en términos de la última ganancia  $k_u$ , también conocida como ganancia crítica, y el último periodo,  $P_u$ , de una oscilación sostenida. Existen dos métodos para la sintonización mediante este método.

- **Primer Método.** Si la respuesta en la salida no tiene oscilaciones y posee una respuesta sobreamortiguada, como se observa en la Figura 2.10, se puede utilizar este método a partir del tiempo del retraso  $L$  y la constante de tiempo  $\tau$  que definen la Ecuación (2.7).

$$\frac{C(s)}{U(s)} = \frac{ke^{-Ls}}{\tau s + 1} \quad (2.7)$$

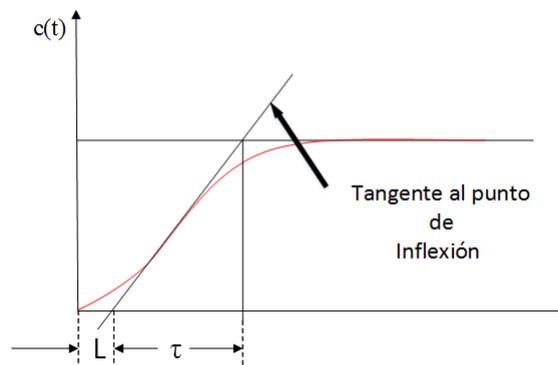


Figura 2.10: Curva experimental para determinar los parámetros del PID

Con los valores de  $L$  y  $\tau$  se puede obtener los valores de  $k_c$ ,  $T_i$  y  $T_d$  de acuerdo a la Tabla 2.1.

- **Segundo Método.** Se utiliza para sistemas que presentan oscilaciones sostenidas. Para encontrar los valores de  $k_u$  y  $P_u$ , se debe trabajar solo con la

Tabla 2.1: Variables para la sintonización, primer método

Controlador	$k_c$	$T_i$	$T_d$
P	$\tau/L$	$\infty$	0
PI	$0.9T/L$	$L/0.3$	0
PID	$1.2T/L$	$2L$	$0.5L$

parte proporcional del controlador PID hasta que el sistema tenga oscilaciones sostenidas, el valor de ganancia,  $k_c$ , con que se logre esto se denomina  $k_u$ , y el periodo  $P_u$  es el que se indica en la Figura 2.11. A partir de estos valores de ganancia se pueden encontrar el resto de parámetros del PID, ver Tabla 2.2.

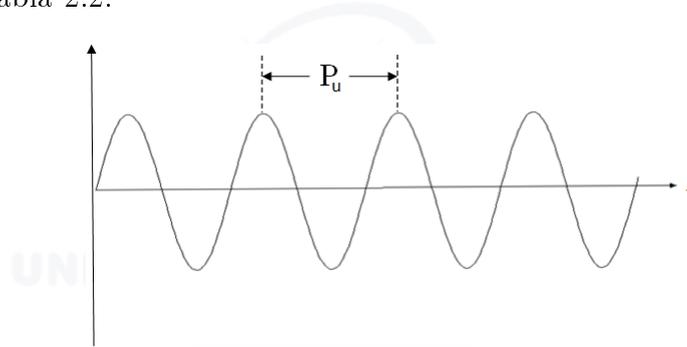


Figura 2.11: Determinación de  $P_u$

Tabla 2.2: Variables para la sintonización, segundo método

Ziegler-Nichols	$k_c$	$T_i$	$T_d$	Observaciones
P	$k_u/2$	-	-	Recomendado
PI	$k_u/2.2$	$P_u/1.2$	-	$0.2 < D/\tau < 2$
PID	$k_u/1.7$	$P_u/2$	$P_u/8$	
Tyres-Luyben	$k_c$	$T_i$	$T_d$	
PI	$k_u/3.2$	$P_u/0.45$	-	Recomendado
PID	$k_u/2.2$	$P_u/0.45$	$P_u/6.3$	$D/\tau < 0.2$
Ciancone-Marlin	$k_c$	$T_i$	$T_d$	
PI	$k_u/3.3$	$P_u/4$	-	Recomendado
PID	$k_u/3.3$	$P_u/4.4$	$P_u/8.1$	$D/\tau > 2$

En la Tabla 2.2 se observan unas variaciones al método de Ziegler – Nichols. Cuando  $D/\tau < 0,2$  el rendimiento del sistema empieza a decrecer, por lo que el método de Tyres-Luyben se basa en el sistema de tiempo muerto mediante la adición de un integrador  $G = k_c e^{-Ds}/s$ ; este método traba-



ja bien para un proceso de constante de tiempo dominante. Mientras que el método de Ciancone-Marlin trabaja mejor para los procesos de tiempo muerto dominantes, con una acción proporcional más conservativa y una acción integral más agresiva.

## 2.6. Definiciones en control moderno

Los métodos convencionales de control no son aplicables para el diseño de sistemas de control óptimo y adaptable, debido a que estos sistemas pueden tener varias entradas y salidas, además, que los mismos sistemas pueden ser variantes en el tiempo y/o no lineales; en este caso, los sistemas convencionales que solo se basan en su función de transferencia para evaluar una entrada y su respectiva salida, no son aptos para el diseño.

El control, entonces, toma un camino diferente y se basa en el análisis de espacio de estados para controlar este tipo de sistemas que en sus estructuras pueden presentar una complejidad alta entre las entradas y salidas.

- **Método en el espacio de estados:** *“Este método se basa en la descripción del sistema en términos de  $n$  ecuaciones en diferencias o diferenciales de primer orden, que pueden combinarse en una ecuación matricial en diferencias o diferencial de primer orden”* [28].
- **Estado:** es la representación mínima de información del sistema en un instante dado,  $t = t_0$ , a partir del cual, junto con el conocimiento de la entrada en  $t > t_0$ , se puede determinar los siguientes estados del sistema por completo para cualquier tiempo  $t > t_0$ .
- **Variables de estado:** son el conjunto de variables que representan la cantidad mínima de información -estado- en el instante  $t = t_0$ . Las variables de estado pueden ser cantidades no medibles ni observables, por lo que representa una ventaja en el control con espacio de estados. Sin embargo, el sistema de control depende de la realimentación ponderada de las variables de estado, por lo que se requiere que éstas sean fácilmente observables.
- **Vector de estado:** es el conjunto de  $n$  variables de estado que determinan de manera singular el estado  $x(t)$  del sistema para cualquier tiempo  $t > t_0$ . La dimensión de este vector de estado, generalmente, en los sistemas físicos reales es finita.



- **Espacio de estado:** es el espacio vectorial donde los estados del sistema toman valores y pueden representarse, por tanto, el espacio de estados tendrá la misma dimensión del vector de estado.
- **Ecuaciones en el espacio de estado:** en el modelo dinámico determinista la representación del diseño solo está ligada a la relación entre las entradas y salidas del sistema. En el modelo actual, en cambio, se añaden los estados del sistema, los cuales, pueden determinar una salida relacionada con la entrada para cualquier instante  $t > t_0$ , debido a que contienen toda la información del sistema.

La representación de un sistema en el espacio de estado no es única, puede tomar diferentes representaciones, pero lo que no varía es la cantidad de variables de estado que están involucradas en el sistema en cualquier representación.

Tanto los sistemas de tiempo continuo como los sistemas de tiempo discreto se pueden modelar de la misma manera.

Los sistemas discretos variantes en el tiempo –lineales o no lineales- se representan mediante su ecuación de estado

$$\mathbf{x}(k+1) = \mathbf{f}[\mathbf{x}(k), \mathbf{u}(k), k]$$

y su ecuación de salida

$$\mathbf{y}(k) = \mathbf{g}[\mathbf{x}(k), \mathbf{u}(k), k]$$

Las ecuaciones anteriores se pueden simplificar para los sistemas lineales variantes en el tiempo, quedando de la siguiente forma:

$$\begin{aligned}\mathbf{x}(k+1) &= \mathbf{G}(k) \mathbf{x}(k) + \mathbf{H}(k) \mathbf{u}(k) \\ \mathbf{y}(k) &= \mathbf{C}(k) \mathbf{x}(k) + \mathbf{D}(k) \mathbf{u}(k)\end{aligned}$$

donde

$\mathbf{x}(k)$	=	vector $n$	(vector de estado)
$\mathbf{y}(k)$	=	vector $m$	(vector de salida)
$\mathbf{u}(k)$	=	vector $r$	(vector de entrada)
$\mathbf{G}(k)$	=	matriz $n \times n$	(matriz de estado)
$\mathbf{H}(k)$	=	matriz $n \times r$	(matriz de entrada)
$\mathbf{C}(k)$	=	matriz $m \times n$	(matriz de salida)
$\mathbf{D}(k)$	=	matriz $m \times r$	(matriz de transmisión directa)

Precisamente, la presencia de la variable  $k$  dentro de la matrices  $\mathbf{G}$ ,  $\mathbf{H}$ ,  $\mathbf{C}$ , y  $\mathbf{D}$  implica la variabilidad del sistema en el tiempo, por lo que si esta variable no aparece de forma explícita indica que el sistema es invariante en el tiempo, simplificándose de la siguiente manera:

$$\mathbf{x}(k+1) = \mathbf{G}\mathbf{x}(k) + \mathbf{H}\mathbf{u}(k) \quad (2.8)$$

$$\mathbf{y}(k) = \mathbf{C}\mathbf{x}(k) + \mathbf{D}\mathbf{u}(k) \quad (2.9)$$

El diagrama de bloques de dicho control es

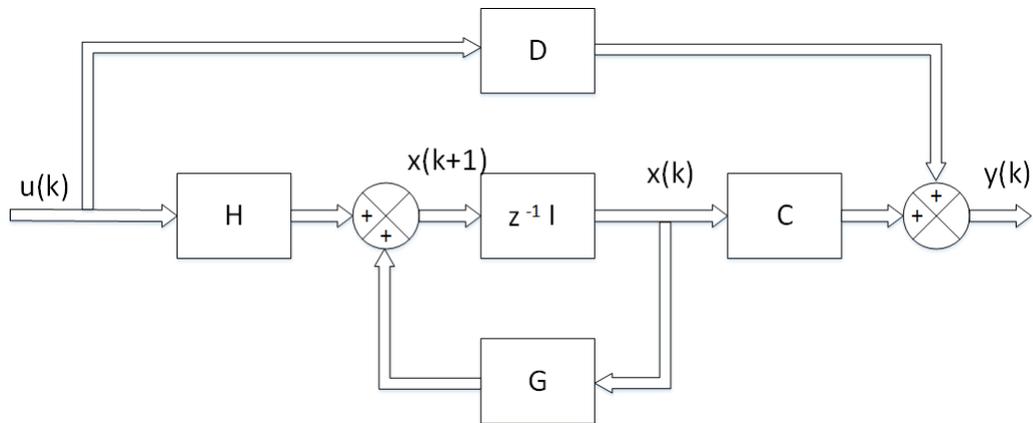


Figura 2.12: Sistema de control en tiempo discreto, lineal e invariante en el tiempo

### 2.6.1. Representación de sistemas en tiempo discreto en espacio de estados

Considerando el modelo de un sistema por su ecuación de diferencias en tiempo discreto

$$y(k) + a_1y(k-1) + a_2y(k-2) + \dots + a_ny(k-n) = b_0u(k) + b_1u(k-1) + \dots + b_nu(k-n) \quad (2.10)$$

donde  $\mathbf{u}(k)$  e  $\mathbf{y}(k)$  representan la entrada y salida del sistema en el instante  $k$ , ponderados por los coeficientes  $a_i$  y  $b_j$ . La Ecuación (2.10) se puede reescribir en forma de su función de transferencia

$$\frac{Y(z)}{U(z)} = \frac{b_0 + b_1z^{-1} + \dots + b_nz^{-n}}{1 + a_1z^{-1} + \dots + a_nz^{-n}} \quad (2.11)$$



si dividimos para  $z^n$ , tanto el numerador como el denominador, tenemos

$$\frac{Y(z)}{U(z)} = \frac{b_0 z^n + b_1 z^{n-1} + \dots + b_n}{z^n + a_1 z^{n-1} + \dots + a_n} \quad (2.12)$$

Se puede tener varias representaciones del sistema en espacio de estados, como se mencionó anteriormente, a partir de la función de transferencia presentada en la Ecuación (2.12). Las representaciones del sistema en espacio de estados pueden ser:

- **Forma canónica controlable:** La representación de las ecuaciones de estado y salida del sistema, respectivamente, son

$$\begin{bmatrix} x_1(k+1) \\ x_2(k+1) \\ \vdots \\ x_{n-1}(k+1) \\ x_n(k+1) \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \\ -a_n & -a_{n-1} & -a_{n-2} & \cdots & -a_1 \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \\ \vdots \\ x_{n-1}(k) \\ x_n(k) \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} u(k) \quad (2.13)$$

$$y(k) = \begin{bmatrix} b_n - a_n b_0 & b_{n-1} - a_{n-1} b_0 & \cdots & b_1 - a_1 b_0 \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \\ \vdots \\ x_n(k) \end{bmatrix} + b_0 u(k) \quad (2.14)$$

- **Forma canónica observable:** La matriz de estado es la transpuesta de la matriz de estado dada en la Ecuación (2.13). Las ecuaciones de estado y de salida son

$$\begin{bmatrix} x_1(k+1) \\ x_2(k+1) \\ \vdots \\ x_{n-1}(k+1) \\ x_n(k+1) \end{bmatrix} = \begin{bmatrix} 0 & 0 & \cdots & 0 & 0 & -a_n \\ 1 & 0 & \cdots & 0 & 0 & -a_{n-1} \\ \vdots & \vdots & & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & 0 & -a_2 \\ 0 & 0 & \cdots & 0 & 1 & -a_1 \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \\ \vdots \\ x_{n-1}(k) \\ x_n(k) \end{bmatrix} + \begin{bmatrix} b_n - a_n b_0 \\ b_{n-1} - a_{n-1} b_0 \\ \vdots \\ b_2 - a_2 b_0 \\ b_1 - a_1 b_0 \end{bmatrix} u(k) \quad (2.15)$$

$$y(k) = \begin{bmatrix} 0 & 0 & \cdots & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \\ \vdots \\ x_{n-1}(k) \\ x_n(k) \end{bmatrix} + b_0 u(k) \quad (2.16)$$

- **Forma canónica diagonal:** Si los polos de la función de transferencia

escrita en la Ecuación (2.12) son todos distintos, su representación diagonal en espacio de estados viene dada por

$$\begin{bmatrix} x_1(k+1) \\ x_2(k+1) \\ \vdots \\ x_n(k+1) \end{bmatrix} = \begin{bmatrix} p_1 & 0 & \cdots & 0 \\ 0 & p_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & p_n \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \\ \vdots \\ x_n(k) \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix} u(k) \quad (2.17)$$

$$y(k) = \begin{bmatrix} c_1 & c_2 & \cdots & c_n \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \\ \vdots \\ x_n(k) \end{bmatrix} + b_0 u(k) \quad (2.18)$$

- Forma canónica de Jordan:** Si la función de transferencia presenta un polo del orden  $m$  en  $z = p_1$ , y todos los demás polos son distintos, entonces, la representación en espacio de estados es

$$\begin{bmatrix} x_1(k+1) \\ x_2(k+1) \\ \vdots \\ x_m(k+1) \\ x_{m+1}(k+1) \\ \vdots \\ x_n(k+1) \end{bmatrix} = \begin{bmatrix} p_1 & 1 & 0 & \cdots & 0 & 0 & \cdots & 0 \\ 0 & p_2 & 1 & \cdots & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & p_1 & 0 & \cdots & 0 \\ 0 & 0 & 0 & \cdots & 0 & p_{m+1} & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 0 & 0 & \cdots & p_n \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \\ \vdots \\ x_m(k) \\ x_{m+1}(k) \\ \vdots \\ x_n(k) \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix} u(k) \quad (2.19)$$

$$y(k) = \begin{bmatrix} c_1 & c_2 & \cdots & c_n \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \\ \vdots \\ x_n(k) \end{bmatrix} + b_0 u(k) \quad (2.20)$$

### 2.6.2. Controlabilidad

Un sistema es controlable si a partir de un estado inicial arbitrario se puede transferir un sistema a otro estado -también arbitrario- en un número finito de periodos de muestreo. Es decir, las variables de estado son controladas en un espacio de tiempo finito mediante cualquier señal de control, por lo que se concluye que las variables de estado no deben ser independientes de la señal de control, puesto que volvería imposible de controlar dichas variables y, por ende, el sistema

también sería no controlable.

“El sistema de control en tiempo discreto dado por la Ecuación (2.21) se dice es de estado completamente controlable, o simplemente de estado controlable, si existe una señal de control constante por intervalos  $u(kT)$  definida a lo largo de un número finito de períodos de muestreo de forma que, al partir de cualquier estado inicial, el estado  $\mathbf{x}(kT)$  pueda ser transferido al estado deseado  $\mathbf{x}_j$  en  $n$  períodos de muestreo como máximo” [28].

$$\mathbf{x}((k+1)T) = \mathbf{G}\mathbf{x}(kT) + \mathbf{H}u(kT) \quad (2.21)$$

La solución a la Ecuación (2.21) es

$$\begin{aligned} \mathbf{x}(nT) &= \mathbf{G}^n \mathbf{x}(0) + \sum_{j=0}^{n-1} \mathbf{G}^{n-j-1} \mathbf{H}u(jT) \\ \mathbf{x}(nT) &= \mathbf{G}^n \mathbf{x}(0) + \mathbf{G}^{n-1} \mathbf{H}u(0) + \mathbf{G}^{n-2} \mathbf{H}u(T) + \dots + \mathbf{H}u((n-1)T) \end{aligned}$$

se obtiene

$$\mathbf{x}(nT) - \mathbf{G}^n \mathbf{x}(0) = \begin{bmatrix} \mathbf{H} & \mathbf{G}\mathbf{H} & \dots & \mathbf{G}^{n-1}\mathbf{H} \end{bmatrix} \begin{bmatrix} u((n-1)T) \\ u((n-2)T) \\ \vdots \\ u(0) \end{bmatrix} \quad (2.22)$$

Como  $\mathbf{H}$  es una matriz de  $n \times 1$ , se deduce que cada conjunto de matrices de  $\mathbf{H}:\mathbf{G}\mathbf{H}:\dots:\mathbf{G}^{n-1}\mathbf{H}$  es una matriz de  $n \times 1$  o un vector columna. A partir de esto, si el rango de la matriz

$$\mathbf{M} = \begin{bmatrix} \mathbf{H} & \mathbf{G}\mathbf{H} & \dots & \mathbf{G}^{n-1}\mathbf{H} \end{bmatrix}$$

es  $n$ , es decir

$$\text{rango} \left( \begin{bmatrix} \mathbf{H} & \mathbf{G}\mathbf{H} & \dots & \mathbf{G}^{n-1}\mathbf{H} \end{bmatrix} \right) = n \quad (2.23)$$

los  $n$  vectores abarcan todo el espacio vectorial de  $n$  dimensiones. A la matriz  $\mathbf{M}$  se la conoce normalmente como matriz de controlabilidad.



### 2.6.3. Observabilidad

De la misma manera que la controlabilidad, se plantea un sistema en tiempo discreto lineal e invariante en el tiempo para el análisis de observabilidad

$$\mathbf{x}((k+1)T) = \mathbf{G}\mathbf{x}(kT) \quad (2.24)$$

$$\mathbf{y}((k+1)T) = \mathbf{C}\mathbf{x}(kT) \quad (2.25)$$

donde

$\mathbf{x}(kT)$  = vector de estado en el  $k$ -ésimo instante de muestreo –dimensión  $n$ –

$\mathbf{y}(kT)$  = vector de salida en el  $k$ -ésimo instante de muestreo –dimensión  $m$ –

$\mathbf{G}$  = matriz de estado de  $n \times n$

$\mathbf{C}$  = matriz de salida de  $m \times n$

*“El sistema se dice ser completamente observable si cualquier estado inicial  $\mathbf{x}(0)$  puede determinarse a partir de la observación de  $\mathbf{y}(\mathbf{kT})$  sobre un número finito de períodos de muestreo. El sistema, por lo tanto, es completamente observable, si cualquier transición del estado de manera eventual afecta a todos los elementos del vector de salida”[28].*

La observabilidad es útil para el problema de variables de estado no medibles, como se verá más adelante, la observación permite realizar una estimación de dichas variables.

La condición para la observabilidad completa del sistema se deduce a partir de la solución de  $\mathbf{x}(kT)$  como sigue

$$\mathbf{x}(kT) = \mathbf{G}^k \mathbf{x}(0)$$

$$\mathbf{y}(kT) = \mathbf{C}\mathbf{G}^k \mathbf{x}(0)$$

Como se observa, la observabilidad implica que, conocidos los valores de  $y(0)$ ,  $y(T)$ ,  $y(2T)$ ,..., es posible determinar los valores de  $x_1(0)$ ,  $x_2(0)$ ,...,  $x_n(0)$ . Por tanto, para determinar  $n$  incógnitas se necesitan  $n$  valores de  $\mathbf{y}(kT)$ . Es decir

$$\mathbf{y}(0) = \mathbf{C}\mathbf{x}(0)$$

$$\mathbf{y}(T) = \mathbf{C}\mathbf{G}\mathbf{x}(0)$$

⋮

$$\mathbf{y}((n-1)T) = \mathbf{C}\mathbf{G}^{n-1}\mathbf{x}(0)$$



Al final se deben obtener un conjunto de  $n$  ecuaciones linealmente independientes. Esto requiere que la matriz de observabilidad  $\mathbf{N}$  debe ser de rango  $n$ .

$$\mathbf{N} = \begin{bmatrix} \mathbf{C} \\ \mathbf{CG} \\ \vdots \\ \mathbf{CG}^{n-1} \end{bmatrix}$$

$$\text{rango} \left( \left[ \mathbf{C}^T : \mathbf{G}^T \mathbf{C}^T : \dots : (\mathbf{G}^T)^{n-1} \mathbf{C}^T \right] \right) = n$$

La matriz  $\mathbf{N}$  se conoce normalmente como matriz de observabilidad.

#### 2.6.4. Realimentación de estados

La regulación de los sistemas se basa en mantener todas las variables de estado en su estado de equilibrio. La técnica de ubicación de polos ayuda en este proceso. Para la utilización de este método se debe primero probar que el sistema es de estado completamente controlable. Es decir

$$\text{rango} \left( \left[ \mathbf{H} : \mathbf{GH} : \dots : \mathbf{G}^{n-1} \mathbf{H} \right] \right) = n$$

Los polos del sistema, por tanto, se pueden ubicar en cualquier localización, la única condición es la anterior, y a partir de ahí se puede realimentar los estados del sistema mediante una matriz de ganancia apropiada, y con una adecuada ponderación.

Entonces, considerando el sistema

$$\mathbf{x}(k+1) = \mathbf{G}\mathbf{x}(kT) + \mathbf{H}u(kT) \quad (2.26)$$

donde

$\mathbf{x}(k)$  = vector de estado en el  $k$ -ésimo instante de muestreo –dimensión  $n$ -

$u(k)$  = señal de control en el  $k$ -ésimo instante de muestreo –escalar-

$\mathbf{G}$  = matriz de estado de  $n \times n$

$\mathbf{H}$  = matriz de salida de  $n \times 1$

cuya representación en lazo abierto se observa en la Figura 2.13.

Suponiendo que la señal de control  $u(k)$  está dada por

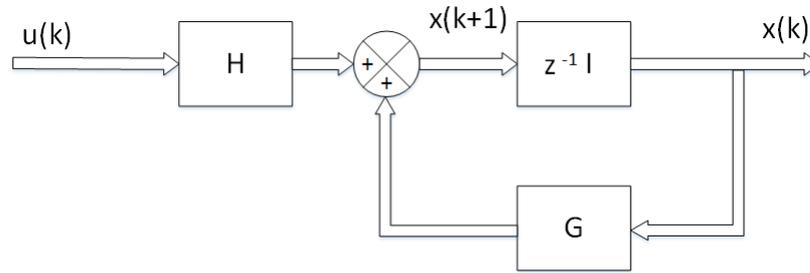


Figura 2.13: Sistema de control en lazo abierto

$$u(k) = -\mathbf{K}\mathbf{x}(k)$$

donde  $\mathbf{K}$  es la matriz de realimentación de estado de dimensión  $1 \times n$ , lo que convierte al sistema en un sistema de control de lazo cerrado, ver Figura 2.14, con su ecuación de estado

$$\mathbf{x}(k+1) = (\mathbf{G} - \mathbf{H}\mathbf{K})\mathbf{x}(k) \quad (2.27)$$

$\mathbf{K}$ , se escoge de manera que los valores característicos de  $\mathbf{G} - \mathbf{H}\mathbf{K}$  sean los polos en lazo cerrado deseados  $\mu_1, \mu_2, \dots, \mu_n$ . Esto determina, también, que si el sistema no fuera controlable, existieran valores característicos de  $\mathbf{G} - \mathbf{H}\mathbf{K}$  que no pudieran ser controlados por la realimentación de estado.

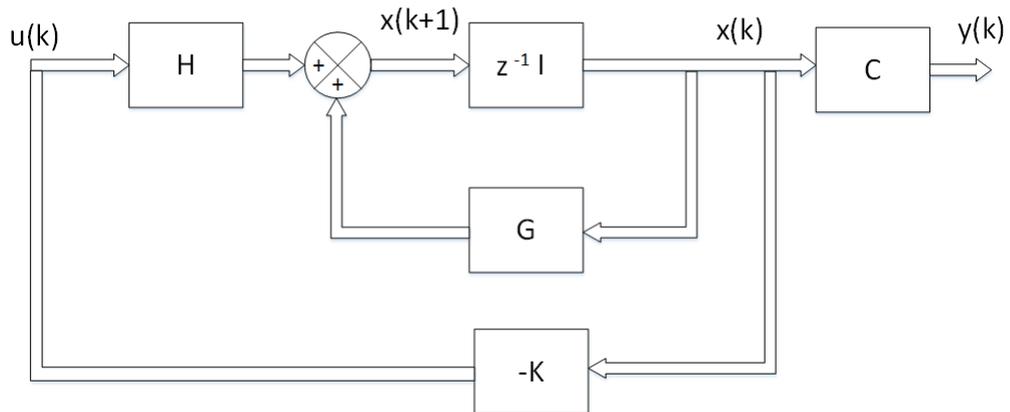


Figura 2.14: Sistema de control con realimentación de estado

### 2.6.5. Fórmula de Ackermann

La fórmula de Ackermann es útil para encontrar la matriz de ganancia de realimentación del estado,  $\mathbf{K}$ .



Considerando el sistema de estado completamente controlable

$$\mathbf{x}(k+1) = \mathbf{G}\mathbf{x}(k) + \mathbf{H}u(k) \quad (2.28)$$

cuya realimentación del estado está dada por

$$u(k) = -\mathbf{K}\mathbf{x}(k)$$

y sus polos en lazo cerrado se desea ubicarlos en

$$\begin{aligned} z_1 &= \mu_1 \\ z_2 &= \mu_2 \\ &\vdots \\ z_n &= \mu_n \end{aligned}$$

Por tanto, la ecuación característica es

$$\begin{aligned} |z\mathbf{I} - \mathbf{G} + \mathbf{H}\mathbf{K}| &= (z - \mu_1)(z - \mu_2) \cdots (z - \mu_n) \\ |z\mathbf{I} - \mathbf{G} + \mathbf{H}\mathbf{K}| &= z^n + \alpha_1 z^{n-1} + \alpha_2 z^{n-2} + \cdots + \alpha_{n-1} z + \alpha_n = 0 \end{aligned}$$

Definiendo

$$\hat{\mathbf{G}} = \mathbf{G} - \mathbf{H}\mathbf{K}$$

El teorema de Cayley-Hamilton define que  $\hat{\mathbf{G}}$  satisface su propia ecuación característica, por lo tanto

$$\hat{\mathbf{G}}^n + \alpha_1 \hat{\mathbf{G}}^{n-1} + \alpha_2 \hat{\mathbf{G}}^{n-2} + \cdots + \alpha_{n-1} \hat{\mathbf{G}} + \alpha_n \mathbf{I} = \phi(\hat{\mathbf{G}}) = \mathbf{0}$$

Este es el criterio para deducir la fórmula de Ackermann.

Sean

$$\begin{aligned} \mathbf{I} &= \mathbf{I} \\ \hat{\mathbf{G}} &= \mathbf{G} - \mathbf{H}\mathbf{K} \\ \hat{\mathbf{G}}^2 &= (\mathbf{G} - \mathbf{H}\mathbf{K})^2 = \mathbf{G}^2 - \mathbf{G}\mathbf{H}\mathbf{K} - \mathbf{H}\mathbf{K}\hat{\mathbf{G}} \\ \hat{\mathbf{G}}^3 &= (\mathbf{G} - \mathbf{H}\mathbf{K})^3 = \mathbf{G}^3 - \mathbf{G}^2\mathbf{H}\mathbf{K} - \mathbf{G}\mathbf{H}\mathbf{K}\hat{\mathbf{G}} - \mathbf{H}\mathbf{K}\hat{\mathbf{G}}^2 \\ &\vdots \\ \hat{\mathbf{G}}^n &= (\mathbf{G} - \mathbf{H}\mathbf{K})^n = \mathbf{G}^n - \mathbf{G}^{n-1}\mathbf{H}\mathbf{K} - \cdots - \mathbf{H}\mathbf{K}\hat{\mathbf{G}}^{n-1} \end{aligned}$$

Multiplicando las ecuaciones anteriores por  $\alpha_n, \alpha_{n-1}, \dots, \alpha_0$  donde  $\alpha_0 = 1$ .

Se obtiene

$$\begin{aligned} \alpha_n \mathbf{I} + \alpha_{n-1} \hat{\mathbf{G}} + \alpha_{n-2} \hat{\mathbf{G}}^2 + \cdots + \hat{\mathbf{G}}^n &= \alpha_n \mathbf{I} + \alpha_{n-1} \mathbf{G} \\ &+ \alpha_{n-2} \mathbf{G}^2 + \cdots + \mathbf{G}^n - \alpha_{n-1} \mathbf{H}\mathbf{K} - \alpha_{n-2} \mathbf{G}\mathbf{H}\mathbf{K} \\ &- \alpha_{n-2} \mathbf{H}\mathbf{K}\hat{\mathbf{G}} - \cdots - \mathbf{G}^{n-1} \mathbf{H}\mathbf{K} - \cdots - \mathbf{H}\mathbf{K}\hat{\mathbf{G}}^{n-1} \end{aligned}$$

reescribiendo

$$\begin{aligned}
 (\hat{\mathbf{G}}) &= \phi(\mathbf{G}) - \alpha_{n-1}\mathbf{HK} - \alpha_{n-2}\mathbf{GHK} - \alpha_{n-2}\mathbf{HK}\hat{\mathbf{G}} - \dots \\
 &\quad - \mathbf{HK}\hat{\mathbf{G}}^{n-1} - \mathbf{G}^{n-1}\mathbf{HK} \\
 &= \phi(\mathbf{G}) - \begin{bmatrix} \mathbf{H}:\mathbf{GH}:\dots:\mathbf{G}^{n-1}\mathbf{H} \end{bmatrix} \begin{bmatrix} \alpha_{n-1}\mathbf{K} + \alpha_{n-2}\mathbf{K}\hat{\mathbf{G}} + \dots + \mathbf{K}\hat{\mathbf{G}}^{n-1} \\ \alpha_{n-2}\mathbf{K} + \alpha_{n-3}\mathbf{K}\hat{\mathbf{G}} + \dots + \mathbf{K}\hat{\mathbf{G}}^{n-2} \\ \vdots \\ \mathbf{K} \end{bmatrix}
 \end{aligned} \tag{2.29}$$

Como

$$\phi(\hat{\mathbf{G}}) = \mathbf{0}$$

entonces

$$\phi(\mathbf{G}) = \begin{bmatrix} \mathbf{H}:\mathbf{GH}:\dots:\mathbf{G}^{n-1}\mathbf{H} \end{bmatrix} \begin{bmatrix} \alpha_{n-1}\mathbf{K} + \alpha_{n-2}\mathbf{K}\hat{\mathbf{G}} + \dots + \mathbf{K}\hat{\mathbf{G}}^{n-1} \\ \alpha_{n-2}\mathbf{K} + \alpha_{n-3}\mathbf{K}\hat{\mathbf{G}} + \dots + \mathbf{K}\hat{\mathbf{G}}^{n-2} \\ \vdots \\ \mathbf{K} \end{bmatrix} \tag{2.30}$$

Como el sistema es de estado completamente controlable, la inversa de la matriz

$$\begin{bmatrix} \mathbf{H}:\mathbf{GH}:\dots:\mathbf{G}^{n-1}\mathbf{H} \end{bmatrix}$$

existe. La forma de la Ecuación (2.30) puede cambiarse de la siguiente manera

$$\begin{bmatrix} \alpha_{n-1}\mathbf{K} + \alpha_{n-2}\mathbf{K}\hat{\mathbf{G}} + \dots + \mathbf{K}\hat{\mathbf{G}}^{n-1} \\ \alpha_{n-2}\mathbf{K} + \alpha_{n-3}\mathbf{K}\hat{\mathbf{G}} + \dots + \mathbf{K}\hat{\mathbf{G}}^{n-2} \\ \vdots \\ \mathbf{K} \end{bmatrix} = \begin{bmatrix} \mathbf{H}:\mathbf{GH}:\dots:\mathbf{G}^{n-1}\mathbf{H} \end{bmatrix}^{-1} \phi(\mathbf{G})$$

Multiplicando ambos miembros de la última ecuación por  $[0 \ 0 \ \dots \ 0 \ 1]$

$$\begin{aligned}
 [0 \ 0 \ \dots \ 0 \ 1] &\begin{bmatrix} \alpha_{n-1}\mathbf{K} + \alpha_{n-2}\mathbf{K}\hat{\mathbf{G}} + \dots + \mathbf{K}\hat{\mathbf{G}}^{n-1} \\ \alpha_{n-2}\mathbf{K} + \alpha_{n-3}\mathbf{K}\hat{\mathbf{G}} + \dots + \mathbf{K}\hat{\mathbf{G}}^{n-2} \\ \vdots \\ \mathbf{K} \end{bmatrix} \\
 &= [0 \ 0 \ \dots \ 0 \ 1] \begin{bmatrix} \mathbf{H}:\mathbf{GH}:\dots:\mathbf{G}^{n-1}\mathbf{H} \end{bmatrix}^{-1} \phi(\mathbf{G})
 \end{aligned}$$

Simplificando, finalmente se obtiene

$$\mathbf{K} = [ 0 \ 0 \ \dots \ 0 \ 1 ] [ \mathbf{H} : \mathbf{GH} : \dots : \mathbf{G}^{n-1} \mathbf{H} ]^{-1} \phi(\mathbf{G}) \quad (2.31)$$

La Ecuación (2.31), conocida como la fórmula de Ackermann, nos permite encontrar la matriz de realimentación del estado requerida,  $\mathbf{K}$ , para los polos inicialmente impuestos.

### 2.6.6. Observadores de estados

El diseño de controladores es claramente dependiente de los estados del sistema, en los cuales se basa la realimentación del estado para reducir el error en caso de presentarse una perturbación externa.

Así, entonces, existe la necesidad de conocer el valor de cada variable de estado del sistema para tiempo  $t > t_0$  con el fin de saber si dichas variables están en su punto de equilibrio o si es necesario realizar un ajuste para regresarlas a su posición original.

En la práctica, no todas las variables de estado están siempre disponibles para ser medidas de manera directa por lo que se necesita estimar el valor cuantificable de aquellas variables que no son medibles directamente. Esta estimación se conoce como observación.

El observador es un subsistema del sistema de control que realiza la estimación de las variables de estado no medibles a partir de las mediciones de las variables de control y de salida, ver Figura 2.15.



Figura 2.15: Observador de estado

El diseño de observadores está supeditado al criterio de observabilidad, por lo que se podrá realizar un observador si, y sólo si, se satisface la condición de observabilidad por parte del sistema en el cual se esté trabajando.

La Figura 2.16 muestra un controlador con observador de estados del sistema

$$\mathbf{x}(k+1) = \mathbf{G}\mathbf{x}(k) + \mathbf{H}\mathbf{u}(k) \quad (2.32)$$

$$\mathbf{y}(k) = \mathbf{C}\mathbf{x}(k) \quad (2.33)$$

donde los términos que intervienen en este sistema son los mismos hasta ahora aplicados, lo único que hay que recalcar es que la matriz  $\mathbf{G}$  es una matriz no singular, es decir, que puede satisfacer su propia ecuación característica.

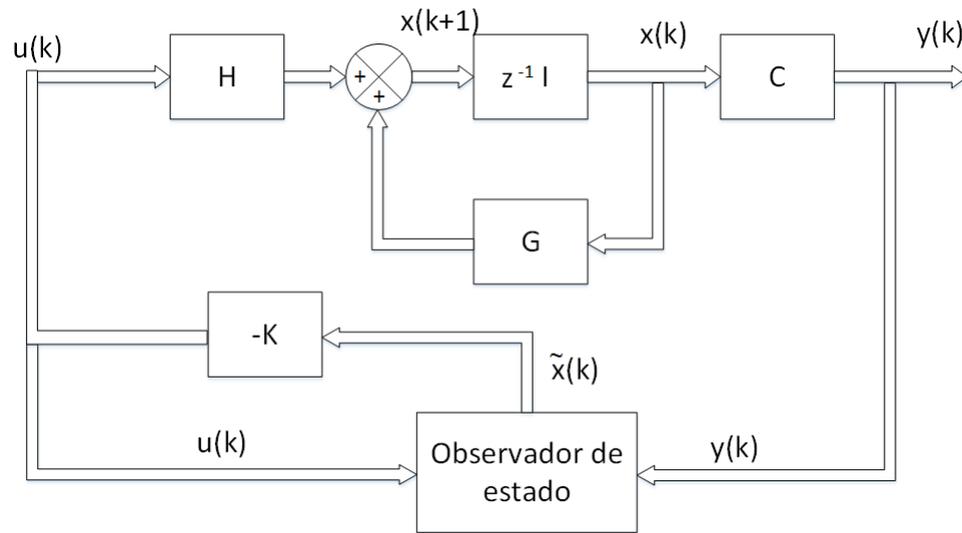


Figura 2.16: Sistema de regulación con un observador de estados

Matemáticamente, la observabilidad significa que se puede determinar  $\mathbf{x}(k+1)$  a partir de  $\mathbf{y}(k), \mathbf{y}(k-1), \dots, \mathbf{y}(k-n+1)$  y  $\mathbf{u}(k), \mathbf{u}(k-1), \dots, \mathbf{u}(k-n+1)$  siempre y cuando se demuestre primero, como se mencionó anteriormente, que el sistema es completamente controlable.

El hecho de que el estado  $\mathbf{x}(k)$  no puede medirse, por lo que se realiza una estimación del mismo  $\tilde{\mathbf{x}}(k)$  a partir de la observación entre sus señales de control y salida, la estimación debe ser muy aproximada a la real, por lo que es conveniente que las matrices  $\mathbf{G}$  y  $\mathbf{H}$  sean las mismas y/o iguales a las del sistema original. Entonces, es posible comparar la salida  $\tilde{\mathbf{y}}(k)$  con  $\mathbf{y}(k)$ , dado que la salida  $\tilde{\mathbf{y}}(k)$  si puede medirse.

El sistema de la Figura 2.14, está determinado por las ecuaciones

$$\mathbf{x}(k+1) = \mathbf{G}\mathbf{x}(k) + \mathbf{H}\mathbf{u}(k) \quad (2.34)$$

$$\mathbf{y}(k) = \mathbf{C}\mathbf{x}(k) \quad (2.35)$$

$$\mathbf{u}(k) = -\mathbf{K}\mathbf{x}(k)$$

y cuyas condiciones de controlabilidad y observabilidad están totalmente cubiertas, pero no es posible medirlo de forma directa. A partir de este inconveniente,

se añade un observador de estados incorporado que se muestra en la Figura 2.17. Se trabaja con el vector de estado estimado, por lo que la señal de control es igual

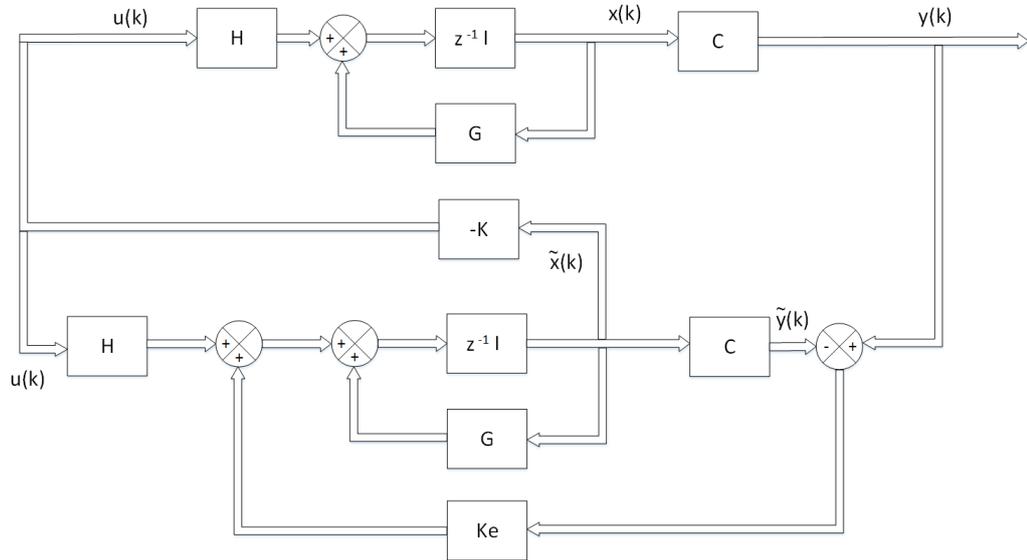


Figura 2.17: Sistema de control con realimentación de estado observado

a

$$\mathbf{u}(k) = -\mathbf{K}\tilde{\mathbf{x}}(k) \quad (2.36)$$

De la Figura 2.17 se tiene que

$$\tilde{\mathbf{x}}(k+1) = \mathbf{G}\tilde{\mathbf{x}}(k) + \mathbf{H}\mathbf{u}(k) + \mathbf{K}_e[\mathbf{y}(k) - \tilde{\mathbf{y}}(k)] \quad (2.37)$$

donde  $\mathbf{K}_e$  es la matriz de realimentación del observador.

Debido a que

$$\tilde{\mathbf{y}}(k) = \mathbf{C}\tilde{\mathbf{x}}(k)$$

la Ecuación (2.37) se puede reescribir de la siguiente forma

$$\tilde{\mathbf{x}}(k+1) = (\mathbf{G} - \mathbf{K}_e\mathbf{C})\tilde{\mathbf{x}}(k) + \mathbf{H}\mathbf{u}(k) + \mathbf{K}_e\mathbf{y}(k) \quad (2.38)$$

La matriz de realimentación la podemos encontrar nuevamente con la fórmula de Ackermann

$$\mathbf{K}_e = \phi(\mathbf{G}) \begin{bmatrix} \mathbf{C} \\ \mathbf{C}\mathbf{G} \\ \vdots \\ \mathbf{C}\mathbf{G}^{n-1} \end{bmatrix}^{-1} \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 1 \end{bmatrix} \quad (2.39)$$



donde

$\phi(\mathbf{G}) =$  polinomio característico deseado de la dinámica del error.

“El observador de estados dado por la Ecuación (2.38) se llama observador de predicción, pues el estimado  $\mathbf{x}(k+1)$  está un período de muestreo adelante de la medición  $\mathbf{y}(k)$ . Los valores característicos de  $\mathbf{G} - \mathbf{K}_e\mathbf{G}$  suelen conocerse como polos del observador” [28].

### 2.6.7. Sistemas de seguimiento

Un seguidor es un sistema de control realimentado que se encarga de mantener la variable controlada muy próxima a la variable de referencia en un sistema que cambia frecuentemente dicha referencia.

En un sistema de seguimiento, por lo general, se necesitan agregar bloques integradores que reduzcan o, si es posible, anulen el error de estado permanente producto del cambio constante del estado del sistema, que a su vez, es consecuencia directa de la frecuente alteración en la referencia.

En un sistema de lazo cerrado, se puede introducir el integrador agregando un nuevo vector de estado, que integre el vector de comando  $\mathbf{R}$  -referencia- y el vector de salida  $\mathbf{Y}$ .

#### 2.6.7.1. Sistema de seguimiento con integrador

Se considera el sistema en espacio de estados, Ecuación (2.32), donde se supone que la planta es completamente controlable y completamente observable, sin embargo, no posee un elemento integrador.

En la Figura 2.18, se presenta un diagrama de bloques al cual se le ha añadido un integrador del error de estado estable, cuya ecuación de estado es

$$\mathbf{v}(k) = \mathbf{v}(k-1) + \mathbf{r}(k) - \mathbf{y}(k) \quad (2.40)$$

donde

$\mathbf{v}(k) =$  vector de error de actuación -dimensión  $m$ -  
 $\mathbf{r}(k) =$  vector de entrada de comando -dimensión  $m$ -

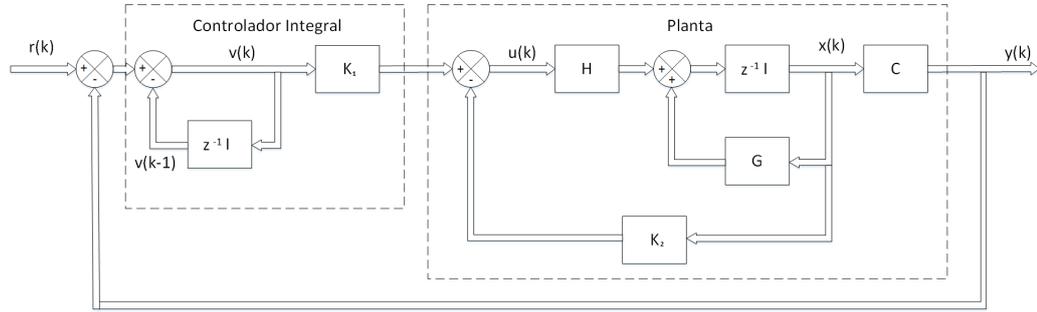


Figura 2.18: Sistema de seguimiento con integrador

La Ecuación (2.40) se puede reescribir como

$$\begin{aligned}
 \mathbf{v}(k+1) &= \mathbf{v}(k) + \mathbf{r}(k+1) - \mathbf{y}(k+1) \\
 &= \mathbf{v}(k) + \mathbf{r}(k+1) - \mathbf{C}[\mathbf{G}\mathbf{x}(k) + \mathbf{H}\mathbf{u}(k)] \\
 &= -\mathbf{C}\mathbf{G}\mathbf{x}(k) + \mathbf{v}(k) - \mathbf{C}\mathbf{H}\mathbf{u}(k) + \mathbf{r}(k+1)
 \end{aligned} \tag{2.41}$$

y el vector de control es

$$\mathbf{u}(k) = -\mathbf{K}_2\mathbf{x}(k) + \mathbf{K}_1\mathbf{v}(k) \tag{2.42}$$

Los parámetros de diseño son los vectores  $\mathbf{K}_1$  y  $\mathbf{K}_2$ .

### 2.6.7.2. Procedimiento para encontrar $\mathbf{K}_1$ y $\mathbf{K}_2$

La determinación de las matrices  $\mathbf{K}_1$  y  $\mathbf{K}_2$  se realiza tomando en consideración los polos deseados en lazo cerrado.

De las Ecuaciones (2.32), (2.41) y (2.42) se obtiene

$$\begin{aligned}
 \mathbf{u}(k+1) &= -\mathbf{K}_2\mathbf{x}(k+1) + \mathbf{K}_1\mathbf{v}(k+1) \\
 &= (\mathbf{K}_2 - \mathbf{K}_2\mathbf{G} - \mathbf{K}_1\mathbf{C}\mathbf{G})\mathbf{x}(k) + (\mathbf{I}_m - \mathbf{K}_2\mathbf{H} - \mathbf{K}_1\mathbf{C}\mathbf{H})\mathbf{u}(k) \\
 &\quad + \mathbf{K}_1\mathbf{r}(k+1)
 \end{aligned} \tag{2.43}$$

Como  $\mathbf{u}(k)$  es una combinación lineal de los vectores de estado  $\mathbf{x}(k)$  y  $\mathbf{v}(k)$ . Se puede formar un nuevo vector de estado conformado por  $\mathbf{x}(k)$  y  $\mathbf{u}(k)$ , por lo que se puede obtener una nueva ecuación de estado con su respectiva ecuación de

salida como sigue

$$\begin{bmatrix} \mathbf{x}(k+1) \\ \mathbf{u}(k+1) \end{bmatrix} = \begin{bmatrix} \mathbf{G} & \mathbf{H} \\ \mathbf{K}_2 - \mathbf{K}_2\mathbf{G} - \mathbf{K}_1\mathbf{C}\mathbf{G} & \mathbf{I}_m - \mathbf{K}_2\mathbf{H} - \mathbf{K}_1\mathbf{C}\mathbf{H} \end{bmatrix} \begin{bmatrix} \mathbf{x}(k) \\ \mathbf{u}(k) \end{bmatrix} + \begin{bmatrix} \mathbf{0} \\ \mathbf{K}_1 \end{bmatrix} \mathbf{r}(k+1) \quad (2.44)$$

$$\mathbf{y}(k) = [\mathbf{C} \quad \mathbf{0}] \begin{bmatrix} \mathbf{x}(k) \\ \mathbf{u}(k) \end{bmatrix} \quad (2.45)$$

los polos del sistema están determinados por los valores característicos de la matriz de estado del sistema, siendo independientes de la entrada de comando  $\mathbf{r}(k)$ .

Se considera una entrada escalón, por lo que el vector de comando es constante

$$\mathbf{r}(k) = \mathbf{r}$$

entonces

$$\begin{bmatrix} \mathbf{x}(k+1) \\ \mathbf{u}(k+1) \end{bmatrix} = \begin{bmatrix} \mathbf{G} & \mathbf{H} \\ \mathbf{K}_2 - \mathbf{K}_2\mathbf{G} - \mathbf{K}_1\mathbf{C}\mathbf{G} & \mathbf{I}_m - \mathbf{K}_2\mathbf{H} - \mathbf{K}_1\mathbf{C}\mathbf{H} \end{bmatrix} \begin{bmatrix} \mathbf{x}(k) \\ \mathbf{u}(k) \end{bmatrix} + \begin{bmatrix} \mathbf{0} \\ \mathbf{K}_1\mathbf{r} \end{bmatrix} \quad (2.46)$$

Debido a que, para una entrada escalón,  $\mathbf{x}(k)$ ,  $\mathbf{u}(k)$  y  $\mathbf{v}(k)$  tienden a los valores de vector constantes  $\mathbf{x}(\infty)$ ,  $\mathbf{u}(\infty)$  y  $\mathbf{v}(\infty)$ , respectivamente. Por tanto

$$\mathbf{v}(\infty) = \mathbf{v}(\infty) + \mathbf{r} - \mathbf{y}(\infty)$$

es decir,

$$\mathbf{y}(\infty) = \mathbf{r}$$

En estado permanente la Ecuación (2.46) se convierte en

$$\begin{bmatrix} \mathbf{x}(\infty) \\ \mathbf{u}(\infty) \end{bmatrix} = \begin{bmatrix} \mathbf{G} & \mathbf{H} \\ \mathbf{K}_2 - \mathbf{K}_2\mathbf{G} - \mathbf{K}_1\mathbf{C}\mathbf{G} & \mathbf{I}_m - \mathbf{K}_2\mathbf{H} - \mathbf{K}_1\mathbf{C}\mathbf{H} \end{bmatrix} \begin{bmatrix} \mathbf{x}(\infty) \\ \mathbf{u}(\infty) \end{bmatrix} + \begin{bmatrix} \mathbf{0} \\ \mathbf{K}_1\mathbf{r} \end{bmatrix} \quad (2.47)$$



Los vectores de error están dados por

$$\begin{aligned}\mathbf{x}_e(k) &= \mathbf{x}(k) - \mathbf{x}(\infty) \\ \mathbf{u}_e(k) &= \mathbf{u}(k) - \mathbf{u}(\infty)\end{aligned}$$

Restando la Ecuación (2.46) menos la Ecuación (2.47)

$$\begin{bmatrix} \mathbf{x}_e(k+1) \\ \mathbf{u}_e(k+1) \end{bmatrix} = \begin{bmatrix} \mathbf{G} & \mathbf{H} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{x}_e(k) \\ \mathbf{u}_e(k) \end{bmatrix} + \begin{bmatrix} \mathbf{0} \\ \mathbf{I}_m \end{bmatrix} \mathbf{w}(k) \quad (2.48)$$

donde

$$\mathbf{w}(k) = [ \mathbf{K}_2 - \mathbf{K}_2\mathbf{G} - \mathbf{K}_1\mathbf{C}\mathbf{G} \quad ; \quad \mathbf{I}_m - \mathbf{K}_2\mathbf{H} - \mathbf{K}_1\mathbf{C}\mathbf{H} ] \begin{bmatrix} \mathbf{x}_e(k) \\ \mathbf{u}_e(k) \end{bmatrix} \quad (2.49)$$

Definiendo

$$\begin{aligned}\xi(k) &= \begin{bmatrix} \mathbf{x}_e(k) \\ \mathbf{u}_e(k) \end{bmatrix} \\ \hat{\mathbf{G}} &= \begin{bmatrix} \mathbf{G} & \mathbf{H} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \\ \hat{\mathbf{H}} &= \begin{bmatrix} \mathbf{0} \\ \mathbf{I}_m \end{bmatrix} \\ \hat{\mathbf{K}} &= -[ \mathbf{K}_2 - \mathbf{K}_2\mathbf{G} - \mathbf{K}_1\mathbf{C}\mathbf{G} ; \mathbf{I}_m - \mathbf{K}_2\mathbf{H} - \mathbf{K}_1\mathbf{C}\mathbf{H} ]\end{aligned} \quad (2.50)$$

Las ecuaciones se transforman en

$$\xi(k+1) = \hat{\mathbf{G}}\xi(k) + \hat{\mathbf{H}}\mathbf{w}(k) \quad (2.51)$$

$$\mathbf{w}(k) = -\hat{\mathbf{K}}\xi(k) \quad (2.52)$$

La matriz  $\hat{\mathbf{K}}$  se puede determinar mediante la técnica de ubicación de polos, una vez que se hayan determinado los polos deseados en lazo cerrado. Entonces, podemos determinar las matrices  $\mathbf{K}_1$  y  $\mathbf{K}_2$  como se detalla a continuación. Se observa que

$$[ \mathbf{K}_2 ; \mathbf{K}_1 ] = \begin{bmatrix} \mathbf{G} - \mathbf{I}_n & \mathbf{H} \\ \mathbf{C}\mathbf{G} & \mathbf{C}\mathbf{H} \end{bmatrix} = [ \mathbf{K}_2\mathbf{G} - \mathbf{K}_2 + \mathbf{K}_1\mathbf{C}\mathbf{G} ; \mathbf{K}_2\mathbf{H} + \mathbf{K}_1\mathbf{C}\mathbf{H} ] \quad (2.53)$$

De las ecuaciones 2.50 y 2.53 se obtiene

$$\hat{\mathbf{K}} = [ \mathbf{K}_2 \mathbf{G} - \mathbf{K}_2 + \mathbf{K}_1 \mathbf{C} \mathbf{G} : - \mathbf{I}_m + \mathbf{K}_2 \mathbf{H} + \mathbf{K}_1 \mathbf{C} \mathbf{H} ]$$

$$\hat{\mathbf{K}} = [ \mathbf{K}_2 : \mathbf{K}_1 ] \begin{bmatrix} \mathbf{G} - \mathbf{I}_n & \mathbf{H} \\ \mathbf{C} \mathbf{G} & \mathbf{C} \mathbf{H} \end{bmatrix} + [ \mathbf{0} : - \mathbf{I}_m ]$$

Por lo tanto, las matrices  $\mathbf{K}_1$  y  $\mathbf{K}_2$  pueden determinarse a partir de la siguiente ecuación

$$[ \mathbf{K}_2 : \mathbf{K}_1 ] = \left( \hat{\mathbf{K}} + [ \mathbf{0} : - \mathbf{I}_m ] \right) \begin{bmatrix} \mathbf{G} - \mathbf{I}_n & \mathbf{H} \\ \mathbf{C} \mathbf{G} & \mathbf{C} \mathbf{H} \end{bmatrix}^{-1} \quad (2.54)$$

Cuando  $\mathbf{u}(k)$  es de dimensión  $m$ , y  $m > 1$ , se pueden obtener más de un conjunto de matrices  $\mathbf{K}_1$  y  $\mathbf{K}_2$ , por lo que se deberá elegir el conjunto que mejor desempeño agregue al sistema.

## 2.7. Control PID neuronal

Las redes neuronales son algoritmos o sistemas de procesamiento que tienen por objetivo asemejarse de la manera más aproximada posible al funcionamiento del cerebro humano, mediante la interconexión de neuronas.

La salida producida por dichas neuronas depende del tipo de procesamiento llevado a cabo en la salida del sistema, también depende de la experiencia y aprendizaje que van obteniendo los sistemas a través de la práctica mediante los llamados métodos de aprendizaje.

Hoy en día, existen diferentes sistemas de procesamiento que incluyen redes neuronales, sin embargo, desde los primeros años de desarrollo aparecieron el perceptrón simple y el ADALINE –por sus siglas en inglés, Adaptive Linear Neuron–.

El perceptrón simple utiliza una función de salida y/o activación, Ecuación (2.55), produciendo valores discretos y realizando una especie de clasificación a la salida, por lo que si existieran salidas con números decimales este procedimiento no es útil.

$$S(t+1) = \begin{cases} 1 & \sum_i w_i x_i(y) > \theta \\ 0 & \text{caso contrario} \end{cases} \quad (2.55)$$



### 2.7.1. ADALINE

El ADALINE se encarga de eliminar el inconveniente de no poder trabajar con salidas con números decimales, ya que la manera de utilizar la regla de aprendizaje en la salida es diferente a la del perceptrón, por lo que, este será el modelo estudiado para el control del interfaz háptico.

Mientras el perceptrón considera solo si se ha equivocado o no, el ADALINE considera cuánto se ha equivocado, realimentando el sistema mediante la tasa de error que existe entre la salida deseada y la medida. Esto que se acaba de mencionar, nos da una perspectiva de que este algoritmo se puede utilizar para el control de una planta, ya que su lógica es idéntica a la de un controlador convencional, PID.

La salida  $y$  de la neurona es

$$\tilde{y} = \sum_{i=1}^n w_i x_i + \theta \quad (2.56)$$

Si tenemos un patrón de entrada  $\mathbf{x}^p$  y una salida  $\mathbf{d}^p$ , el error producido es

$$e = |\mathbf{d}^p - \mathbf{x}^p| \quad (2.57)$$

Esta es la regla de aprendizaje conocida como la regla Delta.

El objetivo del ADALINE es obtener un valor único  $y = d^p$  para un conjunto determinado de valores  $x_i^p$  que se introducen a la entrada donde  $i = 1, 2, \dots, n$ .

La manera de obtener  $y = d^p$  es hallar los pesos  $w_i$  presentes en la Ecuación (2.56), donde  $i = 1, 2, \dots, n$ .

Los pesos  $w_i$  inician con valores aleatorios entre -1 y 1, la regla de aprendizaje, Delta, consiste en actualizar estos pesos hasta lograr los valores óptimos de los mismos que puedan producir  $y = d^p$ .

La actualización de los pesos se basa en multiplicar el error actual con las entradas actuales y se la realiza mediante la fórmula

$$\Delta_p w_j = \gamma (d^p - x^p) x_j \quad (2.58)$$

donde  $\gamma$  es la tasa de aprendizaje, de la cual depende la rápida convergencia, o no, del proceso, mediante la ponderación, esta tasa de aprendizaje se encuentra entre 0 y 1.

Una vez que se obtienen los pesos ponderados actuales, se suma algebraica-



mente con los pesos anteriores, logrando actualizar el vector de pesos, Ecuación (2.59).

$$w_j^p = w_j^{p-1} + \Delta_p w_j \quad (2.59)$$

### 2.7.2. Controlador utilizando la red neuronal ADALINE

Tomando en cuenta la Ecuación (2.6) que modela la salida discreta del PID

$$u_k = u_{k-1} + k_c \left[ \left( 1 + \frac{T}{T_i} + \frac{T_d}{T} \right) e_k - \left( 1 + 2\frac{T_d}{T} \right) e_{k-1} + \frac{T_d}{T} e_{k-2} \right]$$

Se observa que la salida depende del error actual,  $e_k$ , del error de hace una muestra,  $e_{k-1}$ , y del error de hace dos muestras,  $e_{k-2}$ . Por lo tanto, podemos tener un vector de error

$$\mathbf{E} = \begin{bmatrix} e_k \\ e_{k-1} \\ e_{k-2} \end{bmatrix}$$

Dicho esto, la Ecuación (2.6), quedaría

$$u_k = u_{k-1} + k_c \left[ \left( 1 + \frac{T}{T_i} + \frac{T_d}{T} \right) e_k - \left( 1 + 2\frac{T_d}{T} \right) e_{k-1} + \frac{T_d}{T} e_{k-2} \right] \quad (2.60)$$

Ya que los valores de los parámetros del PID, así como el tiempo de muestreo,  $T$ , no son aptos en este tipo de control, podemos considerarlos constantes, por lo que la Ecuación (2.60) podemos reescribirla como

$$u_k = u_{k-1} + w_1 e_k + w_2 e_{k-1} + w_3 e_{k-2} \quad (2.61)$$

Se puede considerar

$$\mathbf{w} = \begin{bmatrix} w_1 \\ w_2 \\ w_3 \end{bmatrix}$$

Entonces

$$u_k = u_{k-1} + \mathbf{w}^T \mathbf{E} \quad (2.62)$$

Como se observa en la Ecuación (2.62), la salida actual del controlador depende únicamente del vector  $\mathbf{w}$  y del vector de error  $\mathbf{E}$ , así como de la muestra de su última salida,  $u_{k-1}$ .

Claramente se puede notar que la Ecuación (2.62) es idéntica a la Ecuación (2.56), por lo que podemos decir que el vector  $\mathbf{w}$  funciona como un vector de



## CAPÍTULO 2. FUNDAMENTOS TEÓRICOS

---

pesos.

Uno de los aspectos más importantes del nuevo controlador es que ya no depende de los parámetros del PID, así como tampoco del tiempo de muestreo.



## Capítulo 3

---

# Construcción de una pala háptica

---

### 3.1. Interfaz háptica, componentes y montaje

El diseño utilizado es de la Universidad de Vanderbilt, que a su vez, es una mejora del diseño original que fue desarrollado y usado como herramienta de enseñanza por parte de la Universidad de Stanford. Este diseño está disponible *-open hardware-* en su sitio web [29]. El diseño original cuenta con una licencia tipo *Creative Commons Attribution-ShareAlike 3.0 Unported License*.

La interfaz háptica posee un grado de libertad y es de tipo *'joystick'* –kinestésico, tipo impedancia-, misma que se puede utilizar para diferentes tipos de aplicaciones, en lo que a sistemas dinámicos se refiere, generando diferentes fuerzas en la interacción con ambientes virtuales.

El sistema consta de diferentes partes modulares fabricadas en acrílico de dos tipos de espesores, además, de componentes mecánicos y electrónicos, los cuales se detallan en las siguientes secciones. La adquisición y procesamiento de datos en la entrada y salida de la planta se la realiza mediante una placa *Arduino Uno*; la interfaz también se compone de un motor de corriente directa (CD) con su respectivo *driver* de control, *Ardumoto Shield*.

#### 3.1.1. Arduino

Es una plataforma de código abierto, tanto de hardware y software, compuesta por entradas y salidas analógicas y digitales, utilizadas para la lectura de sensores y control de diferentes componentes como motores, luces y demás, ver Figura 3.1.

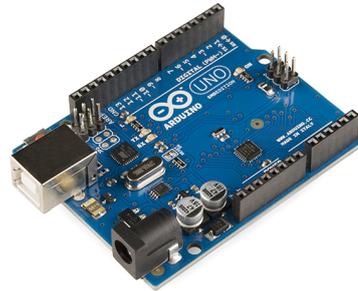


Figura 3.1: Plataforma Arduino<sup>1</sup>

Esta plataforma se puede utilizar para la adquisición de datos y posterior comunicación con un ordenador, ayudando así, al análisis de diferentes sistemas en estudio que corren en el computador. Se utiliza el *Arduino Programming Language* y el *Arduino Development Environment* para la programación del microcontrolador de la placa.

Su bajo costo y flexibilidad para trabajar con diferentes plataformas, además de su clara y, relativamente, fácil programación son las ventajas que esta plataforma ofrece frente a otras plataformas más complejas.

### 3.1.2. Ardumoto Shield

Es un controlador de cargas inductivas para Arduino, el cual, puede manejar fácilmente dos motores de CD por separado, ver Figura 3.2. Cada control consta de un inversor de giro y puede trabajar con altos voltajes y corrientes.

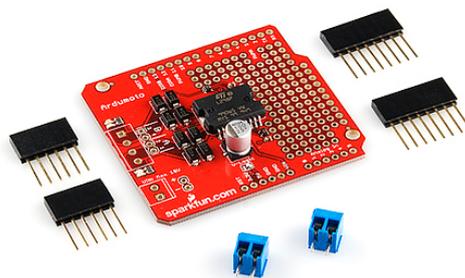


Figura 3.2: Ardumoto Shield<sup>2</sup>

Todas estas características se logran gracias al driver de motor L298 que está incluido en la placa del Ardumoto. Específicamente, este driver tolera grandes

<sup>1</sup><http://diegorys.es/2014/10/02/primeros-pasos-con-arduino/>

<sup>2</sup><https://learn.sparkfun.com/tutorials/ardumoto-shield-hookup-guide#board-overview>

voltajes y corrientes que pueden llegar hasta los 2A, y acepta niveles lógicos TTL para la habilitar o deshabilitar el dispositivo. Puede manejar cargas inductivas como relés, solenoides, motores de paso y de CD, ver Anexo A.

El Ardumoto Shield consta de una entrada exclusiva para la alimentación con una fuente externa de CD y que es alternativa a la que provee la placa Arduino. Esta fuente sirve como soporte para elevar la corriente hasta un punto óptimo con el que trabajan las cargas conectadas al Ardumoto. En el caso de los motores, el aumento de la corriente causa directamente un aumento en el torque del motor, en consecuencia, se puede obtener un mayor rango de fuerzas a lo que se produjeran si se utilizara únicamente la fuente del Arduino. La Figura 3.3 muestra la distribución de pines en la placa del Ardumoto.

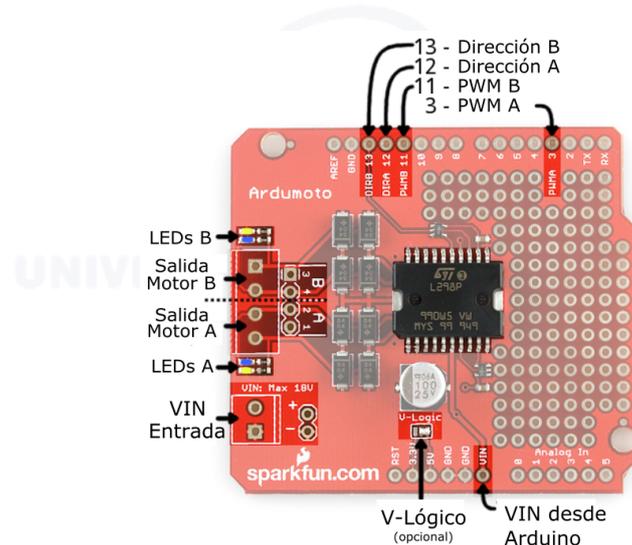


Figura 3.3: Distribución de pines del Ardumoto Shield<sup>3</sup>

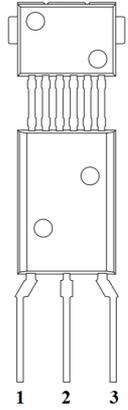
Utiliza un PWM con una resolución de 8 bits para el control de velocidad de los motores. Las características de control, tanto de dirección como de velocidad, así como los pines del Ardumoto que se utilizan, se muestran en la Tabla 3.1.

Tabla 3.1: Distribución de pines de control del Ardumoto Shield

Pines Arduino	Función Ardumoto	Control
3	PWM A	0 = apagado, 255 = máx. velocidad
11	PWM B	0 = apagado, 255 = máx. velocidad
12	Dirección A	Señal Digital de Control (Uno/Cero = Derecha/Izquierda)
13	Dirección B	Señal Digital de Control (Uno/Cero = Derecha/Izquierda)

<sup>3</sup><https://learn.sparkfun.com/tutorials/ardumoto-shield-hookup-guide#board-overview>

Tabla 3.2: Distribución de pines del sensor magnetoresistivo KMA-210

Pin	Símbolo	Descripción	Esquema simplificado
1	$V_{DD}$	alimentación	
2	GND	tierra	
3	Salida/Datos	salida analógica o interfaz de datos	

### 3.1.3. Sensor Magnetoresistivo KMA-210

Este sensor varía su resistencia eléctrica en función del desplazamiento de ángulo de las líneas de campo magnético, paralelas al plano del sensor, a las cuales se encuentra sometido, y que son generadas por un imán.

El sensor magnetoresistivo está compuesto por un sensor magnético y un circuito integrado. La distribución de pines se encuentra especificada en la Tabla 3.2.

La salida analógica del sensor es periódica cada  $180^\circ$ . El ángulo,  $\alpha$ , es medido de acuerdo a la Figura 3.4, donde la línea entrecortada indica la posición cero.

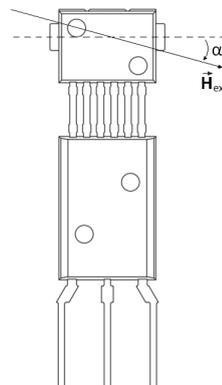


Figura 3.4: Ángulo medido por el sensor magnetoresistivo KMA-210

### 3.1.4. Actuador

Existen algunos tipos de actuadores que dependen del tipo de potencia que se necesite. Están los actuadores hidráulicos, neumáticos y motores.

En el caso de la interfaz háptica, un motor de CD, Maxon 353144, ver Figura 3.5, es utilizado para ejercer el rango de fuerzas de realimentación, necesarias dentro de las propiedades con las que debe cumplir la interfaz, su hoja de especificaciones se encuentra en el Anexo B.



Figura 3.5: Motor utilizado en la interfaz háptica<sup>4</sup>

### 3.1.5. Módulos de la interfaz háptica

#### 3.1.5.1. Base

Soporta todos los demás componentes. Su espesor es de 12mm, este espesor permite brindar estabilidad a la interfaz al momento de su utilización, ver Figura 3.6.

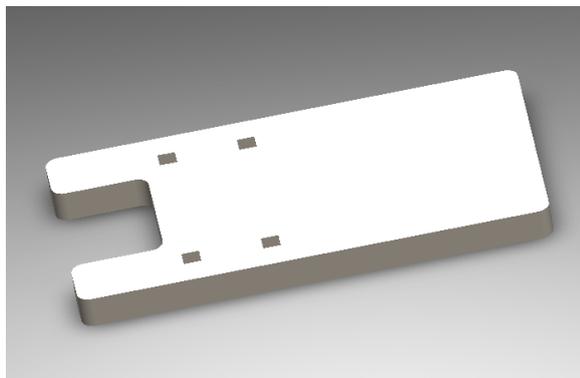


Figura 3.6: Base de la interfaz háptica

<sup>4</sup><http://www.maxonmotor.es/maxon/view/content/products>

### 3.1.5.2. Módulo Frontal

Brinda las facilidades para la sujeción de pala háptica, su espesor es de 6mm. Consta de un mecanismo de ajuste para dicha pala en la parte superior, ver Figura 3.7. También, a este módulo se sujeta el motor de CD.



Figura 3.7: Módulo frontal de la interfaz háptica

### 3.1.5.3. Soportes

Ayudan a la estabilización entre el módulo frontal y la base, dando mayor firmeza a la interfaz, ver Figura 3.8. Como el módulo frontal, esta parte también es de 6mm de espesor.

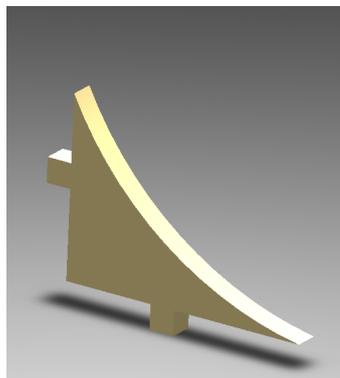


Figura 3.8: Soportes para la estabilización del módulo frontal sobre la base de la interfaz háptica

#### 3.1.5.4. Soporte para la Pala

Este módulo cuenta con un agujero central por donde pasa el eje que contiene a la pala y cuenta con un espesor de 6mm, ver Figura 3.9. Se sujeta al módulo frontal con ayuda de tornillos, permite la calibración de la fuerza de fricción entre la pala y la rueda colocada en el eje del motor.



Figura 3.9: Soporte para sujetar la pala háptica

#### 3.1.5.5. Rueda Motriz

Se trata de una rueda colocada en el eje del motor, sujeta mediante un prisionero al mismo, ver Figura 3.10. Transfiere la energía del motor hacia la pala, que a su vez la transmite al operador. Para su construcción se utilizó aluminio.

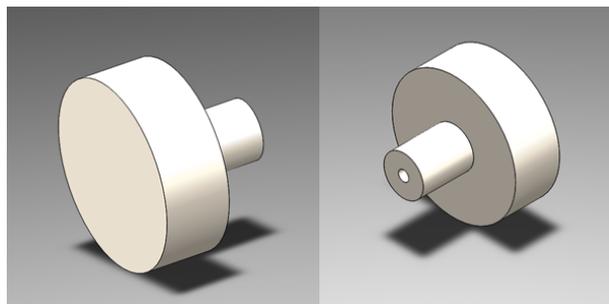


Figura 3.10: Rueda motriz de la interfaz háptica

#### 3.1.5.6. Pala

Se había mencionado, anteriormente, que la interfaz, al ser kinestésica de tipo impedancia, necesita una herramienta para la manipulación e interacción, la pala es, justamente, esa herramienta que permite trabajar con los ambientes virtuales. Es en la pala donde se siente la fuerza o torque proporcionado por el motor, producto de la interacción. El control de la interfaz se realiza con los datos de posición proporcionados, relativos a la pala, cuyo espesor es de 12mm, ver Figura

3.11. En la parte inferior de la misma se le coloca una banda para dar mayor tracción entre la pala y la rueda motriz.



Figura 3.11: Pala de la interfaz háptica

### 3.1.6. Montaje

Con todos los materiales y componentes mecánicos y electrónicos a disposición, el montaje resulta sencillo. Para la unión de las piezas de acrílico se utilizó pegamento epóxico. A continuación se detalla paso a paso el montaje de la interfaz háptica.

1. Se pegan los soportes al módulo frontal de la interfaz.
2. La pieza resultante del paso 1, se la junta con la base. Hasta aquí, ver Figura 3.12.

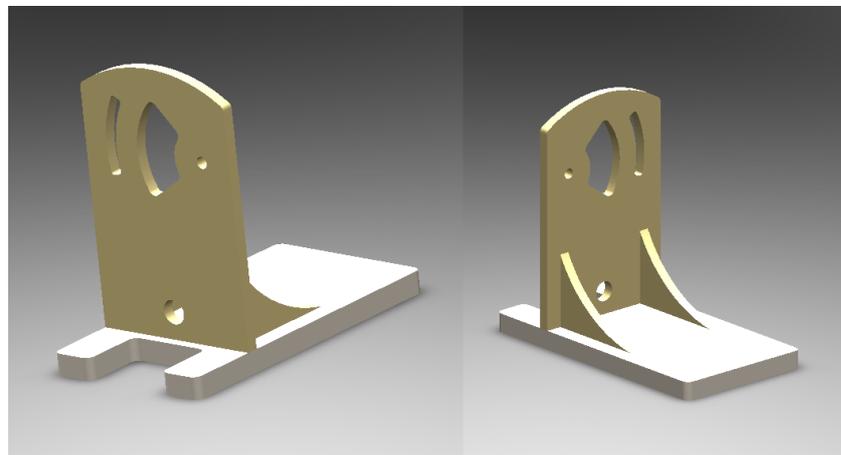


Figura 3.12: Montaje interfaz háptica, pasos 1 y 2

3. Se coloca el motor en su sitio y se lo asegura con unos tornillos en el módulo frontal.

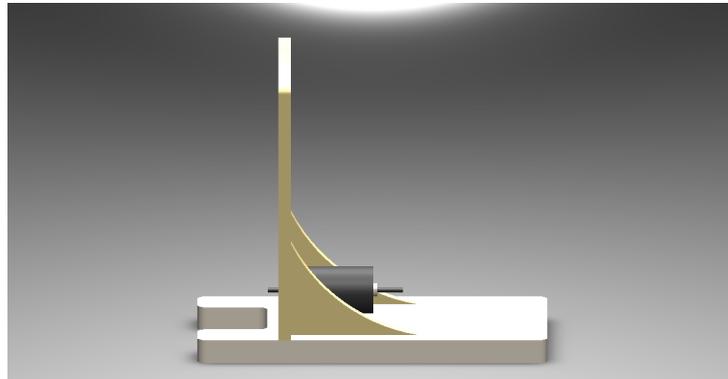


Figura 3.13: Montaje interfaz háptica, paso 3

4. Se pega el imán en el centro de la rueda motriz.

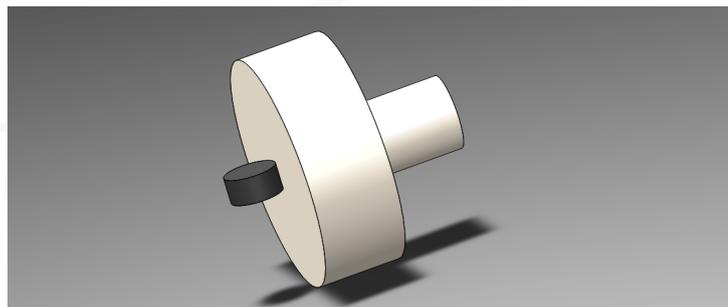


Figura 3.14: Montaje interfaz háptica, paso 4

5. Se coloca la rueda motriz al eje del motor, y se procede a sujetarla mediante un prisionero.

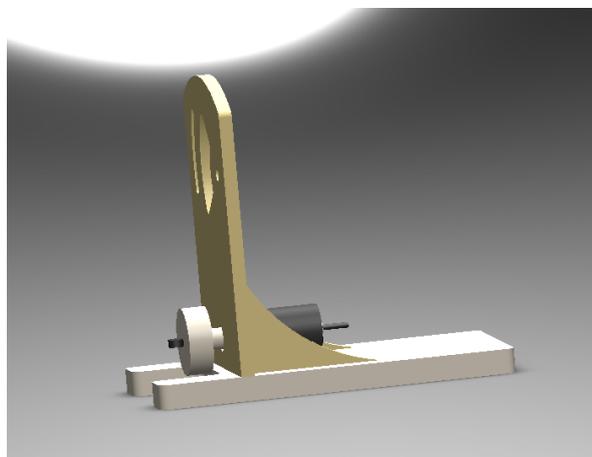


Figura 3.15: Montaje interfaz háptica, paso 5

6. Se atraviesa un eje por el soporte de la pala y por la propia pala. Se utiliza unos collarines para que la pala no se desplace por el eje, además de dejar un espacio razonable entre las dos piezas.

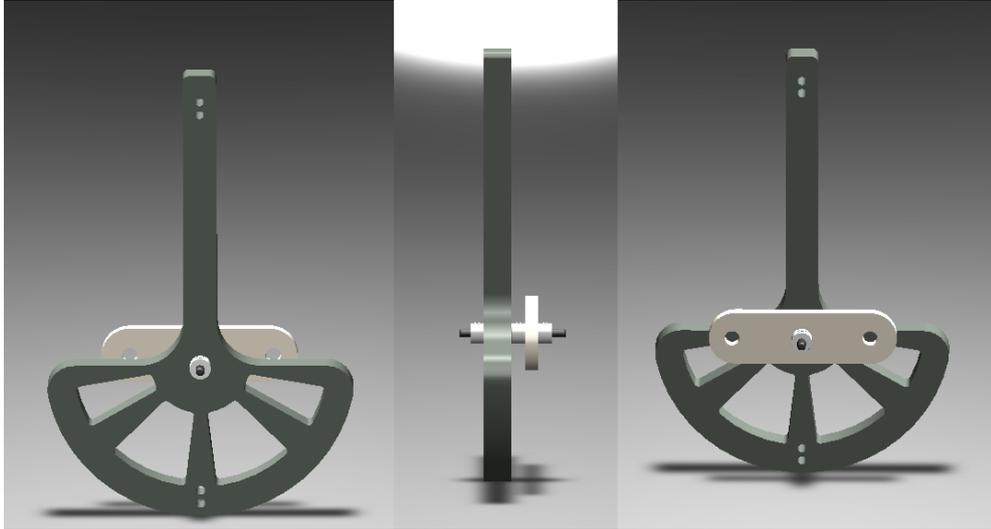


Figura 3.16: Montaje interfaz háptica, paso 6

7. Se sujeta el soporte de la pala en el módulo frontal mediante unos tornillos.

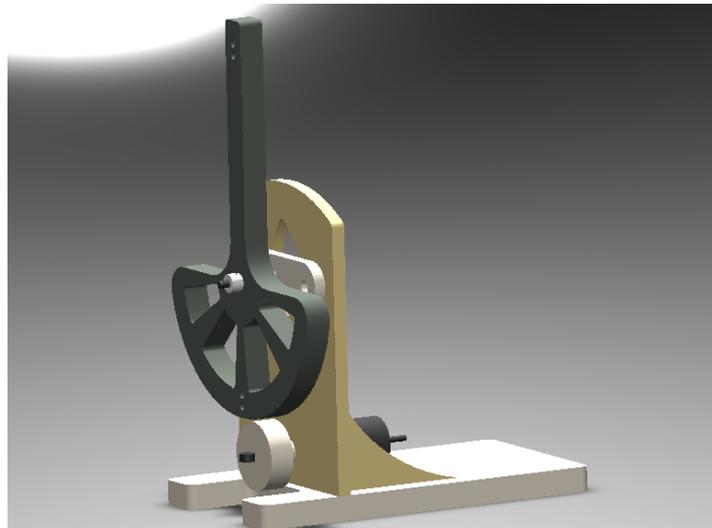


Figura 3.17: Montaje interfaz háptica, paso 7

8. Se pega el sensor magnetoresistivo a la pieza de acrílico correspondiente, ver Figura 3.18, y se la ajusta, antes de atornillarla, ubicando el sensor en frente del eje del imán colocado anteriormente en la rueda motriz.

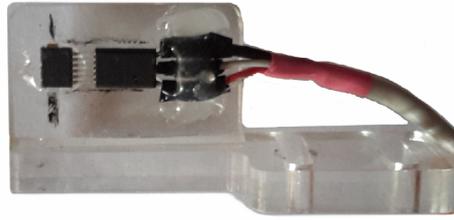


Figura 3.18: Montaje interfaz háptica, paso 8

9. Finalmente, se atornilla la placa Arduino en la base de la interfaz, y también se coloca el Ardumoto Shield, de modo que, la interfaz, totalmente montada, queda como se observa en la Figura 3.19.

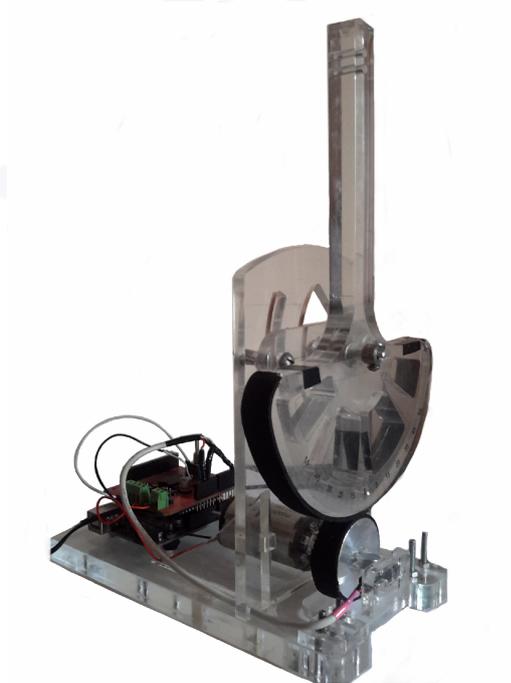


Figura 3.19: Montaje interfaz háptica, paso 9

## Capítulo 4

---

# Control de la interfaz háptica

---

### 4.1. Procesamiento de la señal de posición de la pala háptica

El control de la planta se realiza mediante la herramienta Simulink de MATLAB, y la adquisición de datos mediante la plataforma Arduino.

La plataforma Arduino realiza la lectura de los valores de voltaje entregados por el sensor magnetoresistivo en la entrada analógica A4. El sensor es alimentado por un voltaje de 5V.

La Figura 4.1 muestra la señal de voltaje que proporciona el sensor magnetoresistivo; esta señal es lineal y periódica cada  $180^\circ$ . La discontinuidad que se observa en la señal se procesa mediante software directamente en la plataforma Arduino.

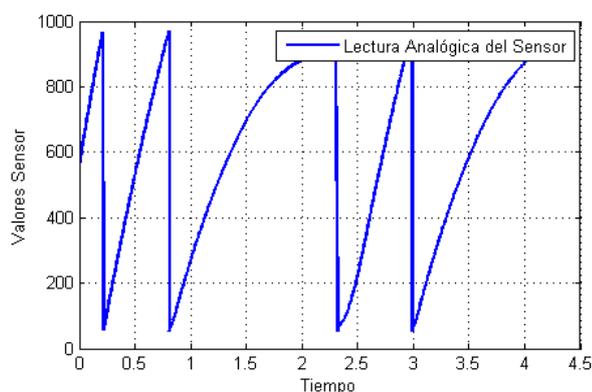


Figura 4.1: Señal a la salida del sensor magnetoresistivo

Las lecturas del sensor, procesadas en Arduino, se envían hacia el ordenador mediante comunicación serial *RS232* a una velocidad de 57600 baudios debido a

que la interfaz permite una interacción en tiempo real. De la misma manera, se realiza la recepción de datos desde el ordenador. El diagrama de flujo mostrado en la Figura 4.2, ilustra el desarrollo del algoritmo utilizado en la programación del procesador Arduino.



Figura 4.2: Diagrama de flujo utilizado en Arduino

#### 4.1.1. Factor de conversión cinemático, *bits-posición*

Para el control de posición se puede utilizar el ángulo de desplazamiento de la pala con respecto a su centro de rotación, o bien, el arco descrito por el mango de la pala, tomando como referencia su parte superior, ver Figura 4.3.

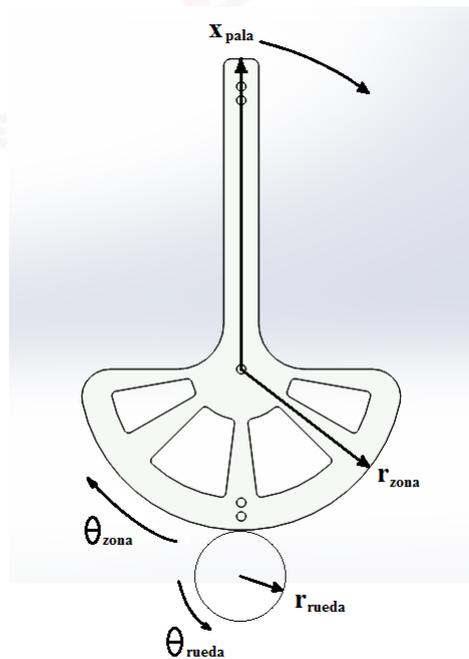


Figura 4.3: Parámetros de la pala háptica y la rueda motriz

La segunda opción es la utilizada debido a que la fuerza generada se puede sentir justamente en esa parte de la pala, por tanto, se necesita un factor de conversión para los bits que se reciben desde el Arduino hacia una cantidad equivalente al arco descrito por la pala.

La relación entre la longitud de los arcos, tanto de la rueda motriz como de la parte inferior pala es

$$L_{zona} = L_{rueda}$$

por lo tanto

$$r_{zona}\theta_{zona} = r_{rueda}\theta_{rueda}$$

La longitud descrita por la parte superior de la pala es

$$x_{pala} = r_{pala}\theta_{zona}$$

Reemplazando

$$x_{pala} = \frac{r_{pala} \cdot r_{rueda}}{r_{zona}} \times \theta_{rueda} \quad (4.1)$$

El factor de conversión se representa como una ganancia a la salida de la planta, que identifica la posición de la pala, ver Figura 4.4.

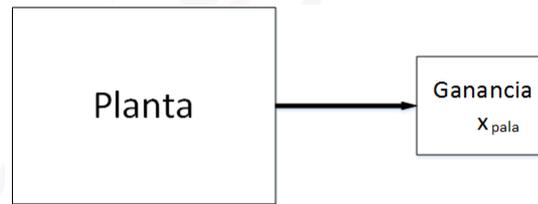


Figura 4.4: Diagrama de bloques, factor de conversión *cinemático*

Se tiene

$$r_{pala} = 0.1355 \text{ m}$$

$$r_{rueda} = 0.0178 \text{ m}$$

$$r_{zona} = 0.072 \text{ m}$$

La rueda motriz da un giro completo de 0 a 1023 bits, con lo que se encuentra una relación para el ángulo  $\theta_{rueda}$ , quedando

$$\theta_{rueda} = \frac{2\pi \cdot \#bit}{1023}$$

La relación final es

$$x_{pala} = \frac{(0,1355) \cdot (0,0178)}{(0,072)} \times \theta_{rueda} \quad (4.2)$$

#### 4.1.2. Factor de conversión *fuerza-torque*

La característica principal de las interfaces hápticas -kinestésicas de tipo impedancia en este caso- es la fuerza que pueden transmitir mediante su retroalimentación. Esta retroalimentación produce una fuerza, y dicha fuerza es generada mediante el torque del motor.

Para la interfaz, se realiza una conversión de la señal obtenida a la salida del controlador -fuerza- a un factor de voltaje que posteriormente, en el Arduino, mediante PWM, se traduce en el torque necesario que el motor debe aplicar para que el usuario sienta la retroalimentación de la pala.

El usuario siente la fuerza en la parte superior de la pala, por lo que intervienen algunos parámetros relacionados con las longitudes de la misma.

El torque del motor está determinado por

$$\tau = Fr$$

La fuerza aplicada por la rueda motriz es igual fuerza que recibe la pala en su parte inferior

$$F_{zona} = F_{rueda}$$

desarrollando

$$\frac{\tau_{zona}}{r_{zona}} = \frac{\tau_{rueda}}{r_{rueda}}$$

la fuerza en la pala es

$$F_{pala} = \frac{\tau_{zona}}{r_{pala}}$$

Por lo tanto, reemplazando

$$F_{pala} = \frac{r_{zona}}{r_{pala} \cdot r_{rueda}} \times \tau_{rueda} \quad (4.3)$$

El factor de conversión *fuerza-torque* se ubica antes de la entrada de la planta, la cual controla el actuador de la interfaz háptica, ver Figura 4.5.

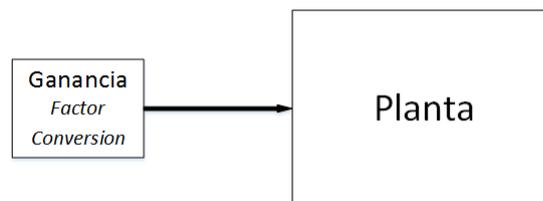


Figura 4.5: Diagrama de bloques, factor de conversión *fuerza-torque*



Teniendo los valores de

$$\begin{aligned}r_{pala} &= 0.1355 \text{ m} \\r_{rueda} &= 0.0178 \text{ m} \\r_{zona} &= 0.072 \text{ m}\end{aligned}$$

además de la resistencia presente en los terminales del motor igual a  $1,84\omega$  -se puede encontrar en el *datasheet* del mismo, Anexo B- y el torque encontrado en la Sección 4.2.2, el factor de conversión, utilizando la relación (4.3), es

$$Factor\_Conversion = \frac{F_{pala} \cdot (0,1355) \cdot (0,0178) \cdot (1,84)}{(0,072) \cdot (0,009)} \quad (4.4)$$

## 4.2. Determinación de parámetros del sistema

Las variables matemáticas involucradas con la interfaz tienen que ver con propiedades físicas relacionadas con la pala y el motor del sistema.

### 4.2.1. Constante de amortiguamiento del motor

El modelo matemático del motor es

$$J\dot{\omega} + b\omega = C \quad (4.5)$$

donde

- $J$  = inercia total del rotor del motor
- $b$  = constante de amortiguamiento del motor
- $C$  = constante de Coulomb
- $\omega$  = velocidad angular del motor

La inercia total del rotor del motor viene dado por la Ecuación (4.6), en la cual, la inercia total es igual a la inercia del rotor más la inercia de la masa que soporta el rotor, y en el caso de la interfaz háptica es la masa correspondiente a la rueda motriz del motor.

$$J = J_{rotor} + J_{masa} \quad (4.6)$$

La inercia del motor la encontramos fácilmente en las hojas de especificación

del motor, ver Anexo B, y es igual a  $10.9 [gr \cdot cm^2]$ . Mientras, la inercia de la rueda motriz adjunta al rotor la encontramos con la fórmula

$$J_{masa} = \frac{1}{2} (m_1 \cdot r_1^2 + m_2 (r_2^2 + r_3^2)) \quad (4.7)$$

donde  $m_1$  y  $m_2$  son las masas correspondientes a los cuerpos rígidos de la rueda motriz, el uno es un cilindro rígido de radio  $r_1$ , y la otra masa es un cilindro hueco con radios  $r_2$  y  $r_3$  –exterior e interior, respectivamente-, ver Figura 4.6.

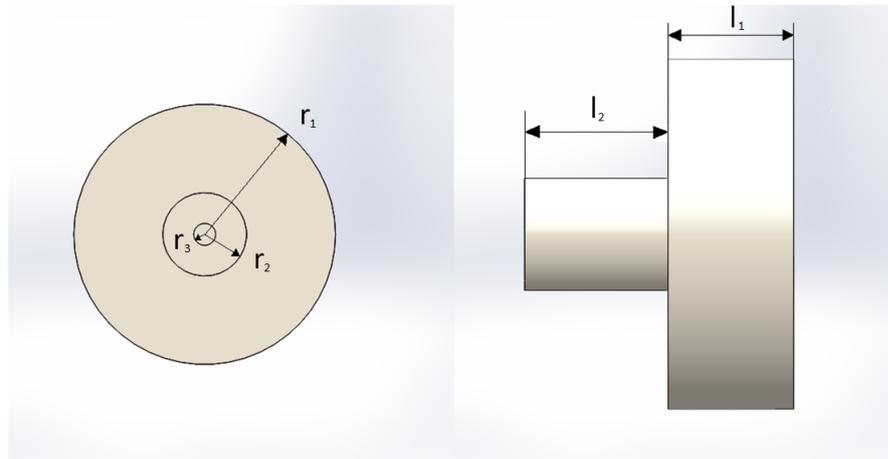


Figura 4.6: Parámetros de la rueda motriz de la interfaz háptica

De acuerdo a las mediciones realizadas tenemos los siguientes datos

$$\begin{aligned} masa\ total &= 41\text{ gr} \\ r_1 &= 1.78\text{ cm} \\ r_2 &= 0.57\text{ cm} \\ r_3 &= 0.15\text{ cm} \end{aligned}$$

también tenemos la longitud de los cilindros que son

$$\begin{aligned} l_1 &= 1.27\text{ cm} \\ l_2 &= 1.45\text{ cm} \end{aligned}$$

Encontramos el volumen de cada masa y se obtiene

$$\begin{aligned} V_1 &= 2\pi \cdot r_1^2 \cdot l_1 = 2\pi \cdot 1.78^2 \cdot 1.27 = 25,28cm^3 \\ V_2 &= 2\pi \cdot (r_2^2 - r_3^2) \cdot l_2 = 2\pi \cdot (0,57^2 - 0,15^2) \cdot 1.45 = 2,755cm^3 \end{aligned}$$

El volumen total es

$$V_T = V_1 + V_2 = 25,28cm^3 + 2,755cm^3 = 28,04cm^3$$

Las masas  $m_1$  y  $m_2$  son

$$m_1 = V_1 \left( \frac{m_T}{V_T} \right) = 25,28cm^3 \left( \frac{41gr}{28,04cm^3} \right) = 36,97gr$$

$$m_2 = V_2 \left( \frac{m_T}{V_T} \right) = 2,755cm^3 \left( \frac{41gr}{28,04cm^3} \right) = 4,0287gr$$

Aplicando la fórmula 4.7 se encuentra la inercia de la masa añadida al rotor

$$J_{masa} = \frac{1}{2} (36,97 \cdot 1,78^2 + 4,028 (0,57^2 + 0,15^2)) = 59,27gr \cdot cm^2$$

Y la inercia total del rotor es

$$J = J_{rotor} + J_{masa} = 10,9 + 59,27 = 70,17gr \cdot cm^2$$

La constante de amortiguamiento  $b$  del motor se determina mediante una comparación entre las respuestas real y simulada con ayuda del modelo del motor, Ecuación (4.5), en Simulink.

El experimento se divide en dos partes y consiste en aproximar la respuesta teórica a la respuesta real de la *fuerza electromotriz* del motor.

#### ■ Primera parte del experimento

1. Realizamos la conexión mostrada en la Figura 4.7.

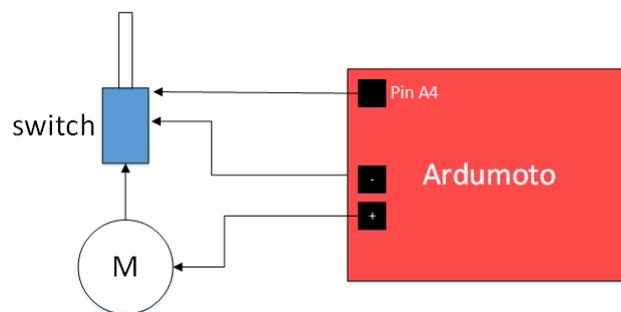


Figura 4.7: Conexión utilizada para determinar la constante de amortiguamiento

2. Se levanta la pala para evitar la fuerza de fricción provocada por la misma.
3. Activamos el *switch* que se observa en la Figura 4.7, dejando que el motor empiece a girar libremente y se estabilice. Generalmente, se tarda de 5 a 10 segundos en alcanzar una velocidad estable.

- Una vez que el motor está girando libremente se desactiva el *switch*, provocando una caída en la tensión en los bornes del motor, provocando su detención. En la Figura 4.8 se observa como el voltaje del motor empieza a descender lentamente antes de llegar a cero. A esta fuerza se la conoce como *Fuerza Electromotriz* del motor.

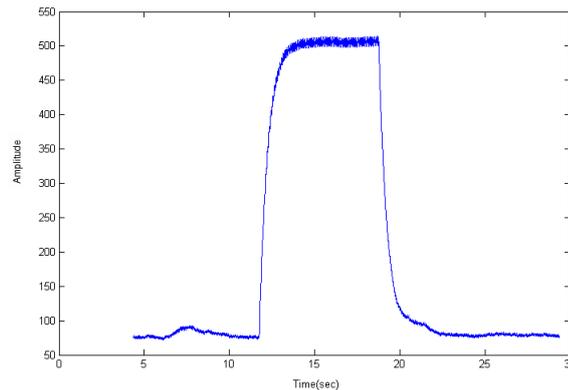


Figura 4.8: Respuesta de la fuerza electromotriz del motor

Las muestras importantes de la Figura 4.8 son los que se encuentran a partir de la desactivación del *switch*. También, los valores se encuentran en una escala de 0 a 1023 que corresponden a los datos recogidos directamente por el sensor, por lo que hay que transformarlos a un rango de 0 – 5V.

- Una vez procesados los datos entre 0 y 5V, y recortada la parte de la señal que nos interesa, obtenemos la señal mostrada en la Figura 4.9. Guardamos estos datos para la siguiente parte del experimento.

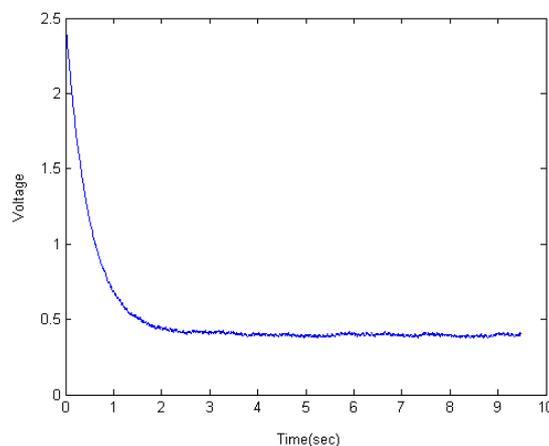


Figura 4.9: Datos válidos de la fuerza electromotriz del motor

### ■ Segunda parte del experimento

1. Simulamos el modelo descrito por la Ecuación (4.5) en Simulink mediante su respectivo diagrama de bloques, ver Figura 4.10. Nótese que el bloque de amortiguamiento pertenece a  $-b/J$ .

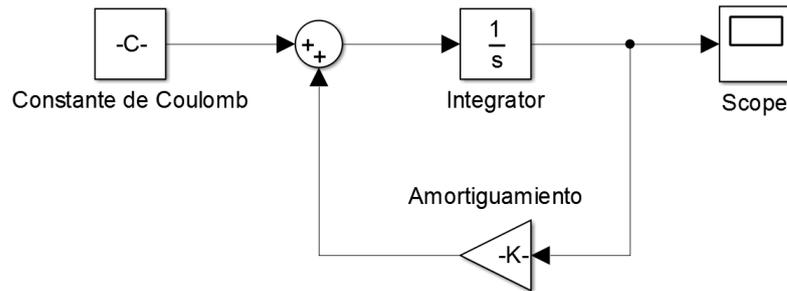


Figura 4.10: Diagrama de bloques del modelo del motor - Simulink

2. Con el modelo corriendo, se empieza a generar respuestas del sistema mediante la manipulación de los bloques pertenecientes a las constantes de *Coulomb* y de *amortiguamiento*. En el bloque integrador se da una condición inicial de 2.5V, misma que encontramos en el *paso 5* de la primera parte del experimento.
3. Las respuestas simuladas se comparan con la respuesta real encontrada en el *paso 5* del anterior procedimiento hasta obtener la mejor aproximación entre las respuestas teórica y experimental, determinando de esta manera las constantes **C** y **b/J**.

En nuestro caso utilizamos una función de costo para determinar dichas constantes de manera óptima como se observa en la Figura 4.12. Obteniendo

$$C = 0.5477 [Nm^2/C^2]$$

$$\frac{b}{J} = 1,42$$

Simulando el diagrama de bloques con los valores anteriormente encontrados obtenemos la respuesta mostrada en la Figura 4.12.

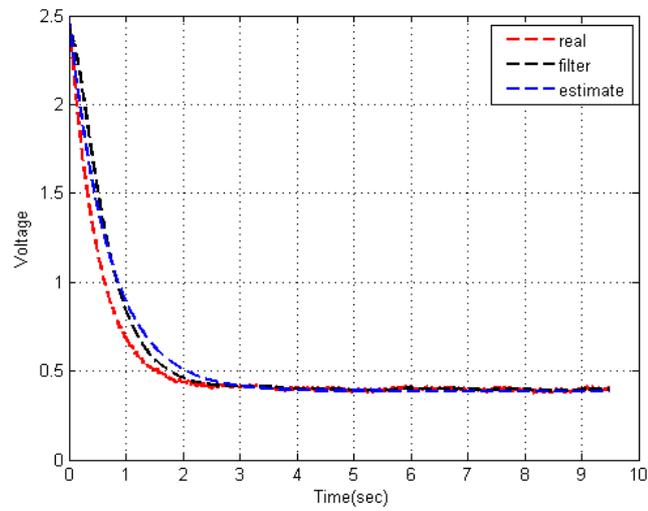


Figura 4.11: Aproximación del modelo teórico y real mediante una función de optimización

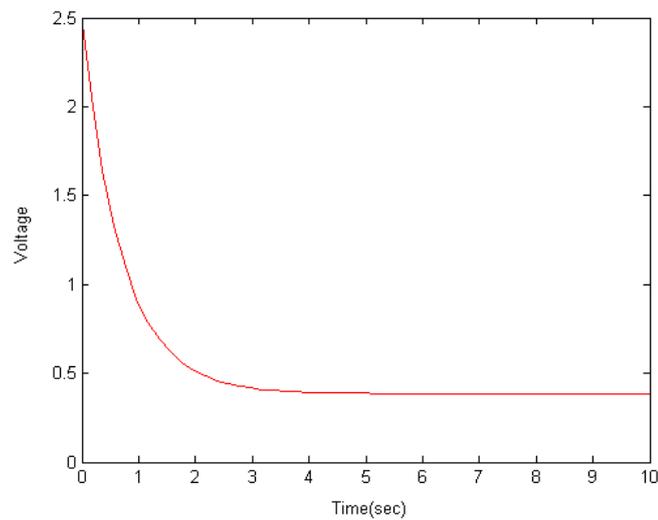


Figura 4.12: Señal obtenida en Simulink del modelo del motor

Como

$$\frac{b}{J} = 1,42$$

y

$$J = 70,17 \text{ [gr} \cdot \text{cm}^2\text{]}$$

entonces

$$b = 1,42 \cdot J = 1,42 \cdot 7,017e - 6$$



$$b = 9,96e - 6 \left[ \frac{Kg \cdot m^2}{seg} \right]$$

### 4.2.2. Torque del motor

Es la fuerza o par que puede generar el motor ante una determinada corriente que circula por el mismo y esta determinado por la Ecuación (4.8). Una característica importante del motor Maxon 353144, elegido para la interfaz, es que ejerce un gran torque a pequeñas corrientes.

$$\tau = k_m i \quad (4.8)$$

La determinación del torque del motor se realizó experimentalmente con ayuda de diferentes instrumentos y materiales.

#### Materiales e instrumentos

- Amperímetro
- Balanza
- Fuente de 8V
- Motor con su rueda motriz
- Diferentes masas con diferentes pesos
- Cinta adhesiva

Los pasos a seguir se presentan a continuación

1. Se conecta el amperímetro en serie entre la salida del *Arduemoto Shield* y el motor.
2. Se coloca una objeto en el extremo de la rueda motriz del motor, y se la sujeta con cinta adhesiva. Previamente se mide la masa del objeto.
3. Se energiza el sistema y se empieza a variar el PWM desde cero hasta que el motor equilibre la masa, tal como se observa en la Figura 4.13.

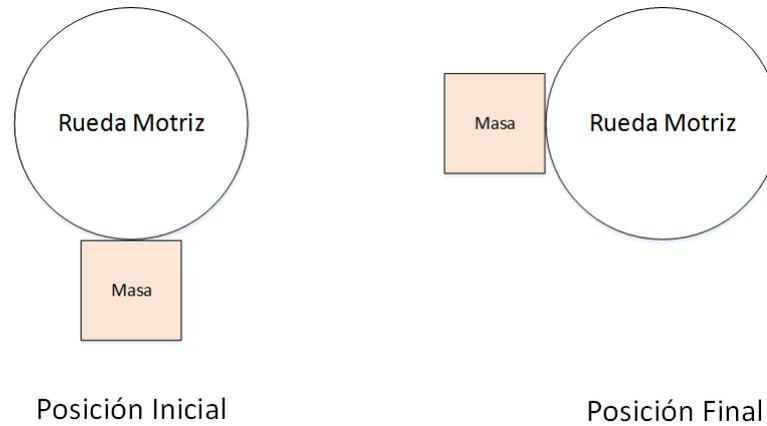


Figura 4.13: Experimento para determinar el torque del motor

4. Se procede a registrar la lectura de corriente proporcionada por el amperímetro.
5. Se repite el proceso variando las masas, tres o cuatro veces más.

Una vez realizado este procedimiento, se calcula el torque producido por el motor para cada masa, con la ecuación

$$\tau = rF \quad (4.9)$$

donde

$$r = 1.78 \text{ cm (radio de la rueda motriz)}$$

$$F = mg$$

La tabulación de los datos obtenidos se presenta en la Tabla 4.1.

Tabla 4.1: Datos de corriente y torque

Corriente (mA)	Torque (Nm)
300	1.920798
440	2.968506
690	5.23854
780	6.286248
880	6.98472

La Figura 4.14 muestra la gráfica de la relación entre la corriente y el torque. La constante del torque del motor está determinada por la pendiente de la ecuación mostrada en la Figura 4.14, y es igual a

$$k_m = 0,009 [Nm/mA]$$

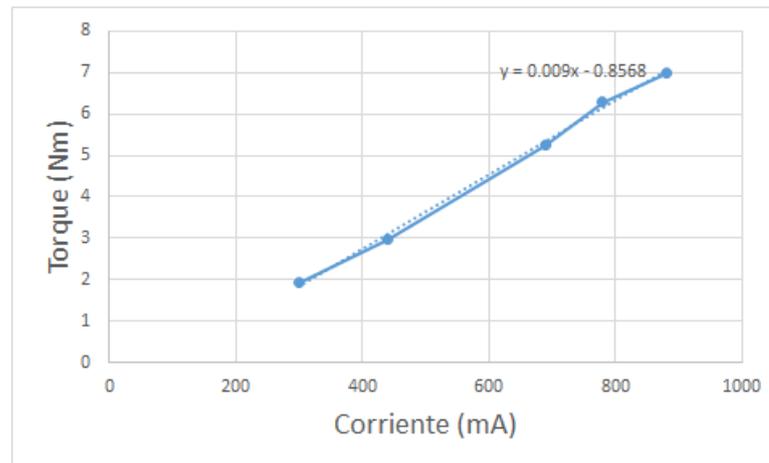


Figura 4.14: Gráfico del torque en función de la corriente para determinar la constante  $k_m$

### 4.2.3. Fricción del motor

Para determinar la fricción del motor se debe medir la mínima cantidad de corriente necesaria para mantener el movimiento del motor y/o para empezar el movimiento del mismo.

Se utiliza la constante  $k_m$  encontrada en la Ecuación (4.9) para determinar el torque o par de fricción del motor.

El experimento consta de los siguientes pasos

1. Se realiza la conexión de acuerdo a la Figura 4.15.

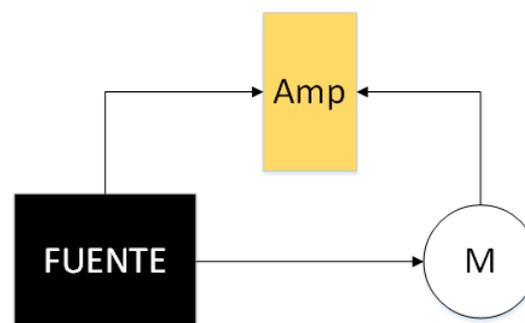


Figura 4.15: Conexión utilizada para determinar la fricción del motor



Tabla 4.2: Medidas de corriente para encontrar la fricción del motor

	Corriente (mA)
Primer Paso	64.8
Segundo Paso	68
Promedio	66.4

2. Con el motor totalmente en reposo y con la fuente también en cero, se incrementa la corriente hasta que el motor empiece a girar. Se debe encontrar un valor de corriente para el cual el motor no pueda girar por su cuenta, pero pueda empezar a girar una vez que se le da un impulso inicial con la mano. Tabular esta corriente.
3. Con el motor girando libremente se reduce la corriente que circula por el mismo y se mide el valor para la cual el motor se detiene completamente. Se debe registrar esta corriente.
4. Una vez realizadas las mediciones, ver Tabla 4.2, se tiene un rango de valores que representan la fricción estática y cinética. Se procede a realizar un promedio entre estos valores para obtener una aproximación de la fricción.

La fricción del motor es

$$\tau = 0,0009 \cdot 66,4 = 0,598 [mN \cdot m]$$

#### 4.2.4. Inercia de la pala háptica

La determinación de la inercia de la pala de la interfaz se realizó experimentalmente, mediante la observación de la frecuencia natural de la misma, utilizando el modelo de un péndulo bifilar, cuya fórmula es

$$I_z = \frac{mgD^2}{16\pi^2 f_n^2 h} \quad (4.10)$$

donde

$m$  = masa de la pala

$D$  = distancia entre los dos sujetadores

$h$  = altura desde el punto de sujeción al soporte

$f_n$  = frecuencia natural de la pala

La mayoría de variables de la Ecuación (4.10) son medibles directamente con excepción de la frecuencia natural de la pala, por lo cual, se utiliza el siguiente experimento para su determinación.

1. Se sujeta la pala de dos extremos. Cada uno, a una distancia  $D/2$  desde el centro de masa de la pala, y a la misma altura como se observa en la Figura 4.16.

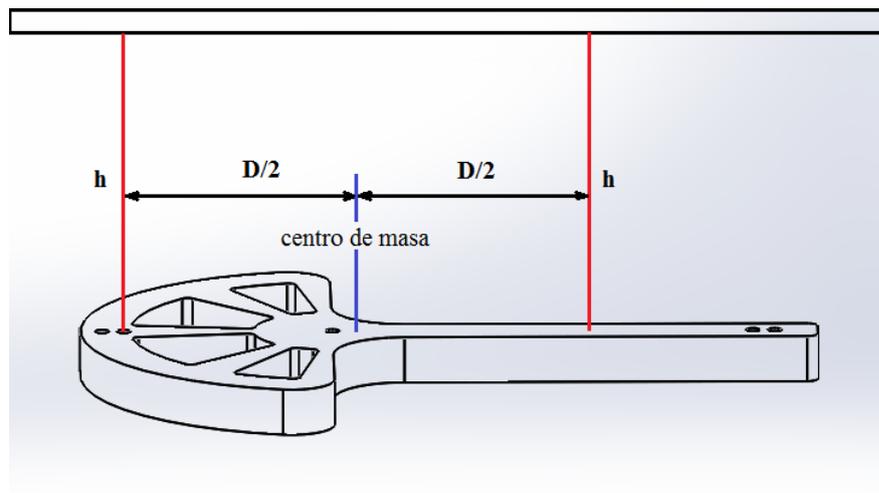


Figura 4.16: Montaje utilizado para determinar la frecuencia natural de la pala háptica, paso 1

El centro de masa de la pala lo ubicamos de manera directa gracias al programa Solidworks, y es

$$\text{Centro\_Masa} = 6.50 [mm]$$

ubicado a partir del eje de rotación de la pala.

2. Se desplaza la pala un ángulo  $\phi$ , el cual, no debe ser muy grande. Otorgándole, así, una condición inicial al sistema, ver Figura 4.17.

Tabla 4.3: Parámetros útiles para encontrar la inercia de la pala

Parámetros	
$f_n$	0.886 Hz
m	106 gr
D	10 cm
h	21 cm

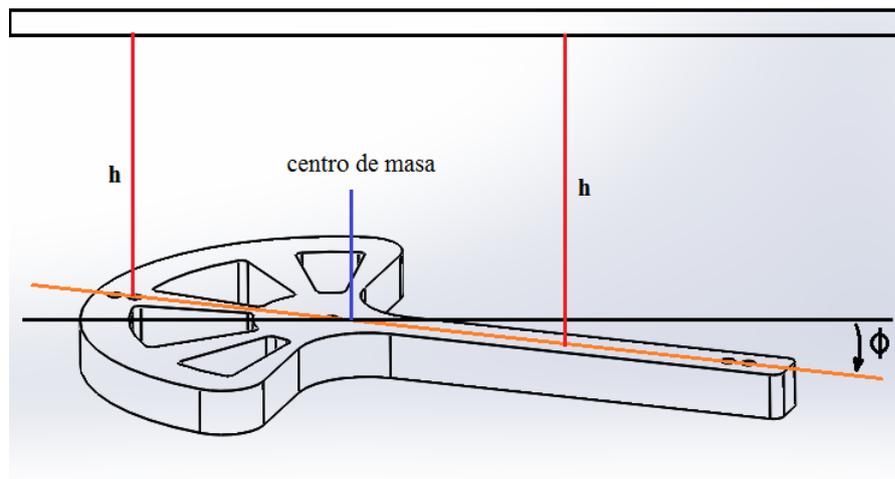


Figura 4.17: Montaje utilizado para determinar la frecuencia natural de la pala háptica, paso 2

- Contamos el número de oscilaciones completas que realiza la pala en un tiempo determinado. En nuestro caso, la pala hizo 20 oscilaciones en 22.571 segundos, lo que nos da una frecuencia natural de 0.886 Hz, ver Tabla 4.3.

Por lo tanto

$$I_z = \frac{(0,106Kg) (9,81m/s^2) (0,1m)^2}{16\pi^2(0,886Hz)^2 (0,21m)} = 0,0004 [Kg \cdot m^2]$$

## 4.3. Control del interfaz háptico mediante PID

### 4.3.1. Control retroalimentado

El PID es el primer controlador utilizado sobre la planta. El diagrama de bloques de la Figura 4.18 ilustra, claramente, y de manera detallada, las variables involucradas en el control de lazo cerrado de la pala háptica. En dicho diagrama se observa el camino que sigue la señal de control.

La señal ingresa al controlador del motor -Arduino en nuestro caso-, el cual, determina el torque que debe imprimir el motor para que se sienta la fuerza en la pala háptica. Antes de enviar la señal directamente al Arduino, se realiza la conversión de fuerza a torque determinado por la relación (4.4).

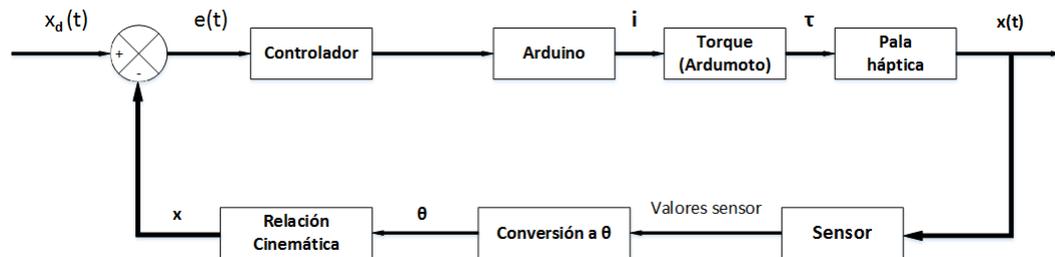


Figura 4.18: Control clásico del interfaz háptico

Mediante la realimentación, el valor que provee el sensor se convierte a un ángulo, agregando transparencia hacia el usuario. El ángulo se puede convertir hacia una relación cinemática de manera opcional, ya que esta relación es útil para la interacción dentro de ambientes virtuales, y no en el control en sí.

El diagrama de bloques en lazo cerrado de Simulink se muestra en la Figura 4.19,

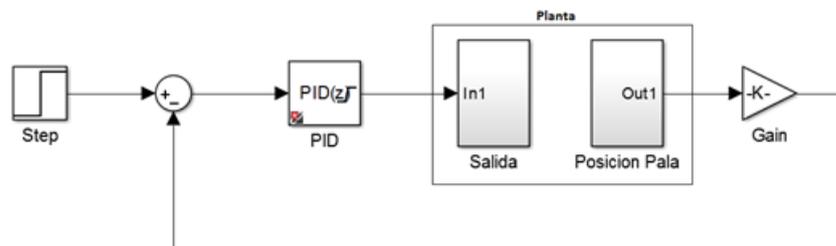


Figura 4.19: Controlador PID para el interfaz háptico - Simulink

### 4.3.2. Sintonización del PID

Se realizó la sintonización para diferentes valores de  $k_c$ ,  $T_i$  y  $T_d$ , y con dos respuestas al escalón, una positiva y la otra negativa, cabe señalar que la referencia se encuentra en metros debido a que se está utilizando el factor de conversión cinemática descrito por la relación (4.2). El tiempo de muestreo es de 0.005 seg.

A continuación se presentan los resultados de las gráficas obtenidas durante la sintonización.

$$\text{PID} > \mathbf{P = 35, D = 0.05, I = 0}$$

Como se observa en las gráficas, el tiempo de respuesta es muy rápido pero presenta sobreoscilaciones con una amplitud considerable.

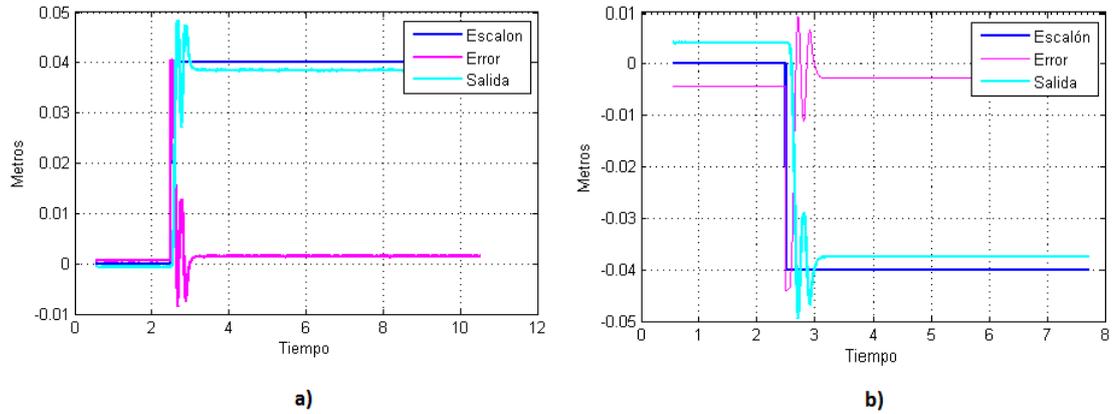


Figura 4.20: Respuesta del controlador PID para una entrada escalón a) positivo y b) negativo

$$\text{PID} > P = 35, D = 0.05, I = 0.1$$

Para este caso las sobreoscilaciones también están presentes como en el caso anterior. Además, el error de estado permanente es mayor que el anterior.

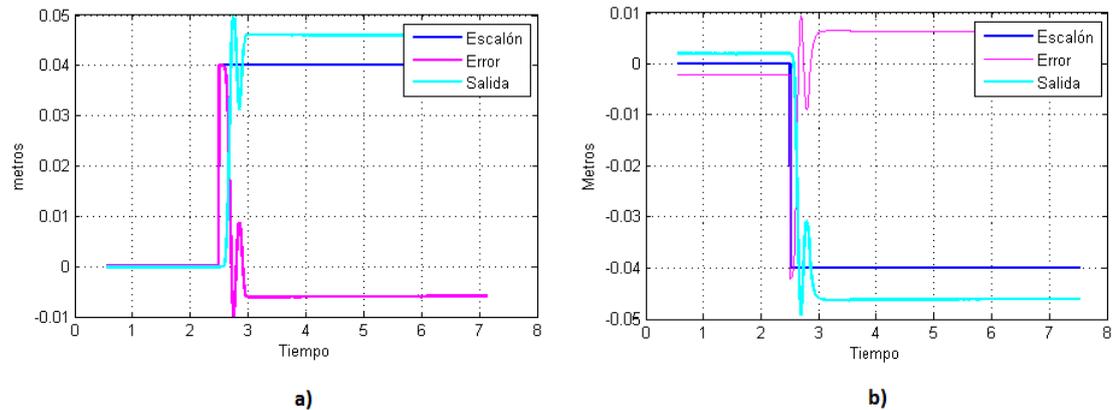


Figura 4.21: Respuesta del controlador PID para una entrada escalón a) positivo y b) negativo

$$\text{PID} > P = 35, D = 0.075, I = 0.5$$

Las respuestas de la planta de la Figura 4.22 no presentan sobreoscilaciones y el tiempo de respuesta es igual o aproximado a los dos casos anteriores.

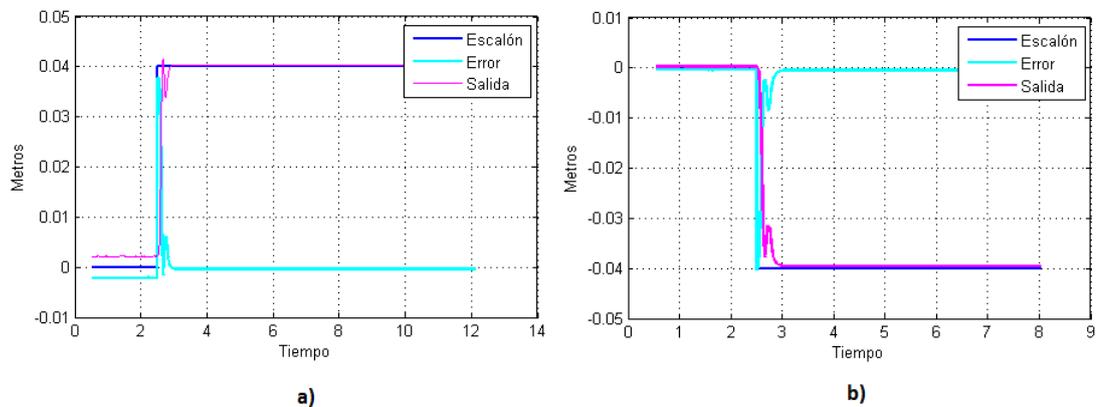


Figura 4.22: Respuesta del controlador PID para una entrada escalón a) positivo y b) negativo

A diferencia de las demás gráficas, en la Figura 4.22, el error de estado permanente es casi cero por lo que estos valores son los óptimos y los escogidos para el controlador PID.

## 4.4. Control del interfaz háptico utilizando técnicas de control moderno

### 4.4.1. Identificación del sistema

La identificación paramétrica de sistemas dinámicos permite obtener un modelo matemático para el diseño de algún controlador, entender fenómenos observados, pronosticar eventos futuros, etc.

La identificación del sistema para encontrar el modelo de la interfaz háptica, se basa en un método experimental que consiste en recolectar un conjunto de datos e inferir en el modelo a partir de un análisis de entradas y salidas luego de someter a la planta a un conjunto de valores -señal- que provocan perturbaciones a la misma, ver Figura 4.23.

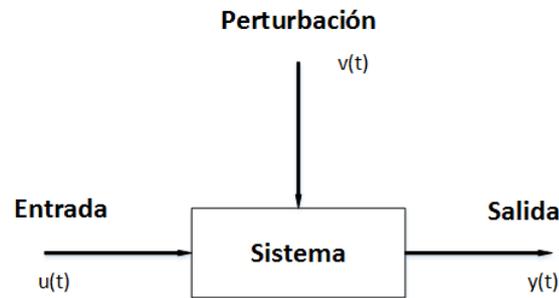


Figura 4.23: Diagrama de bloques para la identificación de los sistemas dinámicos

#### 4.4.2. Identificación de la planta para diseño de controladores

Se utiliza una señal de identificación *-ID Signal-* que puede ser de diferentes tipos como un escalón, sinusoidales, etc. Lo importante es que la planta sea sometida a un amplio rango de excitación. La mejor señal para la identificación sería una señal de ruido blanco que contiene componentes de frecuencia en todo el espectro, pero es complicada de generar, por lo que se utiliza una señal binaria pseudoaleatoria *-PRBS* por sus siglas en inglés- que tiene propiedades similares a los del ruido blanco en su espectro. Es una señal que toma dos posibles valores *-binaria-* dentro de un determinado rango de bits  $N$ . Por cada ciclo de reloj, la señal, sin seguir un patrón, toma un valor aleatorio, de los dos posibles, es decir, cada valor que compone la señal es independientes de los demás, hasta  $2N - 1$ . Luego de este tiempo la señal se repite, de ahí se debe su nombre de *PSEUDO*. Para valores grandes de  $N$ , sus propiedades se asemejan al ruido blanco.

El modelo resultante captura la información esencial del sistema a partir de la relación observada entre la señal que provoca la perturbación y la salida del sistema. Se escoge el modelo que puede interpretar los datos con el menor número posible de parámetros de control.

La identificación de entradas y salidas se la realiza en lazo abierto utilizando un modelo autorregresivo con variable externa *-ARX*, por sus siglas en inglés-, Ecuación (2.10), que modela el sistema de acuerdo a la comparación del error y ajustando los parámetros del *ARX*. La secuencia del diagrama de bloques de Simulink se observa en la Figura 4.24.

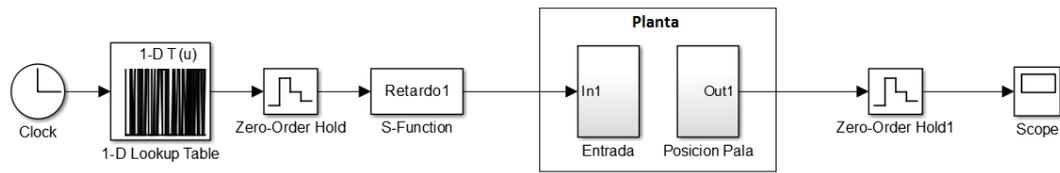


Figura 4.24: Diagrama de bloques para la identificación del sistema - Simulink

Generamos una señal aleatoria en MATLAB, ver Figura 4.25, con las siguientes propiedades:

$N = 128$  -número de muestras-

$T = 0,1seg$  -tiempo de muestreo-

Niveles Binarios =  $[-0,7 \ 0,7]$

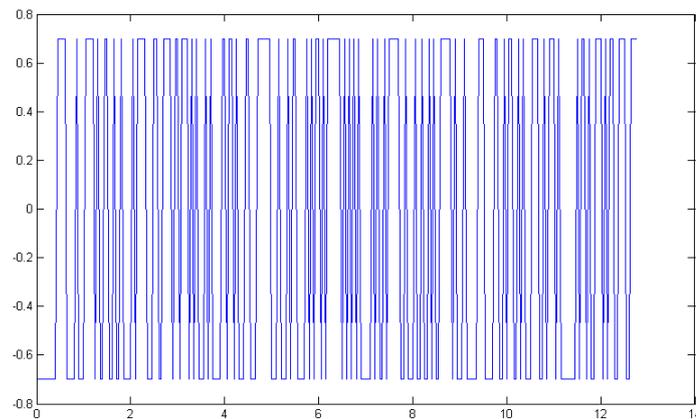


Figura 4.25: Señal PRBS

En el bloque de entrada de la Figura 4.25 se carga la señal de identificación *PRBS*, cuya salida  $\mathbf{u}(k)$  se aplica a la planta haciendo que ésta empiece a operar sobre un punto de trabajo determinado, generalmente cero, generando una respuesta dinámica de la planta que se observa en la Figura 4.26.

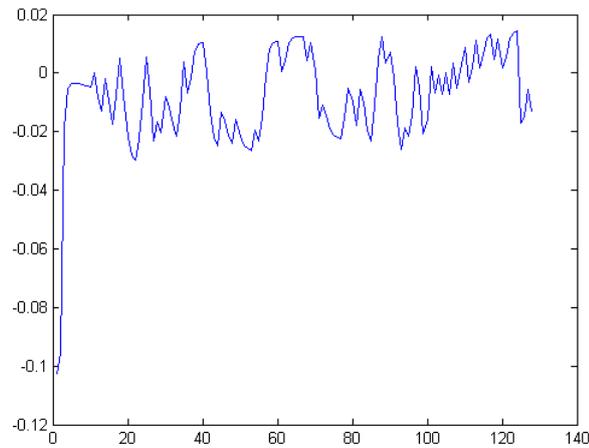


Figura 4.26: Respuesta de la planta

Se procede a realizar un análisis entre las entradas y salidas de la planta, ver Figura 4.27, producto de la excitación a la que fue sometida.

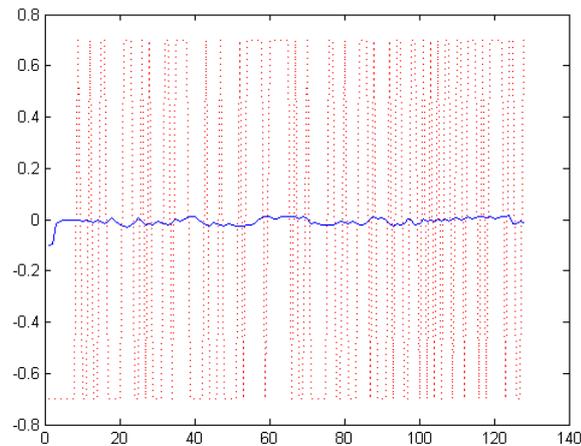


Figura 4.27: Señal PRBS vs salida de la planta

El análisis se realiza con la herramienta de MATLAB: *ident*. Luego de importar la señal de entrada,  $\mathbf{u}(k)$ , y la señal de salida,  $\mathbf{y}(k)$ , y luego de escoger una estimación polinomial del modelo *ARX*, se obtiene una gran cantidad de modelos, ver Figuras 4.28 y 4.29. Pero se procede a elegir el modelo que mejor se ajuste a los requerimientos de diseño de acuerdo al rendimiento que puede proporcionar gracias a las características de sus parámetros. También se presta atención al número de polos que ofrece cada modelo y la bondad de ajuste que ofrecen los mismos entre las señales de salida: medida y estimada.

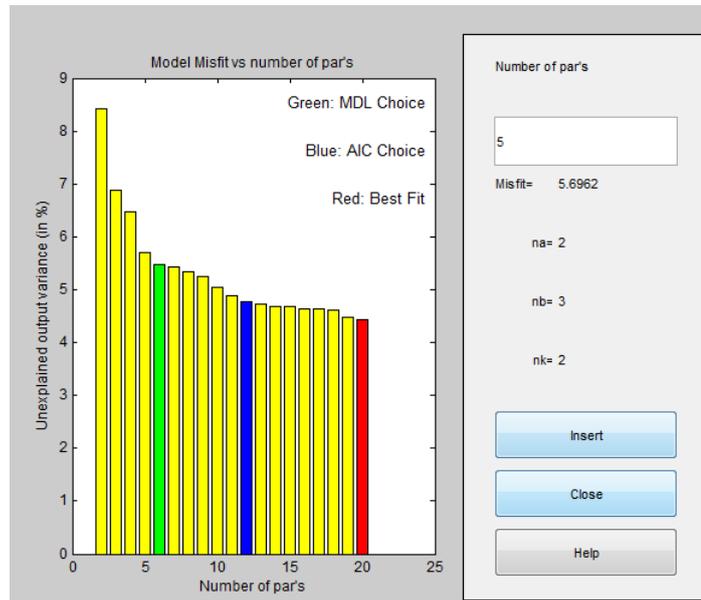


Figura 4.28: Modelos estimados vs varianza de salida

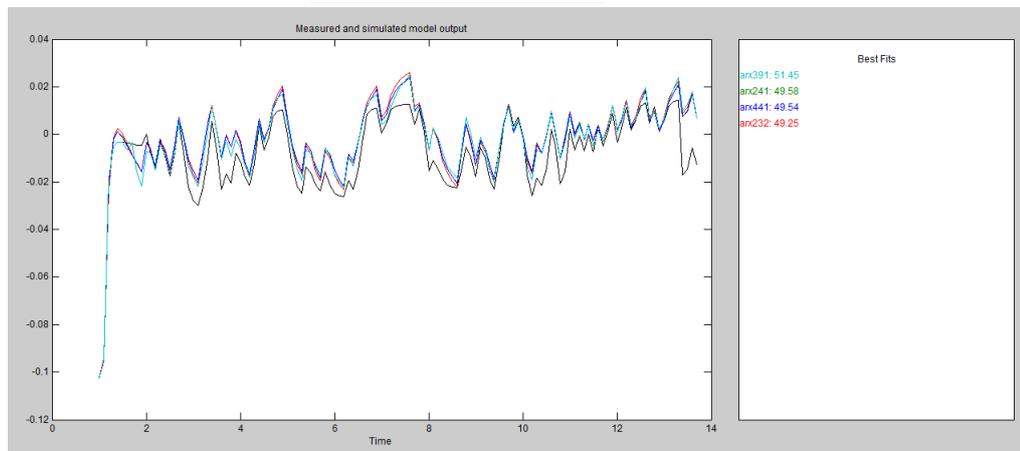


Figura 4.29: Salida medida vs salida simulada de los modelos que mejor rendimiento presentan

Para el caso de la interfaz háptica, se escoge el modelo  $arx232$ , ver Figura 4.30, el cual, consta de dos polos para el control de la planta. Además, brinda un ajuste del 79.14 %, el cual no hay que confundir con la estimación de la Figura 4.29, cuyo porcentaje representa la diferencia del error entre las salidas medida y simulada de la planta.

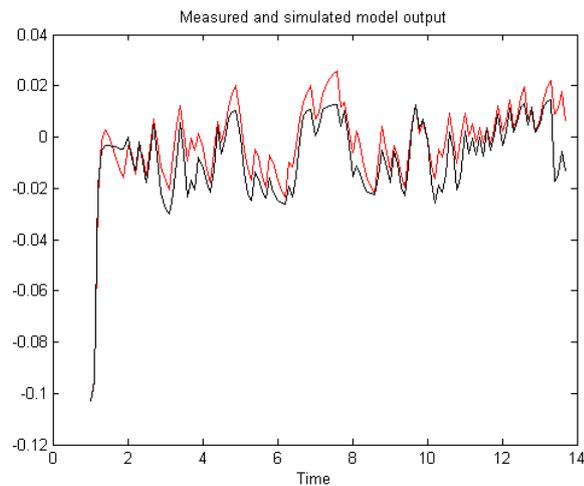


Figura 4.30: Señal de salida de la planta vs aproximación *arx232*

Una vez escogido el modelo a ocupar, se procede a encontrar las matrices de las ecuaciones de estado **G**, **H**, **C** y **D**.

Primero se obtiene la función de transferencia discreta mediante los comandos de MATLAB

```

1 % Determinacion de las matrices de estado G, H, C, D
2 arx232 = arx ([Y U], [2 3 2])
3 HGid = tf (arx232.B, arx232.A,T)

```

entonces

$$HGid = \frac{0,01079z^2 - 0,003185z - 0,002687}{z^2 - 1,229z + 0,3432} \quad (4.11)$$

Hecho esto, se procede a encontrar las matrices del sistema con el comando de MATLAB

```

1 [G H C D] = ssdata (HGid)

```

Las matrices encontradas son

$$\mathbf{G} = \begin{bmatrix} 1,2285 & -0,6863 \\ 0,5 & 0 \end{bmatrix}$$

$$\mathbf{H} = \begin{bmatrix} 0,125 \\ 0 \end{bmatrix}$$

$$\mathbf{C} = [0,0806 \quad -0,1022]$$

$$\mathbf{D} = 0$$



### 4.4.3. Controlabilidad y observabilidad del sistema

Comprobamos la controlabilidad completa y observabilidad completa del sistema mediante

$$\text{rango} \left( \left[ \mathbf{H} : \mathbf{G}\mathbf{H} : \dots : \mathbf{G}^{n-1}\mathbf{H} \right] \right) = n$$

y

$$\text{rango} \left( \left[ \mathbf{C}^T : \mathbf{G}^T \mathbf{C}^T : \dots : (\mathbf{G}^T)^{n-1} \mathbf{C}^T \right] \right) = n$$

respectivamente, donde  $n$  es el rango de  $\mathbf{G}$ , y es igual a

$$\text{rango} \left( \begin{bmatrix} 1,2285 & -0,6863 \\ 0,5 & 0 \end{bmatrix} \right) = 2$$

Entonces, mediante el comando **rank** de MATLAB encontramos que

$$\text{rank} \left( \begin{bmatrix} 0,125 \\ 0 \end{bmatrix} \begin{bmatrix} 1,2285 & -0,6863 \\ 0,5 & 0 \end{bmatrix} \begin{bmatrix} 0,125 \\ 0 \end{bmatrix} \right) = 2$$

$$\text{rank} \left( \begin{bmatrix} [0,0806 & -0,1022]^T & \begin{bmatrix} 1,2285 & -0,6863 \\ 0,5 & 0 \end{bmatrix}^T & [0,0806 & -0,1022]^T \end{bmatrix} \right) = 2$$

Por lo que el sistema es completamente controlable y completamente observable.

### 4.4.4. Control por reubicación de polos

Una vez encontradas las matrices de las ecuaciones de estado, ya se puede realizar algún tipo de control del sistema, el primero a realizar es un regulador a través de la realimentación de los estados del sistema y el diseño de un observador, cuyo diagrama de bloques se muestra en la Figura 4.31.

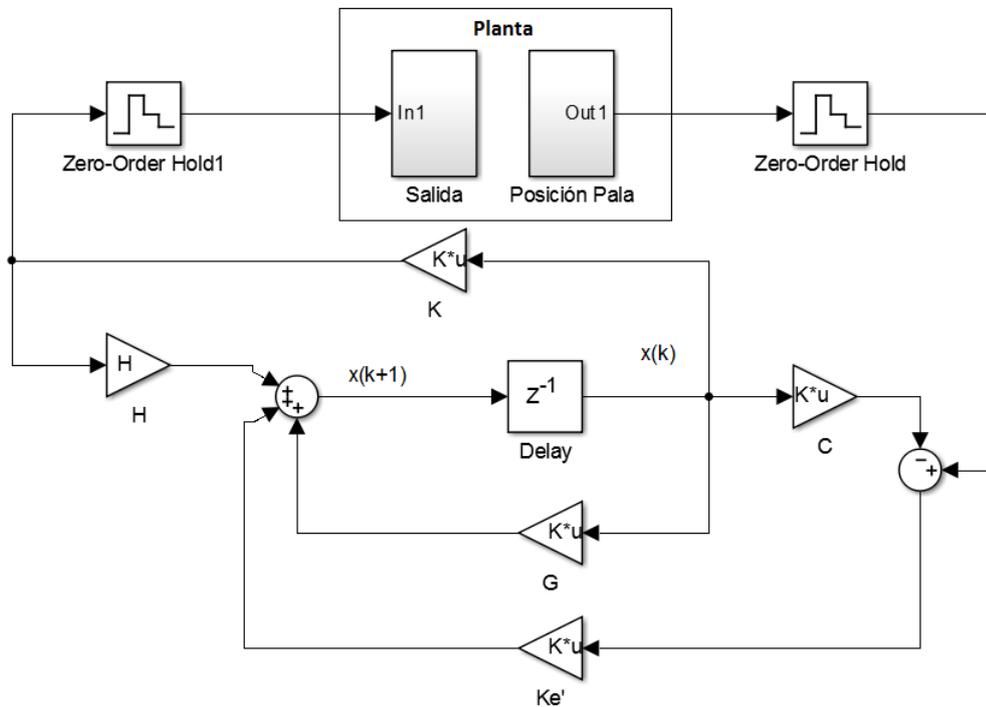


Figura 4.31: Control con estado observado para la interfaz háptica - Simulink

El observador necesita de las matrices de realimentación  $\mathbf{K}$  y  $\mathbf{K}_e$ .

Primero, ubicamos los polos deseados de la planta, en nuestro caso, ocupamos las raíces de los polinomios de *Bessel* normalizados del Anexo C. Como el rango de nuestro sistema es dos, ocupamos los polos

$$p_1 = -4,503 + j 2,34$$

$$p_2 = -4,503 - j 2,34$$

Discretizamos los polos mediante

$$z = e^{T p_i} \quad (4.12)$$

donde  $T$  es el tiempo de muestreo.

Los polos quedan de la siguiente manera

$$z_1 = \exp(T p_1) = \exp(T (-4,503 + j 2,34))$$

$$z_2 = \exp(T p_2) = \exp(T (-4,503 - j 2,34))$$

Para encontrar la matriz de realimentación,  $\mathbf{K}$ , utilizamos los polos ubicados

anteriormente en la fórmula de Ackermann, Sección 2.6.5.

Este paso se simplifica utilizando el comando de MATLAB

```
1 K = acker(G, H, [z1 z2])
```

el cual nos entrega la matriz de realimentación

$$K = [-0,0185 \quad 0,2021]$$

Lo mismo hacemos para encontrar la matriz  $K_e$ . El comando de MATLAB es

```
1 Ke = acker(G', C', [0 0])
```

y la respuesta obtenida es

$$K_e = [116,9292 \quad 80,1450]$$

Con estos datos es posible implementar el regulador en Simulink. Para comprobar el correcto funcionamiento del regulador tenemos que fijarnos que la salida de la planta siempre se ubique en cero, y si existe cualquier perturbación, ésta debe regresar a su punto de origen como se observa en la Figura 4.32.

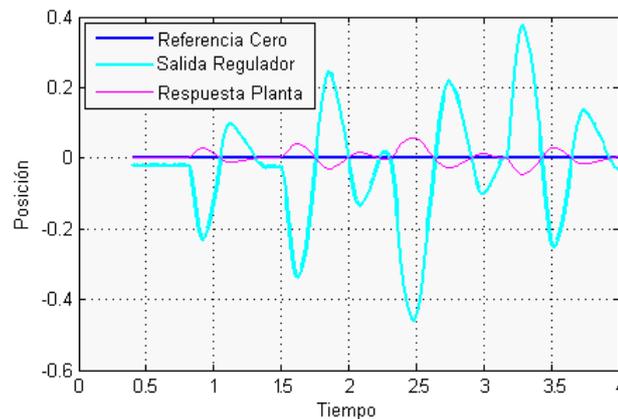


Figura 4.32: Respuesta de la interfaz háptica con regulador implementado en Simulink

#### 4.4.5. Seguidor de referencias constantes

Al regulador realizado le agregamos una etapa de compensación que se encargue de cambiar el punto de trabajo de la pala háptica. El diagrama de bloques se observa en la Figura 4.33.

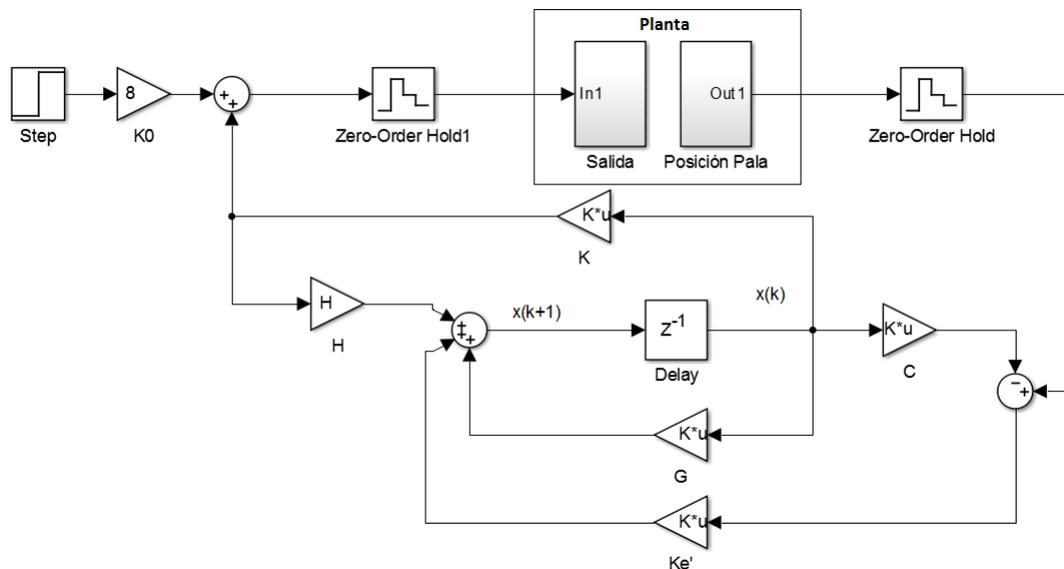


Figura 4.33: Sistema de seguimiento de referencias constantes para el interfaz háptico en Simulink

Al cambiar el punto de trabajo se cambia el punto de equilibrio de las variables de estado. En la Figura 4.34 se presenta la respuesta al escalón de la interfaz háptica, así como la salida de voltaje para el control del torque aplicado al motor. También se puede apreciar el comportamiento de la interfaz al momento que es sometida a perturbaciones externas.

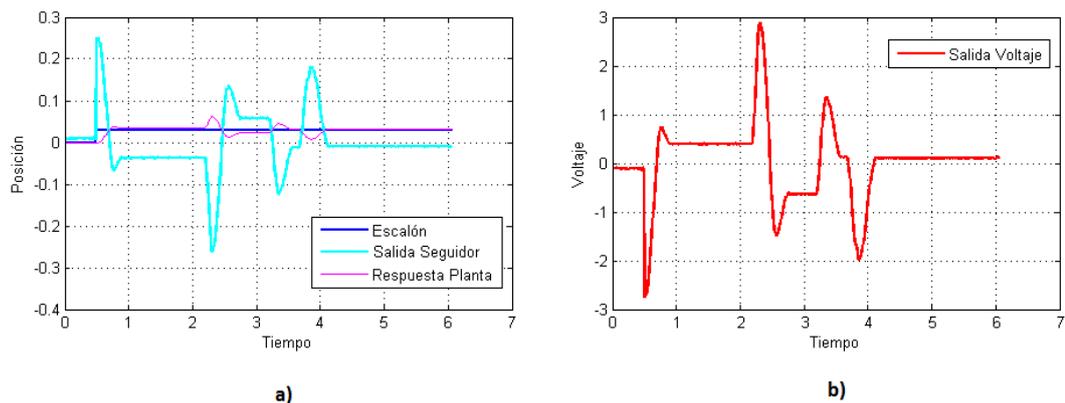


Figura 4.34: Respuesta de la interfaz háptica ante un escalón

En la Figura 4.35 se presenta la respuesta de la interfaz háptica, a la cual, se le cambia el punto de referencia dos veces, y como se observa en dicha imagen, el tiempo de respuesta es muy rápido y eficiente comparado con el PID ya que tampoco presenta demasiadas oscilaciones.

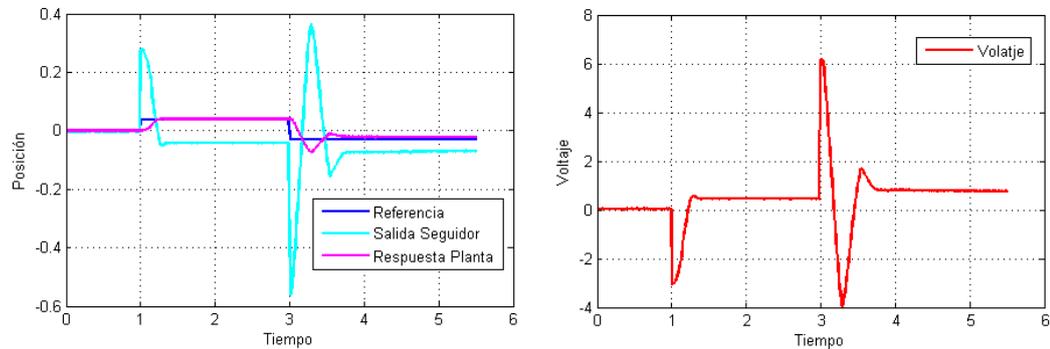


Figura 4.35: Respuesta de la interfaz háptica ante dos cambios de referencia

#### 4.4.6. Sistema de seguimiento con integrador y realimentación de estado observado

Al control desarrollado en la Sección 4.4.5, le sumamos un integrador para eliminar el error de estado permanente. Este integrador añade un polo más al sistema. La Figura 4.36 muestra el diagrama de bloques utilizado en Simulink.

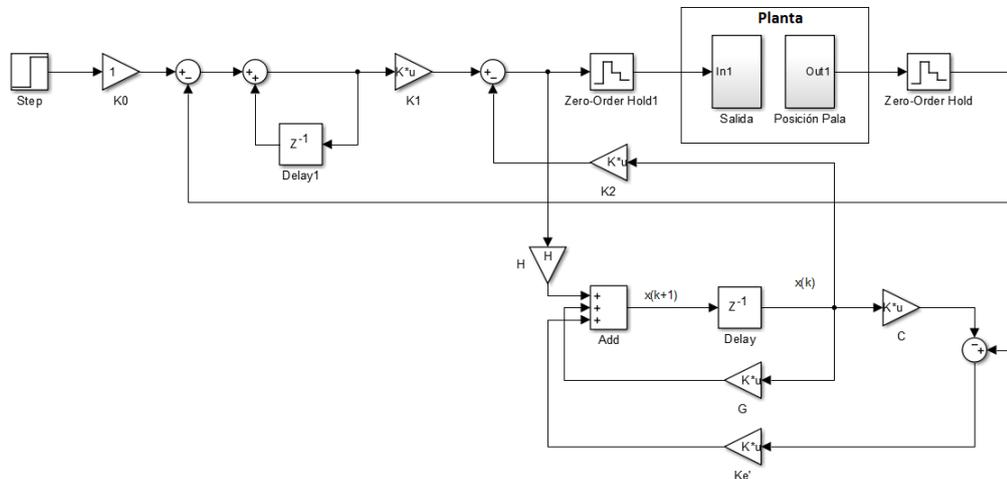


Figura 4.36: Seguimiento con integrador y estado observado para la interfaz háptica - Simulink

De la misma manera que hicimos con el observador, escogemos las raíces pertenecientes a los polinomios de *Bessel* de la tabla descrita en el Anexo C.

$$p_1 = -5,0093$$

$$p_2 = -3,9668 + j 3,7845$$

$$p_3 = -3,9668 - j 3,7845$$



discretizando

$$z_1 = \exp(T(-5,0093))$$

$$z_2 = \exp(T(-3,9668 + j * 3,7845))$$

$$z_3 = \exp(T(-3,9668 - j * 3,7845))$$

Utilizando nuevamente la fórmula de Ackermann, Sección 2.6.5, encontramos la matriz de realimentación  $\mathbf{K}_c$ . Los comandos de MATLAB utilizados son

```
1 Gc = [G H; 0 0 0];  
2 Hc = [0; 0; 1];  
3 Kc = acker(Gc, Hc, [z1 z2 z3]);
```

donde

$$\mathbf{G} = \begin{bmatrix} 1,2285 & -0,6863 \\ 0,5 & 0 \end{bmatrix}$$

$$\mathbf{H} = \begin{bmatrix} 0,125 \\ 0 \end{bmatrix}$$

La matriz de realimentación  $\mathbf{K}_c$  es

$$\mathbf{K}_c = [0,7669 \quad -0,9411 \quad -0,6273]$$

A partir de la cual, podemos encontrar las matrices  $\mathbf{K1}$  y  $\mathbf{K2}$ , como se muestra en el siguiente código de MATLAB

```
1 K2K1 = (Kc + [0 0 1]) * [G-eye(2) H; C*G C*H]^(-1)
```

donde

$$K2K1 = [1,2363 \quad -1,1051 \quad 21,6553]$$

a partir de lo cual

$$K2 = [1,2363 \quad -1,1051]$$

$$K1 = [21,6553]$$

Una vez encontrados todos los parámetros necesarios para el seguidor con integrador, procedemos a realizar la simulación, obteniendo los siguientes resultados.

La Figura 4.37 presenta la respuesta de la interfaz háptica ante varias perturbaciones. Como se observa, el error de estado permanente se elimina con este tipo de seguimiento. El proceso que realiza el seguidor es de integrar el error acumulado para posteriormente eliminarlo.

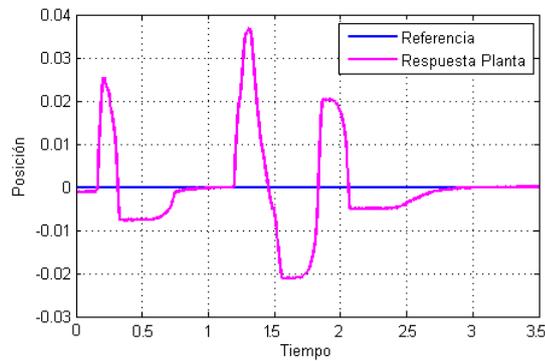


Figura 4.37: Respuesta del seguidor con integrador y estado observado ante una referencia cero - Simulink

La Figura 4.38 muestra la respuesta de la planta ante una entrada tipo escalón. Al inicio la repuesta presenta un error de estado permanente, el cual, como en el caso anterior, es eliminado por completo gracias al seguidor con integrador.

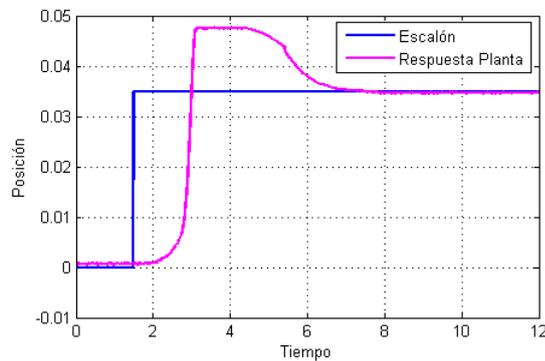


Figura 4.38: Respuesta del seguidor con integrador y estado observado ante una entrada escalón - Simulink

## 4.5. Control PID neuronal del interfaz háptico

Para el control con redes neuronales descrito en la Sección 2.7, utilizamos el diagrama de bloques mostrado en la Figura 4.39.

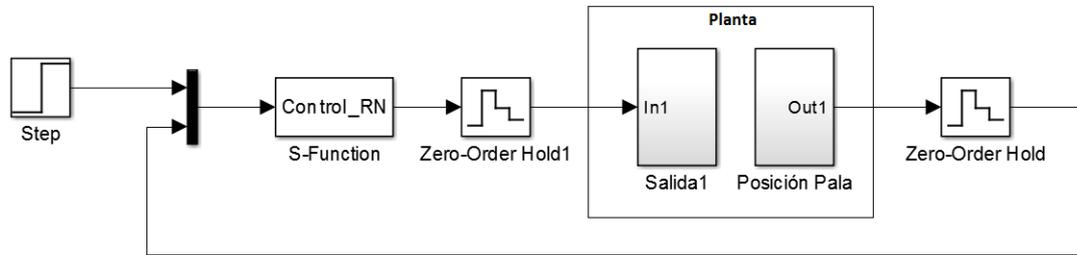


Figura 4.39: Controlador PID neuronal del interfaz háptico - Simulink

El bloque *S-Function* contiene todo el código del controlador con redes neuronales, específicamente, ADALINE.

Primero, se inicializan las variables a utilizar

```
1 OPss=0;
2 PVss=0;
3 SPss=0;
```

y se las guarda en los vectores

```
1 p.OP = [OPss OPss OPss]; % [y(k-1), y(k-2), y(k-3)]
    initialized at ss desde 1867
2 p.PV = [PVss PVss PVss]; % [u(k-1), 0, 0] initialized at ss
3 p.SP = [SPss SPss SPss]; % [w(k), w(k-1), w(k-2)] initialized
    at ss
```

a estos vectores los guardamos en el archivo *pid*

```
1 save pid p;
```

Dentro de la función *mdlOutputs* del bloque *S-Function* agregamos el siguiente código

```
1 function sys = mdlOutputs(t,x,u)
2
3 gama = 1e-5; % constante de aprendizaje
4
5 if t>0
6 load pid; %e cargan los vectores guardados en pid
7
8 ref = u(1); %e obtiene la entrada u(1)
9 y = u(2); %e obtiene la entrada u(2)
10 ER0 = ref-y; %e obtiene el error actual
11 ER1 = ref-p.OP(1); %e obtiene el error anterior
12 ER2 = ref-p.OP(2); %e obtiene el error hace 2 muestras
13 ER3 = ref-p.OP(3); %e obtiene el error hace 3 muestras
```



```
14 w = p.SP; %e carga vector de pesos
15
16 uc = w(1)*ER1 + w(2)*ER2 + w(3)*ER3 + ER0; %procesamiento del
    error
17 w = [ER0* gama *ER1 ER0* gama *ER2 ER0* gama *ER3] + w; %
    actualizacion pesos
18
19 p.OP = [y p.OP(1) p.OP(2)]; %e actualiza el vector p.OP
20 p.PV = [uc p.PV(1) p.PV(2)]; %e actualiza el vector p.PV
21 p.SP = w; %e actualiza el vector p.SP
22
23 save pid p; %e guardan los vectores p.OP, p.PV y p.SP en pid
24
25 sys = uc; %e escribe el error procesado en la salida
26
27 else
28 sys=0;
29 end
```

Los resultados del control mediante redes neuronales se describen a continuación.

La Figura 4.40 muestra la respuesta de la planta, pudiéndose observar una respuesta rápida del controlador. Las oscilaciones que se observan en la Figura 4.40 se deben a perturbaciones del sistema. Algo claro que también se nota es que el error no se reduce a cero en todos los casos. Hay que recordar que este tipo de controlador depende en su mayoría de la tasa de aprendizaje,  $\gamma$ , por lo que la variación de este parámetro es clave para encontrar la mejor respuesta de la planta.

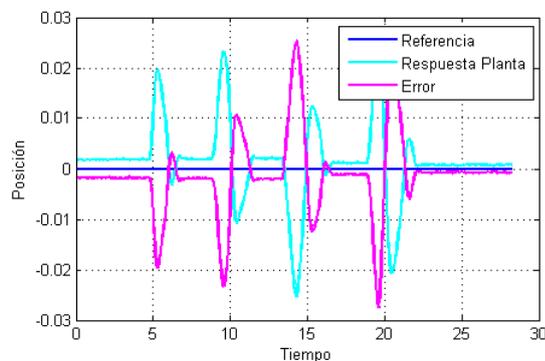


Figura 4.40: Respuesta del controlador PID neuronal del interfaz háptico - Simulink

La Figura 4.41 muestra la respuesta al escalón de la planta utilizando el controlador con redes neuronales, tanto para una referencia positiva como para una negativa. La respuesta no presenta oscilaciones pero si un error de estado permanente.

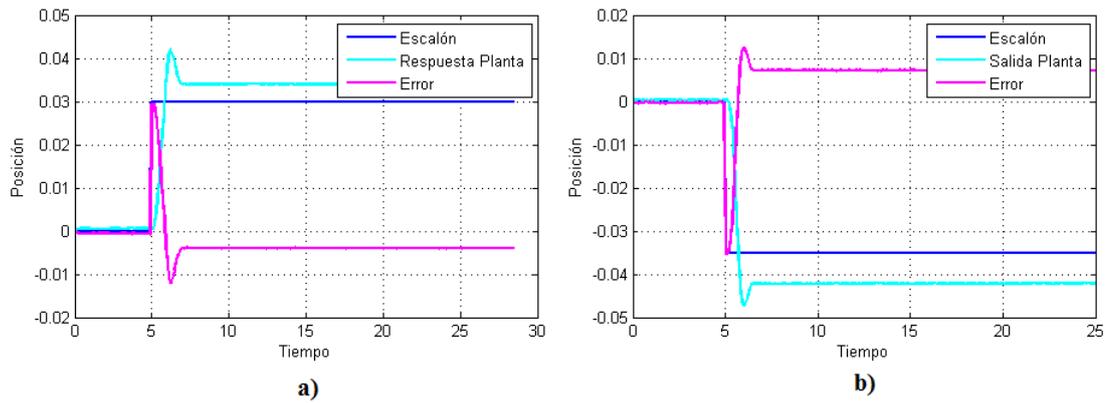


Figura 4.41: Respuesta del controlador con redes neuronales ante una referencia escalón a) positiva y b) negativa

## Capítulo 5

---

# Aplicaciones con la interfaz háptica

---

### 5.1. Péndulo invertido virtual

El péndulo invertido es un sistema comúnmente utilizado en sistemas de control debido a la inestabilidad que presenta si no posee un control.

La perspectiva de control en esta tesis cambia, debido a que la presencia de la interfaz háptica trae consigo la forma o manera de controlar el péndulo invertido y los sistemas en general, ya que el que se encarga de controlar el péndulo no es el sistema de control en su totalidad, sino que se complementa con la retroalimentación del operador que está interactuando con el entorno virtual.

El modelo del péndulo se lo realizo en el programa Solidworks. Consta de dos partes, básicamente: un soporte para el péndulo y el péndulo mismo, las dos piezas se pueden observar en las Figuras 5.1 y 5.2, respectivamente.

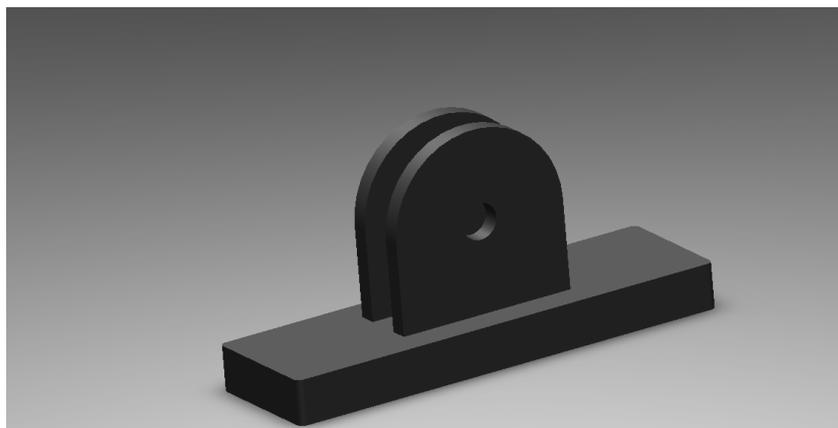


Figura 5.1: Base del péndulo invertido virtual - Solidworks



Figura 5.2: Péndulo virtual - Solidworks

Luego de haber construido las dos piezas, se procedió a ensamblarlas con el mismo programa, resultando el modelo que se observa en la Figura 5.3. Cabe señalar que una vez ensamblado el péndulo en Solidworks, el sistema adquiere todos los parámetros físicos involucrados en la simulación del péndulo invertido, por lo que no se necesitan explícitamente las ecuaciones dinámicas que modelan dicho péndulo.



Figura 5.3: Ensamble final del péndulo invertido virtual - Solidworks

El modelo es exportado desde Solidworks a MATLAB para realizar el control con las herramientas Simulink y Simmechanics. Las Figuras 5.4 y 5.5 muestran el diagrama de bloques en Simulink y el entorno gráfico de Simmechanics, respectivamente.

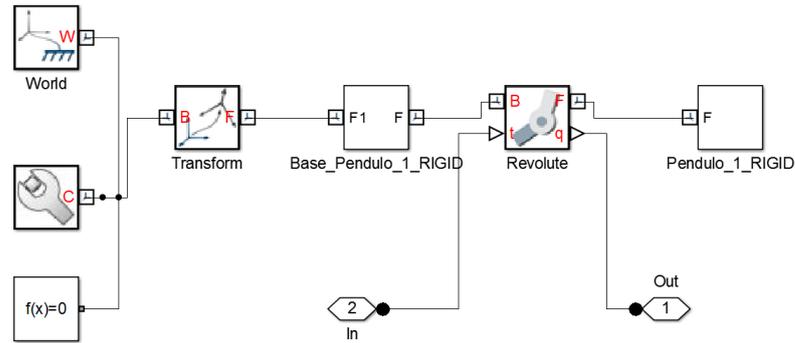


Figura 5.4: Diagrama de bloques importado a Simulink desde Solidworks

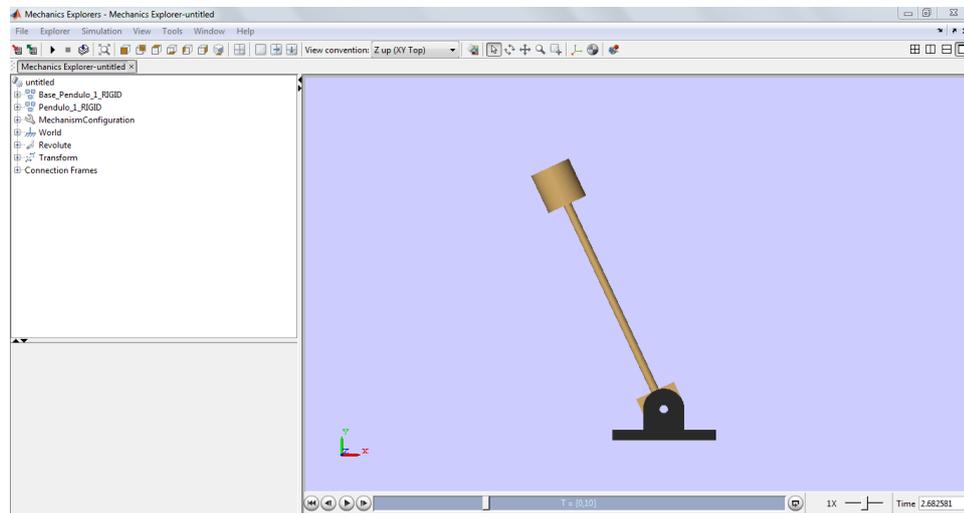


Figura 5.5: Interfaz gráfica en Simmechanics del péndulo virtual

### 5.1.1. Control del péndulo invertido en lazo abierto

La Figura 5.6 muestra el diagrama de bloques del péndulo en lazo abierto. Este tipo de control permite la interacción del usuario con el entorno virtual, permitiéndole que sea él, el que controle el péndulo de manera manual, mediante la realimentación de su visión.

El bloque *VR\_Pendolo* de la Figura 5.6 contiene todos los elementos del péndulo virtual que constan en la Figura 5.4.

En las propiedades del bloque *Revolute* de la Figura 5.4, podemos cambiar los valores de los parámetros del péndulo virtual como el amortiguamiento, rigidez, etc. También podemos seleccionar si existe, o no, realimentación, y qué magnitudes queremos medir: posición, velocidad o aceleración, ver Figura 5.7. Para este control seleccionamos la opción de posición.

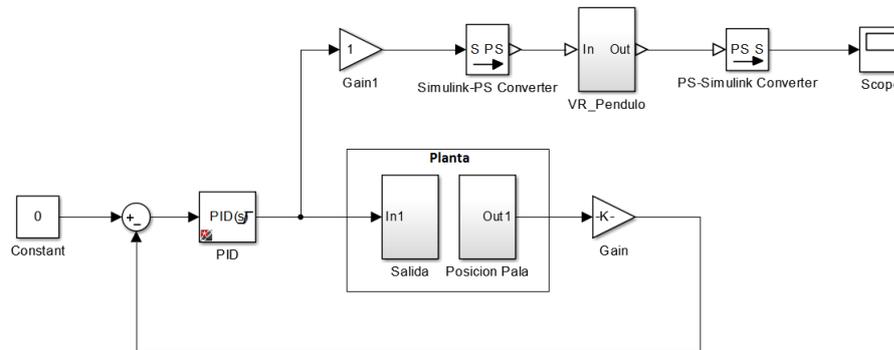


Figura 5.6: Control en lazo abierto del péndulo invertido virtual - Simulink

Tabla 5.1: Valores del PID para el control en lazo abierto del péndulo invertido virtual

Parámetros PID	
$k_c$	17
$T_i$	0.35
$T_d$	0.07

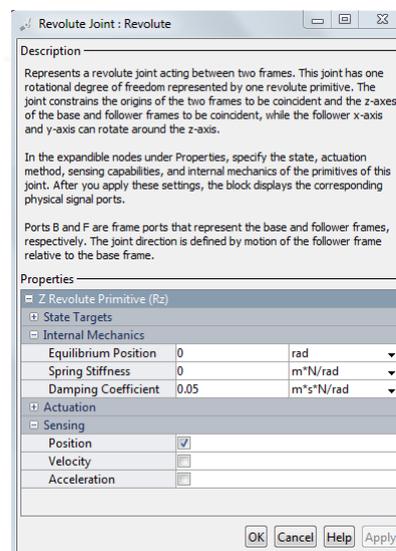


Figura 5.7: Interfaz de las propiedades del bloque *Revolute* - Simulink

El control PID que se observa en la Figura 5.6 es para la interfaz háptica, más no para el péndulo invertido. Los parámetros del PID para esta aplicación se presentan en la Tabla 5.1.

El control consiste en estabilizar el péndulo virtual con el torque provisto por la interfaz háptica. La referencia del control se encuentra en cero para poder dar

el mismo torque en las dos direcciones posibles de movimiento.

### 5.1.2. Control del péndulo invertido en lazo cerrado

La Figura 5.8 muestra el diagrama de bloques utilizado para el control en lazo cerrado del péndulo invertido. Se utilizan dos controles en cascada, uno para la interfaz háptica y el otro para el péndulo virtual.

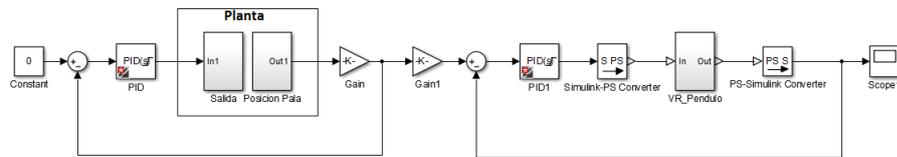


Figura 5.8: Control en lazo cerrado del péndulo invertido virtual - Simulink

La interfaz háptica, a diferencia del control en lazo abierto que entrega directamente el torque al péndulo virtual, esta vez, entrega su posición como referencia al péndulo invertido virtual.

Las Figuras 5.9 y 5.10 presentan la respuesta de salida de los dos controles, tanto de la interfaz háptica como la del péndulo invertido, respectivamente. Los parámetros de los dos PID utilizados se presentan en la Tabla 5.2.

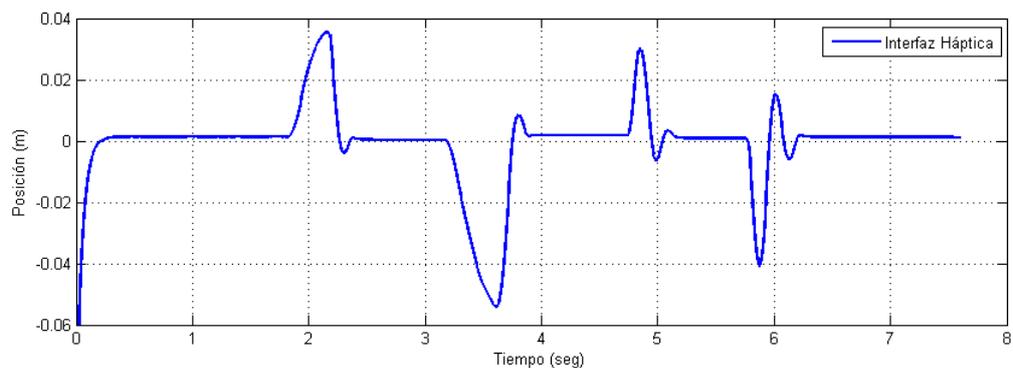


Figura 5.9: Respuesta del controlador de la interfaz háptica en lazo cerrado para la aplicación del péndulo invertido - Simulink

Tabla 5.2: Valores del PID para el control en lazo cerrado del péndulo invertido virtual

	PID Péndulo	PID Interfaz Háptica
$k_c$	5	16
$T_i$	0	0.3
$T_d$	0.05	0.07

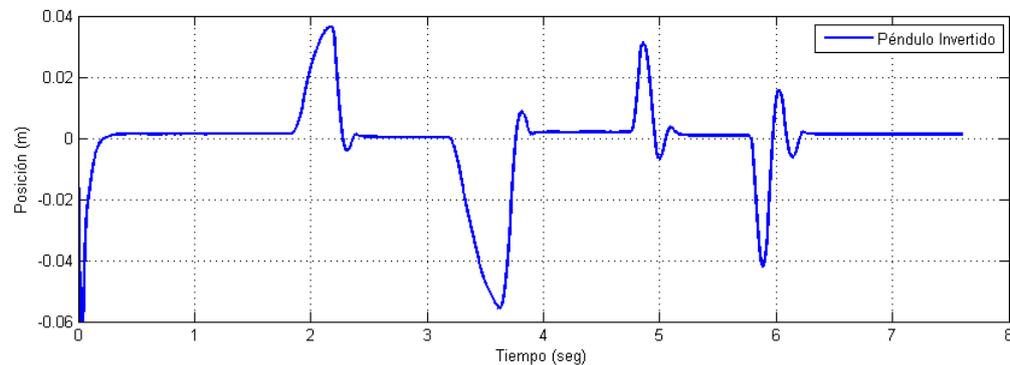


Figura 5.10: Respuesta del controlador del péndulo invertido virtual en lazo cerrado - Simulink

## 5.2. Seguidor del movimiento de una rodilla humana

La interfaz háptica tiene varios campos de aplicación como se lo ha mencionado a lo largo de esta tesis. Una de ellas, es la ayuda que puede brindar a personas que tienen algún tipo de impedimento físico en cualquier articulación del cuerpo humano y que impide su libre movimiento.

En esta sección, se presenta una aplicación que puede ser útil para un estudio más amplio de la interfaz háptica, que por ejemplo, puede ser un exoesqueleto. Se presenta un sistema de seguimiento de una articulación humana, específicamente, la rodilla.

En primer lugar, se procedió a recolectar datos del movimiento de la rodilla con ayuda del programa Kinovea.

En el programa se carga un video<sup>1</sup>, ver Figura 5.11, y se hace el seguimiento de un punto en específico que deseemos, obviamente, en nuestro caso se hizo el

<sup>1</sup><https://www.youtube.com/watch?v=xbT4Z3bn1uA>

seguimiento de la rodilla.

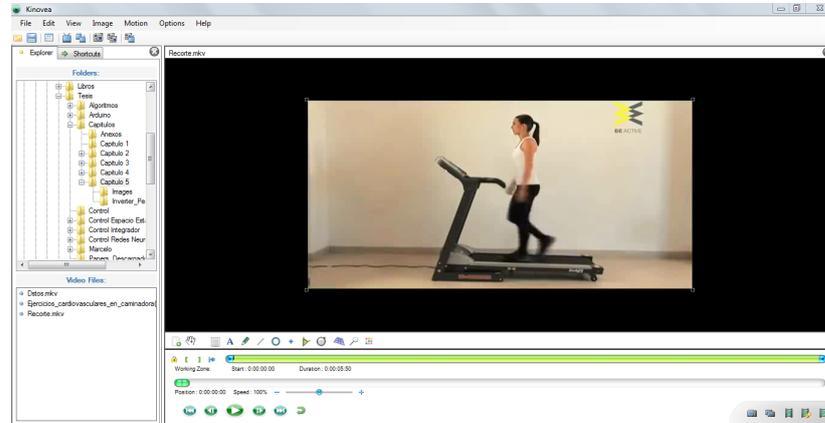


Figura 5.11: Interfaz del programa Kinovea

Para la medida del desplazamiento angular de la articulación se tomó como referencia la cadera de la persona y el piso, tal como se observa en la Figura 5.12.



Figura 5.12: Ángulo medido en la recolección de datos

La cantidad medida por el programa la restamos de  $90^\circ$  y obtenemos el ángulo de desplazamiento de la pala háptica.

Los datos son exportados a Excel y posteriormente a MATLAB. La Figura 5.13 muestra la señal obtenida producto del movimiento de la rodilla a partir de los datos recolectados.

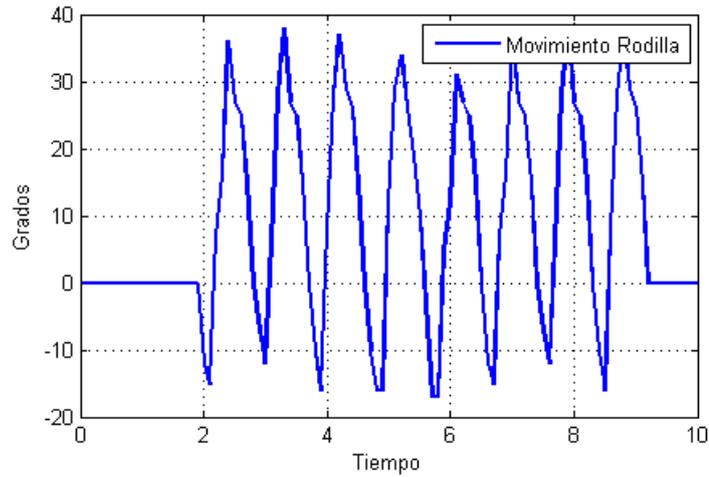


Figura 5.13: Señal obtenida de los datos del movimiento de la rodilla, utilizada como referencia en el seguidor

Se utiliza un PID como sistema de seguimiento. El diagrama de bloques en Simulink se presenta en la Figura 5.14.

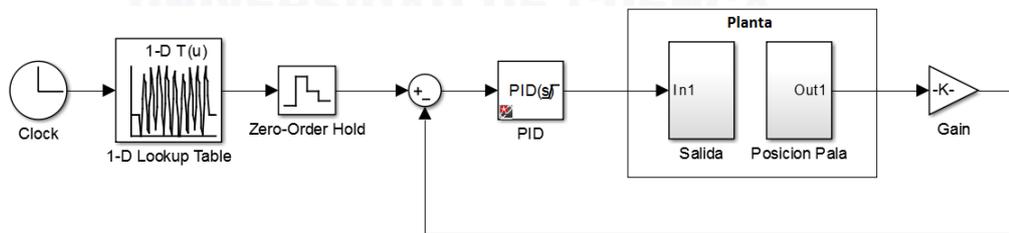


Figura 5.14: Diagrama de bloques para el seguidor de movimiento de la rodilla - Simulink

En el bloque de entrada se carga la señal que contiene el movimiento de la rodilla. Los datos están muestreados cada 0.1 segundos.

Los valores de los parámetros del PID utilizados para el sistema de seguimiento se presentan en la Tabla 5.3.

Tabla 5.3: Valores del PID para el seguidor de movimiento de la rodilla

Parámetros PID	
$k_c$	5
$T_i$	0.43
$T_d$	0.1

La Figura 5.15 muestra el seguimiento de la interfaz con respecto a la señal de referencia, pudiéndose observar un seguimiento bastante aceptable, existiendo un error considerable en los picos superiores de la señal de referencia.

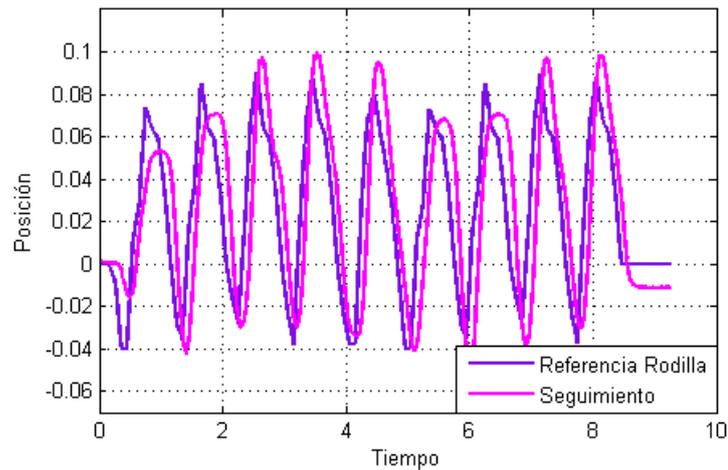


Figura 5.15: Respuesta del seguidor ante el movimiento de la rodilla

### 5.3. Evaluación de la interfaz háptica

En esta sección se realiza una evaluación del péndulo virtual invertido de acuerdo a algunos estamentos de la Sección 2.4 y al diagrama de la Figura 2.5, a partir de los cuales se puede decir lo siguiente:

- La pala háptica no presenta restricciones en su movimiento. La respuesta del motor Maxon, ante cambios de referencia, no influye en el comportamiento del interfaz, además, su fuerza de fricción interna es muy baja como se determinó en la Sección 4.2.3.
- Debido a que no se presenta más de un objeto virtual, la diferenciación de densidades de los objetos virtuales es imposible.
- El diseño realizado en Solidworks, al poseer un grado de libertad, representa un costo computacional bajo. La estabilidad que se obtiene del péndulo invertido en Simmechanics es permanente, sin presentar interrupciones una vez que la aplicación ha iniciado.
- El dispositivo cumple con los parámetros de operación de transparencia y estabilidad; tanto en lazo abierto como en lazo cerrado, el controlador



## CAPÍTULO 5. APLICACIONES CON LA INTERFAZ HÁPTICA

---

PID no tiene respuestas negativas que provoquen una mala interacción del usuario dentro del ambiente virtual.

- La interacción se torna transparente e intuitiva por parte del usuario, la calibración de la fuerza de fricción entre la pala háptica y la rueda motriz es mucho menor que la fuerza mínima que debe aplicar el usuario cuando manipula el péndulo invertido, sin provocar, de esa manera, agotamiento en el usuario mientras éste interactúa con el ambiente virtual.



UNIVERSIDAD DE CUENCA  
desde 1867

## Capítulo 6

---

### Conclusiones

---

- Las pruebas de hardware de la pala háptica mostraron problemas que se solventaron utilizando métodos empíricos. Por ejemplo, fue usada una banda adicional en la rueda motriz, debido a que no se tenía la suficiente tracción entre los dos objetos, lo cual, dificultó el control del interfaz háptico, provocando un error constante en la referencia de la pala.
- La fricción entre la pala háptica y la rueda motriz debe ser mínima, asimismo, se debe asegurar que la pala háptica gire libremente en su punto de rotación sin presentar ninguna oposición al movimiento. Estos dos factores influyen de manera directa en el control del interfaz, ocasionando un gasto variable de energía dependiendo del caso.
- El desempeño que presenta la interfaz háptica permite la interacción en tiempo real. La velocidad en la comunicación serial no influye en el control del interfaz háptico. Se realizaron pruebas entre los 19200 y 115200 baudios, obteniendo similares rendimientos del interfaz.
- El tiempo de muestreo influye directamente en el control con PID, provocando un retardo en la respuesta del controlador si se utilizan menos muestras por segundo, es decir, un tiempo de muestreo grande. En el caso del control moderno, el tiempo de muestreo está implícito en el diseño de los controladores, por lo que no afecta en la práctica. El controlador PID neuronal, no ve influenciado su desempeño con el tiempo de muestreo debido a sus características de adaptación.
- La identificación paramétrica del sistema para el control mediante espacio de estados no siempre arroja una aproximación exacta del sistema, especialmente de la matriz de realimentación, por lo que se ajustó este parámetro



experimentalmente en cada uno de los controladores modernos realizados.

- En el control mediante PID neuronal la constante de aprendizaje influye de manera directa en la convergencia del algoritmo de control haciendo que el sistema responda de manera lenta o rápida, por lo que es el valor más importante a considerar dentro del diseño. El valor de la constante de aprendizaje puede variar entre 0 y 1.
- El péndulo invertido virtual presenta un claro ejemplo del manejo de la interfaz háptica tanto en lazo abierto como en lazo cerrado, mostrando un buen rendimiento en tiempo real, con cero error de fase entre la interfaz y el péndulo virtual. Por tanto, se concluye que la interfaz cumple con su cometido inicial de interacción en ambientes virtuales.
- El seguimiento de la rodilla presenta un error que con un PID clásico no se puede disminuir, por lo que se necesita un controlador moderno para reducir dicho error.
- En la evaluación del interfaz háptico, queda claro que se necesitan más aplicaciones para realizar una evaluación más completa, así como del número de usuarios que puedan definir experimentalmente la calidad y rendimiento del mismo.

---

## Anexos

---



---

**UNIVERSIDAD DE CUENCA**  
desde 1867

---

## Anexo A

---

# Hoja de especificaciones del driver L298

---



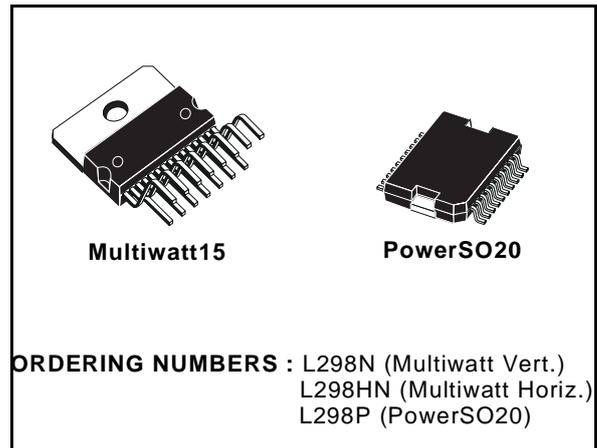
UNIVERSIDAD DE CUENCA  
desde 1867

## DUAL FULL-BRIDGE DRIVER

- OPERATING SUPPLY VOLTAGE UP TO 46 V
- TOTAL DC CURRENT UP TO 4 A
- LOW SATURATION VOLTAGE
- OVERTEMPERATURE PROTECTION
- LOGICAL "0" INPUT VOLTAGE UP TO 1.5 V (HIGH NOISE IMMUNITY)

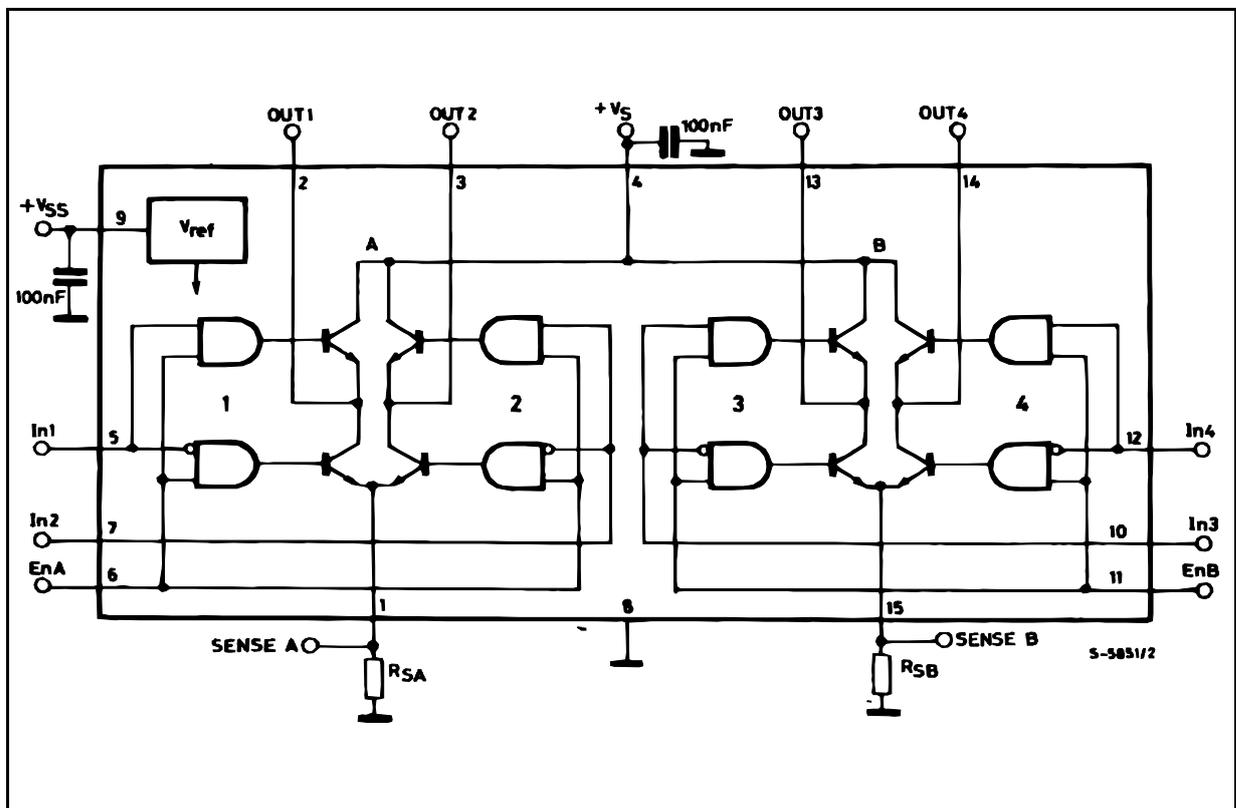
### DESCRIPTION

The L298 is an integrated monolithic circuit in a 15-lead Multiwatt and PowerSO20 packages. It is a high voltage, high current dual full-bridge driver designed to accept standard TTL logic levels and drive inductive loads such as relays, solenoids, DC and stepping motors. Two enable inputs are provided to enable or disable the device independently of the input signals. The emitters of the lower transistors of each bridge are connected together and the corresponding external terminal can be used for the con-



nection of an external sensing resistor. An additional supply input is provided so that the logic works at a lower voltage.

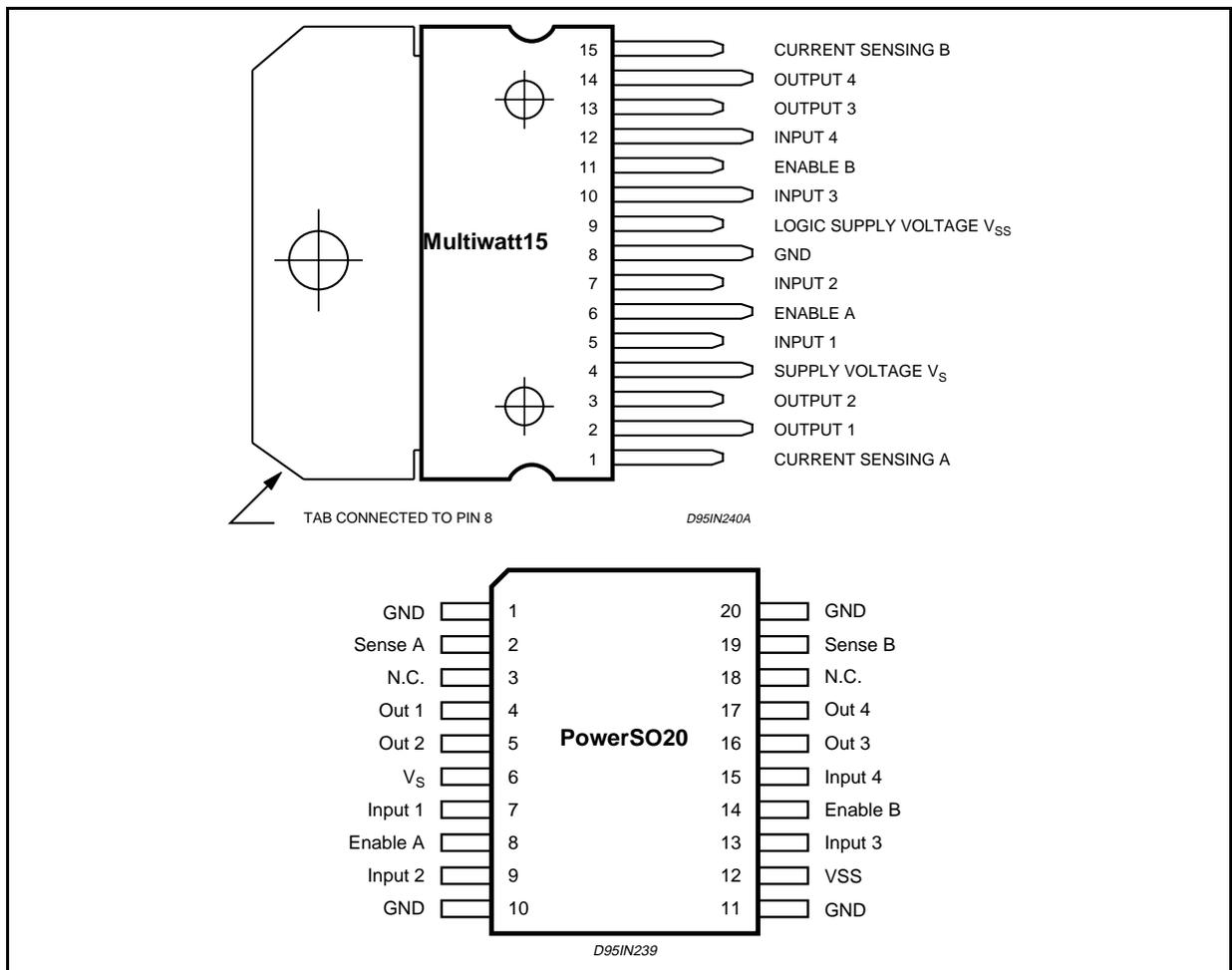
### BLOCK DIAGRAM



**ABSOLUTE MAXIMUM RATINGS**

Symbol	Parameter	Value	Unit
$V_S$	Power Supply	50	V
$V_{SS}$	Logic Supply Voltage	7	V
$V_i, V_{en}$	Input and Enable Voltage	-0.3 to 7	V
$I_o$	Peak Output Current (each Channel) – Non Repetitive ( $t = 100\mu s$ ) – Repetitive (80% on –20% off; $t_{on} = 10ms$ ) – DC Operation	3 2.5 2	A A A
$V_{sens}$	Sensing Voltage	-1 to 2.3	V
$P_{tot}$	Total Power Dissipation ( $T_{case} = 75^\circ C$ )	25	W
$T_{op}$	Junction Operating Temperature	-25 to 130	$^\circ C$
$T_{stg}, T_j$	Storage and Junction Temperature	-40 to 150	$^\circ C$

**PIN CONNECTIONS (top view)**



**THERMAL DATA**

Symbol	Parameter	PowerSO20	Multiwatt15	Unit
$R_{th\ j-case}$	Thermal Resistance Junction-case	Max. –	3	$^\circ C/W$
$R_{th\ j-amb}$	Thermal Resistance Junction-ambient	Max. 13 (*)	35	$^\circ C/W$

(\*) Mounted on aluminum substrate

**PIN FUNCTIONS** (refer to the block diagram)

MW.15	PowerSO	Name	Function
1;15	2;19	Sense A; Sense B	Between this pin and ground is connected the sense resistor to control the current of the load.
2;3	4;5	Out 1; Out 2	Outputs of the Bridge A; the current that flows through the load connected between these two pins is monitored at pin 1.
4	6	V <sub>S</sub>	Supply Voltage for the Power Output Stages. A non-inductive 100nF capacitor must be connected between this pin and ground.
5;7	7;9	Input 1; Input 2	TTL Compatible Inputs of the Bridge A.
6;11	8;14	Enable A; Enable B	TTL Compatible Enable Input: the L state disables the bridge A (enable A) and/or the bridge B (enable B).
8	1,10,11,20	GND	Ground.
9	12	V <sub>SS</sub>	Supply Voltage for the Logic Blocks. A100nF capacitor must be connected between this pin and ground.
10; 12	13;15	Input 3; Input 4	TTL Compatible Inputs of the Bridge B.
13; 14	16;17	Out 3; Out 4	Outputs of the Bridge B. The current that flows through the load connected between these two pins is monitored at pin 15.
–	3;18	N.C.	Not Connected

**ELECTRICAL CHARACTERISTICS** (V<sub>S</sub> = 42V; V<sub>SS</sub> = 5V, T<sub>J</sub> = 25°C; unless otherwise specified)

Symbol	Parameter	Test Conditions	Min.	Typ.	Max.	Unit
V <sub>S</sub>	Supply Voltage (pin 4)	Operative Condition	V <sub>IH</sub> +2.5		46	V
V <sub>SS</sub>	Logic Supply Voltage (pin 9)		4.5	5	7	V
I <sub>S</sub>	Quiescent Supply Current (pin 4)	V <sub>en</sub> = H; I <sub>L</sub> = 0	V <sub>i</sub> = L	13	22	mA
			V <sub>i</sub> = H	50	70	mA
I <sub>SS</sub>	Quiescent Current from V <sub>SS</sub> (pin 9)	V <sub>en</sub> = L	V <sub>i</sub> = X		4	mA
		V <sub>en</sub> = H; I <sub>L</sub> = 0	V <sub>i</sub> = L	24	36	mA
			V <sub>i</sub> = H	7	12	mA
		V <sub>en</sub> = L	V <sub>i</sub> = X		6	mA
V <sub>iL</sub>	Input Low Voltage (pins 5, 7, 10, 12)		–0.3		1.5	V
V <sub>iH</sub>	Input High Voltage (pins 5, 7, 10, 12)		2.3		V <sub>SS</sub>	V
I <sub>iL</sub>	Low Voltage Input Current (pins 5, 7, 10, 12)	V <sub>i</sub> = L			–10	μA
I <sub>iH</sub>	High Voltage Input Current (pins 5, 7, 10, 12)	V <sub>i</sub> = H ≤ V <sub>SS</sub> –0.6V		30	100	μA
V <sub>en</sub> = L	Enable Low Voltage (pins 6, 11)		–0.3		1.5	V
V <sub>en</sub> = H	Enable High Voltage (pins 6, 11)		2.3		V <sub>SS</sub>	V
I <sub>en</sub> = L	Low Voltage Enable Current (pins 6, 11)	V <sub>en</sub> = L			–10	μA
I <sub>en</sub> = H	High Voltage Enable Current (pins 6, 11)	V <sub>en</sub> = H ≤ V <sub>SS</sub> –0.6V		30	100	μA
V <sub>CEsat</sub> (H)	Source Saturation Voltage	I <sub>L</sub> = 1A	0.95	1.35	1.7	V
		I <sub>L</sub> = 2A		2	2.7	V
V <sub>CEsat</sub> (L)	Sink Saturation Voltage	I <sub>L</sub> = 1A (5)	0.85	1.2	1.6	V
		I <sub>L</sub> = 2A (5)		1.7	2.3	V
V <sub>CEsat</sub>	Total Drop	I <sub>L</sub> = 1A (5)	1.80		3.2	V
		I <sub>L</sub> = 2A (5)			4.9	V
V <sub>sens</sub>	Sensing Voltage (pins 1, 15)		–1 (1)		2	V

## Anexo B

---

# Hoja de especificaciones del motor Maxon 353144

---

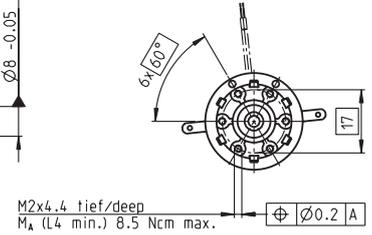
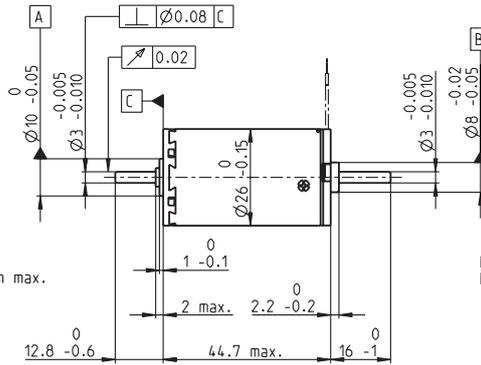
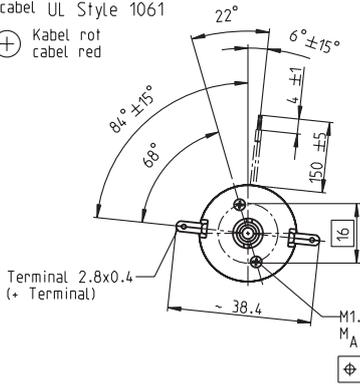


UNIVERSIDAD DE CUENCA  
desde 1867

# A-max 26 Ø26 mm, Graphite Brushes, 6 Watt

Kabel AWG 24/7  
cabel UL Style 1061

⊕ Kabel rot  
cabel red



M 1:2

- Stock program
- Standard program
- Special program (on request)

## Order Number

with terminals	110946	110947	110948	110949	110950	110951	110952	110953	110954	110955	110956	110957
with cables	353143	353144	353145	353146	353147	353148	353149	353150	353151	353152	353153	353154

## Motor Data

Values at nominal voltage																		
1	Nominal voltage	V	7.2	9.0	12.0	12.0	18.0	18.0	24.0	24.0	30.0	36.0	42.0	48.0				
2	No load speed	rpm	9270	10000	10000	8300	8260	7410	8590	7870	8810	8440	8170	6240				
3	No load current	mA	118	104	76.8	59.7	39.2	34.0	30.8	27.6	25.4	20.0	16.4	10.3				
4	Nominal speed	rpm	7160	7620	7600	5590	5640	4790	5880	5100	6210	5850	5550	3550				
5	Nominal torque (max. continuous torque)	mNm	6.73	7.97	11.1	13.0	13.6	13.8	13.1	12.9	13.7	13.8	13.7	13.7				
6	Nominal current (max. continuous current)	A	1.08	1.08	1.08	1.03	0.708	0.642	0.532	0.481	0.452	0.365	0.300	0.201				
7	Stall torque	mNm	38.2	39.7	52.7	43.8	45.6	41.0	43.5	38.1	47.9	46.4	43.7	32.6				
8	Starting current	A	5.50	4.90	4.80	3.29	2.25	1.82	1.67	1.34	1.51	1.16	0.911	0.455				
9	Max. efficiency	%	67	69	73	72	74	73	74	73	75	75	75	72				
Characteristics																		
10	Terminal resistance	Ω	1.31	1.84	2.50	3.65	8.00	9.91	14.4	17.9	19.9	31.0	46.1	106				
11	Terminal inductance	mH	0.101	0.138	0.254	0.372	0.862	1.07	1.42	1.69	2.13	3.35	4.85	10.8				
12	Torque constant	mNm / A	6.94	8.09	11.0	13.3	20.2	22.5	26.0	28.3	31.8	39.9	48.0	71.6				
13	Speed constant	rpm / V	1380	1180	869	718	472	423	367	337	300	239	199	133				
14	Speed / torque gradient	rpm / mNm	260	268	198	197	186	186	203	213	188	186	191	197				
15	Mechanical time constant	ms	33.4	30.5	27.9	27.1	25.4	25.2	24.9	24.9	24.5	24.3	24.2	24.2				
16	Rotor inertia	gcm <sup>2</sup>	12.3	10.9	13.5	13.2	13.0	12.9	11.7	11.2	12.5	12.5	12.1	11.7				

## Specifications

- Thermal data**
- 17 Thermal resistance housing-ambient 13.2 K / W
  - 18 Thermal resistance winding-housing 3.2 K / W
  - 19 Thermal time constant winding 12.4 s
  - 20 Thermal time constant motor 660 s
  - 21 Ambient temperature -30 ... +85°C
  - 22 Max. permissible winding temperature +125°C

- Mechanical data (ball bearings)**
- 23 Max. permissible speed 10400 rpm
  - 24 Axial play 0.1 - 0.2 mm
  - 25 Radial play 0.025 mm
  - 26 Max. axial load (dynamic) 5 N
  - 27 Max. force for press fits (static) (static, shaft supported) 75 N
  - 28 Max. radial loading, 5 mm from flange 1200 N

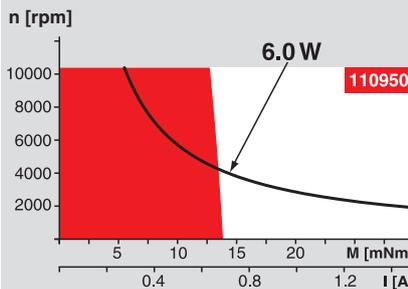
- Mechanical data (sleeve bearings)**
- 23 Max. permissible speed 10400 rpm
  - 24 Axial play 0.1 - 0.2 mm
  - 25 Radial play 0.012 mm
  - 26 Max. axial load (dynamic) 1.7 N
  - 27 Max. force for press fits (static) (static, shaft supported) 80 N
  - 28 Max. radial loading, 5 mm from flange 1200 N

- Other specifications**
- 29 Number of pole pairs 1
  - 30 Number of commutator segments 13
  - 31 Weight of motor 100 g

Values listed in the table are nominal.  
Explanation of the figures on page 49.

**Option**  
Sleeve bearings in place of ball bearings

## Operating Range



## Comments

**Continuous operation**  
In observation of above listed thermal resistance (lines 17 and 18) the maximum permissible winding temperature will be reached during continuous operation at 25°C ambient.  
= Thermal limit.

**Short term operation**  
The motor may be briefly overloaded (recurring).

— Assigned power rating

## maxon Modular System

Overview on page 16 - 21

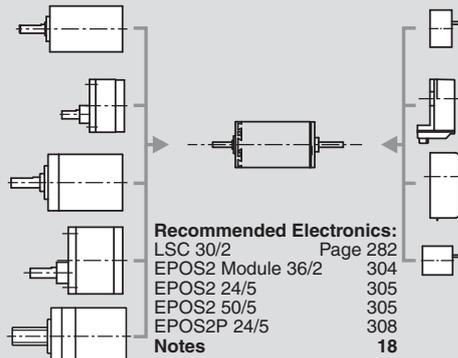
**Planetary Gearhead**  
Ø26 mm  
0.5 - 2.0 Nm  
Page 228

**Spur Gearhead**  
Ø30 mm  
0.07 - 0.2 Nm  
Page 229

**Planetary Gearhead**  
Ø32 mm  
0.4 - 6.0 Nm  
Page 230 / 231 / 233

**Spur Gearhead**  
Ø38 mm  
0.1 - 0.6 Nm  
Page 237

**Spindle Drive**  
Ø32 mm  
Page 249 / 250 / 251



**Recommended Electronics:**  
LSC 30/2 Page 282  
EPOS2 Module 36/2 304  
EPOS2 24/5 305  
EPOS2 50/5 305  
EPOS2P 24/5 308  
**Notes 18**

**Encoder MR**  
128 - 1000 Imp.,  
3 channels  
Page 262

**Encoder Enc**  
22 mm  
100 Imp., 2 channels  
Page 265

**Encoder HED\_ 5540**  
500 Imp.,  
3 channels  
Page 267 / 269

**Encoder MEnc**  
Ø13 mm  
16 Imp., 2 channels  
Page 275

## Anexo C

---

# Raíces de los polinomios normalizados de Bessel

---



UNIVERSIDAD DE CUENCA  
desde 1867

Tabla C.1: Raíces de los polinomios normalizados de Bessel

<b>k</b>									
1	-4,62								
2	-4,053 ± j2,34								
3	-5,0083	-3,9668 ± j3,7845							
4	-4,0156 ± j5,0723	-5,5281 ± j1,6553							
5	-6,448	-4,1104 ± j6,3142	-5,9268 ± j3,0813						
6	-4,2169 ± j7,53	-6,2613 ± j4,4018	-7,1205 ± j1,454						
7	-8,0271	-4,3361 ± j8,7519	-6,5714 ± j5,6786	-7,6824 ± j2,8081					
8	-4,4554 ± j9,9715	-6,8554 ± j6,9278	-8,1682 ± j4,1057	-8,7693 ± j1,3616					
9	9,6585	-4,5696 ± j11,1838	-7,1145 ± j8,1557	-8,5962 ± j5,3655	-9,40113 ± j2,6655				
10	-4,6835 ± j12,4022	-7,3609 ± j9,3777	8,9898 ± j6,6057	-9,9657 ± j3,9342	-10,4278 ± 1,307				



*ANEXO C. RAÍCES DE LOS POLINOMIOS NORMALIZADOS DE BESSEL*

---



UNIVERSIDAD DE CUENCA  
desde 1867

---

## Bibliografía

---

- [1] A. Gallace and C. Spence, “The science of interpersonal touch: an overview,” *Neuroscience & Biobehavioral Reviews*, vol. 34, no. 2, pp. 246–259, 2010.
- [2] K. E. MacLean, “Designing with haptic feedback,” in *Robotics and Automation, 2000. Proceedings. ICRA '00. IEEE International Conference on*, vol. 1. IEEE, 2000, pp. 783–788.
- [3] S. D. Laycock and A. Day, “Recent developments and applications of haptic devices,” in *Computer Graphics Forum*, vol. 22, no. 2. Wiley Online Library, 2003, pp. 117–132.
- [4] V. Hayward, O. R. Astley, M. Cruz-Hernandez, D. Grant, and G. Robles-De-La-Torre, “Haptic interfaces and devices,” *Sensor Review*, vol. 24, no. 1, pp. 16–29, 2004.
- [5] F. P. Brooks Jr, “Walkthrough—a dynamic graphics system for simulating virtual buildings,” in *Proceedings of the 1986 workshop on Interactive 3D graphics*. ACM, 1987, pp. 9–21.
- [6] R. J. Stone, “Haptic feedback: A brief history from telepresence to virtual reality,” in *Haptic Human-Computer Interaction*. Springer, 2001, pp. 1–16.
- [7] F. Barbagli and K. Salisbury, “The effect of sensor/actuator asymmetries in haptic interfaces,” in *Haptic Interfaces for Virtual Environment and Teleoperator Systems, 2003. HAPTICS 2003. Proceedings. 11th Symposium on*. IEEE, 2003, pp. 140–147.
- [8] Y.-S. Lee, S. Lin, and R. Narayan, “Interactive forces analysis and haptic modeling for virtual prototyping and product development,” in *Manufacturing Automation (ICMA), 2010 International Conference on*. IEEE, 2010, pp. 271–278.



- [9] A. Gupta and M. K. O'Malley, "Design of a haptic arm exoskeleton for training and rehabilitation," *Mechatronics, IEEE/ASME Transactions on*, vol. 11, no. 3, pp. 280–289, 2006.
- [10] B. Hannaford and J.-H. Ryu, "Time-domain passivity control of haptic interfaces," *Robotics and Automation, IEEE Transactions on*, vol. 18, no. 1, pp. 1–10, 2002.
- [11] U. Unterhinninghofen, T. Schauß, and M. Buss, "Control of a mobile haptic interface," in *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on*. IEEE, 2008, pp. 2085–2090.
- [12] V. Hayward and O. R. Astley, "Performance measures for haptic interfaces," in *Robotics research*. Springer, 1996, pp. 195–206.
- [13] C. Martín, "Interfaces hápticos, aplicación en entornos virtuales," in *Memorias del XVI Congreso Internacional de Ingeniería Gráfica*, 2004.
- [14] S. D. Eppinger and W. P. Seering, "Understanding bandwidth limitations in robot force control," in *Robotics and Automation. Proceedings. 1987 IEEE International Conference on*, vol. 4. IEEE, 1987, pp. 904–909.
- [15] D. Hecht, M. Reiner, and G. Halevy, "Multimodal virtual environments: response times, attention, and presence," *Presence: Teleoperators and virtual environments*, vol. 15, no. 5, pp. 515–523, 2006.
- [16] W.-C. Wu, C. Basdogan, and M. A. Srinivasan, "Visual, haptic, and bimodal perception of size and stiffness in virtual environments," *ASME DYN SYST CONTROL DIV PUBL DSC.*, vol. 67, pp. 19–26, 1999.
- [17] F. Danieau, A. Lécuyer, P. Guillotel, J. Fleureau, N. Mollet, and M. Christie, "Enhancing audiovisual experience with haptic feedback: a survey on hav," *Haptics, IEEE Transactions on*, vol. 6, no. 2, pp. 193–205, 2013.
- [18] T. W. Fong, F. Conti, S. Grange, and C. Baur, "Novel interfaces for remote driving: gesture, haptic, and pda," in *Intelligent Systems and Smart Manufacturing*. International Society for Optics and Photonics, 2001, pp. 300–311.
- [19] A. Bolopion and S. Régnier, "A review of haptic feedback teleoperation systems for micromanipulation and microassembly," *Automation Science and Engineering, IEEE Transactions on*, vol. 10, no. 3, pp. 496–502, 2013.



- [20] M. Benali-Khoudja, M. Hafez, J.-M. Alexandre, and A. Kheddar, "Tactile interfaces: a state-of-the-art survey," in *Int. Symposium on Robotics*, vol. 31. Citeseer, 2004, pp. 23–26.
- [21] B. Varalakshmi, J. Thriveni, K. Venugopal, and L. Patnaik, "Haptics: state of the art survey," *Int. J. Comput. Sci. Issues (IJCSI)*, vol. 9, no. 5, pp. 234–244, 2012.
- [22] E. Samur, F. Wang, U. Spaelter, and H. Bleuler, "Generic and systematic evaluation of haptic interfaces based on testbeds," in *Intelligent Robots and Systems, 2007. IROS 2007. IEEE/RSJ International Conference on*. IEEE, 2007, pp. 2113–2119.
- [23] M. A. Srinivasan and C. Basdogan, "Haptics in virtual environments: Taxonomy, research status, and challenges," *Computers & Graphics*, vol. 21, no. 4, pp. 393–404, 1997.
- [24] A. E. Kirkpatrick and S. A. Douglas, "Application-based evaluation of haptic interfaces," in *Haptic Interfaces for Virtual Environment and Teleoperator Systems, 2002. HAPTICS 2002. Proceedings. 10th Symposium on*. IEEE, 2002, pp. 32–39.
- [25] G. H. Hostetter, C. J. Savant, R. T. Stefani, and A. B. Ramos, *Sistemas de control*. McGraw-Hill, 1990.
- [26] P. Meshram and R. G. Kanojiya, "Tuning of pid controller using ziegler-nichols method for speed control of dc motor," in *Advances in Engineering, Science and Management (ICAESM), 2012 International Conference on*. IEEE, 2012, pp. 117–122.
- [27] C.-C. Yu, *Autotuning of PID controllers: A relay feedback approach*. Springer Science & Business Media, 2006.
- [28] K. Ogata, *Sistemas de control en tiempo discreto*. Pearson educación, 1996.
- [29] Vanderbilt University, "The Vanderbilt Haptic Paddle," [http://research.vuse.vanderbilt.edu/MEDLab/haptic\\_paddle.html](http://research.vuse.vanderbilt.edu/MEDLab/haptic_paddle.html), accessed: 15 December 2015.