



Universidad Veracruzana

## **TÍTULO:**

**“Desarrollo de un conjunto de experiencias de aprendizaje aplicadas a la enseñanza de la ingeniería de control, utilizando QNET-DCMCT, NI-ELVIS II, LabVIEW y Matlab.”**

**MAESTRIA EN INGENIERÍA ELÉCTRICA  
OPCIÓN: CONTROL**

**Nombre del Alumno:**

**HÉCTOR SADOT DEL ANGEL ZUMAYA**

**Nombre del Director de tesis:**

**M.I. Alberto Pedro Lorandi Medina**



# Titulo

**“Desarrollo de un conjunto de experiencias de aprendizaje aplicadas a la enseñanza de la ingeniería de control, utilizando QNET-DCMCT, NI-ELVIS II, LabVIEW y Matlab.”**

# Resumen

Este trabajo de investigación se sustenta bajo el desarrollo de un conjunto de experiencias de aprendizaje aplicadas a la enseñanza de la ingeniería de control, en las cuales se abordan estrategias de modelado, identificación, control y simulación de un motor de CD utilizando el QNET-DCMCT en conexión con el módulo de interface de adquisición NI-ELVIS II, empleando el software de LabVIEW, Matlab y Simulink. Se busca abordar el proceso de enseñanza aprendizaje por medio de ejercicios prácticos con metodologías propias de la ingeniería de control aplicadas al modelado matemático del sistema, su simulación y control utilizando controladores PID y Difuso (Fuzzy). Además se contrasta el desempeño del sistema cuando es sometido a perturbaciones con cargas estáticas.

# Abstract

This research work is based on the development of a set of learning experiences applied to the teaching of control engineering, in which strategies of modeling, identification, control and simulation of a Motor DC using the QNET-DCMCT in connection with the NI-ELVIS II acquisition interface module, using the LabVIEW, Matlab and Simulink software. It seeks to address the teaching-learning process through practical exercises with control engineering methodologies applied to the mathematical modeling of the system, its simulation and control using PID and Fuzzy controllers. In addition, the performance of the system is contrasted when subjected to disturbances with static charges.

## ÍNDICE

1	CAPÍTULO I.....	11
1.1	Introducción .....	11
1.2	Justificación .....	12
1.3	Objetivos .....	13
1.3.1	Objetivo general .....	13
1.3.2	Objetivos Específicos .....	13
1.4	Antecedentes .....	14
2	CAPÍTULO II .....	16
2.1	Sistemas QNET-DCMCT de Quanser .....	16
2.1.1	Introducción al sistema QNET-DCMCT.....	16
2.1.2	Descripción del sistema.....	17
2.1.3	Esquemático del sistema.....	20
2.1.4	Especificaciones .....	21
2.1.5	Ambiental .....	22
2.1.6	Guía de preparación.....	22
2.1.7	Procedimiento de configuración de ELVIS II. ....	24
2.1.8	Consejos de LabVIEW QNET .....	26
2.1.9	Búsqueda de Fallos .....	30
2.1.10	Apoyo técnico del equipo de Quanser.....	32
3	CAPÍTULO III .....	34
3.1	Modelado matemático del sistema QNET-DCMCT a controlar .....	34
3.1.1	Modelado analítico del sistema .....	34
3.1.2	Modelado experimental o identificación. ....	38
3.1.3	Modelado del sistema utilizando la herramienta <i>System Identification</i> de Matlab .....	39
3.1.4	Modelado usando QNET-DCMCT Modeling de LabVIEW .....	47
4	CAPÍTULO 4 .....	56
4.1	Diseños y estrategias de sintonización usando el controlador PID .....	56



4.1.1	Diseño Analítico del controlador PI a partir de un desempeño determinado al sistema de Quanser.....	56
4.1.2	Diseño del controlador PI Utilizando la herramienta PID Tuner. ....	61
4.1.3	Implementación Virtual de Posición utilizando herramientas de labview y QNET-DCMCT Position Control. ....	71
5	CAPÍTULO 5 .....	78
5.1	Sistema Difuso (Fuzzy) .....	78
5.1.1	Control de posición en el sistema DCMCT aplicando control difuso (Fuzzy) 78	
5.1.2	Diseño del Controlador Difuso (Fuzzy) aplicado en el sistema DCMCT de Quanser 80	
5.1.3	Herramientas de diseño en controladores difusos. ....	83
5.1.4	Tipo y diseño de las funciones de membresía. ....	91
5.1.5	Identificación del método de defusificación.....	92
5.1.6	Desempeño en operación de los diseños de control utilizando Fuzzy. ....	95
6	Conclusiones .....	98
7	Trabajos futuros.....	99
8	Referencias .....	100

## Índice de Figuras

<i>Figura 2-1: Entrenador QNET de control de motores de CD (DCMCT).....</i>	<i>16</i>
<i>Figura 2-2: Disposición de componentes del QNET DCMCT .....</i>	<i>17</i>
<i>Figura 2-3: Componentes del motor QNET DC .....</i>	<i>18</i>
<i>Figura 2-4: Esquema del sistema QNET-DCMCT.....</i>	<i>20</i>
<i>Figura 2-5: Conexión de un entrenador QNET .....</i>	<i>22</i>
<i>Figura 2-6: Componentes del NI-Elvis II .....</i>	<i>23</i>
<i>Figura 2-7: Los LEDs de Listo y Energía sobre el ELVIS II NI .....</i>	<i>25</i>
<i>Figura 2-8: Deben de estar encendidos los LEDs QNET .....</i>	<i>25</i>
<i>Figura 2-9: Las gráficas que se necesitan escalar.....</i>	<i>26</i>
<i>Figura 2-10: La gráfica que se necesita escalar .....</i>	<i>26</i>
<i>Figura 2-11: Eje-y de la gráfica se ha ajustado.....</i>	<i>27</i>
<i>Figura 2-12: Oprima botón derecho ratón en la gráfica y elija Imagen Exportación Simplificada .....</i>	<i>28</i>
<i>Figura 2-13: Caja de diálogo Exportar Imagen Simplificada .....</i>	<i>29</i>
<i>Figura 2-14: Imagen resultante .....</i>	<i>29</i>
<i>Figura 2-15: Restableciendo y actuando el auto-test en el ELVIS II .....</i>	<i>31</i>
<i>Figura 2-16: Alcances en la medición del servomotor QNET-DCMCT en Vi's.....</i>	<i>32</i>
<i>Figura 3-1: Diagrama equivalente de un motor de CD .....</i>	<i>34</i>
<i>Figura 3-2: Archivo .m de Matlab para obtener la curva de respuesta .....</i>	<i>38</i>
<i>Figura 3-3: Gráfico de la respuesta transitoria al sistema.....</i>	<i>39</i>
<i>Figura 3-4: Importar las variables de entrada y salida .....</i>	<i>40</i>
<i>Figura 3-5: Aplicación System Identification de Matlab .....</i>	<i>41</i>
<i>Figura 3-6: Ventana principal de System Identification Toolbox de Matlab. ....</i>	<i>41</i>
<i>Figura 3-7: Importación de modelos.....</i>	<i>42</i>
<i>Figura 3-10: Modelo de salida .....</i>	<i>43</i>
<i>Figura 3-11: Opciones de procesamiento .....</i>	<i>44</i>
<i>Figura 3-12: Selección de la función de transferencia en "Process Models" .....</i>	<i>45</i>
<i>Figura 3-13: Respuesta transitoria de la planta "Model Output" .....</i>	<i>45</i>
<i>Figura 3-14: Importar el modelo "P1" con los datos obtenidos a la prueba analítica.....</i>	<i>46</i>
<i>Figura 3-15: Señal escalón aplicada al sistema .....</i>	<i>47</i>

Figura 3-16: Señal de respuesta en la salida del sistema .....	47
Figura 3-17: Archivo QNET-DCMCT Modeling con su respuesta gráfica de medición .....	49
Figura 3-18: Respuesta al escalón en sistema de lazo abierto .....	50
Figura 3-19: Gráfico utilizando la extracción de datos de la tabla de Excel .....	51
Figura 3-20: Respuesta comparativa entre la respuesta real y las simuladas al DCMCT .....	53
Figura 3-21: Gráfico comparativo entre la respuesta real y las simuladas al DCMCT .....	53
Figura 4-1: Controlador PI aplicado al sistema .....	57
Figura 4-2: Archivo del editor de Matlab para sintonización PI como servo y regulador. ....	60
Figura 4-3: Grafica y parámetros de respuesta del controlador PI .....	60
Figura 4-4: Diagrama de bloques del sistema de control realimentado con PI .....	61
Figura 4-5: Sintonización del PI utilizando la herramienta PID Tuner .....	62
Figura 4-6: Importa el sistema desde "Command Windows" en PID Tuner .....	63
Figura 4-7: Seleccionando el tipo de controlador a sintonizar .....	63
Figura 4-8: Seleccionando el tipo de controlador a sintonizar .....	63
Figura 4-10: Ejecución de la herramienta PID Tuner desde simulink .....	64
Figura 4-9: Respuesta al escalón unitario en el sistema PID Tuner .....	64
Figura 4-10: Ajuste de parámetros en la respuesta del controlador PI en el sistema .....	65
Figura 4-12: Respuesta sintonizada elegida para exportación al sistema DCMCT .....	65
Figura 4-13: curva de respuesta al controlador PI sintonizado desde PID Tuner. ....	66
Figura 4-16: Diagrama de bloques, vi de posición del motor de cd con PID en LabVIEW .....	67
Figura 4-17: Archivo vi de posición del motor de cd utilizando PID (Frontal Panel) .....	68
Figura 4-18: Carga Permanente acoplada al sistema DCMCT .....	69
Figura 4-19: Aplicación del diseño de ganancias en sintonización usando PID .....	69
Figura 4-20: Curva de respuesta al sistema utilizando PID Tuner para sintonización .....	70
Figura 4-19: Diagrama de sistema de lazo cerrado de diseño de Quanser .....	71
Figura 4-20: Instrumentación virtual en LabVIEW QNET-DCMCT Position Control.vi .....	72
Figura 4-21: Grafica de respuesta a los parámetros del controlador usando DCMCT position control. ....	73
Figura 4-22: Curva de respuesta a la señal corregida con la configuración de sintonización. ....	74
Figura 4-23: Diagrama de simulación que permite comparar el desempeño de los métodos utilizados .....	75
Figura 4-24: Curvas de respuesta utilizando los tres controladores según la tabla 4-4 .....	75
Figura 5-1: Lógica difusa (Fuzzy) contra lógica clásica .....	79

<i>Figura 5-2: Valores lingüísticos de la variable difusa “Posición” .....</i>	<i>80</i>
<i>Figura 5-3: Configuración del canal de datos para posición.....</i>	<i>80</i>
<i>Figura 5-4: Diseño de la obtención de la posición con SubVis de QNET .....</i>	<i>81</i>
<i>Figura 5-5: Diagrama de bloques del archivo DCMCTFuzzy.vi .....</i>	<i>81</i>
<i>Figura 5-6: Panel frontal del archivo DCMCTFuzzy.vi .....</i>	<i>82</i>
<i>Figura 5-7: Modo de acceso a la herramienta para Control Difuso.....</i>	<i>83</i>
<i>Figura 5-8: Fuzzy System Designer (Editor de variables lingüísticas) .....</i>	<i>84</i>
<i>Figura 5-9: Fuzzy Set Edit Variable .....</i>	<i>85</i>
<i>Figura 5-10: Variables de términos lingüísticos asignados del diseño 1 (7variables).....</i>	<i>87</i>
<i>Figura 5-11: Variables de términos lingüísticos asignados del diseño 2 (5 Variables). .....</i>	<i>87</i>
<i>Figura 5-12: Grafica de funciones de membresía para la variable de entrada y salida del diseño para la variable de control (diseño 7 variables).....</i>	<i>88</i>
<i>Figura 5-13: Grafica de funciones de membresía para la variable de entrada y salida del diseño para la variable de control (diseño 5 variables).....</i>	<i>88</i>
<i>Figura 5-14: Grafica de funciones de membresía para la variable de salida diseño 1 .....</i>	<i>89</i>
<i>Figura 5-15: Grafica de funciones de membresía para la variable de salida diseño 2 .....</i>	<i>89</i>
<i>Figura 5-16: Rulebase Editor .....</i>	<i>92</i>
<i>Figura 5-17: Reglas Aplicadas en Fuzzy System Designer diseño 1 (7 Variables) .....</i>	<i>93</i>
<i>Figura 5-18: Reglas Aplicadas en Fuzzy System Designer diseño 2 (5 Variables) .....</i>	<i>93</i>
<i>Figura 5-19: Salidas de los controladores Fuzzy en función de las variables de entrada. ....</i>	<i>95</i>
<i>Figura 5-20: Respuesta al diseño 1 (7 variables) sin carga permanente al sistema DCMCT .....</i>	<i>96</i>
<i>Figura 5-21: Respuesta al diseño 1 (5 variables) sin carga permanente al sistema DCMCT .....</i>	<i>96</i>
<i>Figura 5-22: Respuesta con carga permanente al sistema DCMCT utilizando control Fuzzy .....</i>	<i>97</i>
<i>Figura 5-23: Respuesta con carga permanente al sistema DCMCT utilizando control Fuzzy .....</i>	<i>97</i>



## Índice de Tablas

<i>Tabla 2-1 Nomenclatura de componentes del QNET-DCMCT .....</i>	<i>17</i>
<i>Tabla 2-2: Parámetros del modelo y especificaciones del amplificador de potencia PWM .....</i>	<i>21</i>
<i>Tabla 2-3: Especificaciones de los parámetros del sensor QNET-DCMCT.....</i>	<i>21</i>
<i>Tabla 2-4: Condiciones de funcionamiento ambiental del entrenador de control de motor CC de QNET .....</i>	<i>22</i>
<i>Tabla 2-5: Componentes del ELVIS II .....</i>	<i>23</i>
<i>Tabla 3-1: Parámetros obtenidos de placa del sistema QNET-DCMCT de Quanser. ....</i>	<i>35</i>
<i>Tabla 4-1: Características de las constantes del controlador PID .....</i>	<i>56</i>
<i>Tabla 4-2: Descripción del vi diseñado para control de posición del motor CD por PID .....</i>	<i>68</i>
<i>Tabla 4-3: Resumen de resultados del control de posición usando vi de Quanser .....</i>	<i>74</i>
<i>Tabla 4-4: Resultados comparativos en los diseños de sintonización usando PID .....</i>	<i>76</i>
<i>Tabla 5-1: Descripción del vi DCMCT Fuzzy .....</i>	<i>82</i>
<i>Tabla 5-2: diseños del error de señal de posición. ....</i>	<i>86</i>
<i>Tabla 5-3: Los tres conjuntos difusos para la señal <math>e'(t)</math>.....</i>	<i>87</i>
<i>Tabla 5-5: Ahora las funciones de membresía para la variable de salida .....</i>	<i>90</i>
<i>Tabla 5-5: Cuadro de reglas diseño 1 establecidas para el control difuso.....</i>	<i>91</i>
<i>Tabla 5-6: Cuadro de reglas diseño 2 establecidas para el control difuso.....</i>	<i>91</i>
<i>Tabla 5-7: Referencias y base de reglas diseño 1 (7 variables). ....</i>	<i>94</i>
<i>Tabla 5-8: Referencias y base de reglas diseño 2 (5 variables). ....</i>	<i>94</i>



# CAPÍTULO I

# 1 CAPÍTULO I

## 1.1 INTRODUCCIÓN

Existen muchas necesidades para establecer aportes al sistema de enseñanza aprendizaje que puedan ser aplicadas en la ingeniería de control de forma práctica, esto aunado a la falta de laboratorios pertinentes equipados en el área de sistemas de control que converjan en la enseñanza adquirida en la experiencias educativas en la DES del área técnica de la universidad veracruzana región Veracruz, surge la idea de poder efectuar un aporte por medio de ejercicios prácticos con metodologías propias de la ingeniería de control aplicadas por parte ejercicio de laboratorio y realizar pruebas bajo ciertas condiciones de operación valiéndonos del equipo QNET-DCMCT de Quanser para análisis de operación de motores de CD, anexando el control de posición estableciendo un sistema desbalanceado por perturbación con cargas estáticas, utilizando diversas estrategias de modelado, identificación, control y simulación, que pueden ser recurridas en el plan de las experiencias educativas a las cual va dirigido.

Se utiliza una tarjeta de adquisición de datos ELVIS II, una plataforma de interface de software LabVIEW para trabajar en tiempo real, un módulo DCMCT de Quanser hardware exclusivo para laboratorios de control y la adaptación de cargas estáticas que provoquen desbalance sobre una disco de carga acoplado a la fecha del motor de CD, dando así un sistema completo en el cual sea manipulable en su desempeño debido a esto.

Partiendo desde el capítulo 2 donde se denota la utilización de equipos a utilizar aplicados en el sistema de estabilización de un motor de CD en su posición y velocidad, así como su principio de funcionalidad haciendo uso de la planta (proceso) real física de entrenamiento QNET-DCMCT de QUANSER (tarjeta adicional del NI ELVIS II), con el paquete de software de control LabVIEW y de un Laboratorio Educativo en Instrumentación Virtual de la National Instruments,

En el capítulo 3 se describe los procedimientos en el sistema QNET-DCMCT para los métodos para la obtención del modelado del sistema de forma analítica y de forma experimental, donde se utilizaran la representación matemática en simulación para obtención de respuesta en sistemas control al utilizar herramientas de Matlab, System Identification y vi de Quanser de modelado QNET-DCMCT Modeling para LabVIEW y establecer una comparativa en cuanto al comportamiento y la aproximación más cercana a la respuesta real del sistema.

En los capítulos 4 y 5 se efectuaran procedimientos de diseño utilizando el modelo del sistema obtenido en el capítulo 3. Para establecer la implementación del controlador PID parametrizando y usando criterios de desempeño, diseño de un PI utilizando PID Tuner de Simulink bajo el modelo de respuesta a la señal escalón, El uso de herramientas de LabVIEW con vi de Quanser para control de posición del motor de CD en diseño de parámetros PID y utilizando Fuzzy System Designer para generar un controlador en LabVIEW aplicado en el sistema. Esto para establecer la estabilidad y los tiempos de respuesta así como la demostración de operación y preguntas básicas sobre el sistema de control y estabilidad al ponerlo en prueba utilizando perturbaciones con cargas estáticas para su desbalance y comparar el desempeño de todos los métodos utilizados en el sistema de Quanser.

## 1.2 JUSTIFICACIÓN

Es evidente que existe una gran necesidad de poder establecer un conjunto de experiencias de aprendizaje para alcanzar una gran relevancia en el proceso de enseñanza aplicada al área de la ingeniería de control, Recurrir a dispositivos como el QNET-DCMCT, el NI-ELVIS II y los software de aplicación técnica como LabVIEW y Matlab generan una gran oportunidad para utilizarlos como medio para realizar ejercicios prácticos donde se busquen abordar metodologías utilizadas en la ingeniería de control, logrando un gran beneficio en el desempeño de enseñanza aprendizaje mediante el uso de estas herramientas, El desempeño alcanzado al someter al sistema nos genera un gran campo de acción y desarrollo.

## 1.3 OBJETIVOS

### 1.3.1 Objetivo general

Fortalecer el proceso de enseñanza aprendizaje por medio de ejercicios prácticos utilizando el QNET-DCMCT, NI-ELVIS II, empleando herramientas de LabVIEW, Matlab y Simulink en la ingeniería de control utilizando controladores PID y Difuso (Fuzzy) en las experiencias educativas en la DES del área técnica de la Universidad Veracruzana región Veracruz.

### 1.3.2 Objetivos Específicos

Para lograr el objetivo general, a continuación se explican los objetivos específicos involucrados:

- ❖ Desarrollar prueba con la utilización del equipo didáctico QNET-DCMCT de Quanser y NI-ELVIS II con característica de estar sometido a perturbaciones con una carga estática.
- ❖ Obtener el modelo matemático del sistema dinámico de forma analítica y de forma experimental con las características en que se presenta el servomotor del QNET-DCMCT de Quanser.
- ❖ Diseñar mediante una plataforma de interface el software LabVIEW estrategias de control como el control Proporcional-Integral-Derivativo (PID) y el control lógico difuso (Fuzzy).
- ❖ Realizar simulación a través de los datos obtenidos en las pruebas en el sistema para comparar el desempeño de los métodos bajo análisis realizados.

## 1.4 Antecedentes

En algunas tareas industriales a controlar en el pasado y actualmente donde no es necesaria una alta precisión, el diseño de controladores se realiza por medio de esquemas de control clásicos por ser relativamente fáciles de aplicar y de muy buen resultado. Tal es el caso de los controladores (P, PD, PID) con los que se pueden obtener una gran precisión en los sistemas de control de los procesos, pero con el inconveniente de la necesidad de conocer ciertamente el modelo matemático del mismo sistema.

La Mayoría de las plantas industriales exhiben niveles sustanciales de problemas de inestabilidad y de incertidumbre. Se plantean dificultades desde el punto de vista de la ingeniería de control y el diseño usando diferentes metodologías. A fin de corregir estas dificultades en el comportamiento y desempeño del controlador actuando en el sistema. Existen diversos esquemas que se han desarrollado en forma de control, como el control proporcional-integral-derivativo PID, compensación de adelanto de fase, control robusto y otros más. Durante las últimas décadas hemos sido testigo de un creciente en el interés del control difuso de sistemas y muchas aplicaciones de éxitos que están siendo utilizadas de manera eficiente en numerosos problemas de control que presenta grandes dificultades, es por ello la importancia de abordarlos adecuadamente.

Un sistema lineal como lo es la aplicación rotativa para operación de un motor de CD, se presenta en un número importante de aplicaciones académicas para probar diferentes algoritmos de control establecidos en las experiencias educativas de control. En él se puede aplicar diversas estrategias de la ingeniería de control clásico y el control difuso (Fuzzy).

El sistema del motor de CD que contiene del QNET-DCMCT se considera como una configuración de control de ingeniería de referencia cuyo concepto subyacente en que se puede aplicar a un problema en la estabilización, esto bajo diversos arreglos de sistemas en los cuales se pueden generar un desbalance y lograr poder corregirlo en diferentes tipos de sistemas de control (Lorandi-Medina, Hermisa-Saba, Ladrón-de-Guevara-Durán, & Hernández-Silva, 2011). Debido a que las incertidumbres son inevitables en este tipo de sistemas dinámicos, resulta factible también el poder aplicar del mismo modo un sistema de pruebas al sistema de control desarrollando la lógica difusa (borrosa y/o fuzzy), en las experiencias que realicen la enseñanza de este tipo de sistema de control, que busca abatir el problema de ajuste en los ciclos al punto de específico de estabilización provocados por la acción integral de los sistemas clásicos.



# CAPÍTULO II

## 2 CAPÍTULO II

### 2.1 SISTEMAS QNET-DCMCT DE QUANSER

#### 2.1.1 Introducción al sistema QNET-DCMCT

El DC Motor Control Trainer (DCMCT) de Quanser es una unidad versátil diseñada para enseñar y demostrar los fundamentos de servomotores para su control en una variedad de maneras. El sistema se puede configurar fácilmente para controlar la posición del motor.

En particular, el sistema puede usarse para enseñar los fundamentos del control PID, Difuso y otros más. Esto se hace usando un PC con control en tiempo real y el NI ELVIS II. El hardware del QNET-DCMCT que se describe a continuación. Un esquema de los componentes de hardware y las especificaciones son enumeran y se proporcionan consejos útiles de LabVIEW cuando se utilizan los VIs de QNET a la utilización, junto con una guía de solución de problemas que se puedan presentar con el equipo. (Karl-Johan, Apkarian, & Falcon, 2011)



Figura 2-1: Entrenador QNET de control de motores de CD (DCMCT)



## 2.1.2 Descripción del sistema

### 2.1.2.1 Componentes del QNET-DCMCT de Quanser

Los componentes que componen el entrenador de motores de corriente continua están etiquetados en las figuras 2-2 y 2-3. Y se describen en Tabla 2-1.

- Precaución: Asegúrese de que el sistema esté configurado como se indica en la *Guía de configuración de QNET (QNET setup Guide)*. El entrenador QNET-DCMCT es susceptible al deterioro de la protección y de sus partes si no se utiliza como se especificó.

Tabla 2-1 Nomenclatura de componentes del QNET-DCMCT

ID	DESCRIPCION
1	Motor de CD
2	Encoder de alta resolución
3	Carcasa del motor
4	Disco de inercia de carga (se descarta)
5	Conector PCI al NI ELVIS: Interface de módulo QNET con DAQ
6	Tarjeta Encoder/PWM QNET
7	Conector de 24V QNET
8	Fusible
9	LEDs +B, +15V, -15V, +5V

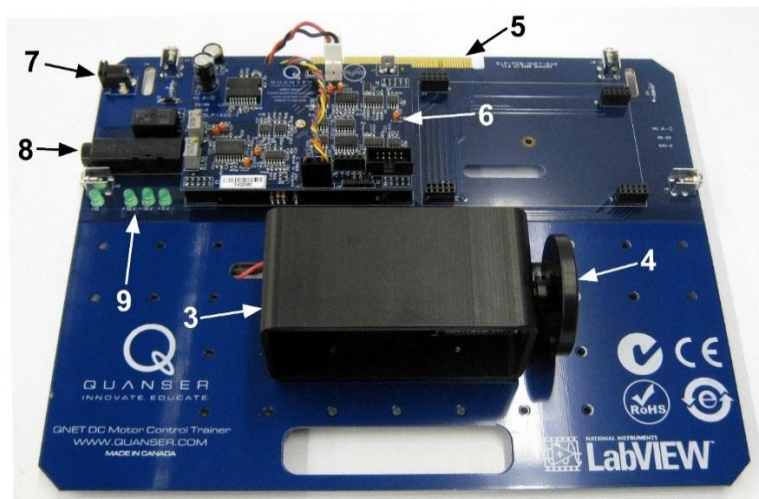


Figura 2-2: Disposición de componentes del QNET DCMCT

## 1. Motor de DC

El motor de corriente continua de 24 voltios con conmutación de grafito, alcanza una velocidad constante de 286 *rpm/V*. y cuenta con una resistencia de bobinado de 8.70 $\Omega$ .

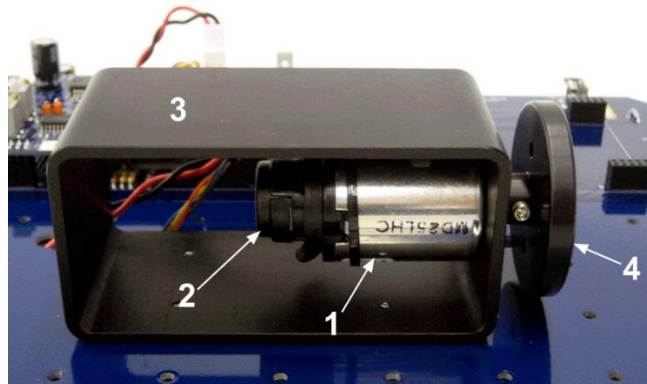


Figura 2-3: Componentes del motor QNET DC

## 2. Amplificador de potencia modulada para el generador de ancho de pulsos

Se utiliza un amplificador de potencia PWM para accionar el motor. La entrada al amplificador es la salida de Digital-Analógico (es decir, D/A) del canal # 0 en la DAQ. El voltaje de salida máximo del amplificador es 24 Volts, su máximo La corriente de pico es de 5 Amp. y la corriente continua máxima es de 4 Amp. La ganancia del amplificador es 2.3 V/V (volts por vuelta).

## 3. Medición de corriente analógica: Resistor de detección de corriente.

Una resistencia de carga en serie de 0.1  $\Omega$  está conectada a la salida del amplificador PWM. La señal se amplifica internamente para Resultan en una sensibilidad de 1.0 V/A. La señal de medición de corriente obtenida está disponible en el modo Analógico-Digital (es decir, A/D) del canal # 0 en la DAQ. Esta medida de corriente se puede utilizar para supervisar la corriente que corre en el motor.

## 4. Medición de posición digital: Encoder (Codificador óptico)

La medición de posición digital se obtiene utilizando un encoder (codificador óptico) de cuadratura de alta resolución. Este encoder está montado directamente en la parte trasera del motor. La medición del recuento del encoder está disponible en la entrada digital (es decir, DI) Canal # 0 de la DAQ.

### **5. Medición de velocidad analógica: Tacómetro**

Una señal analógica proporcional a la velocidad del motor está disponible en el canal de entrada Analógico-Digital (A/D) # 4 en el DAQ. Se deriva digitalmente de la señal del encoder en la tarjeta QNET DCMCT.

### **6. Fusible**

El amplificador de potencia QNET tiene un fusible a 250 V, 3 A.

### **7. Suministro de Potencia al QNET**

El módulo QNET-DCMCT tiene un conector de potencia a 24-Volts DC para energizar el amplificador PWM de la tarjeta. A este se le llama el Fuente de alimentación del QNET. El LED +B sobre la tarjeta QNET se vuelve verde brillante cuando el amplificador se energiza.

### 2.1.3 Esquemático del sistema.

Un esquema del sistema QNET-DCMCT interconectado con un dispositivo DAQ se muestra en la figura 2-4.

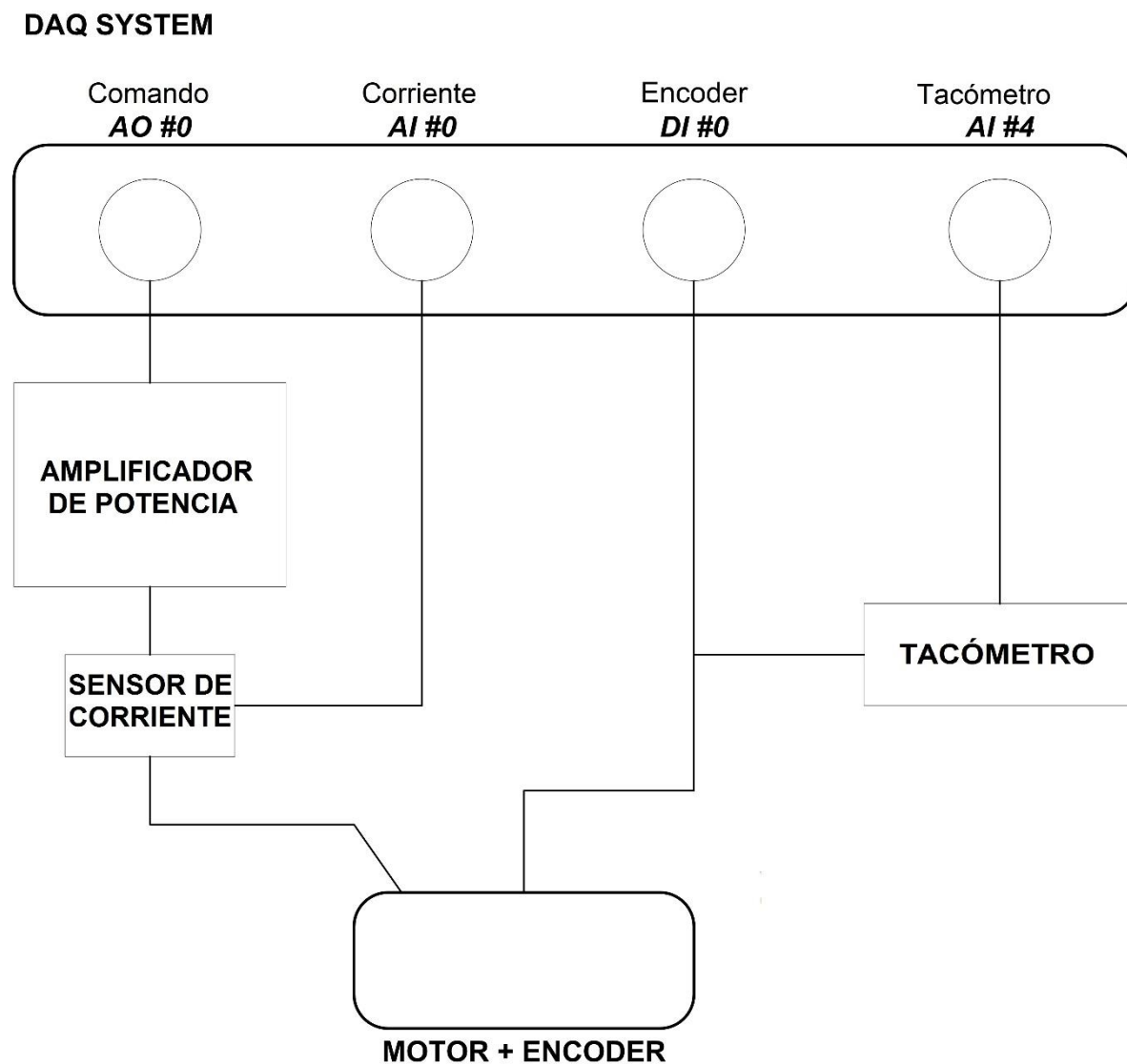


Figura 2-4: Esquema del sistema QNET-DCMCT

## 2.1.4 Especificaciones

Las especificaciones de los parámetros del modelo del sistema QNET-DCMCT están dadas en la Tabla 2-3.

Tabla 2-2: Parámetros del modelo y especificaciones del amplificador de potencia PWM

Símbolo	Descripción	Valor	Unidad
<b>Motor:</b>			
$R_a$	Resistencia de Armadura del motor	8.7	$\Omega$
$K_t$	Constante de par-motor	0.03334	$N \cdot m$
$K_a$		0.03334	$V/(rad/s)$
$J_a$		$1.80 \times 10^{-6}$	$kg \cdot m^2$
$M_l$		0.033	$kg$
$r_l$		0.0242	$m$
	<b>Amplificador modulado Ancho-Pulso</b>		
$V_{MAX}$	Voltaje de salida máximo amplificador PWM	24	V
	Corriente de salida máximo amplificador PWM	5	A
	Ganancia amplificador PWM	2.3	V/V

Las especificaciones sobre los detectores del sistema QNET-DCMCT están dadas en la Tabla 2-3.

Tabla 2-3: Especificaciones de los parámetros del sensor QNET-DCMCT

Descripción, Detección de Corriente:	Valor	Unidad
Calibración Corriente	1.0	A/V
Resistor detector Corriente	0.1	$\Omega$
<b>Encoder del Motor:</b>		
Encoder Contador de líneas	360	Líneas/rev.
Encoder resolución (en modo de cuadratura)	0.25	Grados/cont.
Encoder Tipo	TTL	
Encoder Señal	A,B	
<b>Tacómetro:</b>		
Calibración del tacómetro en la entrada QNET A/D	2987	$^{\circ}C/V$

## 2.1.5 Ambiental

Las condiciones ambientales de operación del entrenador de control de motor CD del DCMCT están dadas en la Tabla 2-4.

Tabla 2-4: Condiciones de funcionamiento ambiental del entrenador de control de motor CC de QNET

Descripción	Valor	Unidad
Temperatura de operación	15 a 35	°C
Humedad	20 a 90	%

- **Precaución:** Asegúrese de que la unidad funcione bajo las condiciones de temperatura y humedad indicadas en la tabla 4. De lo contrario, Puede haber algunos problemas con los resultados del control de movimiento. (Quanser, 2008)

## 2.1.6 Guía de preparación.

Como se ilustra en la Figura 2-5, las tarjetas QNET pueden conectarse fácilmente a un sistema NI ELVIS. La cual detalla el procedimiento de configuración para usar un QNET con un NI ELVIS II.

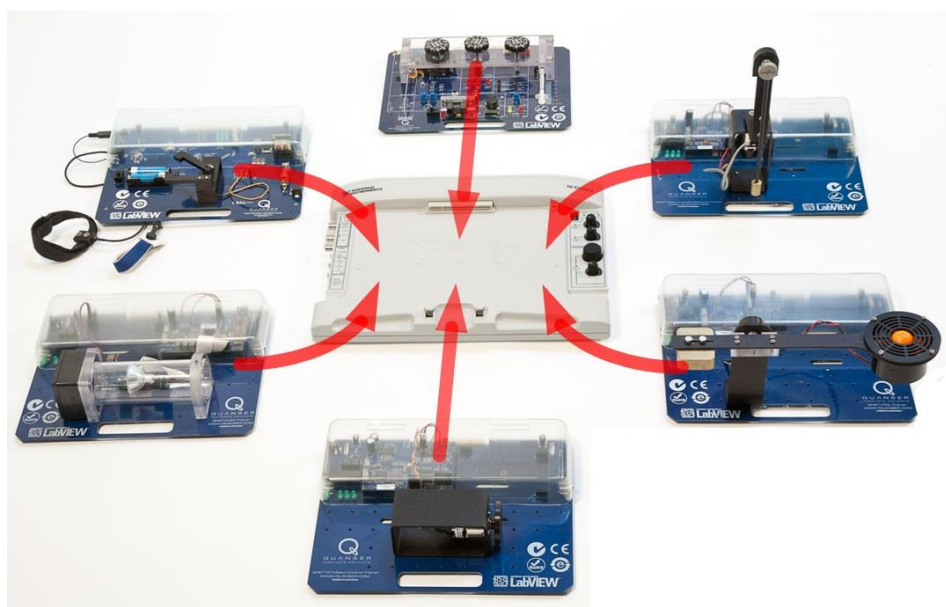


Figura 2-5: Conexión de un entrenador QNET

### 2.1.6.1 Configuración de QNET y NI ELVIS II

El procedimiento para instalar un módulo QNET (Quanser Engineering Trainer) en el NI ELVIS II se detalla de la siguiente manera. Los componentes NI ELVIS II utilizados en el procedimiento de instalación están localizados y marcados con un número de ID en la Figura 2-6, y se describe en la Tabla 2-5

#### 2.6.1.1 Componentes del NI-Elvis II

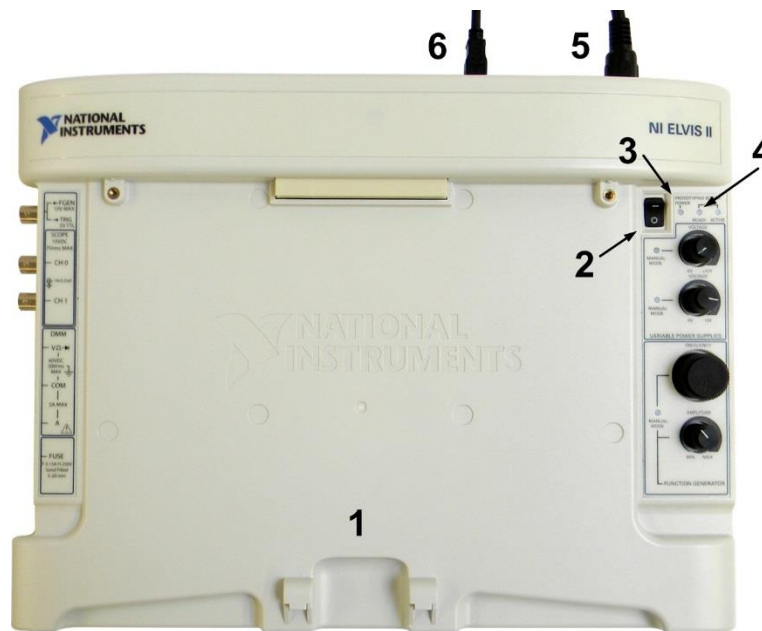


Figura 2-6: Componentes del NI-Elvis II

Configuración de QNET y NI ELVIS II

Tabla 2-5: Componentes del ELVIS II

ID	DESCRIPCION
1	NI ELVIS II
2	Interruptor de energía de la tarjeta y el Prototipo
3	LED de Energía
4	LED de Listo (Ready)
5	Cable de Potencia por el ELVIS II
6	Conexión USB entre PC-ELVIS II

### 2.1.7 Procedimiento de configuración de ELVIS II.

Siga estas instrucciones para configurar una placa QNET en un ELVIS II:

- Precaución: NO realice las siguientes conexiones mientras se suministra alimentación al hardware.
  - Precaución: La unidad se suministra con un cable con conexión a tierra que se utilizará únicamente con una toma es una característica de seguridad, no lo deshabilite.
1. Coloque la pequeña abertura en la parte frontal de la placa QNET sobre el soporte de montaje en el NI ELVIS II.
  2. Deslice el conector PCI del extremo del módulo QNET en el conector hembra del NI ELVIS II. Asegúrese que esté conectado correctamente.
  3. Conecte el cable de alimentación ELVIS II.
  4. Conecte el cable USB de ELVIS II al PC.
  5. Conecte el transformador QNET suministrado a la toma de alimentación QNET del módulo QNET.
  6. Encienda el NI ELVIS II encendiendo el interruptor de encendido del sistema en el panel trasero.
  7. Encienda el interruptor de alimentación de la placa prototipo, ID#2 como en la figura 7.
    - Precaución: Apague el interruptor de alimentación de la tarjeta de prototipos si el motor de CC comienza a girar.
    - Precaución: ¡Tenga mucho cuidado al alimentar el módulo QNET para evitar daños!

Los LED de encendido y encendido de la unidad NI ELVIS II deben encenderse como se ve en la figura 2-7



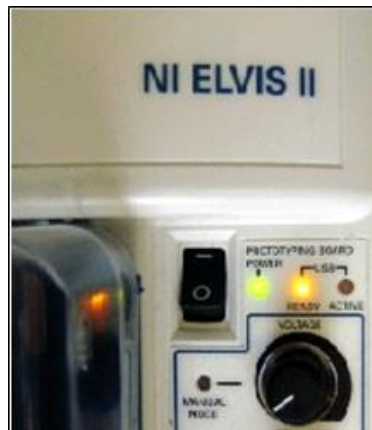


Figura 2-7: Los LEDs de Listo y Energía sobre el ELVIS II NI

Como se muestra en la figura 2-8, verifique que los LEDs +15V, -15V, +5V y +B del módulo QNET estén encendidos. Ellos indican que la tarjeta ha sido correctamente conectada a la unidad ELVIS.



Figura 2-8: Deben de estar encendidos los LEDs QNET

## 2.1.8 Consejos de LabVIEW QNET

### 2.1.8.1 Establecimiento de escalas para gráficas.

Esta sección describe un método manual para cambiar los ejes  $x$  o  $y$  en una gráfica de LabVIEW usando el *QNET\_DCMCT\_Swing\_Up\_Control* VI como un ejemplo. Ver pasos de abajo para reducir el rango del eje- $y$  de la gráfica Ángulo (deg) vista en la figura 2-9 para poder ver el trazo azul más de cerca.

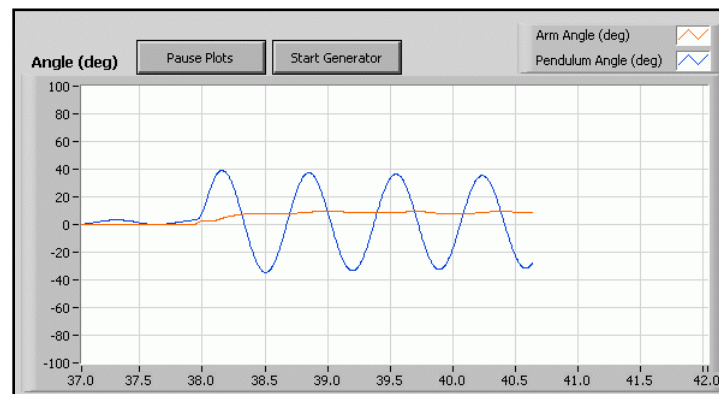


Figura 2-9: Las gráficas que se necesitan escalar

Como se ilustra en la figura 2-10, para disminuir el rango positivo de la gráfica de abajo hasta 40, oprima dos veces sobre '100' en el eje- $y$ , escriba '40', y presione ENTER.

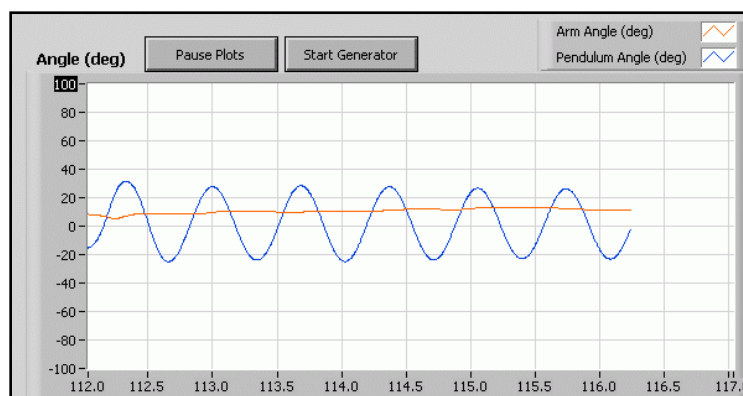


Figura 2-10: La gráfica que se necesita escalar

La gráfica resultante se muestra en la figura 2-11. El trazo azul es ahora más visible.

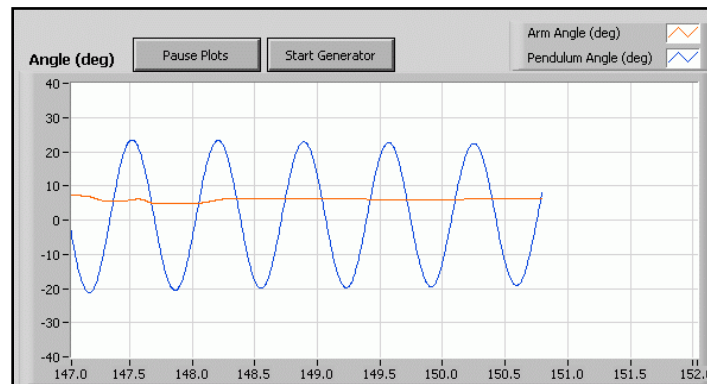


Figura 2-11: Eje-y de la gráfica se ha ajustado

De manera similar, el rango mínimo del eje-y se puede cambiar, así como también el rango del eje-x. Por ejemplo, para ver un rango de tiempo de 10 segundos en lugar de 5 segundos el rango del eje-x se puede cambiar de  $[0.0, 5.0]$  hasta  $[0.0, 10.0]$ . Sin embargo, cuando se cambie el eje-x, es decir, la escala del tiempo, se recomienda hacer lo siguiente:

- Detenga la gráfica o pare el VI y limpie el gráfico (oprime el botón derecho del ratón sobre la gráfica, seleccione Operación de Datos II Limpie el Gráfico).
- Aplice el mismo cambio de escala a las gráficas de la salida y la entrada. De otra manera, los datos trazados en cada gráfica no estarán sincronizadas entre ellas.

### 2.1.8.2 Salvar o guardar las respuestas en las gráficas.

Lea lo siguiente para salvar una respuesta graficada:

1. Oprima el botón derecho del ratón sobre la gráfica y seleccione Imagen de Exportación Simplificada, como se muestra en la figura 2-12.

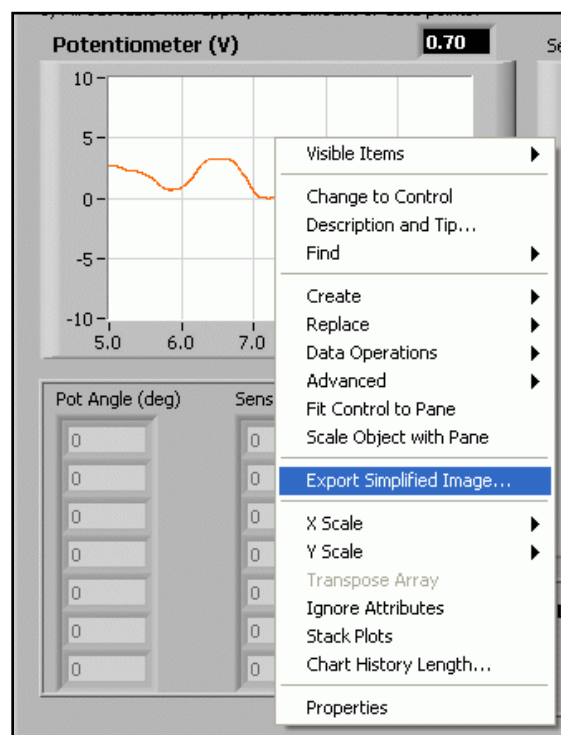


Figura 2-12: Oprima botón derecho ratón en la gráfica y elija Imagen Exportación Simplificada

2. La caja de dialogo mostrada en la figura 2-13 abre y proporciona varias opciones de exportar imagen. Una forma es exportar la imagen al portapapeles como un mapa de bits. Esto entonces se puede pegar en un software gráfico (es decir, MS Paint, Irfanview y salvado en un formato deseado en formato (gif, jpg o png).

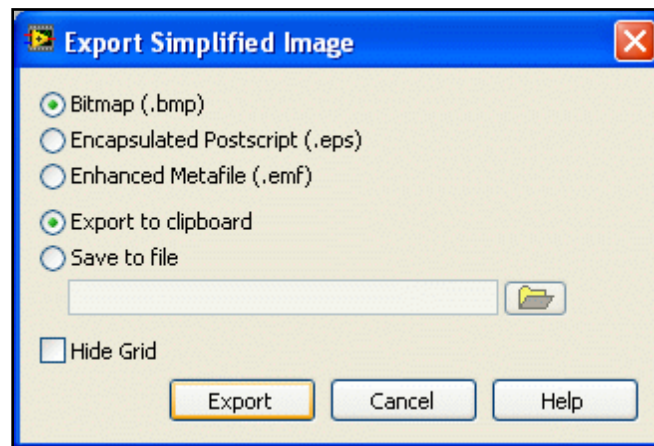


Figura 2-13: Caja de diálogo Exportar Imagen Simplificada

3. La imagen resultante que se salvó se muestra en la figura 2-14.

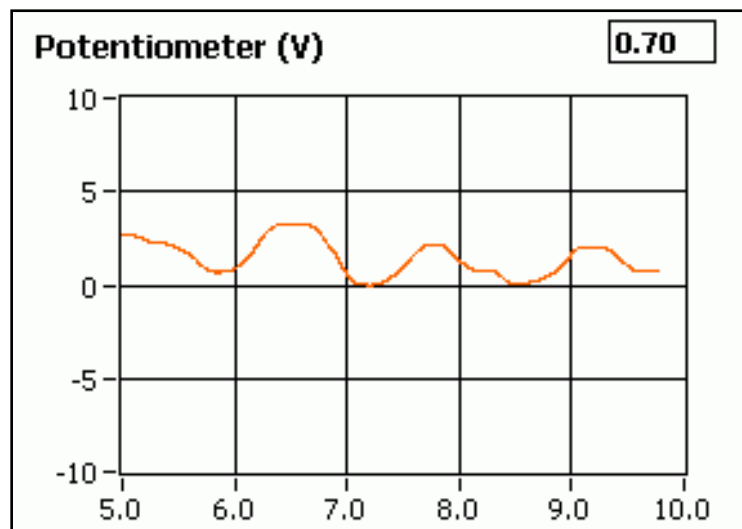


Figura 2-14: Imagen resultante

4. La gráfica se puede salvar ya sea que VI esté corriendo o no. Sin embargo, típicamente es muy fácil parar el VI cuando la respuesta deseada se recopila y entonces se exporta la imagen como se ilustró arriba.

## 2.1.9 Búsqueda de Fallos

### 2.1.9.1 Problemas generales del software

**Pregunta 1.** ¿Cuándo se trata de abrir a un QNET VI, se dice que existen algunos faltantes Vis y ellos tienen un “CD” o “Sim” en el nombre?

**Respuesta=** El módulo de diseño y simulación de LabVIEW no está instalado.

**Pregunta 2.** ¿Cuándo se abre un QNET VI un mensaje le indicará que un VI con el nombre de ELVIS no se puede encontrar?

**Respuesta=**

- ELVIS I: Los VIs QNET utilizan controladores instalados desde el CD ELVIS 3.0 o posterior. Asegúrate de que está instalado. Si la carpeta " \ National Instruments \ NI ELVIS 3.0 " no existe, entonces no está instalado (Disponible para descargar en [www.ni.com](http://www.ni.com)).
- ELVIS II: El QNET Vis usa los drivers ELVISmx. Asegúrese que ud. instaló los contenidos del CD del ELVIS II antes de intentar abrir cualquier QNET Vis (disponible para descargar en [www.ni.com](http://www.ni.com)).

### 2.1.9.2 Problemas generales con el Hardware

**Pregunta 1.** ¿Ninguno de los LEDs en la tarjeta QNET brilla?

**Respuesta=** Asegúrese que los interruptores System Power, que se localiza sobre la parte trasera de las unidades ELVIS II, y Prototyping Board Power, localizado en la esquina superior derecha del ELVIS II, estén en ON. Vea la guía del ajuste del QNET por más información.

**Pregunta 2.** ¿En la tarjeta del QNET, los LEDs de +15V, -15V y +5V estén brillando en verde pero el LED +B no lo esté?

**Respuesta=** Asegúrese de que el conector QNET de la energía en la tarjeta QNET esté conectado con el cable de energía QNET suministrado. Vea la guía del ajuste del QNET por más información.

**Pregunta 3.** ¿Al menos uno de los LEDs de +B, +15V, -15V y +5V en la tarjeta QNET no brille?

**Respuesta=**

- Vea a Q2 si solo el +B no brilla.

**Pregunta 4.** ¿El LED de Listo (Ready) en el ELVIS II no brilla?

**Respuesta=**

1. Vaya al procedimiento de ajuste del ELVIS II indicado en la Guía del Ajuste del QNET.
2. Una vez concluido, empiece el software de *Measurement & Automation Explorer*.
3. Como se ilustra en la Figura 2-15 expanda los artículos el *Devices and interfaces* and *NI-DAWmx* y seleccione el Dispositivo NI ELVIS II.
4. Como se muestra en la Figura 2-15 oprima el botón del ratón sobre el botón *Reset Device*.
5. Una vez restablecido correctamente, oprima el botón derecho sobre el botón *Self-Test*.
6. Si pasó la prueba, restablezca el ELVIS II (es decir, abra el interruptor *Prototyping Board* y el interruptor *System Power* y ciérrelos de nuevo). El LED *Ready* en el ELVIS II deberá de brillar.

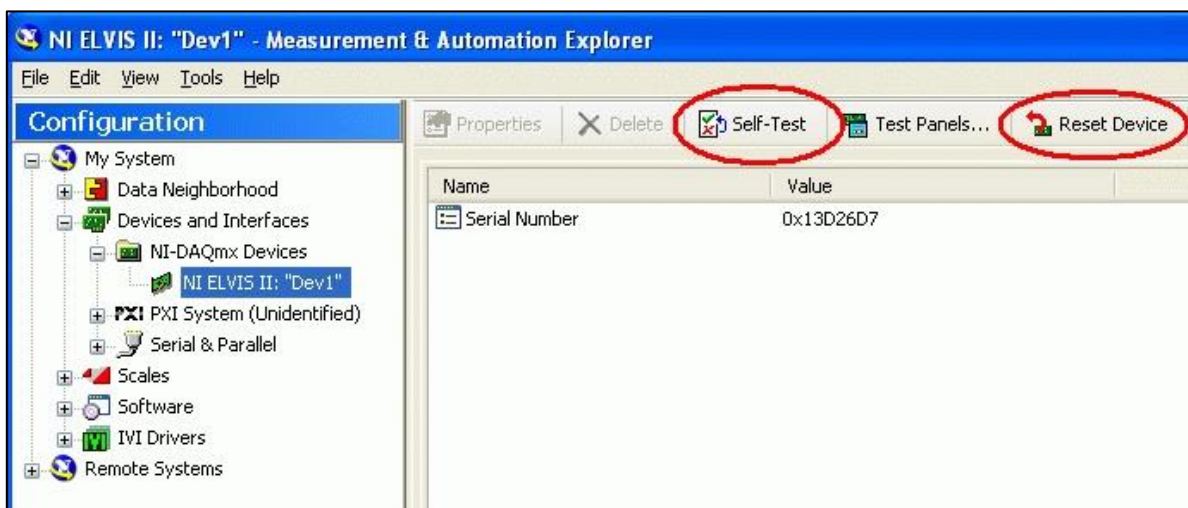


Figura 2-15: Restableciendo y actuando el auto-test en el ELVIS II

### 2.1.9.3 Problemas en el DCMCTM

Cuando abre un QNET-DCMCT (VI), El interruptor de la placa de prototipos no está encendido. El LED al lado del interruptor debe ser de color verde brillante. Por favor Revise la guía de QNET Setup.



Figura 2-16: Alcances en la medición del servomotor QNET-DCMCT en Vi's

El motor no se mueve cuando ejecuto el VI, asegúrese de que el cable de alimentación QNET esté conectado. Los cuatro LED + B, + 15V, -15V, y + 5V en el QNET Bordo debe ser de color verde brillante.

### 2.1.10 Apoyo técnico del equipo de Quanser.

Para obtener el apoyo de Quanser, vaya a <http://www.quanser.com/> y oprima el botón del ratón en el link Tech Support. Llene la forma con toda la información solicitada del software y hardware así como también una descripción del problema encontrado. También, asegúrese de que su número del teléfono y dirección de correo electrónico estén incluidos. Envíe la forma y una persona del apoyo técnico lo contactará. (Quanser, 2008)





# CAPITULO III

### 3 CAPITULO III

#### 3.1 MODELADO MATEMÁTICO DEL SISTEMA QNET-DCMCT A CONTROLAR

Como sabemos el DC Motor Control Trainer (QNET-DCMCT) es un sistema didáctico de Quanser para la enseñanza del control de motores de corriente directa. La variable controlada o salida del sistema puede ser la velocidad del disco de carga del motor o el ángulo de posición del mismo, es decir visto como un sistema servomotor o como un sistema regulador. La señal de control es el voltaje de entrada al amplificador que alimenta el voltaje de armadura del motor y el sistema se puede perturbar manualmente o mediante los diagramas de simulación y control.

El modelo matemático del sistema se determinara de manera analítica y de manera experimental utilizando el sistema DAQ NI ELVIS II, Labview y Matlab.

##### 3.1.1 Modelado analítico del sistema

Para la obtención del modelo matemático de manera analítica se utilizan las leyes físicas que gobiernan el comportamiento dinámico del mismo a esta estrategia también se le conoce como modelado de caja blanca. Consideremos el siguiente diagrama equivalente del motor de CD controlado por armadura mostrado en la figura 3.3

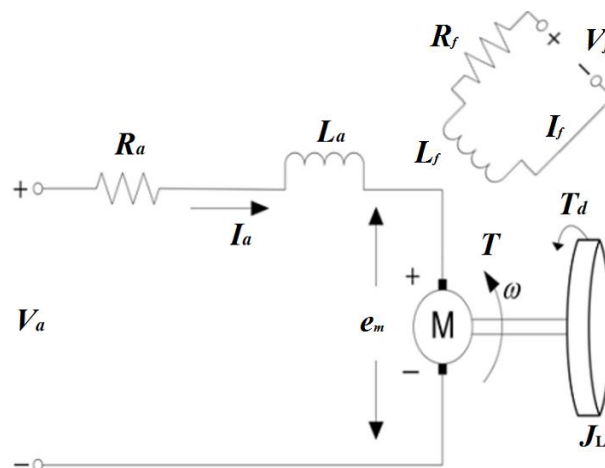


Figura 3-1: Diagrama equivalente de un motor de CD

Tabla 3-1: Parámetros obtenidos de placa del sistema QNET-DCMCT de Quanser.

ID	DESCRIPCION		
1	Voltaje de armadura y de Campo	$V_a, V_f$	
2	Corriente de Armadura y de Campo	$i_a, i_f$	
3	Resistencia de Armadura y de Campo	$R_a, R_f$	$3.30 \Omega$
4	Inductancia de Armadura y de Campo	$L_a, L_f$	
5	Fuerza Contra Electromotriz	$e_m$	
6	Par producido por el servomotor	$T$	
7	Velocidad Angular	$\omega(t)$	$d\theta(t)/dt$
8	Inercia del Servomotor	$J_a$	$9.64 \times 10^{-6} \text{ kgm}^2$
9	Masa de Carga	$m$	$0.033 \text{ kg}$
10	Radio del disco de la carga	$r$	$0.0242 \text{ m}$
11	Inercia de la Carga	$J_L$	$\frac{1}{2}mr^2 = \frac{1}{2}(0.033)(0.0242)^2$ $= 9.663 \times 10^{-6} \text{ kgm}^2$

Aplicando las leyes de Kirchoff y de Newton, tenemos lo siguiente: (Kuo, 1996)

$$V_{R_a} + V_{L_a} + e_m - V_a = 0 \quad (3.1)$$

$$i_a R_a + L_a \frac{di_a}{dt} + e_m = V_a \quad (3.2)$$

- Utilizando la transformada de Laplace queda la función

$$i_a(s)R_a + L_a s i_a(s) + e_m(s) = V_a(s) \quad (3.3)$$

- Esfuerzos producidos por el par mecánico son la sumatoria de la carga del volante de inercia, más el peso de la perturbación  $J = J_L + J_a$ .

$$J(s) \frac{d\omega(t)}{dt} + B\omega(t) = T \quad (3.4)$$

- Haciendo la Transformada de Laplace

$$T(s) = (J(s) + B)\Omega(s) \quad (3.5)$$

Como la armadura está girando en un campo magnético su voltaje es proporcional a la velocidad, donde  $k_1$  es la constante de *FEM* y  $k_2$  es la constante del servomotor. Se relaciona de la siguiente manera (Kuo, 1996).

$$e_m = k_1 \Omega(s) = k_1 \frac{d\theta}{dt} \quad (3.6)$$

- De mismo modo aplicando la transformada de Laplace en ello tenemos:

$$e_m(s) = k_1 \Omega(s) = k_1 s \theta(s) \quad (3.7)$$

- El par creado por el servomotor es proporcional a la corriente de armadura  $T(s) = k_2 i_a$ .

$$T(s) = k_2 i_a(s) \quad (3.8)$$

Las variables de entrada y salida para el sistema las variables a controlar, para el caso de entrada es la velocidad  $\omega(t)$  y de la posición  $\theta(t)$ , en tanto para la variable de salida corresponde al voltaje de armadura  $V_a(t)$  proporcionado al motor para la obtención del modelo.

- Sustituir la ecuación (3.7) en la (3.3)

$$i_a(s)R_a + L_a s i_a(s) + k_1 \Omega(s) = V_a(s) \quad (3.9)$$

- Sustituir la ecuación (3.8) en (3.5)

$$k_2 i_a(s) = (J(s) + B)\Omega(s) \quad (3.10)$$

- Despejando  $i_a$  de (3.10)

$$i_a(s) = \frac{1}{k_2} (J(s) + B)\Omega(s) \quad (3.11)$$

- Sustituyendo (3.11) en (3.9)

$$R_a \left[ \frac{1}{k_2} (J(s) + B)\Omega(s) \right] + L_a s \left[ \frac{1}{k_2} (J(s) + B)\Omega(s) \right] + k_1 \Omega(s) = V_a(s) \quad (3.12)$$

De la ecuación (3.12) se dice que la inductancia de la armadura ( $L_a$ ) y el amortiguador ( $B$ ) es pequeña en comparación con la resistencia con la resistencia de armadura ( $R_a$ ) entonces  $L_a \approx 0$  y  $B \approx 0$  por lo tanto:

- La ecuación se convierte en:

$$R_a \left[ \frac{1}{k_2} (J(s) + B) \Omega(s) \right] + k_1 \Omega(s) = V_a(s) \quad (3.13)$$

- Obteniendo la función de transferencia.

$$\left( \frac{R_a J}{k_2} s + k_1 \right) \Omega(s) = V_a(s) \quad (3.14)$$

$$\frac{\Omega(s)}{V_a} = \frac{1}{\left[ \frac{R_a J + k_1 k_2}{k_2} \right] \frac{1}{R_a J}} \quad (3.15)$$

- Modelo Analítico en términos de Velocidad para el motor de CD.

$$\frac{\Omega(s)}{V_a(s)} = \frac{k_2 / R_a J}{s + \frac{1}{J} \left( \frac{k_1 k_2}{R_a} \right)} \quad (3.16)$$

$$\frac{\Omega(s)}{V_a(s)} = \frac{439.5612}{s + 12.3077} \quad (3.17)$$

$$\frac{\Omega(s)}{V_a(s)} = \frac{K}{\tau s + 1} = \frac{35.7143}{0.0812s + 1} \quad (3.18)$$

Nota: Para poder expresar el modelo matemático en término de la posición, es necesario integrar el modelo en función de la velocidad angular esto es:

$$\frac{\theta(s)}{V_a(s)} = \frac{K}{s(\tau s + 1)}$$

Con este modelo de orden reducido del motor se puede realizar el análisis del sistema en lazo cerrado para calcular las ganancias del controlador diseñado en capítulo siguiente.

### 3.1.2 Modelado experimental o identificación.

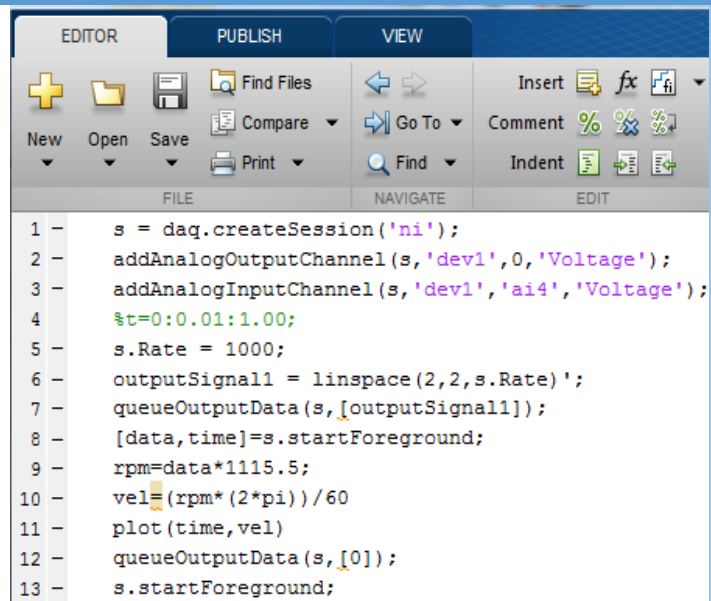
El modelo experimental o modelo de caja negra en el cual se somete el sistema a pruebas utilizando una señal escalón como señal de entrada y así observa el comportamiento de las salidas, en base a ello definir las ecuaciones que determinen este mismo comportamiento y establecer un modelo al sistema.

Para obtener una muestra de los datos del comportamiento del sistema del motor DCMCT de Quanser, Estableceremos una conexión con Matlab utilizando la paquetería de adquisición de datos que reconoce los dispositivos de National Instruments conectado, esto es usando *nidaqmx*. Se diseña en un archivo tipo .m usando el editor de Matlab como muestra la figura 3-4, el cual logra dar una señal de entrada tipo escalón de 2 volts al motor DCMCT mediante la tarjeta NI ELVIS II registrado como dispositivo 1 “*dev1*”. Estableciendo el tiempo de muestreo “*t*” en 1 segundos y con una muestra “simple time” de 0.01 seg.

#### Extracción de la respuesta transitoria al escalón utilizando Matlab

Utilizar los comandos de:  
*daq.createSession*

- Activar el canal de la *daq* Elvis II (ai4)
- Dar un tiempo con *s.Rate* de 1000 para 1seg.
- Obtener los datos de la curva de respuesta por del encoder de Quanser de DCMCT usado la variable “*OutputSignal1*” y una señal de 2 volts en una tiempo de 1seg. usando el comando “*linspace*”
- Almacenar los datos con el comando “*queueOutputData*”.



```

1 - s = daq.createSession('ni');
2 - addAnalogOutputChannel(s, 'dev1', 0, 'Voltage');
3 - addAnalogInputChannel(s, 'dev1', 'ai4', 'Voltage');
4 - %t=0:0.01:1.00;
5 - s.Rate = 1000;
6 - outputSignal1 = linspace(2,2,s.Rate)';
7 - queueOutputData(s,[outputSignal1]);
8 - [data,time]=s.startForeground;
9 - rpm=data*1115.5;
10 - vel=(rpm*(2*pi))/60
11 - plot(time,vel)
12 - queueOutputData(s,[0]);
13 - s.startForeground;
  
```

Figura 3-2: Archivo .m de Matlab para obtener la curva de respuesta

La representación de la curva de respuesta del QNET-DCMCT a través de la adquisición de datos de del encoder de Quanser con la tarjeta ELVIS II en Matlab, usando la paquetería de Matlab para establecer comunicación con la tarjetas de adquisición de *national instruments* usamos la "nidaqmx" la podemos observar en la figura 3-2.

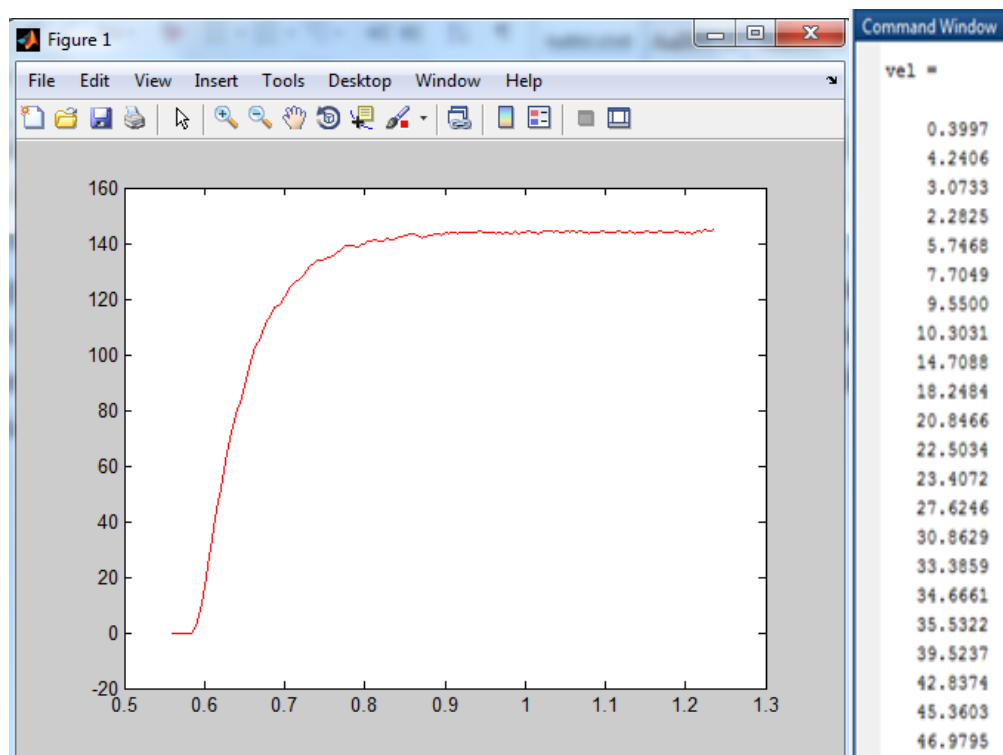


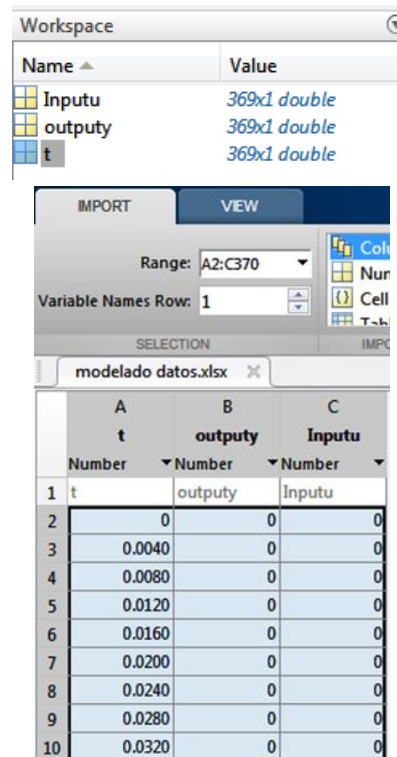
Figura 3-3: Gráfico de la respuesta transitoria al sistema

### 3.1.3 Modelado del sistema utilizando la herramienta *System Identification* de Matlab

Como primera medida del proceso para la obtención del modelo matemático del sistema QNET-DCMCT, tenemos que partir con el experimento realizado anteriormente para conocer el comportamiento de la señal medida del sistema. Se deben establecer las variables a importación desde el "*current folder*" que se encuentren en el *Workspace* de Matlab utilizando la aplicación de identificación de sistemas que cuenta Matlab llamada *System Identification*, en el cual podemos obtener el modelo matemático equivalente

descrito por la prueba de reacción antes realizada a la señal escalón aplicada al sistema QNET-DCMCT de Quanser como se muestra en la figura 3-3.

Se deben de exportar los datos a un archivo de Excel o de base de datos hacia el Workspace de Matlab para tener acceso a utilizar *System Identification* definiendo las variables como “*Inputu*” y “*outputy*” con la variable de tiempo de muestreo estándar *t*.



	A t Number	B outputy Number	C Inputu Number
1	t	outputy	Inputu
2	0	0	0
3	0.0040	0	0
4	0.0080	0	0
5	0.0120	0	0
6	0.0160	0	0
7	0.0200	0	0
8	0.0240	0	0
9	0.0280	0	0
10	0.0320	0	0

Figura 3-4: Importar las variables de entrada y salida

Para abrir la herramienta de *Matlab System identification*, se puede llamar desde la aplicación de Workspace escribiendo *ident* o bien desde el icono de App de Matlab.

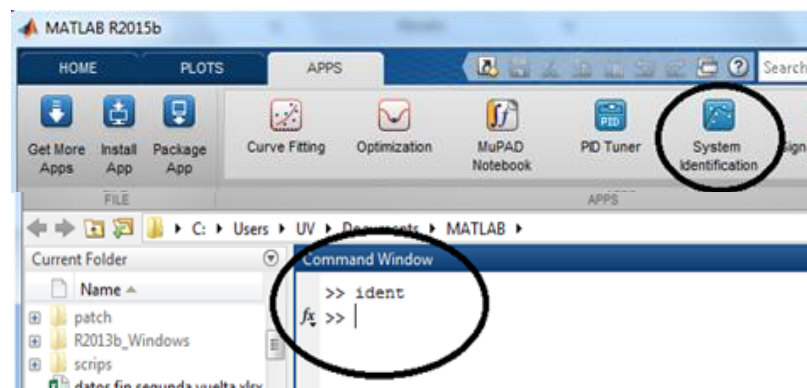




Figura 3-5: Aplicación System Identification de Matlab

Al desplegar la ventana principal del Toolbox, aparecen dos tablas que nos representan los conjuntos de datos de entrada y salida de la señal adquirida en la importación y los modelos o conjuntos precargados para su visualización; además existen otras herramientas que permiten realizar diversas aplicaciones en los sistemas o modelos complementarios a este Toolbox.

La importación de los datos que nos de la generación del modelo del sistema se puede realizar al utilizar el Workspace o directamente de un archivo de almacenamiento de los datos.

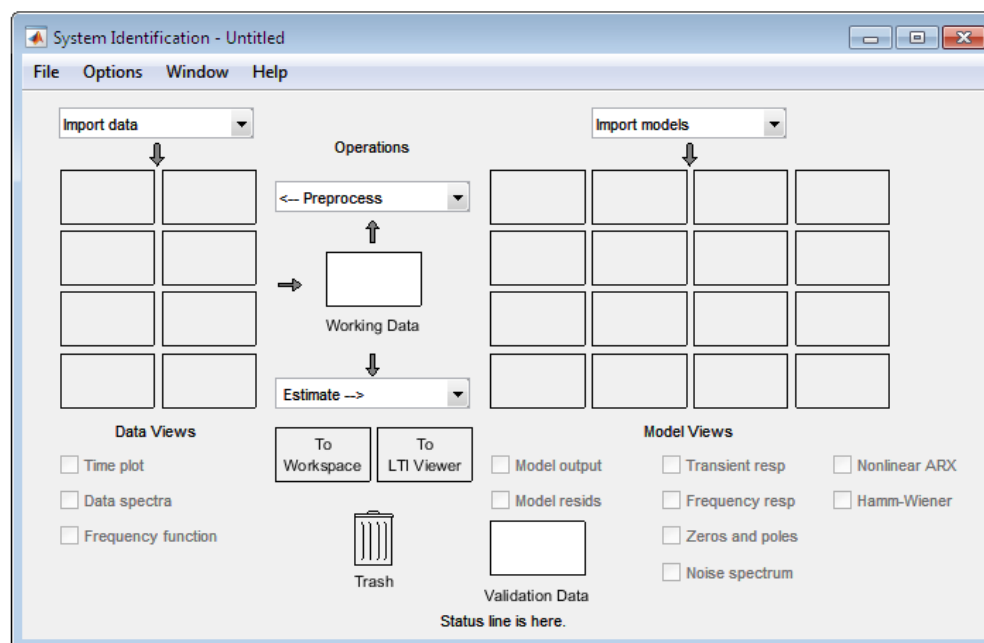


Figura 3-6: Ventana principal de System Identification Toolbox de Matlab.

La implementación del diseño del importar el modelo matemático a partir de los datos adquiridos por el encoder de Quanser, se implementa bajo la operación de importación de datos **Import data**. Se debe de ingresar los datos e incorporarlos al sistema a identificar, en este caso son datos tomados en el dominio del tiempo y están almacenados en el Workspace de Matlab.

Al usar “*Import data*” debemos seleccionar la opción de “*time domain data*” que nos logre permitir ver con la opción “*Operation*” el sistema a identificar y poder establecer y definir grafica en el sistema utilizando “*Data Views*” como se muestra en la figura 3-7.

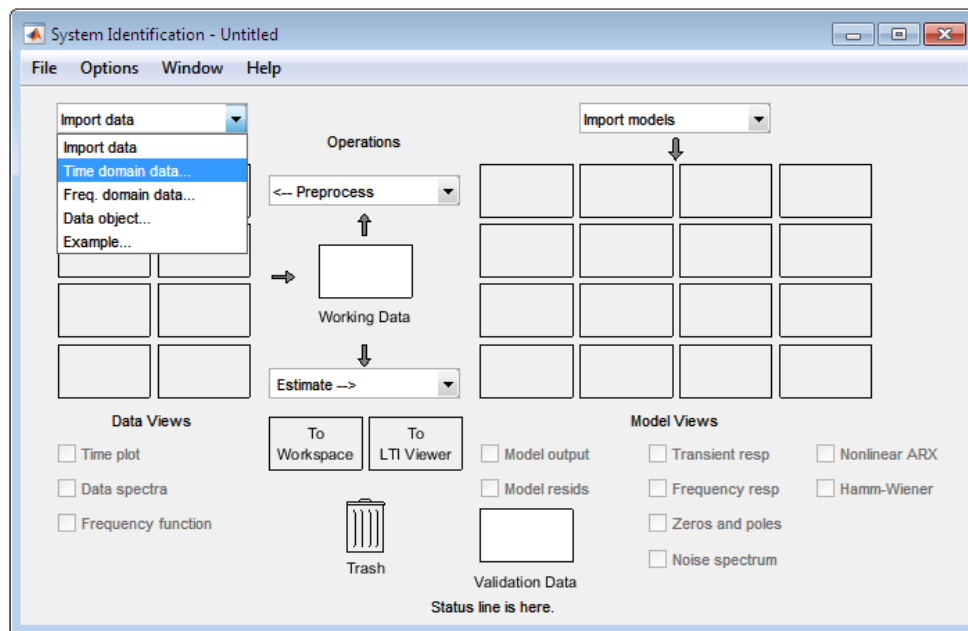


Figura 3-7: Importación de modelos

Se importa los datos con los nombres de las variables previamente guardadas y especificando el nombre, el tiempo de inicio y el intervalo de muestreo, dando ***click*** en “***import***” inmediatamente estos quedaran en la tabla de datos. Es importante tener en cuenta que los vectores deben guardarse como columna si no es el caso aparecerá una ventana que le preguntara si desea transponer, en tal caso lo más conveniente es aceptar.

La determinación de las variables en el dominio del tiempo con “*Time-Domain signals*” se debe realizar bajo el procesamiento previo de obtención de los datos para su análisis de identificación de sistemas. Esto se le llama el pre-procesamiento de dato de un sistema y consiste en tener un registro de entrada previamente optimizado de una señal, donde se puede eliminar un posible desvío en la información y de tal manera aumente la calidad del experimento en la respuesta obtenida. En el caso de encontrar una baja calidad de respuesta en el sistema en base los datos, se deberá realizar otra prueba de señal que arroje mayor número de muestras.

Mediante el la importación de datos, llevamos la variables al sistema llama *mydata*, dándole una tiempo de muestreo acorde a la separación de muestras en el archivo de datos generado por la adquisición e Matlab. La graficar de entrada y salida del modelo en su respuesta a la señal de entrada y la señal de salida respecto al tiempo. Esto en *y1* y *u1* propiamente como se muestra en la figura 3-10.

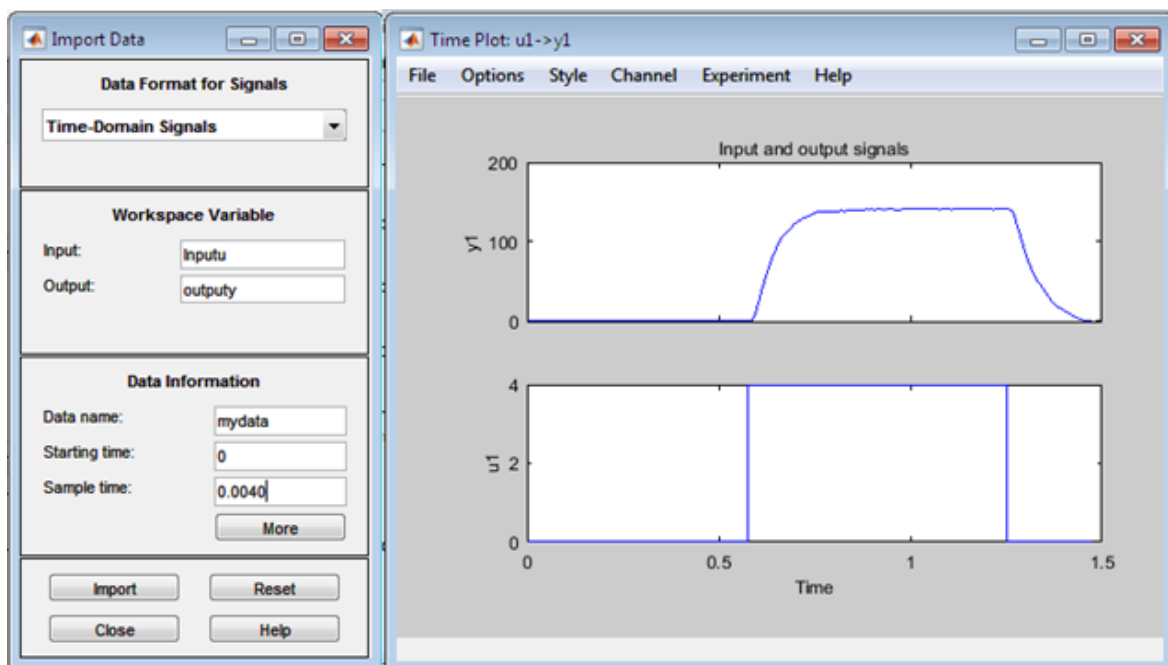


Figura 3-8: Modelo de salida

Como se observa el modelo propuesto no se ajusta a los datos de entrada-salida pues la combinación entre la señal de entrada y la respuesta al impulso del sistema, dan como resultado una secuencia de salida muy diferente como se ve las curvas graficadas de la figura 3-10. Para corregir el modelo, se selecciona el menú de “*estimate*”, eligiendo “*process models*”.

Se muestra la opción “*process model*” en el cual se debe especificar el número de polos para el modelo de función de transferencia, se puede agregar o no el mismo número de ceros marcando el recuadro indicado así como retardos (de la misma forma), la forma de la función de transferencia se muestra a la izquierda; si existen parámetros de los mismo, en “*Initial Guess*” se puede ingresar el valor inicial para cada parámetro a estimar o se puede dejar automáticos (auto) que es lo adecuado para este caso como en la figura 3-11 y 3-12.

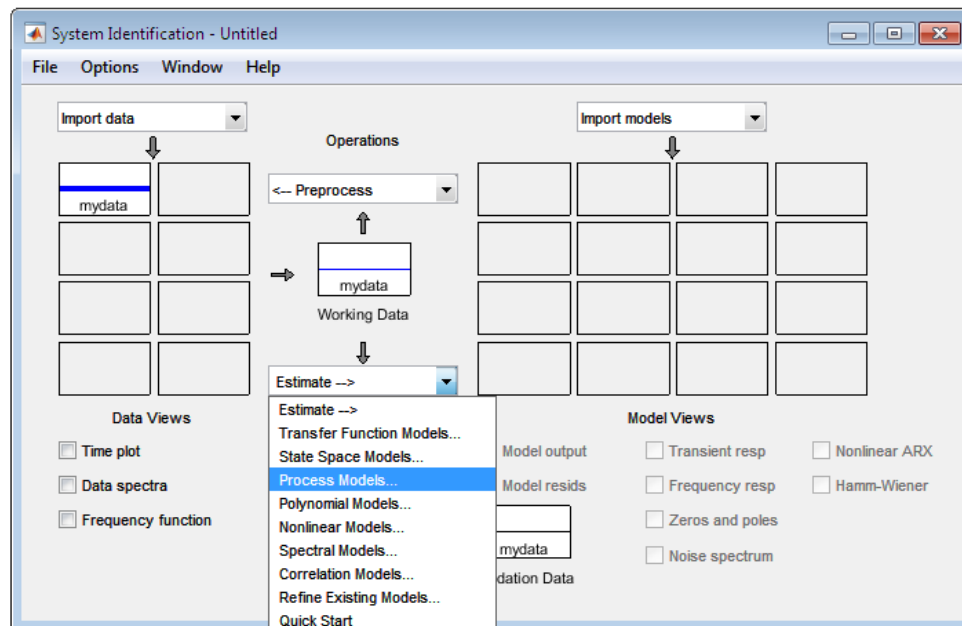


Figura 3-9: Opciones de procesamiento

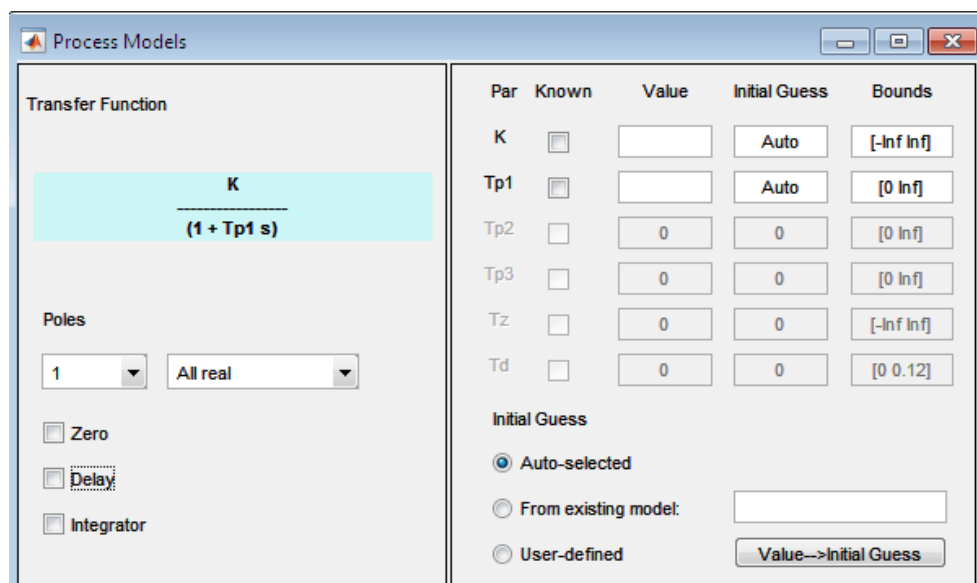


Figura 3-10: Selección de la función de transferencia en “*Process Models*”

Al realizar la estimación y obtener los parámetros de  $K$  y  $t_p = \tau$  para el sistema de aproximación de primer orden del motor DCMCT de Quanser, podemos observar la señal de respuesta transitoria del sistema respecto al tiempo de simulación.

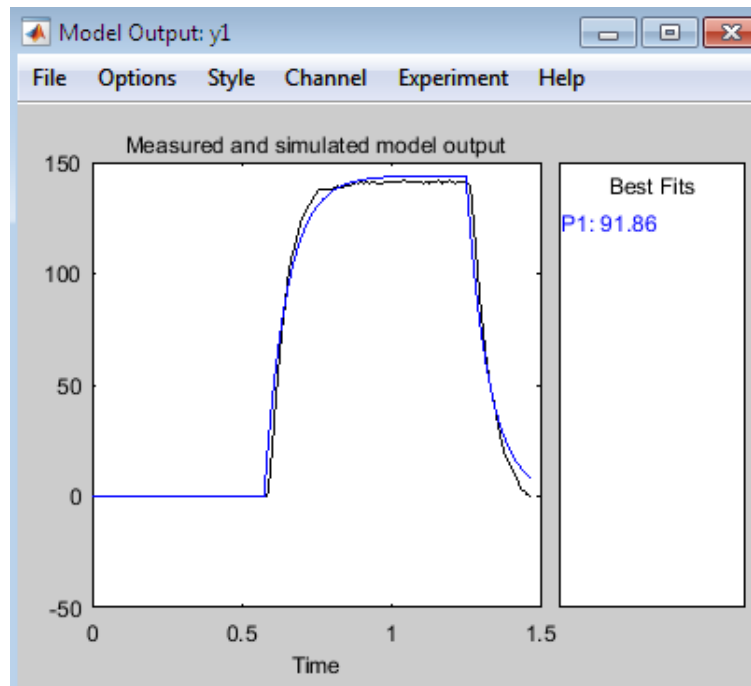


Figura 3-11: Respuesta transitoria de la planta “*Model Output*”

Una vez establecida la respuesta que se identifican del sistema, se procederá a exportar la función de transferencia modelada al Workspace para su visualización para su estudio en base a los datos obtenidos de manera experimental en el DCMCT de Quanser.

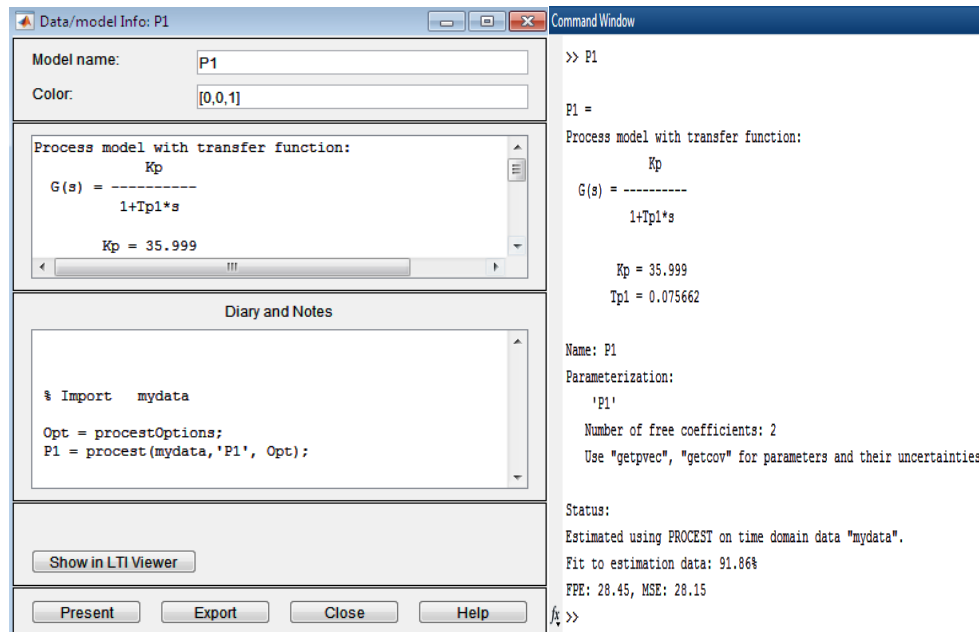


Figura 3-12: Importar el modelo “P1” con los datos obtenidos a la prueba analítica

También del mismo modo una vez verificado el modelo bajo una propuesta de partir del modelo no paramétrico (de manera numérica), la función de transferencia se observa que satisfaga la respuesta estimada por el Toolbox. Este se puede realizar a partir de varios intentos de ejecutar el archivo de comando en Matlab.

Queda establecido el modelo de la función de transferencia de Motor Control Trainer (DCMCT) de Quanser utilizando el System Identification de Matlab.

$$\frac{Y(s)}{U(s)} = \frac{\Omega(s)}{V_a(s)} = \frac{35.999}{0.07566s+1} \quad (3.19)$$

### 3.1.4 Modelado usando QNET-DCMCT Modeling de LabVIEW

#### Prueba a la señal escalón aplicado en LabVIEW

La prueba a la señal escalón como sabemos está basada en la respuesta a una señal aplicada a un sistema estable. Al aplicar una entrada escalonada al sistema y registrar su respuesta podemos delimitar su comportamiento. Considérese el sistema la función de transferencia de primera orden ya conocida del sistema de la ecuación (3.18) para el dispositivo QNET-DCMCT.

La respuesta al escalón como ejemplo es la que se muestran las figuras 3-15 y 3-16.

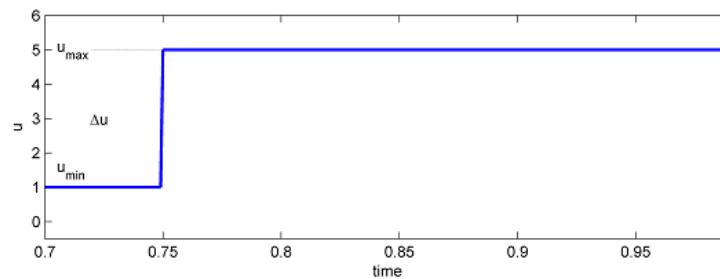


Figura 3-13: Señal escalón aplicada al sistema

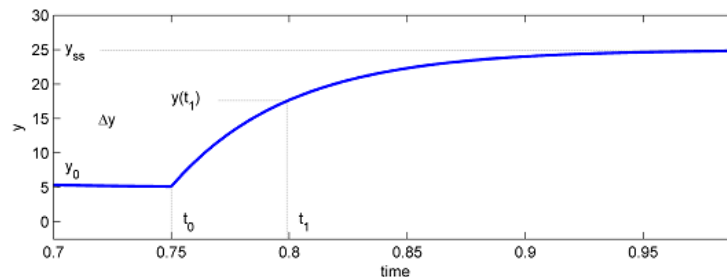


Figura 3-14: Señal de respuesta en la salida del sistema

La señal de entrada al aplicar el escalón comienza en el instante  $t_0$ . La señal de entrada tiene un valor mínimo de  $u_{min}$  y un valor máximo de  $u_{max}$ , la señal de salida resultante está inicialmente en  $y_0$ . Una vez que se aplica el paso, la salida intenta seguirlo y eventualmente se establece en su valor estacionario  $y_{ss}$ . A partir de las señales de salida y entrada, la ganancia de estado estacionario es:

$$K = \frac{\Delta y}{\Delta u} \quad (3.19)$$

Donde  $\Delta y = y_{ss} - y_0$  Y  $\Delta u = u_{max} - u_{min}$  Para encontrar la constante de tiempo del modelo, el valor de  $\tau$  lo podemos calcular primero dónde se supone que la salida está en la constante de tiempo de:

$$y(t_1) = 0.632y_{ss} + y_0 \quad (3.20)$$

Entonces, podemos determinar el tiempo  $t_1$  que corresponde a  $y(t_1)$  a partir de los datos de respuesta de la figura 3-16. Puede obtenerse que el valor en el tiempo  $t_1$  que es igual a:

$$t_1 = t_0 + \tau \quad (3.21)$$

A partir de esto, la constante de  $\tau$  del modelo se puede definir como:

$$\tau = t_1 - t_0 \quad (3.22)$$

Los anteriores métodos aplicados para el modelado pueden validarse mediante el comparativo de las curvas obtenidas con la curva respuesta real del DCMCT y la curva simulada, para ello una forma es utilizando el vi “*QNET-DCMCT Modeling*”.

Se aplica una voltaje por un generador de señal que alimenta el QNET- DCMCT y del mismo modo al modelo parametrizado, de tal manera que tanto el valor simulado como el medido de respuesta se puede pueden visualizar en la misma gráfica que nos permita tener una comparativa de aproximación. El modelo se puede ajustar tanto para la velocidad del motor medida por el encoder por medio del generador de señal como para los parámetros de modelado. (Karl-Johan, Apkarian, & Falcon, 2011)

La aplicación de una tensión al motor de corriente continua y el examen de su velocidad angular se investiga las curvas de respuesta en el laboratorio. El modelo de simulación “*Model Parameters*” se ejecuta en paralelo con el sistema actual para permitir el ajuste y la validación del modelo. Podremos visualizar gráficamente la señal de entrada del sistema y tomar las mediciones establecer una comparación entre la señal simulada con la señal de respuesta real que proporciona el sistema mediante la adquisición de los datos del encoder y la NI-ELVIS.



## Realización del experimento:

Asegúrese de que el VI de modelado de DCCM de QNET está abierto y configurado como se describe en el capítulo 2. Debe ejecutar el archivo *QNET\_DCMCT\_Modeling.vi*. El motor de CD debe comenzar a girar y los ámbitos en el VI deben aparecer similitudes.

En la sección de generador de señal se aplica 4 V.

- Amplitud = 2.0 V
- Frecuencia = 0.40 Hz
- Offset = 2 V

Una vez que haya recopilado una respuesta de paso, haga clic en el botón Detener para detener la ejecución del VI.

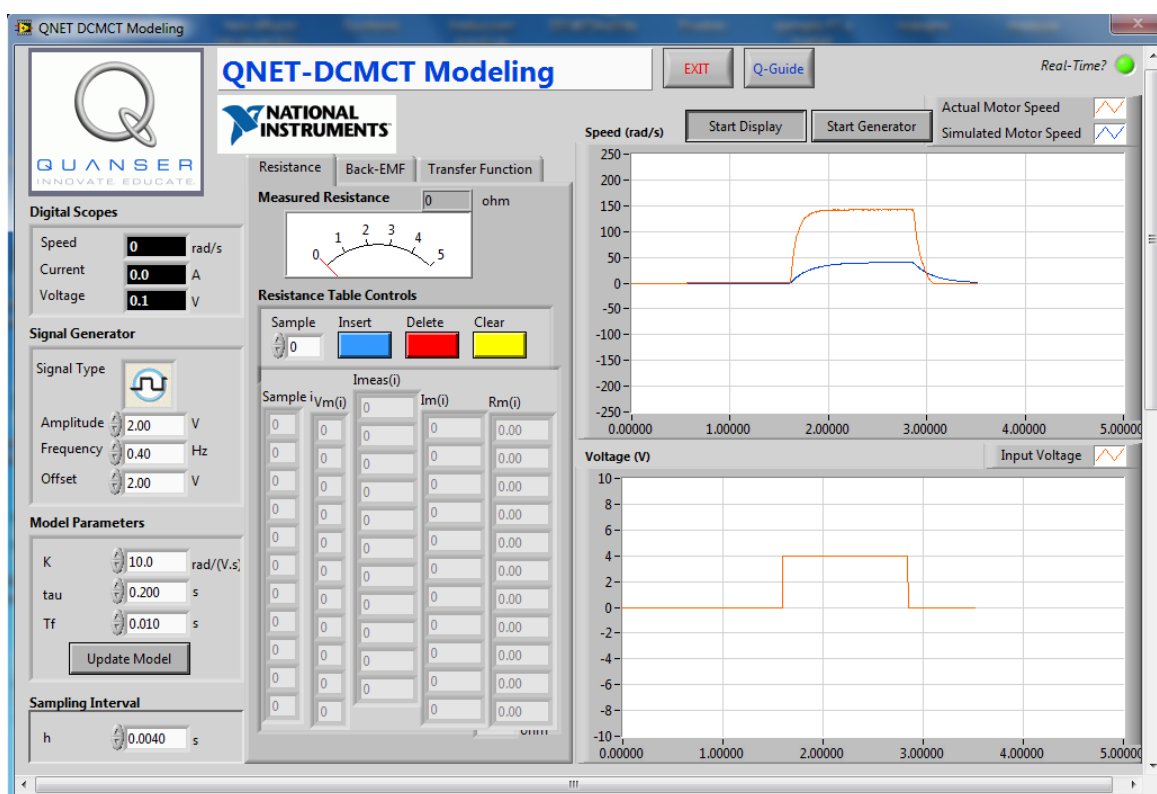


Figura 3-15: Archivo *QNET-DCMCT Modeling* con su respuesta gráfica de medición

Adjuntar las respuestas en la gráfica de Velocidad (rad/ s) y Voltaje (V). Ver en LabVIEW el VI de *QNET-DCMCT Modeling*.

- Si el procedimiento se siguió correctamente, los alcances deberían ser similares a los de la Figura 3-18.
- Selecciones en la curva la exportación de los datos a un archivo de Excel.

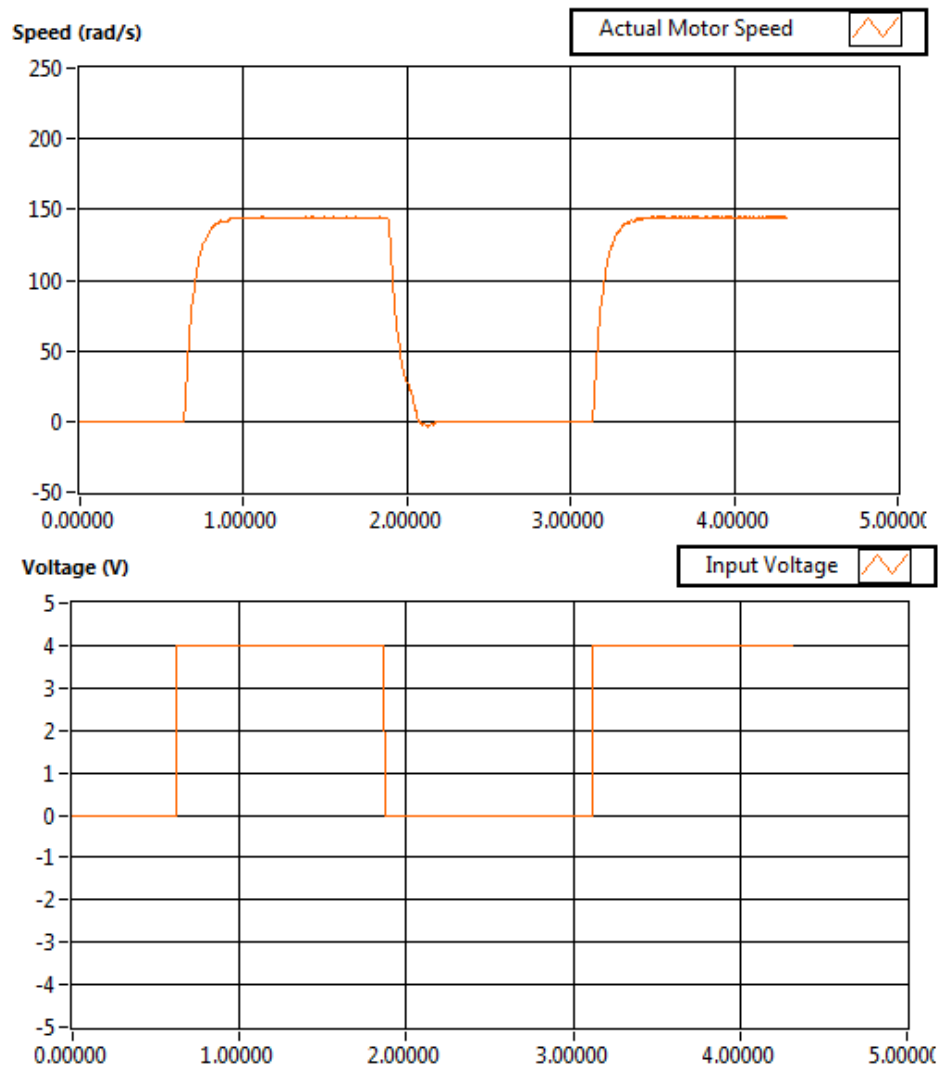


Figura 3-16: Respuesta al escalón en sistema de lazo abierto

Seleccione la pestaña Gráficas de Medición, para exportar los datos y definir los datos con mayor precisión desde Excel y poder ser graficados posteriormente desde Matlab.

218	0.864	139.296	4
219	0.868	139.431	4
220	0.872	140.37	4
221	0.876	140.068	4
222	0.88	140.256	4
223	0.884	140.765	4
224	0.888	140.915	4
225	0.892	140.629	4
226	0.896	140.692	4
227	0.9	141.197	4
228	0.904	140.664	4
229	0.908	140.585	4
230	0.912	141.083	4
231	0.916	140.787	4
232	0.92	141.091	4
233	0.924	140.82	4
234	0.928	140.459	4
235	0.932	141.007	4
236	0.936	140.607	4
237	0.94	141.195	4
238	0.944	140.705	4

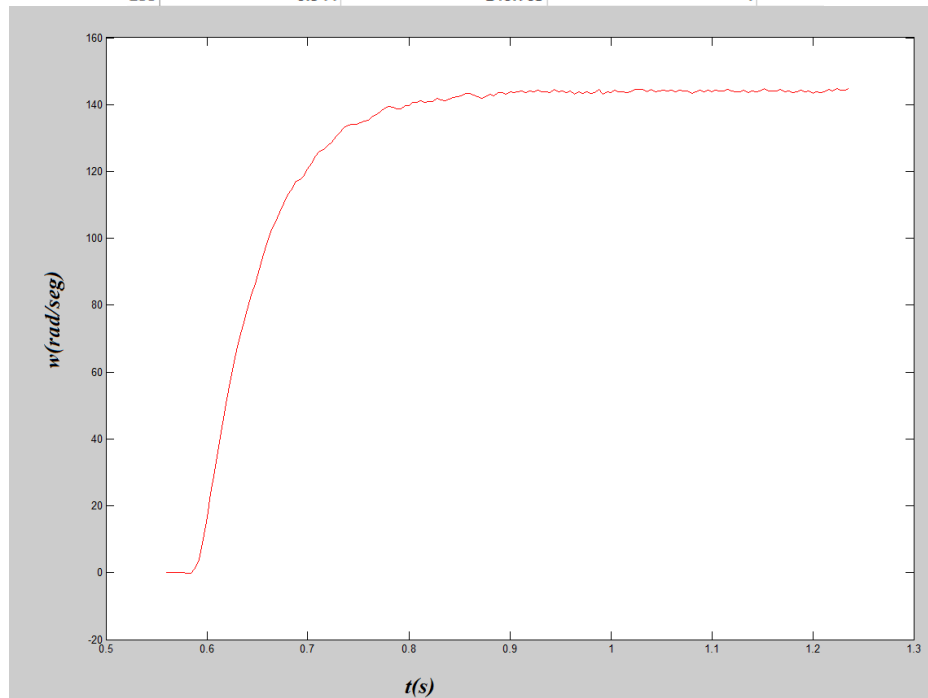


Figura 3-17: Gráfico utilizando la extracción de datos de la tabla de Excel

Utilizar la respuesta de Velocidad (rad/s) y Voltaje (V) para calcular la ganancia en estado estacionario del motor de corriente continua.

De la figura 3-19, las velocidades de eje de carga inicial y de estado estacionario medidas son:

$$t_0 = 0.2 \text{ rad/s} \quad \text{y} \quad t_1 = 141.679 \text{ rad/s}$$

Utilizando la ecuación (3.19) con los datos recogidos anteriores con una señal escalo de 4 volts, la ganancia resultante del estado estacionario es:

$$K = \frac{141.679 - 0.2}{4} = 35.369 \text{ rad/(Vs)}$$

Basado en el método de muestreo se define la constante de tiempo  $\tau$ . Del mismo modo para determinar el tiempo del motor usamos la ecuación  $t_1 = t_0 + \tau$ , La velocidad correspondiente medición definida en la ecuación (3.21). De la figura 3-19, el tiempo a la velocidad del eje y el tiempo de inicio en el escalón son:

$$t_1 = 0.940 \text{ s} \quad \text{Y} \quad t_0 = 0.763 \text{ s}$$

Dado como resultado el tiempo del motor con la ecuación (3.22) en:

$$\tau = t_1 - t_0 = 0.067 \text{ s}$$

Definido los valores de la constante de ganancia y la constante de tiempo que se encuentran en la prueba, queda establecido es sistema.

$$\frac{Y(s)}{U(s)} = \frac{\Omega(s)}{V_a(s)} = \frac{35.369}{0.067s + 1}$$

### 3.1.4.1 Comparativas de respuesta de los modelos

Con la información extraída de las pruebas realizadas anteriormente, podemos proceder a establecer la comparativa del modelos analíticos de forma Analítica e experimental obtenidos, para ello podemos utilizar el software Simulink de Matlab para identificar las variaciones de respuesta expresadas al construir lo modelos y por inspección en las curvas que se generan en la respuesta transitoria a la señal escalón poder analizarlas, esto como se muestra en la figura 3-20 y 3-21.

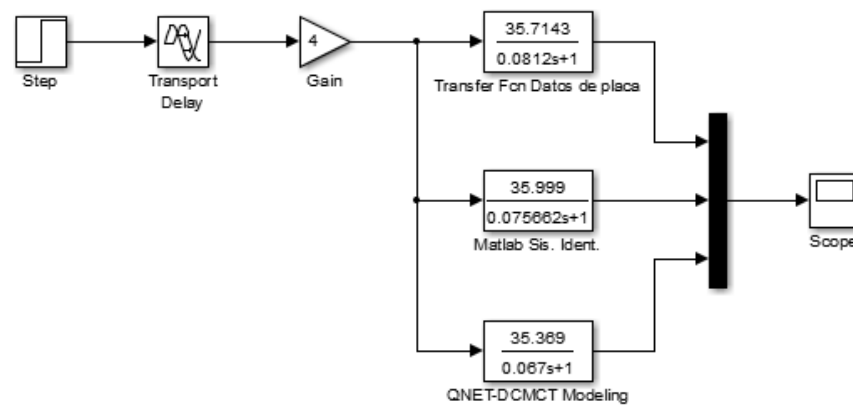


Figura 3-18: Respuesta comparativa entre la respuesta real y las simuladas al DCMCT

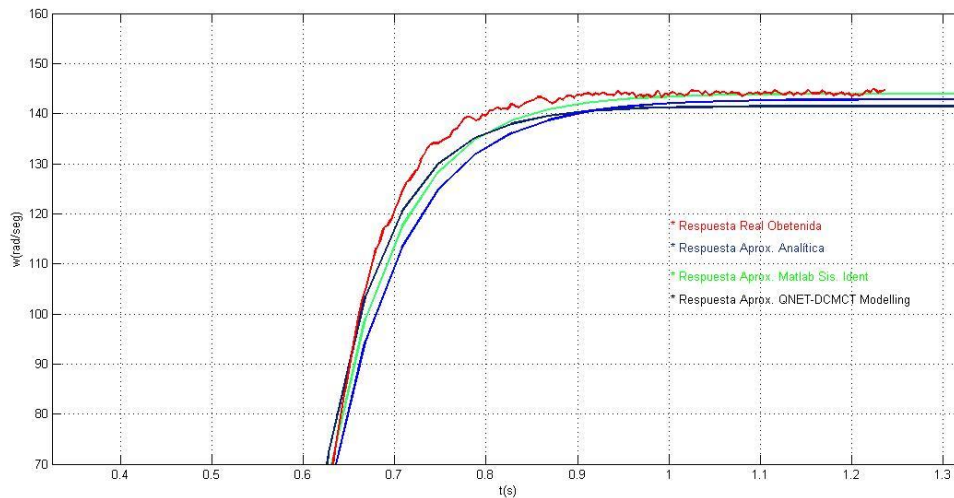


Figura 3-19: Gráfico comparativo entre la respuesta real y las simuladas al DCMCT

Esto nos deja una clara indicación de las pruebas antes hechas, es el modelo experimental del “*system identification*” de Matlab que por inspección en su comportamiento genera la mayor aproximación a la respuesta del sistema DCMCT obtenida al escalón mediante la señal del encoder adquirido por la tarjeta NI-ELVIS II como se muestra en la figura (3-19). Esta es representada por la curva de color verde en la figura 3-21, la curva de respuesta de las pruebas hechas en el vi de modelado para el método experimental del DCMCT “*modelling*” de Quanser se comportó con una alta aproximación, llegó un poco más abajo del valor de estabilización del sistema real obtenida por la respuesta. Esto establece claramente los diversos aspectos que generan las experiencias de aprendizaje al desarrollar distintos métodos de identificación a un mismo sistema.

Sistema seleccionado para la implementación de los controladores:

$$G(s) = \frac{\Omega(s)}{V_a(s)} = \frac{35.666}{0.07566s + 1}$$



# CAPÍTULO IV

## 4 CAPÍTULO 4

### 4.1 DISEÑOS Y ESTRATEGIAS DE SINTONIZACIÓN USANDO EL CONTROLADOR PID

#### 4.1.1 Diseño Analítico del controlador PI a partir de un desempeño determinado al sistema de Quanser

Los controladores PID o controladores de tres términos, son aquellos cuyo desempeño está determinado por la aplicación de las tres acciones básicas de control la acción proporcional (P), acción integral (I) y acción derivativa (D), cada uno de estos modos de control le aporta diferentes características y pueden fácilmente relacionarse con los criterios de desempeño del sistema de control.

En la tabla 4-1 se muestra cómo afecta el incremento de las ganancias de cada una de estas acciones a los distintos criterios de desempeño, como son el tiempo de subida, sobre-pico, tiempo de establecimiento y error en estado estable. (Hernandes, Peza, Garcia, & Torres, 2015)

Tabla 0-1: Características de las constantes del controlador PID

Parámetros del controlador	Tiempo de subida	Sobre-pico Overshoot	Tiempo de estabilización	Error en estado estable
$K_p$	Disminuye	Aumenta	Poco cambio	Disminuye
$K_i$	Disminuye	Aumenta	Aumenta	Elimina
$K_d$	Poco cambio	Disminuye	Disminuye	Poco cambio

El controlador de acción integral en realidad no existe controladores que actúen únicamente con acción integral, siempre actúan en combinación con una acción proporcional, primero actuando la acción proporcional y de manera instantánea, mientras que la acción integral actúa mediante un intervalo de tiempo ( $T_i$ ).

Como se mencionó en capítulo anterior se conforma como la planta a controlar la ecuación (3.18) para el sistema QNET-DCMCT. Debido a ello la función de transferencia



del bloque del controlador PI es aplicado en la forma como muestra la figura 4-1, esto es aplicado en sistema en lazo cerrado con retroalimentación. Estableciendo como propuesta para el controlador.

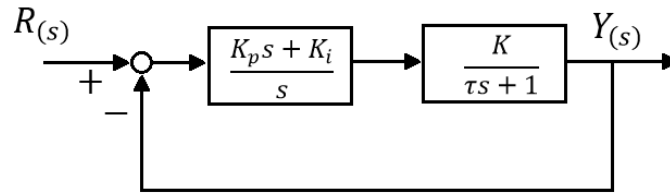


Figura 0-1: Controlador PI aplicado al sistema

Si partimos del sistema de ecuaciones que definen el sistema de control en lazo cerrado con retroalimentación, tendremos primero que definir la ecuación de la planta y desarrollar el polinomio característico del sistema respecto a la forma.

$$Sistema = \frac{Y(s)}{R(s)}$$

Los parámetros que estableceremos por considerar de mejor respuesta al sistema de Quanser visto en el capítulo 3 son:  $K = 35.999$  y  $\tau = 0.07566$ , definiendo la planta al usar.

$$Planta = \frac{K}{\tau s + 1} = \frac{35.999}{0.07566s + 1} \quad (4.1)$$

Estableciendo el controlador PI en el sistema como se muestra en la figura 4-1.

$$PI = \frac{k_p s + k_i}{s} \quad (4.2)$$

Por lo tanto.

$$\frac{Y}{R} = \frac{\frac{K(k_p s + k_i)}{(\tau s + 1)s}}{1 + \frac{K(k_p s + k_i)}{(\tau s + 1)s}} = \frac{K \cdot k_p s + K \cdot k_i}{\tau s^2 + s + K \cdot k_p s + K \cdot k_i} \quad (4.3)$$

Definimos que es un sistema de segundo orden con un cero:

$$\frac{Y}{R} = \frac{K \cdot k_p s + K \cdot k_i}{\tau s^2 + (K \cdot k_p + 1)s + K \cdot k_i} \quad (4.4)$$

Desarrollando para la forma general del sistema, con la ecuación característica.

$$s^2 + \frac{K \cdot k_p + 1}{\tau} s + \frac{K \cdot k_i}{\tau} = s^2 + 2\zeta \omega_n s + \omega_n^2 \quad (4.5)$$

Esto es:

$$\omega_n^2 = \frac{K \cdot k_i}{\tau} \quad (4.6)$$

Por lo tanto si despejamos  $k_i$  en (4.6) tenemos:

$$k_i = \frac{\tau \omega_n^2}{K} \quad (4.7)$$

De mismo modo de la ecuación (4.5):

$$2\zeta \omega_n = \frac{K \cdot k_p + 1}{\tau} \quad (4.8)$$

El  $k_p$  queda definido de (4.8) como:

$$k_p = \frac{2\zeta \omega_n \tau - 1}{K} \quad (4.9)$$

A partir del desempeño requerido del sistema de control, se determinaran los valores de las ganancias  $k_p$  y  $k_i$ . En este caso asumiremos un determinado  $OS$  (porcentaje de sobre pico, Overshoot) y un determinado tiempo de establecimiento  $t_s$ . Sabiendo que:

$$OS = e^{\frac{-\pi\zeta}{\sqrt{1-\zeta^2}}} * 100 \quad (4.10)$$

Y que:

$$t_s = \frac{4}{\zeta \omega_n} \rightarrow \text{para } \pm 2\% \quad \text{y} \quad t_s = \frac{3}{\zeta \omega_n} \rightarrow \text{para } \pm 5\% \quad (4.11)$$

Se propone los siguientes criterios de desempeño para la sintonización en el sistema, con un porcentaje de Overshoot no mayor al 20% y un tiempo de estabilización de 0.15 segundos.

$$OS < 20\%, t_s = 0.15 \text{ seg.}$$

Para el coeficiente de amortiguamiento  $\zeta$  se obtiene partiendo de la ecuación característica del tiempo de estabilidad. El cual se encuentra en referencia bajo dos criterios que son con un 2% o un 4%. Debemos establecer del mismo modo el factor de amortiguamiento en el sistema utilizando la siguiente ecuación ya habiendo definido el porcentaje del sobre-pico (Overshoot).

$$\zeta = \frac{-\ln(\%OS/100)}{\sqrt{\pi^2 + \ln^2(\%OS/100)}} = \quad (4.12)$$

$$\zeta = \frac{-\ln(0.2)}{\sqrt{\pi^2 + \ln^2(0.2)}} = \frac{1.61}{\sqrt{9.87 + 2.59}} = 0.4549$$

Para determinar la razón de amortiguamiento del sistema, se utiliza la ecuación (4.1.12) y se despeja  $\omega_n$ .

$$t_s = \frac{4}{\zeta \omega_n} \rightarrow \omega_n = \frac{4}{0.15 * 0.4549} = 58.486 \quad (4.13)$$

Podemos definir parámetros del controlador PI utilizando las ecuaciones (4.7) y (4.9)

$$k_p = \frac{2\zeta\omega_n\tau - 1}{k} = \frac{2(0.4549)(58.486)(0.07566) - 1}{35.999} = 0.0843$$

$$k_i = \frac{\tau\omega_n^2}{k} = \frac{0.07566 * (58.486)^2}{35.999} = 7.1819$$

Se desarrolla un archivo .m en Matlab para la obtención de controlador PI bajo las ecuaciones anteriormente establecidas en el sistema analítico se emplea como se ve en la figura 4-2 y con los parámetros que determinan las condiciones para un criterio de

desepeño deseado en la curva de respuesta al aplicar la señal escalon y la señal de perturbación del sistema como se muestra en la figura 4-3.

```

PISADOT.m
1 %Implementación de PI para sistema de primer orden.
2 %definición de la planta
3 K=35.999;
4 tau=0.07566;
5 planta=tf(K,[tau 1]);
6 %Función de transferencia del controlador del PID.
7 %s^2+((K*Kp+1)/tau)*s+(K*Ki)/tau
8 %Tiempo de estabilidad mínimo "ts" en seg.
9 ts=0.15;
10 %Overshoot, porcentaje de sobre paso mínimo "%OS"
11 OS=0.2;
12 xi=-log(OS)/sqrt((pi^2)+(log(OS)^2))
13 wn=(4/(ts*xi))
14 %Acción proporcional
15 Kp=((2*xi*wn*tau)-1)/K
16 %Acción Integral
17 Ki=((tau*wn^2)/K)
18 %Diseño del controlador PI
19 PI=tf([Kp Ki],[1 0]);
20 %Sistema en lazo abierto con implementación del controlador PI
21 Gla=series(PI,planta);
22 %Sistema de en lazo cerrado como servo con el controlador PI
23 Glcs=feedback(Gla,1);
24 %Respuesta al escalon del sistema
25 step(Glcs)
26 hold
27 %Sistema de en lazo cerrado como Regulador con el controlador PI
28 Glcr=feedback(planta,PI);
29 step(Glcr)
30 G=tf(Glcs);
    
```

Figura 0-2: Archivo del editor de Matlab para sintonización PI como servo y regulador.

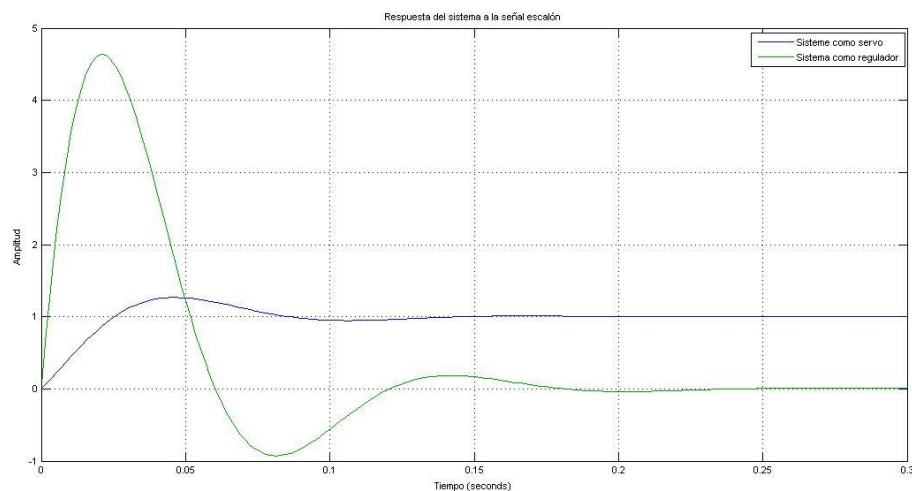


Figura 0-3: Grafica y parámetros de respuesta del controlador PI

Los criterios de respuesta del motor de acuerdo a la señal escalón proporcionada, se apreciar una respuesta satisfactoria, se observa que al establecer una perturbación estática presenta un sobrepaso de 68%, Esto dicta para poder hacer un ajuste de sintonización de los parámetros utilizando un controlador PI sintonizado en el sistema.

#### 4.1.2 Diseño del controlador PI Utilizando la herramienta PID Tuner.

En este método de sintonización del controlador se agrega una perturbación estática permanente, establecida por una carga externa al disco de carga de propio sistema. Esta perturbación permanente permea al sintonizar el controlador como un sistema regulador, para establecer un control de posición con una adaptación de escala de medición.

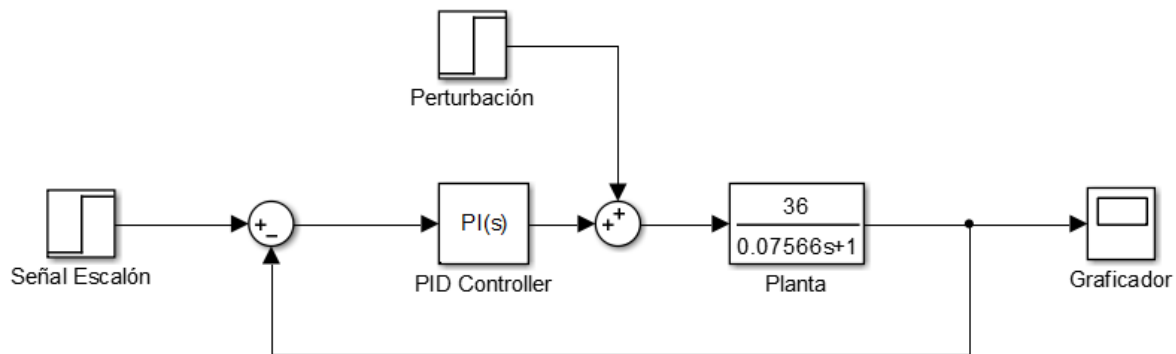


Figura 0-4: Diagrama de bloques del sistema de control realimentado con PI

Existen herramientas que nos pueden ayudar a mejorar la sintonización del controlador utilizado en el sistema. Uno de ellos es la aplicación PID Tuner perteneciente al software de Matlab, en ella se aplica un mecanismo de control en retroalimentación del sistema para modificar los criterios de respuesta y establecer los parámetros que cumplan con una mejora en el requerimiento del desempeño.

Se ejecuta la aplicación PID Tuner de Matlab, escribiendo “**PID Tuner**” en la línea de comandos desde “**command window**” o bien haciendo doble clic desde el bloque de simulink del **PID controller**, este al desplegar sus opciones también contara la opción de sintonización con la aplicación “**Tuner**”, otra opción es seleccionando la aplicación desde la galería de aplicaciones.

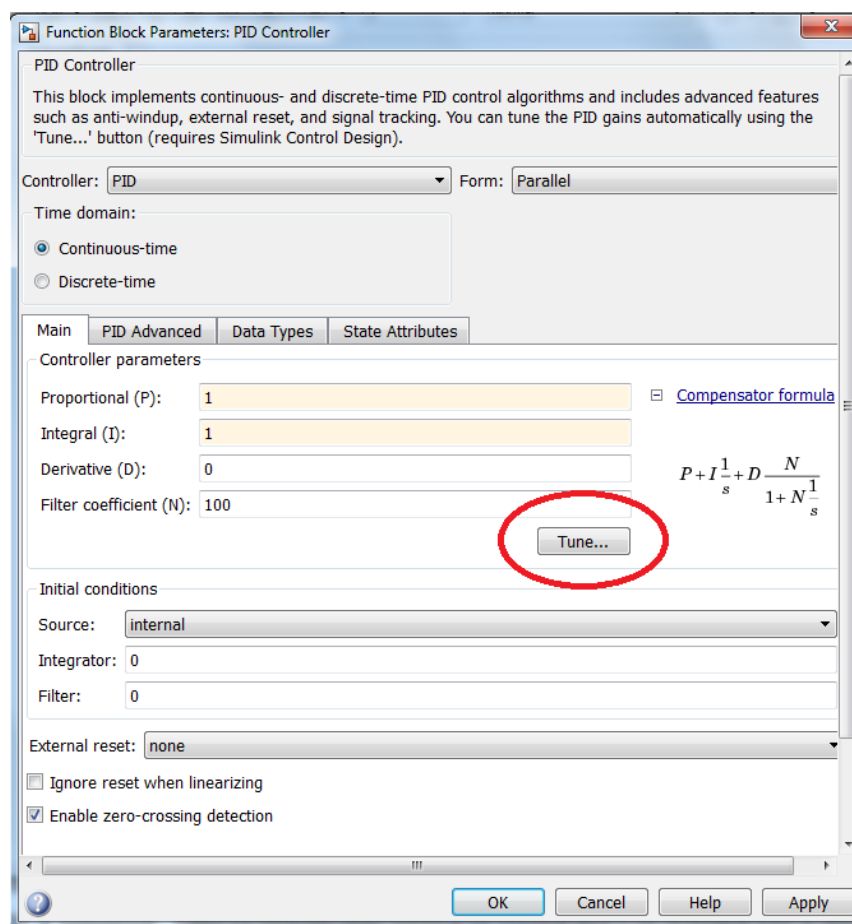


Figura 0-5: Sintonización del PI utilizando la herramienta PID Tuner

Podemos generar la importación de los datos de entrada y de salida que se obtuvieron del capítulo 3 del sistema. Seleccionar la opción para identificar una nueva planta “*Pant*” definido como la planta del sistema, Esto para lograr dar pie a especificar ambas señales. Vemos especificamos como la “*planta*” que es la que contiene los datos almacenados del sistema.

La importación de datos usando PID Tuner se muestran en la figura 4-6. Al realizar la dinámica del sistema, insertamos una variación en él a los cinco segundos y registramos la salida (disponible como *output* de variable en el espacio de trabajo de Matlab). El análisis visual de los datos de salida muestra que parece capturar los cambios en la dinámica del sistema. La entrada escalonada es una forma habitual de estimular la dinámica del sistema, pero se pueden emplear otras señales de impulso en su lugar. (Turevskiy, 2018)

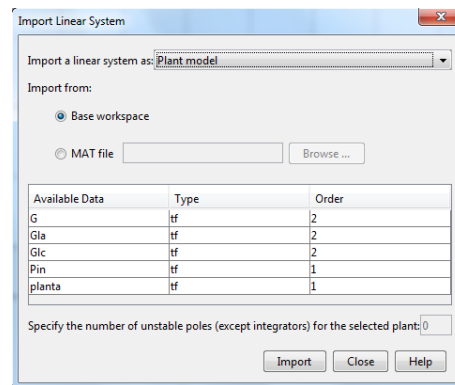


Figura 0-6: Importa el sistema desde “*Command Windows*” en PID Tuner

Si nos basamos el diseño de la planta como de la figura 4-4, podemos llamar a la simulación en simulink y acceder al PID controller para su sintonización directa. Lo primero que debemos definir es el tipo de controlador a utilizar en el apartado de “*controller*” y seleccionamos PI como muestra la figura 4-7, del mismo modo debemos seleccionar el modo de respuesta, las que se abordan aquí es “*Reference tracking*” el sistema en respuesta a la señal escalón como referencia y “*input disturbance rejection*” la respuesta del sistema a entrada de perturbación como se muestra en la figura 4-8. Esto para analizar el QNET-DECMT como sistema tipo servo y como regulador.

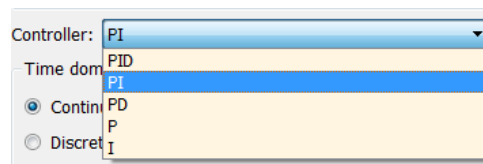


Figura 0-7: Seleccionando el tipo de controlador a sintonizar

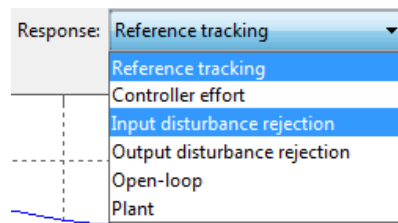


Figura 0-8: Seleccionando el tipo de controlador a sintonizar

Se carga la herramienta del PID Tuner desde el bloque de implementación continua de PID controller del simulink y se da pie al diseño de respuesta que seleccionamos con las características del control establecido.

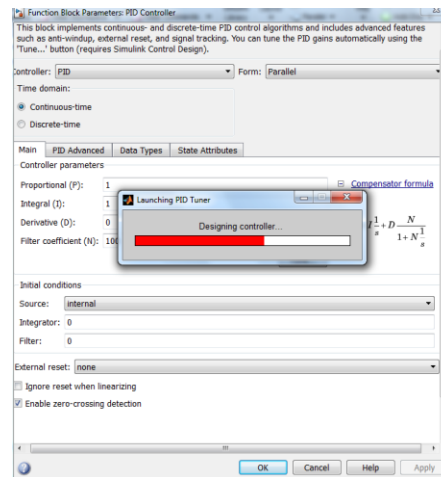


Figura 0-9: Ejecución de la herramienta PID Tuner desde simulink.

Utilizando los controles deslizantes interactivos con la finalidad de ajustar el rendimiento de lazo cerrado que son necesarios para diseñar un controlador rápido y estable, necesitamos asegurarnos de que la salida del controlador enviada al motor no supere el valor máximo que puede proporcionar el QNET-DCMCT. En la gráfica de la figura 4-9 muestra el resultado del controlador al diagrama de respuesta escalón de lazo cerrado, se realiza ajustes que puedan lograr el rendimiento deseado: una respuesta rápida y estable en el motor del sistema.

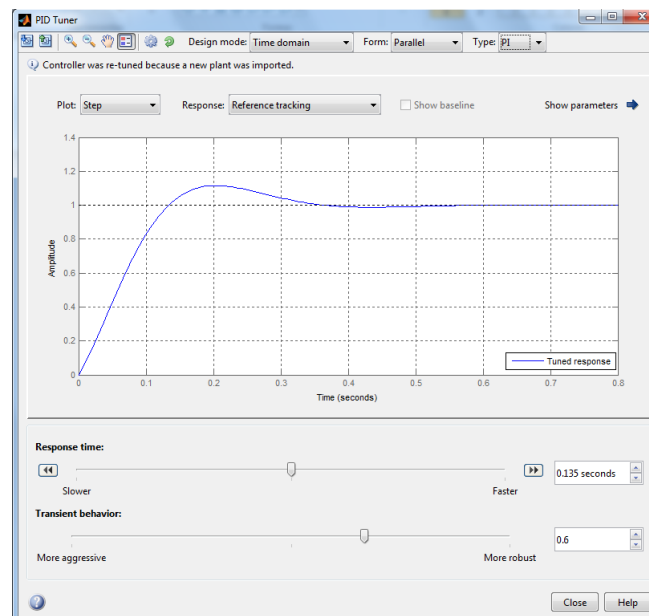


Figura 0-10: Respuesta al escalón unitario en el sistema PID Tuner



Se observa la gráfica y cómo se comporta al modificar la respuesta con las opciones de tiempo de respuesta “*Response Time*” y la respuesta transitoria “*Transient Behavios*”. Esto para hacer el controlador más robusto en su comportamiento, generando así las modificaciones del diseño gracias a la aplicación de Matlab.

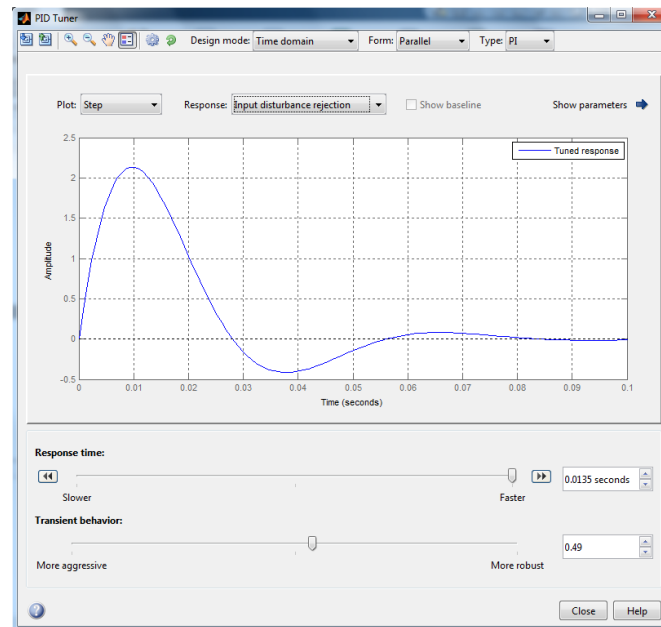


Figura 0-11: Ajuste de parámetros en la respuesta del controlador PI en el sistema

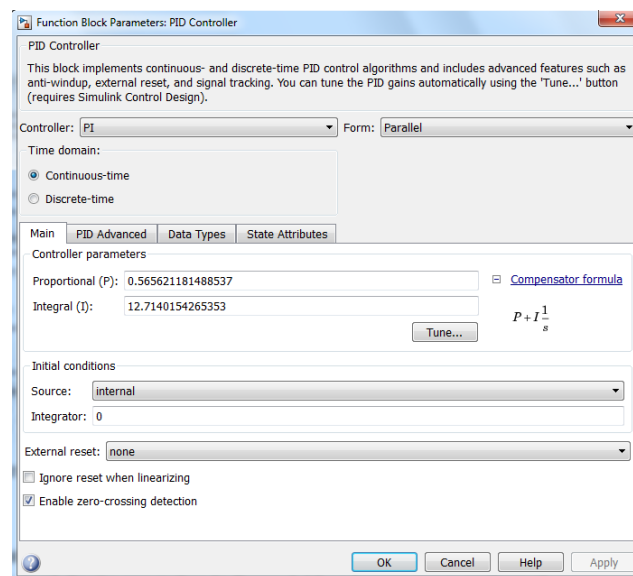


Figura 0-12: Respuesta sintonizada elegida para exportación al sistema DCMCT

Para extraer los datos y ver el comportamiento del controlador es necesario recurrir nuevamente a simulink, bajo el diseño que indica la figura 4-4 se lograra observar y analizar el comportamiento y el desempeño con los valores de sintonización obtenida por la aplicación **PID Tuner** en los parámetros del controlador PI aplicado al sistema.

A partir del análisis del diagrama de bloques, se obtiene la respuesta, de primera instancia con el controlador unitario, al ser aplicada al 0.5 segundo de haber aplicado la señal escalón al sistema, se observó un sobre paso de 60% de Overshoot, al establecer la sintonización por PID Tuner y considerando los parámetros del control PI, Observamos que se logra limita el Overshoot hasta un 25% así como el modo integral logra eliminar rápidamente el error en estado estable, disminuyendo la estabilidad relativa del sistema como muestra la figura 4-13.

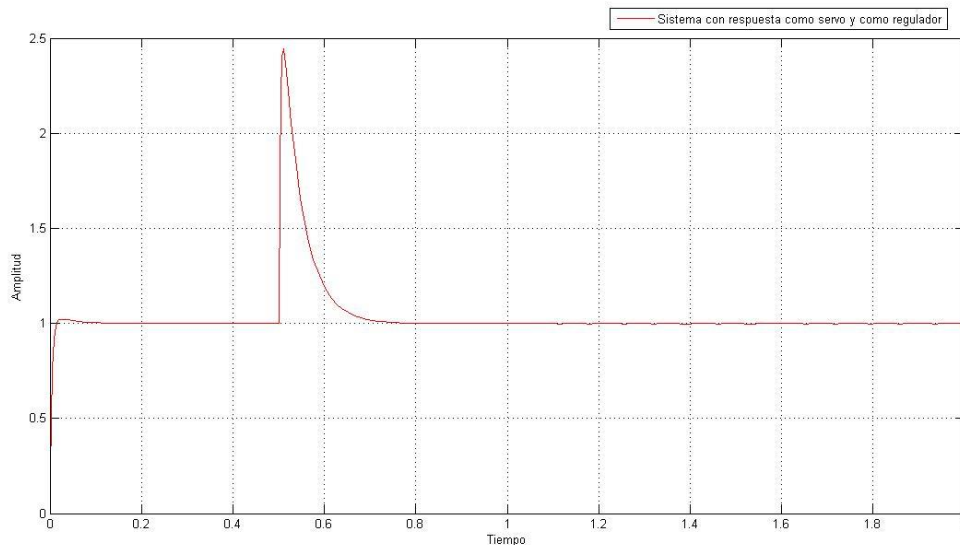


Figura 0-13: curva de respuesta al controlador PI sintonizado desde PID Tuner.

Los resultados entregados por el desempeño del controlador PI sintonizado bajo el sistema PID Tuner serán aplicado en un vi de diseño, Ahora implementaremos un PID estándar de LabVIEW aplicado al sistema QNET-DCMCT mediante la construcción de un vi y contrastar físicamente los resultados. Esta respuesta cabe mencionar que cuenta permanentemente aplicada la carga en acoplamiento al disco de carga al sistema y que busca ser corregido mediante la compensación y el desempeño del ajuste de parámetros del controlador.

Para implementar en la utilización de los datos obtenidos en la sintonización del controlador PI, creamos un archivo vi para el control de sistema. Procedemos a abrir un archivo nuevo, creado para PID en LabVIEW. Se debe involucrar un sistema de lazo *while* para su ejecución continua y establecer en ello en el diagrama de bloques los *subvi* de Quanser que contiene el sistema del encoder para la selección del dispositivo de adquisición (ELVIS II), así como el *subvi* lector de posición (*Pos Read*) que nos proporciona la posición del sistema a controlar del QNT-DCMCT, Este adjudica la señal como la variable del proceso de sistemas y la compararla con la de control establecida personalmente a través del “*pointer slide*”. Se utiliza la salida de control de PID de LabVIEW a un asistente de la DAQ “*data acquisition*” y a su vez comprar la señal de entrada y de salida con una indicador gráfico, las figuras 4-16 y 4-17 muestra elaborado el vi desarrollado en el sistema usando LabVIEW. (Lewis, 2018). Un vez ejecutado el vi se puede guiar desde la tabla 4-2 para su operación.

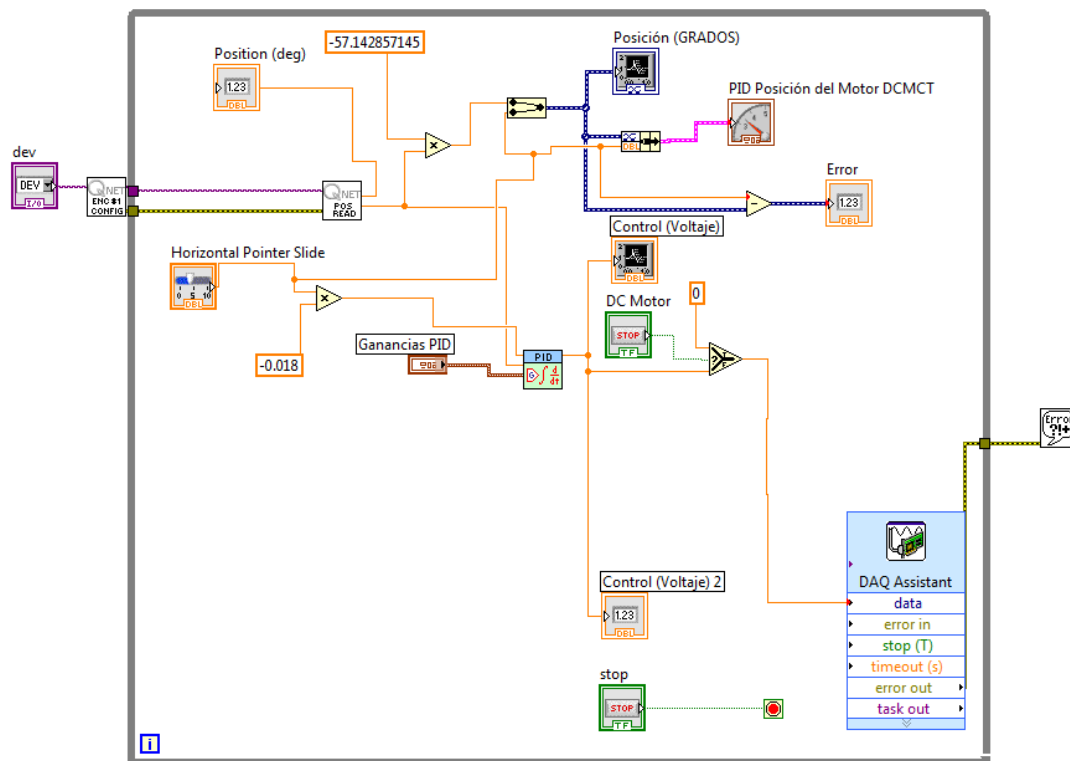


Figura 0-14: Diagrama de bloques, vi de posición del motor de cd con PID en LabVIEW

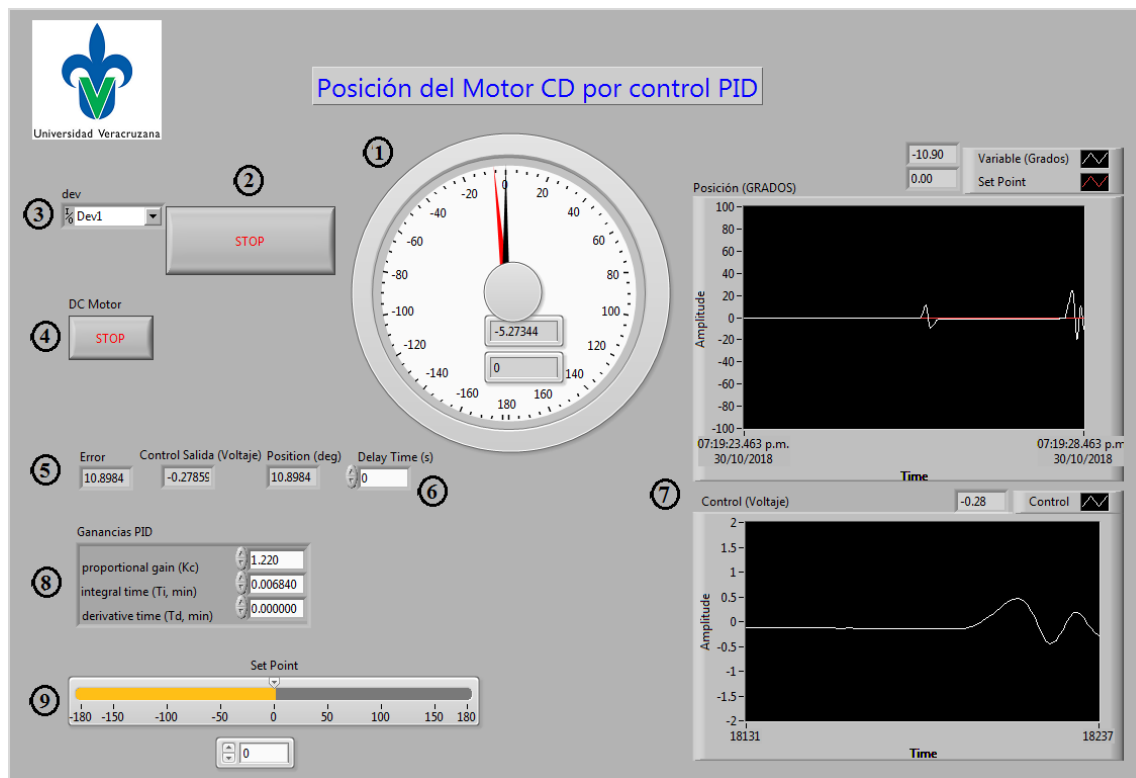


Figura 0-15: Archivo vi de posición del motor de cd utilizando PID (Frontal Panel)

Tabla 0-2: Descripción del vi diseñado para control de posición del motor CD por PID

Número	Descripción
1	Medidor de posición del motor CD
2	Botón de paro en la simulación del vi
3	Selección del dispositivo de adquisición de datos (NI ELVIS II)
4	Botón de paro (stop de emergencia) del motor CD
5	Medidores de error, voltaje de salida y posición
6	Selector de retardo de tiempo $t_d$
7	Pantalla grafica de las señales de posición y control de voltaje de salida
8	Ganancias programadas del controlador PID
9	Set-point del sistema

Podemos ver su comportamiento y respuesta si acoplamos una carga estática permanente al sistema como una perturbación más acoplada vista en la figura 4-18.

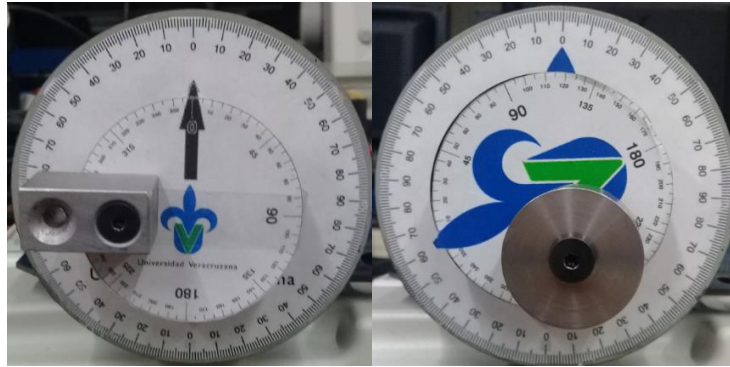


Figura 0-16: Carga Permanente acoplada al sistema DCMCT

La respuesta la podemos observar el grafica comprando el *set point* contra la variable que arroja el encoder en el graficador “Posición (GRADOS)” y el indicador gráfico de posición.

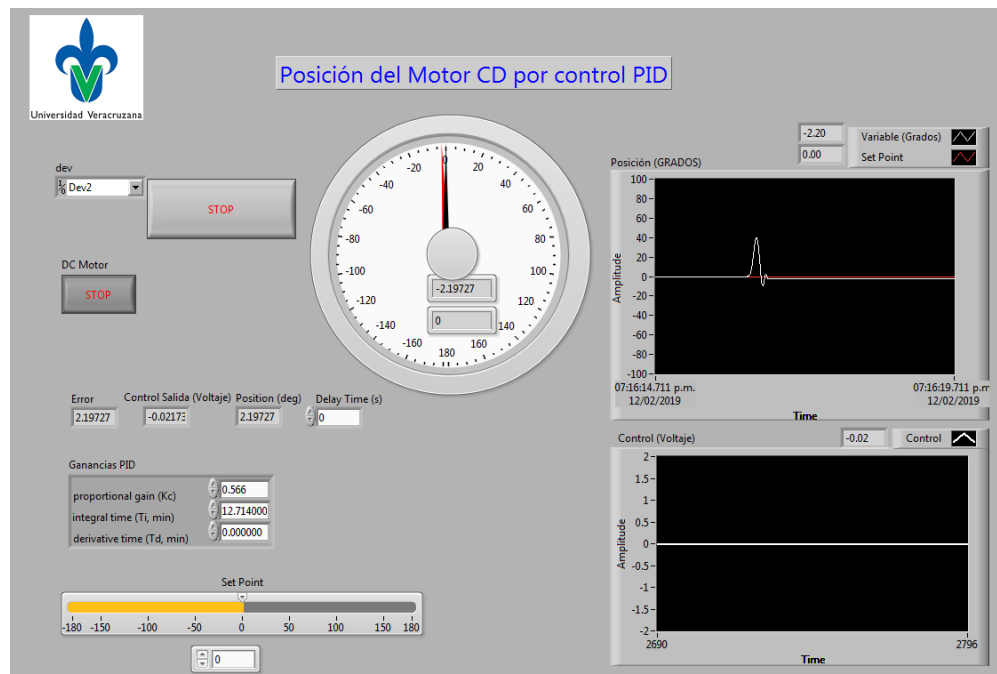


Figura 0-17: Aplicación del diseño de ganancias en sintonización usando PID

La respuesta puede ser exportada a un archivo de Excel o bien como imagen para su mejor análisis del comportamiento del controlador con los criterios de desempeño establecidos en la sintonización como se muestra en la figura 4-20.

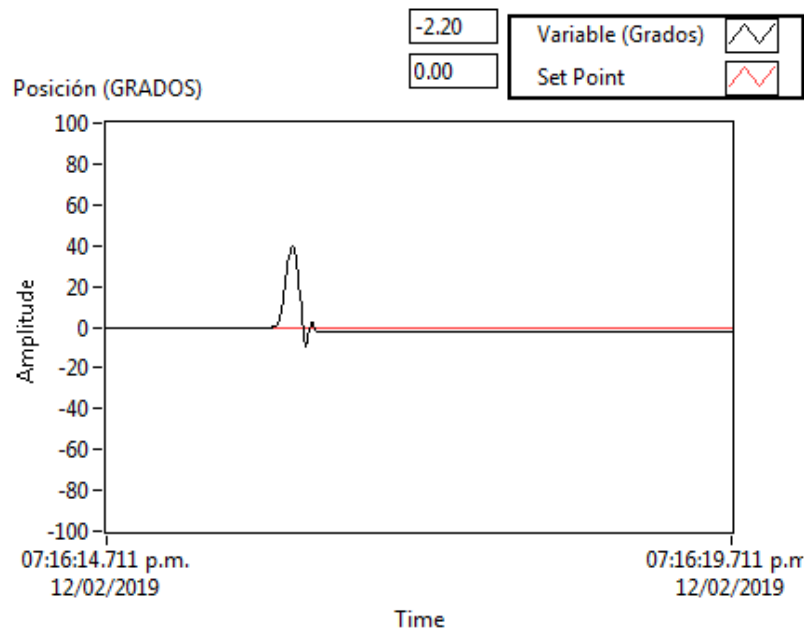


Figura 0-18: Curva de respuesta al sistema utilizando PID Tuner para sintonización

### 4.1.3 Implementación Virtual de Posición utilizando herramientas de labview y QNET-DCMCT Position Control.

El control de la posición en el motor del sistema. Bajo este esquema también se puede diseñar un controlador proporcional con derivados integrales de acuerdo con las especificaciones del QNET-DCMCT. El diagrama de bloques para el control PID en lazo cerrado se muestra a continuación. (Quanser, 2008)

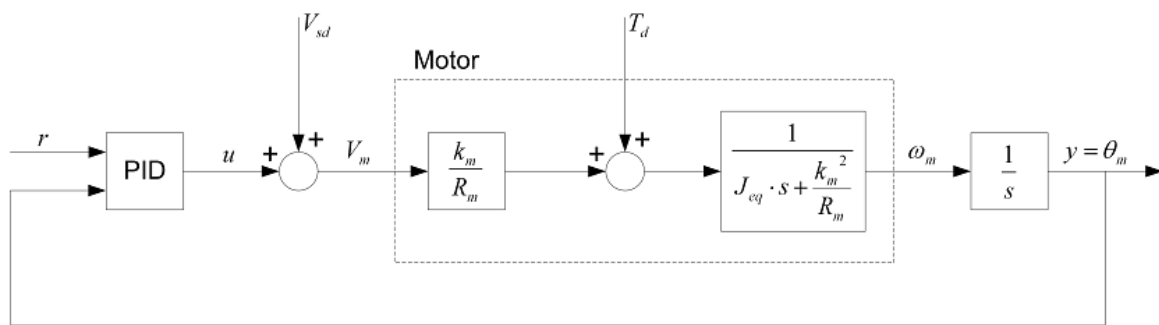


Figura 0-19: Diagrama de sistema de lazo cerrado de diseño de Quanser

El sistema de bloques que representa al motor de CD en conjunto como se aprecia en la figura 4-9 es en términos de la constante del motor  $k_m$ , la resistencia de armadura del motor eléctrico  $R_m$ , y el momento equivalente de inercia del pivote del motor  $J_{eq}$ , la cual es equivalente a  $J$  establecida en el capítulo de 3. La perturbación estática aplicada a la inercia está representada por la variable de par  $T_d$ .

Establecemos el seguimiento de una posición basado en una señal onda cuadrada aplicada al sistema QNET-DCMCT mediante un generador de señal en el vi, esto para analizar los disturbios y los efectos con una carga estática acoplada de forma directa al motor, es importante mencionar que el vi de Quanser cuenta con la opción de usar un interruptor de control (**Disturbance**) para perturbar el sistema al estar en ejecución y puede ser utilizado para observar su comportamiento y su respuesta para la estabilización propia del sistema. El instrumento virtual de LabVIEW (**QNET\_DCMCT\_Position\_Control**) para el control de posición se muestra en la figura 4-20. (Quanser, 2008)

Asegúrese de que *QNET\_DCMCT\_Position\_Control.vi* esté abierto y configurado como se describe en el capítulo 3, Ejecutar el archivo vi en LabVIEW con el botón de *play*, el motor debe estar girando y los alcances en el VI debe aparecer similitud como se muestra en la figura 4-21.

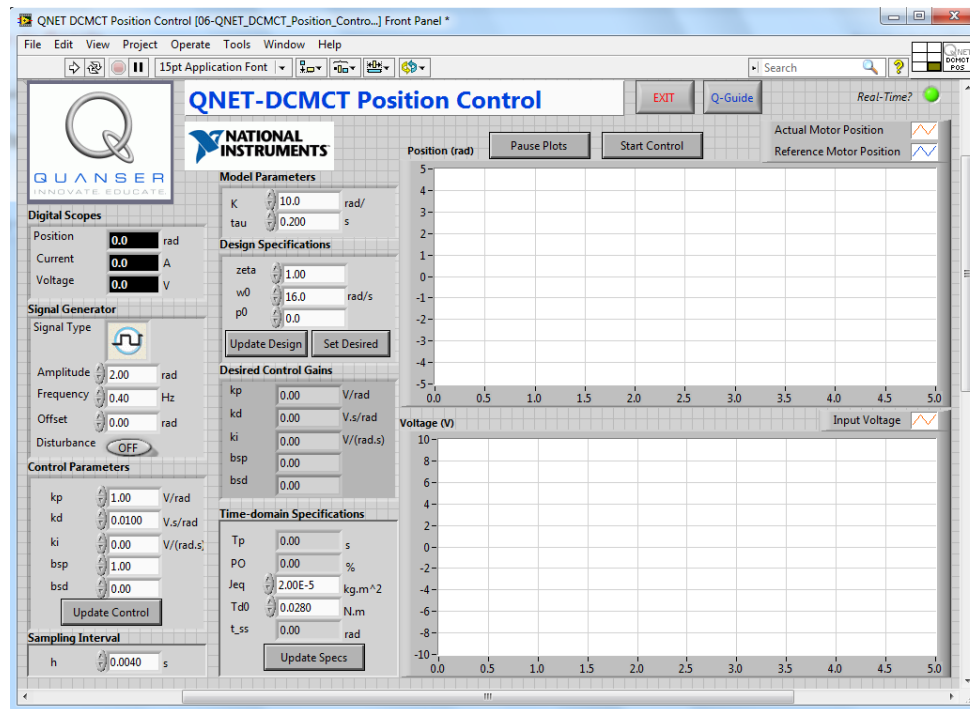


Figura 0-20: Instrumentación virtual en LabVIEW *QNET-DCMCT Position Control.vi*

Establecer los parámetros del generador de señales “*signal generator*”:

- Amplitud = 2.00 rad
- Frecuencia = 0.40 Hz
- Offset = 2.00 rad

Anexamos el modelo del sistema obtenido en el capítulo 3:

- $K = 36$  rad
- $\tau = 0.0757$  seg.

Definimos las especificaciones de criterio de desempeño del controlador:

- $\zeta = 1$
- $\omega = 16$  rad/s



Determinar el comportamiento de la posición medida (roja) con respecto a la posición de referencia (azul) en la gráfica. El motor al entrar en funcionamiento girara siguiendo la señal de onda cuadrada.

La respuesta se vuelve más oscilatoria a medida que vamos aumentando la ganancia proporcional, el rebasamiento inicial de la señal de control también aumenta con  $k_p$ , la respuesta se estabiliza reduciendo el error en estado estable con la ganancia integral  $k_i$ , en tanto la velocidad con que cambia el error aumentando la amplitud o disminuyéndola se define con la ganancia derivativa  $k_d$  en el sistema.

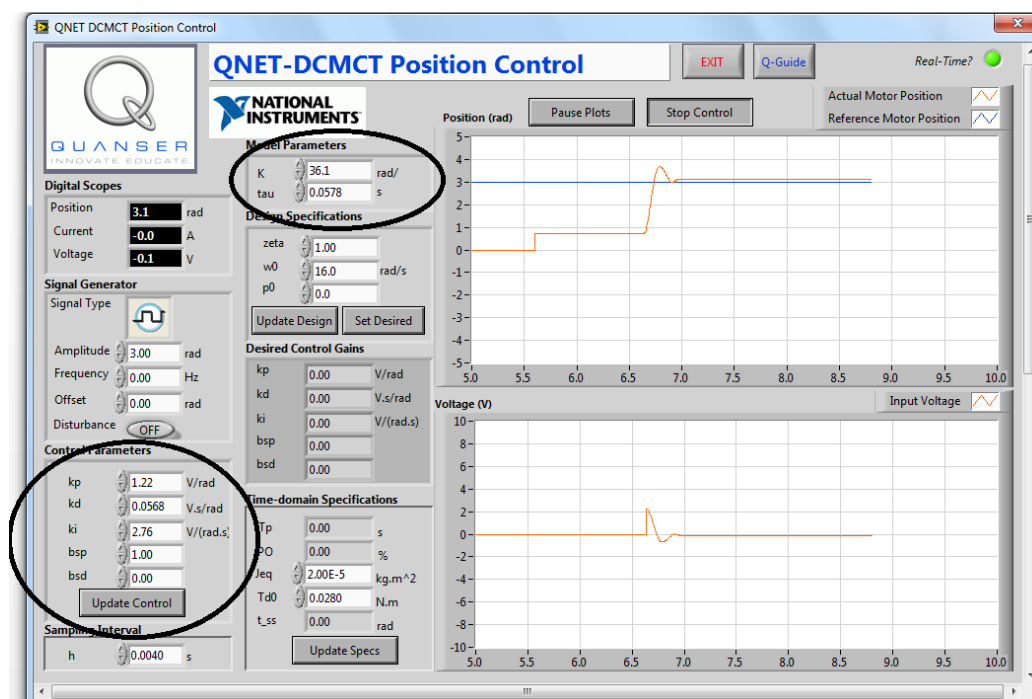


Figura 0-21: Grafica de respuesta a los parámetros del controlador usando DCMCT position control.

Al utilizar el diseñador de control de ganancias “*Desired Control Gains*” del vi QNET-DCMCT position control, obtendremos los parámetros de diseño del PID, para después cargarlos al controlador “*Control parameter*” observando la respuesta del sistema. Se debe capturar la respuesta grafica de la señal de posición y la señal de controlada, del mismo modo la señal de entrada de Voltaje (V) respecto al desplazamiento visto la figura 4-23.

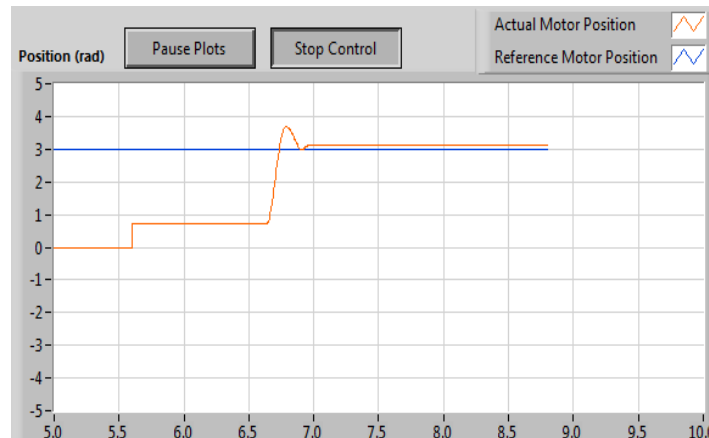


Figura 0-22: Curva de respuesta a la señal corregida con la configuración de sintonización.

Los resultados fueron satisfactorios al desempeño del controlador PID sintonizado previamente establecido por el método de sintonización.

Tabla 0-3: Resumen de resultados del control de posición usando vi de Quanser

Descripción	Símbolo	Valor	Unidad
Ganancia del modelo utilizado	$K$	35.999	$rad/s$
Constante de tiempo del modelo utilizado	$\tau$	0.0757	$s$
Ganancia proporcional	$k_p$	1.22	$V/rad$
Ganancia derivativa	$k_d$	0.0568	$V/(rad/s)$
Ganancia integral	$k_i$	2.76	$V/(rad \cdot s)$
Medición de porcentaje de sobrepaso (Overshoot)	$PO$	4.0	%

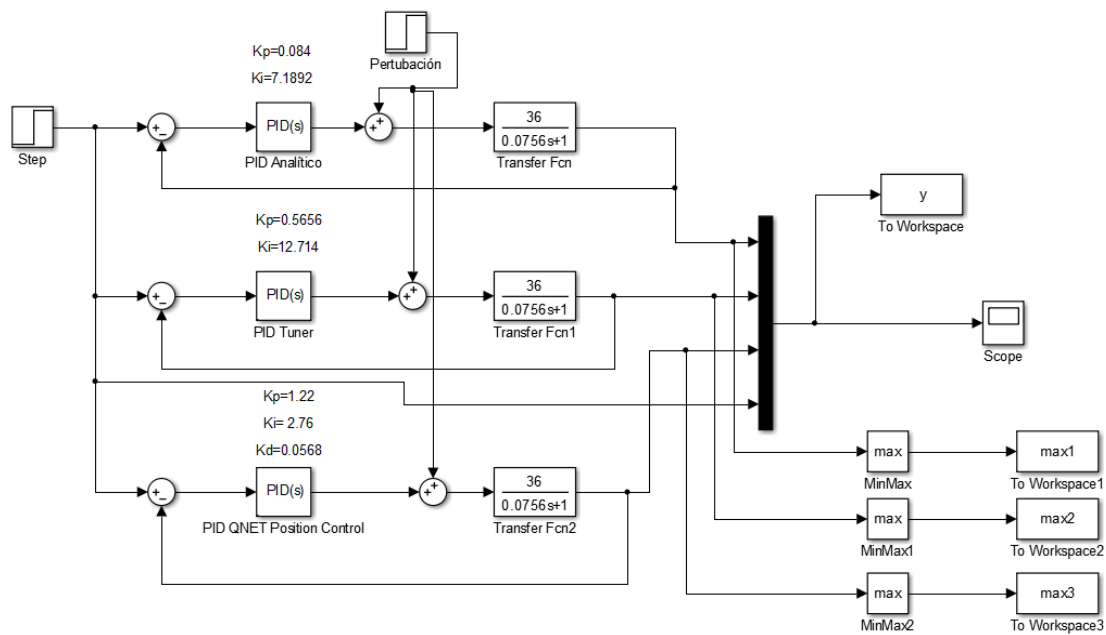


Figura 0-23: Diagrama de simulación que permite comparar el desempeño de los métodos utilizados

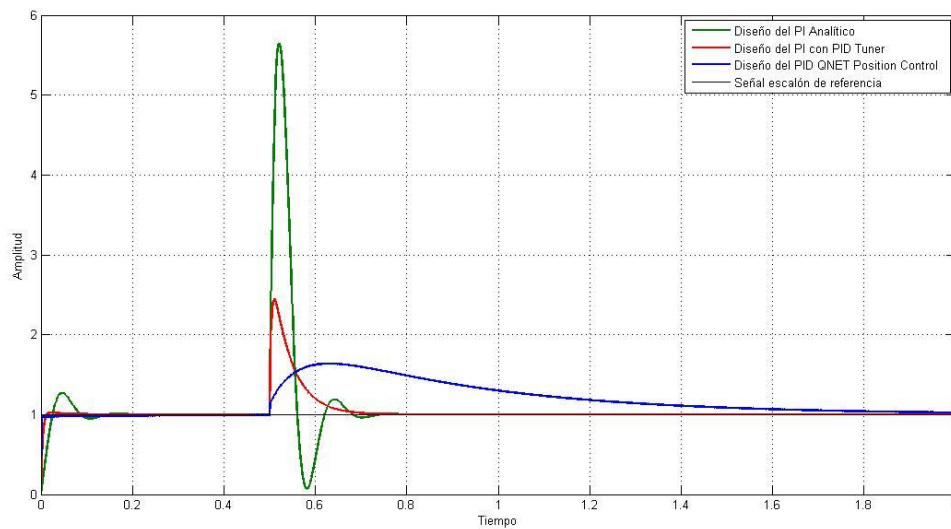


Figura 0-24: Curvas de respuesta utilizando los tres controladores según la tabla 4-4

Tabla 0-4: Resultados comparativos en los diseños de sintonización usando PID

Diseño del controlador	Overshoot	Tiempo de estabilidad
PI Analítico	5.6434	0.7417 seg.
PI usando PID Tuner	2.4475	0.7278 seg.
PID usando PID QNET Position Control	1.6348	1.994 seg.

La confiabilidad del desempeño del controlador no puede ser determinada con mucha precisión debido a que las pruebas fueron realizadas en Windows y LabVIEW, dando a ello son programas que requiere de muchos recursos en el procesador y no transmite la información en tiempo real (esto se vio de manera muy clara cuando se intentaba abrir o trabajar en otros programas al mismo tiempo que corría LabVIEW), también importante mencionar los problemas de recurso limitados de la PC hace que se incremente. Los drivers, que son procesadores que tienen como única tarea comunicarse con el sistema, en caso poder trabajo con ellos pueden cumplir con esta función y así obtener mejores resultados en el desempeño del controlador.



# CAPÍTULO V

## 5 CAPÍTULO 5

### 5.1 SISTEMA DIFUSO (FUZZY)

#### 5.1.1 Control de posición en el sistema DCMCT aplicando control difuso (Fuzzy)

La lógica difusa es una lógica alternativa a lo que es la lógica clásica, ya que esta permite trabajar con información la cual no es exacta, lo cual no ocurre con la lógica clásica o con un controlador PID donde se requiere de que la información sea exacta y definida.

Esta lógica fue implementada con el fin de emular en cierta forma el conocimiento humano, ya que el razonamiento del ser humano se relaciona con este tipo de información imprecisa para realizar ciertas actividades, un ejemplo de este tipo de razonamiento humano es cuando trata de describir la estatura de una persona, se acostumbra a decir alto, mediano o pequeño aunque estas descripciones no son sujetas a un valor exacto, sino que más bien se trata de asociarlas a un rango de valores.

En el año de 1965 Lofti Asker Zadeh hizo por primera vez la presentación de un trabajo sobre conjuntos difusos con un trabajo bajo el nombre de “*Fuzzy Sets*” (Conjuntos difusos), de modo que se le conoció a Zadeh como el padre del término Fuzzy (borroso). (Lorandi-Medina, Hermisa-Saba, Ladrón-de-Guevara-Durán, & Hernández-Silva, 2011)

Más tarde adoptarían estos conceptos de control para el desarrollo de máquinas de vapor, sistemas de procesos industriales, en proyectos de laboratorio y hasta en el ámbito del transporte. Es por ello que se continúa con las investigaciones en este campo en áreas como las redes neuronales y como se relacionan con los sistemas Fuzzy que puedan relacionarse en ambos métodos y por ello seguir desarrollando aplicaciones en diversos campos.

### 5.1.1.1 Conjuntos difusos

Un conjunto difuso es aquel que puede contener elementos con grados parciales de pertenecía, partiendo de la teoría clásica de conjuntos, definiendo al grado de pertenecía como un número real entre cero y uno.

### 5.1.1.2 Conjuntos clásicos

Los conjuntos clásicos no son más que una colección de elementos, que surgen de la necesidad del ser humano de clasificar objetos y conceptos.

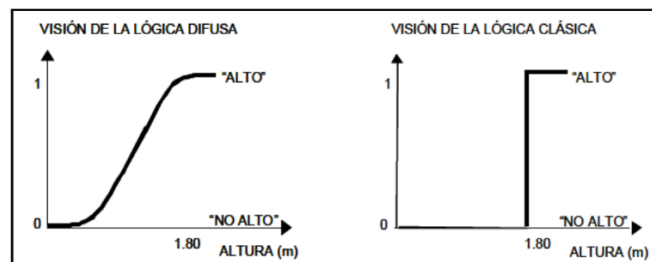


Figura 0-1: Lógica difusa (Fuzzy) contra lógica clásica

### 5.1.1.3 Las variables lingüísticas

Una variable lingüística como su nombre lo dice, es una variable cuyos valores son palabras o sentencias en un lenguaje natural, Por ejemplo, La velocidad de un coche, “Velocidad” es una variable lingüística si su valor es “muy lento”, “lento”, “normal”, “rápido” y “muy rápido”. También está el caso de la posición de sistema DCMCT, “Posición” es la variable lingüística y su valor es “lejos izquierda”, “cerca izquierda”, “muy cerca izquierda”, “centrado”, “muy cerca derecha”, “cerca derecha” y “lejos derecha”. De tal manera de saber en qué lugar se encuentra desbalanceada la viga.

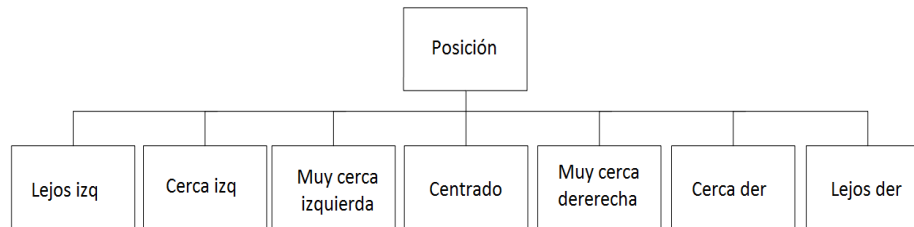


Figura 0-2: Valores lingüísticos de la variable difusa "Posición"

## 5.1.2 Diseño del Controlador Difuso (Fuzzy) aplicado en el sistema DCMCT de Quanser

### 5.1.2.1 Diseño en el vi utilizando herramientas de LabVIEW.

Para iniciar debemos de realizar un archivo o llamado vi utilizando el software de LabVIEW en el que comprenda la adquisición de datos en respuesta al sistema, para esto debemos de utilizar diferentes bloques de QNET ya elaboradas por Quanser como subVI's en propio a realizar. Se configura al ecoder para leer los grados de posición donde se ubica el motor de cd. Se utiliza el bloque de "dev" para poder seleccionar el canal que especifica la tarjeta de Quanser en donde capta la señal y la digitaliza. A su vez se enlaza con un bloque Quanser "Enc#1 config"; el cual configura los datos de entrada de la tarjeta dados por el encoder, para después enlazarlo o unirlo al bloque "Position Read" que los convierte en unos datos flotantes con el valor de posición del motor.

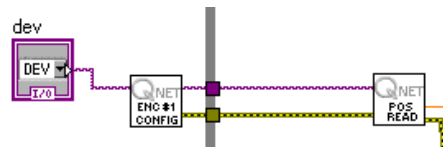


Figura 0-3: Configuración del canal de datos para posición

Este dato obtenido de la configuración de posición en el subVI de Quanser no arroja datos que debemos dimensionar con los grados en la superficie que rota los 360°. Esto lo logramos ajustando el valor flotante con un dato fijo al multiplicarlo con una constante de radianes que definimos. Para finalmente graficar el valor y visualizarlo a su vez con un indicador de posición como se muestra en la figura 5-4.



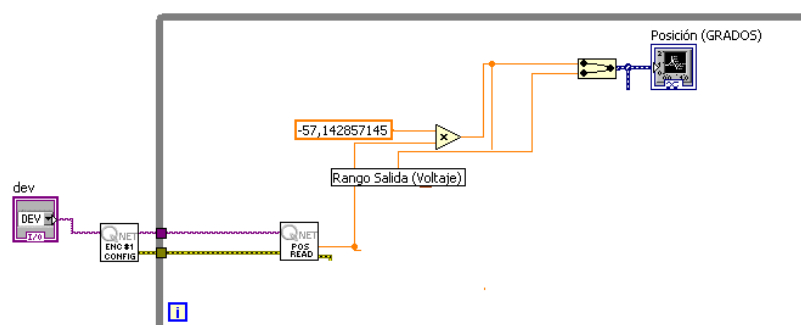


Figura 0-4: Diseño de la obtención de la posición con SubVis de QNET

En el diagrama de bloques es donde se ve la estructura de la posición del motor por medio de un indicador gráfico, en el podemos observar el arreglo del *Fuzzy Controller* para un sistema de una simple entrada y una simple salida tipo “SISO”, este demanda la señal de posición de entrada llamado “set point” en grados. Se debe controlar la proporción de salida para que no exceda al modelo sistema sienta un voltaje mucho mayor al que el propio motor del DCMCT está limitado, es por ello que se introduce un arreglo para delimitar los límites de señal de salida que son los mismos establecidos por el sub-vi de *QNet-Motor-Write*, este nos permitirá mandar la salida defusificación al QNET-DCMCT como se muestra en la figura 5-5. (Worldwide-Technical-Support, 2013)

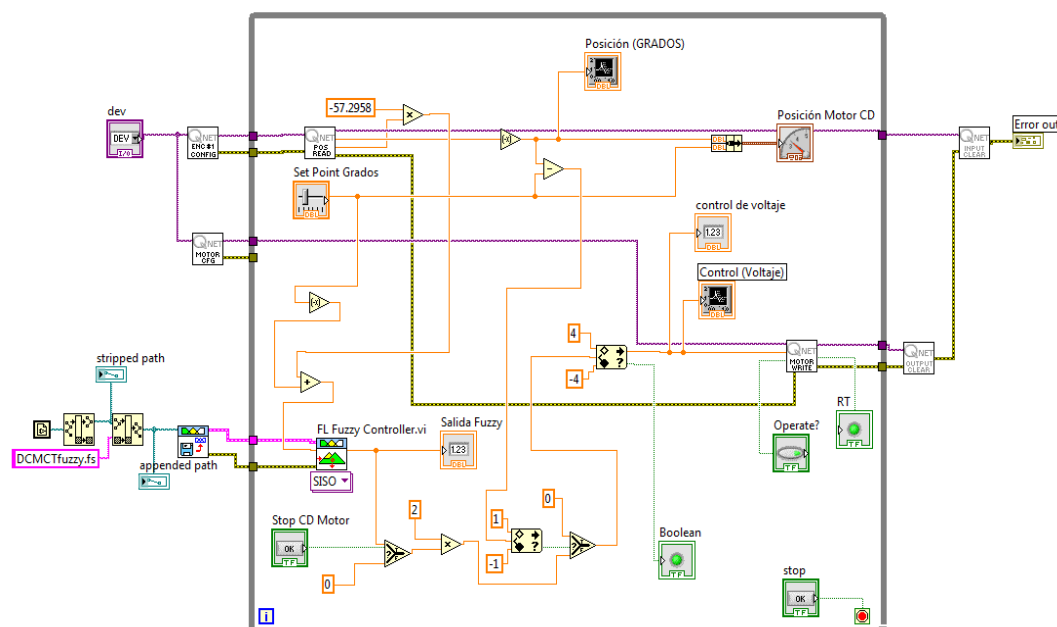


Figura 0-5: Diagrama de bloques del archivo *DCMCTFuzzy.vi*

Utilizando el archivo vi para funcionamiento en pruebas ejecutando *DCMCTFuzzy.vi*, el cual desplegara el panel frontal de operación para el sistema con acción de control con lógica Fuzzy como se muestra en la figura 5-6.

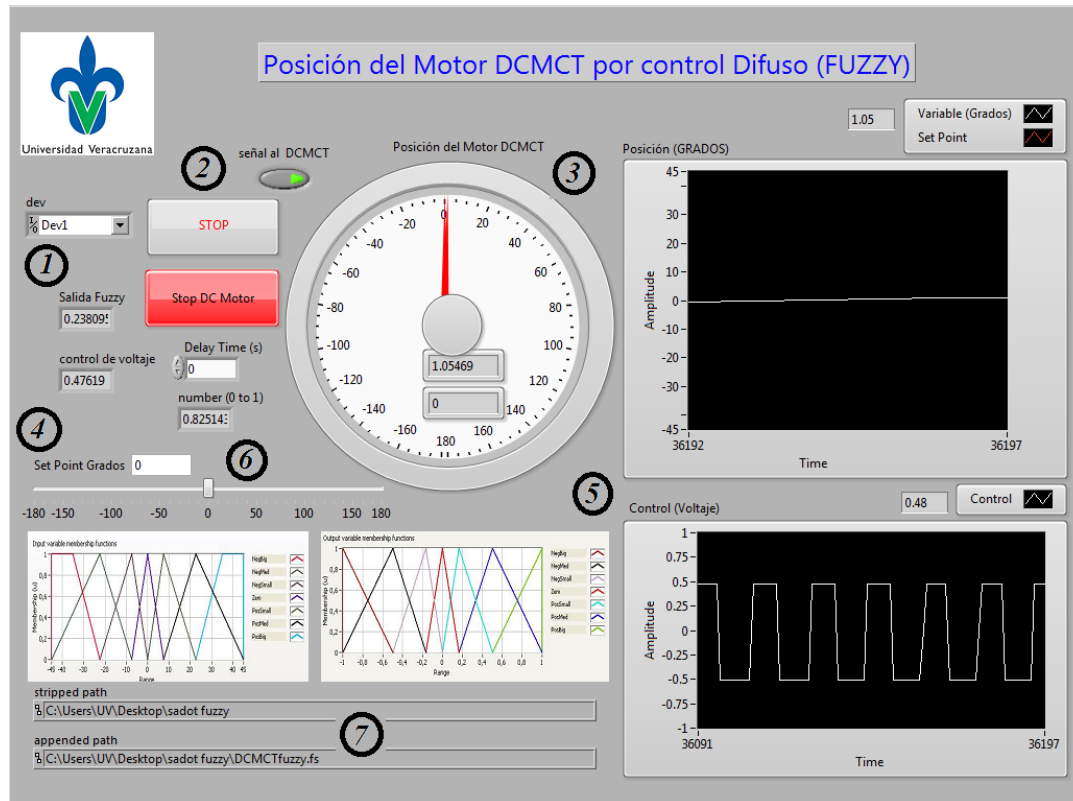


Figura 0-6: Panel frontal del archivo *DCMCTFuzzy.vi*

Tabla 0-1: Descripción del vi DCMCT Fuzzy

Número	Descripción
1	Selección del dispositivo de adquisición de datos (NI ELVIS II)
2	Botón de paro en la simulación del vi
3	Medidor de posición del motor DCMCT
4	Medidores de error, voltaje de salida y posición
5	Pantalla grafica de las señales de posición y control de voltaje de salida
6	Set Point del Sistema DCMCT
7	Ruta de archivo .fs de reglas de fusificación y des-fusificación del sistema

### 5.1.3 Herramientas de diseño en controladores difusos.

LabVIEW a través de su Toolkit de Lógica Difusa, permite instalar una herramienta poderosa para el diseño de controladores difusos. A esta interface virtual se accede a través la opción *Fuzzy Logic Controller Design*, ubicada en el menú Tools de LabVIEW, como se visualiza en la figura 5-7.

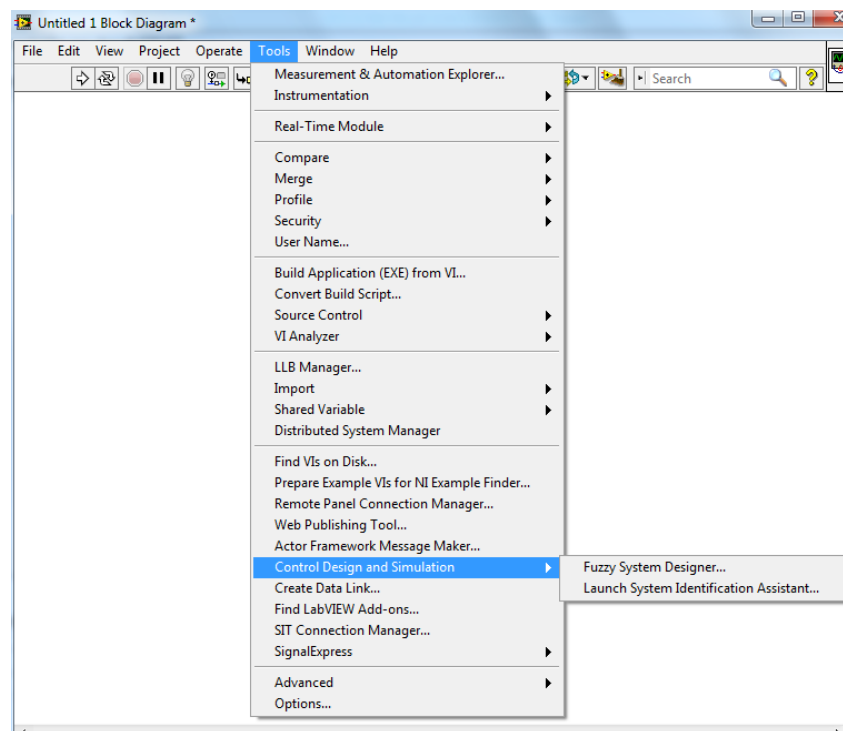


Figura 0-7: Modo de acceso a la herramienta para Control Difuso

El *Fuzzy Logic Controller Design VI*, consiste de las siguientes partes.

- Project Manager: Maneja el proyecto de lógica difusa.
- *Fuzzy Set Editor*: Define y modifica las variables lingüísticas, incluyendo sus términos lingüísticos.
- *Rule Base Editor*: Define y modifica la base de reglas del sistema difuso a ser diseñado.
- Utilitario para pruebas y simulación.

Cabe señalar que el Fuzzy Logic Controller Design VI, tiene las siguientes limitaciones:

- El máximo número de variables lingüísticas es cuatro.
- El máximo número de términos lingüísticos para cada variable lingüística es nueve.
- Los tipos de funciones de membresía están normalizada para tener la forma triangular, trapezoidal y singletons.

### 5.1.3.1 Diseño del *PROJECT MANAGER*.

Este archivo vi es una aplicación independiente con una interface de usuario gráfica para diseño y edición de un controlador difuso. Aunque no tiene entradas o salidas, puede ser utilizado como un subVI.

El panel frontal del Project Manager, tiene un campo para descripción del proyecto, donde se puede ingresar información importante referente al proyecto. Adicionalmente existe un campo para identificación del proyecto, en el cual el diseñador puede ingresar su nombre.

LabVIEW automáticamente llama al *Fuzzy Set Editor*, cuando se crea un nuevo proyecto.

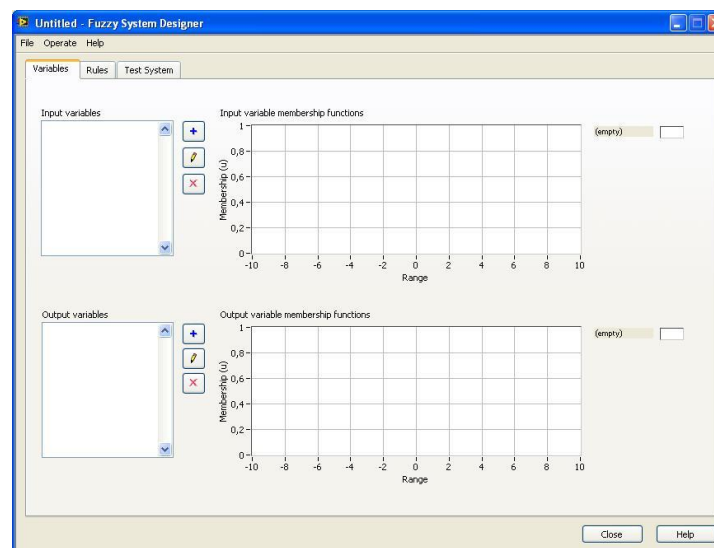


Figura 0-8: *Fuzzy System Designer* (Editor de variables lingüísticas)

### 5.1.3.2 Diseño de *FUZZY SET EDITOR*.

El Fuzzy-Set-Editor-Variable, permite crear y editar las funciones de membresía para cada una de las variables lingüísticas de entrada y salida (Figura 5-9).

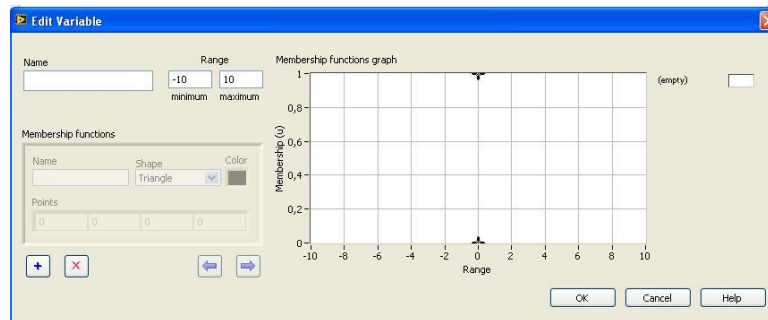


Figura 0-9: *Fuzzy Set Edit Variable*

Cada Nuevo proyecto, tiene ciertas configuraciones por defecto. El *Fuzzy Edit Variable* controla las modificaciones para los términos lingüísticos, con respecto a posibles restricciones, diseñamos dos funciones de membresías en las variables subdivididas de entrada, con la finalidad de comprar el desempeño del controlador en ambos casos. Los extremos de las variables se establecerán de manera trapezoidal para una mayor área de acción de respuesta en la integral de áreas más adelante establecidas por los criterios de antecedente y de grado de soporte en su membresías; del mismo modo las variables intermedias de manera triangular para lograr reducir área de acción y una estabilidad en la respuesta.

### 5.1.3.3 Definición de la base de reglas *RULES*.

Las reglas aplicadas requieren un conocimiento de la planta a ser controlada, es decir, la forma en que ésta se comportaría en caso de realizar un cambio en alguna de sus entradas. En este sentido, cualquier persona con un mínimo de conocimiento acerca de un proceso en particular podría generar una base de reglas para su controlador.

No existe una regla formal que indique el número de términos para cada variable lingüística que se debe usar, todo depende del diseñador y de los objetivos a alcanzar, otra observación es, que la mayoría de variables lingüísticas tienen un número impar de términos es debido al hecho que estos términos son definidos simétricamente, y un término debe describir la región media entre los dos extremos. (Lorandi-Medina, Hermisa-Saba, Ladrón-de-Guevara-Durán, & Hernández-Silva, 2011)

Para la elección del universo del discurso se tiene en cuenta el máximo ángulo de alcance del sistema es de  $(+80^\circ$  a  $-80^\circ)$  para obtener respuesta. Para el diseño se establecieron dos tipos de función en difuso, el primer diseño que se plantea realizar es utilizando siete funciones de membresía para el error  $e(t)$  y tres para la derivada del error  $e'(t)$ . Los términos lingüísticos asignados para cada uno de los nueve conjuntos difusos del error  $e(t)$ , se abrevian de la siguiente manera, NegBig, NegMed, NegSmall, Zero, PosSmall, PosMed, PosBig. Estos corresponden a las variables lingüísticas (en inglés) grande, medio y pequeño tanto positivo como negativo.

Para el segundo el diseño, reducimos el número de variables a cinco funciones de membresía para el error  $e(t)$  NegBig, NegMed, Zero, PosMed, PosBig; y mantendremos tres para la derivada del error  $e'(t)$ , esto queda definido como muestra la tabla 5-2 y las figuras 5-10 y 5-11.

Tabla 0-2: diseños del error de señal de posición.

Error Diseño 1 (7 variables)	Error Diseño 2 (5 variables)
<b>NegBig:</b> El error de entrada es Negativo Grande.	NegBig: El error de entrada es Negativo Grande.
<b>NegMed:</b> El error de entrada es Negativo Mediano.	NegSmall: El error de entrada es Negativo Pequeño.
<b>NegSmall:</b> El error de entrada es Negativo Pequeño.	Zero: El error de entrada es Cero, está en el Set Point.
<b>Zero:</b> El error de entrada es Cero, está en el Set Point.	PosSmall: El error de entrada es Positivo Pequeño.
<b>PosSmall:</b> El error de entrada es Positivo Pequeño.	PosBig: El error de entrada es Positivo Grande.
<b>PosMed:</b> El error de entrada es Positivo Mediano.	
<b>PosBig:</b> El error de entrada es Positivo Grande.	

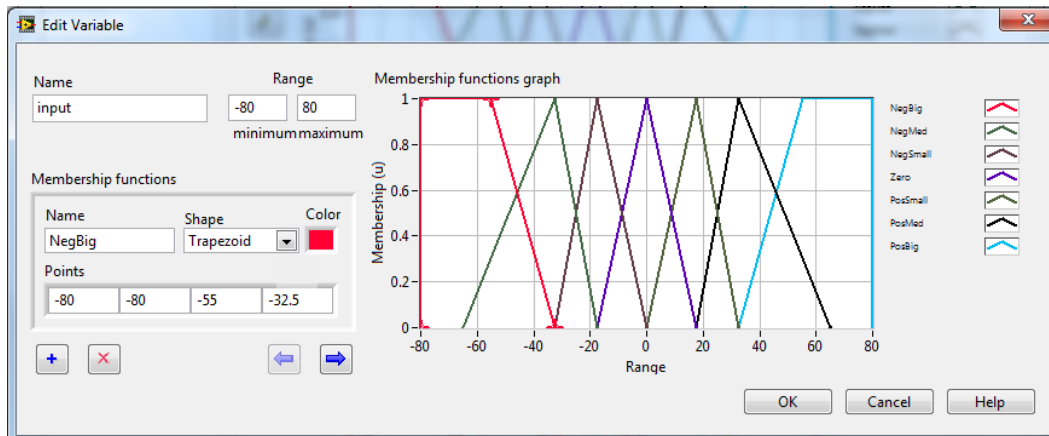


Figura 0-10: Variables de términos lingüísticos asignados del diseño 1 (7variables).

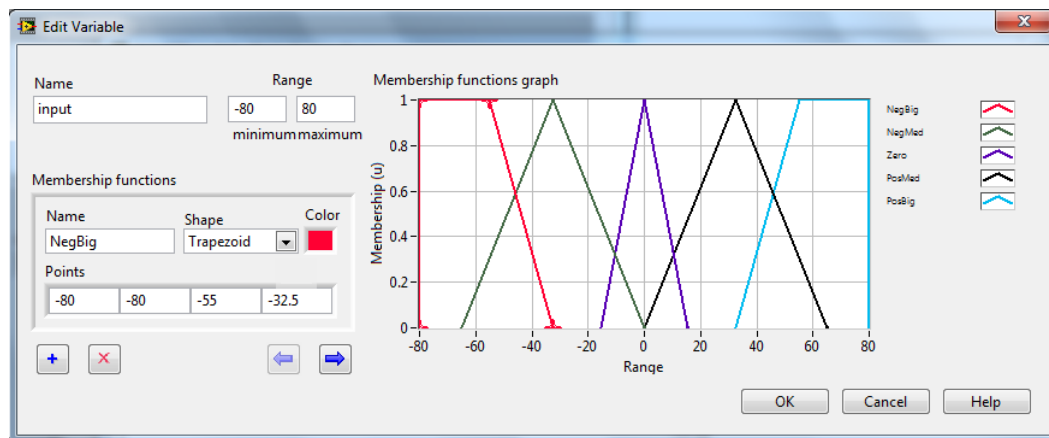


Figura 0-11: Variables de términos lingüísticos asignados del diseño 2 (5 Variables).

Los términos lingüísticos asignados para cada uno de los tres conjuntos difusos del error  $e'(t)$ , se abrevian con las letras N, Z y P que representan negativo, cero y positivo respectivamente.

Tabla 0-3: Los tres conjuntos difusos para la señal  $e'(t)$

Cambio Error
<b>Pos:</b> El cambio de error es Positivo.
<b>Zero:</b> El cambio de error es Cero.
<b>Neg:</b> El cambio de error es Negativo.

Esto queda establecido en el diseño de fusificación en ambos casos con las reglas establecidas.

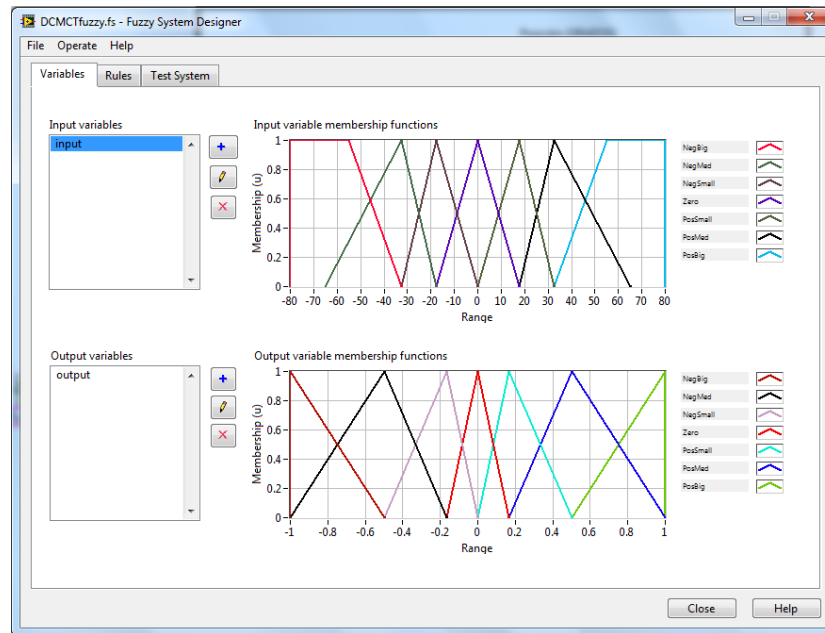


Figura 0-12: Grafica de funciones de membresía para la variable de entrada y salida del diseño para la variable de control (diseño 7 variables).

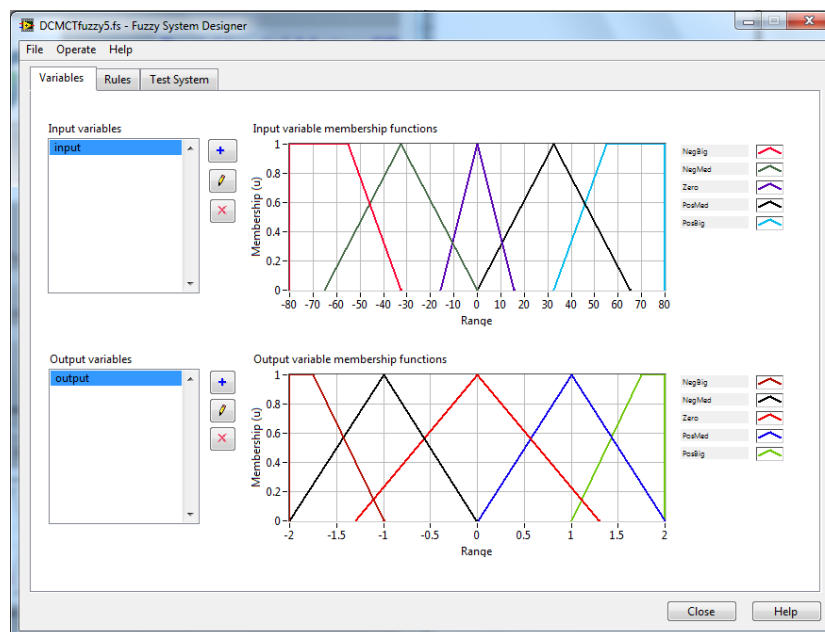


Figura 0-13: Grafica de funciones de membresía para la variable de entrada y salida del diseño para la variable de control (diseño 5 variables).



Así los nueve conjuntos difusos que se desprenden para la etapa de control en la señal de error en input (señal de posición) son:

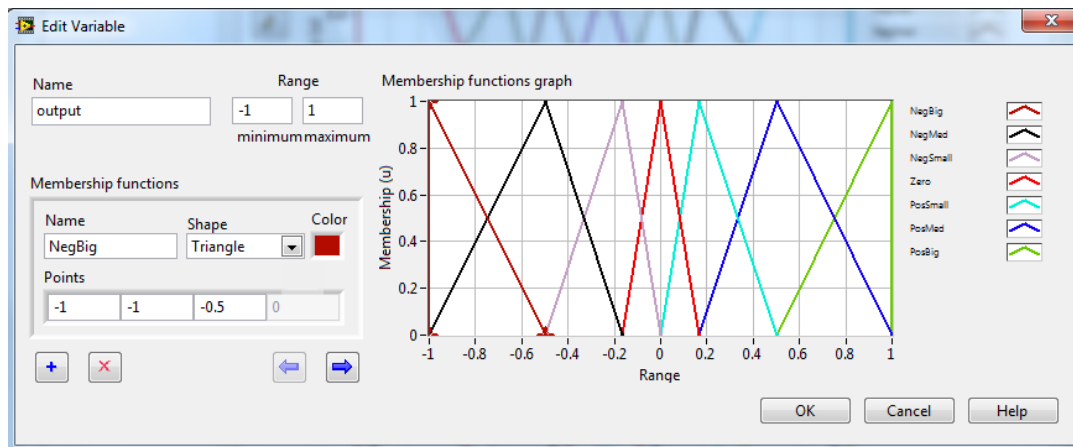


Figura 0-14: Grafica de funciones de membresía para la variable de salida diseño 1

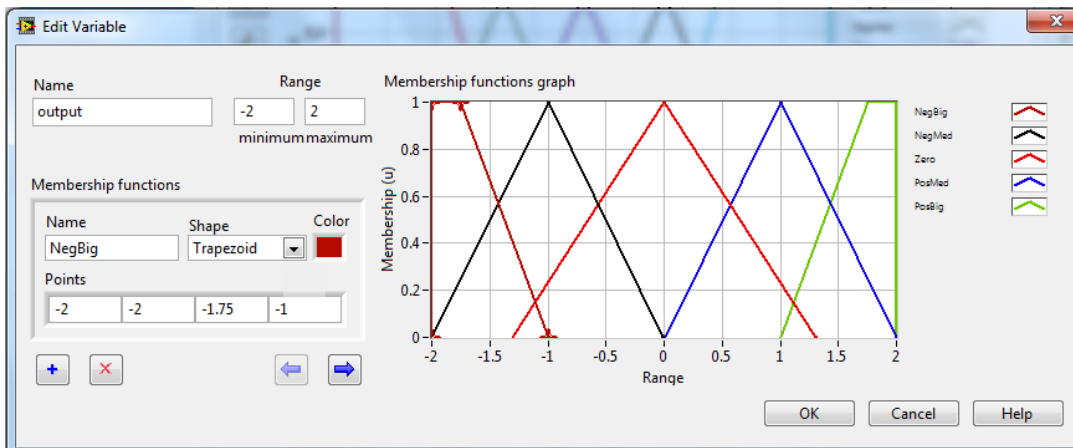


Figura 0-15: Grafica de funciones de membresía para la variable de salida diseño 2

Las membresías para las variables designadas en la fusificación las dejamos también estructuradas por el área de acción en la tabla 5-5 y del mismo modo en cada uno de los diseños de variables estipuladas se delimitar los rangos de acción para la defusificación, respetando el número de membrecías que se establecerán de manera automática en la base de reglas.

Tabla 0-4: Ahora las funciones de membresía para la variable de salida

Membrecía Diseño 1	Membrecías Diseño 2
<b>NegBig</b> : Incrementa al valor actual de salida, un valor Grande.	NegBig : Incrementa al valor actual de salida, un valor Grande.
<b>NegMed</b> : Incrementa al valor actual de salida, un valor Mediano.	NegMed : Incrementa al valor actual de salida, un valor Mediano.
<b>NegSmall</b> : Incrementa al valor actual de salida, un valor Pequeño.	Zero : Mantenga el valor de salida.
<b>Zero</b> : Mantenga el valor de salida.	PosMed : Decremento del valor actual de salida, un valor Mediano.
<b>PosSmall</b> : Decremento del valor actual de salida, un valor Pequeño.	PosBig: Decremento del valor actual de salida, un valor Grande.
<b>PosMed</b> : Decremento del valor actual de salida, un valor Mediano.	
<b>PosBig</b> : Decremento del valor actual de salida, un valor Grande.	

Después se procedió a delimitar la salida de control, determinando los rangos de respuesta. Para esta operación se determinó un rango de -1 hasta 1 para poder controlar el cambio de posición del DCMCT con la carga permanente ya acoplada al sistema. Esto nos permitirá no estar fuera del rango de acción y que se establecerá en las reglas de controlador.

Es importante que la delimitación de las proporciones en las reglas antes establecidas, donde las formas de las funciones fueron tipo triángulo en segmentado en la respuesta al ángulo que delimite la respuesta cada lado del círculo graduado de control del DCMCT, de tal manera que alcanzara una respuesta satisfactoria.

Se utiliza de modo práctico una diferenciación de colores en cada una de las áreas de respuesta, tal manera de reconocer en que membresía en que se encuentra cada variable de control establecida para ambos diseños.

### 5.1.4 Tipo y diseño de las funciones de membresía.

El tipo de funciones de membresía escogido es la función triangular, debido a que ésta presenta mayor facilidad para ser llevada a un sistema digital como es un CPU, por la simplicidad de su descripción. Así que las funciones de membresía para el error  $e'(t)$  y la salida  $u'(t)$ , se muestran en la figura 5-9 respectivamente.

La tabla 5-7 establece la base de las reglas para la realización del control de velocidad del QNET-DCMCT. La primera fila y la primera columna indican los conjuntos difusos de la variable E y de la variable CE, respectivamente, y las funciones de pertenencia de la variable de salida CU es mostrada en el cuerpo de la tabla.

Existen  $7 \times 3 = 21$  reglas posibles para el diseño 1 de la tabla 5-5 y que se pueden realizar con los conjuntos difusos de entrada, donde la lectura de un valor actúa de la forma que  $E = \text{NegBig}$  y  $S = \text{Zero}$ , entonces la respuesta será "*PosBig*".

Tabla 0-5: Cuadro de reglas diseño 1 establecidas para el control difuso

Entrada	NegBig	NegMed	NegSmall	Zero	PosSmall	PosMed	PosBig
Salida							
Negativa	PosBig	PosMed	PosSmall	Zero	NegSmall	NegMed	NegBig
Zero	PosBig	PosMed	PosSmall	Zero	NegSmall	NegMed	NegBig
Positiva	PosBig	PosMed	PosSmall	Zero	NegSmall	NegMed	NegBig

Existen  $5 \times 3 = 15$  reglas posibles para el diseño 2, de la tabla 5-6 y que se pueden realizar con los conjuntos difusos de entrada respectivamente, de la misma forma que diseño anterior.

Tabla 0-6: Cuadro de reglas diseño 2 establecidas para el control difuso

Entrada	NegBig	NegMed	Zero	PosMed	PosBig
Salida					
Negativa	PosBig	PosMed	Zero	NegMed	NegBig
Zero	PosBig	PosMed	Zero	NegMed	NegBig
Positiva	PosBig	PosMed	Zero	NegMed	NegBig

Se puede dejar sin reglas algunos bloques de la tabla 5-6, para que los valores de salida sean activados con solo algunos conjuntos difusos de entrada.

### 5.1.5 Identificación del método de defusificación.

El objetivo de un método de defusificación es derivar un valor no difuso, que refleje mejor los valores difusos de la variable lingüística de salida. Los resultados de dichos métodos son un área final, como resultado de un conjunto de áreas solapadas entre sí (cada área es resultado de una regla de inferencia). Para éste controlador se utiliza el método de defusificación de centroide, que es el método más usado en aplicaciones difusas, donde la salida final será el centro de gravedad del área total resultante, además contribuye a que la salida se mueva más suavemente a lo largo de la superficie de control, que además, en la simulación, brinda los mejores resultados.

#### 5.1.5.1 Programa de control

En este programa se procesa la señal analógica de entrada, que lo proporciona el encoder y los Subvi de Quanser utilizados para que la señal se traduzca en datos de entrada por el puerto *dev01*, que es uno de los puertos analógicos de entrada de dicha tarjeta. Este programa luego de procesar la señal, entrega una señal analógica de 0 a 5v, que es enviada al exterior a través del puerto analógico de salida *a0* de la misma tarjeta.

#### 5.1.5.2 RULEBASE EDITOR (Editor de Reglas).

El *Rulebase Editor*, es el Editor de reglas o *Rules* permite ingresar la base de reglas, cuya dimensión está determinada por la cantidad de variables y términos lingüísticos de entrada. Cada regla está asociada con un factor de peso, para acentuar o reducir la influencia de una determinada regla en las características del controlador. En las figuras 5-16 y 5-17, se muestra la base de reglas para un ejemplo determinado.

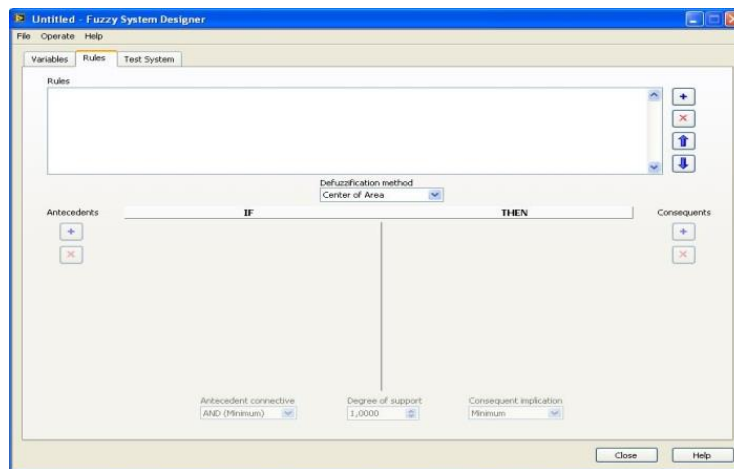


Figura 0-16: *Rulebase Editor*

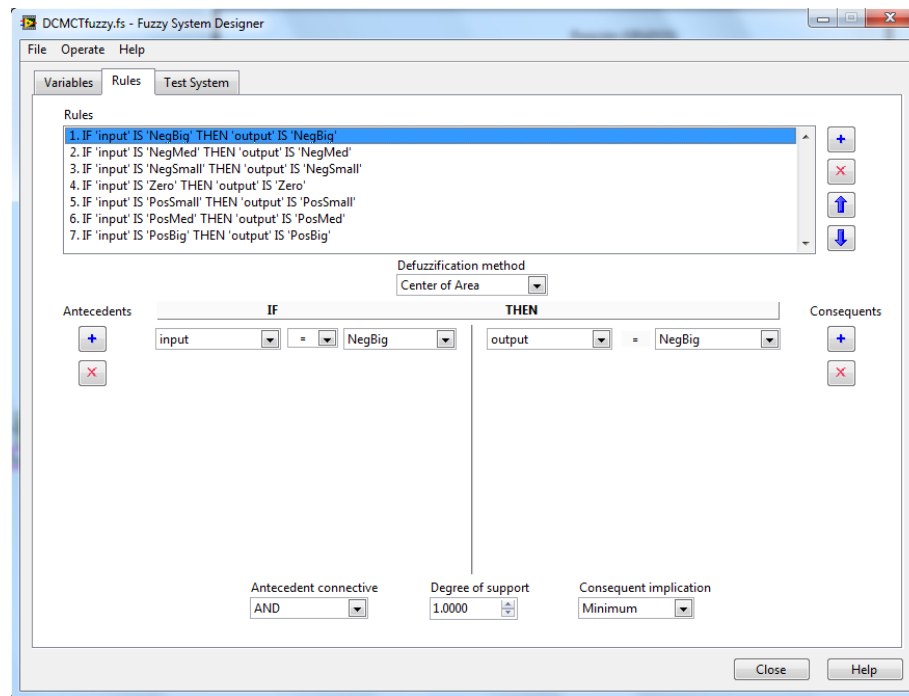


Figura 0-17: Reglas Aplicadas en *Fuzzy System Designer* diseño 1 (7 Variables)

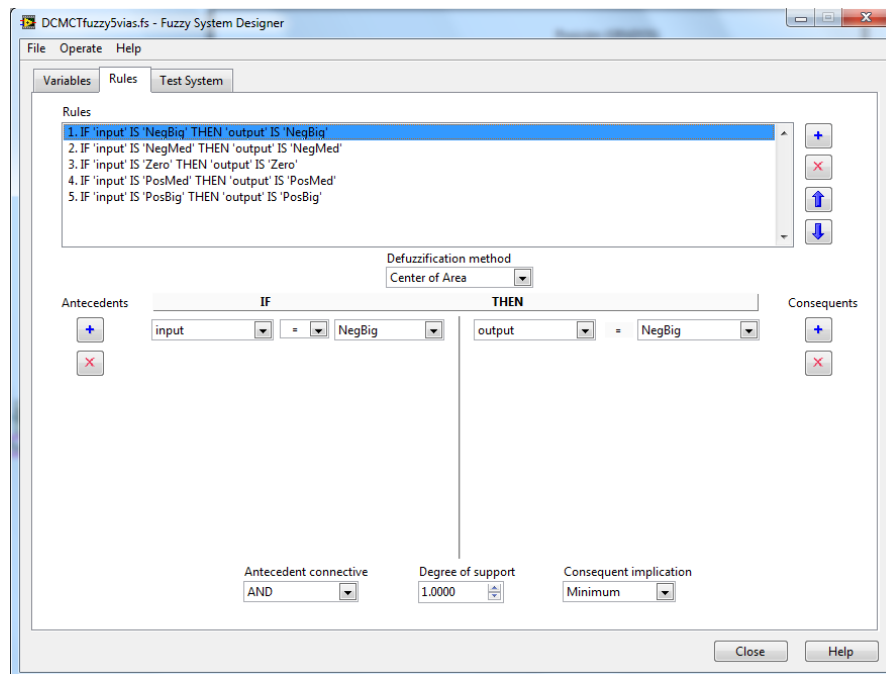


Figura 0-18: Reglas Aplicadas en *Fuzzy System Designer* diseño 2 (5 Variables)

Una vez definidas las variables difusas y sus respectivos valores lingüísticos, es posible expresar el modelo cualitativo del controlador difuso mediante el conjunto de reglas que las relacionan.

Tabla 0-7: Referencias y base de reglas diseño 1 (7 variables).

N°	IF (en caso)	THEN (entonces)
1	IF “input” IS “NegBig”	THEN “output” IS “NegBig” [AND(mín.), Degree of support = 1.0]
2	IF “input” IS “NegMed”	THEN “output” IS “NegMed” [AND(mín.), Degree of support = 0.75]
3	IF “input” IS “NegSmall”	THEN “output” IS “NegSmall” [AND(mín.), Degree of support = 0.5]
4	IF “input” IS “Zero”	THEN “output” IS “Zero” [AND(mín.), Degree of support=0.0]
5	IF “input” IS “PosSmall”	THEN “output” IS “PosSmall” [AND(mín.), Degree of support = 0.5]
6	IF “input” IS “PosMed”	THEN “output” IS “PosMed” [AND(mín.), Degree of support = 0.75]
7	IF “input” IS “PosBig”	THEN “output” IS “PosBig” [AND(mín.), Degree of support = 1.0]

Tabla 0-8: Referencias y base de reglas diseño 2 (5 variables).

N°	IF (en caso)	THEN (entonces)
1	IF “input” IS “NegBig”	THEN “output” IS “NegBig” [AND(mín.), Degree of support = 1]
3	IF “input” IS “NegSmall”	THEN “output” IS “NegSmall” [AND(mín.), Degree of support = 0.5]
4	IF “input” IS “Zero”	THEN “output” IS “Zero” [AND(mín.), Degree of support=0.0]
5	IF “input” IS “PosSmall”	THEN “output” IS “PosSmall” [AND(mín.), Degree of support = 0.5]
7	IF “input” IS “PosBig”	THEN “output” IS “PosBig” [AND(mín.), Degree of support = 0.1]

Se puede especificar un grado de apoyo, entre 0 y 1, para cada regla de un sistema difuso. El grado de apoyo que representa la importancia relativa de cada regla y permite la puesta a punto de la base de reglas. En la mayoría de los casos, el grado de apoyo es de 1. El peso final de una regla es igual al grado de apoyo multiplica por el valor de verdad del antecedente regla agregada.

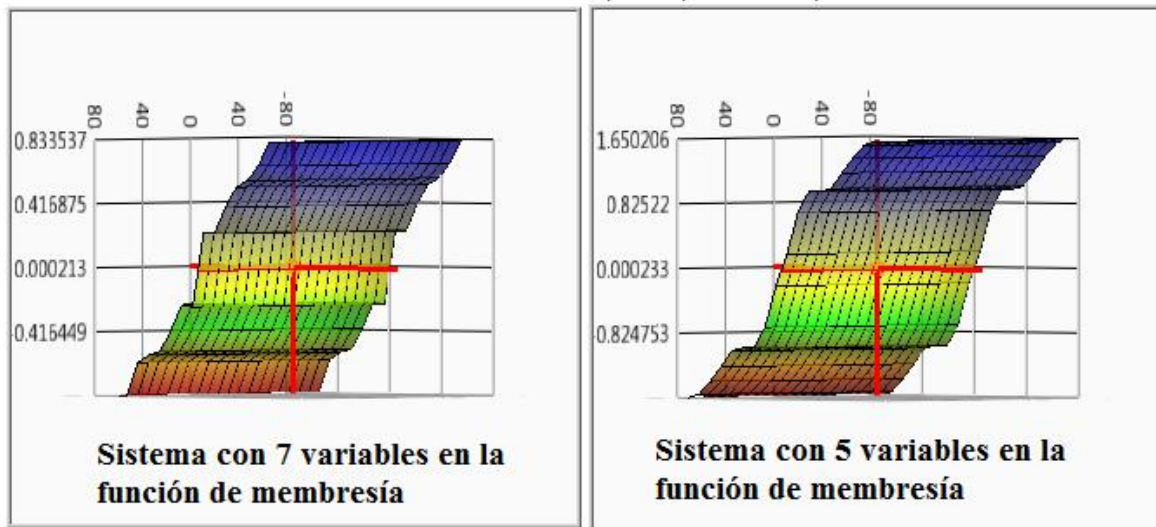


Figura 0-19: Salidas de los controladores Fuzzy en función de las variables de entrada.

### 5.1.6 Desempeño en operación de los diseños de control utilizando Fuzzy.

Debido a que el universo de la señal de estabilidad es de  $\pm 80$ , lo que se basa en la suposición de que el sistema de laboratorio QNET-DCMCT no se mueve a las de  $80^\circ$ . Se emplearon en este caso funciones de pertenencia gaussianas con desviaciones establecidas y medias de -80, 0 y 80 en el diseño.

Mediante la captación en graficas de la respuesta, podemos analizar los modelos de los dos sistemas Fuzzy, observar el comportamiento y el mejor desempeño de los dos modelos planteados que cuentan cuenta con siete y cinco variables de membresía para ubicar la posición y el grado de corrección para el error.

El análisis de los resultados a la respuesta aplicándole una carga estática permanente al sistema DCMCT aplicando los dos diseños de controlador Fuzzy para observar el desempeño obtenido por cada uno de ellos se puede observar en las figura 5-20 y 5-21.

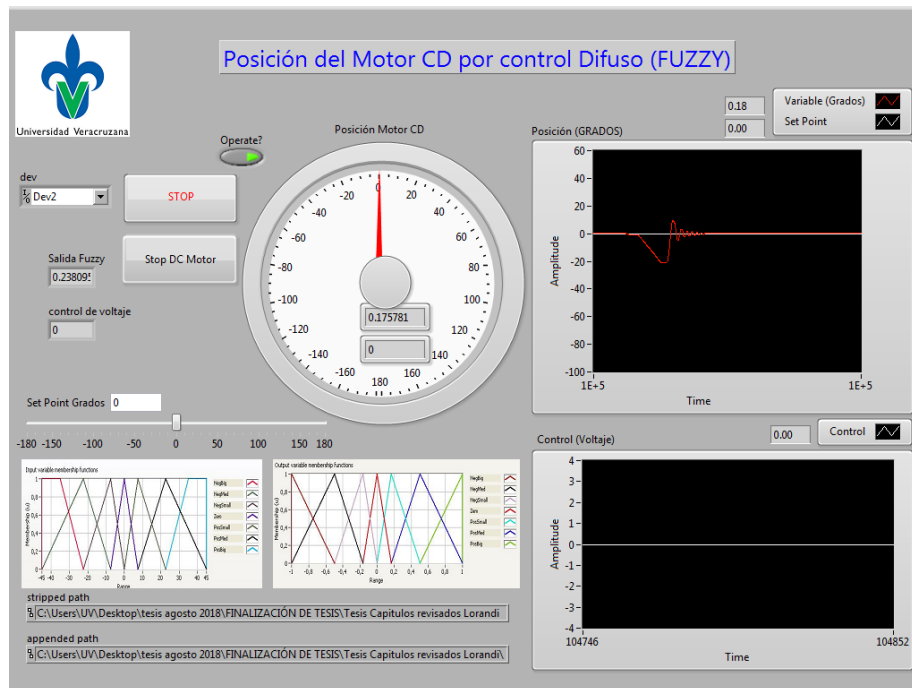


Figura 0-20: Respuesta al diseño 1 (7 variables) sin carga permanente al sistema DCMCT

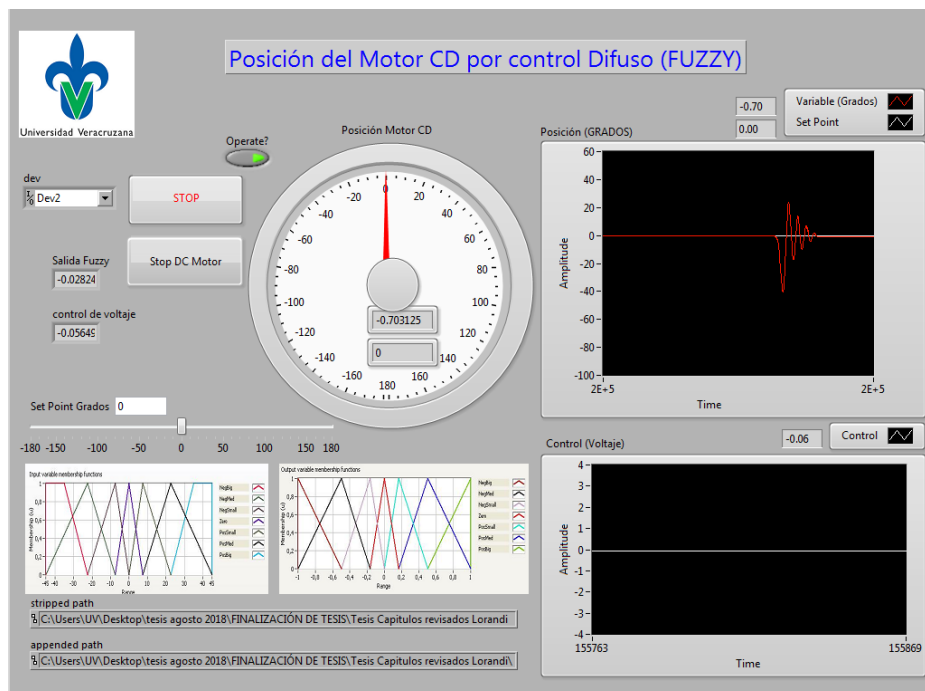


Figura 0-21: Respuesta al diseño 1 (5 variables) sin carga permanente al sistema DCMCT

Del mismo modo se le aplico la carga permanente al sistema DCMCT y se observó los resultados de comportamiento y el tiempo de estabilización por los dos diseños antes establecidos.



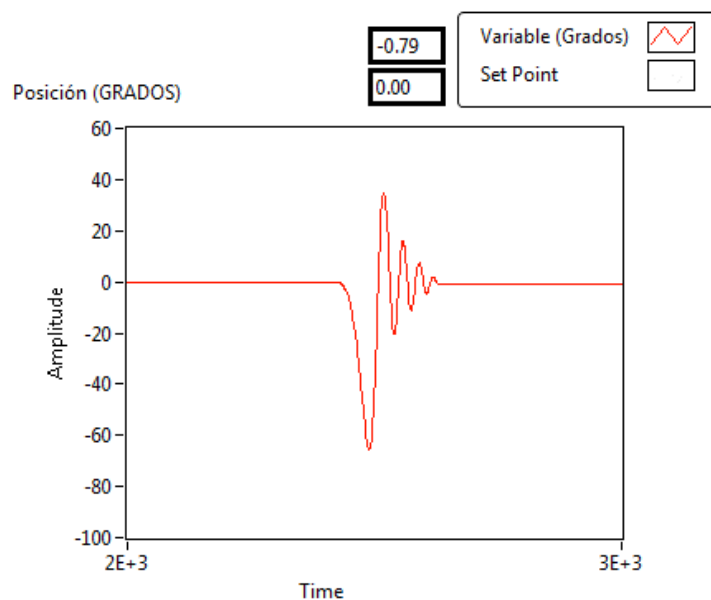


Figura 0-22: Respuesta con carga permanente al sistema DCMCT utilizando control Fuzzy

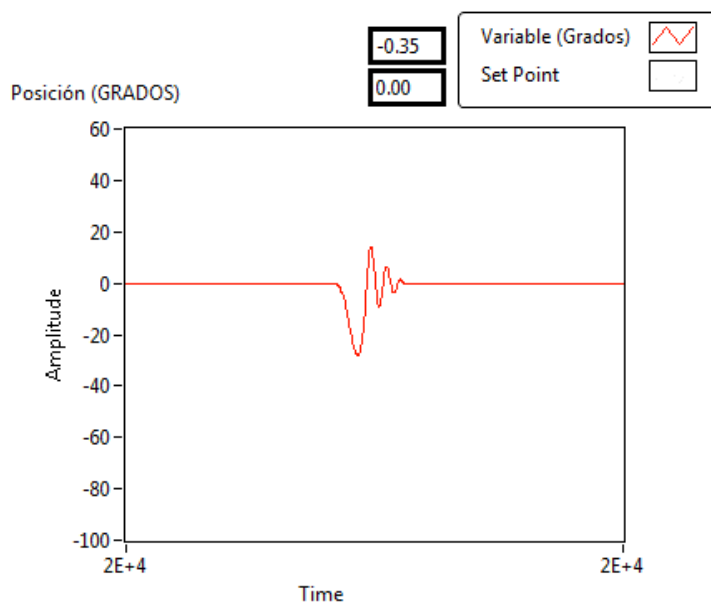


Figura 0-23: Respuesta con carga permanente al sistema DCMCT utilizando control Fuzzy

Queda establecido como un mejor comportamiento de desempeño en el tiempo de estabilización con la carga estática permanente aplicada al sistema, el diseño aplicado con 5 variables de membresías en la señal de entrada, el cual proporciona una mejor estabilidad a la referencia del setpoint.

## 6 CONCLUSIONES

El desarrollo de este trabajo de investigación permitió implementar y sustentar estrategias de control PID y Fuzzy aplicadas al sistema DCMCT de forma práctica y que de modo que se puedan lograr obtener en medida aportación al conjunto de experiencias de aprendizaje utilizadas. Los resultados fueron satisfactorios desde los objetivos planteados, estableciéndose una serie de casos aplicativos que pueden ser repicados para el uso y pruebas al utilizar el equipo QNET-DCMCT. Es importante destacar que al utilizar la interface de la tarjeta de adquisición del NI-ELVIS II permite el desarrollo de estrategias de modelado matemático del sistema, así como de identificación, control y simulación en el motor esto usando los softwares de Matlab y LabVIEW en comunicación con la tarjeta ELVIS II. Del mismo modo se abordó esquemáticamente el determinar una serie de pasos a razón de prácticas para el fáciles manejo de los dispositivos y puedan realizar el uso de lo aprendido la experiencias educativas de sistemas control que estén cursando en la DES del área técnica de la universidad veracruzana región Veracruz.

## 7 TRABAJOS FUTUROS

Es importante comentar que se podrían realizar más pruebas en el mismo prototipo NI-DAQ y QNET-DCMCT de Quanser, Estables una serie de estrategias más actuales y bajo plataformas que ayuden a los estudiantes acceder a ellas y que sean de más fácil manejo sin utilizar el básico manual operativo, esto promoviendo la creatividad para el desarrollo de adquirir el conocimiento y desarrollando experimentos de laboratorio o practicas aplicativas con equipo didáctico con el que se cuente, Quanser cuenta con una alta gama para poder ser aplicado a ello. Es importante establecer en el diseño y en las estrategias de modelado del sistema el utilizar un tiempo de retardo de respuesta, y así poder establecer mejores métodos de sintonización en el controlador PID y criterios para comprobar el desempeño de los mismos, del mismo modo utilizar otras formas de funciones que sean de diferentes tipos en las membresías de las respuesta al cambio de la posición, de tal manera de poder ver si mejora o no la respuesta y si a su vez es satisfactoria. Analizar diferentes opciones de aplicación del control difuso o algún otro método de control. Establecer un sistema de cargas más controladas y acumulativas en la perturbación permanente del sistema, así como el llegar a realizar un cambio del motor de cd, para tener un sistema diferente de acoplamiento de carga y poder ser utilizado mediante otro software que no sea los archivos de control establecidos de Quanser para LabVIEW.

## 8 REFERENCIAS

- Hernandes, C., Peza, E., Garcia, E., & Torres, J. (2015). Simulación de los Sistemas de Control y Proporcional Derivativo y Proporcional Integral de un Servomecanismo de segundo orden bajo la plataforma de LabVIEW. *Revista de Tecnología e Innovación*, 690-693.
- Karl-Johan, Á., Apkarian, J., & Falcon, J. (2011). *Instructor Workbook Qnet DC Motor Control Trainer for NI ELVIS*. Markham, Ontario: Quanser Inc.
- Kuo, B. C. (1996). *Sistema de Control Automático* (Vol. Séptima Edición). Nahucalpan de Juárez, Edo. de México: Prentice Hall Hispanoamericano, S.A.
- Lewis, J. (2018). Tuning PID systems. *Automatic Control*, 20(12), 22-30.
- Lorandi-Medina, A.-P., Hermisa-Saba, G., Ladrón-de-Guevara-Durán, E., & Hernández-Silva, J. (2011). Controladores PID y Controladores Difuso. *Revista de la ingeniería Industrial, Academia Journals*, 5(1), 2-4.
- Quanser, T. S. (2008). DC Motor Control Trainer (DCMCT) User Manual. *Quanser Innovate Educate*, 1-5.
- Turevskiy, A. (2018). PID Control Design with Control System Toolbox. *MathWorks*, 1. Obtenido de <https://es.mathworks.com/videos/pid-control-design-with-control-system-toolbox-68748.html>
- Worldwide-Technical-Support, P.-a. (2013). Getting Starte with LabVIEW. 4-1 a 5-3.