

TESIS CARRERA DE MAESTRÍA EN INGENIERÍA

**DISEÑO E IMPLEMENTACIÓN DE DIFERENTES
CONTROLADORES EN UN LEVITADOR MAGNÉTICO PARA
RECHAZO DE PERTURBACIONES DE POSICIÓN**

Ing. Marcelo Javier Castelao Caruana

Maestrando

Ing. Félix Maciel

Director

Ing. Juan Pablo Pierini

Co-Director

Miembros del jurado:

Dr. Andrés Etchepareborda

UNLPam

Dr. Rogelio Hecker

UNLPam

Ing. Andrés Laudari

Instituto Balseiro

Abril 2024

Centro Atómico Bariloche

Instituto Balseiro

Universidad Nacional de Cuyo

Comisión Nacional de Energía Atómica

Argentina

*“Hay que cultivarse. Cuanto más se sabe,
más se puede crear, inventar
y gozar de la vida”*

Quino

1 Glosario de Símbolos

u	Acción de control. Tensión de entrada al electroimán.
R	Resistencia eléctrica
I	Corriente que circula por el electroimán
I_e	Corriente de equilibrio
h	Huelgo entre el elemento levitado y el electroimán
h_e	Huelgo de equilibrio
h_{ref}	Huelgo de referencia
g	Aceleración de la gravedad
L	Inductancia
m	Masa del elemento levitado
K_{mag}	Constante de proporcionalidad entre fuerza magnética y corriente-huelgo
P	Peso del elemento levitado
F_{mag}	Fuerza magnética
ρ	Señal de perturbación
ZOH	Modelo matemático de retenedor de orden cero.
G_{aaf}	Modelo matemático del filtro anti-aliasing

2 Índice

1	Glosario de Símbolos	3
2	Índice	4
3	Resumen	7
4	Abstract	9
5	Introducción	10
5.1	Objetivos	10
5.2	Motivación	10
6	Descripción de la Planta	16
6.1	Esquema experimental	16
6.2	Descripción del hardware utilizado	18
6.2.1	Placa Electrónica	19
6.2.2	Electroimán tipo Ventosa	20
6.2.3	Elemento Levitado	21
6.2.4	Sensor de efecto Hall	22
6.2.5	Motor Vibrador	23
6.2.6	Computadora de Placa Única	23
7	Identificación y Modelado Matemático del Sistema	25
7.1	Caracterización del Tiempo de Muestreo	25
7.2	Medición de los Parámetros de la Planta	28
7.2.1	Identificación del Bloque de corriente	29
7.2.2	Bloque Fuerza Magnética	33
7.2.3	Modelo Matemático	35
7.2.4	Identificación de la Perturbación	41
8	Diseño de Controladores	44

8.1	Definición de los requerimientos del controlador	44
8.2	Control Clásico	45
8.2.1	Lazo Interno	45
8.2.2	Lazo Externo	46
8.2.2.1	Diseño del Controlador del Lazo Interno	47
	Diseño del Controlador del Lazo Externo	48
8.2.3	Respuesta dinámica del sistema	52
8.3	Control Repetitivo	55
8.3.1	Introducción al Control Repetitivo Digital	55
8.3.2	Análisis de estabilidad del Algoritmo de Control Repetitivo	57
8.3.3	Diseño de Control Repetitivo	58
8.4	Diseño por Variable de Estado con Estimador Extendido para Rechazo de Perturbación.	60
8.4.1	Diseño de la Ley de Control	64
8.4.2	Diseño del Estimador Extendido	65
8.4.2.1	Estimador Extendido MIMO	66
8.4.2.2	Estimador a partir de un Filtro de Kalman	69
8.5	Control Adaptativo con Estimador Extendido para Rechazo de Perturbación variable.	71
9	Resultados	75
9.1	Control Clásico	76
9.2	Controlador Repetitivo	78
9.3	Control Moderno con Estimador MIMO (Modelo de Perturbación Fija)	82
9.4	Control Moderno con Filtro de Kalman (Modelo de Perturbación Fija)	87
9.5	Control Moderno con Filtro de Kalman (Modelo Perturbación Variable)	92
9.6	Comparación de Performance de los distintos controladores diseñados	97
10	Conclusiones	101

11	Observaciones, Trabajos a Futuro o Preguntas a Responder	102
12	Apéndices	105
12.1	Instalación de Kernel PREEMPT_RT	105
12.2	Librería LibLevitador.c	106
12.2.1	Inicialización Levitador Magnético	106
12.2.2	Finalización Levitador Magnético	107
12.2.3	Medición Puertos Analógicos	108
12.2.4	Salida PWM	108
12.3	Algoritmos de Control implementados	109
12.3.1	Algoritmo de Control Clásico	109
12.3.2	Algoritmo de Control con Estimador MIMO	126
12.3.3	Algoritmo de Control Moderno con Filtro Kalman Perturbación Variable	134
13	Agradecimientos	146
14	Referencias	147

3 Resumen

En el marco de la Maestría en Ingeniería del Instituto Balseiro, se implementaron controladores diseñados con diferentes técnicas de control para el rechazo de perturbaciones del tipo vibratorias en un levitador magnético. La motivación de realizar este levitador y atenuar perturbaciones del tipo vibratorias radica en los cojinetes magnéticos activos de máquinas rotantes. Este levitador trata de captar la dinámica de un cojinete magnético sometidos a cargas axiales, para luego ensayar diferentes algoritmos de control que atenúen esta vibración.

Como primera etapa se debió diseñar al Levitador en su conjunto, definiendo estructuras, actuador, perturbación, sensores de posición y corriente, placas amplificadoras, y acondicionamiento de señales, placa de control, etcétera.

Como segunda etapa, se caracterizaron e identificaron los distintos parámetros que definen la dinámica del sistema. A partir de estos parámetros, se realizó un modelo matemático del mismo, que permitió diseñar y ensayar los diferentes algoritmos de control a implementar.

En una tercera etapa del trabajo, se estudiaron distintos controladores capaces de rechazar perturbaciones del tipo vibratoria en el elemento levitado. A partir de esta investigación se diseñaron e implementaron distintas técnicas de control al sistema Levitador Magnético. Las técnicas de control implementadas fueron: control clásico, control repetitivo, control moderno con estimador MIMO, control moderno con filtro de Kalman para perturbación de frecuencia fija y filtro de Kalman para perturbación de frecuencia variable.

Por último, se estudió la performance de los controladores implementados para la atenuación de la perturbación introducida al elemento levitado. Para el estudio de la performance, se calculó el espectro de potencia de la señal de posición del elemento levitado, la desviación estándar de la señal posición y

la energía contenida (a través del teorema de Parseval) en la señal de posición con respecto a la posición media.

4 Abstract

In the context of the Master's Degree in Engineering at the Balseiro Institute, different control techniques were implemented for the rejection of vibration-type disturbances in a magnetic levitator. The motivation for realising this levitator and attenuating vibration-type disturbances lies in the active magnetic bearings of rotating machines. This levitator tries to capture the dynamics of a magnetic bearing subjected to axial loads, and then implement different control algorithms to attenuate this vibration.

To do this, the Levitator had to be designed as a whole, defining structures, actuator, disturbance, position and current sensors, amplifier boards, signal conditioning, control board, etcetera.

As a second stage, the different parameters that define the dynamics of the system were characterised and identified. Based on these parameters, a mathematical model of the system was created, which allowed the design and testing of the different control algorithms to be implemented. Then, different controllers capable of rejecting vibration-type disturbances in the levitated element were studied. Based on this research, different control techniques were designed and implemented for the Magnetic Levitator system. The control techniques implemented were: classical control, repetitive control, modern control with MIMO estimator, modern control with Kalman filter for fixed frequency disturbance and Kalman filter for variable frequency disturbance. Finally, the performance of the controllers implemented for the attenuation of the disturbance introduced to the levitated element was studied. For the performance study, the power spectrum of the position signal of the levitated element, the standard deviation of the position signal and the energy contained (through Parseval's theorem) in the position signal with respect to the equilibrium were calculated.

5 Introducción

“Quizá el error estuviera en aceptar que ese objeto era un tornillo por el hecho de que tenía la forma de un tornillo. Picasso toma un auto de juguete y lo convierte en el mentón de un cinocéfalo. A lo mejor el napolitano era un idiota, pero también pudo ser el inventor de un mundo. Del tornillo a un ojo, de un ojo a una estrella... ¿Por qué entregarse a la gran costumbre?”

Rayuela – Julio Cortázar

5.1 Objetivos

El objetivo de este trabajo es estudiar diferentes controladores que permitan rechazar, de manera activa, perturbaciones del tipo vibratorias en un sistema tipo cojinete magnético axial.

Para llevar a cabo este objetivo, como primer paso, se diseñará un levitador magnético, incluyendo elemento levitado, actuador, sensor y hardware, que permita implementar los diferentes controladores. Luego, se identificarán los parámetros del sistema y la perturbación de posición introducida. Se realizará un modelo matemático lineal y no lineal de la planta que permita el diseño y la verificación de los diferentes controladores diseñados. Se implementarán, de forma digital, los algoritmos de control en el levitador para luego comparar su performance ante perturbaciones de posición del elemento levitado.

5.2 Motivación

La levitación magnética consiste en mantener un objeto suspendido sin existir contacto mecánico entre las partes. Los sistemas tipo *MagLev* (Magnetic Levitation) permiten movimientos relativos entre las partes sin fricción, permitiendo alta velocidad relativa, sin desgaste mecánico (con su consecuente no desprendimiento de material), bajo ruido y sin contaminación por lubricación debido a que no existe contacto entre las partes. Por estas

características estos tipos de sistemas son utilizados en trenes de alta velocidad, aislamiento de vibraciones en maquinaria de precisión, rodamientos magnéticos de diversas aplicaciones industriales y tecnológicas. La contracara de estos tipos de sistema es su rápida dinámica inestable y sus marcadas alinealidades [1]. Estas características hacen que el diseño de los lazos de control deba realizarse de manera minuciosa, para lograr una performance adecuada en el punto de operación del levitador.

La tecnología más asociada al termino de Levitación magnética, para el común de la población, es la de los trenes de alta velocidad tipo MagLev. En 1912, el ingeniero e inventor Emile Bachelet obtiene la patente para un “aparato transmisor de levitación” de la oficina de patentes de estados unidos. El invento fue descrito como una maquina capaz de transferir paquetes a muy alta velocidad desde un punto a otro. Alrededor de 20 años después, Hermann Kemper propone en Alemania un tren levitado magnéticamente a través de un lazo de control realimentado. Luego de décadas de desarrollo su idea eventualmente evoluciona en el sistema Transrapid usado en el primer tren tipo MagLev comercial construido en Shanghái, el cual fue inaugurado en 2003. Este tren alcanza una velocidad de hasta 430km/h. Un diseño similar fue utilizado en la construcción de un tren tipo MagLev urbano de baja velocidad en la ciudad de Nagoya (Japón), el cual funciona desde 2005. La ventaja de este tipo de trenes de baja velocidad es su conducción suave, silencioso, seguro, confiable y posee bajos costos de mantenimiento y operación debido al bajo desgaste de sus componentes.

La tecnología de Levitación Magnética también podría ayudar a lanzar naves espaciales poniéndolas en órbita. El sistema StarTram (Figura 1), el cual está siendo desarrollado por la NASA, es un sistema de asistencia de lanzamiento el cual conduciría un vehículo espacial a lo largo de una vía. La nave espacial levitada magnéticamente se aceleraría a velocidades de hasta 950km/h para luego cambiar a propulsores tipo cohete para su lanzamiento a la órbita. El peso del propelente es uno de los principales culpables del alto

costo de los lanzamientos de cohetes convencionales. Pero debido a que StarTram usa electricidad externa para ayudar al lanzamiento, el peso del vehículo en el despegue es aproximadamente un 20% menor que el de un cohete típico, lo cual disminuye los costos en el despegue [2].

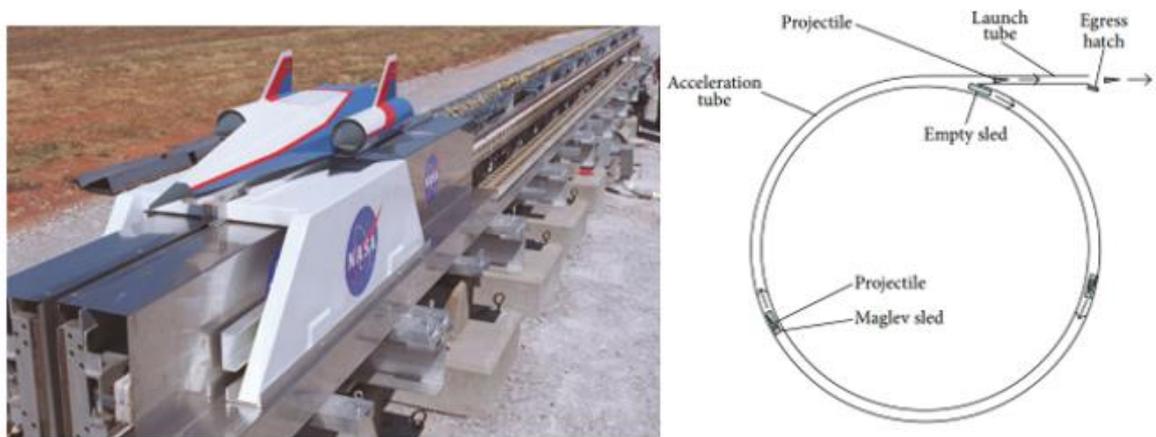


Figura 1: Esquema del sistema de asistencia de lanzamiento magnético StarTram

Otro de los usos de la levitación magnética es el de caracterizar muestras de alimentos y agua a partir de mediciones de densidad. Las medidas de densidad se obtienen suspendiendo un objeto diamagnético en un recipiente lleno del fluido paramagnético al cual se desea realizar la medición a partir de la posición vertical del objeto suspendido. La levitación magnética se utiliza para estimar la salinidad del agua, comparar diversos aceites vegetales en función de la proporción de grasa poliinsaturada y grasa monoinsaturada, comparar el contenidos de grasa en la leche, el queso, pasta de maní, y para determinar la densidad de los granos, etcétera [3].

En base a la misma tecnología, se está desarrollando una bomba centrífuga para implantes de corazón (Figura 2), el cual el impulsor es levitado magnéticamente. Esta bomba utilizada como un asistente ventricular para pacientes que sufrieron ataques cardíacos, consiste en un cojinete magnético

activo que levita el impulsor permitiendo la rotación sin fricción. Las principales ventajas de una transmisión sin fricción son la hemólisis reducida, la trombogénesis minimizada y la mayor durabilidad mecánica necesaria para el apoyo circulatorio mecánico a largo plazo. Varios fabricantes se encuentran en diversas etapas de desarrollo y aplicación clínica de bombas de sangre rotativas de tercera generación [4].

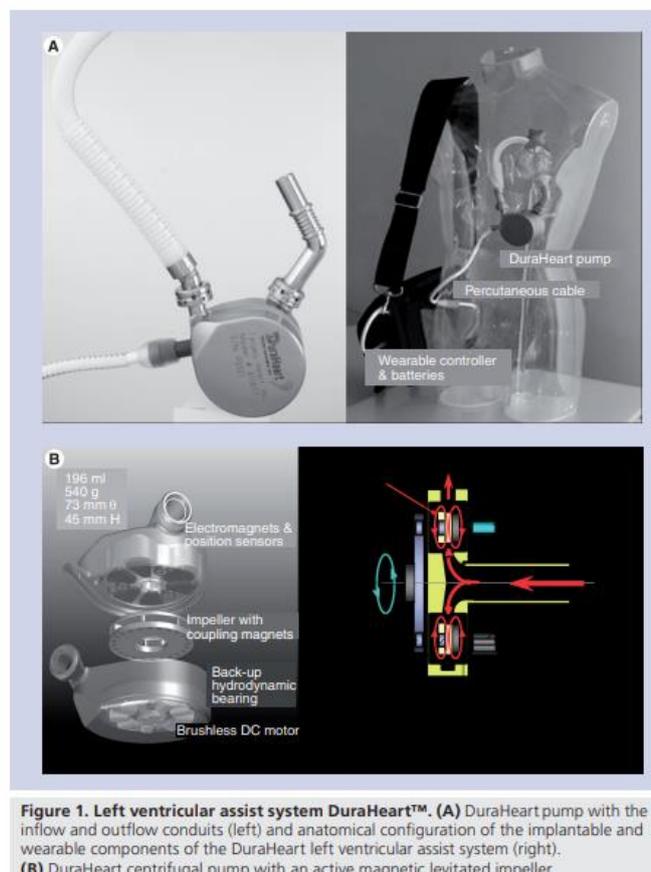


Figura 2: Esquema de sistema ventricular centrifugo levitado magneticamente DuraHeart

Como último ejemplo de utilización de esta tecnología, puede hacerse mención a los cojinetes magnéticos activos (AMB: Active Magnetic Bearing), los cuales levitan magnéticamente el eje en movimiento (Figura 3). Estos cojinetes permiten altas velocidades de operación, hasta 360000rpm (6kHz) para aplicaciones industriales [5]. Debido a que no hay contacto físico entre las partes fijas y móviles, no existe desgaste mecánico y no hay necesidad de

lubricación, evitando así posibles impurezas en el medio en el cual trabaja el cojinete. Dos ejemplos típicos son la industria de los equipos para semiconductores y las compañías de alimentación y bebidas. A su vez, una operación libre de lubricación significa que ya no se requieren los equipos relacionados con la lubricación, como las bombas y los filtros. Esto supone una reducción considerable de los costos totales.

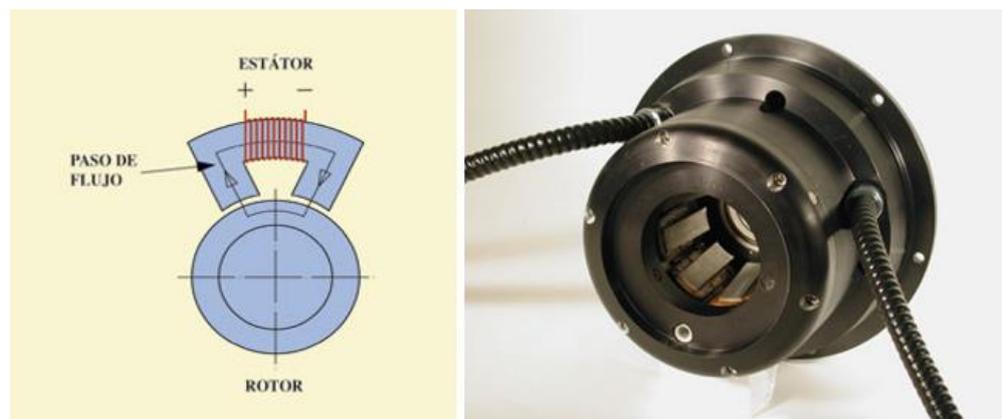


Figura 3: Esquema de un Cojinete magnético activo.

En un AMB es necesario implementar un sistema de control que regule la fuerza de atracción magnética. A partir de este sistema de control, se pueden compensar las fuerzas de desequilibrio y/o reducir las vibraciones de la máquina rotante. Estos sistemas AMB se utilizan a menudo para controlar las frecuencias de resonancia estructural provenientes del rotor o de los soportes elásticos. Ya que las frecuencias de resonancia pueden variar significativamente con la velocidad de rotación, el diseño del controlador para los sistemas AMB es importante para el rendimiento del sistema. Otras de las ventajas que permiten los AMB es que su sistema de control ofrece inherentemente la posibilidad de registrar continuamente las fuerzas de los rodamientos y los desplazamientos del rotor, permitiendo el monitoreo en línea de los parámetros críticos del proceso y la detección temprana de fallas incipientes, aumentando así la confiabilidad. Además, los AMB permiten

modificar el punto de operación, ajustando y optimizando de acuerdo a los cambios del proceso.

Debido a estas ventajas, se utilizan en un número cada vez mayor de aplicaciones comerciales de alto rendimiento en el ámbito de la maquinaria rotativa, ejemplo de esto son: bombas de ultra alto vacío, turbinas de vapor de alta temperatura de 10MW (540°C) [6], industria de semiconductores, industrias alimenticias husillos de fresado y rectificado de alta velocidad, volantes para almacenamiento de energía, giroscopios para navegación espacial, selectores de neutrones (neutron chopper), centrifugadoras Zippe, etc.

En este trabajo se desarrollará un sistema de levitación magnético activo. Se implementarán distintos controladores y se estudiará el comportamiento de estos ante distintas perturbaciones de posición en el elemento levitado del tipo vibratoria. La perturbación se realizará a partir del giro de un motor desbalanceado ubicado en el propio elemento levitado.

6 Descripción de la Planta

—¿Es aquella la Montaña? —preguntó Bilbo con voz solemne, mirándola con asombro. Nunca había visto antes algo que pareciese tan enorme. —¡Desde luego que no! —dijo Balin—. Esto es sólo el principio de las Montañas Nubladas, tenemos que cruzarlas de algún modo, por encima o por debajo, antes de que podamos internarnos en las Tierras Áspers de más allá.

El Hobbit

6.1 Esquema experimental

Un levitador magnético es un dispositivo capaz de mantener una masa ferromagnética en suspensión en un rango de alturas determinado a través de un campo magnético generado por un electroimán. En la Figura 4 se muestra el esquema conceptual del mismo.

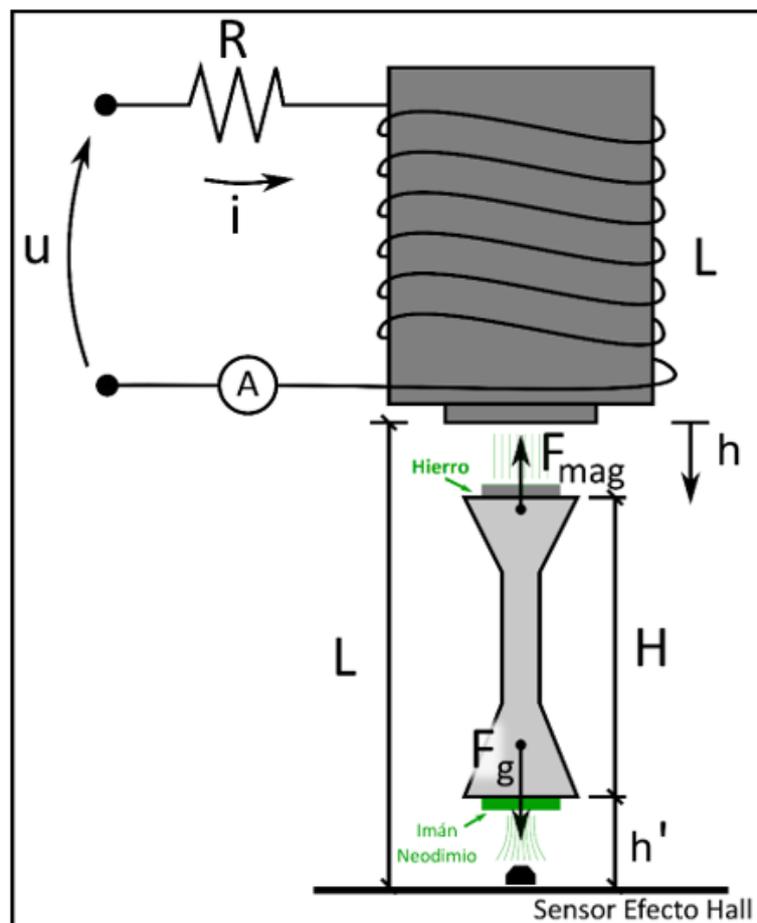


Figura 4: Esquema conceptual del levitador magnético diseñado

Al aplicar una tensión (u) al electroimán, se produce una corriente (i), generando a su vez una fuerza magnética (F_{mag}). Esta fuerza magnética se opone al peso del cuerpo libre, atrayéndolo hacia el electroimán. Para generar la levitación, la fuerza magnética deberá ser igual (y de signo contrario) al peso del cuerpo libre, en un punto de operación deseado (h). Debido a que la fuerza magnética es función de la corriente que circula por el electroimán y del hueco que hay en la parte superior del elemento levitado, estos parámetros deberán ser medidos y controlados para lograr la levitación magnética.

Para medir la corriente del electroimán se agregó en serie al electroimán una resistencia de bajo valor. La caída de tensión que se produce en la resistencia será proporcional a la corriente que circula por el electroimán.

Para realizar la medición de posición del elemento levitado, se agregó un imán en la parte inferior de este y un sensor de efecto hall en la base del levitador. El sensor de efecto hall entrega una tensión entre bornes proporcional a la intensidad de campo medida. A partir de este fenómeno se puede medir la distancia entre el sensor y el imán y por ende la posición del elemento levitado. Cabe aclarar que, debido a la gran distancia entre el electroimán y el sensor de efecto hall, el campo magnético generado por el primero no afecta la medición de posición.

Para implementar el sistema de control se utilizó una computadora de placa única (SBC por sus siglas en inglés Single Board Computer), la cual es capaz de medir entradas analógicas y posee una salida tipo PWM para regular la tensión que se aplica al electroimán.

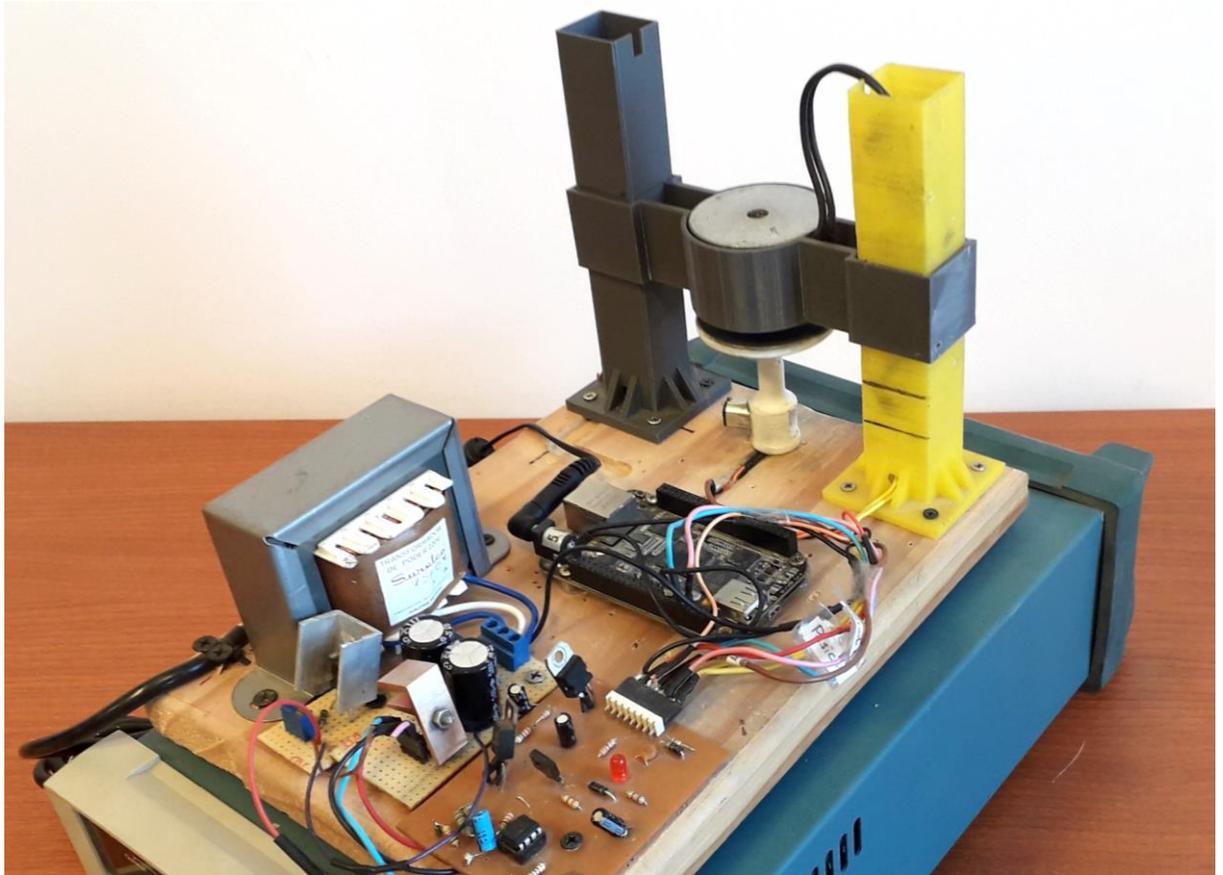


Figura 5: Levitador Magnético desarrollado para probar los distintos controladores.

En la Figura 5 se puede observar la fuente de alimentación; la placa electrónica que adapta y filtra las distintas señales involucradas; el electroimán tipo ventosa que oficia como actuador; el sensor de efecto Hall 49e para medir la posición del elemento levitado y la SBC que permite medir las variables involucradas, implementar los algoritmos de control y generar una tensión modulada por ancho de pulsos (PWM).

6.2 Descripción del hardware utilizado

6.2.1 Placa Electrónica

En la Figura 6 se muestra el circuito electrónico diseñado para acondicionar las señales necesarias para controlar la planta. La misma se compone de una entrada PWM, proveniente de la SBC, la cual es filtrada para atenuar el ripple de tensión sobre el electroimán, logrando acciones de controla más suaves. El transistor se utiliza como amplificador lineal, siendo el encargado de manejar la tensión aplicada al electroimán. En el emisor del transistor se encuentra una resistencia de 0.8Ω , la cual produce una pequeña caída de tensión proporcional a la corriente que circula por el electroimán. Esta caída de tensión es filtrada y luego amplificada por el circuito operacional 741. El filtro es del tipo *pasabajo* con una frecuencia de corte de 146Hz ($\sim 917\text{rad/s}$) el cual actúa como un filtro antialiasing. La salida de este operacional se conecta a la entrada analógica P9_39 de la SBC.

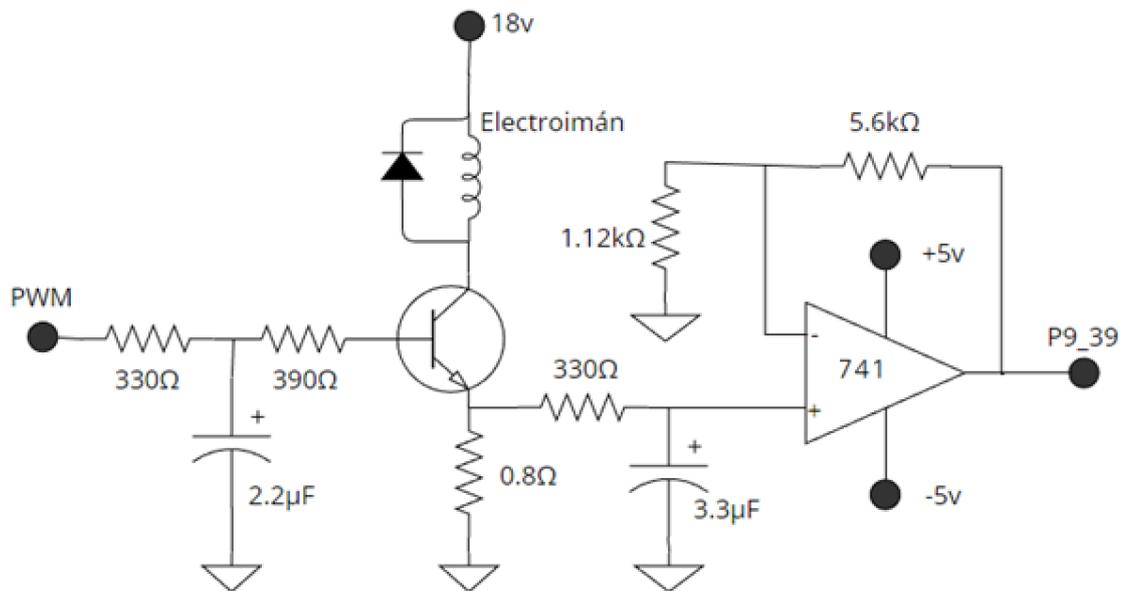


Figura 6: Esquema del circuito electrónico utilizado para la medición de corriente y la actuación del electroimán

En la Figura 7 se muestra el esquema electrónico correspondiente a la medición de posición. La misma consiste en el sensor de posición lineal de Efecto Hall 49e al cual se le agregó a la salida un divisor resistivo. Este divisor

resistivo cumple la función de no superar la tensión máxima permitida por las entradas analógicas de la SBC. También se agregó un filtro antialiasing con una frecuencia de corte de 159Hz ($\sim 998\text{rad/s}$).

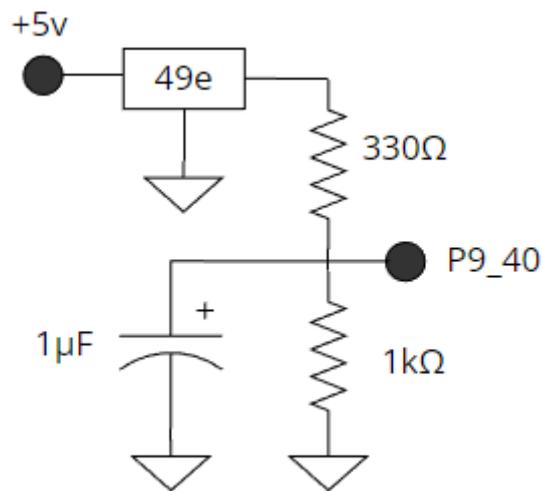


Figura 7: Esquema del circuito electrónico utilizado para acondicionar la señal de posición.

6.2.2 Electroimán tipo Ventosa

El electroimán utilizado es del tipo ventosa (Figura 8) modelo T8020 de Técnica Gama, el cual posee una tensión de trabajo de 12v@300mA y una fuerza máxima de trabajo, según el fabricante, de 50kg¹.

¹ Fuerza de trabajo con entrehierro de 0mm en chapa de 3mm espesor.



Figura 8: Electroimanes tipo ventosa como el utilizado en el sistema.

6.2.3 Elemento Levitado

El elemento levitado consiste en un cilindro impreso en 3D, el cual tiene una masa de 50gr. En su cara superior posee una placa tipo arandela de material ferromagnético de 1mm de espesor. Esta placa, de igual diámetro externo que el electroimán, posee una abertura por donde pasa un eje paramagnético de aluminio de manera de limitar el movimiento del elemento levitado en la dirección longitudinal. En la cara inferior se encuentra un imán de neodimio de 12mm de diámetro que, junto al sensor de efecto hall, es utilizado para determinar la posición del elemento levitado. Por sobre la cara inferior, el elemento levitado posee una abertura por donde se introduce un motor vibrador el cual será responsable de inducir las perturbaciones a ser rechazadas por los distintos controladores.



Figura 9: Elemento levitado del sistema Levitador Magnético

6.2.4 Sensor de efecto Hall

Para medir la posición del elemento levitado se utilizó un sensor lineal de efecto Hall, el cual mide la intensidad de campo del imán permanente ubicado en el fondo de este. Este sensor 49E es un pequeño circuito integrado el cual entrega una tensión proporcional a la intensidad campo magnético medido (Figura 10).



Figura 10: Sensor 49e de efecto hall utilizado para medir la posición del elemento levitado.

Las principales características del sensor son:

- Consumo de energía de 3.5mA@VCC = 5V
- Voltaje operativo: 3V a 6.5V dc
- Sensibilidad: 17mV/mT @Vcc=5v
- Rango medición -90 a +90mT
- Rango de temperatura de -40°C a 85°C
- Ancho de banda: no especificado por el fabricante.

6.2.5 Motor Vibrador

Para lograr introducir la perturbación de vibración en el elemento levitado se introdujo, dentro de este, un motor vibrador (Figura 11) el cual funciona a partir de tener una masa desbalanceada en su eje. Este vibrador posee una velocidad de giro de 250rpm@3v y tiene una masa de 15gr.



Figura 11: Motor vibrador utilizado para introducir perturbaciones al sistema.

6.2.6 Computadora de Placa Única

Para la implementación de los algoritmos de control es necesaria la utilización de un sistema capaz de realizar la medición de las variables de interés y en

función de estas aplicar una acción de control a la planta. Debido a esto se utilizó una Computadora de Placa Única o SBC. La SBC es una computadora completa, desarrollada en una sola placa electrónica que incluye microprocesadores, memoria, entradas/salidas (I/O) entre otras características.

Para este proyecto se utilizó una BeagleBone® Black, la cual es una SBC de bajo costo (65U\$\$) que incluye un microprocesador Sitara™ AM3359 ARM Cortex™-A8 de Texas Instrument que funciona a 1GHz, 4GB de memoria flash integrada además de 512MB de memoria RAM DDR3 a 400MHz. También 2x46 pines de entrada/salida los cuales, 6 son entradas ADC de 12bit y posee salidas PWM de hasta 400MHz. A su vez es compatible con Debian, Android, Ubuntu, entre otros. Debido a estas prestaciones es muy atractiva para el desarrollo de este proyecto. En la Figura 12 se muestra la placa de desarrollo BeagleBone® Black con sus puertos de entrada/salida.

P9					P8			
Function	Physical Pins	Function	Function		Physical Pins	Function		
DGND	1	2	DGND	DGND	1	2	DGND	
VDD 3.3 V	3	4	VDD 3.3 V	MMC1_DAT6	3	4	MMC1_DAT7	
VDD 5V	5	6	VDD 5V	MMC1_DAT2	5	6	MMC1_DAT3	
SYS 5V	7	8	SYS 5V	GPIO_66	7	8	GPIO_67	
PWR_BTN	9	10	SYS_RESET	GPIO_69	9	10	GPIO_68	
UART4_RXD	11	12	GPIO_60	GPIO_45	11	12	GPIO_44	
UART4_TXD	13	14	EHRPWM1A	EHRPWM2B	13	14	GPIO_26	
GPIO_48	15	16	EHRPWM1B	GPIO_47	15	16	GPIO_46	
SPIO_CS0	17	18	SPIO_D1	GPIO_27	17	18	GPIO_65	
I2C2_SCL	19	20	I2C_SDA	EHRPWM2A	19	20	MMC1_CMD	
SPIO_DO	21	22	SPIO_SCLK	MMC1_CLK	21	22	MMC1_DAT5	
GPIO_49	23	24	UART1_TXD	MMC1_DAT4	23	24	MMC1_DAT1	
GPIO_117	25	26	UART1_RXD	MMC1_DAT0	25	26	GPIO_61	
GPIO_115	27	28	SP11_CS0	LCD_VSYNC	27	28	LCD_PCLK	
SP11_DO	29	30	GPIO_112	LCD_HSYNC	29	30	LCD_AC_BIAS	
SP11_SCLK	31	32	VDD_ADC	LCD_DATA14	31	32	LCD_DATA15	
AIN4	33	34	GND_ADC	LCD_DATA13	33	34	LCD_DATA11	
AIN6	35	36	AIN5	LCD_DATA12	35	36	LCD_DATA10	
AIN2	37	38	AIN3	LCD_DATA8	37	38	LCD_DATA9	
AIN0	39	40	AIN1	LCD_DATA6	39	40	LCD_DATA7	
GPIO_20	41	42	ECAPWMO	LCD_DATA4	41	42	LCD_DATA5	
DGND	43	44	DGND	LCD_DATA2	43	44	LCD_DATA3	
DGND	45	46	DGND	LCD_DATA0	45	46	LCD_DATA1	

LEGEND	
Power, Ground, Reset	
Digital Pins	
PWM Output	
1.8 Volt Analog Inputs	
Shared I2C Bus	
Reconfigurable Digital	

Figura 12 PinOut de la placa de desarrollo BeagleBone® Black

7 Identificación y Modelado Matemático del Sistema

"Controlling a laser with Linux is crazy, but everyone in this room is crazy in his own way. So if you want to use Linux to control an industrial welding laser, I have no problem with your using PREEMPT_RT."

Linus Torvalds

7.1 Caracterización del Tiempo de Muestreo

Para este proyecto en particular, los tiempos de muestreo (T_s) que la placa debe cumplir son determinantes para el correcto control del sistema. Es por eso que se procedió a caracterizar la placa BeagleBone® Black a partir de un programa en código C. En la Figura 6 se muestra el diagrama de flujo del código implementado.

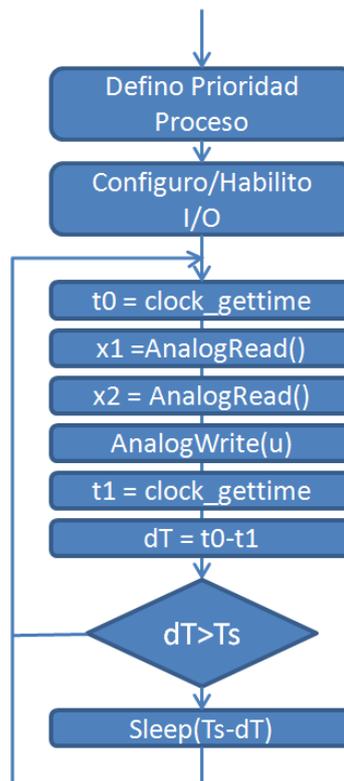


Figura 6: Diagrama de flujo del algoritmo para garantizar tiempo de muestreo constante en la placa de control.

En una primera instancia este programa define la prioridad del proceso como alta, de manera que el sistema operativo priorice la ejecución del mismo, obteniendo así tiempos de muestreo (T_s) lo más constantes posibles. Luego se habilita las entradas/salidas del sistema, para luego pasar al loop de control. En este loop se mide el tiempo t_0 en el cual inicia, se procede a realizar la medición analógica de las variables x_1 y x_2 (mediciones que permitirán a futuro cerrar los lazos de control de corriente y posición del levitador. Luego se genera una señal de salida PWM. Por último, se calcula el tiempo dT (siendo $dT=t_1-t_0$) que tardó el loop en ejecutarse y se procede a la suspensión del proceso un tiempo T_s-dT de manera de garantizar que el tiempo de ejecución del loop sea de un periodo de muestreo.

Este script descrito anteriormente se ejecutó con el sistema operativo preinstalado Debian GNU/Linux 10 kernel 4.19.94-ti-r42 y un kernel 4.19.190-bone-rt-r65. La diferencia entre kernels radica en que este último es un patch PREEMPT_RT que convierte al sistema operativo Linux en un sistema de Tiempo Real. La instalación se realizó según [7].

En la Figura 7 se puede observar la respuesta de los kernels mencionados anteriormente ante 10000 iteraciones para diferentes tiempos de muestreo.

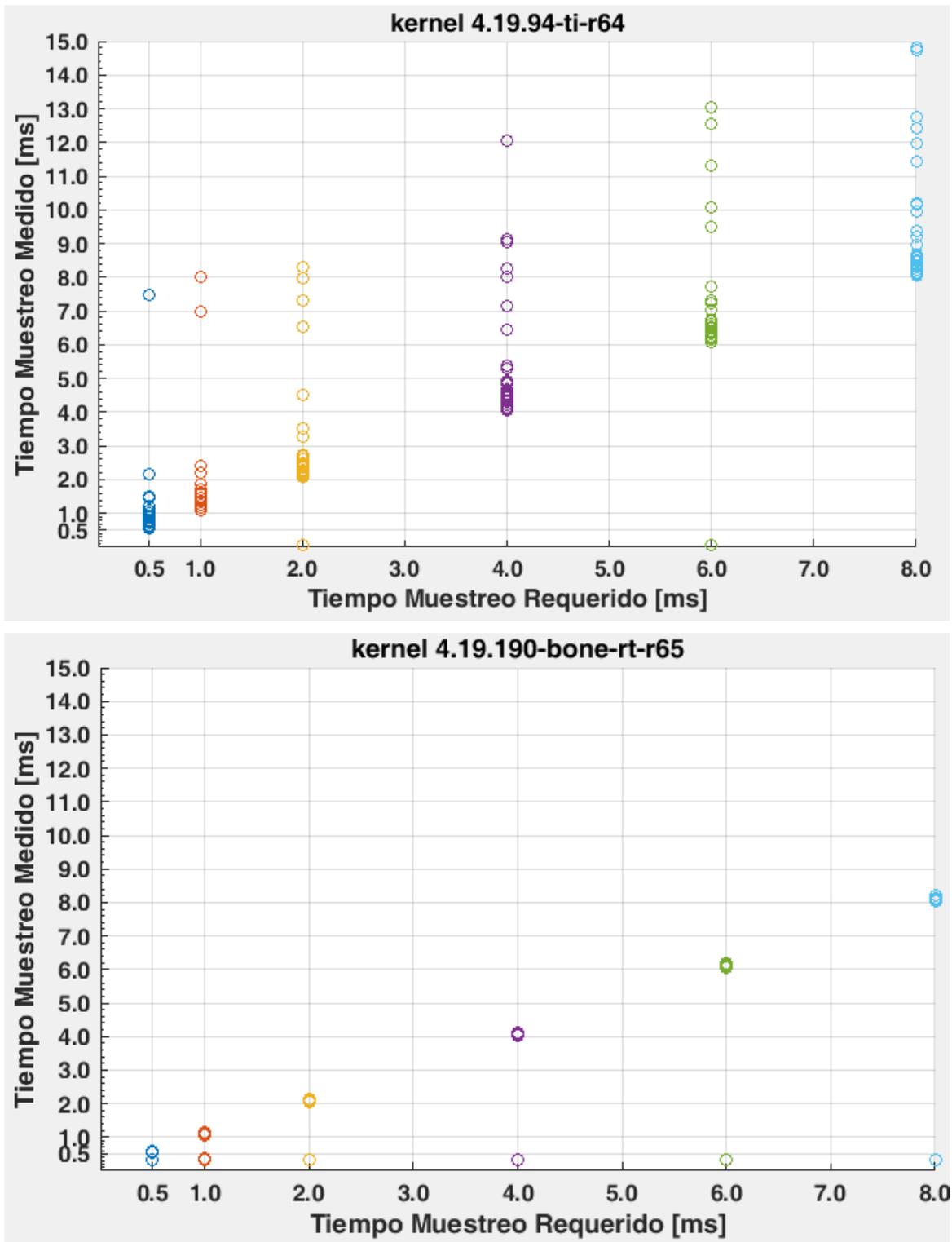


Figura 7: Tiempo de muestreo medido en función del T_s requerido para 10000 muestras evaluado en SO no-RT y RT.

En la figura correspondiente al kernel 4.19.94-ti-r64 se puede observar que, ante un T_s determinado, la gran dispersión de valores que puede llegar a demorar la iteración es demasiado grande. Estos valores son inaceptables para sistemas de control en tiempo real debido a que pueden llegar a ser de 1.8 a 7 veces más grande que el T_s requerido. Descartando así el uso de este kernel para este proyecto en particular. Para el caso del kernel 4.19.190-bone-rt-r65 el sistema operativo responde de forma precisa, garantizando un tiempo de muestreo uniforme a lo largo de toda la medición, salvo en la primera iteración, que arroja valores del orden de los 0.5ms. En la Tabla 1 se muestra esta comparación, ante determinados T_s , de los dos kernels utilizados.

Kernel						
Tiempo Muestreo [ms]	4.19.94-ti-r64			4.19.190-bone-rt-r65		
	mean(T_s) [ms]	std(T_s) [ms]	Error relativo [%]	mean(T_s) [ms]	std(T_s) [ms]	Error relativo [%]
0.50	0.59	0.08	12.80	0.524	0.007	1.34
1.00	1.1	0.1	8.77	1.000	0.015	1.50
2.00	2.1	0.1	5.64	2.03	0.02	0.93
4.00	4.1	0.1	3.18	4.03	0.08	2.07
6.00	6.1	0.1	2.29	6.0	0.1	1.65
8.00	8.1	0.1	1.64	8.03	0.08	0.97

Tabla 1: Tabla comparativa entre promedios de T_s utilizados y su respectivo desvío estándar

7.2 Medición de los Parámetros de la Planta

El diagrama en bloques del sistema puede ser representado como el mostrado en la Figura 13:

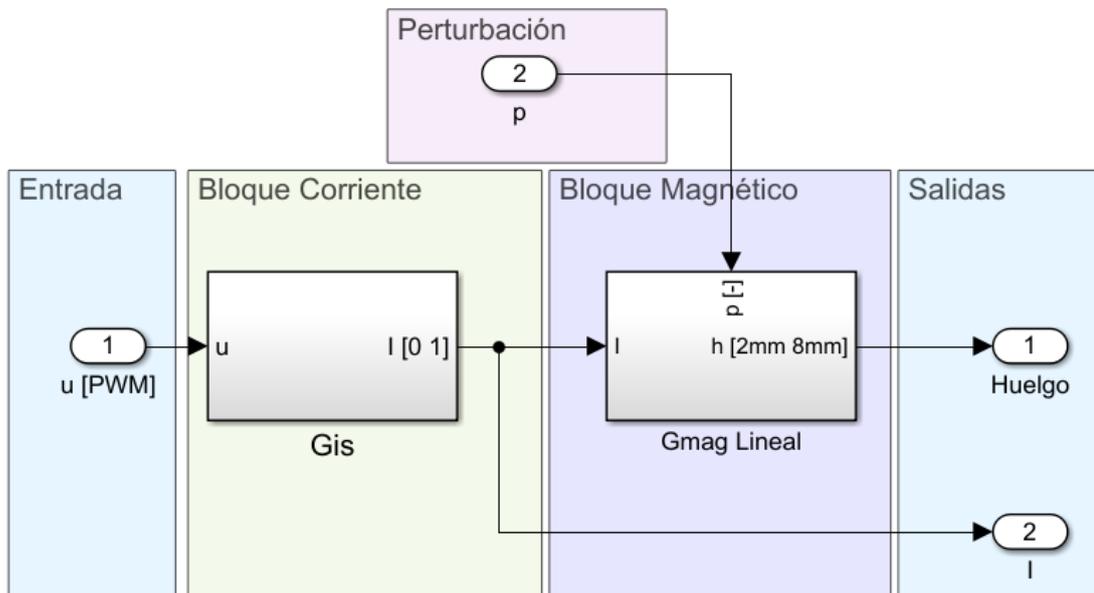


Figura 13: Diagrama de bloques del sistema Levitador Magnético a lazo abierto.

En el mismo se observa la entrada de tensión (u) como una señal del tipo PWM, el bloque de corriente y el bloque magnético. El sistema posee dos mediciones, la corriente que circula por el electroimán (I) y la medición del hueco entre el electroimán y el elemento levitado (*Huelgo*). A su vez, sobre el bloque magnético ingresa la perturbación de posición del elemento levitado.

Para poder diseñar los lazos de control correctamente es necesario obtener un modelo matemático del sistema lo más confiable posible. Es por eso que se procedió a identificar y validar los distintos bloques que componen el sistema de levitación magnética. A continuación, se presenta la caracterización y modelo de este sistema.

7.2.1 Identificación del Bloque de corriente

Como primer paso se midió la relación tensión/corriente del bloque, obteniéndose la relación mostrada en la Figura 14 .

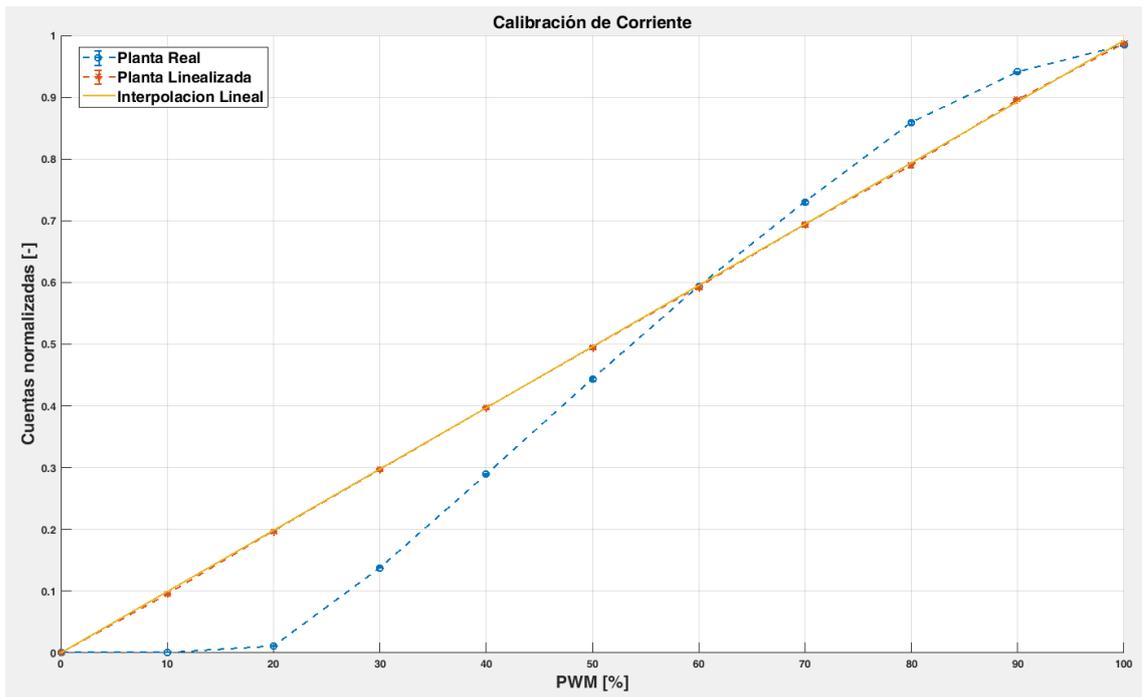


Figura 14: Relación entre la tensión de entrada y la corriente que circula por el electroimán.

Se observa que la relación es de carácter no lineal. Esto se corrigió por software de manera de linealizar la relación entre la tensión de entrada y la salida. Se aprecia que la ganancia de estado estacionario es 0.99 cuentas normalizadas.

Para determinar la dinámica del Bloque Corriente, se excitó al mismo con una señal del tipo escalón de tensión entre 0% y 100% de ciclo útil en pasos de 10% (Figura 15) con un huelgo fijo de 4mm del elemento levitado.

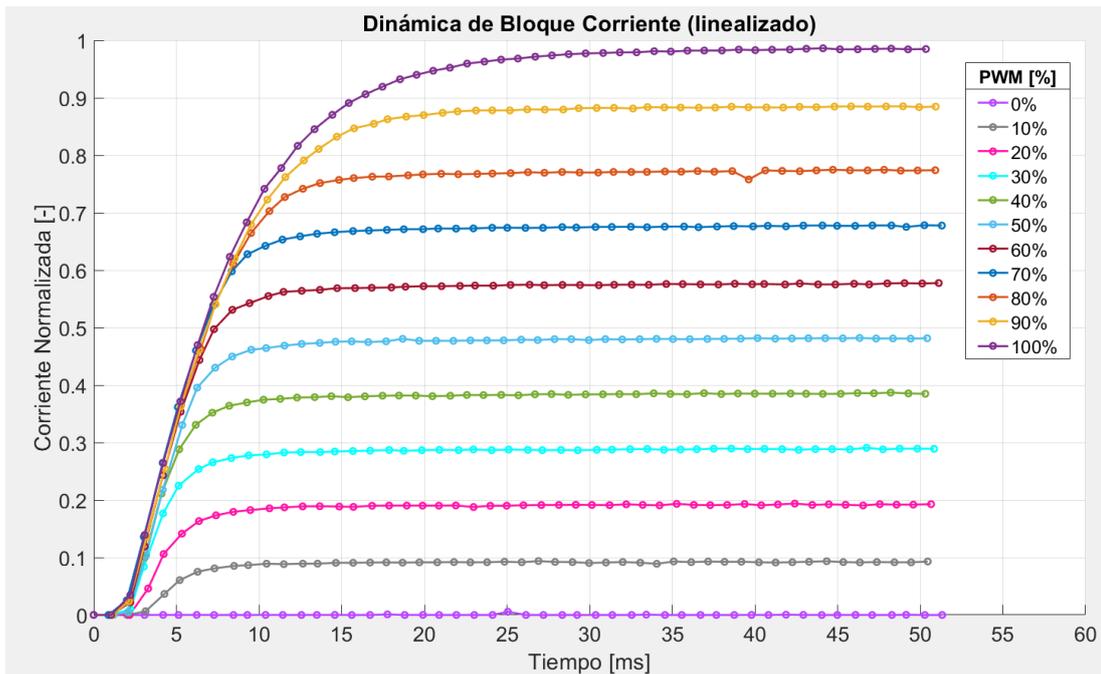


Figura 15: Respuesta dinámica del bloque corriente ante entradas del tipo escalón de tensión.

En la figura anterior se observa que, como era de esperar, el bloque corriente tiene un comportamiento de sistema de primer orden. A partir de esta respuesta, se modela al bloque como:

$$G_i(s) = \frac{I(s)}{U(s)} = \frac{K_i}{s/\tau + 1}$$

En la Tabla 2 se muestran los distintos valores obtenidos a través de interpolar los datos obtenidos.

$$I(t) = u(t) * K_i(1 - e^{-t/\tau})$$

PWM [%]	$u(t) * K_i$ [-]	Cota error	τ [seg]	Cota error	R [-]
10	0.09236	(0.091, 0.0936)	315.5	(282.2, 348.7)	0.962
20	0.1916	(0.190, 0.192)	395.6	(373.2, 418)	0.988
30	0.2877	(0.287, 0.288)	489.5	(474.9, 504.1)	0.997
40	0.3855	(0.384, 0.386)	420.4	(405.7, 435.1)	0.992
50	0.482	(0.480, 0.484)	350.1	(334, 366.2)	0.986
60	0.5779	(0.576, 0.580)	335	(324.5, 345.4)	0.992
70	0.6793	(0.677, 0.68)	281.7	(272.3, 291.1)	0.990

80	0.7762	(0.774, 0.780)	236.2	(228.8, 243.5)	0.991
90	0.8881	(0.886, 0.890)	189.9	(185.6, 194.3)	0.995
100	0.9866	(0.985, 0.98)0	168.2	(166.4, 170)	0.998

Tabla 2: Valores de ganancia y tiempo característico obtenidos con escalones de corriente de distinta amplitud.

A partir de la Tabla 2, se observa dispersión de valores en el tiempo característico de respuesta del sistema. Debido a que la planta de levitación magnética trabajará alrededor del punto de equilibrio $I=0.5$, se adoptará este valor para determinar el tiempo característico del bloque de Corriente.

Como se mencionó anteriormente, se aproxima la dinámica del bloque Corriente como un sistema lineal del tipo:

$$G_i(s) = \frac{0.99}{\frac{s}{350} + 1}$$

A partir del modelo matemático propuesto, se validó la dinámica del sistema con el bloque lineal propuesto a partir de una entrada cuadrada de amplitud y periodo pseudorandom, obteniéndose lo mostrado en la Figura 16.

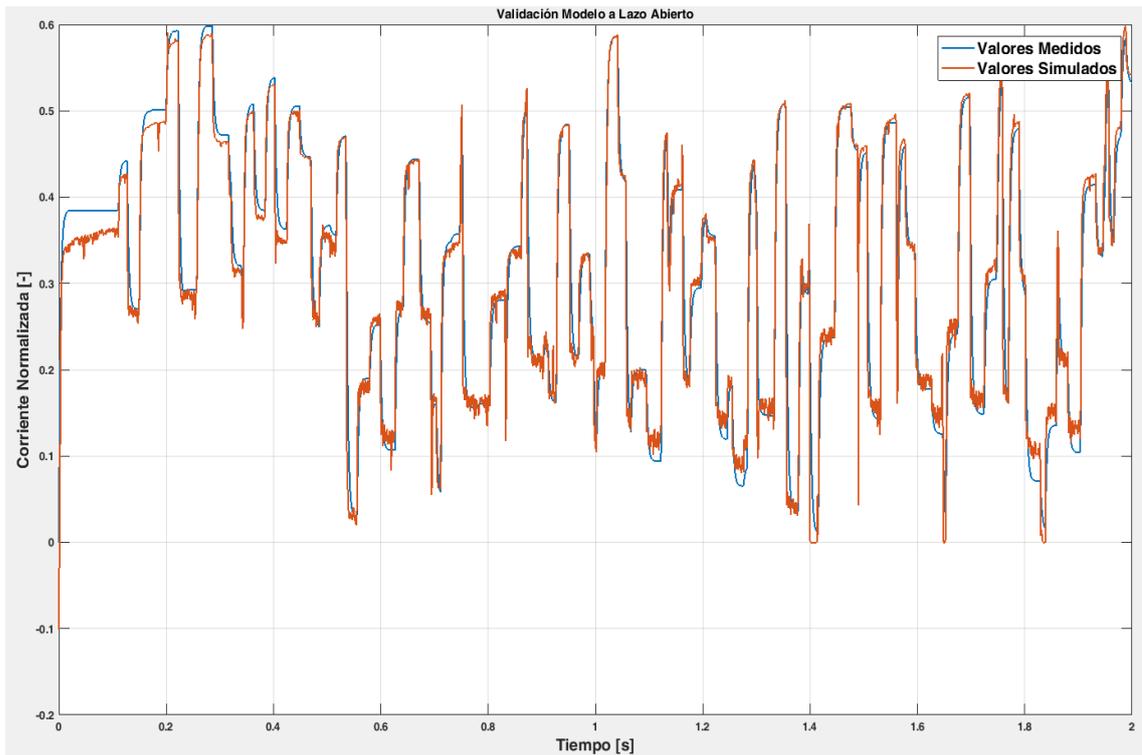


Figura 16: Comparación entre la respuesta del Bloque Corriente del sistema modelado (rojo) y medido (azul)

Se verifica que la dinámica del modelo matemático corresponde con bastante exactitud a la dinámica medida.

7.2.2 Bloque Fuerza Magnética

Como primer paso se calibró el sensor de posición del elemento levitado con respecto a la superficie del electroimán (Figura 17). Se observa que, si bien el sensor de posición posee cierto grado de alinealidad, la medición resulta ser precisa debido a la poca dispersión de valores ante un determinado huelgo.

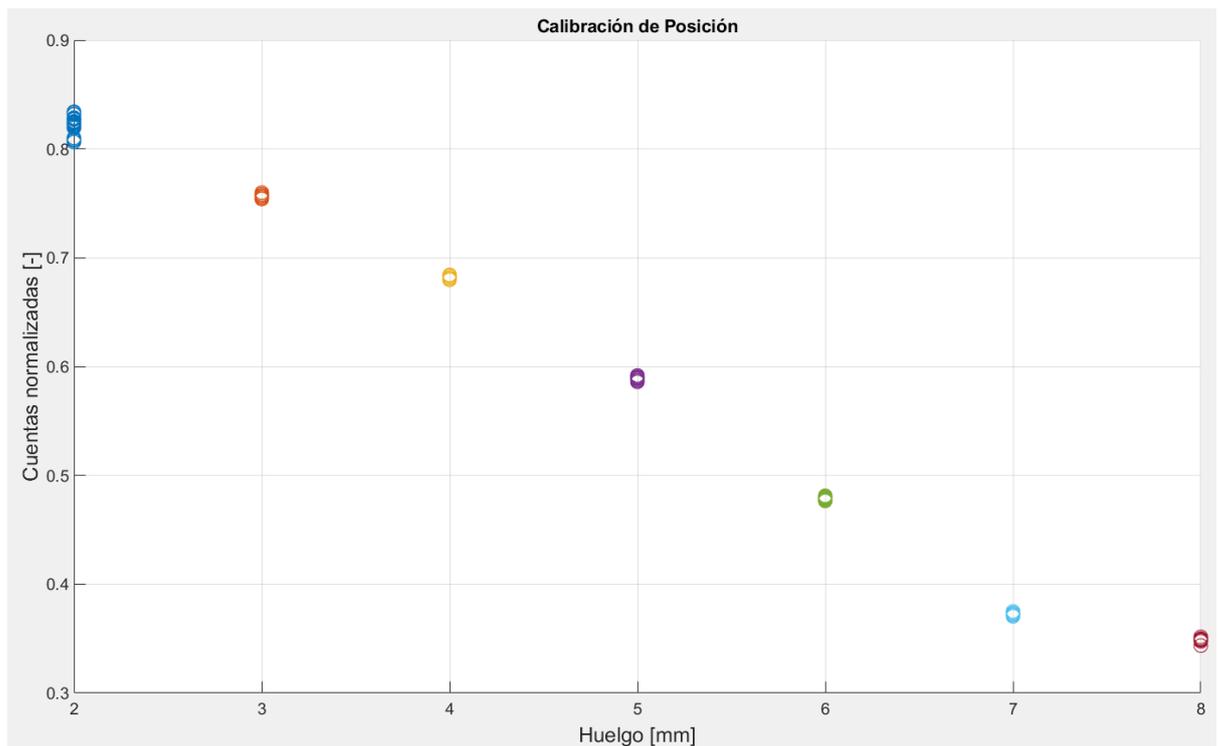


Figura 17: Calibración del sensor de posición de efecto Hall E49.

Para medir la relación entre la fuerza magnética en función del huelgo y la corriente, se utilizó una balanza. A partir de definir la distancia entre el elemento levitado y la superficie del electroimán, se midió la diferencia de peso del elemento levitado cuando se aplicaba una corriente de valor establecido. Para mantener esta corriente constante a lo largo de la medición, se utilizó un control Proporcional-Integral en el bloque de corriente. A partir de estas mediciones se pudo hallar el campo escalar mostrado en la Figura 18.

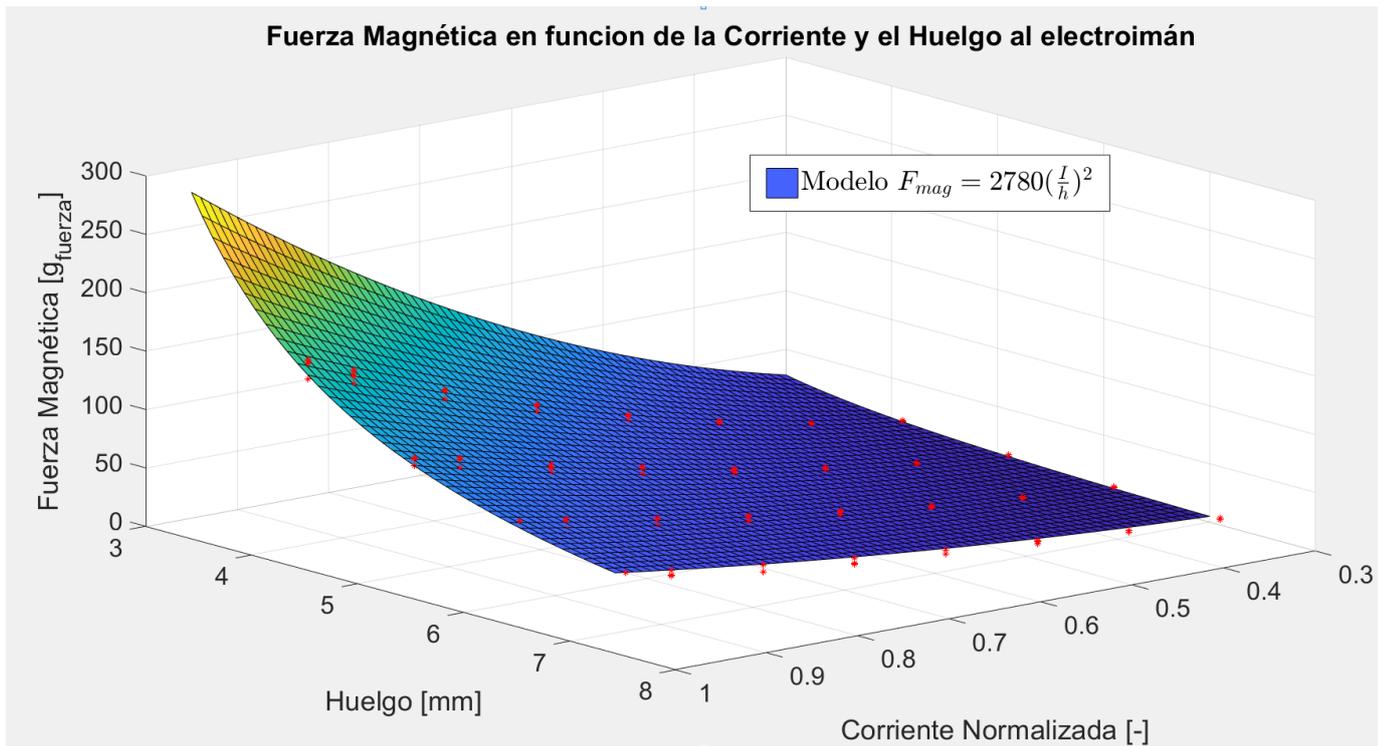


Figura 18: Relación entre la Fuerza magnética aplicada al elemento levantado en función del huelgo y la corriente aplicada.

En esta figura se observa que la fuerza magnética aplicada al elemento levantado es proporcional al cuadrado de la corriente e inversamente proporcional al cuadrado del huelgo, pudiéndose expresar como:

$$F_{mag} = g K_{mag} \left(\frac{I}{h}\right)^2 \quad (1)$$

7.2.3 Modelo Matemático

A partir de la identificación de los distintos parámetros que componen el bloque de corriente y el bloque magnético, se procedió a realizar el modelo matemático de la planta. Para esto se recuerda el esquema del sistema mostrado anteriormente:

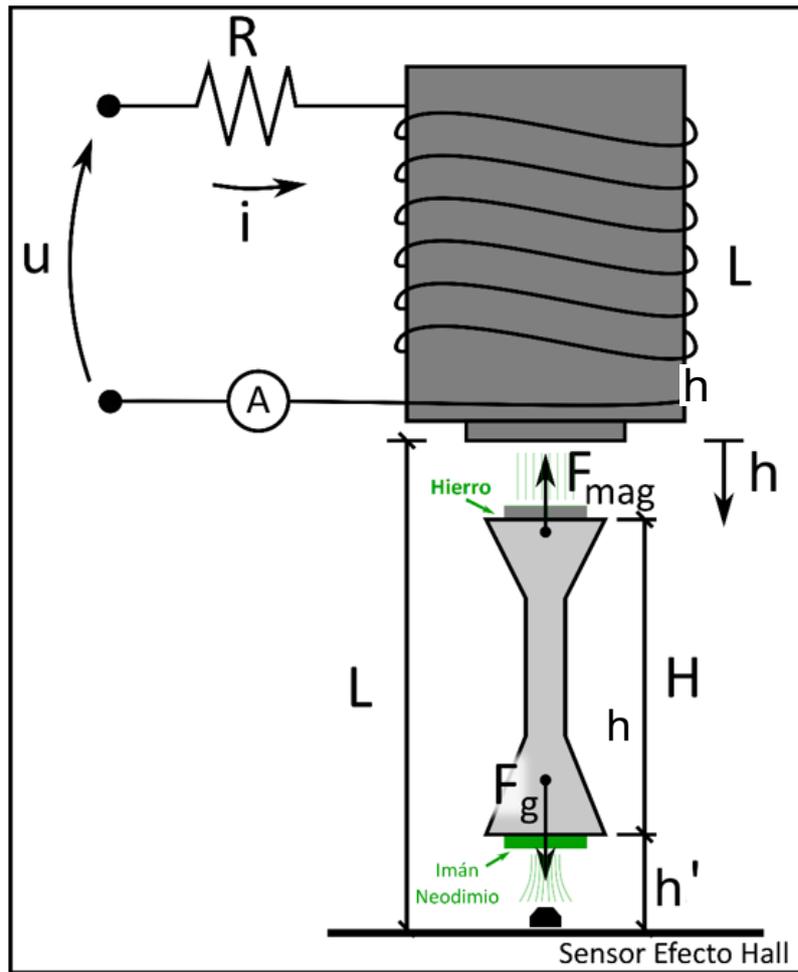


Figura 19: Esquema conceptual del levitador magnético diseñado

Siendo las hipótesis de modelado:

- Inductancia del electroimán constante. Ante cambios en el huelgo, la reluctancia del circuito magnético se modifica, variando así la inductancia (L) del circuito electromagnético. En el modelo matemático este parámetro se considera constante.
- Movimiento del elemento levitado solo en el eje “h”, es decir el movimiento del elemento levitado se considera axial, despreciando movimientos radiales.

- Se desprecia el rozamiento producido por la guía vertical de aluminio que permite mantener centrado al elemento levitado.

A partir de estas hipótesis, las ecuaciones que describen a la planta son:

$$u = RI + L\dot{I} \quad (2)$$

$$m\ddot{h} = F_{mag} - P \quad (3)$$

Siendo:

$$P = mg \quad (4)$$

$$F_{mag} = g K_{mag} \left(\frac{I}{h}\right)^2 \quad (5)$$

Por lo que, el sistema de ecuaciones diferenciales del sistema es:

$$u = RI + L\dot{I} \quad (6)$$

$$m\ddot{h} = -g K_{mag} \left(\frac{I}{h}\right)^2 + mg \quad (7)$$

El punto de equilibrio del sistema se calcula igualando la ec.(7) a cero:

$$0 = -g K_{mag} \left(\frac{I}{h}\right)^2 + mg \therefore \boxed{\sqrt{\frac{K_{mag} I_e^2}{m}} = h_e} \quad (8)$$

Para facilitar el control del sistema, se opta por considerar la corriente de equilibrio $I_e = 0.5$, por lo que la posición de equilibrio es:

$$\sqrt{\frac{2780 \frac{\text{gr mm}^2}{-2} 0.5^2}{49.15 \text{gr}}} = h_e \therefore h_e = 3.7604 \text{ mm} \quad (9)$$

Linealizando la ec. (7) alrededor del punto de equilibrio se obtiene:

$$\ddot{\Delta h} = -2g \frac{K_{mag}}{m} \frac{I_e}{h_e^2} \Delta I + 2g \frac{K_{mag}}{m} \frac{I_e^2}{h_e^3} \Delta h \quad (10)$$

Transformando la ec. (10) por Laplace y expresándola como función transferencia:

$$G_{mag} = \frac{\Delta H(s)}{\Delta I(s)} = \frac{-2g \frac{K_{mag}}{m} \frac{I_e}{h_e^2}}{\left(s^2 - 2g \frac{K_{mag}}{m} \frac{I_e^2}{h_e^3} \right)} \quad (11)$$

Reemplazando por los valores medidos anteriormente:

$$G_{mag} = \frac{\Delta H(s)}{\Delta I(s)} = \frac{-2 \cdot 9810 \frac{mm}{seg^2} \frac{2780 \frac{gr \cdot mm^2}{-2}}{49.1gr} \cdot 0.5}{\left(s^2 - 2 \cdot 9810 \frac{mm}{seg^2} \frac{2780 \frac{gr \cdot mm^2}{-2}}{49.1gr} \frac{(0.5)^2}{(3.76mm)^3} \right)} \quad (12)$$

Por lo que la función transferencia del Bloque Magnético se puede expresar como:

$$G_{mag} = \frac{\Delta H(s)}{\Delta I(s)} = \frac{-39288}{(s^2 + 4798.3)} = \frac{-39288}{(s - 70.63)(s + 70.63)} \quad (13)$$

Recordando que la función transferencia del Bloque Corriente es:

$$G_i(s) = \frac{0.99}{\frac{s}{350} + 1} \quad (14)$$

A partir del modelo matemático de la planta a Lazo Abierto, se puede analizar el comportamiento de la misma, la cual posee los siguientes polos:

$$s_1 = -350.00 \text{ seg}^{-1}$$

$$s_2 = 69.27 \text{ seg}^{-1}$$

$$s_3 = -69.27 \text{ seg}^{-1}$$

Se observa una dinámica del bloque de corriente rápida ($s_1 \sim 55\text{Hz}$) mientras que el bloque magnético posee dinámica inestable ($s_2 \sim 11\text{Hz}$). Debido a esta dinámica inestable no es posible validar el modelo frente a la planta real a lazo abierto.

La representación del sistema por variable de estado es:

$$\begin{cases} \dot{x}(t) = Ax(t) + Bu(t) \\ y(t) = Cx(t) \end{cases} \quad (15)$$

$$\begin{bmatrix} \dot{i} \\ \dot{h} \\ \ddot{h} \end{bmatrix} = \begin{bmatrix} \frac{-R}{L} & 0 & 0 \\ 0 & 0 & 1 \\ \frac{K_{mag} I_e}{m h_e^2} & -\frac{K_{mag} I_e^2}{m h_e^3} & 0 \end{bmatrix} \begin{bmatrix} i \\ h \\ \dot{h} \end{bmatrix} + \begin{bmatrix} 1/L \\ 0 \\ 0 \end{bmatrix} u \quad (16)$$

$$y = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} i \\ h \\ \dot{h} \end{bmatrix}$$

Para el diseño de los distintos controladores se utilizará el modelo lineal de la planta (ec. (13) y (14) o bien ec. (16), según corresponda), mientras que la verificación del funcionamiento de los algoritmos de control se realizará a partir del modelo no lineal del sistema (Figura 20), el cual incluye ruido de medición, topes superiores e inferiores del huelgo y entrada de perturbación. En la Figura 21 y Figura 22 se muestra el modelo de los bloques que forman a la planta en su conjunto.

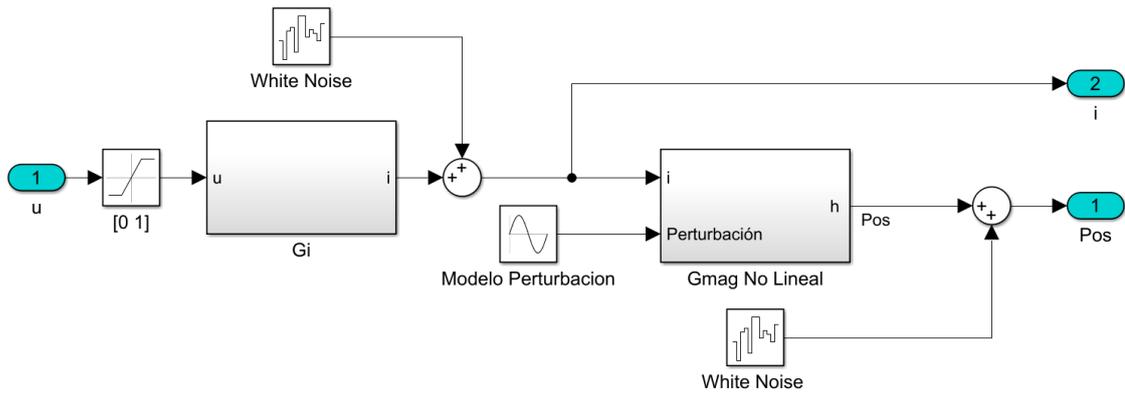


Figura 20: Modelo de la Planta de Levitación Magnética utilizada para la verificación de los controladores diseñados.

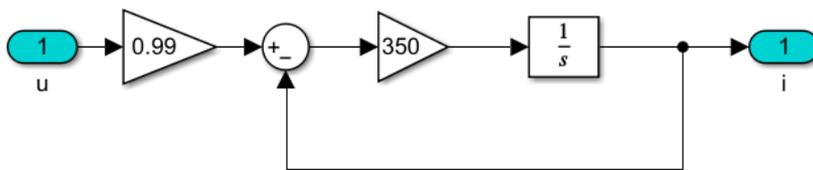


Figura 21: Modelo del Bloque Corriente.

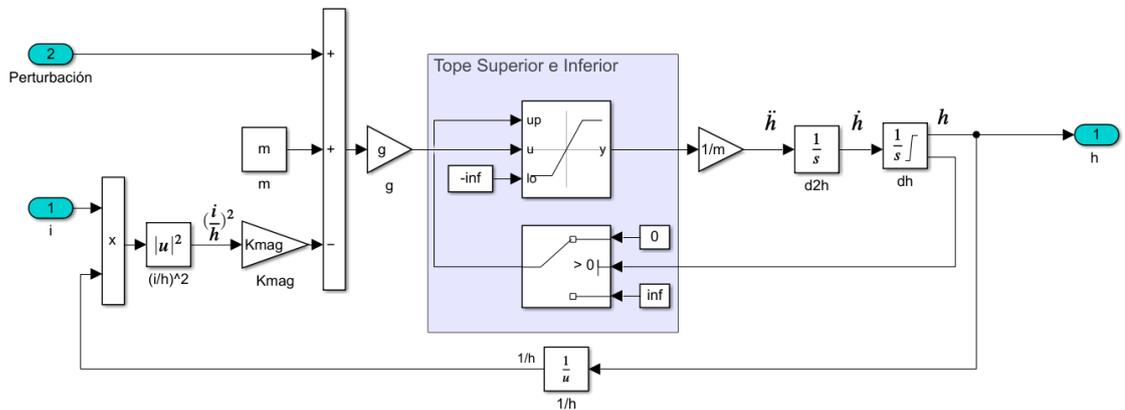


Figura 22: Modelo No Lineal del Bloque Magnético

Como se mencionó anteriormente, el modelo de la planta no incluye fricción entre el elemento levitado y la guía vertical que lo mantiene centrado con respecto al electroimán. A su vez, el modelo considera que no hay “rebote”

cuando el elemento levitado impacta contra los topes inferiores o superiores. Por último, el modelo considera movimiento unidireccional en el eje h. Debido a la distancia existente entre el eje y la parte interna del elemento levitado, se produce desplazamientos en el plano xy, cuando se introduce la perturbación de posición, agregándole ruido a la medición de posición.

7.2.4 Identificación de la Perturbación

Para poder determinar la frecuencia de giro del motor vibrador, y por ende la frecuencia de la perturbación, se midió la vibración introducida en la medición de posición del elemento levitado en función de la tensión aplicada al motor en un rango de tensión de 0.96v a 1.80v. Luego, a partir de esta medición de posición, se realizó el espectro de potencia de la señal y se determinó la frecuencia de giro (Figura 23). En esta figura se puede observar las distintas frecuencias características de la vibración medida.

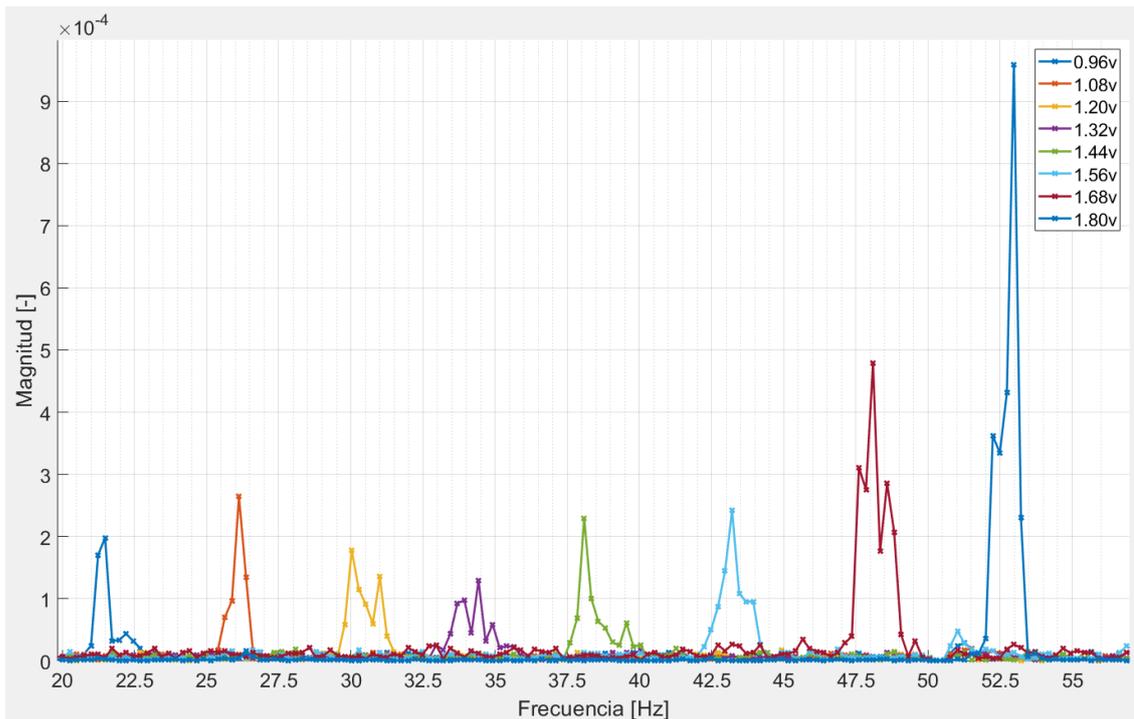


Figura 23: Espectro de potencia de la medición de posición para distintas tensiones aplicadas al motor vibrador.

En la Figura 24, se observa la calibración obtenida entre la tensión de entrada y la velocidad de giro del motor vibrador. Esta frecuencia va desde los 21 Hz a los 53 Hz aproximadamente (132 rad/s a 333 rad/s).

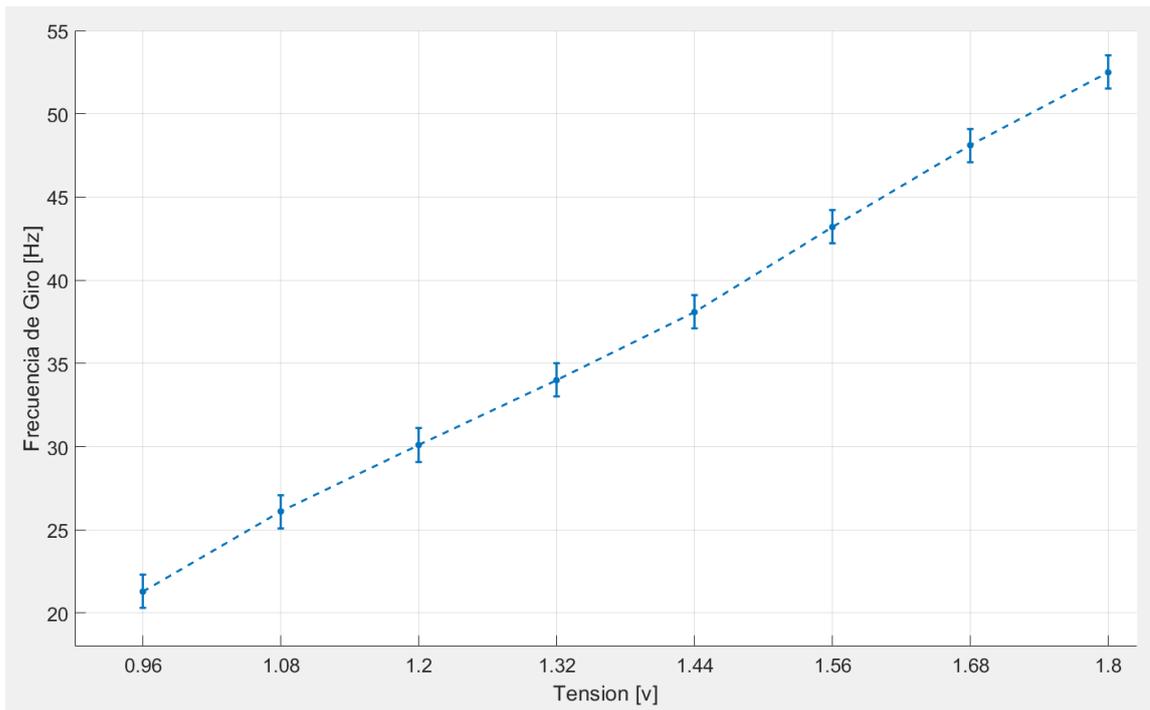


Figura 24: Relación entre la frecuencia de giro y la tensión aplicada al motor vibrador.

Se debe tener en cuenta que, debido a que la frecuencia de perturbación es inferida indirectamente a través de la tensión entre bornes del motor vibrador, puede resultar que la velocidad de giro del motor varíe por razones externas (calentamiento, rozamiento, etc.) logrando velocidades de giro con cierta variabilidad.

8 Diseño de Controladores

En este capítulo se describen los diferentes controladores diseñados para el rechazo de perturbación del tipo vibratoria en el elemento levitado. Los controladores implementados fueron diseñados a partir de técnicas de control clásico, control repetitivo y control moderno con estimadores MIMO y Filtro de Kalman. A estos estimadores se les introdujo el modelo de perturbación para que atenúen la perturbación introducida.

8.1 Definición de los requerimientos del controlador

Como se vio anteriormente, la planta posee polos en el semiplano derecho, es decir dinámica inestable. Debido a esto, se introducen limitaciones en los requerimientos que debe cumplir el sistema a lazo cerrado. Partiendo del análisis dado en Multivariable Feedback Control [8], el controlador debe ser lo suficientemente rápido para estabilizar al sistema, introduciendo así requerimientos de diseño. Para un polo real " p " en el semiplano derecho, el ancho de banda del sistema a lazo cerrado debe ser, al menos $2p$ aproximadamente. A partir de esto, se propone realizar un controlador que sea, en lo posible, 2 veces más rápido que la dinámica del polo inestable a lazo abierto [8] ($\sim 140\text{rad/s}$). Para que la dinámica del lazo interno no afecte a la del lazo externo se busca que esta sea mayor a 5 veces. Por esta razón, la dinámica del lazo interno, se ubicará en una frecuencia mayor a 750rad/s ($\sim 120\text{Hz}$). De esta forma, la dinámica más rápida que se desea controlar quedará en torno a los 120Hz , se propone que el tiempo de muestreo sea de 2ms (entre 4 y 5 veces la dinámica del lazo interno). A partir de estas especificaciones se procede a realizar el diseño del controlador en el campo discreto.

8.2 Control Clásico

El primer controlador que se diseñó fue a partir de técnicas de control clásico. Este controlador no incluye rechazo de perturbaciones en su diseño, resultando útil para la comparación de la performance frente a otros controladores.

Para esto se diseñarán dos lazos de control (Figura 25), un lazo interno que controlará la corriente que circula por el electroimán y otro lazo externo para el control de posición del elemento levitado. El lazo externo genera el valor de referencia de corriente en función del error de posición del elemento levitado. Cabe mencionar que este diseño se realizará en el plano discreto.

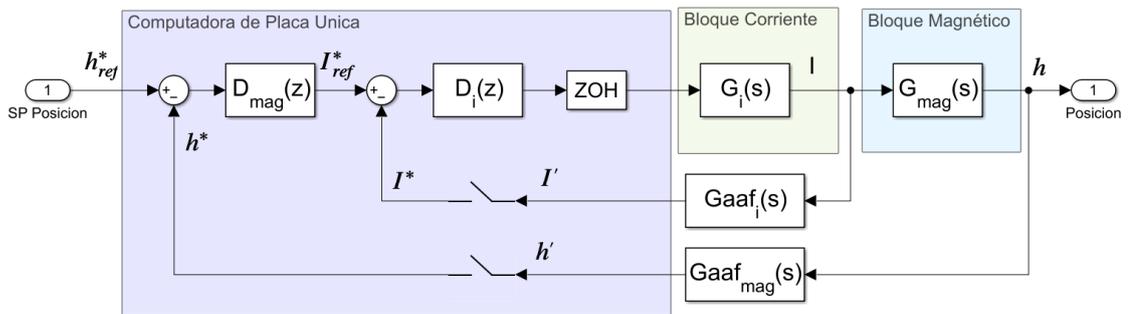


Figura 25: Esquema de los lazos de control implementados.

8.2.1 Lazo Interno

Como primer paso se hallará la función transferencia del sistema a lazo cerrado, en el plano discreto, para luego realizar el controlador del lazo interno.

Las ecuaciones, que rigen la dinámica de la planta a lazo cerrado, pueden expresarse como:

$$(I_{ref}^* - I^*)D_i(z) ZOH G_i(s) Gaaf_i(s) = I' \quad (17)$$

$$(h_{ref}^* - h^*)D_{mag}(z) = I_{ref}^* \quad (18)$$

$$[(h_{ref}^* - h^*)D_{mag}(z) - I^*]D_i(z) ZOH G_i(s)G_{mag}(s)G_{aaf_{mag}}(s) = h' \quad (19)$$

Definiendo, aplicando transformada Z en la ec. (17) y despejando, se obtiene la función transferencia del lazo interno (21):

$$Z[I'] = I(z) \quad (20)$$

$$Z[h] = H(z)$$

$$I_{ref}(z)D_i(z) Z[ZOH G_i G_{aaf_i}] = I(z)(1 + D_i(z) Z[ZOH G_i G_{aaf_i}]) \quad (21)$$

$$\boxed{\frac{I(z)}{I(z)_{ref}} = T_i(z) = \frac{D_i(z) \overline{G_i G_{aaf_i}(z)}}{(1 + D_i(z) \overline{G_i G_{aaf_i}(z)})}} \quad (22)$$

8.2.2 Lazo Externo

La función transferencia del lazo externo puede hallarse reemplazando la ec. (22) en ec. (19).

$$[(h_{ref}^* - h^*)D_{mag}(z) - T_i(z) I(z)_{ref}]D_i(z) ZOH G_i(s)G_{mag}(s)G_{aaf_{mag}}(s) = h' \quad (23)$$

Aplicando transformada Z en ec. (18) y reemplazando en ec. (23)

$$[(h_{ref}^* - h^*)D_{mag}(z) - T_i(z)(h_{ref}^* - h^*)D_{mag}(z)]D_i(z) ZOH G_i(s)G_{mag}(s)G_{aaf_{mag}}(s) = h' \quad (24)$$

$$[1 - T_i(z)](h_{ref}^* - h^*)D_{mag}(z)D_i(z) ZOH G_i(s)G_{mag}(s)G_{aaf_{mag}}(s) = h' \quad (25)$$

Aplicando transformada Z en ec.(25):

$$[1 - T_i(z)](H_{ref}(z) - H(z))D_{mag}(z)D_i(z)Z[ZOH G_i G_{mag} G_{aaf_{mag}}] = H(z) \quad (26)$$

Reemplazando ec.(22) en ec. (26) y reordenando, se obtiene:

$$\left[\frac{D_{mag}(z)D_i(z)\overline{G_l G_{mag} G_{aafmag}(z)}}{(1 + D_i(z) \overline{G_l G_{aafi}(z)})} \right] (H_{ref} - H) = H \quad (27)$$

Siendo la función transferencia del lazo externo, la mostrada en ec (28):

$$\frac{H(z)}{H(z)_{ref}} = \frac{\frac{D_{mag}(z)D_i(z)\overline{G_l G_{mag} G_{aafmag}(z)}}{(1 + D_i(z) \overline{G_l G_{aafi}(z)})}}{1 + \frac{D_{mag}(z)D_i(z)\overline{G_l G_{mag} G_{aafmag}(z)}}{(1 + D_i(z) \overline{G_l G_{aafi}(z)})}} \quad (28)$$

8.2.2.1 Diseño del Controlador del Lazo Interno

El controlador del lazo interno deberá garantizar que la dinámica de este sea mucha más rápida que la del lazo externo, de manera de afectarla lo menos posible.

Como primer paso del diseño del controlador $D_i(z)$ se realiza un diagrama del lugar geométrico de raíces (Figura 26) de la ecuación característica de la F.T.

$$\frac{I(z)}{I(z)_{ref}}$$

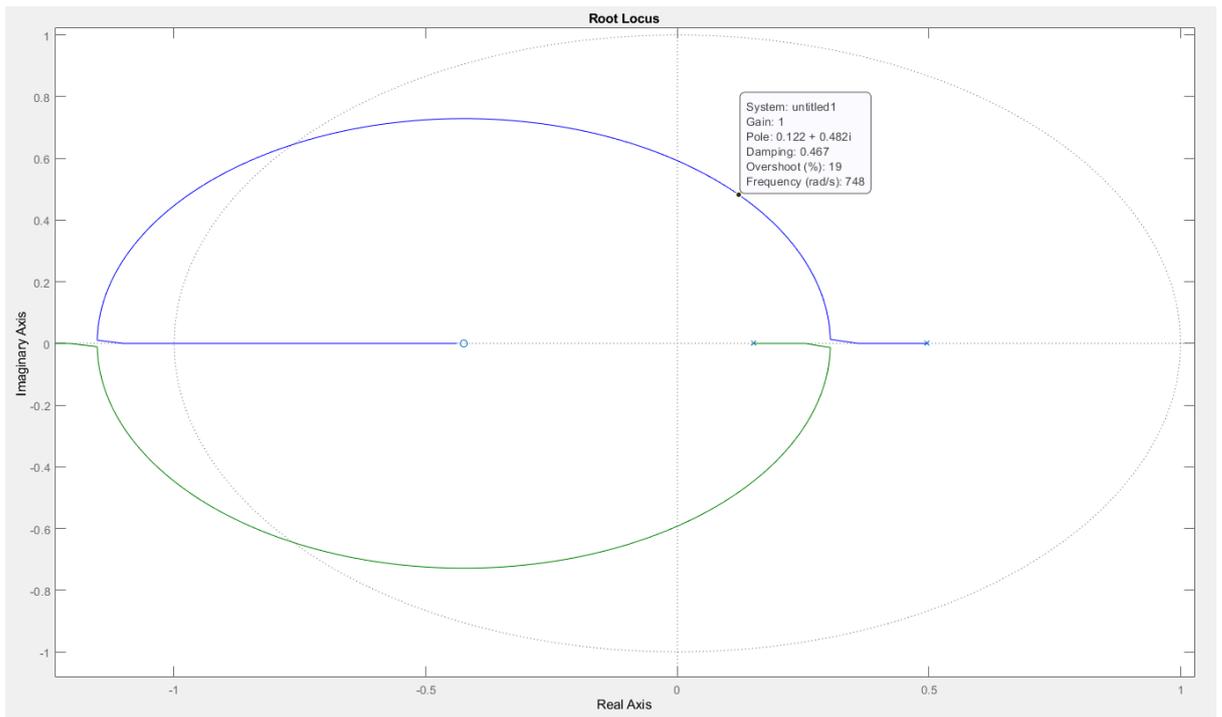


Figura 26: Lugar geométrico de las raíces para el lazo interno del controlador clásico

En este diagrama se puede observar que con solo un controlador proporcional de ganancia unitaria se logra cumplir con la especificación de velocidad de respuesta impuesta anteriormente. Al situar el polo de lazo cerrado de corriente en este punto, se garantiza la dinámica de corriente no afectará la dinámica de posición, por lo que el controlador de corriente será:

$$\boxed{D_i(z) = 1.0} \quad (29)$$

Diseño del Controlador del Lazo Externo

Para diseñar el controlador de posición, se realiza el diagrama de bode de la ecuación característica del sistema discretizado (ec. (28)), para distintos puntos de equilibrio (de 2 a 7mm). Se asume que el controlador de posición D_{mag} es del tipo proporcional y unitario,

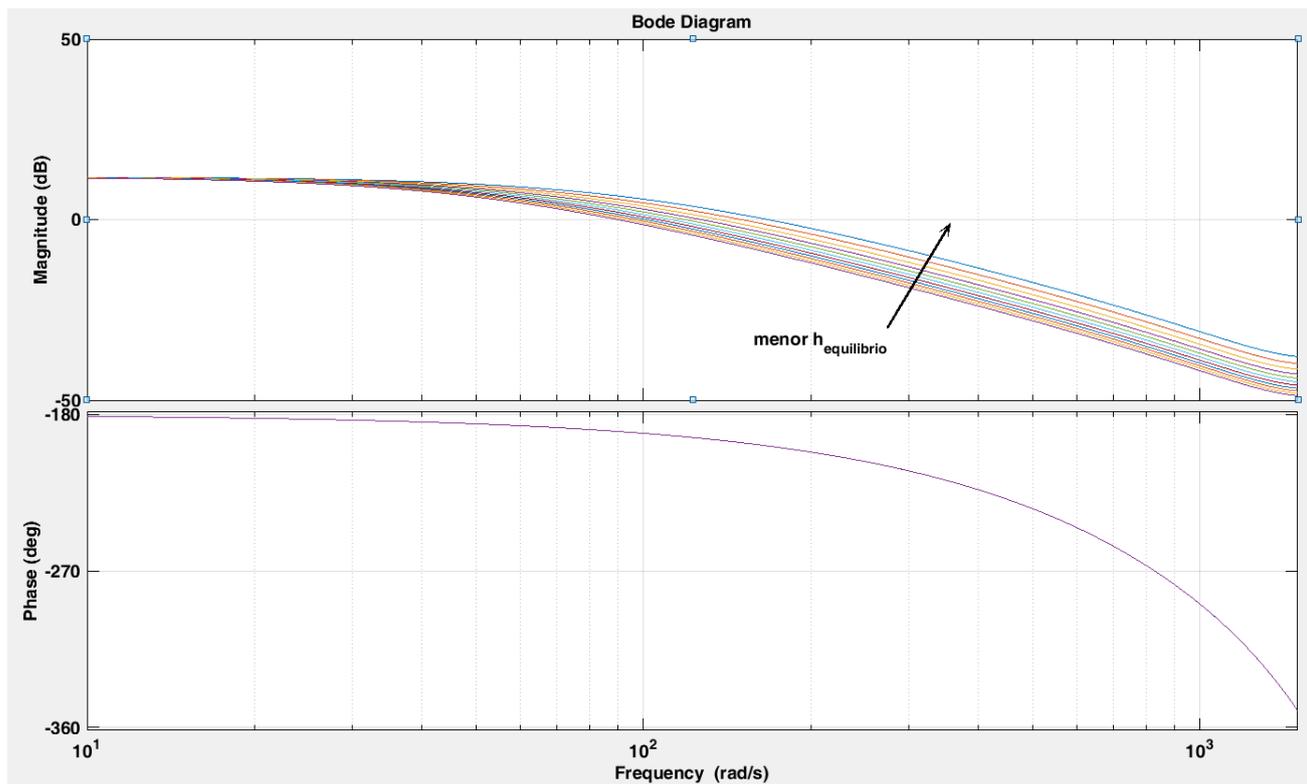


Figura 27: Diagrama de Bode del sistema con lazo interno realimentado para distintos huecos de equilibrio.

En este diagrama se puede observar que no es posible controlar al sistema inestable a partir de un controlador proporcional. También se observa que para puntos de equilibrio más cerca del electroimán (menor hueco), el sistema se vuelve más rápido (mayor ancho de banda) pero con menor margen de fase. Debido a esto, al disminuir el hueco, el controlador estará más exigido para lograr la estabilidad del sistema.

El punto de equilibrio del levitador magnético, como se mencionó anteriormente, será el correspondiente a una acción de control u de 0.5 ($h = 3.76\text{mm}$). Esto es debido a que la acción de control está acotada entre 0 y 1 (acción de control con respecto al equilibrio acotada entre -0.5 y 0.5).

A partir de definir el punto de operación del sistema a una distancia de 3.76mm del electroimán (ec. 8), se procede a diseñar un controlador que

cumpla los requerimientos antes descritos para luego verificarlo en puntos de operación menos favorables.

Como se dijo anteriormente, por cuestiones de robustez, se desea que el sistema realimentado tenga al menos una frecuencia de corte de $\sim 140 \text{ rad/s}$ (al menos el doble que el polo inestable de la planta). Si se realiza el diagrama del bode (con ganancia positiva y controlador proporcional), a partir de la ecuación característica de la Función Transferencia del lazo externo (ec. (28)), se obtiene:

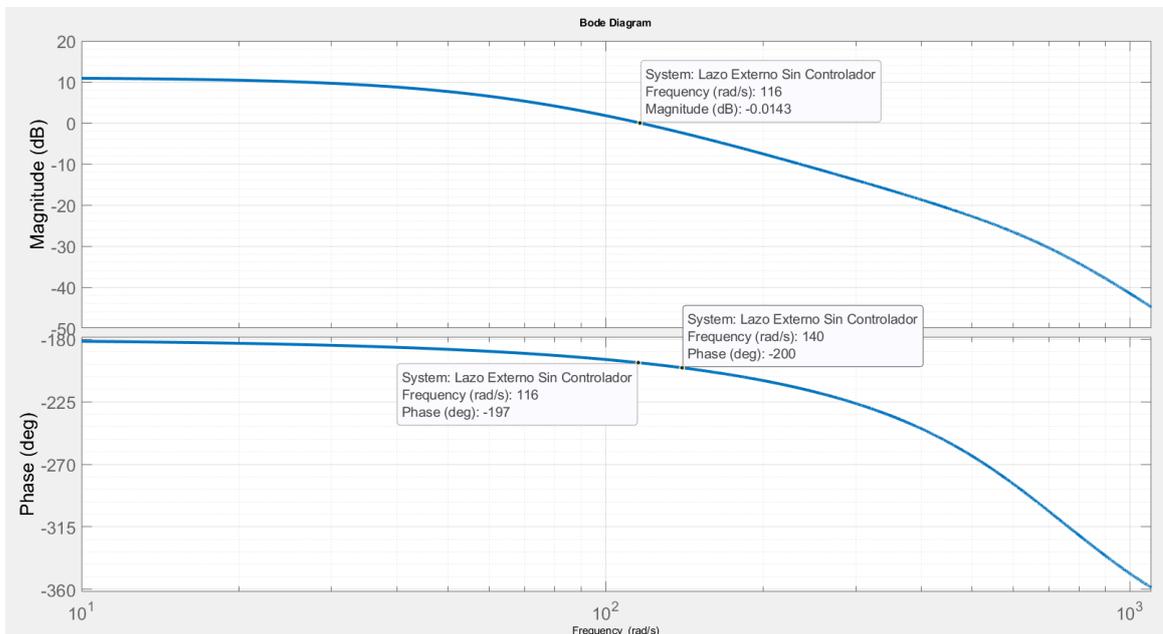


Figura 28: Diagrama de Bode con lazo interno realimentado y un huelgo de equilibrio de 3.76mm

En esta frecuencia de corte deseada ($\omega_{cd} \sim 140 \text{ rad/s}$), el sistema tiene una fase de -200° . Por lo que se procede a aportar fase en este punto de manera de estabilizar al sistema y obtener un margen de fase adecuado para el control. Debido a esto, se diseñará un controlador tipo lead de manera de aportar fase al sistema y aumentarle el ancho de banda al mismo.

La máxima contribución de fase (ϕ_{max}) dada por el controlador será de 70° en la frecuencia de corte deseada ($\omega_{cd} = 140 \text{ rad/s}$).

El controlador posee una estructura del tipo:

$$D_{mag}(s) = K \frac{T s + 1}{\alpha T s + 1} \quad (30)$$

Siendo:

$$\alpha = \frac{1 - \text{sen}(\phi_{max})}{1 + \text{sen}(\phi_{max})}; \omega_{cd} = \frac{1}{T\sqrt{\alpha}} \quad (31)$$

Debido a que no hay requerimientos de error de estado estacionario, se procede a ajustar la ganancia del controlador de manera de lograr el ancho de banda deseado con un margen de fase aceptable, pero cuidando no degradar la ganancia de estado estacionario al disminuir demasiado la ganancia. A partir de este ajuste se obtiene que el controlador, en el plano continuo, es:

$$D_{mag}(s) = 7.244 \frac{s + 53.59}{s + 746.4} \quad (32)$$

La implementación del algoritmo de control discreto, a fin de evitar distorsión de frecuencia, se realiza un mapeo de polos y ceros del tipo:

$$z = e^{sT_s} \quad (33)$$

Obteniéndose:

$$D_{mag}(z) = -4.0 \frac{(z - 0.8983)}{(z - 0.1452)} \quad (34)$$

Si se realiza el diagrama de bode del sistema realimentado con el compensador $D_{mag}(z)$ (ec. (34)) se puede observar que el sistema posee un

margen de fase de 36° (en 157rad/s) y un margen de ganancia de 6.41dB (en 354rad/s) (Figura 29).

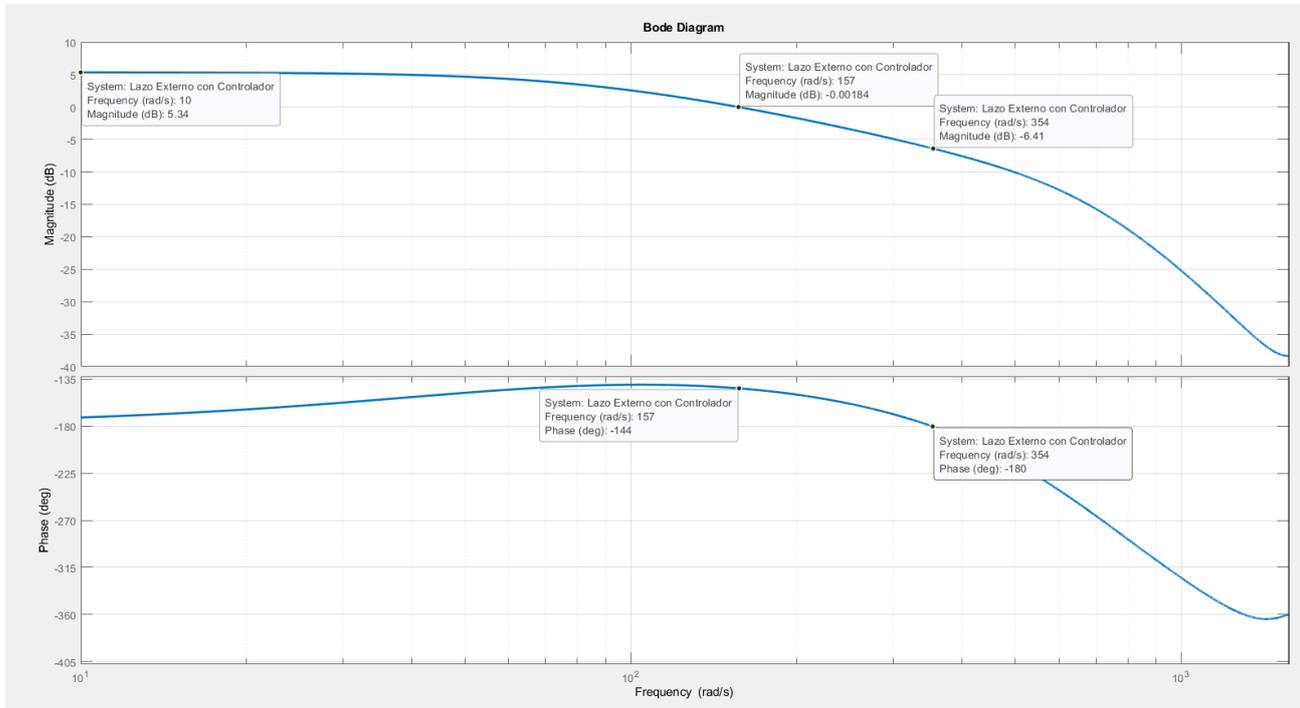


Figura 29: Diagrama de bode del sistema realimentado.

En el diagrama de bode anterior se observa que se obtiene un ancho de banda mayor a los 140rad/s que se había fijado como requerimiento necesario para estabilizar al sistema.

8.2.3 Respuesta dinámica del sistema

Debido a la característica inestable del sistema, solo es posible validarlo al cerrar el lazo de control. Para realizar esta validación se implementó el controlador diseñado anteriormente y se comparó la respuesta entre el modelo lineal y no lineal de la planta ante cambios abruptos del valor deseado de posición (Figura 30). Se observa al comparar la medición de posición contra el modelo lineal y no lineal que el modelo matemático planteado es

adecuado ya que, si bien posee diferencias, la dinámica es representada adecuadamente.

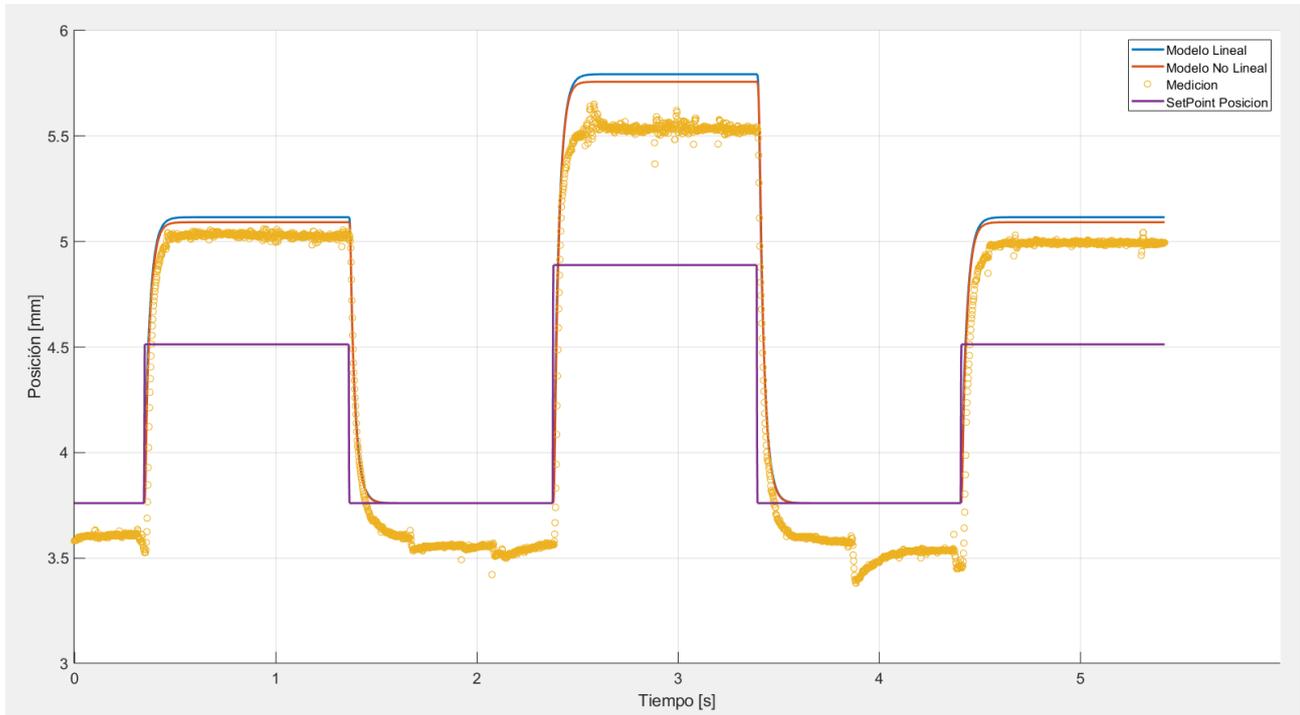


Figura 30: Respuesta dinámica del sistema realimentado con un controlador clásico ante cambios de posición de referencia.

En la Figura 31 se aprecia una ampliación temporal de la dinámica de posición ante un cambio abrupto en la referencia (entrada tipo escalón). En ella se observa un tiempo de establecimiento de $\sim 100\text{ms}$ y errores de estado estacionario de hasta el 100% con respecto al setpoint de posición.

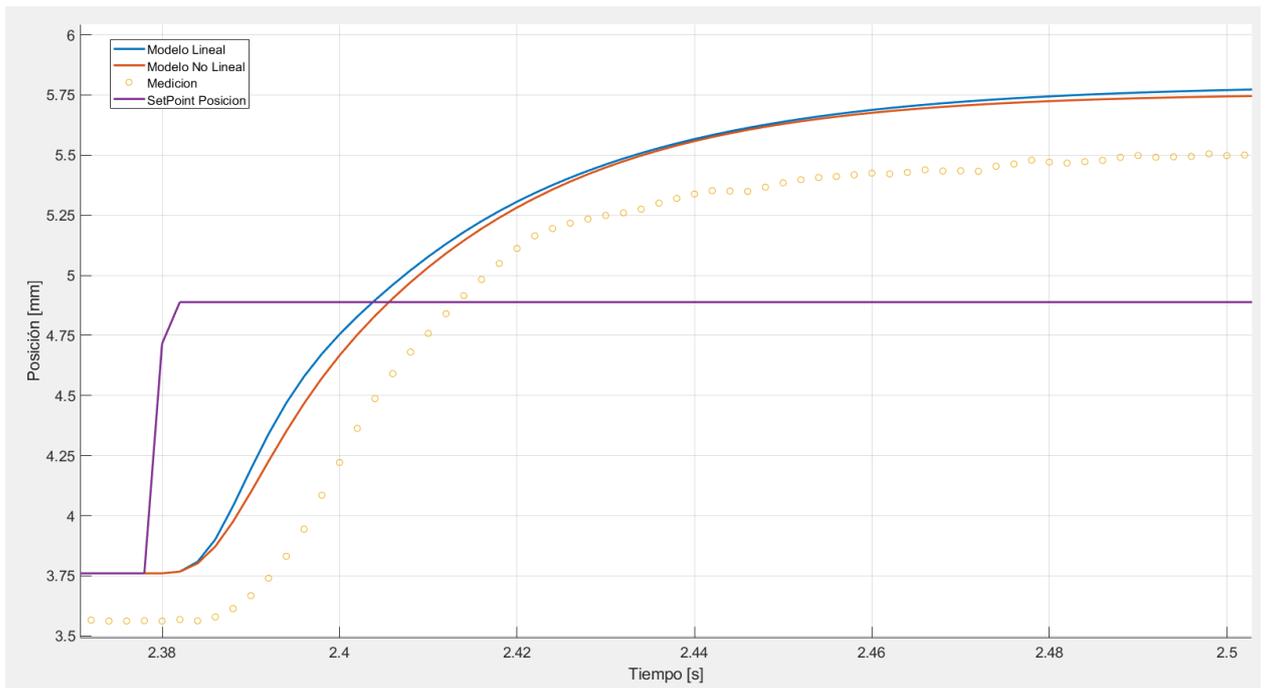


Figura 31: Dinámica de posición ante un cambio tipo escalón en la referencia

En la Figura 32 se muestra la acción de control tanto modelada como medida ante los mismos cambios de setpoint de posición.

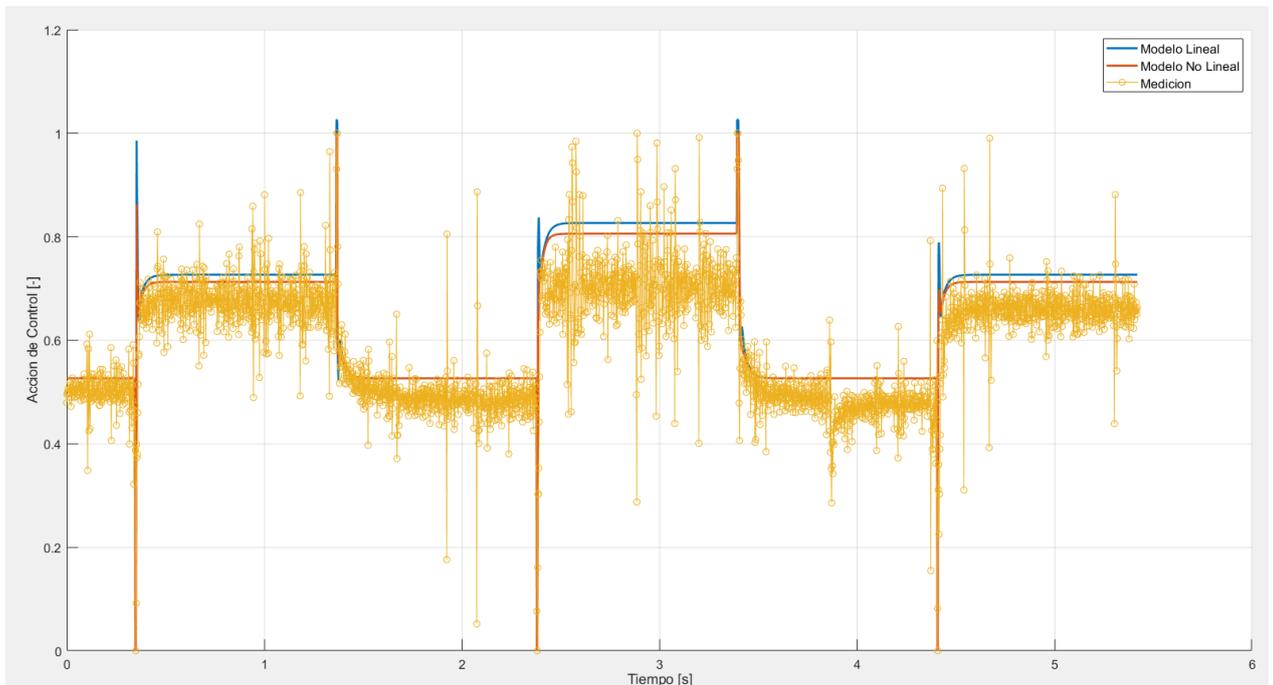


Figura 32: Acción de control del controlador clásico ante cambios de referencia del tipo escalón.

8.3 Control Repetitivo

El control repetitivo digital es una técnica de control que permite el seguimiento de referencias periódicas y el rechazo de perturbaciones periódicas. Los controladores repetitivos son diseñados asumiendo que las señales a seguir/rechazar tienen una frecuencia fija y conocida, siendo su principal desventaja la degradación de desempeño cuando esta frecuencia varía [9].

8.3.1 Introducción al Control Repetitivo Digital

El control repetitivo (RC) es una técnica basada en el principio de modelo interno (IMP). Este establece que para seguir/rechazar una señal es necesario que su generador este incluido en el lazo de control. El RC utiliza el IMP para el seguimiento/rechazo de señales periódicas asumiendo que la frecuencia de las señales a seguir/rechazar es constante y conocida [10]. Esto lleva a la selección de un número de muestras por periodo que es incorporado estructuralmente en el algoritmo de control. La implementación digital del RC involucra la discretización de las señales externas. De este modo, una señal T_p -periodica muestreada con periodo T_s se convierte en una señal N -periodica ($N=T_p/T_s$). El modelo interno de este tipo de señales es de la forma [11] [12]:

$$C_r(z) = \frac{z^{-N}}{1 - z^{-N}} = \frac{1}{z^N - 1} \quad (35)$$

Este modelo interno se obtiene a partir de la transformada Z de la representación en series de Fourier discreta de la señal periódica.

La principal característica de la ec. (35) es que tiene N polos uniformemente espaciados sobre el círculo unitario del plano complejo Z . De esta manera el controlador proporciona ganancia infinita en las frecuencias $\omega=2k\pi/N$ (con $K=0,1,\dots,N-1$). Esta elevada ganancia compromete la estabilidad y robustez del sistema. Por eso se incluyen un filtro $Q(z)$ empleado para proveer robustez ante dinámicas no modeladas. Aunque existen diversas estructuras, la más utilizada es la mostrada en la Figura 33.

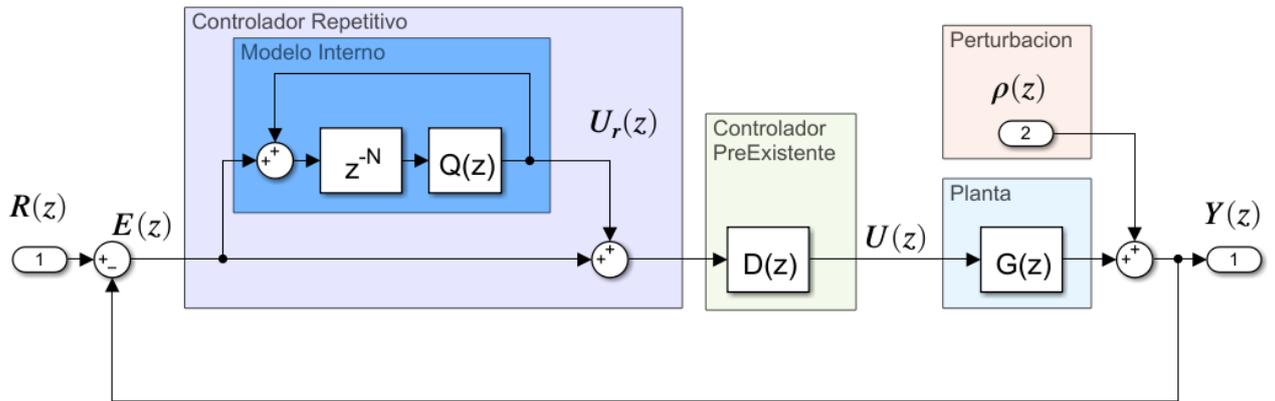


Figura 33: Diagrama de Bloques del controlador Repetitivo

En esta estructura el controlador repetitivo es adicionado a un controlador preexistente $D(z)$ el cual es diseñado para la estabilidad de la planta $G(z)$. De esta manera la ec. (35) queda de la forma:

$$C_r(z) = \frac{z^{-N} Q(z)}{1 - z^{-N} Q(z)} \quad (36)$$

8.3.2 Análisis de estabilidad del Algoritmo de Control Repetitivo

El análisis de estabilidad del sistema a lazo cerrado no resulta trivial por ello se sigue el análisis planteado por [10] y [13]. La función sensibilidad del sistema a lazo cerrado resulta:

$$S(z) = \frac{E(z)}{R(z)} = \frac{1}{1 + D(z)G(z)} \frac{1}{1 + C_r(z)T_0(z)} = S_0(z)S_M(z)$$

Siendo:

$S_0(z) = \frac{1}{1 + D(z)G(z)}$	Función de sensibilidad
$T_0(z) = \frac{D(z)G(z)}{1 + D(z)G(z)}$	Función sensibilidad complementaria del sistema sin Control Repetitivo

$S_M(z) = \frac{1 - z^{-N}Q(z)}{1 - z^{-N}Q(z)(1 - T_0(z))}$	Función sensibilidad modificante
--	----------------------------------

El sistema de la Figura 33 es estable si:

1. $S_0(z)$ es estable, lo que proviene de la estabilidad del lazo subyacente
2. $1 - z^{-N}Q(z)$ es estable, lo cual se cumple definiendo $Q(z)$ como un filtro FIR.
3. $\|Q(z)\|_\infty < 1.0$
4. $\|Q(z)(1 - T_0(z))\|_\infty < 1.0$

8.3.3 Diseño de Control Repetitivo

Como se mencionó anteriormente, el bloque de control repetitivo se anexa a un sistema ya estabilizado con anterioridad. En este caso se utilizará la estructura de control diseñada anteriormente con las técnicas de control clásico, utilizando los controladores $D_i(z)$ y $D_{mag}(z)$ definidos en las ecuaciones (29) y (34).

Como primer paso para el diseño del controlador repetitivo, se debe definir la frecuencia de la perturbación a atenuar. Si se supone que se desea rechazar una perturbación de 33Hz, se debe calcular el retraso temporal del generador de funciones:

$$N = \frac{w_s}{w_p} = \frac{2\pi 500\text{Hz}}{2\pi 33\text{Hz}} \approx 15$$

Para el filtro $Q(z)$ se propone un filtro pasabajo FIR de orden 10, con una frecuencia de corte igual a la frecuencia de la perturbación (33Hz). Este filtro FIR es modulado a partir de ventana tipo Hamming, el cual fue calculado mediante uso de software matemático.

$$Q(z) = \frac{a_0 z^{10} + a_1 z^9 + \dots + a_9 z + a_{10}}{z^{10}}$$

$$Q(z) = 0.005 \frac{(0.69z^{10} + 2.07z^9 + 6.2z^8 + 12.4z^7 + 18.2z^6 + 20.5z^5 + 18.2z^4 + 12.4z^3 + 6.2z^2 + 2.07z + 0.69)}{z^{10}}$$

En la Figura 34 se analiza el bode de $Q(z)(1 - T_0(z))$ de manera de garantizar que la norma infinito sea menor a 1.0 (menor a 0dB) y así garantizar la estabilidad del controlador repetitivo.

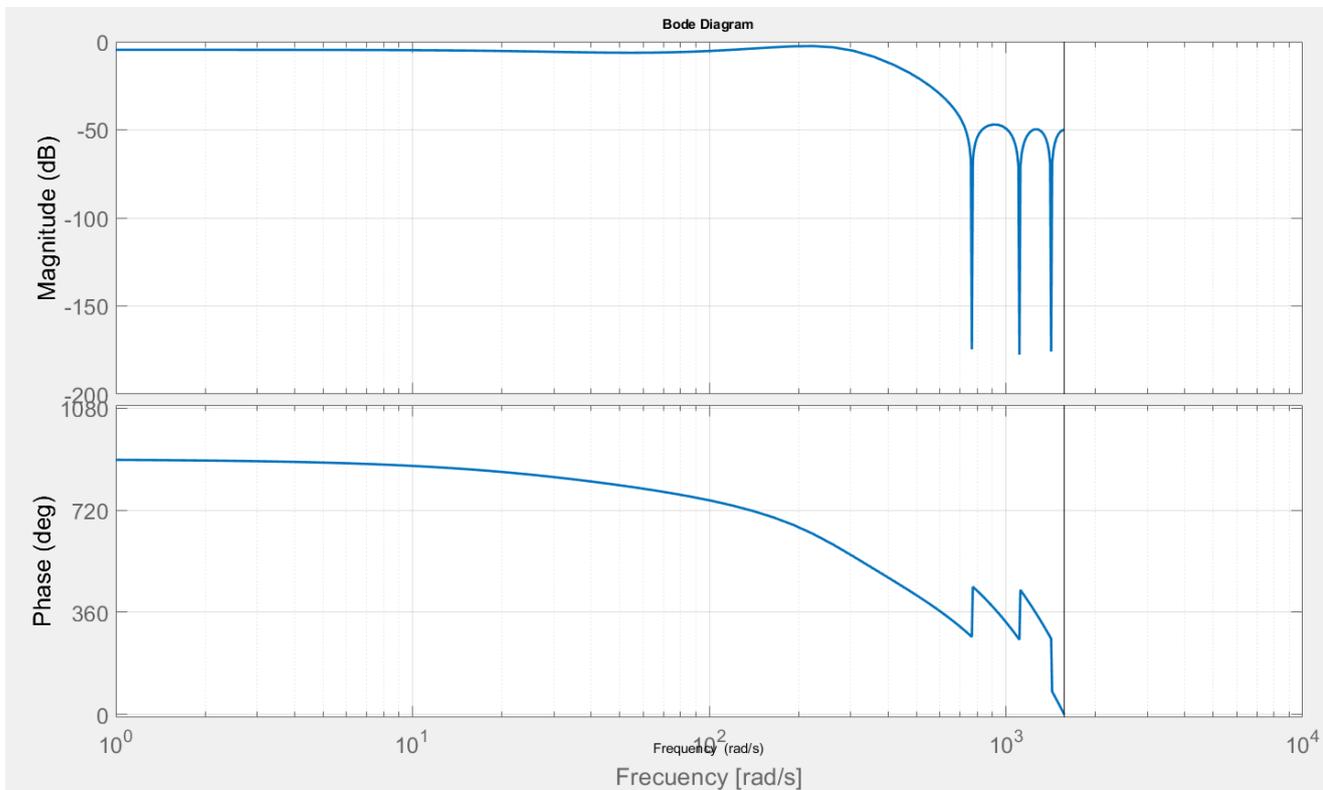


Figura 34: Diagrama de bode de $Q(z)(1 - T_0(z))$

Para terminar de validar el diseño, se comparó en simulación el controlador repetitivo frente al controlador clásico. En la Figura 35 se muestra la posición del elemento levitado ante una perturbación del tipo sinusoidal en la frecuencia de diseño.

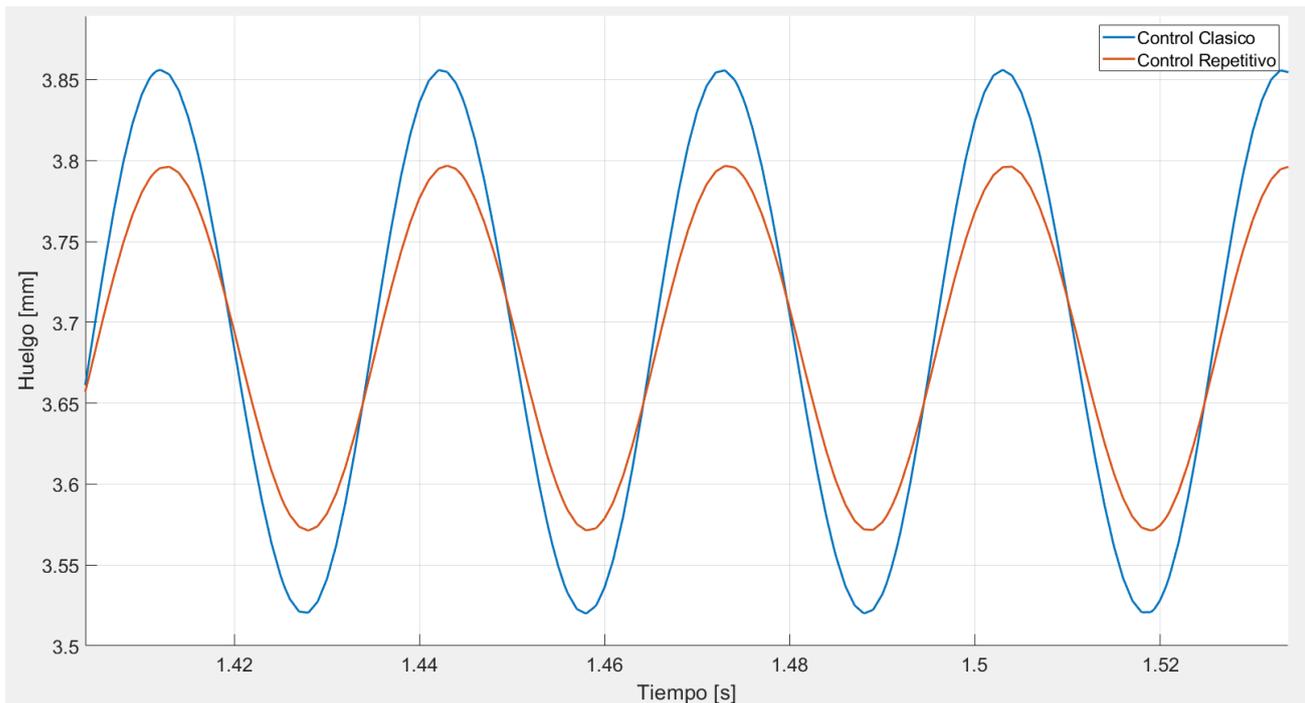


Figura 35: Simulación donde se compara la performance del Controlador Repetitivo frente a un Controlador Clásico ante una perturbación de 33Hz.

Se observa en simulación que el RC disminuye la perturbación en la posición del elemento levitado un 30% aproximadamente (considerando la amplitud pico a pico).

8.4 Diseño por Variable de Estado con Estimador Extendido para Rechazo de Perturbación.

Se realizará un controlador que sea capaz de rechazar perturbaciones periódicas de frecuencia ω_p conocida, la cual podría ser fija o variable. Este controlador se diseñará en el espacio de estados, es decir que el modelo del sistema estará descrito a partir de ecuaciones diferenciales ordinarias (EDO) y el control será a partir de las variables de estados del sistema. Debido a que no todas las variables de estado del sistema a controlar son medidas, se

implantarán estimadores de manera de calcular estas variables de estado a partir de las mediciones de la planta y su modelo matemático.

Los pasos del método de diseño son los siguientes:

- Seleccionar la ubicación de los polos a lazo cerrado de la planta para cumplir satisfactoriamente la respuesta dinámica del sistema.
- Diseñar un observador extendido que permita estimar las variables del estado de la planta y la perturbación a partir de: las mediciones, la acción de control y el modelo matemático.
- Combinar la ley de control y el estimador.

En la Figura 36 se muestra la planta $G(s)$ con la perturbación $\rho(t)$ y el esquema de Control más el Observador Extendido utilizado.

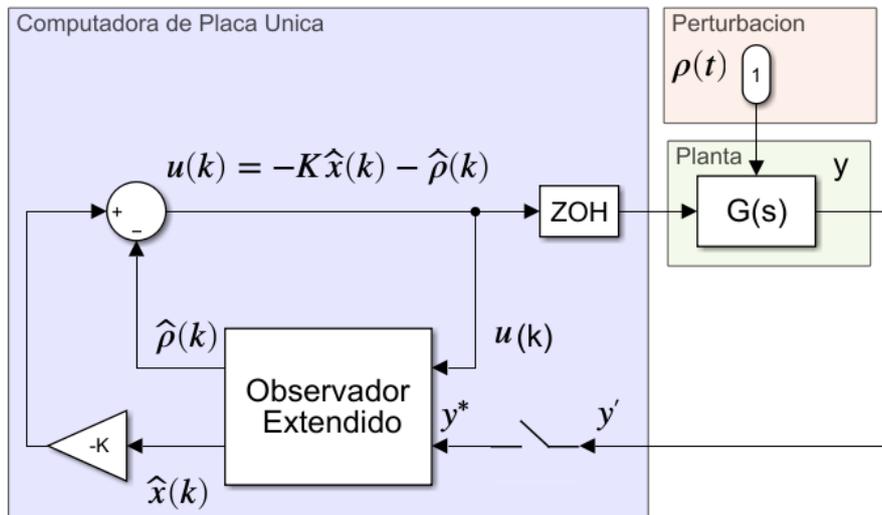


Figura 36: Diagrama del esquema de control moderno con estimador extendido para rechazo de perturbaciones

Este método de control se basa en aumentar el orden del estimador con la ecuación de la perturbación de manera que nos permita cancelar su efecto en la salida [14].

Primero es necesario recordar y definir la representación del sistema como variables de estado. Como se mencionó anteriormente, en las ecuaciones (15) y (16), el modelo lineal continuo de la planta puede representarse como:

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t) \\ y(t) &= Cx(t) \end{aligned} \quad (37)$$

Mientras que el mismo sistema, puede ser representado en variable de estado discreta, según nomenclatura de [15], como:

$$\begin{aligned} x(k+1) &= Gx(k) + Hu(k) \\ y(k) &= Cx(k) \end{aligned} \quad (38)$$

Donde:

$$G = e^{AT_s} \quad (39)$$

$$H = \int_0^{T_s} e^{A\eta} d\eta \quad (40)$$

Siendo e^A la función exponencial matricial definida como:

$$e^A = \sum_{k=0}^{\infty} \frac{A^k}{k!} \quad (41)$$

Si se considera una perturbación ρ_1 que ingresa al sistema, esta puede ser descrita por las siguientes ecuaciones:

$$\begin{aligned} \dot{x} &= Ax + Bu + B_1\rho_1 \\ y &= Cx + J\rho_1 \end{aligned} \quad (42)$$

El modelo de perturbación ρ_1 satisface la siguiente ecuación:

$$\ddot{\rho}_1 + \alpha_1\dot{\rho}_1 + \alpha_2\rho_1 = 0 \quad (43)$$

Siendo la perturbación armónica de frecuencia ω_p , los coeficientes α_1 y α_2 serán 0 y $-\omega_p^2$ respectivamente.

Se puede demostrar, según [14], que hay una perturbación equivalente ρ que satisfice la misma ecuación que ρ_1 e ingresa al sistema en el mismo lugar que la señal de control.

Si asumimos que la planta no posee ceros en ninguna raíz (lo cual es cierto para esta planta en particular), podemos reemplazar ρ_1 por ρ :

$$\begin{aligned}\dot{x} &= \mathbf{A}x + \mathbf{B}(u + \rho) \\ y &= \mathbf{C}x\end{aligned}\tag{44}$$

Si podemos estimar la perturbación equivalente, podemos agregar a la señal de control un término $-\hat{\rho}$ que cancele el efecto de la perturbación real en estado estacionario. Para hacer esto se combinan las ecuaciones (43) y (44) en una descripción extendida por variable de estado:

$$\begin{aligned}\dot{z} &= \mathbf{A}_p z + \mathbf{B}_p u \\ y &= \mathbf{C}_p z\end{aligned}\tag{45}$$

Donde $z = [\rho \ \dot{\rho} \ x^T]^T$ y las matrices son:

$$\mathbf{A}_p = \begin{bmatrix} 0 & 1 & \mathbf{0} \\ -\alpha_2 & -\alpha_1 & \mathbf{0} \\ \mathbf{B} & \mathbf{0} & \mathbf{A} \end{bmatrix}, \mathbf{B}_p = \begin{bmatrix} 0 \\ 0 \\ \mathbf{B} \end{bmatrix}\tag{46}$$

$$\mathbf{C}_p = [0 \quad 0 \quad \mathbf{C}]$$

Cabe aclarar que el sistema descrito por la ec. (46) no es controlable debido a que no se puede afectar ρ desde u . Sin embargo, si (\mathbf{A}, \mathbf{C}) es observable y el sistema $(\mathbf{A}, \mathbf{B}, \mathbf{C})$ no posee cancelación cero-polo, el sistema extendido si es observable. Esto significa que podemos construir un observador extendido que estime los estados de la planta y de la perturbación ρ , para luego cancelarla. Las ecuaciones del estimador extendido y la acción de control son:

$$\dot{\hat{z}} = \mathbf{A}_p \hat{z} + \mathbf{B}_p u + \mathbf{L}(y - \mathbf{C} \hat{z}) \quad (47)$$

$$u = -[1 \ 0 \ \mathbf{K}] \hat{z} \quad (48)$$

La descripción por variable de estado, para el caso del levitador magnético, es:

$$\begin{bmatrix} \dot{\rho} \\ \ddot{\rho} \\ \dot{i} \\ \ddot{h} \\ \dot{h} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ -\omega_p^2 & 0 & 0 & 0 & 0 \\ \frac{1}{L} & 0 & -\frac{R}{L} & 0 & 0 \\ 0 & 0 & \frac{-2K_{mag} I_{eq}}{m h_{eq}^2} & 0 & \frac{-2K_{mag} I_{eq}^2}{m h_{eq}^3} \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \rho \\ \dot{\rho} \\ I \\ \dot{h} \\ h \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \frac{1}{L} \\ 0 \\ 0 \end{bmatrix} u \quad (49)$$

$$y = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \rho \\ \dot{\rho} \\ I \\ \dot{h} \\ h \end{bmatrix}$$

A partir de esta formulación extendida se realizará un observador MIMO y un Estimador a partir de un filtro de Kalman, solo considerando la medición de posición. Antes, sin embargo, se desarrollará el método para determinar la acción de control del sistema.

8.4.1 Diseño de la Ley de Control

Una de las técnicas más efectivas y ampliamente utilizadas para el diseño de sistemas de control lineal es el regulador cuadrático lineal óptimo (Linear Quadratic Regulator LQR) [14] [16]. Dado un sistema lineal:

$$\begin{aligned} \dot{x}(t) &= \mathbf{A}x(t) + \mathbf{B}u(t) \\ y(t) &= \mathbf{C}x(t) \end{aligned} \quad (50)$$

Se desea definir una acción de control $u(t)$ que minimice el funcional:

$$J = \int_0^{\infty} [\phi x^2(t) + u^2(t)] dt \quad (51)$$

Donde ϕ es un factor de peso dado por el diseñador del control que modifica el peso relativo entre el desempeño de los estados de la planta y su acción de control. La ley de control que minimiza J es la de la ley de control dada por:

$$u(k) = -\mathbf{K}\mathbf{x} = -[K_1 \quad K_2 \quad \dots \quad K_n] \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \quad (52)$$

Aquí el valor óptimo de K es el que ubica los polos en lazo cerrado en las raíces estables de la ecuación del lugar simétrico de las raíces

$$1 + \phi G_0(s)G_0(-s) = 0 \quad (53)$$

Donde $G_0(s)$ es la función transferencia a lazo abierto desde u a y :

$$G_0(s) = \frac{Y(s)}{U(s)} = C(s\mathbb{I} - A)^{-1}B \quad (54)$$

Al variar el parámetro ϕ , se modifica el peso del costo del error de seguimiento x^2 con respecto al esfuerzo de control u^2 .

Si se analizan los polos estables óptimos para la planta de levitación magnética con un factor de peso de 0.39, se obtiene:

$$1 + 0.39G_i(-s)G_{mag}(-s)G_i(s)G_{mag}(s) = [-354.4 \quad -140.0 \quad -140.0] \quad (55)$$

Este valor de ϕ fue elegido para que los polos a lazo cerrado del sistema sea, al menos, 2 veces la del polo inestable de la planta.

8.4.2 Diseño del Estimador Extendido

Como se mencionó anteriormente, se diseñarán dos observadores para determinar los estados y la perturbación del sistema:

- Observador Extendido MIMO (Multiple Input Multiple Output)

- Estimador Extendido a partir de un Filtro de Kalman

8.4.2.1 Estimador Extendido MIMO

Debido a que se tiene acceso a las mediciones de corriente (l) y altura (h) del levitador magnético, se procedió a realizar un estimador extendido del tipo MIMO. Los estados del sistema pueden estimarse como:

$$\begin{aligned}\hat{x}(k+1) &= G\hat{x}(k) + Hu(k) + L[y(k) - \hat{y}(k)] \\ \hat{y}(k) &= C\hat{x}(k)\end{aligned}\tag{56}$$

Se puede demostrar que la dinámica del error de estimación([14] [15]) está dada por los n autovalores de la matriz (G-LC). Debido a que el sistema es de múltiples salidas, existen infinitas matrices L que permiten la ubicación deseada para la dinámica del error de estimación. Para solucionar este problema se propone una única salida ficticia (y^*) que resulta ser una combinación lineal de las salidas existentes [17].

$$y^*(k) = f_1y_1(k) + f_2y_2(k) + \dots + f_ny_n(k) = F \cdot y(k)\tag{57}$$

Con

$$F = [f_1 \ f_2 \ \dots \ f_q]\tag{58}$$

En donde los valores de f_i son elegidos arbitrariamente.

El modelo de estado discreto con la nueva salida resulta:

$$\begin{aligned}x(k+1) &= Gx(k) + Hu(k) \\ y^*(k) &= FCx(k) = C^*x(k)\end{aligned}\tag{59}$$

Resultando C^* vector fila.

Una vez establecidos los autovalores para el estimador ($\lambda_1 \ \lambda_2 \ \dots \ \lambda_n$) se halla el vector L^* para el estimador a partir de la salida ficticia. El modelo del estimador resultante es:

$$\hat{x}(k + 1) = (G - L^*C^*)\hat{x}(k) + Hu(k) + L^*y^*(k) \quad (60)$$

Reemplazando y^* y C^* por su equivalente del modelo original:

$$\hat{x}(k + 1) = (G - L^*FC)\hat{x}(k) + Hu(k) + L^*Fy(k) \quad (61)$$

Siendo:

$$L = L^* \cdot F \quad (62)$$

Características particulares:

- Las columnas de la matriz L son combinación lineal entre si
- Se pueden utilizar salidas cuyo contenido de ruido sea bajo y el error cometido en la estima puede verse reducido comparado con el que se obtendría desde una sola salida.

Con $\omega_p = 207 \frac{rad}{s}$ ($\sim 33Hz$) y $F = [0.2 \ 1.0]$

Para hallar los polos deseados de estimador se realizó el Root Locus Simétrico (Figura 37) de manera de obtener la ubicación de los polos de estimación óptimos:

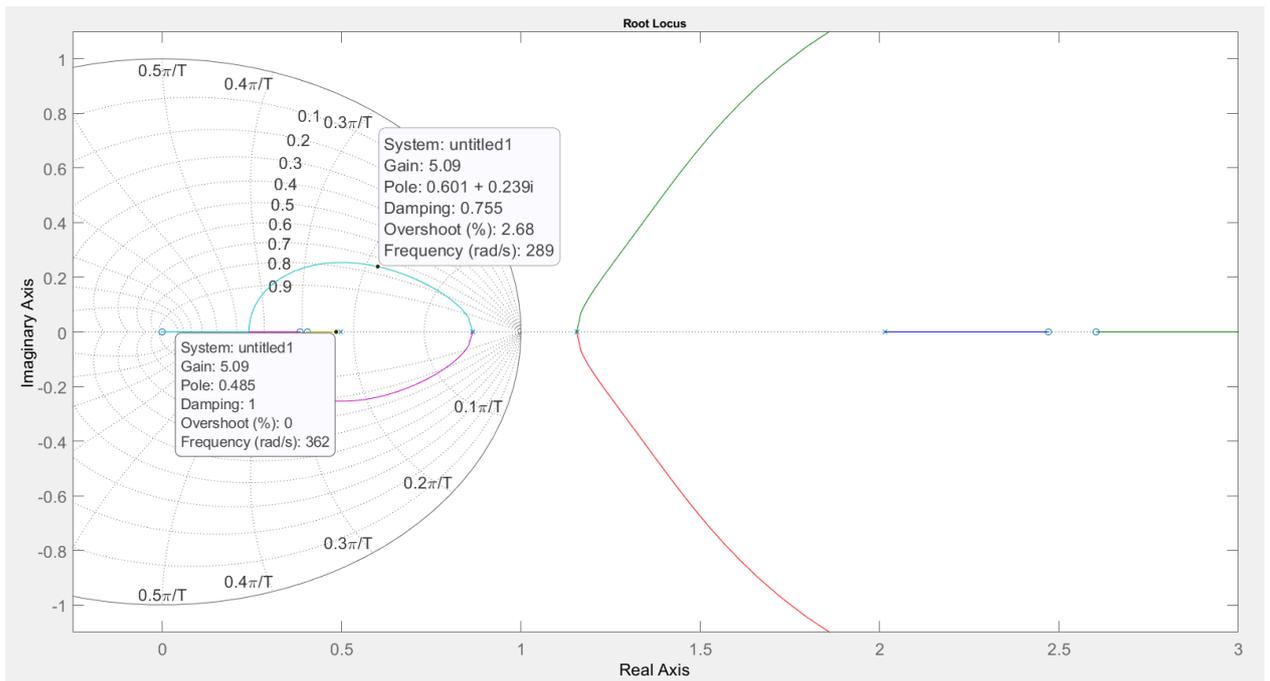


Figura 37: Root Locus simetrico para determinar la posicion optimo de los polos del estimador.

El valor de peso elegido ($Gain=5.09$) corresponde al doble de la frecuencia en la dinámica de posición de la ley de control. En cambio, la dinámica de estimación para la variable de estado de corriente es comparable a la dinámica elegida de la ley de control. Nótese a su vez, que solo se escogen el valor óptimo de estimación para los estados originales de la planta (sin considerar la estimación de perturbación). Al realizar el RLS se produce una cancelación cero-polo para la dinámica de perturbación, haciendo que el valor óptimo se encuentre en la misma frecuencia de la perturbación. La ubicación de los polos del estimador radica en un compromiso entre velocidad de estimación y la amplificación del ruido de medición [14]. Cuando la dinámica de estimación aumenta se amplifica, a partir de la señal medida, el ruido de los estados estimados.

Es decir, el valor de ganancia del estimador, puede calcularse como:

$$L^* = \left(acker(G^T, (F \cdot C)^T, [e^{-\omega_p T_s}; e^{-\omega_p T_s}; 0.48; 0.60 + j0.23; 0.60 - j0.23]) \right)^T$$

$$L = L^*F = \begin{bmatrix} 0.1263 & 0.6314 \\ 32.5371 & 162.6854 \\ 0.0224 & 0.1119 \\ 46.3202 & 231.6000 \\ 0.2653 & 1.3266 \end{bmatrix}$$

8.4.2.2 Estimador a partir de un Filtro de Kalman

El filtro de Kalman es un algoritmo utilizado en procesamiento de señales y control de sistemas que proporciona estimaciones óptimas de variables de estado en un sistema dinámico con ruido que puede ser variable en el tiempo.

Recordando que el sistema en variable discreta, puede expresarse como:

$$\begin{aligned} x(k+1) &= G \cdot x(k) + H \cdot u(k) + v(k) \\ y(k) &= C \cdot x(k) + w(k) \end{aligned} \quad (63)$$

Siendo:

- $v(k)$: ruido del sistema
- $w(k)$: ruido de las medidas

Como consideración, el ruido debe ser ruido blanco, de media cero e independientes, es decir:

$$E\{x\} = \bar{x} = \sum_{i=1}^n x_i p(x_i) \quad \text{Esperanza Matemática (Valor Medio)}$$

$$\sigma(x, y) = E\{(x - \bar{x})(y - \bar{y})^T\} \quad \text{Covarianza (distancia al valor medio)}$$

El filtro de Kalman opera en dos etapas principales: *Predicción* y *Actualización*:

Etapa de predicción, se utiliza un modelo matemático del sistema para predecir el estado futuro del sistema basado en el estado actual y el control aplicado. Se actualiza la estimación del estado y la covarianza de la predicción.

Ec. de Predicción de Estado:
$$\hat{x}_{k|k-1} = G_{k-1}\hat{x}_{k-1|k-1} + H_{k-1}u_{k-1}$$

Ec. de Predicción de la Covarianza:
$$P_{k|k-1} = G_{k-1}P_{k-1|k-1}G_{k-1}^T + Q_{k-1}$$

En la etapa de actualización, se utiliza la medición actual del sistema para actualizar la estimación del estado. La diferencia entre la medición real y la predicción se utiliza para corregir la estimación.

Ganancia de Kalman
$$K_k = P_{k|k-1}C_k^T(C_kP_{k|k-1}C_k^T + R_k)^{-1}$$

Actualización del Estado
$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k(y_k - C_k\hat{x}_{k|k-1})$$

Actualización de la Covarianza
$$P_{k|k} = (\mathbb{I} - K_kC_k)P_{k|k-1}$$

Siendo:

- $\hat{x}_{k|k}$: Estimación estado en tiempo k después de incorporar la medida
- $P_{k|k}$: Covarianza de la estimación del estado.
- G_k : Matriz de transición de estado.
- H_k : Matriz de control.
- u_k : Vector de control.
- Q_k : Covarianza del ruido del proceso.
- C_k : Matriz de observación.
- R_k : Covarianza del ruido de la medida.
- y_k : Medida en el tiempo k

La matriz de covarianza del ruido del proceso se determinó midiendo la posición y corriente del sistema a lazo abierto y calculando la velocidad a partir de la derivada de la posición de forma numérica. Con estas tres mediciones se calculó la covarianza de cada estado obteniéndose.

$$Q = \begin{bmatrix} 0.01 & 0 & 0 & 0 & 0 \\ 0 & 0.01 & 0 & 0 & 0 \\ 0 & 0 & 0.0009 & -0.0028 & 0.0001 \\ 0 & 0 & -0.0028 & 361.839 & -0.366 \\ 0 & 0 & 0.0001 & -0.366 & 0.001 \end{bmatrix}$$

Para la medición de la posición (salida del sistema) se consideró una varianza (R) de:

$$R = 0.01$$

A partir de estos valores de ruido obtenidos se procedió a realizar el filtro de Kalman.

8.5 Control Adaptativo con Estimador Extendido para Rechazo de Perturbación Variable.

A partir del diseño del estimador extendido, se procedió a realizar un controlador con autoajuste de la frecuencia de perturbación de manera offline. Es decir, que si bien el controlador será capaz de rechazar perturbaciones armónicas de frecuencias conocidas (ω_p) que puedan modificar su valor, esta variación, para el análisis de este trabajo, no se realizará mientras el algoritmo de control esté en funcionamiento. Para implementar este controlador con autoajuste, se procedió a medir la tensión en bornes del motor vibrador, infiriendo de esta manera la velocidad de giro de este. A partir de este valor, se introdujo esta velocidad de giro al estimador extendido de manera de poder rechazar perturbaciones de frecuencia variable.

Siendo el sistema:

$$\begin{cases} \dot{x}(t) = A(w_p)x(t) + Bu(t) \\ y(t) = Cx(t) \end{cases}$$

Recordando la ec. (16), que describe al sistema por variable de estado, para el caso del levitador magnético:

$$\begin{bmatrix} \dot{\rho} \\ \ddot{\rho} \\ \dot{i} \\ \ddot{h} \\ \dot{h} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ -\omega_p^2 & 0 & 0 & 0 & 0 \\ \frac{1}{L} & 0 & -\frac{R}{L} & 0 & 0 \\ 0 & 0 & \frac{-2K_{mag}I_{eq}}{m h_{eq}^2} & 0 & \frac{-2K_{mag}I_{eq}^2}{m h_{eq}^3} \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \rho \\ \dot{\rho} \\ I \\ \dot{h} \\ h \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \frac{1}{L} \\ 0 \\ 0 \end{bmatrix} u$$

$$y = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \rho \\ \dot{\rho} \\ I \\ \dot{h} \\ h \end{bmatrix}$$

Debido a que la implementación del estimador se realizará de forma discreta es necesario representar al sistema en el plano discreto. Asumiendo que la acción de control pasa por un ZOH, esta representación está dada por:

$$\begin{cases} x(k+1) = G(w_p)x(k) + Hu(k) \\ y(k) = Cx(k) \end{cases}$$

Para la implementación de este estimador de perturbación variable, se discretizó al sistema en función de la frecuencia de perturbación w_p , a partir de las ecuaciones (39) y (40), obteniéndose:

$$G(w_p) = \begin{bmatrix} a_{11} & a_{12} & 0.000 & 0.000 & 0.000 \\ a_{21} & a_{22} & 0.000 & 0.000 & 0.000 \\ a_{31} & a_{32} & 0.496 & 0.000 & 0.000 \\ a_{41} & a_{42} & -56.65 & 1.010 & 10.47 \\ a_{51} & a_{52} & -0.063 & 0.002 & 1.010 \end{bmatrix}$$

$$H = \begin{bmatrix} 0.000 \\ 0.000 \\ 0.478 \\ -20.97 \\ -0.0147 \end{bmatrix}$$

Donde:

$$a_{11} = \cos(T_s w_p)$$

$$a_{12} = \frac{\sin(T_s w_p)}{w_p}$$

$$a_{21} = -w_p \sin(T_s w_p)$$

$$a_{22} = \cos(T_s w_p)$$

$$a_{31} = \frac{1.164e5 \cos(T_s w_p) + 332.5 w_p \sin(T_s w_p) - 57790}{(w_p^2 + 350.0^2)}$$

$$a_{32} = \frac{165.1 w_p + 1.164e5 \sin(T_s w_p) - 332.5 w_p \cos(T_s w_p)}{w_p (w_p^2 + 350.0^2)}$$

$$a_{41}$$

$$= \frac{-(4.56 \cdot 10^9 w_p \cdot \sin(T_s w_p) - 1.30 \cdot 10^7 w_p^2 \cdot \cos(T_s w_p) + 6.59 \cdot 10^6 w_p^2 + 1.34 \cdot 10^{10})}{w_p^4 + 1.28 \cdot 10^5 \cdot w_p^2 + 6.39 \cdot 10^8}$$

$$a_{42}$$

$$= \frac{(4.56 \cdot 10^9 \cos(T_s w_p) + 1.30 \cdot 10^7 \cdot w_p \cdot \sin(T_s w_p) - 1.88 \cdot 10^4 \cdot w_p^2 - 4.57 \cdot 10^9)}{w_p^4 + 1.28 \cdot 10^5 \cdot w_p^2 + 6.39 \cdot 10^8}$$

$$a_{51}$$

$$= \frac{(4.56 \cdot 10^9 \cdot \cos(T_s w_p) + 1.30 \cdot 10^7 \cdot w_p \cdot \sin(T_s w_p) - 1.88 \cdot 10^4 \cdot w_p^2 - 4.57 \cdot 10^9)}{w_p^4 + 1.28 \cdot 10^5 \cdot w_p^2 + 6.39 \cdot 10^8}$$

$$a_{52}$$

$$= \frac{(3.91 \cdot 10^6 \cdot w_p + 4.56 \cdot 10^9 \cdot \sin(T_s w_p) - 1.30 \cdot 10^7 \cdot w_p \cdot \cos(T_s w_p) - 20.97 \cdot w_p^3)}{w_p \cdot (w_p^4 + 1.28 \cdot 10^5 \cdot w_p^2 + 6.39 \cdot 10^8)}$$

A partir de esta matriz variable, se desarrolló un filtro de Kalman de manera de estimar la perturbación introducida a partir de la medición de la tensión aplicada al motor vibrador. Si bien el autoajuste de la frecuencia de

perturbación se realiza de manera offline, se implementó por completitud, un filtro en la medición de tensión del motor en caso de que esta varíe mientras el sistema este a lazo cerrado. Este filtro es de la forma:

$$H_{filter}(z) = \frac{\omega_p}{\omega_{medido}} = \frac{1 + z^{-1} + \dots + z^{-N}}{N} \quad (N = 200)$$

Este filtro, en caso de que la tensión del motor vibrador varíe, permitirá que el valor de ω_p modifique su valor de manera lenta con respecto a las dinámicas de la planta.

9 Resultados

“Una obra de arte nunca se termina, solo se abandona”

Leonardo da Vinci

En este capítulo se analizará individualmente y luego se comparará conjuntamente la performance y robustez de los distintos controladores diseñados anteriormente en función de atenuar la perturbación introducida por el motor desbalanceado al elemento levitado. Para este análisis, se calculará la densidad espectral de potencia de la señal Posición para distintas velocidades de giro del motor desbalanceado y se comparará la atenuación del pico en la frecuencia de giro. También se calculará, para los distintos controladores, la desviación estándar de la señal Posición en la frecuencia de diseño de la señal Perturbación de manera de cuantificar la dispersión de señal Posición con respecto al equilibrio. Por último, se analizará la energía contenida en la señal Posición a través del teorema de Parseval, cuantificando de esta manera la energía del elemento Levitado debido a la Perturbación introducida. Es esperable que un controlador ideal capaz de rechazar completamente una perturbación de origen vibratoria, presente una energía en el elemento levitado nula (es decir, elemento levitado totalmente quieto).

Para determinar el espectro de potencia de la señal Posición en los distintos controladores diseñados, se estabilizó el elemento levitado en la posición de equilibrio. Luego, se introdujo la señal Perturbación a una determinada frecuencia durante 90seg (45000 puntos). A partir de esta medición, se dividió la muestra obtenida en 9 segmentos de 5000 puntos (10seg) cada uno y se obtuvo la densidad espectral de potencia de cada segmento. Por último, se las promedió y se obtuvo la densidad de potencia mostrada en las figuras correspondientes.

Cabe recordar que la Perturbación se realiza a partir de un motor vibrador, el cual varía su velocidad a partir de la tensión aplicada en los bornes del

motor. Esta frecuencia de giro no está medida directamente. Si bien no se espera grandes variaciones de velocidad para una determinada tensión aplicada, no es posible asegurar una velocidad de giro repetible para las distintas mediciones y constante a lo largo de una determinada medición.

9.1 Control Clásico

Como primer análisis, y caso testigo, se realizó el análisis de performance al controlador clásico sometido a vibraciones de entre 22Hz y 53Hz obteniéndose el espectro que se muestra en la Figura 38.

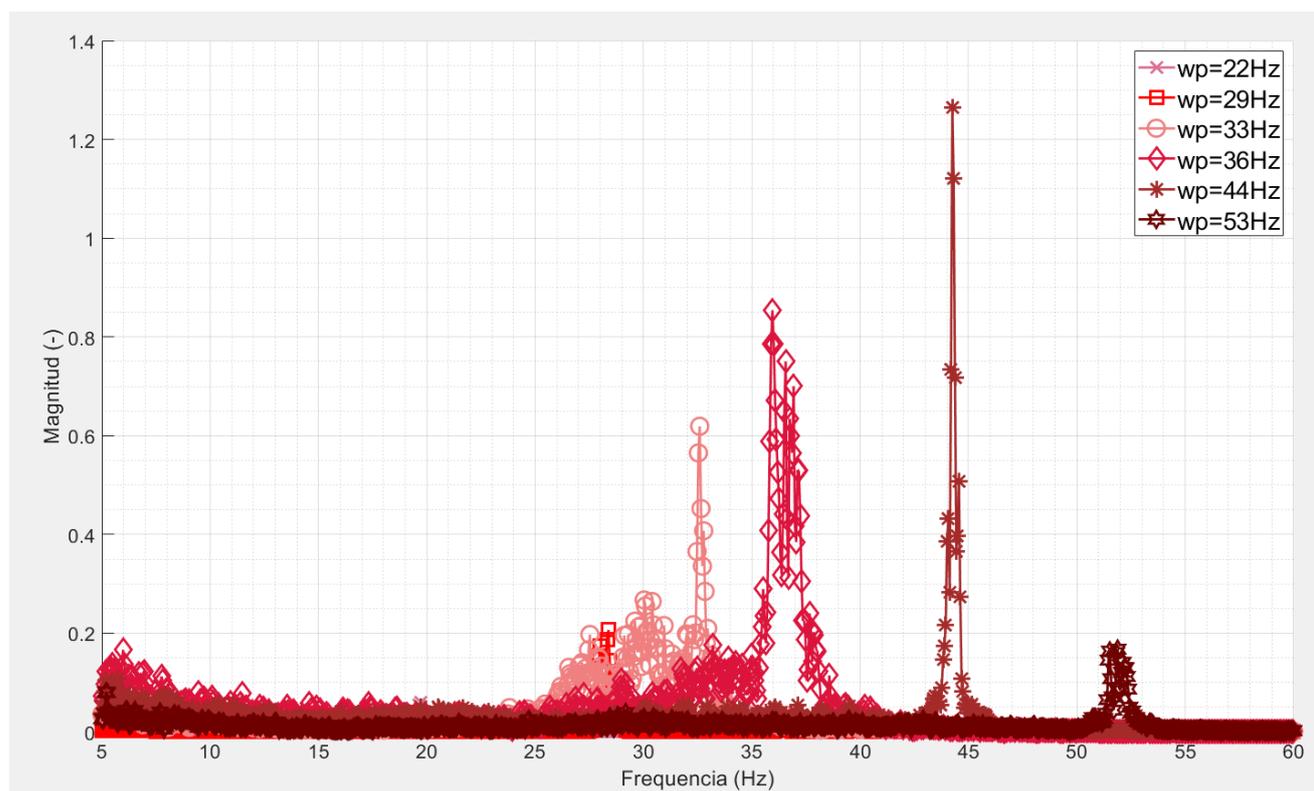


Figura 38: Espectro de Potencia de la señal de posición con Controlador Clásico.

Se observa que los picos del espectro de potencia coinciden con la velocidad de rotación del motor vibrador esperada según la calibración realizada con

anterioridad. Nótese que los picos poseen una magnitud que se encuentran principalmente entre 0.2 y 1.3, valores que nos ayudaran a comparar la performance de los controladores.

En la Figura 39 se muestra la señal de posición (con respecto al equilibrio) del elemento levitado, ante una perturbación de 33Hz.

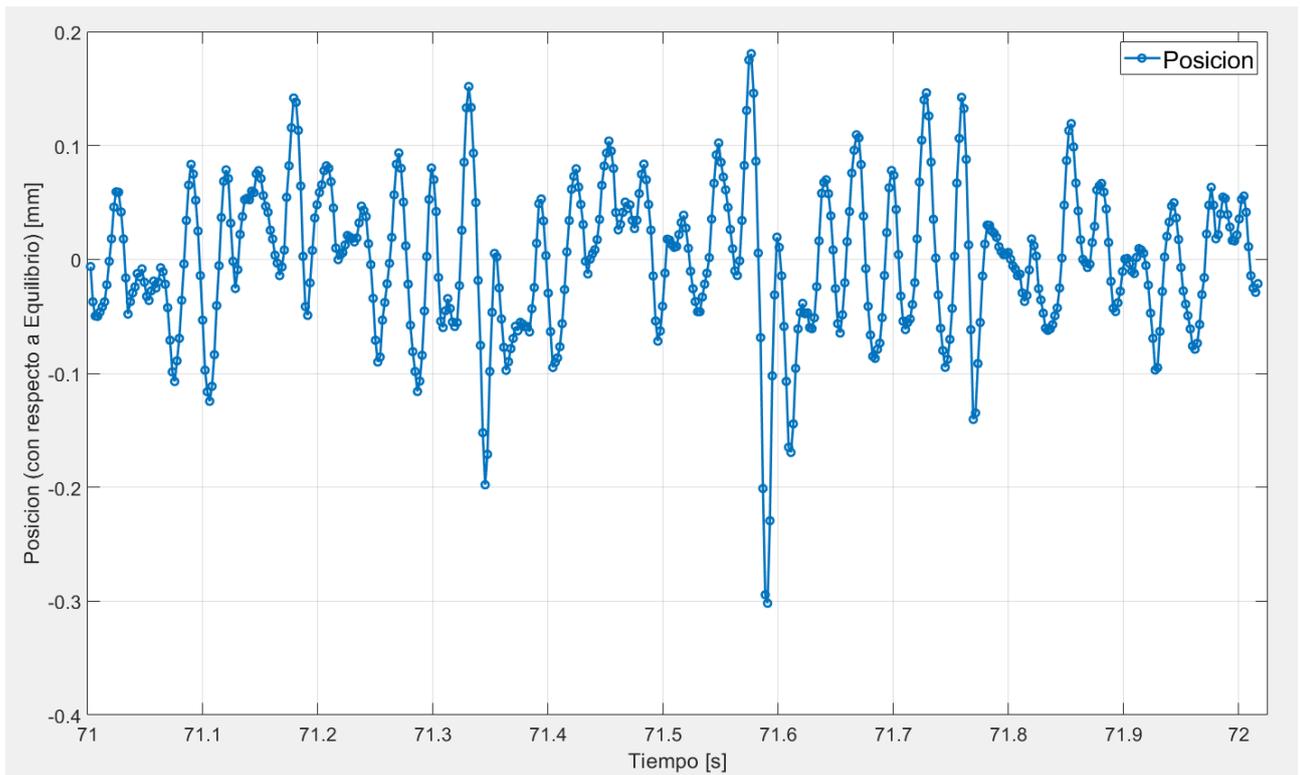


Figura 39: Señal de posición con controlador Clásico ante perturbación de 35Hz.

En la Figura 40 se muestra la acción de control correspondiente, la cual se encuentra en el rango $[-0.5 \ 0.5]$ no llegando a saturar en ningún momento, indicando que el controlador se encuentra operando en su rango lineal.

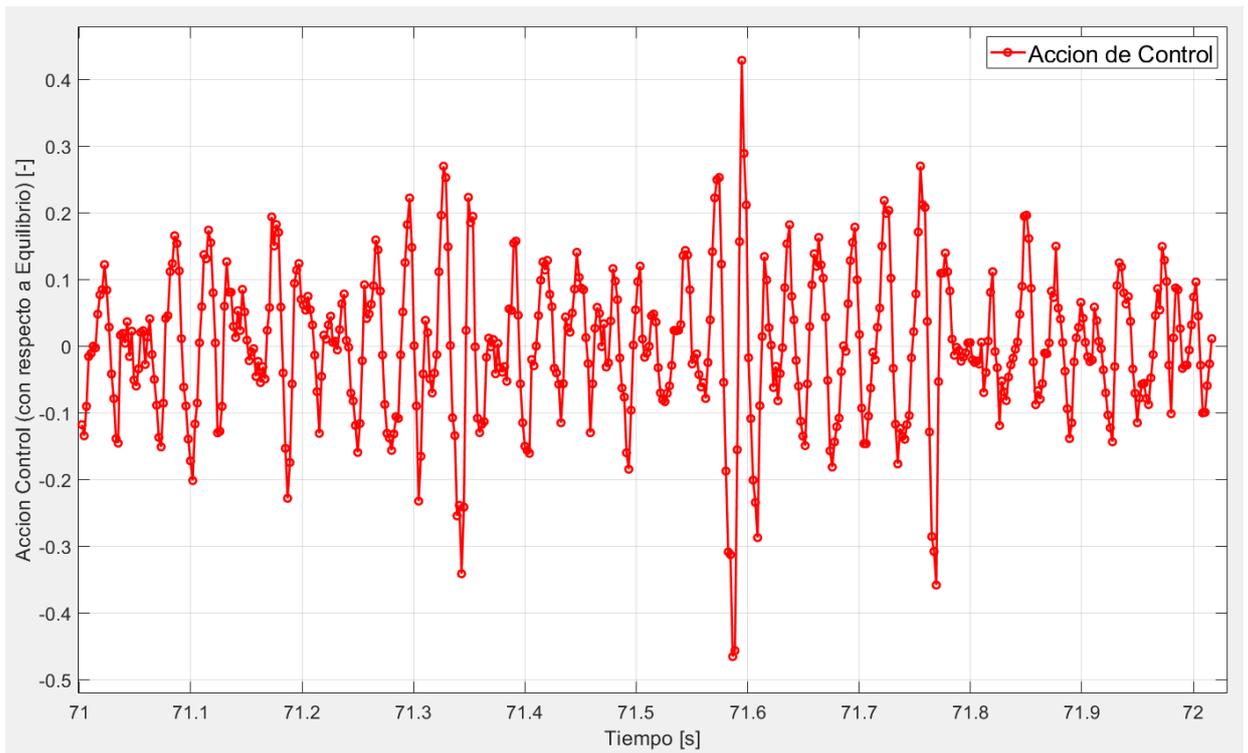


Figura 40: Acción de control con controlador Clásico ante perturbación de 35Hz.

9.2 Controlador Repetitivo

Como segundo caso se realiza el análisis de performance del controlador repetitivo diseñado para rechazar perturbaciones de 33Hz de frecuencia. Al igual que en el Control Clásico se varió la frecuencia de perturbación entre 22Hz y 53Hz. En la Figura 41 se observa el espectro de frecuencias medido.

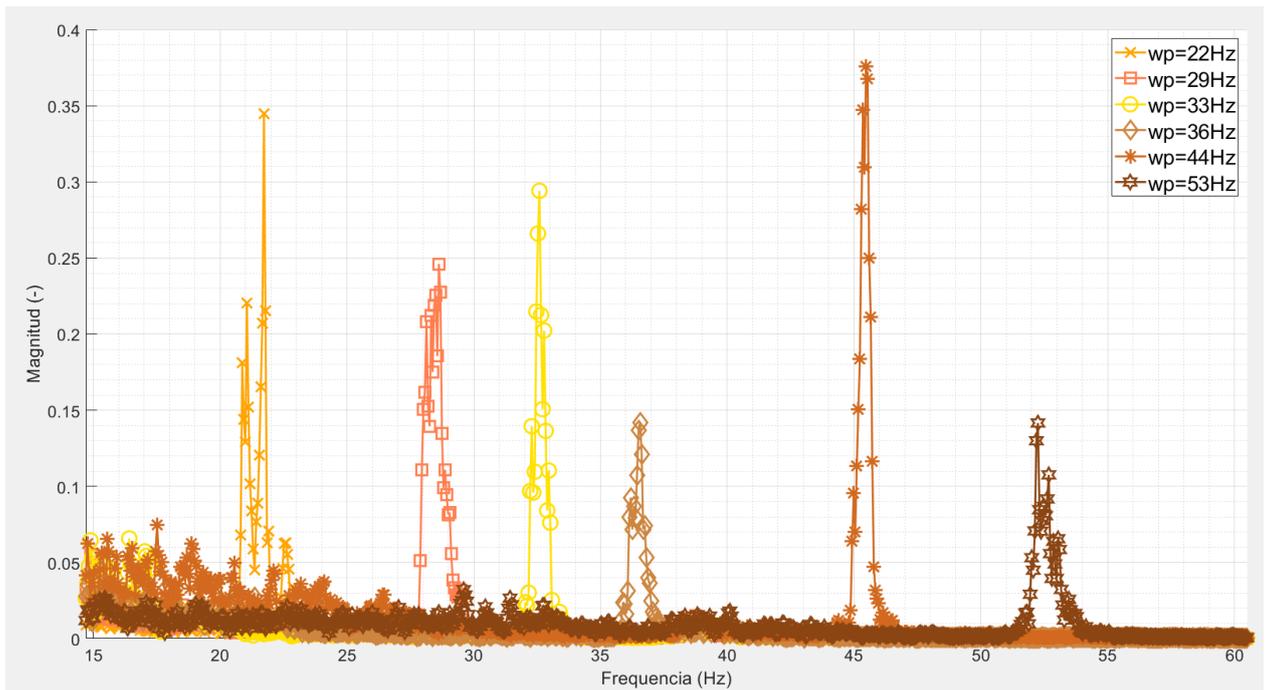


Figura 41: Comparación de performance ante variación en la frecuencia de perturbación del controlador repetitivo.

En esta figura se observa una atenuación de los componentes principales de frecuencia con respecto al Control Clásico para todo el rango de frecuencias estudiado, logrando para la frecuencia de diseño una atenuación en el componente principal de frecuencia (33Hz) de casi el 54%.

En la Figura 42 se observa la señal de posición medida con respecto al equilibrio, para la perturbación de frecuencia de diseño (33Hz). En ella se puede apreciar una cierta periodicidad aproximada a la de la perturbación.

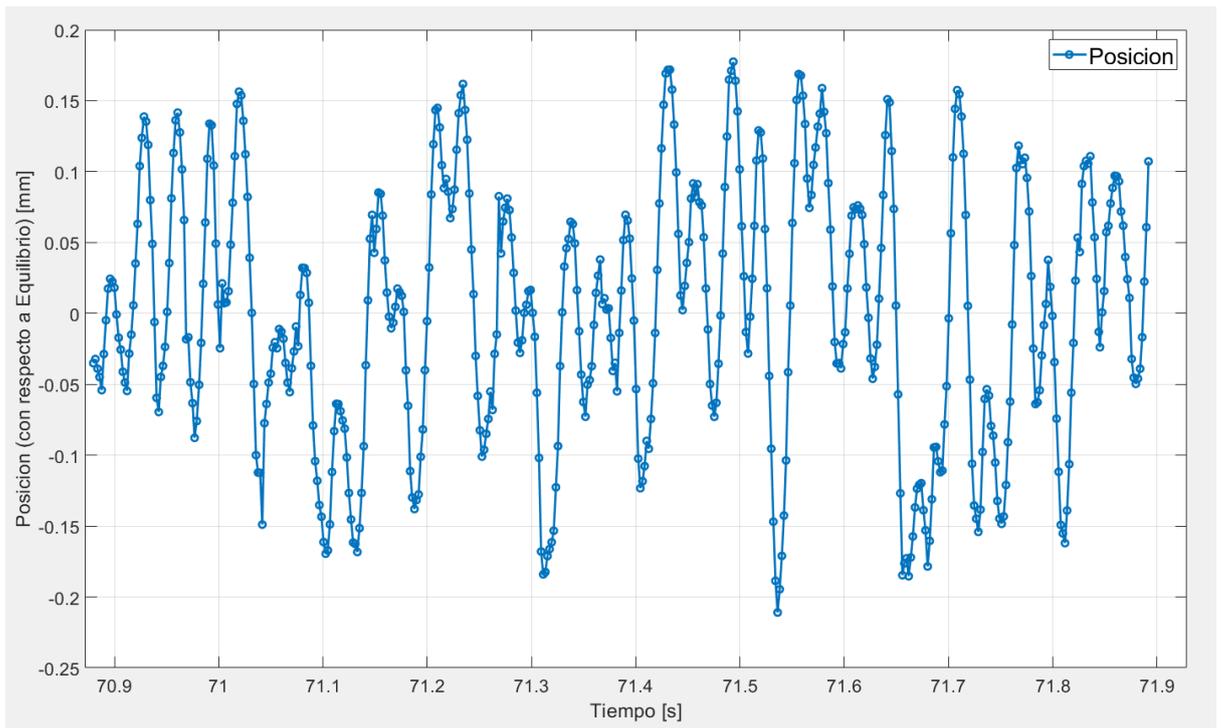


Figura 42: Señal de posición con controlador Repetitivo ante perturbación de 35Hz.

En la Figura 43 se muestra la Acción de Control del controlador Repetitivo, el cual se encuentra dentro del rango lineal de funcionamiento.

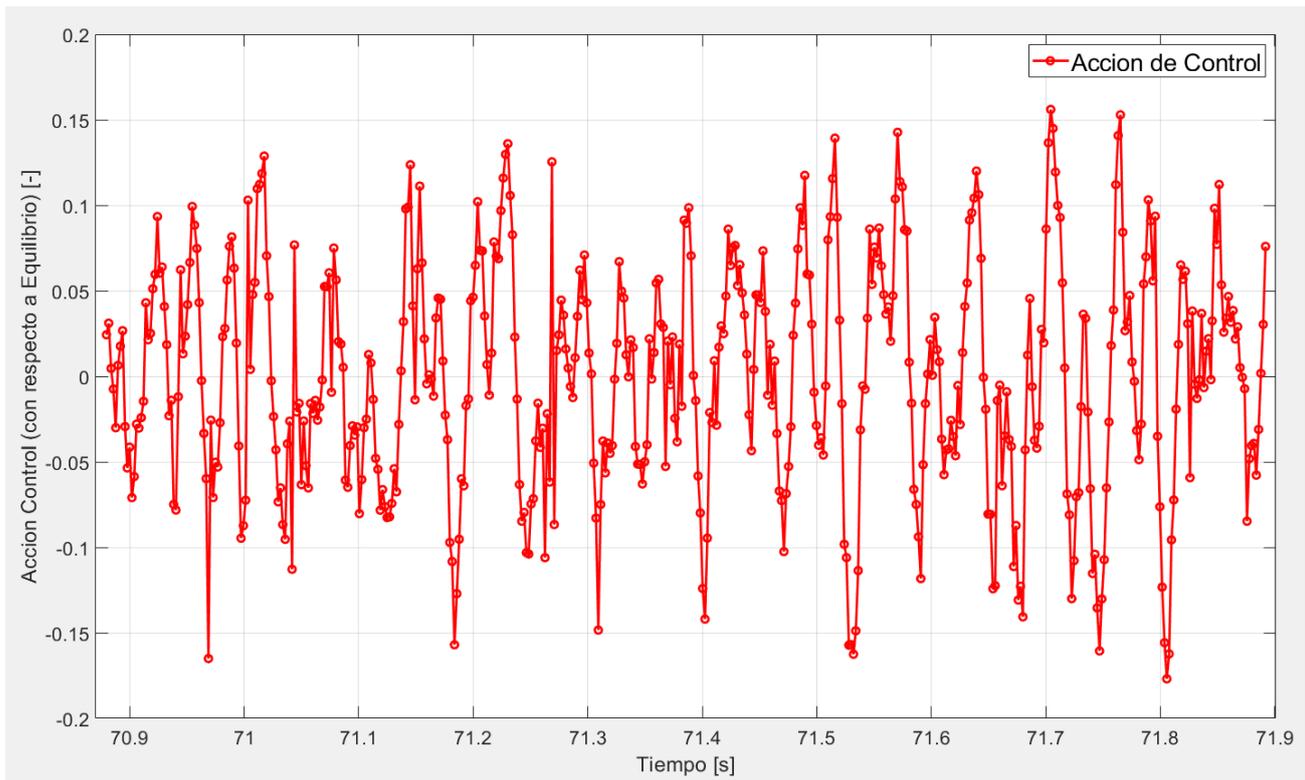


Figura 43: Acción de control correspondiente al Controlador Repetitivo

En la Figura 44 se observa el error de posición (azul) y la salida del bloque repetitivo (rojo). La suma de ambas señales será la entrada al controlador de posición implementado.

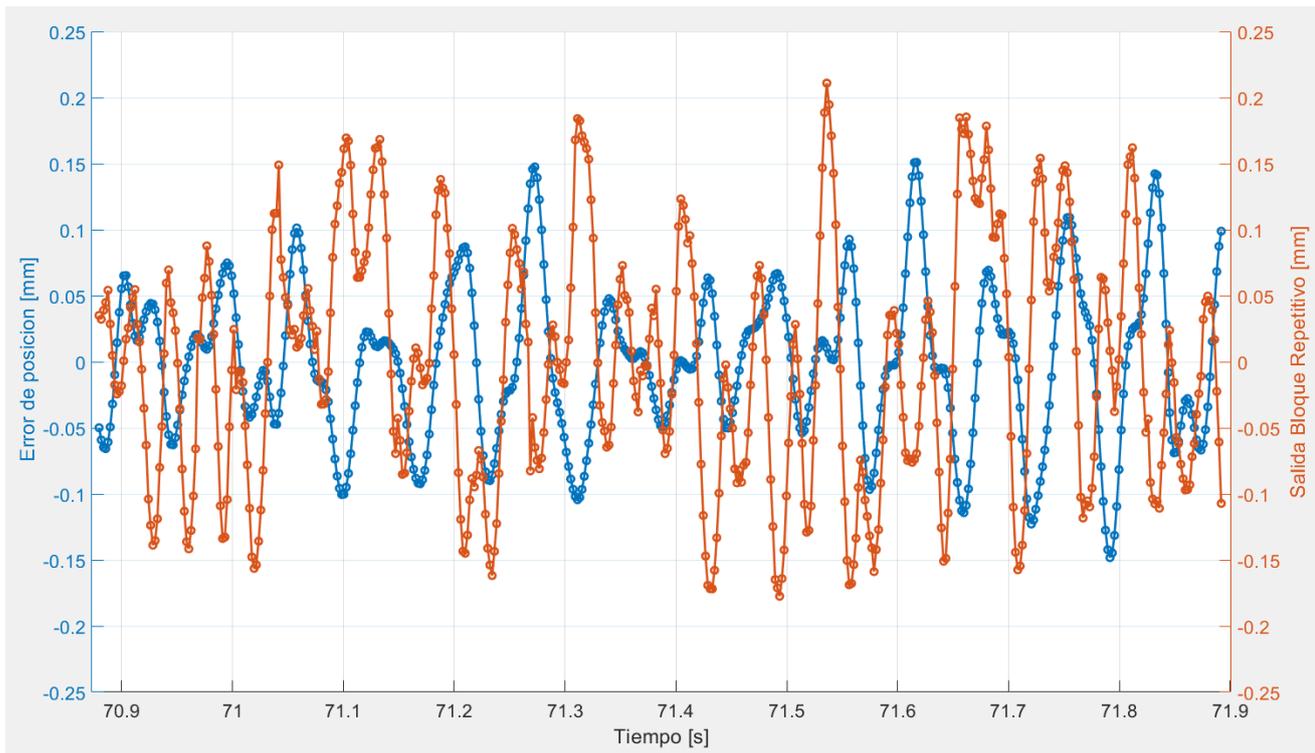


Figura 44: Error de Posición y Salida del bloque Repetitivo.

9.3 Control Moderno con Estimador MIMO (Modelo de Perturbación Fija)

A partir de la implementación del controlador con estimador MIMO, se realiza el análisis de robustez. En la Figura 45 se muestra el espectro de potencia de la señal de posición para frecuencias de perturbación entre 21Hz y 53Hz.

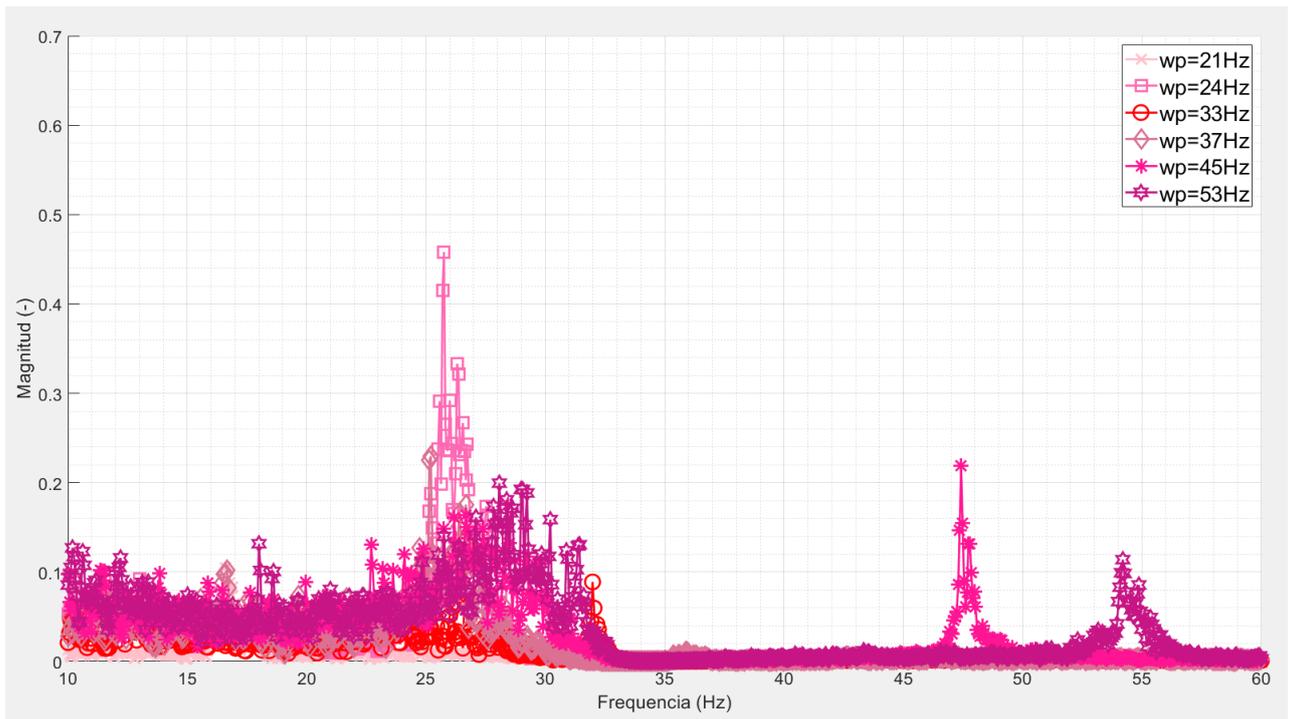


Figura 45: Espectro de Potencia para Controlador con Estimador MIMO de la señal de posición.

En el espectro de potencia de la figura anterior se observa una buena performance para atenuar frecuencias de 33Hz (frecuencia de diseño) y 37Hz (1.12 veces frecuencia de diseño). Puede observarse como contrapartida una amplificación de componentes de baja frecuencia. También puede observarse que las magnitudes máximas de los picos de frecuencia obtenidos están en el rango de 0.1 a 0.45 aproximadamente.

A continuación, se muestran las señales de interés del sistema en el dominio temporal, ante una perturbación de 33Hz. En la Figura 46 y Figura 47 se muestra la posición del elemento levitado y la corriente medida (con respecto al equilibrio) junto con su respectiva estimación. Tanto la estimación de posición como la de corriente convergen satisfactoriamente a la variable medida.

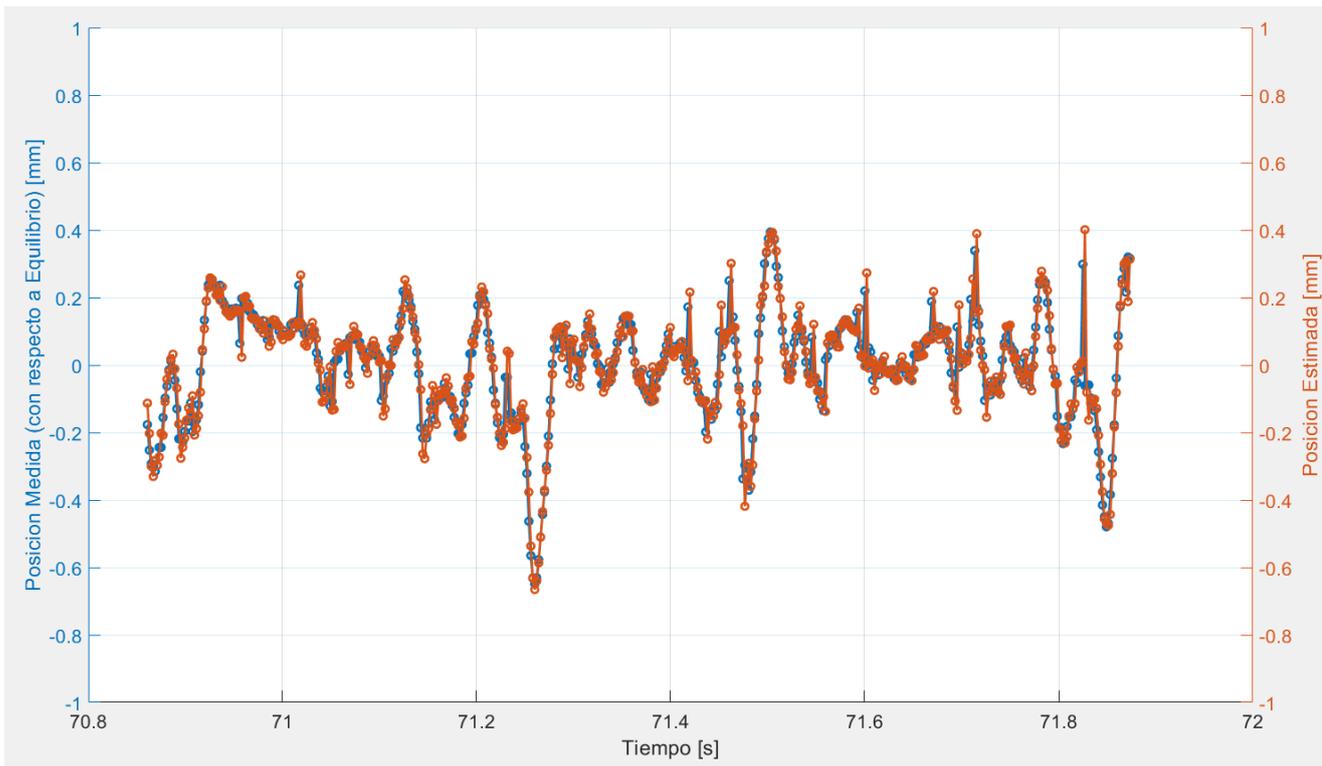


Figura 46: Huelgo medido y estimado para un controlador con estimación MIMO

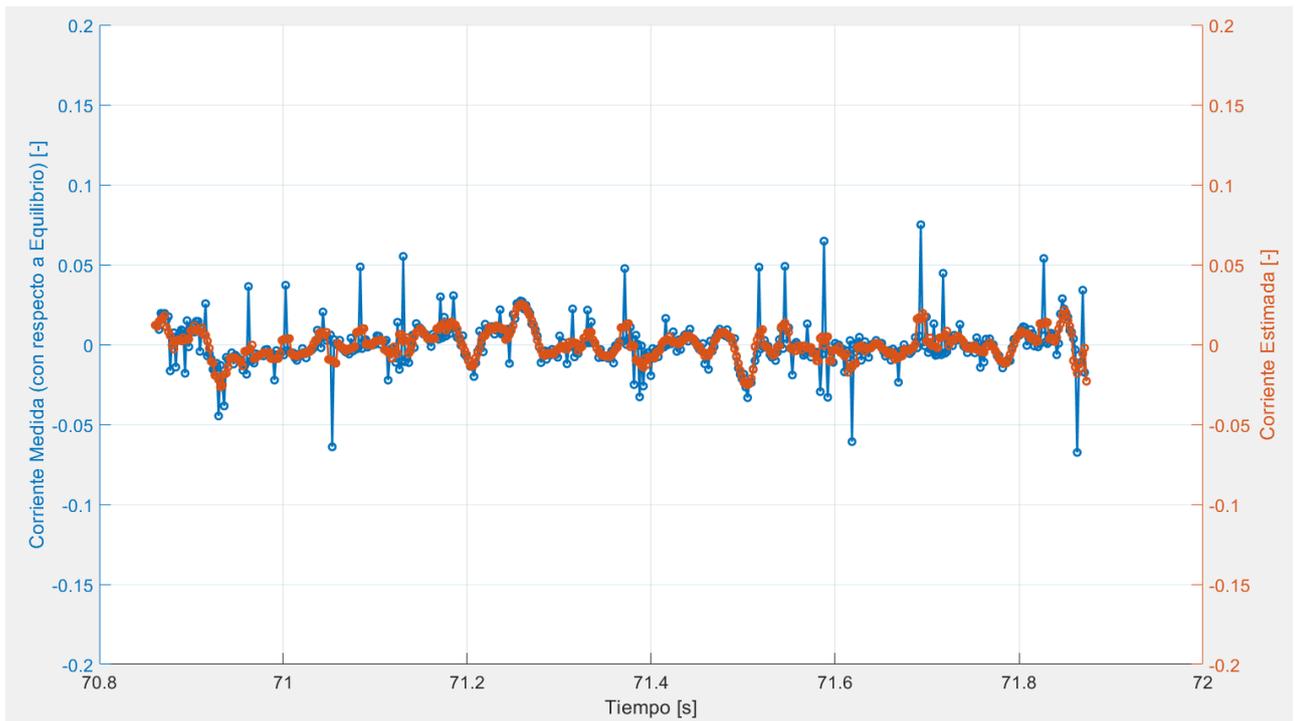


Figura 47: Corriente medida y Estimada para un controlador con estimación MIMO

En la Figura 48 se muestra la acción de control obtenida para el sistema de control con Estimador MIMO, el cual se encuentra dentro del rango lineal de funcionamiento.

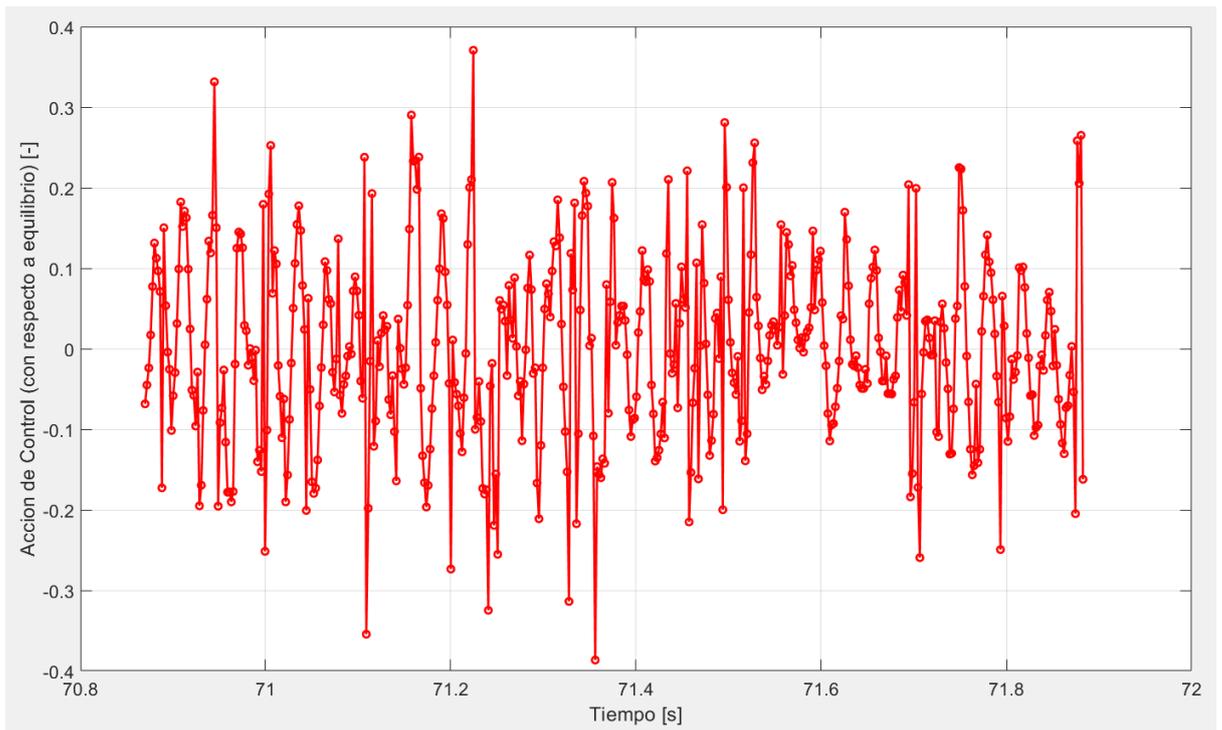


Figura 48: Acción de Control para Estimador MIMO

En la Figura 49 se muestra la estimación de la perturbación, la cual posee una frecuencia de oscilación de 33Hz (aproximadamente), coincidiendo con la frecuencia de perturbación esperada.

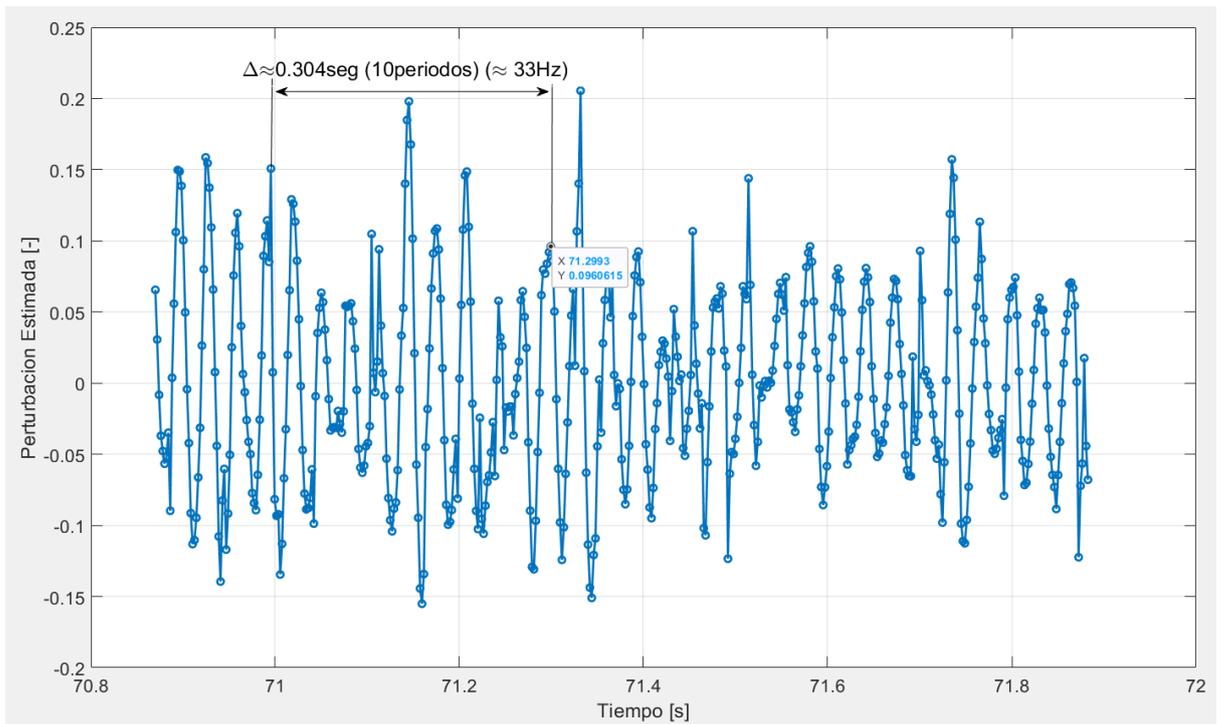


Figura 49: Perturbación Estimada para el controlador con Estimador MIMO.

9.4 Control Moderno con Filtro de Kalman (Modelo de Perturbación Fija)

Se realiza el análisis de robustez del sistema a partir del Control Moderno con filtro de Kalman (Modelo de Perturbación Fija). Se varia, nuevamente, la frecuencia de la perturbación entre 22Hz y 53Hz y se realiza el espectro de potencia de la señal de posición (Figura 50).

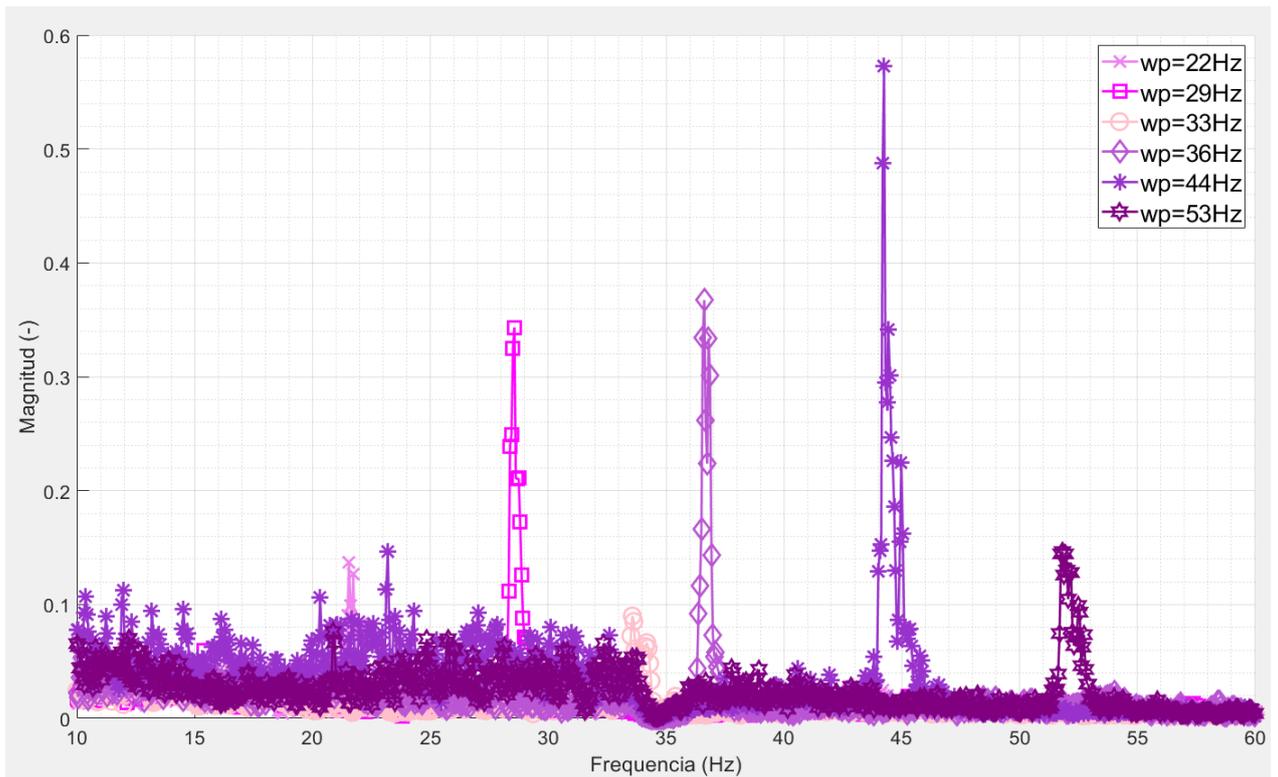


Figura 50: Espectro de Potencia de la señal de Posición con Filtro de Kalman Fijo

En el espectro de potencia de la figura anterior se observa una disminución en la magnitud de los picos de frecuencia con respecto al Control Clásico. Al igual que en el Estimador MIMO, se produce una gran atenuación en la frecuencia de diseño (33Hz), aunque el ancho de la frecuencia atenuada es más angosto, volviendo al filtro más selectivo.

En la Figura 51 y Figura 52 se muestra la posición y corriente medida y estimada del elemento levitado ante una perturbación en la frecuencia de diseño, respectivamente. Para el caso de la posición, el filtro estima de manera satisfactoria la posición medida, no así para el caso de la corriente. Cabe mencionar que el filtro de Kalman realiza sus estimaciones a partir de la medición de posición, por lo que, si el modelo del bloque corriente no es del todo satisfactorio, esto puede ser motivo de diferencias entre el valor medido y el estimado.

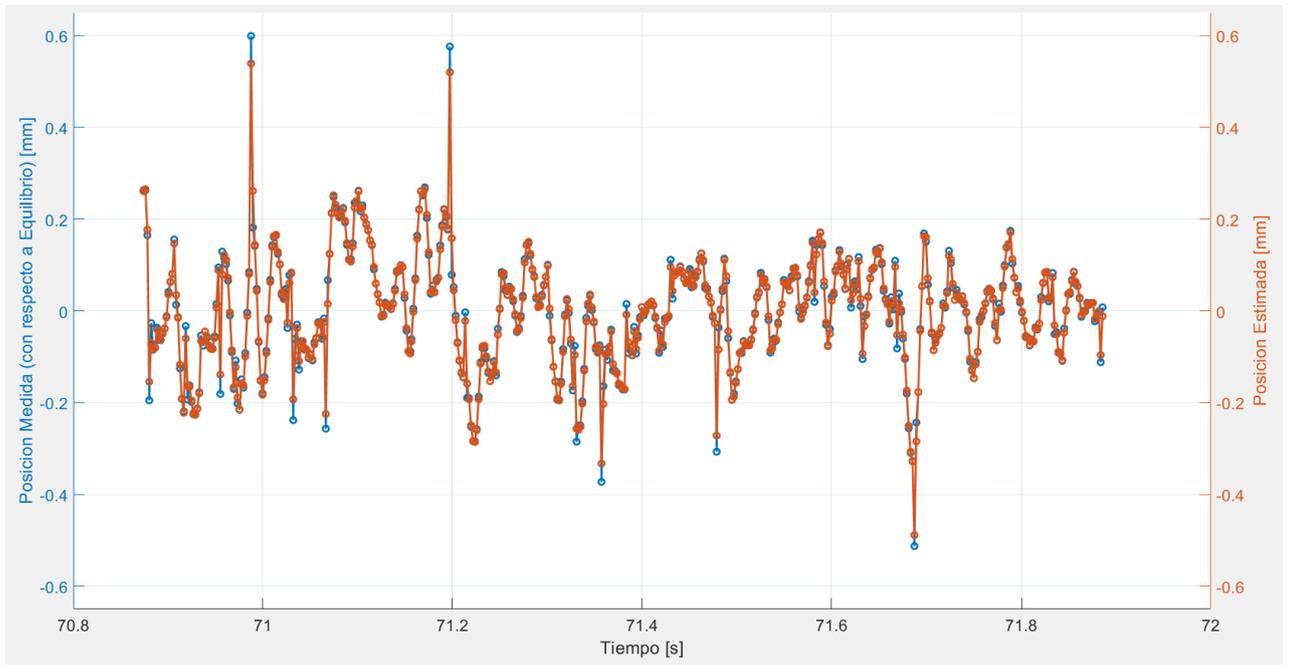


Figura 51: Posición medida y estimada para Controlador Moderno con Filtro de Kalman (Modelo de Perturbación Fija).

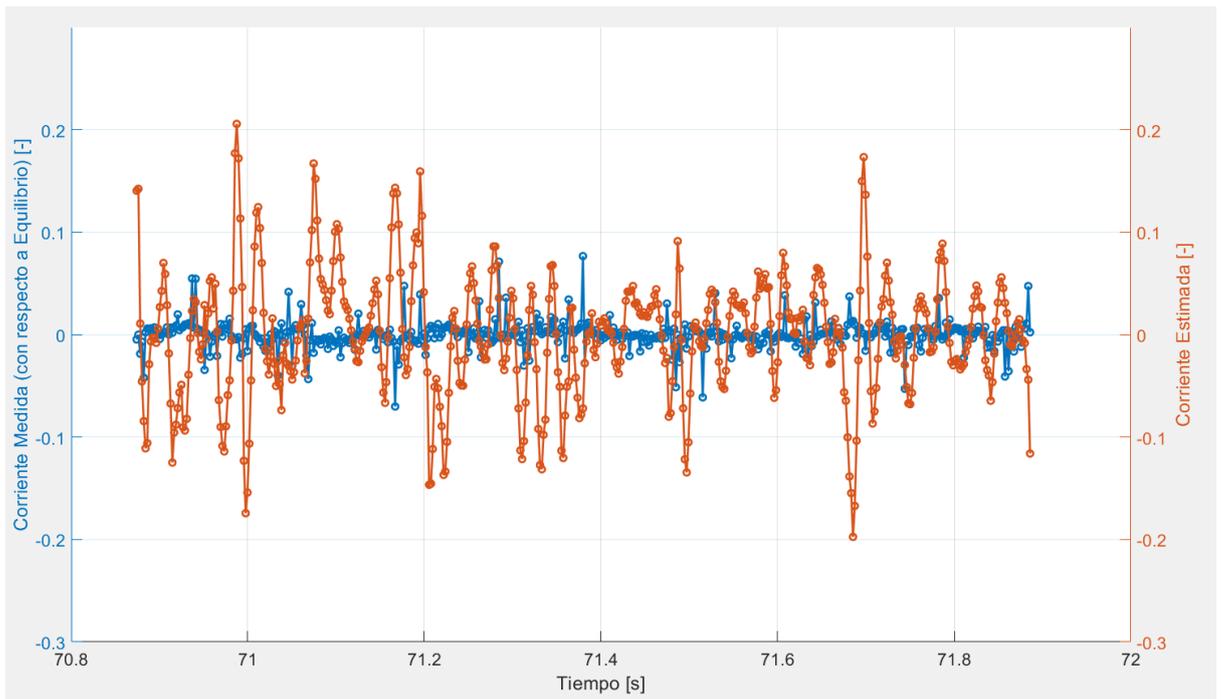


Figura 52: Corriente medida y estimada para Controlador Moderno con Filtro de Kalman (Modelo de Perturbación Fija)

En la Figura 53 se muestra la acción de control, la cual llega a saturación en ciertas ocasiones agregándole alinealidades al sistema.

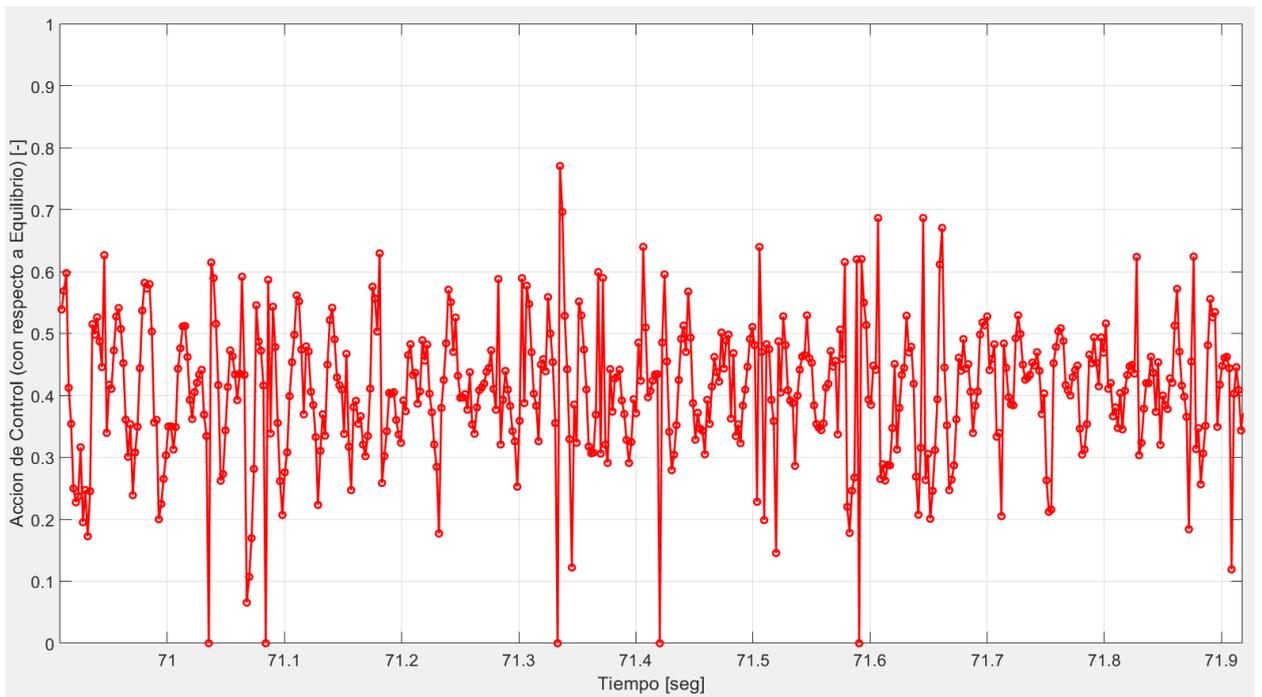


Figura 53: Acción de Control de Controlador Moderno con Filtro de Kalman (Modelo de Perturbación Fija)

En la Figura 54 se puede observar que el componente de perturbación estimada es una señal periódica con frecuencia característica similar al modelo de perturbación.

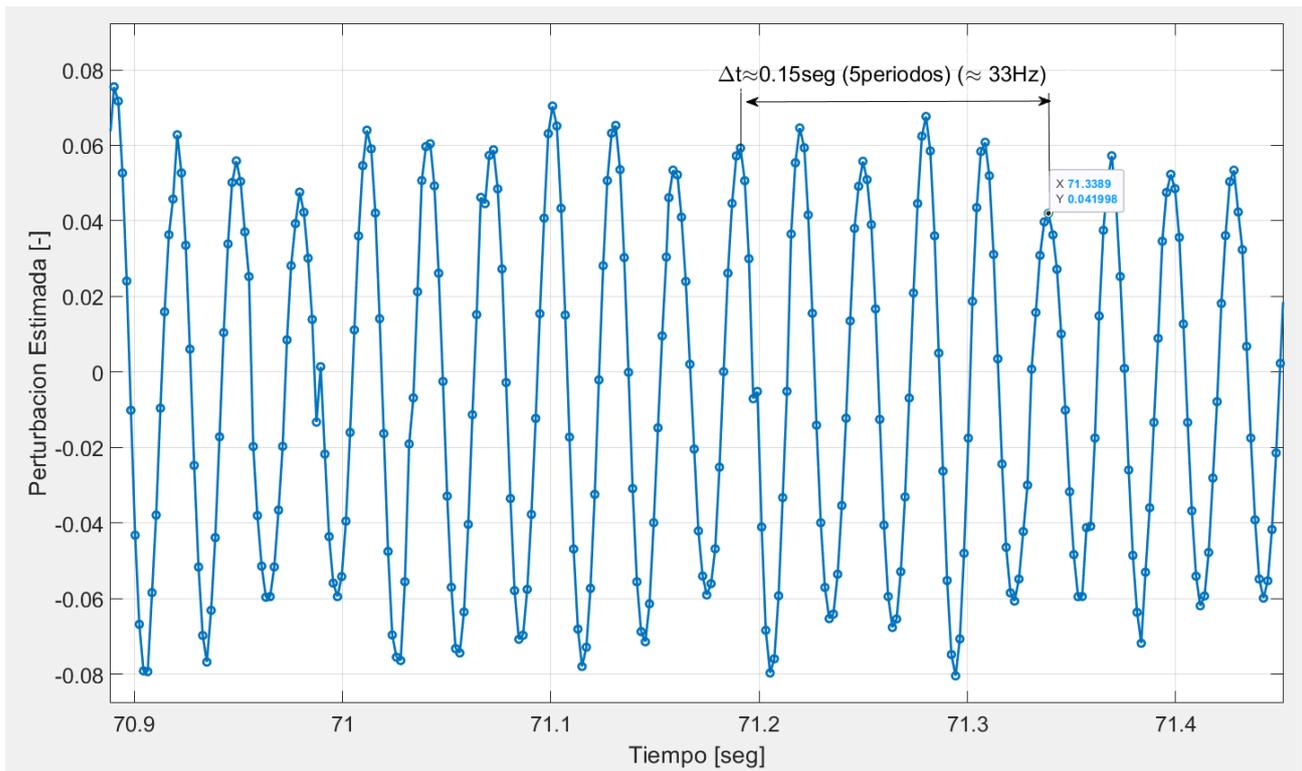


Figura 54: Perturbación estimada para Controlador Moderno con Filtro de Kalman (Modelo de Perturbación Fija).

9.5 Control Moderno con Filtro de Kalman (Modelo Perturbación Variable)

Por último, se analiza la performance del Controlador Moderno con Filtro de Kalman (Modelo de Perturbación Variable). En la Figura 55 se muestra el espectro de potencia obtenido para distintas frecuencias de perturbación. Puede observarse la atenuación en las distintas frecuencias de diseño, aunque de manera muy selectiva. Debido a la incertidumbre en la medición indirecta de la frecuencia de perturbación, esto podría ser perjudicial para la implementación del Modelo de Perturbación Variable. No obstante, los picos principales en el espectro de potencia se ven atenuados con respecto al

Control Clásico. También se observa una amplificación de los componentes de baja frecuencia, disminuyendo la performance del controlador.

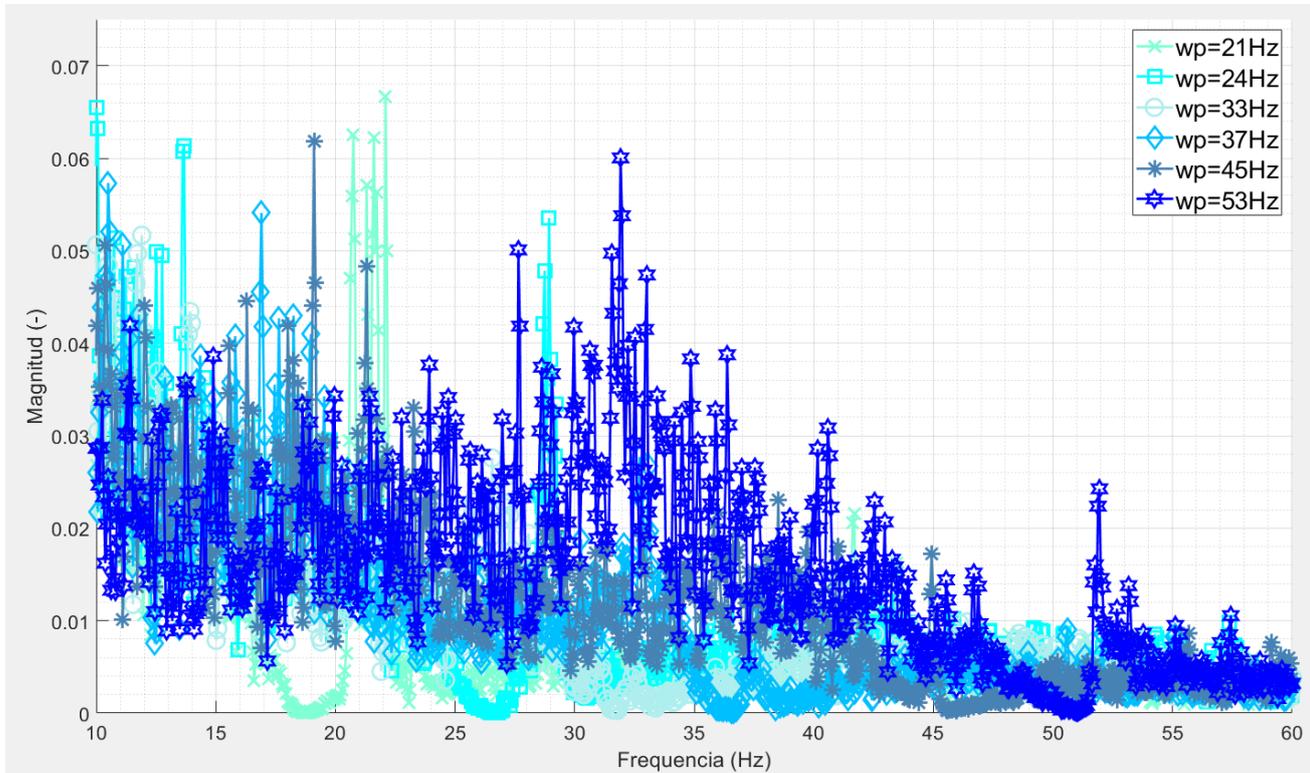


Figura 55: Espectro de Potencia de la señal de Posición del controlador Moderno con Filtro de Kalman (Modelo de Perturbación Variable)

En la Figura 56 y Figura 57 se observa posición y corriente, medida y estimada del elemento levitado ante una perturbación con frecuencia 33Hz, respectivamente. Al igual que en caso anterior, para el caso de la posición, el filtro estima de manera satisfactoria la posición medida, no así para el caso de la corriente.

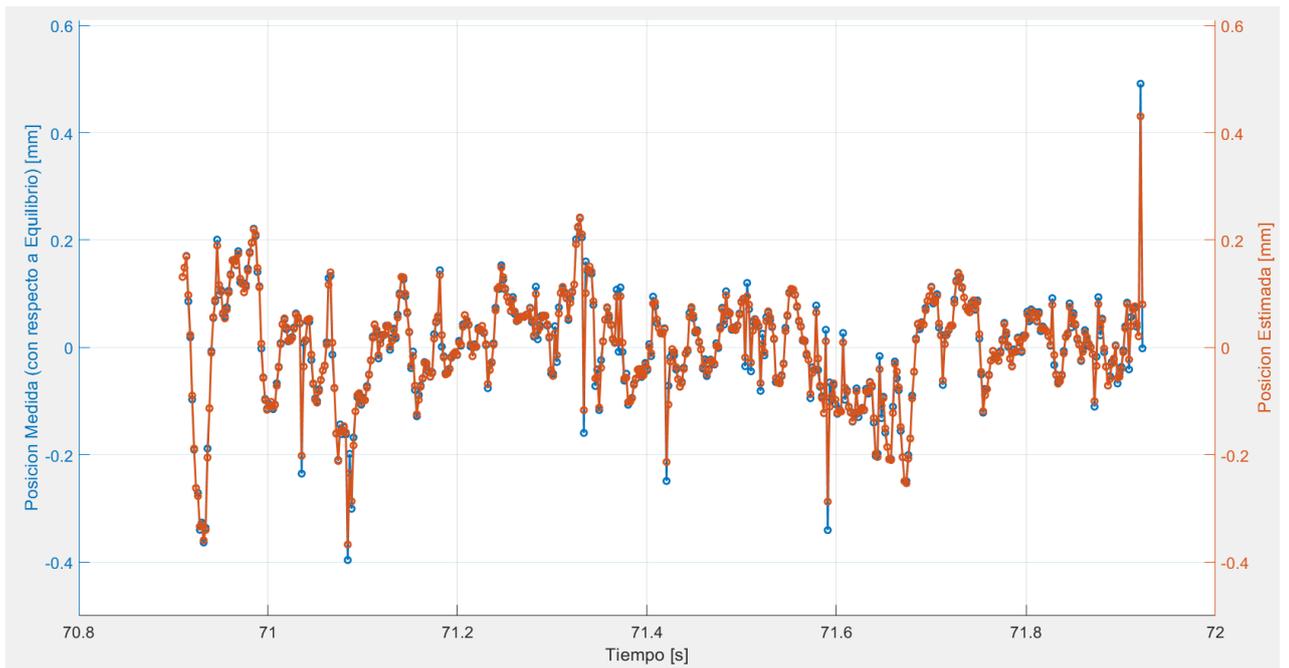


Figura 56: Señal de posición medida y estimada para controlador Moderno con Filtro de Kalman (Modelo de Perturbación Variable)

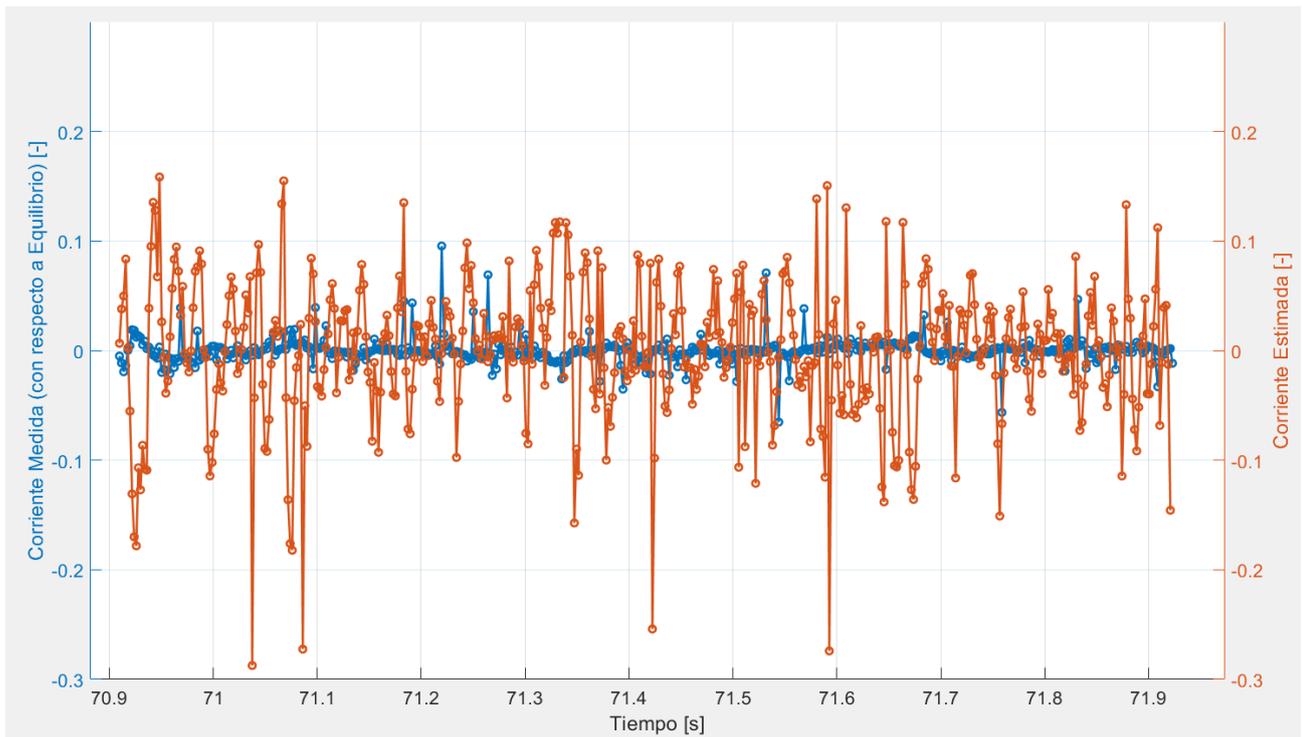


Figura 57: Señal de corriente medida y estimada para para controlador Moderno con Filtro de Kalman (Modelo de Perturbación Variable)

En la Figura 58 se muestra la acción de control obtenida, la cual, al igual que en el caso anterior, llega a saturación en ciertas ocasiones agregándole alinealidades al sistema.

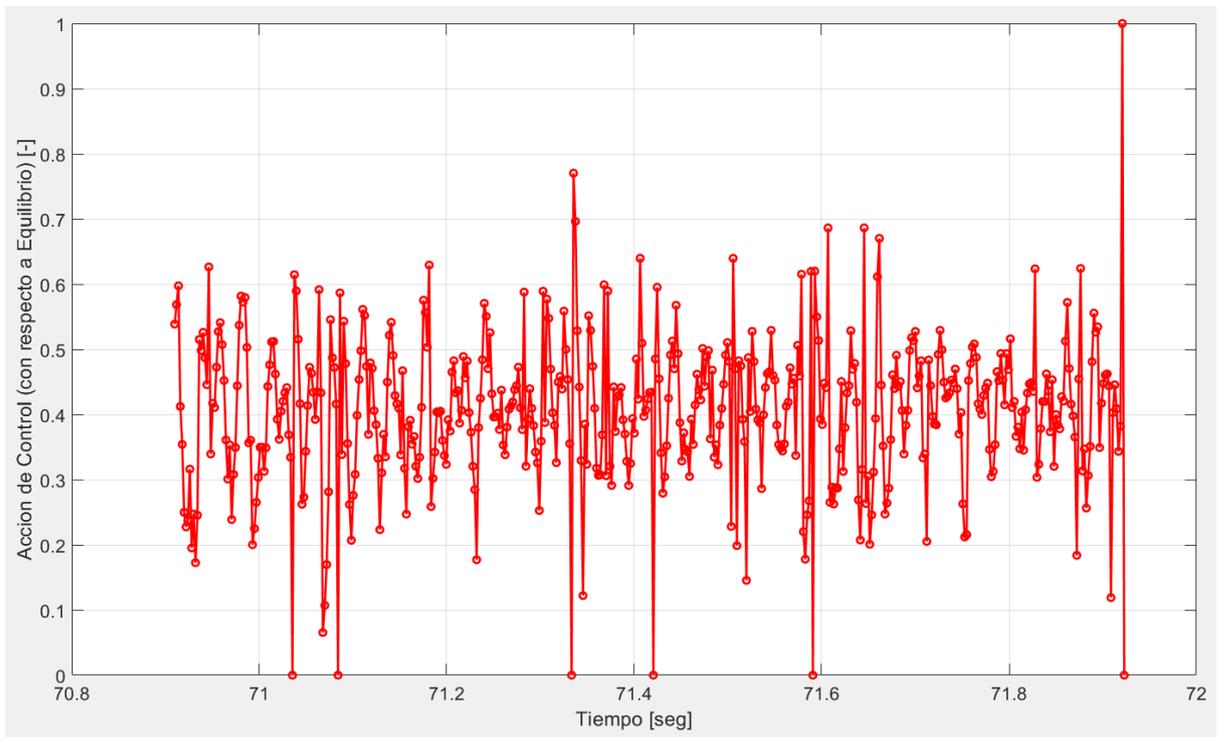


Figura 58: Acción de Control del Controlador Moderno con Filtro de Kalman (Modelo de Perturbación Variable)

En la Figura 59 se observa que la perturbación estimada es una señal periódica con frecuencia característica similar al modelo de perturbación.

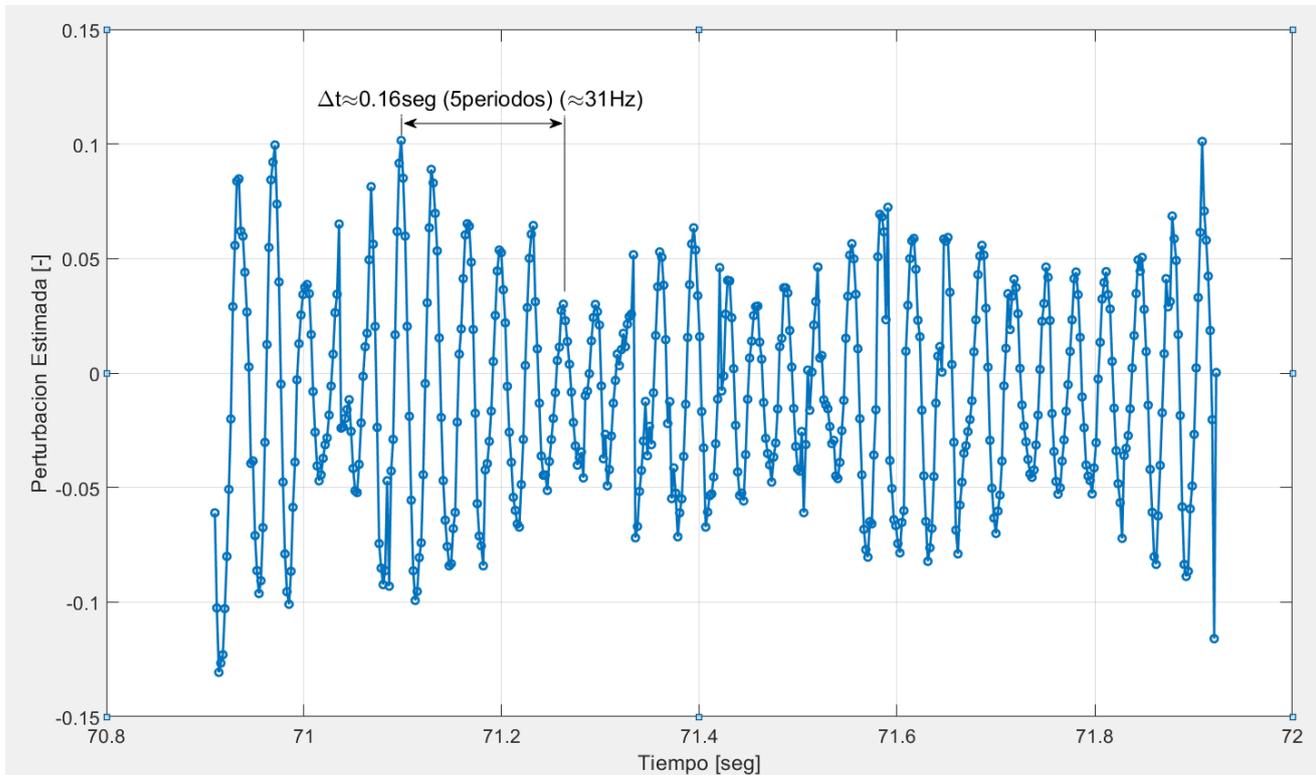


Figura 59: Perturbación Estimada del Controlador Moderno con Filtro de Kalman (Modelo de Perturbación Variable)

9.6 Comparación de Performance de los distintos controladores diseñados

Para comparar la performance y robustez de los controladores antes diseñados, se muestra la densidad espectral de potencia (Figura 60) obtenida con una perturbación de frecuencia de 33Hz (frecuencia de diseño).

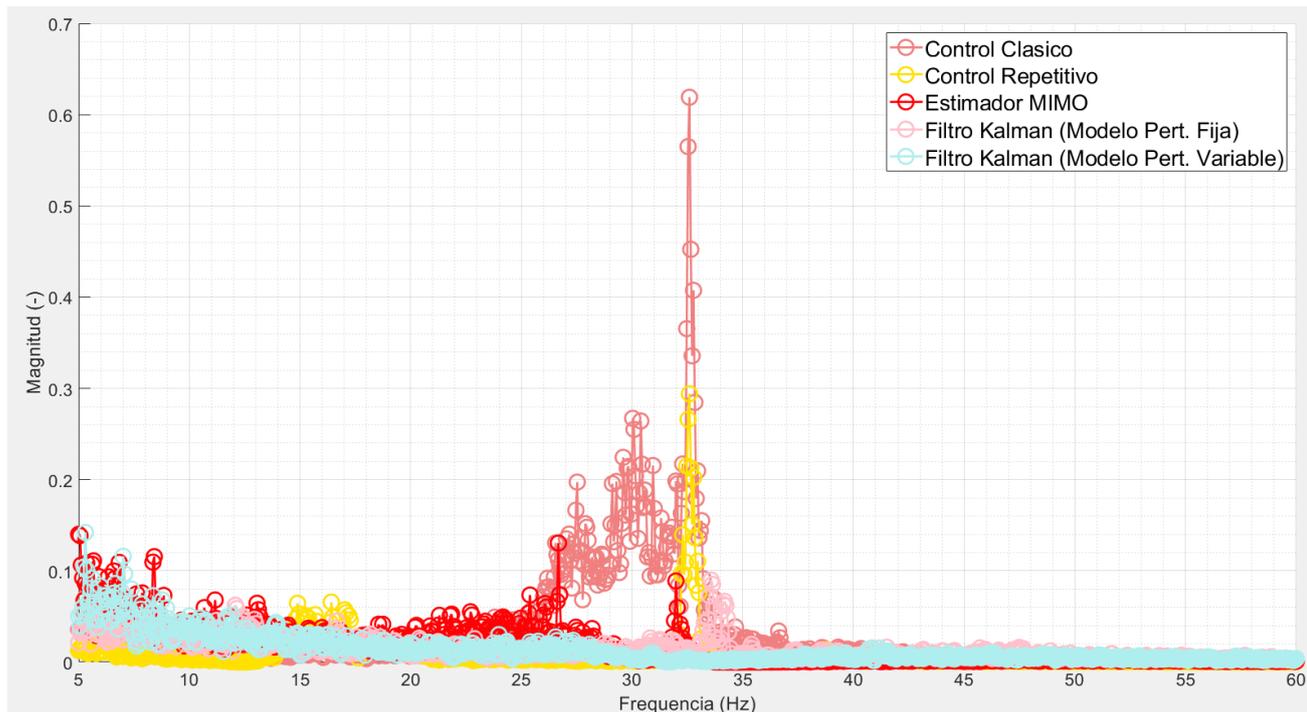


Figura 60: Densidad espectral de potencia de los distintos controladores para una perturbación de 33Hz (frecuencia de diseño)

En la figura anterior se observa como el Controlador Clásico presenta la menor atenuación para la perturbación antes mencionada. A su vez, el controlador Repetitivo presenta atenuación en la frecuencia de la perturbación de diseño manteniendo un nivel de amplificación aceptable para bajas frecuencias. El Controlador Moderno con Estimador MIMO posee una mejor capacidad de atenuación de la perturbación en la frecuencia de diseño con respecto a los anteriores, aunque presenta amplificación de la señal a baja frecuencia. El Control Moderno con Filtro de Kalman (Modelo de Perturbación Fija), también presenta una gran atenuación en la frecuencia de diseño de la perturbación con una menor amplificación de los componentes de baja frecuencia con respecto al controlador anterior. Por último, en el caso del Control Moderno con Filtro de Kalman (Modelo de Perturbación Variable) se observa como el pico en la frecuencia de perturbación se encuentra muy atenuado,

presentando casi un espectro de potencia plano, con una cierta amplificación a baja frecuencia.

Otra manera de estudiar la performance de los distintos controladores, como se mencionó al comienzo del capítulo, es a través del desvío estándar de la señal de posición ante la perturbación de diseño (33Hz). En la Tabla 3 se compara los valores calculados y la atenuación resultante con respecto al Control Clásico.

Tipo Controlador	Desvío Estándar [mm]	Atenuación (respecto a Clásico)
Clásico	0,27	----
Repetitivo	0,09	0,67
MIMO	0,17	0,37
Kalman Fijo	0,13	0,52
Kalman Variable	0,14	0,48

Tabla 3: Comparación de los distintos controladores en la atenuación del desvío estándar para la señal de posición.

Se observa como en todos los casos el desvío estándar de la señal de posición se logra atenuar con respecto al Control Clásico, representando que el valor de la señal Posición posee menos fluctuaciones con respecto al equilibrio.

En la Figura 61 se muestra la energía contenida en la señal de posición para las distintas frecuencias de perturbación con los distintos controladores implementados. Este análisis se realizó a partir de calcular la energía contenida en los espectros de potencia obtenidos experimentalmente.

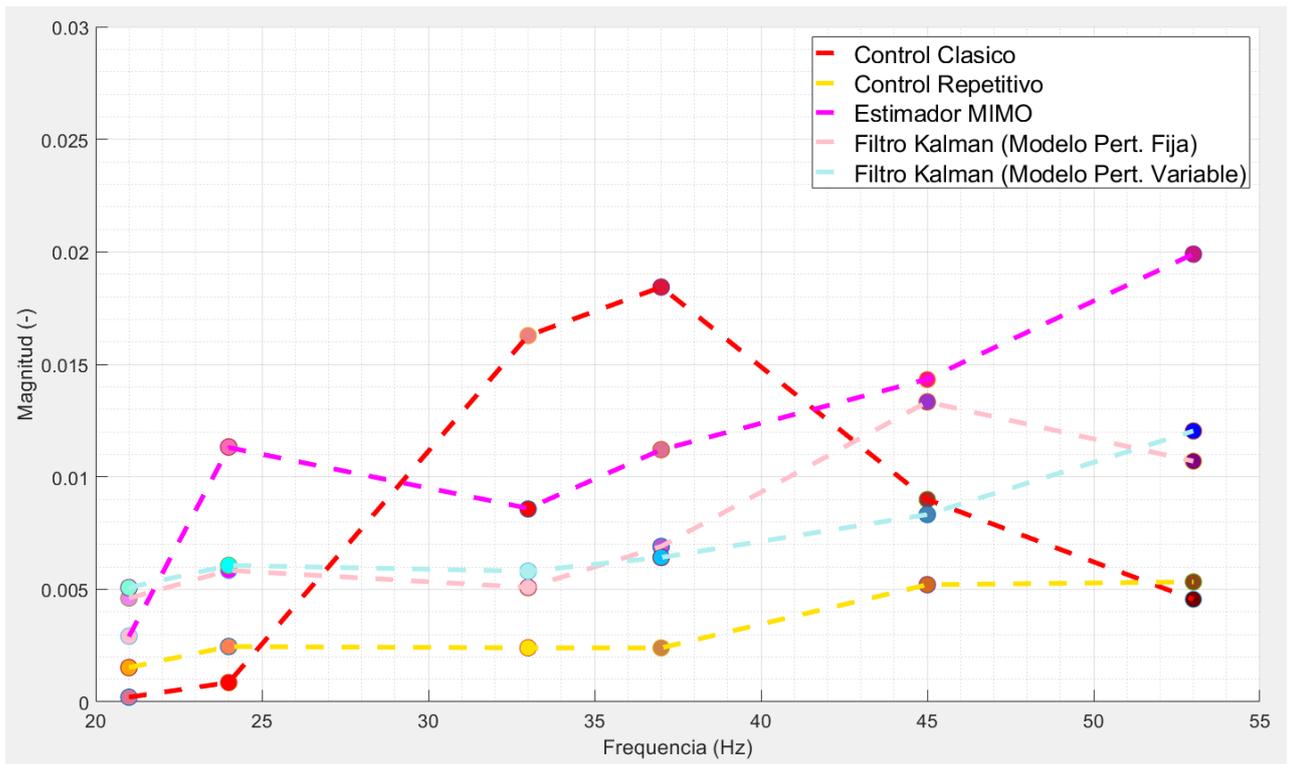


Figura 61: Energía contenida en los espectros de potencia obtenidos para diferentes frecuencias de perturbación con los diferentes controladores implementados

Puede observarse como la energía contenida en la señal de posición para el Control Clásico es mayor en el rango entre los 30 Hz y 40 Hz que en los demás controladores.

El control moderno con estimador MIMO presenta alta energía en la señal de posición por fuera de los valores de frecuencia de diseño, haciendo que este controlador sea poco robusto atenuando la perturbación ante el cambio de frecuencias de la misma por fuera de la frecuencia de diseño

El control Moderno con Filtro de Kalman, tanto Modelo de Perturbación Fijo como Variable, presentan una buena performance ante cambios en la frecuencia de perturbación, siendo un poco mejor este último.

A su vez, se aprecia como el Controlador Repetitivo presenta mejor desempeño debido a que la señal posición posee la menor energía introducida

por el motor vibrador. Se observa, a su vez, una muy buena robustez para todo el rango de frecuencia.

10 Conclusiones

Se diseñó y construyó un Levitador Magnético que permitió estudiar diferentes técnicas de control para el rechazo de perturbaciones del tipo vibratorias introducida al elemento levitado. Para introducir estas perturbaciones vibratorias, se insertó un motor desbalanceado al elemento levitado, al cual se le varió la tensión en bornes del motor variando a su vez la velocidad de giro de este.

Se controló la posición del elemento levitado a partir de técnicas de control clásico, control repetitivo, control moderno con estimador MIMO y control moderno con filtro de Kalman. A los estimadores antes mencionados se les introdujo con modelo de perturbación permitiendo estimar la perturbación introducida y a partir de esto, atenuarla. Debido a que, en la literatura analizada, el rechazo de perturbación era para una frecuencia fija, se procedió a diseñar un Filtro de Kalman con Modelo de Perturbación Variable. El cual, a partir de la medición de tensión del motor desbalanceado, permitió estimar la frecuencia de la perturbación y atenuarla. Cabe destacar que esta técnica fue desarrollada para este trabajo de maestría.

Se compararon la performance de los distintos controladores para atenuar la perturbación introducida, como así también, se analizó la robustez de los controladores implementados para atenuar perturbaciones por fuera de la frecuencia de diseño. A partir de las mediciones analizadas, se puede afirmar que, si bien todos los controladores presentaron grandes ventajas en el rechazo de la perturbación con respecto al control clásico, hay diferencias entre ellos. El Control Repetitivo logró atenuar, satisfactoriamente la frecuencia de perturbación de diseño manteniendo una baja amplificación de la señal para bajas frecuencias. A su vez presentó el mejor desempeño en

cuento a energía acumulada en la señal de posición y un bajo nivel del desvío estándar con respecto a los demás controladores. El Controlador Moderno con Estimador MIMO, si bien mostró una mejor capacidad de atenuación de la perturbación en la frecuencia de diseño con respecto al control clásico y repetitivo, presentó amplificación de la señal a baja frecuencia y el menor desempeño si se considera la energía contenida en la señal de posición y su desvío estándar. El Control Moderno con Filtro de Kalman (Modelo de Perturbación Fija), presentó una gran atenuación en la frecuencia de diseño de la perturbación con una menor amplificación de los componentes de baja frecuencia con respecto a los anteriores. Todos los controladores anteriores, mostraron una degradación en el desempeño al variar la frecuencia de la perturbación por fuera de la de diseño. En el caso del Control Moderno con Filtro de Kalman (Modelo de Perturbación Variable), como era de esperar, se observó el pico en la frecuencia de perturbación muy atenuado para todas las frecuencias analizadas, presentando casi un espectro de potencia plano, con una cierta amplificación a baja frecuencia. Este controlador logró buen desempeño, con respecto a la energía contenida en la señal de posición y su desvío estándar.

11 Observaciones, Trabajos a Futuro o Preguntas a Responder

El levitador magnético posee un eje para guiar al elemento levitado, este eje al tener huelgo con el elemento levitado genera ciertas oscilaciones por fuera del movimiento axial, aumentando el ruido del proceso. A menor huelgo genera mayor rozamiento haciendo que se modifique la dinámica del sistema. Se genera un compromiso entre reducir el ruido producto del desplazamiento radial y la fricción agregada al sistema.

Debido que el levitador magnético diseñado trata de captar la dinámica de un cojinete magnético axial, sería interesante lograr la rotación del elemento levitado para comprobar si el momento angular impacta en la estabilidad del sistema en sus desplazamientos horizontales (escapa del alcance de este trabajo, pero es interesante a futuro). También estudiar cómo impacta la colocación de electroimanes tanto arriba y abajo en la velocidad del sistema.

En el modelo matemático de la planta no se tiene en cuenta como varía la reluctancia del bloque corriente en función de la posición del elemento levitador, lo cual introduce un error de modelado de la planta.

El tiempo de muestreo de los controladores se lo determino a partir de la frecuencia de los polos del sistema levitador. Cuando se caracterizó la perturbación se encontró un valor adecuado en 33Hz (relación energía de la perturbación, frecuencia capaz de estimar). Debido a que se está muestreando a 2 ms (correspondiente a una frecuencia de Nyquist de 250Hz), y siendo en la práctica un valor frecuencia de muestreo recomendado es 10 veces la frecuencia a observar. La frecuencia de perturbación es 7.5 veces la frecuencia de Nyquist ($250\text{Hz}/33\text{Hz} \approx 7.5$), disminuyendo para frecuencias de perturbación mayores. Sería interesante poder muestrear mucho más rápido ($T_s \approx 500\mu\text{s}$) lograr una mejor performance en los controladores.

En una primera etapa, se utilizó como fuente de tensión, un transformador. Esta fuente de tensión utilizada introdujo un componente de error proveniente de la frecuencia de red. Como se trató de rechazar perturbación de frecuencia comparable a la de red, esto introducía problemas, por eso que, en la etapa final de medición de la performance de los controladores, se reemplazó la fuente por una fuente de tensión continua de laboratorio.

Quedaría a mejorar, medir directamente la velocidad de rotación del motor a partir de un tacómetro para evitar corrimientos en la velocidad de giro del motor.

Debido a que la variación en la frecuencia de perturbación, para el caso de Filtro de Kalman de Perturbación Variable, se realiza de manera offline, restaría analizar cómo afecta esta dinámica en caso de que la ajuste se realice on-line y cómo afecta el cambio en la frecuencia de perturbación a la estabilidad del sistema.

¿Qué aprendí? Aprendí lo que hay detrás de toda esta matemática, y como, sumando y restando, multiplicando y dividiendo, se pueden modificar las dinámicas de las cosas.

12 Apéndices

12.1 Instalación de Kernel PREEMPT_RT

A continuación, se detalla el procedimiento de instalación de Kernel Real Time en la placa Beagle Bone Black.

Para instalar el “real time kernel”, primero ver que se tiene instalado:

```
bone$ uname -a  
Linux ece434 4.19.94-ti-r45 #1buster SMP PREEMPT Thu Jun 18 19:18:41 UTC  
2020 armv7l GNU/Linux
```

A partir de esto, se busca un ‘kernel real time’ que posea un número similar:

```
bone $ cd /opt/scripts/tools  
bone$ sudo ./update_kernel.sh --bone-rt-kernel  
info: checking archive  
2020-08-06 15:12:44 URL:https://rcn-ee.com/repos/latest/buster-  
armhf/LATEST-bone-rt [195/195] -> "LATEST-bone-rt" [1]  
-----  
Kernel Options:  
ABI:1 LTS44 4.4.145-bone-rt-r23  
ABI:1 LTS49 4.9.178-bone-rt-r15  
ABI:1 LTS414 4.14.185-bone-rt-r37  
ABI:1 LTS419 4.19.127-bone-rt-r53  
ABI:1 LTS54 5.4.52-bone-rt-r31  
ABI:1 STABLE 5.6.17-bone-rt-r14  
-----  
Kernel version options:  
-----  
LTS414: --lts-4_14  
LTS419: --lts-4_19  
LTS54: --lts-5_4  
STABLE: --stable  
TESTING: --testing  
-----  
Please pass one of the above kernel options to update_kernel.sh  
-----
```

Debido a que el Sistema Operativo de la placa original posee un kernel 4.19, voy a instalar el 4.19.127-bone-rt-r53 que es la versión más actual del 'real time kernel'. Se procede a su instalación:

```
bone$ sudo ./update_kernel.sh --kernel 4.19.127-bone-rt-r53
```

Si se abre el archivo **uEnv.txt**, puede chequear el kernel que posee el sistema operativo instalado en la placa.

```
bone head /boot/uEnv.txt
#Docs: http://elinux.org/Beagleboard:U-boot\_partitioning\_layout\_2.0

uname r=4.19.94-ti-r45
#uuid=
#dtb=
bone$ sudo reboot
```

Después de reiniciar la placa, se observa el cambio de kernel instalado.

```
bone$ uname -a
Linux beaglebone 4.14.52-ti-rt-r60 #1 SMP PREEMPT Tue Jun 26 23:28:55
UTC 2018 armv71 GNU/Linux
```

12.2 Librería LibLevitador.c

Se detallan las funciones generales utilizadas en los distintos algoritmos de control

12.2.1 Inicialización Levitador Magnético

```
int InicioLevitador(int frecPWM)
{
    int periodo = 0;
    // ##### Configuración Salida PWM #####
    // Abro Puerto PWM
    char str[] = "0";

    FILE* configureP8_19 = fopen("/sys/class/pwm/pwmchip7/export", "w");
    if (configureP8_19 == NULL) {
        perror("Error: configureP8_19");
        return(-1);
    }
    else{
```

```

        fwrite(str,1,sizeof(str), configureP8_19);
        fclose(configureP8_19);
        printf("---- PWM: Configuro P8_19 ----\n");
    }

    printf("---- PWM: Configuro Periodo ----\n");
    FILE* fpwm_period = fopen("/sys/class/pwm/pwmchip7/pwm0/period", "w");
    if ( fpwm_period == NULL ){
        perror("Error: fpwm_period");
        return(-1);
    }
    else{
        char periodo_str[20];
        periodo = (int) 1e9/frecPWM;    //Periodo [nseg] 200kHz = 1/5000nseg
        sprintf(periodo_str, "%d", periodo); //Convierte int a char
        fwrite (periodo_str, sizeof(char), sizeof(periodo_str),
fpwm_period);
        fclose(fpwm_period);
    }

    // Habilito Salida
    char enable_pwm[] = "1";    //[nseg] 20kHz = 1/50,000nseg

    FILE* fpwm_enable = fopen("/sys/class/pwm/pwmchip7/pwm0/enable", "w");
    if ( fpwm_enable == NULL ){
        perror("Error: fpwm_enable");
        return(-1);
    }
    else{
        fwrite (enable_pwm, sizeof(char), sizeof(enable_pwm), fpwm_enable);
        fclose(fpwm_enable);
        printf("---- PWM: Habilito Salida ----\n");
    }
    return(0);
}

```

12.2.2 Finalización Levitador Magnético

```
int FinLevitador(void){
```

```

char enable_pwm[] = "0";
//fclose("/sys/bus/iio/devices/iio:device0/in_voltage0_raw");
//fclose("/sys/bus/iio/devices/iio:device0/in_voltage1_raw");

FILE* fpwm_enable = fopen("/sys/class/pwm/pwmchip7/pwm0/enable", "w");
if ( fpwm_enable == NULL ){
    perror("Error: fpwm_enable");
    return(-1);
}
else{
    fwrite (enable_pwm, 1, sizeof(enable_pwm), fpwm_enable);
    fclose(fpwm_enable);
    printf("---- PWM: Deshabilito PWM ----\n");
    return(0);
}
}

```

12.2.3 Medición Puertos Analógicos

```

int MedicionP9_38(){
    int valuePos=0;

    FILE* f = fopen("/sys/bus/iio/devices/iio:device0/in_voltage3_raw",
"r"); //Corriente PIN9_38 AIN2
    if ( f == NULL ){
        perror("Error: f0");
        return(-1);
    }
    int itemRead = fscanf(f,"%d",&valuePos); //Medicion Posicion
    if(itemRead<0.0)
        printf("ERROR Leyendo P9_38\n");

    fclose(f); // Cierro Puerto medicion

    return valuePos;
}

```

12.2.4 Salida PWM

```

int SeteoDuty(float u,int frecPWM)
{
    int periodo;
    int duty_cycle;
    char duty_cycle_str[10] ;    //[nseg] 20kHz = 1/50,000nseg

    FILE* fpwm_duty = fopen("/sys/class/pwm/pwmchip7/pwm0/duty_cycle", "w");
    if ( fpwm_duty == NULL ){
        perror("Error: fpwm_duty");
        return -1;
    }
    else{
        periodo = (int) 1e9/frecPWM;
        if(u>0.999)
            u=0.999;
        if(u<0.001)
            u=0.001;
        duty_cycle = (int)(u*periodo);

        sprintf(duty_cycle_str, "%i", duty_cycle);
        fwrite (duty_cycle_str, sizeof(char), sizeof(duty_cycle_str),
fpwm_duty);
        fclose(fpwm_duty);
        return duty_cycle;
    }
}

```

12.3 Algoritmos de Control implementados

12.3.1 Algoritmo de Control Clásico

```

//Calibracion de la medicion de corriente, para distintas frecuencias de PWM
y valores de PWM

/*gcc HolaMundo.c -o HolaMundo -lm (lm: para linkear la math library) */
/* ./HolaMundo */
// gcc Control.c -o Control -lm -I../..//Para que tome las librerias con
la LibLevitador (dos directorios arriba)

// ls -ltar /dev/pwm
// sudo su

```

```

// config-pin P9_22 pwm

#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <sys/time.h>
#include <time.h>
#include <errno.h>
#include <sys/resource.h>
#include <sched.h>
#include "LibLevitador.h"
#include <math.h>

//#define _POSIX_C_SOURCE 200809L
//#define _XOPEN_SOURCE 500

int cmpfunc (const void * a, const void * b) {
    return ( *(int*)a - *(int*)b );
}

/* ##### */
/* ##### Def. Tiempo Muestreo ##### */
/* ##### */
float Ts = 0.002;          // [seg] Tiempo Muestreo

char file[] = "Clasico_Borrar.dat";

// Start measuring time
int k=0;
int N=50000;

float Posicion[5];        //[mm] Posicion Iman
float PosicionMedio = 0;
float HuelgoFiltro = 0;

float Corriente[5];       // Descarto Min y Max y me quedo con la Mediana
float CorrienteMedio = 0;
float CorrienteFiltro = 0;

```

```

int smooth = 200;          //Ventana de promedio en medicion de perturbacion
int Vibracion[200];

float VibracionMedio = 0;

float u=0;                 //[0 - 1] Accion de Control

float u_lin;
int prueba; //Variable para ver porque en el osciloscopio se ven cero en la
accion de control,
//Seteo Frec. PWM
int frecPWM = 100e3; // Frecuencia de PWM [Hz]

int periodo;

float dT; // [seg] Tiempo q tarda el bucle
struct timespec begin, t0, t1, end, duerme;
struct sched_param sp = {.sched_priority = 90};

// Parametros Controlador Corriente en plano Z (discreto)
float ei, ei_old=0; //Error
float u_old;
float SPi;                //SetPoint Corriente

// Parametros Controlador Posicion en plano Z (discreto)
float ePos, ePos_old=0; //Error
float uPos_old;
// Accion de control u(k-1)

float SPpos = 0;         //Posicion de Equilibrio
double I_eq = 0;        //Corriente de equilibrio para SPpos

float wp = 0;           // [rad/s] Frecuencia Perturbacion
float Huelgo;           // Distancia entre Arandela del Elemento Levitado y
electroiman

```

```

//float DPosz_cero;
//float DPosz_polo;
//float DPosz_Gain;

int main(){
    for (int i=0;i<smooth;i=i+1){
        Vibracion[i] = 1000;
    }

    float Datos[N][7];      //[Tiempo Corriente SPi Posicion Sph u wp]

printf("Tiempo de Muestreo: %f\n",Ts);

// *****
// **** Seteo Prioridad del Proceso ****
// *****

    setpriority(PRIO_PGRP,getpid(),-20); //Aumento la prioridad del proceso
al maximo (-20) Hace falta correrlo como "sudo su"

    printf("Prioridad del Proceso ");
    printf("%i\n",getpriority(PRIO_PGRP,getpid()));

    sched_setscheduler(0,SCHED_FIFO,&sp);

// *****
// **** Setup del Entradas/Salidas ****
// *****

    system("config-pin P8_19 pwm"); //Configuro P8_13 como PWM

    InicioLevitador(frecPWM); // Configuro Entradas/Salidas del levitador

    u=0.0;
    SeteoDuty(u,frecPWM);
    printf("---- PWM: Configuro Duty Cycle ----\n");

    printf("---- INICIO PROGRAMA ----\n");

```

```

clock_gettime(CLOCK_REALTIME, &begin);

u_lin = LinealCorriente(u);

I_eq = 0.5;
SPpos = sqrt(2780*pow(I_eq,2)/48.4); //sqrt((Kmag*i_eq^2)/masa)
//SPpos = 4.0;
printf("---- SetPoint: %f ----\n", SPpos);

for (k=1;k<200;k=k+1){

clock_gettime(CLOCK_REALTIME, &t0);

(float)MedicionP9_39();
(float)MedicionP9_40();
SeteoDuty(0, frecPWM);

clock_gettime(CLOCK_REALTIME, &t1);

duerme.tv_sec = 0;
duerme.tv_nsec = 1.0e9*Ts-(t1.tv_nsec-t0.tv_nsec);

if (duerme.tv_nsec>0)
    nanosleep(&duerme, NULL);
else
    printf("---- No Cumple Tiempo Real ----\n");

}

k=0;

clock_gettime(CLOCK_REALTIME, &begin);

while(k<N){

/*
        if (k*Ts>2.0){

```

```

        //I_eq = 0.6;
        SPpos = sqrt(2780*pow(0.6,2)/49.15);
//sqrt((Kmag*i_eq^2)/masa)
    }

    if (k*Ts>3.0){
        //I_eq = 0.5;
        SPpos = sqrt(2780*pow(0.5,2)/49.15);
//sqrt((Kmag*i_eq^2)/masa)
    }

    if (k*Ts>4.0){
        //I_eq = 0.6;
        SPpos = sqrt(2780*pow(0.65,2)/49.15);
//sqrt((Kmag*i_eq^2)/masa)
    }

    if (k*Ts>5.0){
        //I_eq = 0.7;
        SPpos = sqrt(2780*pow(0.5,2)/49.15);
//sqrt((Kmag*i_eq^2)/masa)
    }

    if (k*Ts>6.0){
        //I_eq = 0.5;
        SPpos = sqrt(2780*pow(0.6,2)/49.15);
//sqrt((Kmag*i_eq^2)/masa)
    }
*/

    clock_gettime(CLOCK_REALTIME, &t0);
// #####
// ##### Medicion Variables #####
// #####

    Posicion[0] = (float)MedicionP9_39();
    Posicion[1] = (float)MedicionP9_39();
    Posicion[2] = (float)MedicionP9_39();
    Posicion[3] = (float)MedicionP9_39();
    Posicion[4] = (float)MedicionP9_39();

```

```

qsort (Posicion, sizeof (Posicion) / sizeof (float), sizeof (float), cmpfunc);
//qSort (Lista; NroItems, Funcion que dice cuando es mayor)

        PosicionMedio = (Posicion[1]+Posicion[2]+Posicion[3])/3;
//Descarto los valores Max y Min
        PosicionMedio = PosicionMedio/4095.0;

        Huelgo = Cuentas2Posicion (PosicionMedio);

        Corriente[0] = (float) MedicionP9_40 ();
        Corriente[1] = (float) MedicionP9_40 ();
        Corriente[2] = (float) MedicionP9_40 ();
        Corriente[3] = (float) MedicionP9_40 ();
        Corriente[4] = (float) MedicionP9_40 ();

qsort (Corriente, sizeof (Corriente) / sizeof (float), sizeof (float), cmpfunc);
//qSort (Lista; NroItems, Funcion que dice cuando es mayor)

        CorrienteMedio = (Corriente[1]+Corriente[2]+Corriente[3])/3;
//Descarto los valores Max y Min

        CorrienteMedio = CorrienteMedio/4095.0;

        /* ***** */
        /* ***** Bloque Medicion de Frecuencia ***** */
        /* ***** */

        Vibracion[0] = MedicionP9_38 ();
        VibracionMedio = 0;
        for (int i=0; i<smooth; i=i+1){
        VibracionMedio = VibracionMedio+Vibracion[i];
        }

        VibracionMedio = VibracionMedio/smooth; //Saco Promedio
        VibracionMedio = VibracionMedio/4095.0;

        wp = 2*3.14*Cuentas2Frecuencia (VibracionMedio);
        for (int i=smooth; i>0; i=i-1){

```

```

        Vibracion[i] = Vibracion[i-1];
    }

    // #####
    // ##### Controlador Dmag #####
    // #####

    ePos = (SPpos - Huelgo);

    // Controlador(float error,float error_old, float u_old,float
cero,float polo,float K)
    //SPi = Controlador(ePos,ePos_old,uPos_old,0.9,0.15,-4.5);
    //SPi = Controlador(ePos,ePos_old,uPos_old,0.9,0.15,-4.1);
    SPi = Controlador(ePos,ePos_old,uPos_old,0.8983,0.1452,-4.0);
//ANDA

    ePos_old = ePos;
    uPos_old = SPi;

    // #####
    // ##### Controlador Di #####
    // #####

    ei = (SPi-CorrienteMedio+I_eq);

    // u = Controlador(ei,ei_old,ui_old,0.4,1.0,2.4); //
Controlador(error(k),error(k-1),u(k-1),cero,polo,K)
    u = 1.0*ei;

    //u = Controlador(ei,ei_old,u_old,0.1,0.8,0.5);
    ei_old = ei;
    u_old = u;

    u = u+I_eq/1.0;

    if (u>1.0)
        u = 1.0;

    if (u<0.0)
        u=0.0;

    u_lin = LinealCorriente( u );

```

```

prueba = SeteoDuty(u_lin, frecPWM);

// ##### //
// ##### Tiempo Muestreo ##### //
// ##### //

clock_gettime(CLOCK_REALTIME, &end);

Datos[k][0] = (end.tv_sec-begin.tv_sec)+1e-9*(end.tv_nsec-
begin.tv_nsec);
Datos[k][1] = CorrienteMedio; //[-] Corriente
Datos[k][2] = SPi; // SP Corriente
Datos[k][3] = Huelgo; //[mm] Posicion
Datos[k][4] = SPpos; //[mm] SP Posicion
Datos[k][5] = u; //[-] Accion control
Datos[k][6] = VibracionMedio; //[-] Perturbacion

clock_gettime(CLOCK_REALTIME, &t1);

duerme.tv_sec = 0;
duerme.tv_nsec = 1.0e9*Ts-(t1.tv_nsec-t0.tv_nsec);

if (duerme.tv_nsec>0)
    nanosleep(&duerme, NULL);
else
    printf("---- No Cumple Tiempo Real ----\n");

    k=k+1;
}
printf("---- FIN WHILE ----\n");
u=0.0;
SeteoDuty(u, frecPWM);

// ##### //
// ##### Archivo Salida ##### //
// ##### //
printf("---- FILE: Genero Archivo Salida ----\n");

```

```

FILE* f = fopen(file,"w");

for (k=0;k<N;k++) {

fprintf(f,"%f\t%f\t%f\t%f\t%f\t%f\t%f\n",Datos[k][0],Datos[k][1],Datos[k][2]
,Datos[k][3],Datos[k][4],Datos[k][5],Datos[k][6]);
}
fclose(f);

FinLevitador();

printf("---- FIN PROGRAMA ----\n");

return 0;
}

```

Algoritmo de Control Repetitivo

```
// Control Repetitivo
```

```
/*gcc HolaMundo.c -o HolaMundo -lm (lm: para linkear la math library) */
```

```
/* ./HolaMundo */
```

```
// ls -ltar /dev/pwm
```

```
// sudo su
```

```
// config-pin P9_22 pwm
```

```
// gcc Control.c -o Control -lm -I../..//Para que tome las librerias con
la LibLevitador (dos directorios arriba)
```

```
#include <stdio.h>
```

```
#include <stdlib.h>
```

```
#include <unistd.h>
```

```
#include <sys/time.h>
```

```
#include <time.h>
```

```
#include <errno.h>
```

```
#include <sys/resource.h>
```

```
#include <sched.h>
```

```
#include "LibLevitador.h"
```

```
#include <math.h>
```

```

#define PI 3.14159265359

int cmpfunc (const void * a, const void * b) {
    return ( *(int*)a - *(int*)b );
}
/* ##### */
/* ##### Def. Tiempo Muestreo ##### */
/* ##### */
float Ts = 0.002;          // [seg] Tiempo Muestreo
char file[] = "Borrar_RC_FIR.dat";

// Start measuring time
int k=0;
int N=50000;

int i = 0;
float Posicion[5];        //[mm] Posicion Iman
float PosicionMedio = 0;

float Corriente[5];       // Descarto Min y Max y me quedo con la Mediana
float CorrienteMedio = 0;
float CorrienteFiltro = 0;

float u=0;                 //[0 - 1] Accion de Control

float u_lin;

//Seteo Frec. PWM
int frecPWM = 500e3; // Frecuencia de PWM [Hz]

int periodo;

float dT; // [seg] Tiempo q tarda el bucle
struct timespec begin, t0, t1, end, duerme;
struct sched_param sp = {.sched_priority = 90};

// *****
// * Parametros Controlador Posicion en plano Z (discreto) *
// *****

```

```

float ePos=0, ePos_old=0; //Error
float uPos_old;
// Accion de control u(k-1)

float SPpos = 0;          //Posicion de Equilibrio
double I_eq = 0;         //Corriente de equilibrio para SPpos

float Huelgo;            // Distancia entre Arandela del Elemento Levitado y
electroiman

float in_Dmag=0, in_Dmag_old=0;
// *****
// * Parametros Controlador Corriente en plano Z (discreto) *
// *****

float ei=0, ei_old=0;    //Error
float ui_old;            // Accion de control u(k-1)
float SPi;                //SetPoint

int main(){

    // *****
    // * Parametros Controlador Repetitivo *
    // *****

    float wp = 2*PI*33; // [rad/s] Frec. de Perturbacion
    float Krc = 0.5;

    // Q(z)
    float OutQ;
    float InQ[11] = {0,0,0,0,0,0,0,0,0,0,0};
    float Qnum[12] =
{0.006142,0.01956,0.06095,0.1247,0.1843,0.2087,0.1843,0.1247,0.06095,0.01956
,0.006142};

    //OutQ = a0*InQ(k)+a1*InQ(k-1)+a2*InQ(k-2)+...+aN*InQ(k-N)

    int delay;
    delay = round((2*PI/Ts)/wp+0.5);
    float Z[delay];
    for (i=0; i<delay;i=i+1){

```

```

        Z[i]=0;
    }
    // *****
    // MEDICIONES
    // *****
    float Datos[N][5];          //[Tiempo Corriente Posicion u uRC]

printf("Tiempo de Muestreo: %f\n",Ts);

// *****
// **** Seteo Prioridad del Proceso ****
// *****
    setpriority(PRIO_PGRP,getpid(),-20); //Aumento la prioridad del proceso
al maximo (-20) Hace falta correrlo como "sudo su"
    printf("Prioridad del Proceso ");
    printf("%i\n",getpriority(PRIO_PGRP,getpid()));
    sched_setscheduler(0,SCHED_FIFO,&sp);

// *****
// **** Setup del Entradas/Salidas ****
// *****

    system("config-pin P8_19 pwm"); //Configuro P8_13 como PWM
    InicioLevitador(frecPWM); // Configuro Entradas/Salidas del levitador

    u=0.0;
    SeteoDuty(u,frecPWM);
    printf("---- PWM: Configuro Duty Cycle ----\n");
printf("---- INICIO PROGRAMA ----\n");
    clock_gettime(CLOCK_REALTIME,&begin);
    u_lin = LinealCorriente(u);
    //Cuando pongo la I_eq segun modelo no llega a chupar el elemento
levitado
    I_eq = 0.5;
    SPpos = sqrt(2780*pow(I_eq,2)/49.15); //sqrt((Kmag*i_eq^2)/masa)
    // SPpos = 3.5;
    printf("---- SetPoint: %f ----\n",SPpos);

    for (i=1;i<200;i=i+1){

        clock_gettime(CLOCK_REALTIME,&t0);

```

```

(float)MedicionP9_39();
(float)MedicionP9_40();
SeteoDuty(0,frecPWM);

clock_gettime(CLOCK_REALTIME,&t1);

duerme.tv_sec = 0;
duerme.tv_nsec = 1.0e9*Ts-(t1.tv_nsec-t0.tv_nsec);

if (duerme.tv_nsec>0)
    nanosleep(&duerme,NULL);
else
    printf("---- No Cumple Tiempo Real ----\n");
}
k=0;
clock_gettime(CLOCK_REALTIME,&begin);

while(k<N){

    clock_gettime(CLOCK_REALTIME,&t0);
    // #####
    // ##### Medicion Variables #####
    // #####
    Posicion[0] = (float)MedicionP9_39();
    Posicion[1] = (float)MedicionP9_39();
    Posicion[2] = (float)MedicionP9_39();
    Posicion[3] = (float)MedicionP9_39();
    Posicion[4] = (float)MedicionP9_39();

qsort(Posicion,sizeof(Posicion)/sizeof(float),sizeof(float),cmpfunc);
//qSort(Lista;NroItems,Funcion que dicecuando es mayor)
    PosicionMedio = (Posicion[1]+Posicion[2]+Posicion[3])/3;
//Descarto los valores Max y Min
    PosicionMedio = PosicionMedio/4095.0;
    Huelgo = Cuentas2Posicion(PosicionMedio);

    Corriente[0] = (float)MedicionP9_40();
    Corriente[1] = (float)MedicionP9_40();
    Corriente[2] = (float)MedicionP9_40();
    Corriente[3] = (float)MedicionP9_40();

```

```

Corriente[4] = (float)MedicionP9_40();

qsort(Corriente, sizeof(Corriente)/sizeof(float), sizeof(float), cmpfunc);
//qSort(Lista;NroItems,Funcion que dicecuando es mayor)

CorrienteMedio = (Corriente[1]+Corriente[2]+Corriente[3])/3;
//Descarto los valores Max y Min
CorrienteMedio = CorrienteMedio/4095.0;

// ##### //
// ##### Controlador Dmag ##### //
// ##### //

ePos = (SPpos - Huelgo);

// ##### //
// ##### Controlador Repetitivo ##### //
// ##### //

// Controlador(float in,float in_old, float out_old,float
cero,float polo,float K)
//out_Qz = Controlador(Z[delay-1],Z[delay-
2],out_Qz_old,0,0.3141,0.6859);

InQ[0] =ePos*Krc+Z[delay-1];

OutQ = 0;
for(i=0;i<11;i=i+1){
    OutQ = OutQ+InQ[i]*Qnum[i];
}
//    printf("%f\n",OutQ);

for(i=10;i>0;i=i-1){
    InQ[i] = InQ[i-1];
}

for (i=delay-1;i>0;i=i-1){
    Z[i] = Z[i-1];
}

```

```

z[0] = -1.0*OutQ;

// ##### //
// ## Sumador Error+u Repetitivo ### //
// ##### //

in_Dmag = ePos+Z[delay-1];

// ##### //
// ##### Controlador Posicion ##### //
// ##### //

// Controlador(float error,float error_old, float u_old,float
cero,float polo,float K)
SPi = Controlador(in_Dmag,in_Dmag_old,uPos_old,0.8983,0.1452,-
4.4);

in_Dmag_old = in_Dmag;

uPos_old = SPi;

// ##### //
// ##### Controlador Di ##### //
// ##### //

//ei = (SPi-CorrienteMedio+I_eq);

ei = (SPi-CorrienteMedio+I_eq);

u = 1.0*ei;
ei_old = ei;
ui_old = u;

u = u+I_eq/0.95;

if (u>1.0)
    u = 1.0;

if (u<0)
    u=0;

```

```

u_lin = LinealCorriente( u );

SeteoDuty(u_lin,frecPWM);

// ##### //
// ##### Tiempo Muestreo ##### //
// ##### //

clock_gettime(CLOCK_REALTIME,&end);
Datos[k][0] = (end.tv_sec-begin.tv_sec)+1e-9*(end.tv_nsec-
begin.tv_nsec);
Datos[k][1] = CorrienteMedio; //[A] Corriente
Datos[k][2] = u; // SP Corriente
Datos[k][3] = Huelgo; // Salida Lazo Control Repetitivo
Datos[k][4] = Z[delay-1]; // Salida Lazo Control
Repetitivo

// Datos[k][0] = (end.tv_sec-begin.tv_sec)+1e-9*(end.tv_nsec-
begin.tv_nsec);
// Datos[k][1] = CorrienteMedio; //[A] Corriente
// Datos[k][2] = SPi; // SP Corriente
// Datos[k][3] = Huelgo; //[mm] Posicion
// Datos[k][4] = SPpos;
// Datos[k][5] = u;

clock_gettime(CLOCK_REALTIME,&t1);

duerme.tv_sec = 0;
duerme.tv_nsec = 1.0e9*Ts-(t1.tv_nsec-t0.tv_nsec);

if (duerme.tv_nsec>0)
    nanosleep(&duerme,NULL);
else
    printf("---- No Cumple Tiempo Real ----\n");
k=k+1;
}
printf("---- FIN WHILE ----\n");

```

```

u=0.0;
SeteoDuty(u, frecPWM);

// ##### //
// ##### Archivo Salida ##### //
// ##### //
printf("---- FILE: Genero Archivo Salida ----\n");

FILE* f = fopen(file, "w");

for (k=0; k<N; k++) {
//
fprintf(f, "%f\t%f\t%f\t%f\t%f\n", Datos[k][0], Datos[k][1], Datos[k][2], Datos[k][3], Datos[k][4], Datos[k][5]);

fprintf(f, "%f\t%f\t%f\t%f\n", Datos[k][0], Datos[k][1], Datos[k][2], Datos[k][3], Datos[k][4]);
}
fclose(f);

FinLevitador();

printf("---- FIN PROGRAMA ----\n");

return 0;
}

```

12.3.2 Algoritmo de Control con Estimador MIMO

```

//Control Moderno con Rechazo de Perturbacion
// ModernoPert_MIMO.c

/*gcc HolaMundo.c -o HolaMundo -lm (lm: para linkear la math library) */
/* ./HolaMundo */
// gcc ModernoPert_MIMO.c -o ModernoPert_MIMO -lm -I../..//Para que tome
las librerias con la LibLevitador (dos directorios arriba)

```

```

// ls -ltar /dev/pwm
// sudo su
// config-pin P9_22 pwm

#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <sys/time.h>
#include <time.h>
#include <errno.h>
#include <sys/resource.h>
#include <sched.h>
#include "LibLevitador.h"
#include <math.h>

int cmpfunc (const void * a, const void * b) {
    return ( *(float*)a - *(float*)b );
}

/* ##### */
/* ##### Def. Tiempo Muestreo ##### */
/* ##### */
float Ts = 0.002;      // [seg] Tiempo Muestreo

char file[] = "Borrar_MIMO.dat";

// Start measuring time
int k=0;
int N=50000;

int Posicion[5];      //[mm] Posicion Iman
float PosicionMedio = 0;

int Corriente[5];     // Descarto Min y Max y me quedo con la Mediana
float CorrienteMedio = 0;
float CorrienteFiltro = 0;

float u = 0;
float uTotal = 0; // [0 - 1] Accion de Control
float u_lin;

```

```

//Seteo Frec. PWM
int frecPWM = 500e3; // Frecuencia de PWM [Hz]

int periodo;

float dT; // [seg] Tiempo q tarda el bucle
struct timespec begin, t0, t1, end, duerme;
struct sched_param sp = {.sched_priority = 90};

float PosEq = 0; //Posicion de Equilibrio
float I_eq = 0; //Corriente de equilibrio para SPpos
float y[2] = {0,0}; // Salida MIMO [Corriente, Posicion] con respecto a
Equilibrio

float Huelgo; // Distancia entre Arandela del Elemento Levitado y
electroiman

int main(){

    float Datos[N][7]; // [Tiempo Corriente SPi Posicion Sph u]

    printf("Tiempo de Muestreo: %f\n", Ts);

    // *****
    // **** Seteo Prioridad del Proceso ****
    // *****
    setpriority(PRIO_PGRP, getpid(), -20); //Aumento la prioridad del proceso
al maximo (-20) Hace falta correrlo como "sudo su"

    printf("Prioridad del Proceso ");
    printf("%i\n", getpriority(PRIO_PGRP, getpid()));

    sched_setscheduler(0, SCHED_FIFO, &sp);

    // *****
    // **** Setup del Entradas/Salidas ****
    // *****

```

```

printf("---- PWM: Configuro Duty Cycle ----\n");
system("config-pin P8_19 pwm"); //Configuro P8_13 como PWM
InicioLevitador(frecPWM); // Configuro Entradas/Salidas del levitador
SeteoDuty(0.0,frecPWM);

printf("---- INICIO PROGRAMA ----\n");
I_eq = 0.5;
PosEq = sqrt(2780*pow(I_eq,2)/48.4); //sqrt((Kmag*i_eq^2)/masa)

printf("---- SetPoint: %f ----\n",PosEq);

// *****
// ***** Bloque Inicializacion Control Moderno *****
// *****

// Ancho Banda Controlador wc = 200.0 rad/s
// Ancho Banda Estimador wo = 4*wc

float A[5][5] = {{ 0.9048 ,0.00190,0.000000,0.0000,0.00000},\
                 {-93.6338,0.90480,0.000000,0.0000,0.00000},\
                 {0.460300,0.00050,0.496600,0.0000,0.00000},\
                 {-20.6056,-0.0146,-56.6540,1.0105,10.4683},\
                 {-0.0146,-6.3008e-06,-0.0631,0.002,1.0105}};

/*
float A[5][5] = {{ 0.0 ,0.0,0.000000,0.0000,0.00000},\
                 {0.0,0.0,0.000000,0.0000,0.00000},\
                 {0.460300,0.00050,0.496600,0.0000,0.00000},\
                 {-20.6056,-0.0146,-56.6540,1.0105,10.4683},\
                 {-0.0146,-6.3008e-06,-0.0631,0.002,1.0105}};
*/

float B[5] = {0.000, 0.000,0.4782,-20.9713,-0.0147};
float C[2][5] = {{0.0,0.0,1.0,0.0,0.0},\
                 {0.0,0.0,0.0,0.0,1.0}};

// ORIGINAL
float K[5] = {1,0,0.55271,-0.0066045,-0.61257};
float L[5][2] =
{{0.12627,0.63136},{32.5371,162.6854},{0.022387,0.11194},{46.3202,231.6008},
{0.26532,1.3266}};

```

```

float x_est[5], x_est_new[5];

x_est[0] = 0;
x_est[1] = 0;
x_est[2] = 0;
x_est[3] = 0;
x_est[4] = 0;

float y_est[2] = {0.0,0.0}; //[Corriente Huelgo]
//
*****
// Espero un par de segundos para acomodar el Elemento Levitado

for (k=1;k<200;k=k+1){

clock_gettime(CLOCK_REALTIME,&t0);
SeteoDuty(0,frecPWM);
clock_gettime(CLOCK_REALTIME,&t1);

duerme.tv_sec = 0;
duerme.tv_nsec = 1.0e9*Ts-(t1.tv_nsec-t0.tv_nsec);

if (duerme.tv_nsec>0)
    nanosleep(&duerme,NULL);
else
    printf("---- No Cumple Tiempo Real ----\n");
}

k=0;
clock_gettime(CLOCK_REALTIME,&begin);

while(k<N){

    clock_gettime(CLOCK_REALTIME,&t0);
    // ##### //
    // ##### Medicion Variables ##### //
    // ##### //
    Posicion[0] = MedicionP9_39();
    Posicion[1] = MedicionP9_39();
    Posicion[2] = MedicionP9_39();
    Posicion[3] = MedicionP9_39();
}

```

```

        Posicion[4] = MedicionP9_39();

qsort (Posicion, sizeof (Posicion) / sizeof (float), sizeof (float), cmpfunc);
//qSort (Lista; NroItems, Funcion que dice cuando es mayor)

        PosicionMedio = (Posicion[0]+Posicion[1]+Posicion[2])/3;
//Descarto los valores Max y Min
        PosicionMedio = PosicionMedio/4095.0;
        Huelgo = Cuentas2Posicion (PosicionMedio);

        Corriente[0] = MedicionP9_40();
        Corriente[1] = MedicionP9_40();
        Corriente[2] = MedicionP9_40();
        Corriente[3] = MedicionP9_40();
        Corriente[4] = MedicionP9_40();

qsort (Corriente, sizeof (Corriente) / sizeof (int), sizeof (int), cmpfunc);
//qSort (Lista; NroItems, Funcion que dice cuando es mayor)
        CorrienteMedio = (Corriente[1]+Corriente[2]+Corriente[3])/3;
//Descarto los valores Max y Min
        CorrienteMedio = CorrienteMedio/4095.0;

        //CorrienteMedio = (float)MedicionP9_40()/4095.0;
        // #####
        // ##### Estimador + Accion control #####
        // #####

u = -
(K[0]*x_est[0]+K[1]*x_est[1]+K[2]*x_est[2]+K[3]*x_est[3]+K[4]*x_est[4]);

        //printf ("---- Accion Control: %f ----\n", u);

        y[0] = CorrienteMedio-I_eq;
        y[1] = Huelgo - PosEq;

        y_est[0] =
C[0][0]*x_est[0]+C[0][1]*x_est[1]+C[0][2]*x_est[2]+C[0][3]*x_est[3]+C[0][4]*
x_est[4];

```

```

        y_est[1] =
C[1][0]*x_est[0]+C[1][1]*x_est[1]+C[1][2]*x_est[2]+C[1][3]*x_est[3]+C[1][4]*
x_est[4];

        x_est_new[0] = L[0][0]*(y[0]-y_est[0])+L[0][1]*(y[1]-
y_est[1])+B[0]*u+A[0][0]*x_est[0]+A[0][1]*x_est[1]+A[0][2]*x_est[2]+A[0][3]*
x_est[3]+A[0][4]*x_est[4];

        x_est_new[1] = L[1][0]*(y[0]-y_est[0])+L[1][1]*(y[1]-
y_est[1])+B[1]*u+A[1][0]*x_est[0]+A[1][1]*x_est[1]+A[1][2]*x_est[2]+A[1][3]*
x_est[3]+A[1][4]*x_est[4];

        x_est_new[2] = L[2][0]*(y[0]-y_est[0])+L[2][1]*(y[1]-
y_est[1])+B[2]*u+A[2][0]*x_est[0]+A[2][1]*x_est[1]+A[2][2]*x_est[2]+A[2][3]*
x_est[3]+A[2][4]*x_est[4];

        x_est_new[3] = L[3][0]*(y[0]-y_est[0])+L[3][1]*(y[1]-
y_est[1])+B[3]*u+A[3][0]*x_est[0]+A[3][1]*x_est[1]+A[3][2]*x_est[2]+A[3][3]*
x_est[3]+A[3][4]*x_est[4];

        x_est_new[4] = L[4][0]*(y[0]-y_est[0])+L[4][1]*(y[1]-
y_est[1])+B[4]*u+A[4][0]*x_est[0]+A[4][1]*x_est[1]+A[4][2]*x_est[2]+A[4][3]*
x_est[3]+A[4][4]*x_est[4];

        x_est[0] = x_est_new[0];
        x_est[1] = x_est_new[1];
        x_est[2] = x_est_new[2];
        x_est[3] = x_est_new[3];
        x_est[4] = x_est_new[4];

        // ##### //
        // ##### Controlador Di ##### //
        // ##### //

        uTotal = u+I_eq/1.1;

        if (uTotal>1.0)
            uTotal = 1.0;

        if (uTotal<0)
            uTotal=0;

        u_lin = LinealCorriente(uTotal);

        SeteoDuty(u_lin,frecPWM);

```

```

// ##### //
// ##### Tiempo Muestreo ##### //
// ##### //

clock_gettime(CLOCK_REALTIME, &end);

Datos[k][0] = (end.tv_sec-begin.tv_sec)+1e-9*(end.tv_nsec-
begin.tv_nsec);
Datos[k][1] = CorrienteMedio; //[A] Corriente
Datos[k][2] = y_est[0]+I_eq; //[A] Corriente Estimado
Datos[k][3] = Huelgo; //[mm] Posicion
Datos[k][4] = y_est[1]+PosEq; // Pos Estimado
Datos[k][5] = uTotal; // Accion Control
Datos[k][6] = x_est[0]; // Accion Control

clock_gettime(CLOCK_REALTIME, &t1);

duerme.tv_sec = 0;
duerme.tv_nsec = 1.0e9*Ts-(t1.tv_nsec-t0.tv_nsec);

if (duerme.tv_nsec>0)
    nanosleep(&duerme, NULL);
else
    printf("---- No Cumple Tiempo Real ----\n");

k=k+1;
}
printf("---- FIN WHILE ----\n");

SeteoDuty(u, frecPWM);

// ##### //
// ##### Archivo Salida ##### //
// ##### //
printf("---- FILE: Genero Archivo Salida ----\n");

FILE* f = fopen(file, "w");

for (k=0; k<N; k++) {

```

```

fprintf(f, "%f\t%f\t%f\t%f\t%f\t%f\t%f\n", Datos[k][0], Datos[k][1], Datos[k][2]
, Datos[k][3], Datos[k][4], Datos[k][5], Datos[k][6]);
    }
    fclose(f);

    FinLevitador();
    printf("---- FIN PROGRAMA ----\n");
    return 0;
}

```

12.3.3 Algoritmo de Control Moderno con Filtro Kalman Perturbación Variable

```

//Control Moderno con Filtro de Kalman Rechazo de Perturbacion Variable

/*gcc HolaMundo.c -o HolaMundo -lm (lm: para linkear la math library) */
/* ./HolaMundo */
// gcc ModernoPert_MIMO.c -o ModernoPert_MIMO -lm -I../..//Para que tome
las librerias con la LibLevitador (dos directorios arriba)

// ls -ltar /dev/pwm
// sudo su
// config-pin P9_22 pwm

#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <sys/time.h>
#include <time.h>
#include <errno.h>
#include <sys/resource.h>
#include <sched.h>
#include "LibLevitador.h"
#include <math.h>

int cmpfunc (const void * a, const void * b) {

```

```

    return ( *(float*)a - *(float*)b );
}

/* ##### */
/* ##### Def. Tiempo Muestreo ##### */
/* ##### */
float Ts = 0.002;          // [seg] Tiempo Muestreo

char file[] = "KalmanVariable_Borrar.dat";

// Start measuring time
int k=0;
int N=50000;

int Posicion[5];          //[mm] Posicion Iman
float PosicionMedio = 0;

int Corriente[5];        // Descarto Min y Max y me quedo con la Mediana
float CorrienteMedio = 0;
float CorrienteFiltro = 0;
int smooth = 200;        //Ventana de promedio en medicion de perturbacion
int Vibracion[200];

float VibracionMedio = 0;

float uTotal = 0; // [0 - 1] Accion de Control
float u_lin;

//Seteo Frec. PWM
int frecPWM = 500e3; // Frecuencia de PWM [Hz]

int periodo;

float dT; // [seg] Tiempo q tarda el bucle
struct timespec begin, t0, t1, end, duerme;
struct sched_param sp = {.sched_priority = 90};

float PosEq = 0;          //Posicion de Equilibrio
float I_eq = 0;           //Corriente de equilibrio para SPos
float y = 0;              // Salida Corriente con respecto a Equilibrio

```

```

float Huelgo;          // [mm] Distancia entre Arandela del Elemento Levitado y
electroiman
float wp = 0;          // [rad/s] Frecuencia Perturbacion
int main(){

    float Datos[N][7];    //[Tiempo Corriente SPi Posicion Sph u]

    printf("Tiempo de Muestreo: %f\n",Ts);

// *****
// **** Seteo Prioridad del Proceso ****
// *****
    setpriority(PRIO_PGRP,getpid(),-20); //Aumento la prioridad del proceso
al maximo (-20) Hace falta correrlo como "sudo su"

    printf("Prioridad del Proceso ");
    printf("%i\n",getpriority(PRIO_PGRP,getpid()));

    sched_setscheduler(0,SCHED_FIFO,&sp);

// *****
// **** Setup del Entradas/Salidas ****
// *****

    printf("---- PWM: Configuro Duty Cycle ----\n");
    system("config-pin P8_19 pwm"); //Configuro P8_13 como PWM
    InicioLevitador(frecPWM); // Configuro Entradas/Salidas del levitador
    SeteoDuty(0.0,frecPWM);

    printf("---- INICIO PROGRAMA ----\n");
    I_eq = 0.5;
    PosEq = sqrt(2780*pow(I_eq,2)/49.3); //sqrt((Kmag*i_eq^2)/masa)
    //PosEq = 4.0;
    printf("---- SetPoint: %f ----\n",PosEq);

// *****
// ***** Bloque Inicializacion Control Moderno *****
// *****

    // Ancho Banda Controlador wc = 200.0 rad/s

```

```

// Ancho Banda Estimador   wo = 4*wc

// [rho; drho; I; dh; h]

float den1 = 0.0;
float den2 = 0.0;
// Inicializacion con wp=2*pi*35Hz;
float A[5][5] = {{ 0.8628 ,0.00190,0.000000,0.0000,0.00000},\
                 {-133.9663,0.8628,0.000000,0.0000,0.00000},\
                 {0.4523,0.00050,0.496600,0.0000,0.00000},\
                 { -20.4419,-0.0146,-56.6540,1.0105,10.4683},\
                 {-0.0145,-7.5e-6,-0.06310,0.0020,1.01050}};

float At[5][5];
traspuesta(5,5,A,At);

float B[5][1] = {{0}, {0},{0.4782},{-20.9713},{-0.0147}};

/***** Pruebo Distintas Covarianzas *****/
// float B1[5][1] = {{1},{1},{1},{1},{1}};
// float B1t[1][5];
// traspuesta(5,1,B1,B1t);
// float Rw[1][1] = {{0.05}}; // Proceso
/*
float Rv[1][1] = {{0.001}}; //Medicion 0.001 = Covarianza la saqe de una
medicion de posicion (sin pertur)

float Cov[5][5] = {{0.00001,0.0000,0.0000 ,0.0000 ,0.0000},\
                  {0.0000,0.00001,0.0000 ,0.0000 ,0.0000},\
                  {0.0000,0.0000,0.0009 ,-0.0028,0.0001},\
                  {0.0000,0.0000,-0.0028,361.839,-0.366},\
                  {0.0000,0.0000,0.00010,-0.366 ,0.0010}};
*/
float Rv[1][1] = {{0.001}}; //Medicion 0.001 = Covarianza la saqe de una
medicion de posicion (sin pertur)

float Cov[5][5] = {{0.001,0.00000,0.0000 ,0.0000 ,0.0000},\
                  {0.00000,0.0001,0.0000 ,0.0000 ,0.0000},\
                  {0.00000,0.00000,0.0009 ,-0.0028,0.0001},\

```

```

                                {0.00000,0.00000,-0.0028,361.839,-0.366},\
                                {0.00000,0.00000,0.00010,-0.366 ,0.001}};

float C[1][5] = {{0.0,0.0,0.0,0.0,1.0}};
float Ct[5][1];
traspuesta(1,5,C,Ct);

//float K[1][5] = {{0.6692,-0.8109,-0.0080,1.000,0.000}}; //
// [rho; drho; I; dh; h]

// float K[1][5] = {{1.0,0.0,0.5527,-0.0066,-0.6126}}; // Original con
0.27 Ganancia RLS
// float K[1][5] = {{1.0,0.0,0.4489,-0.0052,-0.4280}}; // con
0.25*0.27 Ganancia RLS (abs(Pk)=[351;104;104]
float K[1][5] = {1,0,0.37091,-0.005112,-0.36156}; // con 0.1*0.27
Ganancia RLS (abs(Pk)=[365;193;193]
// float K[1][5] = {{1.0,0.0,0.9583,-0.0095,-0.7803}}; // [-600 -140 -
140]

float x_est[5][1];
inicializarMatriz(5,1,x_est);

float xbar[5][1];
inicializarMatriz(5,1,xbar);

float M[5][5];
inicializarMatriz(5,5,M);

float P[5][5];
inicializarMatriz(5,5,P);

float u[1][1] = {{0}};

//
*****
// Espero un par de segundos para acomodar el Elemento Levitado

for (k=1;k<200;k=k+1){

clock_gettime(CLOCK_REALTIME,&t0);

```

```

SeteoDuty(0,frecPWM);
clock_gettime(CLOCK_REALTIME,&t1);

duerme.tv_sec = 0;
duerme.tv_nsec = 1.0e9*Ts-(t1.tv_nsec-t0.tv_nsec);

if (duerme.tv_nsec>0)
    nanosleep(&duerme,NULL);
else
    printf("---- No Cumple Tiempo Real ----\n");
}

k=0;
clock_gettime(CLOCK_REALTIME,&begin);

while(k<N){

    clock_gettime(CLOCK_REALTIME,&t0);
    // ##### //
    // ##### Medicion Variables ##### //
    // ##### //
    Posicion[0] = MedicionP9_39();
    Posicion[1] = MedicionP9_39();
    Posicion[2] = MedicionP9_39();
    Posicion[3] = MedicionP9_39();
    Posicion[4] = MedicionP9_39();

qsort(Posicion,sizeof(Posicion)/sizeof(float),sizeof(float),cmpfunc);
//qSort(Lista;NroItems,Funcion que dicecuando es mayor)

    PosicionMedio = (Posicion[0]+Posicion[1]+Posicion[2])/3;
//Descarto los valores Max y Min
    PosicionMedio = PosicionMedio/4095.0;
    Huelgo = Cuentas2Posicion(PosicionMedio);

    Corriente[0] = MedicionP9_40();
    Corriente[1] = MedicionP9_40();
    Corriente[2] = MedicionP9_40();
    Corriente[3] = MedicionP9_40();

```

```

Corriente[4] = MedicionP9_40();

qsort(Corriente, sizeof(Corriente)/sizeof(int), sizeof(int), cmpfunc);
//qSort(Lista; NroItems, Funcion que dice cuando es mayor)
    CorrienteMedio = (Corriente[1]+Corriente[2]+Corriente[3])/3;
//Descarto los valores Max y Min
    CorrienteMedio = CorrienteMedio/4095.0;

/* ***** */
/* ***** Bloque Medicion de Frecuencia ***** */
/* ***** */

Vibracion[0] = MedicionP9_38();

VibracionMedio = 0;
for (int i=0; i<smooth; i=i+1){
VibracionMedio = VibracionMedio+Vibracion[i];
}

VibracionMedio = VibracionMedio/smooth; //Saco Promedio
VibracionMedio = VibracionMedio/4095.0;

wp = 2*3.14*Cuentas2Frecuencia(VibracionMedio);
for (int i=smooth; i>0; i=i-1){
    Vibracion[i] = Vibracion[i-1];
}

// #####//
// ##### Medicion de Frecuencia Perturbacion #####//
// #####//

//wp = 2*3.14*46.6;

//CorrienteMedio = (float)MedicionP9_40()/4095.0;
// #####//
// ##### Estimador + Accion control #####//
// #####//

```

```

den1 = pow(wp,2)+pow(350.0,2);
den2 = pow(wp,4)+1.2772e5*pow(wp,2)+6.3901e8;
A[0][0] = cos(Ts*wp);
A[0][1] = sin(Ts*wp)/wp;
A[0][2] = 0.0;
A[0][3] = 0.0;
A[0][4] = 0.0;
A[1][0] = -wp*sin(Ts*wp);
A[1][1] = cos(Ts*wp);
A[1][2] = 0.0;
A[1][3] = 0.0;
A[1][4] = 0.0;
A[2][0] = (1.164e5*cos(Ts*wp)+332.5*wp*sin(Ts*wp)-57790)/den1;
A[2][1] = (165.1*wp+1.164e5*sin(Ts*wp)-
332.5*wp*cos(Ts*wp))/(wp*den1);
A[2][2] = 0.4966;
A[2][3] = 0.0;
A[2][4] = 0.0;
A[3][0] = -(4.5653e9*wp*sin(Ts*wp)-
1.3043e7*pow(wp,2)*cos(Ts*wp)+6.5860e6*pow(wp,2)+1.3399e10)/den2;
A[3][1] = (4.5653e9*cos(Ts*wp)+1.3043e7*wp*sin(Ts*wp)-
1.8843e4*pow(wp,2)-4.5744e9)/den2;
A[3][2] = -56.65;
A[3][3] = 1.01;
A[3][4] = 10.47;
A[4][0] = (4.5653e9*cos(Ts*wp)+1.3043e7*wp*sin(Ts*wp)-
1.8843e4*pow(wp,2)-4.5744e9)/den2;
A[4][1] = (3.9087e6*wp+4.5653e9*sin(Ts*wp)-
1.3043e7*wp*cos(Ts*wp)-20.9713*pow(wp,3))/(wp*den2);
A[4][2] = -0.06307;
A[4][3] = 0.002007;
A[4][4] = 1.01;

y = Huelgo - PosEq;
// P(k)=M(k)-M(k)*C'*inv[C*M(k)*C'+Rv]*C*M(k)

float aux00[1][5];
float aux01[1][1];
float aux02[1][1];
float aux03[5][1];
float aux04[5][1];

```

```

float aux05[1][5];
float aux06[5][5];

inicializarMatriz(1,5,aux00);
inicializarMatriz(1,5,aux01);

ProductoMatriz(1,5,5,5,C,M,aux00); //C*M(k)
ProductoMatriz(1,5,5,1,aux00,Ct,aux01); //C*M(k)*C'
aux02[0][0] = 1/(aux01[0][0]+Rv[0][0]);

ProductoMatriz(5,5,5,1,M,Ct,aux03); //M(k)*C'
ProductoMatriz(5,1,1,1,aux03,aux02,aux04);
//M(k)*C'*inv(C*M(k)*C'+Rv)
ProductoMatriz(1,5,5,5,C,M,aux05); //C*M(k)
ProductoMatriz(5,1,1,5,aux04,aux05,aux06);
//M(k)*C'*inv[C*M(k)*C'+Rv]*C*M(k)

restaMatriz(5,5,M,aux06,P); //P(k)=M(k)-
M(k)*C'*inv[C*M(k)*C'+Rv]*C*M(k)

// xest(k)=xbar(k)+P(k)*C'*inv(Rv)*[y(k)-C*xbar(k)]
float aux07[1][1];
float aux08[1][1];
float aux09[5][1];
float aux10[5][1];

inicializarMatriz(1,1,aux07);
inicializarMatriz(1,1,aux08);

ProductoMatriz(1,5,5,1,C,xbar,aux07); // C*xbar(k)
aux08[0][0] = (y-aux07[0][0])/Rv[0][0];
ProductoMatriz(5,5,5,1,P,Ct,aux09); //
ProductoMatriz(5,1,1,1,aux09,aux08,aux10); // P(k)*Ct*Rv^-
1*(y(k)-C*xbar(k))

sumaMatriz(5,1,xbar,aux10,x_est);

// M(k+1)=A*P(k)*A'+B*Rw*B'
float aux11[5][5];
float aux12[5][5];
float aux13[5][1];

```

```

float aux14[5][5];

inicializarMatriz(5,5,aux11);
inicializarMatriz(5,5,aux12);
inicializarMatriz(5,1,aux13);
inicializarMatriz(5,5,aux14);

traspuesta(5,5,A,At);
ProductoMatriz(5,5,5,5,A,P,aux11); //A*P(k)
ProductoMatriz(5,5,5,5,aux11,At,aux12); //A*P(k)*At

//ProductoMatriz(5,1,1,1,B1,Rw,aux13); // B1*Rw
//ProductoMatriz(5,1,1,5,aux13,B1t,aux14); // B1*Rw*B1t
/*
de proceso
aux14[0][0] = Rw[0][0]; // aux14 = B1*Rw*B1t Diagonal con Ruido
aux14[1][1] = Rw[0][0];
aux14[2][2] = Rw[0][0];
aux14[3][3] = Rw[0][0];
aux14[4][4] = Rw[0][0];

sumaMatriz(5,5,aux12,aux14,M); // M(k+1) = A*P(k)*At+B1*Rw*B1t
*/

sumaMatriz(5,5,aux12,Cov,M); // M(k+1) =
A*P(k)*At+CovarianzaProceso
// u(k) = -K*x_est(k);
ProductoMatriz(1,5,5,1,K,x_est,u); //
u[0][0] = -1.0*u[0][0];

// xbar(k+1)=A*x_est(k)+B*u(k)

float aux15[5][1];
float aux16[5][1];
inicializarMatriz(1,5,aux15);

ProductoMatriz(5,5,5,1,A,x_est,aux15); // A*x_est(k)
ProductoMatriz(5,1,1,1,B,u,aux16);

sumaMatriz(5,1,aux15,aux16,xbar); // xbar(k+1) =
A*x_est(k)+B*u(k)

```

```

// ##### //
// ##### //
// ##### //

uTotal = u[0][0]+I_eq/1.0;

if (uTotal>1.0)
    uTotal = 1.0;

if (uTotal<0)
    uTotal=0;
u_lin = LinealCorriente(uTotal);

SeteoDuty(u_lin,frecPWM);
// ##### //
// ##### Tiempo Muestreo ##### //
// ##### //

clock_gettime(CLOCK_REALTIME, &end);
/*
    Datos[k][0] = (end.tv_sec-begin.tv_sec)+1e-9*(end.tv_nsec-
begin.tv_nsec);
    Datos[k][1] = x_est[0][0]; // Perturbacion
    Datos[k][2] = x_est[1][0]; //velocidad Perturbacion
    Datos[k][3] = Huelgo; // [mm] Posicion
    Datos[k][4] = x_est[4][0]+PosEq; // Posicion Estimada
    Datos[k][5] = uTotal;
    Datos[k][6] = wp;
*/
    Datos[k][0] = (end.tv_sec-begin.tv_sec)+1e-9*(end.tv_nsec-
begin.tv_nsec);
    Datos[k][1] = CorrienteMedio; // [A] Corriente
    Datos[k][2] = x_est[2][0]+I_eq; // [A] Corriente Estimado
    Datos[k][3] = Huelgo; // [mm] Posicion
    Datos[k][4] = x_est[4][0]+PosEq; // Posicion Estimada
    Datos[k][5] = uTotal;
    Datos[k][6] = x_est[0][0];

clock_gettime(CLOCK_REALTIME, &t1);

```

```

    duerme.tv_sec = 0;
    duerme.tv_nsec = 1.0e9*Ts-(t1.tv_nsec-t0.tv_nsec);

    if (duerme.tv_nsec>0)
        nanosleep(&duerme,NULL);
    else
        printf("---- No Cumple Tiempo Real ----\n");

        k=k+1;
}
printf("---- FIN WHILE ----\n");

SeteoDuty(0,frecPWM);

// ##### //
// ##### Archivo Salida ##### //
// ##### //
printf("---- FILE: Genero Archivo Salida ----\n");

FILE* f = fopen(file,"w");

for (k=0;k<N;k++) {

fprintf(f,"%f\t%f\t%f\t%f\t%f\t%f\t%f\n",Datos[k][0],Datos[k][1],Datos[k][2]
,Datos[k][3],Datos[k][4],Datos[k][5],Datos[k][6]);
}
fclose(f);

FinLevitador();

printf("---- FIN PROGRAMA ----\n");

return 0;
}

```

13 Agradecimientos

Bueno, va llegando el momento de agradecer, momento con sabor a finalización, que no es poco. Primero, y siempre, agradecer a la educación pública, gratuita, universal y de calidad. A Natalia, Salvador, Olivia que no están explícitamente en este trabajo, pero están en las horas de descanso, alegrías y cosas que hacen que valga la pena hacerlo. A mis compañeros de trabajo, Gonzalo, Alfredo, Juancito, Fabian, Norbert, Andres, que compartimos el gusto por el Control y la Automatización, el aprender y el enseñar. A Felix, mi psicólogo técnico. A Juan Pablo, por enseñarme desde cómo hacer una cerveza a cómo controlar un sistema inestable, por escucharme, preguntarme, irse a pensar y volver a preguntarme. A Graciela, que siempre me pincho para que termine con este trabajo. Muchas gracias y ahora a ver que sale...

14 Referencias

- [1] Federal Institute of Technology Zurich, «ICMB International Center of Magnetic Bearing,» 2024. [En línea]. Available: <https://www.ifr.mavt.ethz.ch/research/icmb/>. [Último acceso: 2023].
- [2] J. Powell y G. Maise, «StarTram: The Magnetic Launch Path to Very Low Cost, Very High Volume Launch to Space,» *2008 14th Symposium on Electromagnetic Launch Technology*, 2008.
- [3] S. P. C. M. G. M. W. Katherine Mirica, «Magnetic Levitation in the Analysis of Foods and Water,» *Journal of Agricultural and Food Chemistry*, 2010.
- [4] M. Morshuis, C. Nojiri y M. Schönbrodt, «DuraHeart magnetically levitated centrifugal left ventricular assist system for advanced heart failure patients,» 2010.
- [5] S. G., «Active magnetic bearings-chances and limitations,» 2002.
- [6] S. Düsterhaupt, H. Neumann, T. Rottenbach y F. Worlitz, «High temperature active magnetic bearings in industrial steam turbines,» *The 15th International Symposium on Magnetic Bearings*, 2016.
- [7] M. A. Yoder, «Embedded Linux Wiki,» 19 Marzo 2024. [En línea]. Available: https://elinux.org/EBC_Exercise_36_PREEMPT_RT.
- [8] S. Skogestad y I. Postlethwaite, *Multivariable Feedback Control. Analysis and Design* (2nd Edition), Wiley, 2007.
- [9] G. Hillerström y K. Walgama, «Repetitive Control Theory and Applications - A Survey,» *13th Triennial World Congress*, vol. 29, nº ISSN 1474-6670, 1996.

- [10] J. Cortes-Romero y G. Ramos-Fuentes, «Control GPI-repetitivo para sistemas lineales con incertidumbre/variación en los parámetros,» *Tecno Lógicas*, vol. 18, nº ISSN 0123-7799, pp. 13-24, 2015.
- [11] M. Tomizuka y C. Kempf, «Design of Discrete Time Repetitive Controllers with Applications to Mechanical Systems,» *IFAC Proceedings Volumes*, nº ISSN 1474-6670, pp. 243-248, 1990.
- [12] D. W. B. Z. Y. W. J. F. S. d. H. Keliang Zhou, «Dual-mode structure digital repetitive control,» *Automatica*, Vols. %1 de %2Volume 43, Issue 3, nº ISSN 0005-1098, pp. 546-554, 2007.
- [13] L. Cuiyan, Z. Dongchun y Z. Xianyi, «A Survey of Repetitive Control,» *RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1160-1166 vol.2, 2004.
- [14] G. Franklin, D. Powell y A. Emami-Naeimi, *Feedback Control of Dynamics Systems* 6th ed., Pearson, 2009.
- [15] K. Ogata, *Sistemas de Control en Tiempo Discreto - 2da Edición*, Pearson Educación, 1996.
- [16] R. Dorf y R. Bishop, *Modern Control Systems* 12th ed., Prentice Hall, 2011.
- [17] W. P. Kloster, «Estimador Asintótico de Estados. Caso MIMO Observable,» Mar del Plata, 2020.
- [18] H. Yaghoubi, «The most important maglev applications,» 2013.
- [19] N. R. a. G. H. a. S. E. L. Hemenway, «Magnetic Bearing Technology for Industrial Bearingless Motor Systems,» *2019 IEEE Workshop on Electrical Machines Design, Control and Diagnosis (WEMDCD)*, vol. 1, 2019.
- [20] G. Yang, Z. Shi, N. Mo y L. Zhao, «Research on active magnetic bearing applied in Chinese modular high-temperature gas-cooled reactor,» vol. 77, nº 0149-1970.

[21] MatLab - MathWorks, «Centro de Ayuda - pwelch,» [En línea]. Available:
<https://la.mathworks.com/help/signal/ref/pwelch.html>. [Último acceso: 2023].