

CONTROL DE VELOCIDAD DE UN SISTEMA MOTOR GENERADOR
EMPLEANDO UN CONTROLADOR PID DE DOS GRADOS DE LIBERTAD

JULIAN ANDRES PABON VARGAS
CRISTIAN DAVID OSPINO BENITEZ

UNIVERSIDAD PONTIFICIA BOLIVARIANA
ESCUELA DE INGENIERA
FACULTAD DE INGENIERIA ELÉCTRICA Y ELECTRÓNICA
FLORIDABLANCA
2019

CONTROL DE VELOCIDAD DE UN SISTEMA MOTOR GENERADOR
EMPLEANDO UN CONTROLADOR PID DE DOS GRADOS DE LIBERTAD

JULIAN ANDRES PABON VARGAS
CRISTIAN DAVID OSPINO BENITEZ

PROYECTO DE GRADO PARA OPTAR POR EL TITULO DE INGENIERO
ELECTRONICO

DIRECTOR
LUIS ANGEL SILVA PhD.

UNIVERSIDAD PONTIFICIA BOLIVARIANA
ESCUELA DE INGENIERA
FACULTAD DE INGENIERIA ELÉCTRICA Y ELECTRÓNICA
FLORIDABLANCA

2019

Nota de aceptación

Firma del director del proyecto

Firma del jurado

Firma del jurado

AGRADECIMIENTOS

Agradezco a Dios por tantas bendiciones a lo largo de los años, por darme fuerza para seguir adelante y completar mi carrera satisfactoriamente.

Quiero agradecer a mis padres por el apoyo que me brindaron durante el desarrollo de mi formación profesional, por sus aportes y consejos que me ayudaron en momentos difíciles y me alentaron a seguir adelante con mis propósitos de vida.

Agradezco a los profesores que me enseñaron no solo conocimientos académicos sino que me dieron un ejemplo a seguir a partir de su experiencia en la profesión y sus valores como persona.

A los compañeros que compartieron conmigo tantas experiencias que hicieron memorable el tiempo que pasé en la universidad.

Julián Andrés Pabón Vargas

AGRADECIMIENTOS

Primero quiero darle las más grandes gracias a Dios por todo su apoyo en este proyecto. Dedicar este proyecto a mis padres por toda la ayuda y la confianza que empeñaron en mí para que fuera un profesional como ahora que lo estoy consiguiendo. A los profesores de la universidad agradecer por compartir todos esos conocimientos que me ayudaron a alcanzar la meta. A mis amistades que me brindaron una mano cuando la necesitaba y me brindaron mucho ánimo para seguir adelante. Todo este proyecto fue dedicado a esas personas que ayudaron a que fuera un buen profesional.

Gracias a todos.

Cristian David Ospino Benítez

TABLA DE CONTENIDO

	Pág.
1. INTRODUCCIÓN	13
2. OBJETIVOS	15
2.1 Objetivo general:	15
2.2 Objetivos específicos:	15
3. MARCO TEÓRICO	16
3.1 Control PID	16
3.2 Métodos de diseño de PID	17
3.3 Control de modelo interno	18
3.4 Control PID de dos grados de libertad.....	20
3.4.1 Aplicaciones del PID-2DOF.	23
3.5 Arduino	24
3.5.1 Arduino como tarjeta de adquisición	25
3.6 Controlador Lógico Programable.....	26
3.6.1 PLC F1616 TRIANGLE RESEARCH INTERNATIONAL	27
3.7 Sistema motor-generador	29
3.7.1 Descripción del sistema motor-generador.....	29
3.7.2 Motor de corriente continua.....	30
3.7.3 Generador de corriente continua.....	30
4. IDENTIFICACIÓN DEL MODELO DEL SISTEMA	32
4.1 Descripción del montaje	32
4.2 Generación de señales y adquisición de datos	33
4.3 Identificación del modelo linealizado de la planta	36
5. DISEÑO DE LOS CONTROLADORES	38
5.1 Sintonización del controlador PID-1DOF por IMC.....	38
5.2 Diseño de controlador PID-2DOF	39
6. IMPLEMENTACIÓN PRÁCTICA Y EN SIMULACIÓN DE LOS CONTROLADORES PID-1DOF Y PID-2DOF	41

6.1 Método anti Wind-up Back Calculation.....	41
6.2 Discretización de los controladores	42
6.2.1 Discretización del controlador PID-1DOF	42
6.2.2 Discretización del controlador PID-2DOF	43
6.3 Implementación de los controladores PID-1DOF y PID-2DOF	44
6.3.1 Simulación de Controladores utilizando la herramienta StateFlow de Matlab.	44
6.3.2 Implementación de controladores PID-1DOF y PID-2DOF en el PLC F1616 BA de TRIANGLE RESEARCH INTERNATIONAL.	51
6.4 Validación de los controladores PID-1DOF y PID-2DOF.....	56
7. ANALISIS DE DESEMPEÑO DE LOS CONTROLADORES PID-1DOF Y PID-2DOF CON PRESENCIA DE PERTURBACIONES.....	61
7.1 Presencia de perturbación de entrada.....	61
7.2 Ruido de medida en la señal de realimentación.	65
7.3 Carga resistiva en la salida del generador del sistema	68
7.4 Análisis de resultados	71
8. CONCLUSIONES	74
9. BIBLIOGRAFÍA.....	76
10. ANEXOS	79

LISTA DE TABLAS

	Pág.
Tabla 1. Índices de desempeño para la trayectoria de validación 1	72
Tabla 2. Índices de desempeño para la trayectoria de validación 2	73

LISTA DE FIGURAS

	Pág.
Figura 1. Esquema del controlador PID-1DOF estándar.	17
Figura 2. Esquema del controlador IMC.....	18
Figura 3. Esquema de controlador IMC con filtro Gf.....	20
Figura 4. Controlador PID-1DOF con la parte derivativa aplicada solo a la señal realimentada.....	21
Figura 5. Esquema de control PID-1DOF separado en acciones de control.....	21
Figura 6. Esquema de controlador PID de dos grados de libertad	22
Figura 7. Directorio de librería Arduino IO en Simulink.....	25
Figura 8. Bloques de funciones de Arduino IO en Simulink.....	26
Figura 9. Ciclo de funcionamiento de un PLC	27
Figura 10. PLC F1616 BA.....	28
Figura 11. Interfaz de software TRILOGI	28
Figura 12. Ventana de edición de scripts de TRILOGI	29
Figura 13. Motor-generator	29
Figura 14. Tablero de la planta	30
Figura 15. Montaje del sistema	32
Figura 16. Señal de identificación	33
Figura 17. Señal de validación.....	33
Figura 18. Generador de señal escalonada.	34
Figura 19. Esquema de conexiones de identificación.....	35
Figura 20. Diagrama de transferencia de señales.....	35
Figura 21. Respuesta del motor a señal de identificación.	36
Figura 22. Respuesta del motor a señal de validación.	36
Figura 23. Gráfica de ajuste.....	37
Figura 24. Ajuste modelo primer orden	37
Figura 25. Diagrama de simulación de PID-2DOF	40
Figura 26. Comparación de los resultados de β	40
Figura 27. Diagrama de bloques Anti Wind-up Back Calculation.....	41
Figura 28. Controlador PID-1DOF en StateFlow.	45
Figura 29. Controlador PID-2DOF en StateFlow.	45
Figura 30. Estado de Inicio en StateFlow para ambos controladores.....	46
Figura 31. Estado de control en StateFlow para controlador PID-1DOF.	46
Figura 32. Estado de control en StateFlow para controlador PID-2DOF.	46
Figura 33. Etapa de Anti Wind-Up en StateFlow para ambos controladores.	47
Figura 34. Estado de actualización en StateFlow para controlador PID-1DOF	47
Figura 35. Estado de actualización en StateFlow para controlador PID-2DOF	47
Figura 36. Línea de Código para conversión de modelo continuo a discreto.	48
Figura 37. Estructura de un controlador discreto PID-1DOF en StateFlow.....	49
Figura 38. Estructura de un controlador discreto PID-2DOF en StateFlow.....	49
Figura 39. Respuesta del controlador discreto PID-1DOF en StateFlow a un paso unitario.	50
Figura 40. Respuesta del controlador discreto PID-2DOF en StateFlow a un paso unitario.	50

Figura 41. Respuesta de los controladores discretos en StateFlow para trayectoria de prueba.	51
Figura 42. Estructura de controlador PID-1DOF en Software Trilogi	52
Figura 43. Estructura de Controlador PID-2DOF en Software Trilogi.	53
Figura 44. Etapa de inicio en Trilogi para un controlador PID-1DOF y PID-2DOF.....	53
Figura 45. Etapa de control en Trilogi para el controlador PID-2DOF.	54
Figura 46. Etapa de Anti Wind-Up en Trilogi para un controlador PID-1DOF y PID-2DOF.	54
Figura 47. Etapa de actualización y envío de datos en Trilogi para un controlador PID-2DOF.....	55
Figura 48. Implementación de controladores en el sistema motor-generator utilizando el PLC F1616 BA.....	55
Figura 49. Respuesta temporal (a), acción de control (b), y señal de error (c) de los controladores PID-1DOF y PID-2DOF para la trayectoria pasó unitario en operación nominal.....	57
Figura 50. Trayectoria de validación 1	58
Figura 51. Trayectoria de validación 2	58
Figura 52. Respuesta temporal (a), acción de control (b), y señal de error (c) de los controladores PID-1DOF y PID-2DOF para la trayectoria 1 en operación nominal. ..	59
Figura 53. Respuesta temporal (a), acción de control (b), y señal de error (c) de los controladores PID-1DOF y PID-2DOF para la trayectoria 2 en operación nominal. ..	60
Figura 54. Escalón unitario para perturbación.....	61
Figura 55. Esquema de conexiones para perturbación	62
Figura 56. Respuesta temporal (a), acción de control (b), y señal de error (c) de los controladores PID-1DOF y PID-2DOF para la trayectoria de validación 1 con presencia de perturbación.	63
Figura 57. Respuesta temporal (a), acción de control (b), y señal de error (c) de los controladores PID-1DOF y PID-2DOF para la trayectoria de validación 2 con presencia de perturbación.	64
Figura 58. Respuesta temporal con ruido de medida (a), acción de control (b) y señal de error (c) de los controladores PID-1DOF y PID-2DOF para la trayectoria 1 de validación	66
Figura 59. Respuesta temporal con ruido de medida (a), acción de control (b) y señal de error (c) de los controladores PID-1DOF y PID-2DOF para la trayectoria 2 de validación	67
Figura 60. Respuesta temporal con carga resistiva (a), acción de control (b) y señal de error (c) de los controladores PID-1DOF y PID-2DOF para la trayectoria 1 de validación	69
Figura 61. Respuesta temporal con carga resistiva (a), acción de control (b) y señal de error (c) de los controladores PID-1DOF y PID-2DOF para la trayectoria 2 de validación.	70

RESUMEN GENERAL DE TRABAJO DE GRADO

TITULO: CONTROL DE VELOCIDAD DE UN SISTEMA MOTOR GENERADOR EMPLEANDO UN CONTROLADOR PID DE DOS GRADOS DE LIBERTAD.

AUTOR(ES): Julián Andrés Pabón Vargas
Cristian David Ospino Benítez

PROGRAMA: Facultad de Ingeniería Electrónica

DIRECTOR(A): Luis Ángel Silva

RESUMEN

En este proyecto se presenta la implementación de un sistema de control de velocidad empleando un controlador PID de dos grados de libertad (PID-2DOF) para un sistema motor-generador. Inicialmente se presenta la estrategia de control PID general y la estrategia PID-2DOF junto con el método de sintonización por IMC. Posteriormente, se realiza la identificación del sistema motor-generador empleando una trayectoria escalonada como señal de referencia. El envío y adquisición de datos se realiza mediante una tarjeta Arduino UNO junto con la herramienta Simulink de Matlab. A partir de los datos obtenidos se realiza la identificación del modelo lineal del sistema motor-generador utilizando la herramienta IDENT de Matlab. Al obtener el modelo matemático del sistema, se procede a diseñar los controladores PID-1DOF y PID-2DOF usando el método de sintonización IMC. Posteriormente se discretizan los controladores PID-1DOF y PID-2DOF por medio del método de diferencia atrás y aplicando la técnica antiwindup back calculation. La implementación se realiza de forma simulada utilizando la herramienta StateFlow de Matlab y de forma práctica por medio de un PLC. Luego, se evalúa el desempeño de los dos controladores ante la presencia de una perturbación en la entrada del sistema, presencia de ruido de medida y una carga resistiva en la salida del generador. Finalmente, se realiza un contraste cuantitativo comparando los índices de desempeño de cada controlador en las distintas pruebas. Los resultados indican que el controlador PID-2DOF presenta un mejor desempeño al realizar el seguimiento de trayectorias comparado con el controlador PID-1DOF.

PALABRAS CLAVE:

PID-2DOF, Anti Wind-up, Simulink, Stateflow, PLC, Arduino.

V° B° DIRECTOR DE TRABAJO DE GRADO

GENERAL SUMMARY OF WORK OF GRADE

TITLE: SPEED CONTROL FOR A MOTOR-GENERATOR SYSTEM USING A TWO DEGREES OF FREEDOM PID CONTROLLER.

AUTHOR(S): Julián Andrés Pabón Vargas
Cristian David Ospino Benítez

FACULTY: Facultad de Ingeniería Electrónica

DIRECTOR: Luis Ángel Silva

ABSTRACT

This project presents the implementation of a speed control system using a PID controller with two degrees of freedom (PID-2DOF) for a motor-generator system. Initially, the general PID control strategy and the PID-2DOF strategy are presented together with the IMC tuning method. Subsequently, the identification of the motor-generator system is made using a stepped path as a reference signal. The sending and acquisition of data is done through an Arduino UNO card together with the Matlab Simulink tool. The identification of the linear model of the motor-generator system using the IDENT tool of Matlab is made from the data obtained. After obtaining the mathematical model of the system, it is proceeded to design the PID-1DOF and PID-2DOF controllers using the IMC tuning method. Subsequently, the PID-1DOF and PID-2DOF controllers are discretized by means of the difference method behind and applying the Anti wind-up back calculation technique. The implementation is done in a simulated way using the Matlab StateFlow tool and in a practical way by means of a PLC. Then, the performance of the two controllers is evaluated in the presence of a disturbance in the input of the system, the presence of measurement noise and a resistive load in the output of the generator. Finally, a quantitative contrast is made comparing the performance index of each controller in the different tests. The results indicate that the PID-2DOF controller presents a better performance when tracking paths compared to the PID-1DOF controller.

KEYWORDS:

PID-2DOF, Anti Wind-up, Simulink, Stateflow, PLC, Arduino

V° B° DIRECTOR OF GRADUATE WORK

1. INTRODUCCIÓN

En el área de control de la industria existen diversas estrategias de control las cuales tienen como objetivo el mejoramiento del sistema o proceso en que se trabaja, pero todo esto depende del tipo de ajuste que se le aplique al controlador.

El controlador PID clásico o PID de 1 grado de libertad (PID 1 Degrees Of Freedom o PID-1DOF) es la estrategia de control más utilizada en la industria por múltiples factores como su robustez, su sencillo diseño al tener múltiples técnicas de sintonización, su sencilla implementación en múltiples procesos y además un bajo costo con respecto al beneficio que tiene. El problema que presenta el PID-1DOF es que por lo general, es sintonizado para cumplir una de las siguientes condiciones: seguimiento del punto de referencia o el rechazo de perturbaciones en el sistema de lazo cerrado, siendo necesario la elección de un modo de sintonía teniendo en cuenta que solo una de las condiciones puede cumplirse al tiempo. Generalmente, a nivel industrial el modo de sintonía es seleccionado para la condición de seguimiento del punto de referencia, de manera que cuando existen perturbaciones en el sistema se generan acciones de control más grandes lo que conlleva a saturaciones del elemento final de control, un mayor gasto de energía y por ende a un mayor gasto económico. En el caso del sistema motor-generador presente en el laboratorio de control de procesos de la Universidad Pontificia Bolivariana seccional Bucaramanga, se observa que una de las perturbaciones que más afecta el desempeño del sistema es la presencia de ruido en el lazo de realimentación, por lo que un controlador PID-1DOF no sería la mejor estrategia para el control de velocidad del sistema.

Se presenta la estrategia de control PID general y la estrategia PID-2DOF junto con el método de sintonización por IMC (Internal Model Control). Posteriormente, se realiza la identificación del sistema motor-generador empleando una trayectoria escalonada como señal de referencia. El envío y adquisición de datos se realiza mediante una tarjeta Arduino UNO junto con la herramienta Simulink de Matlab. A partir de los datos obtenidos se realiza la identificación del modelo lineal del sistema motor-generador utilizando la herramienta IDENT de Matlab. Al obtener el modelo matemático del sistema, se procede a diseñar los controladores PID-1DOF y PID-2DOF usando el método de sintonización IMC. Posteriormente se discretizan los controladores PID-1DOF y PID-2DOF por medio del método de diferencia atrás y aplicando la técnica antiwindup back calculation. La implementación se realiza de forma simulada utilizando la herramienta StateFlow de Matlab y de forma práctica por medio de un PLC (Programmable Logic Controller). Luego, se evalúa el desempeño de los dos controladores ante la presencia de una perturbación en la

entrada del sistema, presencia de ruido de medida y una carga resistiva en la salida del generador.

Este documento tiene la siguiente estructura. Inicialmente dentro del marco teórico se presenta el concepto de controlador PID y PID de dos grados de libertad junto con el método de sintonización por IMC. Posteriormente, se presenta la identificación del modelo del sistema. Luego, se realiza el diseño de los controladores PID-1DOF y PID-2DOF. Seguidamente, describe el método anti wind-up back calculation, luego se presenta el método de discretización de diferencia atrás y se realiza la implementación de los controladores PID-1DOF y PID-2DOF en simulación y de forma práctica en el motor-generador. Luego, se muestra el proceso de validación de los controladores y en análisis de desempeño ante perturbaciones externas. Finalmente, se presentan las conclusiones.

2. OBJETIVOS

2.1 Objetivo general:

Diseñar e implementar un control de velocidad para un sistema motor generador empleando un controlador PID de dos grados de libertad.

2.2 Objetivos específicos:

- 1) Identificar el modelo linealizado del sistema motor generador a partir de datos experimentales tomados usando Arduino como tarjeta de adquisición y la herramienta Ident de Matlab.
- 2) Diseñar un controlador PID de un grado de libertad empleando la técnica IMC para el modelo linealizado del sistema motor generador.
- 3) Diseñar un controlador PID de dos grados de libertad empleando la técnica IMC para el modelo linealizado del sistema motor generador.
- 4) Implementar los controladores PID de un grado de libertad y PID de dos grados de libertad utilizando el PLC F1616 BA de TRIANGLE RESEARCH INTERNATIONAL.
- 5) Validar el desempeño de los controladores PID de un grado de libertad y PID de dos grados de libertad ante perturbaciones de entrada, presencia de ruido de medida y variaciones de carga.

3. MARCO TEÓRICO

3.1 Control PID

EL controlador PID es la técnica de control más utilizada a nivel industrial por características como su sencillo diseño e implementación en muchos procesos. El controlador PID ha disfrutado de tanto éxito como de crítica en los sistemas dinámicos pero aun así se estima que el control PID clásico está implementado en más del 85% de todos los procesos industriales ya que proporciona un buen manejo de respuesta transitoria y de estado estacionario y porque genera soluciones a muchos problemas de control en la industria. (R. A. Paz, 2001)

El controlador PID se conforma de tres acciones de control, proporcional, integral y derivativa, que al sumarse de distintas maneras pueden dar origen a varios tipos de controladores (P, PI, PD, PID). Cada acción de control tiene una función diferente que compone el controlador PID. (K. Ogata, 2010)

El control proporcional up está matemáticamente expresado en (1), y representa el presente de la acción de control.

$$up(t) = K * e(t) \quad (1)$$

donde K es la ganancia proporcional del sistema y $e(t)$ corresponde a la señal de error.

El control integral ui está matemáticamente expresado en (2), y representa el pasado de la acción del control.

$$ui(t) = K \left(\frac{1}{T_i} \int e(\tau) d\tau \right) \quad (2)$$

donde T_i es la ganancia integral.

El control derivativo ud está matemáticamente expresado en (3), y representa el futuro de la acción de control.

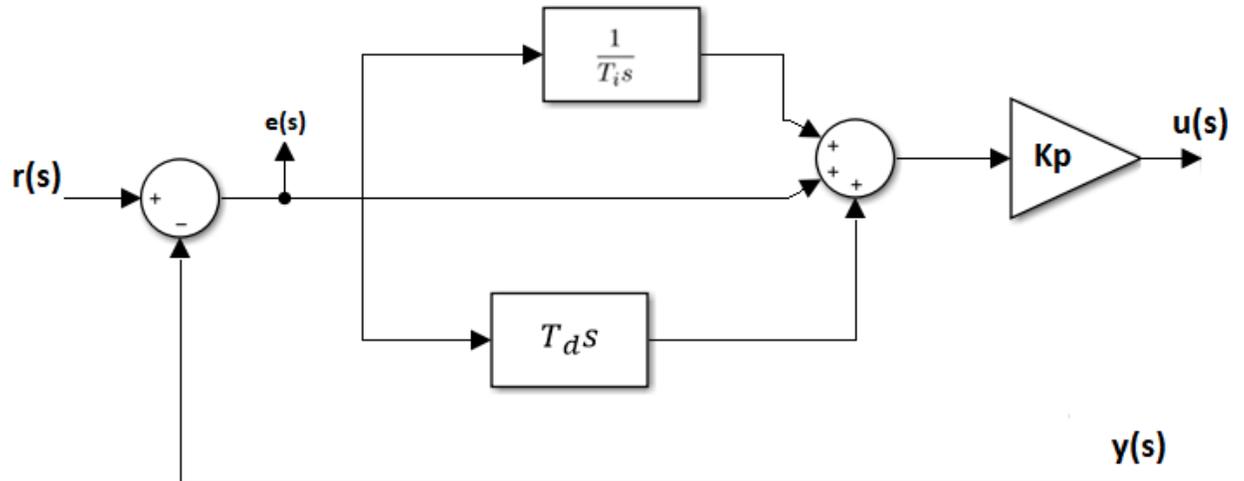
$$ud(t) = K \left(T_d \frac{de(t)}{dt} \right). \quad (3)$$

donde T_d es la ganancia derivativa del sistema.

Al combinar las tres acciones de control (1), (2) y (3), se obtiene el PID clásico expresado en la ecuación (4) y mostrado en la Figura 1.

$$u(t) = K \left(e(t) + \frac{1}{T_i} \int_0^t e(\tau) d\tau + T_d \frac{de(t)}{dt} \right) \quad (4)$$

Figura 1. Esquema del controlador PID-1DOF estándar.



3.2 Métodos de diseño de PID

El diseño de un controlador PID, consiste en determinar sus parámetros (K_p , T_i , T_d), los cuales permitan obtener una respuesta del sistema que se ajuste y sea acorde a las necesidades de desempeño requeridas. El primer paso para lograr la sintonización de un controlador consiste en identificar el comportamiento de la planta por medio de pruebas, en donde se evalúa la respuesta a una señal de entrada específica. Al obtener la dinámica del proceso, se puede proceder a escoger un método de sintonización con el cual se van a determinar los parámetros para el controlador. (K. J. Åström, 2009)

El proceso de seleccionar los parámetros del controlador que cumplan con las especificaciones de comportamiento dadas se conoce como sintonía del controlador. (V. Alfaro, 2003) Existen diferentes métodos de sintonización empleados según el modelo del sistema y sus características, algunos que pueden destacarse son: Ziegler y Nichols, Ziegler y Nichols mejorado, SIMC (Skogestad internal model control), MIGO, AMIGO, asignación de polos, IMC, entre otros. (M. Paez, 2017)

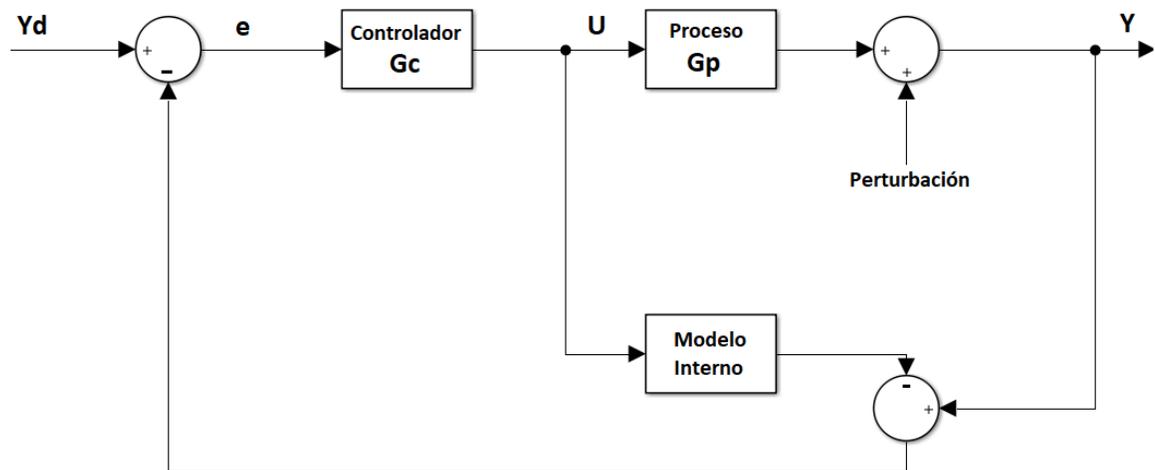
Por medio de reglas de sintonización dadas por el método de Control por Modelo Interno (IMC), se realizará la determinación de los parámetros necesarios para realizar el diseño del controlador PID.

3.3 Control de modelo interno

El control por modelo interno (IMC) es una técnica utilizada para diseñar sistemas de control basados en la Q-parametrización y además para compensar la presencia de perturbaciones en el lazo de realimentación. La función del controlador IMC es hacer un controlador que simule una compensación de la planta, cuando hay presencia de perturbaciones, por medio de un modelo interno que está dado por una aproximación al modelo de la planta y acoplado a la señal de realimentación. Su estructura está conformada por un modelo del proceso real y un controlador. Para obtener las reglas de sintonía para un controlador PID, el diseño IMC es aplicado a funciones de transferencia de bajo orden. (D.E Rivera, 2007)

Para implementar el control es necesario conocer el modelo del sistema que se quiere controlar, por tanto es necesario realizar una identificación previa de éste. En la Figura 2 se muestra una representación de la integración del modelo interno a un sistema realimentado.

Figura 2. Esquema del controlador IMC



Se tiene que Y es la salida del sistema, U corresponde a la acción de control del sistema, e es la señal de error, Yd es la entrada con un valor deseado, Gp corresponde a la planta a controlar, Gc es el controlador y el modelo interno (mi) dado por una aproximación de la planta a controlar. Al proceso se le adiciona una

perturbación, la cual afecta indirectamente a la diferencia que se genera entre el modelo interno y la realimentación.

En el proceso se tienen involucradas las siguientes ecuaciones. Donde el modelo interno es un aproximado de la función de la planta y que por medio de diagrama de bloques se obtiene que el controlador está dado por el inverso del modelo interno.

$$m_i = \widetilde{G}_p \quad (5)$$

$$\widetilde{G}_p \approx G_p \quad (6)$$

$$G_c = \frac{1}{\widetilde{G}_p} \quad (7)$$

Al tener G_c como el inverso de \widetilde{G}_p , esto produce que el sistema se comporte de forma no deseada porque se ha convertido en una función impropia, lo cual hace que su implementación no sea posible de esta manera, Para ello se debe aplicar un filtro, que adicionalmente se encargará de atenuar los efectos del error que se produzcan en el lazo de control. (Y. García, 2009)

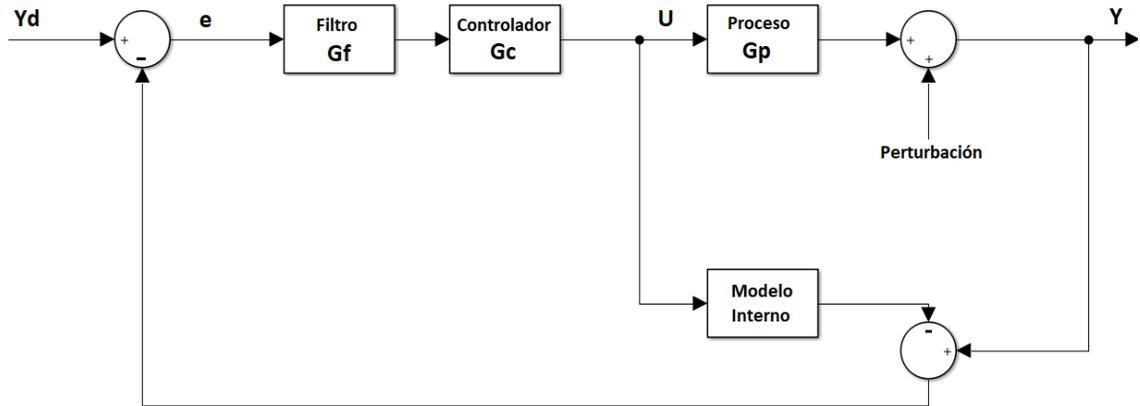
La ecuación que define al filtro está dada por (8).

$$G_f(s) = \frac{1}{(\lambda s + 1)^n} \quad (8)$$

donde λ corresponde a un parámetro que se debe determinar según las necesidades del sistema y n corresponde al orden del filtro.

Ya definido el orden del filtro y el valor del parámetro λ , puede realizarse la implementación en el modelo del proceso. En la Figura 3 se muestra la representación del modelo con el filtro aplicado.

Figura 3. Esquema de controlador IMC con filtro G_f .



De la figura anterior se puede obtener la ecuación (9).

$$U(s) = \frac{G_c(s)G_f(s)}{1-G_c(s)G_f(s)\tilde{G}_p(s)} (Yd - Y) \quad (9)$$

3.4 Control PID de dos grados de libertad

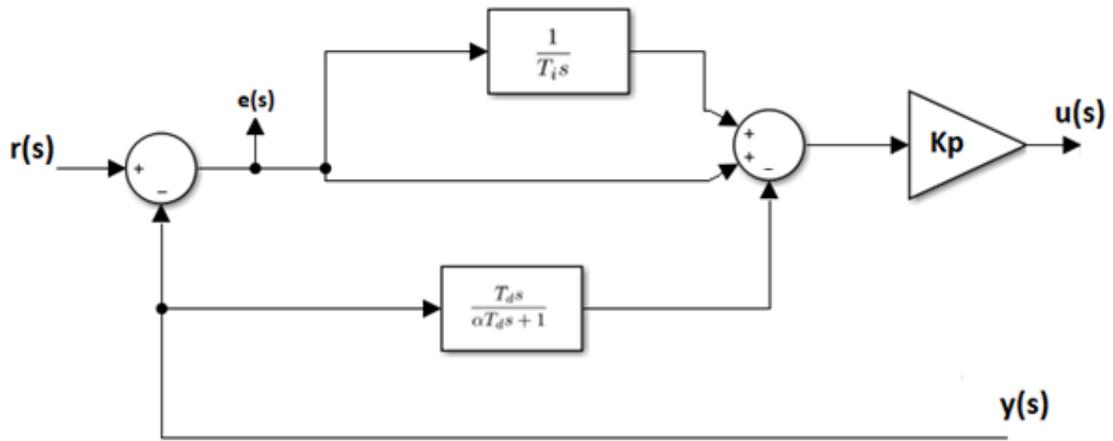
En un sistema de control PID de dos grados de libertad (PID-2DOF) se busca tener una restricción adicional del comportamiento de la variable a controlar al someter el sistema a cambios o perturbaciones. Para esto es necesario poder realizar un control independiente al establecido inicialmente por los parámetros del PID-1DOF.

Para llegar al algoritmo de control PID de dos grados de libertad, es necesario partir del análisis de la estructura del controlador PID de un grado de libertad.

Se tiene la expresión del PID-1DOF en el dominio de la frecuencia mostrada en (10) y gráficamente en la Figura 4, en el cual la parte derivativa solo se le aplica a la señal de realimentación. (V. Alfaro, 2016)

$$u(s) = K_p \left\{ r(s) - y(s) + \left(\frac{1}{T_i s} \right) [r(s) - y(s)] - \left(\frac{T_d s}{\alpha T_d s + 1} \right) y(s) \right\} \quad (10)$$

Figura 4. Controlador PID-1DOF con la parte derivativa aplicada solo a la señal realimentada



El control de un grado de libertad se puede observar en la ecuación (10) a partir de los parámetros K_p , T_i y T_d , los cuales al ser fijados determinan un comportamiento el cual no puede modificarse sin tener que cambiar dichos parámetros.

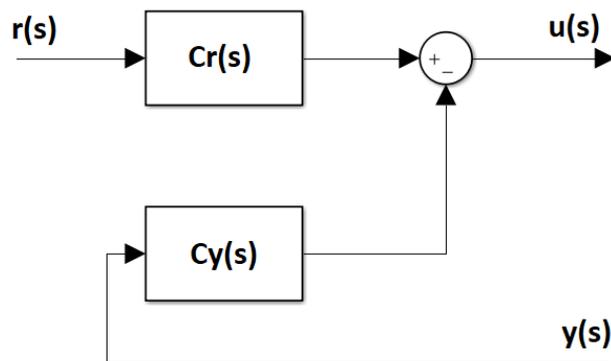
Para realizar un control PID de dos grados de libertad, se incluye el parámetro β en la ecuación mostrada en (11).

$$u(s) = K_p \left\{ \beta r(s) - y(s) + \left(\frac{1}{T_i s} \right) [r(s) - y(s)] - \left(\frac{T_d s}{\alpha T_d s + 1} \right) y(s) \right\} \quad (11)$$

Es necesario ver el algoritmo como la unión de dos controladores, por lo tanto la expresión se puede escribir como la ecuación (12) y representarla en la Figura 5.

$$u(s) = C_r(s)r(s) - C_y(s)y(s) \quad (12)$$

Figura 5. Esquema de control PID-1DOF separado en acciones de control



C_r y C_y se representan matemáticamente en las ecuaciones (13) y (14).

$$C_r(s) = K_p \left(\beta + \frac{1}{T_i s} \right) \quad (13)$$

$$C_y(s) = K_p \left(1 + \frac{1}{T_i s} + \frac{T_d s}{\alpha T_d s + 1} \right) \quad (14)$$

En este punto se observa que si se escogen los parámetros K_p , T_i y T_d , el parámetro β es independiente a estos valores fijados. Esto permite que puedan realizarse modificaciones al comportamiento del sistema ante un cambio necesario en el controlador del valor deseado C_r . Lo anterior define el segundo grado de libertad del controlador PID-2DOF.

La ecuación general del controlador PID de dos grados de libertad está dada por (15).

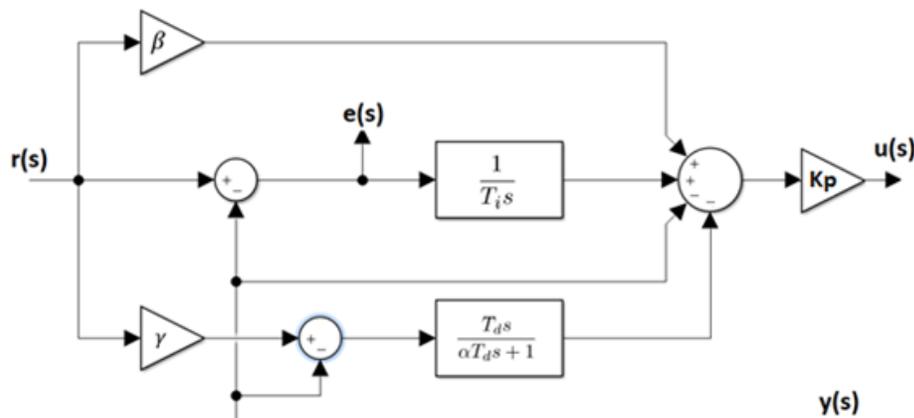
$$u(s) = K_p \left\{ \beta r(s) - y(s) + \left(\frac{1}{T_i s} \right) [r(s) - y(s)] - \left(\frac{T_d s}{\alpha T_d s + 1} \right) [\gamma r(s) - y(s)] \right\} \quad (15)$$

donde β es el factor de peso del valor deseado y γ es el factor de peso derivativo.

Los valores de estos parámetros deben ajustarse a medida de la necesidad de las modificaciones que quieran hacerse a la respuesta del sistema. El valor de β puede variar entre 0 y 1, en donde el valor de 1 convierte al controlador en un PID de un grado de libertad, al no afectar la parte del valor deseado proporcional. El parámetro γ es conveniente que se fije en 0 para evitar cambios bruscos en la parte derivativa ante entradas tipo escalon, lo que hace que la acción derivativa se aplique solo a la señal realimentada. (V. Alfaro, 2016)

El esquema general del controlador PID de dos grados de libertad se representa en la Figura 6.

Figura 6. Esquema de controlador PID de dos grados de libertad



3.4.1 Aplicaciones del PID-2DOF.

El controlador PID-2DOF fue propuesto con el fin de mejorar las características de desempeño y estabilidad robusta del sistema control ante la presencia de perturbaciones externas, permitiendo corregir el comportamiento del sistema de acuerdo a las necesidades del usuario.

En (M. Kumar, 2015) se presenta la sintonización de controladores PID-2DOF para procesos de segundo orden sin retardo haciendo una comparación en rendimiento con un controlador PID-1DOF, el resultado se mostró favorable al PID-2DOF ya que presentaba un sobre impulso mucho menor y además tenía mejor rechazo a las perturbaciones en la salida a las que fue sometido el sistema.

En (V. M. Alfaro, 2009) se muestra la sintonización robusta para controladores PI/PID de dos grados de libertad en cascada para sistemas de primer y segundo orden con retardo. Los resultados muestran que para un diseño de controlador de este tipo se garantiza un sistema más robusto frente a perturbaciones en la salida que se puedan presentar.

En 2009, para un congreso anual de la asociación de México de control automático se presentó (R. J. Vazquez, 2009), donde se utilizó un controlador PID-2DOF para sistemas inestables con retardo usando un enfoque polinomial.

En (V. M. Alfaro, 2009), se hicieron pruebas de validación al controlador PID-2DOF con un conjunto de plantas sugeridas por Åström and Hägglund (2000). El diseño del controlador se realizó a partir de la minimización de funciones de coste de la señal de error y de esfuerzo de control. Los resultados arrojaron que para procesos donde se tienen cambios en la entrada, como perturbaciones, el PID-2DOF permite tener un mejor funcionamiento y además menor gasto de energía con respecto al controlador PID-1DOF.

En (G. L. Deshmukh) se presenta un diseño para un sistema de control de temperatura con un controlador PID-2DOF. Se concluyó que el controlador PID-2DOF cumple tanto el seguimiento del punto de referencia como el rechazo de perturbaciones de carga.

El control de velocidad para un motor DC sin escobillas utilizando un controlador PID-2DOF y la técnica Anti-Windup es presentado en (A. K. Kadhim, 2017). La estrategia de control se diseñó para un control de velocidad por medio de PWM donde se parametrizó digitalmente a partir de la herramienta de Matlab-Simulink. A partir de resultados se comprobó que el PID-2DOF mostraba mejores comportamiento que el PID-1DOF.

En (M. E. Hmidi, 2017) se propone un modelo dinámico y un enfoque de control PID-2DOF sistemático y robusto para una fase térmica del tren de transmisión para vehículos híbridos.

En el campo de la robótica se tienen los siguientes trabajos. En (O. Y. Tenga, 2015) se presenta un controlador PID-2DOF para el control de posición de un servo motor aplicado a una mesa XY. Los resultados mostraron un mejor desempeño del controlador PID-2DOF con respecto a los controladores PID-1DOF y PI.

Una comparación en tiempo real entre los controladores PID-1DOF y PID-2DOF para un brazo de robot de 6 grados de libertad es presentada en (N. G. Adar, 2016). Los controladores fueron implementados usando la herramienta Simulink de Matlab para tener una interacción en tiempo real entre los movimientos del robot y el computador. Los resultados experimentales mostraron que un PID-2DOF tiene un mejor desempeño que un PID-1DOF para el control del manipulador en tiempo real.

Finalmente, trabajos que emplean el sistema motor-generator que se encuentra en la Universidad Pontificia Bolivariana seccional Bucaramanga se presentan a continuación. En (L. Silva, 2017) se presenta el diseño y análisis estadístico de robustez de controladores FOPID, IOPID y SIMC PID aplicado al control de velocidad del sistema motor-generator. Los controladores fueron implementados por medio de la herramienta StateFlow de Matlab y con la tarjeta de adquisición NI 6008. Los resultados mostraron una mayor robustez en el controlador FOPID al tener menos afectación a variaciones de carga pero con mayor afectación ante la presencia de perturbaciones de entrada y ruido de medida en el lazo de realimentación.

En el proyecto de grado (M.Paez, 2017), se presenta la comparación entre las técnicas de diseño por asignación de polos y control por modelo interno (IMC) para el control de velocidad de un sistema motor-generator. Se analizó el desempeño de los controladores ante la presencia de perturbaciones de entrada, ruido de medida y variaciones de carga. Se concluyó que el controlador diseñado por la técnica IMC tenía mayor robustez para el seguimiento de trayectorias de referencia.

3.5 Arduino

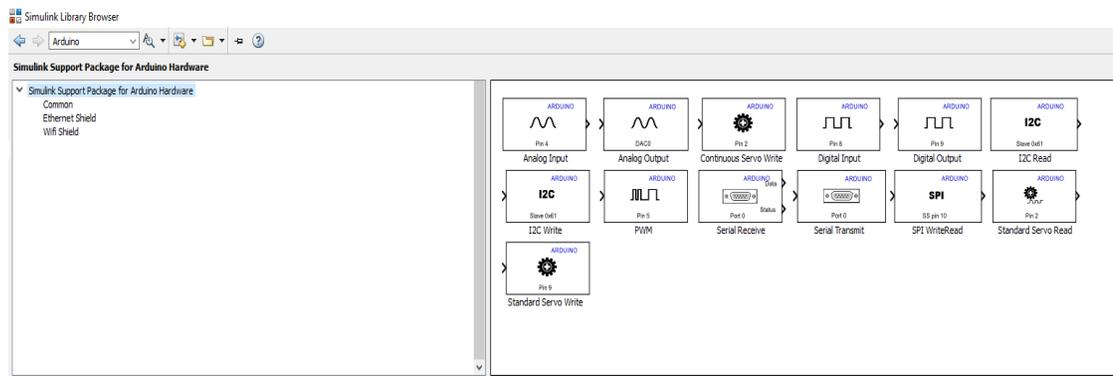
Arduino es una plataforma electrónica de código abierto que cuenta con hardware y software de uso sencillo y asequible. Una tarjeta Arduino cuenta con entradas y salidas digitales y análogas las cuales pueden ser programadas para realizar instrucciones por medio de un micro controlador. Su lenguaje de programación está basado en C++ y cuenta con una gran cantidad de librerías que permiten su integración con complementos externos a la tarjeta. (What is Arduino, 2018)

3.5.1 Arduino como tarjeta de adquisición

El software Matlab por medio de su herramienta Simulink, permite realizar una integración del hardware de Arduino mediante un paquete de soporte el cual habilita unas librerías específicas que agregan una serie de funciones que permiten desarrollar y simular algoritmos que corran de forma independiente en la tarjeta Arduino. (Arduino Support for Simulink, 1994-2019) A su vez, es posible ingresar señales tanto digitales como analógicas y realizar una lectura de esta información por medio de los bloques dentro de un diagrama realizado en Simulink.

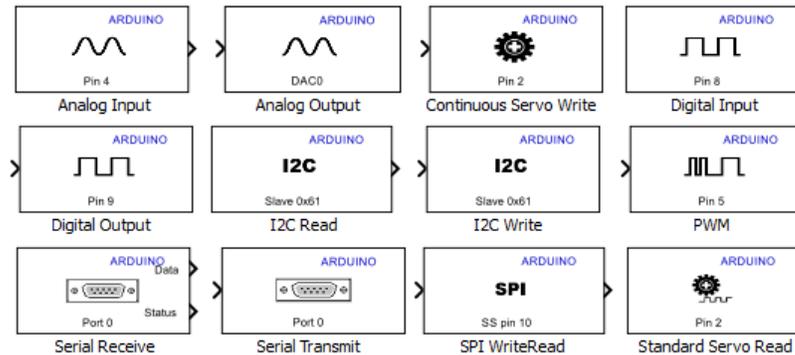
En el directorio de simulink se encuentran los bloques correspondientes al paquete de soporte para arduino. La herramienta cuenta con distintos bloques que representan cada una de las funciones disponibles en la tarjeta arduino, esto puede variar dependiendo de la tarjeta utilizada. En la Figura 7 se muestra el directorio de la librería de arduino io que tiene simulink.

Figura 7. Directorio de librería Arduino IO en Simulink



Esto permite realizar diagramas de bloques en donde se configurarán las conexiones hacia Arduino, definiendo los pines de entrada y salida que se van a utilizar. Mediante esta herramienta es posible optimizar el trabajo de captación de señales en tiempo real y al mismo tiempo se facilita la creación de señales por medio de los bloques de Simulink y las salidas analógicas y de PWM de Arduino. En la Figura 8 se encuentran los bloques de trabajo de la librería de Arduino.

Figura 8. Bloques de funciones de Arduino IO en Simulink



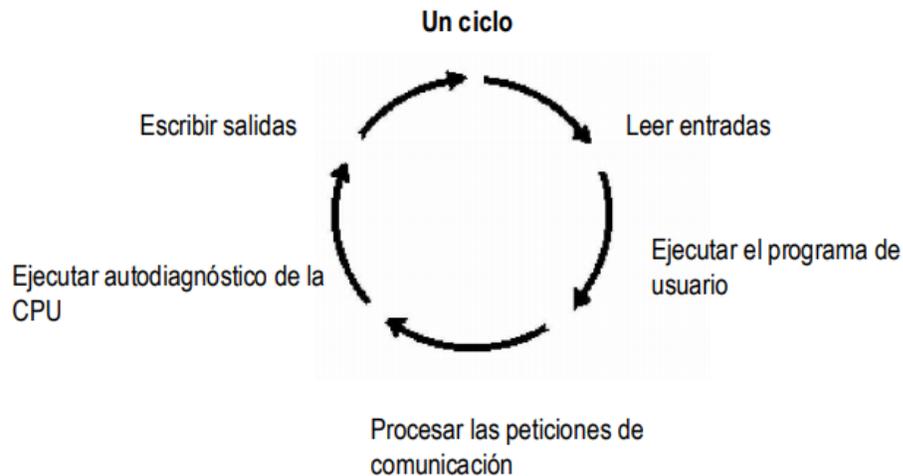
3.6 Controlador Lógico Programable

Un controlador lógico programable (PLC) es un dispositivo electrónico muy utilizado en el control de procesos industriales y automatización. El PLC cuenta con una memoria programable para el almacenamiento de instrucciones lógicas que permite recibir y enviar señales digitales o análogas y a su vez aplicar estrategias de control. (Controlador Lógico Programable, 2018)

La estructura general de un PLC consiste de una unidad de procesamiento o CPU, interfaz de entradas e interfaz de salidas. La CPU es la encargada de ejecutar las instrucciones programadas por el usuario en donde están involucradas entradas y salidas del PLC. Mediante la memoria, realiza el almacenamiento de la información que ingresa por las entradas para ser procesada y se toma una decisión dependiendo del algoritmo. Las entradas y salidas de un PLC pueden ser análogas y digitales, también pueden clasificarse por su tensión de alimentación y por el tipo de aislamiento. (R. M. M, 2018)

El ciclo de funcionamiento de un PLC consiste en ejecutar de forma secuencial las instrucciones que están definidas en el programa almacenado en su memoria. Éste debe leer las entradas que comúnmente provienen de una planta, luego ejecuta el algoritmo basado en los valores de entrada y posteriormente genera una acción en la salida. Esto puede ser una acción de control proveniente a una estrategia aplicada en la programación. En la Figura 9 se muestra en detalla el ciclo de trabajo de un PLC.

Figura 9. Ciclo de funcionamiento de un PLC



3.6.1 PLC F1616 TRIANGLE RESEARCH INTERNATIONAL

El modelo F1616 es el PLC que se va a utilizar en este proyecto para realizar la implementación de los controladores descritos previamente. El PLC cuenta con 8 entradas analógicas de 12 bits con 0-10 VDC, 4 salidas analógicas de 10 bits con 0-10 VDC, 16 entradas digitales y 16 salidas digitales. 12 de las salidas digitales son relés mientras que las 4 restantes son MOSFET de alta corriente. Éstas también pueden ser usadas como salidas de PWM. Adicionalmente, 6 de las entradas digitales son entradas de alta velocidad que pueden ser usadas para decodificación y medición de pulsos recibidos de los 3 encoders digitales con los que cuenta. [24]

En la Figura 10 se muestra el PLC F1616 BA de TRIANGLE RESEARCH INTERNATIONAL con todas sus entradas y salidas que fueron acopladas a pequeños puertos de color rojo por LATIN TECH para facilitar las conexiones.

Su programación se realiza en el software TRILOGI el cual cuenta con una interfaz con sus respectivas herramientas para crear circuitos de control, como se muestra en la figura. En la Figura 11 se muestra la interfaz de trabajo del software TRILOGI.

Figura 10. PLC F1616 BA

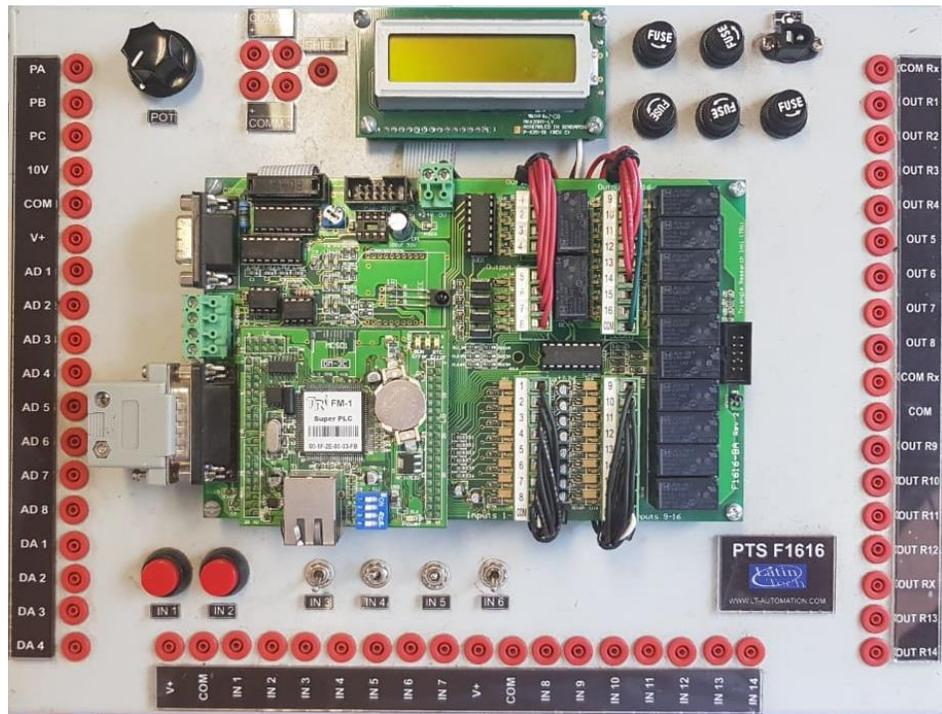


Figura 11. Interfaz de software TRILOGI



Dentro de los elementos que pueden agregarse en un circuito, está dCusF, el cual es el que permite realizar scripts en un lenguaje de programación propio del software que está basado en C++. La ventana de edición para la programación de PLC se muestra en la Figura 12.

Figura 12. Ventana de edición de scripts de TRILOGI

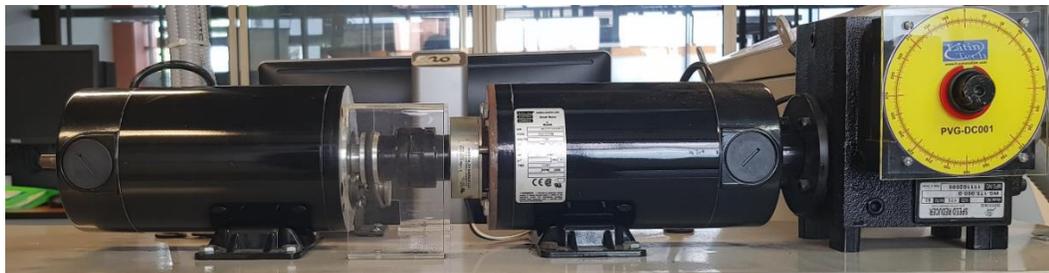
```
Custom Function #1 - cnt
p=1
t=3/10
x=1/10
i=2
a=ADC(1)
b=ADC(2)
e=a-b
m=p*e
q=n+(2*e/10)
u=m+q
if (u>4095)
u=4095
endif
if (u<-4095)
u=-4095
endif
L=(u+4095)/2
setlcd 2,7,"q="+str$(q)
setlcd 1,1,"e="+str$(e*10/4095)
setlcd 2,1,"MiK="+str$(n)
setlcd 1,7,"V="+str$(L*1000/4095)
setdac 1,L
n=q
```

3.7 Sistema motor-generador

3.7.1 Descripción del sistema motor-generador

El sistema motor-generador que se va a utilizar en este proyecto consiste de un motor de escobillas acoplado a un generador de corriente continua. El generador se encuentra a su vez acoplado a un reductor de velocidad con un encoder. En la Figura 13 se muestra detalladamente las conexiones del motor-generador.

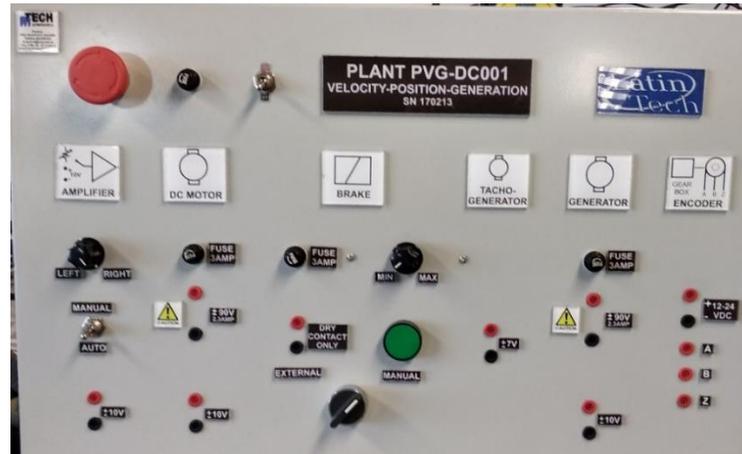
Figura 13. Motor-generador



Cuenta con un tablero en el cual están ubicadas todas las entradas y salidas correspondientes a cada uno de los elementos que se encuentran acoplados. El sistema motor-generador cuenta con una entrada de $\pm 10V$ y el taco-generador tiene

una salida de $\pm 7V$. También se observan entradas y salidas adicionales como la del encoder, frenos de forma manual y de emergencia y también la salida de carga eléctrica del generador. El tablero de control del sistema motor-generator esta detallado en la Figura 14.

Figura 14. Tablero de la planta



3.7.2 Motor de corriente continúa

Un motor eléctrico realiza el trabajo de convertir energía eléctrica en energía mecánica. El funcionamiento de un motor de corriente continua consiste en la repulsión que ejercen los polos magnéticos de un imán permanente cuando estos interactúan con los polos magnéticos de un electroimán el cual está sobre un eje. Al electroimán se le conoce como rotor y realiza una rotación entre los polos magnéticos del imán permanente que se encuentra dentro de la armadura del motor. Al energizarse la bobina del electroimán, se genera un campo magnético que interactúa con el campo magnético del imán permanente. Esta interacción produce un torque en el rotor y su dirección de giro depende de la ubicación de los polos del electroimán con respecto a los del imán permanente. (Funcionamiento del motor de corriente directa, 2018)

3.7.3 Generador de corriente continúa

El generador es una máquina que convierte la energía mecánica en energía eléctrica. La energía mecánica debe ser suministrada por un motor acoplado en su eje. El funcionamiento de un generador consiste en una bobina que gira en un campo magnético, de esta manera aparece una fuerza electromotriz que se establece sobre la bobina debido a las variaciones de flujo producidas por la rotación. La corriente generada se produce cuando el campo magnético proveniente

de un imán, atraviesa la bobina del rotor y produce un torque que origina el movimiento de rotación del generador. (M. A Rodríguez, 2017).

4. IDENTIFICACIÓN DEL MODELO DEL SISTEMA

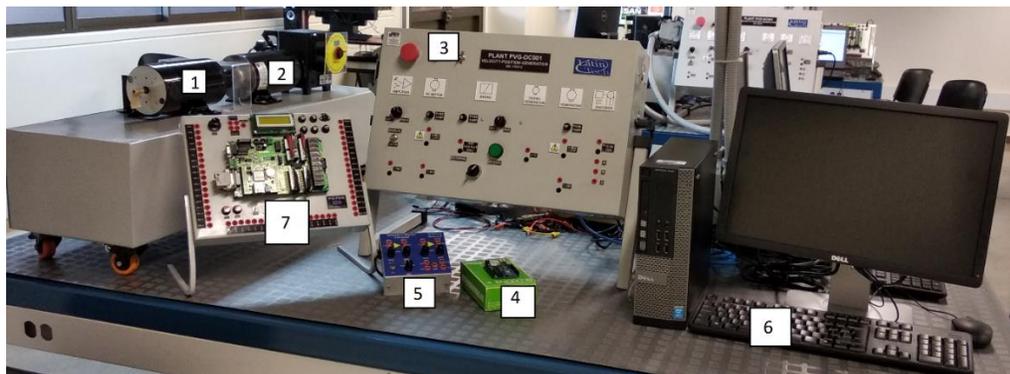
A continuación se presentará el proceso realizado para la adquisición de datos con la cual se hizo la identificación del modelo linealizado del sistema para posteriormente utilizarse en el diseño de las estrategias de control PID-1DOF y PID-2DOF.

4.1 Descripción del montaje

El sistema descrito a continuación se muestra en la Figura 15. Está compuesto de un motor eléctrico (1) que cuenta con una tensión nominal de $\pm 130V$ DC y 2500 rpm. Éste se encuentra acoplado a otra máquina eléctrica (2) que en este caso funciona como un generador, formando de esta manera el sistema motor-generador. El sistema cuenta con un tablero (3) en donde se manejan todas las entradas y salidas con las que cuenta, entre las referentes a este trabajo de grado, se tiene la entrada de $\pm 10V$ DC del motor (1) y la salida de $\pm 7V$ DC del tacogenerador.

Para realizar la adquisición de datos se utilizó la tarjeta Arduino UNO (4), la cual por medio de un diagrama en Simulink (6), se programó para que recibiera y enviara los datos de las señales de trayectorias para realizar la identificación y validación. Para llevar a cabo la comunicación entre el Arduino y el motor-generador fue necesario hacer uso de un convertidor unipolar-bipolar (5). Por último, la estrategia de control se llevó a cabo por medio del PLC FA1616 BA (7).

Figura 15. Montaje del sistema



4.2 Generación de señales y adquisición de datos

El proceso de identificación consiste en aplicar una señal de entrada compuesta de pulsos los cuales tendrán amplitudes aleatorias. El objetivo es observar cómo varía la velocidad del motor al ser sometido a los distintos niveles de voltaje y de esta manera identificar el comportamiento del sistema.

Para la identificación se emplearon dos trayectorias: La trayectoria de la Figura 16, la cual es utilizada para el proceso de identificación y obtención del modelo linealizado y la trayectoria de la Figura 17 se emplea para realizar una validación del modelo previamente obtenido.

Figura 16. Señal de identificación

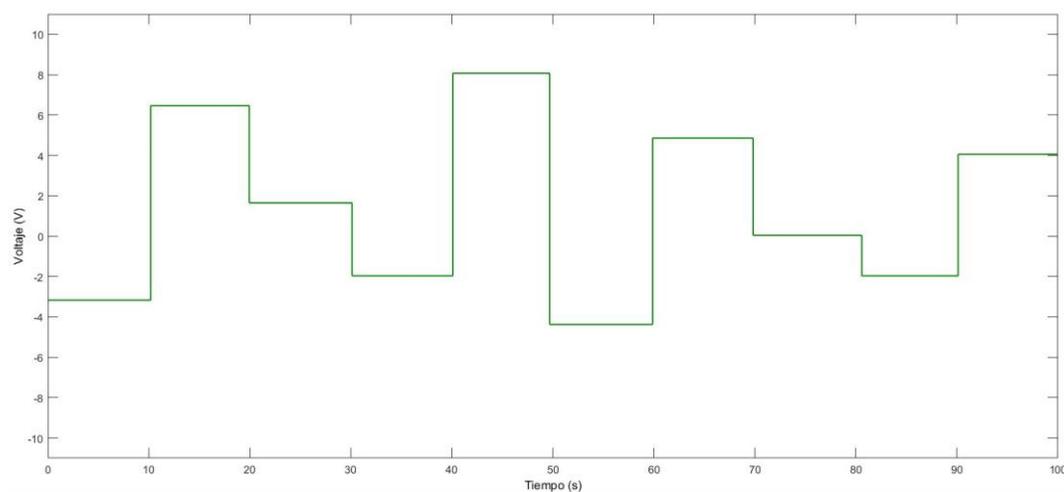
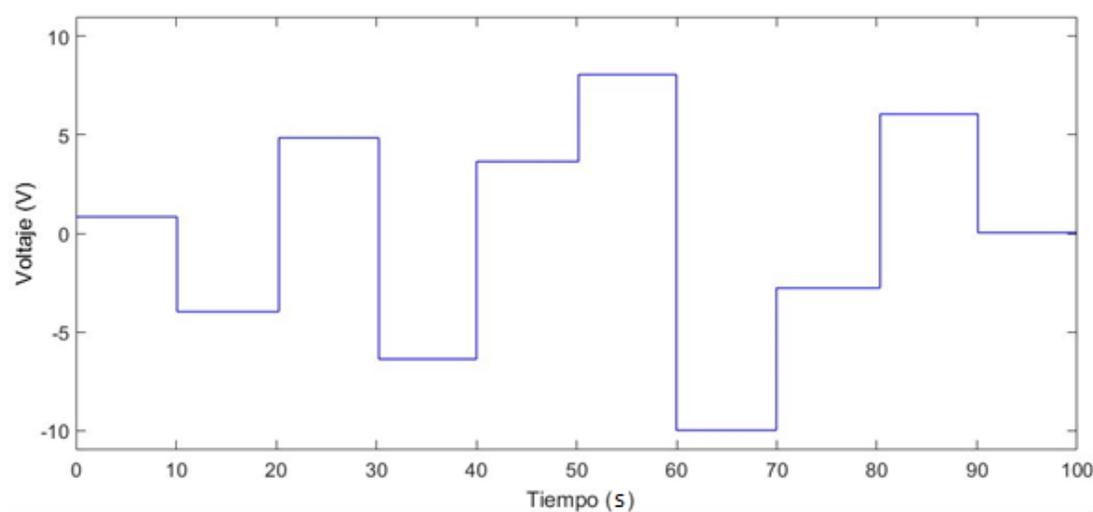
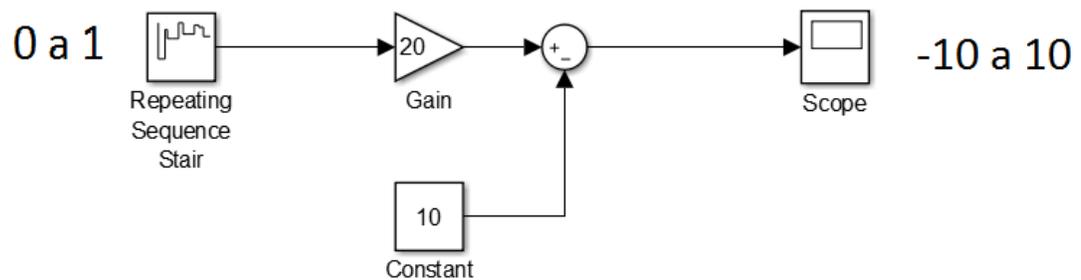


Figura 17. Señal de validación



Para generar las señales escalonadas se utilizó el diagrama de Simulink que se observa en la Figura 18. El primer bloque es el encargado de generar una señal con amplitudes aleatorias entre 0 y 1, luego se realizan las operaciones para convertir estas amplitudes al rango entre -10 y 10. Las velocidades máximas en ambos sentidos de giro están representadas por los valores de 10 y -10 y el valor de 0 es el voltaje en el cual el motor se encuentra detenido. Para el ancho de los pulsos de las trayectorias se tuvo en cuenta el tiempo de establecimiento esperado de 2 segundos, por tanto se escogió un ancho de pulso de 10 segundos, lo cual ofrece una ventana de tiempo suficiente para observar algún cambio de ser necesario.

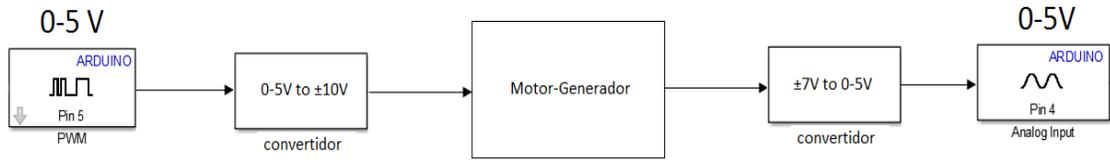
Figura 18. Generador de señal escalonada.



Para poder enviar estas señales al motor-generador fue necesario hacer un acondicionamiento de señal. La tarjeta Arduino UNO cuenta con un voltaje de hasta 5V en sus salidas de PWM, la cual pasa por un filtro RC para convertirla en una señal analógica y luego se utilizó un circuito amplificador para llevar las señales hasta el rango de 0 a 10V. Por último en este procedimiento, se utilizó un convertidor unipolar-bipolar para convertir la señal al rango de -10 a 10V para ser enviada a la entrada del motor-generador. El proceso anterior es necesario hacerlo de forma inversa para que la señal de salida del generador pueda ser enviada al Arduino UNO y de esta forma se adquiere la señal de respuesta proveniente del motor-generador.

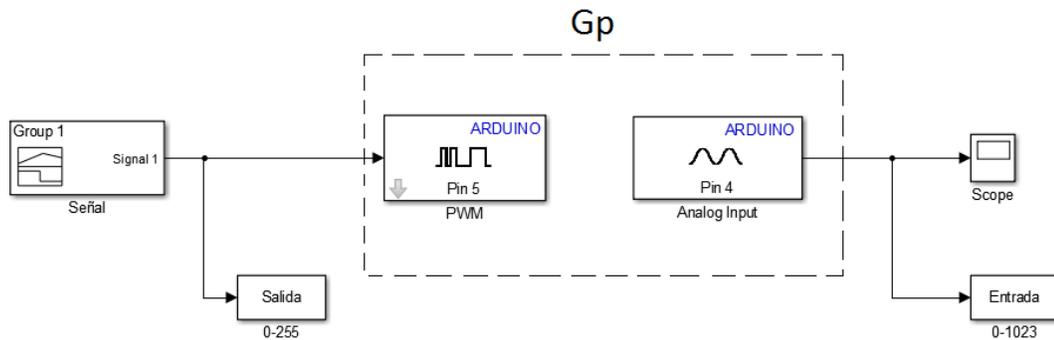
En la Figura 19 se observa un esquema de las conexiones descritas anteriormente. Para enviar la señal al motor se utilizó un pin de salida de PWM en el Arduino UNO, luego se acondiciona la señal por medio de un convertidor para ser conectada al motor. El sistema motor generador por medio del taco-generador produce un voltaje de $\pm 7V$ que será acondicionado de nuevo para que ingrese por el pin de entrada analógica del Arduino y así poder analizar las señales en Simulink.

Figura 19. Esquema de conexiones de identificación



En la Figura 20 se observa cada uno de los bloques utilizados para enviar la señal de $\pm 10V$ al motor y recibir la señal de salida de ± 7 del generador. El proceso consistió en recrear las señales obtenidas del diagrama de generación de la Figura 18 mediante la herramienta Signal Builder de Simulink, la cual permite crear una señal a partir de trazos y de esta manera obtener la forma escalonada de las trayectorias. Para realizar la adquisición de datos se definió un tiempo de muestreo de 0.1 segundos en Simulink. Como se observó en la Figura 16 y Figura 17, las señales tienen un tiempo de 100 segundos por lo que se almacenaron 10.000 muestras para cada prueba. Estos datos fueron guardados en variables del workspace para luego ser utilizados en la identificación mediante la herramienta Ident.

Figura 20. Diagrama de transferencia de señales.



En la Figura 21 y Figura 22 se presentan las señales obtenidas del generador y su respectiva comparación con las señales de entrada para cada caso.

Figura 21. Respuesta del motor a señal de identificación.

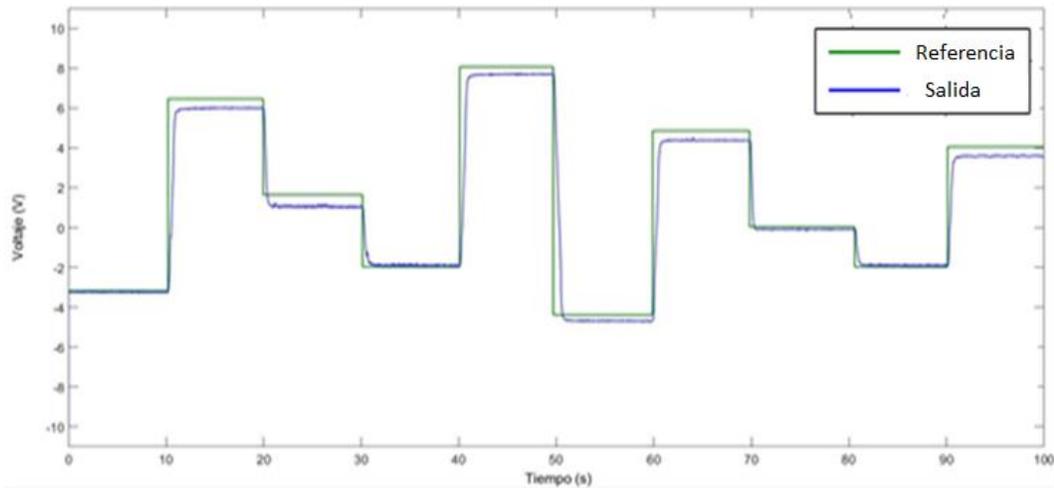
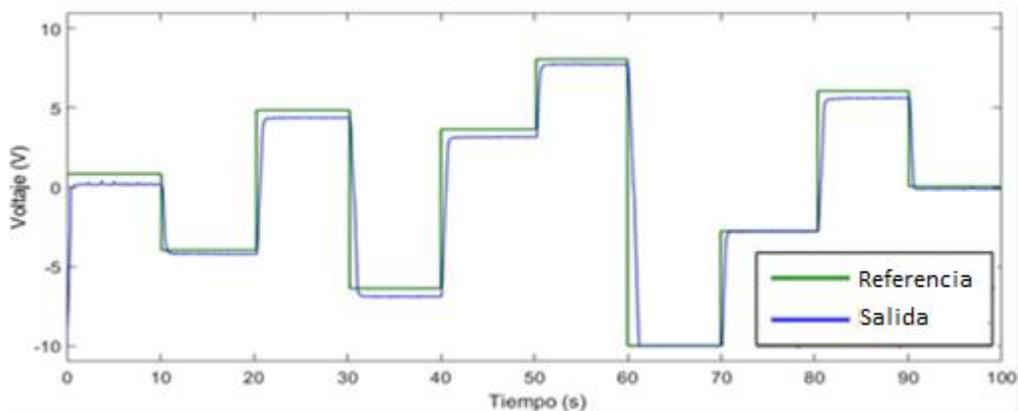


Figura 22. Respuesta del motor a señal de validación.



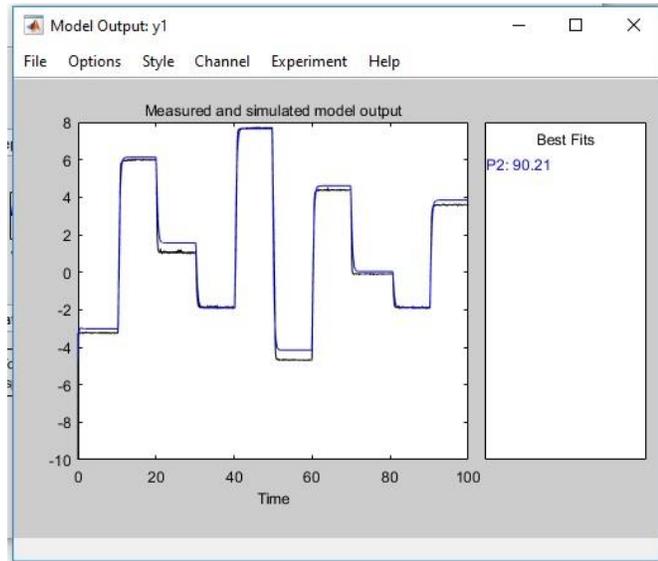
4.3 Identificación del modelo linealizado de la planta

Partiendo de los resultados obtenidos de la señal de identificación, se tomaron los datos almacenados en el workspace de Matlab y haciendo uso de la herramienta Ident, se obtuvo un modelo linealizado de segundo orden con dos polos reales que se muestra en (16). La identificación tuvo un ajuste del 90.21%.

$$Gp(s) = \frac{0.95}{(0.2118s+1)(0.2122s+1)} \quad (16)$$

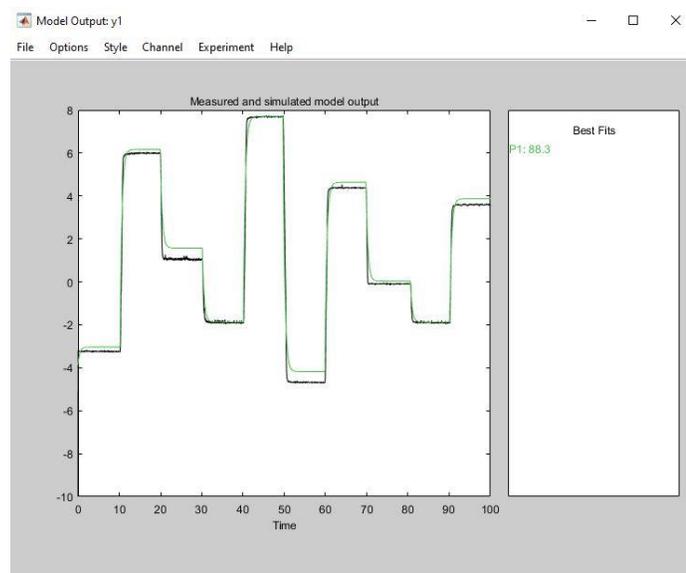
En la Figura 23 se muestra la gráfica del ajuste obtenido por medio de Ident de Matlab.

Figura 23. Gráfica de ajuste



Previamente, se realizó una prueba para identificar un modelo del sistema que constaba de una función de transferencia de primer orden sin ceros. Se obtuvo un ajuste de 88.3%, por tanto se optó por el modelo de segundo orden que proporcionaba un mayor ajuste. El modelo de primer orden se muestra en la Figura 24.

Figura 24. Ajuste modelo primer orden



5. DISEÑO DE LOS CONTROLADORES

A continuación se presentará el diseño de los controladores PID-1DOF y PID-2DOF utilizando el método de sintonización IMC que se presentó en la sección 3. Para el diseño del controlador se utiliza el modelo linealizado obtenido en la etapa de identificación dado por la ecuación (16).

5.1 Sintonización del controlador PID-1DOF por IMC

La función de transferencia de un sistema de segundo orden con polos reales está dada por

$$G_p(s) = \frac{k_p}{(\tau_1 s + 1)(\tau_2 s + 1)} \quad (17)$$

La ecuación del controlador IMC está dada por: (Y. García, 2009)

$$G_c(s) = \frac{f(s)G_p^{-1}}{1 - f(s)G_p^{-1}\tilde{G}_p} \quad (18)$$

Asumiendo un filtro de primer orden y reemplazando (17) en (18), el controlador IMC para el modelo de segundo orden esta dado por:

$$G_c(s) = \frac{(\tau_1 s + 1)(\tau_2 s + 1)}{k\lambda s} \quad (19)$$

La ecuación (19) se puede expresar de la siguiente manera:

$$G_c(s) = \frac{\tau_1 + \tau_2}{\lambda k_p} \left(1 + \frac{1}{(\tau_1 + \tau_2)s} + \frac{\tau_1 \tau_2}{\tau_1 + \tau_2} s \right) \quad (20)$$

Al hacer una semejanza entre la ecuación (20) y el controlador PID-1DOF.

$$PID(s) = k \left(1 + \frac{1}{\tau_i s} + \tau_d s \right) \quad (21)$$

Se obtiene las constantes del controlador dadas por (22):

$$\begin{aligned} k &= \frac{\tau_1 + \tau_2}{\lambda k_p} \\ \tau_i &= (\tau_1 + \tau_2) \\ \tau_d &= \frac{\tau_1 \tau_2}{(\tau_1 + \tau_2)} \end{aligned} \quad (22)$$

Como se observa en (22) la constante k depende del valor de la constante λ mientras que τ_i y τ_d dependen de los parámetros del modelo del sistema. Los parámetros de diseño deseados para el controlador son: tiempo de establecimiento en lazo cerrado igual a 2 segundos y un sobreimpulso menor al 10%.

El parámetro λ define la velocidad de respuesta del sistema en lazo cerrado y se ajusta experimentalmente hasta obtener las especificaciones de diseño deseadas. El valor de λ hallado es de 0.22 utilizando el modelo de simulación.

Teniendo en cuenta el valor de λ y el valor de los parámetros del modelo del sistema dados por la ecuación (16), las constantes del controlador PID (23) son:

$$\begin{aligned}k &= \frac{0.2118+0.2122}{(0.22)*0.95} = 2 \\ \tau_i &= 0.2118 + 0.2122 = 0.424 \\ \tau_d &= \frac{0.2118*0.2122}{0.2118+0.2122} = 0.105\end{aligned}\tag{23}$$

5.2 Diseño de controlador PID-2DOF

Como se mencionó en el capítulo 3, el controlador PID-2DOF está conformado por las mismas constantes del controlador PID-1DOF más un nuevo parámetro β , la cual en este trabajo de grado, se escogió realizando simulaciones con valores entre 0 y 1 hasta encontrar el que ofrezca mejor desempeño.

A continuación se presentan las pruebas realizadas para tres valores de β , en las cuáles se puede observar el efecto que tiene éste parámetro en la respuesta del sistema ante la entrada de un paso unitario. En la Figura 25 se muestra el diagrama de simulación empleado durante la realización de estas pruebas. Las constantes K , τ_i y τ_d son dadas por (23).

6. IMPLEMENTACIÓN PRÁCTICA Y EN SIMULACIÓN DE LOS CONTROLADORES PID-1DOF Y PID-2DOF

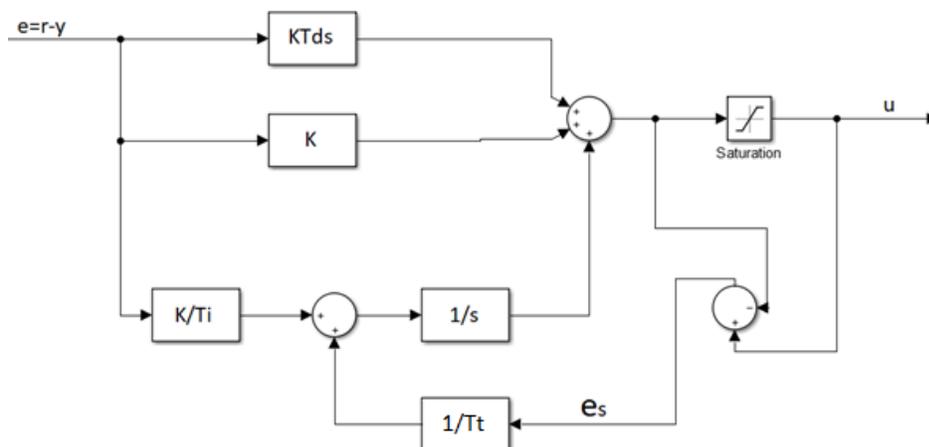
En este capítulo se presenta la simulación y la implementación de los controladores PID-1DOF y PID-2DOF. Para llevar a cabo la implementación, fue necesario realizar la discretización de los controladores empleando el método de discretización de diferencia atrás. De la misma manera, se va a presentar la implementación del método anti Windup de la acción integral para cada caso.

6.1 Método anti Wind-up Back Calculation

El fenómeno Windup se presenta en los controladores PID cuando el sistema tiene un rango de operación muy amplio por tanto cuando el actuador alcanza un nivel de saturación, el controlador no es capaz de aplicar la acción de control correspondiente, lo que lleva a que el sistema tenga un comportamiento como si se encontrara en un lazo abierto. Lo anterior se debe a la parte integral del controlador, ya que ésta integrará el valor del error hasta el punto en que éste valor sea muy grande y se produzca el efecto Windup.

El método anti Wind-up back calculation se implementa tal como se muestra en la Figura 27. Se observa que el sistema tiene un saturador ubicado en la salida del controlador. Su función es llevar la señal del controlador a un valor límite establecido en caso de que el valor de la salida del sistema alcance un valor mayor al permitido. Cuando el actuador se encuentra en un rango de valores menor a los límites, el valor de la señal del saturador será igual a la del controlador, por tanto la nueva señal e_s es cero por lo que la acción de control integral no se ve afectada. Ésta señal es realimentada por medio de la ganancia $1/Tt$, en donde se puede considerar $Tt = \sqrt{t_i t_d}$. (H. Markaroglu, 2006)

Figura 27. Diagrama de bloques Anti Wind-up Back Calculation



6.2 Discretización de los controladores

El objetivo del proceso de discretización es obtener los controladores digitales representados como ecuaciones de diferencia para implementarlos directamente en el lenguaje de programación tanto de StateFlow como en PLC F1616 BA.

6.2.1 Discretización del controlador PID-1DOF

Para aplicar el método de diferencia atrás debe considerarse la ecuación general de un controlador PID-1DOF (K. Ogata, 2010)

$$PID(s) = Kp\left(1 + \frac{1}{\tau_i s} + \tau_d s\right) \quad (24)$$

En la práctica, al controlador PID se le aplica un filtro en la acción derivativa para mitigar los efectos de ruido que pueden afectar la respuesta del sistema, por tanto la ecuación queda como se muestra en (25)

$$PID(s) = Kp\left(1 + \frac{1}{\tau_i s} + \frac{\tau_d s}{N s + 1}\right) \quad (25)$$

donde N corresponde al valor del filtro de acción derivativa y en este proyecto de grado N=10.

El método de discretización de diferencia atrás permite relacionar la transformada de Laplace y la transformada z mediante la siguiente expresión:

$$s = \frac{1-z^{-1}}{T_s} \quad (26)$$

Al reemplazar (26) en (25) se obtiene el PID-1DOF digital dado por (27).

$$PID(z) = Kp + \frac{KpT_s}{\tau_i} \left(\frac{1}{1-z^{-1}}\right) + \frac{\frac{kp*\tau_d}{T_s}(1-z^{-1})}{\left(\frac{\tau_d}{NT_s}+1\right) - \frac{\tau_d}{NT_s}z^{-1}} \quad (27)$$

Teniendo en cuenta que $PID(z) = U(z)/E(z)$ donde $U(z)$ es la acción de control y $E(z)$ es la señal de error, la ecuación (27) se puede expresar como

$$U(z) = KpE(z) + \frac{KpT_s}{\tau_i} \left(\frac{1}{1-z^{-1}}\right)E(z) + \frac{\frac{kp*\tau_d}{T_s}(1-z^{-1})}{\left(\frac{\tau_d}{NT_s}+1\right) - \frac{\tau_d}{NT_s}z^{-1}} E(z) \quad (28)$$

La ecuación (28) puede expresarse de forma separada en cada una de las componentes del controlador PID como se muestra en (29)

$$U(z) = U_p(z) + U_i(z) + U_d(z) \quad (29)$$

En forma independiente cada componente se expresa como

$$\begin{aligned} U_p(z) &= K_p E(z) \\ U_i(z) &= \frac{K_p T_s}{\tau_i} \left(\frac{1}{1-z^{-1}} \right) E(z) \\ U_d(z) &= \frac{\frac{k_p \tau_d}{T_s} (1-z^{-1})}{\left(\frac{\tau_d}{N T_s} + 1 \right) - \frac{\tau_d}{N T_s} z^{-1}} E(z) \end{aligned} \quad (30)$$

A partir de estas ecuaciones se pueden hallar las ecuaciones de diferencia que corresponden al controlador discreto y que van a ser implementadas tanto en StateFlow como en el PLC F1616 BA.

Las ecuaciones de diferencia para cada componente del controlador PID-1DOF discreto son las siguientes

$$\begin{aligned} U_p(k) &= K_p e(k) \\ U_i(k) &= U_i(k-1) + \frac{K_p T_s}{\tau_i} e(k) \\ U_d(k) &= \frac{\frac{k_p \tau_d}{T_s} (e(k) - e(k-1)) + \frac{\tau_d}{N T_s} U_d(k-1)}{\left(\frac{\tau_d}{N T_s} + 1 \right)} \end{aligned} \quad (31)$$

6.2.2 Discretización del controlador PID-2DOF

El procedimiento para realizar la discretización del PID-2DOF es similar a la realizada para el PID-1DOF presentado en la sección anterior. A partir de la ecuación del PID-2DOF e incluyendo el filtro de la acción derivativa se tiene la ecuación dada por (32).

$$U(s) = K_p (\beta r(s) - y(s)) + \frac{K_p}{\tau_i s} (r(s) - y(s)) + \left(\frac{\tau_d s}{\tau_d s + 1} \right) (-y(s)) \quad (32)$$

Aplicando la relación dada por (26) y teniendo en cuenta que $PID(z) = U(z)/E(z)$, la ecuación queda de la siguiente forma:

$$U(z) = K_p (\beta r(z) - y(z)) + \frac{K_p T_s}{\tau_i} \left(\frac{1}{1-z^{-1}} \right) (r(z) - y(z)) + \frac{\frac{k_p \tau_d}{T_s} (1-z^{-1})}{\left(\frac{\tau_d}{N T_s} + 1 \right) - \frac{\tau_d}{N T_s} z^{-1}} (-y(z)) \quad (33)$$

donde $r(z) - y(z)$ corresponde a la señal de error $E(z)$.

La ecuación (33) se puede expresar de forma separada en cada una de las componentes del PID-2DOF de la siguiente manera

$$\begin{aligned}
 U_p(z) &= K_p(\beta r(z) - y(z)) \\
 U_i(z) &= \frac{K_p T_s}{\tau_i} \left(\frac{1}{1-z^{-1}} \right) (r(z) - y(z)) \\
 U_d(z) &= \frac{\frac{k_p \tau_d}{T_s} (1-z^{-1})}{\left(\frac{\tau_d}{N T_s} + 1 \right) - \frac{\tau_d}{N T_s} z^{-1}} (-y(z))
 \end{aligned} \tag{34}$$

Las ecuaciones de diferencia para el controlador PID-2DOF digital son las siguientes

$$\begin{aligned}
 U_p(k) &= K_p(\beta r(k) - y(k)) \\
 U_i(k) &= U_i(k-1) + \frac{K_p T_s}{\tau_i} e(k) \\
 U_d(k) &= - \frac{\frac{k_p \tau_d}{T_s} (y(k) - y(k-1)) + \frac{\tau_d}{N T_s} U_d(k-1)}{\left(\frac{\tau_d}{N T_s} + 1 \right)}
 \end{aligned} \tag{35}$$

6.3 Implementación de los controladores PID-1DOF y PID-2DOF

6.3.1 Simulación de Controladores utilizando la herramienta StateFlow de Matlab.

La implementación de los controladores se realizó inicialmente empleando la herramienta StateFlow de Matlab. La Implementación implica la solución de las ecuaciones de diferencia dadas por (31) y (35).

En la Figura 28 y Figura 29 se muestra en detalle la programación de los controladores PID-1DOF y PID-2DOF respectivamente, programados por la herramienta StateFlow de Matlab. Ambas líneas de programación cuentan con dos estados, inicio y control, lo cuales conforman la estructura del controlador programado con la herramienta StateFlow de Matlab.

Figura 28. Controlador PID-1DOF en StateFlow.

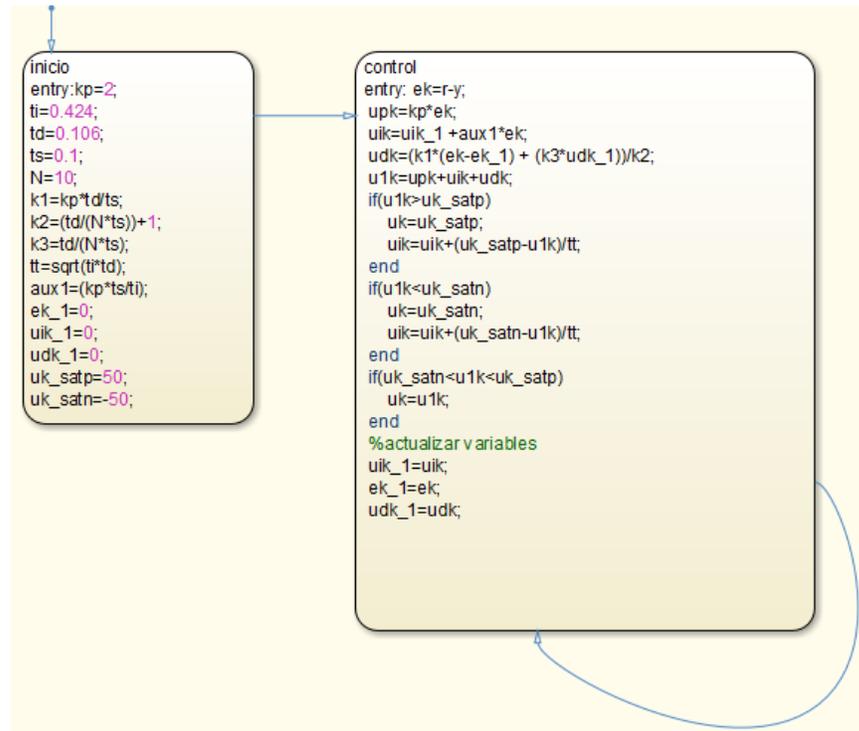
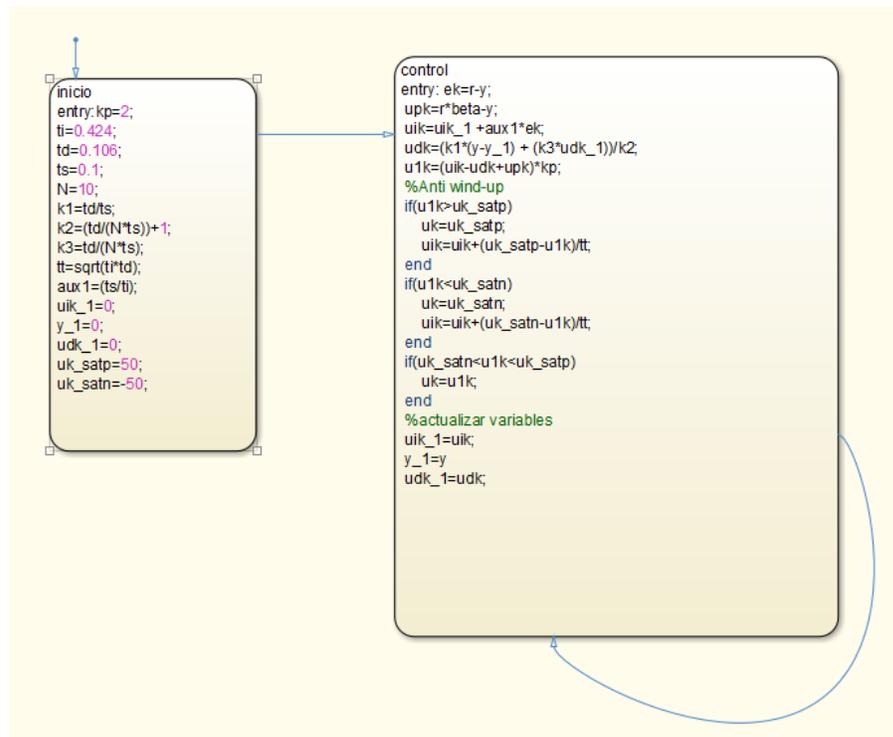
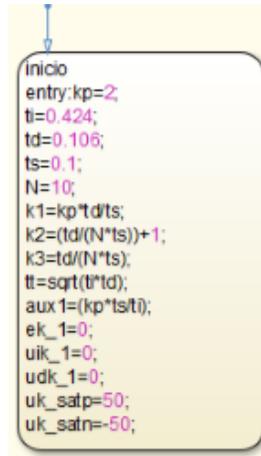


Figura 29. Controlador PID-2DOF en StateFlow.



El estado de Inicio presentado en la Figura 30, se caracteriza por ser una etapa donde se inicializan las variables que componen al controlador y además asignar valores a constantes que se pueden utilizar durante el estado de control. Para este proyecto de grado los estados de inicio de ambos controladores son exactamente el mismo ya que fueron diseñados con el mismo método de sintonización.

Figura 30. Estado de Inicio en StateFlow para ambos controladores.



Los estados de control presentados en la Figura 31 y Figura 32 corresponden a los controladores PID-1DOF y PID-2DOF respectivamente. El estado de control se caracteriza por ser el encargado de calcular inicialmente la señal de error, luego la acción de control total del sistema a partir de la sumatoria de acciones individuales conformadas por la acción proporcional, integral y derivativa, aplicar el método Anti Wind-up y actualizar las variables del controlador. Esta etapa también incluye la adquisición de datos de referencia y de realimentación del sistema con los cuales se hace el cálculo de error.

Figura 31. Estado de control en StateFlow para controlador PID-1DOF.

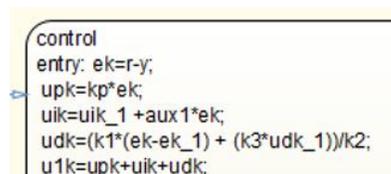
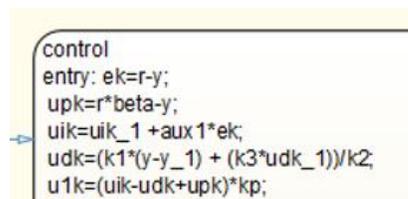


Figura 32. Estado de control en StateFlow para controlador PID-2DOF.



La etapa Anti Wind-Up se caracteriza por evaluar si la acción de control total supera los límites de trabajo del actuador. De darse esta condición, se procede a una serie de cálculos para aumentar o disminuir los valores en la acción de control integral y evitar una saturación del sistema. Para ambos controladores el estado Anti Wind-Up es el mismo y es mostrado en la Figura 33.

Figura 33. Etapa de Anti Wind-Up en StateFlow para ambos controladores.

```
%Anti wind-up
if(u1k>uk_satp)
    uk=uk_satp;
    uik=uik+(uk_satp-u1k)/tt;
end
if(u1k<uk_satn)
    uk=uk_satn;
    uik=uik+(uk_satn-u1k)/tt;
end
if(uk_satn<u1k<uk_satp)
    uk=u1k;
end
```

La etapa de Actualización presentada en la Figura 34 y Figura 35 corresponde a los controladores PID-1DOF y PID-2DOF respectivamente. Esta etapa actualiza las variables que dependen de su valor anterior. Esta etapa se diferencia en los controladores debido a que en el PID-1DOF se actualiza la variable de error y en el PID-2DOF se actualiza el valor de realimentación.

Figura 34. Estado de actualización en StateFlow para controlador PID-1DOF

```
%actualizar variables
uik_1=uik;
ek_1=ek;
udk_1=udk;
```

Figura 35. Estado de actualización en StateFlow para controlador PID-2DOF

```
%actualizar variables
uik_1=uik;
y_1=y;
udk_1=udk;
```

Finalmente, después de terminada la etapa de actualización se forma un bucle el cual retorna de nuevo a el estado de control y volviendo a calcular variables que sean óptimas para el control total del sistema.

Con el fin de obtener resultados por medio de simulación de los controladores, se debe tener en cuenta también la discretización de la planta o del modelo linealizado que fue calculada en el capítulo de identificación. Para tener una semejanza entre las dos plantas se utilizó el comando de Matlab c2d (Continuos to Discrete), la cual hace una conversión de un modelo continuo a un modelo discreto. Este comando requiere dos variables, las cuales son el tiempo de muestreo y el método de discretización por el cual se va hacer la conversión del modelo. En la Figura 36 se puede observar la línea de código de Matlab en detalle para hacer la conversión del modelo.

Inicialmente se ingresa el modelo continuo mostrada en la ecuación (36) y por medio de la línea de código que indica la flecha en la Figura 36 se hace la conversión de modelos utilizando la función continua, un tiempo de muestreo de 0.1s y el método de discretización retenedor de orden cero ("zoh", Zero-order hold), dando como resultado una función en el dominio el tiempo discreto mostrado en la ecuación (37).

$$G_p(s) = \frac{0.95}{(0.2122s+1)(0.2118s+1)} \quad (36)$$

Figura 36. Línea de Código para conversión de modelo continuo a discreto.

```
>> s=tf('s');
>> sysc=0.95/((0.2122*s+1)*(0.2118*s+1))

sysc =

          0.95
-----
0.04494 s^2 + 0.424 s + 1

Continuous-time transfer function.

>> sysD=c2d(sysc,0.1,'zoh') ←
sysD =

    0.07766 z + 0.05669
-----
    z^2 - 1.248 z + 0.3893

Sample time: 0.1 seconds
Discrete-time transfer function.
```

$$G_p(z) = \frac{0.07766z+0.05669}{z^2-1.248z+0.3893} \quad (35)$$

En la Figura 37 y Figura 38 se muestran los modelos de simulación para los controladores discretos PID-1DOF y PID-2DOF respectivamente, utilizando la herramienta StateFlow con los cuales se hicieron pruebas al modelo hallado convertido en el dominio discreto.

Figura 37. Estructura de un controlador discreto PID-1DOF en StateFlow.

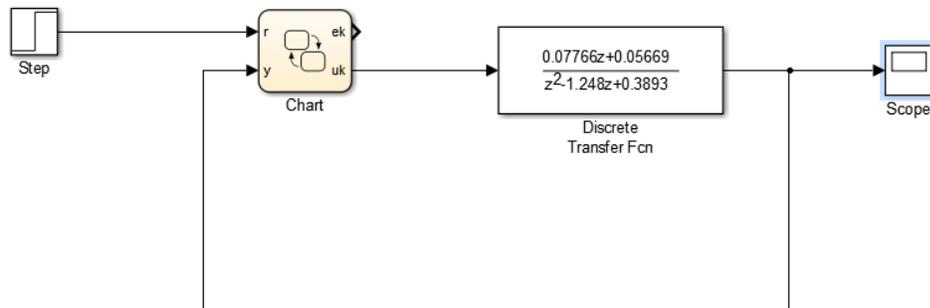
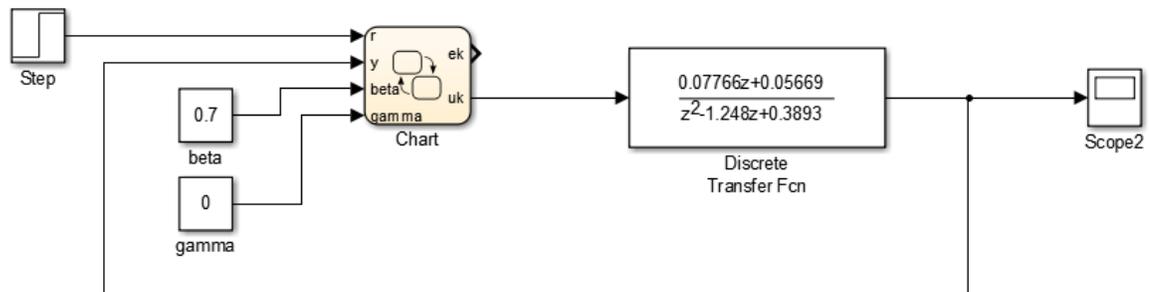


Figura 38. Estructura de un controlador discreto PID-2DOF en StateFlow.



Los controladores fueron sometidos inicialmente a una señal de paso unitario y luego a una trayectoria de pasos escalonada con cambios en tiempo y en amplitud con el fin de comprobar si los controladores cumplen los requisitos de diseño. En las Figura 39 y Figura 40 se muestran las respuestas de los controladores discretos a un paso unitario. Se puede apreciar que ambos controladores cumplen con los requisitos de diseño al tener un tiempo de establecimiento igual a dos segundos y además de no tener un sobreimpulso mayor al 10%.

Figura 39. Respuesta del controlador discreto PID-1DOF en StateFlow a un paso unitario.

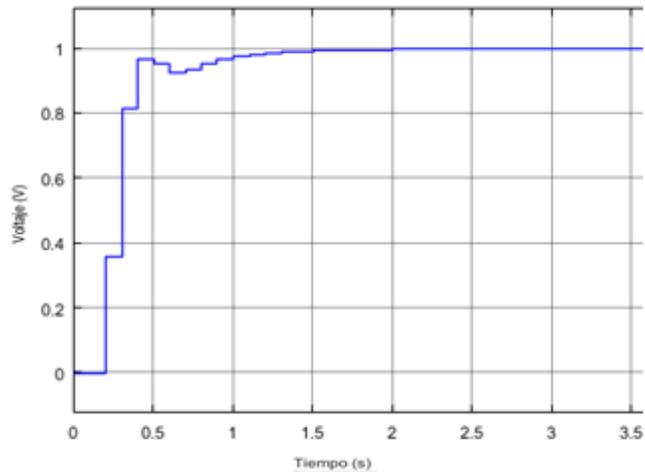
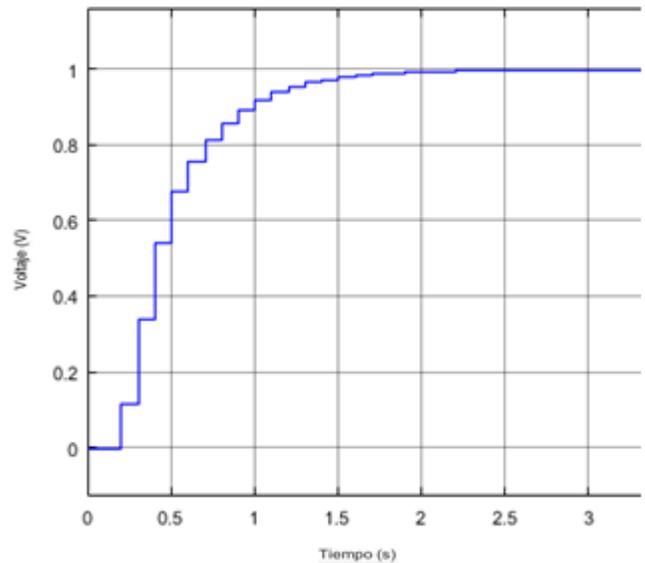
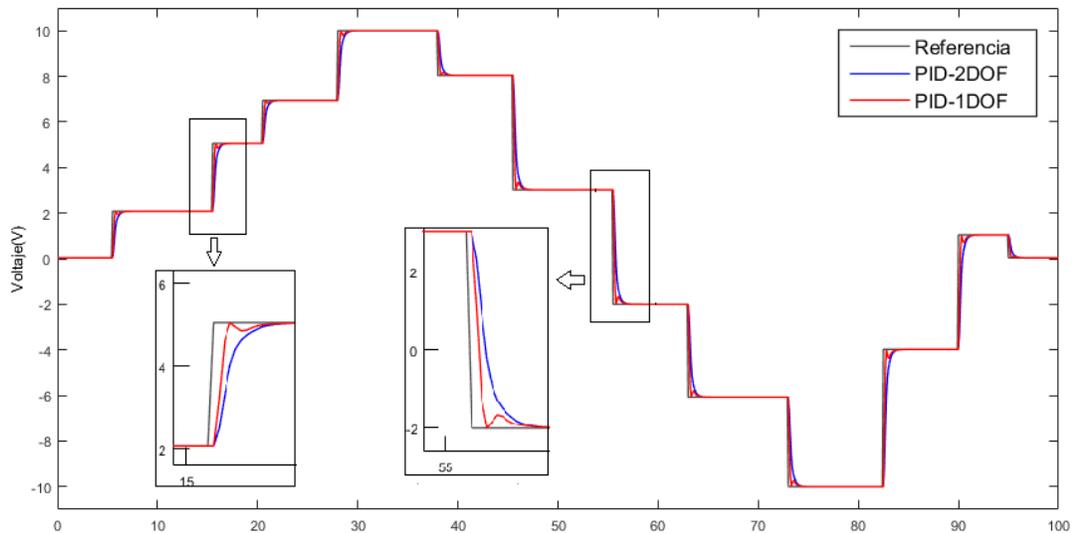


Figura 40. Respuesta del controlador discreto PID-2DOF en StateFlow a un paso unitario.



La respuesta de los controladores para la trayectoria escalonada se muestra en la Figura 41. Se puede observar que ambos controladores cumplen con los requisitos del diseño y además dan un buen seguimiento para cada valor en amplitud a los que fueron sometidos. Se puede ver en el acercamiento de la Figura 41 una diferencia entre los controladores donde el controlador PID-1DOF alcanza el estado estacionario de una manera no tan suave como lo hace el controlador PID-2DOF, esto debido a la presencia del factor β .

Figura 41. Respuesta de los controladores discretos en StateFlow para trayectoria de prueba.



6.3.2 Implementación de controladores PID-1DOF y PID-2DOF en el PLC F1616 BA de TRIANGLE RESEARCH INTERNATIONAL.

La implementación práctica de los controladores en este proyecto de grado fue realizada con el PLC F1616 BA de TRIANGLE RESEARCH INTERNATIONAL mostrado en la Figura 10. Para la implementación se utilizó la interfaz de programación proporcionada por el software Trilogi, la cual es controlada por medio de un lenguaje basado en C++ y enviado al PLC por medio de un cable tipo serial, todo esto con ayuda de un computador con una versión de Windows XP. En la Figura 42 y Figura 43 se presenta en detalle la programación del software Trilogi para los controladores PID-1DOF y PID-2DOF respectivamente, los cuales serán utilizados para el control del sistema motor-generator. Como se observa la programación es de tipo secuencial e implementa las ecuaciones de los controladores digitales dadas por (31) y (35).

De manera similar que en la herramienta StateFlow de Matlab, la programación de los controladores en el Software de Trilogi cuenta con las mismas etapas en su código para la realización de la acción de control total del sistema.

En la Figura 44 se muestra la etapa de inicio de un controlador PID-1DOF o el de un controlador PID-2DOF programado en el software Trilogi, se puede resaltar como diferencia que el software Trilogi no cuenta con punto flotante por lo que cada constante, si tiene números decimales, debe ser escrito de manera fraccionaria tal como se muestra en el código.

Figura 42. Estructura de controlador PID-1DOF en Software Trilogi

```
Custom Function #1 - control
' constantes de la planta
t=1/100      'ts
p=2         'kp
i=42/100    'ti
l=105/1000  'td
DM[3]=10    'N

c=4095*3    'Saturacion positiva con ganancia
j=-4095*3   'Saturacion negativa con ganancia

'control
a=ADC(4)    'referencia
b=ADC(5)    'retroalimentacion

e=a-b       'senal de error
m=p*e       'parte proporcional = kp*e
q=(e/21)+n  'parte integral = (kp/ti)*ts*e + n
d=10*(b-g)+ 51*h/100 'parte derivativa = (td*kp)*(e - e(k-1))/ts
r=m+q+d     'control pid

if(r>c)     'efecto antiwind up
u=c
q=q+(c-r)*21/100
endif

if(r<j)
u=j
q=q+(j-r)*21/100
endif

if(j<r<c)
u=r
endif      'fin del antiwind up

o=(u+4095)/2 'acoplamiento de la senal
y=(e+4095)/2
v=o*100/4095
w=e*100/4095
setlcd 1,1, "a="+str$(a)
setlcd 1,8, "b="+str$(b)
setlcd 2,1, "PID-1DOF"
setdac 1,o
setdac 2,y
n=q
g=e
h=d
```

Figura 43. Estructura de Controlador PID-2DOF en Software Trilogi.

```

Custom Function #1 - control
' constantes de la planta
t=1/100      'ts
p=2         'kp
i=42/100    'ti
l=105/1000  'td
DM[3]=10    'N

c=4095*3    'Saturacion positiva con ganancia
j=-4095*3   'Saturacion negativa con ganancia

'control
a=ADC(4)    'referencia
b=ADC(5)    'retroalimentacion
DM[1]=3/10  'beta
DM[2]=0     'gamma
e=a-b
k=2*((7*a/10) -b ) 'parte proporcional
q=(e/21)+ n     'parte integral = (ts/ti)*e + n
d=10*(b-g)+ 51*h/100 'parte derivativa = (td/ts)*(h - h(k-1))
r=k+q-d 'control pid

if(r>c)      'efecto antiwind up
u=c
q=q+(c-r)*21/100
endif
if(r<j)
u=j
q=q+(j-r)*21/100
endif
if(j<r<c)
u=r
endif      'fin del antiwind up

o=(u+4095)/2 'acoplamiento de la senal
y=(e+4095)/2
v=o*100/4095
w=e*100/4095
setlcd 1,1, "a="+str$(a)
setlcd 1,8, "b="+str$(b)
setlcd 2,1, "PID-2DOF"
setdac 1,o
setdac 2,y
h=d
n=q
g=b
    
```

Figura 44. Etapa de inicio en Trilogi para un controlador PID-1DOF y PID-2DOF.

```

' constantes de la planta
t=1/100      'ts
p=2         'kp
i=42/100    'ti
l=105/1000  'td
DM[3]=10    'N

c=4095*3    'Saturacion positiva con ganancia
j=-4095*3   'Saturacion negativa con ganancia
    
```

La etapa de control de un controlador PID-2DOF programado en Trilogi es mostrado en la Figura 45, tiene la misma función que el estado de control programado en StateFlow pero con un código diferente, ya que los cálculos están realizados en cada acción individual de control porque el software no es capaz de realizar cálculos con punto flotantes. La etapa de control también tiene la función de hacer la lectura de datos que provienen del sistema motor-generator utilizando los comandos ADC en Trilogi, como la referencia y la retroalimentación, cerrando así el lazo de control.

Figura 45. Etapa de control en Trilogi para el controlador PID-2DOF.

```
'control
a=ADC(4)      'referencia
b=ADC(5)      'retroalimentacion
DM[1]=3/10    'beta
DM[2]=0       'gamma
e=a-b         'senal de error
k=2*((7*a/10) -b ) 'parte proporcional
q=(e/21)+ n   'parte integral = (ts/ti)*e + n
d=10*(b-g)+ 51*h/100 'parte derivativa = (td/ts)*(h - h(k-1))
r=k+q-d       'control pid
```

La etapa de Anti Wind-Up programado en Trilogi que se muestra en la Figura 46, es la técnica por medio de cálculos para que el control total no se sature por la sumatoria de valores en la acción de control integral. Para este proyecto de grado este código es válido para ambos controladores.

Figura 46. Etapa de Anti Wind-Up en Trilogi para un controlador PID-1DOF y PID-2DOF.

```
if(r>c)        'efecto antiwind up
u=c
q=q+(c-r)*21/100
endif
if(r<j)
u=j
q=q+(j-r)*21/100
endif
if(j<r<c)
u=r
endif         'fin del antiwind up
```

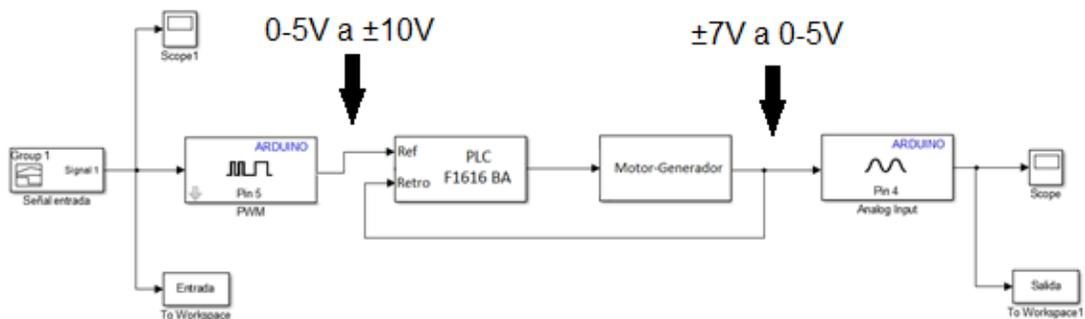
En la etapa de actualización para un controlador PID-2DOF en Trilogi mostrado en la Figura 47 tiene la misma finalidad que el programado en StateFlow, pero en este caso también tiene la función de enviar los datos calculados a las salidas del PLC, a partir del comando setdac y asignando los pines, las cuales serán conectadas al sistema motor-generador creando así el control PID, todo esto con cálculos previos por el acoplamiento de las señales. Al terminar su ejecución el programa retornara al estado de inicio repitiendo todo el proceso de control.

Figura 47. Etapa de actualización y envío de datos en Trilogi para un controlador PID-2DOF.

```
o=(u+4095)/2      'acoplamiento de la señal
y=(e+4095)/2
v=o*100/4095
w=e*100/4095
setlcd 1,1, "a="+str$(a)
setlcd 1,8, "b="+str$(b)
setlcd 2,1, "PID-2DOF"
setdac 1,o
setdac 2,y
h=d
n=q
g=b
```

En la Figura 48 se muestra el esquema general para la Implementación de los controladores en el sistema motor-generator utilizando el PLC F1616 BA y utilizando Arduino como tarjeta de adquisición. Se puede observar el lazo cerrado de control entre el sistema motor-generator y el PLC, donde están implementados los controladores. Los demás bloques corresponden al Arduino como tarjeta de adquisición y envío de datos de la señal de referencia que leerá el PLC F1616 BA. Para recibir y enviar datos con el Arduino hubo la necesidad de acoplar convertidores de voltajes de 0-5V a $\pm 10V$ y $\pm 7V$ a 0-5V.

Figura 48. Implementación de controladores en el sistema motor-generator utilizando el PLC F1616 BA.

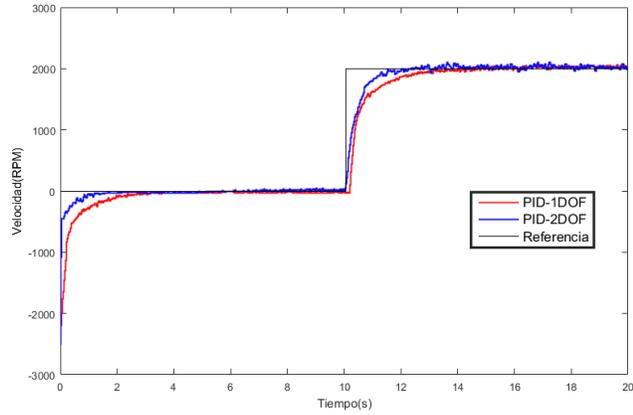


6.4 Validación de los controladores PID-1DOF y PID-2DOF

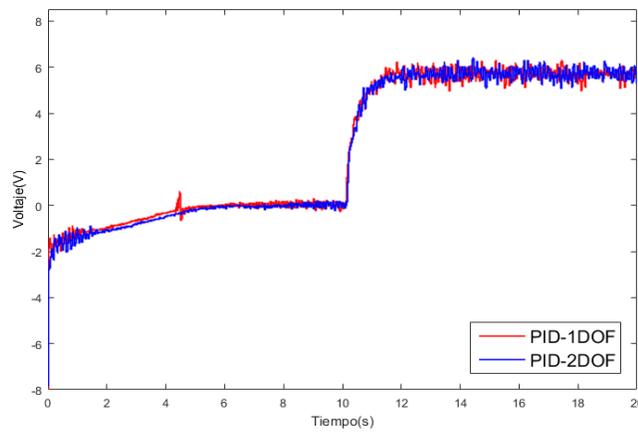
Para el proceso de validación en la práctica de los controladores PID-1DOF y PID-2DOF se planteó llevar a cabo una prueba en la que se emplearon 3 trayectorias, las cuáles fueron aplicadas a los dos controladores y se registró el seguimiento para cada caso.

Para el primer caso se planteó una trayectoria de paso unitario. En la Figura 49a se tiene la respuesta temporal del sistema motor-generador ante esta trayectoria. Como se puede observar los controladores PID-1DOF y PID-2DOF realizan un seguimiento adecuado de esta trayectoria, sin embargo, se observa que el controlador PID-2DOF tiene una mejora en el tiempo de establecimiento debido a que es más cercano al valor deseado de 2 segundos. Al analizar la acción de control se observa que los dos controladores presentan un comportamiento similar a lo largo de la trayectoria. En la señal de error se observa que el PID-2DOF presenta una respuesta más rápida con respecto al controlador PID-1DOF.

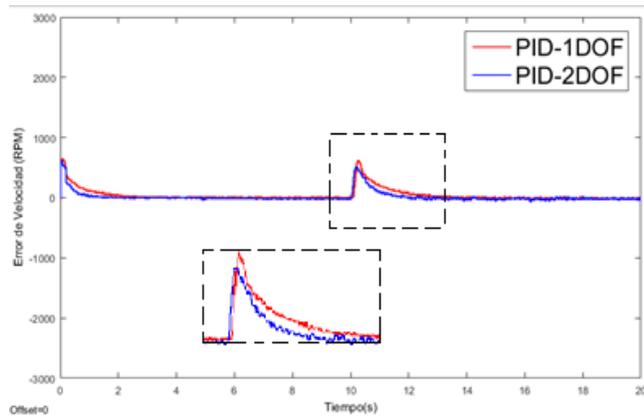
Figura 49. Respuesta temporal (a), acción de control (b), y señal de error (c) de los controladores PID-1DOF y PID-2DOF para la trayectoria pasó unitario en operación nominal.



(a)



(b)



(c)

Para realizar la validación de los controladores PID-1DOF y PID-2DOF se implementaron dos trayectorias escalonadas. Las trayectorias que se observan en la Figura 50 y

Figura 51, contienen cambios aleatorios en las amplitudes de los escalones para de esta manera observar el comportamiento de los dos controladores ante distintos valores de tensión en la entrada.

Figura 50. Trayectoria de validación 1

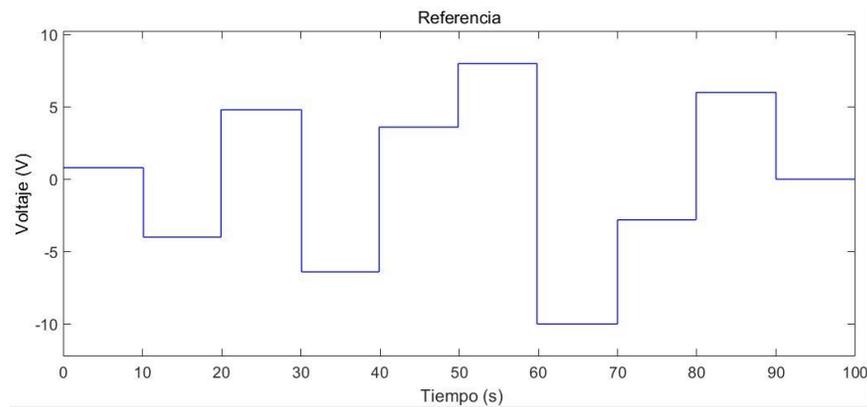
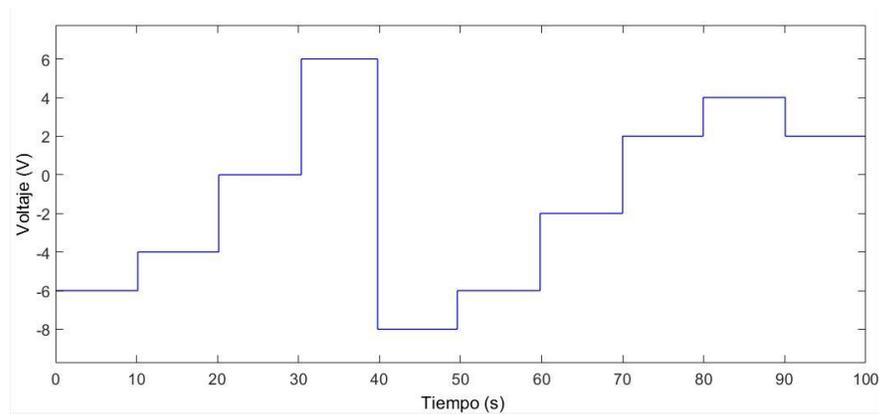
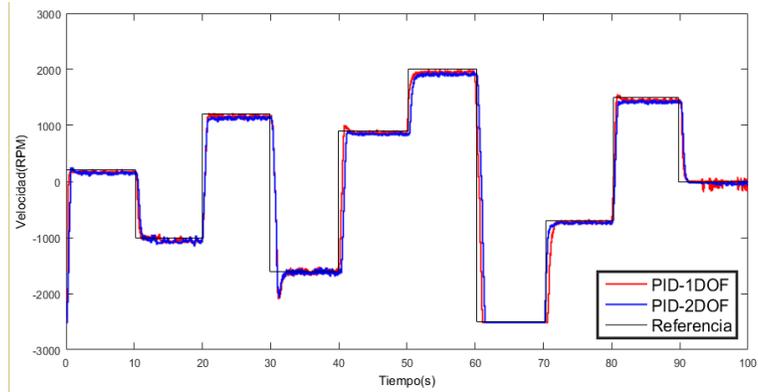


Figura 51. Trayectoria de validación 2

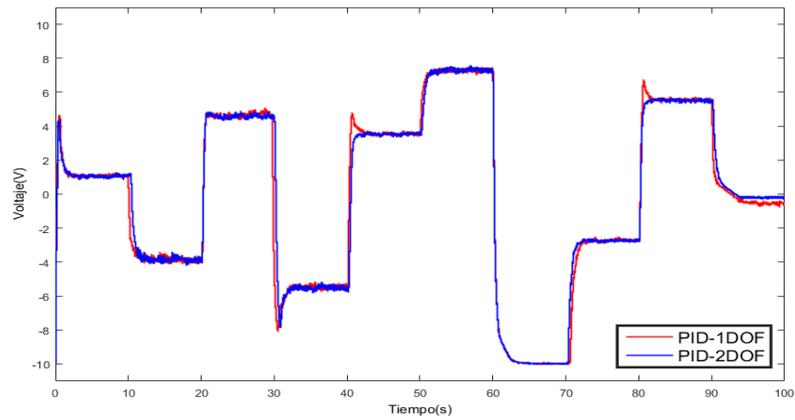


Se realizaron las pruebas correspondientes a la respuesta temporal, acción de control y señal de error para la trayectoria 1 y trayectoria 2. Como se observa en la Figura 52 y Figura 53, los controladores PID-1DOF y PID-2DOF realizan un seguimiento apropiado de las trayectorias 1 y 2, y de igual manera la acción de control tiene una respuesta acorde a lo esperado. Con respecto a la señal de error en la trayectoria 1 se observa que los dos controladores mantienen un desempeño similar pero el PID-2DOF logra volver al valor de 0 en forma más eficiente que el PID-1DOF.

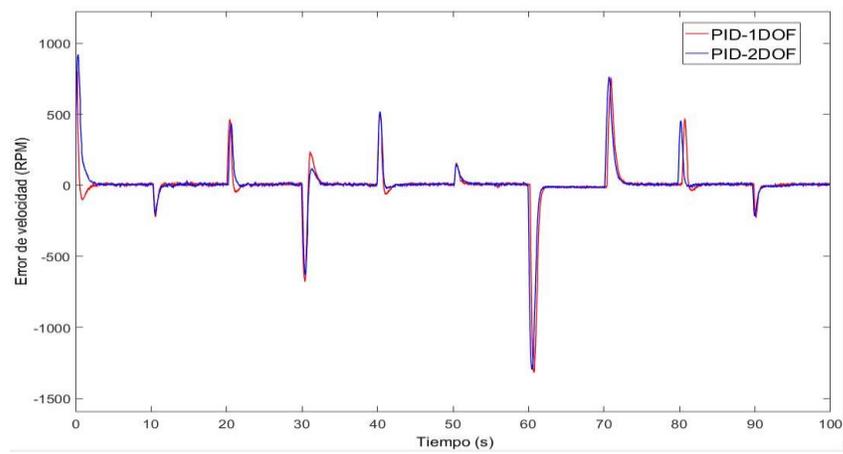
Figura 52. Respuesta temporal (a), acción de control (b), y señal de error (c) de los controladores PID-1DOF y PID-2DOF para la trayectoria 1 en operación nominal.



(a)

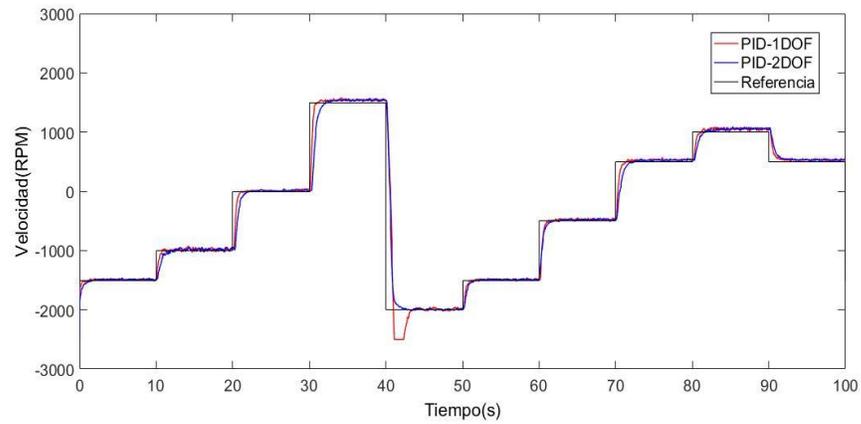


(b)

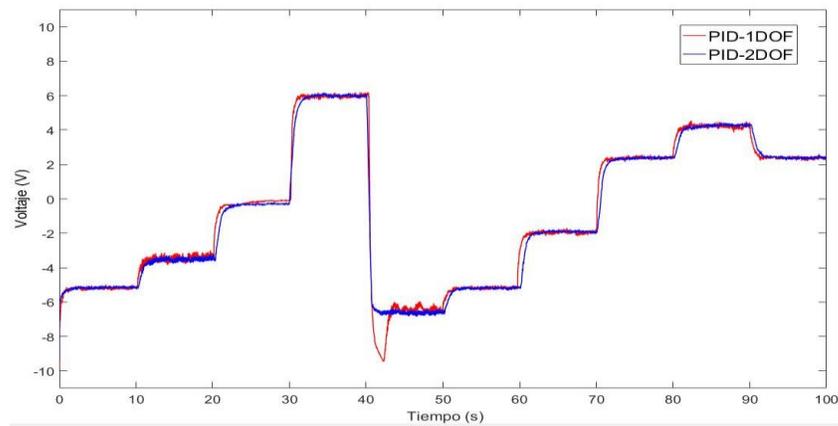


(c)

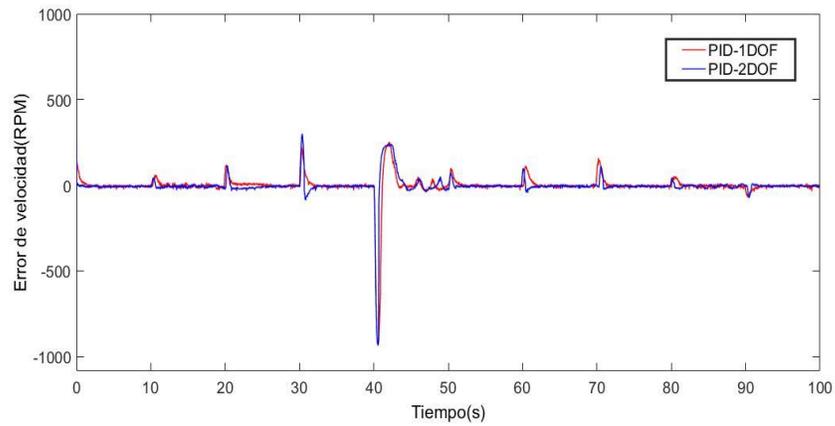
Figura 53. Respuesta temporal (a), acción de control (b), y señal de error (c) de los controladores PID-1DOF y PID-2DOF para la trayectoria 2 en operación nominal.



(a)



(b)



(c)

7. ANALISIS DE DESEMPEÑO DE LOS CONTROLADORES PID-1DOF Y PID-2DOF CON PRESENCIA DE PERTURBACIONES

El análisis del desempeño de los controladores PID-1DOF y PID-2DOF se realizó mediante tres pruebas en donde se observó el comportamiento de los controladores al someterlos a perturbaciones externas. Las pruebas consistieron en: una perturbación externa adicionada a la acción de control, ruido de medida en la señal de realimentación y una carga resistiva aplicada en la salida del generador del sistema. Para realizar las pruebas se emplearon las trayectorias de validación 1 y 2 mostradas anteriormente en la sección 6.4.

Adicionalmente, se evaluó el desempeño de los controladores PID-1DOF y PID-2DOF realizando los cálculos de los valores RMSE de la señal de error y el valor RMS de la acción de control.

7.1 Presencia de perturbación de entrada.

La perturbación se introdujo como un cambio de velocidad y sentido de giro del motor en $t=55s$ para la trayectoria 1 y $t=35s$ para la trayectoria 2, esto se logró por medio de un escalón unitario como se muestra en la Figura 54. Cada perturbación tuvo una duración de 1s y una amplitud de 3V la cual se agregó al lazo de realimentación como se observa en la Figura 55.

Figura 54. Escalón unitario para perturbación

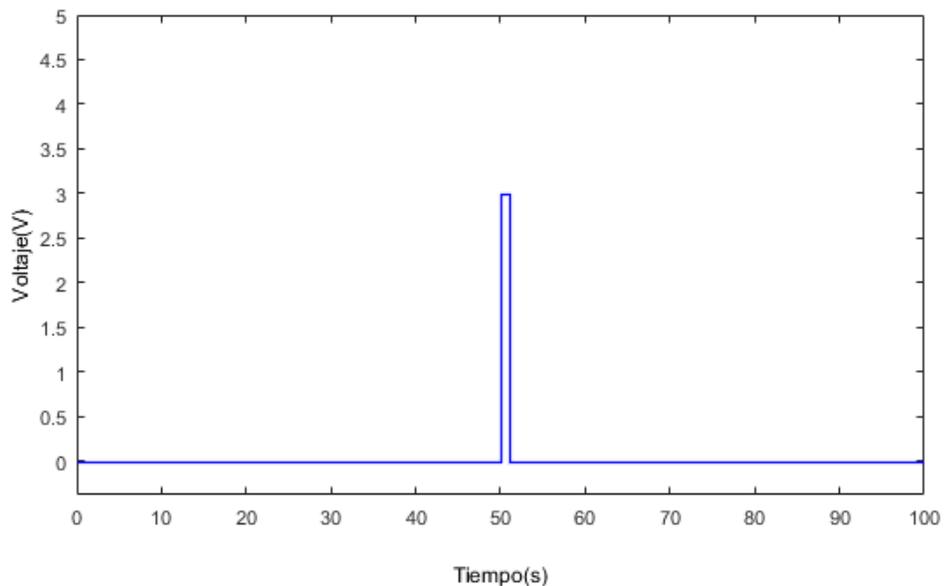
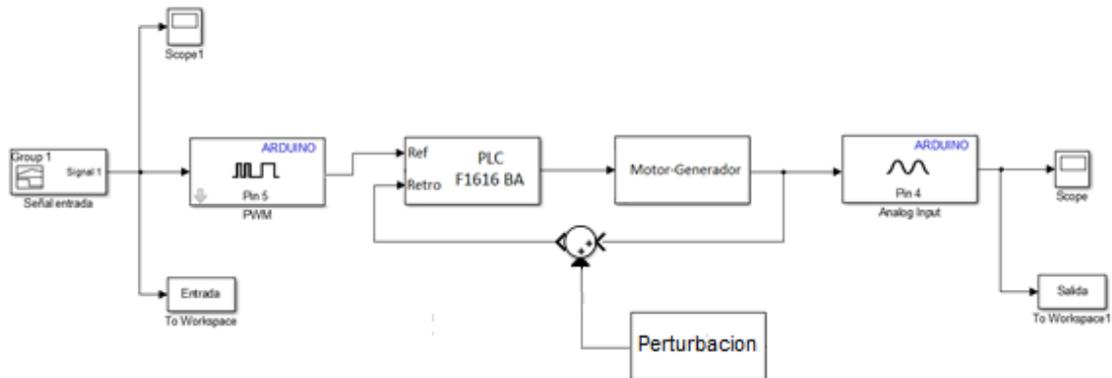
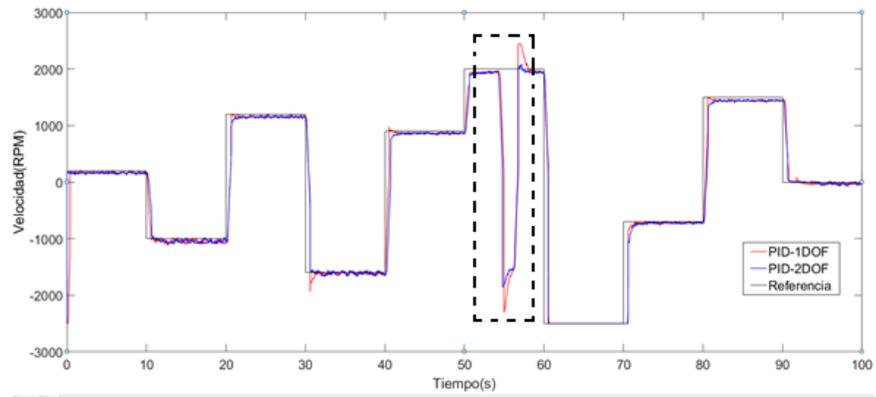


Figura 55. Esquema de conexiones para perturbación

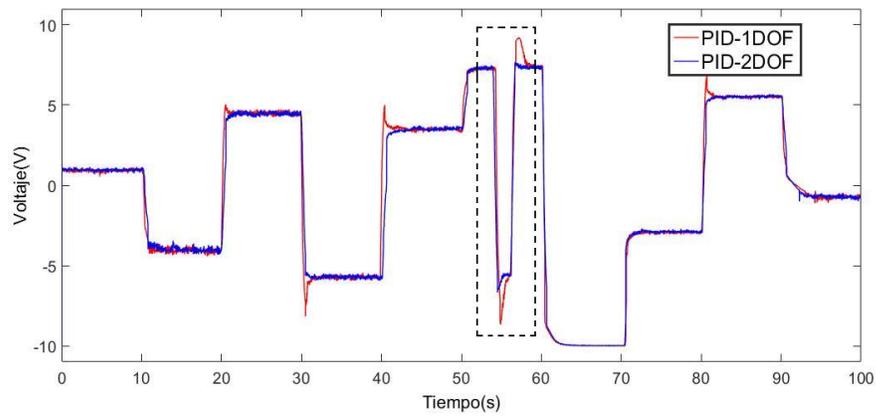


En la Figura 56 y Figura 57 se presentan la respuesta temporal, la acción de control y la señal de error para las trayectorias de validación 1 y 2. Como se observa en la Figura 56a, el PID-1DOF se afecta en mayor medida ante la presencia de la perturbación, lo cual se refleja con un sobreimpulso mayor tal como se resalta en el cuadro punteado. En el caso de la trayectoria 2, los controladores realizan un seguimiento cercano a la señal de referencia. De la acción de control mostrada en la Figura 56b se puede decir que el controlador PID-2DOF realiza menos esfuerzo para volver a su valor original luego al salir de la perturbación, lo cual se traduce en un menor consumo de energía. Finalmente en la Figura 56c y Figura 57c, la señal de error alcanza el valor de 0 de manera más rápida en el caso del controlador PID-2DOF con respecto al controlador PID-1DOF.

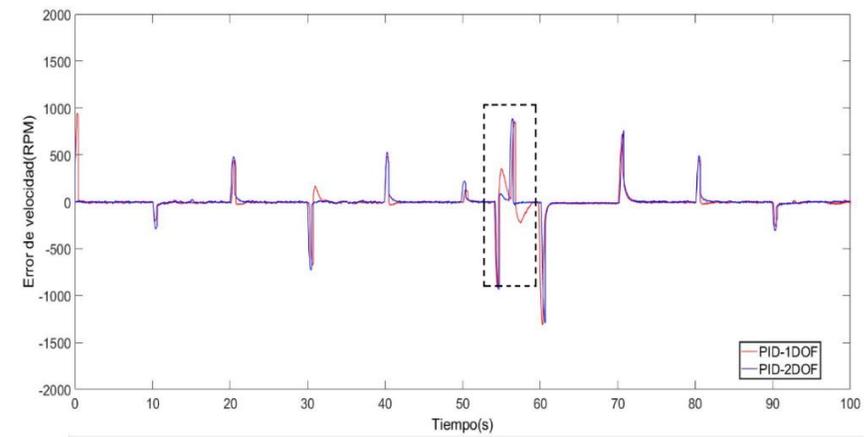
Figura 56. Respuesta temporal (a), acción de control (b), y señal de error (c) de los controladores PID-1DOF y PID-2DOF para la trayectoria de validación 1 con presencia de perturbación.



(a)

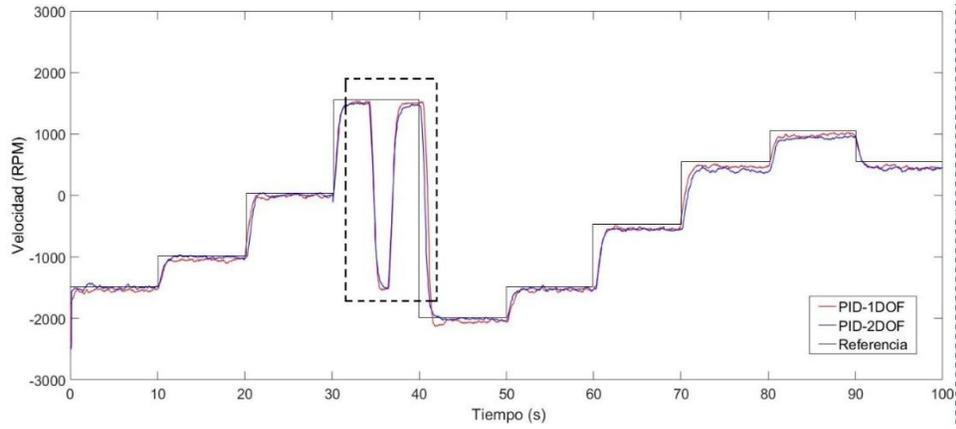


(b)

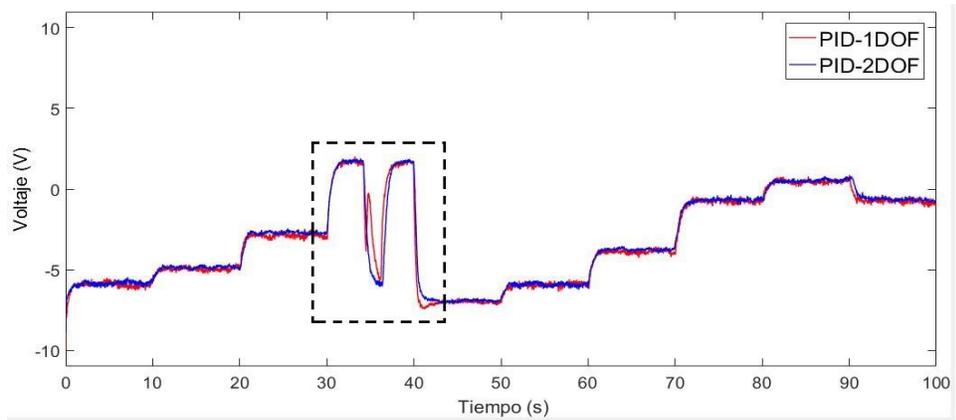


(c)

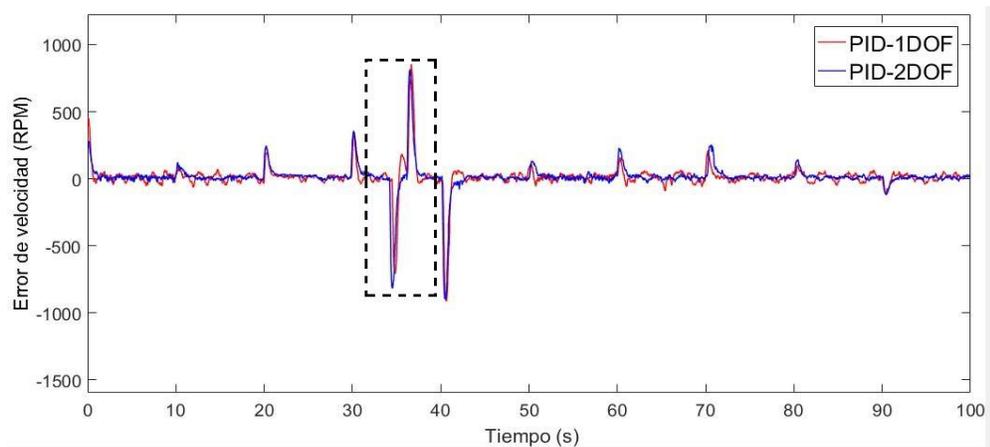
Figura 57. Respuesta temporal (a), acción de control (b), y señal de error (c) de los controladores PID-1DOF y PID-2DOF para la trayectoria de validación 2 con presencia de perturbación.



(a)



(b)

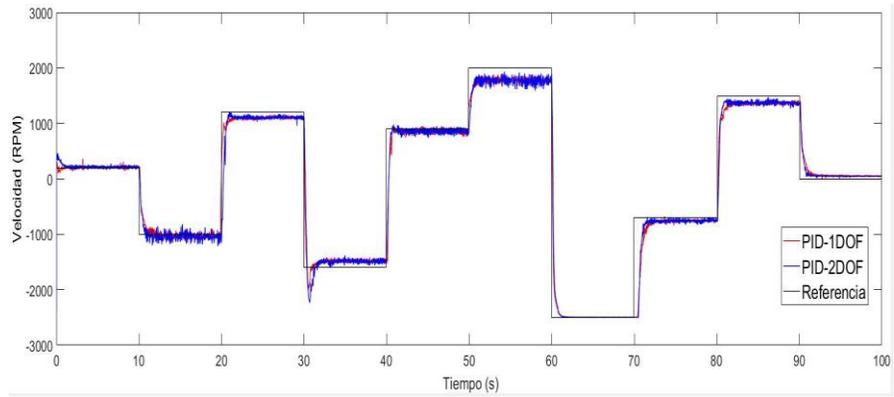


(c)

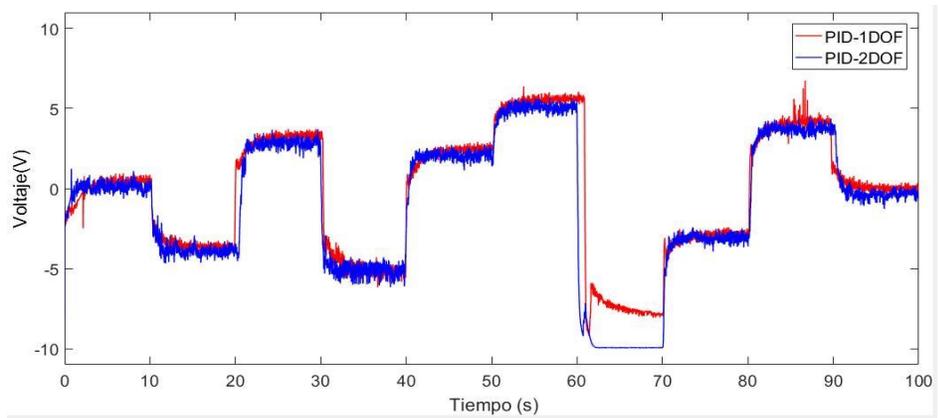
7.2 Ruido de medida en la señal de realimentación.

La salida de $\pm 7V$ del taco generador es la que provee la señal de realimentación de velocidad del sistema motor-generador. Ésta salida generalmente se compone de una señal ruidosa, por lo que para realizar la prueba de desempeño con presencia de ruido de medida, se optó por retirar el condensador que es utilizado para filtrar la señal de realimentación. Como se observa en las Figura 58 y Figura 59, los controladores PID-1DOF y PID-2DOF realizan un seguimiento de las trayectorias de validación 1 y 2 cercano a la trayectoria de referencia. En la acción de control se observa que el controlador PID-2DOF tiene una mejora con respecto al tiempo de establecimiento comparado con el PID-1DOF. Al observar la señal de error se puede notar que el controlador PID-2DOF presenta una mejora en el desempeño debido a que éste alcanza el valor de cero antes que el controlador PID-1DOF para algunos valores como se observa en la ampliación de la Figura 58c.

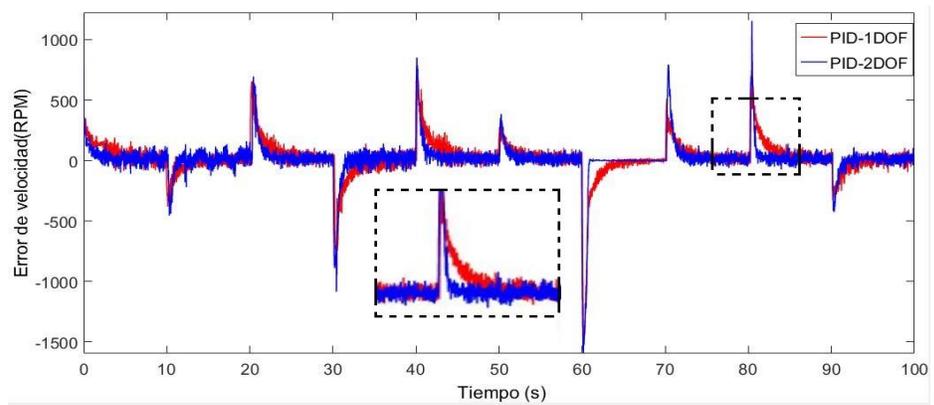
Figura 58. Respuesta temporal con ruido de medida (a), acción de control (b) y señal de error (c) de los controladores PID-1DOF y PID-2DOF para la trayectoria 1 de validación



(a)

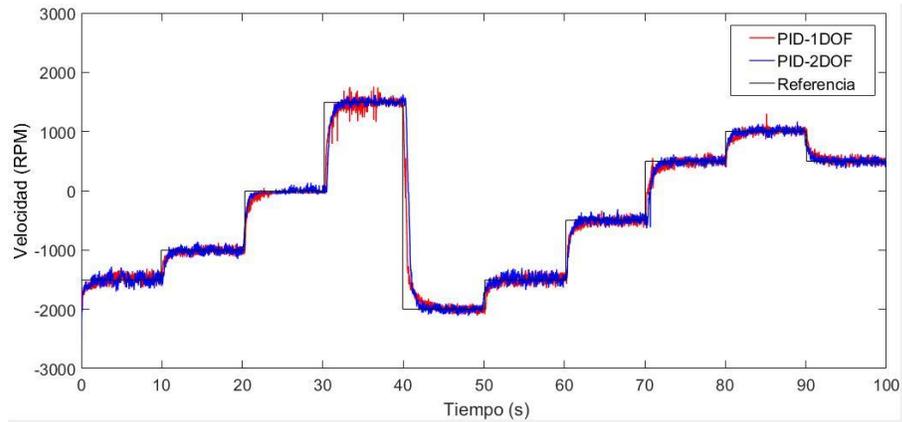


(b)

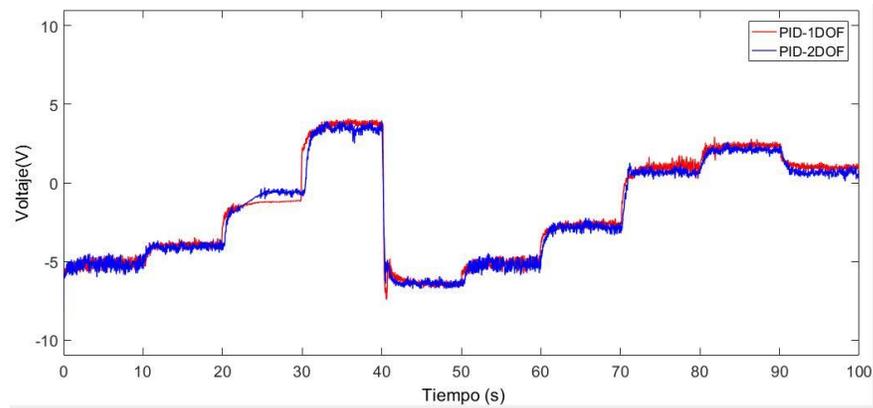


(c)

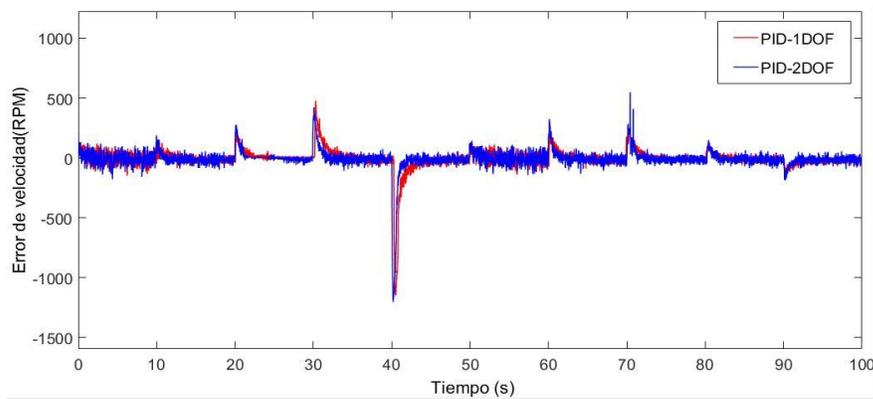
Figura 59. Respuesta temporal con ruido de medida (a), acción de control (b) y señal de error (c) de los controladores PID-1DOF y PID-2DOF para la trayectoria 2 de validación



(a)



(b)



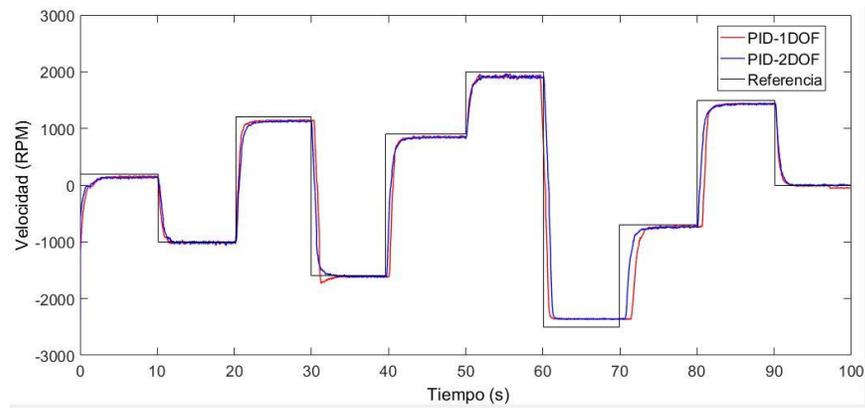
(c)

7.3 Carga resistiva en la salida del generador del sistema

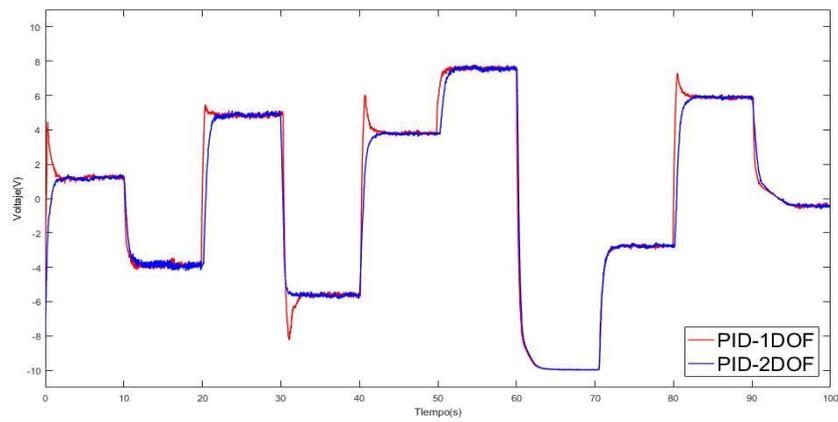
Se realizó la prueba de desempeño para los controladores PID-1DOF y PID-2DOF con una carga resistiva colocada en la salida de del generador del sistema. Para realizar esta prueba se conectaron dos bombillos de 60 Watts en paralelo, para de esta forma obtener una carga de 120 Watts.

En la Figura 60 y Figura 61 se muestra los resultados obtenidos de la respuesta temporal, acción de control y señal de error de los controladores PID-1DOF y PID-2DOF para las trayectorias de validación 1 y 2 con presencia de carga resistiva. Como se observa en las Figura 60a y Figura 61a, debido al efecto de la carga resistiva en la salida del motor-generador, los controladores PID-1DOF y PID-2DOF no logran acercarse a los valores de las amplitudes de las trayectorias recorridas. Para la acción de control se observa en la Figura 60b que el controlador PID-1DOF presenta sobreimpulso al realizar los cambios de amplitudes para la trayectoria 1 de validación. En la trayectoria 2 de validación, la acción de control en ambos casos presenta un desempeño similar. En la señal de error, se observa en la Figura 61c que para la trayectoria 2 de validación el controlador PID-2DOF presenta un tiempo de establecimiento menor a comparación del PID-1DOF.

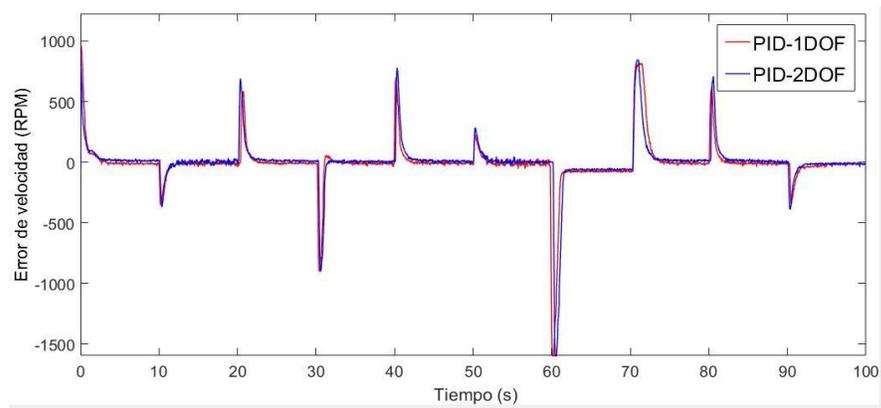
Figura 60. Respuesta temporal con carga resistiva (a), acción de control (b) y señal de error (c) de los controladores PID-1DOF y PID-2DOF para la trayectoria 1 de validación



(a)

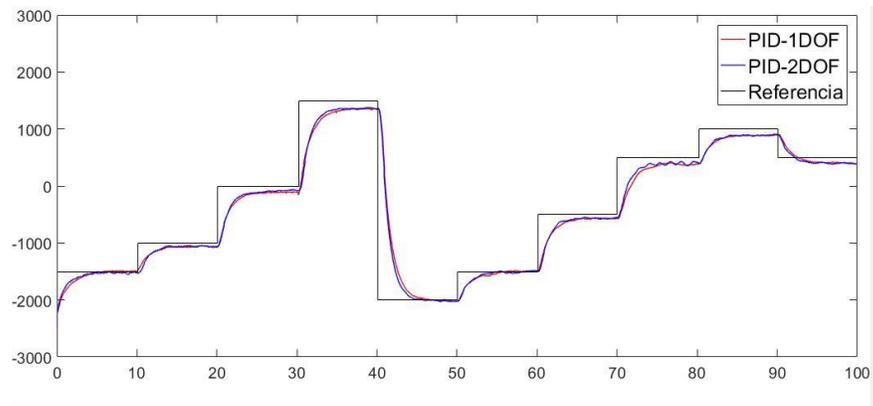


(b)

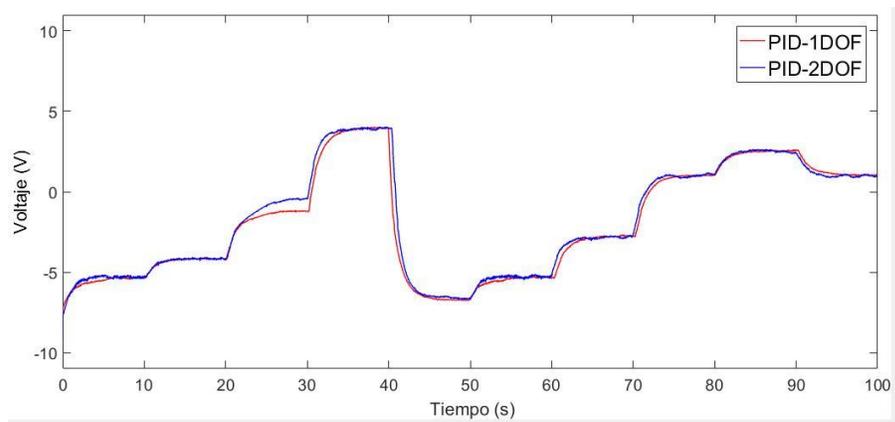


(c)

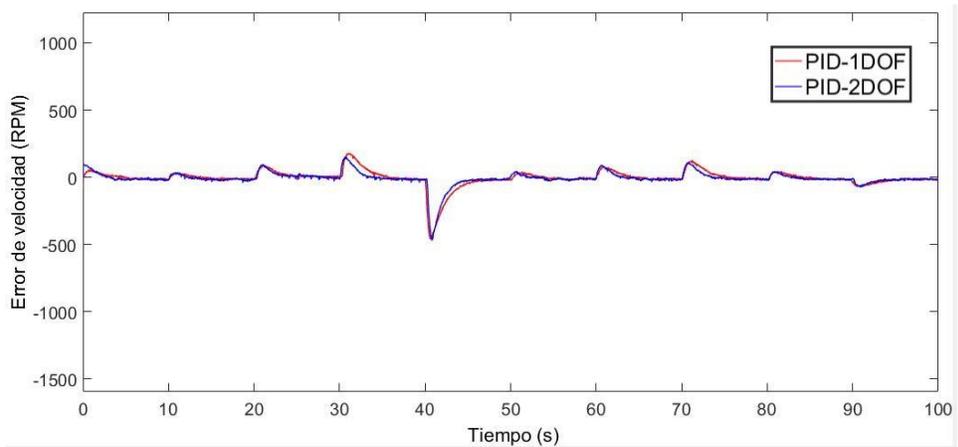
Figura 61. Respuesta temporal con carga resistiva (a), acción de control (b) y señal de error (c) de los controladores PID-1DOF y PID-2DOF para la trayectoria 2 de validación.



(a)



(b)



(c)

7.4 Análisis de resultados

Para poder establecer una comparación a partir de resultados cuantitativos entre los controladores PID-1DOF y PID-2DOF, se calcularon los valores RMSE y RMS (Root Mean Square Error y Root Mean Square). Éstos índices permiten cuantificar el desempeño teniendo en cuenta la señal de error y la acción de control respectivamente.

El valor RMSE mide la cantidad de error que hay entre dos conjuntos de datos, uno correspondiente a datos de referencia y el otro a datos observados. El valor RMSE se calcula por medio de la ecuación:

$$RMSE = \sqrt{\frac{1}{N} \sum_{i=1}^N (X_i - \tilde{X}_i)^2} \quad (36)$$

El valor RMS se define como la raíz cuadrada del promedio de un conjunto de elementos elevados al cuadrado. (Cálculo de RMSE, 2019) Se calcula por medio de la ecuación:

$$RMS = \sqrt{\frac{1}{N} \sum_{i=1}^N (X_i)^2} \quad (37)$$

En la Tabla 1 se presentan los resultados obtenidos a partir de los cálculos de los valores RMSE y RMS para la trayectoria de validación 1 con las condiciones de operación nominal, perturbación en la entrada, ruido de medida y carga resistiva en la salida del generador. Se puede observar que el controlador PID-2DOF presenta una mejora respecto al valor RMSE en cada una de las pruebas realizadas para la trayectoria de validación 1. En el caso de la presencia de perturbación es en donde más se destaca el desempeño del controlador PID-2DOF con respecto al controlador PID-1DOF con un porcentaje de mejora del 8.8%. Al analizar los valores RMS se observa que con respecto a la acción de control, no se tiene una diferencia notable, los dos controladores tienen un desempeño similar en este aspecto.

Tabla 1. Índices de desempeño para la trayectoria de validación 1

Prueba	Controlador	Índices de desempeños		% de mejora	
		RMSE	RMS	RMSE	RMS
Operación Nominal	PID1-DOF	0.608	5.207	5.1	1.44
	PID2-DOF	0.59	5.134		
Perturbación	PID1-DOF	0.584	5.206	8.8	2.43
	PID2-DOF	0.538	5.082		
Ruido	PID1-DOF	0.7115	4.482	3.28	1.58
	PID2-DOF	0.6888	4.412		
Carga	PID1-DOF	0.7887	5.364	5.5	2.67
	PID2-DOF	0.7475	5.224		

En la Tabla 2 se presentan los valores de los índices de desempeño para la trayectoria de validación 2 con las condiciones de operación nominal, perturbación en la entrada, ruido de medida y carga resistiva en la salida del generador. Como se puede observar, los resultados obtenidos para esta trayectoria son mejores en cuanto a la operación nominal y la presencia de carga resistiva para los controladores PID-1DOF y PID-2DOF. En la operación nominal y presencia de carga resistiva se tuvo un aumento considerable en el porcentaje de mejora del valor RMSE con respecto a la trayectoria de validación 1, pasando de un 5.1 y 5.5 a un 13.09 y 17.43 respectivamente. Con respecto a la prueba de perturbación, se observó que a pesar de que los valores RMSE obtenidos fueron menores en comparación a la trayectoria de validación 1, el porcentaje de mejora no superó el 3% a diferencia del 8.8% obtenido anteriormente. Con respecto al valor RMS, al igual que en las pruebas para la trayectoria de validación 1, no hubo una diferencia notable entre los desempeños de los controladores PID-1DOF y PID-2DOF, aunque cabe destacar que se obtuvieron valores menores con respecto a las pruebas realizadas anteriormente.

Tabla 2. Índices de desempeño para la trayectoria de validación 2

Prueba	Controlador	Índices de desempeños		% de mejora	
		RMSE	RMS	RMSE	RMS
Operación Nominal	PID1-DOF	0.3096	4.263	13.09	1.04
	PID2-DOF	0.2738	4.219		
Perturbación	PID1-DOF	0.4197	4.2	2.74	1.22
	PID2-DOF	0.4315	4.149		
Ruido	PID1-DOF	0.3861	3.718	1.29	0.43
	PID2-DOF	0.3814	3.702		
Carga	PID1-DOF	0.2642	3.908	17.43	1.92
	PID2-DOF	0.225	3.834		

Teniendo en cuenta los resultados obtenidos a partir de las pruebas realizadas para determinar los índices de desempeño de los controladores PID-1DOF y PID-2DOF, puede afirmarse que el controlador PID-2DOF que se implementó en el sistema motor-generador ofrece un desempeño notablemente superior al del controlador PID-1DOF al realizar trabajos de seguimiento en condiciones de operación nominal y presencia de carga resistiva. De lo anterior, se puede agregar que los dos controladores presentan una mejora general en el desempeño con una trayectoria que esté compuesta de escalones menos distantes como lo es la trayectoria de validación 2.

8. CONCLUSIONES

El proyecto presentó el diseño y la implementación de un control de velocidad para un sistema motor-generador por medio de un controlador PID de dos grados de libertad. Se realizó la identificación de la planta por medio de la herramienta IDENT de Matlab utilizando una trayectoria escalonada y se obtuvo un modelo de segundo orden con polos reales. Partiendo del modelo lineal obtenido, se diseñó un controlador PID-1DOF por medio del método de sintonización IMC para de esta manera realizar el diseño del controlador PID-2DOF teniendo en cuenta las constantes obtenidas. La finalidad del proyecto fue realizar un contraste entre los dos controladores y analizar su desempeño ante la presencia de perturbaciones, ruido de medida y presencia de carga resistiva.

Se realizó el proceso de discretización utilizando el método de diferencia atrás y posteriormente se implementó el método anti-windup back calculation para evitar que los controladores lleguen a un punto de saturación no deseado. La implementación se llevó a cabo de forma simulada empleado la herramienta Stateflow de Matlab y de forma práctica por medio de un controlador lógico programable.

Para analizar y contrastar el desempeño de los controladores PID-1DOF y PID-2DOF se realizaron pruebas con dos trayectorias escalonadas. Para evaluar el desempeño de los controladores se utilizaron los valores RMSE y RMS los cuales involucran las señales de error y acción de control respectivamente.

Al analizar los resultados obtenidos se observó que el controlador PID-2DOF presenta un desempeño superior al realizar seguimiento de trayectorias escalonadas. Con respecto a la señal de error, se obtuvo que el controlador PID-2DOF presentó valores RMSE menores con una diferencia notable respecto al controlador PID-1DOF. Los resultados obtenidos para valores RMS mostraron que para la acción de control no existe una diferencia importante entre los dos controladores, lo que quiere decir que los dos presentan un desempeño similar en este aspecto.

Al realizar una comparación entre los resultados obtenidos para las dos trayectorias, se observó que para una trayectoria escalonada que tenga cambios de valores distantes como la trayectoria de validación 1, los controladores PID-1DOF y PID-2DOF presentan valores RMSE Y RMS mayores en comparación a los obtenidos con la trayectoria de validación 2, la cual contiene cambios menos distantes entre sus escalones, lo que quiere decir que el desempeño de ambos controladores es mejor para éste tipo de trayectorias. De lo anterior, se puede resaltar que en el caso de la trayectoria de validación 2, el controlador PID-2DOF realiza un mejor trabajo ante la presencia de carga resistiva.

En términos generales se puede decir que el controlador PID-2DOF presenta una robustez notablemente mayor con respecto al controlador PID-1DOF cuando está en presencia de perturbaciones y ante la presencia de una carga resistiva.

9. BIBLIOGRAFÍA

- [1] R.A. Paz. (2001). the design of the PID controller. Klipsch School of Electrical and Computer Engineering. pp 1.
- [2] K. Ogata. (2010). Ingeniería de Control Moderna, Madrid: PEARSON EDUCACIÓN S.A. p.567.
- [3] K. J. Åström and T. Häggglund. (2009). Control PID Avanzado. Madrid: PEARSON EDUCACIÓN S.A
- [4] V. Alfaro. (2003) “Métodos de sintonización de controladores PID que operan como servomecanismos” Revista de la Universidad de Costa Rica, Vol 13, No 1 y 2
- [5] M. Paez. (2017). Control de velocidad de un Sistema motor generador empleando técnicas de control por modelo interno (Tesis de pregrado). Universidad Pontificia Bolivariana, Bucaramanga, Colombia. p 16.
- [6] D. E. Rivera. “Una metodología para la identificación integrada con el diseño de controladores imc-pid”. Iberoam. Automática e Informática Ind. RIAI, Vol. 4, no. 4, 2007.
- [7] Y. García and I. Lobo “Controlador PID integrado por la estructura de modelo interno (IMC) y lógica difusa”. Ciencia e Ingeniería, 2009.
- [8] V Alfaro Sistemas de Control proporcional, integral y derivativo: Algoritmos, análisis y ajuste. Universidad de Costa Rica, Costa Rica. 2016
- [9] M. Kumar and V.V. Patel. (2015). “Tuning of Two Degree of Freedom PID Controller for Second Order Processes”. International Journal of Science, Engineering and Technology Research (IJSETR), Vol.4 (5), 1543-1546.
- [10] V.M. Alfaro, R. Vilanova and O. Arrieta. (2009). “Robust tuning of Two-Degree-of-Freedom (2-DoF) PI/PID based cascade control systems”. ELSEVIER, Vol. 19(10), 1658-1670.
- [11] R. J. Vazquez Guerra, J.F. Marquez Rubio y B. del Muro Cuellar. (2009). “Controlador de dos grados de libertad para sistemas inestables con retardo usando un enfoque polinomial”. Asociación de México de control automático (AMCA), Congreso anual, 1-6.
- [12] V.M. Alfaro, O. Arrieta and R. Vilanova. (2009). “Control de Dos-Grados-de-Libertad (2-GdL) aplicados al “Benchmark” de Sistemas para Controladores PID”. Revista Iberoamericana de Automática e Informática Industrial (RIAI), Vol. 6(2), 56-67.

- [13] G.L.Deshmukh and C.B.Kadu. (2016). "Design of two degree of freedom PID controller for temperature control system". International Conference on Automatic Control and Dynamic Optimization Techniques (ICACDOT), 586-589.
- [14] A.K. Kadhim and A.A. Obed. (2017). "Brushless DC Motor Speed Control Based on PID Controller with 2-DOF and Anti-Windup Techniques". The Second Engineering Conference for Graduate Research Middle Technical University - Electrical Engineering Technical College Baghdad – Iraq, 1-13.
- [15] M.E. Hmidi, I.B. Salem and L.E. Amraoui. (2017). "Design and Modeling of PID Controller with 2DOF: Application to Thermal Phase of a Hybrid Vehicle". 14th International Multi-Conference on Systems, Signals & Devices (SSD), 419-424.
- [16] O.Y. Tenga, C.S. Horngb and H.W. Keat. (2015). "Positioning Control of XY Table Using 2-DOF PID Controller". Applied Mechanics and Materials, Vol. 761, 137-141.
- [17] N.G. Adar and R. Kozan. (2016). "Comparison between Real Time PID and 2-DOF PID Controller for 6-DOF Robot Arm", ACTA PHYSICA POLONICA A, Vol. 130(1), 269-271.
- [18] L. Angel and J. Viola, (2017). "Design and Statistical Robustness Analysis of FOPID, IOPID and SIMC PID Controllers Applied to a Motor-Generator System". IEEE LATIN AMERICA TRANSACTIONS, Vol.13 (2), 3724-3734.
- [19] M. Paez. (2017). Control de velocidad de un Sistema motor generador empleando técnicas de control por modelo interno (Tesis de pregrado). Universidad Pontificia Bolivariana, Bucaramanga, Colombia.
- [20] What is Arduino. Consultado noviembre 2018. Disponible en línea: <https://www.arduino.cc/en/Guide/Introduction>
- [21] Arduino Support from Simulink. Disponible en línea: <https://www.mathworks.com/hardware-support/arduino-simulink.html>
- [22] PLC (Controlador Lógico Programable). Consultado Noviembre 2018. Disponible en línea: <http://www.logicbus.com.mx/plc.php>
- [23] R. M. M. Introducción a los controladores Lógicos Programables. Digital I
- [24] F1616 User manual. TRIANGLE RESEARCH INTERNATIONAL
- [25] Funcionamiento del motor de corriente directa. Consultado Noviembre 2018. Disponible en línea: http://www.asifunciona.com/electrotecnia/af_motor_cd/af_motor_cd_6.htm

[26] M. A. Rodríguez. Máquinas de corriente continua. Universidad de Cantabria, España. 2017

[27] H. Markaroglu. (2006) "Tracking time adjustment in back calculation anti-windup scheme". Istanbul Technical University, Faculty of Electrical and Electronics Engineering, Control Engineering Departement.

[28] Cálculo de RMSE. Consultado Abril 2019. Disponible en línea: <https://www.gestiondeoperaciones.net/proyeccion-de-demanda/calculo-de-la-raiz-del-error-cuadratico-medio-o-rmse-root-mean-squared-error/>

10. ANEXOS

