



UNIVERSIDAD CATÓLICA
DE SANTIAGO DE GUAYAQUIL

TESIS DE GRADO

PREVIO A LA OBTENCIÓN DEL TÍTULO DE
INGENIEROS EN TELECOMUNICACIONES

TEMA:

DISEÑO E IMPLEMENTACIÓN DE ENTRENADORES
ELECTRÓNICOS INDUSTRIALES ORIENTADOS A PRÁCTICAS
EN CONTROL Y AUTOMATIZACIÓN ELECTRÓNICO

REALIZADO POR:

JHONNY ANDRES ZÚÑIGA CENTENO
JHONNY ENRIQUE BARRERA SILVA
ABEL ISAAC CASTILLO FLORES

DIRECTOR:

ING. MARCOS ANDRADE REYES

Guayaquil – Ecuador
Marzo del 2010

FACULTAD DE EDUCACIÓN TÉCNICA PARA EL DESARROLLO

TESIS DE GRADO

TÍTULO:

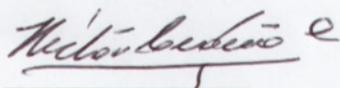
DISEÑO E IMPLEMENTACIÓN DE ENTRENADORES ELECTRÓNICOS
INDUSTRIALES ORIENTADOS A PRÁCTICAS EN CONTROL Y
AUTOMATIZACIÓN ELECTRÓNICO

PRESENTADA A LA FACULTAD DE EDUCACIÓN TÉCNICA PARA EL
DESARROLLO, DE LA UNIVERSIDAD CATÓLICA DE SANTIAGO DE GUAYAQUIL

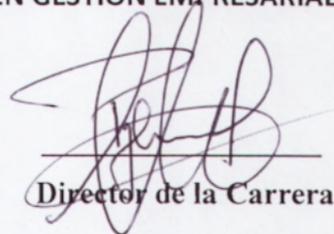
POR:

JHONNY ANDRÉS ZÚÑIGA CENTENO
JHONNY ENRIQUE BARRERA SILVA
ABEL ISAAC CASTILLO FLORES

PARA DAR CUMPLIMIENTO CON UNO DE LOS REQUISITOS PARA OPTAR POR EL TÍTULO
DE: INGENIERO EN TELECOMUNICACIONES CON MENCIÓN EN GESTIÓN EMPRESARIAL



Decano



Director de la Carrera

Director de Tesis

Vocal Principal

Vocal Principal

Director de Tesis

DEDICATORIA

A **DIOS**, porque gracias a su Amor Infinito, nos hace saber que todo es posible, y que cuando se quiere se puede siempre que se haga de corazón. A mis **PADRES Carmen y Jhonny**, quienes en todo momento me han dado su amor, estímulo y esfuerzo para salir adelante en toda mi vida. A mi **FAMILIA**, quienes en todo momento me han dado su apoyo y cariño, lo que me hace seguir adelante.

Jhonny Andrés Zúñiga Centeno

A **DIOS** por ser quien ha estado a mi lado en todo momento dándome las fuerzas necesarias para continuar luchando día tras día y seguir adelante rompiendo todas las barreras que se me presente. Dedico también a mi mamá **Elvia Flores** y a mi **FAMILIA** ya que gracias a ellos soy quien soy hoy en día.

Abel Isaac Castillo Flores

Quiero dedicar este trabajo a **DIOS** por la bendición de terminar mi carrera. A mis **PADRES Melanea y Homero**, por todo lo que me han dado en esta vida, especialmente por sus sabios consejos y por estar a mi lado en los momentos difíciles. A mi hermana **Viviana**, quien me ha acompañado en silencio con una comprensión a prueba de todo. A mi "Mamita" **Guillermina y Luz María** quien con su simpleza me ha ayudado a encontrar la luz cuando todo es oscuridad. A mis tíos Manolo y Fanny por estar siempre dispuestos para ayudarme.

Jhonny Enrique Barrera Silva

AGRADECIMIENTO

A **DIOS**, por darme la bendición y oportunidad de poder alcanzar mis metas. A mis **PADRES**, por guiarme y darme el amor y confianza en todo momento y así alcanzar mis sueños. Gracias por su amor, los amo. A toda mi **FAMILIA**, porque siempre han estado allí, brindándome en todo momento el apoyo. A mis **AMIGOS**, por brindarme su apoyo, cariño y solidaridad a lo largo de toda mi vida. A mi **TUTOR** de Tesis el Ing. Marcos Andrade, por el soporte brindado para elaborar este documento.

Jhonny Andrés Zúñiga Centeno

Primero y antes que nada, dar gracias a **DIOS**, por estar conmigo en cada paso que doy, por fortalecer mi corazón e iluminar mi mente y por haber puesto en mi camino a aquellas personas que han sido mi soporte y compañía durante todo el periodo de estudio. Agradecer hoy y siempre a mi madre **ELVIA** porque a pesar de no estar presente conmigo, siempre se ha preocupado por mí desde España. A mis hermanos que desde aquí han estado siempre apoyándome en todo.

Abel Isaac Castillo Flores

Agradezco a **DIOS**, mi señor, mi guía, mi proveedor, mi fin último. A mis hermanos, mis padres, por darme la estabilidad emocional, económica, sentimental; para poder llegar hasta este logro, que definitivamente no hubiese podido ser realidad sin ustedes. A mi fabuloso equipo de tesis; Jhonny, Abel a Marcos nuestro tutor que ha sido un pilar en el desarrollo de esta tesis, gracias a todos por alcanzar este sueño, que ahora es una realidad.

Jhonny Enrique Barrera Silva

RESUMEN

El presente estudio está dirigido a diseñar e implantar un grupo de (EPC) Entrenadores Electrónicos para Procesos de Control Industrial, los cuales son controlados mediante programación gráfica basada en la herramienta LabVIEW.

Estos entrenadores tienen como finalidad dotar al laboratorio de electrónica de herramientas de estudio, aportando a la capacitación de los estudiantes en controles industriales, permitiendo simular el proceso de control con LabVIEW de una planta industrial a gran escala.

Los Entrenadores Electrónicos para Procesos de Control Industrial, también permiten convergir el aprendizaje de distintas carreras como Ing. en Telecomunicaciones, Ing. Electromecánica e Ing. Agrícola.

ÍNDICE GENERAL

DEDICATORIAS	III
AGRADECIMIENTOS	IV
RESUMEN	V
INTRODUCCIÓN	1
CAPÍTULO I: PROBLEMA DE INVESTIGACIÓN	
1.1 Planteamiento del Problema	3
1.1.1 Formulación del Problema	4
1.1.2 Sistematización de la variable	4
1.2 Objetivos	4
1.2.1 Generales	4
1.2.2 Específicos	4
1.3 Justificación	5
1.4 Delimitación	5
1.4.1 Espacial	5
1.4.2 Temporal	5
CAPÍTULO II: MARCO TEÓRICO	
2.1 Antecedentes de la Investigación	6
2.1.1 Con qué cuenta el laboratorio electrónica	6
2.2 Bases Teóricas de Control	6
2.2.1 Sistema de Control	6
2.2.2 Tipos de Señales y Sistemas	7
2.3 Sistema de Control PID	9
2.3.1 Controlador PID	9
2.3.2 Fundamentos de Controladores PID	11
2.3.3 Principio de Realimentación	13
2.3.4 Análisis estático de los Sistemas Realimentados	16
2.3.5 Control PID	17
2.3.6 Parámetros PID	18
2.3.7 Implementación Digital	27
2.3.8 Aspectos Operacionales	29
2.3.9 Modelos de Simulación	30
2.3.10 Control PID de Temperatura	34
CAPÍTULO III: PLANTA DE ENTRENAMIENTO	
3.1 Concepto EPC	39
3.2 Funciones	40
3.2.1 Medición de Control de Temperatura	40
3.2.2 Medición de Control de Velocidad	41
3.2.3 Control de Posición	42

3.2.4	Análisis de Sonido	42
3.2.5	Relé de Propósito General	42
3.3	Diagrama Esquemático de la Planta de Entrenamiento	43
3.4	Diagrama Topológico y Conexiones Básicas	46
3.5	Diseño Final Propuesto EPC para la FETD	48
3.6	Diseño Electrónico propuesto EPC para la FETD	49

CAPÍTULO IV: PRÁCTICAS PARA EL ENTRENADOR EPC

4.1	Introducción a Prácticas EPC	50
4.2	Práctica de Control y Medición de Temperatura	50
4.2.1	Práctica T1: Medición de Temperatura	51
4.2.2	Práctica T2: Filtrado de Señal de Temperatura	55
4.2.3	Práctica T3: Control de Temperatura Manual	57
4.2.4	Práctica T4: Control de Temperatura On/Off	63
4.2.5	Práctica T5: Control de Temperatura PID	67
4.3	Prácticas de Medición y Control de Velocidad	74
4.3.1	Práctica M1: Control Manual de Velocidad de motor DC	75
4.3.2	Práctica M2: Señal De Encoder – Forma Analógica	79
4.3.3	Práctica M3: Señal De Encoder – Forma Digital	84
4.3.4	Práctica M4: Caracterización Motor DC	88
4.3.5	Práctica M5: Control PID de Velocidad de Motor DC	93
4.4	Práctica de Control de Posición	106
4.4.1	Práctica S1: Secuencia de Encendido Manual	107
4.4.2	Práctica S2: Control de Posición	111
4.5	Prácticas de Análisis de Sonido	116
4.5.1	Práctica D1: Medición de Armónicos	117
4.5.2	Práctica D2: Detección de DTMF	125
4.6	Prácticas de Relé de Propósito General	131
4.6.1	Práctica R1: Test de Conmutación	132

CAPÍTULO V: CONCLUSIONES Y RECOMENDACIONES

5.1	Conclusión	136
5.2	Recomendaciones	137

ANEXOS

ANEXO 1	138
ANEXO 2	140
ANEXO 3	141

BIBLIOGRAFÍA

ÍNDICE GRÁFICO

CAPÍTULO II

Figura 2.1: Diagrama de un Sistema de Control	7
Figura 2.2: Señal Analógica	8
Figura 2.3: Señal Discreta en el Tiempo	8
Figura 2.4: Señal Discreta en Amplitud	9
Figura 2.5: Diagrama PID	10
Figura 2.6: Diagrama de bloques del sistema de control de un proceso	14
Figura 2.7: (a) Característica de un controlador (b) zona muerta Y (c) histéresis	15
Figura 2.8: Característica de un controlador proporcional	15
Figura 2.9: Característica estática del proceso	17
Figura 2.10: Diagrama de Bloques de un Lazo de Realimentación Simple	19
Figura 2.11: Simulación de un Sistema de Control en Lazo Cerrado con Control Proporcional	21
Figura 2.12: Implementación de la acción integral concebida como un Reset automático	23
Figura 2.13: Simulación de un sistema en lazo cerrado con control Proporcional e Integral	24
Figura 2.14: Interpretación geométrica de la acción derivativa como un Control predictivo.	25
Figura 2.15: Simulación de un sistema en lazo cerrado con control Proporcional, integral y derivativo	26
Figura 2.16: Diagrama Simulado de Control Proporcional	30
Figura 2.17: Diagrama Modelado de Control Proporcional	31
Figura 2.18: Diagrama Simulado de Control Proporcional e Integral	32
Figura 2.19: Diagrama Modelado de Control Proporcional e Integral	32
Figura 2.20: Diagrama Simulado de Control Proporcional, Integral y Derivativo	33
Figura 2.21: Diagrama Modelado de Control Proporcional, Integral y Derivativo	33
Figura 2.22: Sistema de Control de Temperatura.	34
Figura 2.23: Controlador PID paralelo	35
Figura 2.24: Esquema Simplificado del Controlador PID Electrónico	35
Figura 2.25: Diagrama de Bloque de la Etapa de Potencia	36
Figura 2.26: Curva de Reacción	37
Figura 2.27: Comparación del Modelo Obtenido.	38
Figura 2.28: Respuesta del Sistema con el Controlador PID	38

CAPÍTULO III

Figura 3.1: Planta de Entrenamiento	39
Figura 3.2: Esquemático del Halógeno	43
Figura 3.3: Esquemático del Encoder	43
Figura 3.4: Esquemático del Micrófono	44
Figura 3.5: Esquemático del Motor DC	44
Figura 3.6: Esquemático del Relé	45
Figura 3.7: Esquemático Motor Stepper	45
Figura 3.8: Diagrama Esquemático EPC (Layout o Topológico)	46
Figura 3.9: Diagrama Esquemático EPC para la FETD	49

CAPÍTULO IV

Figura 4.1: Conexión Práctica T1	51
Figura 4.2: Panel Frontal Práctica T1	52
Figura 4.3: Diagrama de Bloques Práctica T1	54
Figura 4.4: Diagrama de Bloques Práctica T2	56
Figura 4.5: Conexión de EPC a Usb-6009	58
Figura 4.6: Panel Frontal de Práctica T3	60
Figura 4.7: Diagrama de Bloque Programa T3	61
Figura 4.8: Gráfica Tipo Exponencial	62
Figura 4.9: Panel Frontal Práctica T4	65
Figura 4.10: Niveles de Setpoint	65
Figura 4.11: Diagrama de Bloque Práctica T4	66
Figura 4.12: Panel Frontal Práctica T5	69
Figura 4.13: Diagrama de Bloques de la Práctica T5	70
Figura 4.14: Parámetros PID	71
Figura 4.15: Respuesta a Cambios Parámetros PID	72
Figura 4.16: Respuesta del Sistema	73
Figura 4.17: Conexión tarjeta DAQ Práctica M1	75
Figura 4.18: Diagrama Frontal Práctica M1	77
Figura 4.19: Diagrama de Bloque Práctica M1	78
Figura 4.20: Conexión Tarjeta DAQ Práctica M2	79
Figura 4.21: Panel Frontal Práctica M2	81
Figura 4.22: Diagrama de Bloque Práctica M2	82
Figura 4.23: Curva de Arranque	83
Figura 4.24: Conexión Tarjeta DAQ Práctica M3	85
Figura 4.25: Panel Frontal Práctica M3	86
Figura 4.26: Diagrama de Bloque Práctica M3	87
Figura 4.27: Conexión Tarjeta DAQ Práctica M4	88
Figura 4.28: Panel Frontal Práctica M4	90
Figura 4.29: Diagrama de Bloques Práctica M4	91
Figura 4.30: Aumento Progresivo de Voltaje	92
Figura 4.31: Comportamiento de Velocidad	92
Figura 4.32: Conexión Tarjeta DAQ Práctica M5	94
Figura 4.33: Panel Frontal Práctica M5	96
Figura 4.34: Diagrama de Bloque Práctica M5	97
Figura 4.35: $K_c=1$	98
Figura 4.36: $K_c=0.1$	98
Figura 4.37: $K_c=0.01$	99
Figura 4.38: $K_c=0.001$	99
Figura 4.39: $K_c=0.005$	100
Figura 4.40: $K_c= 0.0025$	100
Figura 4.41: $K_c= 0.003$	101
Figura 4.42: $F_{cr} = 4.9 \text{ Hz}$; $P_{cr} = 1/4.9 = 0.2 \text{ segundos}$	102
Figura 4.43: $K_c= 0.00135$	103
Figura 4.44: Señal de Respuesta	103
Figura 4.45: $K_c=0.00210$	104
Figura 4.46: $K_c= 0.0005$	104
Figura 4.47: Diagrama de Bloque Práctica M5	105
Figura 4.48: Conexión Tarjeta DAQ Práctica S1	107
Figura 4.49: Panel Frontal Práctica S1	108

Figura 4.50: Diagrama de Bloques de la Práctica S1	109
Figura 4.51: Conexión tarjeta DAQ Práctica S2	111
Figura 4.52: Panel Frontal Práctica S2	113
Figura 4.53: Diagrama de Bloque Práctica S2	114
Figura 4.54: Conexión Tarjeta DAQ Práctica DSA1	117
Figura 4.55: Panel Frontal Práctica DSA1	119
Figura 4.56: Diagrama de Bloques Práctica DSA1	120
Figura 4.57: Teclado DTMF	121
Figura 4.58: Señal Ruidosa	121
Figura 4.59: Espectro de Frecuencia	122
Figura 4.60: Componentes Armónicos	123
Figura 4.61: Espectro de Potencia	123
Figura 4.62: Diagrama de Bloque DSA1	124
Figura 4.63: Conexión tarjeta DAQ Práctica DSA2	125
Figura 4.64: Panel Frontal Práctica DSA2	127
Figura 4.65: Diagrama de Bloques Práctica DSA2	129
Figura 4.66: Conexión Tarjeta DAQ Práctica R1	133
Figura 4.67: Panel Frontal Práctica R1	134
Figura 4.68: Diagrama de Bloques Práctica R1	135

INTRODUCCIÓN

La Facultad de Educación Técnica para el Desarrollo, dentro de su proceso de implementación de estrategias, encaminadas al fortalecimiento de sus procesos académicos apunta con gran oportunidad al uso de una infraestructura tecnológica (hardware – software), recientemente adquirida para el laboratorio de electrónica, sin embargo carece de equipamiento para implementar prácticas de laboratorio en el área de controles industriales.

De acuerdo a lo propuesto en la malla curricular de las carreras de Ingeniería de la Facultad Técnica, se dictan materias que proponen al estudiante de ingeniería la introducción a los procesos industriales; tales como Automática I, II y III. Estas materias y otras afines podrán utilizar el entrenador electrónico para procesos industriales, el cual es objeto del presente proyecto, para implementar los diferentes sistemas de control.

El diseño propuesto y la implementación de hardware a realizar contemplan un estudio completo del control en procesos electrónicos industriales, lo cual ayudará en gran medida al desarrollo de la creatividad del estudiante de nuestra facultad.

El presente documento consta de 5 capítulos.

CAPÍTULO I: Este incluye todos los aspectos relacionados con el planteamiento de Problema, Objetivos, Justificación, y Delimitación de la Investigación.

CAPÍTULO II: Se refiere al marco teórico, contiene los antecedentes teóricos y prácticos relacionados con la investigación a Control Automático, los fundamentos respectivos, la definición de términos básicos y manejo de variables de proceso.

CAPÍTULO III: Comprende el marco metodológico, el diseño propuesto para el nuevo EPC FETD.

CAPÍTULO IV: En este capítulo se expone una guía práctica para el estudiante, con ejercicios propuestos para el funcionamiento del EPC.

CAPÍTULO V: Se refiere a las conclusiones y recomendaciones del estudio de la investigación.

CAPÍTULO 1: EL PROBLEMA

1.1 PLANTEAMIENTO DEL PROBLEMA

En un ambiente evolutivo de la tecnología en la educación y pedagogía de las carreras tecnológicas; lejos de las limitaciones en nuestra facultad, el presente proyecto se basa en las fortalezas existentes en la FETD (Facultad de Educación Técnica para el Desarrollo) que en lo estructural cuenta con un laboratorio de electrónica que actualmente posee el software de LabVIEW con licencia profesional; la misma que fue adquirida por la Universidad Católica de Santiago de Guayaquil, y que a medida como avanzan en los semestres han dado resultados satisfactorios en la implementación de sistemas, equipos industriales y de adquisición de datos; de manera que, lo que hace poco indicábamos como una falencia (nuestros laboratorios), actualmente se está transformando en una de nuestras mejores fortalezas, dejando de lado las debilidades de la estructura tecnológica de la facultad;convirtiendo esto en una amenaza para otras universidades del medio, por el mejoramiento en calidad educativa.

Es necesario estar a la vanguardia en la actualización y enseñanza tecnológica de nuestra facultad, mejorando el ámbito pedagógico para un mejor desarrollo académico y educación de calidad para los estudiantes, como en otras instituciones de educación superior que ofrecen carreras tecnológicas.

Es necesario brindar a nuestros estudiantes, herramientas de estudios que colaboren en su formación profesional e incentiven la investigación.

1.1.1 FORMULACIÓN DEL PROBLEMA

Necesidad de mejorar el laboratorio de electrónica para el estudio e investigación de procesos industriales y de control, con la cual se pueda lograr un desarrollo intelectual de nuestros estudiantes.

1.1.2 SISTEMATIZACIÓN DE LA VARIABLE

¿Cuál es la situación actual del recurso de aprendizaje en el laboratorio de electrónica?

¿Permiten los recursos del laboratorio de electrónica la enseñanza práctica en la carrera de Control y Automatismo?

1.2 OBJETIVOS

1.2.1 OBJETIVO GENERAL

Implementación de entrenadores electrónicos (EPC), para el área de controles industriales y automatización de procesos, de manera que se pueda obtener un mejor resultado en las prácticas de laboratorio de electrónica en las materias que tengan injerencia o en áreas pertinentes de electrónica de control.

1.2.2 OBJETIVOS ESPECÍFICOS

- 1 Implementar Entrenadores Electrónicos Industriales.
- 2 Elaborar prácticas para el desarrollo de la planta de entrenamiento en LabVIEW.
- 3 Diseño esquemático propuesto (EPC FETD), para mejoras futuras.

1.3 JUSTIFICACIÓN

El presente texto tiene como finalidad diseñar e implementar herramientas para el laboratorio de electrónica, que serán de gran utilidad para el proceso de enseñanza práctica de materias que son esenciales de la Carrera de ingeniería en Telecomunicaciones, Electromecánica y Control y Automatismo.

1.4 DELIMITACIONES

1.4.1 DELIMITACIÓN ESPACIAL

La presente investigación en diseño y la implementación del hardware tecnológico se lo desarrollará en el laboratorio de electrónica de la facultad Técnica para el Desarrollo de la Universidad Católica Santiago de Guayaquil

1.4.2 DELIMITACIÓN TEMPORAL

El estudio del diseño e implementación de la tesis se realizará en un periodo de tres meses a partir de la fecha que fue aprobado el tema de tesis.

CAPÍTULO 2: MARCO TEÓRICO

2.1 ANTECEDENTES DE LA INVESTIGACIÓN

2.1.1 CON QUÉ CUENTA EL LABORATORIO DE ELECTRÓNICA

El laboratorio de electrónica cuenta con una herramienta educativa muy importante para el proceso de enseñanza – aprendizaje de los estudiantes de la facultad Técnica, como lo es el software de programación LabVIEW; esta aplicación más la suma de herramientas, para prácticas como el EPC ofrecerán una nueva forma de enseñanza en materias de la carreras de Ingeniería en Control y Automatismo, Telecomunicaciones y Electromecánica; esto es, no sea sólo de forma teórica, sino también la práctica, simulando a través de LabVIEW un proceso de control real, de un equipo tecnológico industrial o mecánico.

2.2 BASES TEÓRICAS DEL SISTEMA DE CONTROL

2.2.1 SISTEMA DE CONTROL

Desde el punto de vista de la teoría de control, un sistema o proceso está formado por un conjunto de elementos relacionados entre sí que ofrecen señales de salida, en función de señales o datos de entrada.

Es importante resaltar el hecho, de que, no es necesario conocer el funcionamiento interno o cómo actúan entre sí los diversos elementos para caracterizar el sistema. Para ello, sólo se precisa conocer la relación que existe entre la entrada y la salida del proceso que realiza el mismo (principio de cajonera). El aspecto más importante de un sistema es el conocimiento de su dinámica, es decir, cómo se comporta la señal de salida frente a una variación de la señal de entrada.

Un conocimiento preciso de la relación entrada/salida permite predecir la respuesta del sistema y seleccionar la acción de control adecuada para mejorarla. De esta manera, el diseñador, conociendo cuál es la dinámica deseada, ajustará la acción de control para conseguir el objetivo final frente a perturbaciones externas del sistema.

En vista de todo lo expuesto, se puede definir un sistema de control como el conjunto de elementos que interactúan, para conseguir que la salida de un proceso se comporte tal y como se desea, mediante una acción de control. (Ver *Figura 2.1*)

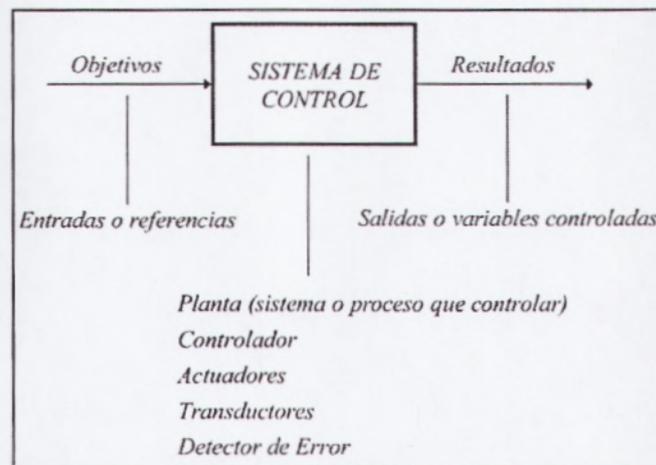


Figura 2.1: Diagrama de un Sistema de Control

2.2.2 TIPOS DE SEÑALES Y SISTEMAS

Consideraremos como señales las variaciones a lo largo del tiempo de las entradas o salidas de un sistema. Obviamente, estas señales pueden ser de distinta naturaleza, y por tanto sus unidades físicas pueden ser diversas. Según cómo sea la variación de estas señales, podemos clasificarlas dentro de dos grandes grupos: señales analógicas y señales discretas.

-**Señales Analógicas:** Son aquellas cuya variación, tanto en amplitud como a lo largo del tiempo, es continua. Es decir, pueden tomar cualquier valor real, en cualquier instante de tiempo (Ver Figura. 2.2)

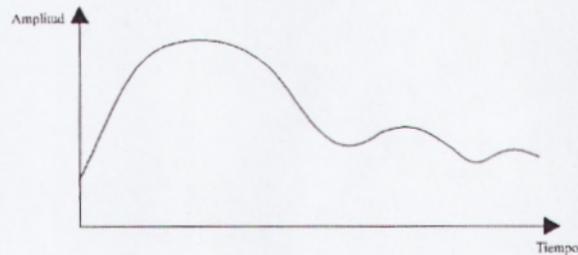


Figura 2.2: Señal Analógica

- **Señales Discretas:** Este tipo de señales no tiene una variación continua como las anteriores, sino que su evolución se rige por un determinado conjunto finito de valores posibles. Según dónde tome este conjunto de valores, podremos distinguir entre señales discretas en amplitud o discretas en tiempo.

- o **Señales Discretas en Tiempo:** Sólo tienen valor en instantes de tiempo predeterminados. Y aunque su amplitud puede ser cualquier valor dentro del rango de los reales, el valor de la señal entre dos instantes de tiempo consecutivos no está definido. (Ver Figura 2.3)

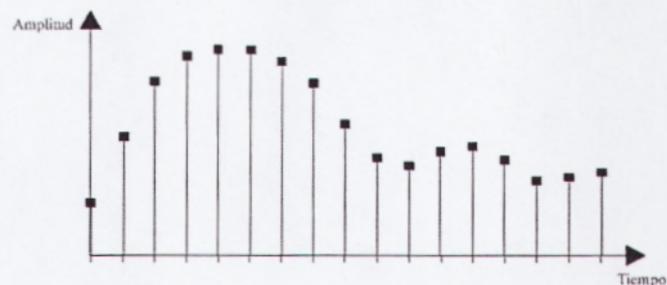


Figura 2.3: Señal Discreta en el Tiempo

- **Señales Discretas en Amplitud:** En este caso, la señal toma valor en cualquier instante de tiempo, pero estos valores de amplitud pueden encontrarse entre los definidos en el conjunto predeterminado. (Ver Figura2.4)

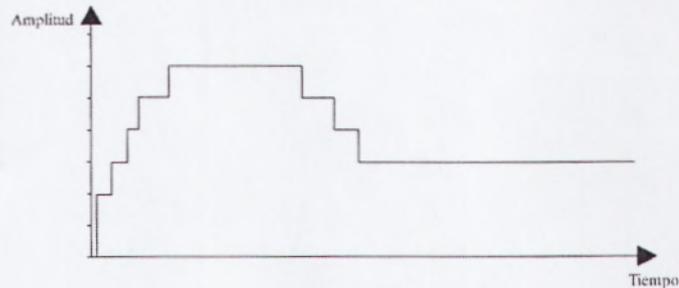


Figura2.4: Señal Discreta en Amplitud

2.3 SISTEMA DE CONTROL PID

2.3.1 CONTROLADOR PID

El controlador PID (Proporcional, Integral y Derivativo) es un controlador realimentado cuyo propósito es hacer que el error en estado estacionario, entre la señal de referencia y la señal de salida de la planta, sea cero de manera asintótica en el tiempo, lo que se logra mediante el uso de la acción integral. Además el controlador tiene la capacidad de anticipar el futuro a través de la acción derivativa que tiene un efecto predictivo sobre la salida del proceso.

Los controladores PID son suficientes para resolver el problema de control de muchas aplicaciones en la industria, particularmente cuando la dinámica del proceso lo permite (en general procesos que pueden ser descritos por dinámicas de primer y segundo orden), y los requerimientos de desempeño son modestos (generalmente limitados a especificaciones del comportamiento del error en estado estacionario y una rápida respuesta a cambios en la señal de referencia).

El controlador PID, de lejos, es el algoritmo de control más común. Numerosos lazos control utilizan este algoritmo (Ver Figura 2.5), que puede ser implementado de diferentes maneras: como controlador stand-alone, como parte de un paquete de control digital directo o como parte de un sistema de control distribuido. Su estudio puede ser abordado desde múltiples puntos de vista. Puede ser tratado como un dispositivo que puede ser operado utilizando unas cuantas reglas prácticas, pero también puede ser estudiado analíticamente.

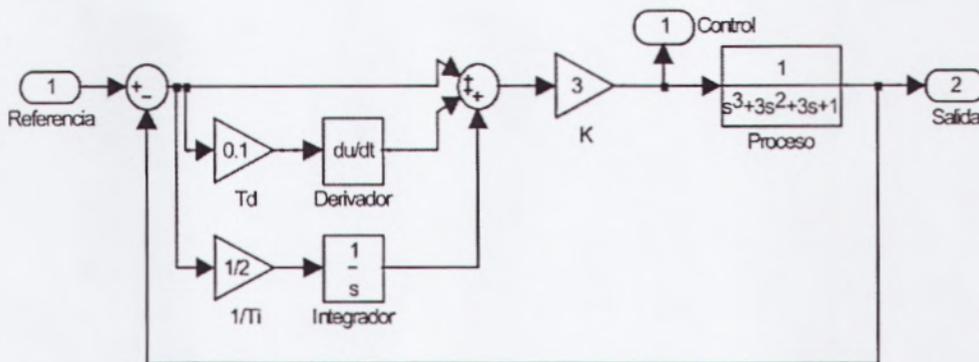


Figura2.5: Diagrama PID

Su uso extensivo en la industria es tal que el 95% de los lazos de control que existen en las aplicaciones industriales son del tipo PID, de los cuales la mayoría son controladores PI, lo que muestra la preferencia del usuario en el uso de leyes de control muy simples. En general, el usuario no explota todas las características de estos controladores, quizás por falta de una mejor comprensión desde el punto de vista de la teoría de control.

Los algoritmos actuales se combinan con funciones lógicas y secuenciales y una serie de mecanismos y funciones adicionales para adecuarse a los requerimientos de los modernos sistemas de control y automatización industrial, lo que da lugar a dispositivos especializados para el control de temperatura, velocidad, distribución de energía, transporte, máquinas-herramientas, reacción química, fermentación, entre otros.

Los controladores PID son generalmente usados en el nivel de control más bajo, por debajo de algunos dispositivos de mediano nivel como PLCs, supervisores, y sistemas de monitoreo. Sin embargo, su importancia es tal que se convierte en el "pan de cada día" del Ingeniero de Control.

2.3.2 FUNDAMENTOS DE CONTROLADORES PID

Los controladores PID han sobrevivido a muchos cambios en la tecnología a lo largo de su historia. Desde los antiguos reguladores de Watt, de la época de la revolución industrial, pasando por los controladores neumáticos, los controladores analógicos eléctricos y electrónicos (primero implementados con válvulas y luego con circuitos integrados) hasta los modernos controladores basados en microprocesadores, que proporcionan una mayor flexibilidad debido a su programabilidad.

El microprocesador ha tenido una influencia dramática sobre el desarrollo del controlador PID; ha permitido brindar nuevas oportunidades para implementar funciones adicionales como el ajuste automático de parámetros y los cambios de modos de control. Para los efectos de estos apuntes, se considera la frase "ajuste automático" en el sentido de que los parámetros del controlador se ajustan automáticamente en base a la demanda de un operador o de una señal externa, desactivando para ello el controlador. Esto hace que esta función sea diferente a la función de adaptación, propias de los controladores adaptivos, que ajustan en línea (o de manera continua) los parámetros del controlador.

El desarrollo de los sistemas de control PID está también influenciado por el desarrollo en el campo de la comunicación de datos de campos, lo que ha permitido su inserción como módulos importantes en los esquemas de control distribuido. En este sentido, la capacidad de comunicación de estos dispositivos con otros dispositivos de campo como PLCs y otros sistemas de

control de niveles superiores, es una función necesaria en los modernos controladores PID.

Si bien a nivel industrial existen grupos de Ingenieros de procesos e instrumentación que están familiarizados con los controladores PID, en el sentido de que llevan una práctica continua de instalación, puesta en marcha y operación de sistemas de control con lazos PID, también es cierto que existe mucho desconocimiento acerca de los detalles involucrados en la construcción de los algoritmos. Prueba de ello es que muchos controladores son puestos en modo manual y, entre aquellos que están en el modo automático, frecuentemente la acción derivativa se encuentra desactivada. La razón es obvia, el ajuste de los controladores es un trabajo tedioso y requiere de cierta intuición basada en los principios de funcionamiento tanto de los procesos físicos controlados como de la misma teoría de control.

Otras razones del pobre desempeño tienen que ver con problemas en la instrumentación y los equipos y accesorios utilizados en el lazo de control, como son los sensores, actuadores, dispositivos de comunicación, interfaces de adquisición de datos, etc.

Los principales problemas de los actuadores están generalmente relacionados con fallas de dimensionamiento (en general están subdimensionados) y los problemas de histéresis que introducen no linealidades importantes. Por su parte, los dispositivos asociados con la medición de las señales de la planta (sensores, dispositivos de adquisición de datos, adecuación de señales y sistemas de comunicación de datos de campo) a menudo se encuentran mal calibrados y, es frecuente que estén dotados de mecanismos inadecuados de filtraje pobre o bien de filtraje excesivo (producido en los llamados sensores inteligentes). Más aún, muchos sistemas de control no cumplen con las condiciones mínimas para su operación en tiempo real.

Es así que quedan por hacer muchas mejoras sustanciales con respecto al desempeño de los procesos industriales. Por su parte, la industria, a medida que la demanda de productos requiere una mejor calidad, está obligada a mejorar sus lazos de control, lo que a su vez requiere un mayor conocimiento acerca de los procesos y de sus mecanismos de regulación.

2.3.3 PRINCIPIO DE REALIMENTACIÓN

La idea de la realimentación es bastante simple y muy poderosa. A lo largo de su historia, ha tenido una fuerte influencia en la evolución de la tecnología. Las aplicaciones del principio de realimentación han tenido éxito en los campos del control, comunicaciones e instrumentación. Para entender el concepto, asuma que el proceso es tal que cuando el valor de la variable manipulada se incrementa, entonces se incrementan los valores de las variables del proceso. Bajo este concepto simple, el principio de realimentación puede ser expresado como sigue:

Incrementar la variable manipulada cuando la variable del proceso sea más pequeña que la referencia y disminuirla cuando ésta sea más grande.

Este tipo de realimentación se llama "realimentación negativa" debido a que la variable manipulada se mueve en la dirección opuesta a la variable del proceso. El principio puede ser ilustrado por el diagrama de bloques que se muestra (VerFigura2.6). En este diagrama el proceso y el controlador están representados por cajas negras y las flechas denotan las entradas y salidas a cada bloque.

Note que existe un símbolo especial que denota una suma de señales. El diagrama de bloques muestra que el proceso y el controlador están conectados en un lazo realimentado.

La presencia del signo en el bloque de retorno indica que la realimentación es negativa.

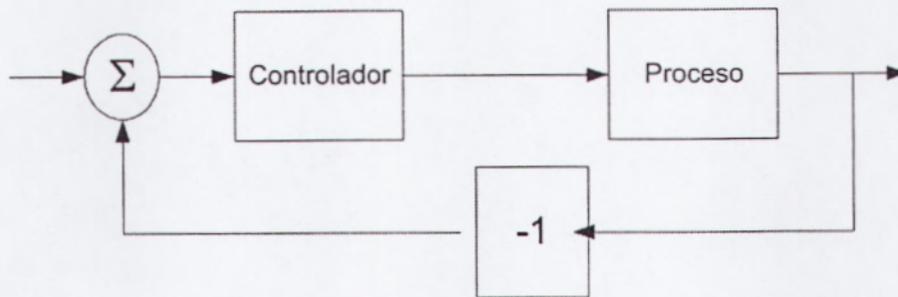


Figura2.6: Diagrama de bloques del sistema de control de un proceso.

CONTROL ON-OFF

El mecanismo de realimentación más simple se puede describir matemáticamente como sigue:

$$u = \begin{cases} u_{\max} & e > 0 \\ u_{\min} & e < 0 \end{cases} \quad \text{Ec.1}$$

Donde $e = y_{sp} - y$ (diferencia entre la referencia especificada por el operador y la salida medida del proceso) es el denominado "error de control". Esta ley de control implica que siempre se usa la acción correctiva máxima. De esta manera, la variable manipulada tiene su valor más grande cuando el error es positivo y su valor más pequeño cuando el error es negativo. La realimentación de este tipo se llama "control on-off". Es simple y no tiene parámetros que configurar, aparte de las acciones mínima y máxima que se ejecutan en el cálculo de la señal de control. El control on-off muchas veces es apropiado para mantener la variable controlada del proceso cerca del valor de la referencia que fue especificada, pero típicamente resulta en un sistema donde las variables oscilan. Note en la Ec. 1 que la variable de control no está definida cuando el error es cero.

Es común tener algunas modificaciones ya sea introduciendo histéresis o una zona muerta como se muestra (Ver Figura2.7)

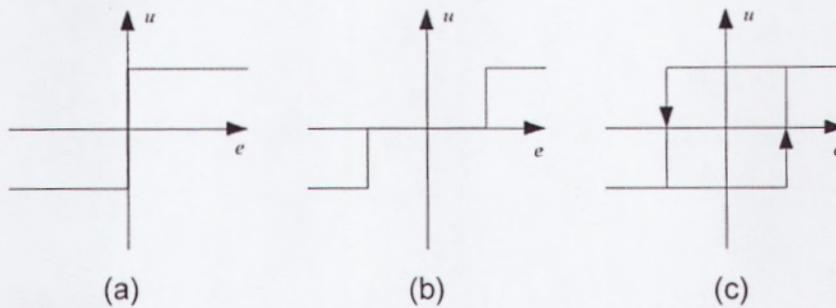


Figura2.7: (a) Característica de un controlador on-off ideal y modificaciones con (b) zona muerta y (c) histéresis.

CONTROL PROPORCIONAL

La razón por la que el control on-off resulta en oscilaciones es que el sistema sobreactúa cuando ocurre un pequeño cambio en el error que hace que la variable manipulada cambie sobre su rango completo. Este efecto se evita en el control proporcional, donde la característica del controlador es proporcional al error de control cuando éstos son pequeños. La Figura2.8 muestra la característica de un controlador proporcional. De esta manera, el controlador está caracterizado por la función no lineal $u=f_c(e)$, dependiente del error de control, que se muestra en la figura.

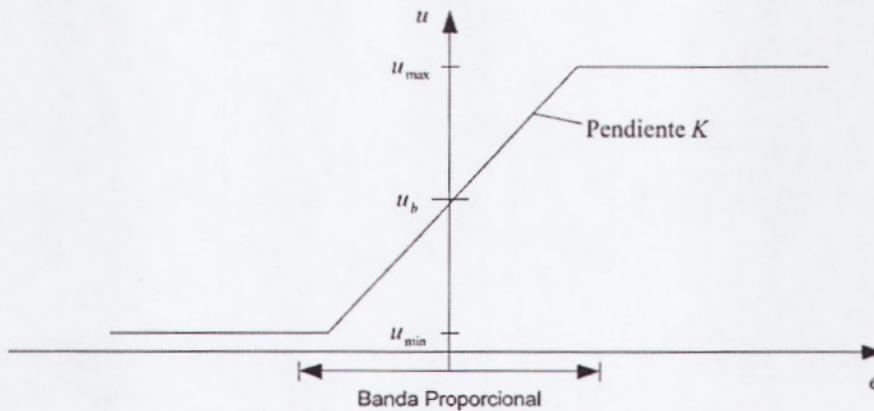


Figura2.8: Característica de un controlador proporcional. La entrada es el error de control e y la salida es la señal de control u .

Para describir la característica del controlador proporcional se debe dar los límites u_{\max} y u_{\min} de la variable de control. El rango lineal puede ser especificado, ya sea, por la pendiente dada en la curva característica (ganancia K del controlador) o, bien, por el rango donde la curva característica es lineal (conocida como banda proporcional P_b). Este rango está normalmente centrado alrededor de la referencia. La banda proporcional y la ganancia del controlador están relacionadas a través de:

$$u_{\max} - u_{\min} = KP_b \quad \text{Ec. 2}$$

Normalmente se asume que $u_{\max} - u_{\min} = 100$, lo cual implica que:

$$K = \frac{100}{P_b} \quad \text{Ec. 3}$$

Note que un controlador proporcional actúa como un controlador on-off cuando los errores de control son grandes.

2.3.4 ANÁLISIS ESTÁTICO DE LOS SISTEMAS REALIMENTADOS

Algunas propiedades de un sistema de control se pueden comprender mediante un análisis estático simple. Para esto, se introduce la "característica estática del proceso", que es una curva que muestra el valor estacionario de su salida y como función de su entrada u (Ver Figura 2.9). Note que la curva tiene una interpretación física sólo si el proceso es estable. La característica estática del proceso es importante para determinar el rango de las señales de control que son requeridas para cambiar la salida del proceso sobre un rango deseado. También es usada para dimensionar el tamaño de los actuadores, seleccionar la resolución del sensor, así como para evaluar qué tan grandes son las variaciones en la ganancia estática.

Todas estas consideraciones se deben tomar en cuenta a la hora de diseñar el sistema de control.

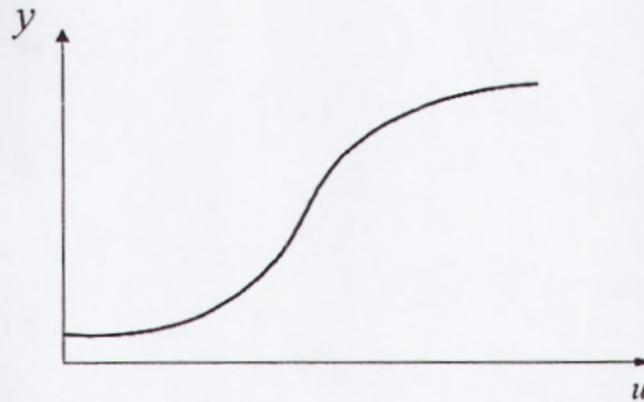


Figura 2.9: Característica estática del proceso. Muestra la salida del proceso y como una función de su entrada, bajo consideraciones estáticas.

2.3.5 CONTROL PID

En la sección anterior se vio que el control proporcional tiene la desventaja de que, en la mayoría de los casos, resulta en un error estático o de estado estacionario diferente de cero. Los algoritmos de control usados en la práctica son, por tanto, normalmente más complejos que el del controlador proporcional. Se puede mostrar empíricamente que el llamado “controlador PID” es una estructura útil. Dentro de la banda proporcional el comportamiento del algoritmo PID en su versión de “libro de texto” se puede describir como:

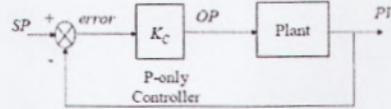
$$u(t) = K \left(e(t) + \frac{1}{T_i} \int_0^t e(\tau) d\tau + T_d \frac{de(t)}{dt} \right) \quad \text{Ec.4}$$

Donde u es la variable de control y e es el error de control dado por $e = y_{SP} - y$. De esta manera, la variable de control es una suma de tres términos: el término P, que es proporcional al error; el término I, que es proporcional a la integral del error; y el término D, que es proporcional a la derivada del error. Los parámetros del controlador son: la ganancia proporcional K , el tiempo integral T_i y el tiempo derivativo T_d .

2.3.6 PARÁMETROS PID

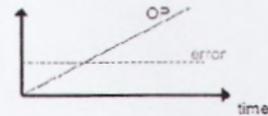
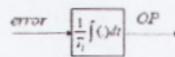
- **Proporcional**

- Lleva al punto de ajuste
- $Error \rightarrow 0, OP \rightarrow 0$
- "Error en estado estable"



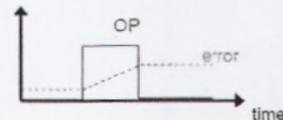
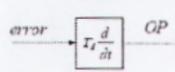
- **Integral**

- Elimina el error en estado estable
- OP proporcional a $\int error$



- **Derivativo**

- Incrementa la tasa de respuesta
- OP proporcional a la tasa de cambio del error



ACCIÓN PROPORCIONAL

En el caso de un control proporcional puro, la ley de control de la ecuación (4) se reduce a:

$$u(t) = Ke(t) + u_b \quad \text{Ec.5}$$

La acción de control es simplemente proporcional al error de control. La variable u_b es una señal de polarización o un reset. Cuando el error de control e es cero, la variable de control toma el valor $u(t) = u_b$. La polarización u_b a menudo se la fija en $(u_{\max} + u_{\min})/2$, pero, algunas veces, puede ser ajustada manualmente de forma que el error de control en estado estacionario sea cero en una referencia dada.

ANÁLISIS ESTÁTICO

Muchas de las propiedades del control proporcional se pueden entender mediante el siguiente argumento, que está basado en consideraciones estáticas puras. Considere un lazo realimentado simple, como el mostrado en (VerFigura2.10), que consiste de un proceso y un controlador, sometidos a perturbaciones.

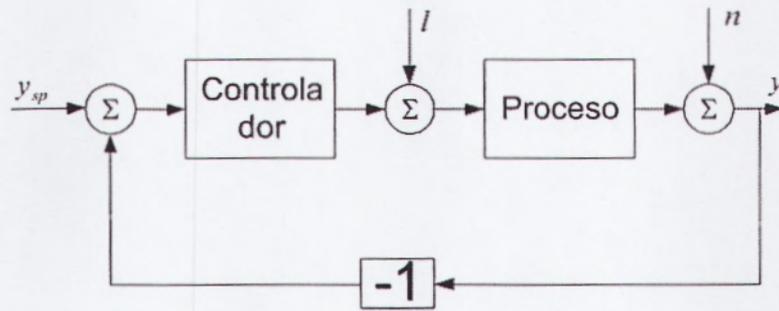


Figura 2.10: Diagrama de Bloques de un Lazo de Realimentación Simple

Asuma que el controlador tiene acción proporcional y que el proceso está representado por un modelo estático:

$$x = K_p(u + l) \quad \text{Ec.6}$$

Donde x es la variable del proceso, u es la variable de control, l es una perturbación de carga K_p es la ganancia estática del proceso. Las siguientes ecuaciones se obtienen a partir del diagrama de bloques:

$$\begin{aligned} y &= x + n \\ x &= K_p(u + l) \\ u &= K(y_{sp} - y) + u_b \end{aligned} \quad \text{Ec.7}$$

La eliminación de las variables intermedias da la siguiente relación entre la variable del proceso x , la referencia y_{sp} , la perturbación de carga l y el ruido de medición n :

$$x = \frac{KK_p}{1 + KK_p}(y_{sp} - n) + \frac{K_p}{1 + KK_p}(l + u_b) \quad \text{Ec.8}$$

De la Ec.8 se pueden leer muchas propiedades interesantes del sistema en lazo cerrado. Primero asuma que n y u_b son cero. La ganancia de lazo debe ser alta para asegurar que la salida del proceso x sea cercana a la referencia y_{sp} . Un valor alto de la ganancia de lazo permitirá hacer que el sistema sea insensible a la perturbación de carga l .

Sin embargo, si n es diferente de cero, de la Ec.8 se sigue que el ruido de medición n influye sobre la salida del proceso de la misma forma que lo hace la referencia y_{sp} .

Para evitar que el sistema sea sensible al ruido de medición, la ganancia de lazo no debe ser muy grande. Más aún, la polarización u_b del controlador influye en el sistema de la misma forma en que lo hace la perturbación de carga. Por tanto, es obvio que el diseño de la ganancia de lazo debe ser considerado como un compromiso entre dos objetivos de control diferentes, por lo que no existe una respuesta simple que permita encontrar una fórmula que determine la mejor ganancia de lazo a ser aplicada en el sistema.

Esto dependerá de cuál objetivo de control es más importante para la aplicación en cuestión.

También, de la Ec.8 se puede ver que el controlador proporcional normalmente producirá un error en estado estacionario. Esto puede ser deducido intuitivamente a partir de la observación de la Ec.7, donde el error de control es cero sólo cuando $u = u_b$ en estado estacionario. Por tanto, el error puede hacerse cero en una condición de operación dada manipulando la polarización u_b del controlador.

El análisis estático anterior está basado en la suposición de que el proceso se puede describir mediante un modelo estático. Cuando se considera la dinámica del sistema se introducen otras propiedades sobre el comportamiento del sistema en lazo cerrado. Lamás importante es que el sistema en lazo cerrado normalmente será inestable si se eligen altas ganancias de lazo. En la práctica, es la dinámica del sistema la que determina la máxima ganancia de lazo que puede ser utilizada.

Un ejemplo típico del control proporcional se ilustra (VerFigura2.11). La figura muestra el comportamiento de la salida del proceso y de la señal de control, después de un cambio al escalón en la señal de referencia. El error en estado estacionario puede ser calculado a partir de la Ec.8

El término de polarización u_b , la perturbación de carga l y el ruido de medición n son cero en la simulación. Con una ganancia del controlador $K = 1$ y una ganancia estática del proceso $K_P=1$, se obtiene un error de control del 50%. La figura muestra que el error en estado estacionario decrece a medida que se incrementa la ganancia del controlador, tal como se predice en la Ec.8. Note también que la respuesta se vuelve más oscilatoria al incrementar la ganancia del controlador. Esto se debe a la dinámica del proceso.

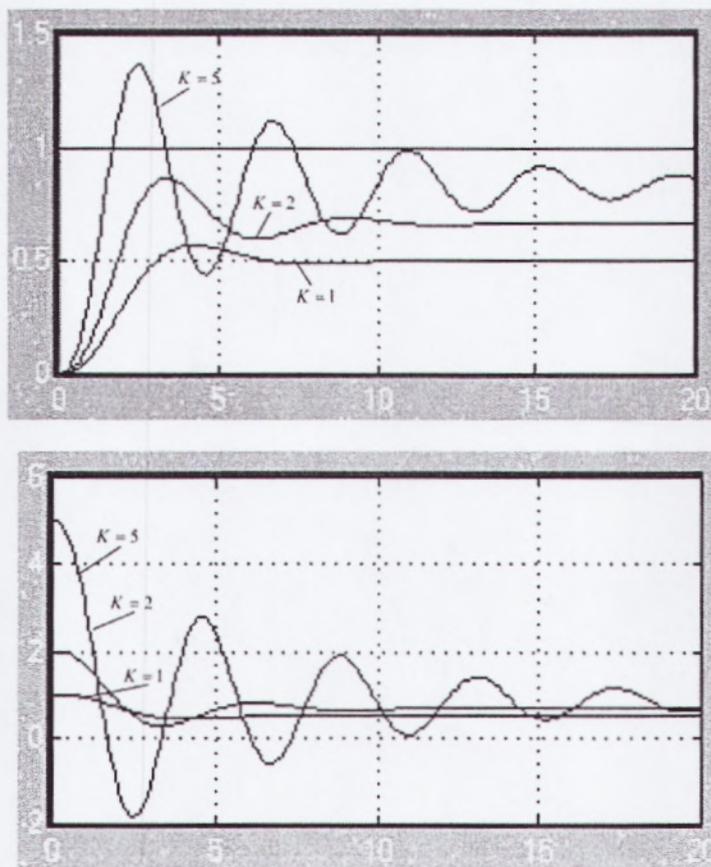


Figura 2.11: Simulación de un Sistema de Control en Lazo Cerrado con Control Proporcional

ACCIÓN INTEGRAL

La función principal de la acción integral es asegurar que la salida del proceso concuerde con la referencia en estado estacionario. Con el controlador proporcional, normalmente existiría un error en estado estacionario. Con la acción integral, un pequeño error positivo siempre producirá un incremento en la señal de control y, un error negativo siempre dará una señal decreciente sin importar cuán pequeño sea el error.

El siguiente argumento simple muestra que el error en estado estacionario siempre será cero con la acción integral. Asuma que el sistema está en estado estacionario con una señal de control constante, u_0 y un error constante, e_0 . De la Ec. 4 se tiene que la señal de control está dada por:

$$u_0 = K \left(e_0 + \frac{e_0}{T_i} t \right) \quad \text{Ec. 9}$$

Como se tiene que $e_0 \neq 0$, claramente se contradice el supuesto de que la señal de control u_0 se mantiene constante. Por tanto, como resultado de esto, un controlador con acción integral siempre dará un error en estado estacionario cero.

La acción integral también puede ser vista como un dispositivo que automáticamente restablece el término de polarización u_b de un controlador proporcional.

Esto se ilustra en el diagrama de bloques (Ver Figura 2.12), que muestra un controlador proporcional con un "reset" que se ajusta automáticamente. El ajuste se hace realimentando una señal, que es un valor filtrado de la salida del controlador, a un punto de suma.

El "reset automático" fue el que dio origen a la acción integral del controlador del tipo PID.

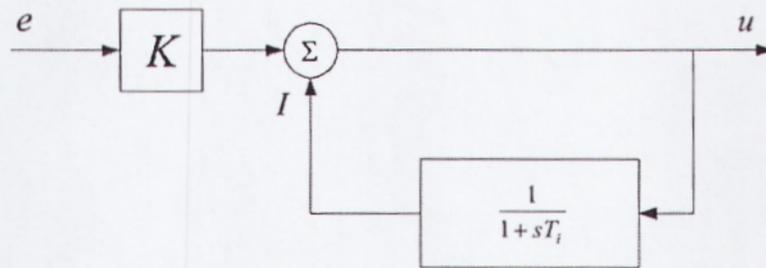


Figura2.12: Implementación de la acción integral concebida como un reset automático.

La implementación mostrada (Ver Figura 2.12), todavía es usada por muchos fabricantes de controladores. Sin embargo, a partir del diagrama de bloques, se pueden deducir las siguientes ecuaciones:

$$u = Ke + I$$

$$T_i \frac{dI}{dt} + I = u$$

De donde, la eliminación de u entre estas ecuaciones produce:

$$T_i \frac{dI}{dt} + I = Ke + I$$

Y, de aquí:

$$T_i \frac{dI}{dt} = Ke$$

Que muestra que el controlador (Ver Figura 2.12), es en los hechos, un controlador del tipo PI.

Las propiedades de la acción integral se ilustran (VerFigura2.13), que muestra la simulación de un sistema de control PI. La ganancia proporcional es $K= 1$

en todas las curvas. El caso $T_i = \infty$ corresponde a un control proporcional puro, que es idéntico al de $K = 1$, donde el error final es del 50%. El error es eliminado cuando T_i toma valores finitos. Para valores grandes de T_i , la respuesta se desliza lentamente hacia la referencia.

El acercamiento es aproximadamente exponencial con constante de tiempo T_i/KK_p más rápido para valores pequeños de T_i ; pero es, también, más oscilatorio.

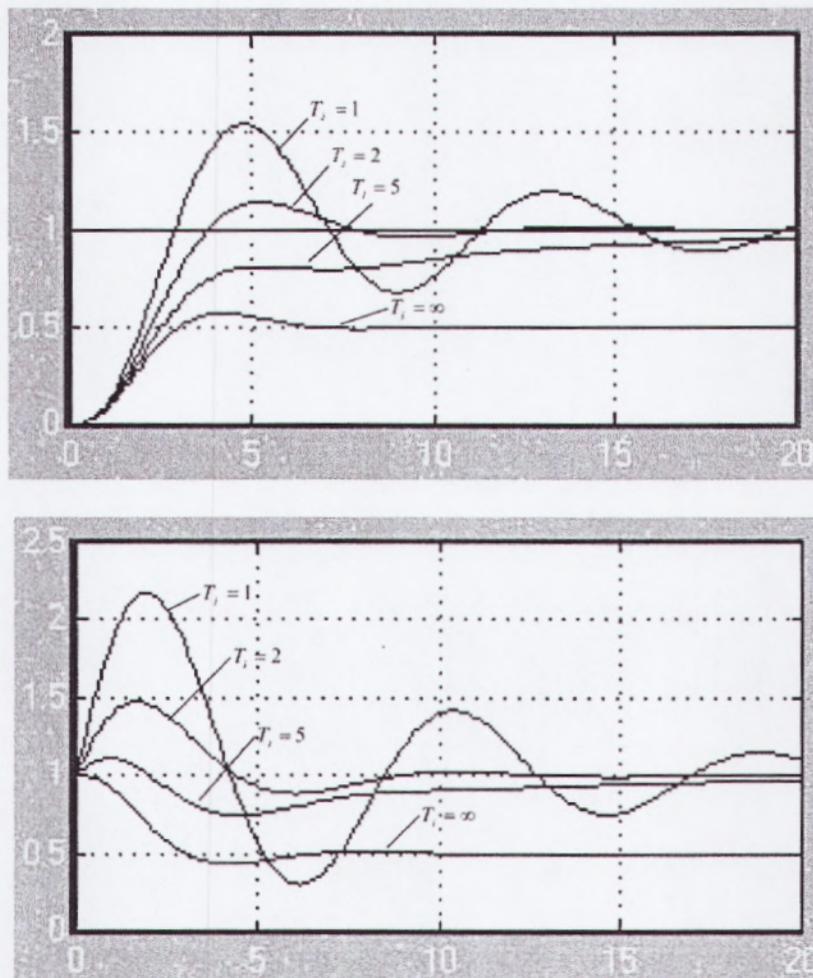


Figura 2.13: Simulación de un sistema en lazo cerrado con control proporcional e integral.

ACCIÓN DERIVATIVA

El propósito de la acción derivativa es mejorar la estabilidad de lazo cerrado. El mecanismo de inestabilidad puede ser descrito intuitivamente como sigue. Debido a la dinámica del proceso, pasa algún tiempo antes de que la variable de control se note en la salida del proceso. De esta manera, el sistema de control tarda en corregir el error.

La acción de un controlador con acción proporcional y derivativa puede ser interpretada como si el control proporcional fuese hecho para predecir la salida del proceso. La predicción se hace por la extrapolación del error de control en la dirección de la tangente a su curva respectiva, como se muestra en la (VerFigura 2.14)

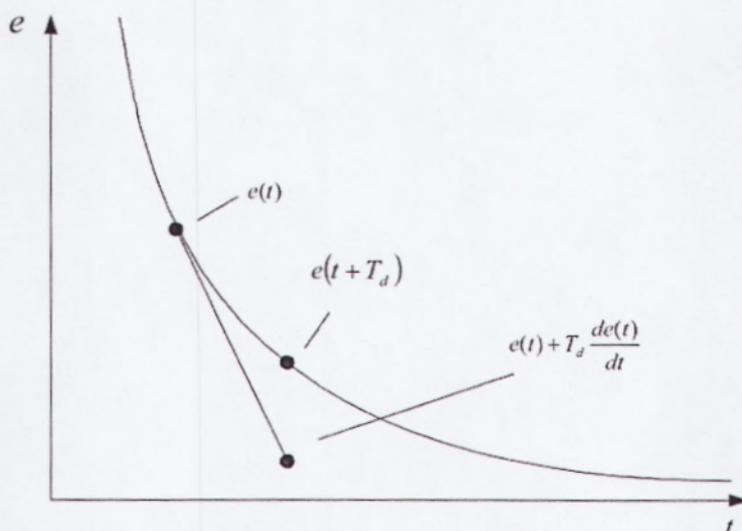


Figura2.14: Interpretación geométrica de la acción derivativa como un control predictivo, donde la predicción se obtiene mediante extrapolación lineal.

La estructura básica de un controlador PD está dada por:

$$u(t) = K \left(e(t) + T_d \frac{de(t)}{dt} \right) \quad \text{Ec.10}$$

La expansión en series de Taylor de $e(t + T_d)$ da:

$$e(t + T_d) \approx e(t) + T_d \frac{de(t)}{dt}$$

De esta manera, la señal de control es proporcional a un estimado del error de control en él, un tiempo T_d hacia adelante, donde el estimado se obtiene mediante extrapolación lineal, como fue mostrado (VerFigura 2.14)

Las propiedades de la acción derivativa se ilustran (VerFigura 2.15), que muestra la simulación de un sistema con control PID. La ganancia del controlador y el tiempo de estimación se mantienen constantes con $K= 3$ y $T_i = 2$, y se varía el tiempo derivativo T_d . Para $T_d = 0$ se tiene un control PI puro. El sistema de lazo cerrado es oscilatorio con los parámetros elegidos. Inicialmente el amortiguamiento se incrementa con el incremento del tiempo derivativo, pero disminuye cuando el tiempo derivativo se vuelve más grande.

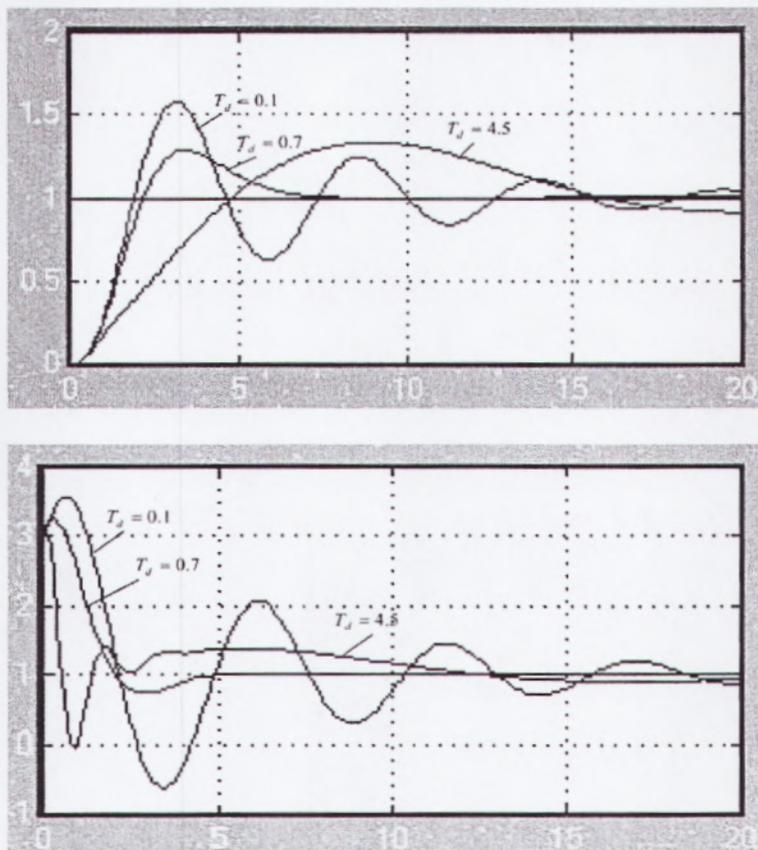


Figura 2.15: Simulación de un sistema en lazo cerrado con control proporcional, integral y derivativo.

2.3.7 IMPLEMENTACIÓN DIGITAL

Los controladores PID fueron originalmente implementados usando técnicas analógicas. Los antiguos sistemas usaban relés neumáticos, y otros dispositivos. Los motores eléctricos con relés, circuitos realimentados y amplificadores operacionales fueron usados con posterioridad. Muchas de las características como el windup y las derivaciones de la salida del proceso, en lugar del error de control, fueron incorporadas como trucos en la implementación. En la actualidad, es una práctica común implementar controladores PID usando microprocesadores, tal que algunos de los viejos trucos han sido redescubiertos.

Muchos aspectos deben ser considerados con respecto a las implementaciones digitales. Los más importantes tienen que ver con el muestreo, la discretización y la cuantización.

- MUESTREO

Cuando se usa una computadora digital para implementar una ley de control, todo el procesamiento de señales se hace en instantes de tiempo discretos. La secuencia de operaciones es como sigue:

1. Esperar una interrupción proveniente de un reloj de tiempo real.
2. Leer la entrada analógica a través del puerto de entrada/salida.
3. Calcular la señal de control, utilizando el algoritmo de control preestablecido.
4. Enviar la salida analógica al puerto de entrada/salida.
5. Actualizar las variables del controlador que serán utilizadas en el siguiente instante de muestreo.
6. Ir al paso 1

Las acciones de control están basadas en los valores de la salida del proceso, obtenidos solamente en instantes discretos en el tiempo. Este procedimiento se llama "muestreo". El caso normal es que las señales son muestreadas periódicamente con periodo h . El mecanismo de muestreo introduce algunos fenómenos no esperados que deben ser tomados en cuenta en una buena implementación digital de un controlador PID.

- DISCRETIZACIÓN

Para implementar una ley de control de tiempo continuo, como un controlador PID, en una computadora digital, es necesario aproximar las derivadas y las integrales que aparecen en la ley de control.

- CUANTIZACIÓN Y LONGITUD DE PALABRA

Una computadora digital permite sólo precisión finita en los cálculos. Algunas veces es difícil implementar el término integral en computadoras con longitud de palabra corta.

Para evitar el redondeo a cero del término de corrección, es necesario tener una precisión de al menos 20 bits. Aún así, se requieren más bits para obtener valores numéricos significativos. La situación es particularmente importante cuando se usa un motor a pasos que produce incrementos. Entonces, es necesario recurrir a trucos especiales para evitar el redondeo a cero de la integral. Una forma simple es usar un periodo de muestreo largo en el término integral. Por ejemplo, si se usa un periodo de muestreo de 1 segundo en vez de 0.02 segundos en el ejemplo anterior, será suficiente una precisión de 14 bits.

2.3.8 ASPECTOS OPERACIONALES

Prácticamente todos los controladores pueden correr en dos modos: manual o automático. En el modo manual la salida del controlador es manipulada directamente por el operador, típicamente presionando botones que incrementan o disminuyen la salida del controlador. Un controlador puede también operar en combinación con otros controladores, como en el caso de una conexión en cascada, o con elementos no lineales, como multiplicadores o selectores. Esto incrementa los modos operacionales. Los controladores también tienen parámetros que pueden ser ajustados durante la operación. Cuando existen cambios de modos y parámetros, es esencial evitar los transitorios de conmutación. La forma en que se haga la conmutación de modos y los cambios en los parámetros depende de la estructura del controlador elegido.

- TRANSFERENCIA SUAVE ENTRE MODOS MANUAL Y AUTOMÁTICO

Puesto que el controlador es un sistema dinámico, es necesario asegurar que el estado del sistema sea el correcto cuando se conmute el controlador entre modos manual y automático. Cuando el sistema está en modo manual, el algoritmo de control produce una señal de control que puede ser diferente de la señal de control manualmente generada. Entonces, es necesario asegurar que las dos salidas coincidan al tiempo de conmutar. Esto se llama "transferencia suave" (bumpless).

- TRANSFERENCIA SUAVE EN EL CAMBIO DE PARÁMETROS

Un controlador es un sistema dinámico. Un cambio en los parámetros de un sistema dinámico naturalmente resulta en cambios de su salida. Los cambios en la salida pueden ser evitados, en algunos casos mediante cambios simultáneos del estado del sistema. Los cambios en la salida también dependerán de la realización elegida. Con un controlador PID se requiere que no existan cambios drásticos en la señal de salida, cuando se cambian los parámetros y el error es cero.

2.3.9 MODELOS DE SIMULACIÓN

EJEMPLO 1: MODELO DE SIMULACIÓN CONTROL PROPORCIONAL

El siguiente modelo muestra corresponde al modelo de simulación del sistema de control en lazo cerrado con control proporcional utilizado para obtener las gráficas (VerFigura2.11)

El modelo permite obtener dos gráficas:

- La gráfica de las curvas de respuesta de la salida de la planta con respecto a la referencia.
- La gráfica de las curvas la señal de control calculada por el controlador, respectivas a cada una de las curvas de la gráfica anterior.

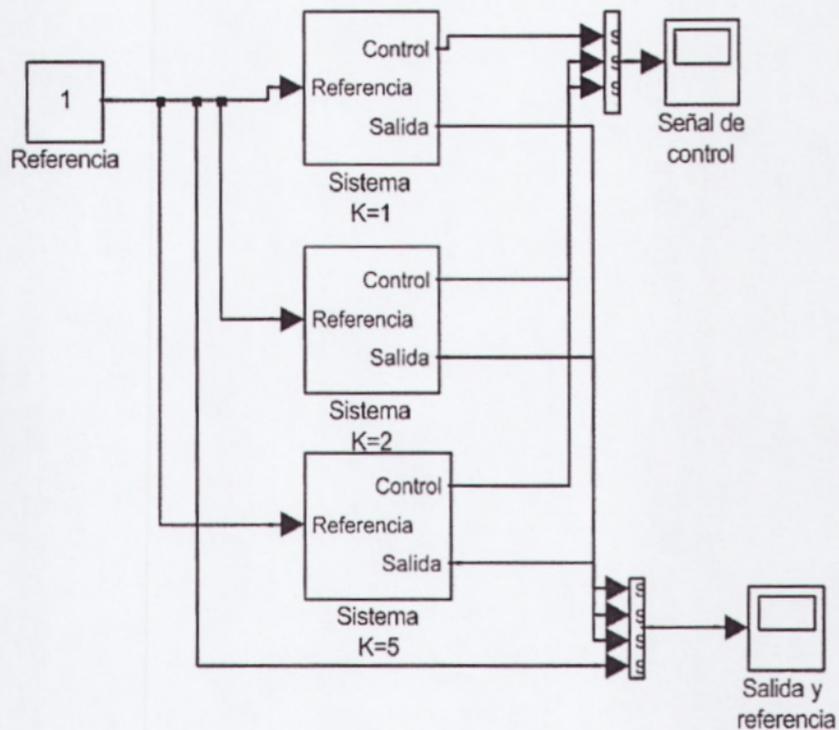


Figura 2.16: Diagrama Simulado de Control Proporcional

Por su parte, el control proporcional (P), está modelado por:

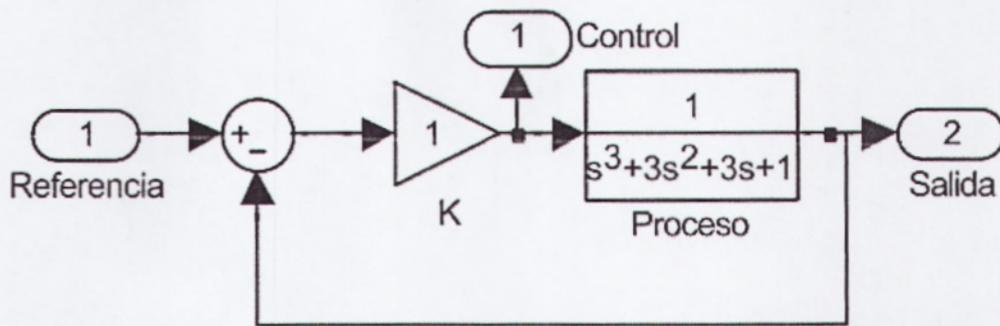


Figura 2.17: Diagrama Modelado de Control Proporcional

EJEMPLO 2: MODELO DE SIMULACIÓN CONTROL PROPORCIONAL E INTEGRAL

El modelo de simulación de un sistema en lazo cerrado con control proporcional e integral, utilizado para obtener las gráficas de la (VerFigura 2.13), se muestra a continuación en las siguientes figuras 2.18 y 2.19:

El modelo permite obtener dos gráficas:

- La gráfica de las curvas de respuesta de la salida de la planta con respecto a la referencia.
- La gráfica de las curvas de la señal de control calculada por el controlador, respectivas a cada una de las curvas de la gráfica anterior.

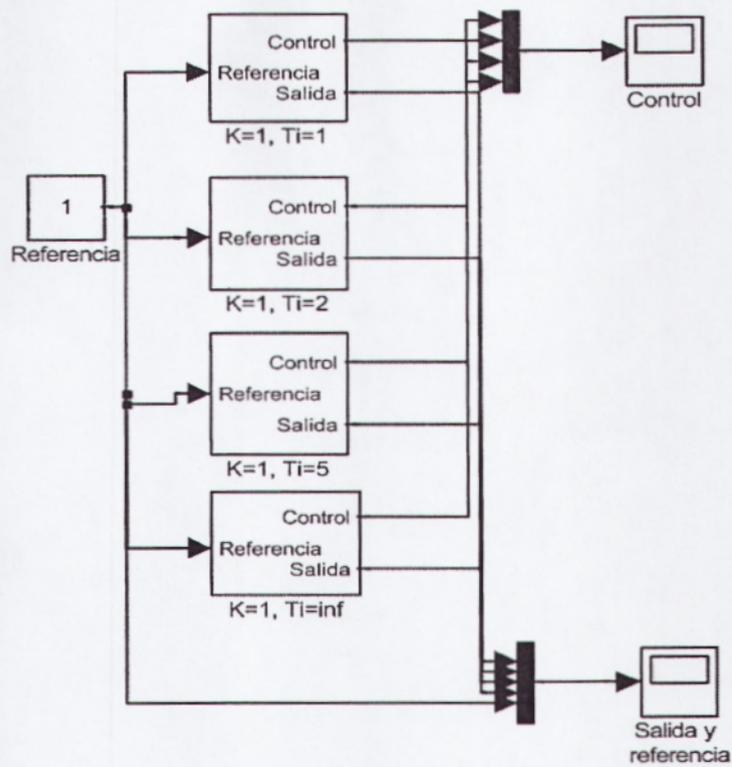


Figura 2.18: Diagrama Simulado de Control Proporcional e Integral

El sistema de control Proporcional e Integral (PI) está modelado como se muestra a continuación:

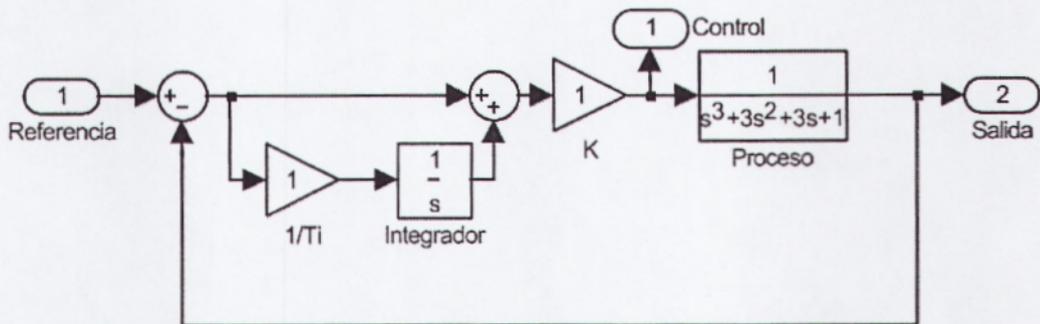


Figura 2.19: Diagrama Modelado de Control Proporcional e Integral

EJEMPLO 3: MODELO DE SIMULACIÓN FIGURA 15

El modelo de simulación de un sistema en lazo cerrado con control proporcional, integral y derivativo, utilizado en las gráficas de la (VerFigura 2.15), se muestra a continuación en las siguientes figuras 2.20 Y 2.21:

El modelo permite obtener dos gráficas:

- La gráfica de las curvas de respuesta de la salida de la planta con respecto a la referencia.
- La gráfica de las curvas de la señal de control calculada por el controlador, respectivas a cada una de las curvas de la gráfica anterior.

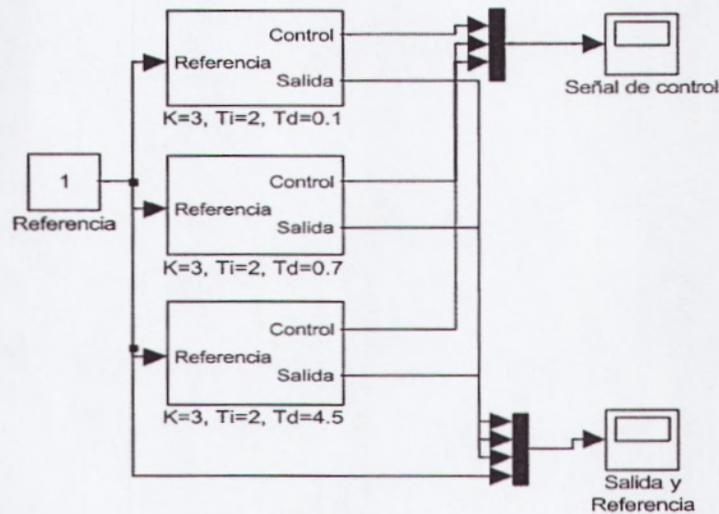


Figura 2.20: Diagrama Simulado de Control Proporcional, Integral y Derivativo.

El sistema con control Proporcional Integral y Derivativo (PID), está modelado como se muestra a continuación:

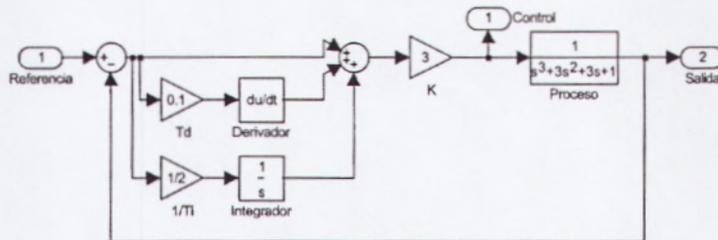


Figura 2.21: Diagrama Modelado de Control Proporcional, Integral y Derivativo.

2.3.10 CONTROL PID DE TEMPERATURA

El siguiente sistema que se expone como ejemplo consiste en un controlador PID analógico, basado en amplificadores operacionales de propósito general. El controlador está implementado de tal manera que fácilmente se distinguen cada una de las partes del controlador: sumador, acción proporcional, acción integral y acción derivativa. El sistema también cuenta con una etapa de potencia, que utiliza el método de "paquetes de onda" para regular la potencia entregada a una parrilla eléctrica.

DESCRIPCIÓN DEL SISTEMA

En el siguiente gráfico (Ver Figura 2.22), se muestra un diagrama simplificado del sistema de control de temperatura. El sistema está constituido por un controlador PID analógico, una etapa de potencia, una parrilla eléctrica y un sensor de temperatura.

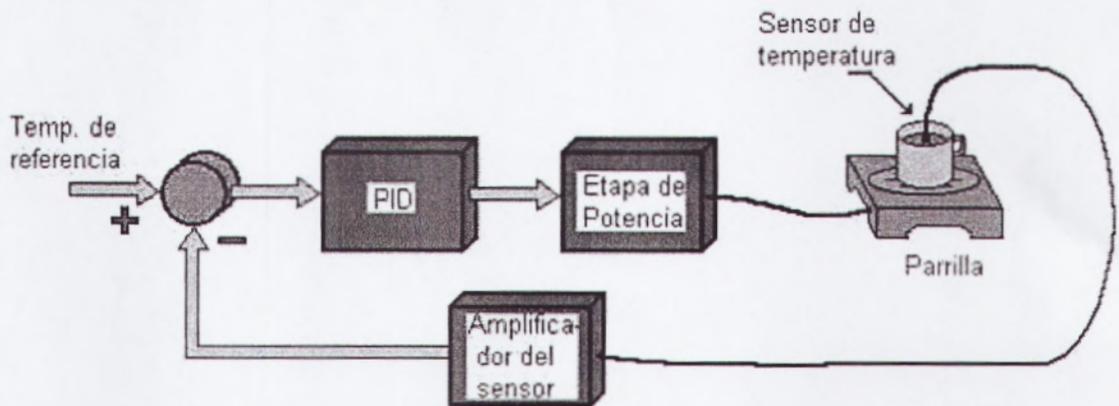


Figura 2.22: Sistema de Control de Temperatura.

CONTROLADOR PID: El controlador utilizado es un controlador PID paralelo, el cual está constituido por una etapa de acción de control proporcional, una etapa derivativa y una etapa integral, de acuerdo con el diagrama (ver Figura 2.23).

El controlador tiene entonces una función de transferencia dada por:

$$U(s) = K_p \left(1 + \frac{1}{T_i s} + T_d s \right) E(s)$$

Donde K_p es la constante de acción proporcional, T_i es la constante de tiempo de la acción integral, T_d es la constante de tiempo de la acción derivativa, $E(s)$ es la señal de error y $U(s)$ es la señal de control.

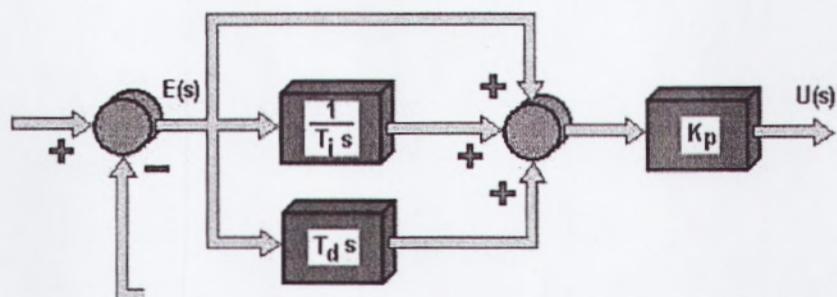


Figura 2.23: Controlador PID paralelo.

En la (VerFigura 2.24), se muestra un diagrama simplificado de la implementación electrónica utilizada, la cual consta de cuatro etapas. Por facilidad de implementación, se utilizaron circuitos integradores y Derivadores inversores, así como un sumador inversor y un circuito de control de ganancia (acción proporcional, K_p) inversor. Esto genera tres etapas inversoras, y para compensar el signo, se añadió un amplificador diferencial que calcula el error, pero con signo negativo. De esta manera, el circuito da la apariencia de tener retroalimentación negativa, pero en realidad no es así.

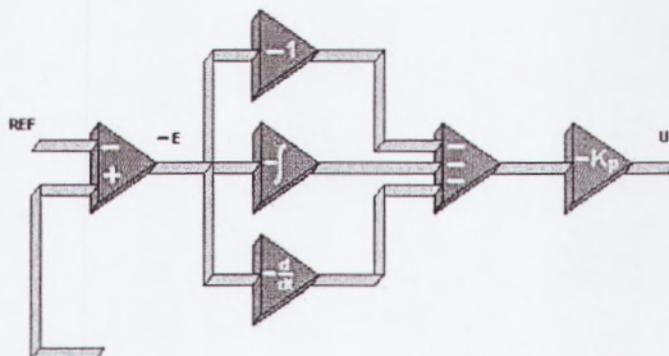


Figura 2.24: Esquema Simplificado del Controlador PID Electrónico.

ETAPA DE POTENCIA: En la (VerFigura 2.25), se puede observar el diagrama de bloques de la etapa de potencia. Ésta consiste de un generador de tiempo proporcional, una etapa de aislamiento y un interruptor de estado sólido (triac). El circuito generador de tiempo proporcional recibe la señal de salida del controlador PID (señal de control) y la transforma en un pulso cuya duración es proporcional a la señal de control.

La etapa de aislamiento está formada por un optoacoplador MOC3030, el cual cuenta con un circuito detector de cruce por cero. Finalmente, la etapa de potencia cuenta con un triac, con el cual se controla el voltaje de línea que se aplica a la parrilla.

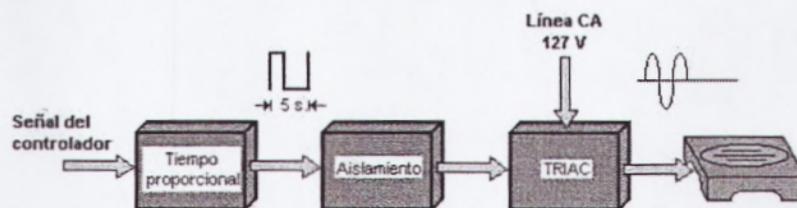


Figura 2.25: Diagrama de Bloque de la Etapa de Potencia.

CARACTERIZACIÓN DEL SISTEMA : Para caracterizar el sistema se escogió el método de la curva de reacción, debido a que el sistema térmico presenta características dinámicas que nos permiten aproximarlos a un sistema de primer orden con tiempo muerto como el dado por la siguiente expresión

$$\frac{Y(s)}{X(s)} = \frac{Ke^{-ts}}{\tau s + 1}$$

donde K es la constante de ganancia del proceso, t es la constante de tiempo y q es el tiempo muerto.

El método de la curva de reacción consiste de los siguientes pasos:

1. Permitir que el proceso alcance estado estacionario
2. Introducir un cambio tipo escalón en la variable de entrada.
3. Recolectar datos de la entrada y respuesta de salida hasta que el proceso nuevamente alcance estado estacionario.
4. Realizar el cálculo gráfico de los parámetros a partir de la curva de reacción.

En la (VerFigura 2.26), se muestra la curva de temperatura de la parrilla. Una vez que se ha estabilizado la temperatura de la parrilla, lo cual ocurre alrededor de los 44 °C, se incrementa la acción de control un 33% y a partir de ahí se genera una curva de reacción mediante la cual se definen los parámetros de aproximación del proceso. De acuerdo con la figura 2.26 se pueden estimar $t_1= 10$ minutos y $t_2= 18$ minutos; por lo tanto, aplicando las expresiones (7) y (8) se tienen $t=12$ minutos y $yq=6$ minutos.

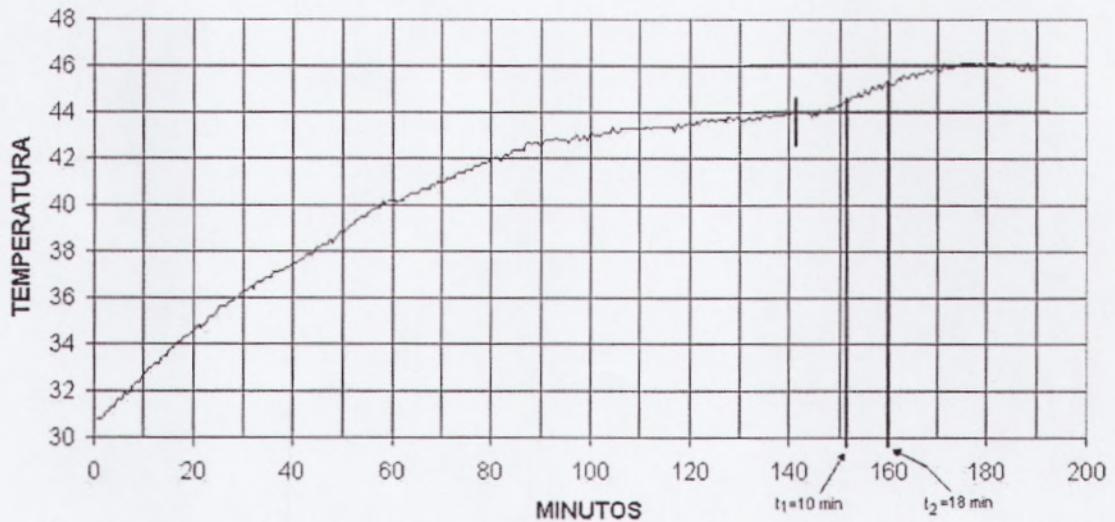


Figura 2.26: Curva de Reacción.

En la Figura 2.27 se muestra una comparación de los datos experimentales con el modelo matemático encontrado para la respuesta de la planta.

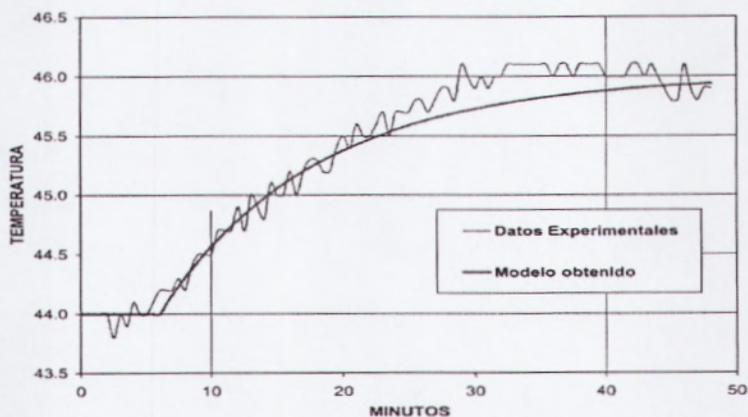


Figura 2.27: Comparación del Modelo Obtenido.

RESULTADOS: Se implementó el controlador PID analógico, y junto con el sensor y la etapa de potencia se realizó una prueba experimental, la cual se muestra en la (Ver Figura 2.28). Como puede observarse, el controlador permite reducir el tiempo de asentamiento de la temperatura. Existe un pequeño sobrepaso, el cual puede reducirse mediante un proceso adicional de sintonización manual.

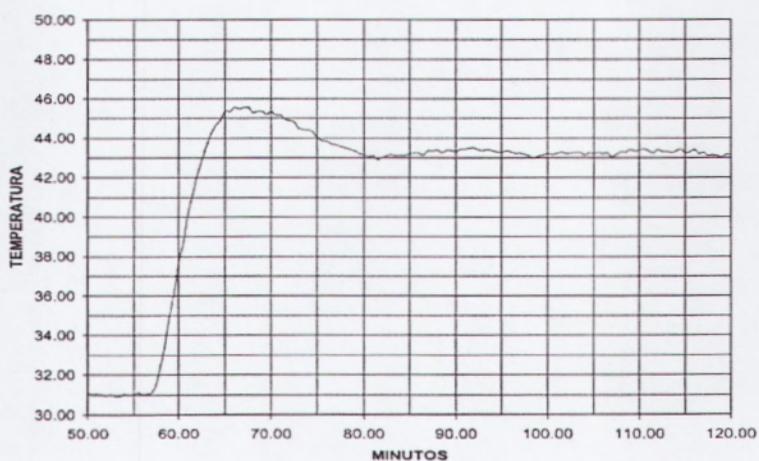


Figura 2.28: Respuesta del Sistema con el Controlador PID.

CAPÍTULO III. PLANTA DE ENTRENAMIENTO

3.1 CONCEPTO EPC

El Entrenador de Planta de Control “EPC” es una placa electrónica que incluye varios sensores y actuadores típicos en los sistemas de instrumentación y control tales como temperatura, velocidad, posición, señales analógicas de corriente continua, alterna, digital, y tren de pulsos.

El objetivo de este equipo es facilitar el aprendizaje de conceptos de teoría de control e instrumentación al poner a disposición del usuario varios experimentos prácticos listos para usar. De esta forma se minimiza el tiempo de diseño y construcción electrónico, se asegura la compatibilidad de los sensores con los experimentos, y se obtiene una experiencia de primera mano con las características y problemas de los sistemas físicos reales tales como ruido, precisión, acoplamiento AC/DC, etc. En lugar de usar simulaciones por computadora.

Además, habilita la metodología de Aprendizaje Activo (aprender por medio del desarrollo de proyectos prácticos) que aporta significativamente al aprendizaje que usando exclusivamente medios teóricos tales como libros de texto, dictados, y resolución de ejercicios.

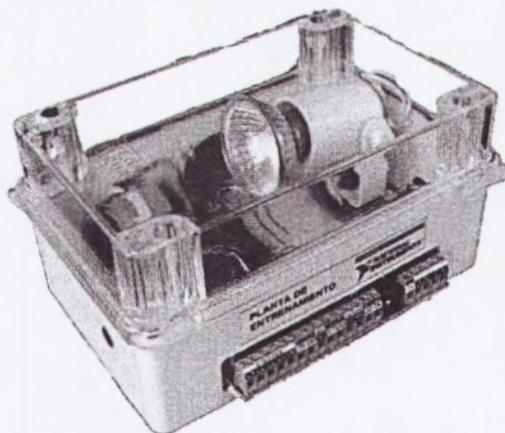


Figura 3.1: Planta de Entrenamiento

3.2 FUNCIONES EPC

Mediante el uso de la EPC (Ver Figura 3.1), el estudiante podrá desarrollar experimentos de adquisición y procesamiento de señales análogas/digital de control automático, programación, realizando la demostración de temas teóricos utilizando una tarjeta de adquisición de datos tipo USB-6009; con las que cuenta la Facultad, y el software LabVIEW de National Instruments del cual existe licencia académica.

Además se podrán desarrollar prácticas de laboratorio donde se puedan analizar y lograr experiencias en el manejo de procesos de control al proporcionar plantas físicas reales que funcionan con señales típicas. Esto ayudara de gran manera al estudiante a realizar tomas de decisión como si estuviera en el campo real de labores.

Los experimentos que contiene el EPC son los siguientes:

- Control de Temperatura
- Control de Velocidad de Motor DC
- Control de Posición de Motor Stepper
- Análisis de Sonido (adquisición y análisis de señales AC en Dominio del tiempo y de la frecuencia)
- Relé de Propósito General

3.2.1 MEDICIÓN Y CONTROL DE TEMPERATURA

Temperatura: medición, filtrado, control manual, control PID mediante PWM, introducción de perturbaciones.

El EPC incluye un bombillo halógeno que produce calor cuando se enciende. Este elemento simula un dispositivo de calentamiento tipo On/Off como puede ser una níquelina, o una válvula todo-nada.

Un sensor de temperatura convierte la señal de calor en una señal de voltaje según la siguiente ecuación.

$$^{\circ}\text{C} = V * 100$$

Donde:

$^{\circ}\text{C}$ es la temperatura en grados celcius

V es el voltaje que entrega el sensor de temperatura

100 es una constante numérica

Un ventilador instalado frente al halógeno permite sacar el aire caliente del EPC, introduciendo también una perturbación en el sistema térmico.

3.2.2 MEDICIÓN Y CONTROL DE VELOCIDAD

Velocidad: Caracterización Voltaje-Velocidad de un motor DC para comprobar linealidad, control manual, demostración de sintonización de parámetros PID mediante el segundo método de Ziegler-Nichols, experimentación de diferentes valores de ganancias del controlador PID.

El EPC incluye un motor de corriente continua (Motor DC) en cuyo eje está acoplado un encoder de 36 pulsos por revolución para medir la velocidad. El motor es controlado por una señal de voltaje DC que puede variar entre 0 y 5 voltios. La salida del encoder es una señal pulsante.

El EPC incluye un amplificador electrónico lineal que permite ampliar el voltaje de control de un máximo de 5V a un máximo de 12V, además de entregar la corriente necesaria para el funcionamiento del Motor DC.

3.2.3 CONTROL DE POSICIÓN

Posición: control de motor de pasos (Stepper), secuencias de energización de bobinas.

El EPC incluye un motor de pasos unipolar (Stepper) de cuatro entradas digitales disponibles en los terminales L0, L1, L2 y L3. Conforme se activan las líneas digitales en determinadas secuencias, el motor gira paso a paso. El EPC también incluye un encoder de 36 pulsos por revolución, de modo que cada pulso representará 10°

3.2.4 ANÁLISIS DE SONIDO

Análisis de señales AC: Adquisición de datos de un micrófono a alta velocidad, análisis de espectros, detección de evento de teclado DTMF.

El EPC incluye un micrófono que permite adquirir la señal de sonido mediante la tarjeta DAQ. Estas señales de tipo corriente alterna, y el procesamiento que se realiza es un análisis de espectros y otras mediciones típicas.

3.2.5 RELÉ DE PROPÓSITO GENERAL

El EPC incluye un relé electromecánico para aplicaciones de conmutación de propósito general. Este relé es tipo SPDT, por lo tanto incluye un contacto común C, un contacto normalmente abierto NO (por "Normally Open"), y un contacto normalmente cerrado NC (por "Normally Closed"). Para controlar la bobina IN (por "Input") se la alimenta con una señal tipo TTL.

3.3 DIAGRAMA ESQUEMÁTICO DE LA PLANTA DE ENTRENAMIENTO

3.3.1 ESQUEMÁTICO DEL HALÓGENO

Circuito que permite encender el bombillo halógeno de la planta de entrenamiento.

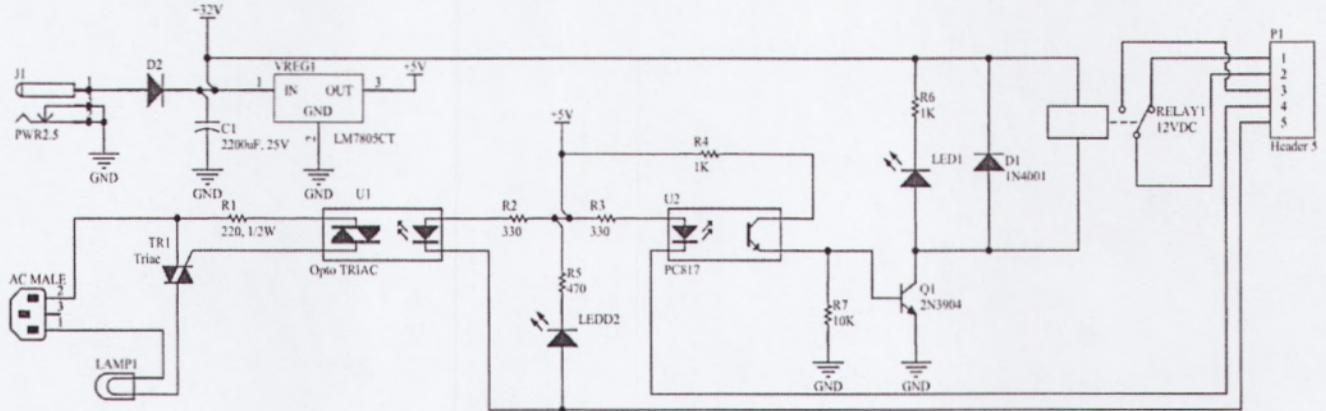


Figura 3.2: Esquemático del Halógeno

3.3.2 ESQUEMÁTICO DEL ENCODER

Circuito que permite el uso del encoder de la planta de entrenamiento.

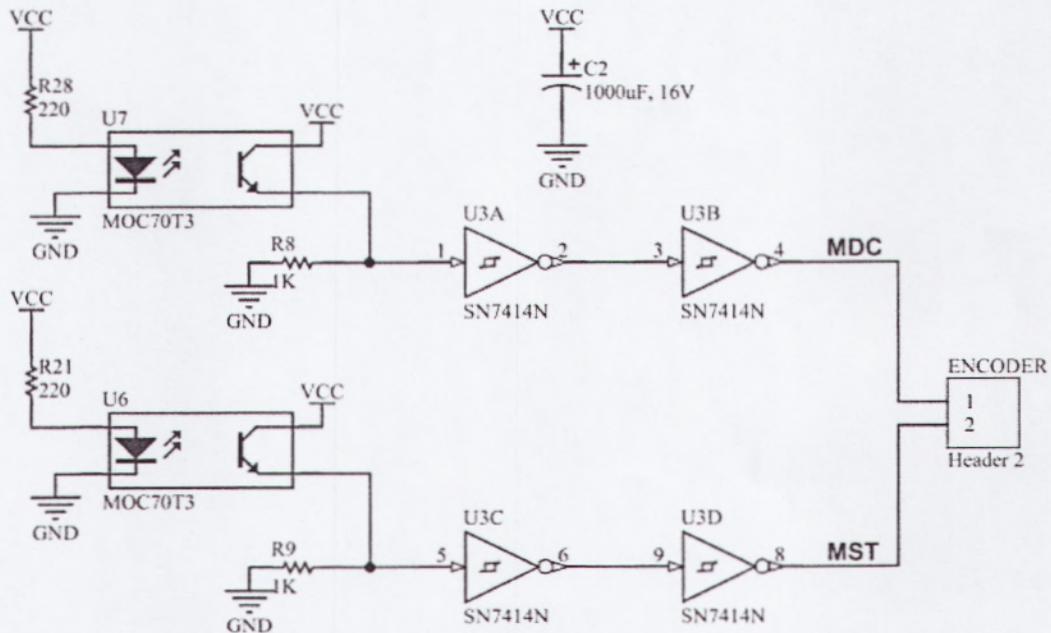


Figura 3.3: Esquemático del Encoder

3.3.3 ESQUEMÁTICO DEL MICRÓFONO

Circuito que permite el uso del micrófono de la planta de entrenamiento.

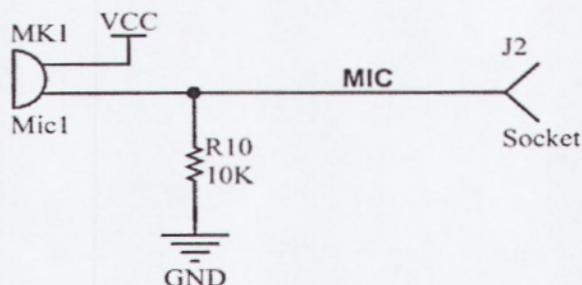


Figura 3.4: Esquemático del Micrófono

3.3.4 ESQUEMÁTICO DEL MOTOR DC

Circuito que permite el uso del motor DC de la planta de entrenamiento.

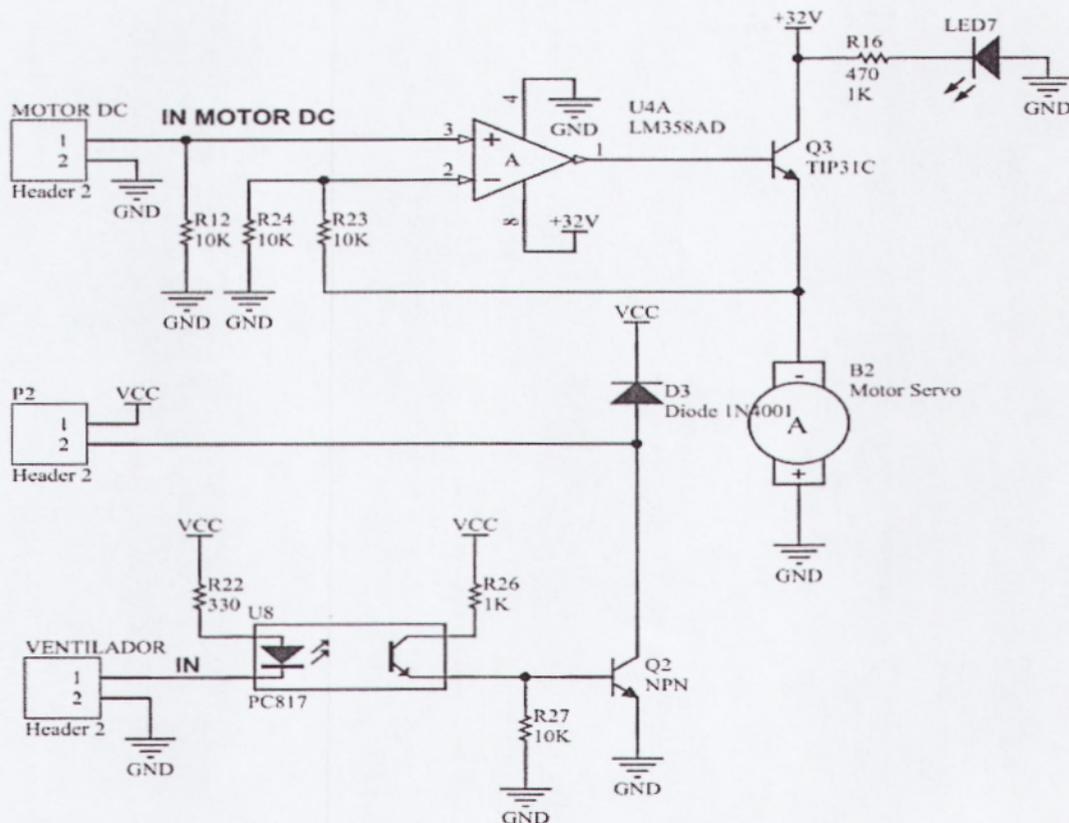


Figura 3.5: Esquemático del Motor DC

3.3.5 ESQUEMÁTICO DEL RELÉ

Circuito que permite el uso del relé de la planta de entrenamiento.

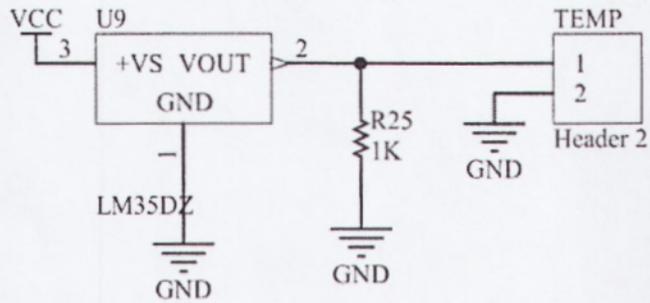


Figura 3.6: Esquemático del Relé

3.3.6 ESQUEMÁTICO DEL MOTOR STEPPER

Circuito que permite el uso del motor stepper de la planta de entrenamiento.

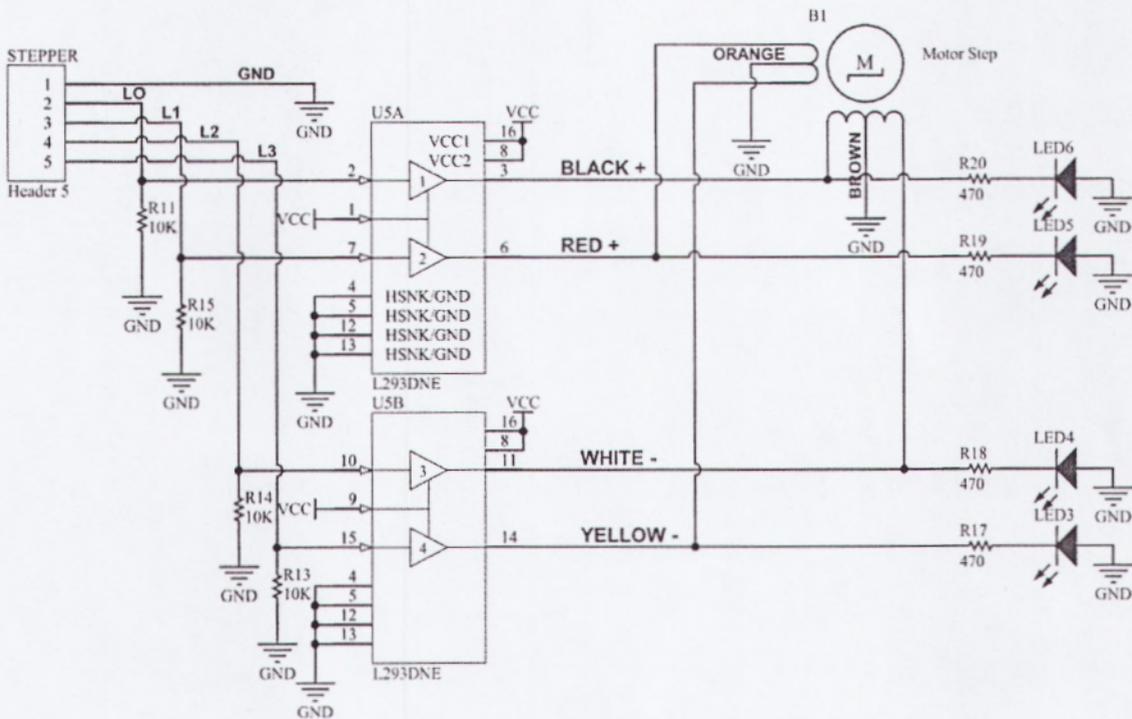


Figura 3.7: Esquemático Motor Stepper

3.4 DIAGRAMA TOPOLÓGICO Y CONEXIONES BÁSICA

A continuación se presenta un diagrama esquemático ubicando las partes principales del equipo (Ver Figura 3.8)

NOTA IMPORTANTE: Antes de conectar una fuente de poder al equipo lea completamente esta sección, pues dependiendo del equipo de control que esté usando, puede encenderse en condiciones que conduzcan a calentamiento y posibles daños a los circuitos, e incluso leves quemaduras en la piel.

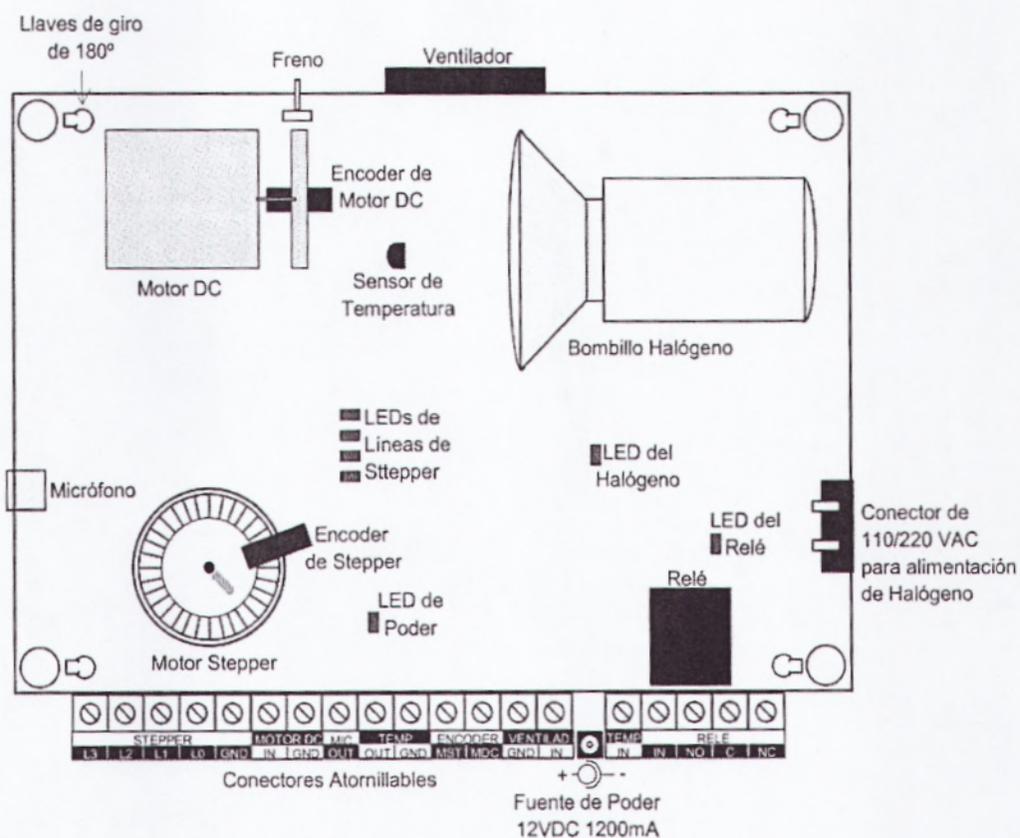


Figura 3.8: Diagrama Esquemático EPC (Layout o Topológico)

El LED(LED1), indicador de presencia de alimentación; se enciende cuando se conecta la Fuente de Poder de 12VDC. La polarización está indicada en el diagrama esquemático (pin interno positivo, carcasa negativo).

El Halógeno(Conectado a P2, pin 2)funciona con lógica inversa; es decir, cuando se envía un FALSE a la línea de control del Halógeno **TEMP IN**, se enciende el LED de Halógeno. Si el cable de alimentación del Halógeno está conectado al Conector de 110/220 VAC, el Halógeno se encenderá.

NOTA IMPORTANTE: *Cuando se enciende el Halógeno durante pocos minutos, éste se calentará considerablemente, pudiendo causar leves quemaduras en la piel y daños en el equipo. Para evitarlo, desconecte el cable de poder de 110/220 VAC para quitar la alimentación al Halógeno. También puede activar la línea de control del Halógeno desde un programa de LabVIEW o desde el utilitario de la tarjeta DAQ Measurement And Automation Explorer.*

El Relé funciona con lógica inversa; es decir, cuando se envía un FALSE a la línea del Relé, éste conmutará el terminal común C con el terminal normalmente abierto NO (por sus siglas en inglés, Normally Open).

Las bobinas del Motor StepperB1(Motor Paso a Paso) L0, L1, L2, L3 tienen lógica directa; es decir, cuando se envía una señal TRUE a una línea del Stepper, la bobina correspondiente se activa. Si al conectar la tarjeta DAQ se enciende uno o más de los LEDs de Líneas del Stepper, desconecte inmediatamente la Fuente de Poder de 12VDC, envíe un FALSE a todas las líneas digitales de la tarjeta DAQ desde el utilitario de la tarjeta DAQ Measurement and Automation Explorer, tal como muestra la figura (en este caso se asume que se han

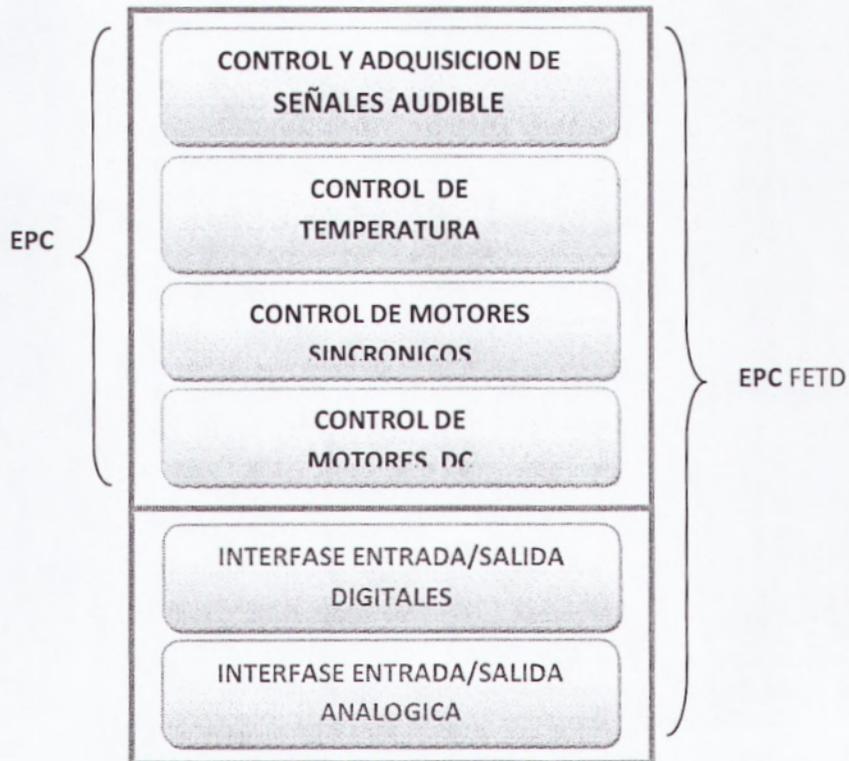
conectado las cuatro líneas del puerto P1 de la tarjeta DAQ a las líneas del Stepper).

El ventilador funciona con lógica inversa; es decir, cuando se envía una señal FALSE el ventilador se enciende.

En caso de que se requiera abrir la tapa de acrílico transparente que protege el EPC, se deben girar las cuatro Llaves de Giro a la posición de apertura; es decir, 180 grados en sentido contrario al giro de las manecillas del reloj. Estas llaves no son tipo tornillo, por lo tanto, es necesario no forzar el giro de las mismas.

3.5 DISEÑO FINAL PROPUESTO EPC PARA LAFETD

Diseño en diagrama de Bloques del entrenador de Planta de Control electrónico "EPC" que se deja propuesto para mejoras:



3.6 DISEÑO ELECTRÓNICO PROPUESTO EPC PARA LA FETD

El diseño esquemático del EPC FET que se deja propuesto toma en cuenta las salidas digitales del USB-6009 las cuales van a poder manejar cualquier voltaje y corriente debido a que la salida se ha puesto una separación galvánica a través de un acoplador óptico y tiene dos posibilidades de conexión:

- A través del transistor Driver de salida para corriente cercanas al amperio.
- A través de los Borners del Relay para corrientes mayores



Figura 3.9: Diagrama Esquemático EPC para la FETD

CAPÍTULO IV. PRÁCTICAS PARA EL ENTRENADOR

4.1 INTRODUCCIÓN A PRÁCTICAS EPC

Este capítulo explica la forma básica de utilización del EPC, pero no pretende ser un texto de teoría de control, instrumentación, o programación de LabVIEW.

Cada una de estas prácticas incluye una breve descripción del hardware y uno o más programas de aplicación desarrollados en LabVIEW, ordenados por cada una de las características que brinda el EPC. Estos programas se presentan con el código abierto de forma que puedan ser utilizados como base para programas más complicados, o como casos de estudio.

4.2 PRÁCTICAS DE MEDICIÓN Y CONTROL DE TEMPERATURA

Estas prácticas deben ser realizadas estrictamente según las instrucciones, pues de lo contrario las temperaturas generadas podrían dañar el equipo, e incluso provocar quemaduras en la piel si entra en contacto directo con el bombillo Halógeno. Por lo tanto se debe utilizar la tapa protectora de acrílico transparente debidamente asegurada y cuidar de no sobrepasar los 70 grados celcius de temperatura durante más de 2 minutos.

4.2.1 PRÁCTICA T1: MEDICIÓN DE TEMPERATURA

En esta práctica se medirá la temperatura interna del EPC mediante el sensor de temperatura integrado.

1. Realizar las siguientes conexiones entre el EPC y la tarjeta DAQ:
 - a. Conectar la salida del sensor de temperatura del EPC TEMPOUT a la entrada analógica de la tarjeta DAQ. En este manual se asume que se está utilizando el canal de entrada analógica número 5 en configuración RSE, con un rango de 0-5 V para medir la señal del sensor, de modo que el conector de la tarjeta DAQ será el AI5. Se puede utilizar cualquier otro canal, ya sea en configuración RSE o Diferencial. En caso de tener dudas sobre sistemas de adquisición de datos consulte a National Instruments a cerca de las opciones de capacitación.
 - b. Conecte la salida de referencia GND del EPC al GND de la tarjeta DAQ. Todos los terminales GND del EPC son comunes, al igual que todos los terminales GND de la tarjeta DAQ USB-6009.
 - c. Conectar la fuente de poder al EPC

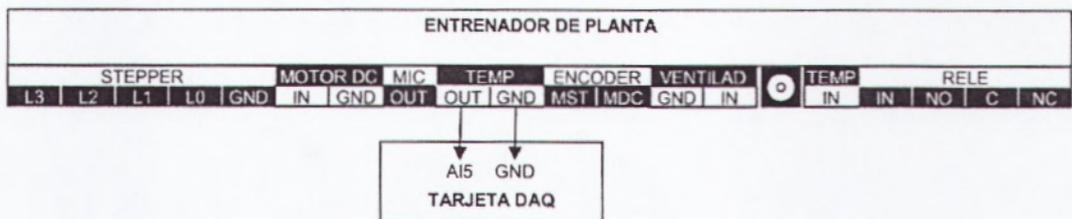


Figura4.1: Conexión Práctica T1

2. Se desarrolla el programa Termómetro.vi. (Ver Figura4.2) En el software de Programación LabVIEW. Este programa muestra la Temperatura medida tanto en un indicador tipo termómetro con display digital, como en un gráfico tipo Chart. Para esto se realiza los siguientes pasos:
 - a. En la pantalla del Panel Frontal se dibuja un Indicador Grafico, para esto seleccionamos de la paleta de controles el menú Modern, Graph, WaveForm Chart.
 - b. Se dibuja un indicador numérico como Termómetro, para esto seleccionamos de la paleta de controles el menú Modern, Numeric, Thermometer.
 - c. Se dibuja un botón de Stop, esto lo realizamos al seleccionar de la paleta de controles el menú Modern, Boolean, Stop Button.

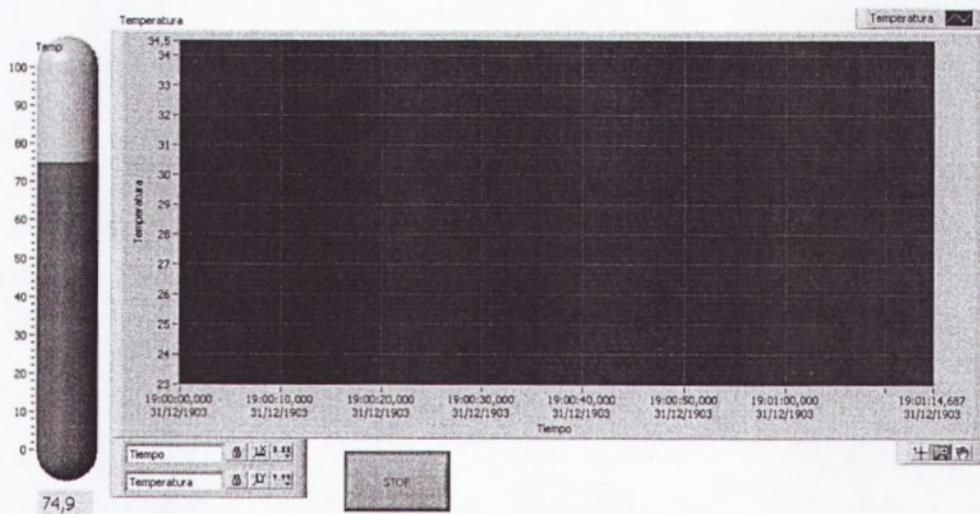


Figura4.2: Panel Frontal Práctica T1

3. Abrir el diagrama de bloques. Se utiliza un Timed Loop con un Período de 100ms en lugar de un While Loop para mejorar la temporización cuando se ejecuten programas de control.

Tomando en cuenta las especificaciones del sensor, se multiplica el voltaje que entrega el sensor por una constante de 100 y se obtiene la temperatura en grados celcius. Para esto se realiza los siguientes pasos:

- a. Seleccione de la paleta de Funciones el menú Programing, Structure, Time Structure, Time Loop.
- b. Seleccione una entrada mediante un DAQ Assistant, elija de la paleta de funciones el menú Express, Input, DAQ Assistant y configúrelo de la siguiente manera:
 1. Seleccione Acquire Signals
 2. Seleccione Analog Input
 3. Seleccione Voltaje
 4. Utilice el dispositivo USB-6009
 5. Seleccione canal analógico N° 5
 6. Configure Modo de conexión RSE (referenciado sencillo)
 7. Seleccione Modo de muestreo en 1 Sample On Demand
 8. De click en OK.
- c. Seleccione una constante al dar click en el Menú programing, Numeric, multiply.

d. Realice las conexiones en el diagrama de bloques (VerFigura4.3)

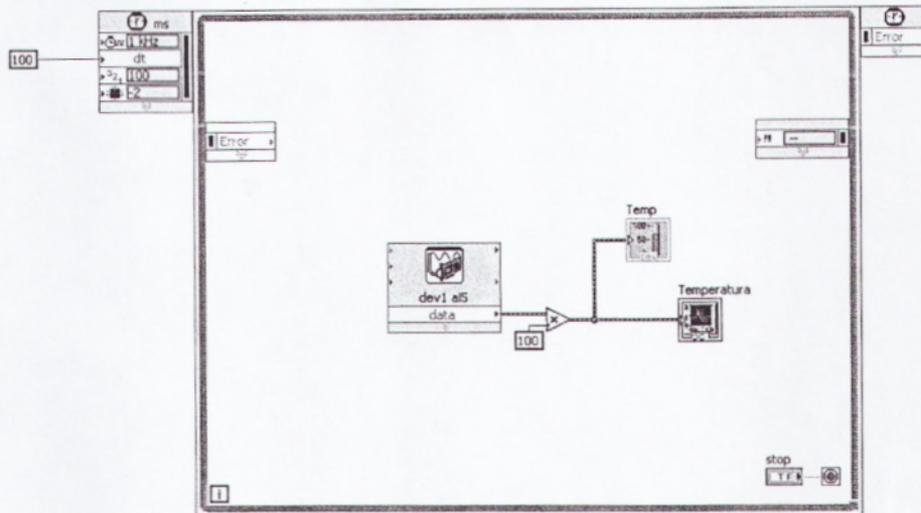


Figura4.3: Diagrama de Bloques Práctica T1

4. Cerrar el configurador presionando el botón de OK.
5. Correr el programa. Confirmar que la temperatura medida esté acorde con la temperatura ambiente dentro del EPC.
6. Notar que la señal de temperatura tiene una cantidad de ruido considerable, en este caso de más de un 1° celcius.
9. Parar el programa utilizando el botón de STOP.
10. Cerrar el programa sin hacer cambios.

4.2.2 PRÁCTICA T2: FILTRADO DE SEÑAL DE TEMPERATURA

En esta práctica se corregirá el problema del ruido de lectura de temperatura de la práctica anterior utilizando la técnica de promediación. Se tomarán 1000 muestras a una velocidad de 40,000 muestras por segundo y se promediarán para obtener una medición más precisa y libre de ruido.

1. Mantener las conexiones realizadas (*VerFigura4.1*)
2. Realizo el programa termómetro filtrado.vi. El panel frontal es similar al Programa de la práctica anterior.*Literales a,by c del Ítem 2 de la Practica T1, (VerFigura4.2)*
3. Abrir el diagrama de bloques. Este programa es similar al anterior, con la diferencia de que en la adquisición de datos se toman varias muestras de voltaje y luego se promedian. Esta sencilla técnica reduce el ruido en gran medida. Para esto se usa la herramienta Amplitud and Level Measurements.
 - a. Se aplica en la ventana de diagrama de bloques los literales **a, b, c** del Ítem 3, de la práctica T1.
 - b. Selecciono en paleta de funciones el menú Programing, WaveForm, WeveForm Measure, Amplitud and Level Measurements.
 - c. Realizo las conexiones en el diagrama de boques (*VerFigura4.4*)

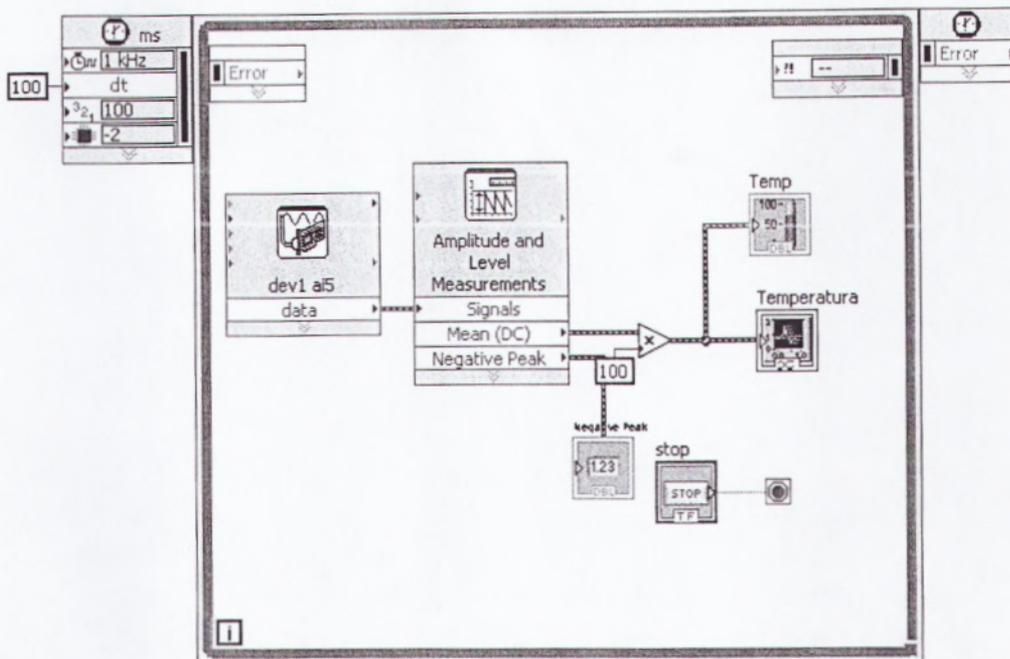


Figura4.4: Diagrama de Bloques Práctica T2

4. Se aplica el literal **b**, del Ítem 3 de la práctica T1 para la configuración de la tarjeta DAQ.
5. Nótese la configuración del canal. Es similar a la práctica anterior, con la diferencia de que el modo de adquisición es de múltiples muestras (N Samples), adquiriendo 1,000 muestras (Samples to Read: 1k) a una velocidad de 40,000 muestras por segundo (Rate: 40k). Así, esta herramienta entrega un arreglo de 1,000 muestras.
6. Cerrar el configurador presionando el botón de OK.
7. Correr el programa. Confirmar que la temperatura medida esté acorde con la temperatura ambiente dentro del EPC.
8. Notar que la señal de temperatura, aunque aún tiene ruido, éste está en el orden de los 0.02° grados celcius; es decir, mucho menor que en la práctica anterior.

9. Parar el programa utilizando el botón de STOP.

10. Cerrar el programa sin hacer cambios.

4.2.3 PRÁCTICA T3: CONTROL DE TEMPERATURA MANUAL

En esta práctica se controlará el encendido del bombillo halógeno manualmente desde el computador. También se controlará el encendido del ventilador, cuya función es sacar el aire caliente del EPC reduciendo la temperatura y creando una perturbación en el sistema. Para medir la temperatura se utiliza como base el programa usado en la práctica anterior.

1. Conectar el EPC a la tarjeta DAQ (*VerFigura4.5*) de la siguiente manera:
 - a. Confirmar que no esté conectada la fuente de poder del EPC ni el cable de alimentación del Halógeno.
 - b. Conectar una salida digital tipo TTL de la tarjeta DAQ (por ejemplo, P1.0) a la entrada de control del bombillo Halógeno en el EPC llamada TEMP IN.
 - c. Conectar una salida digital tipo TTL de la tarjeta DAQ (por ejemplo, P1.0) a la entrada de control del bombillo Halógeno en el EPC llamada TEMP IN.

- d. Conectar una segunda salida digital tipo TTL de la tarjeta DAQ (por ejemplo, P1.1) a la entrada de control del ventilador VENTILADOR IN.
- e. Conectar la salida del sensor de temperatura TEMP OUT a una entrada analógica de la tarjeta DAQ (por ejemplo, AI5).
- f. Conectar una referencia GND del EPC a una referencia GND de la tarjeta DAQ. Todos los terminales GND del EPC son comunes, al igual que todos los terminales GND de una tarjeta DAQ USB-6009.

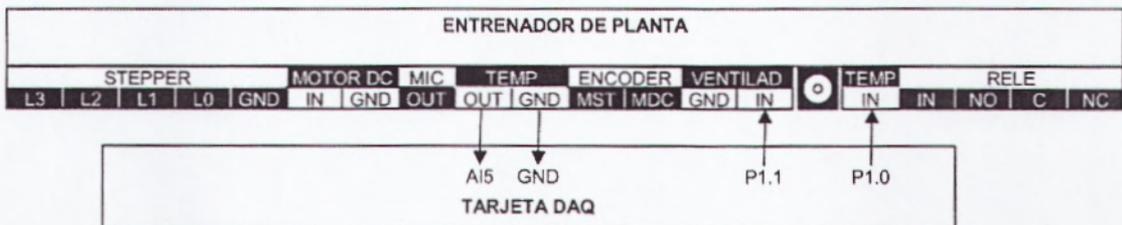


Figura4.5: Conexión de EPC a Usb-6009

- g. Conectar la fuente de poder del EPC y el cable de alimentación del bombillo halógeno ala toma de poder

NOTA IMPORTANTE: Si el bombillo se enciende inmediatamente, desconecte solamente el cable de alimentación del bombillo halógeno de 110/220 VAC para evitar que se caliente. Posteriormente el Halógeno se apagará manualmente desde el computador. Observando el **Led del Halógeno** en la tarjeta electrónica del EPC se puede detectar si la línea está activada o desactivada. Recordar que el Halógeno funciona con lógica inversa.

2. Se realiza el programa control manual temp y vent.vi. (VerFigura4.6)

En el panel frontal se dispone de un indicador de temperatura tipo Termómetro como en las prácticas anteriores. Igualmente, un gráfico tipo Chart despliega la temperatura, pero adicionalmente muestra también el encendido y apagado del Halógeno y del Ventilador.

Adicionalmente, el panel frontal dispone de dos controles tipo Switch, uno para comandar el Halógeno, y otro para comandar el ventilador. Para esto se realizan los siguientes Pasos:

- a. En la pantalla del Panel Frontal se dibuja un Indicador Grafico, para esto seleccionamos de la paleta de controles el menú Modern, Graph, WaveForm Chart.
- b. Se dibuja un indicador numérico como Termómetro, para esto seleccionamos de la paleta de controles el menú Modern, Numeric, Thermometer.
- c. Se dibuja un botón de Stop, esto lo realizamos al seleccionar de la paleta de controles el menú Modern, Boolean, Stop Button.
- d. Se dibuja dos Switch, uno para el ventilador y otro para el halógeno, esto se realiza al seleccionar de la paleta de controles el menú Modern, Boolean, vertical Toggle Switch.

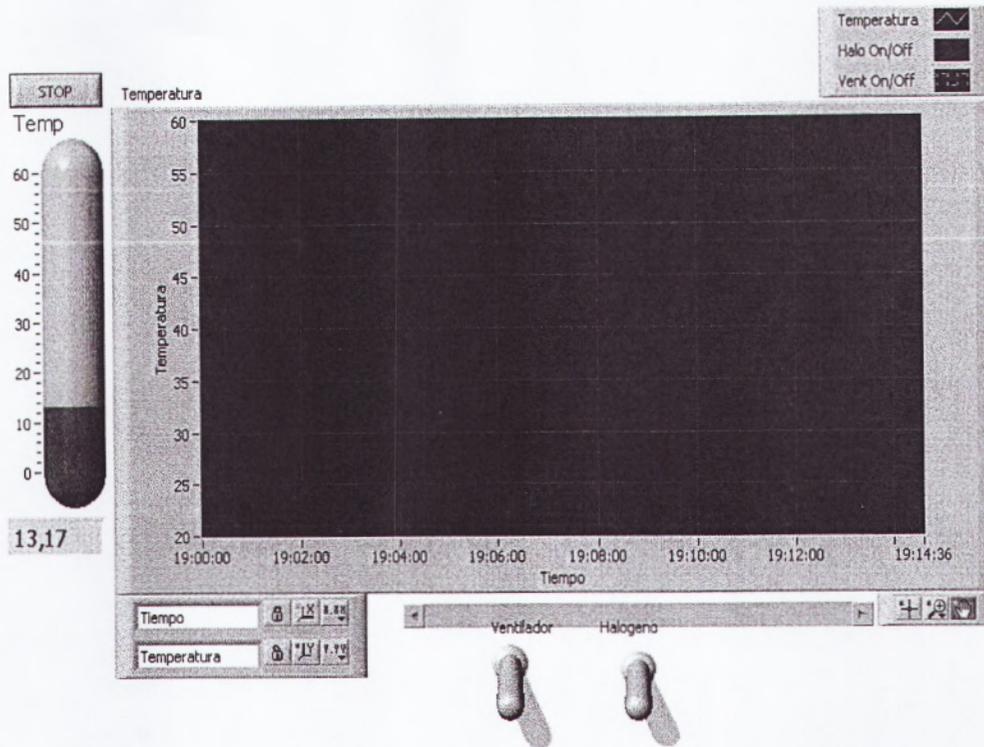


Figura4.6: Panel Frontal de Práctica T3

3. Realicelo siguiente en el diagrama de Bloques.
 - a. Se aplica los literales a y b, del Ítem 3 de la Práctica T2.
 - b. Selecciono de la paleta de funciones el menú Express, signal Manipulation, Convert From Dynamic Data.
 - c. Selecciono de la paleta de funciones el menú Express, signal Manipulation, Marge Signal.
 - d. Selecciono de la paleta de funciones el menú programing, Comparison, Greater?
 - e. Selecciono de la paleta de funciones el menú Programing, bololean, Or.

f. Realizo las conexiones en el diagrama de boques (VerFigura4.7)

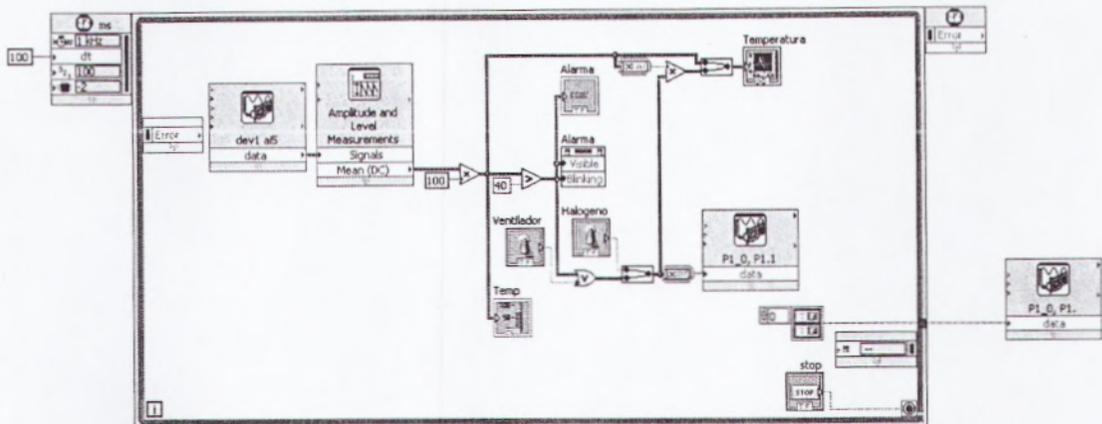


Figura 4.7: Diagrama de Bloque Programa T3

4. Correr el programa. Operar sobre los controles del Halógeno y del Ventilador para observar el comportamiento de la temperatura dentro del EPC.

NOTA IMPORTANTE: La temperatura se eleva rápidamente; puede llegar de 30° a 100° celcius en aproximadamente un minuto. Para evitar daños tanto en personas como en el equipo se recomienda no mantener la temperatura por encima de 70° celcius por más de dos minutos. Recordar que para estas prácticas es necesario mantener la tapa colocada y asegurada debidamente, y por ninguna razón entrar en contacto directo con el bombillo Halógeno.

5. El programa incluye una alarma que se despliega en caso de Sobre pasar los 70° celcius, pero no provee de control automático para bajarla. Es necesario apagar manualmente el Halógeno para bajar la temperatura.
6. Con este programa se puede observar (VerFigura4.8) la curva característica tipo exponencial de la respuesta al escalón unitario de un sistema de primer orden, como lo es este sistema térmico.

La respuesta no es exactamente exponencial debido a que el bombillo Halógeno no es un calefactor perfecto.

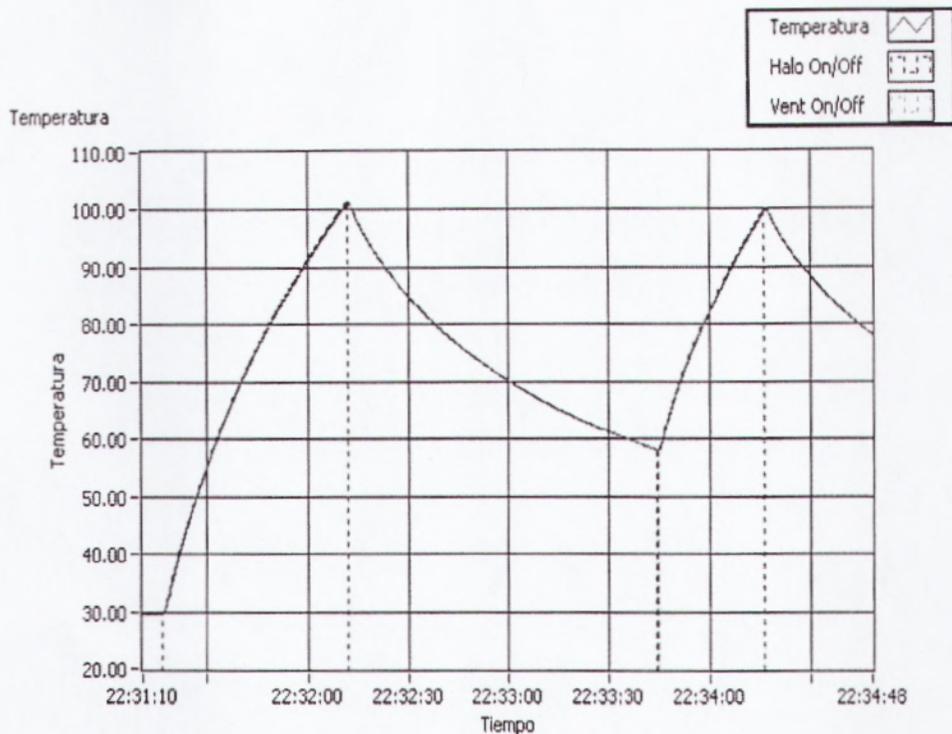


Figura4.8: Gráfica Tipo Exponencial

7. Parar el programa presionando el botón de STOP. Esto asegura que al terminar el programa tanto el Halógeno como el Ventilador se apaguen. Por lo general, los programas de LabVIEW no deben pararse con el botón de abortar, disponible en la barra de herramientas.
8. Abrir el diagrama de bloques para revisar el código del programa.
 - a. El DAQ Assistant de salidas digitales escribe a las líneas desde las cuales se controla el Halógeno y el Ventilador. Debido a que estos dispositivos operan con lógica inversa, la inversión de la línea se hace dentro del DAQ Assistant.

- b. El nodo de propiedad hace que la alarma se despliegue y parpadee solamente cuando la temperatura sea mayor a 70°C.
 - c. El estado del control del Halógeno y del Ventilador se multiplican por la temperatura y se envían a graficar en la Chart para poder apreciar el tiempo en que estas líneas se activan y desactivan.
9. Cerrar el programa sin hacer cambios, asegurándose que tanto el Halógeno como el Ventilador queden apagados.

4.2.4 PRÁCTICA T4: CONTROL DE TEMPERATURA ON/OFF

En esta práctica se realizará un control de temperatura tipo On/Off (Encendido/Apagado) utilizando el bombillo Halógeno como actuador, y el sensor de temperatura para medir la variable controlada.

1. Las conexiones de la tarjeta DAQ son las mismas que los literales del Ítem 1 de la práctica T3.
2. Se realiza el programa control temp on-off.vi. El panel frontal muestra la temperatura del EPC en un indicador tipo Termómetro. El operador debe seleccionar la temperatura deseada en el control Setpoint.

Si la temperatura del EPC está por debajo del Setpoint, el Halógeno se enciende automáticamente. Cuando la temperatura supera el Setpoint el Halógeno se apaga automáticamente.

Un indicador booleano Halo indica el estado del Halógeno.

La gráfica tipo Chart Historial de Temperatura muestra el valor de Temperatura, el Setpoint, y el estado del Halógeno.

Adicionalmente, el panel frontal presenta un control boleano tipo switch Ventilador para controlar el ventilador y acelerar el proceso de enfriado o introducir una perturbación.

3. Para esto se realizan los siguientes Pasos:(*VerFigura4.9*)
 - a. En la pantalla del Panel Frontal se dibuja un Indicador Grafico, para esto seleccionamos de la paleta de controles el menú Express, Graph Indicators, WaveForm Chart.
 - b. Se dibuja un indicador numérico como Termómetro, para esto seleccionamos de la paleta de controles el menú Modern, Numeric, Thermometer.
 - c. Se dibuja un indicador numérico como Setpoint, para esto seleccionamos de la paleta de controles el menú Modern, Numeric, Vertical Pointer Slide.
 - d. Se dibuja un botón de Stop, esto lo realizamos al seleccionar de la paleta de controles el menú Modern, Boolean, Stop Button.
 - e. Se dibuja un Switch, para el ventilador, esto se realiza al seleccionar de la paleta de controles el menú Modern, Boolean, vertical Toggle Switch.
 - f. Se dibuja un halógeno, para esto se selecciona de la paleta de controles el menú Modern, Boolean, Round Led.

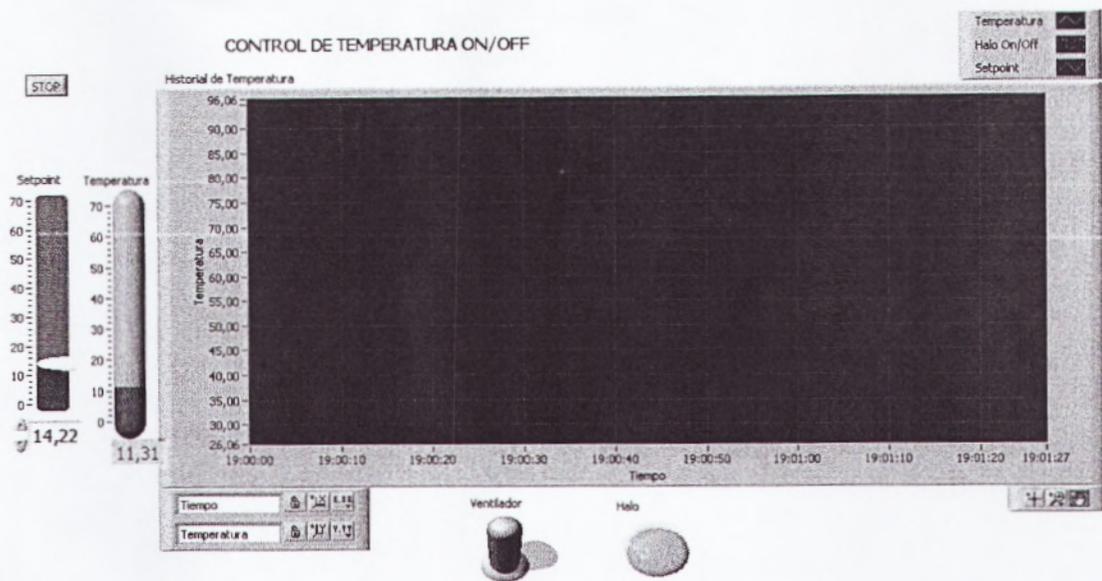


Figura 4.9: Panel Frontal Práctica T4

En la siguiente gráfica(VerFigura4.10), se presenta un ejemplo de control con tres niveles de Setpoint diferentes.



Figura 4.10: Niveles de Setpoint

Puede observarse el rizado típico en sistemas de control On/Off. Haciendo un Zoom en la gráfica puede medirse este rizado con mayor precisión. En este caso, el rizado es de aproximadamente 0.6° por aumento y 0.2° por defecto, dando un total de 0.8° .

4. Realice el diagrama de Bloques:

a. Se aplica los literales *a,b,c,d* y *e* del Ítem 3 de la Práctica T3.

b. Se realiza las conexiones en el diagrama de bloque (*VerFigura4.11*)

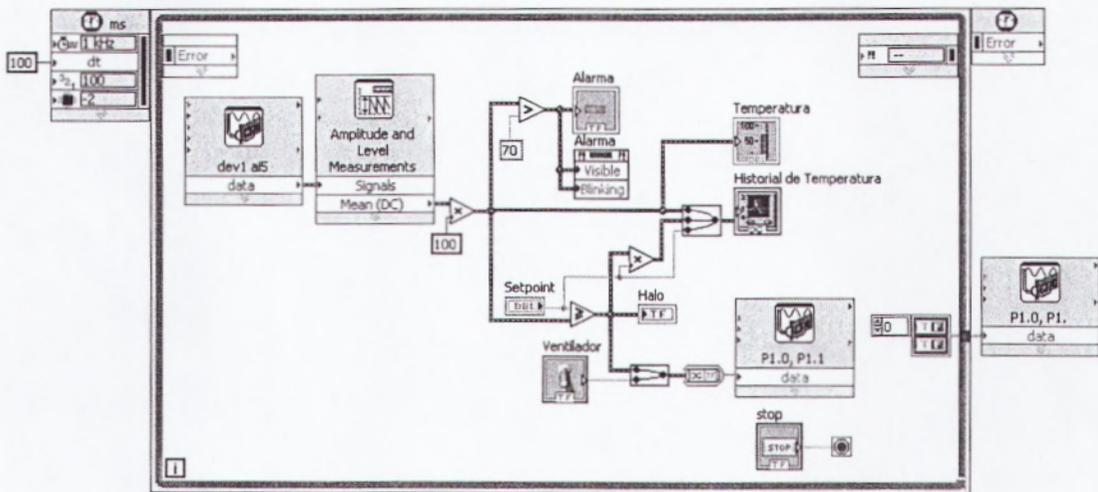


Figura4.11: Diagrama de Bloque Práctica T4

5. Parar el programa presionando el botón de STOP. Esto asegura que al terminar el programa tanto el Halógeno como el Ventilador se apaguen. Por lo general, los programas de LabVIEW no deben pararse con el botón de abortar disponible en la barra de herramientas.
6. Abrir el diagrama de bloques para revisar el programa. En general, el programa es similar al de la práctica T3.

- a. Si la temperatura del EPC es mayor que el Setpoint, el Halógeno se apaga. Cuando la temperatura es mayor que el Setpoint el Halógeno se enciende.
 - b. Cuando se presiona el botón de STOP el Timed Loop termina sus ciclos y se envían señales de FALSE al Halógeno y al Ventilador
7. Cerrar el programa sin hacer cambios, asegurándose que tanto el Halógeno como el Ventilador queden apagados.
 8. RETO: Implementar un control On/Off con brecha diferencial.

4.2.5 PRÁCTICA T5: CONTROL DE TEMPERATURA PID(Mediante PWM)

En esta práctica se mejorará el control de temperatura utilizando un controlador PID. Puesto que el bombillo halógeno tiene una operación tipo On/Off, se convertirá la salida analógica del PID en una señal PWM de baja frecuencia observable que será la señal de control del bombillo halógeno. En sistemas industriales más avanzados se utiliza un variador de voltaje, o un PWM de alta frecuencia.

La selección de los valores de las ganancias PID se puede obtener por varios métodos experimentales y analíticos. En este caso se trabajarán con diferentes valores de ganancias de K_p y T_i , manteniendo $T_d=0$, y observando el comportamiento de la variable controlada Temperatura.

Los resultados observados en la práctica pueden variar con los expuestos en este manual. Estas variaciones se deberían a varios factores como tiempo de vida del Halógeno, orientación del Halógeno hacia el sensor, temperatura ambiente, y perturbaciones como ráfagas de aire.

Estas prácticas se realizan con el ventilador apagado, pero puede encenderse para acelerar el descenso de temperatura o para introducir perturbaciones en el sistema.

1. Las conexiones de la tarjeta DAQ son las mismas que los literales del Ítem 1 de la práctica T3.

2. Realice un programa con nombre control temp pid.vi.
 - a. En el panel frontal se dispone de un indicador de temperatura tipo Termómetro y un control tipo Slide para establecer el valor de temperatura deseado Setpoint, además de un gráfico que muestra tanto la Temperatura como el Setpoint, similar a la práctica anterior. También incluye un control boleano para encender o apagar el Ventilador, y un indicador boleano que muestra el estado del Halógeno.

 - b. En este programa se dispone adicionalmente de un control que permite ingresar las Ganancias PID del sistema de control, y de un indicador tipo Slide que muestra la salida del controlador PID Out.

 - c. Se dibuja los mismo elementos en el panel frontal de los literales a, b, c, d, e, y f, aumentando un indicador Para el PID.

 - d. Se dibuja también un grupo de controles numéricos para la ganancia Pid. Esto lo realizamos seleccionando de la paleta de controles el menú Modern, Numeric, Numeric Control (*VerFigura4.12*)

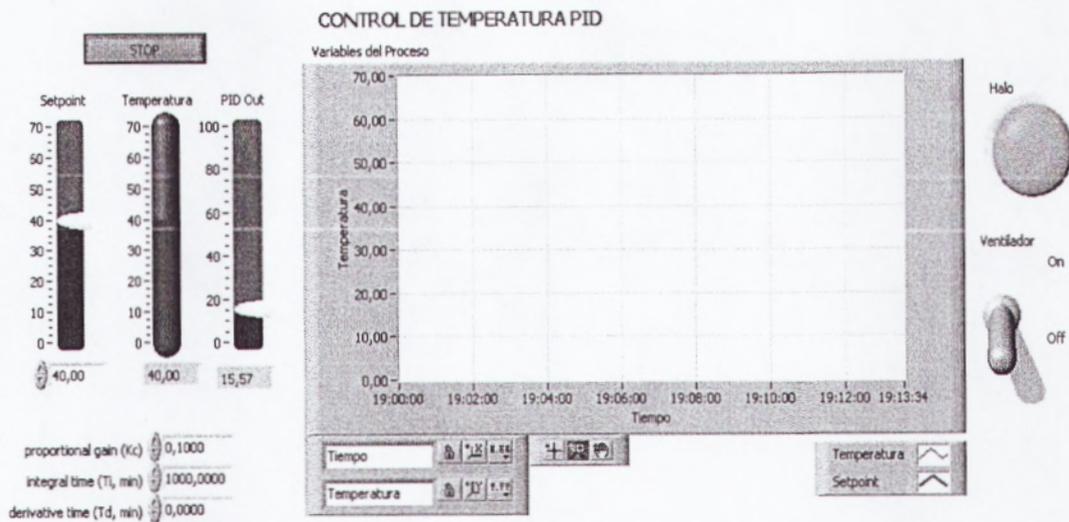


Figura4.12: Panel Frontal Práctica T5

3. Colocar el Setpoint a 60°C
4. Ingresar los siguientes valores en Ganancias PID.
 $K_c = 1.0$; $T_i = 1000.00$; $T_d = 0$
5. Se realiza el Diagrama de Bloques.
 - a. Se aplica el literal a del ítem 3 de la Práctica T4
 - b. Se utiliza la estructura de secuencia, para esto seleccionamos de la paleta de funciones el menú Programing, Structures, Stacked Sequence Structure.
 - c. Selecciono de la paleta de Funciones el menú Programing, Array, Build Array.
 - d. Selecciono de la paleta de Funciones el menú Programing, Comparison, Select.
 - e. Realizo las conexiones de la grafica. (VerFigura4.13)

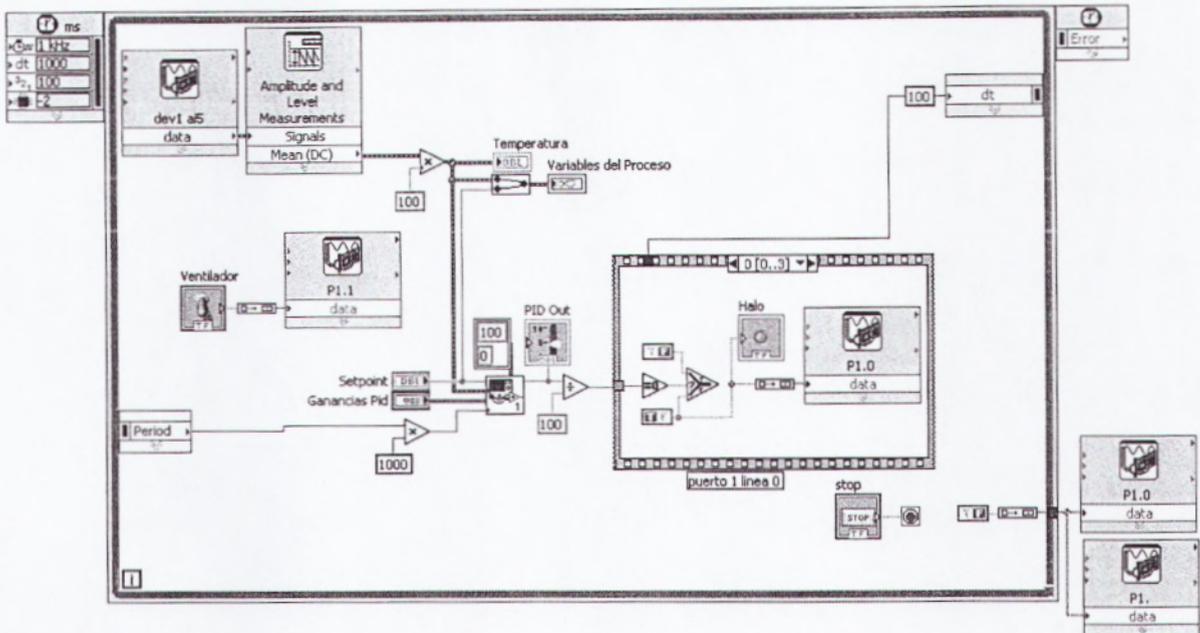


Figura4.13: Diagrama de Bloques de la Práctica T5

6. Correr el programa, y observar el funcionamiento del programa.
7. En un sistema PID se aplica una ganancia al error, consistente en la diferencia entre el valor deseado (Setpoint) y el valor de la variable (en este caso, la Temperatura). Se puede observar la salida del PID (PID Out) como un valor entre 0% y 100%, que regula el tiempo del pulso alto del PWM, cuya frecuencia es 100ms.
8. Como se puede observar (Fig. 38), con estos valores de parámetros PID el sistema es estable, pero oscila considerablemente, tendiendo a estabilizarse en el setpoint (en este caso, 60°C).

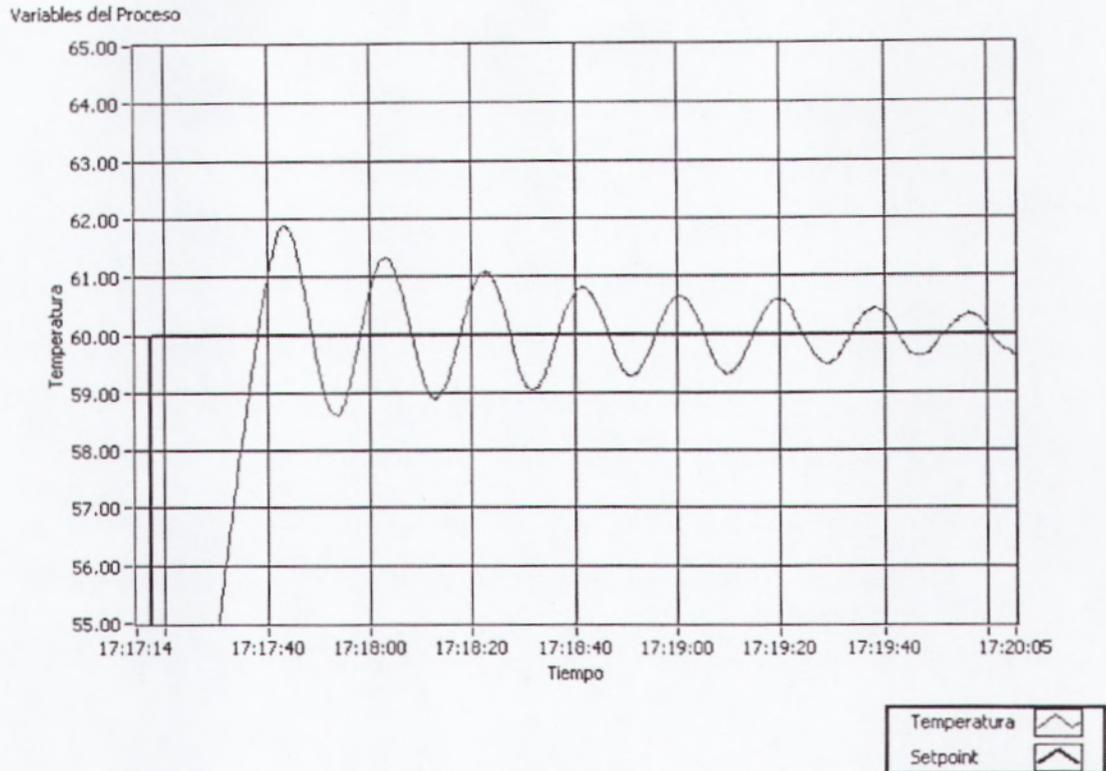


Figura4.14: Parámetros PID

9. Para observar el efecto de usar otras Ganancias PID es necesario antes reducir la temperatura a valores más cercanos que la temperatura ambiente.
 - a. Colocar el Setpoint a 20°C y encender el ventilador hasta que la temperatura baje al menos a unos 40°C.
 - b. Una vez que la temperatura se acerque a estos valores, apagar el ventilador. Es posible que la temperatura tienda a subir ligeramente por el calentamiento del Halógeno y del EPC en general.
10. Borrar la gráfica anterior haciendo click con el botón secundario del ratón sobre la gráfica Variables del Proceso y seleccionando en el menú desplegable la opción Clear Chart.

11. Cambiar las ganancias PID a los siguientes valores

$$K_c = 0.1 ; T_i = 1000.00 ; T_d = 0$$

12. Cambiar el Setpoint nuevamente a 60°C.

13. Observar la respuesta del sistema (Ver Figura 4.15)



Figura 4.15: Respuesta a Cambios Parámetros PID

14. Como se puede apreciar, la reducción de la ganancia proporcional produce una respuesta más amortiguada, aunque el tiempo de respuesta se incrementa.

15. Probar nuevamente la respuesta del sistema, esta vez con una ganancia proporcional aún menor. Para observar el efecto de usar otras Ganancias PID es necesario antes reducir la temperatura a valores más cercanos que la temperatura ambiente.

- a. Colocar el Setpoint a 20°C y encender el ventilador hasta que la temperatura baje al menos a unos 40°C.
 - b. Una vez que la temperatura se acerque a estos valores, apagar el ventilador. Es posible que la temperatura tienda a subir ligeramente por el calentamiento del Halógeno y del EPC en general.
16. Borrar la gráfica anterior haciendo click con el botón secundario del ratón sobre la gráfica Variables del Proceso y seleccionando en el menú desplegable la opción Clear Chart.
 17. Cambiar las ganancias PID a los siguientes valores
 $K_c = 0.1$; $T_i = 1000.00$; $T_d = 0$
 18. Cambiar nuevamente el Setpoint a un valor alto, por ejemplo 60°C.
 19. Observar la respuesta del sistema (Ver Figura 4.16)

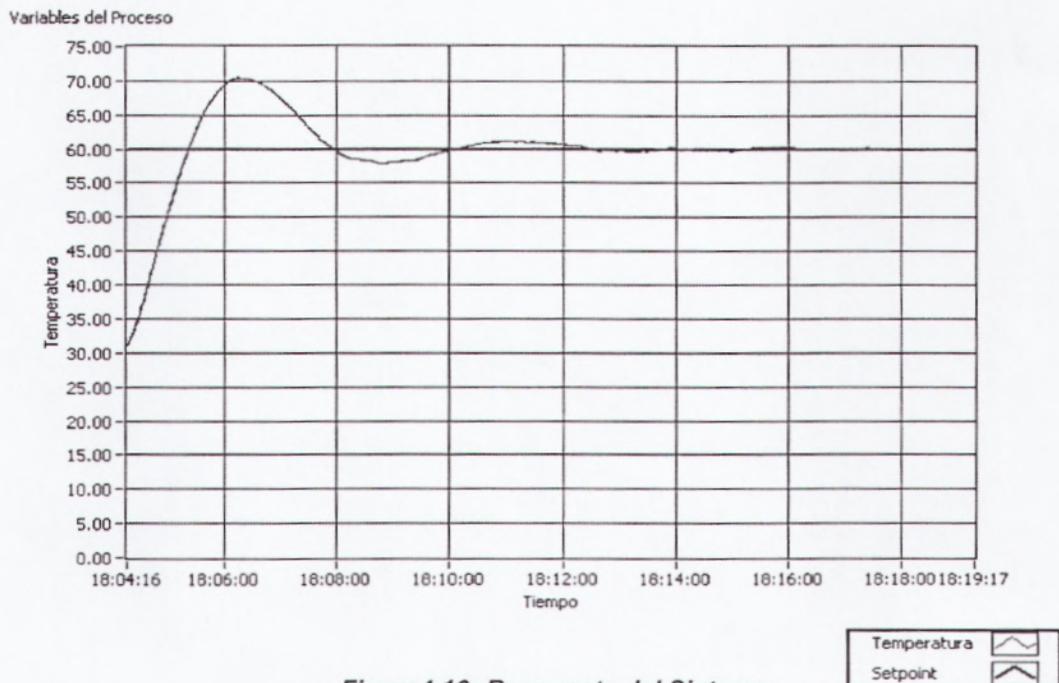


Figura 4.16: Respuesta del Sistema

20. Como se puede apreciar, la nueva reducción de la ganancia proporcional produce una respuesta aún más amortiguada que en el caso anterior.

21. También se puede observar que el error es menor que en control On/Off.

Los gráficos mostrados pueden diferir de los obtenidos experimentalmente en las prácticas, dependiendo de las variaciones de la temperatura ambiental, ráfagas de aire que ingresen por las ranuras del ventilador aún cuando éste esté apagado, tiempo de vida del Halógeno, etc.

De esta forma se han desarrollado prácticas de instrumentación y control de sistemas térmicos. El equipo puede también utilizarse conjuntamente con el Toolkit de Identificación de Sistemas de LabVIEW para modelar matemáticamente el sistema térmico, y el Toolkit de Diseño de Sistemas de Control para diseñar un controlador más eficiente.

4.3 PRÁCTICAS DE MEDICIÓN Y CONTROL DE VELOCIDAD

En estas prácticas se utilizarán programas para producir la señal de control de corriente continua desde la tarjeta DAQ. Típicamente estas tarjetas soportan señales de hasta 10V y de baja corriente, usualmente por debajo de los 5mA. En estas prácticas se utiliza un voltaje de control máximo de 5V. El EPC incluye un amplificador electrónico lineal que permite ampliar el voltaje de control de un máximo de 5V a un máximo de 12V, además de entregar la corriente necesaria para el funcionamiento del Motor DC.

4.3.1 PRÁCTICA M1: CONTROL MANUAL DE VELOCIDAD DE MOTOR DC

En esta práctica se utilizará un control numérico en el Panel Frontal para controlar el voltaje de alimentación del Motor DC, y así controlar manualmente la velocidad.

1. Realizar las siguientes conexiones entre el EPC y la tarjeta DAQ:
 - a. Conectar una salida analógica de la tarjeta DAQ a la entrada de control del Motor DC MOTOR DC IN. En este ejemplo se utiliza la salida analógica número cero (AO0) de la tarjeta instalada como Device 1.
 - b. Conectar la referencia GND de la tarjeta DAQ a la referencia GND del EPC. Todas las referencias GND de la tarjeta DAQ están conectadas internamente (en el modelo USB-6009 y similares); igualmente, todas las referencias GND del EPC están conectadas internamente.
 - c. Conectar la fuente de poder del EPC.

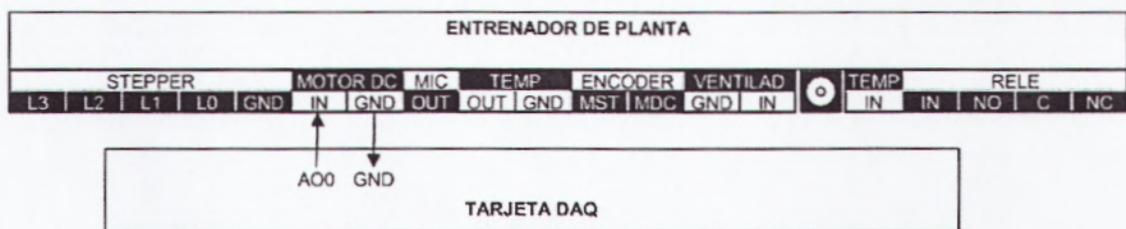


Figura4.17: Conexión tarjeta DAQ Práctica M1

2. Se realiza el programa control manual velocidad.vi. El panel frontal presenta los siguientes controles e indicadores:
 - a. El control numérico tipo Slide Voltaje es el voltaje de salida de la tarjeta DAQ, que puede tomar valores entre 0 y 5 voltios.

- b. El indicador gráfico tipo Chart Señales muestra un historial de los valores de voltaje que se han generado.
 - c. El botón de STOP detiene el programa. Es importante detener el programa utilizando este botón y no el botón de abortar de la barra de herramientas porque así se asegura que al parar el programa se detenga el motor, pues se envía un valor de 0V a la salida analógica.
3. Para esto se realiza los siguientes pasos:(*VerFigura4.18*)
- a. En la pantalla del Panel Frontal se dibuja un Indicador Grafico, para esto seleccionamos de la paleta de controles el menú Express, Graph Indicators, WaveForm Chart.
 - b. Se dibuja un indicador numérico velocidad, para esto seleccionamos de la paleta de controles el menú Modern, Numeric, Vertical Pointer Slide.
 - c. Se dibuja un botón de Stop, esto lo realizamos al seleccionar de la paleta de controles el menú Modern, Boolean, Stop Button.
 - d. Se dibuja un control numérico,Esto lo realizamos seleccionando de la paleta de controles el menú Modern, Numeric, Numeric Control



Figura 4.18: Diagrama Frontal Práctica M1

4. Se abre la ventana de Diagrama de bloque y se realiza lo siguiente:
 - a. Selecciono de la paleta de Funciones el menú programing, Structure, Time Structure, Time Loop.
 - b. Selecciono una entrada mediante un DAQ Asistant, escojo la paleta de funciones el menú Express, Input, DAQ Asistant.
 - c. Selecciono de la paleta de Funciones el Menú Express, signal Manipulation, Marge Signals
 - d. Realizo las conexiones del diagrama de bloque como la grafica. (VerFigura4.19)

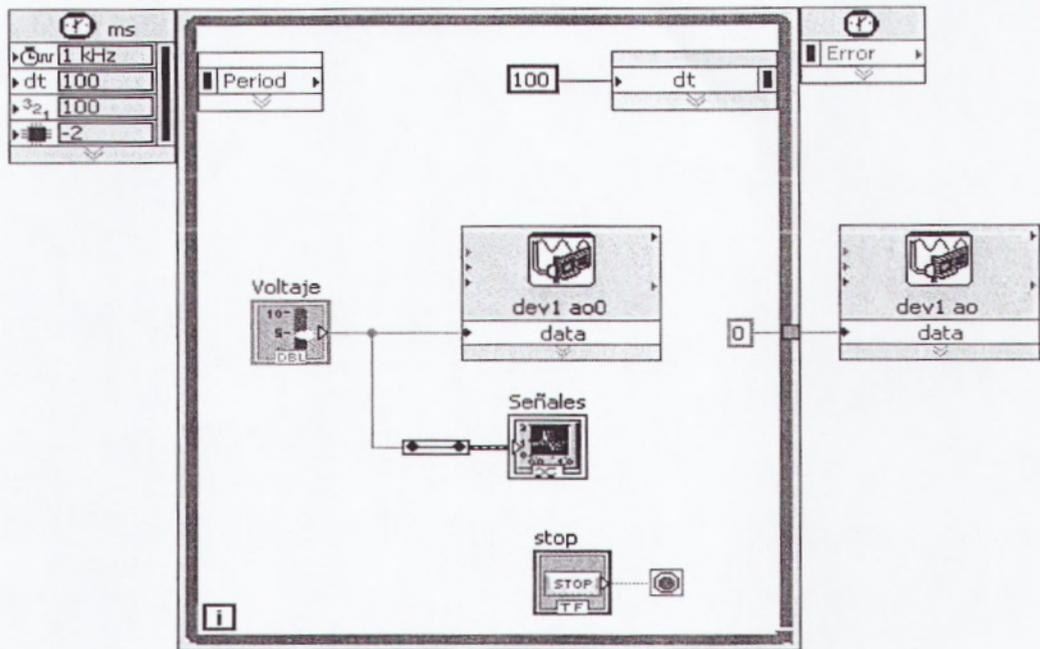


Figura4.19: Diagrama de Bloque Práctica M1

5. Cambiar el valor de Voltaje.
 - a. Observar que si el voltaje es relativamente pequeño (usualmente por debajo de 0.5V) el motor no se mueve. Esto se debe a que el campo magnético generado por la corriente que circula a bajo voltaje no es suficiente para vencer la fricción e inercia del eje.
 - b. Observar que mientras mayor es el voltaje, mayor es la velocidad del eje. En las prácticas posteriores se medirá esta velocidad.
6. Abrir el diagrama de bloques para examinar el programa. Nótese que al terminar la ejecución del lazo se envía un valor de cero a la salida analógica para parar el motor cuando se detenga el programa.
7. Cerrar el programa sin hacer cambios.

4.3.2 PRÁCTICA M2:SEÑAL DE ENCODER – FORMA ANALÓGICA

En esta práctica se adquirirá la señal de salida del encoder del Motor DC para analizarla en forma analógica. Además se medirá la frecuencia para determinar la velocidad del motor.

1. Realizar las siguientes conexiones entre el EPC y la tarjeta DAQ:
 - a. Conectar una salida analógica de la tarjeta DAQ a la entrada de control del Motor DC MOTOR DC IN. En este ejemplo se utiliza la salida analógica número cero (AO0) de la tarjeta instalada como Device 1.
 - b. Conectar la referencia GND de la tarjeta DAQ a la referencia GND del EPC. Todas las referencias GND de la tarjeta DAQ están conectadas internamente (en el modelo USB-6009 y similares); igualmente, todas las referencias GND del EPC están conectadas internamente.
 - c. Conectar la salida del encoder del Motor DC del EPC llamada ENCODER MDC a una entrada analógica de la tarjeta DAQ. En este ejemplo se utiliza la entrada analógica número cinco (AI5) en conexión tipo RSE.
 - d. Conectar la fuente de poder del EPC.

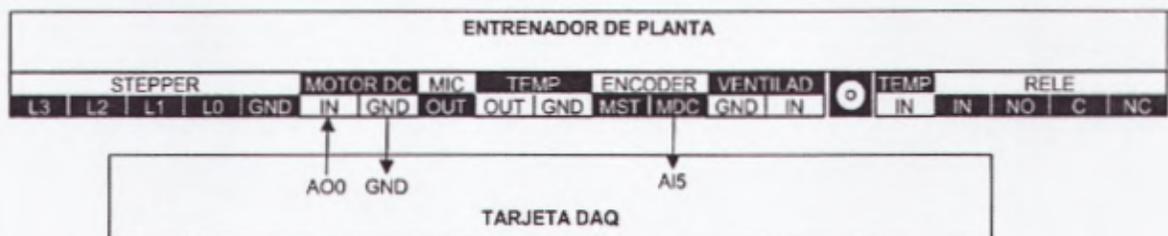


Figura4.20: Conexión Tarjeta DAQ Práctica M2

2. se realiza el programa medición analógica velocidad.vi. El panel frontal presenta los siguientes controles e indicadores:
- a. El control numérico tipo Slide Voltaje es el voltaje de salida de la tarjeta DAQ, que puede tomar valores entre 0 y 5 voltios.
 - b. El indicador gráfico tipo Chart Señales muestra un historial de los valores de voltaje y de velocidad que se han generado. La magnitud de estas señales están en Valores Por Unidad (VPU). Para el Voltaje el límite máximo es 5 Voltios. Para la Velocidad el límite máximo es 4500 RPM.
 - c. El indicador gráfico tipo Graph Señal Encoder muestra en forma de osciloscopio la señal de salida del Encoder que es tipo tren de pulsos. Obsérvese que mientras mayor es el Voltaje de control del Motor DC, mayor es la rapidez de giro del eje, y por lo tanto mayor es la frecuencia (menor es el período) de la señal del Encoder.
 - d. El indicador tipo Gauge Frecuencia muestra cuántos pulsos por segundo (Hz) tiene la Señal del Encoder, mientras que el indicador tipo Gauge Velocidad indica la velocidad del eje en revoluciones por minuto (RPM). El disco del encoder tiene 36 ranuras.
 - e. El botón de STOP detiene el programa. Es importante detener el programa utilizando este botón y no el botón de abortar de la barra de herramientas porque así se asegura que al parar el programa se detenga el motor, pues se envía un valor de 0V a la salida analógica.

3. Para esto se realiza los siguiente Pasos: (VerFigura4.21)

- a. En la pantalla del Panel Frontal se dibuja dos Indicador Grafico, para esto seleccionamos de la paleta de controles el menú Express, Graph Indicators, WaveForm Chart.
- b. Se dibuja un indicador numérico de velocidad, para esto seleccionamos de la paleta de controles el menú Modern, Numeric, Vertical Pointer Slide.
- c. Se dibuja un botón de Stop, esto lo realizamos al seleccionar de la paleta de controles el menú Modern, Boolean, Stop Button.
- d. Se dibuja un control numérico, Esto lo realizamos seleccionando de la paleta de controles el menú Modern, Numeric, Numeric Control.
- e. Se dibuja dos controles numéricos tipo Gauge, para esto seleccionamos de la paleta de controles el menú Modern, Numeric, Gauge.

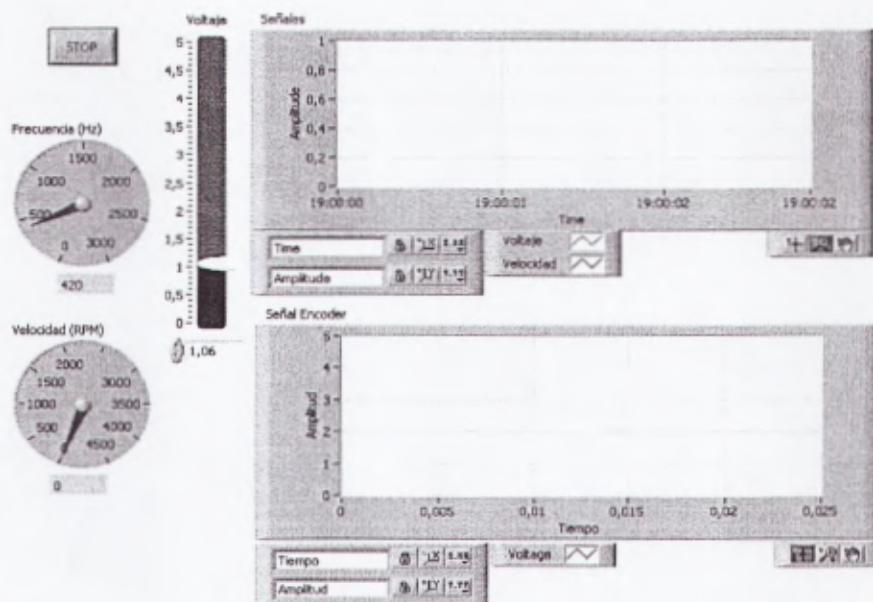


Figura 4.21: Panel Frontal Práctica M2

4. Se abre el diagrama de Bloques y se realiza lo siguiente:
 - a. Se aplica los literales a y b, del Ítem 3 de la Práctica M.
 - b. Selecciono en paleta de funciones el menú Programing, WaveForm, WeveForm Measure, Amplitud and Level Measurements.
 - c. Seleccione de la paleta de Funciones el menú Programing, Comparison, Select.
 - d. Seleccione de la paleta de funciones el menú Express, signal Manipulation, Marge Signal.
 - e. Realice las conexiones del diagrama de bloque como la grafica. (VerFigura4.22)

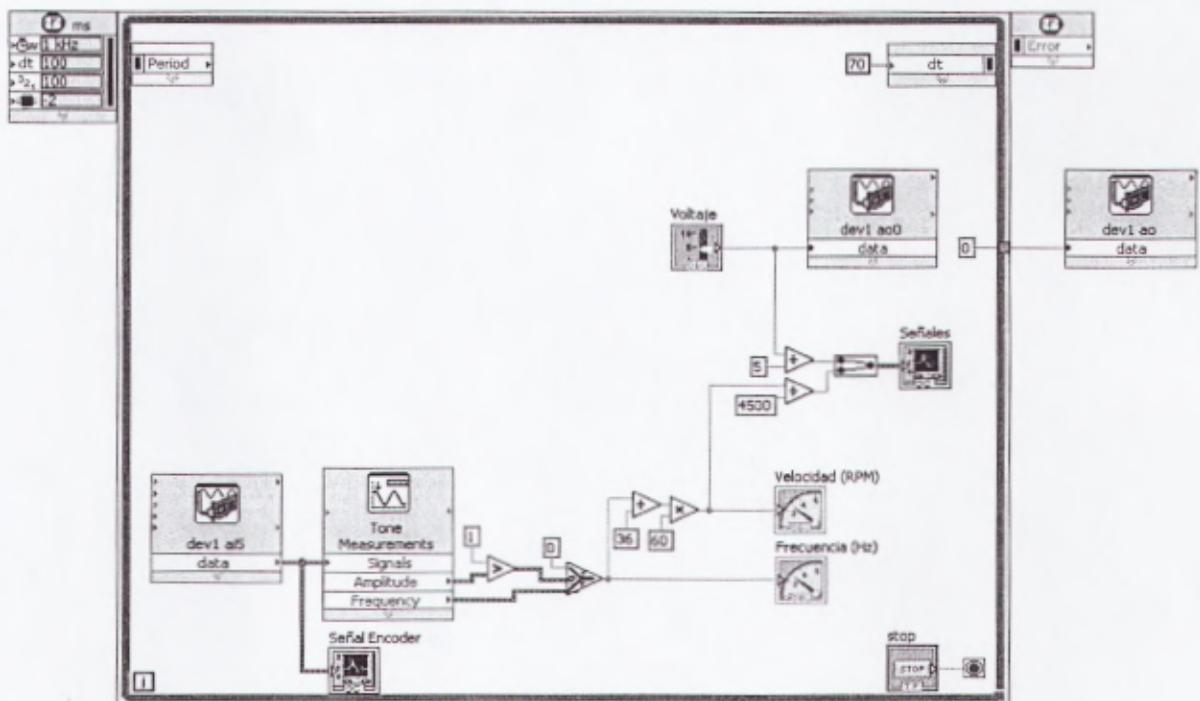


Figura 4.22: Diagrama de Bloque Práctica M2

5. Modificar el valor de Voltaje y observar cómo varía la velocidad. Cuando el voltaje está en su máximo valor (5 V), la velocidad toma su máximo valor (cerca de 4500 RPM).
6. Para observar (Ver Figura 4.23), la curva de arranque y parada del motor puede aplicarse sucesivamente valores de 0V y luego 5V al voltaje, y notar en la Gráfica Señales la forma en que la velocidad responde a esta señal de estímulo tipo escalón, típica de un motor DC.

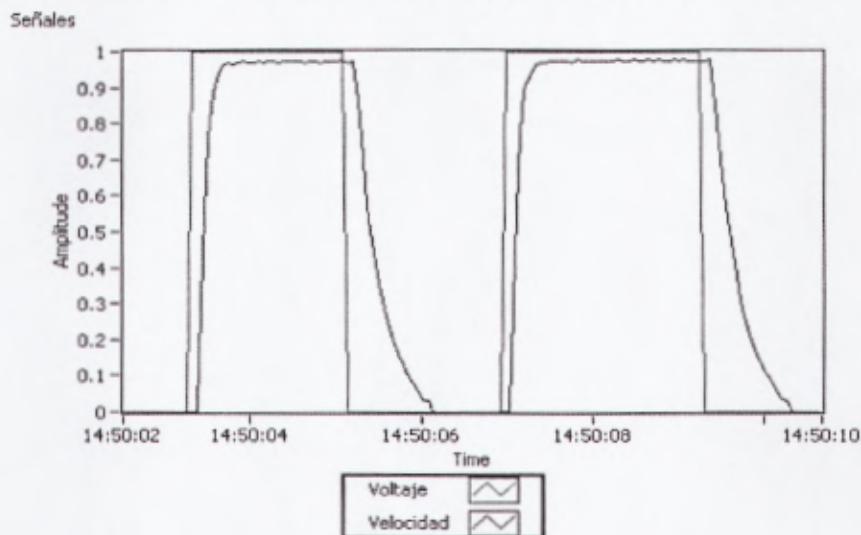


Figura 4.23: Curva de Arranque

7. Abrir el diagrama de bloques para analizarlo.
 - a. La señal adquirida del Encoder en forma analógica es procesada para obtener su frecuencia y su amplitud.
 - b. Si la amplitud es menor que 1 significa que el Motor DC no está girando, por lo tanto el encoder entregará una señal sin pulsos (y que puede estar en estado alto o bajo según donde haya parado el disco), y la frecuencia que se mide será la del ruido de la señal, mas no de la velocidad.

- c. Si la amplitud de la señal es mayor que 1 el motor estará girando y el Encoder entregará una señal tipo tren de pulsos. Entonces la frecuencia medida sí será la cantidad de pulsos por segundo.
 - d. Las operaciones siguientes son para transformar la frecuencia en revoluciones por minuto (RPM).
8. Cerrar el programa sin hacer cambios.

4.3.3 PRÁCTICA M3:SEÑAL DE ENCODER – FORMA DE CONTADOR DIGITAL

En esta práctica se adquirirá la señal del encoder para analizarla en forma de tren de pulsos, y se medirá la cantidad de pulsos por iteración para determinar la velocidad del motor.

1. Realizar las siguientes conexiones entre el EPC y la tarjeta DAQ(*VerFigura4.24*)
 - a. Conectar una salida analógica de la tarjeta DAQ a la entrada de control del Motor DC MOTOR DC IN. En este ejemplo se utiliza la salida analógica número cero (AO0) de la tarjeta instalada como Device 1.
 - b. Conectar la referencia GND de la tarjeta DAQ a la referencia GND del EPC. Todas las referencias GND de la tarjeta DAQ están conectadas internamente (en el modelo USB-6009 y similares); igualmente, todas las referencias GND del EPC están conectadas internamente.

- c. Conectar la salida del encoder del Motor DC del EPC llamada ENCODER MDC a la entrada de contadores de la tarjeta DAQ. En este ejemplo se utiliza el contador número cero, cuyo terminal en la tarjeta DAQ está identificado como PF10 (Programable Function input 0).
- d. Conectar la fuente de poder del EPC.

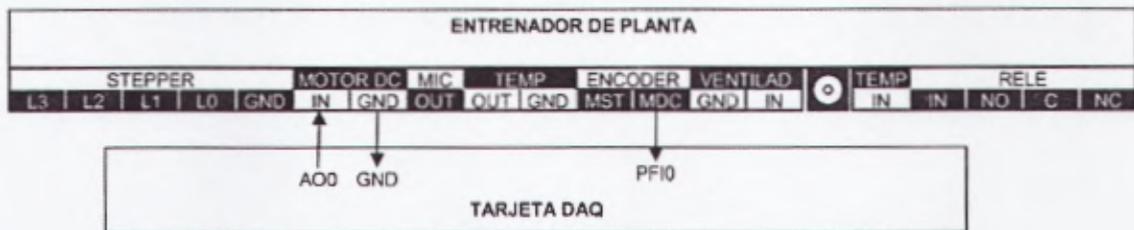


Figura4.24: Conexión Tarjeta DAQ Práctica M3

2. Realice el programa medición contador velocidad.vi. El panel frontal presenta los siguientes controles e indicadores:
 - a. El control numérico tipo Slide Voltaje es el voltaje de salida de la tarjeta DAQ, que puede tomar valores entre 0 y 5 voltios.
 - b. El indicador gráfico tipo Chart Señales muestra un historial de los valores de voltaje y de velocidad que se han generado. La magnitud de estas señales está en Valores Por Unidad (VPU). Para el Voltaje el límite máximo es 5 Voltios. Para la Velocidad el límite máximo es 4500 RPM.
 - c. El indicador tipo Gauge Velocidad indica la velocidad del eje en revoluciones por minuto (RPM). El disco del encoder tiene 36 ranuras.
 - d. El botón de STOP detiene el programa. Es importante detener el programa utilizando este botón y no el botón de abortar de la

barra de herramientas porque así se asegura que al parar el programa se detenga el motor, pues se envía un valor de 0V a laSalida analógica

3. Para esto realice los siguientes Pasos: (VerFigura 4.25)

- a. Se aplica los literales a, b, c y d del Ítem 3 de la práctica M2.
- b. Se dibuja dos controles numéricos tipo Gauge, para esto seleccionamos de la paleta de controles el menú Modern, Numeric, Gauge.

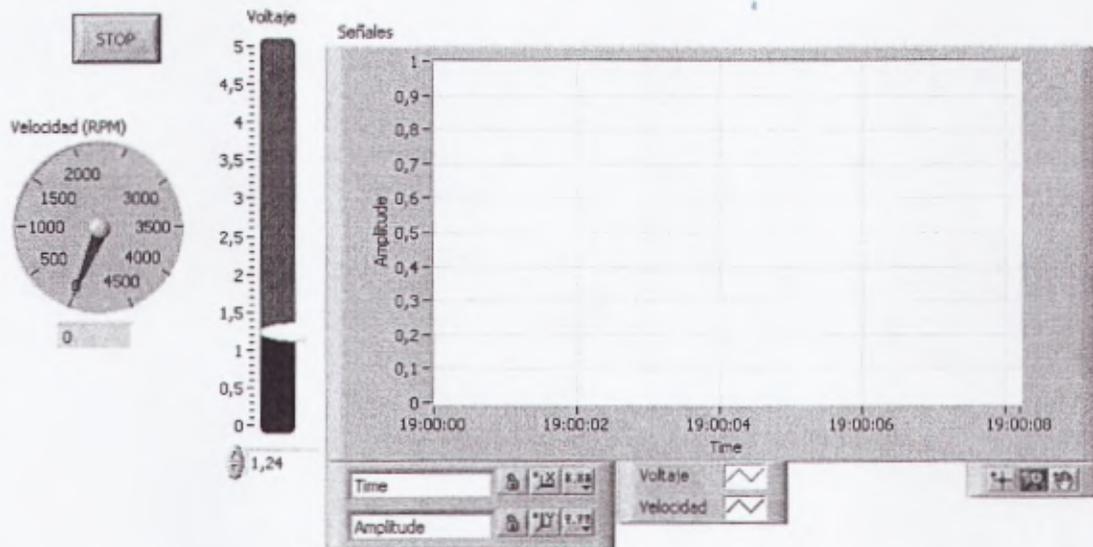


Figura 4.25: Panel Frontal Práctica M3

4. Modificar el valor de Voltaje de alimentación del Motor DC. Observar la forma en cómo varía la Velocidad al variar el Voltaje. Ésta es igual a la práctica anterior, excepto por la cantidad de ruido a diferentes velocidades. Este método es preferible cuando se tiene un encoder de alta cantidad de pulsos por revolución, pues el programa es más liviano porque no necesita adquisición de datos analógica sino solamente leer del registro del contador de la tarjeta DAQ.

5. Se abre el diagrama de Bloques y se realiza lo siguiente:
 - a. Se aplica los literales a y b, del Ítem 3 de la Práctica M2.
 - b. Seleccione de la paleta de funciones el menú Express, signal Manipulation, Marge Signal.
 - c. Seleccione Operadores numérico, en la paleta de funciones elija el menú Programing, Numeric, multiple-divide-subtract.
 - d. Realice las conexiones del diagrama de bloque como la grafica.
(VerFigura4.26)

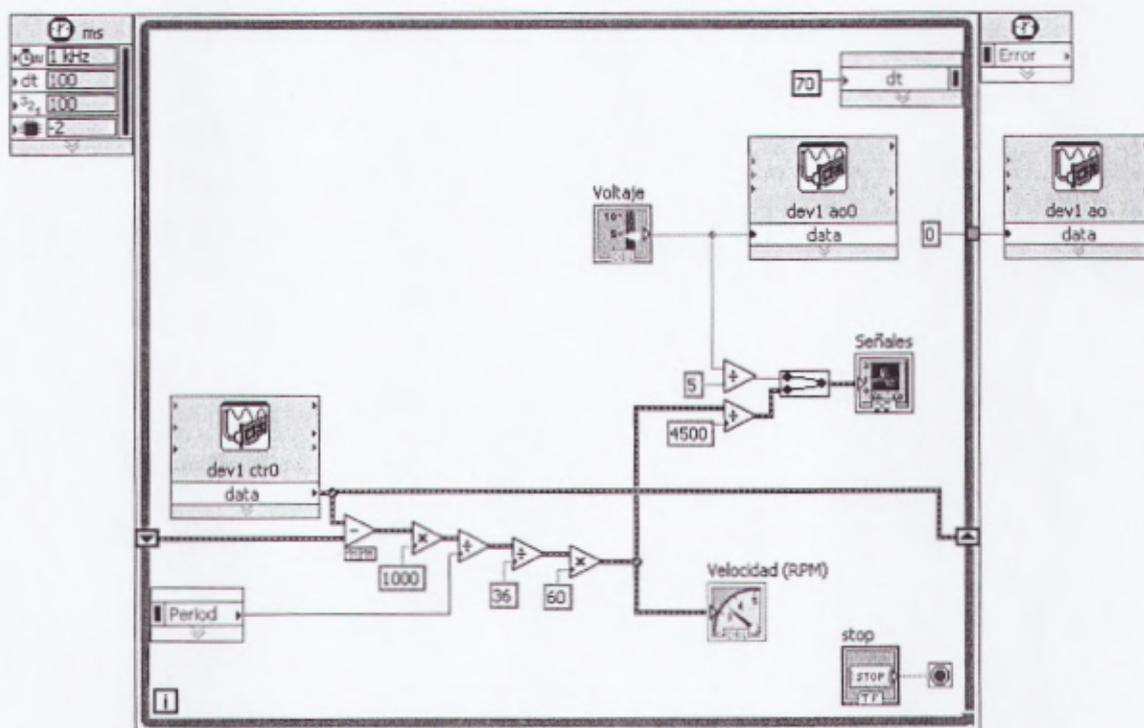


Figura 4.26:Diagrama de Bloque Práctica M3

6. Se analiza el diagrama de bloque:
 - a. En este caso se ha adquirido la señal del Encoder por medio del contador de la tarjeta DAQ.
 - b. Se resta el valor actual del contador con el valor en la iteración anterior para obtener el número de pulsos en una iteración.
 - c. Los cálculos siguientes permiten obtener la velocidad en revoluciones por segundo

7. Cerrar el programa sin hacer cambios.

4.3.4 PRÁCTICA M4: CARACTERIZACIÓN DE MOTOR DC

En esta práctica se utilizarán como bases las prácticas anteriores para obtener una caracterización del Motor en cuanto a linealidad entre la variable de control (Voltaje DC) y la variable controlada (Velocidad). Si el sistema es lineal se puede aplicar un controlador PID. También se podrá observar los rangos en los cuales el sistema es lineal o tiene discontinuidades. Este análisis se hará de manera visual solamente, pero se puede ampliar el programa para mostrar una línea superpuesta al comportamiento del motor para obtener resultados matemáticos exactos.

1. Mantener las conexiones de la Práctica M2: Señal de Encoder – Forma Analógica.

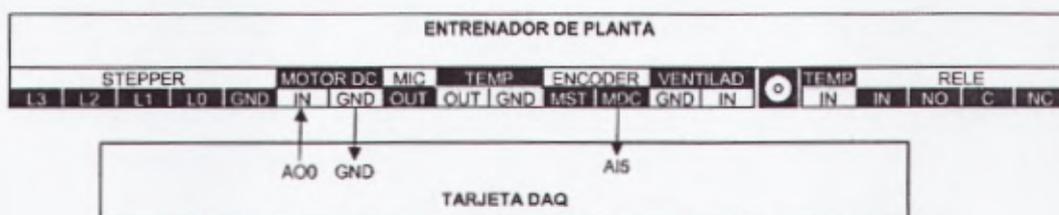


Figura4.27: Conexión Tarjeta DAQ Práctica M4

Realice un programa con nombre caracterización motor.vi. Este programa incrementa automáticamente el Voltaje de control del Motor DC de 0 a 5 voltios en pasos de 0.05 V cada 100ms. Al llegar a los 5V el programa se detiene automáticamente. El panel frontal tiene los siguientes objetos:

- a. El indicador numérico tipo Slide Voltaje muestra el voltaje enviado al Motor DC.
- b. El indicador numérico tipo Slide Velocidad muestra la velocidad del Motor DC en revoluciones por minuto (RPM).
- c. El gráfico tipo XY Voltaje Vs. Velocidad muestra el par de datos ordenados: en el eje X el Voltaje de control del Motor DC, y en el eje Y la Velocidad.
- e. El botón STOP permite detener el programa antes que llegue a los 5V

2. Para esto realice lo siguiente en el Panel Frontal:(VerFigura4.28)

- a. En la pantalla del Panel Frontal se dibuja un Indicador Grafico, para esto seleccionamos de la paleta de controles el menú Modern, Graph, WaveForm Chart.
- b. Se dibuja dos indicadores numérico de para velocidad y voltaje, para esto seleccionamos de la paleta de controles el menú Modern, Numeric, Vertical Pointer Slide.
- c. Se dibuja un botón de Stop, esto lo realizamos al seleccionar de la paleta de controles el menú Modern, Boolean, Stop Button.

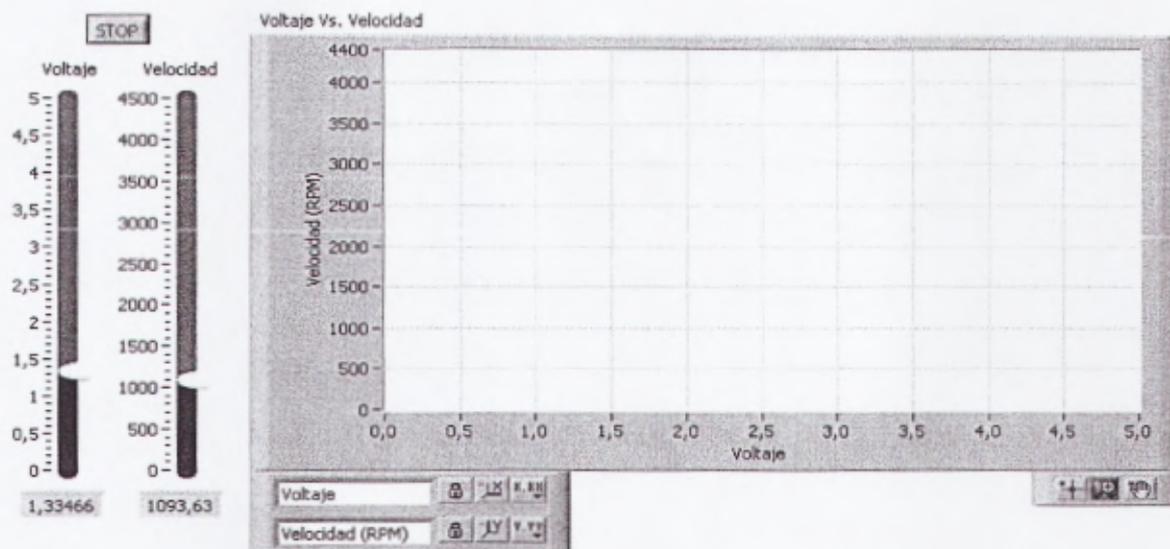


Figura4.28: Panel Frontal Práctica M4

3. Se Abre el diagrama de Bloques y se realiza lo siguiente:
 - a. Se aplica los literales a y b, del Ítem 3 de la Práctica M1.
 - b. Seleccione en paleta de funciones el menú Programing, WaveForm, WeveForm Measure, Amplitud and Level Measurements.
 - c. Seleccione de la paleta de Funciones el menú Programing, Comparison, Select.
 - d. Seleccione de la paleta de Funciones el menú Programing, Array, Build Array.
 - e. Realce las conexiones de la imagen.(VerFigura4.29)

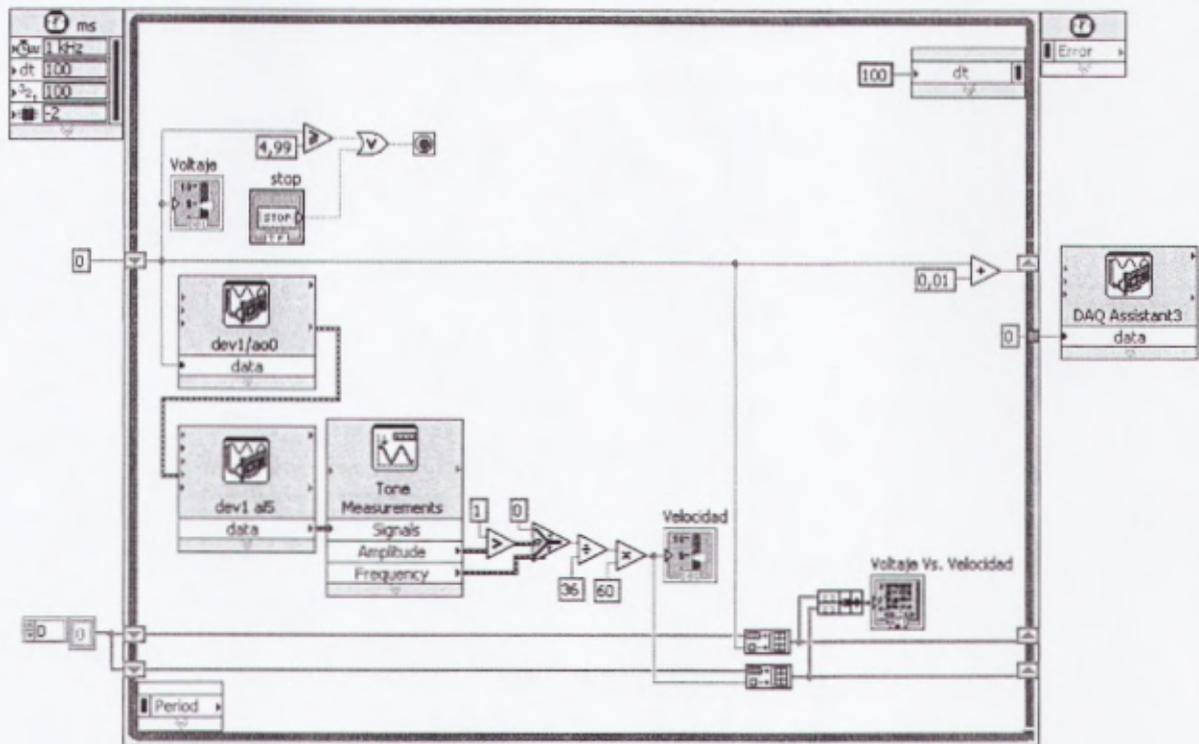


Figura4.29: Diagrama de Bloques Práctica M4

3. Correr el programa.

- a. Observar cómo automáticamente el voltaje sube progresivamente. (VerFigura4.30)
- b. De cero a aproximadamente 0.5 V, el voltaje no es suficiente para vencer el rozamiento e inercia del rotor, de modo que éste no rota y la velocidad es cero.
- c. Por sobre los 0.5 V aproximadamente, el rotor empieza a girar repentinamente presentando una discontinuidad. Es posible que existan pequeños pulsos de velocidad debido a que el motor presenta un ligero movimiento antes de arrancar definitivamente.

```

1  library ieee;
2  use ieee.std_logic_1164.all ;
3  entity ffsr is port (
4      S,R,clk: in std_logic;
5      Q, Qn: inout std_logic);
6  end ffsr;
7  architecture a_ffsr of ffsr is
8  begin
9  process (clk, S, R)
10 begin
11     if (clk'event and clk = '1') then
12         if (S = '0' and R = '1') then
13             Q <= '0';
14             Qn <= '1' ;
15         elsif (S = '1' and R = '0') then
16             Q <= '1';
17             Qn <= '0';
18         elsif (S = '0' and R = '0') then
19             Q <= Q;
20             Qn <= Qn;
21         else
22             Q <= Q;
23             Qn <= '-';
24         end if;
25     end if;
26 end process;
27 end a_ffsr;

```

Figura 4. 2: Listado de programación VHDL de la práctica 1.
Fuente: Los autores

En la figura 4.3 se muestra la ejecución y reporte de compilación del programa mostrado en la figura 4.2 de la práctica 1.

The screenshot shows the 'Compilation Report - Flow Summary' window in Quartus II. The 'Flow Summary' table is as follows:

Flow Status	Successful - Mon Mar 05 17:36:00 2012
Quartus II Version	7.2 Build 151 09/26/2007 5: Web Edition
Revision Name	ffsr
Top-level Entity Name	ffsr
Family	Cyclone II
Device	EP2K10K10-3
Timing Model	Fast
Met timing requirements	Yes
Total logic elements	2 / 18,752 (<1%)
Total combinational functions	2 / 18,752 (<1%)
Dedicated logic registers	2 / 18,752 (<1%)
Total registers	2
Total pins	5 / 315 (2%)
Total virtual pins	0
Total memory bits	0 / 239,616 (0%)
Embedded Multiplier 9bit elements	0 / 52 (0%)
Total PLLs	0 / 4 (0%)

The 'Messages' window at the bottom shows the following messages:

```

Info: sh for register "Qn-reg0" (data pin = "R", clock pin = "clk") is -2.853 ns
Info: Quartus II Classic Timing Analyzer was successful. 0 errors, 1 warning
Info: Quartus II Full Compilation was successful. 0 errors, 13 warnings

```

Figura 4. 3: Resultado de la compilación de programación VHDL de la práctica 1.
Fuente: Los autores

Una vez realizada la compilación sin ningún tipo de error de sintaxis en la programación, procederemos a ejecutar el diseño RTL realizado por el software Quartus II, para eso nos dirigimos *Tools* <herramientas> + *Netlist Viewers* y se

escoge *RTL Viewer*, en la figura 4.4 se muestra el resultado obtenido del diseño una vez que se escogió *RTL*.

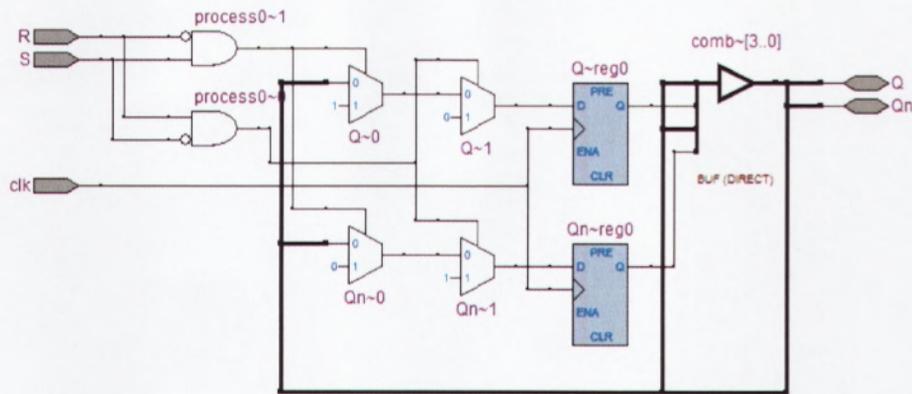


Figura 4. 4: Diseño RTL de la práctica 1.
Fuente: Los autores

4.2. PRÁCTICA 2. Diseño de un registro de 4 bits.

Realizar la programación en VHDL de un registro de 4 bits, como el mostrado en la figura 4.5, en el diseño deben utilizarse instrucciones *if - then - else* y procesos.

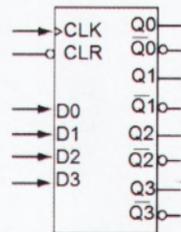


Figura 4. 5: Registro de 4 bits de la práctica 2.
Fuente: Los autores

Como puede observarse en la tabla 4.2, donde se describe el comportamiento del registro de 4 bits, si $CLR = 0$, las salidas Q son '0'; pero si $CLR = 1$, toman el valor de las entradas $D0$, $D1$, $D2$ y $D3$.

CLR	D	Q	QN
0		0	1
1	D_n	D_n	D_n

Tabla 4.2: Tabla de estados del registro de 4 bits.

Fuente: Los autores

De acuerdo a lo establecido en la tabla 4.2 y de las condiciones dadas, procedemos a realizar el código del programa para un registro de 4 bits en VHDL, el mismo que se muestra en la figura 4.6.

```

1  library ieee;
2  use ieee.std_logic_1164.all;
3  entity reg4 is port (
4      D: in std_logic_vector(3 downto 0);
5      CLK, CLR: in std_logic;
6      Q, Qn: inout std_logic_vector(3 downto 0));
7  end reg4;
8  architecture a_reg4 of reg4 is
9  begin
10     process (CLK, CLR) begin
11         if (CLK'event and CLK = '1') then
12             if (CLR = '1') then
13                 Q <= D;
14                 Qn <= not Q;
15             else
16                 Q <= "0000";
17                 Qn <= "1111";
18             end if ;
19         end if ;
20     end process;
21 end a_reg4;

```

Figura 4. 6: Programa VHDL del registro de 4 bits de la práctica 2.

Fuente: Los autores

Una vez creado el programa en VHDL se realiza la compilación del mismo, en la figura 4.7 se muestra el reporte sin ningún error de sintaxis para la práctica 2.

Flow Status	Successful - Mon Mar 05 17:50:56 2012
Quartus II Version	7.2 Build 151 09/26/2007 SJ Web Edition
Revision Name	reg4
Top-level Entity Name	reg4
Family	Cyclone II
Device	EP2C20F484C7
Timing Models	Final
Met timing requirements	Yes
Total logic elements	8 / 18,752 (< 1 %)
Total combinational functions	8 / 18,752 (< 1 %)
Dedicated logic registers	8 / 18,752 (< 1 %)
Total registers	8
Total pins	14 / 315 (4 %)
Total virtual pins	0
Total memory bits	0 / 239,616 (0 %)
Embedded Multiplier 9-bit elements	0 / 52 (0 %)
Total PLLs	0 / 4 (0 %)

Figura 4. 7: Resultado de la compilación de programación VHDL de la práctica 2.

Fuente: Los autores

En la figura 4.8 se muestra el diseño RTL de acuerdo a la programación del registro de 4 bits en VHDL, el procedimiento ya se lo explicó en la práctica anterior.

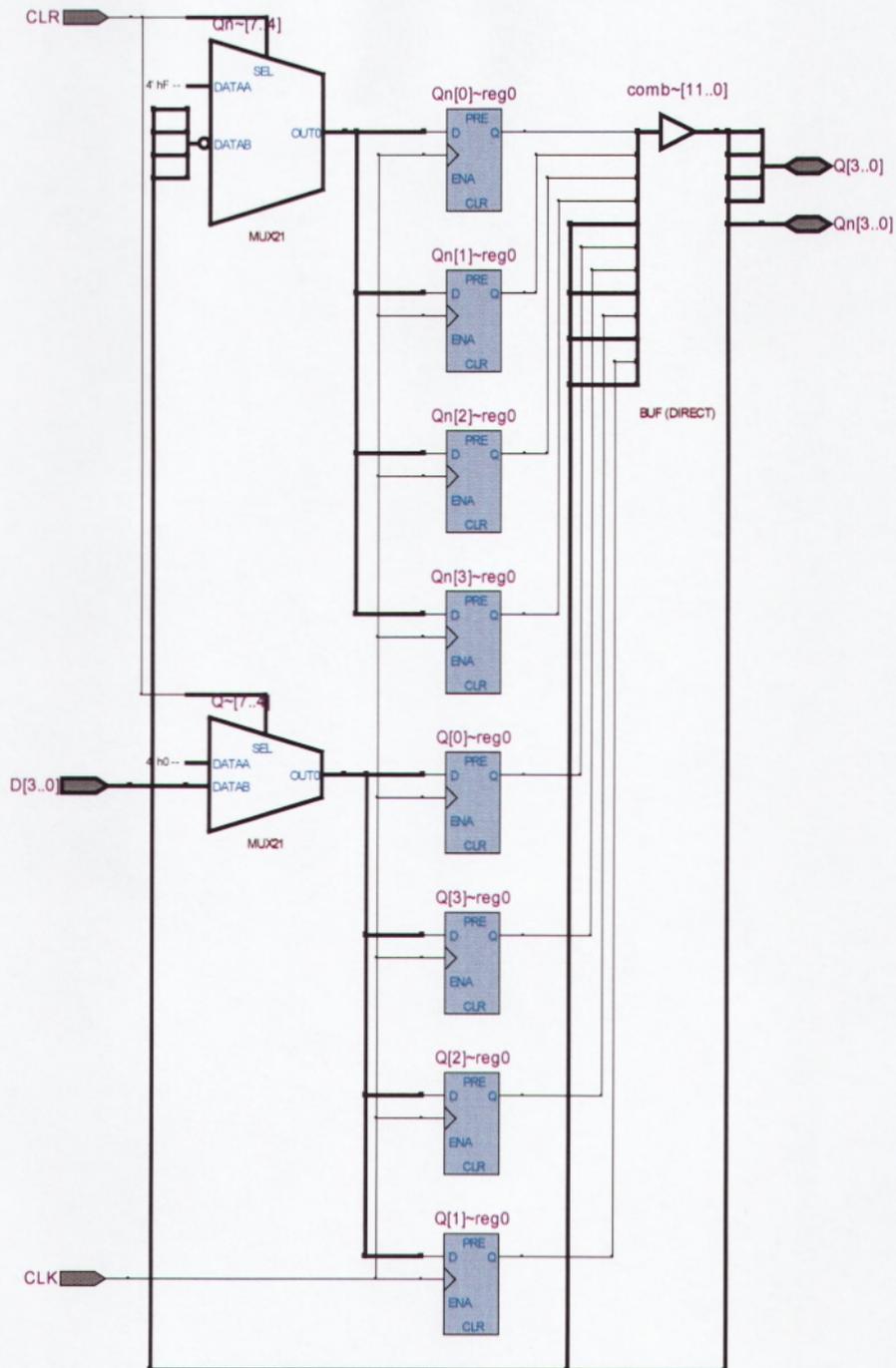


Figura 4. 8: Diseño RTL de la práctica 2.
Fuente: Los autores

4.3. PRÁCTICA 3. Diseño a partir de diagrama de estados.

Se requiere programar el diagrama de estados de la figura 4.9.

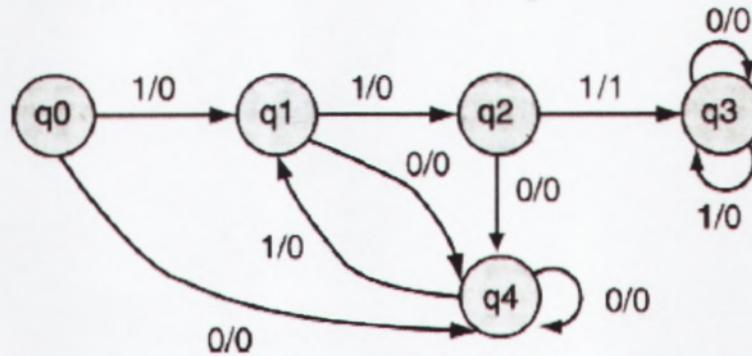


Figura 4. 9: Diagrama de estados para la práctica 3.

Fuente: Los autores

Por ejemplo, en la figura 4.9 notamos que el circuito pasa al estado q1 con el primer '1' de entrada y al estado q2 con el segundo. Las salidas asociadas con estas dos entradas es '0'. La tercera entrada consecutiva de '1' genera un '1' en la salida y hace que el circuito pase al estado q3. Una vez que se encuentra en q3, el circuito permanecerá en el mismo estado, emitiendo salidas 0. En lo que respecta a la programación en VHDL, para la práctica 3 seguiremos con la misma metodología que las prácticas anteriores, la cual consiste en usar estructuras *case - when* y tipo de datos enumerado, que en este caso contiene los cinco estados (q0, q1, q2, q3 y q4) que componen el diagrama de estados mostrado en la figura 4.9.

En base al análisis realizado del diagrama de estados mostrado en la figura 4.9, procedemos a realizar la respectiva programación en VHDL. En la figura 4.10 se muestra el listado correspondiente al programa de la práctica 3.

```
1  library ieee;
2  use ieee.std_logic_1164.all;
3  entity diag is port (
4      clk,x: in std_logic;
5      z: out std_logic);
6  end diag;
7  architecture arq_diag of diag is
8      type estados is (q0,q1,q2,q3,q4);
9      signal edo_pres, edo_fut: estados;
10 begin
11     procesos: process (edo_pres,x) begin
12         case edo_pres is
13             when q0=> z <= '0';
14             if x = '0' then
15                 edo_fut <= q4;
16             else
17                 edo_fut <= q1;
18             end if;
19             when q1 => z <= '0';
20             if x = '0' then
21                 edo_fut <= q4;
22             else
23                 edo_fut <= q2;
24             end if;
25             when q2 =>
26             if x = '0' then
27                 edo_fut <= q4;
28                 z <= '0';
29             else
30                 edo_fut <= q3;
31                 z <= '1';
32             end if;
33             when q3 => z <= '0';
34             if x = '0' then
35                 edo_fut <= q3;
36             else
37                 edo_fut <= q3;
38             end if;
39             when q4 => z <= '0';
40             if x = '0' then
41                 edo_fut <= q4;
42             else
43                 edo_fut <= q1;
44             end if;
45         end case;
46     end process procesos;
47     proceso2: process (clk) begin
48     if (clk'event and clk='1')then
49         edo_pres <= edo_fut;
50     end if;
51     end process proceso2;
52 end arq_diag;
```

Figura 4. 10: Programa VHDL del registro de 4 bits de la práctica 3.

Fuente: Los autores

En la figura 4.11 se muestra el resultado obtenido de la máquina de estado generada a partir de la programación en VHDL del registro de 4 bits, donde se pudo comprobar la igualdad del diagrama de estados de la figura 4.9.

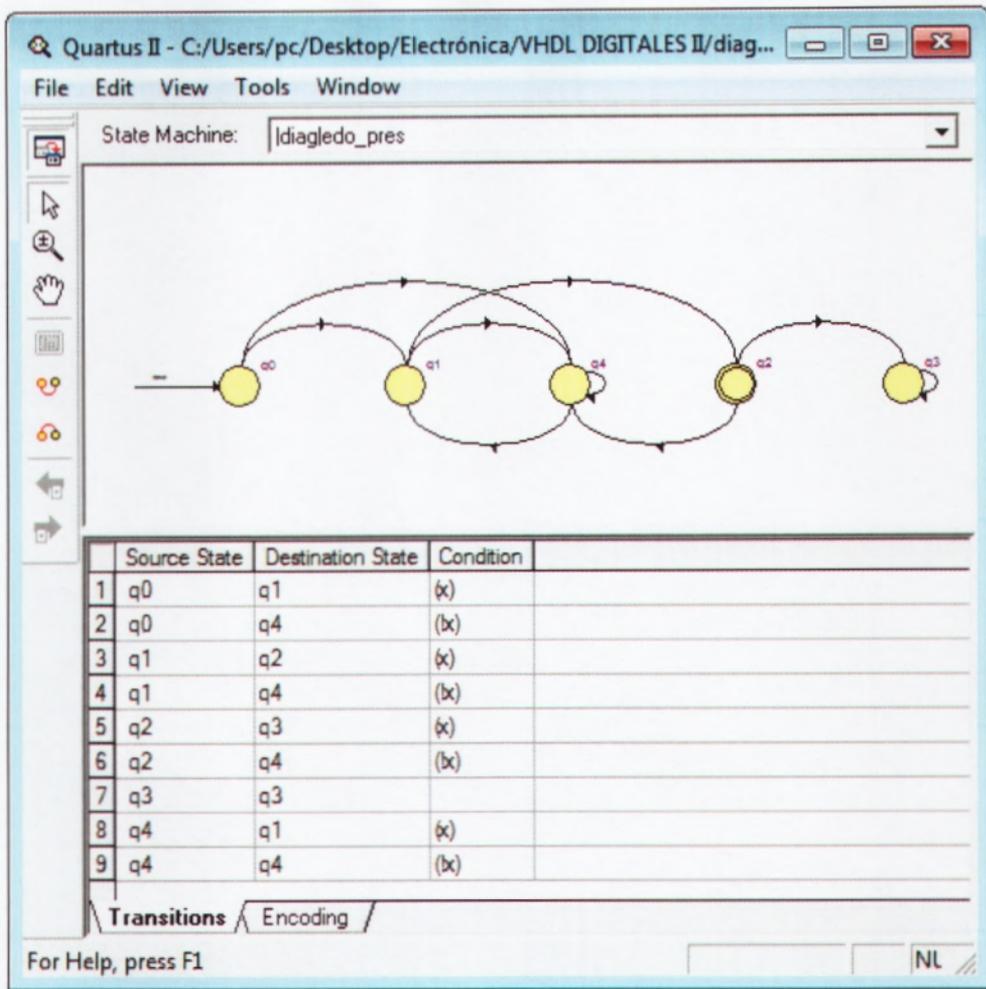


Figura 4. 11: Máquinas de estados del registro de 4 bits de la práctica 3.

Fuente: Los autores

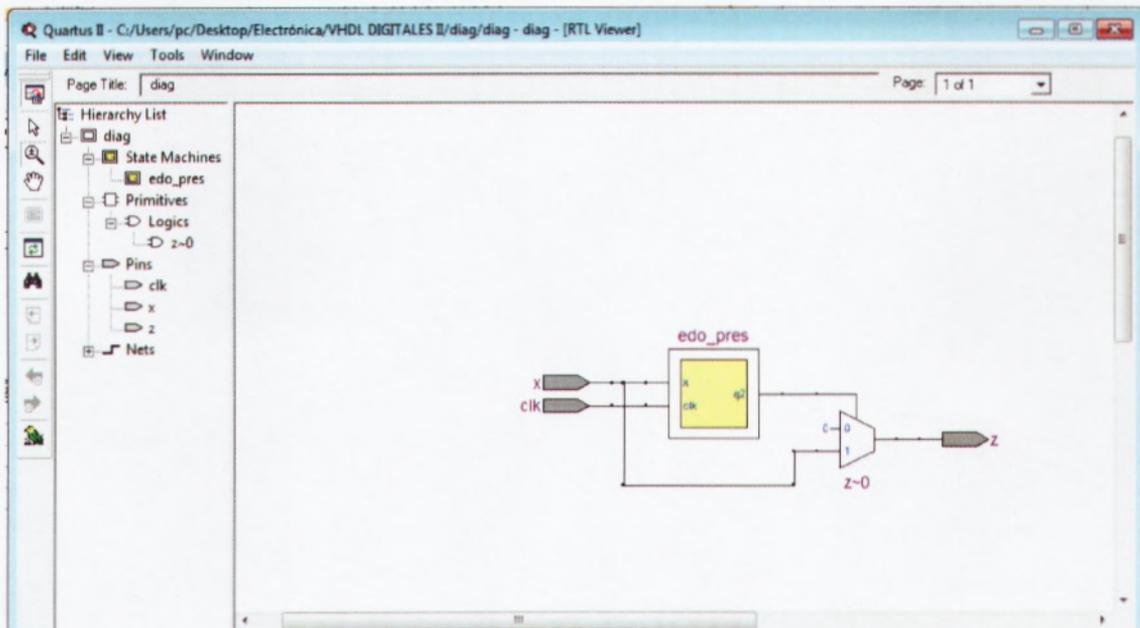


Figura 4. 12: Diseño RTL del registro de 4 bits de la práctica 3.

Fuente: Los autores

4.4. PRÁCTICA 4. Integración de entidades en VHDL.

Hasta el momento se ha empleado la programación en VHDL para diseñar entidades individuales (bloques lógicos mínimos), con el único objeto de familiarizarlo con los diversos estilos de diseño y programación o ambos, así como con el uso y aplicación de las palabras reservadas en VHDL. Sin embargo, es obvia que esta herramienta de diseño no fue creada para este fin.

Existen diversas razones para su empleo; pero tal vez su verdadera fortaleza radica en que permite integrar "sistemas digitales" que contienen una gran cantidad de subsistemas electrónicos con el fin de minimizar el tamaño de la aplicación. En primera instancia, en un solo circuito integrado y si el problema es muy complejo, a través de una serie sucesiva de circuitos programables, sea que se llamen CPLD (dispositivo lógico programable complejo) o FPGA (arreglos de compuertas programables en campo).

Esquema básico de integración de entidades

La integración de entidades puede realizarse mediante el diseño individual de cada bloque lógico a través de varios procesos internos que posteriormente pueden unirse mediante un programa común. Otra posibilidad es observar y analizar de manera global todo el sistema evaluando su comportamiento sólo a través de sus entradas y salidas. En ambos casos el resultado es satisfactorio; más bien, nuestra tarea consiste en analizar las ventajas y desventajas que existen en ambas alternativas de solución. En el primer caso, el inconveniente principal es el número excesivo de terminales utilizadas en el dispositivo, debido a que al diseñar entidades individuales, se tendrían que declarar las terminales de entrada-salida de cada entidad (ver figura 4.13a).

En el segundo caso, cuando observamos el sistema como un todo (ver figura 4.13b) el número de terminales de entrada y salida disminuye, por lo que nuestro trabajo es desarrollar no sólo un algoritmo interno capaz de interpretar el funcionamiento de cada bloque, sino también cómo conectar cada uno de ellos.

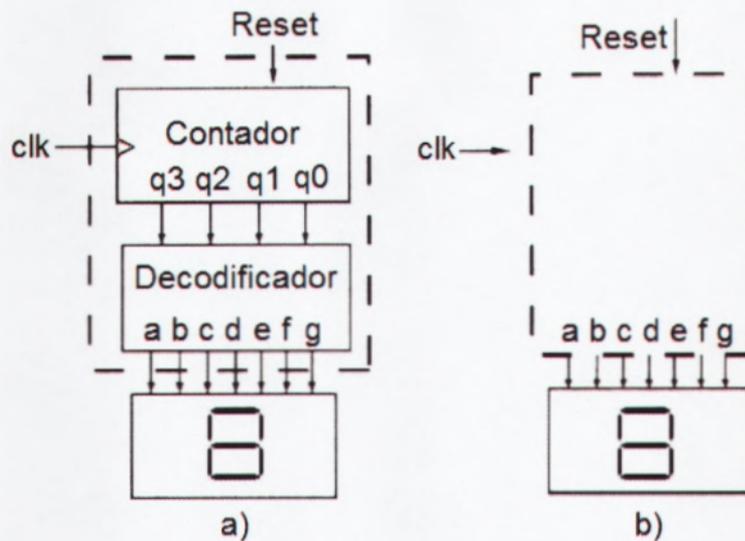


Figura 4. 13: a) Diseño mediante entidades individuales, b) Diseño mediante la relación de entradas/salidas.

Fuente: Los autores

Programación de entidades individuales

Procederemos a la programación del contador y decodificador de la figura 4.13a), mediante la programación de entidades individuales, como el mostrado en la figura 4.14.

```

Quartus II - C:/Users/pc/Desktop/Electrónica/VHDL DIGITALES II/display/display - display - [display.vhd]
File Edit View Project Processing Tools Window
1  library ieee;
2  use ieee.std_logic_1164.all;
3  use ieee.std_logic_unsigned.all;
4  entity display is port (
5      clk,reset: in std_logic;
6      d: inout std_logic_vector(6 downto 0); -- salidas del decodificador
7      q: inout std_logic_vector(3 downto 0); -- salidas del contador
8  end display;
9  architecture arqdisplay of display is
10 begin
11     process (clk,reset)
12     begin
13         if (clk'event and clk = '1') then
14             q <= q+1;
15             if (reset = '1' or q = "1001") then
16                 q <= "0000";
17             end if ;
18         end if;
19     end process;
20     process (q) begin
21     case q is
22         when "0000" => d <= "1111110";
23         when "0001" => d <= "0110000";
24         when "0010" => d <= "1101101";
25         when "0011" => d <= "1111001";
26         when "0100" => d <= "0110011";
27         when "0101" => d <= "1011011";
28         when "0110" => d <= "1011111";
29         when "0111" => d <= "1110000";
30         when "1000" => d <= "1111111";
31         when "1001" => d <= "1110011";
32         when others => d <= "0000000";
33     end case;
34     end process;
35 end arqdisplay;

```

Figura 4. 14: Programación en VHDL mediante entidades individuales.

Fuente: Los autores

Como podemos observar, en la declaración de la entidad (entity) se han asignado las terminales de entrada y salida correspondientes a cada bloque lógico de manera individual; así por ejemplo, el contador tiene las salidas q [q3, q2, q1, q0] y el decodificador las salidas d (da, db, de, dd, de, df, dg). Note que la programación requiere dos procesos: el primero (líneas 11 a 19) describe el funcionamiento del contador; el segundo (líneas 20 a 35), el funcionamiento del decodificador.

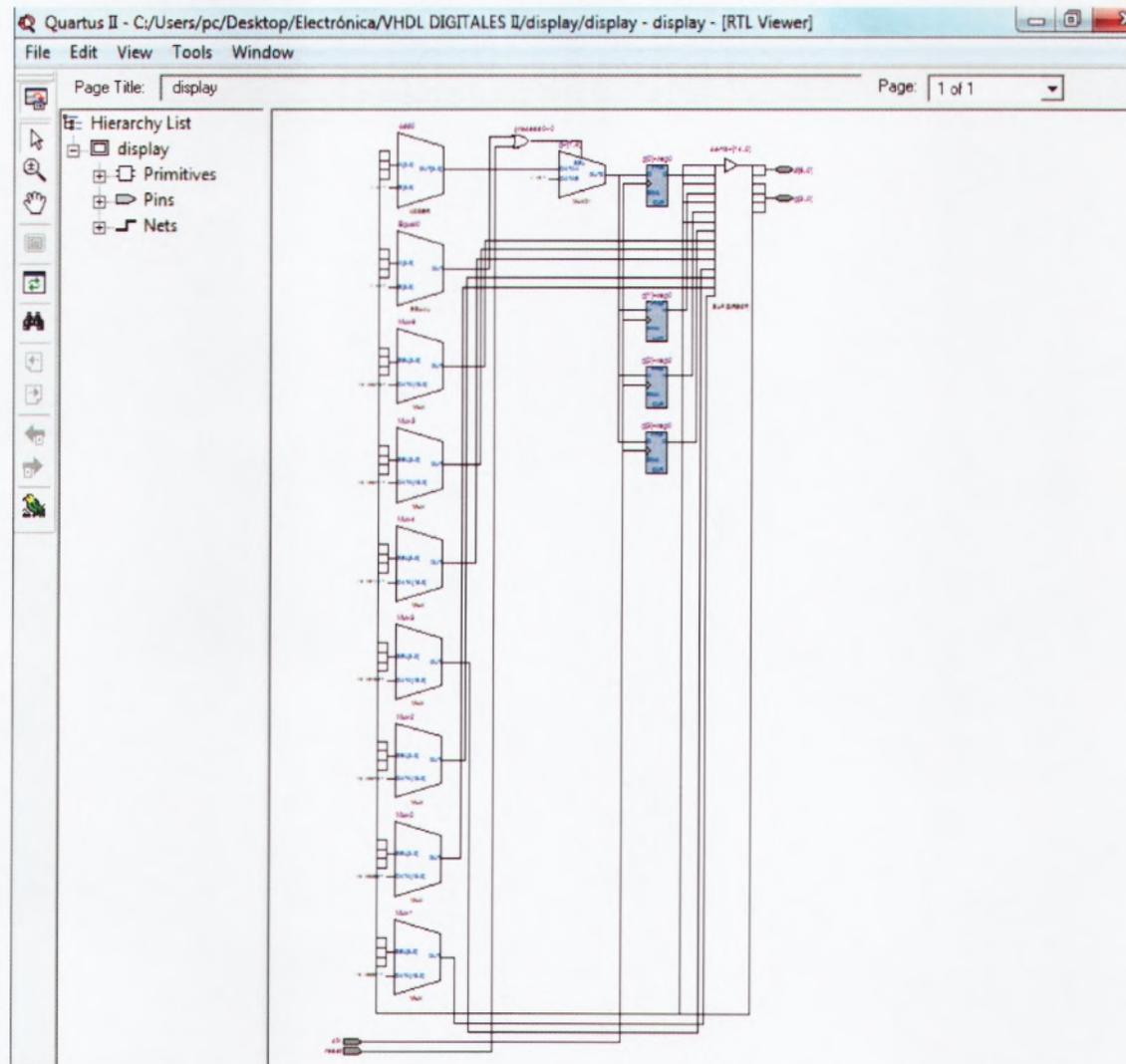


Figura 4. 15: Diseño RTL mediante entidades individuales.

Fuente: Los autores

Programación mediante relación entradas/salidas

Por otro lado, en la figura 4.16 se muestra el programa en VHDL del sistema de la figura 4.13b. Observemos cómo nuevamente el programa supone dos procesos: el primero (líneas 11 a 19) programa el contador y el segundo (líneas 20 a 34) describe el decodificador.

```
Quartus II - C:/Users/pc/Desktop/Electrónica/VHDL DIGITALES II/display1/display1 - display1 - [display1.vhd]
File Edit View Project Processing Tools Window
1 library ieee;
2 use ieee.std_logic_1164.all ;
3 use ieee.std_logic_unsigned.all ;
4 entity display1 is port (
5     clk,reset: in std_logic;
6     d: out std_logic_vector (6 downto 0));
7 end display1;
8 architecture a_displ of display1 is
9     signal q: std_logic_vector (3 downto 0);
10    begin
11        process (clk,reset) begin
12            if (clk'event and clk = '1') then
13                q <= q + 1;
14                if (reset = '1' or q = "1001") then
15                    q <= "0000";
16                end if;
17            end if;
18        end process;
19        process (q) begin
20            case q is
21                when "0000" => d <= "1111110";
22                when "0001" => d <= "0110000";
23                when "0010" => d <= "1101101";
24                when "0011" => d <= "1111001";
25                when "0100" => d <= "0110011";
26                when "0101" => d <= "1011011";
27                when "0110" => d <= "1011111";
28                when "0111" => d <= "1110000";
29                when "1000" => d <= "1111111";
30                when "1001" => d <= "1110011";
31                when others => d <= "0000000";
32            end case ;
33        end process;
34    end a_displ;
```

Figura 4. 16: Programación en VHDL mediante relación de entradas / salidas.
Fuente: Los autores

Sin embargo, a diferencia del programa anterior en éste sólo se asignan las terminales de salida correspondientes al decodificador (línea 6), el enlace interno entre el contador y este último se realiza mediante señales (signal) línea 9. De nuevo, es importante mencionar que los dos programas son correctos y que la disponibilidad de terminales y capacidad de integración del dispositivo empleado determinan la utilización de uno u otro.

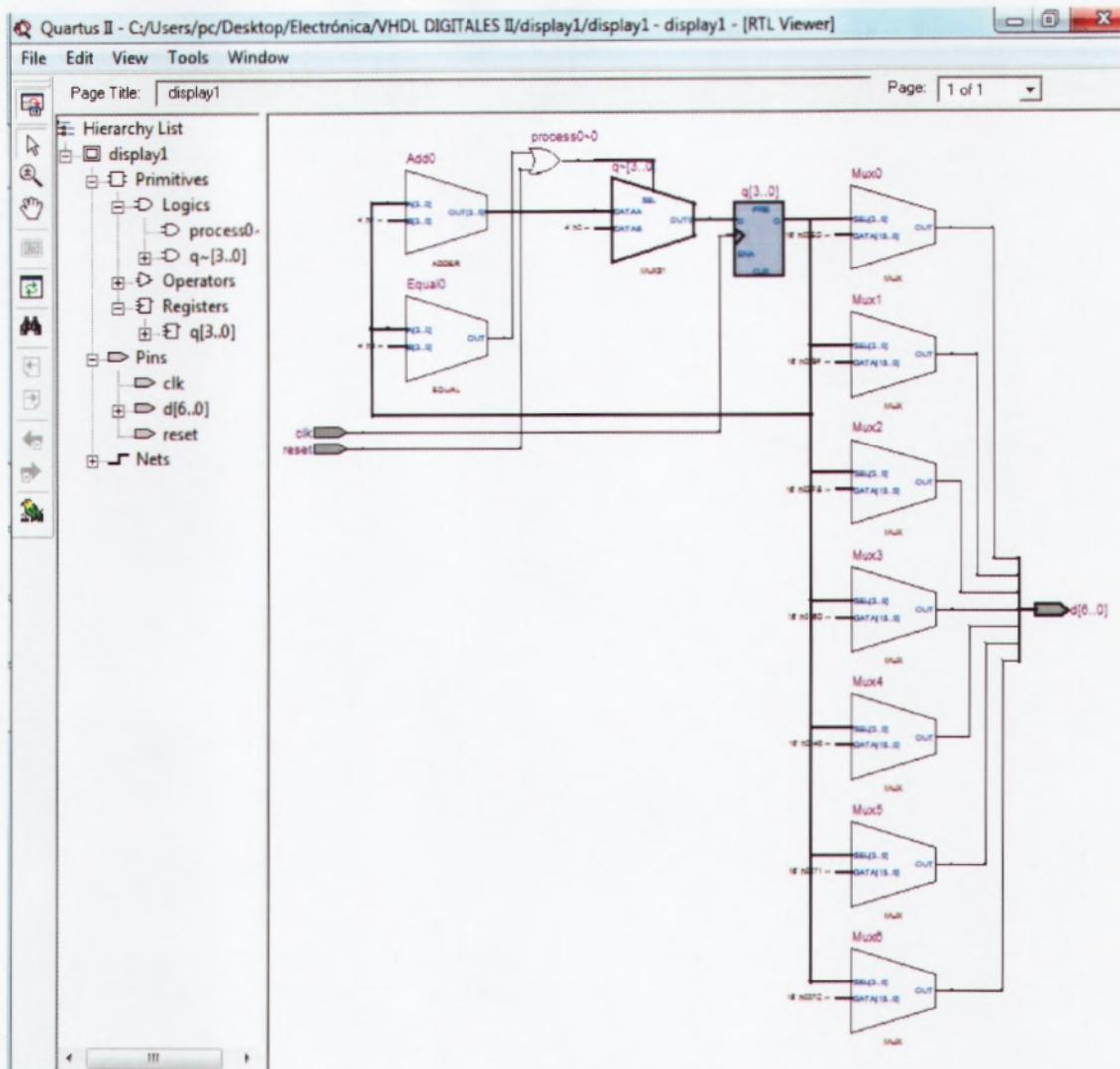


Figura 4. 17: Diseño RTL mediante relación de entradas / salidas.
Fuente: Los autores

4.5. PRÁCTICA 5. Integración de bloques lógicos en VHDL.

Para la siguiente práctica realizaremos la programación en VHDL para integrar diferentes bloques lógicos, como el mostrado por la figura 4.18. El circuito de la figura 4.18 consiste en automatizar procesos de empaquetamiento de muñecas. Considerar que las muñecas se deben empaquetar en forma individual o con un máximo de nueve unidades por paquete. Al inicio el operador selecciona mediante el teclado decimal la cantidad de piezas a empaquetar. Como sabemos, este número decimal se convierte a BCD mediante el circuito codificador y luego pasa mediante el registro hacia el decodificador de 7 segmentos para mostrar en el display el valor del número seleccionado.

Observe que este valor binario es a su vez la entrada A (A3, A2, A1 y A0) del circuito comparador de cuatro bits.

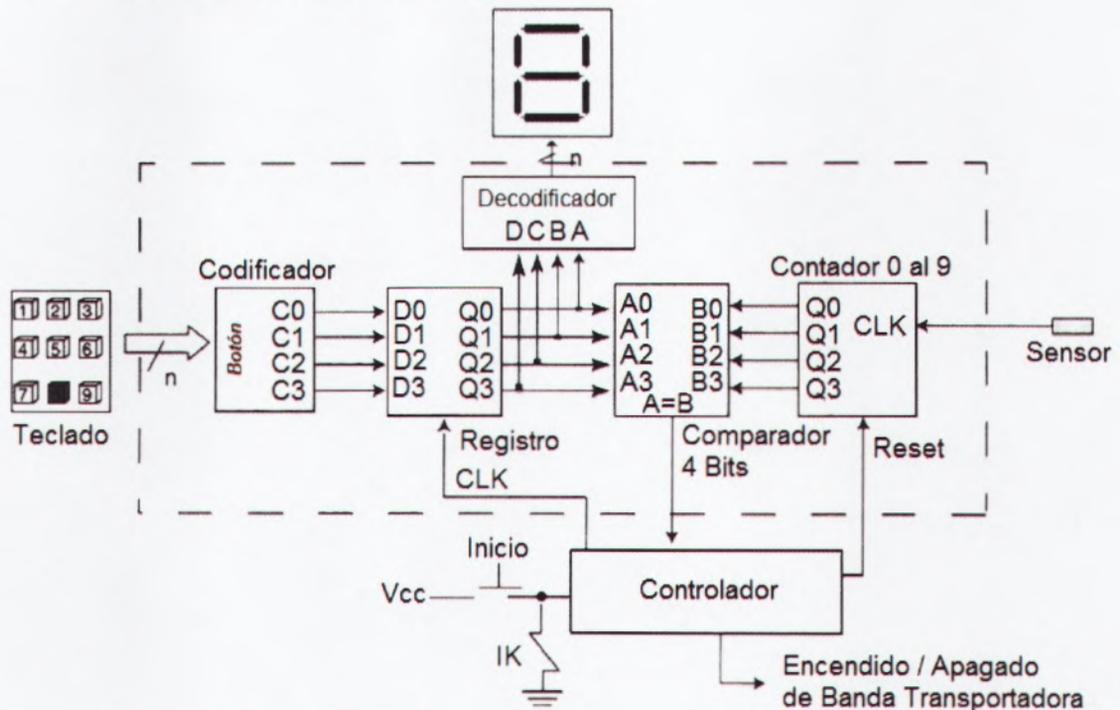


Figura 4. 18: Circuito para automatizar proceso de empaquetamiento de muñecas.
Fuente: Los autores

Una vez explicado el funcionamiento del circuito descrito en la figura 4.18, y de seleccionar la cantidad de muñecas por empaquetar, el operador presiona el botón de inicio, el cual desencadena una serie de acciones controladas por el bloque denominado controlador. El mismo comienza con una señal de salida llamada Reset, que coloca al contador binario en el estado de cero; enseguida envía la señal de arranque (Encendido) al motor que controla el avance de la banda transportadora. Ahora bien, cada vez que una de las muñecas colocadas sobre la banda transportadora pasa por el "sensor" (ver figura 4.19), donde se origina un impulso eléctrico (pulso) que hace que el contador aumente en uno su conteo.

Este procedimiento continúa incrementando al contador en una unidad, hasta el momento en que la cifra actual del contador (número B) es igual al "número A" dentro del comparador ($A = B$), con lo cual este último envía una señal al bloque controlador que detiene la banda transportadora (Apagado) y marca el fin del proceso.

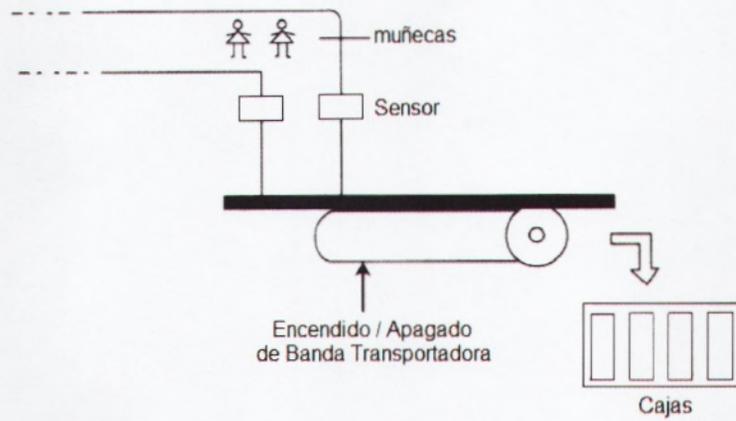


Figura 4. 19: Banda transportadora para empaquetamiento de muñecas.
Fuente: Los autores

En la figura 4.20 se observa el intercambio de señales que se realiza entre el controlador y cada uno de los bloques lógicos del sistema.

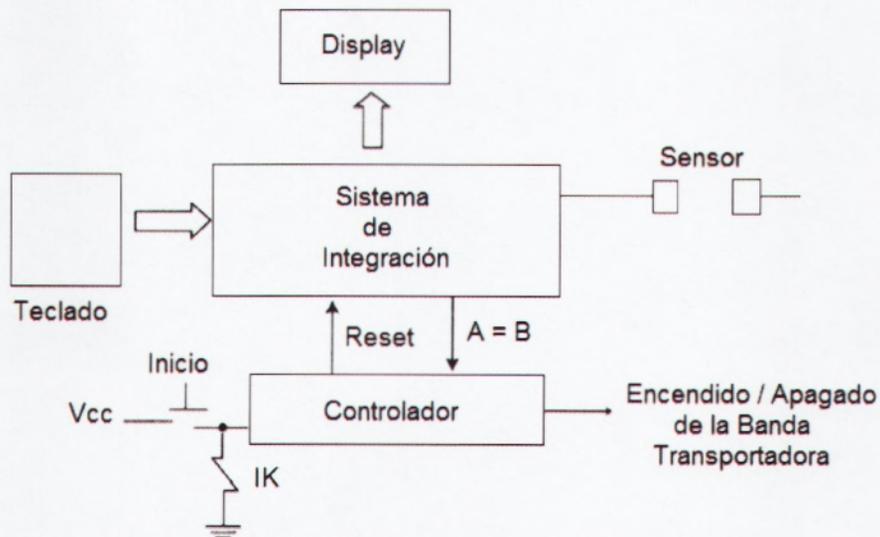


Figura 4. 20: Diagrama de bloques para automatizar proceso de empaquetamiento de muñecas.
Fuente: Los autores

Una vez entendido lo que se requiere el proceso de automatizar el empaquetamiento de muñecas procedemos a realizar la programación correspondiente a esta integración que se muestra en la figura 4.21.

```

1  library ieee;
2  use ieee.std_logic_1164.all;
3  use ieee.std_logic_unsigned.all;
4  entity control is port (
5      elk,reset: in std_logic; boton: in std_logic_vector (0 to 8); sensor: in std_logic;
6      deco: out std_logic_vector (0 to 6); compara: out std_logic);
7  end control;
8  architecture a_control of control is
9      signal Q,C,R: std_logic_vector (3 downto 0);
10     begin
11     proceso1: process (sensor,reset,q)begin
12         if (sensor' event and sensor = '1') then
13             Q <= "0000";
14             Q <= Q + 1;
15         if (reset = '1' or Q = "1001") then
16             Q <= "0000";
17         end if;
18     end if;
19     end process;
20     proceso2: process (elk,boton,R) begin
21         if (elk'event and elk = '1') then
22             R <= C;
23             if (boton = "100000000") then
24                 C <= "0001";
25             elsif (boton = "010000000") then
26                 C <= "0010";
27             elsif (boton = "001000000") then
28                 C <= "0011";
29             elsif (boton = "000100000") then
30                 C <= "0100";
31             elsif (boton = "000010000") then
32                 C <= "0101";
33             elsif (boton = "000001000") then
34                 C <= "0110";
35             elsif (boton = "000000100") then
36                 C <= "0111";
37             elsif (boton = "000000010") then
38                 C <= "1000";
39             else
40                 C <= "1001";
41             end if;
42         end if;
43         case R is
44             when "0000" => deco <= "0000001";
45             when "0001" => deco <= "1001111";
46             when "0010" => deco <= "0010010";
47             when "0011" => deco <= "0000110";
48             when "0100" => deco <= "1001100";
49             when "0101" => deco <= "0100100";
50             when "0110" => deco <= "0100000";
51             when "0111" => deco <= "0001111";
52             when "1000" => deco <= "0000000";
53             when others => deco <= "0001100";
54         end case;
55     end process;
56     compara <= '1' when Q = R else '0';
57 end a_control;

```

Figura 4. 21: Programación en VHDL proceso de empaquetamiento de muñecas.
Fuente: Los autores

Asimismo como en anteriores prácticas procedemos siempre a realizar la compilación, la misma que no debe tener ningún error para así obtener el diseño RTL que se muestra en la figura 4.22, del circuito que permite automatizar el proceso de empaquetamiento de muñecas.

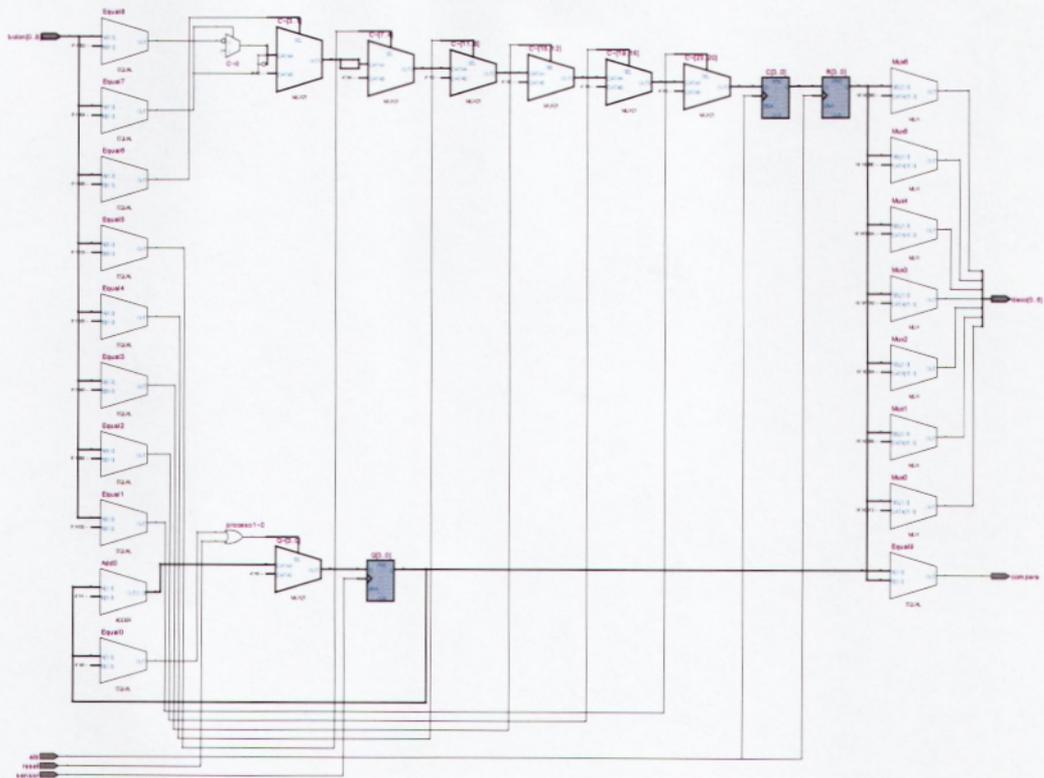


Figura 4. 22: Circuito RTL del proceso de empaquetamiento de muñecas.
Fuente: Los autores

CAPÍTULO 5: CONCLUSIONES Y RECOMENDACIONES.

5.1. CONCLUSIONES

- A través del estado del arte o fundamentos teóricos de la programación en VHDL y de FPGA se pudo determinar que dichas herramientas son empleadas en instituciones de educación superior para la enseñanza de sistemas digitales en cuanto al diseño de controladores.
- Se determinó la eficiencia de la herramienta a utilizar en el Laboratorio de Electrónica específicamente en la asignatura de Sistemas Digitales II en la Carrera de Ingeniería en Telecomunicaciones, para contribuir a su aprendizaje significativo a los estudiantes de la Facultad de Educación Técnica para el Desarrollo en la Universidad Católica de Santiago de Guayaquil.
- Con el desarrollo de la programación y posterior implementación de estas prácticas de sistemas digitales II, el estudiante de VI Ciclo de la Carrera de Ingeniería en Telecomunicaciones estará en capacidad de realizar diseños electrónicos que a futuro servirán como parte de su vida profesional, así adquirir nuevos conocimientos y generar nuevas competencias.

5.2. RECOMENDACIONES

- Realizar la compra de la licencia profesional del software Quartus II de Altera para aplicaciones de mayor envergadura que requieren de librerías que no cuentan la versión gratuita para estudiantes, así como también las gestiones apropiadas para la adquisición de nuevas tarjetas de entrenamiento en FPGA como Cyclone II o mayores, para la materia de Sistemas Digitales II, tomando como referencia el valor de estas tarjetas (ver Anexo 1).

- Contribuir al desarrollo investigativo con aplicaciones y adaptaciones orientados en la carrera de telecomunicaciones, tales como la biomedicina.

- Capacitar al personal involucrado en el cuidado y mantenimiento del laboratorio de electrónica específicamente en las actualizaciones de licencia e instalación de los drivers para compatibilidad de la tarjeta Cyclone II de Altera para así aprovechar en forma eficiente los recursos ya existentes.

REFERENCIAS BIBLIOGRÁFICAS

[AGRAWAL, 1986] AGRAWAL, Om, SHANKAR, Kapil, "PAL device buries registers, brings state machines to life", *Electronic Design*, 24-7-1986, pp. 101-106.

[CHAM, 1988] CHAM, K.M., SOO-YOUNG, CHIN, D., MOLL, J.L., LEE, K., VOORDE, P.V., "*Computer-aided design and VLSI device development*", Kluwer academic publishers, 1988.

[Delgado, 1993] Delgado C., Lecha E., Moré M., Teres Ll., Sánchez L., *Introducción a los lenguajes VHDL, Verilog y UDLI*. Novática No. 112, España, 1993.

[IEEE, 1997] IEEE, Instituto de Ingeniería Eléctrica y Electrónica, *Revista Computer IEEE*, 1977.

[LUPON, 1988] LUPON, Emili, RUBIO, José Antonio, "*Improving performance of PLDs*", Departamento de Ingeniería Electrónica, Universidad de Cataluña, Barcelona, 1988.

[MM, 1981] "Programmable array logic handbook", Monolithic Memories, 1981.

[MM, 1986] "LSI data book", Monolithic Memories, 1986.

[NI, 2011] NI, National Instruments, "Introducción a la Tecnología FPGA: Los Cinco Beneficios Principales" Tutorial NI, 2011.

[PHILIPS, 1987a] "Semi-custom programmable logic devices data handbook", Philips, 1987.

[PHILIPS, 1987b] "System Cell 1987. Design manual. Cell library", Philips, 1987.

[**RANDELL, 1982**] RANDELL, B., TRELEAVEN, P.C., "VLSI architecture", Prentice Hall, New Jersey, 1982.

[**Teres, 1998**] Teres Ll., Torroja Y., Olcoz S., Villar E., *VHDL Lenguaje Estándar de Diseño Electrónico*. McGraw-Hill, 1998.

[**TEXAS, 1990**] "Semiconductor products, circuits design tools and support", Texas Instruments, 1990.

[**XilinxDK, 2003**] Xilinx Inc., "CoolRunner-II RealDigital CPLD," Xilinx Design Kit, USA, Nov. 2003.

[**Pardo, 2000**] F. Pardo y J. A. Boluda, "VHDL Lenguaje para Síntesis y Modelado de Circuitos," Alfaomega ra-ma, España, 2000.

Bibliografía complementaria

T.L. Floyd: *Fundamentos de sistemas digitales*. Prentice Hall, 1998.

Ll. Teres; Y. Torroja; S. Olcoz; E. Villar: *VHDL Lenguaje estándar de diseño electrónico*. McGraw-Hill, 1998.

David G. Maxinez, Jessica Alcalá: *Diseño de Sistemas Embebidos a través del Lenguaje de Descripción en Hardware VHDL*. XIX Congreso Internacional Académico de Ingeniería Electrónica. México, 1997.

C. Kloos, E. Cerny: *Hardware Description Language and their Applications. Specification, modelling, verification and synthesis of microelectronic systems*. Chapman & Hall, 1997.

IEEE: *The IEEE standard VHDL Language Reference Manual*. IEEE-Std-1076-1987, 1988

Zainalabedin Navabi: *Analysis and Modeling of Digital Systems*. McGraw-Hill, 1988.

E J. Ashenden: *The Designer's guide to VHDL*. Morgan Kauffman Publishers, Inc., 1995.

R. Lipsett, C. Schaefer: *VHDL Hardware Description and Design*. Kluwer Academic Publishers, 1989.

S. Mazor, P Langstraat: *A guide to VHDL*. Kluwer Academic Publishers, 1993.

J.R. Armstrong y F. Gail Gray: *Structured Design with VHDL*. Prentice Hall, 1997.

K. Skahill: *VHDL for Programmable Logic*. Addison Wesley, 1996.

J. A. Bhasker: *A VHDL Primer*. Prentice Hall, 1992.

H. Randolph: *Applications of VHDL to Circuit Design*. Kluwer Academic Publisher.

Anexo 1

COMPROBANTE DE PEDIDO Y COMPRA DE 8 TARJETAS FPGA



Orders 1-800-344-4539
Fax 218-681-3380
www.digikey.com

Invoice # 36487894
U.S. \$

701 Brooks Ave. South, P.O. Box 677, Thief River Falls, MN 56701-0677 USA

Sold To:
CUSTOMER 6296633
JANET PARRA ORTEGA
CDLA KENNEDY NORTE Y AV
A ANDRADE ED CLINICA SAN FRANC
GUAYAQUIL, GUAYA 090150
ECUADOR

Bill To:
JANET PARRA O.
SOLNET
CDLA KENNEDY NORTE Y AV
A ANDRADE ED CLINICA SAN FRANC
GUAYAQUIL GUAYA EC090150
ECUADOR

Terms: Mastercard	Invoice Date: 26-OCT-2011	Page: 1
Customer Purchase Order		Sales Order 31295561
Back Orders Accepts to 16-NOV-2011		Account 1826353
Entered By / Date A0FX/17-OCT-2011	Shipped Via UGT	Ship Date 26-OCT-2011
Easy to Remember: 1-800-DIGI-KEY		

For Office Use Only	Received INTERNET	Prev Sales Order 0	Prev Invoice 0	Billing BILL SHIP	Pack List No. 1	Printing Date 26-OCT-2011	Currency Type U.S. \$	MSC # 0
----------------------------	-----------------------------	------------------------------	--------------------------	-----------------------------	---------------------------	-------------------------------------	---------------------------------	-------------------

Idx	Box	Ordered	Cancelled	Shipped	Item Number/Description	Back Order	Unit Price US \$	Amount US \$
1	1	8	0	8	P0528 BOARD DRV DR1 ALTERA CUST REF #: CARLOS ANTONIO BENALCAZAR PARRA (TERRASIC) HTSUS: 8471.41.0150 RCCN: 3A001A7A LIC: EXC RNC LEAD: LEAD FREE ROHS: ROHS COMP COUNTRY/ORIGIN: TAIWAN BOX 1 SHIPPED UGT WEIGHT 18 LBS 0 OZS BOX ID 125674320337712178 ATTN RECIPIENT: EXPORT OF THESE OR ANY COMMODITIES MAY CONSTITUTE A ROUTED TRANSACTION. THIS INVOICE CONTAINS THE NECESSARY INFORMATION TO FILE THE EXPORT UNDER THE U. S. DEPARTMENT OF COMMERCE AUTOMATED EXPORT SYSTEM (AES), WHEN REQUIRED BY 15 CFR PART 30, FOREIGN TRADE REGULATIONS. TOTAL INVOICED 1250.00 SHIPPING CHARGES APPLIED 21.21 ** CHARGES SUBTOTAL ** 1271.21 TOTAL CHARGED TO CREDIT CARD 1271.21 U.S. \$ YOUR CREDIT CARD HAS BEEN CHARGED THE ABOVE INDICATED AMOUNT THE ORDER IS COMPLETE Ship To: HERBERT ALVAREZ ABCARD CARGO SERVICE INC 8560 NW 72 STREET MIAMI FL 33166-0000 Ship From: DIGI-KEY CORPORATION 701 BROOKS AVE. SOUTH P.O. BOX 677 THIEF RIVER FALLS MN 56701-0677 General - WEB ORDER ID: 37692495 17-OCT-2011 CALLED JANET TO VERIFY PERSONAL OR COMPANY ORDER. SHE IS ORDERING FOR HER SON CARLOS TO USE AT UNIVERSITY. A3P9/2234		1250.00	1250.00

Claims for pricing errors, shortages, and defective product must be reported within 30 days of invoice date.

Contact Customer Service at 1-800-858-3616

Anexo 2

Tutorial of ALTERA Cyclone II FPGA Starter Board

This is a simple project which makes the LED and seven-segment display count from 0 to 9.

You will get familiar with Quartus II design software—You will understand basic design steps about Quartus II projects, such as designing projects using *schematic editor* and *HDL*, compiling your design, pin assignment, and downloading it into the FPGA board.

Procedures:

Create a Project:

1. Launch the Quartus II software, select **File** → **New Project Wizard** → **Next**;
2. Choose the Directory and name of the project:
 - a. **Working directory**: Choose the E: where your flash disk will be.
 - b. **Name of this project**: Type counter.
 - c. **Name of the top-level design entity**: Type counter_top.
3. Click **Finish**, when prompted, choose **Yes**;

Assign the Device

4. Choose **Assignments** → **Device**;
5. Under **Family**, choose *Cyclone II*;
6. Under **Available devices**, choose **EP2C20F484C7**;
7. Click **OK**;

Design Entry

8. Choose **File** → **New** → **Block Diagram/Schematic File** to create a new file, Block1.bdf;
9. Click **OK**;
10. Choose **File** → **Save As** and enter the File name: counter_top;
11. Click **Save**;
12. Choose **File** → **New** → **Verilog HDL File** to create a new Verilog File.
13. Click **OK** to create a new file Verilog1.v;
14. Select **File** → **Save As** and enter the File name: counter;
15. Copy the Verilog HDL code from **decade.v** (Verilog source file along with this tutorial) into the blank counter.v file;
16. Save the file by choosing **File** → **Save**;
17. Choose **File** → **Create/Update** → **Create Symbol Files for Current File** to convert the counter.v file to a Symbol File (.sym). You use this Symbol File to add the HDL code to your BDF schematic;
18. Click **Ok**;
19. To add the counter.v symbol to the top-level design, click the **counter_top.bdf** tab;
20. Choose **Edit** → **Insert Symbol**;
21. Double-click the **Project** directory to expand it;
22. Select **Decade**, which is module name of created symbol;
23. Click **Ok**;
24. Move the cursor to the BDF grid Click to place the counter symbol onto the BDF.
25. Add an input pin and an output bus with the following steps:

- a. Choose **Edit** → **Insert Symbol**;
 - b. Under **Libraries**, select **quartus/libraries > primitives > pin > input**;
 - c. Click **OK**;
 - d. Place the input symbol to the BDF grid, move the symbol so that it is touching the **CLK** input to the **Decade** symbol;
 - e. Use the mouse to click and drag the new input pin to the left; notice that the ports remain connected;
 - f. Change the pin name by double-clicking pin name;
26. Do the same procedures as 25.a to 25.f to add input and output pins to other ports. Select **quartus/libraries > primitives > pin > output** for output pins;
 27. When rename the output port 'digit', remember to enter digit[6..0], which means they have 7 pins.
 28. When rename the output port 'Q', remember to enter Q[3..0], which means they have 4 pins.
 29. Save the file by choosing **File** → **Save**;

Pin Assign

30. Choose **Processing** → **Start** → **Start Analysis & Elaboration** in preparation for assigning pin locations;
31. Click **OK** in the message window that appears after analysis and elaboration completes;
32. Choose **Assignments** → **Pins**, which opens the Pin Planner, a spreadsheet-like table of specific pin assignments. The Pin Planner shows the design's pins.
33. In the **Location** column next to each of the node names, add the coordinates (pin numbers) for the actual values to use with the board. For the pin layout of components on the Board, please refer to Appendix A;

Compile Your Project

34. In the **Processing** menu, choose **Start Compilation**, If you are asked to save changes to your BDF, click **Yes**;
35. When compilation is complete, the Quartus II software displays a message. Click **OK** to close the message box.

Program the Device

36. Choose **Tools** → **Programmer**. The Programmer window opens.
37. Connect the USB Cable from the Computer to the FPGA Board;
38. Set the **RUN/PROG** switch (SW12) to the **RUN** position;
39. Turn the board on using the on/off switch (SW11)Power;
40. Click **Start**. The file downloads to the development board.

Congratulations, you have created, compiled, and programmed your design!

If everything is Ok, you will see the LEDs and 7-Segment show the count sequence from 0 to 9.

For further development, please refer to the following documents:

UserGuide: http://www.altera.com/literature/ug/ug_cii_starter_board.pdf

Pin Assignment: http://www.altera.com/literature/manual/mnl_cii_starter_board_rm.pdf

Schematic: http://www.altera.com/products/devkits/altera/documents/cy2_fpga_starter_board_schematic.pdf

Pin Connections on FPGA Board

Table 2-9. Clock Circuit FPGA Pin Connections

Signal Name	FPGA Pin	Description
CLOCK_27	PIN_D12	27 MHz clock input
CLOCK_50	PIN_L1	50 MHz clock input
CLOCK_24	PIN_B12	24 MHz clock input from USB Blaster
EXT_CLOCK	PIN_M21	External (SMA) clock input

Table 2-10. Push Button Switch FPGA Pin Connections

Switch	FPGA Pin	Description
KEY[0]	PIN_R22	Pushbutton[0]
KEY[1]	PIN_R21	Pushbutton[1]
KEY[2]	PIN_T22	Pushbutton[2]
KEY[3]	PIN_T21	Pushbutton[3]

Table 2-11. Toggle Switch FPGA Pin Connections

Switch	FPGA Pin	Description
SW[0]	PIN_L22	Toggle Switch[0]
SW[1]	PIN_L21	Toggle Switch[1]
SW[2]	PIN_M22	Toggle Switch[2]
SW[3]	PIN_V12	Toggle Switch[3]
SW[4]	PIN_W12	Toggle Switch[4]
SW[5]	PIN_U12	Toggle Switch[5]
SW[6]	PIN_U11	Toggle Switch[6]
SW[7]	PIN_M2	Toggle Switch[7]
SW[8]	PIN_M1	Toggle Switch[8]
SW[9]	PIN_L2	Toggle Switch[9]

Table 2-12. LED FPGA Pin Connections

Signal Name	FPGA Pin	Description
LEDR[0]	PIN_R20	LED Red[0]
LEDR[1]	PIN_R19	LED Red[1]
LEDR[2]	PIN_U19	LED Red[2]
LEDR[3]	PIN_Y19	LED Red[3]
LEDR[4]	PIN_T18	LED Red[4]
LEDR[5]	PIN_V19	LED Red[5]
LEDR[6]	PIN_Y18	LED Red[6]
LEDR[7]	PIN_U18	LED Red[7]
LEDR[8]	PIN_R18	LED Red[8]
LEDR[9]	PIN_R17	LED Red[9]
LEDG[0]	PIN_U22	LED Green[0]
LEDG[1]	PIN_U21	LED Green[1]
LEDG[2]	PIN_V22	LED Green[2]
LEDG[3]	PIN_V21	LED Green[3]
LEDG[4]	PIN_W22	LED Green[4]
LEDG[5]	PIN_W21	LED Green[5]
LEDG[6]	PIN_Y22	LED Green[6]
LEDG[7]	PIN_Y21	LED Green[7]

Table 2-13. Seven-Segment Display FPGA Pin Connections (Part 1 of 2)

Signal Name	FPGA Pin	Description
HEX0[0]	PIN_J2	Seven-Segment segment 0[0]
HEX0[1]	PIN_J1	Seven-Segment segment 0[1]
HEX0[2]	PIN_H2	Seven-Segment segment 0[2]
HEX0[3]	PIN_H1	Seven-Segment segment 0[3]
HEX0[4]	PIN_F2	Seven-Segment segment 0[4]
HEX0[5]	PIN_F1	Seven-Segment segment 0[5]
HEX0[6]	PIN_E2	Seven-Segment segment 0[6]
HEX1[0]	PIN_E1	Seven-Segment segment 1[0]
HEX1[1]	PIN_H6	Seven-Segment segment 1[1]

Table 2-13. Seven-Segment Display FPGA Pin Connections (Part 2 of 2)

Signal Name	FPGA Pin	Description
HEX1[2]	PIN_H5	Seven-Segment segment 1[2]
HEX1[3]	PIN_H4	Seven-Segment segment 1[3]
HEX1[4]	PIN_G3	Seven-Segment segment 1[4]
HEX1[5]	PIN_D2	Seven-Segment segment 1[5]
HEX1[6]	PIN_D1	Seven-Segment segment 1[6]
HEX2[0]	PIN_G5	Seven-Segment segment 2[0]
HEX2[1]	PIN_G6	Seven-Segment segment 2[1]
HEX2[2]	PIN_C2	Seven-Segment segment 2[2]
HEX2[3]	PIN_C1	Seven-Segment segment 2[3]
HEX2[4]	PIN_E3	Seven-Segment segment 2[4]
HEX2[5]	PIN_E4	Seven-Segment segment 2[5]
HEX2[6]	PIN_D3	Seven-Segment segment 2[6]
HEX3[0]	PIN_F4	Seven-Segment segment 3[0]
HEX3[1]	PIN_D5	Seven-Segment segment 3[1]
HEX3[2]	PIN_D6	Seven-Segment segment 3[2]
HEX3[3]	PIN_J4	Seven-Segment segment 3[3]
HEX3[4]	PIN_L8	Seven-Segment segment 3[4]
HEX3[5]	PIN_F3	Seven-Segment segment 3[5]
HEX3[6]	PIN_D4	Seven-Segment segment 3[6]

2. Realice el un Programa con nombre analizador de espectros.vi. El panel frontal presenta los siguientes objetos.

- a. Un indicador tipo gráfico Señal muestra la señal en dominio del tiempo en forma de osciloscopio. Las herramientas de escala y gráficas permiten observar en detalle la forma de onda, hacer zoom, etc.
- b. Un indicador tipo gráfico Espectro de Frecuencia muestra el espectro de potencia de la señal adquirida, calculado a partir de una transformada rápida de Fourier (FFT). Las herramientas de escala, gráfica y de cursor permiten observar en detalle el espectro, hacer zoom, mover el cursor, etc.
- c. El botón Congelar permite congelar las imágenes para hacer los análisis.
- d. El botón STOP termina la ejecución del programa.

3. Se Realiza el panel Frontal con los siguientes pasos:(*VerFigura4.55*)

- a. En la pantalla del Panel Fontal se dibuja dos IndicadoresGráficos, para esto seleccionamos de la paleta de controles el menú Modern, Graph, WaveForm Chart.
- b. Seleccione de la paleta de controles el menú modern, Boolean, Stop Button.
- c. Seleccione de la paleta de controles el menú modern, Boolean, Stop Button(Cambio de Nombre a Congelar).

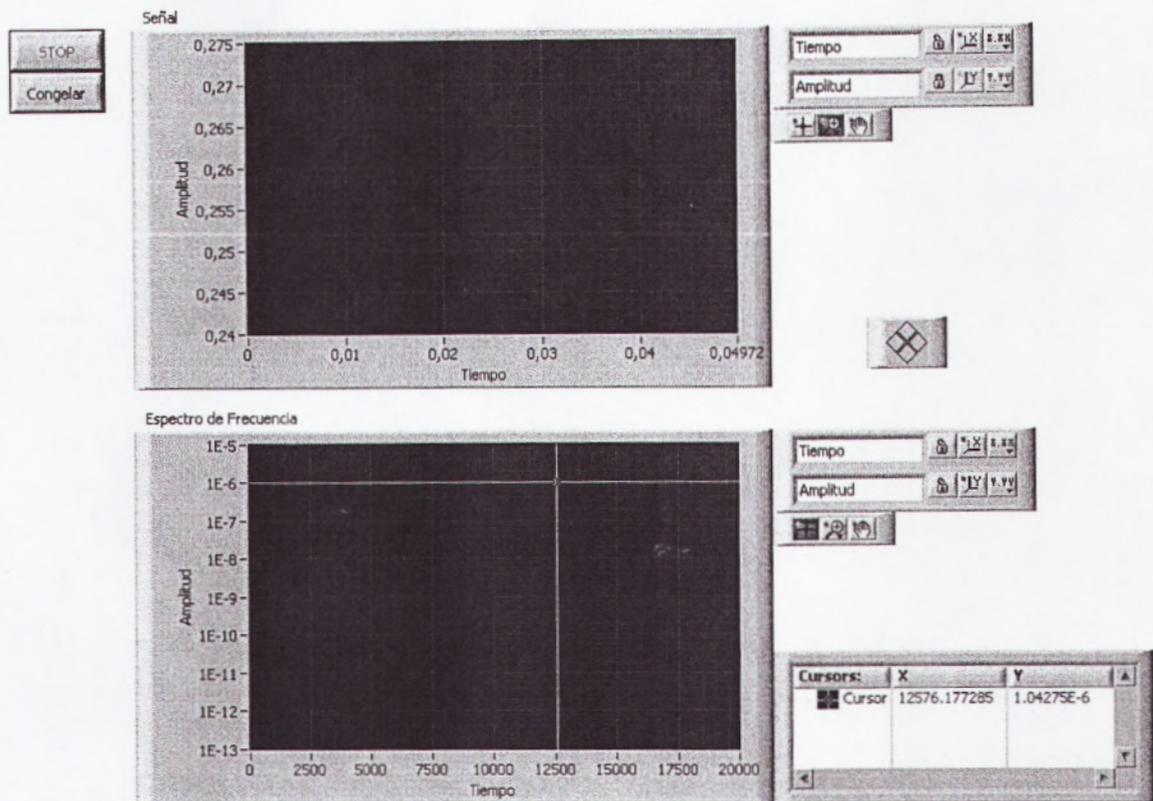


Figura4.55: Panel Frontal Práctica DSA1

4. Abra el diagrama de bloque y realice lo siguiente:
 - a. Se aplica el literal a del ítem 4 de la práctica S1.
 - b. Seleccione de la paleta de funciones el menú programing, Structures, While Loop.
 - c. Seleccione de la paleta de funciones el menú programing, Structures, Case Structure.
 - d. Seleccione de la paleta de funciones el menú Programing, Waveform, Analog Waveform, Waveform Measure, FFT Power Spectrum.vi

- e. Seleccione de la paleta de funciones el menú Programing, Waveform, Analog Waveform, Waveform Measure, Basic Averaget, DC-RMS.
- f. Seleccione de la paleta funciones el menú Programing. Timing, Wait.
- g. Realizar las conexiones de la imagen. (VerFigura4.56)

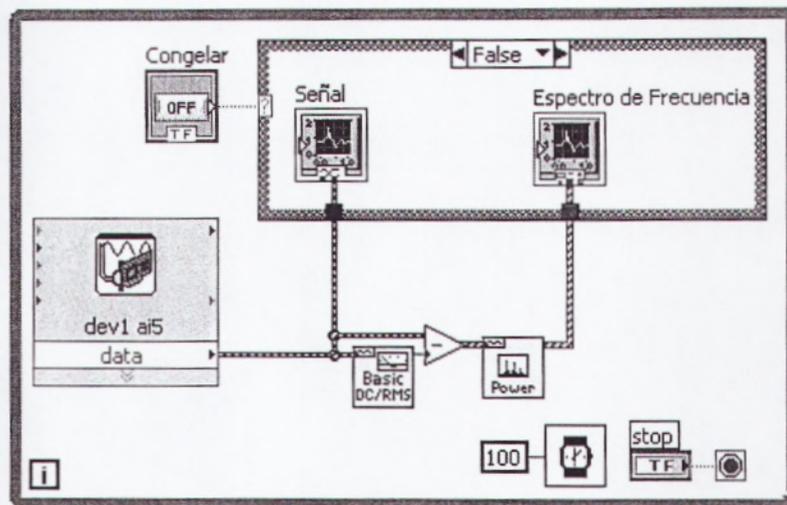


Figura4.56: Diagrama de Bloques Práctica DSA1

5. Correr el Programa.
 - a. Producir un sonido cerca del micrófono. Utilizar el sonido que produce al presionar diferentes teclas de un teléfono celular (DTMF). En la siguiente gráfica se presenta la composición de tonos de un teclado de este tipo.(VerFigura4.57)

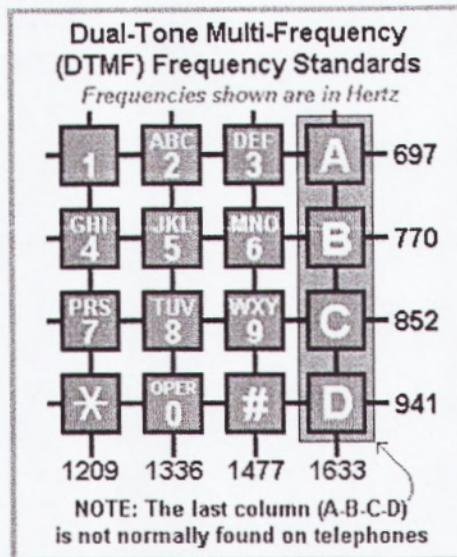


Figura4.57: Teclado DTMF

- b. Observar que en ausencia de sonido existe una señal ruidosa con una componente de corriente continua cerca de 0.25V. Esta componente es también típica en las señales adquiridas de un acelerómetro. (VerFigura4.58)

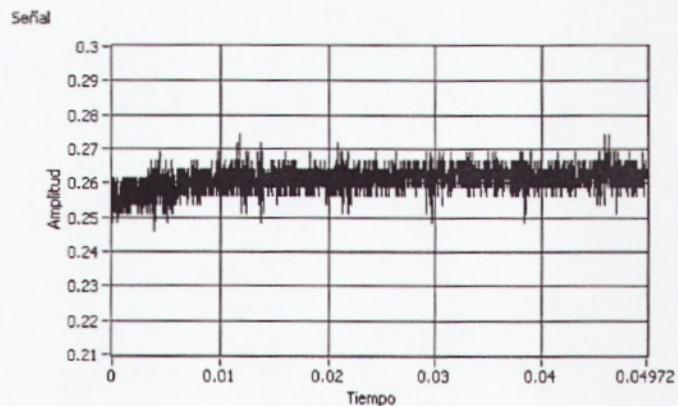


Figura 4.58: Señal Ruidosa

- c. La desventaja de tener una señal de estas características es que el rango de voltaje de la tarjeta se desperdicia en toda la zona en que no existe señal. Por esta razón es necesario utilizar una tarjeta con acoplamiento AC/DC y alta resolución en aplicaciones

profesionales o de alto desempeño. En estas prácticas se corregirá esta componente DC mediante software.

- d. Presionar la tecla #5. Congelar la imagen para analizarla.
- e. En el gráfico Espectro de Frecuencia (*VerFigura4.59*), se podrán observar dos componentes de armónicos, el principal en 1336 Hz, y una más baja en 770 Hz.

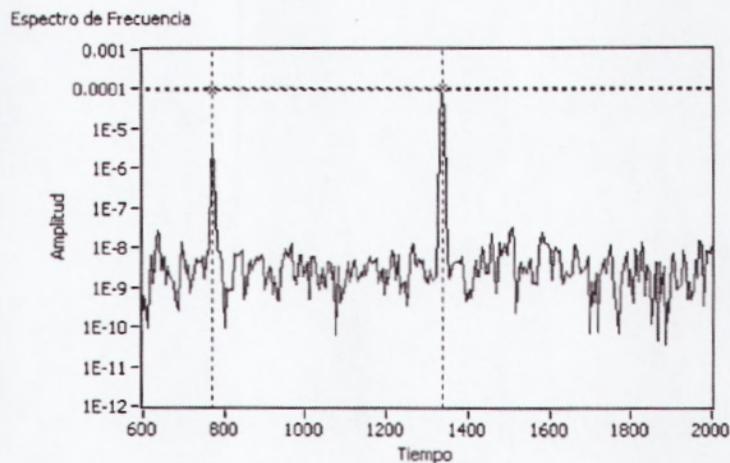


Figura4.59: Espectro de Frecuencia.

- f. Descongelar la imagen presionando nuevamente sobre el botón Congelar. Las gráficas deben mostrar nuevamente en tiempo real la señal adquirida.
- g. Presionar la tecla #2 del teclado del teléfono y adquirir el sonido que produce. Cuando se tenga una señal clara, presionar el botón Congelar para que la imagen no se pierda al soltar la tecla.
- h. En el gráfico Espectro de Frecuencia(*VerFigura4.60*),se podrán observar dos componentes de armónicos, el principal en 1336 Hz, y una más baja en 697 Hz.

Espectro de Frecuencia

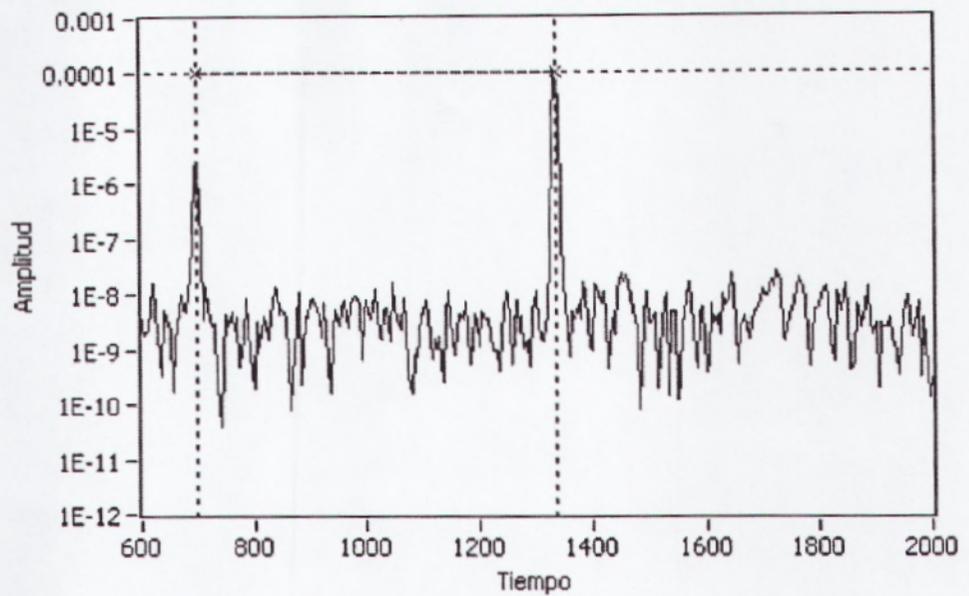


Figura4.60: Componentes Armónicos

- i. Repetir el análisis, esta vez presionando la tecla #9 Observar los componentes del espectro de potencia, el alto en 1477Hz y el bajo en 852 Hz. (VerFigura4.61)

Espectro de Frecuencia

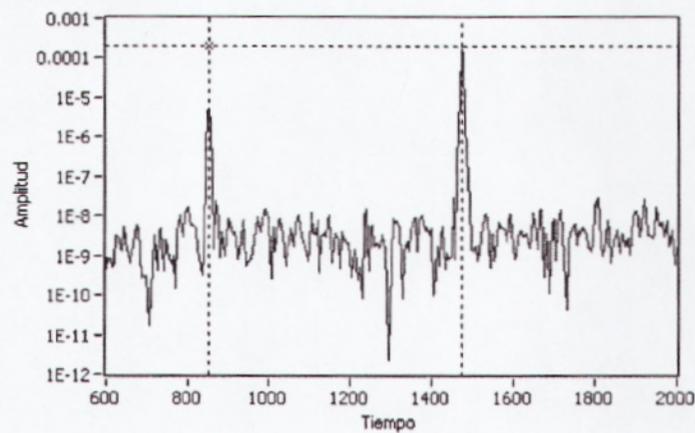


Figura 4.61: Espectro de Potencia

- j. Confirmar las mediciones con la gráfica que especifica las frecuencias estándar para un teclado DTMF.

6. Parar el programa presionando el botón de STOP.
7. Abrir el diagrama de bloques para analizar el algoritmo. (VerFigura4.62)
 - a. La adquisición de datos se está realizando a una velocidad de 40kS/s, y se adquieren 8k muestras.
 - b. El programa mide la componente de corriente continua DC y la resta de la señal original para obtener el espectro de potencia sin dicha componente.
 - c. El botón Congelar inhibe la escritura de nuevos datos en las gráficas Señal y Espectro de Frecuencia.

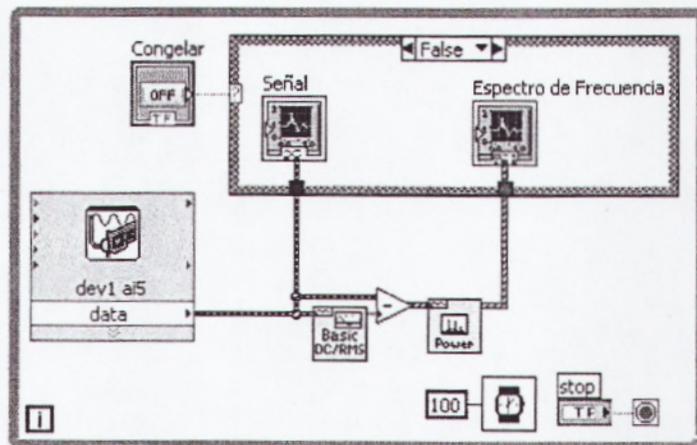


Figura4.62: Diagrama de Bloque DSA1

8. Cerrar el Programa sin hacer Cambios.

4.5.2 PRÁCTICA D2: DETECCIÓN DTMF

En esta práctica se utilizará como base la práctica anterior, junto a herramientas de análisis más avanzadas para detectar qué tecla del teléfono se ha presionado, tales como eliminación de la componente de corriente continua, filtrado de frecuencias no deseadas, y medición de tonos.

1. Realizar las siguientes conexiones entre el EPC y la tarjeta DAQ:
 - a. Conectar la salida del micrófono MIC OUT a una entrada analógica de la tarjeta DAQ. En estas prácticas se utiliza la entrada analógica número cinco AI5 en conexión RSE.
 - b. Conectar la referencia GND de la tarjeta DAQ a la referencia GND del EPC. Todas las referencias GND de la tarjeta DAQ están conectadas internamente (en el modelo USB-6009 y similares); igualmente, todas las referencias GND del EPC están conectadas internamente.
 - c. Conectar la fuente de poder del EPC.

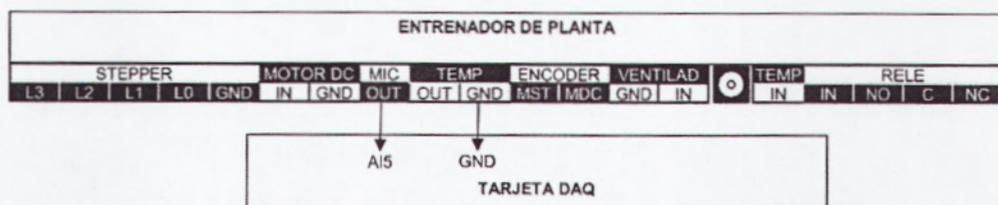


Figura 4.63: Conexión tarjeta DAQ Práctica DSA2

2. Realice un programa con nombre detector dmft.vi. El panel frontal presenta los siguientes objetos.
 - a. Un indicador tipo gráfico Señal muestra la señal en dominio del tiempo en forma de osciloscopio. Las herramientas de escala y gráficas permiten observar en detalle la forma de onda, hacer zoom, etc.
 - b. Un indicador tipo gráfico Espectro de Frecuencia muestra el espectro de potencia de la señal adquirida, calculado a partir de una transformada rápida de Fourier (FFT). Las herramientas de escala, gráfica y de cursor permiten observar en detalle el espectro, hacer zoom, mover el cursor, etc.
 - c. Notar que las autoescalas están desactivadas, de forma que se puedan observar solamente la porción de información que interesa.
 - d. Un indicador tipo string Tecla que muestra qué botón se ha detectado presionado.
 - e. El control numérico Umbral define el valor sobre el cual serán considerados los tonos. Esto permite que al detectar ruido de baja amplitud, éste no sea considerado en las mediciones. El valor predeterminado es $1E-8$, pero puede ser cambiado para teclados con volumen más bajo.
 - f. El botón Congelar permite congelar las imágenes para hacer los análisis sin necesidad de parar el programa.
 - g. El botón STOP termina la ejecución del programa.
3. Realice los siguientes pasos en el Panel Frontal. (Ver Figura 4.64)

- a. Se aplica el literal a, b y c del Ítem 3 de la Práctica SDA1.
- b. Seleccione de la paleta de controles el menú Modern. String & Path, String Indicator.
- c. Seleccione de la paleta de controles el menú Modern, Numeric, Numeric Control.

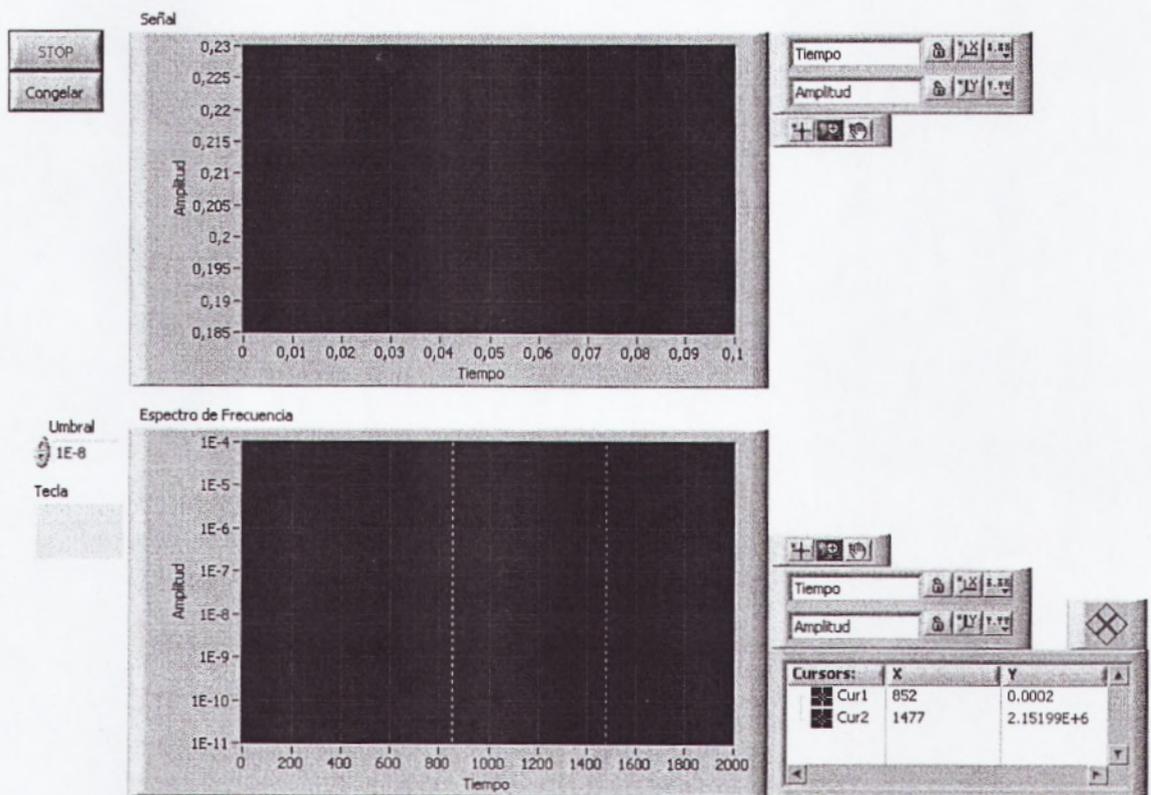


Figura4.64: Panel Frontal Práctica DSA2

4. Abra el diagrama de Bloque y realice lo siguiente:

- a. Se aplica el literal a del ítem 4 de la Práctica S1.
- b. Seleccione de la paleta de funciones el menú programing, Structures, While Loop.
- c. Seleccione de la paleta de funciones el menú programing, Structures, escoja tres Case Structure.
- d. Seleccione de la paleta de funciones el menú programing, Structures, For Loop.
- e. Seleccione de la paleta de funciones el menú Programing, Waveform, Analog Waveform, Waveform Measure, FFT Power Spectrum.vi
- f. Seleccione de la paleta de funciones el menú Programing, Waveform, Analog Waveform, Waveform Measure, Basic Averaget, DC-RMS.
- g. Seleccione de la paleta de funciones el menú Programing, Waveform, Analog Waveform, Waveform Measure, Extract Multiple Tone Information.
- h. Seleccione de la paleta de funciones el menú Express, Signal Analysis, Filter.
- i. Seleccione de la paleta de Funciones el menú Programing, escoja dos Array, Index Array.
- j. Seleccione de la paleta funciones el menú Programing. Timing, Wait.
- k. Realice las conexiones de la Grafica. (*VerFigura4.65*)

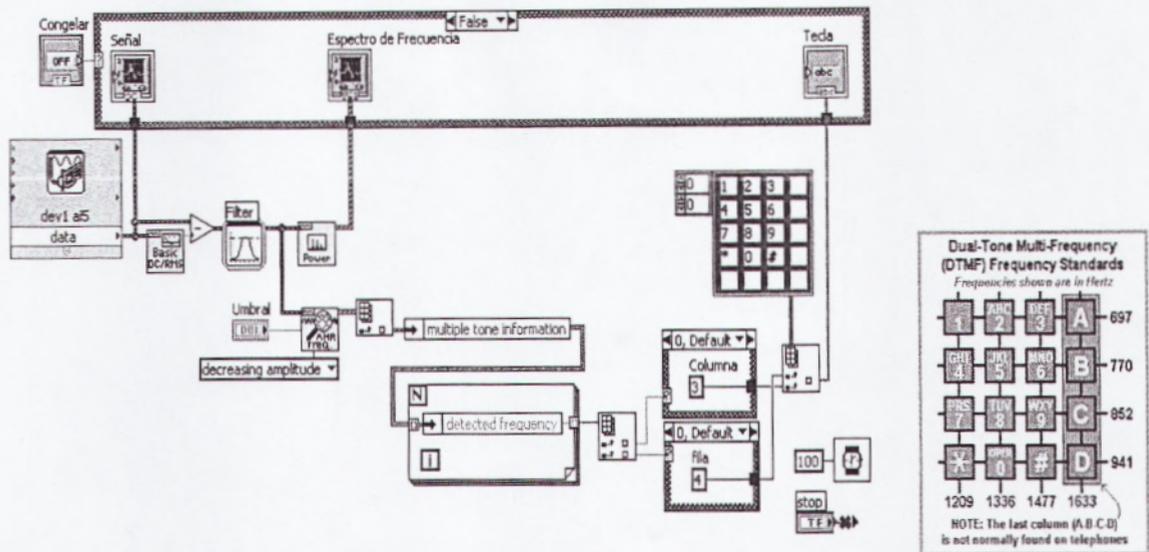


Figura 4.65: Diagrama de Bloques Práctica DSA2

5. Correr el Programa.

- Acercar el audífono del teléfono al micrófono del EPC y presionar una tecla.
- Observar la forma de la onda, el espectro de frecuencia, y el indicador Tecla para confirmar que se ha detectado la tecla presionada.
- En caso de no detectarse, acercar más el teléfono al micrófono.

6. Parar el Programa Presionando El Botón Stop.

7. Abrir el diagrama de bloques para analizar el algoritmo.

- a. Puesto que la señal original tiene una componente de corriente continua, se resta la señal original menos el valor DC para eliminarla.
 - b. Puede existir una componente de ruido, dependiendo de la frecuencia de red de alimentación, de la fuente de poder, del muestreo y del ruido electromagnético. Por lo tanto es conveniente filtrar la señal de entrada con un filtro de paso bajo para eliminar esas componentes "alias" de la señal y detectar únicamente las frecuencias alrededor de los tonos buscados.
 - c. El programa utiliza la función de medición de varios componentes de frecuencia, los organiza por orden de amplitud, de la más alta a la más baja, y toma los dos componentes más altos.
 - d. Las estructuras de caso detectan si los dos tonos principales están cerca de las frecuencias estándar de un teclado DTMF con una precisión de ± 10 Hz.
 - e. Si ambas frecuencias están en rango, se indexa un arreglo bidimensional de strings que contienen la disposición del teclado según los tonos.
8. Cerrar el programa sin hacer cambios.

3.6 PRÁCTICAS DE RELÉ DE PROPÓSITO GENERAL

El EPC incluye un relé electromecánico para aplicaciones de conmutación de propósito general. Este relé es tipo SPDT, por lo tanto incluye un contacto común C, un contacto normalmente abierto NO (por "Normally Open"), y un contacto normalmente cerrado NC (por "Normally Closed"). Para controlar la bobina IN (por "Input") se la alimenta con una señal tipo TTL.

El relé funciona con lógica inversa; es decir:

- Cuando se envía un estado BAJO (0V) a la bobina del relé, éste se activa conmutando el terminal común C con el terminal normalmente abierto NO.
- Cuando se envía un estado ALTO (5V) a la bobina del relé, éste se desactiva conmutando el terminal común C con el terminal normalmente cerrado NC.



NOTA IMPORTANTE: Aunque en la superficie del relé puede leerse que soporta conmutaciones de hasta 10A, debido a consideraciones de aislamiento no se deben conmutar cargas más altas que 1A.

4.6.1 PRÁCTICA R1: TEST DE CONMUTACIÓN

En esta práctica se probará el funcionamiento del relé enviando una salida digital de la tarjeta DAQ a la bobina RELE IN del EPC, y monitoreando los terminales de conmutación del relé (NO y NC) mediante líneas de entradas digitales de la tarjeta DAQ.

1. Realizar las siguientes conexiones entre el EPC y la tarjeta DAQ:
 - a. Conectar la entrada de la bobina del relé del EPC marcada como RELE IN a una salida digital de la tarjeta DAQ. En este ejemplo se utiliza la línea digital P1.0.
 - b. Conectar la referencia GND de la tarjeta DAQ a la referencia GND del EPC. Todas las referencias GND de la tarjeta DAQ están conectadas internamente (en el modelo USB-6009 y similares); igualmente, todas las referencias GND del EPC están conectadas internamente.
 - c. Conectar el terminal común del relé del EPC marcado como RELE C a una referencia GND del mismo EPC.
 - d. Conectar las salidas de conmutación del relé del EPC Normalmente Abierto NO y Normalmente Cerrado NC a dos líneas de entrada digital de la tarjeta DAQ respectivamente. En este ejemplo se utilizan las líneas digitales P1.1 y P1.2.
 - e. Conectar la fuente de poder del EPC. Confirmar que el voltaje de la fuente sea de 12V.

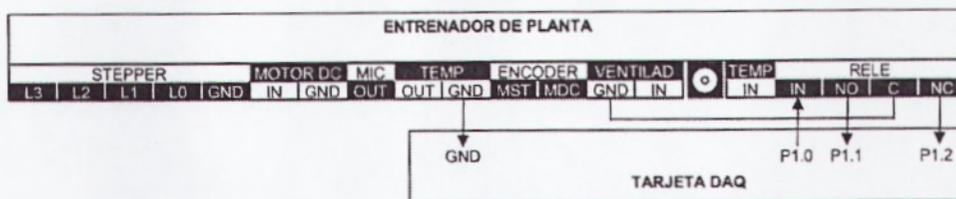


Figura 4.66: Conexión Tarjeta DAQ Práctica R1

2. Realice un Programa con nombre relé.vi. El panel frontal presenta los siguientes objetos:
 - a. Un control booleano tipo switch llamado Relé permite operar sobre la bobina del relé.
 - b. Cuando el Relé está en estado BAJO, la tarjeta DAQ produce una señal de 0V, activando el relé del EPC y conmutando entre el terminal común C y el Normalmente Abierto NO. Cuando el Relé está en estado ALTO, la tarjeta DAQ produce una señal de 5V, desactivando el relé y conmutando entre el terminal común C y el Normalmente Cerrado NC. El indicador gráfico muestra la forma en que opera el relé del EPC.
 - c. Al operar sobre el switch Relé del programa, se podrá escuchar el mecanismo del relé del EPC al conmutar entre los diferentes terminales.
 - d. El botón STOP termina la ejecución del programa.
3. Realice los siguientes pasos en el Panel frontal:
 - a. Seleccione de la paleta de controles el menú modern, boolean, stop Button.

- b. Seleccione de la paleta de controles el menú modern, boolean, Vertical Toggle Switch.
- c. Seleccione de la paleta de controles el menú modern, boolean, escojo tres round led.

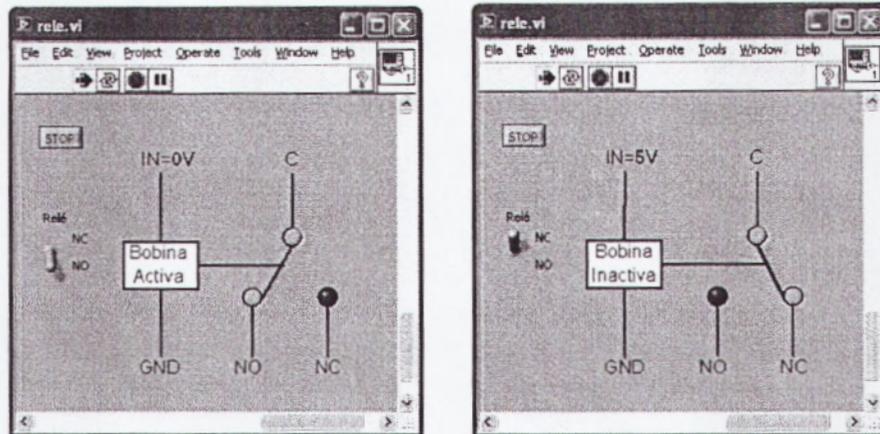


Figura 4.67: Panel Frontal Práctica R1

4. Abrir el Diagrama de Bloques y se realiza lo siguiente:
 - a. Se aplica el literal a del ítem 4 de la Práctica S1, con dos Daq. Assistant.
 - b. Seleccione de la paleta de Funciones el menú Programing, escoja dos Array, Index Array.
 - c. Seleccione de la paleta de Funciones el menú Programing, escoja dos Array, Buil Array.
 - d. Seleccione de la paleta funciones el menú Programing. Timing, Wait Until Next ms multiple.

e. Realice las conexiones de la Grafica. (VerFigura4.68)

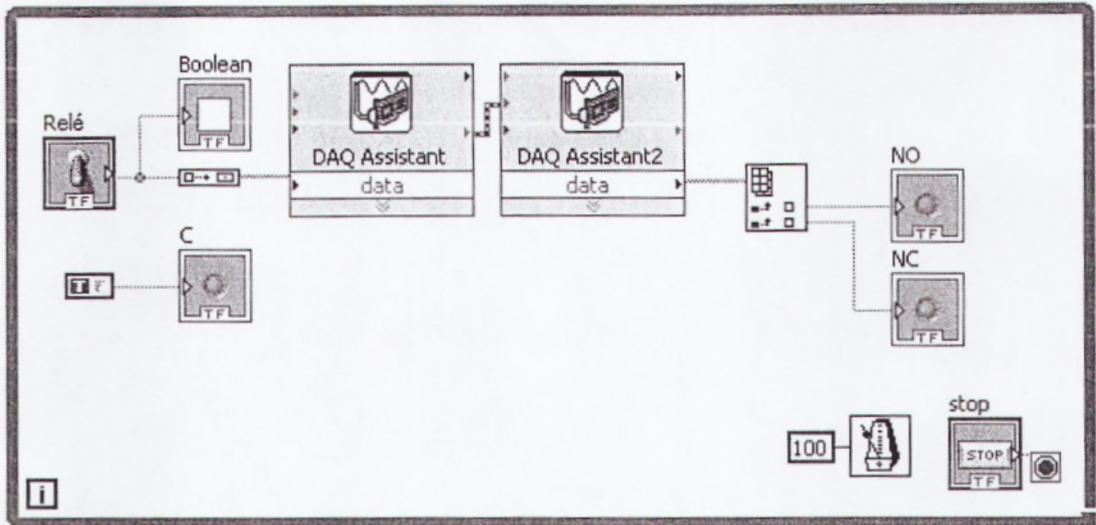


Figura 4.68: Diagrama de Bloques Práctica R1

5. Correr el programa y operar sobre el control Relé. Observar el funcionamiento del relé.
6. Si se desea, se puede conectar una carga externa con su propia alimentación. Para ello es necesario desconectar previamente las líneas digitales de la tarjeta DAQ que se usaron para monitorear el estado de los contactos NO y NC.
7. Parar el programa presionando el botón STOP.
8. Cerrar el programa sin hacer cambios.

CAPÍTULO V: CONCLUSIONES Y RECOMENDACIONES

5.1 CONCLUSIONES

Una vez realizado un amplio estudio de un sistema de control, en el cual se profundizó en Control PID para ver sus algoritmos de trabajo en procesos industriales; nos lleva a presentar el EPC, el Entrenador Electrónico, como herramienta de práctica en el Laboratorio de Electrónica, junto con un diseño prototipo para mejoras futuras del EPC.

Al efectuar análisis de control Industrial, se demostró las bondades de estudio con las que cuenta el entrenador electrónico, en la forma del aprendizaje del estudiante, de las carreras de Ingeniería de la Facultad Técnica para el Desarrollo. El analizar señales de sistemas de medición de temperaturas aplicada por ejemplo para industrias que trabajen con calderos, o sistema de control de velocidad aplicada en la industria Automotriz, etc. Hacen que el EPC sea una herramienta pedagógica muy útil para los estudiantes.

El presente estudio e implementación del EPC es aplicado, ya que se dirige a la simulación de un proceso de control en un campo laboral real, con lo que no solo se queda en fundamentos teóricos, es decir, se basa en acciones concretas para afrontar un problema. Dar soluciones mediante la toma de decisiones con la que el estudiante de hoy, se encuentra en el ámbito profesional, logrando así una mejor operatividad de determinado sistema de control.

El laboratorio de la Facultad Técnica con el EPC, contará con otro instrumento de estudio, con el que incentivara la investigación y desarrollo de tecnología de los estudiantes. Además sumara recursos permitiendo estar a la vanguardia en la enseñanza tecnológica.

5.2 RECOMENDACIONES

Dada las conclusiones obtenidas en el presente texto, se sugieren las siguientes recomendaciones:

- Sugerir a los profesores de cátedra como, Automática, Cálculo, Física, Electrónica, el uso en sus materias de los EPC, para que todos esos fundamentos teóricos se vean simulados en un proceso de control mediante el EPC.
- Si a futuro se necesita implementar mayor EPC en el laboratorio de Electrónica, se deja el diseño para la fabricación de un EPC con más mejoras; en el que inclusive los estudiantes podrán realizarlo.
- Realizar con los EPC cursos de procesos de control industrial, para estudiantes de la universidad o público en general, proporcionando así una nueva alternativa de estudio para nuestra Facultad, esto hará que se genere recursos económicos y mayor cantidad de estudiantes en las carreras de Ingeniería que ofrece la Facultad Técnica para el Desarrollo.

ANEXO 1

ESPECIFICACIONES GENERALES

REQUERIMIENTOS

■ Requerimientos de Hardware

- Entrenador de Planta de Control EPC
- Fuente de poder AC/DC de 12V, 1200Ma
- Cable de conexión para bombillo halógeno de 110/220 VAC 50/60Hz.
- Tarjeta de adquisición de datos genérica de National Instruments (Ejemplo DAQ USB-6009) con las siguientes características mínimas:
 - 4 salidas digitales tipo TTL
 - 1 entrada analógica de 14 bits, +/-10V, 40KS/s, soporte de Adquisición de formas de onda de corriente alterna (N Samples) y de corriente continua.
 - 1 salida analógica de 12 bits, 100 S/s, 0-5V.
 - 1 entrada de contadores de 5MHz
- Destornillador pequeño
- Computador con Windows XP/Vista

■ Requerimientos de Software

- LabVIEW 8.6 o superior
- Toolkit de control PID
- Recomendado: Toolkit de diseño e identificación de sistemas de control.

■ Prerrequisitos

- Conocimientos de programación en LabVIEW fundamental (consulte con National Instruments para conocer sobre las opciones decapacitación)
- Conocimientos básicos de sistemas de adquisición de datos y sensores.
- Para las prácticas de control, conocimientos básicos de teoría de control moderno

ANEXO 2

SEÑALES DE MEDICIÓN Y CONTROL

Entradas Digitales: Bombillo 1 Ventilador 1 Relé 1 Stepper 4 Total: 7 0-5VDC Compatibles TTL	Entradas Analógicas: Motor DC 1 Total: 1 Nivel 0-5 VDC Tasa Muestreo 40kS/s
Salidas de Tren de Pulsos Motor DC 1 Stepper 1 Total: 2 0-5 VDC Compatibles TTL	Salidas Analógicas: Temperatura 1 Micrófono 1 Total: 2 Nivel 0-5 VDC Consumo 5mA
Relé Electromecánico Tipo SPDT Bobina 1 comandado TTL Común 1 Normalmente Abierto 1 Normalmente Cerrado 1 Voltaje de Conmutación 120VAC, 48VDC Corriente de Conmutación 1A	

ALIMENTACIÓN

12 VDC, 1200 mA para alimentación del EPC, motores y componentes electrónicos

120 VAC, 50W para alimentación del bombillo dicroico

DIMENSIONES FÍSICAS

Ancho 19cm x Alto 9,5cm x Profundidad 13cm

ANEXO 3

ESPECIFICACIONES MÍNIMAS DEL SISTEMA DAQ

Características Mínimas del Sistema de Adquisición de Datos

Digital I/O

Number of channels.....	12 total
8 (P0.<0..7>)	
4 (P1.<0..3>)	
Direction control.....	Each channel individually programmable as input or output
Output driver type	
USB-6008.....	Open-drain
USB-6009.....	Each channel individually programmable as push-pull or open-drain.
Compatibility	CMOS, TTL, LVTTL
Internal pull-up resistor.....	4.7 k Ω to +5 V
Power-on state	Input (high impedance)
Absolute maximum voltage range	-0.5 to +5.8 V

Analog Input

Input range, single-ended	± 10 V
ADC Resolution (bits).....	14 bits (differential)
Input range, differential	$\pm 20, \pm 10, \pm 5, \pm 4, \pm 2.5, \pm 2, \pm 1.25, \pm 1$ V
Maximum working voltage	± 10 V
Oversvoltage protection.....	± 35 V
FIFO buffer size.....	512 B
Timing resolution.....	41.67 ns (24 MHz timebase)
Timing accuracy.....	100 ppm of actual sample rate
Input Impedance.....	144 k Ω
Trigger source.....	Software or external digital trigger
System noise	0.3 LSB _{rms} (± 10 V range)

Analog Output

Absolute accuracy (no load).....	7 mV typical, 36.4 mV maximum at full scale
Number of channels.....	2
Type of DAC.....	Successive approximation
DAC resolution	12 bits
Maximum update rate.....	150 Hz, software-timed
Output range	0 to +5 V
Output impedance	50 Ω
Output current drive	5 mA
Power-on state	0 V
Slew rate	1 V/ μ s
Short-circuit current	50 mA

Counter

Number of counters	1
Resolution.....	32 bits
Counter measurements.....	Edge counting (falling edge)
Pull-up Resistor	4.7 k Ω to 5 V
Maximum input frequency	5 MHz
Minimum high pulse width	100 ns
Minimum low pulse width	100 ns
Input high voltage	2.0 V
Input low voltage	0.8 V

BIBLIOGRAFÍA

- National Instrument
- Automatización y Control Industrial. Universidad de Quilmes
- LabVIEW Basic Training. National Instrument
- Control PID. Universidad de san Andrés
- Sistema de Control PID. National Instrument
- Tutoriales de control con Matlab – Universidad de Michigan
- Smith, Carlos A.; Corripio, Armando B. Principles and practice of automatic process control. Second edition. John Wiley & Sons. 1997
- Marlin, Thomas E. Process control: designing processes and control systems for dynamic performance. Mc Graw-Hill. 1975.
- Corripio, Armando B. Tuning of industrial control systems. Instrument Society of America. 1990.
- Roots, William K. Fundamentals of temperature control. Academic Press. 1969.

4.3.2 PRÁCTICA M2:SEÑAL DE ENCODER – FORMA ANALÓGICA

En esta práctica se adquirirá la señal de salida del encoder del Motor DC para analizarla en forma analógica. Además se medirá la frecuencia para determinar la velocidad del motor.

1. Realizar las siguientes conexiones entre el EPC y la tarjeta DAQ:
 - a. Conectar una salida analógica de la tarjeta DAQ a la entrada de control del Motor DC MOTOR DC IN. En este ejemplo se utiliza la salida analógica número cero (AO0) de la tarjeta instalada como Device 1.
 - b. Conectar la referencia GND de la tarjeta DAQ a la referencia GND del EPC. Todas las referencias GND de la tarjeta DAQ están conectadas internamente (en el modelo USB-6009 y similares); igualmente, todas las referencias GND del EPC están conectadas internamente.
 - c. Conectar la salida del encoder del Motor DC del EPC llamada ENCODER MDC a una entrada analógica de la tarjeta DAQ. En este ejemplo se utiliza la entrada analógica número cinco (AI5) en conexión tipo RSE.
 - d. Conectar la fuente de poder del EPC.

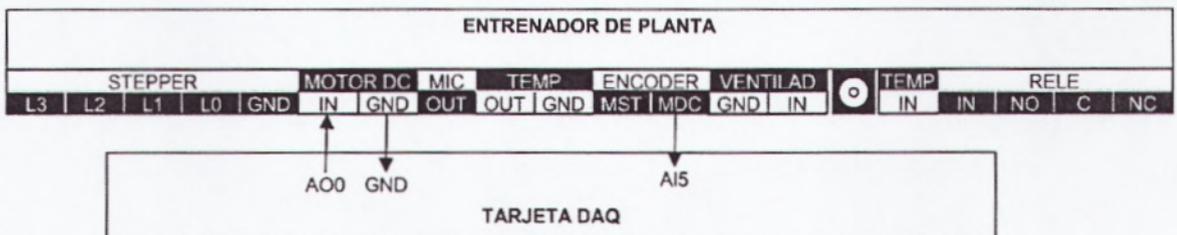


Figura4.20: Conexión Tarjeta DAQ Práctica M2

2. se realiza el programa medición analógica velocidad.vi. El panel frontal presenta los siguientes controles e indicadores:
- a. El control numérico tipo Slide Voltaje es el voltaje de salida de la tarjeta DAQ, que puede tomar valores entre 0 y 5 voltios.
 - b.El indicador gráfico tipo Chart Señales muestra un historial de los valores de voltaje y de velocidad que se han generado. La magnitud de estas señales están en Valores Por Unidad (VPU). Para el Voltaje el límite máximo es 5 Voltios. Para la Velocidad el límite máximo es 4500 RPM.
 - c. El indicador gráfico tipo Graph Señal Encoder muestra en forma de osciloscopio la señal de salida del Encoder que es tipo tren de pulsos. Obsérvese que mientras mayor es el Voltaje de control del Motor DC, mayor es la rapidez de giro del eje, y por lo tanto mayor es la frecuencia (menor es el período) de la señal del Encoder.
 - d. El indicador tipo Gauge Frecuencia muestra cuántos pulsos por segundo (Hz) tiene la Señal del Encoder, mientras que el indicador tipo Gauge Velocidad indica la velocidad del eje en revoluciones por minuto (RPM). El disco del encoder tiene 36 ranuras.
 - e. El botón de STOP detiene el programa. Es importante detener el programa utilizando este botón y no el botón de abortar de la barra de herramientas porque así se asegura que al parar el programa se detenga el motor, pues se envía un valor de 0V a la salida analógica.

3. Para esto se realiza los siguiente Pasos: (VerFigura4.21)

- a. En la pantalla del Panel Frontal se dibuja dos Indicador Grafico, para esto seleccionamos de la paleta de controles el menú Express, Graph Indicators, WaveForm Chart.
- b. Se dibuja un indicador numérico de velocidad, para esto seleccionamos de la paleta de controles el menú Modern, Numeric, Vertical Pointer Slide.
- c. Se dibuja un botón de Stop, esto lo realizamos al seleccionar de la paleta de controles el menú Modern, Boolean, Stop Button.
- d. Se dibuja un control numérico, Esto lo realizamos seleccionando de la paleta de controles el menú Modern, Numeric, Numeric Control.
- e. Se dibuja dos controles numéricos tipo Gauge, para esto seleccionamos de la paleta de controles el menú Modern, Numeric, Gauge.

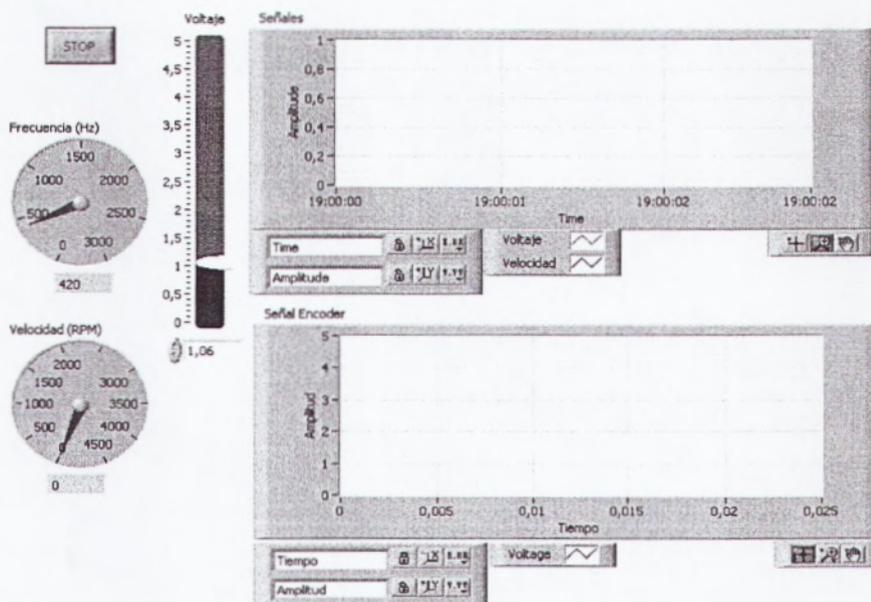


Figura 4.21: Panel Frontal Práctica M2

4. Se abre el diagrama de Bloques y se realiza lo siguiente:
 - a. Se aplica los literales a y b, del Ítem 3 de la Práctica M.
 - b. Selecciono en paleta de funciones el menú Programing, WaveForm, WeveForm Measure, Amplitud and Level Measurements.
 - c. Seleccione de la paleta de Funciones el menú Programing, Comparison, Select.
 - d. Seleccione de la paleta de funciones el menú Express, signal Manipulation, Marge Signal.
 - e. Realice las conexiones del diagrama de bloque como la grafica. (VerFigura4.22)

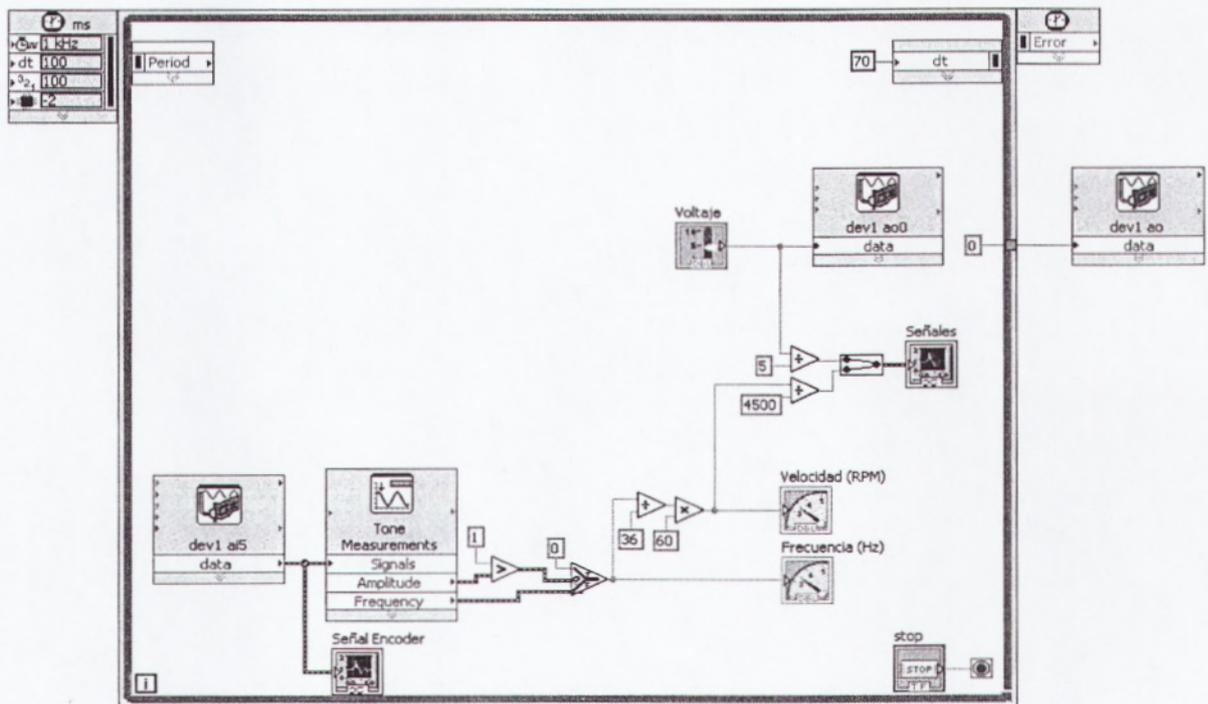


Figura 4.22: Diagrama de Bloque Práctica M2

5. Modificar el valor de Voltaje y observar cómo varía la velocidad. Cuando el voltaje está en su máximo valor (5 V), la velocidad toma su máximo valor (cerca de 4500 RPM).
6. Para observar (Ver Figura 4.23), la curva de arranque y parada del motor puede aplicarse sucesivamente valores de 0V y luego 5V al voltaje, y notar en la Gráfica Señales la forma en que la velocidad responde a esta señal de estímulo tipo escalón, típica de un motor DC.

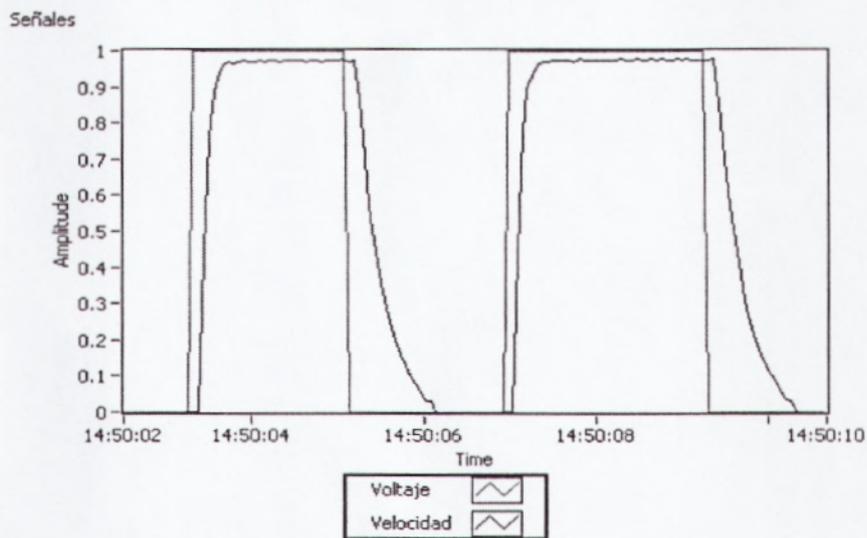


Figura 4.23: Curva de Arranque

7. Abrir el diagrama de bloques para analizarlo.
 - a. La señal adquirida del Encoder en forma analógica es procesada para obtener su frecuencia y su amplitud.
 - b. Si la amplitud es menor que 1 significa que el Motor DC no está girando, por lo tanto el encoder entregará una señal sin pulsos (y que puede estar en estado alto o bajo según donde haya parado el disco), y la frecuencia que se mide será la del ruido de la señal, mas no de la velocidad.

- c. Si la amplitud de la señal es mayor que 1 el motor estará girando y el Encoder entregará una señal tipo tren de pulsos. Entonces la frecuencia medida sí será la cantidad de pulsos por segundo.
 - d. Las operaciones siguientes son para transformar la frecuencia en revoluciones por minuto (RPM).
8. Cerrar el programa sin hacer cambios.

4.3.3 PRÁCTICA M3:SEÑAL DE ENCODER – FORMA DE CONTADOR DIGITAL

En esta práctica se adquirirá la señal del encoder para analizarla en forma de tren de pulsos, y se medirá la cantidad de pulsos por iteración para determinar la velocidad del motor.

1. Realizar las siguientes conexiones entre el EPC y la tarjeta DAQ(*VerFigura4.24*)
 - a. Conectar una salida analógica de la tarjeta DAQ a la entrada de control del Motor DC MOTOR DC IN. En este ejemplo se utiliza la salida analógica número cero (AO0) de la tarjeta instalada como Device 1.
 - b. Conectar la referencia GND de la tarjeta DAQ a la referencia GND del EPC. Todas las referencias GND de la tarjeta DAQ están conectadas internamente (en el modelo USB-6009 y similares); igualmente, todas las referencias GND del EPC están conectadas internamente.

- c. Conectar la salida del encoder del Motor DC del EPC llamada ENCODER MDC a la entrada de contadores de la tarjeta DAQ. En este ejemplo se utiliza el contador número cero, cuyo terminal en la tarjeta DAQ está identificado como PF10 (Programable Function input 0).
- d. Conectar la fuente de poder del EPC.

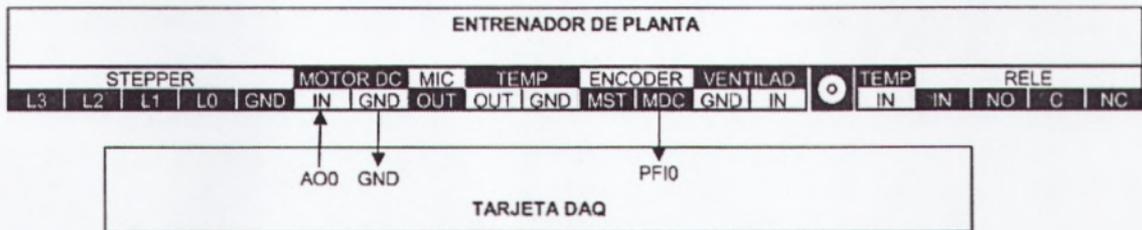


Figura4.24: Conexión Tarjeta DAQ Práctica M3

2. Realice el programa medición contador velocidad.vi. El panel frontal presenta los siguientes controles e indicadores:
 - a. El control numérico tipo Slide Voltaje es el voltaje de salida de la tarjeta DAQ, que puede tomar valores entre 0 y 5 voltios.
 - b. El indicador gráfico tipo Chart Señales muestra un historial de los valores de voltaje y de velocidad que se han generado. La magnitud de estas señales está en Valores Por Unidad (VPU). Para el Voltaje el límite máximo es 5 Voltios. Para la Velocidad el límite máximo es 4500 RPM.
 - c. El indicador tipo Gauge Velocidad indica la velocidad del eje en revoluciones por minuto (RPM). El disco del encoder tiene 36 ranuras.
 - d. El botón de STOP detiene el programa. Es importante detener el programa utilizando este botón y no el botón de abortar de la

barra de herramientas porque así se asegura que al parar el programa se detenga el motor, pues se envía un valor de 0V a laSalida analógica

3. Para esto realice los siguientes Pasos: (VerFigura 4.25)

- a. Se aplica los literales a, b, c y d del Ítem 3 de la práctica M2.
- b. Se dibuja dos controles numéricos tipo Gauge, para esto seleccionamos de la paleta de controles el menú Modern, Numeric, Gauge.

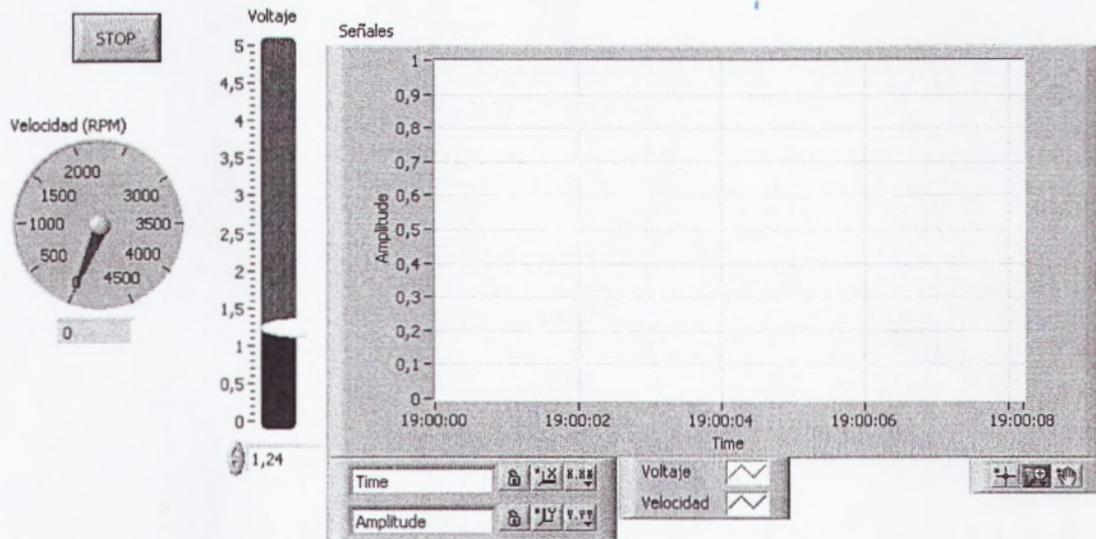


Figura 4.25: Panel Frontal Práctica M3

- 4. Modificar el valor de Voltaje de alimentación del Motor DC. Observar la forma en cómo varía la Velocidad al variar el Voltaje. Ésta es igual a la práctica anterior, excepto por la cantidad de ruido a diferentes velocidades. Este método es preferible cuando se tiene un encoder de alta cantidad de pulsos por revolución, pues el programa es más liviano porque no necesita adquisición de datos analógica sino solamente leer del registro del contador de la tarjeta DAQ.

5. Se abre el diagrama de Bloques y se realiza lo siguiente:
 - a. Se aplica los literales a y b, del Ítem 3 de la Práctica M2.
 - b. Seleccione de la paleta de funciones el menú Express, signal Manipulation, Marge Signal.
 - c. Seleccione Operadores numérico, en la paleta de funciones elija el menú Programing, Numeric, multiple-divide-subtract.
 - d. Realice las conexiones del diagrama de bloque como la grafica.
(VerFigura4.26)

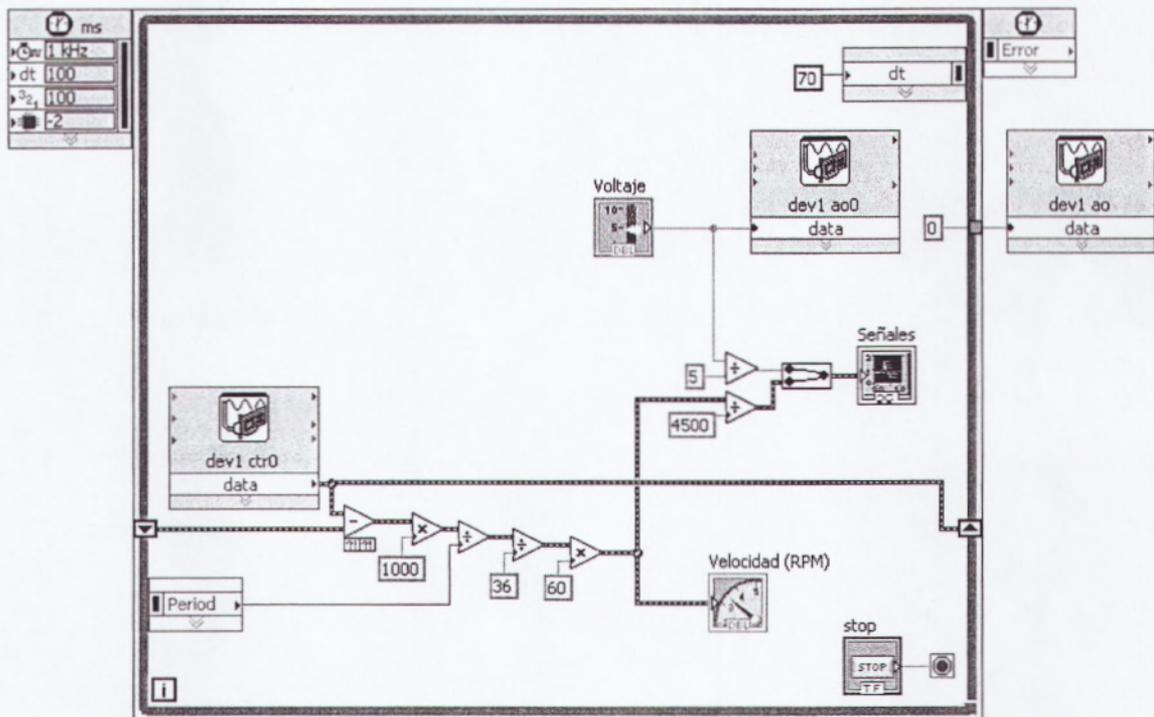


Figura 4.26:Diagrama de Bloque Práctica M3

6. Se analiza el diagrama de bloque:

- a. En este caso se ha adquirido la señal del Encoder por medio del contador de la tarjeta DAQ.
- b. Se resta el valor actual del contador con el valor en la iteración anterior para obtener el número de pulsos en una iteración.
- c. Los cálculos siguientes permiten obtener la velocidad en revoluciones por segundo

7. Cerrar el programa sin hacer cambios.

4.3.4 PRÁCTICA M4: CARACTERIZACIÓN DE MOTOR DC

En esta práctica se utilizarán como bases las prácticas anteriores para obtener una caracterización del Motor en cuanto a linealidad entre la variable de control (Voltaje DC) y la variable controlada (Velocidad). Si el sistema es lineal se puede aplicar un controlador PID. También se podrá observar los rangos en los cuales el sistema es lineal o tiene discontinuidades. Este análisis se hará de manera visual solamente, pero se puede ampliar el programa para mostrar una línea superpuesta al comportamiento del motor para obtener resultados matemáticos exactos.

1. Mantener las conexiones de la Práctica M2: Señal de Encoder – Forma Analógica.

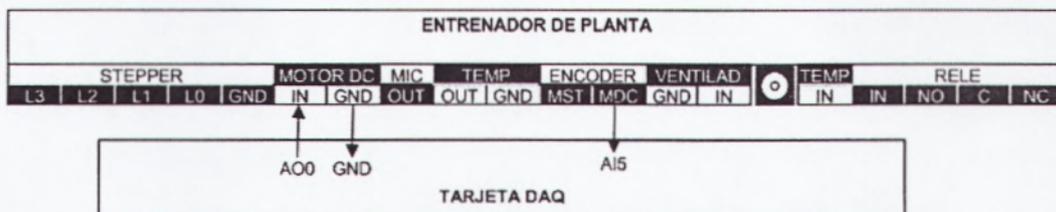


Figura4.27: Conexión Tarjeta DAQ Práctica M4

Realice un programa con nombre caracterización motor.vi. Este programa incrementa automáticamente el Voltaje de control del Motor DC de 0 a 5 voltios en pasos de 0.05 V cada 100ms. Al llegar a los 5V el programa se detiene automáticamente. El panel frontal tiene los siguientes objetos:

- a. El indicador numérico tipo Slide Voltaje muestra el voltaje enviado al Motor DC.
- b. El indicador numérico tipo Slide Velocidad muestra la velocidad del Motor DC en revoluciones por minuto (RPM).
- c. El gráfico tipo XY Voltaje Vs. Velocidad muestra el par de datos ordenados: en el eje X el Voltaje de control del Motor DC, y en el eje Y la Velocidad.
- e. El botón STOP permite detener el programa antes que llegue a los 5V

2. Para esto realice lo siguiente en el Panel Frontal:(VerFigura4.28)

- a. En la pantalla del Panel Frontal se dibuja un Indicador Grafico, para esto seleccionamos de la paleta de controles el menú Modern, Graph, WaveForm Chart.
- b. Se dibuja dos indicadores numérico de para velocidad y voltaje, para esto seleccionamos de la paleta de controles el menú Modern, Numeric, Vertical Pointer Slide.
- c. Se dibuja un botón de Stop, esto lo realizamos al seleccionar de la paleta de controles el menú Modern, Boolean, Stop Button.

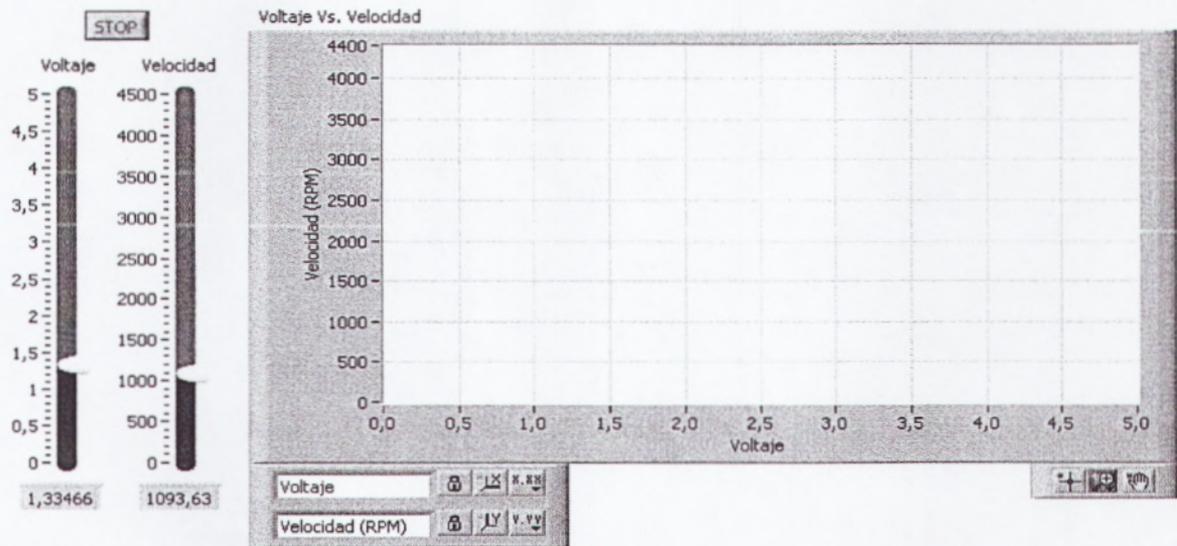


Figura4.28: Panel Frontal Práctica M4

3. Se Abre el diagrama de Bloques y se realiza lo siguiente:
 - a. Se aplica los literales a y b, del Ítem 3 de la Práctica M1.
 - b. Selecciono en paleta de funciones el menú Programing, WaveForm, WeveForm Measure, Amplitud and Level Measurements.
 - c. Seleccione de la paleta de Funciones el menú Programing, Comparison, Select.
 - d. Seleccione de la paleta de Funciones el menú Programing, Array, Build Array.
 - e. Realce las conexiones de la imagen.(VerFigura4.29)

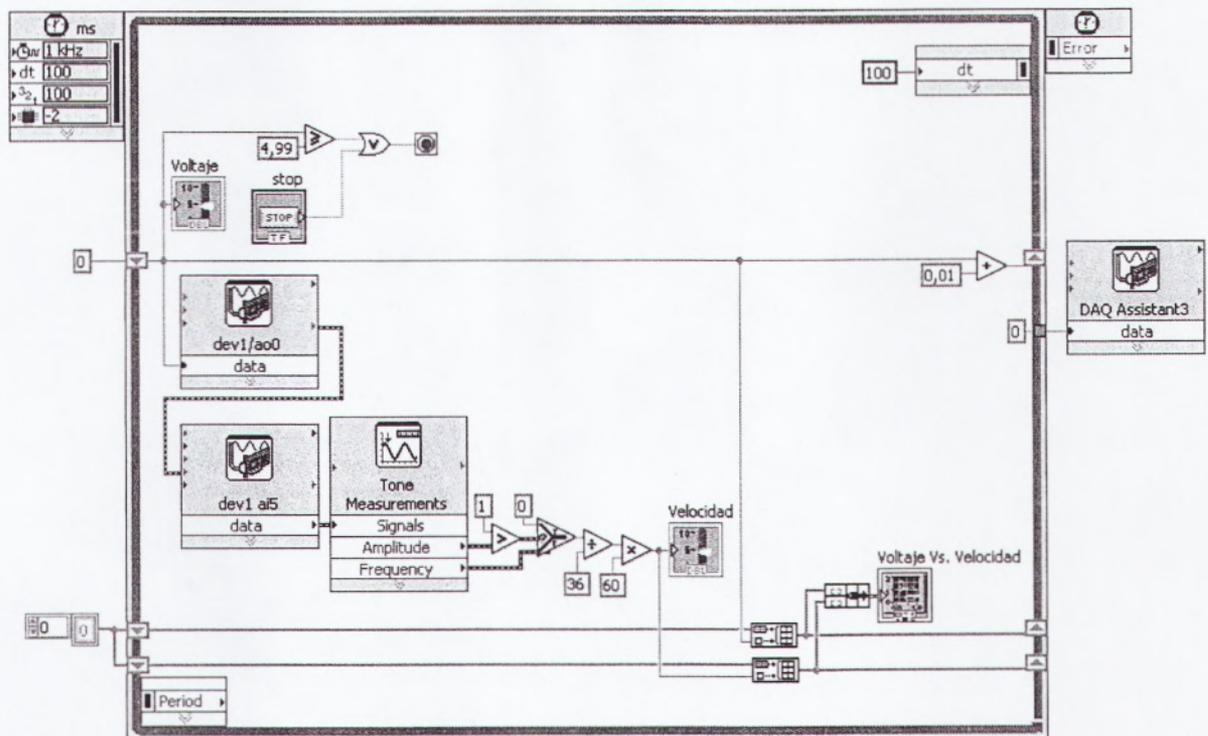


Figura4.29: Diagrama de Bloques Práctica M4

3. Correr el programa.

- a. Observar cómo automáticamente el voltaje sube progresivamente. (Ver Figura 4.30)
- b. De cero a aproximadamente 0.5 V, el voltaje no es suficiente para vencer el rozamiento e inercia del rotor, de modo que éste no rota y la velocidad es cero.
- c. Por sobre los 0.5 V aproximadamente, el rotor empieza a girar repentinamente presentando una discontinuidad. Es posible que existan pequeños pulsos de velocidad debido a que el motor presenta un ligero movimiento antes de arrancar definitivamente.

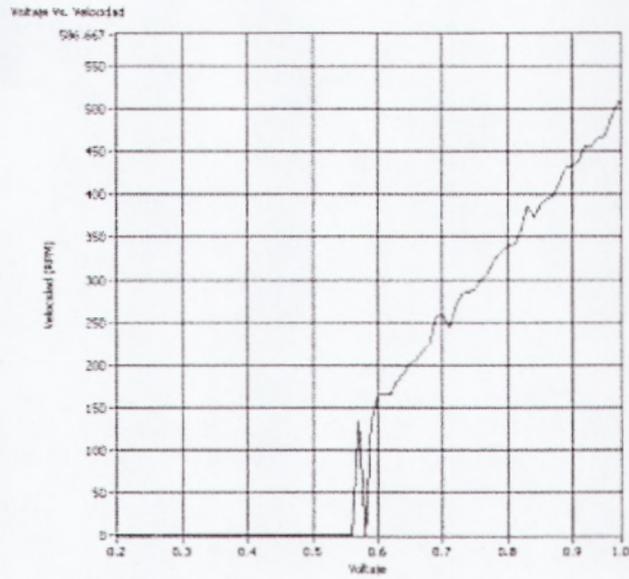


Figura4.30: Aumento Progresivo de Voltaje

- d. A partir de este punto la velocidad presenta un comportamiento lineal con respecto al voltaje. Por lo tanto se puede utilizar un controlador tipo PID en el rango lineal del motor; esto es, desde 0.6V hasta 5V. (VerFigura4.31)

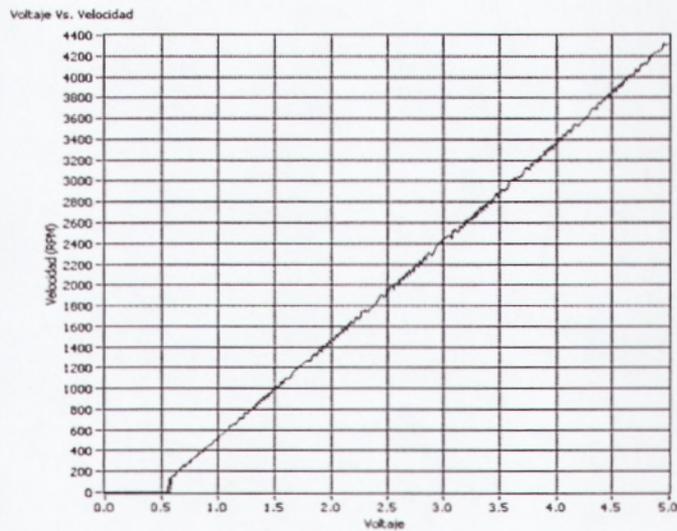


Figura4.31: Comportamiento de Velocidad

4. Abrir el diagrama de bloques para analizar el algoritmo.
 - a. La sección del primer Shift Register contando desde la parte superior del programa incrementa progresivamente el valor de Voltaje de salida al Motor DC desde cero hasta 5V en saltos de 0.01V.
 - b. La sección de adquisición de datos y determinación de la velocidad es similar a la utilizada en la Práctica M2: Señal de Encoder – Forma Analógica.
 - c. La sección del segundo y tercer Shift Register crean los arreglos de valores X (Voltaje) y Y (Velocidad) para ingresarlos al gráfico tipo XY Voltaje Vs. Velocidad.
5. Cerrar el programa sin hacer cambios.

3.3.5 PRÁCTICA M5:CONTROL PID DE VELOCIDAD DE MOTOR DC

En esta práctica se utilizarán como bases las prácticas anteriores para obtener un control PID de la velocidad del Motor DC. En muchos sistemas en los que es necesario controlar una variable no es posible modelar matemáticamente la planta debido a la falta de especificaciones. Tal es el caso del Motor DC del EPC. Puesto que no se tienen especificaciones del fabricante sobre la resistencia de armadura, inductancia, amortiguamiento por fricción viscosa, inercia del eje, etc. se deben utilizar métodos experimentales para obtener un conjunto de valores para las ganancias del controlador PID. En esta práctica se utilizará la segunda regla experimental de Ziegler-Nichols.

En prácticas más avanzadas puede utilizarse el Toolkit de Identificación de Sistemas de National Instruments para obtener una

descripción matemática del Motor DC a partir de señales de estímulo y respuesta.

1. Conectar el EPC a la tarjeta DAQ de la siguiente manera:
 - a. Conectar una salida analógica de la tarjeta DAQ a la entrada de control del Motor DC MOTOR DC IN. En este ejemplo se utiliza la salida analógica número cero (AO0) de la tarjeta instalada como Device 1.
 - b. Conectar la referencia GND de la tarjeta DAQ a la referencia GND del EPC. Todas las referencias GND de la tarjeta DAQ están conectadas internamente (en el modelo USB-6009 y similares); igualmente, todas las referencias GND del EPC están conectadas internamente.
 - c. Conectar la salida del encoder del Motor DC del EPC llamada ENCODER MDC a una entrada analógica de la tarjeta DAQ. En este ejemplo se utiliza la entrada analógica número cinco (AI5) en conexión tipo RSE.
 - d. Conectar la fuente de poder del EPC.

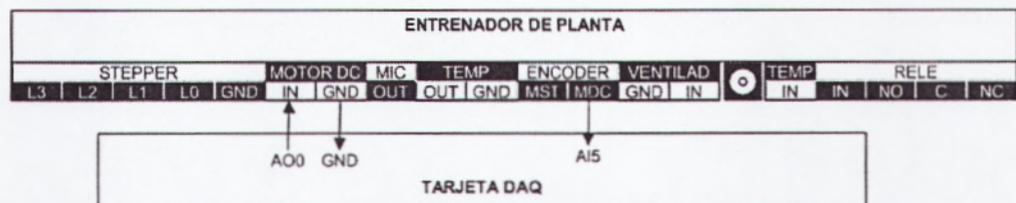


Figura4.32: Conexión Tarjeta DAQ Práctica M5

2. Realice el programa control PID velocidad.vi. El panel frontal presenta los siguientes controles e indicadores:
 - a. El usuario selecciona el valor de velocidad de consigna, es decir, el valor de velocidad al cual se necesita que llegue el Motor, mediante el control numérico tipo Slide Setpoint.
 - b. El indicador numérico tipo Slide Velocidad indica la velocidad a la que el Motor está girando.
 - c. El indicador numérico tipo Slide Voltaje indica el voltaje de salida de la tarjeta DAQ al Motor DC.
 - d. El clúster de controles numéricos Ganancias PID contiene los valores de constantes Proporcional K_c , Tiempo Integral T_i (en minutos), y el Tiempo Derivativo T_d (en minutos).
 - e. El indicador gráfico tipo Chart Señales muestra el valor de Setpoint (velocidad deseada) y el valor de Velocidad (velocidad real), ambas en RPM.
3. Para esto realice lo siguiente en el Panel Frontal:(VerFigura4.33)
 - a. En la pantalla del Panel Fontal se dibuja un Indicador Grafico, para esto seleccionamos de la paleta de controles el menú Modern, Graph, WaveForm Chart.
 - b. Se dibuja tres indicadores numérico de para velocidad, voltaje y RPM, para esto seleccionamos de la paleta de controles el menú Modern, Numeric, Vertical Pointer Slide.
 - c. Se dibuja un botón de Stop, esto lo realizamos al seleccionar de la paleta de controles el menú Modern, Boolean, Stop Button.

- d. Se dibuja también un grupo de controles numéricos para las ganancias Pid. Esto lo realizamos seleccionando de la paleta de controles el menú Modern, Numeric, Numeric Control.

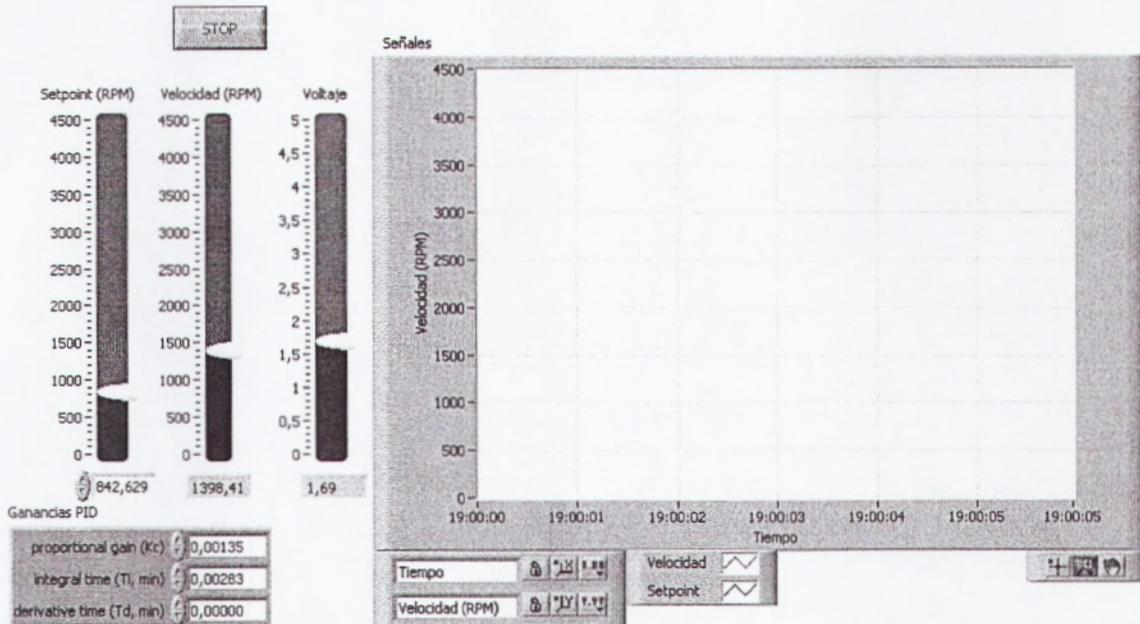


Figura4.33: Panel Frontal Práctica M5

4. Se abre el diagrama de Bloque y se realiza lo siguiente:
- Se aplica los literales a y b, del Ítem 3 de la Práctica M1.
 - Selecciono en paleta de funciones el menú Programing, WaveForm, WaveForm Measure, Amplitud and Level Measurements.
 - Seleccione de la paleta de Funciones el menú Programing, Comparison, Select.
 - Seleccione de la paleta de funciones el menú Express, signal Manipulation, Marge Signal.

e. Realice las conexiones de la imagen. (Ver Figura 4.34)

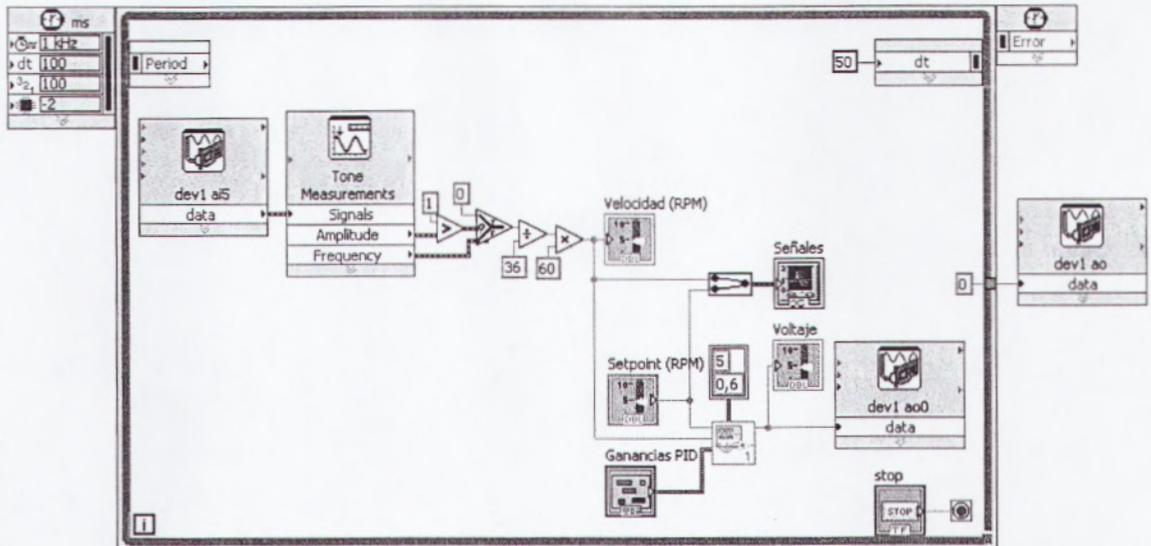


Figura 4.34: Diagrama de Bloque Práctica M5

5. Hallar un conjunto de valores para las Ganancias PID utilizando el segundo método de Ziegler-Nichols según los siguientes pasos:

- a. Asignar $K_c=0$, $T_i=999999$, $T_d=0$
- b. Cambiar los valores de Setpoint. Observar que el Voltaje no varía, pues no existe acción de control con $K_c=0$.
- c. Hallar un valor de ganancia proporcional K_c crítico, llamado K_{cr} , en el cual el voltaje de salida, y por lo tanto la velocidad, varíe de forma periódica en respuesta al escalón, y bajo el cual la salida sea estable en respuesta al escalón. Se recomienda empezar por la unidad, y a continuación ir en incrementos o decrementos de potencias de 10 hasta encontrar K_{cr} .

Para crear la función escalón se puede colocar el valor de Setpoint en 0, borrar la gráfica tipo Chart, correr el programa, e ingresar un valor de 3000 en Setpoint y observar la respuesta.

En el siguiente ejemplo se muestra un procedimiento sugerido.

NOTA IMPORTANTE: Los Motores de diferentes EPC podrían tener diferentes especificaciones. Por lo tanto, los valores del siguiente ejemplo pueden diferir de los valores experimentales obtenidos con un EPC diferente.

Asignar $K_c=1$. Se obtendrá una señal similar a la siguiente:

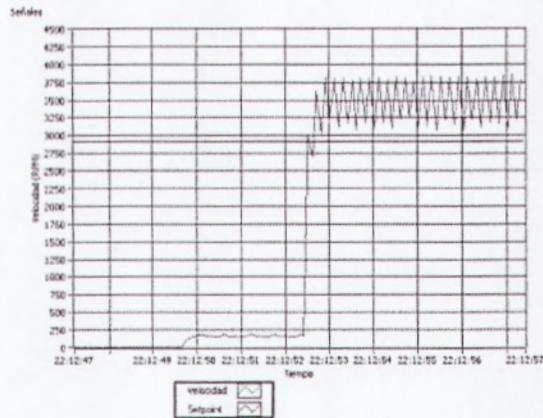


Figura4.35: $K_c=1$

Como se puede observar, la salida varía periódicamente, así que se debe decrecer en una potencia de 10 para confirmar si la señal aún varía periódicamente.

Asignar $K_c=0.1$. Se obtendrá una señal similar a la siguiente:

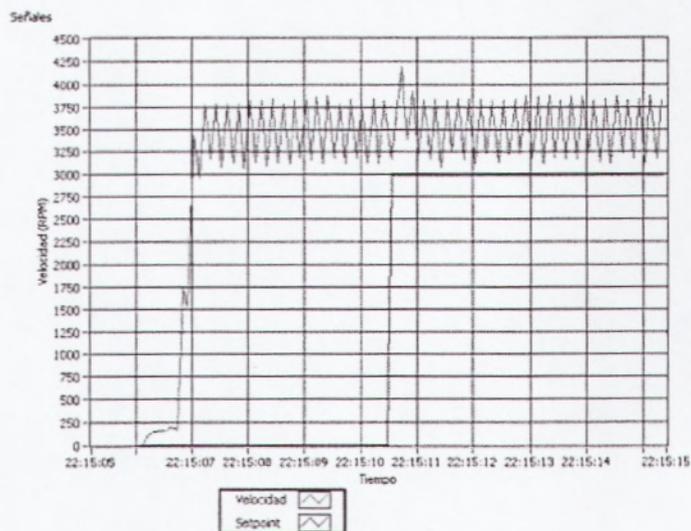


Figura 4.36: $K_c=0.1$

Como se puede observar, la señal aún varía periódicamente, así que debe decrecer en una potencia de 10 para confirmar si la señal aún vería periódicamente.

Asignar $K_c=0.01$. Se obtendrá una señal similar a la siguiente:

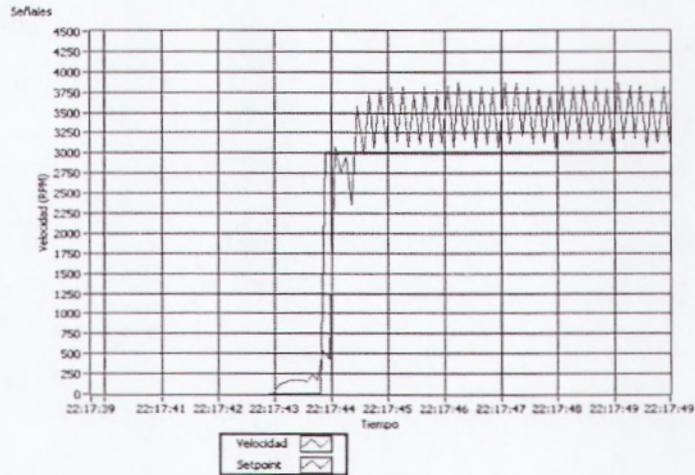


Figura 4.37: $K_c=0.01$

Como se puede observar, la salida aún varía periódicamente, así que debe decrecer en una potencia de 10 para confirmar si la señal aún vería periódicamente.

Asignar $K_c=0.001$. Se obtendrá una señal similar a la siguiente:

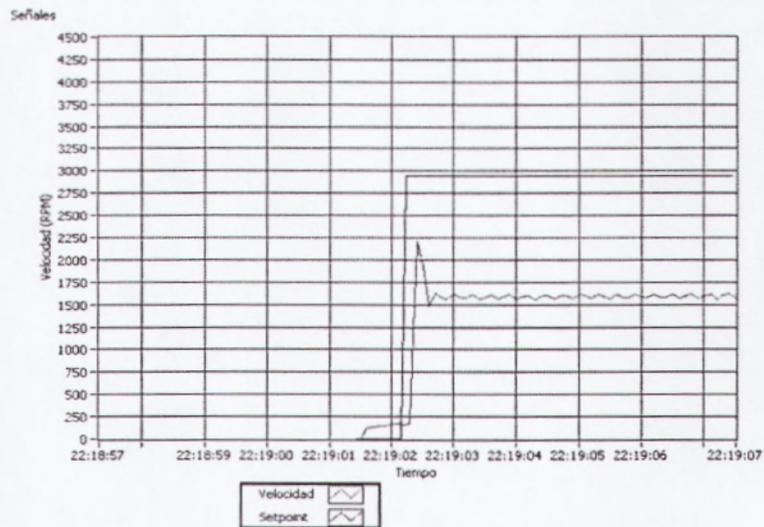


Figura 4.38: $K_c=0.001$

Como se puede observar, la salida ya no oscila. Por lo tanto el valor de K_c crítico está entre 0.001 y 0.01.

Asignar $K_c=0.005$. Se obtendrá una señal similar a la siguiente:

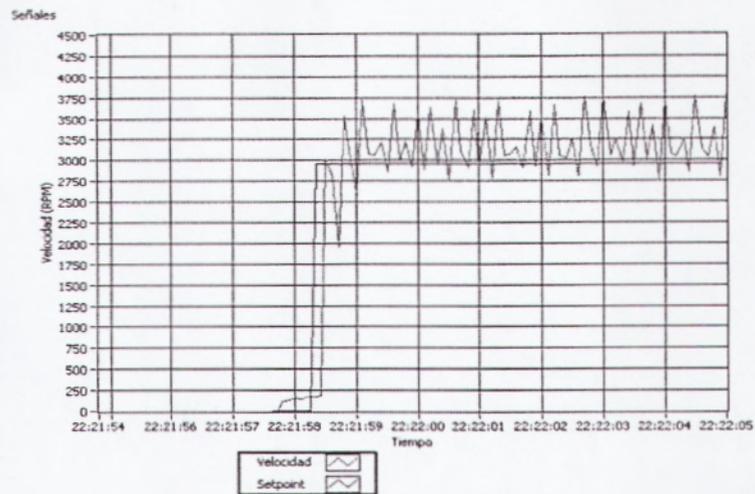


Figura 4.39: $K_c=0.005$

Puesto que la señal oscila, el valor de K_c crítico está entre 0.001 y 0.005. Asignar $K_c=0.0025$. Se obtendrá una señal similar a la siguiente:

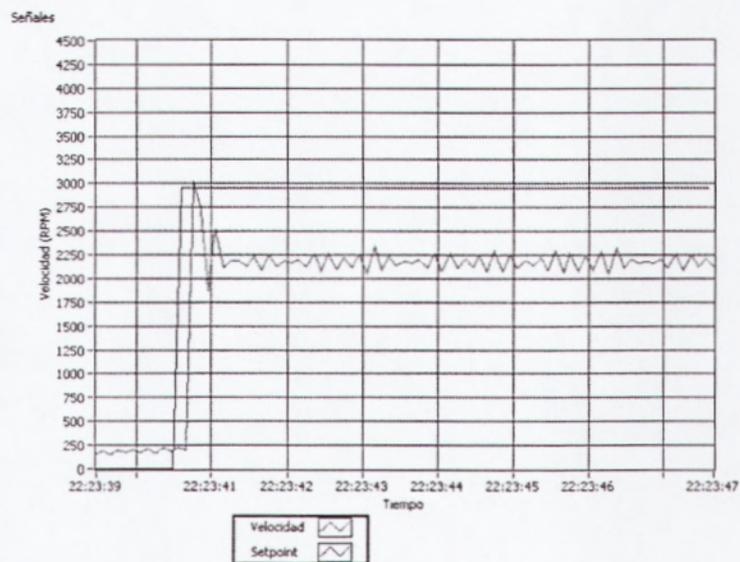


Figura 4.40: $K_c= 0.0025$

Se puede observar que la salida oscila, se estabiliza, vuelve a oscilar, y así sucesivamente. Puede intentarse un nuevo valor de K_c para confirmar una oscilación más evidente.

Asignar $K_c=0.003$. Se obtendrá una señal similar a la siguiente:

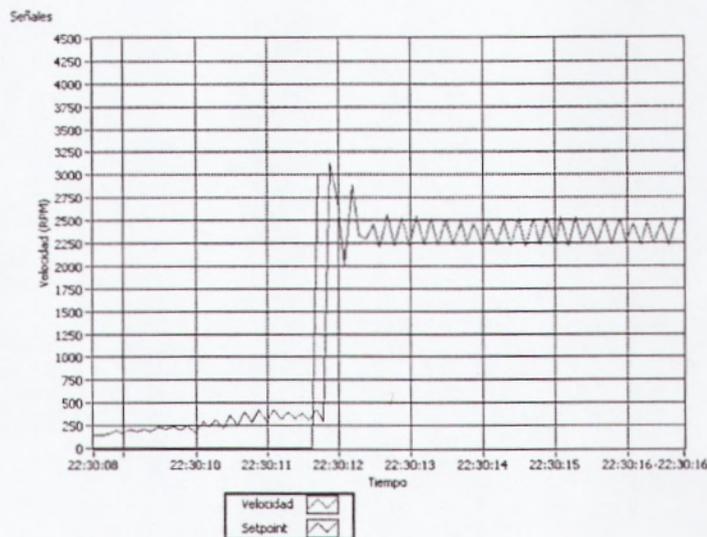


Figura 4.41: $K_c= 0.003$

Se puede observar que la señal varía periódicamente, así que podría tomarse a éste como el valor de K_c Crítico; así:

$$K_{cr}=0.003$$

A continuación se mide el período crítico. Para ello podemos observar la frecuencia crítica de oscilación de la salida F_{cr} .

El período crítico P_{cr} será el inverso de la frecuencia crítica F_{cr} . En este caso se tienen aproximadamente 4.9 ciclos por segundo. Las herramientas de Zoom del gráfico Señales pueden ser de mucha utilidad para medir esta frecuencia, tal como se muestra a continuación.



Figura 4.42: $F_{cr} = 4.9 \text{ Hz}$ $P_{cr} = 1/4.9 = 0.2 \text{ segundos}$

Aplicando la tabla recomendada por el segundo método de sintonía de Ziegler-Nichols se obtienen los siguientes valores (notar que el método proporciona T_i y T_d en segundos, mientras que el controlador PID de LabVIEW especifica sus ganancias T_i y T_d en minutos).

Tipo de Controlador	K_c	T_i	T_d
P	$0.5 K_{cr}$	inf	0
PI	$0.45 K_{cr}$	$1/1.2 P_{cr}$	0
PID	$0.6 K_{cr}$	$0.5 P_{cr}$	$0.125 P_{cr}$

VALORES MEDIDOS EXPERIMENTALMENTE		
K_{cr}	0.003	
P_{cr}	0.20	seg
F_{cr}	4.90	Hz

Tipo de Control	K_c	T_i (seg)	T_d (seg)	T_i (min)	T_d (min)
P	0.0015	inf	0	inf	0
PI	0.00135	0.1701	0	0.00283	0
PID	0.0018	0.10	0.03	0.00170	0.00043

Observando la respuesta al escalón podemos considerar al Motor DC del EPC como un sistema de primer orden. Por lo tanto podemos aplicar un controlador PI. Así, aplicando un valor de $K_c=0.00135$, $T_i(\text{min})=0.00283$, $T_d(\text{min})=0$, se obtiene la siguiente señal de respuesta al escalón.

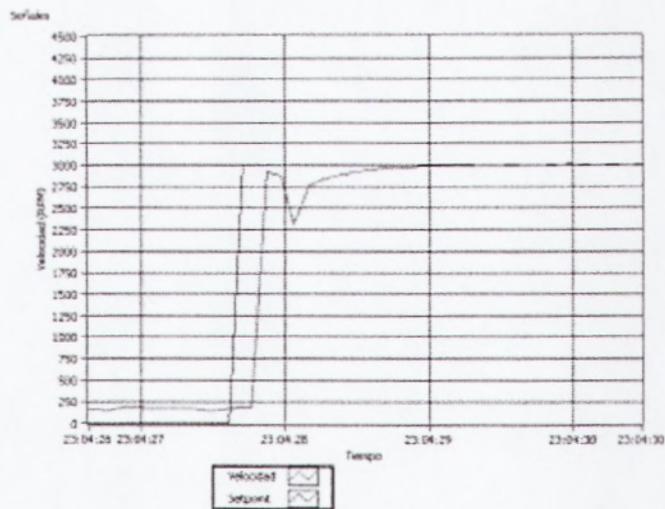


Figura 4.43: $K_c= 0.00135$

Enviando diferentes valores de escalón pueden observarse que la señal de respuesta sigue correctamente al Setpoint.

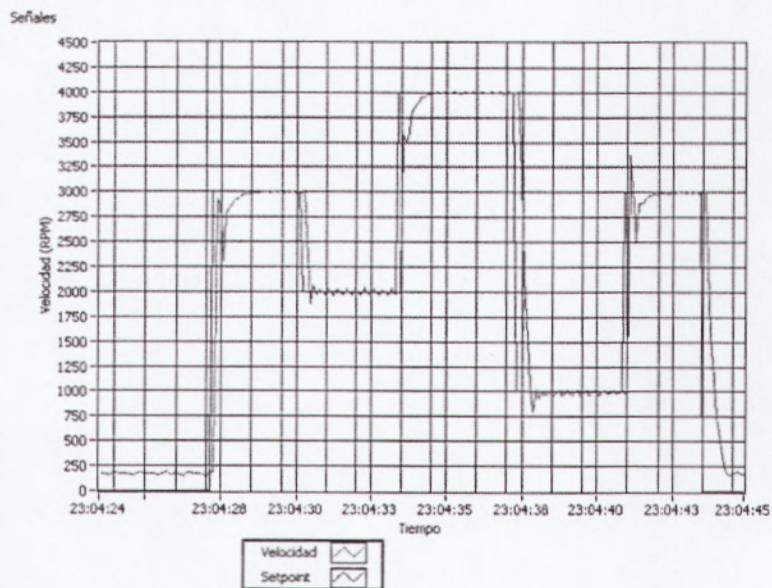


Figura 4.44: Señal de Respuesta

Se puede cambiar los valores de K_c y T_d para observar diferentes tipos de respuestas. Por ejemplo, incrementando $K_c = 0.00210$ y estimulando el sistema con un escalón de 0 a 2000 RPM, se obtiene la siguiente respuesta subamortiguada.

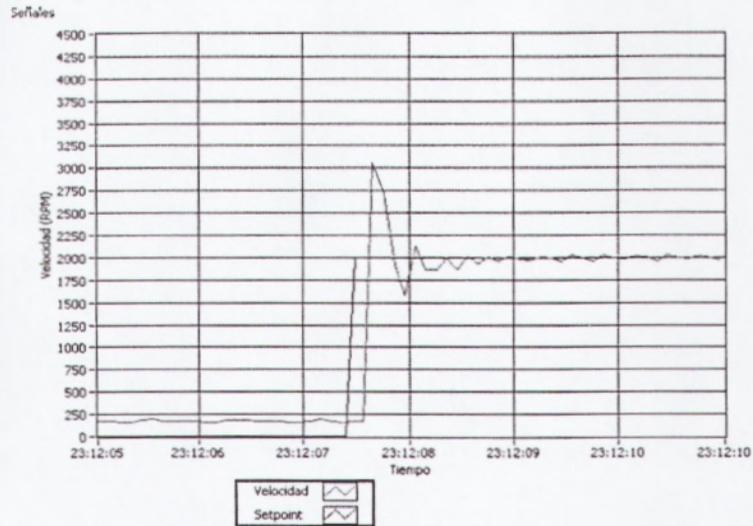


Figura 4.45: $K_c=0.00210$

Por otro lado, reduciendo $K_c=0.0005$ y estimulando el sistema con un escalón de 0 a 2000 RPM, se obtiene la siguiente respuesta sobreamortiguada.

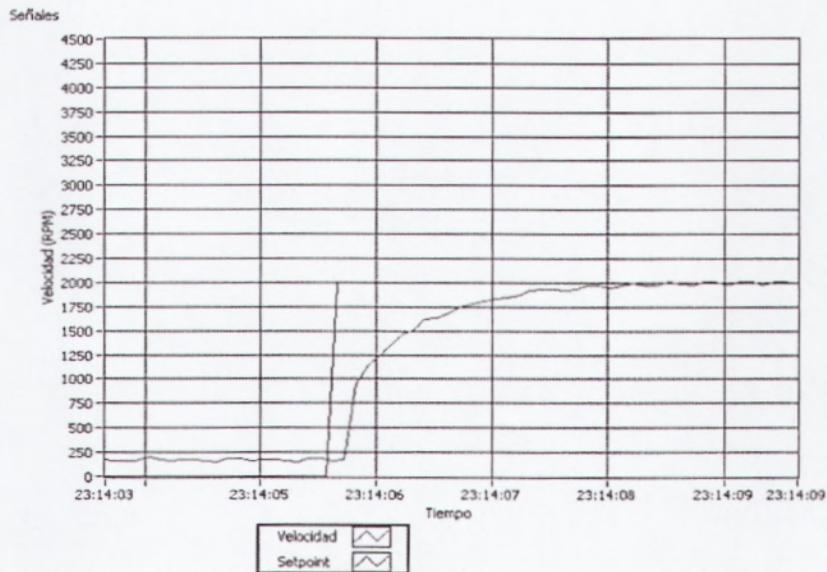


Figura 4.46: $K_c=0.0005$

4. Abrir el diagrama de bloques para analizar el algoritmo.

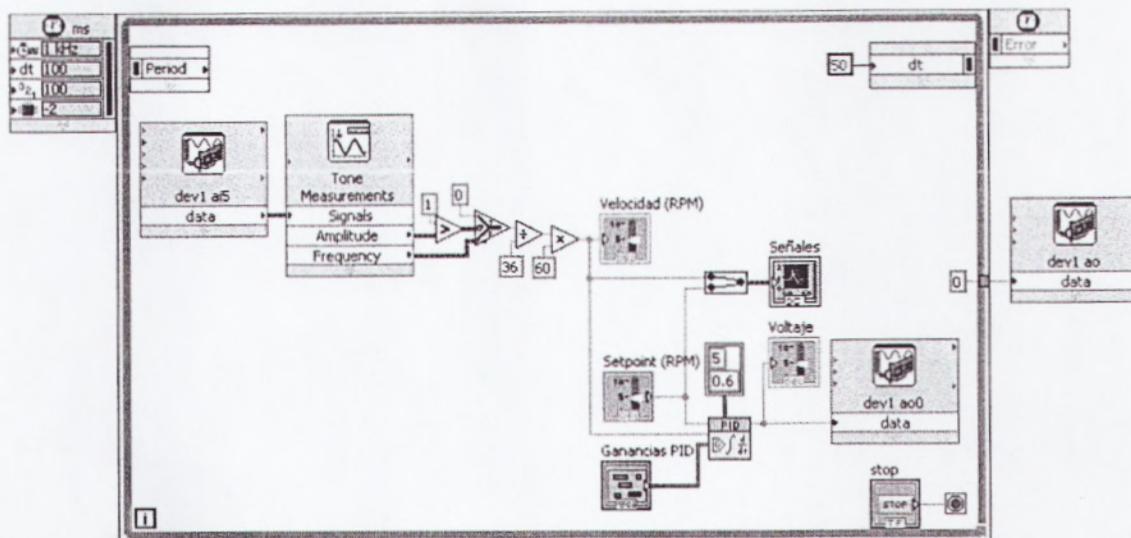


Figura4.47: Diagrama de Bloque Práctica M5

- a. La sección de adquisición de datos y medición de velocidad analizando la señal del Encoder como analógica es similar al utilizado en la práctica M2.
- b. En el diagrama de bloques puede observarse el ícono de PID, el cual como entradas se encuentran el Setpoint, la variable del proceso Velocidad, y las Ganancias PID. Además se especifica el rango de salida. Puesto que el sistema presentaba una discontinuidad en 0.5 en la cual la velocidad aumentaba en forma de escalón, se ha tomado el rango mínimo de salida como 0.6 V. La salida es el voltaje, el cual se envía a la salida analógica de la tarjeta DAQ que es la que alimenta al Motor DC.

5. Cerrar el programa sin hacer cambios.

3.4 PRÁCTICAS DE CONTROL DE POSICIÓN

En estas prácticas se utilizarán cuatro líneas de un puerto digital la tarjeta DAQ para comandar las bobinas del Stepper de modo que se pueda configurar la secuencia de energizado. El Stepper tiene un giro de aproximadamente 7.5° por cada paso. Por lo tanto, necesitará 48 pasos para girar una revolución completa de 360° .

Las bobinas del Stepper funcionan con lógica directa; es decir, cuando se envía una señal TRUE a una línea del Stepper, la bobina correspondiente se activa.

Si al conectar la tarjeta DAQ se enciende uno o más de los LEDs de Líneas del Stepper (para ubicar estos LEDs referirse al Capítulo I – Diagrama Esquemático y Conexiones Básicas) , desconectar inmediatamente la Fuente de Poder de 12VDC, y enviar un FALSE a todas las líneas digitales de la tarjeta DAQ desde el utilitario de la tarjeta DAQ Measurement and Automation Explorer, tal como muestra la figura (en este caso se asume que se han conectado las cuatro líneas del puerto P1 de la tarjeta DAQ a las líneas del Stepper). Esto pondrá en FALSE a todas las líneas desenergizando todas las bobinas del Stepper.

NOTA IMPORTANTE: *Cuando se activan las bobinas del Stepper, los LEDs de Líneas de Stepper se encienden. Si uno o más de estos LEDs se encienden al conectar la tarjeta DAQ, las bobinas estarán energizadas y el Stepper se calentará considerablemente si no se desactivan dentro de pocos minutos, pudiendo provocar daños al equipo, e incluso leves quemaduras en la piel.*

4.4.1 PRÁCTICA S1: SECUENCIA DE ENCENDIDO MANUAL

En esta práctica se utilizarán cuatro controles boléanos para encender manualmente cada las bobinas, del Stepper mediante cuatro líneas digitales de la tarjeta DAQ. En estos ejemplos se utilizan las cuatro primeras líneas digitales del puerto cero de la tarjeta DAQ (P1.0, P1.1, P1.2, y P1.3).

1. Realizar las siguientes conexiones entre el EPC y la tarjeta DAQ:
 - a. Conectar las líneas L0, L1, L2 y L3 del Stepper en el EPC a las líneas P1.0, P1.1, P1.2, y P1.3 de la tarjeta DAQ respectivamente. Recordar que si en este punto una o más líneas del Stepper se encienden indicarán que las bobinas están activadas, provocando calentamiento y posibles daños al equipo si esta situación se mantiene durante algunos minutos.
 - b. Conectar la referencia GND de la tarjeta DAQ a la referencia GND del EPC. Todas las referencias GND de la tarjeta DAQ están conectadas internamente (en el modelo USB-6009 y similares); igualmente, todas las referencias GND del EPC están conectadas internamente.
 - c. Conectar la fuente de poder del EPC.

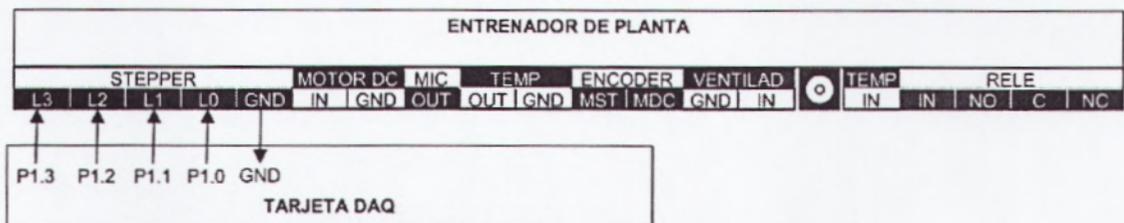


Figura 4.48: Conexión Tarjeta DAQ Práctica S1

2. Realice el programa con nombre secuencia manual.vi. El panel frontal presenta los siguientes objetos:
 - a. Un arreglo de controles booleanos tipo botón Líneas envía, línea por línea, señales de TRUE o FALSE al Stepper. Cuando estas líneas se activan o desactivan, se puede observar el estado en los Leds de Líneas de Stepper.
 - b. Un botón de STOP que detiene el programa. Es importante detener el programa utilizando este botón y no el botón de abortar de la barra de herramientas porque así se asegura que al parar el programa se desactiven todas las líneas del Stepper, pues se envía un valor de FALSE a todas las líneas digitales.
3. Para esto se realiza lo siguiente en el Panel Frontal.
 - a. Seleccione de la paleta de controles el menú modern, Boolean, y escoja cuatro push Button.
 - b. Seleccione de la paleta de controles el menú modern, Boolean, Stop Button.

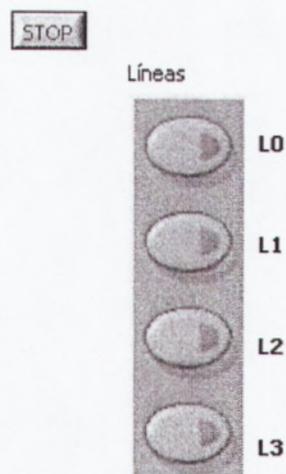


Figura4.49: Panel Frontal Práctica S1.

4. Abra el diagrama de bloque y realice lo siguiente:
- Seleccione una entrada mediante un DAQ Assistant, elija de la paleta de funciones el menú Express, Input, DAQ Assistant y configúrelo de la siguiente manera:
 1. Seleccione Acquire Signals
 2. Seleccione Analog Input
 3. Seleccione Voltaje
 4. Utilice el dispositivo USB-6009
 5. Seleccione canal analógico N° 5
 6. Configure Modo de conexión RSE (referenciado sencillo)
 7. Seleccione Modo de muestreo en 1 Sample On Demand
 8. De click en OK.
 - Seleccione de la paleta de funciones el menú programing, Timing, Wait Until Next ms Multiple.
 - Realizar las conexiones de la grafica.(VerFigura4.50)

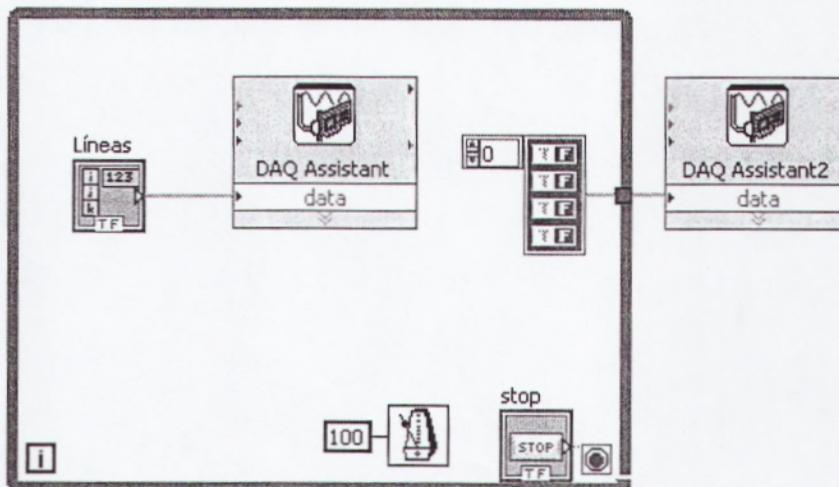


Figura 4.50: Diagrama de Bloques de la Práctica S1

5. Correr el Programa:
 - a. Notar la posición actual del Stepper. Si se desea se puede remover la tapa de acrílico transparente para colocar manualmente el disco en una posición referencial clara, por ejemplo apuntando al Encoder.
 - b. Activar solamente la línea L0 y observar el movimiento del disco.
 - c. Desactivar la línea L0 y activar la línea L1. El disco se habrá movido 7.5° .
 - d. Desactivar la línea L1 y activar la línea L2. Nuevamente el disco se habrá movido 7.5° , un total de 15° en los dos pasos.
 - e. Calcular cuántos pasos deben darse para completar 45° de giro y repetir la secuencia anterior hasta completarlo. Cuando se llega a activar la línea L3, para dar el siguiente paso se debe desactivar la línea L3 y activar la L0, comenzando nuevamente la secuencia.
 - f. Para que el disco gire en sentido contrario se requiere enviar la secuencia en sentido inverso.
6. Parar el programa presionando el botón de STOP.
7. Cerrar el programa sin guardar los cambios.

4.4.2 PRÁCTICA S2: CONTROL DE POSICIÓN

En esta práctica se utiliza un control numérico para especificar la posición angular a la que debe apuntar el disco del Stepper. Dada las especificaciones de este equipo, la posición final tendrá una precisión de 7.5° debido a la resolución del paso.

1. Realizar las siguientes conexiones entre el EPC y la tarjeta DAQ:
 - a. Conectar las líneas L0, L1, L2 y L3 del Stepper en el EPC a las líneas P1.0, P1.1, P1.2, y P1.3 de la tarjeta DAQ respectivamente. Recordar que si en este punto una o más líneas del Stepper se encienden indicarán que las bobinas están activadas, provocando calentamiento y posibles daños al equipo si esta situación se mantiene durante algunos minutos.
 - b. Conectar la referencia GND de la tarjeta DAQ a la referencia GND del EPC. Todas las referencias GND de la tarjeta DAQ están conectadas internamente (en el modelo USB-6009 y similares); igualmente, todas las referencias GND del EPC están conectadas internamente. En este ejemplo se utiliza el GND de la sección de temperatura.
 - c. Conectar la fuente de poder del EPC.

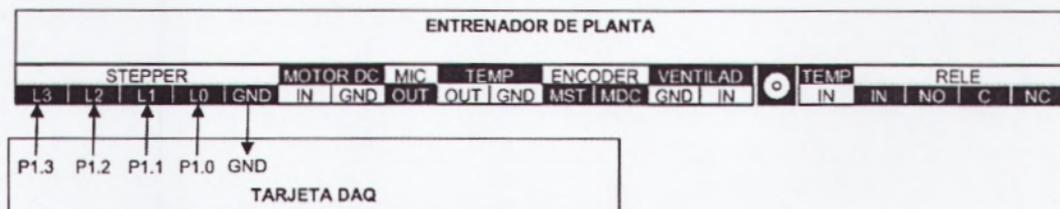


Figura4.51: Conexión tarjeta DAQ Práctica S2

2. Realice un programa con el nombre control posición.vi. El panel frontal presenta los siguientes objetos:
 - a. Un control numérico tipo Slide Posición Deseada (grad) para especificar la posición angular en grados a la cual debe apuntar el disco del Stepper. Incluye un control tipo numérico para colocar valores exactos.
 - b. Un control numérico Tiempo Entre Pasos (ms) para especificar cuántos milisegundos debe esperar el programa antes de realizar el siguiente paso.
 - c. Un indicador numérico tipo Slide Posición Actual (grad) que especifica la posición angular en grados a la que está apuntando en determinado momento el disco del Stepper.
 - d. Un botón de STOP que detiene el programa. Es importante detener el programa utilizando este botón y no el botón de abortar de la barra de herramientas porque así se asegura que al parar el programa se desactiven todas las líneas del Stepper, pues se envía un valor de FALSE a todas las líneas digitales.

3. En el panel Frontal se realiza los siguientes pasos:(VerFigura4.52)
 - a. Seleccione de la paleta de controles el menú modern, Boolean, Stop Button.
 - b. Seleccione dos indicadores numérico para la posición deseada y posición actual, para esto se elige de la paleta de controles el menú Modern, Numeric, horizontal pointer slide.

- c. Seleccione dos indicadores numérico para el tiempo entre pasos, para esto se elije de la paleta de controles el menú Modern, Numeric, Numeric Control.

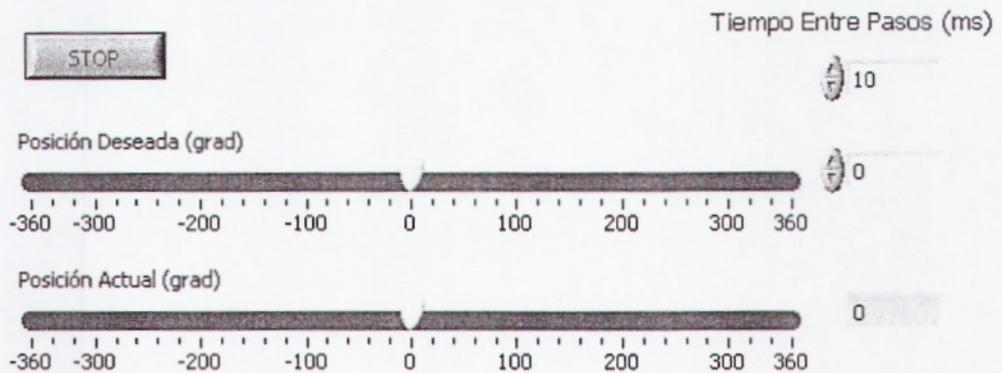


Figura4.52: Panel Frontal Práctica S2

4. Se abre el diagrama de bloques y se realiza lo siguiente:
- Se aplica el literal **a** del ítem 4 de la práctica S1.
 - Seleccione de la paleta de funciones el menú programming, Structures, timed Structures, Time Loop.
 - Seleccione de la paleta de funciones el menú programming, Structures, y escoja dos Case Structure.
 - Seleccione de la paleta de funciones, Programming, Boolean, Boolean array to number.
 - Seleccione de la paleta de Funciones el menú Programming, Comparison, (Select, less than 0?, greater than 0?, Equal?)

- f. Seleccione de la paleta de Funciones el menú Programing, array, buid array.
- g. Seleccione de la paleta de Funciones el menú Programing, array, index array.
- h. Seleccione de la paleta de Funciones el menú Programing, numeric, (Round to nearest, multiply, divide, subtract).
- i. Se realiza las conexiones del la grafica.(VerFigura4.53)

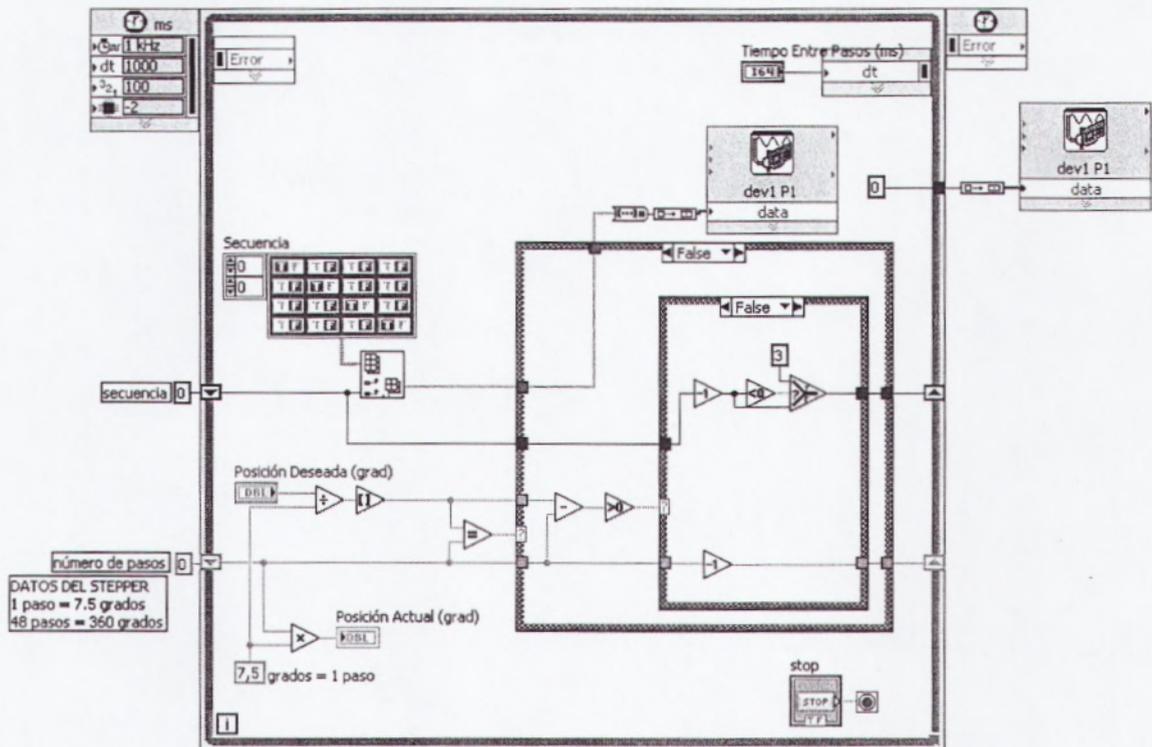


Figura4.53: Diagrama de Bloque Práctica S2

5. Correr el Programa.

- a. Notar la posición actual del Stepper. Si se desea se puede remover la tapa de acrílico transparente para colocar manualmente el disco en una posición referencial clara, por ejemplo apuntando al Encoder.

- b. Seleccionar un Tiempo Entre Pasos de 100 ms.

- c. Introducir un valor de Posición Deseada de 180° e inmediatamente notar el movimiento del disco del Stepper y del indicador numérico Posición Actual. Luego de cada paso el programa esperará 100 ms antes de dar el siguiente. También puede notarse la secuencia de encendido de las líneas observando los LEDs de Líneas de Stepper en la tarjeta electrónica del EPC.

- d. Utilizar el desplazador del control tipo Slide Posición Deseada para cambiar la posición utilizando el ratón. Notar el movimiento del disco del Stepper, así como del indicador Posición Actual, y el encendido de los LEDs de Líneas del Stepper.

- e. Cambiar el Tiempo Entre Pasos a 10 ms y repetir los pasos c y d. Introducir diferentes valores en Posición Deseada, como por ejemplo un valor de 3600° que equivale a 10 revoluciones completas. Comprobar la precisión del movimiento.

- f. Cambiar el Tiempo Entre Pasos a 0 ms y repetir los pasos c y d. Introducir nuevamente diferentes valores en Posición Deseada, como por ejemplo un valor de 3600° que equivale a 10 revoluciones completas. Comprobar la precisión del movimiento. Notar cómo eventualmente el Stepper puede saltarse pasos debido a que el programa corre más rápido de lo que el Stepper puede girar.

6. Cerrar el Programa

3.5 PRÁCTICAS DE ANALISIS DE SONIDO

En estas prácticas se asume que se utiliza una tarjeta DAQ genérica como la USB-6009, y el objetivo es comprender las bases de este tipo de análisis. Sin embargo, para aplicaciones de alto desempeño es necesario utilizar una tarjeta de adquisición de datos específica para análisis dinámico de señales (DSA).

Las principales características de este tipo de tarjetas son:

1. Alta resolución, necesaria para detectar armónicos de alta y baja magnitud.
2. Filtros Anti-Alias para eliminar el Alias debido al muestreo.
3. Alta resolución, necesaria para detectar armónicos de alta y baja magnitud.
4. Muestreo simultáneo para evitar el desfasaje en adquisición de datos en varios canales.
5. Acoplamiento AC/DC para eliminar las componentes de corriente continua Este tipo de análisis es similar al que se requiere en aplicaciones de medición de calidad de energía, vibración, telecomunicaciones, y aplicaciones DSP en general.

4.5.1 PRÁCTICA D1: MEDICIÓN DE ARMONICOS

En esta práctica se utilizará una entrada analógica de alta velocidad para adquirir la señal del micrófono, observar su forma de onda, realizar un análisis de armónicos, y medir las frecuencias fundamentales.

Se recomienda utilizar un teléfono para, presionando los botones del teclado numérico, generar sonido y analizar su contenido de frecuencia.

1. Realizar las siguientes conexiones entre el EPC y la tarjeta DAQ:

- a. Conectar la salida del micrófono MIC OUT a una entrada analógica de la tarjeta DAQ. En estas prácticas se utiliza la entrada analógica número cinco AI5 en conexión RSE.
- b. Conectar la referencia GND de la tarjeta DAQ a la referencia GND del EPC. Todas las referencias GND de la tarjeta DAQ están conectadas internamente (en el modelo USB-6009 y similares); igualmente, todas las referencias GND del EPC están conectadas internamente.
- c. Conectar la fuente de poder del EPC.

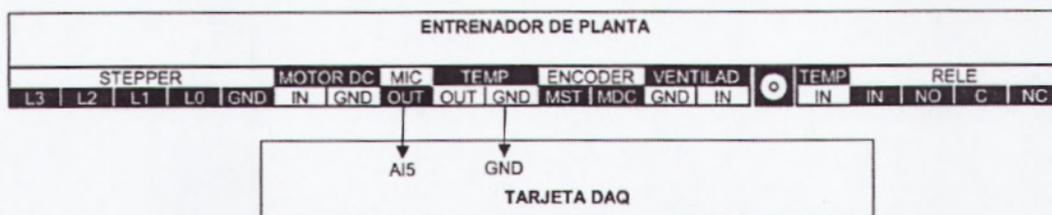


Figura4.54: Conexión Tarjeta DAQ Práctica DSA1