



UNIVERSIDAD POLITÉCNICA SALESIANA

SEDE GUAYAQUIL

CARRERA:

INGENIERÍA ELECTRÓNICA

TRABAJO DE TITULACIÓN PREVIO A LA OBTENCIÓN DEL TÍTULO DE:

INGENIERO ELECTRÓNICO

TÍTULO

**DISEÑO E IMPLEMENTACIÓN DE CONTROLADOR PID PARA UN BALANCÍN
DE DOS GRADOS DE LIBERTAD.**

AUTORES:

SANAGUANO PINCAY DAVID ERNESTO

ARAUJO SÁNCHEZ ÁLVARO PAÚL

TUTOR:

MSC. VICENTE PEÑARANDA

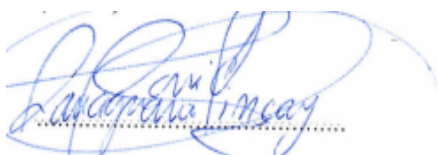
GUAYAQUIL-ECUADOR

2020

CERTIFICADO DE RESPONSABILIDAD Y AUTORÍA DEL TRABAJO DE TITULACIÓN.

Nosotros, David Ernesto Sanaguano Pincay y Álvaro Paúl Araujo Sánchez, autores del proyecto técnico denominado Diseño e implementación de controlador PID para un balancín de dos grados de libertad, certificamos que los análisis realizados, diagramas desarrollados y conclusiones son de nuestra exclusiva responsabilidad y autoría.

Guayaquil, agosto del 2020.



David Ernesto Sanaguano Pincay

C.I: 0919377127



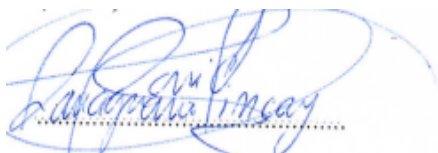
Álvaro Paúl Araujo Sánchez

C.I: 0706647021

CERTIFICADO DE CESIÓN DE DERECHOS

A través de la presente declaración, cedemos nuestros derechos de propiedad intelectual correspondiente en este trabajo de titulación a la Universidad Politécnica Salesiana, según lo establecido por la ley de la propiedad intelectual y por su normatividad institucional vigente.

Guayaquil, agosto del 2020.



David Ernesto Sanaguano Pincay

C.I: 0919377127



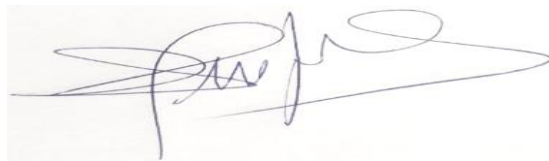
Álvaro Paúl Araujo Sánchez

C.I: 0706647021

CERTIFICADO DE DIRECCIÓN DEL TRABAJO DE TITULACIÓN

En calidad de director del trabajo de titulación **DISEÑO E IMPLEMENTACIÓN DE CONTROLADOR PID PARA UN BALANCÍN DE DOS GRADOS DE LIBERTAD**, elaborado por Sanaguano Pincay David Ernesto y Araujo Sánchez Álvaro Paúl. Autorizo la aprobación del presente trabajo de titulación basándome en la supervisión y dirección, obteniendo los resultados esperados, cumpliendo con los objetivos planteados, e informe final considerados por la Universidad Politécnica Salesiana como trabajo final de titulación.

Guayaquil, agosto del 2020.



Msc. Vicente Avelino Peñaranda Idrovo.
DIRECTOR DEL TRABAJO DE TITULACIÓN.

DEDICATORIA

Dedico este trabajo de titulación a Dios, que me cubre siempre de sabiduría y bendiciones, a mi Papa que a pesar de haber estado una gran parte de mi vida fuera de casa por su profesión, siempre está pendiente de nosotros su familia, a mi Madre que ha sido un soporte fundamental en mi vida y más aún en los momentos en que más la necesitaba, a mi hermana que ha sido mi gran compañera en mi carrera universitaria, y a todos aquellos que depositaron su confianza y me dieron su apoyo absoluto e incondicional en mi vida.

David Ernesto Sanaguano Pincay

DEDICATORIA

Dedico este trabajo de titulación a Dios, a mis padres quienes han sido parte fundamental en mi vida, ya que gracias a sus consejos y apoyo incondicional me han alentado para seguir con mi carrera, a mis hermanos que han estado para mí en todo momento, a mi esposa que es mi mano amiga y de manera muy especial a mi pequeña hija que es mi fuente de inspiración para seguir creciendo como persona y profesional. A todos aquellos que depositaron su confianza y me apoyaron en lo absoluto en mi etapa universitaria.

Álvaro Paúl Araujo Sánchez.

RESUMEN

Año	Título	Alumnos	Tutor	Tema de titulación
2020	Ingeniero Electrónico	Sanaguano Pincay David Ernesto Araujo Sánchez Álvaro Paúl	MSC. Vicente Peñaranda	Diseño E Implementación De Controlador PID Para Un Balancín De Dos Grados De Libertad

El presente proyecto: DISEÑO E IMPLEMENTACIÓN DE CONTROLADOR PID PARA UN BALANCÍN DE DOS GRADOS DE LIBERTAD, consiste en el diseño y control de un balancín de dos grados de libertad con el fin de proporcionar a la Universidad Politécnica Salesiana un prototipo para la implementación de las prácticas de laboratorio de las cátedras de Teoría de Control 1 y 2.

Las prácticas ayudan a mejorar conocimientos y fomentar enseñanza dentro de los temas de Controladores PID. Los mismos que son usados por los alumnos de la Universidad Politécnica Salesiana sede Guayaquil para que puedan fortalecer conocimientos teóricos, desarrollar destrezas en el análisis y diseño de sistemas de control en un software de ingeniería como LabVIEW. El presente proyecto consta de 7 prácticas, donde cada práctica está elaborada para guiar el procedimiento, paso a paso para el diseño, adquisición de datos y control del sistema.

El balancín consta de una barra pivotada en su centro, en un extremo posee un motor brushless de 12V el cual realiza el movimiento en el eje Y, y en otro extremo posee 2 motores brushless de 12V, los cuales realizan el movimiento para el eje X, se modifica la velocidad de cada motor para posicionar el balancín en sus dos ejes, y se estabiliza la posición mediante el controlador PID y un algoritmo de control.

La adquisición de datos se realiza en el Software LabVIEW por medio de un microcontrolador Arduino Mega 2560. Mediante el uso de un sensor Rotacional, se detecta los cambios de posición. La comunicación entre el Arduino y el Software LabView se realiza a través de un puerto serial el cual es seleccionado por el bloque

serial Port en la Interfaz gráfica de la aplicación y a su vez el bloque de Baud Rate donde se establece la velocidad en baud/s.

ABSTRACT

Year	Profession	Students	Director	Technical Project
2020	Electronic Engineer	Sanaguano Pincay David Ernesto Araujo Sánchez Álvaro Paúl	MSC. Vicente Peñaranda	DESIGN AND IMPLEMENTATION OF PID CONTROLLER FOR A BALANCIN OF TWO DEGREES OF FREEDOM

The present project: DESIGN AND IMPLEMENTATION OF PID CONTROLLER FOR A BALANCIN OF TWO DEGREES OF FREEDOM, consists of the development of a 2 degrees of freedom helicopter structure in order to provide the Universidad Politecnica Salesiana with a prototype for the implementation of the laboratory practices of the chairs of Control Theory 1 and 2.

The practices will help improve knowledge and promote teaching within the topics of PID Controllers. The same ones that we use by the students of the Universidad Politecnica Salesiana of headquarters Guayaquil so that they can strengthen theoretical knowledge, develop skills in the analysis and design of control systems in engineering software such as LabVIEW. This project contain 7 practices, where each practice is elaborated to guide the procedure, step by step for the design, data acquisition and control of the system.

The structure which consists of a pivoted bar in its center, at one end it has a 12v brushless motor which will perform the movement on the Y axis, and at another end it has 2 brushless 12 v motors, which will perform the movements for the X axis, the pulse width of each motor is modified to manipulate the rocker position on its two axes, and the position is stabilized by the PID controller and a control algorithm.

Data acquisition is performed in LabVIEW Software by means of an Arduino Mega 2560 microcontroller. By using a rotational sensor, changes in position are detected. The communication between the Arduino and the LabWiew Software is done through

a serial port which will be selected by the serial Port block in the graphical interface of the application and in turn the Baud Rate block where the speed is set in baud/s

ÍNDICE GENERAL

CERTIFICADO DE RESPONSABILIDAD Y AUTORÍA DEL TRABAJO DE TITULACIÓN.....	II
CERTIFICADO DE CESIÓN DE DERECHOS.....	III
CERTIFICADO DE DIRECCIÓN DEL TRABAJO DE TITULACIÓN	IV
DEDICATORIA.....	V
DEDICATORIA.....	VI
ABSTRACT	IX
ÍNDICE GENERAL.....	XI
ÍNDICE DE FIGURAS.....	XIII
ÍNDICE DE TABLAS	XV
INTRODUCCIÓN	1
1. EL PROBLEMA.....	2
1.1. Planteamiento del problema.	2
1.2. Importancia y alcance	2
1.3. Delimitación	2
1.3.1. Delimitación temporal	3
1.3.2. Delimitación espacial	3
1.3.3. Delimitación social	3
1.3.4. Delimitación tecnológica.....	3
1.3.5. Impacto.....	4
1.4. Justificación	4
1.5. Grupo objetivo	4
1.6. Explicación del problema.....	5
1.7. Objetivos.....	5
1.7.1. Objetivo general	5
1.7.2. Objetivos específicos	5
2. FUNDAMENTOS TEORICOS	6
2.1. Conceptos básicos en el sistema de control	6
2.1.1. Sistema	6
2.1.2. Perturbación.....	6
2.1.3. Control realimentado.....	6
2.1.4. Entrada.....	7
2.1.5. Salida.....	7
2.1.6. Controlador PID	7
2.1.7. Control Proporcional	8
2.1.8. Control Derivativo.....	8
2.1.9. Control Integral.....	8

2.2.	Motores sin escobilla	9
2.3.	Tarjeta Arduino Mega2560.....	9
2.4.	Software LabVIEW	9
2.5.	Software Matlab.....	10
2.6.	Método Ziegler-Nichols	10
2.7.	Método Ziegler-Nichols en lazo cerrado	10
3.	MARCO METODOLOGICO.....	13
3.1.	Diseño mecánico del balancín de dos grados de libertad.....	13
3.1.1.	Parte fija	13
3.1.2.	Parte móvil	14
3.2.	Diseño del algoritmo.....	14
3.2.1.	Comunicación del Arduino con LabVIEW.	14
3.2.2.	Control de en LabVIEW.	15
3.2.3.	Lectura de encoder.	16
3.2.4.	Control manual-automático.....	16
3.2.5.	Control PID.....	16
3.2.6.	Recopilación de datos.....	17
3.3.	Diseño del circuito electrónico	18
3.4.	Diseño del diagrama de bloques en LabVIEW.....	20
3.5.	Diseño de interfaz gráfica en LabView	23
3.5.1.	Interfaz gráfica modo de operación	23
4.	ANÁLISIS E INTERPRETACIÓN DE LOS RESULTADOS.....	26
4.1.	Análisis de los parámetros PID por Ziegler-Nichols en el eje Y.....	26
4.1.1.	Obtención de los parámetros del PID en el eje Y.	27
4.2.	Análisis de los parámetros PID por Ziegler-Nichols en el eje X.....	28
4.2.1.	Obtención de los parámetros del PID en X.....	29
4.3.	Funcionamiento del controlador PID.....	31
4.3.1.	Funcionamiento del controlador PID en el eje de las Y.	31
4.3.2.	Funcionamiento del controlador PID en el eje X.....	35
	CONCLUSIONES.....	40
	RECOMENDACIONES.....	41
	BIBLIOGRAFÍA.....	42
	ANEXOS	44

ÍNDICE DE FIGURAS.

Figura 1. Ubicación geográfica de la Universidad Politécnica Salesiana.....	3
Figura 2. Diagrama de control realimentado. (Ogata K. , 2010).....	7
Figura 3. Control proporcional integral derivativo. (PID) (Sánchez, 2012).....	7
Figura 4. Diagrama de control proporcional. (P) (Sánchez, 2012).	8
Figura 5. Diagramas de control derivativo (D) (Sánchez, 2012).....	8
Figura 6. Motor sin escobilla (Padmaraja Yedamale, Microchip Technology Inc., 2003)	9
Figura 7. Tarjeta Arduino Mega 2560. (Arduino corp., 2020)	9
Figura 8. Plataforma LabVIEW (National Instruments, 2020).....	10
Figura 9. Lazo cerrado del método Ziegler-Nichols. (Garcia, 2020).....	11
Figura 10. Tabla de sintonía de Z. N en lazo cerrado. (Departamento de Control, División de Ingeniería Eléctrica Facultad de Ingeniería UNAM., 2006).....	11
Figura 11. Balancín de 2 grados de libertad	13
Figura 12. Diagrama del Balancín de dos grados de libertad.....	14
Figura 13. Bloque de comunicación entre el Arduino y LabVIEW.....	15
Figura 14. Bloques para el control de ancho de pulso de los 3 motores.....	15
Figura 15. Bloque para lectura de encoder.	16
Figura 16. Control manual-automático de la planta.	16
Figura 17. Control PID de la planta	17
Figura 18. Bloque para guardar los datos en archivo de Excel.	18
Figura 19. Conexión de la fuente de voltaje a 120v.....	18
Figura 20. Conexión de los motores y drivers con el Arduino	19
Figura 21. Librería de MakerHub.....	20
Figura 22. While Loop para el envío y recepción de datos desde el Arduino.	20
Figura 23. Conexión de los módulos de control de ancho de pulso para cada motor.	21
Figura 24. Conversión en grados de los datos obtenidos del sensor rotacional en Y	21
Figura 25. Conexión del control PID para el movimiento en el eje Y del sistema....	22
Figura 26. Conexión del control PID para el movimiento en el eje X del sistema....	22
Figura 27. Panel frontal para la inicialización del motor 1	24
Figura 28. Control manual de ancho de pulso para inicializar los motores 2-3.....	24
Figura 29. Panel frontal para el control manual en el eje en X y el eje en Y.....	25
Figura 30. Panel frontal para la obtención y almacenamiento de datos.....	25
Figura 31. Gráfica que se usa para el método de Ziegler-Nichols.	26
Figura 32. Gráfica con los valores de L y T para Ziegler-Nichols en el eje Y.	27
Figura 33. Controlador PID sintonizado con Ziegler-Nichols en el eje Y.....	28
Figura 34. Gráfica de lazo abierto para la planta de la planta en el eje X.....	29
Figura 35. Gráfica con los valores de L y T para Ziegler-Nichols en el eje X	29
Figura 36. Controlador PID sintonizado con Ziegler-Nichols en el eje X.....	31
Figura 37. Inicio de funcionamiento del sistema en el eje Y en 0 grados.	31
Figura 38. Funcionamiento del controlador PID para 20 grados de punto de ajuste	32
Figura 39. Controlador PID estabilizando el sistema en los 20 grados de punto de ajuste.....	32
Figura 40. Aplicación de perturbación en 20 grados de punto de ajuste.	33

Figura 41. Cambio de 20 grados a 40 grados de punto de ajuste.....	33
Figura 42. Sistema estabilizado en 40 grados de punto de ajuste.	34
Figura 43. Respuesta del controlador PID ante una perturbación en 40 grados de punto de ajuste.....	34
Figura 44. Funcionamiento del controlador PID en 60 grados de punto de ajuste.	35
Figura 45. Estabilización del sistema en 60 grados del eje X.	36
Figura 46. Respuesta del controlador PID frente a perturbación inducida.	36
Figura 47. Respuesta del sistema frente al cambio de punto de ajuste de 60 a 90 grados.....	37
Figura 48. Estabilización del sistema en 90 grados de punto de ajuste en el eje X.	37
Figura 49. Respuesta del controlador PID frente a perturbación en los 90 grados del eje X.	38
Figura 50. Vista general de la interfaz gráfica en respuesta del sistema frente a una mínima perturbación en el eje Y.	38
Figura 51. Vista general de la interfaz gráfica en respuesta del sistema frente a una mayor perturbación en el eje Y.	39
Figura 52. Instalación de motores sin escobilla en el balancín.	89
Figura 53. Conexiones de todos los componentes de la planta hacia Arduino.....	89
Figura 54. Cableado de motores hacia Arduino.....	90
Figura 55. Pruebas de Comunicación entre Arduino y LabView.	90
Figura 56. Prueba de motores sin escobillas con Arduino.....	91
Figura 57. Calibración de Àngulo de giro del balancín.	91
Figura 58. Diseño de estructura.	92
Figura 59. Conexión de driver controlador de velocidad del motor sin escobillas. ..	92
Figura 60. Montaje de motores y cableado eléctrico	93
Figura 61. Primeras pruebas de funcionamiento de la planta	93

ÍNDICE DE TABLAS

Tabla I Parámetros de sintonización de Z.N para el eje Y.....	11
Tabla II. Parámetros de sintonización de Z.N para el eje X.....	12
Tabla III Tabla con los valores del PID en el eje Y.	28
Tabla IV. Tabla con los valores del PID en el eje X.	30

INTRODUCCIÓN

Ecuador es un país en vía de desarrollo en donde su industria no es ajena a esta evolución. Los equipos empleados en las industrias son muy determinantes para la ejecución de las tareas de cada sector. Un aliado de las industrias es establecer mecanismos que ayuden a optimizar el trabajo de las máquinas para poder realizar los procedimientos industriales de manera eficaz reduciendo de esta forma la presencia de errores.

Los controladores PID son una herramienta fundamental para reducir errores en los sistemas automatizados permitiendo mejores resultados para los procesos industriales.

Un controlador PID ayuda a controlar un proceso o variable, automáticamente sin la necesidad de utilizar mano de obra humana ayudando a reducir costos a largo plazo y mejor capacidad de producción con una mejor calidad.

El presente proyecto tiene por objeto la implementación de un balancín de dos grados de libertad implementado con un controlador PID lo que permite reforzar los conocimientos adquiridos en las materias de Teoría de Control y proponer casos prácticos y reales de los diferentes tipos de conflictos existentes en la industria para optimizar los procesos y así mejorar la producción.

Además, se realiza una interfaz gráfica de usuario didáctica y de fácil uso para poder analizar los diferentes valores obtenidos por el Hardware del balancín para su mejor análisis y entendimiento permitiendo tomar las decisiones correctas para optimizar procesos.

1. EL PROBLEMA

1.1. Planteamiento del problema.

En la actualidad las empresas necesitan profesionales con sólidos conocimientos y destrezas en programación de tarjetas microcontroladoras y manejo de softwares de ingeniería, debido a que las industrias cada día buscan la optimización y la automatización de los procesos de producción.

Por tal motivo, se debe desarrollar destrezas para rediseñar procesos controlados. Con un módulo controlado por software se pueden fortalecer esas destrezas en el desarrollo de diseño de controladores PID y en el manejo de software, como Matlab o LabVIEW, donde se puede analizar, experimentar y comparar los conocimientos teóricos adquiridos en clases, y obtener mayores destrezas con la práctica.

1.2. Importancia y alcance

El control automático con el pasar de los años se ha convertido en una parte fundamental en el avance de la ciencia y de la ingeniería, ya que están presentes en sistemas electrónicos aéreos, terrestres y navales, robots con inteligencia artificial, o de cualquier otro tipo de control inteligente u automático. (Ogata, 2010)

El presente trabajo de titulación trata de la interacción entre una interfaz de usuario en LabVIEW y el balancín con dos grados de libertad de bajo costo.

Este proyecto va dirigido a los estudiantes de la carrera de ingeniería electrónica y a su vez a los docentes, para que se impartan clases didácticas.

1.3. Delimitación

Acorde a lo descrito en este documento, se especifica el lugar donde será desarrollado el proyecto de titulación y su ubicación geográfica.

1.3.1. Delimitación temporal

La planificación correspondiente al diseño, desarrollo e implementación del proyecto está delimitada al año 2020.

1.3.2. Delimitación espacial

El proyecto de titulación propuesto se desarrolló en el laboratorio de electrónica y automatización, de la Universidad Politécnica Salesiana sede Guayaquil, la cual se encuentra ubicada en las calles, Chambers 227 y 5 de junio del cantón Guayaquil de la provincia del Guayas

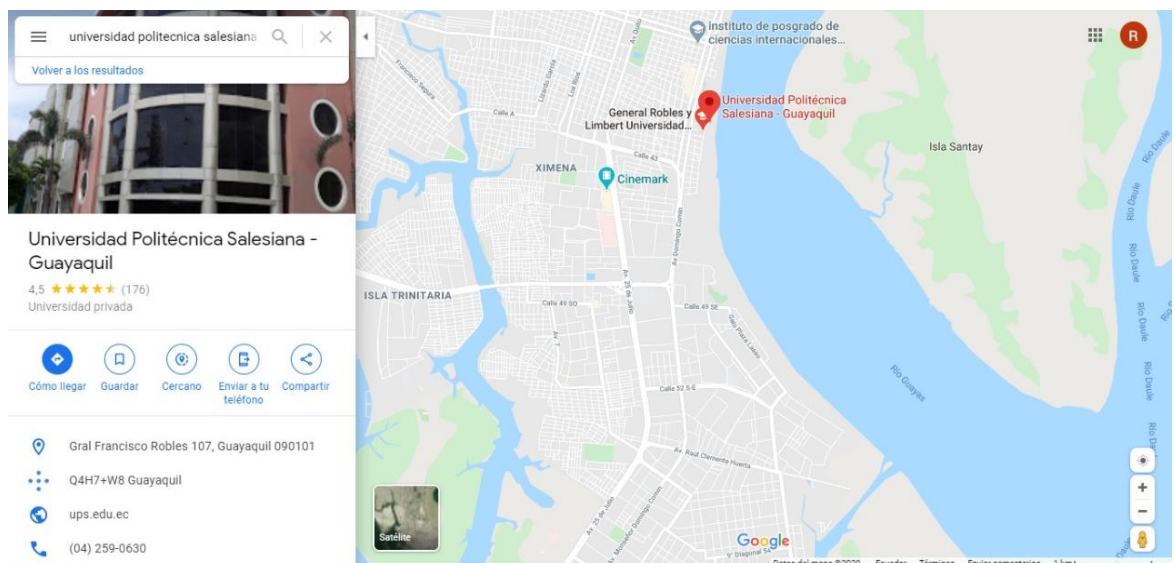


Figura 1. Ubicación geográfica de la Universidad Politécnica Salesiana

1.3.3. Delimitación social

El proyecto de titulación propuesto beneficia, a estudiantes de la Universidad Politécnica Salesiana sede Guayaquil, está diseñado para ser: desmontable, asequible y económico.

1.3.4. Delimitación tecnológica.

El presente proyecto de titulación incorpora lineamientos actuales para el desarrollo de conocimientos teóricos prácticos básicos e indispensables muy

utilizados en la industria, que permitirán a los futuros profesionales una verdadera preparación en materia de control y automatización de procesos.

Dentro del plano económico este proyecto de titulación implementa una serie de sensores de medición angular de bajo costo, se utiliza la tarjeta electrónica Arduino Mega para la comunicación entre la interfaz gráfica y la planta, se usan dos motores para el posicionamiento de los ejes X, Y. Debido al excesivo costo de un Driver (variador de velocidad) cuya función principal es realizar el cambio de giro de un motor.

1.3.5. Impacto

Por medio de la implementación del Balancín con 2 grados de libertad se incentiva las destrezas cognitivas de los estudiantes de la facultad de ingeniería electrónica mejorando los conocimientos en el área industrial y las destrezas de la comunidad universitaria para así competir en el ámbito laboral con conocimientos más sólidos en control y automatización específicamente en modelos de PID por medio de la realización de las prácticas de laboratorio.

1.4. Justificación

Este proyecto tiene por justificación reforzar los conocimientos teóricos de control automático aplicados en la ingeniería adquiridos en el aula de clase, este proyecto es de vital ayuda para los estudiantes, ya que mediante la elaboración de prácticas experimentan y analizan los datos obtenidos de una planta real con software y hardware usados actualmente en las industrias para el análisis de una planta, permitiendo así adquirir nuevas destrezas para mejorar no sólo en el ámbito académico sino también en el campo laboral.

1.5. Grupo objetivo

Los beneficiarios de este proyecto son los estudiantes que cursen las asignaturas de Teoría de control y Electiva1 de la Universidad Politécnica Salesiana sede Guayaquil, cuya finalidad es complementar lo adquirido teóricamente en clases.

El objetivo de las prácticas es desarrollar en los estudiantes el aprendizaje técnico práctico del control automático aplicado en la ingeniería.

1.6. Explicación del problema.

Este proyecto tiene como prioridad reforzar los conocimientos teóricos de control automático adquiridos en clases, por medio de la realización de un prototipo Balancín de dos grados de libertad, al cual se implementa un controlador PID permitiendo impulsar el desarrollo de algoritmos, el análisis de sistemas, control de procesos mediante interfaz gráfica y sintonización de controladores PID, ya que estas competencias se necesitan en el entorno profesional dentro de la industria ecuatoriana.

1.7. Objetivos

1.7.1. Objetivo general

Diseñar e implementar controlador PID para un balancín de dos grados de libertad.

1.7.2. Objetivos específicos

- Diseñar un algoritmo para el control de estabilidad del sistema.
- Diseñar la interfaz gráfica.
- Obtener la función de transferencia del sistema mediante software.
- Diseñar e implementar un control para un balancín de dos grados de libertad.
- Elaborar un manual de mantenimiento y operación del módulo de control

2. FUNDAMENTOS TEÓRICOS

Han pasado más de dos décadas de avances y desarrollos tecnológicos sobre métodos de control e implementación de controladores PID, la mayoría de los artículos sobre el uso de motores sin escobillas “Brushless”, aún se centran únicamente en el funcionamiento del motor, proveyendo solo una explicación básica, mínima y elemental de control. (Krishnan, 2010).

Los avances tecnológicos con controladores PID tienen como objetivo fomentar e impulsar el uso del mismo, en todas las áreas referentes a control, el control PID en el siglo actual es de gran interés para todo académico especializado con PID y un estímulo para futuras investigaciones para los profesionales, con misión de remover problemas aplicando controladores PID. (Ramon Vilanova, 2012)

2.1. Conceptos básicos en el sistema de control

2.1.1. Sistema

Conjunto de componentes que interactúan entre sí para realizar un determinado objetivo. (Ogata K. , Ingeniería de control moderno, 2010)

2.1.2. Perturbación

Es una señal que tiende a afectar la señal de salida de un sistema negativamente. (Ogata K. , Ingeniería de control moderno, 2010)

2.1.3. Control realimentado

Operación que tiende a reducir la diferencia entre la salida de un sistema y el valor de entrada de referencia. (Ogata K. , Ingeniería de control moderno, 2010)

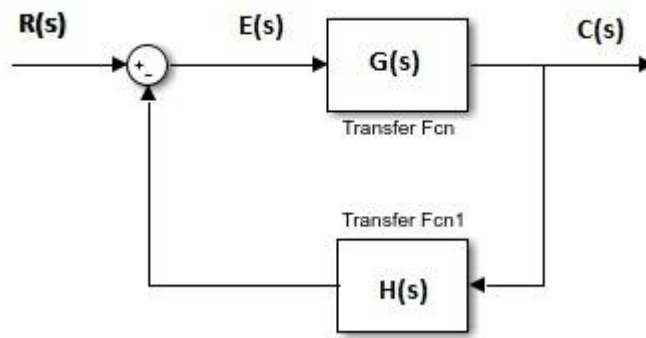


Figura 2. Diagrama de control realimentado. (Ogata K. , 2010)

2.1.4. Entrada

Señal que ingresa a un sistema, puede ser referencial o perturbación.

2.1.5. Salida

Señal resultante del proceso de control en el sistema.

2.1.6. Controlador PID

Algoritmo de control compuesto de tres elementos que nos proporcionan acción: proporcional, integral y derivativa, que permite controlar a los sistemas realimentados para alcanzar la estabilidad deseada de un sistema. (Ogata K. , Sistema de control en tiempo discreto, 2010).

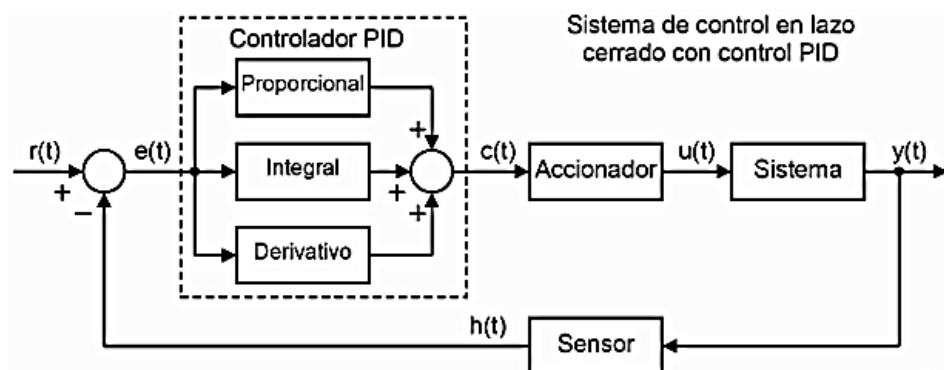


Figura 3. Control proporcional integral derivativo. (PID) (Sánchez, 2012)

2.1.7. Control Proporcional

Algoritmo de control que proporciona una mayor velocidad de respuesta en el sistema, disminuyendo el error en el sistema, la señal de error es proporcional a la acción de control. (Sánchez, 2012)

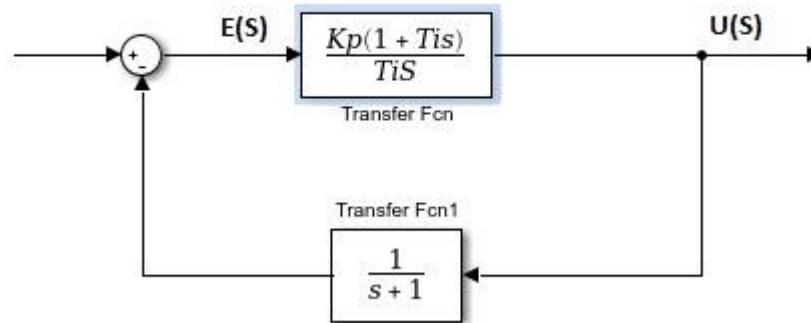


Figura 4. Diagrama de control proporcional. (P) (Sánchez, 2012).

2.1.8. Control Derivativo

Algoritmo de control que proporciona aumento en la estabilidad del sistema, amplificando la señal de salida cuando la respuesta es muy inestable (oscila demasiado), este control reduce ruidos y perturbaciones, es una derivada la relación entre la señal de salida y la entrada. (Sánchez, 2012)

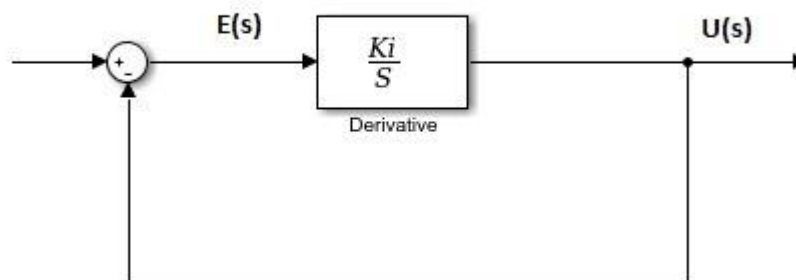


Figura 5. Diagrama de control derivativo (D) (Sánchez, 2012)

2.1.9. Control Integral

Algoritmo de control que proporciona inercia al sistema reduciendo el error con la rapidez deseada, es una integral la relación entre la señal de salida y la entrada. (Sánchez, 2012)

2.2. Motores sin escobilla

Motor síncrono constituido por un rotor de imanes permanentes y un estator bobinado, los campos magnéticos generados por el estator y por el rotor giran a la misma frecuencia. (Padmaraja Yedamale, Microchip Technology Inc., 2003)



Figura 6. Motor sin escobilla (Padmaraja Yedamale, Microchip Technology Inc., 2003)

2.3. Tarjeta Arduino Mega2560

Placa microcontroladora que posee 54 salidas/entradas digitales, basada en el microcontrolador Atmega2560 con conexión USB, ICSP. (Arduino corp., 2020)



Figura 7. Tarjeta Arduino Mega 2560. (Arduino corp., 2020)

2.4. Software LabVIEW

Software de ingeniería diseñado para aplicaciones de control y medida, ofrece un enfoque de programación gráfica útil para desarrollar algoritmos de análisis de datos y diseño de interfaces de usuario personalizadas. (National Instruments, 2020)

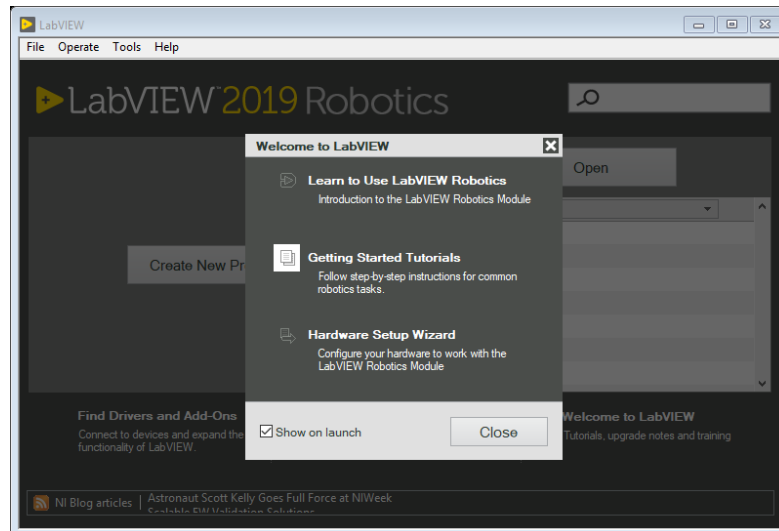


Figura 8. Plataforma LabVIEW (National Instruments, 2020)

2.5. Software Matlab

Software de ingeniería diseñado para análisis, procesos, programación de arreglos y matrices, con tool-box que permiten obtener resultados deseados del funcionamiento de diferentes algoritmos con capacidad de escalar análisis para la ejecución de clústeres y manejo de datos para la automatización de un trabajo. (Mathworks, 2020)

2.6. Método Ziegler-Nichols

El método Ziegler-Nichols permite desarrollar o ajustar de manera empírica un controlador PID, sin la necesidad de tener el modelado matemático del sistema. El método propone conseguir un sobre pulso máximo del 25%, valor de buena característica de estabilidad y rapidez en la mayoría de los sistemas retro alimentado ante una respuesta escalón, permite obtener parámetros de ganancias: proporcionales, integrales y derivativas, a partir de la respuesta de un sistema en lazo abierto o lazo cerrado. (Pardo, 2013)

2.7. Método Ziegler-Nichols en lazo cerrado

Método que se usa en sistemas que pueden tener sostenidas oscilaciones, elimina los efectos de la parte integral y la parte derivativa, usa únicamente la ganancia k_p para obtener oscilaciones sostenidas, la ganancia obtenida en el sistema

de oscilación sostenida se llama ganancia crítica k_{cr} , que es obtenida con un tiempo crítico t_{cr} . (Controladores PID).

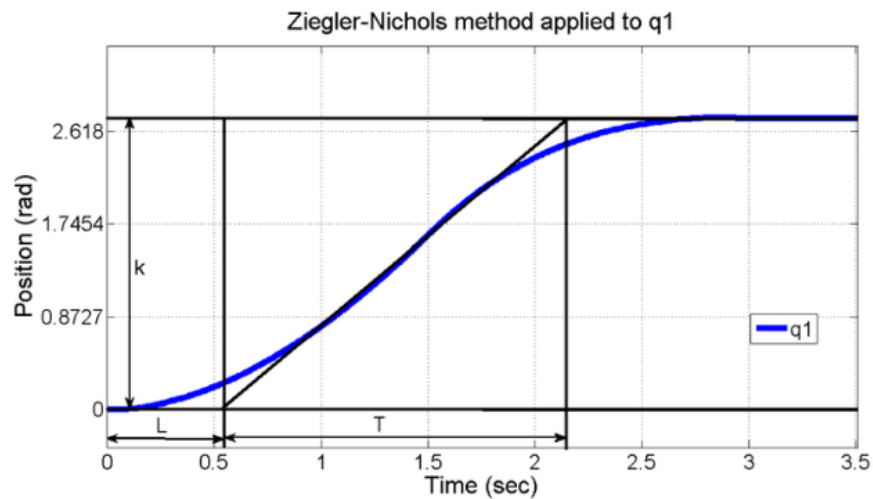


Figura 9. Lazo cerrado del método Ziegler-Nichols. (Garcia, 2020)

Tipo de controlador	K_p	τ_i	τ_d
P	$0.5K_{cr}$	∞	0
PI	$0.45K_{cr}$	$\frac{1}{2}P_{cr}$	0
PID	$0.6K_{cr}$	$0.5P_{cr}$	$0.125P_{cr}$

Figura 10. Tabla de sintonía de Ziegler-Nichols en lazo cerrado. (Departamento de Control, División de Ingeniería Eléctrica Facultad de Ingeniería UNAM., 2006)

Con los parámetros de Ziegler-Nichols, se determina los valores para el controlador PID que regirá el sistema para los movimientos de la planta en el eje X y en el eje Y.

Tabla I

Parámetros de sintonización de Ziegler-Nichols para el eje Y.

Parámetro para Y	Kc	Ti	Td
P	2,425	0	0
PI	2,18	0,067	0
PID	2,91	0,04	0,01

Tabla II.

Parámetros de sintonización de Ziegler-Nichols para el eje X.

Parámetro para x	Kc	Ti	Td
P	1,50	0	0
PI	1,35	0,011	0
PID	1,80	0,067	0,0167

3. MARCO METODOLÓGICO

En esta sección se detalla cómo se realizó el algoritmo de control que permite controlar los motores del balancín, así como el diseño de su estructura, almacenamiento de datos y toda infraestructura electrónica involucrada para que el proyecto de titulación funcione de acuerdo a lo propuesto.

3.1. Diseño mecánico del balancín de dos grados de libertad

La planta como se observa en la siguiente figura, está compuesta de dos partes: una parte fija y una parte móvil.



Figura 11. Balancín de 2 grados de libertad

3.1.1. Parte fija

La parte fija está compuesta de un tubo de acero inoxidable cuadrado hueco de 5cm de lado por 18cm de largo, en la parte superior encontramos una base cuadrada de 5cm por lado donde se encuentra una rodadera la cual sirve de pivote para la base cilíndrica donde está fijada la parte móvil, la parte inferior está anclada a una caja rectangular de 25 cm de largo por 20 cm de ancho con una altura de 3cm donde se encuentran todos los dispositivos electrónicos que sirven para el control del sistema.

3.1.2. Parte móvil

La parte móvil está compuesta de un tubo de aluminio cuadrado hueco de 2 cm de lado por 30 cm de largo, perforado en el centro por un pasador que permite el movimiento en el eje Y. Anclado en la rodadera que esta fija en la base cilíndrica de 2,5 cm de radio fijada en la base de la parte superior, en los dos extremos del tubo de la parte móvil están adaptadas las bases que sirven para fijar los motores sin escobillas, los cuales permiten el movimiento sobre el eje X y el eje Y.

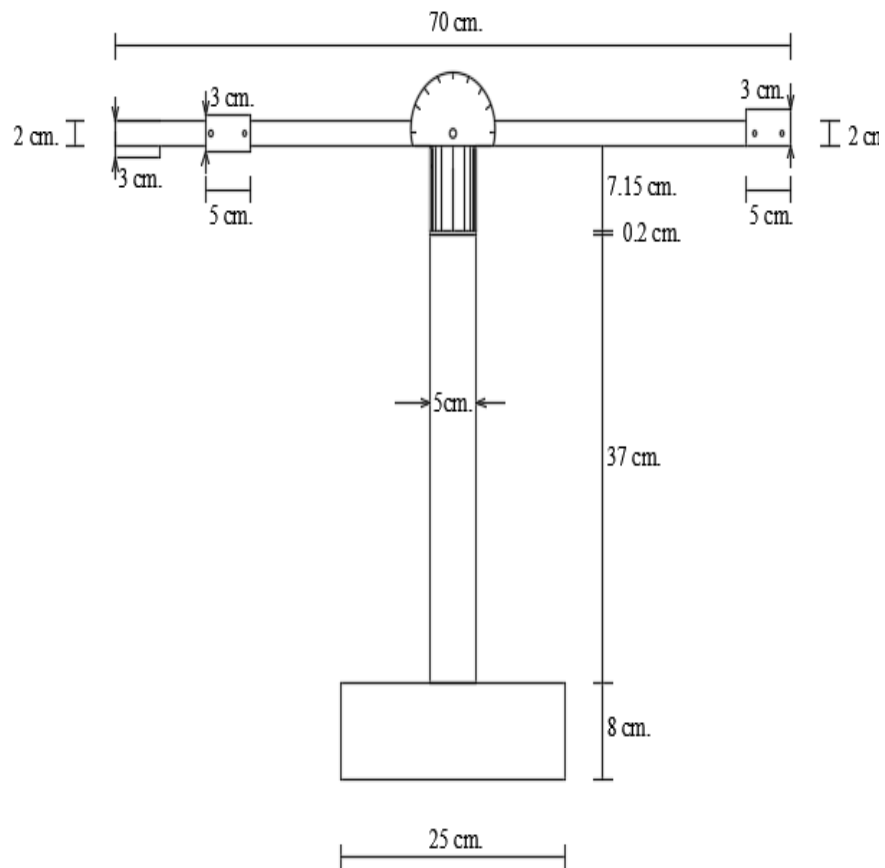


Figura 12. Diagrama del Balancín de dos grados de libertad.

3.2. Diseño del algoritmo.

3.2.1. Comunicación del Arduino con LabVIEW.

Los bloques que se usan para habilitar la comunicación entre la placa de pruebas Arduino y LabVIEW se muestran a continuación.

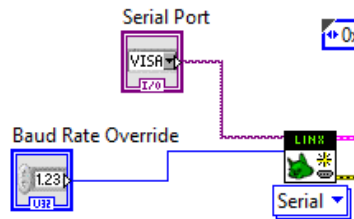


Figura 13. Bloque de comunicación entre el Arduino y LabVIEW.

- **Baud rate:** bloque donde se establece la velocidad en baud/s, cuyo valor debe ser 9600.
- **Serial Port:** bloque que se usa para seleccionar el número de puerto por el cual el Arduino se comunica con la interfaz gráfica de LabVIEW.

3.2.2. Control de en LabVIEW.

Los bloques que se usan para el control de ancho de pulso de cada motor y el detalle de cada conexión se observan en la siguiente figura.

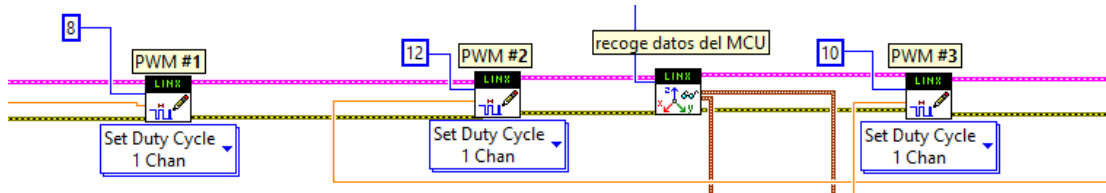


Figura 14. Bloques para el control de ancho de pulso de los 3 motores

- **PWM#1:** bloque en el cual se imprime el ciclo de trabajo del motor 1.
- **PWM#2:** bloque en el cual se imprime el ciclo de trabajo del motor 2
- **PWM#3:** bloque en el cual se imprime el ciclo de trabajo del motor 3

Todos los módulos del Arduino están conectados en serie con la siguiente identificación:

- **Cable rosado:** se usa para la comunicación serial entre cada módulo.
- **Cable amarillo:** se usa para la seguridad interna de la comunicación entre cada módulo, si hay un error en cualquier módulo, el error afecta todo y detiene la comunicación.

3.2.3. Lectura de encoder.

Bloque que permite la lectura del voltaje que genera el sensor al mover la barra móvil, se usa la entrada analógica del Arduino para la lectura de los datos.

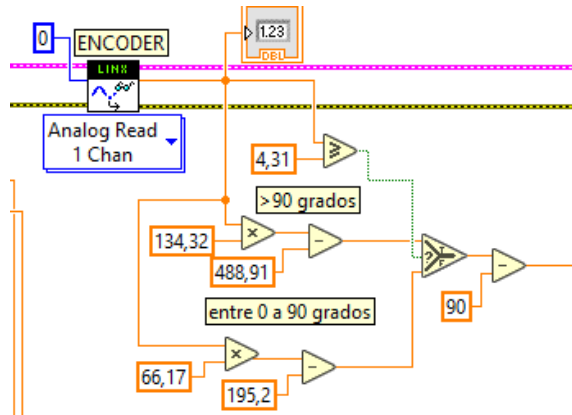


Figura 15. Bloque para lectura de encoder.

3.2.4. Control manual-automático.

Bloque que posee 2 funciones: escoger el control sobre los motores, ya sea de forma manual ingresando el ancho de pulso en el bloque "SET DUTY CYCLE (0-1)", o que el motor reciba los valores que imprime el control PID de manera automática.

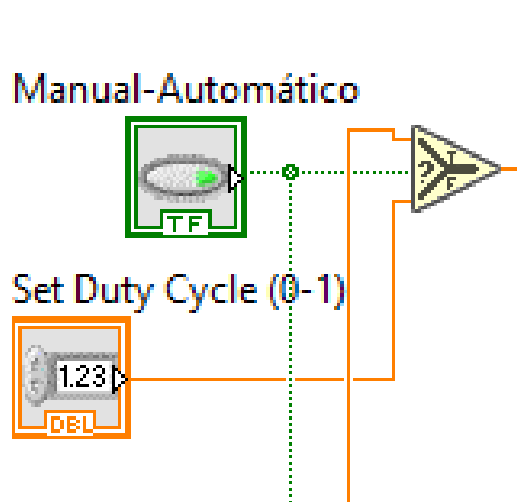


Figura 16. Control manual-automático de la planta.

3.2.5. Control PID

En el módulo PID se realizan los cálculos matemáticos, al cual le ingresan las señales: set point, output range, PID gains.

- **Set point:** constante que se desea tener como valor para aproximar una señal.
- **Output range:** valores del rango de salida, del cual se desea que el PID tome como límites operacionales (superior e inferior).
- **PID gains:** valores que se ingresan, deben tener la constante proporcional, derivativa e integral.

El módulo “SP, VP, PID”, bloque que gráfica las variables (set point, output range y PID gains), en un solo gráfico, los otros bloques son módulos indicadores de los valores ingresados.

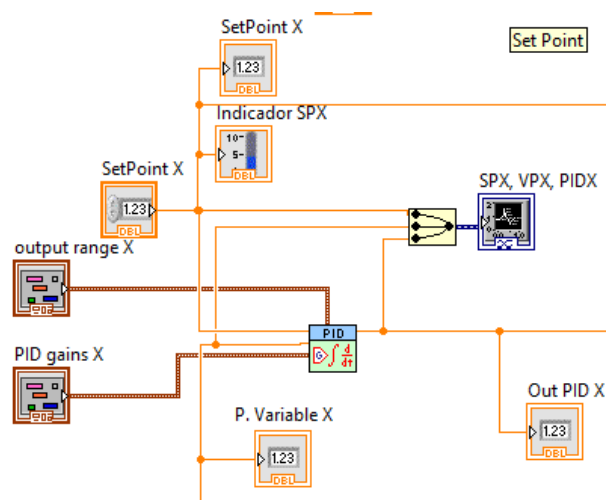


Figura 17. Control PID de la planta

3.2.6. Recopilación de datos.

En la figura 18 se observan los módulos que se usan para guardar los datos generados en el sistema.

- **Write to measurement:** en este módulo se ingresan valores dinámicos, ya sea en escalar o en matriz de 1x2
- **Set dynamic data attributes:** en este módulo se transforma un valor double en un valor dinámico, en el cual se puede poner el nombre de la columna.

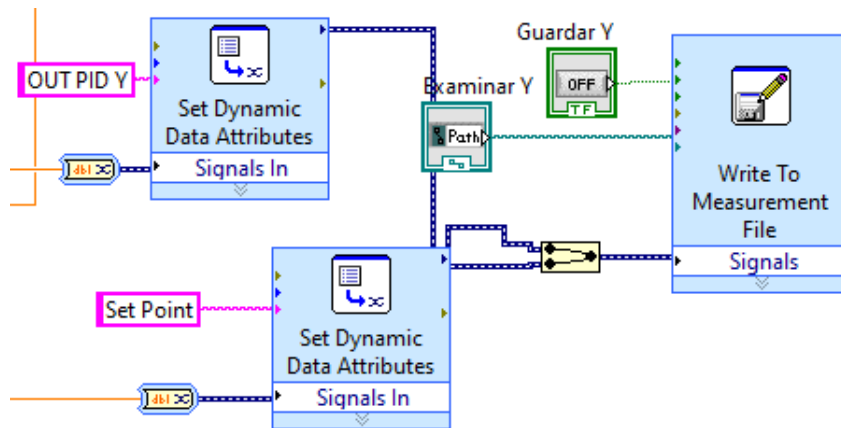


Figura 18. Bloque para guardar los datos en archivo de Excel.

3.3. Diseño del circuito electrónico

A continuación, se muestran las conexiones eléctricas entre Arduino, motores sin escobillas y fuente de alimentación.

Para poder energizar este proyecto se utilizó una fuente conmutada de 12V DC 5 A, a la cual a los terminales de salida de voltaje se suelda un fusible de 5A como medida de seguridad para proteger motores y la placa de pruebas Arduino ya que son partes vitales de este proyecto.

El circuito electrónico en general se fragmenta en dos partes para una fácil conexión, los motores 1, 2 y 3 se conectan a sus respectivos drivers 1, 2 y 3, tal cual como se observa en la figura 20.

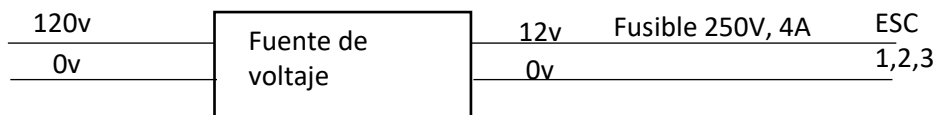


Figura 19. Conexión de la fuente de voltaje a 120v

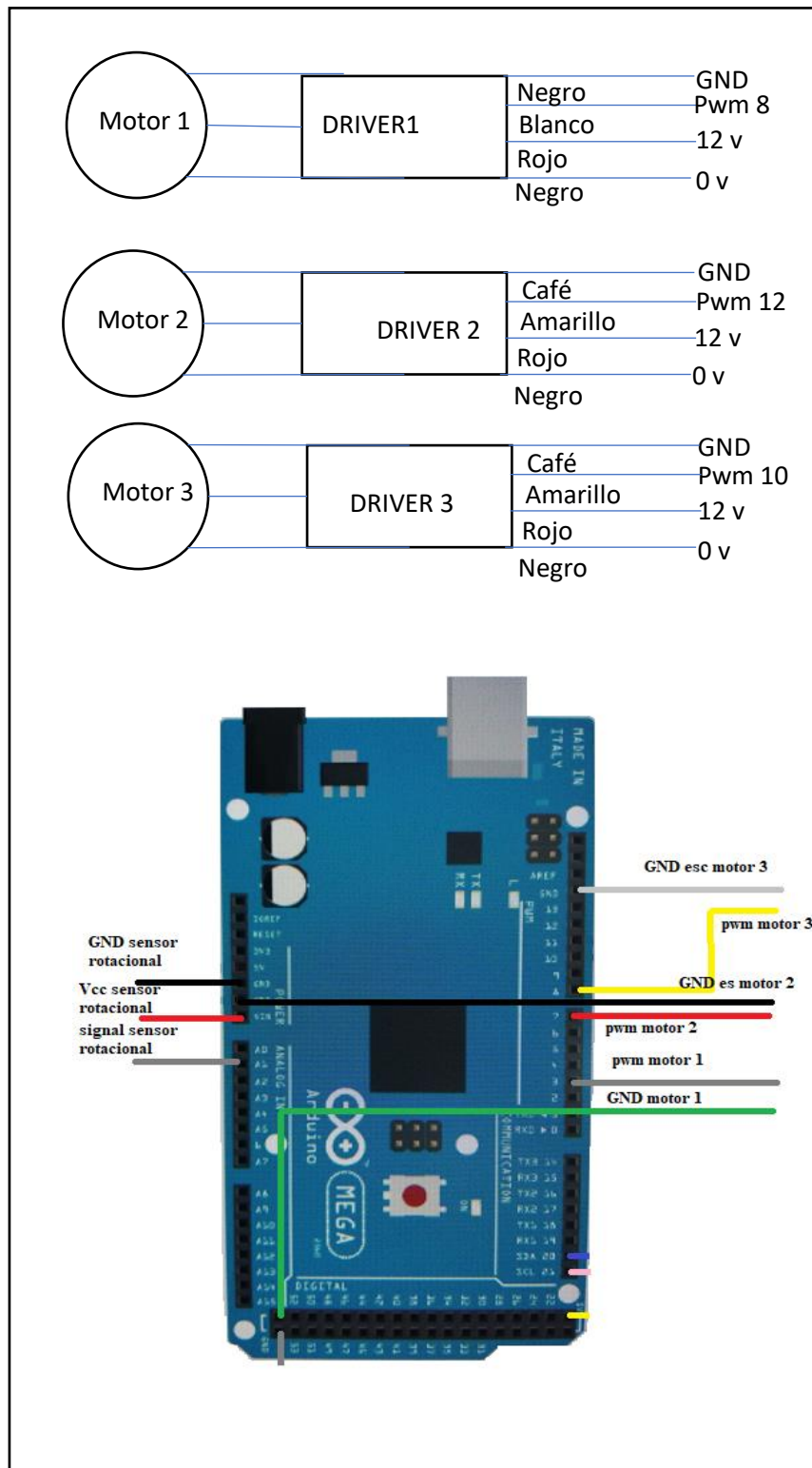


Figura 20. Conexión de los motores y drivers con el Arduino

3.4. Diseño del diagrama de bloques en LabVIEW

En el siguiente apartado se muestran los elementos más relevantes del algoritmo de control basado en diagramas de bloques, creado para controlar un Balancín de dos grados de libertad.

Para poder realizar la comunicación bidireccional entre Arduino y PC se conectan los módulos de Open y Close en una estructura de While Loop, en el módulo open se crea un control para el serial Port encargado de controlar por cual puerto se realiza la comunicación entre el Arduino y el algoritmo en LabVIEW, se crea un control Baud Rate Override encargado de controlar la velocidad de transmisión de datos, en el módulo close se crea un indicador error out el cual indica que tipo de errores se están desarrollando en la transmisión de datos.

Para el envío y recepción de datos, se usan los módulos de LINX de la librería de MakerHub.

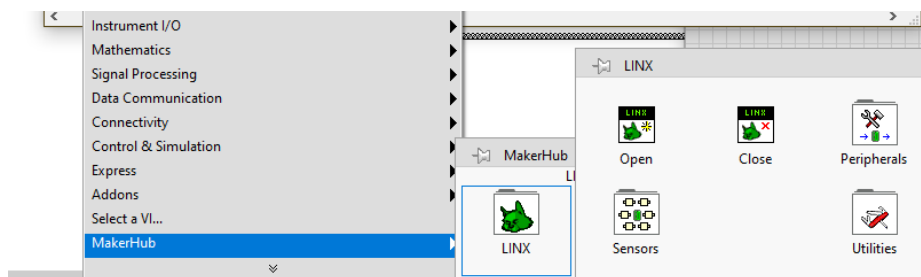


Figura 21. Librería de MakerHub.

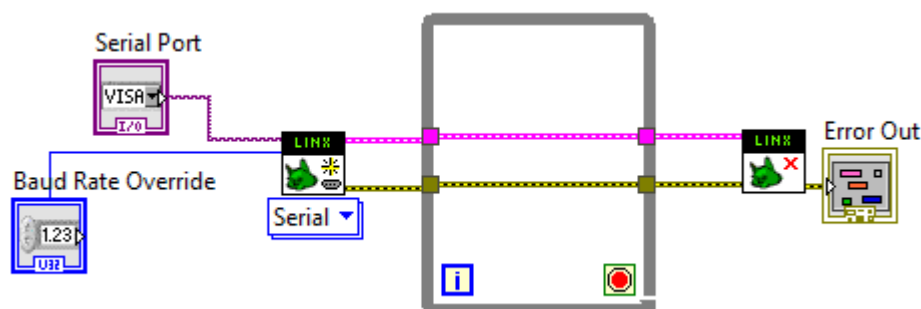


Figura 22. While Loop para el envío y recepción de datos desde el Arduino.

Para el control de los motores dentro del While Loop se coloca los módulos "Set duty cycle", los cuales sirven para transmitir el ancho de pulso en cada motor, en

cada módulo de “Set duty cycle” se crea una constante con el número que indica a que pin en el Arduino se conecta cada motor.

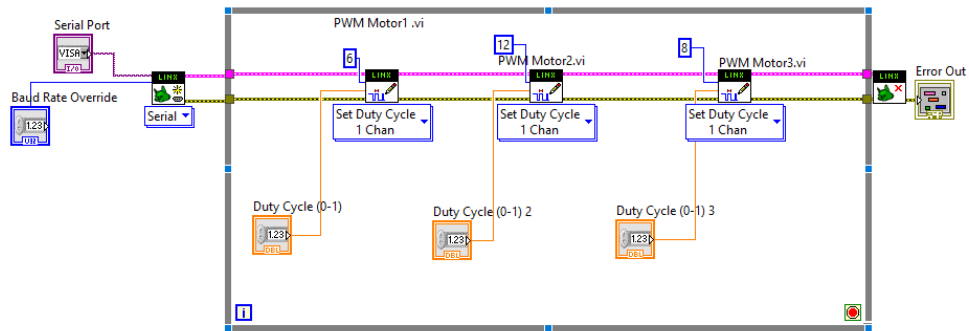


Figura 23. Conexión de los módulos de control de ancho de pulso para cada motor.

En la siguiente figura se observa los diferentes bloques con operación matemática que se encargan de convertir los datos obtenidos por el sensor rotacional en grados de posición del balancín.

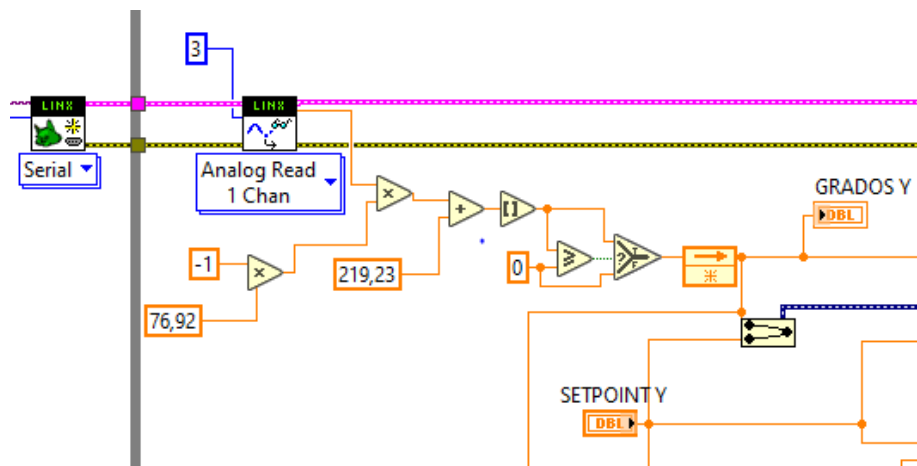


Figura 24. Conversión en grados de los datos obtenidos del sensor rotacional en Y

Para el diseño del controlador PID que regula el movimiento en el eje Y del balancín, la conexión se detalla en la figura 25, tiene como entradas: set point Y, output range Y, variable process y PID gains, tiene una única salida que se conecta a la entrada de ancho de pulso del motor 1.

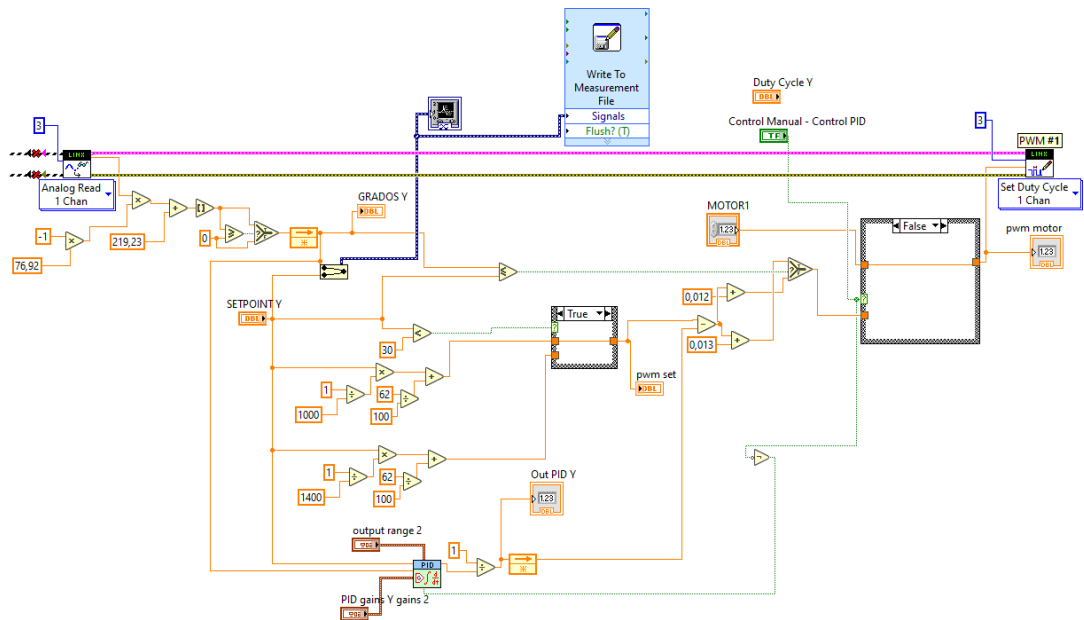


Figura 25. Conexión del control PID para el movimiento en el eje Y del sistema

El diseño del controlador PID que regula el movimiento en el eje X del balancín, la conexión se detalla en la figura 26, tiene como entradas: set point X, output range X, variable process y PID gains, tiene una única salida que se conecta a un Estructure Case que dependiendo del punto de ajuste que se desea, envía la señal de ancho de pulso al motor 2 y motor 3.

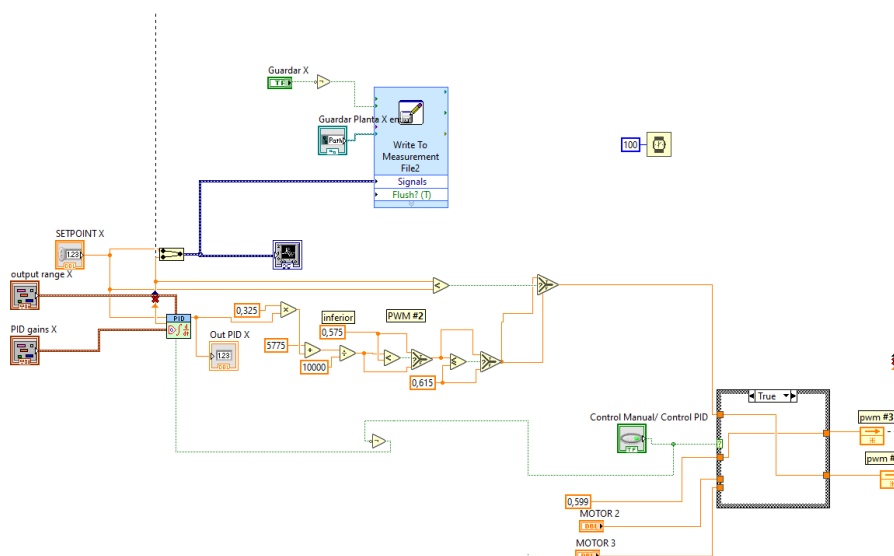


Figura 26. Conexión del control PID para el movimiento en el eje X del sistema.

La salida de ambos controladores PID tanto como de X y Y están conectados a un indicador que gráfica la señal de salida en un determinado tiempo de muestreo, a su vez se conectan a un módulo Write to Measurement File para el almacenamiento de datos generados a la salida de cada controlador PID.

3.5. Diseño de interfaz gráfica en LabView

Para el diseño de la interfaz gráfica se consideró seccionar en diferentes partes las funcionalidades que tendrá la interfaz gráfica, ya que se necesita tener modos de operación manuales y automáticos para el control de los motores, inicialización de motores, gráfica de datos y almacenamiento.

3.5.1. Interfaz gráfica modo de operación

Cada motor funciona con un ESC con las mismas especificaciones, pero funcionan con diferente ancho de pulso, se establecen dos modos de operación, manual y automático es este modo interviene el control por medio de un PID para ambos modos los motores deben ser inicializados, la inicialización se realiza de la siguiente manera:

- Se realiza el esquema de control manual para los motores.
- Para la operación de cada motor, en la interfaz gráfica se aprecia los 3 módulos que sirven para la operación manual de cada motor, con límite inferior de cero y límite superior de 0,8 de ancho de pulso, debido a que si el ancho de pulso supera los 0,8 podría afectar a los motores.
- Para inicializar los motores, con el módulo selector se manipula hasta aproximar el valor de ancho de pulso a 0,6 o hasta escuchar los sonidos consecutivos de inicialización de los motores.
- Después de inicializar el motor, regreso al valor 0 de ancho de pulso.
- Con el mouse se manipula los valores de ancho de pulso para el modo manual.

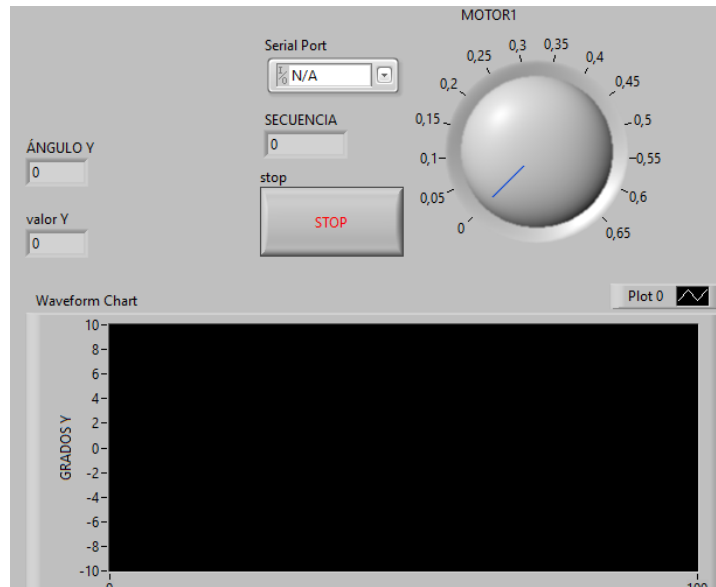


Figura 27. Panel frontal para la inicialización del motor 1

En cada uno de las secciones para la inicialización de motores se puede visualizar de forma gráfica el comportamiento de cada motor y adicional a esto se pueden gestionar manualmente con el controlador en forma de perilla.

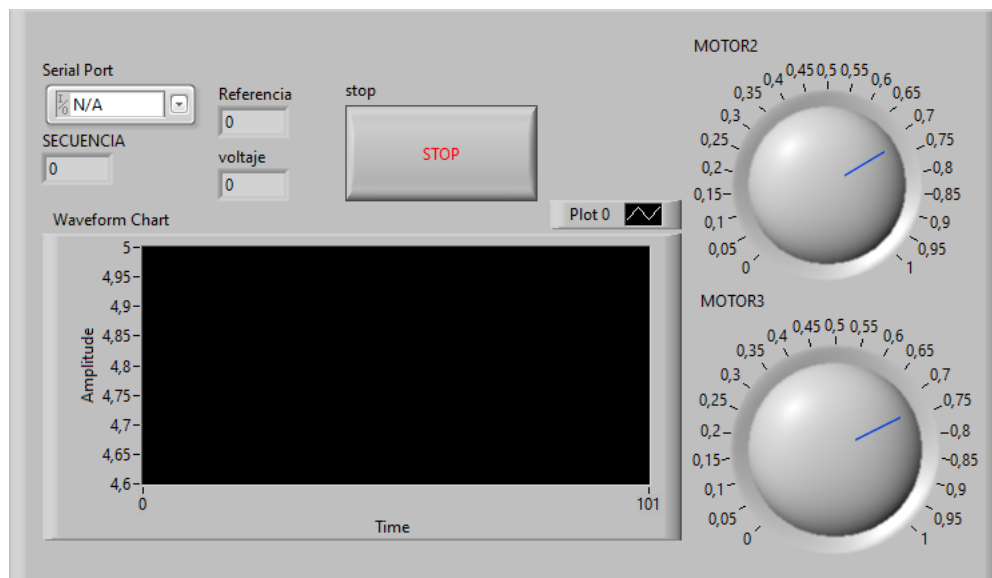


Figura 28. Control manual de ancho de pulso para inicializar los motores 2-3.

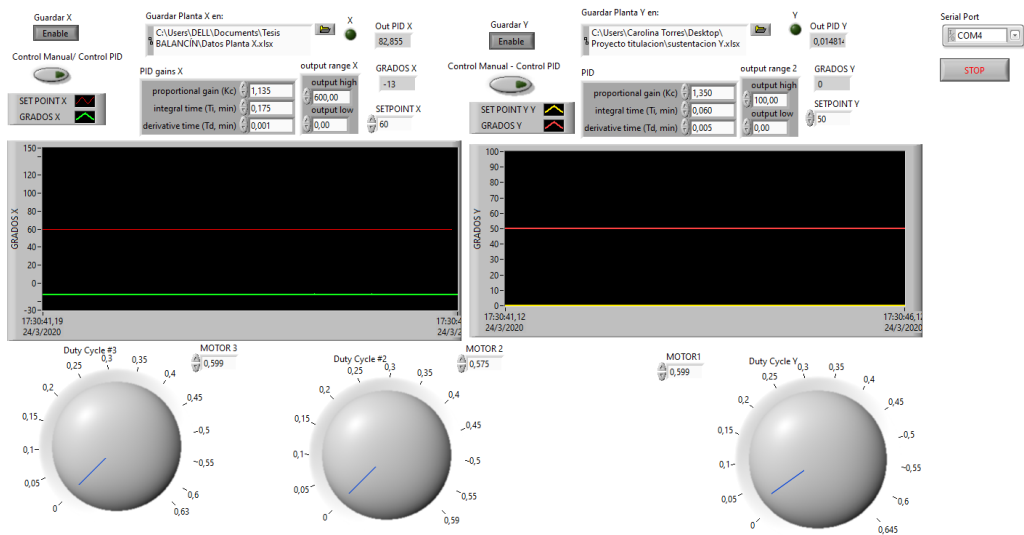


Figura 29. Panel frontal para el control manual en el eje en X y el eje en Y

Esta sección del panel principal se encarga del almacenamiento de los datos y creación de un archivo de Excel para poder graficar los valores obtenidos y hacer comparación del comportamiento de la planta.

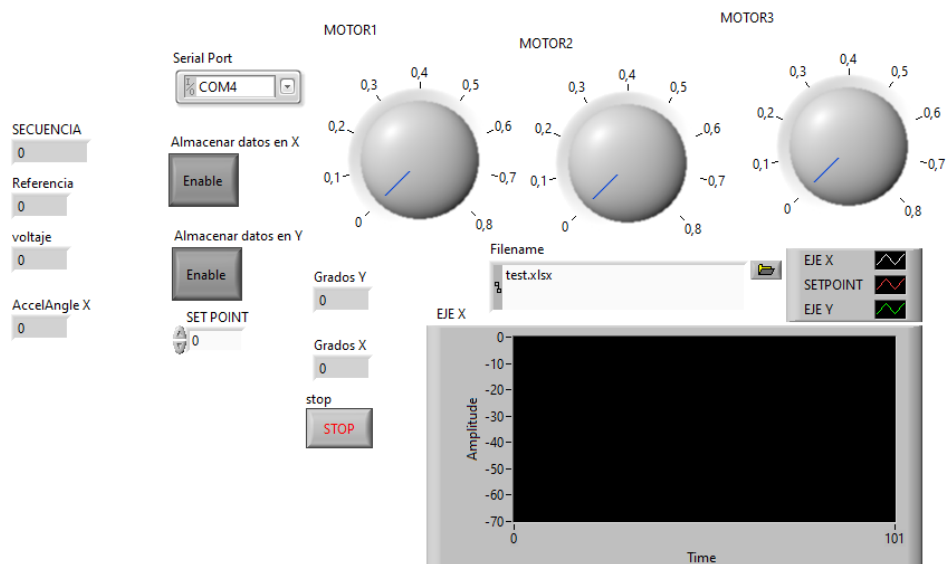


Figura 30. Panel frontal para la obtención y almacenamiento de datos.

4. ANÁLISIS E INTERPRETACIÓN DE LOS RESULTADOS

4.1. Análisis de los parámetros PID por Ziegler-Nichols en el eje Y.

Se aplica una señal escalón a la planta en lazo abierto, en modo manual se ingresaron los valores de ancho de pulso para colocar al balancín en 60 grados en el eje y.

Mediante el diagrama de control se toman valores del sistema y se almacenan los datos de la planta en lazo abierto en un archivo de Excel de nombre, DATOS ZIEGGER NICHOLS.XLSX. En el archivo Excel se guarda los valores obtenidos del posicionamiento en grados de la planta en el eje Y, con los datos obtenidos, se desarrolla una gráfica en Excel donde se aprecia el comportamiento de la planta en respuesta a la señal escalón.

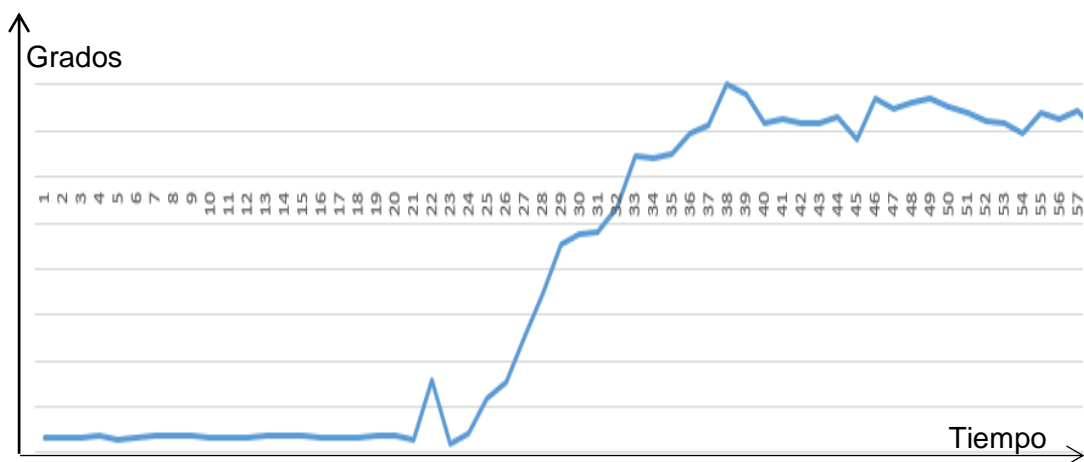


Figura 31. Gráfica que se usa para el método de Ziegler-Nichols.

Se observa en la figura 35 el comportamiento general del sistema y se identifica el tiempo muerto, tiempo de subida y se obtienen los valores de sintonización del PID.

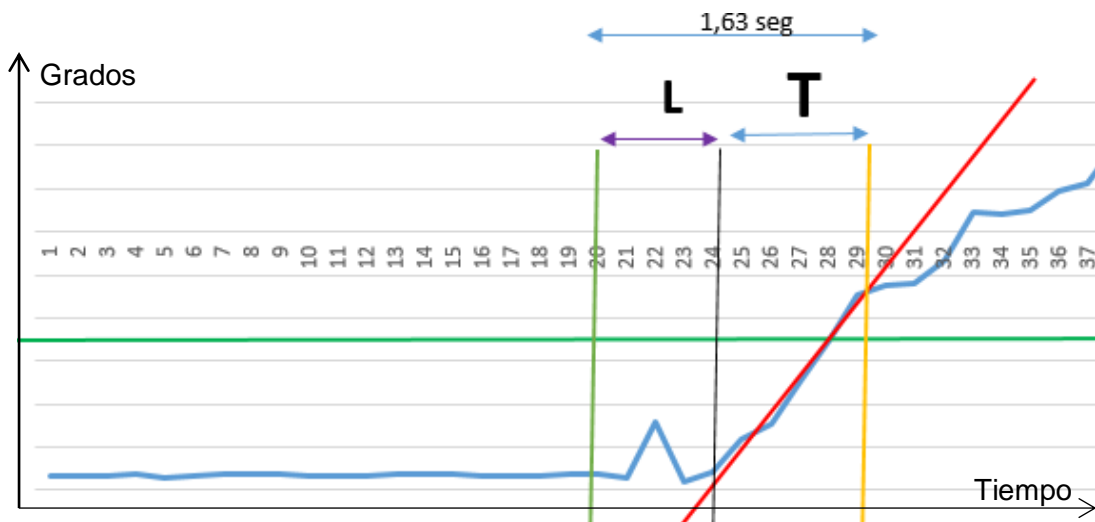


Figura 32. Gráfica con los valores de L y T para Ziegler-Nichols en el eje Y.

4.1.1. Obtención de los parámetros del PID en el eje Y.

Se calcula de forma teórica los parámetros de PID en el eje Y con el fin de compararlo con los valores que obtenemos de la planta.

$$L = \frac{\text{tiempo total} \times \text{iteracionesL}}{\text{total de iteraciones}}$$

$$L = \frac{1,63 \times 4}{10}$$

$$L = \frac{6,53}{10} = \underline{\underline{0,652 \text{ seg.}}}$$

$$T = \frac{\text{tiempo total} \times \text{iteracionesT}}{\text{total de iteraciones}}$$

$$T = \frac{1,63 \times 5,5}{10}$$

$$T = \frac{8,965}{10} = \underline{\underline{0,8965 \text{ seg.}}}$$

$$K_c = \frac{1,2 T}{L}$$

$$K_c = \frac{1,2 \times 0,8965}{0,63}$$

$$K_c = \underline{\underline{1,65}}$$

$$T_i = 2 \times L$$

$$T_i = 2 \times 0,652$$

$$T_i = \underline{1.304 \text{ seg.}}$$

$$T_d = 0,5 \times L$$

$$T_d = 0,5 \times 0,652$$

$$T_d = \underline{0,326 \text{ seg.}}$$

Los valores del tiempo se obtienen en segundos, para implementar en el controlador PID el tiempo se transforma en minutos.

$$T_i = 1.304 \text{ seg.} / 60$$

$$T_i = \underline{0,021 \text{ seg.}}$$

$$T_d = 0,326 \text{ seg.} / 60$$

$$T_d = \underline{0,005 \text{ seg.}}$$

Tabla III

Tabla con los valores del PID en el eje Y.

Parámetro para Y	Kc	Ti	Td
P	0,825	0	0
PI	0,7425	0,7470	0
PID paralelo	1,65	0,021	0,005

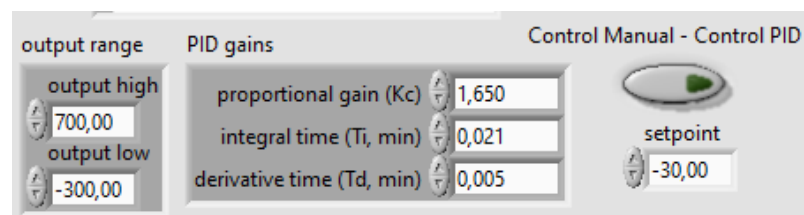


Figura 33. Controlador PID sintonizado con Ziegler-Nichols en el eje Y

4.2. Análisis de los parámetros PID por Ziegler-Nichols en el eje X.

Se aplica una señal escalón a la planta en lazo abierto, en modo manual se ingresa el valor de ancho de pulso para colocar al balancín en 60 grados en el eje X. Mediante el diagrama de control se toman los valores del sistema y se almacena los datos en un archivo de Excel, DATOS ZIEGGER NICHOLS X.XLSX.

Con los datos obtenidos, se desarrolla una gráfica en Excel para determinar los valores del PID en X. Con los datos obtenidos, se desarrolla una gráfica en Excel donde se aprecia el comportamiento de la planta en respuesta a la señal escalón.

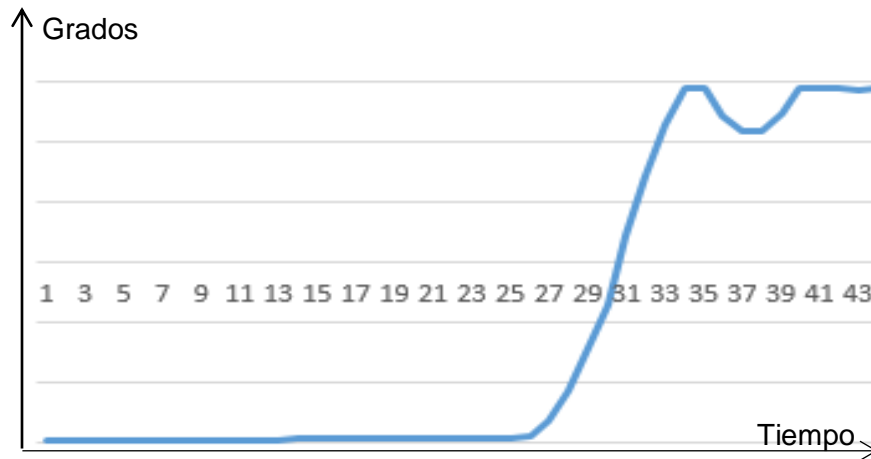


Figura 34. Gráfica de lazo abierto para la planta de la planta en el eje X

Se observa en la figura 31 el comportamiento general del sistema y se identifica el tiempo muerto, tiempo de subida y se obtiene los valores de sintonización del PID.

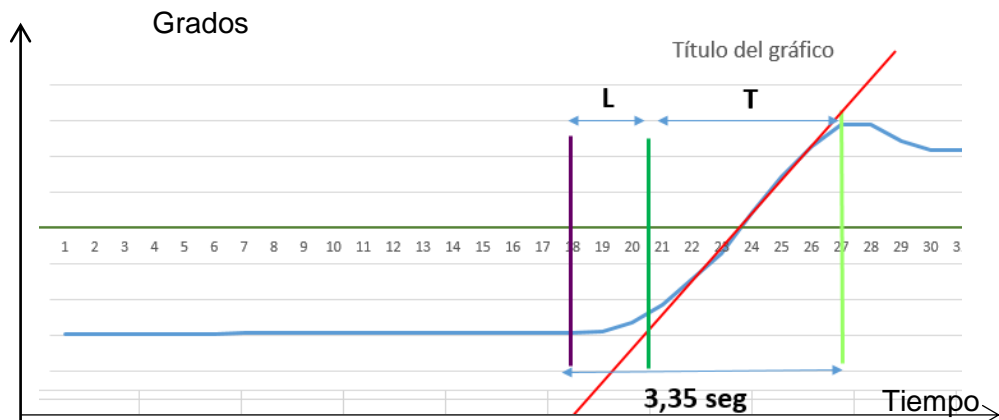


Figura 35. Gráfica con los valores de L y T para Ziegler-Nichols en el eje X

4.2.1. Obtención de los parámetros del PID en X

$$L = \frac{\text{tiempo total} \times \text{iteracionesL}}{\text{total de iteraciones}}$$

$$L = \frac{3,35 \times 2,5}{9}$$

$$L = \frac{8,375}{9} = \mathbf{0,930 \text{ seg.}}$$

$$T = \frac{\text{tiempo total} \times \text{iteraciones}}{\text{total de iteraciones}}$$

$$T = \frac{3,35 \times 6,5}{9}$$

$$T = \frac{21,775}{9} = \underline{\underline{2,41 \text{ seg.}}}$$

$$Kc = \frac{1.2 T}{L}$$

$$Kc = \frac{1.2 \times 2,41}{0,930}$$

$$Kc = \underline{\underline{3,109}}$$

$$Ti = 2 \times L$$

$$Ti = 2 \times 0,930$$

$$Ti = \underline{\underline{1,86 \text{ seg.}}}$$

$$Td = 0,5 \times L$$

$$Td = 0,5 \times 0,930$$

$$Td = \underline{\underline{0,465 \text{ seg.}}}$$

Los valores del tiempo se obtienen en segundos, para implementarlos en el controlador PID se transforma el tiempo en minutos.

$$Ti = 1,86 \text{ seg.} / 60$$

$$Ti = 0,031$$

$$Td = 0,465 \text{ seg.} / 60$$

$$Td = 0,00775$$

Tabla IV.

Tabla con los valores del PID en el eje X.

Parámetro para Y	Kc	Ti	Td
P	1,55	0	0
PI	1,395	2,008	0
PID paralelo	3,1096	0,031	0,0075

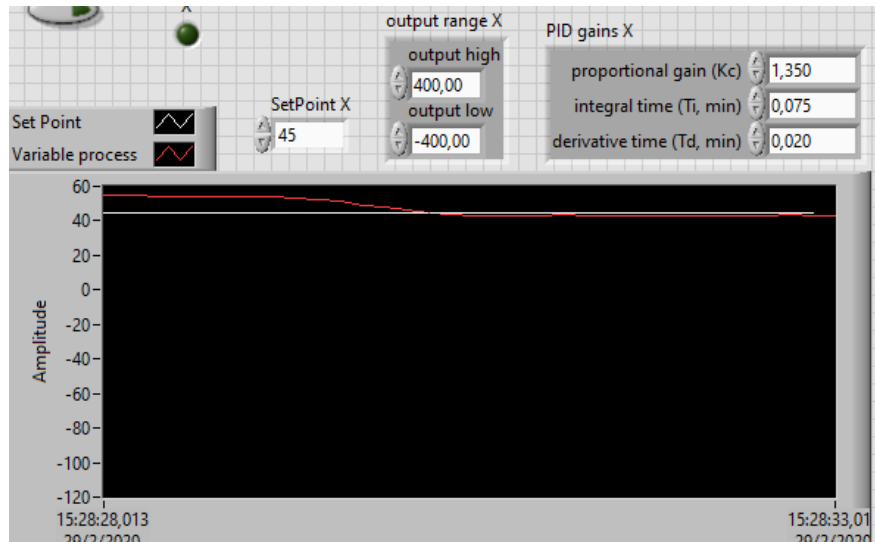


Figura 36. Controlador PID sintonizado con Ziegler-Nichols en el eje X

4.3 Funcionamiento del controlador PID.

4.3.1. Funcionamiento del controlador PID en el eje de las Y.

Para el análisis del controlador PID se usan dos valores de punto de ajuste en el eje: 20 grados y 40 grados, en ambos puntos de ajustes se aplica una perturbación, para observar como responde el controlador PID frente a la aplicación de perturbación en el sistema. En el eje Y, el rango de funcionamiento empieza desde 0 hasta los 60 grados.

Inicialización del sistema desde 0 grados en modo manual.

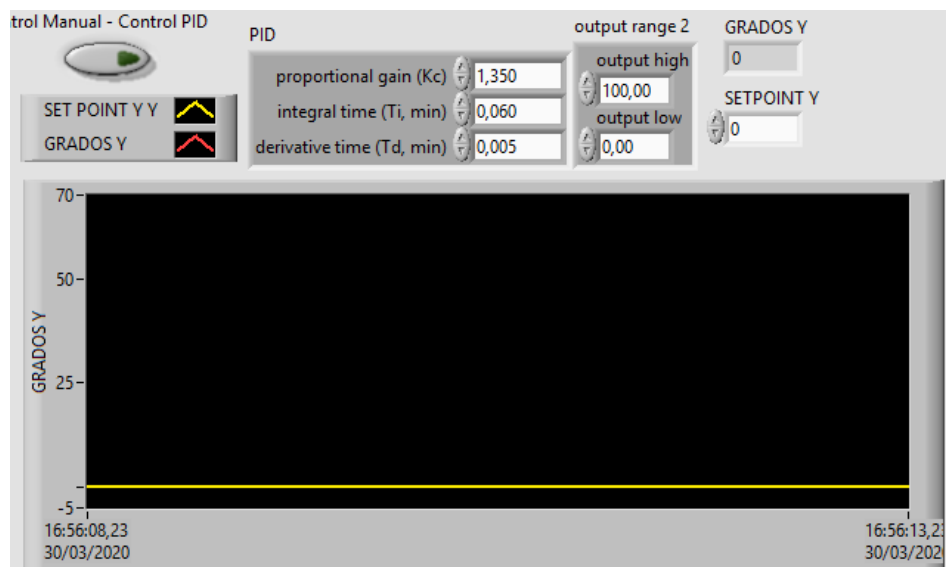


Figura 37. Inicio de funcionamiento del sistema en el eje Y en 0 grados.

El sistema inicia en 0 grados del eje Y.

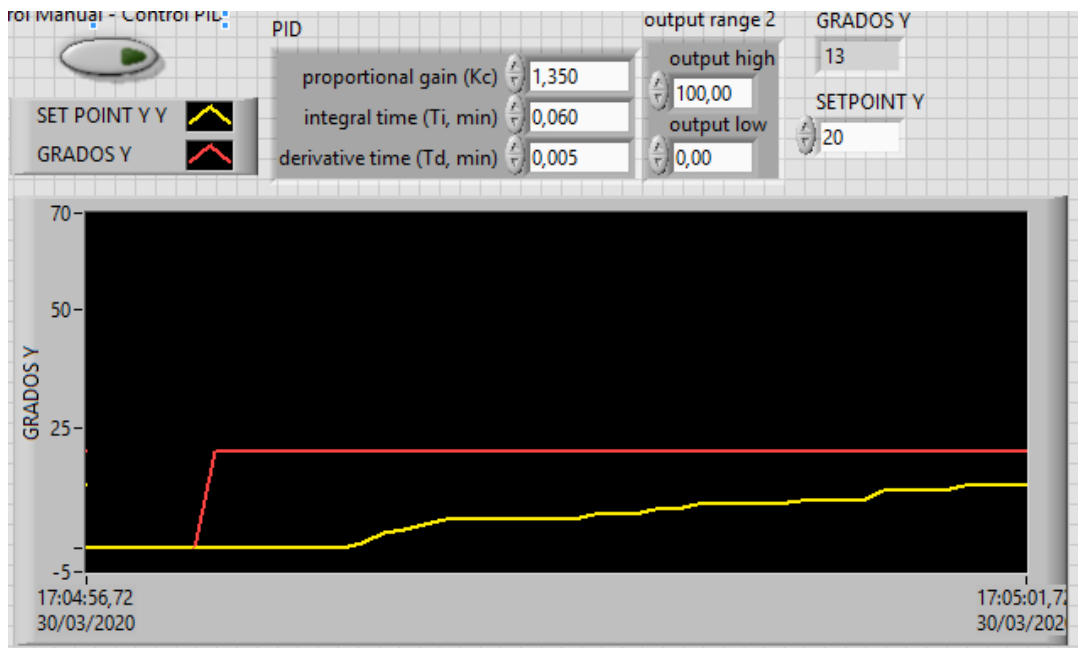


Figura 38. Funcionamiento del controlador PID para 20 grados de punto de ajuste

El controlador PID se sintoniza de tal manera que tenga una respuesta rápida, para que en pocos segundos pueda estabilizarse en el punto de ajuste deseado.

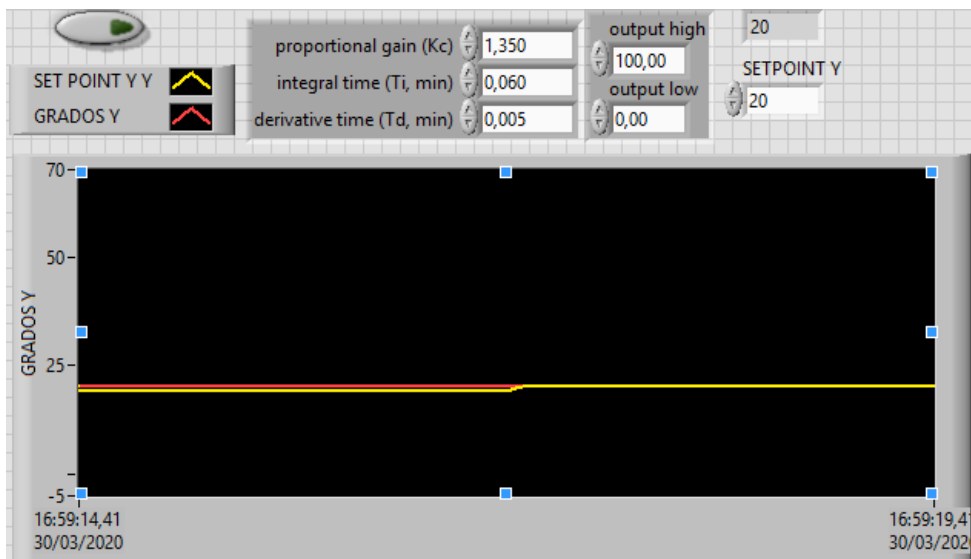


Figura 39. Controlador PID estabilizando el sistema en los 20 grados de punto de ajuste.

El sistema se estabiliza, para probar el funcionamiento del controlador PID se induce al sistema una pequeña perturbación.

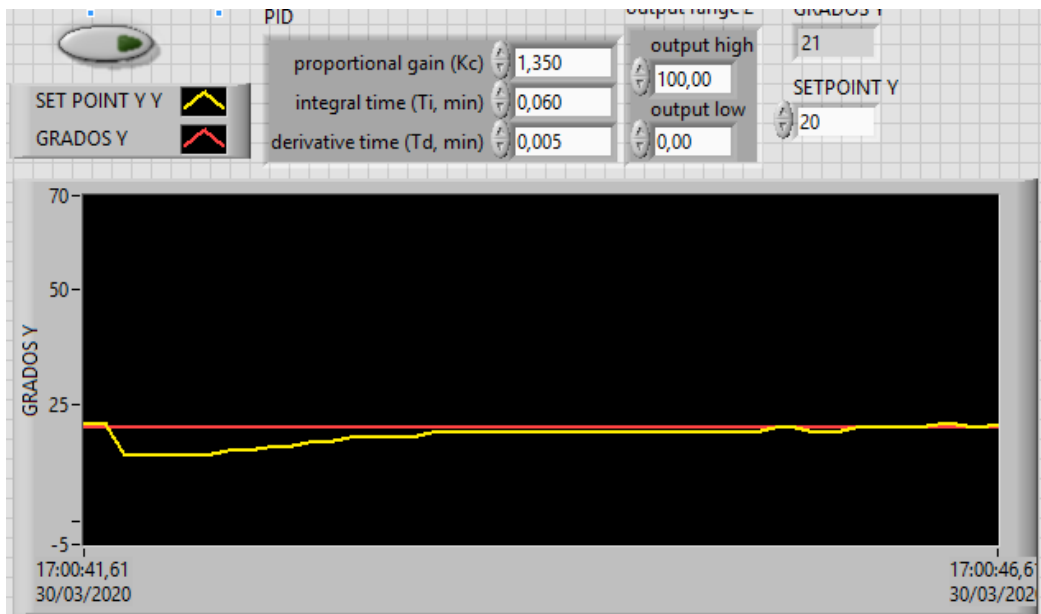


Figura 40. Aplicación de perturbación en 20 grados de punto de ajuste.

Se comprueba el funcionamiento del controlador PID en 20 grados de punto de ajuste, se cambia su punto de ajuste a 40 grados y se procede a analizar la respuesta del sistema frente al cambio del punto de ajuste.

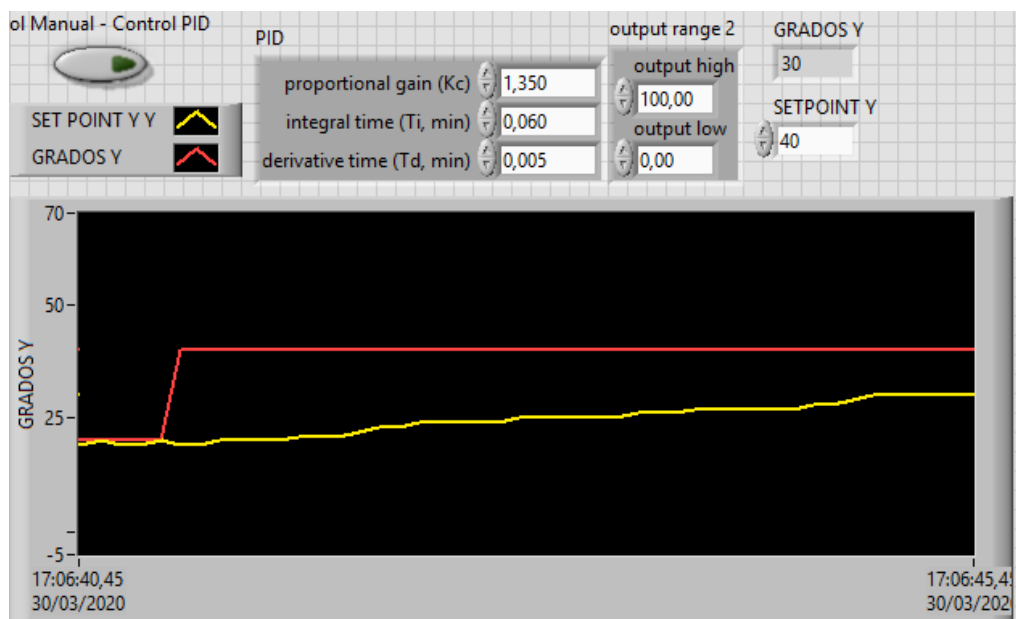


Figura 41. Cambio de 20 grados a 40 grados de punto de ajuste

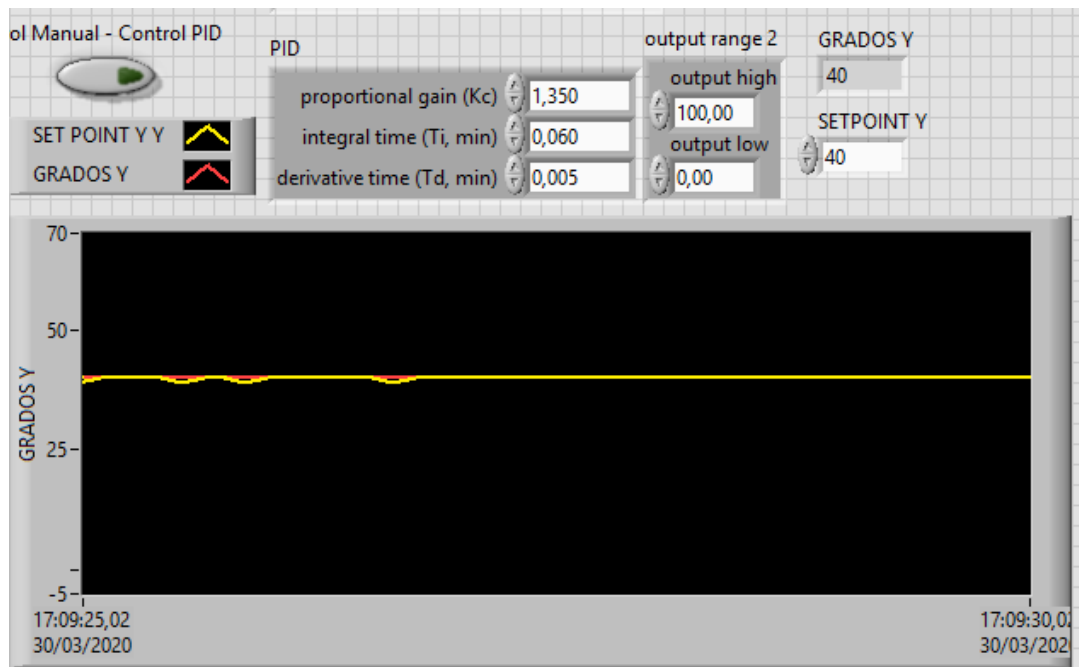


Figura 42. Sistema estabilizado en 40 grados de punto de ajuste.

Al llegar al punto de estabilización en los 40 grados, se aplica una perturbación al sistema para ver la respuesta del controlador PID.

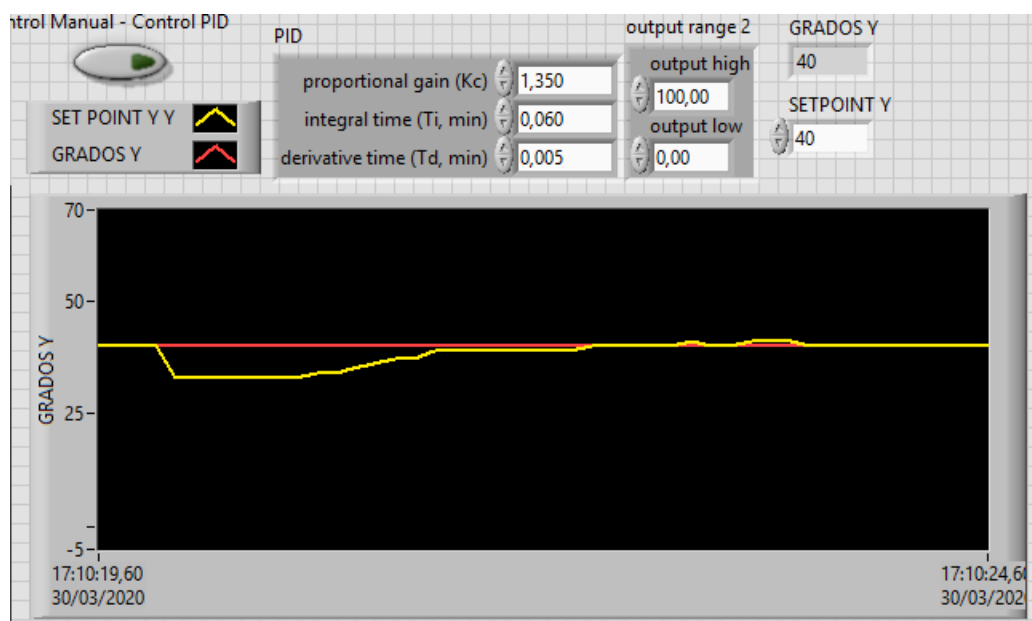


Figura 43. Respuesta del controlador PID ante una perturbación en 40 grados de punto de ajuste.

4.3.2. Funcionamiento del controlador PID en el eje X.

Para el análisis del controlador PID se usa dos valores de punto de ajuste en el eje: 60 grados y 90 grados, para ambos puntos de ajustes se le aplica una perturbación para observar cómo responde el controlador PID frente a la aplicación de perturbación en el sistema.

En el eje X, el rango de funcionamiento empieza desde los 0 grados hasta los 100 grados. Se inicia el sistema desde los 0 grados, en modo automático se coloca el valor de punto de ajuste a 60 grados, cabe recalcar que se produce una perturbación constante en el eje X debido a la fuerza de rotación que provoca el funcionamiento del motor1 que actúa en el eje Y.

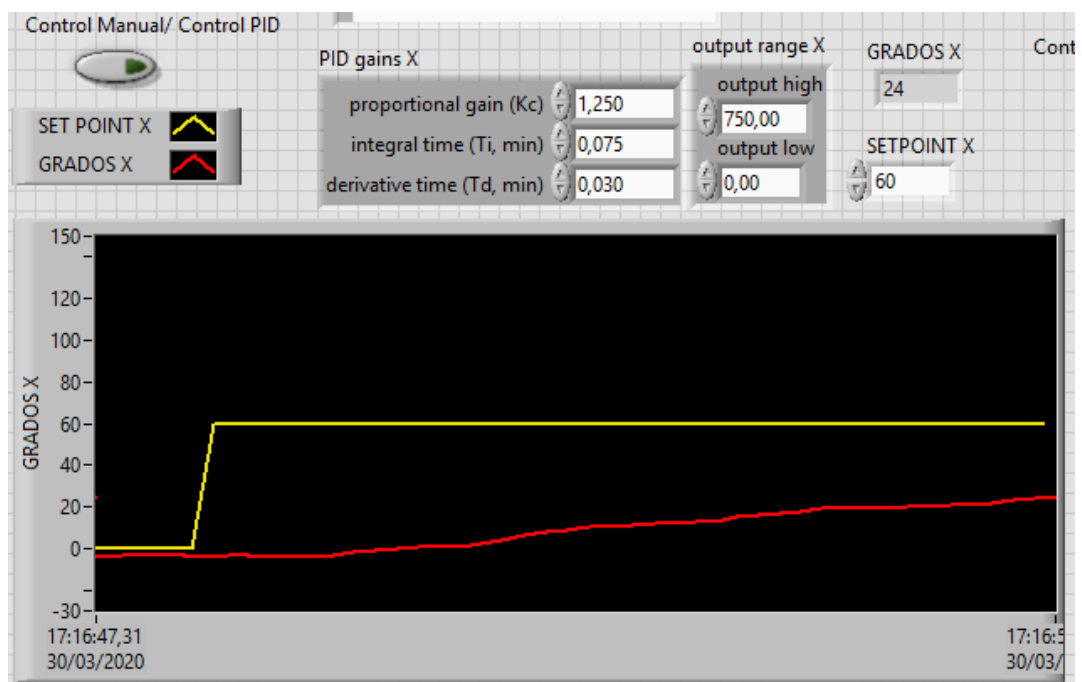


Figura 44. Funcionamiento del controlador PID en 60 grados de punto de ajuste.

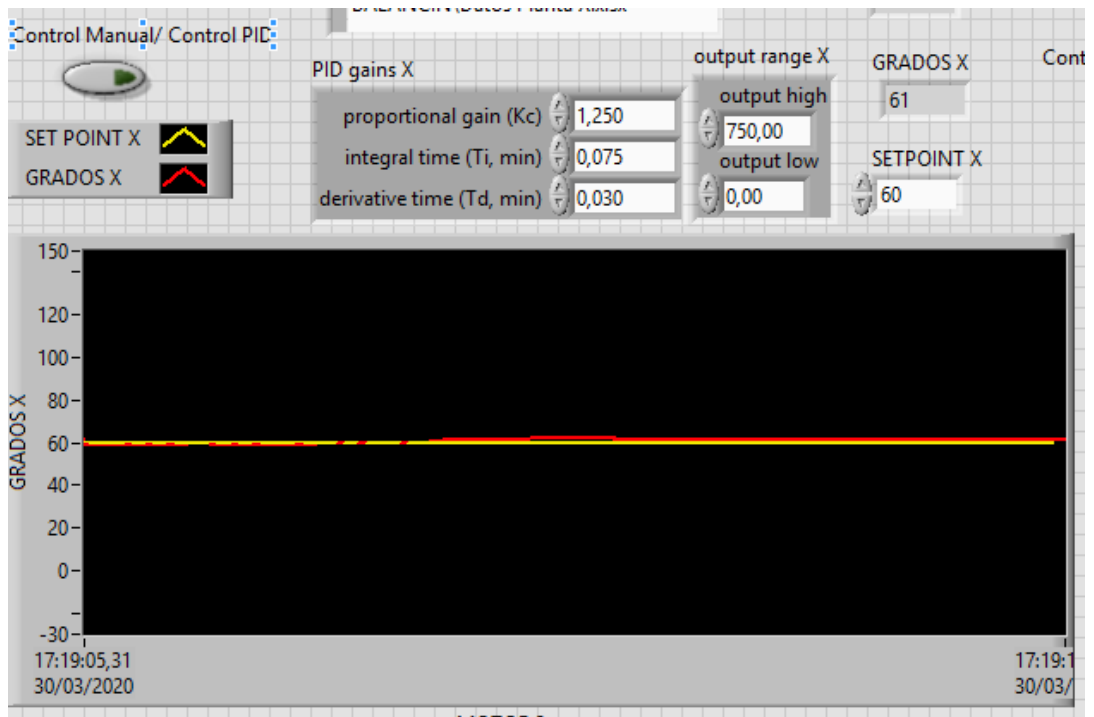


Figura 45. Estabilización del sistema en 60 grados del eje X.

Al estabilizarse el sistema, se induce una perturbación para analizar el tipo de respuesta del controlador PID ante mencionada perturbación.

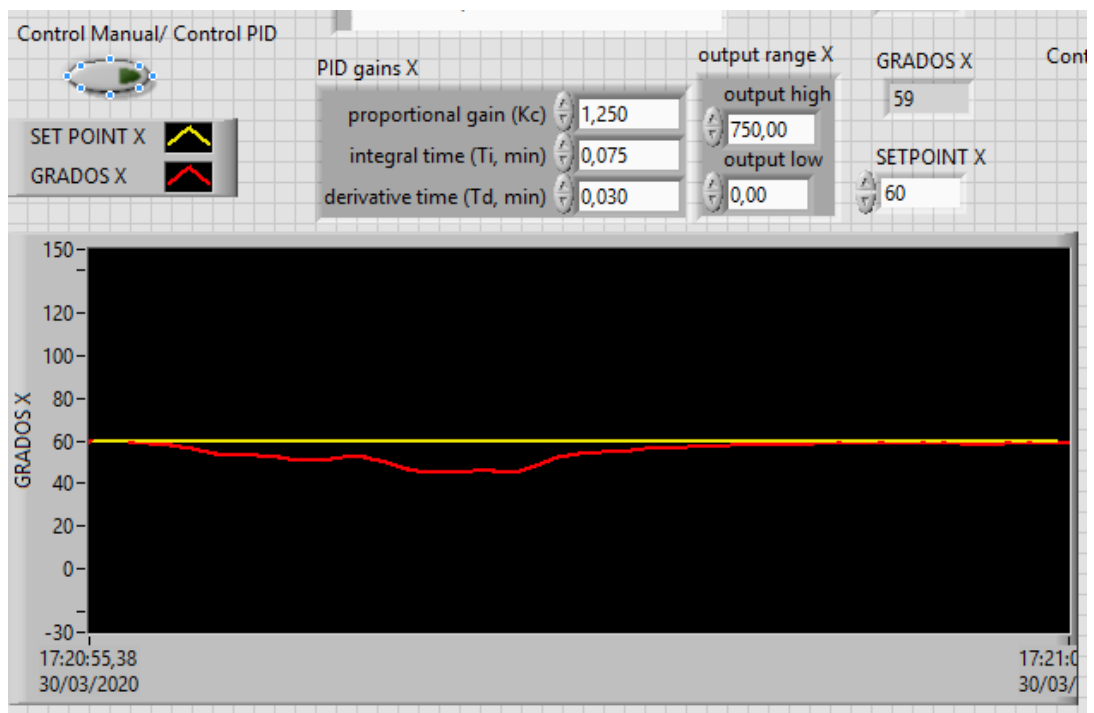


Figura 46. Respuesta del controlador PID frente a perturbación inducida.

Al inducirle una perturbación, el sistema reacciona de manera acorde a lo planificado, se produce un cambio en el valor de punto de ajuste para analizar el funcionamiento del controlador PID frente al cambio de los 60 grados hasta los 90 grados en el eje X.

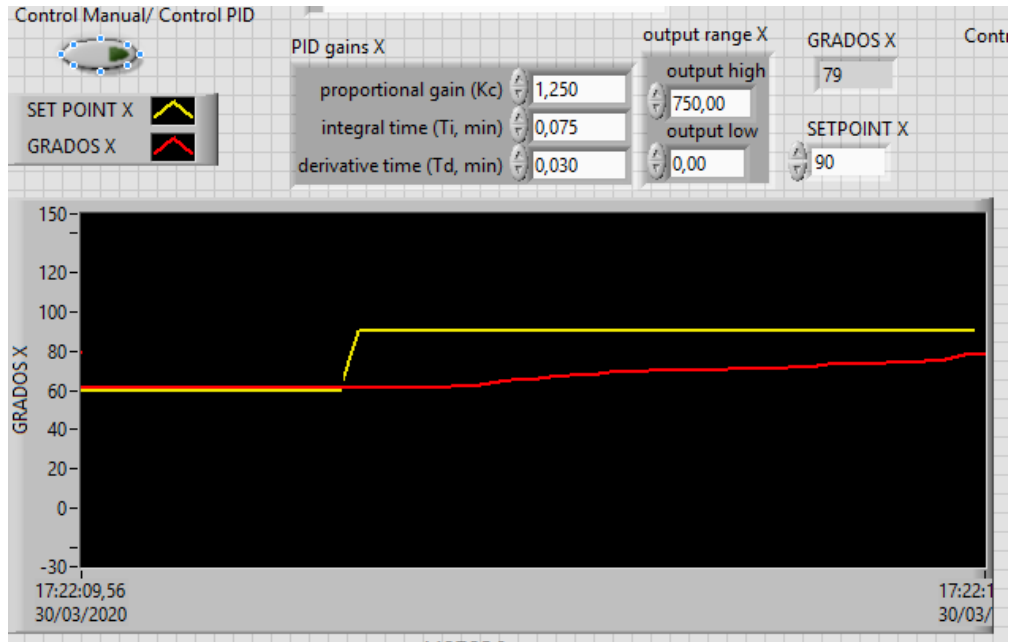


Figura 47. Respuesta del sistema frente al cambio de punto de ajuste de 60 a 90 grados.

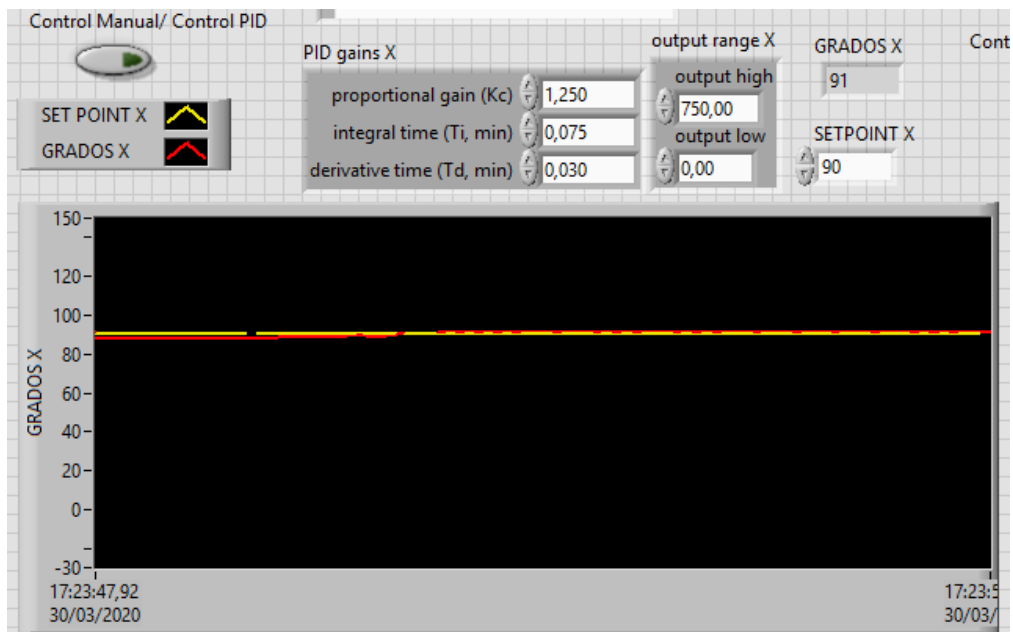


Figura 48. Estabilización del sistema en 90 grados de punto de ajuste en el eje X.

Al alcanzar la estabilidad del sistema en los 90 grados, se induce una perturbación para comprobar el correcto funcionamiento del controlador PID.

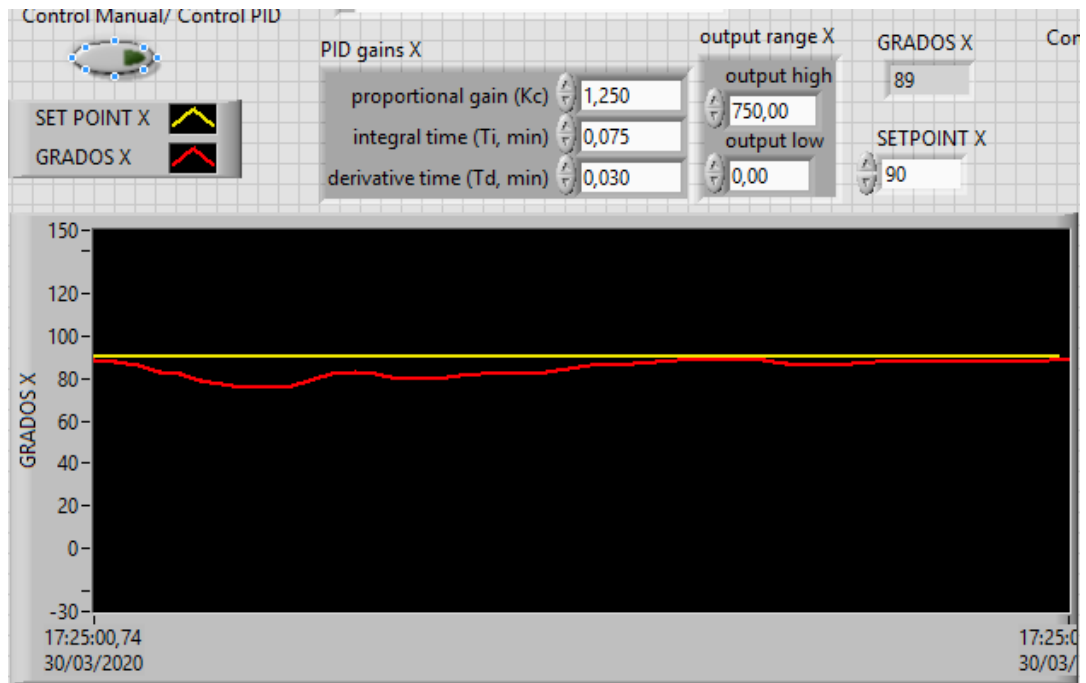


Figura 49. Respuesta del controlador PID frente a perturbación en los 90 grados del eje X.

Se procede a realizar una toma general de la interfaz gráfica, para analizar la respuesta del controlador PID en ambos ejes, el sistema está operando en 20 grados para el eje Y, y 60 grados en el eje X.

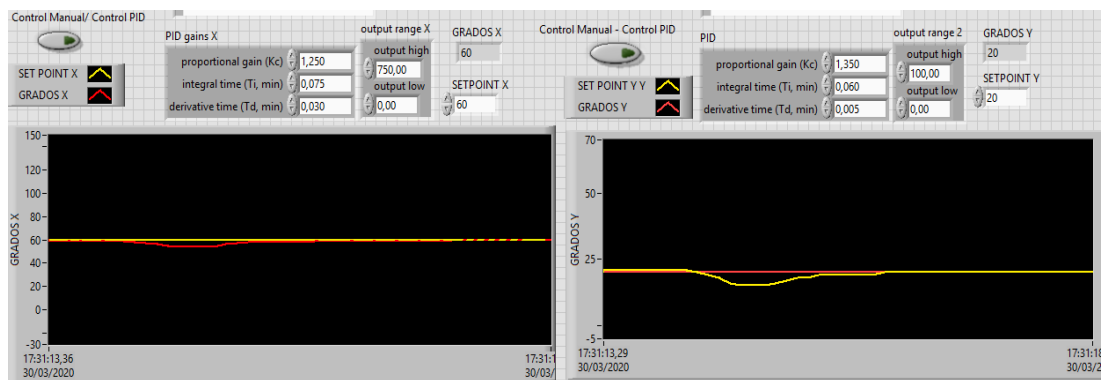


Figura 50. Vista general de la interfaz gráfica en respuesta del sistema frente a una mínima perturbación en el eje Y.

Se coloca el eje Y en 40 grados, y 90 grados en el eje X, al llegar a la estabilización de valor de punto de ajuste se procede a inducir una perturbación y comprobar el funcionamiento del controlador PID en ambos sentidos.

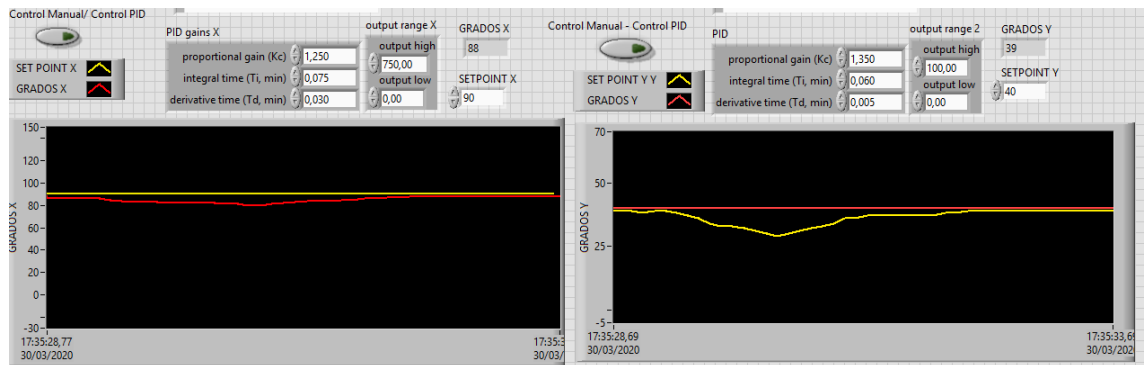


Figura 51. Vista general de la interfaz gráfica en respuesta del sistema frente a una mayor perturbación en el eje Y.

CONCLUSIONES.

- La sintonización del PID en ambos ejes, funciona acorde a lo planteado, cabe recalcar que los valores sintonizados son óptimos para el funcionamiento en los dos puntos de ajustes del análisis realizado, el controlador PID puede funcionar con los valores sintonizados en otros punto de ajuste, pero se puede llegar a obtener una sobre oscilación, debido a que para obtener un menor tiempo de estabilidad se amplió el valor de la proporcional para obtener una respuesta rápida de sistema y así reducir el tiempo en el que el sistema llegue al valor de punto de ajuste deseado.
- A pesar de las perturbaciones constantes que provee el funcionamiento de cada motor, el controlador PID cumple con las expectativas, posicionando al balancín en los grados que se desea para cada eje, cabe recalcar que los motores son no lineales lo cual provoca pequeños rangos de pwm donde su velocidad no varía
- Los motores brushless al tener un rango de trabajo en ancho de pulso demasiado corto, estabilizarlos en cierto punto de ajuste se vuelve complicado, la implementación de un controlador PID de vital ayuda.
- Para la adquisición de los datos necesarios para la sintonización del PID mediante Ziegler-Nichols se aplicó una señal escalón a los motores que producen el mayor trabajo en el posicionamiento del balancín en cada eje, obteniendo valores considerables con respecto a la estabilización de la planta.
- En el eje Y al posicionar el balancín, se obtienen porcentajes de error muy bajos, comparado al error que presenta el balancín en el eje X, ya que la fuerza centrífuga del motor en el eje en Y, produce perturbaciones en el eje X proporcional a la velocidad de giro del motor en el eje Y.

RECOMENDACIONES.

- Antes de iniciar el funcionamiento manual de los motores se debe inicializar cada motor, caso contrario ningún motor podrá ser controlado manual o automáticamente.
- Los cables de comunicación entre el Arduino y los dispositivos electrónicos que permiten el funcionamiento y la medición de posición del balancín deben estar correctamente realizadas, ya que si un cable se llegara a desconectar presentaría falla de comunicación y la interfaz gráfica daría error provocando que la comunicación general en el sistema se pierda y se deba reiniciar la comunicación.
- Antes de la conexión de los motores a la fuente de poder, se emplea un fusible que permita proteger a los dispositivos electrónicos en caso de alguna variación o sobre pico del voltaje.
- Debido a que el procesamiento de datos es alto, producido por el constante envío y recepción de datos, primero se debe posicionar un eje y después el otro eje, ya que al hacer funcionar los dos ejes al mismo tiempo puede producir errores en el sistema o retardos en el procesamiento de datos, dejando paralizado el funcionamiento del computador.

BIBLIOGRAFÍA.

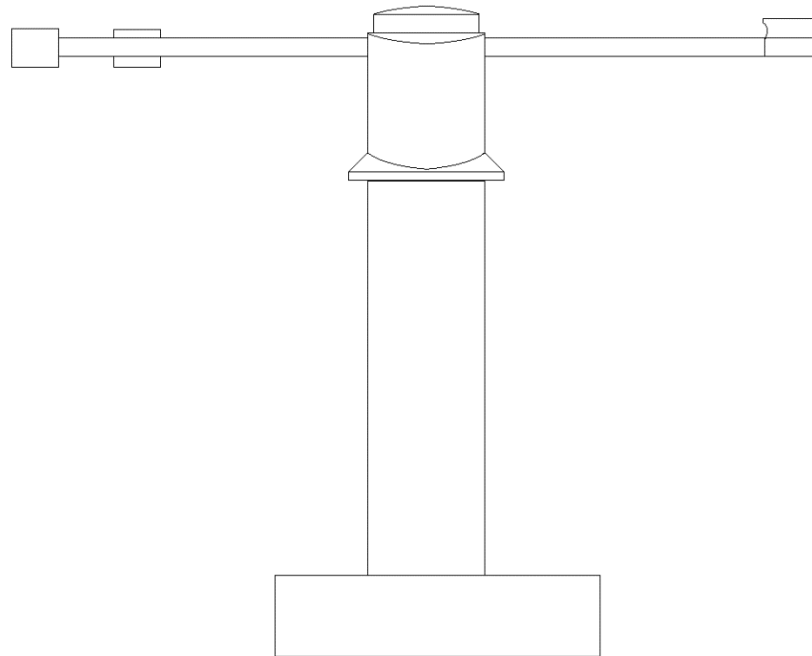
- Arduino. (s.f.). Arduino Mega 2650 Rev3. Obtenido de <https://store.arduino.cc/usa/mega-2560-r3>
- Arduino corp. (2020, 05 1). store.arduino. From arduino.mega: <https://store.arduino.cc/usa/mega-2560-r3>
- Controladores PID. (s.f.). Obtenido de <http://www3.fi.mdp.edu.ar/control4c7/APUNTES/Clase%207%20-%20PID.pdf>
- Departamento de Control, División de Ingeniería Eléctrica Facultad de Ingeniería UNAM. (14 de Noviembre de 2006). Reglas de sintonía de controladores PID. . Obtenido de Reglas de sintonía de controladores PID. : <https://www.slideserve.com/prema/reglas-de-sinton-a-de-controladores-pid>
- Fuente, M. d. (s.f.). Sintonia de reguladores PID. Obtenido de <http://www.isa.cie.uva.es/~maria/pids.pdf>
- Garcia, A. (20 de 05 de 2020). Parametric Identification of a 3 DOF Robot Manipulator Using a DSP. Obtenido de Parametric Identification of a 3 DOF Robot Manipulator Using a DSP: https://www.researchgate.net/figure/Ziegler-Nichols-method-applied-to-the-transient-response-of_fig3_268436846
- Krishnan, R. (2010). Permanent Magnet Synchronous DC Motor Drives. CRC PRESS.
- Mathworks. (1 de 05 de 2020). MatLab. Obtenido de MatLab: <https://la.mathworks.com/products/matlab.html>
- Mathworks. (s.f.). MATLAB. Obtenido de <https://la.mathworks.com/products/matlab.html>
- National Instruments. (1 de 05 de 2020). labview. Obtenido de labview: <https://www.ni.com/es-cr/shop/labview.html>
- National Instruments. (s.f.). National Instruments. Obtenido de <https://www.ni.com/es-cr/shop/labview.html>
- Ogata. (2010). Introduccion a los sistemas de control. En ogata, Ingenieria de control moderna (pág. 988). Pearson Education.
- Ogata, K. (2010). Ingenieria de control moderno. Pearson Educacion.
- Ogata, K. (2010). Sistema de control en tiempo discreto. Prentice Hall.
- Padmaraja Yedamale, Microchip Technology Inc. (2003). Burshless DC (BLDC) Motor Fundamentals. MICROCHIP, 20.
- Pardo, C. (2013). Metodo de Ziegler-Nichols: Picuino. Obtenido de Picuino Web site: <https://www.picuino.com/es/arduprog/control-ziegler-nichols.html>
- Ramon Vilanova, A. V. (2012). PID CONTROL IN THE THIRD MILENIUM . Londres: Springer.
- Sánchez, A. b. (30 de 08 de 2012). Control de procesos industriales. Obtenido de <https://es.slideshare.net/BlogsalDescubierto/cpi2-clase-4-parte-1-acciones-de-control>

Ziad Tarik Al-Dahan, N. K. (2016). Design and Implementation of Fall Detection System Using MPU6050 Arduino. International Conference on Smart homes and Health telematics. Springer.

ANEXOS
Anexos 1. Manual de uso



MANUAL DE USO



**DISEÑO E IMPLEMENTACIÓN DE CONTROLADOR PID
PARA BALANCÍN DE 2 GRADOS DE LIBERTAD**

AUTORES:

**ÁLVARO PAÚL ARAUJO SÁNCHEZ.
DAVID ERNESTO SANAGUANO PINCAY.**

2020

Requisitos de cumplimiento.

El módulo balancín está construido para operar con las siguientes especificaciones.

- Alimentación 12V DC, 5 A.
- Fusible europeo de 4 A, 250 V AC.
- Cable 18 awg, (suministro de voltaje)
- Cable 24 awg, (envío de datos).
- Motor brushless A22212/13T 1000Kv.
- Tarjeta micro - controladora Arduino Mega 2650.
- ESC 30 a.
- Sensor de rotación.
- Borneras de 5 A.
- Software LabVIEW.
- Software Matlab.
- Vi Package Manager.
- Toolkit LINX
- Cable de datos USB 2.0 AAB
- Procesador Intel Core i5 o superior.

Especificaciones

Tabla 1. Tabla de especificaciones.

ESPECIFICACIONES	Medidas.
Largo de la barra de aluminio	30 cm.
Ancho de la barra de aluminio	2,5 cm.
Radio del cilindro	2 cm.
Altura del cilindro	7 cm.
Largo de la barra de acero	15 cm.
Ancho de la barra de acero	5 cm.
Lado de la base cuadrada superior	5 cm.
Hélice del motor 1	10 in.
Hélice del motor 2	8 in.
Hélice del motor 3	8 in.
Tensión de los ESC	12v dc.
Tensión de los motores	12v dc.
Tensión en el sensor de rotación y	5v dc.
Tensión en el sensor de rotación x	5v dc.
Amperios en el motor1	2 a.
Amperios en el motor2	2 a.
Amperios en el motor3	2 a.
Amperios en los ESC	30 a.
Peso de los motores	51 gr.
Peso de los ESC.	25 gr.
Peso de la barra de aluminio	280 gr.
Perno sujetador.	3 in.
Radio del rodamiento	500 mm.
Altura de la caja del balancín	3,5 cm.
Largo de la caja del balancín	10 cm.
Ancho de la caja del balancín	8,5 cm.

Instalación

1. Se coloca la barra de acero en la caja de la base del balancín, y se la emperna en el centro de la base.
2. Se coloca el rodamiento de bolitas en la base cuadrada.
3. En el rodamiento de bolitas se acopla el sensor de rotación.
4. Se coloca la base cuadrada en la parte superior del tubo de acero.
5. La base cilíndrica se la acopla en el rodamiento de la base cuadrada.
6. Se acopla la barra de aluminio a la base cilíndrica mediante un perno ajustable.
7. Se fija la base del motor 1 a un extremo del tubo de aluminio.
8. Se fija el motor 1 a la base en el tubo de aluminio.
9. Se fija la base del motor 2 al otro extremo del tubo de aluminio.
10. Se fija el motor 2 a la base en el tubo de aluminio.
11. Se fija la base del motor 3 al lado opuesto de la base del motor 2 del tubo de aluminio.
12. Se fija el motor 3 a la base en el tubo de aluminio.
13. Se conectan los ESC a cada motor.
14. Los cables de datos de los motores se conectan de la siguiente manera:
 - Señal del motor 1 al pin 3 pwm del Arduino.
 - Tierra del motor 1 al pin gnd del Arduino.
 - Señal del motor 2 al pin 7 pwm del Arduino.
 - Tierra del motor 2 al pin gnd del Arduino.
 - Señal del motor 3 al pin 8 pwm del Arduino.
 - Tierra del motor 3 al pin gnd del Arduino.
15. Los cables de alimentación son conectados con la salida de la fuente.
16. Los pines del encoder rotacional del eje Y, se conectan de la siguiente manera.
 - Vcc al pin de 5v del Arduino.
 - Señal s al pin A1 del Arduino.
 - Tierra al pin gnd del Arduino.
17. Los pines del encoder rotacional del eje X, se conectan de la siguiente manera.
 - Vcc al pin de 5v del Arduino.
 - Señal s al pin A3 del Arduino.
 - Tierra al pin gnd del Arduino.

Estructura de la planta.

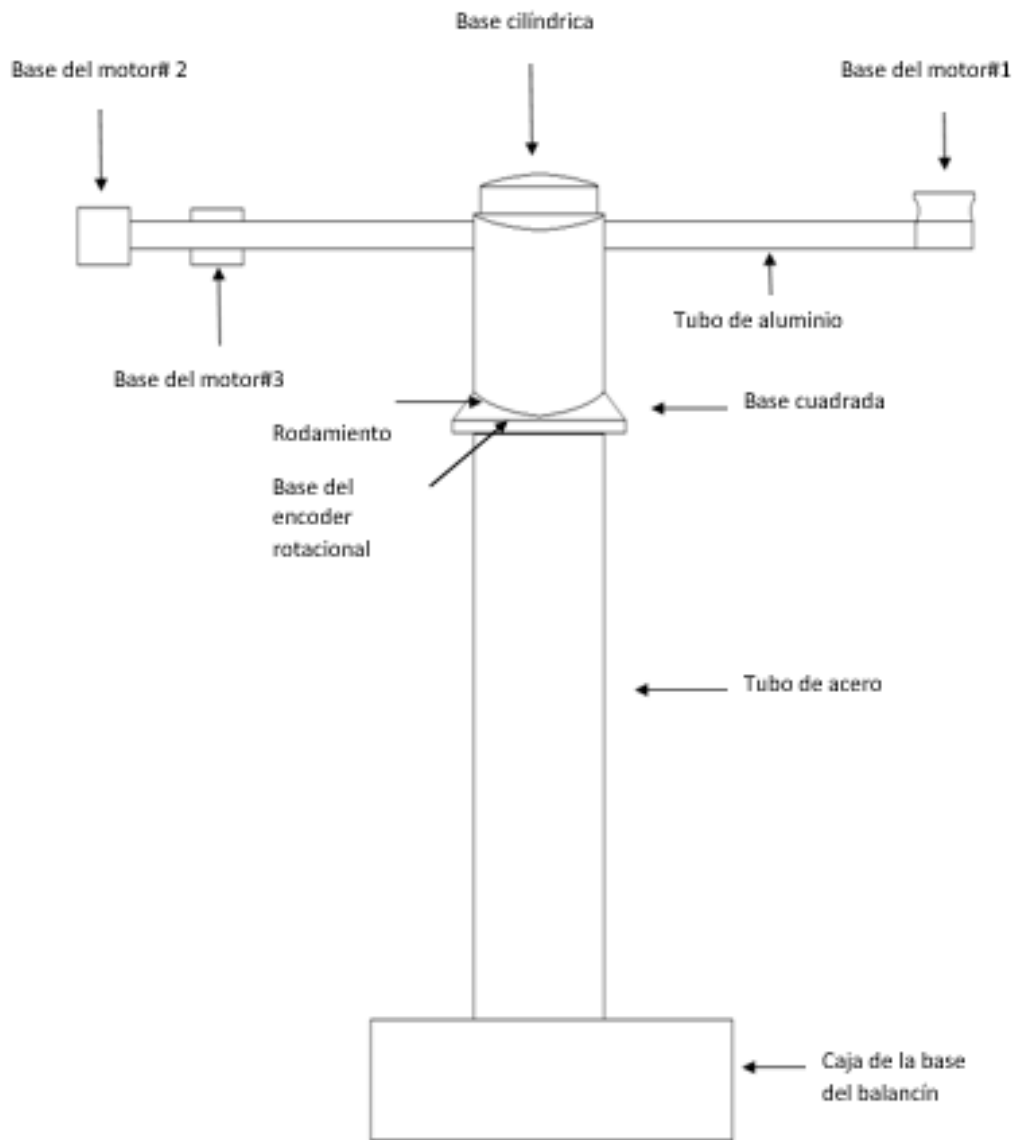


Figura 1. Estructura de la planta.

Diagrama de conexión en el Arduino

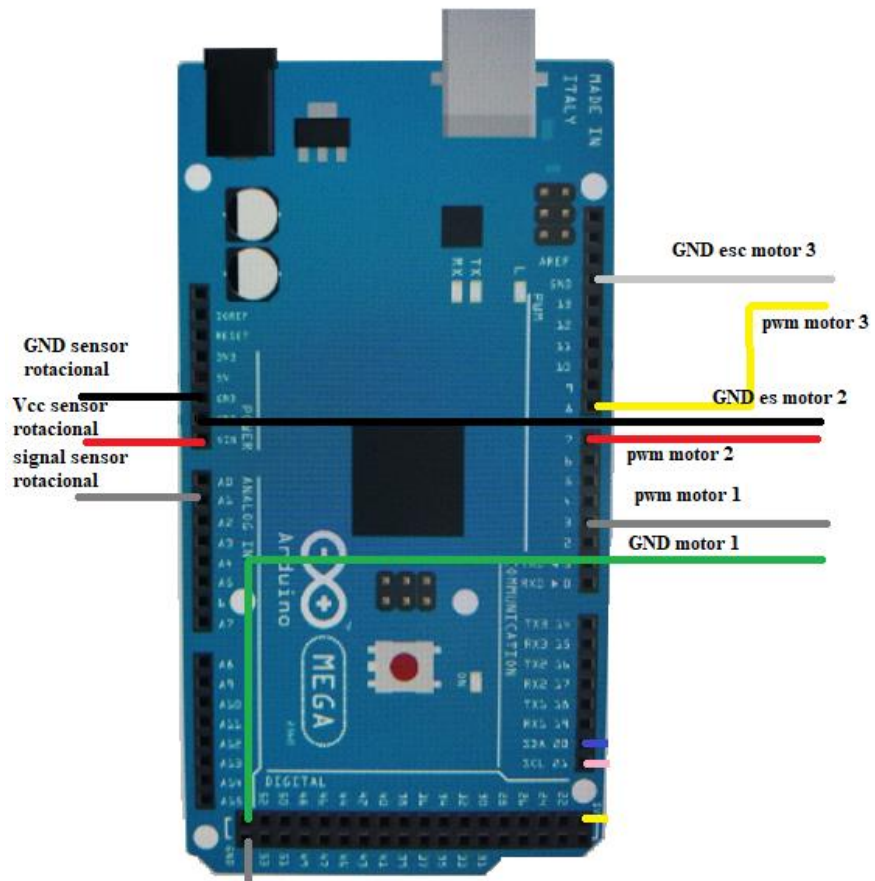


Figura 2. Conexión electrónica del Arduino.

Modo de operación

Cada motor funciona con un ESC con las mismas especificaciones, pero funcionan con diferente ancho de pulso, la planta tiene 2 modos de operación: modo control manual y modo control PID, para ambos modos los motores deben ser inicializados, la inicialización se realiza de la siguiente manera:

1. Se realiza el esquema de control manual para los motores.
2. Para la operación de cada motor, en la interfaz gráfica se aprecia los 3 módulos que sirven para la operación manual de cada motor, con límite inferior de cero y límite superior de 0,8 de ancho de pulso, debido a que si el ancho de pulso supera los 0,8 podría afectar a los motores.
3. Para inicializar los motores, con el módulo selector se manipula hasta aproximar el valor de 0,6 de ancho de pulso, o hasta escuchar los sonidos consecutivos de inicialización de los motores.
4. Después de inicializar el motor, regreso al valor 0 de ancho de pulso.
5. Con el mouse se manipula los valores de ancho de pulso para el modo manual.

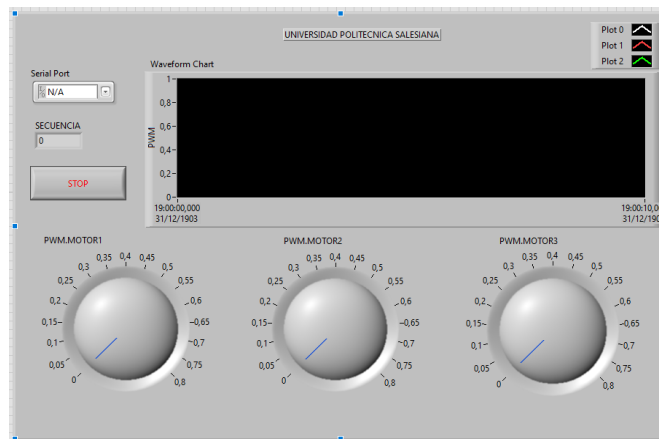


Figura 3. Interfaz gráfica de control manual.

Recomendaciones

Tabla 2. Tabla descriptiva de error.

Descripción del error	Causa probable	Soluciones posibles
No enciende el motor	Falta de suministro de poder al motor	Comprobar si el ESC está conectado a la fuente de poder
Error 8	Localización indefinida de la dirección que almacena los datos	Revisar si los datos se están guardando en la dirección correcta.
Error 5006	No se transmiten los datos desde el Arduino al LabVIEW	Revisar si el cable USB está conectado.

Mantenimiento.


A pesar de que los motores brushless no presentan desgastes por fricción, se recomienda limpiarlos con limpia contacto horas antes de su funcionamiento para eliminar suciedad y humedad dentro del motor.

En caso de que se queme algún driver, puede ser cambiado por otro driver con las mismas características (20 a).

Se debe aplicar "penetrante" en los cojinetes para evitar presencia de óxido debido a la humedad.

Antes de conectar la fuente a la toma de 110 v, revisar el fusible de protección.

Anexos 2. Prácticas

		GUIA DE PRÁCTICA DE LABORATORIOS
Carrera: Ingeniería Electrónica.		Asignatura: Teoría de control
Nro. Práctica	1	Práctica1: Conocimiento del equipo y su interfaz gráfica en LabVIEW.
Objetivos: <ul style="list-style-type: none"> • Identificar las partes que conforma el equipo. • Reconocer el funcionamiento de los módulos que encontramos en LabVIEW. • Diseñar un diagrama de control para el envío y recepción de datos desde el Arduino. 		
Instrucciones.		<ul style="list-style-type: none"> • Identificar como está constituida la planta. • Reconocer el funcionamiento de cada módulo de LabVIEW que se usa en el diagrama de bloques. • Diseñar un diagrama de control para la transmisión y recepción de datos desde el ordenador hasta el Arduino.
Actividades por desarrollar: <ul style="list-style-type: none"> • Identificar en cuantas partes está constituida la planta. • Reconocer el funcionamiento de cada módulo que se usa en el diagrama de bloques para el control del sistema. • Diseñar un diagrama de control para la transmisión y recepción de datos desde la interfaz gráfica en LabVIEW hasta el Arduino. 		

Se diseña el balancín de tal manera que pueda ser desmontable, está dividido en dos partes, la parte fija que contiene la placa Arduino, y todas las conexiones eléctricas del sistema.

En la parte móvil del balancín se encuentra empernados los motores a cada extremo del tubo de aluminio, y los drivers que mediante envío de anchos de pulso permiten el funcionamiento del motor, el tubo de aluminio que permite el movimiento en el eje Y se emperna en su parte central a una base cilíndrica la cual se acopla a un rodamiento en el centro del tubo de acero inoxidable, el cual permite mover el balancín en el eje X.

En la parte inferior del acople entre la base cilíndrica y el rodamiento de la base del tubo de acero inoxidable se encuentra un sensor rotacional, el cual permite censar los grados en el cual se posiciona el balancín en el eje X.

Los motores están conectados a su driver, que conectado a los pines de salida pwm del Arduino se imprime los diferentes valores de ancho de pulso.

Se fija el Arduino a la caja protectora de acero en la parte inferior del balancín, aislada con una placa de acrílico, para evitar el contacto entre los dispositivos electrónicos y la base metálica.

Se alimentan los motores por una fuente de poder de 12 v y 5 a, posee un fusible para proteger el sistema de alguna sobrecarga que provenga de la fuente de voltaje, el Arduino se conecta al puerto USB del ordenador.

Desarrollo del algoritmo de control:

1. Se crea un archivo VI nuevo, con el nombre del proyecto.
2. Se procede a colocar el módulo open de LINX para el inicio de la comunicación.
3. Se crea un while loop donde se desarrolla el algoritmo de control para los motores.
4. Dentro del while loop se coloca los módulos "set duty cycle" para la impresión de los valores de ancho de pulso en cada motor.
5. A cada motor se le crea una constante donde se indica a que puerto del Arduino estará conectado, para la transmisión de datos.
6. Se coloca un control para ciclo de trabajo pwm en cada motor, donde se imprime valores desde 0 a 1 para hacer funcionar los motores.
7. Se cierra la comunicación del Arduino con el módulo close de LINX
8. Se coloca un indicador de error en la parte derecha del módulo close para notificar algún error en el funcionamiento del algoritmo de control.

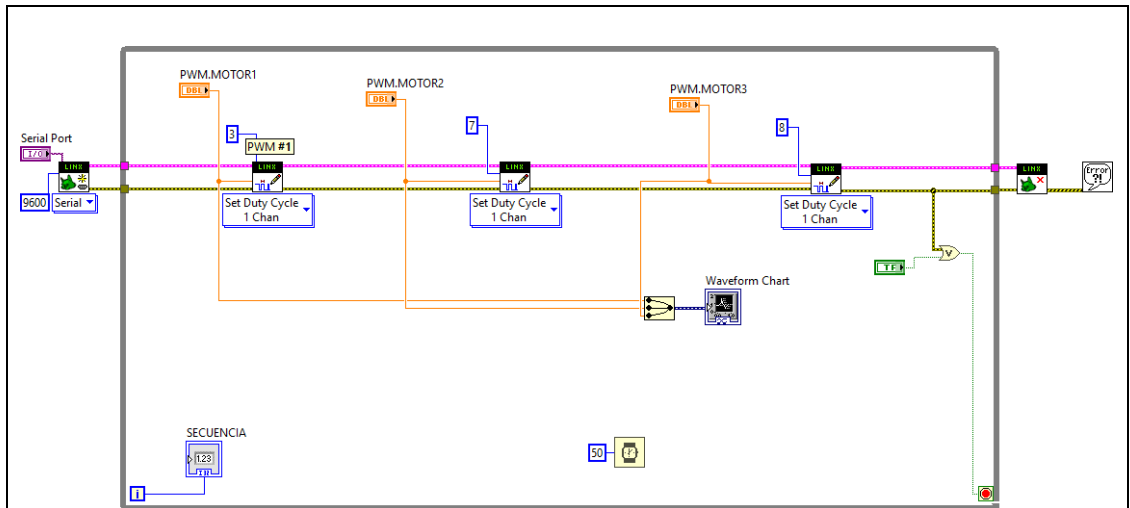


Figura 4. Diagrama de control para transmisión y recepción de datos.

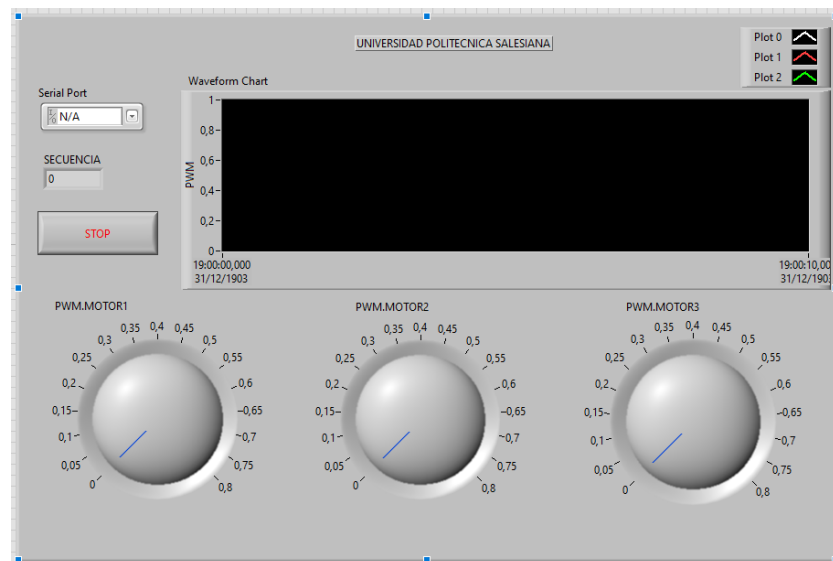


Figura 5. Interfaz gráfica del diagrama de control de los motores.

Para la comunicación entre el Arduino y LabVIEW siempre se debe cargar el firmware de LINX de la siguiente manera:

1. En la casilla de tools se da clic izquierdo y se despliega una ventana.
2. Se da clic izquierdo en la casilla de LINX, y se despliega 4 opciones.
3. Selecciona la opción de LINX Firmare Wizard y se carga la información al Arduino.

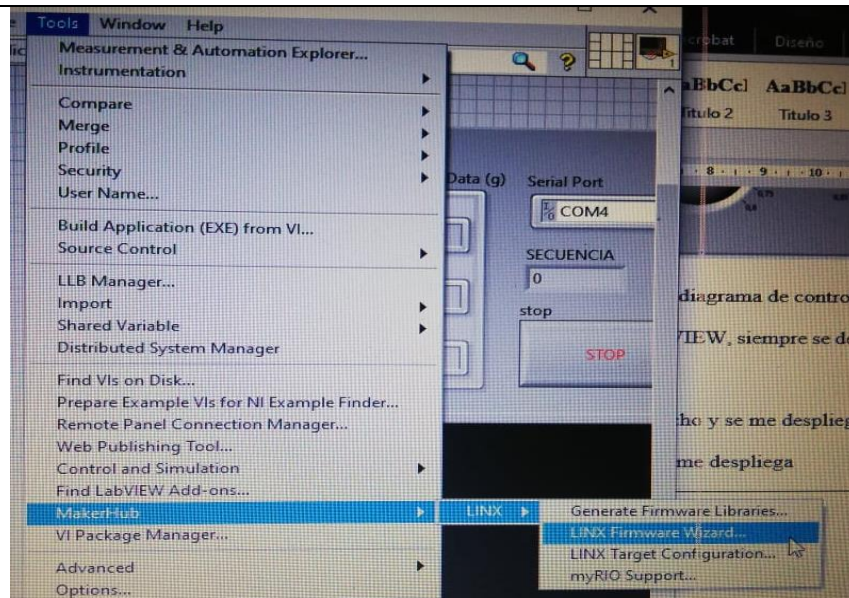


Figura 6. Se selecciona LINX Firmware Wizard



Figura 7. Se selecciona la placa de Arduino que se está usa.

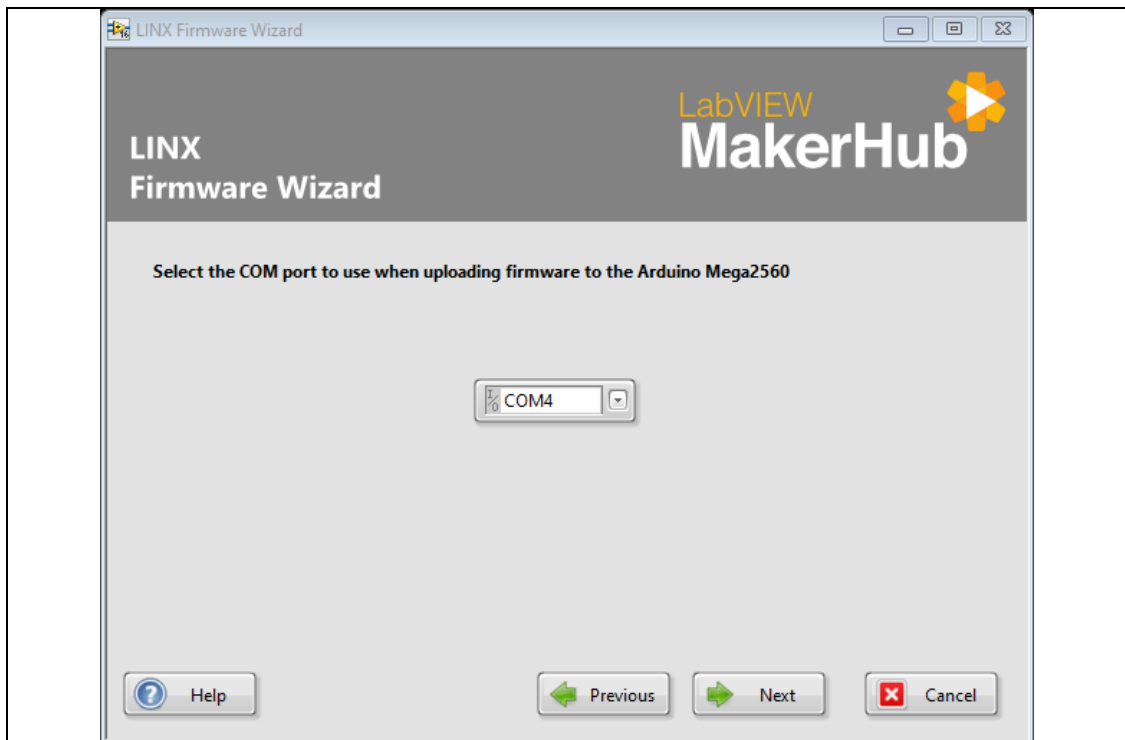


Figura 8. Se selecciona el puerto de comunicación.

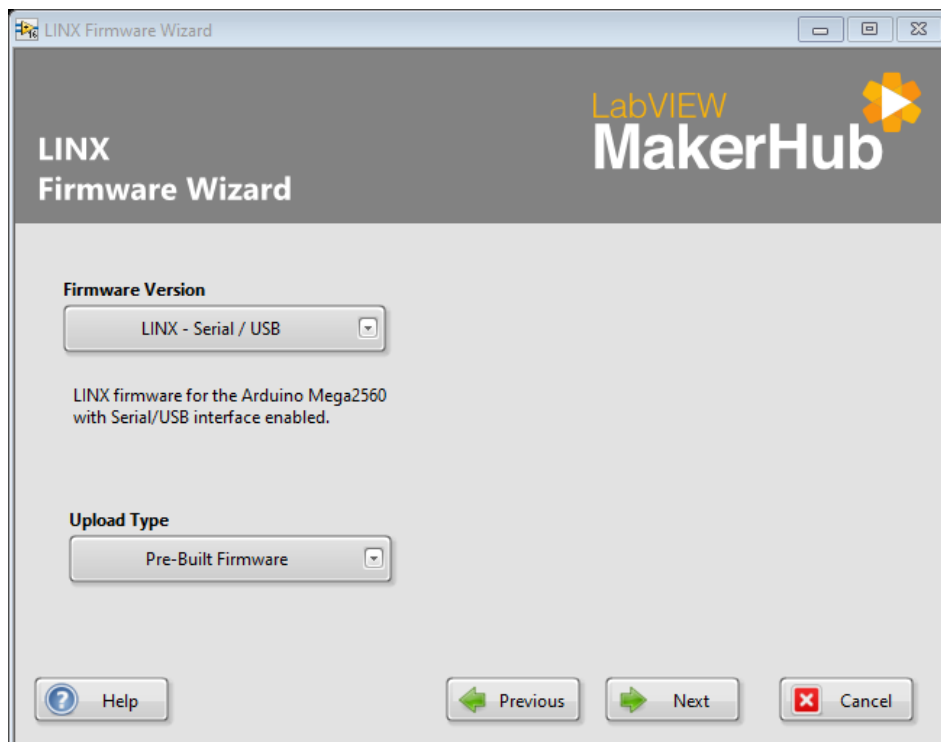


Figura 9. Se revisa el tipo de firmware a cargar.

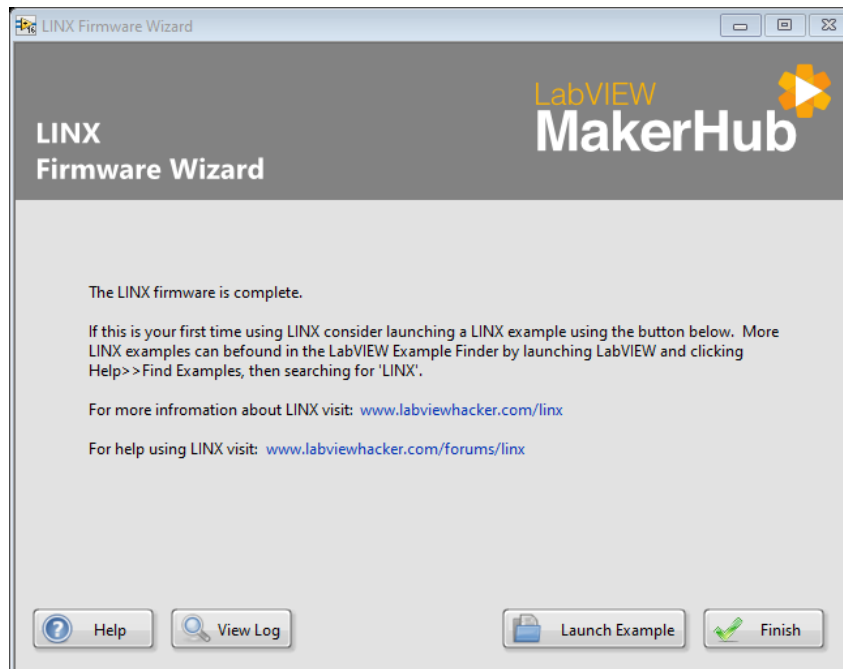


Figura 10. Notificación de carga completa del firmware.

Resultados

Se identifica el funcionamiento de cada parte del sistema, reconociendo que motores permiten posicionarse al balancín en el eje X, y que motor le permite posicionarse en eje Y, se desarrolla un diagrama de control base para transmisión de señales de ancho de pulso desde la interfaz gráfica a mi tarjeta de Arduino.

Conclusiones.

Al reconocer cada parte del balancín, se diseña un algoritmo de control sencillo que permite comunicar el Arduino con el algoritmo de control, mediante el envío y recepción de datos desde la tarjeta Arduino hasta la interfaz gráfica de LabVIEW.

Anexo



Práctica #1.vi

[Proyecto titulación\prácticas\Práctica #1.vi](#)

Carrera: Ingeniería Electrónica.		Asignatura: Teoría de control
Nro. Práctica	2	Práctica 2: Posicionamiento del sistema en sentido vertical
<p>Objetivos:</p> <ul style="list-style-type: none"> • Identificar el tipo de control que se necesita para posicionarse en el eje Y. • Determinar tres tipos de ancho de pulso para posicionarse en el eje Y. • Reconocer el tipo de control manual que se necesita para posicionarse en el eje Y. 		
Instrucciones.	<ul style="list-style-type: none"> • Inicializar los motores. • Introducir los valores de ancho de pulso para que el motor funcione. • Determinar en qué valores de ancho de pulso el motor se posiciona en tres diferentes tipos de mediciones. 	
<p>Actividades por desarrollar:</p> <ul style="list-style-type: none"> • Realizar el respectivo proceso para inicializar el motor. • Determinar el valor de ancho de pulso en el cual el motor empieza a funcionar. • Determinar diferentes valores de ancho de pulso en el cual el balancín se posiciona en tres tipos de mediciones. 		
<p>1. En el diagrama de bloques se realiza la comunicación entre la interfaz y LabVIEW con los módulos de LINX open y close.</p>		

2. Se coloca el módulo “set duty cycle” dentro del while loop donde se transmiten los valores de ancho de pulso al motor.
3. Para la variación del ancho de pulso se crea un control a la entrada del módulo “set duty cycle”, en el cual se imprime valores dentro del rango de 0 a 1.
4. Se coloca un indicador gráfico, para muestrear los valores de ancho de pulso.

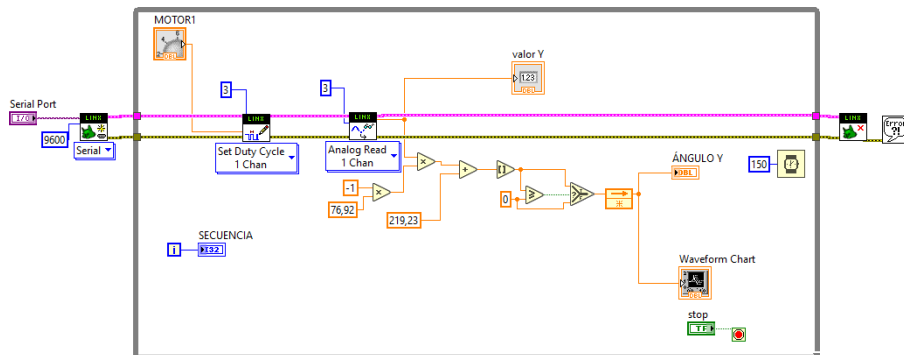


Figura 1. Algoritmo para el control en el eje Y.

5. En el panel frontal se reproduce el diagrama de control, se realiza la inicialización del motor.

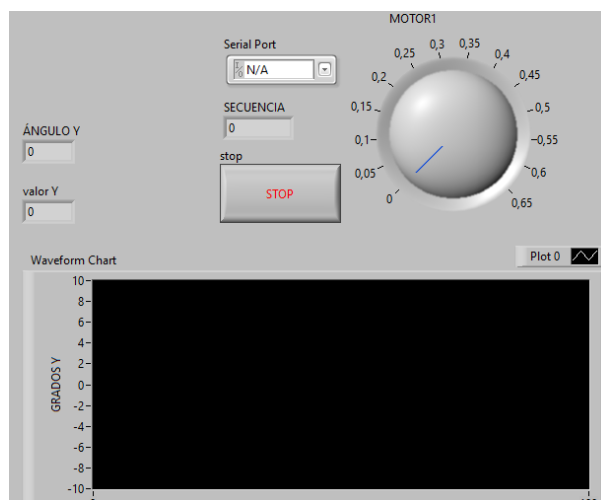


Figura 2. Panel frontal para la inicialización del motor

6. Posterior a la inicialización, se determina los valores en el cual puedo posicionar el balancín en tres puntos de ajustes diferentes.

Resultados

Después de realizar el diagrama de bloques, se inicializa colocando en el control de ancho de pulso el valor de 0.580, posterior con el rodillo del mouse se disminuye el valor a 0 o hasta escuchar el sonido que notifica que el motor ya se inicializo.

Determinar 3 valores de ancho de pulso para obtener los puntos de ajuste en 0, 15 y 30 grados en el eje Y.

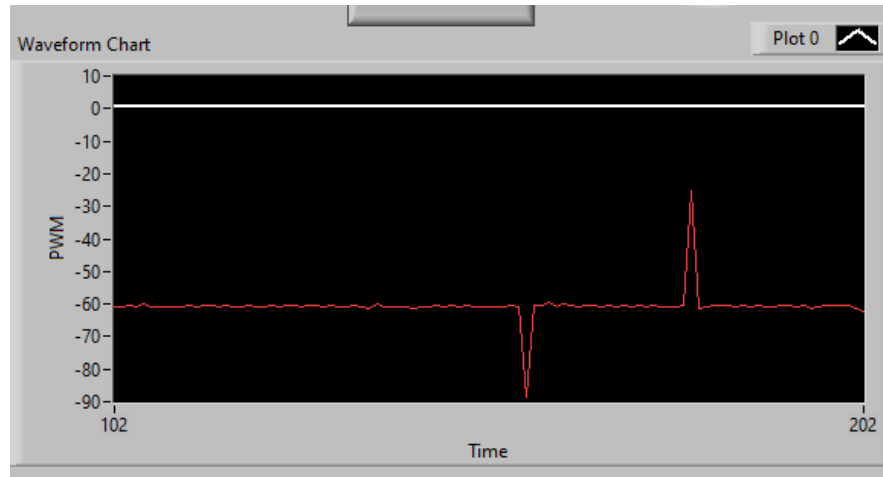


Figura 3. Gráfica del motor 1 inicializado.

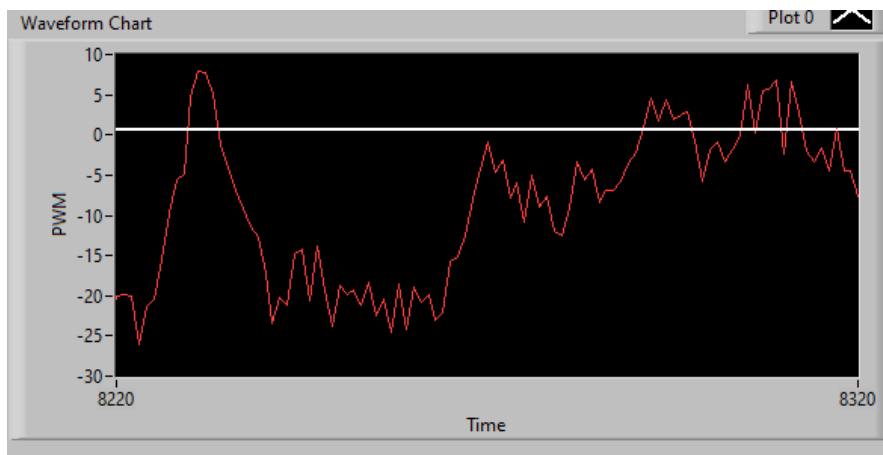


Figura 4. Gráfica del motor 1 con punto de ajuste de 0 grados en modo manual.

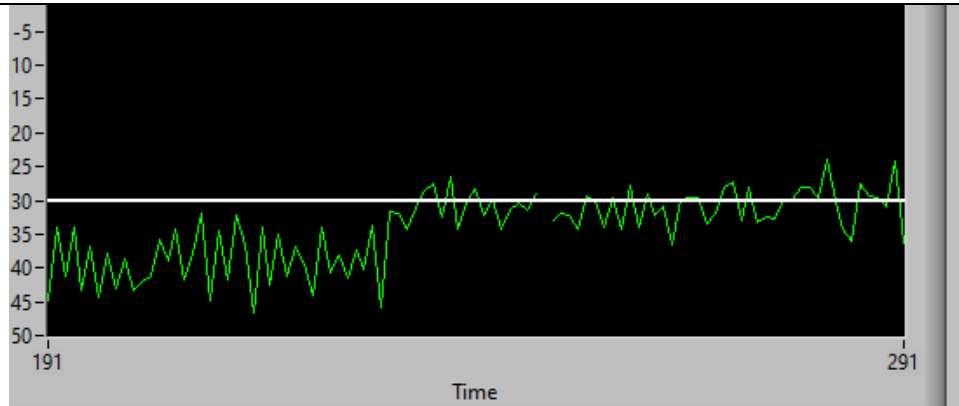


Figura 5. Gráfica del motor 1 con punto de ajuste de 30 grados en modo manual.

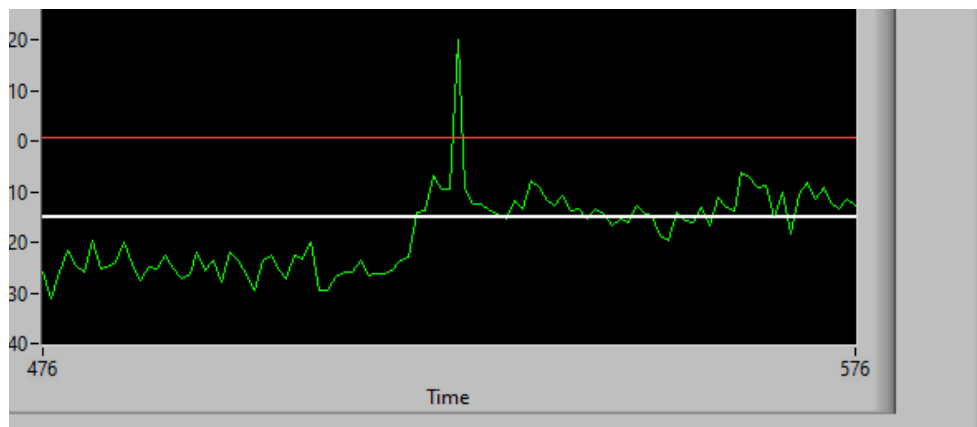


Figura 6. Gráfica del motor 1 con punto de ajuste de 15 grados en modo manual.

Conclusiones.

Se logra comunicar el Arduino y el algoritmo de control mediante el envío y recepción de datos, se determina los valores para posicionar el balancín en diferentes grados en el eje Y incrementando o disminuyendo el ancho de pulso "pwm" del motor, sin llegar al punto muerto en el cual el motor se apaga.

Recomendaciones.

Para lograr un mayor control manual para fijar el motor en el punto de ajuste deseado, se debe disminuir la escala del control dial.

Anexos



Práctica #2.vi

[Proyecto titulación\prácticas\Práctica #2.vi](#)



Carrera: Ingeniería Electrónica.		Asignatura: Teoría de control
Nro. Práctica	3	Práctica 3: Posicionamiento del sistema en sentido horizontal
<p>Objetivos:</p> <ul style="list-style-type: none"> • Identificar el tipo de control que necesita el posicionamiento en el eje X. • Determinar tres tipos de ancho de pulso para el posicionamiento en el eje X. • Reconocer el tipo de control manual que necesita el posicionamiento en el eje X. 		
Instrucciones.		<ul style="list-style-type: none"> • Inicializar los motores. • Introducir los valores de ancho de pulso para que los motores funcionen. • Determinar en qué valores de ancho de pulso los motores se posicionen en tres diferentes tipos de mediciones.
<p>Actividades por desarrollar:</p> <ul style="list-style-type: none"> • Realizar el respectivo proceso para inicializar los motores. • Determinar el valor de ancho de pulso, en el cual los motores empiezan a funcionar. • Determinar diferentes tipos de valores de ancho de pulso, en el cual el balancín se posicione en tres diferentes tipos de mediciones. • En el diagrama de bloques se realiza la comunicación entre la interfaz y LabVIEW con los módulos de LINX open y close. • Se coloca el módulo “set duty cycle” dentro del while loop, donde se transmite los valores de ancho de pulso a los motores. • Para la variación del ancho de pulso se crea un control a la entrada del módulo “set duty cycle”, en el cual se imprime valores dentro del rango de 0 a 1. <ol style="list-style-type: none"> 1. Se coloca un indicador gráfico, para mostrar los valores de ancho de pulso. 		

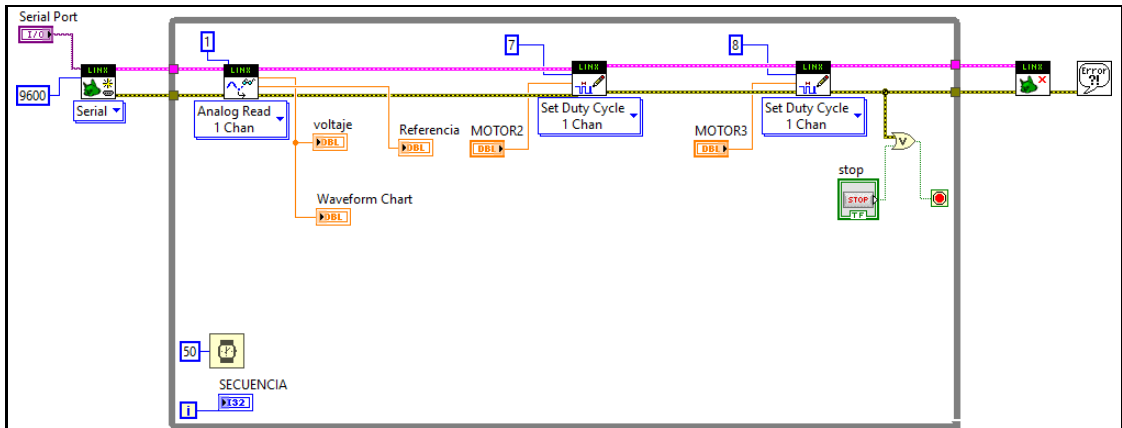


Figura 1. Algoritmo de control para el eje X

2. Se realiza las conexiones para la conversión de los datos obtenidos por el encoder rotacional.

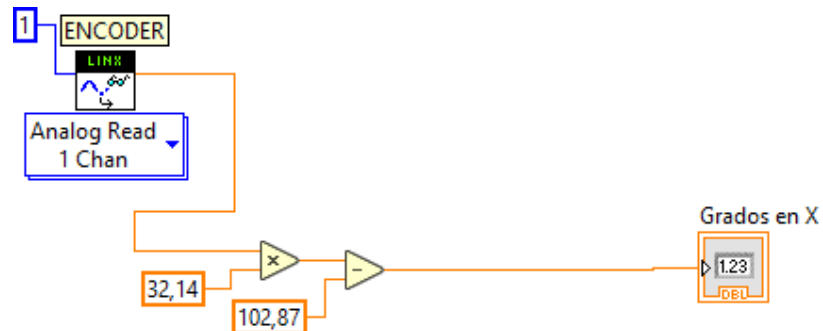


Figura 2. Obtención de los grados en X con el encoder rotacional.

3. En el panel frontal se reproduce el diagrama de control, se realiza la inicializa de los motores 2 y 3 con los datos detallados en la siguiente figura.

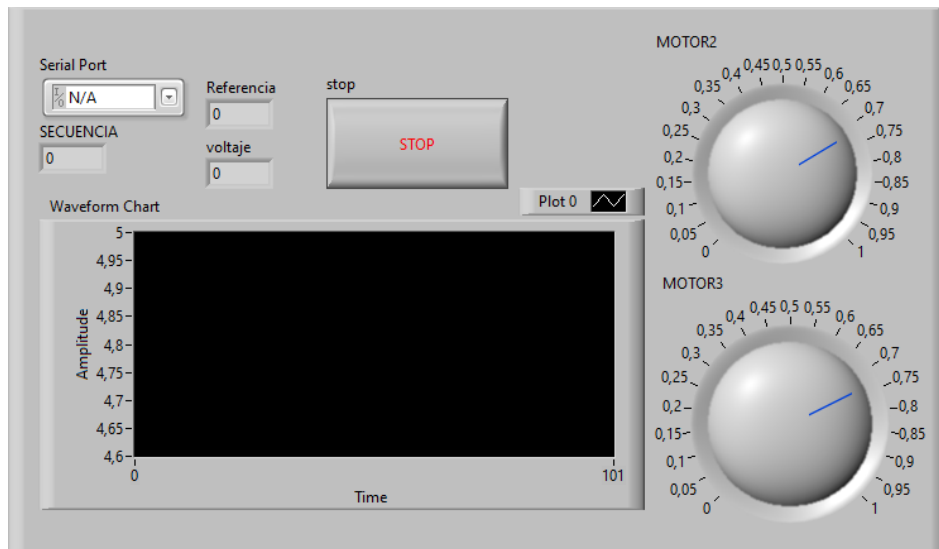


Figura 3. Control manual de ancho de pulso para inicializar los motores.

4. Posterior a la inicialización se determina los valores en el cual puedo posicionar al balancín en 3 puntos de ajustes diferentes.

Resultados

Después de realizar el diagrama de bloques se inicializa los motores colocando en el control de ancho de pulso el valor de 0.580, posterior con el rodillo del mouse se disminuye el valor a 0 o hasta escuchar el sonido que notifica que el motor ya se inicializo.

Determinar los valores de ancho de pulso para posicionar la planta en 45, 30 y -30 grados en el eje X.

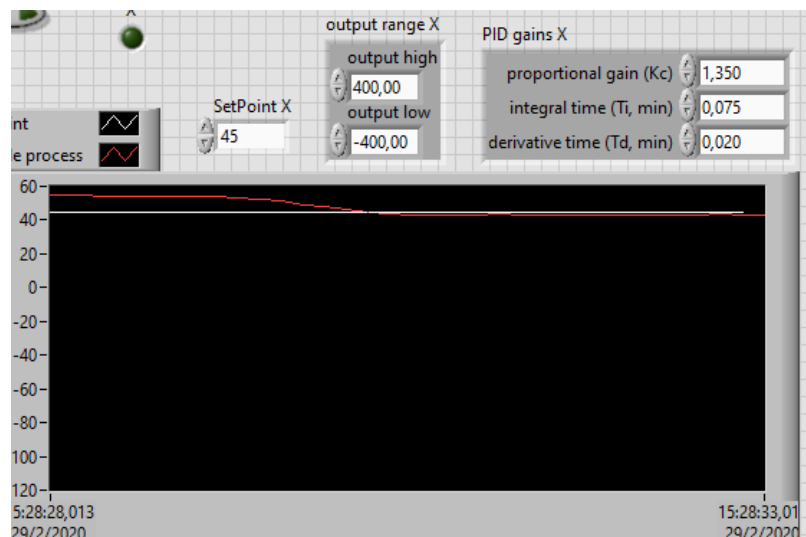


Figura 4. Control manual del eje X en 45 grados.

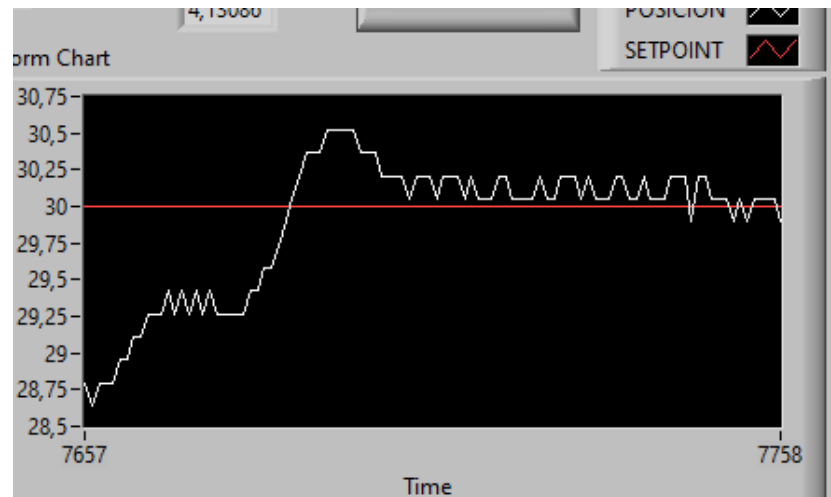


Figura 5. Control manual del eje X en 30 grados.

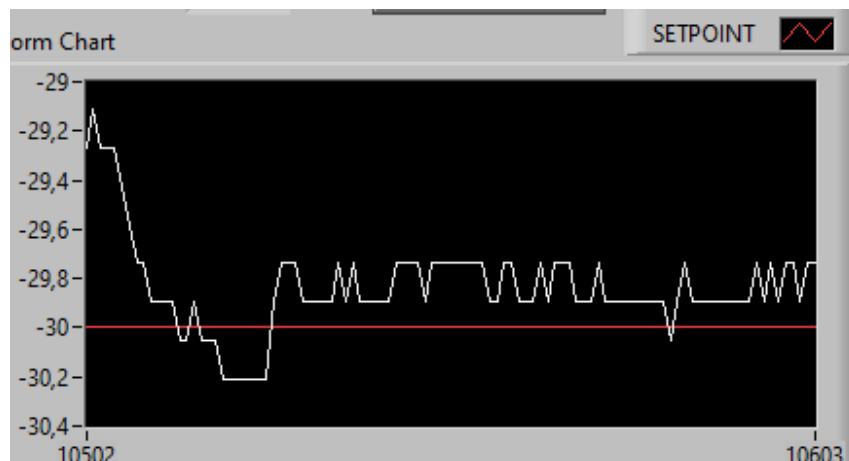


Figura 6. Control manual del eje X en -30 grados.

Conclusiones.

Se logra comunicar el Arduino y el algoritmo de control mediante el envío y recepción de datos con el Arduino. Se determina los valores para posicionar el balancín en diferentes grados en el eje X incrementando o disminuyendo el ancho de pulso de los motores, sin llegar al punto muerto en el cual los motores se apagan.

Recomendaciones.

Para lograr un mayor control manual para fijar la planta en el punto de ajuste deseado del eje X, se debe disminuir la escala del control de cada dial.

Anexos



Práctica #3.vi



GUIA DE PRÁCTICA DE LABORATORIOS

Carrera: Ingeniería
Electrónica.

Asignatura: Teoría de control

Nro. Práctica

4

Práctica 4: Posicionamiento del sistema en sentido horizontal y sentido vertical.

Objetivos:

- Identificar el tipo de control que se necesita para el posicionamiento en el eje X y en el eje Y.
- Determinar dos tipos de ancho de pulso para el posicionamiento en el eje X.
- Determinar dos tipos de ancho de pulso para el posicionamiento en el eje Y.

Instrucciones.

- Inicializar los motores.
- Introducir los valores de ancho de pulso para que los motores funcionen.
- Determinar en qué valores de ancho de pulso el motor se posiciona en dos diferentes tipos de mediciones para el eje en X y para el eje en Y.

Actividades por desarrollar:

- Realizar el respectivo proceso para inicializar los motores.
- Determinar el valor de ancho de pulso en el cual los motores empiezan a funcionar.
- Determinar dos tipos de valores de ancho de pulso en el cual el balancín se posicione en dos diferentes mediciones para el eje en X y para el eje en Y.

1. En el diagrama de bloques se realiza la comunicación entre la interfaz y LabVIEW con los módulos de LINX open y close.
2. Se coloca los módulos “set duty cycle” dentro del while loop donde se transmite los valores de ancho de pulso a los motores.
3. Se coloca un indicador gráfico, para mostrar los valores de ancho de pulso.

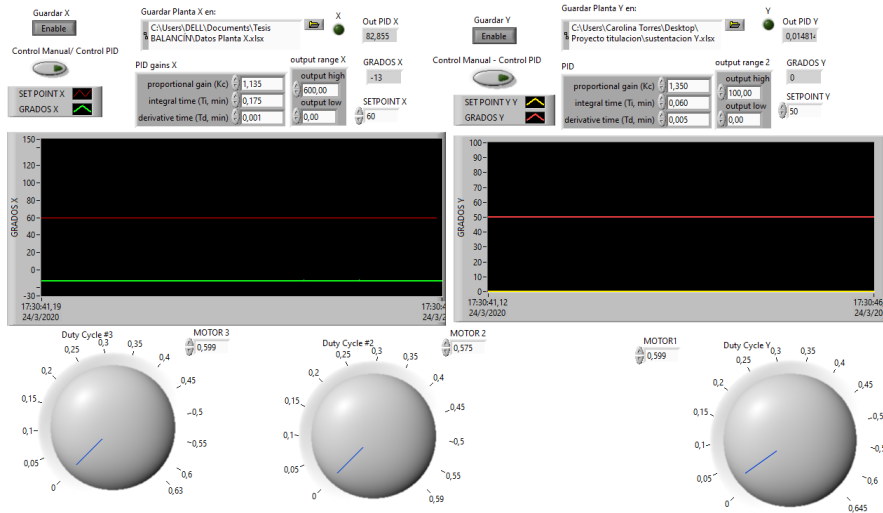


Figura 1. Panel frontal para el control manual en el eje en X y el eje en Y.

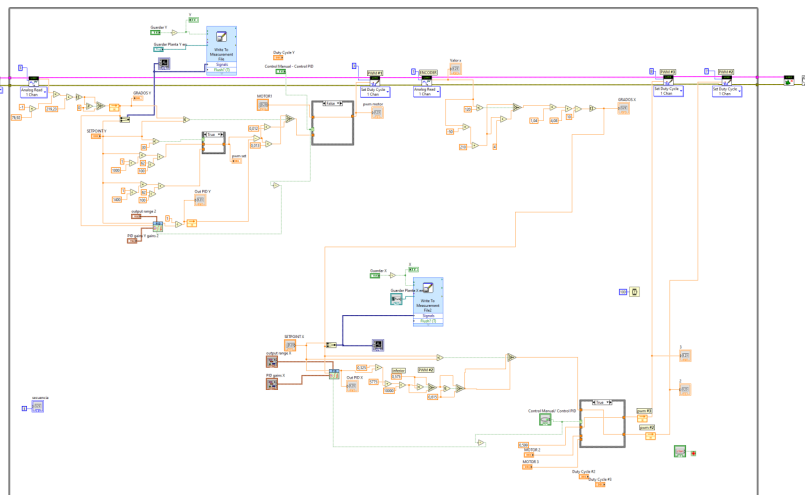


Figura 2. Diagrama de bloques para el control manual en el eje en X y el eje en Y.

4. Se inicializan los motores con los valores de ancho de pulso que se detalla en la figura a continuación.

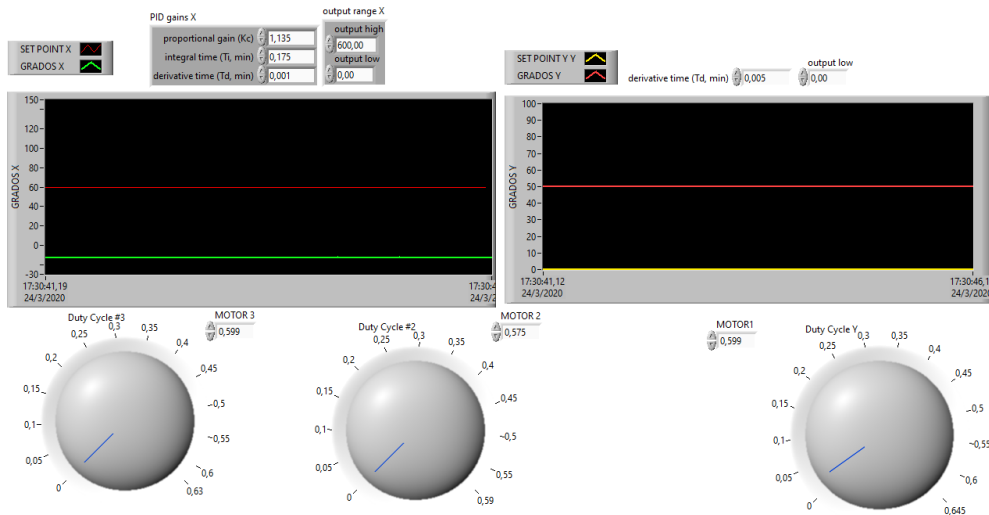


Figura 3. Control manual de ancho de pulso para la inicialización de los motores

5. Posterior a la inicialización se determina los valores en el cual se puede posicionar al balancín en 2 tipos de puntos de ajustes diferentes.

Resultados

Después de realizar el diagrama de bloques se inicializa los motores colocando en el control de ancho de pulso el valor de 0.580, posterior con el rodillo del mouse se disminuye el valor a 0 o hasta escuchar el sonido que notifica que los tres motores ya se inicializo.

Determinar los valores de ancho de pulso para posicionar la planta en: 0 y 30 grados en el eje X y en el eje Y.

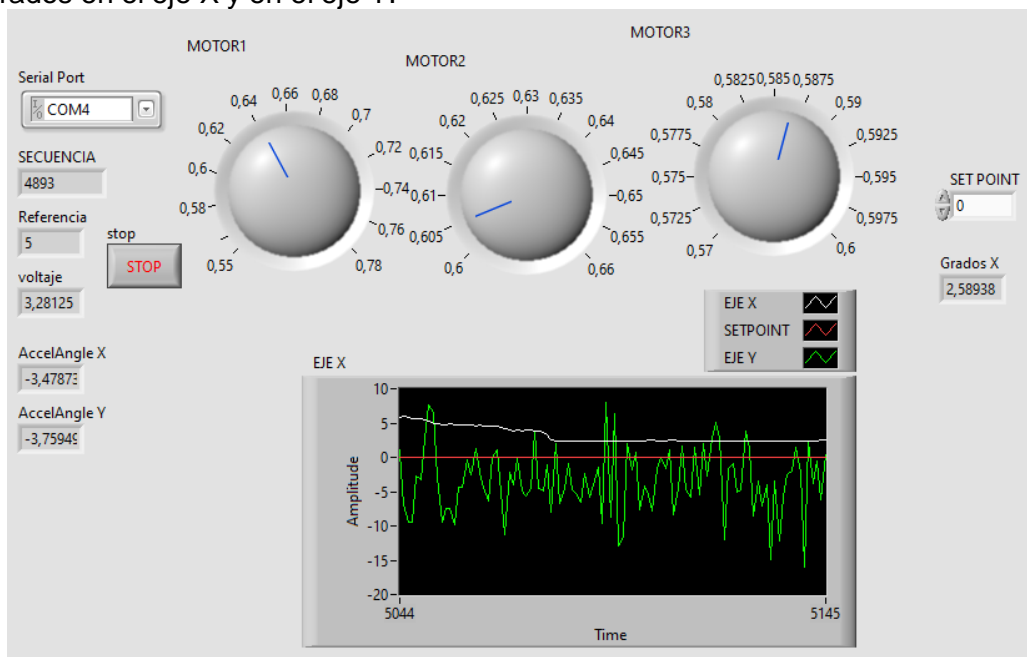


Figura 4. Control manual del eje X y el eje Y en 0 grados.

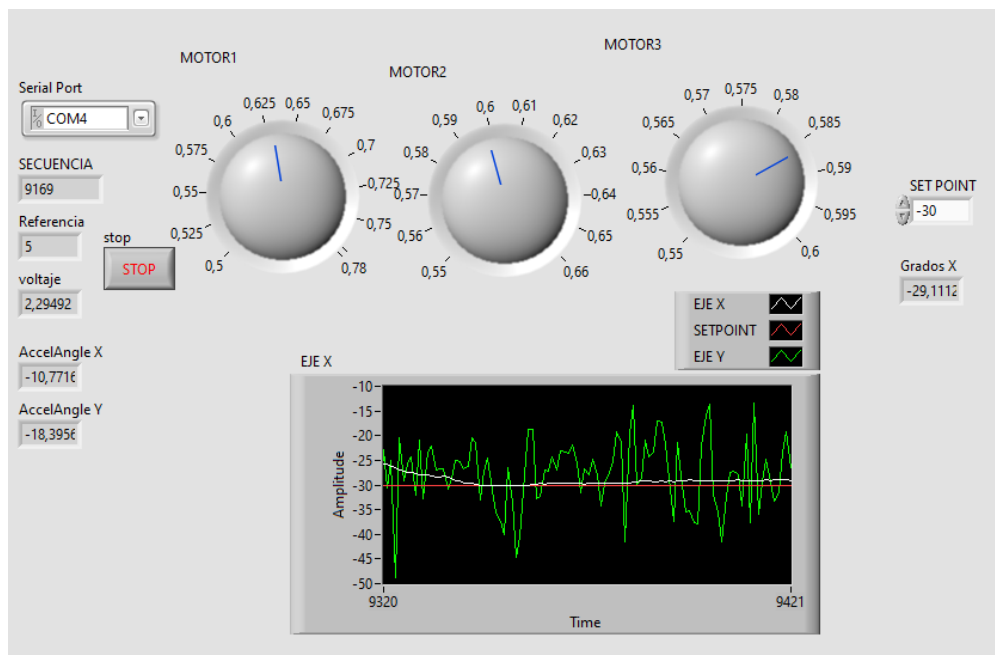


Figura 5. Control manual del eje X y el eje Y en 30 grados.

Conclusiones.

Se logra comunicar el Arduino y el algoritmo de control mediante el envío y recepción de datos con el Arduino, se determina los valores para posicionar el balancín en diferentes grados en el eje X y en el eje Y, incrementando o disminuyendo el ancho de pulso de cada motor sin llegar al punto muerto en el cual los motores se apagan.

Recomendaciones.


Para lograr un mayor control manual para fijar la planta en el punto de ajuste deseado del eje X, se debe disminuir la escala del control de cada dial.

Anexos



Práctica #4.vi

[Proyecto titulación\prácticas\Práctica #4.vi](#)

		GUIA DE PRÁCTICA DE LABORATORIOS
Carrera: Ingeniería Electrónica.		Asignatura: Teoría de control
Nro. Práctica	5	Práctica 5: Adquisición de datos del sistema en 4 diferentes puntos de ajuste.
<p>Objetivos:</p> <ul style="list-style-type: none"> • Desarrollar el algoritmo de control para la adquisición de datos del sistema • Almacenar los datos adquiridos del sistema en un archivo Excel. • Realizar una gráfica con los datos obtenidos de la planta. 		
Instrucciones.		<ul style="list-style-type: none"> • Crear el control para el almacenamiento de datos. • Almacenar los datos en un archivo Excel. • Obtener la gráfica con los datos obtenidos en el sistema.
<p>Actividades por desarrollar:</p> <ul style="list-style-type: none"> • Realizar el respectivo proceso para inicializar los motores. • Desarrollar el control para almacenar los datos adquiridos. • Obtener la gráfica de respuesta del sistema. 		

1. En el diagrama de bloques se realiza la comunicación entre la interfaz y LabVIEW con los módulos de LINX open y close.
2. Se coloca los módulos “set duty cycle” dentro del while loop, donde se transmite los valores de ancho de pulso a los motores.
3. Para la variación del ancho de pulso se crea un control a la entrada del módulo “set duty cycle”, en el cual se imprime valores dentro del rango de 0 a 1.
4. Se coloca un indicador gráfico para mostrar los valores de ancho de pulso.
5. Se coloca los módulos de almacenamiento de datos de la siguiente manera:
6. Se da clic izquierdo en el diagrama de bloques y se selecciona la función File I/O.

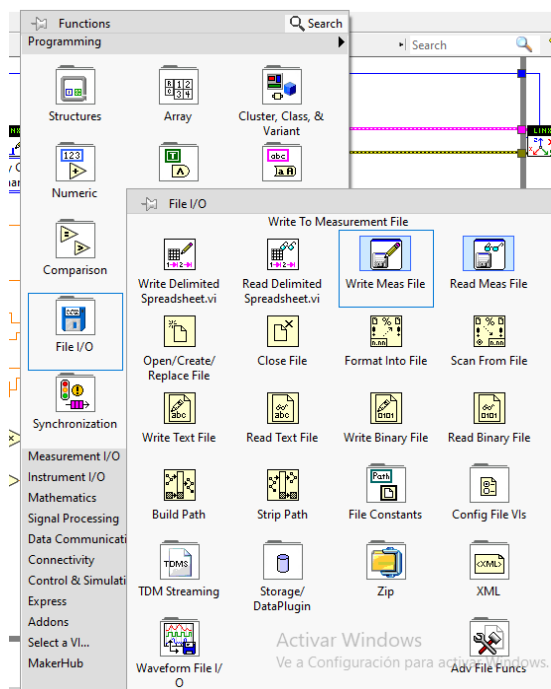


Figura 1. Creación de Módulo Write Measurement File.

- Se selecciona el módulo Write Measurement File para cada sentido de giro.
- Se configura el módulo Write Measurement File para guardar los datos en un archivo Excel.

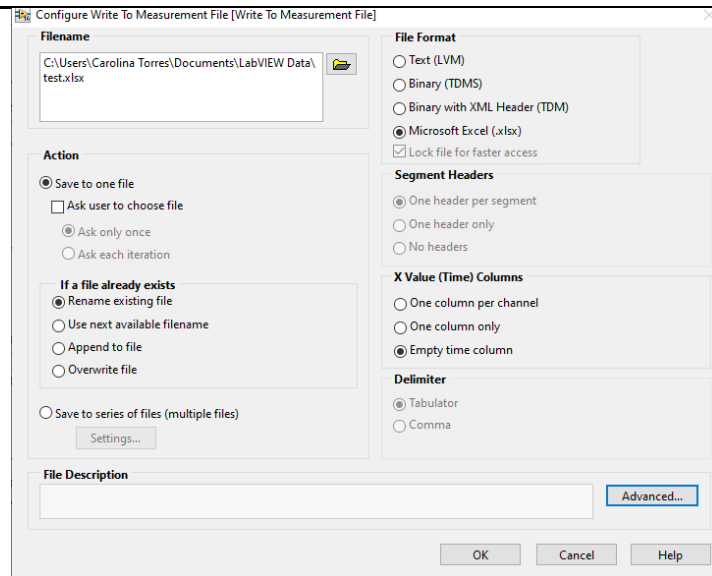


Figura 2. Configuración write to measurement.

7. Se conecta cada módulo write measurement a cada salida de los sensores de medición.

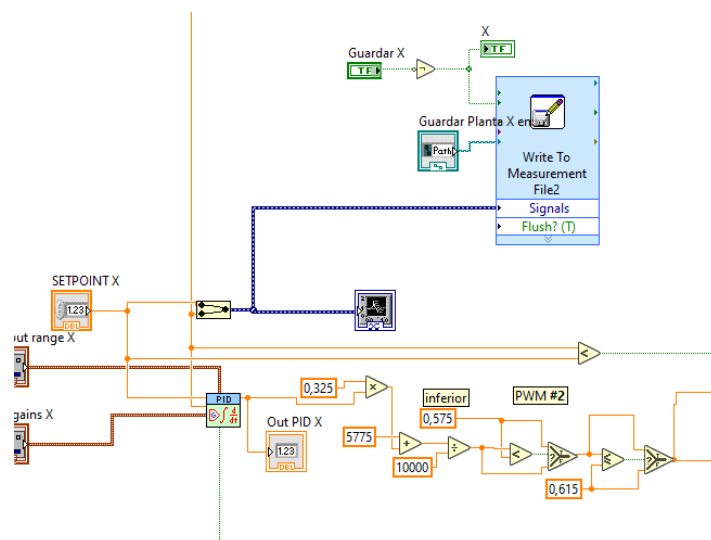


Figura 3. Diagrama de bloques para almacenar los datos obtenidos en LabVIEW.

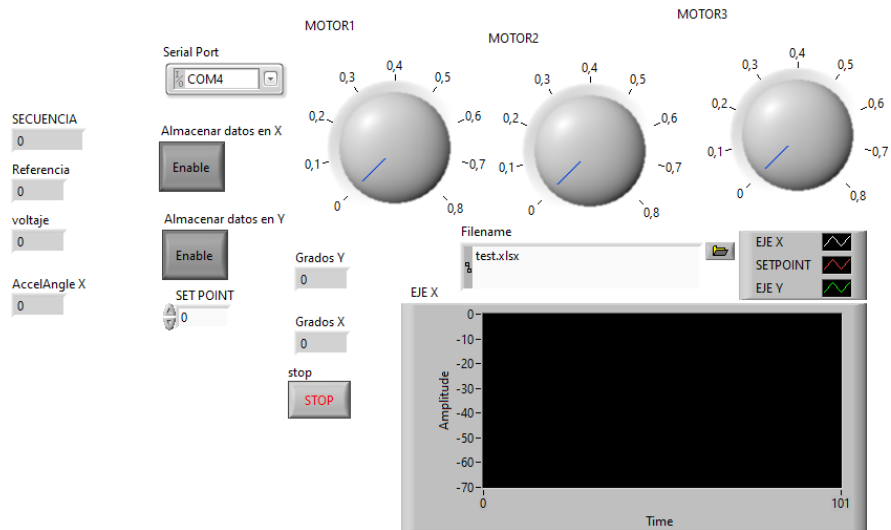


Figura 4. Panel frontal para la obtención y almacenamiento de datos.

	A	B	C	D
1			Time 1	PID 1
2			0	-50,045747
3			1	-51,202201
4			2	-50,151705
5			3	-48,751461
6			4	-41,928451
7			5	-39,714797
8			6	-35,890873
9			7	-32,703083
10			8	-28,067428
11			9	-25,124814
12			10	-21,357202
13			11	-19,019001
14			12	-22,199933
15			13	-24,076779
16			14	-26,588575
17			15	-27,369888
18			16	-27,638634
19			17	-27,884563
20			18	-27,586465
21			19	-27,298291
22			20	-27,497426
23			21	-27,394962
24			22	-28,347467
25			23	-28,977486
26			24	-28,958353
27			25	-28,832811

Figura 5. Archivo de Excel con los datos obtenidos en LabVIEW.

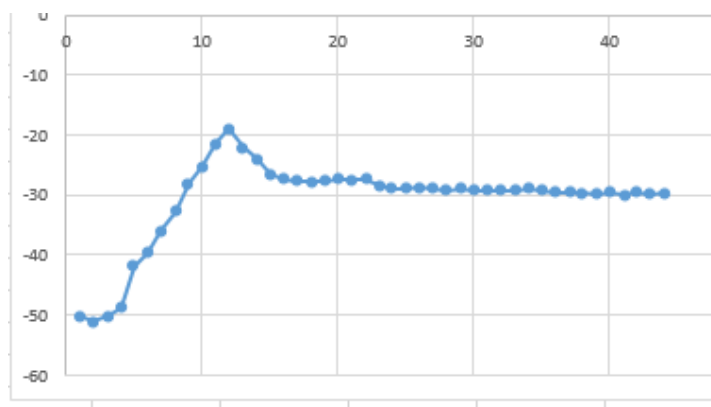


Figura 6. Gráfica en Excel con los datos obtenidos en LabVIEW.

Conclusiones.

Se almacena los datos en Excel los valores de salida del sistema, los datos obtenidos se guardan en dos columnas, la primera representa el tiempo de

muestreo y la segunda columna representa los valores de salida de la planta, con los valores obtenidos en Excel se pueden graficar y apreciar el comportamiento en la planta.

Recomendaciones.

Se coloca un control on/off en el módulo write measurement file para controlar el momento desde que se desea almacenar y prevenir que el módulo de almacenamiento guarde los valores desde que se reproduce el sistema.

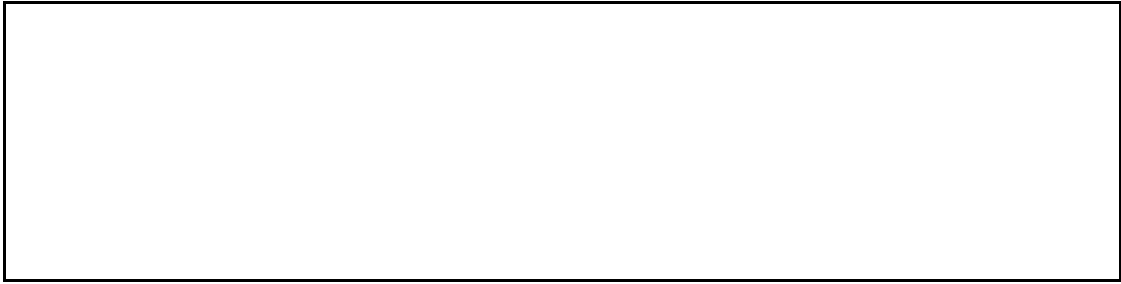
Para obtener una gráfica más precisa en Excel, se disminuye el tiempo de retardo en cada iteración para guardar valores consecutivos que mide cada sensor.


Anexos



Práctica #5.vi

[Proyecto titulación\prácticas\Práctica #5.vi](#)



		GUIA DE PRÁCTICA DE LABORATORIOS
Carrera: Ingeniería Electrónica.		Asignatura: Teoría de control
Nro. Práctica	6	Práctica 6: Diseño del modelado matemático en Matlab con los datos obtenidos en LabVIEW.
Objetivos: <ul style="list-style-type: none">• Determinar los datos necesarios para la obtención de la función de transferencia.• Almacenar los datos del sistema para la obtención de la función de transferencia.• Obtener la función de transferencia con los datos almacenados en LabVIEW.		
Instrucciones.	<ul style="list-style-type: none">• Crear el control para el almacenamiento de datos.• Almacenar los datos en un archivo Excel.• Obtener la función de transferencia con los datos obtenidos en LabVIEW mediante la herramienta Ident.	
Actividades por desarrollar: <ul style="list-style-type: none">• Realizar el respectivo proceso para inicializar los motores.• Desarrollar el control para almacenar los datos adquiridos.		

- Obtener la función de transferencia.

1. Se abre el software de Matlab y se realiza el procesamiento de los datos del sistema que se obtienen en LabVIEW.

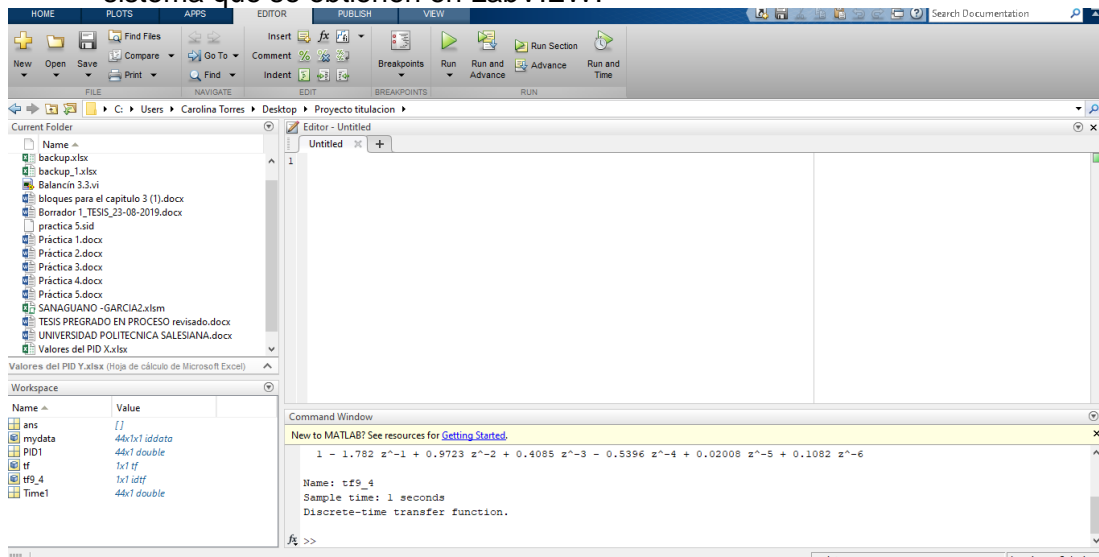


Figura 1. Vista de inicio de Matlab.

2. Se da clic izquierdo en el icono de import data y se selecciona el archivo que contiene los datos adquiridos en LabVIEW.

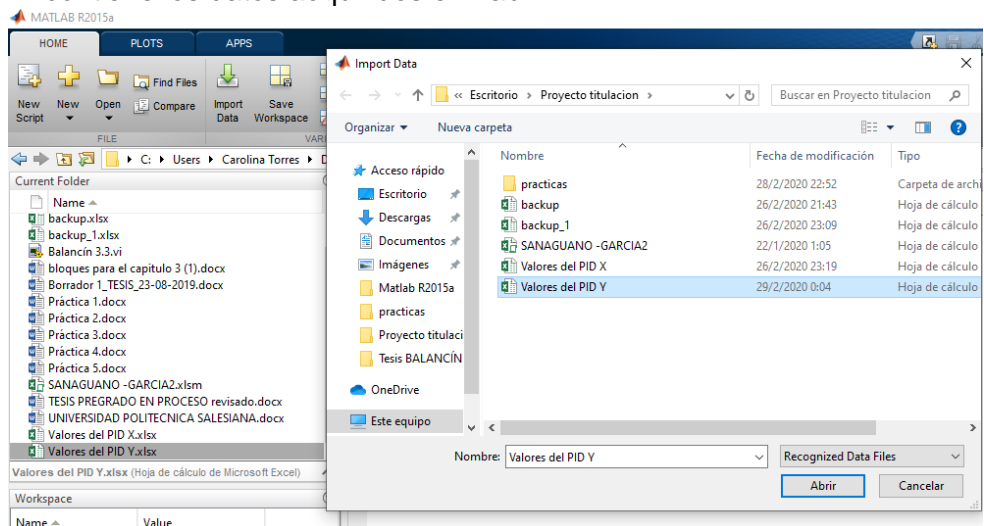


Figura 2. Ventana para importar el archivo Excel con los datos adquiridos en LabVIEW.

3. Se selecciona el archivo que contiene los datos en la current folder e importamos los datos.
4. En la ventana de importar datos nos aparece de columnas de datos, la primera que contiene el tiempo de muestreo y la segunda que contiene los datos de salida del sistema, se seleccionan los datos en las columnas y se importan los datos al workspace.

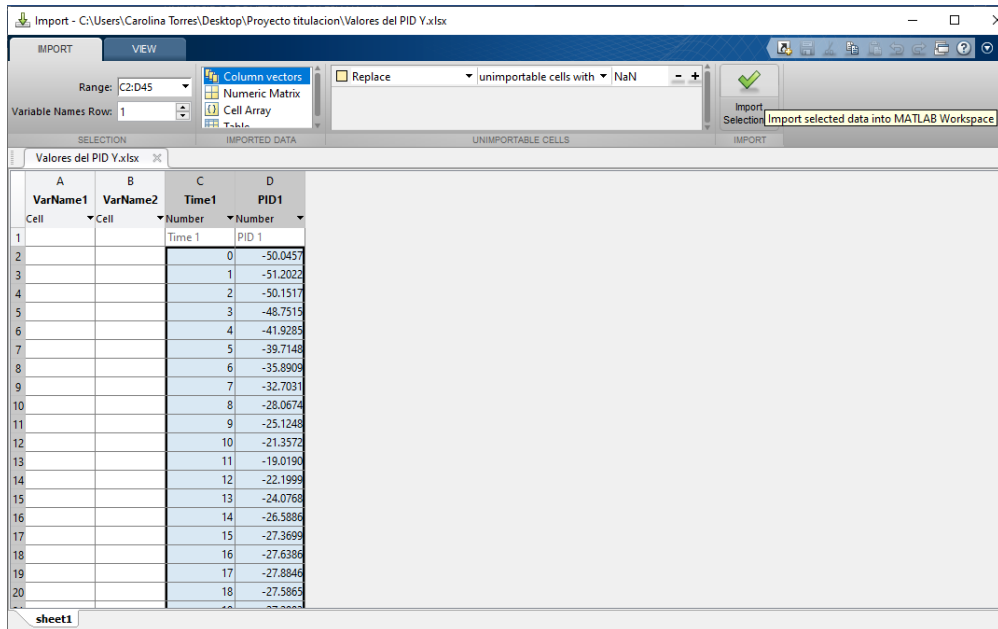


Figura 3. Ventana para importación de datos al workspace de Matlab.

5. En el command window se escribe la palabra ident, para abrir la herramienta de System identification.

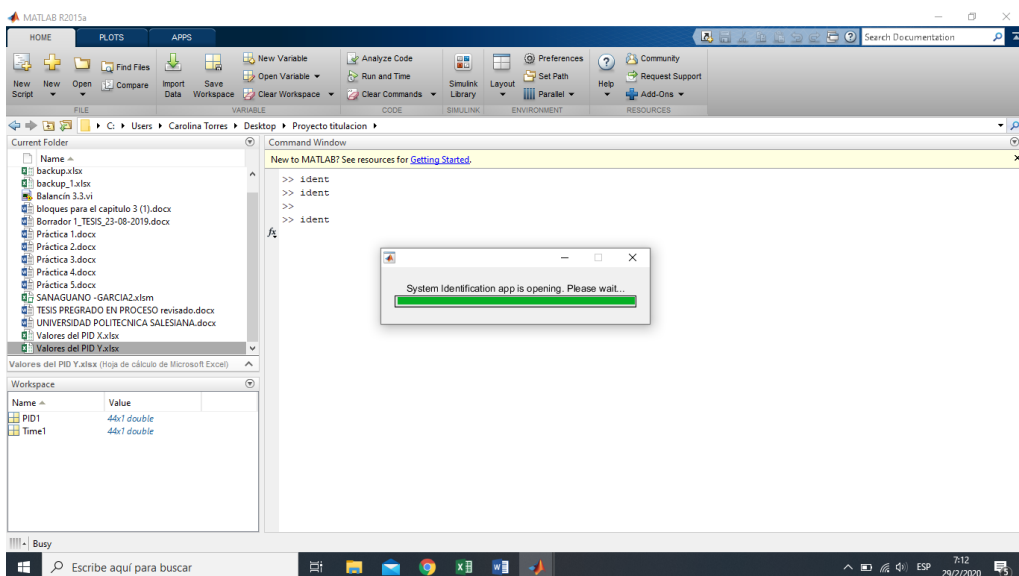


Figura 4. Inicialización de la herramienta Ident.

- En el "data format for signal" se escogen los datos de input y output desde las variables del workspace.

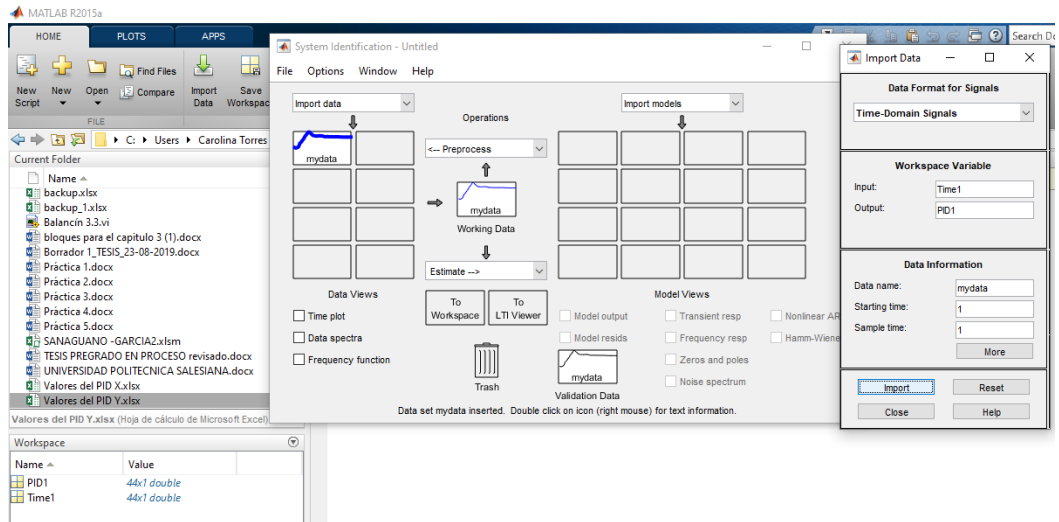


Figura 5. Selección de datos de entrada y salida para identificación del sistema.

- Se realiza una estimación para obtener un modelo de salida con un valor máximo o cercano al 100% y se los exporta la workspace.

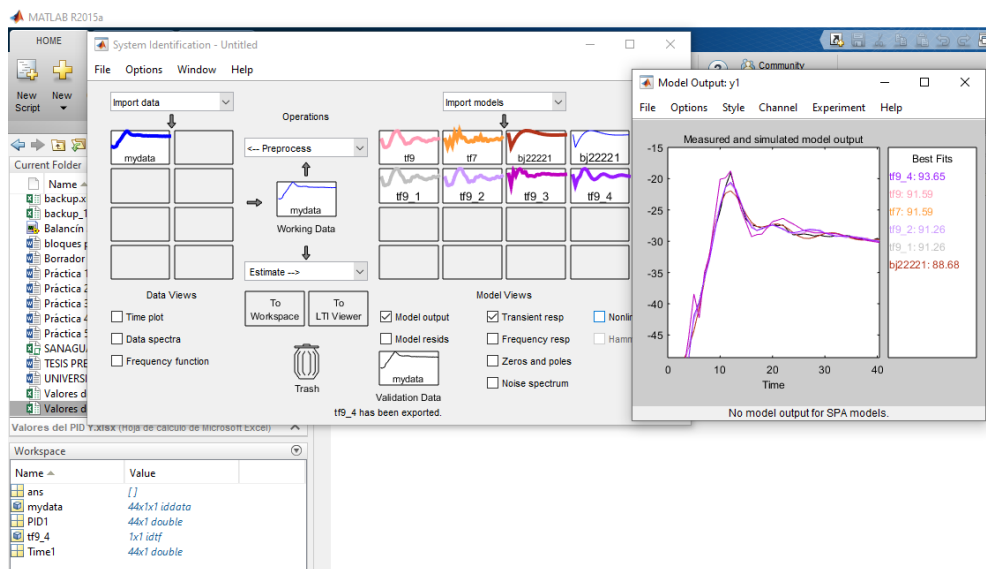


Figura 6. Estimación del modelo de salida del sistema

- En el comando Window con los datos obtenidos en la estimación del sistema, se obtiene la función de transferencia del sistema.

```

Command Window
New to MATLAB? See resources for Getting Started.

>> tf=tf(tf9_4)

tf =

From input "u1" to output "y1":
          -9.57 z^-1 + 14.3 z^-2 - 4.751 z^-3
-----
1 - 1.782 z^-1 + 0.9723 z^-2 + 0.4085 z^-3 - 0.5396 z^-4 + 0.02008 z^-5 + 0.1082 z^-6

Name: tf9_4
Sample time: 1 seconds
Discrete-time transfer function.

```

Figura 7. Obtención de la función de transferencia del sistema.

Conclusiones.

Al importar los datos a Matlab se los procesa y se obtiene la función de transferencia que rige al sistema y se obtiene la gráfica de respuesta, la función de transferencia depende de la estimación que se tenga en el momento de identificar el sistema

Recomendaciones.

Para obtener una estimación más precisa, se procura identificar el sistema con un valor superior al 90 o más cercano a 100.

Anexos




Datos Practica 6.xlsx

[Proyecto titulación\prácticas\Datos Práctica 6.xlsx](#)



Práctica #6.vi

[Proyecto titulación\prácticas\Práctica #6.vi](#)

		GUIA DE PRÁCTICA DE LABORATORIOS
Carrera: Ingeniería Electrónica.		Asignatura: Teoría de control
Nro. Práctica	7	Práctica 7: Diseño e implementación del controlador PID en el sistema.
Objetivos: <ul style="list-style-type: none"> • Almacenar los datos del sistema para la obtención de la función de transferencia. • Diseñar el diagrama de control para el sistema en el eje X y eje Y. • Determinar los valores de: K_c, T_i y T_d del controlador PID. • Implementar el controlador PID en el sistema. 		
Instrucciones.		<ul style="list-style-type: none"> • Diseñar el diagrama de control del sistema y el control para almacenamiento de datos. • Identificar los parámetros del PID para el control de estabilidad del sistema. • Implementar el controlador PID y estabilizar el sistema.
Actividades por desarrollar: <ul style="list-style-type: none"> • Realizar el respectivo proceso para inicializar los motores. • Desarrollar el control del sistema y para almacenar los datos adquiridos. • Obtener los parámetros del PID e implementarlo en el sistema. 		

1. Se diseña el algoritmo de control para los motores.
2. Se diseña el algoritmo para la medición de los grados de posición.
3. Se diseña el algoritmo para almacenar los valores adquiridos del sistema.

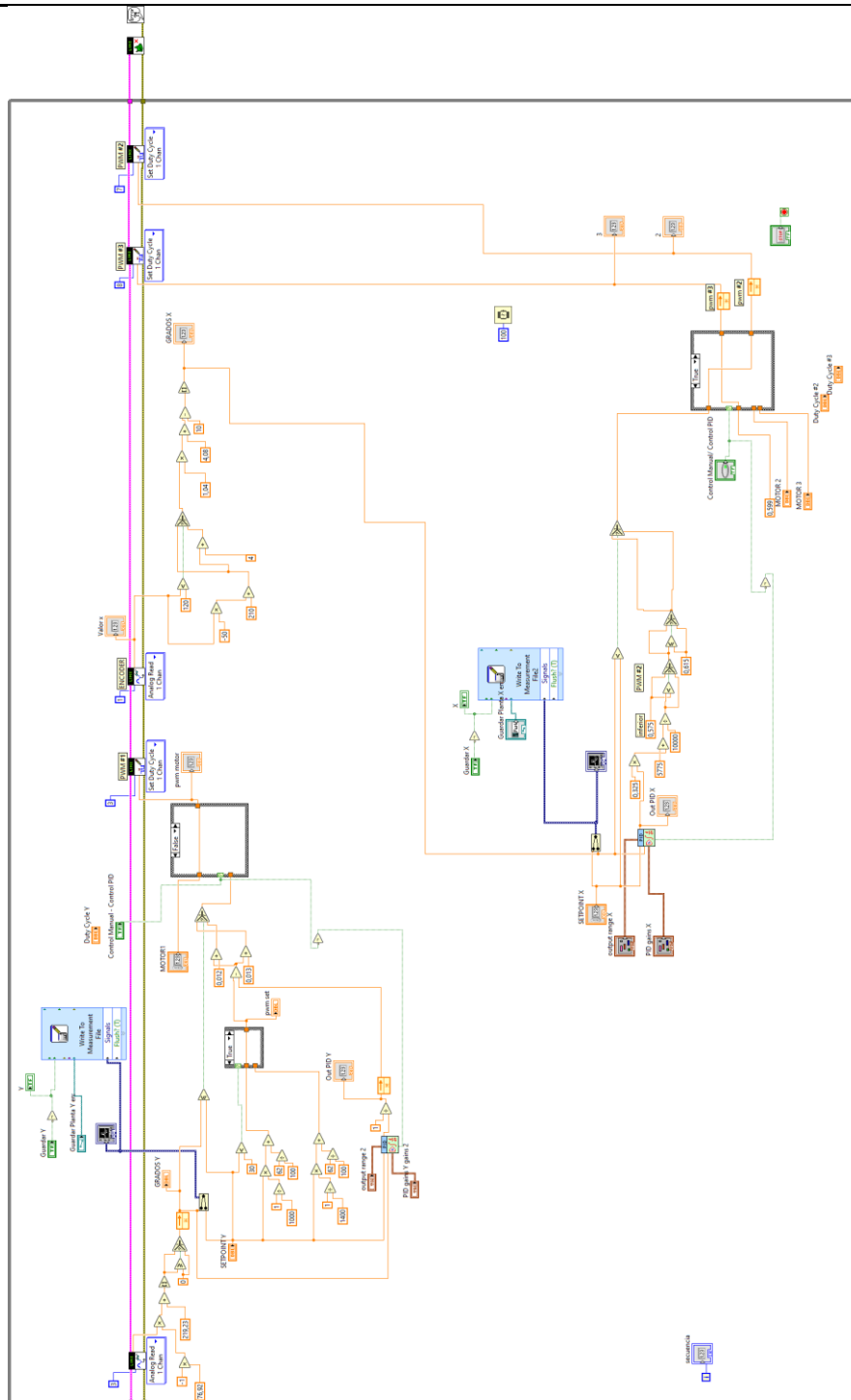


Figura 1. Diagrama de control del sistema.

4. Se realizan los pasos de la práctica 6 para la obtención de la función de transferencia del sistema.
5. Con la ayuda de Excel graficamos los valores que se adquiere del sistema en cada eje y se aplica la regla de sintonización de Ziegler-Nichols.

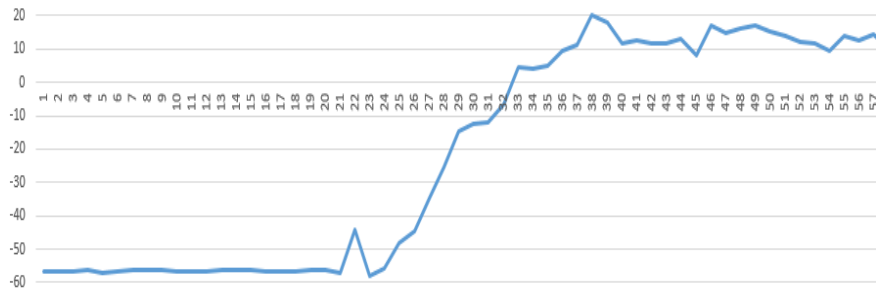


Figura 2. Gráfica de lazo cerrado para la planta de la planta en el eje Y.

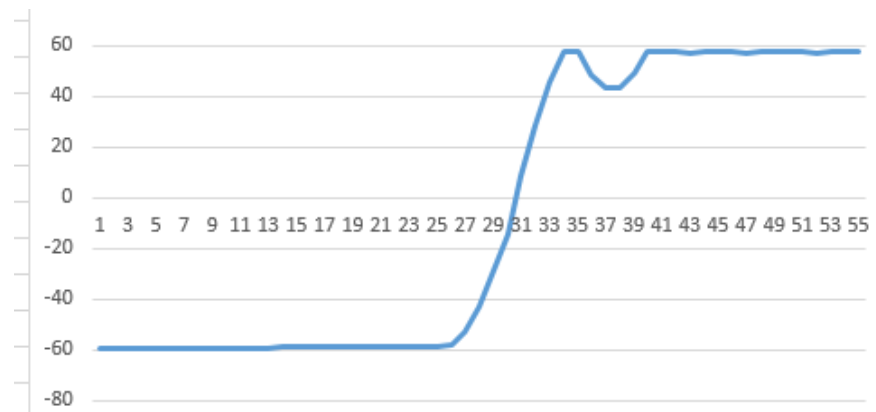


Figura 3. Gráfica de lazo cerrado para la planta de la planta en el eje X

6. Aplicando la tabla de sintonización de PID se obtiene los valores de K_c , T_i , T_d , en Y.

K_c : 1,65

T_i : 0,021

T_d : 0,0326

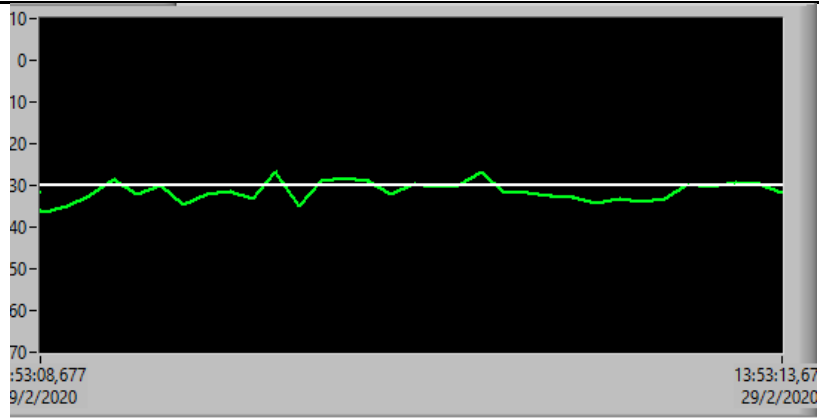


Figura 4. Estabilidad en el eje Y, su punto de ajuste de 30 grados.

7. Aplicando la tabla de sintonización de PID se obtiene los valores de K_c , T_i , T_d , en X.

K_c : 1.35, T_i : 0.075, T_d : 0.020

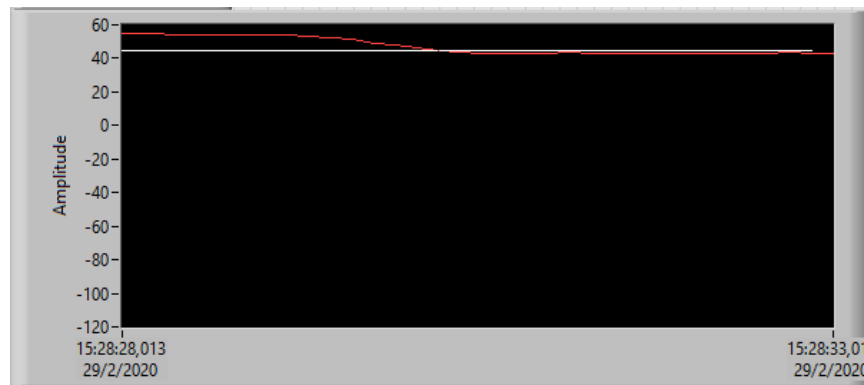


Figura 5. Estabilidad en el eje X en 40 grados.

Conclusiones.

Al implementar el PID con los valores sintonizados por Ziegler-Nichols se reduce el sobre pico y se aumenta la estabilización, el eje Y funciona correctamente desde 0 grados hasta 50 grados, y en el eje X correctamente desde los 0 grados hasta los 90 grados.

Recomendaciones.

Para una mayor precisión en la sintonización del PID, los tiempos deben ser transformados de segundos a minutos para que cumpla la regla.

Anexos



Práctica #7.vi

[Proyecto titulación\prácticas\Práctica #7.vi](#)

Anexos 3. Cronograma de actividades.

Cronograma de actividades	Semestre						Semestre					
	1	2	3	4	5	6	1	2	3	4	5	6
Análisis y diseño de la planta	■											
Adquisición de los dispositivos electrónicos de la planta		■	■									
Ensamblaje de la planta				■								
Diseño del algoritmo de control manual de la planta en LabVIEW					■							
Diseño del algoritmo de control automático de la planta en LabVIEW						■						
Prueba y obtención de datos en diferentes puntos de ajuste.							■					
Diseño del controlador PID con la sintonización de Ziegler-Nichols								■				
Implementación del controlador PID con los valores obtenidos por Ziegler-Nichols									■			
Elaboración de prácticas 1-4										■		
Elaboración de prácticas 5-7 y manual de operación											■	
Prueba final de prácticas en la planta										■	■	■

Anexos 4. Presupuesto.

Descripción	Cantidad	V. Unitario	V. Total
Arduino mega	1	\$30,00	\$30,00
Motor sin escobillas	3	\$25,00	\$75,00
Controlador de velocidad ESC	3	\$15,00	\$45,00
Diseño y construcción del módulo	1	\$800,00	\$800,00
Encoder rotacional	2	\$10,00	\$20,00
Fuente de alimentación	1	\$20,00	\$20,00
Transporte	1	\$200,00	\$200,00
Impresiones	1	\$80,00	\$80,00
Herramientas de trabajo	1	\$20,00	\$20,00
Total			\$1.290,00

Anexo 5. Estadística en el eje Y

Tabla V. Tabla de porcentaje de eficiencia del PID en el eje Y.

punto de ajuste = 20		punto de ajuste =40		punto de ajuste = 50	
# DE PRUEBA	RESULTAD O	# DE PRUEBA	RESULTAD O	# DE	RESULTAD O
1	1	1	1	1	1
2	1	2	1	2	1
3	1	3	1	3	1
4	1	4	1	4	1
5	1	5	1	5	1
6	1	6	1	6	1
7	1	7	1	7	0
8	1	8	1	8	1
9	1	9	1	9	0
10	1	10	1	10	1
11	1	11	1	11	1
12	1	12	1	12	0
13	1	13	1	13	1
14	1	14	1	14	1
15	1	15	1	15	1
16	0	16	1	16	0
17	1	17	1	17	1
18	1	18	1	18	0
19	1	19	1	19	1
20	1	20	1	20	1
21	1	21	1	21	0
22	1	22	1	22	1
23	1	23	0	23	1
24	1	24	1	24	0
25	1	25	1	25	1
26	1	26	1	26	1
27	1	27	1	27	0
28	1	28	1	28	1
29	1	29	1	29	1
30	1	30	1	30	0
31	1	31	1	31	0
32	1	32	1	32	1
33	1	33	1	33	1
34	1	34	1	34	1
35	1	35	1	35	0
36	1	36	1	36	1
37	1	37	1	37	0
38	1	38	1	38	1
39	1	39	1	39	1
40	1	40	1	40	0
41	1	41	1	41	1
42	1	42	1	42	1
43	1	43	1	43	0
44	1	44	1	44	1
45	1	45	1	45	1
46	1	46	1	46	0
47	1	47	1	47	1
48	1	48	1	48	1
49	1	49	1	49	0
50	1	50	1	50	1
TOTAL	98,00%	TOTAL	98,00%	TOTAL	68,00%

Anexo 6. Estadística en el eje X

Tabla VI. Tabla de porcentaje de eficiencia del PID en el eje X.

punto de ajuste = 0		punto de ajuste = 60		punto de ajuste =90	
# DE PRUEBA	RESULTADO	# DE PRUEBA	RESULTADO	# DE PRUEBA	RESULTADO
1	1	1	1	1	1
2	1	2	1	2	1
3	1	3	1	3	1
4	1	4	1	4	1
5	1	5	1	5	1
6	1	6	1	6	1
7	1	7	1	7	1
8	1	8	1	8	1
9	1	9	1	9	1
10	1	10	1	10	1
11	1	11	1	11	1
12	1	12	1	12	1
13	1	13	1	13	1
14	1	14	1	14	1
15	1	15	1	15	1
16	0	16	1	16	1
17	1	17	1	17	1
18	1	18	1	18	1
19	1	19	1	19	1
20	1	20	1	20	1
21	1	21	1	21	1
22	1	22	1	22	1
23	1	23	0	23	1
24	1	24	1	24	1
25	1	25	1	25	1
26	1	26	1	26	1
27	1	27	1	27	1
28	1	28	1	28	1
29	1	29	1	29	1
30	1	30	1	30	1
31	1	31	1	31	1
32	0	32	1	32	1
33	1	33	1	33	1
34	1	34	1	34	1
35	1	35	1	35	1
36	1	36	1	36	1
37	1	37	1	37	1
38	1	38	1	38	0
39	1	39	1	39	1
40	1	40	1	40	1
41	0	41	1	41	1
42	1	42	1	42	1
43	1	43	1	43	1
44	1	44	1	44	1
45	1	45	1	45	1
46	1	46	1	46	1
47	0	47	1	47	1
48	1	48	1	48	1
49	1	49	1	49	1
50	1	50	1	50	1
TOTAL	92,00%	TOTAL	98,00%	TOTAL	98,00%

Anexo 7. Fotografías



Figura 52. Instalación de motores sin escobilla en el balancín.

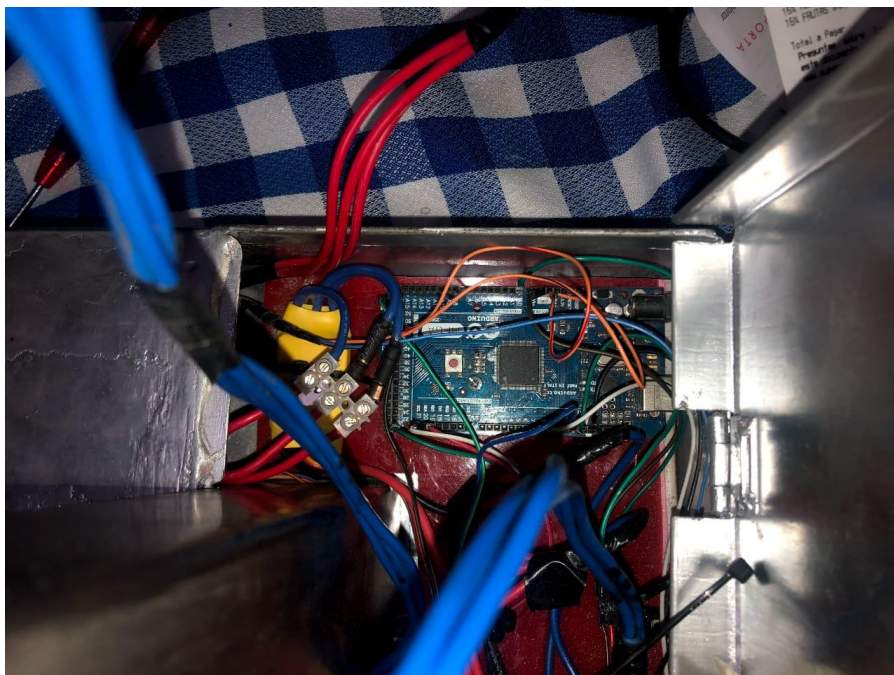


Figura 53. Conexiones de todos los componentes de la planta hacia Arduino.

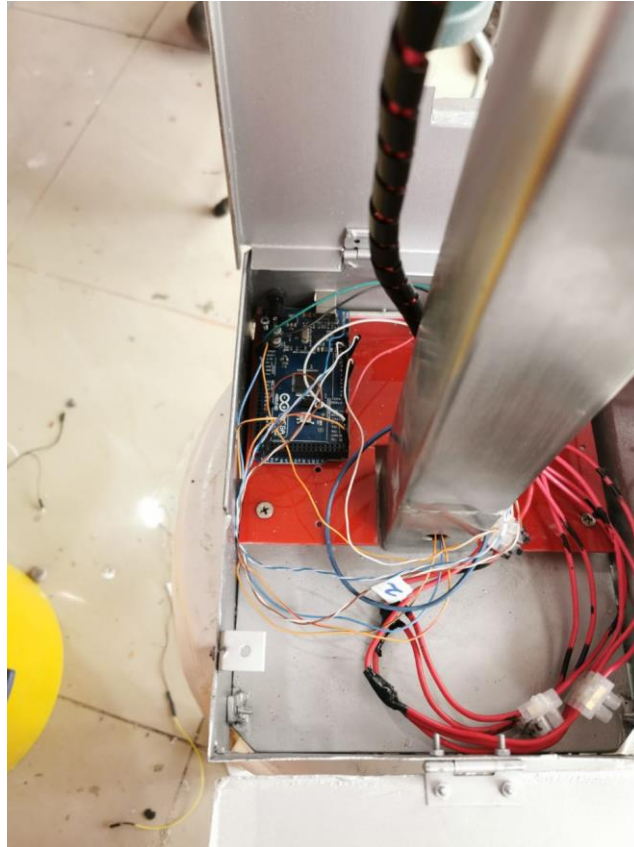


Figura 54. Cableado de motores hacia Arduino.

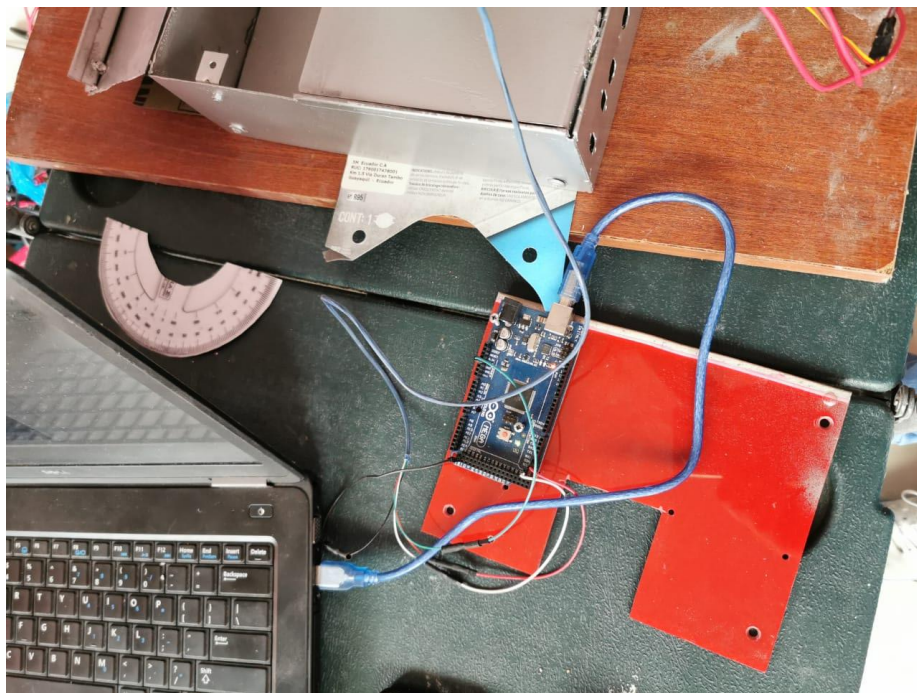


Figura 55. Pruebas de Comunicación entre Arduino y LabView.

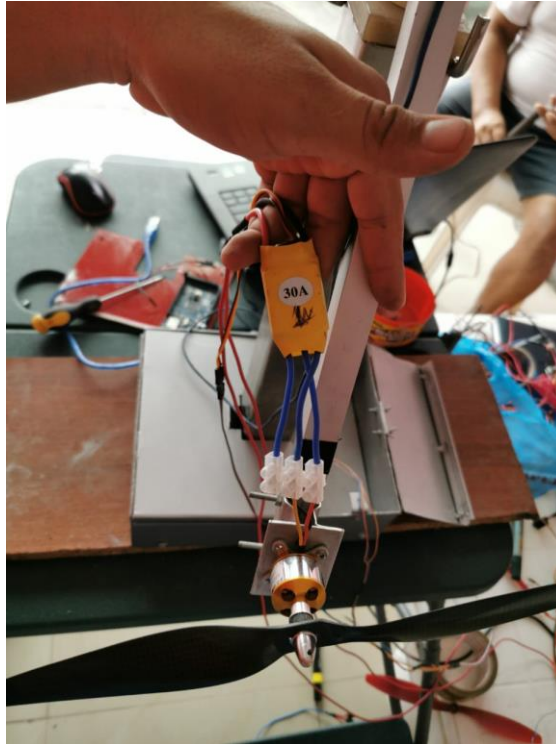


Figura 56. Prueba de motores sin escobillas con Arduino.



Figura 57. Calibración de Angulo de giro del balancín.

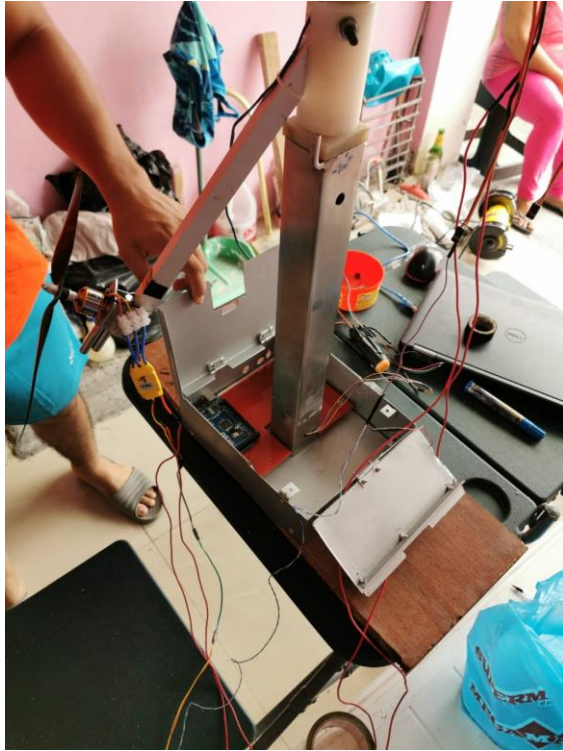


Figura 58. Diseño de estructura.



Figura 59. Conexión de driver controlador de velocidad del motor sin escobillas.



Figura 60. Montaje de motores y cableado eléctrico.



Figura 61. Primeras pruebas de funcionamiento de la planta.