

ESCUELA POLITÉCNICA DEL EJÉRCITO



DEPARTAMENTO DE ELÉCTRICA Y ELECTRÓNICA

**DISEÑO E IMPLEMENTACIÓN DE UN CONTROLADOR DE
POSICIÓN DE PEDALES DE AUTOMÓVIL PARA PERSONAS
DISCAPACITADAS.**

AUTOR

DAVID PATRICIO PAREDES FLOR

Tesis presentada como requisito previo a la obtención del grado de:

**INGENIERO EN ELECTRÓNICA, AUTOMATIZACIÓN Y
CONTROL**

SANGOLQUÍ – ECUADOR

JULIO 2013

Declaración de Responsabilidad

ESCUELA POLITÉCNICA DEL EJÉRCITO

INGENIERÍA ELECTRÓNICA, AUTOMATIZACIÓN Y CONTROL

DECLARACIÓN DE RESPONSABILIDAD

David Patricio Paredes Flor

DECLARO QUE:

El proyecto de grado denominado “Diseño e implementación de un controlador de posición de pedales de automóvil para personas discapacitadas”, ha sido desarrollado con base a una investigación exhaustiva, respetando derechos intelectuales de terceros, conforme las citas que constan el pie de las páginas correspondiente, cuyas fuentes se incorporan en la bibliografía.

Consecuentemente este trabajo es mi autoría.

En virtud de esta declaración, me responsabilizo del contenido, veracidad y alcance científico del proyecto de grado en mención.

Sangolquí, 16 de Julio del 2013

Paredes Flor David Patricio

Certificado de tutoría

ESCUELA POLITÉCNICA DEL EJÉRCITO

INGENIERÍA ELECTRÓNICA, AUTOMATIZACIÓN Y CONTROL

CERTIFICADO

Ing. Edgar Tipán

Ing. Rodolfo Gordillo

CERTIFICAN

Que el trabajo titulado “Diseño e implementación de un controlador de posición de pedales de automóvil para personas discapacitadas” realizado por David Patricio Paredes Flor, ha sido guiado y revisado periódicamente y cumple normas estatutarias establecidas por la ESPE, en el Reglamento de Estudiantes de la Escuela Politécnica del Ejército.

Debido a que se trata de un trabajo innovador recomiendan su publicación.

El mencionado trabajo consta de un documento empastado y un disco compacto el cual contiene los archivos en formato portátil de Acrobat (pdf). Autorizan a David Patricio Paredes Flor que lo entregue al Ing. Luis Orozco, en su calidad de Director de la Carrera.

Sangolquí, 16 de Julio del 2013

Ing. Edgar Tipán

DIRECTOR

Ing. Rodolfo Gordillo

CODIRECTOR

Autorización de publicación

ESCUELA POLITÉCNICA DEL EJÉRCITO

INGENIERÍA ELECTRÓNICA, AUTOMATIZACIÓN Y CONTROL

AUTORIZACIÓN

Yo, David Patricio Paredes Flor

Autorizo a la Escuela Politécnica del Ejército la publicación, en la biblioteca virtual de la Institución del trabajo “Diseño e implementación de un controlador de posición de pedales de automóvil para personas discapacitadas”, cuyo contenido, ideas y criterios son de mi exclusiva responsabilidad y autoría.

Sangolquí, 16 de Julio del 2013

Paredes Flor David Patricio

DEDICATORIA

Dedico el esfuerzo, la dedicación y el sacrificio mostrados en el desarrollo de este trabajo a mis bellos hijos que han sido mi motor e inspiración desde el día de su nacimiento, logrando sacar fuerzas de mí de donde ya no había más, así mismo a mi madre y a mi esposa por ser un pilar fundamental, que con paciencia y amor me ayudaron sobrellevar el trabajo en los momentos más difíciles. Este trabajo es para ustedes, mi familia entera, mis seres queridos.

AGRADECIMIENTO

Un agradecimiento infinito al Señor, mi Dios, por darme la fortaleza, sabiduría, la templanza y por escuchar mis peticiones en los momentos más difíciles de mi vida, no solo durante el desarrollo de este proyecto, sino en todo momento y circunstancia.

A mi esposa e hijos que han sido mi fuente de inspiración y por quienes con ayuda del Señor vamos a llegar muy alto.

También les agradezco con todo mi corazón a mis padres por sus constantes consejos, atención y ayuda, a mi madre quien a pesar de las circunstancias, con sabiduría, ejemplo y amor ha hecho de mí la persona que soy.

Al Ing. Edgar Tipán y al Ing. Gordillo por brindarme su tiempo y sus valiosas observaciones vitales para el desarrollo de este proyecto.

ÍNDICE DE CONTENIDOS

CAPÍTULO 1	1
DEFINICIÓN DEL PROYECTO.	1
1.1. Antecedentes.	1
1.2. Justificación e Importancia.....	2
1.3. Alcance del proyecto.	2
1.4. Objetivos.	3
1.4.1. General.	3
1.4.2. Específicos.	3
CAPÍTULO 2	5
MARCO TEÓRICO	5
2.1. Introducción a los Sistemas de Control Clásicos.	5
2.1.1. Control ON-OFF	6
2.1.2. Control Proporcional (P)	8
2.1.3. Control Integrativo (I)	10
2.1.4. Control Proporcional - Integrativo (PI).....	13
2.1.5. Control Derivativo (D).....	14
2.1.6. Control Proporcional - Derivativo (PD).....	17
2.1.7. Control Proporcional –Integral - Derivativo (PID).....	18
2.2. Introducción a los Sistemas de Control Difusos.	20
2.2.1. Conjuntos Difusos.	21
2.2.2. Funciones de Pertenencia.	22
2.2.3. Operaciones con conjuntos difusos.	25
2.3. Sistemas Difusos.	27
2.3.1. Tipo Mamdani.	28
2.3.2. Tipo Takagi-Sugeno.	32
CAPÍTULO 3	34
DISEÑO DEL HARDWARE.....	34
3.1. Elementos del Sistema de Control.....	34
3.2. Actuadores Lineales.	34
3.2.1. Pruebas de Funcionamiento.	36
3.2.1.1. Pruebas de Velocidad.	36
3.2.1.2. Pruebas de Consumo de Corriente.	37

3.3.	Microcontrolador PIC18F4550.	39
3.3.1.	Generalidades.	39
3.3.2.	Arquitectura.	40
3.3.3.	Organización de Memoria.	40
3.3.4.	Interrupciones.	41
3.3.5.	Unidades Funcionales.	42
3.3.5.1.	Puertos de E/S.	42
3.3.5.2.	Temporizadores.	44
3.3.5.3.	Conversores ADC.	45
3.3.5.4.	Unidad de Captura, Comparación y PWM (CCP).	46
3.3.5.5.	Canales de Comunicación.	47
3.3.6.	Dotando de Salida Analógica al Microcontrolador.	50
3.4.	Sensores.	51
3.4.1.	Sensores de Posición	52
3.4.1.1.	Para los pedales.	52
3.4.1.2.	Para los gatillos.	53
3.4.2.	Inclinación del Automóvil.	55
3.4.3.	Montaje del GPS para medición de Velocidad.	58
3.4.4.	Pulsador indicador de cambio de marcha.	62
3.5.	Elaboración de la Tarjeta de Adquisición y Control (TAC).	62
3.5.1.	Especificaciones de componentes.	63
3.5.1.1.	Circuito Regulador de Voltaje (CRV).	64
3.5.1.2.	Controlador.	66
3.5.1.3.	Circuito Integrador – Seguidor (CIS).	68
3.5.1.4.	Demultiplexor (DEMUX).	70
3.5.1.5.	Puente H de Potencia.	72
3.5.1.6.	Circuito de memoria EEPROM.	74
3.5.2.	Diseño y Vista general de tarjeta.	76
CAPÍTULO 4		80
DESARROLLO DEL SOFTWARE.		80
4.1.	PICC.	80
4.1.1.	Palabras de Control.	80
4.1.2.	Librerías.	81
4.1.3.	Configuración del Módulo ADC.	83
4.1.4.	Configuración del Módulo PWM.	85
4.1.5.	Comunicaciones.	85

4.1.5.1.	RS-232.....	86
4.1.5.2.	I2C.....	86
4.1.5.3.	USB_CDC.....	87
4.1.6.	Requerimientos Generales del Sistema de Control.....	87
4.1.7.	Programación del PIC	89
4.1.7.1.	Manejo de variables.	89
4.1.7.2.	Sensamiento.	96
4.1.7.3.	Sistema de Control Supervisor.....	100
4.1.7.3.1.	Modo Manual.	100
4.1.7.3.2.	Modo Semiautomático	101
4.1.7.4.	Sistema de Control Regulatorio.	104
4.1.7.4.1.	Diseño de un controlador experto usando algoritmos de Lógica Difusa.	106
4.1.7.4.1.1.	Definiciones generales.	106
4.1.7.4.1.2.	Funciones de pertenencia.	107
4.1.7.4.1.3.	Definición y evaluación de las funciones de pertenencia	110
4.1.7.4.1.4.	Métodos de implicación	113
4.1.7.4.1.5.	Definición y evaluación de la base de reglas	114
4.1.7.4.1.6.	Cálculo de la salida	115
4.1.7.5.	Transmisión de datos USB.....	116
4.1.8.	Generación del (.HEX).....	119
4.2.	PICKit2.....	121
4.2.1.	Tarjeta de Programación.	121
4.2.2.	Programación del Microcontrolador.	121
4.3.	Visual Basic 2010 Express.....	123
4.3.1.	Módulo para comunicaciones IO.Ports.SerialPort.	123
4.3.2.	Introducción a Windows Forms para creación de gráficas en tiempo real. .	124
4.3.3.	Diseño de la Interface de Configuración de parámetros de Control.	126
CAPÍTULO 5		133
IMPLEMENTACIÓN DEL SISTEMA CONTROLADOR		133
5.1.	Instalación.	133
5.2.	Ajustes y Calibraciones.	139
CAPÍTULO 6		144
PRUEBAS Y RESULTADOS		144
6.1.	Pruebas de Funcionamiento.....	144
6.1.1.	Sistema de control regulatorio.....	144
6.1.1.1.	Desempeño por gatillo.....	144

6.1.1.2. Desempeño por constante.....	145
6.2. Análisis de Resultados.....	145
CAPÍTULO 7	148
CONCLUSIONES Y RECOMENDACIONES	148
7.1. Conclusiones.....	148
7.2. Recomendaciones	150
7.2.1. Programación del PIC	150
7.2.2. Montaje del sistema de control.....	150
7.2.3. Calibraciones y puesta en marcha	150
GLOSARIO.....	152
BIBLIOGRAFÍA.....	154
ANEXOS	A

ÍNDICE DE FIGURAS

CAPÍTULO II: Marco Teórico.

FIGURA 2. 1 REPRESENTACIÓN DE UN SISTEMA REALIMENTADO CON CONTROLADOR ON-OFF.....	7
FIGURA 2. 2 RESPUESTA EN LAZO CERRADO, CONTROLADOR ON-OFF.....	7
FIGURA 2. 3 CURVA DE HISTÉRESIS.....	8
FIGURA 2. 4 REPRESENTACIÓN DE UN SISTEMA REALIMENTADO CON CONTROLADOR PROPORCIONAL.....	8
FIGURA 2. 5 RESPUESTA AL IMPULSO CON UN CONTROLADOR PROPORCIONAL	10
FIGURA 2. 6 REPRESENTACIÓN DE UN SISTEMA REALIMENTADO CON CONTROLADOR INTEGRAL.....	12
FIGURA 2. 7 EFECTO DEL CONTROL INTEGRAL SOBRE EL LUGAR GEOMÉTRICO DE LAS RAÍCES DEL SISTEMA.....	12
FIGURA 2. 8 REPRESENTACIÓN DE UN SISTEMA REALIMENTADO CON CONTROLADOR PROPORCIONAL INTEGRAL.....	14
FIGURA 2. 9 RESPUESTA AL IMPULSO CON UN CONTROLADOR PROPORCIONAL INTEGRAL ...	14
FIGURA 2. 10 REPRESENTACIÓN DE UN SISTEMA REALIMENTADO CON CONTROLADOR DERIVATIVO.....	16
FIGURA 2. 11 REPRESENTACIÓN DE UN SISTEMA REALIMENTADO CON CONTROLADOR PROPORCIONAL DERIVATIVO.....	18
FIGURA 2. 12 REPRESENTACIÓN DE UN SISTEMA REALIMENTADO CON CONTROLADOR PROPORCIONAL INTEGRAL DERIVATIVO.....	19
FIGURA 2. 13 CONJUNTOS DIFUSOS PARA LA EDAD DE UNA PERSONA ADULTA.....	22
FIGURA 2. 14 FUNCIÓN DE PERTENENCIA TRIANGULAR	23
FIGURA 2. 15 FUNCIÓN DE PERTENENCIA TRAPEZOIDAL.....	24
FIGURA 2. 16 FUNCIÓN DE PERTENENCIA GAUSSIANA.....	24
FIGURA 2. 17 FUNCIÓN DE PERTENENCIA GAMMA.....	25
FIGURA 2. 18 INTERSECCIÓN DE CONJUNTOS.....	26
FIGURA 2. 19 UNIÓN DE CONJUNTOS	26
FIGURA 2. 20 COMPLEMENTO DE CONJUNTOS	27
FIGURA 2. 21 PARTES DE UN SISTEMA DIFUSO	28
FIGURA 2. 22 PROCESAMIENTO GENERAL DE SISTEMA DIFUSO TIPO MAMDANI.....	28
FIGURA 2. 23 PROCESAMIENTO GENERAL DE SISTEMA DIFUSO TIPO TAKAGI-SUGENO	32

FIGURA 2. 24 EJEMPLO DE OPERACIÓN DE UN SISTEMA TAKAGI-SUGENO	33
CAPÍTULO III: Diseño del Hardware.	
FIGURA 3. 1 ELEMENTOS DEL HARDWARE.....	34
FIGURA 3. 2 ACTUADOR LINEAL DE PRECISIÓN.	36
FIGURA 3. 3 CURVA RESPUESTA EN CORRIENTE, ACTUADOR 200 LBS.	38
FIGURA 3. 4 CURVA RESPUESTA EN CORRIENTE, ACTUADOR 400 LBS.	39
FIGURA 3. 5 DIAGRAMA DE PINES PIC18F4550.....	44
FIGURA 3. 6 SEÑAL PWM TÍPICA.....	47
FIGURA 3. 7 CIRCUITO INTEGRADOR.	51
FIGURA 3. 8 CONFIGURACIÓN DE POTENCIÓMETRO LINEAL EN PEDALES.....	52
FIGURA 3. 9 GRÁFICA DE RESPUESTA EN VOLTAJE EN FUNCIÓN DEL DESPLAZAMIENTO DEL POTENCIÓMETRO LINEAL.	53
FIGURA 3. 10 GALGA EXTENSIOMÉTRICA FLEX22 (FS-L-005-253-ST).	54
FIGURA 3. 11 GRÁFICA DE RESPUESTA EN VOLTAJE EN FUNCIÓN DEL DESPLAZAMIENTO DEL POTENCIÓMETRO LINEAL.	55
FIGURA 3. 12 ACCELERÓMETRO MMA7361, MINI PCB	57
FIGURA 3. 13 MÉTODO DE LA HIPOTENUSA.	59
FIGURA 3. 14 GPS CLICK.	60
FIGURA 3. 15 PINES USADOS DEL GPS.....	60
FIGURA 3. 16 PULSADOR INDICADOR DE CAMBIO DE MARCHA.....	62
FIGURA 3. 17 ESQUEMA BÁSICO DE TARJETA DE CONTROL.	63
FIGURA 3. 18 COMPONENTES DEL SISTEMA CONTROLADOR.....	64
FIGURA 3. 19 CIRCUITO REGULADOR DE VOLTAJE.....	64
FIGURA 3. 20 CIRCUITO REGULADOR, 3.15 VDC.....	65
FIGURA 3. 21 CIRCUITO INTEGRADOR - AMPLIFICADOR DE CORRIENTE.....	69
FIGURA 3. 22 MAX232 USADO COMO FUENTE NEGATIVA DE VOLTAJE.....	70
FIGURA 3. 23 PRINCIPIO DE FUNCIONAMIENTO DEL DEMUX.	70
FIGURA 3. 24 DIAGRAMA DE CONEXIÓN DEL DEMUX ANÁLOGO CD4053.....	71
FIGURA 3. 25 ESQUEMA DE CONEXIÓN DEL PUENTE H DE POTENCIA	73
FIGURA 3. 26 PUENTE H - CIRCUITO IMPLEMENTADO.	74
FIGURA 3. 27 ESQUEMA CONEXIÓN EEPROM ST24C08.	76
FIGURA 3. 28 ESQUEMA ELECTRÓNICO DE ELEMENTOS Y CIRCUITOS EN ARES.	77
FIGURA 3. 29 EXPORTAR PISTAS EN FORMATO PDF.....	77

FIGURA 3. 30 PISTAS GENERADAS EN FORMATO PDF.....	78
FIGURA 3. 31 VISTA DE LA TAC, CAPA SUPERIOR. A). VISTA 3D GENERADA EN SOFTWARE ARES. B). VISTA REAL.....	78
FIGURA 3. 32 VISTA DE LA TAC, CAPA INFERIOR. A). VISTA 3D GENERADA EN SOFTWARE ARES. B). VISTA REAL.....	79
CAPÍTULO IV: Desarrollo del Software.	
FIGURA 4. 1 ERROR DE CONVERSIÓN DEL ADC.....	84
FIGURA 4. 2 SETEO DE FRECUENCIA PARA LA SEÑAL PWM.....	85
FIGURA 4. 3 DEFINICIÓN DEL SISTEMA DE CONTROL.....	88
FIGURA 4. 4 SISTEMA CONTROLADOR, BLOQUES FUNCIONALES.....	89
FIGURA 4. 5 FUNCIONES DE PERTENENCIA: A). ERROR. B). CAMBIO DEL ERROR.....	92
FIGURA 4. 6 SINGLETONS DE SALIDA	93
FIGURA 4. 7 BASE DE REGLAS.....	94
FIGURA 4. 8 DIAGRAMA DE FLUJO DE LECTURA DE VARIABLES	96
FIGURA 4. 9 DIAGRAMAS DE FLUJO: A). PARA LA LECTURA DE LOS PEDALES. B). PARA CALCULAR LA INCLINACIÓN DEL AUTOMÓVIL.....	97
FIGURA 4. 10 DIAGRAMA DE FLUJO PARA LECTURA DE GATILLOS	98
FIGURA 4. 11 DIAGRAMA DE FLUJO PARA LECTURA DE LA VELOCIDAD DEL AUTOMÓVIL	99
FIGURA 4. 12 DECISIÓN DE MODO, CONTROL SUPERVISOR	100
FIGURA 4. 13 MODO MANUAL	101
FIGURA 4. 14 MODO SEMIAUTOMÁTICO	102
FIGURA 4. 15 GENERACIÓN DE SET POINTS	103
FIGURA 4. 16 CONTROL DE CAMBIO DE MARCHA.....	104
FIGURA 4. 17 DIAGRAMA DE FLUJO PARA EL CONTROL REGULATORIO	105
FIGURA 4. 18 SIENDO LA SEÑAL DE CONTROL U EN FUNCIÓN DEL ERROR Y CAMBIO DEL ERROR	106
FIGURA 4. 19 FUNCIONES DE PERTENENCIA PARA EL ERROR.....	108
FIGURA 4. 20 FUNCIONES DE PERTENENCIA PARA EL CAMBIO DEL ERROR	108
FIGURA 4. 21 SINGLETONS DE SALIDA	109
FIGURA 4. 22 EJEMPLOS DE FUNCIONES DE PERTENENCIA EL DIAGRAMA DE FLUJO PARA EL PROCESO DE FUZZIFICACIÓN SE MUESTRA EN LA SIGUIENTE FIGURA.....	111
FIGURA 4. 23 DIAGRAMA DE FLUJO PARA LA FUZZIFICACIÓN	111
FIGURA 4. 24 EVALUAR FUNCIONES DE PERTENENCIA ACTIVAS	112

FIGURA 4. 25 OBTENER VALORES DE PERTENENCIA	113
FIGURA 4. 26 BASE DE REGLAS.....	114
FIGURA 4. 27 DIAGRAMA DE FLUJO PARA EL MOTOR DE INFERENCIA DIFUSA.....	115
FIGURA 4. 28 DIAGRAMA DE FLUJO PARA EL INICIO Y TRANSMISIÓN DE DATOS.....	118
FIGURA 4. 29 COMPILACIÓN DEL CÓDIGO EN EL PICC.....	119
FIGURA 4. 30 RESUMEN DE COMPILACIÓN	120
FIGURA 4. 31 GENERACIÓN DEL .HEX EN LA CARPETA DEL PROYECTO	120
FIGURA 4. 32 VISTA DE LA TARJETA DE PROGRAMACIÓN CONECTADA A LA PC	121
FIGURA 4. 33 PROGRAMACIÓN DEL PIC MEDIANTE EL PICKIT 2	122
FIGURA 4. 34 GRABACIÓN EXITOSA DEL PIC.....	123
FIGURA 4. 35 INSERTAR UN CHART EN EL FORMULARIO.....	125
FIGURA 4. 36 INSERTAR Y CONFIGURAR SERIES EN LA GRÁFICA	125
FIGURA 4. 37 CONFIGURACIÓN DE TIPO DE GRÁFICA	126
FIGURA 4. 38 ASIGNAR RANGOS PARA LOS GATILLOS.....	127
FIGURA 4. 39 ASIGNAR RANGOS PARA LOS PEDALES	128
FIGURA 4. 40 ASIGNAR RANGOS PARA INCLINACIÓN Y VELOCIDAD.....	128
FIGURA 4. 41 ASIGNAR RANGOS DE FUNCIONES DE PERTENENCIA PARA EL ERROR.....	129
FIGURA 4. 42 ASIGNAR RANGOS DE FUNCIONES DE PERTENENCIA PARA EL CAMBIO DEL ERROR	129
FIGURA 4. 43 ASIGNAR GANANCIAS Y VALORES DE SINGLETONS DE SALIDA	130
FIGURA 4. 44 ASIGNAR BASE DE REGLAS.....	130
FIGURA 4. 45 INICIAR AL DESEMPEÑO	131
FIGURA 4. 46 DESEMPEÑO POR CONSTANTE	132
FIGURA 4. 47 DESEMPEÑO POR SEÑAL CUADRADA	132

CAPÍTULO V: Implementación del sistema controlador.

FIGURA 5. 1 ESTRUCTURA MONTADA SOBRE LOS RIELES DEL ASIENTO DEL CONDUCTOR	133
FIGURA 5. 2 ESTRUCTURA METÁLICA SOPORTE PARA LOS ACTUADORES	134
FIGURA 5. 3 UNIÓN DE ESTRUCTURA METÁLICA CON ACTUADORES LINEALES.....	135
FIGURA 5. 4 UNIÓN ENTRE ACTUADORES Y PEDALES	135
FIGURA 5. 5 MONTAJE DE SENSORES SOBRE ACTUADORES LINEALES	136
FIGURA 5. 6 POSICIÓN DE LOS PIES DEL CONDUCTOR ENTRE LA ESTRUCTURA	137
FIGURA 5. 7 GALGA DEL FRENO EN DEDO ÍNDICE DE GUANTE IZQUIERDO.....	137

FIGURA 5. 8 GALGAS DEL ACELERADOR Y EMBRAGUE EN EL GUANTE DERECHO	138
FIGURA 5. 9 CASE DE LA TARJETA DE ADQUISICIÓN Y CONTROL	139

CAPÍTULO VI: Pruebas y resultados.

FIGURA 6. 1 DESEMPEÑO POR GATILLO.....	144
FIGURA 6. 2 RESPUESTA A VARIOS SET POINTS CONSECUTIVOS	145
FIGURA 6. 3 GRÁFICA DE RESPUESTA DETALLADA.....	145

ÍNDICE DE TABLAS

CAPÍTULO II: Marco Teórico.

TABLA 2. 1 CONJUNTOS DIFUSOS PARA LA EDAD DE UNA PERSONA ADULTA	22
TABLA 2. 2 VALORES DE FUNCIÓN DE PERTENENCIA TRIANGULAR.....	23
TABLA 2. 3 VALORES DE FUNCIÓN DE PERTENENCIA TRAPEZOIDAL	24
TABLA 2. 4 EJEMPLO DE VARIABLES LINGÜÍSTICAS PARA LOS CONJUNTOS DIFUSOS.	30
TABLA 2. 5 EJEMPLO DE LA BASE DE REGLAS.	30

CAPÍTULO III: Diseño del Hardware.

TABLA 3. 1 CARACTERÍSTICAS DE ACTUADORES LINEALES	35
TABLA 3. 2 RESPUESTA DE VELOCIDAD DE ACTUADORES LINEALES.	37
TABLA 3. 3 RESPUESTA EN CORRIENTE, ACTUADOR 200 LBS.	37
TABLA 3. 4 RESPUESTA EN CORRIENTE, ACTUADOR 400 LBS.	38
TABLA 3. 5 ORGANIZACIÓN DE LA MEMORIA PIC18F4550.....	41
TABLA 3. 6 INTERRUPCIONES EN PIC18F4550.	42
TABLA 3. 7 DISTRIBUCIÓN DE PINES DE E/S DEL PIC18F4550	43
TABLA 3. 8 CARACTERÍSTICAS DE TEMPORIZADORES, PIC18F4550.....	45
TABLA 3. 9 DISTRIBUCIÓN DE PINES DEL CONVERTOR ADC DEL PIC18F4550.....	46
TABLA 3. 10 LISTA DE VARIABLES SUS RESPECTIVOS SENSORES.....	51
TABLA 3. 11 MEDICIONES DE VOLTAJE EN FUNCIÓN DEL DESPLAZAMIENTO DEL POTENCIÓMETRO LINEAL.	52
TABLA 3. 12 MEDICIONES DE VOLTAJE EN FUNCIÓN DEL ÁNGULO DE FLEXIÓN DE LA GALGA EXTENSIOMÉTRICA.	55
TABLA 3. 13 ESPECIFICACIONES GENERALES ACELERÓMETRO MMA7361.....	56
TABLA 3. 14 CONSTANTES PARA CÁLCULO DE INCLINACIÓN	58
TABLA 3. 15 CARACTERÍSTICAS GPS CLICK.	60
TABLA 3. 16 CONTENIDO DE LA TRAMA NMEA.....	61
TABLA 3. 17 DISTRIBUCIÓN DE ENTRADAS AL MICROCONTROLADOR PRINCIPAL.....	67
TABLA 3. 18 DISTRIBUCIÓN DE SALIDAS DEL MICROCONTROLADOR PRINCIPAL.	68
TABLA 3. 19 TABLA DE VERDAD DEL DEMUX ANÁLOGO CD4053.	72
TABLA 3. 20 CARACTERÍSTICAS EEPROM ST24C08.	75

CAPÍTULO IV: Desarrollo del Software.

TABLA 4. 1 PICC - LISTA DE FUSIBLES UTILIZADOS	81
TABLA 4. 2 LIBRERÍA Y COMANDOS PARA EL MANEJO DEL PUERTO USB.....	82
TABLA 4. 3 LIBRERÍA Y COMANDOS PARA EL MANEJO DE LA MEMORIA EEPROM.....	83
TABLA 4. 4 LIBRERÍA Y COMANDO PARA EL MANEJO DE VECTORES.	83
TABLA 4. 5 DIRECCIONES PARA LOS LÍMITES DE UNIVERSOS DE DISCURSO.....	91
TABLA 4. 6 DIRECCIONES PARA LOS LÍMITES DE FUNCIONES DE PERTENENCIA PARA EL ERROR Y CAMBIO DEL ERROR.	92
TABLA 4. 7 DIRECCIONES PARA LAS POSICIONES DE LOS SINGLETONS	93
TABLA 4. 8 DIRECCIONES PARA LAS CONSTANTES DE LA BASE DE REGLAS.....	94
TABLA 4. 9 DIRECCIONES PARA LAS GANANCIAS.....	94
TABLA 4. 10 DIRECCIONES PARA LAS VARIABLES DEL PROCESO.....	95
TABLA 4. 11 DESCRIPCIÓN DE SUFIJOS PARA VARIABLES.....	96
TABLA 4. 12 VARIABLES DE INCREMENTO DE LECTURA DE GATILLOS	99
TABLA 4. 13 ENTRADAS Y SALIDAS DEL SISTEMA REGULADOR DIFUSO	107
TABLA 4. 14 COMANDOS DE TRANSMISIÓN DE DATOS	117
TABLA 4. 15 CATEGORÍAS DE COMANDOS DE TRANSMISIÓN.....	118
TABLA 4. 16 PARÁMETROS DE CONFIGURACIÓN DEL PUERTO SERIAL.....	124
TABLA 4. 17 DETALLE DE PESTAÑAS DE LA INTERFAZ DE CONFIGURACIÓN.....	127
TABLA 4. 18 MODOS DE FUNCIONAMIENTO DEL DESEMPEÑO.....	131

CAPÍTULO V: Implementación del sistema controlador.

TABLA 5. 1 LÍMITES UNIV. DISCURSO PARA LOS GATILLOS	140
TABLA 5. 2 LÍMITES UNIV. DISCURSO PARA LOS PEDALES.....	140
TABLA 5. 3 RANGOS PARA LA INCLINACIÓN Y VELOCIDAD	140
TABLA 5. 4 RANGOS DE FUNCIONES DE PERTENENCIA DE LAS ENTRADAS PARA LOS TRES PEDALES	141

CAPÍTULO VI: Pruebas y resultados.

TABLA 6. 1 RESUMEN DE PARÁMETROS DE RESPUESTA DEL SISTEMA	146
---	-----

RESUMEN

El presente documento detalla el diseño e implementación de un sistema electrónico de posicionamiento de pedales para un automóvil de tres marchas, controlado mediante gatillos, dirigido a personas con capacidades limitadas en sus extremidades inferiores. El sistema está compuesto por dos partes fundamentales, el hardware y software. El hardware que lo conforma en conjunto los elementos finales de control que son los actuadores lineales de precisión encargados de mover los pedales del automóvil, y la tarjeta de adquisición y control que es un sistema microcontrolado que cuenta con dos partes, la parte de procesamiento cuya base es el PIC18F4550 y la parte de fuerza mediante un grupo de tres puentes H analógicos de 10 amperios para activar los actuadores lineales, además, está la estructura metálica de soporte para instalar dichos actuadores en los pedales. El software, que consta de la programación del microcontrolador que aplica algoritmos de lógica difusa en el control regulatorio de posición y lógica de estados en el control supervisor, encargado de manejar las variables y diferentes condiciones de funcionamiento. Además, se tiene una interface HMI de configuración de parámetros de control, la cual se comunica a la tarjeta de control mediante cable USB usando comunicación serial USB_CDC en la cual se pueden configurar las constantes para los controladores difusos de los pedales. Esta interface permite la configuración de parámetros sobre la marcha, permitiendo además visualizar la respuesta de los controladores mediante la herramienta de desempeño.

CAPÍTULO 1

DEFINICIÓN DEL PROYECTO.

1.1. Antecedentes.

A mediados de los años 70, el control difuso nace como una tecnología emergente, orientada hacia las aplicaciones industriales específicamente en el control de velocidad de una máquina de vapor (Mamdani, Basic Aplicaciones, 1974), con el objetivo de resolver problemas de control que mediante métodos tradicionales presentaban una mayor complejidad en su solución, esto es, cuando no se puede obtener un modelo matemático preciso o razonablemente aproximado de un sistema complejo (Lotfi, 1973), y en particular cuando la descripción del sistema requiere de conocimientos previos sobre el mismo, en términos generales, la metodología de lógica difusa para el diseño de controladores tiene características sobresalientes sobre muchas otras técnicas de control.

Desde entonces se ha dado paso a la utilización del conocimiento humano en el diseño de controladores (Mamdani, Intl. J. Man machine study, 1975), lo cual no solo representa una ventaja, sino que es prácticamente necesario. La relativa nueva tecnología de control difuso tiende a ser una mejor alternativa para el diseño de sistemas de control complejos. Comparada con los enfoques convencionales, esta no se apoya en la modelación matemática de un sistema físico sino que lo trabaja emulando la inteligencia del ser humano.

En este contexto, la aplicación de lógica difusa en el diseño e implementación de un controlador difuso de posición, se lleva a cabo en el software PICC, utilizado en la programación de microcontroladores, además del Software Visual Basic 2010 Express para la creación una interface de configuración rápida de parámetros de control. Para la implementación del sistema controlador se dispone de un microcontrolador PIC18F4550 y sus respectivos actuadores lineales de precisión

Las pruebas de la efectividad del control difuso se verán en la variación dinámica de posición de los pedales en un automóvil convencional de marchas.

1.2. Justificación e Importancia.

Gracias al creciente desarrollo tecnológico en el campo del control inteligente, se hace posible la creación más natural e intuitiva de controladores utilizando algoritmos de lógica difusa. Esta metodología permite evitar las complicaciones matemáticas propias de un desarrollo clásico de controladores. En el caso del presente proyecto, donde el sistema pretende emular el movimiento de los pies sobre los pedales de un automóvil al manejarlo, el control intuitivo realizado por un humano puede parecer obvio en el desarrollo del controlador.

Además, como futuros profesionales, es nuestro deber utilizar los conocimientos adquiridos a lo largo de nuestra carrera, y aplicarlos en la solución de problemas que en este caso en particular, está orientado hacia mejorar la calidad de vida de personas cuyas cualidades físicas les impiden o limitan en cierta manera el conducir un automóvil debido a la incapacidad de utilizar sus piernas.

1.3. Alcance del proyecto.

Se realizó la recopilación de información y la revisión teórica de conceptos que fundamentan la tecnología de control difuso y su impacto en el diseño de controladores inteligentes, una vez estudiados los fundamentos teóricos, se procedió con la interacción teórico experimental, diseñando la tarjeta de control y desarrollando el software, para lo cual se dispone del software PICC, utilizado en la programación de microcontroladores, en este caso se utilizó el microcontrolador PIC18F4550. Además del software Visual Basic 2010 Express para la creación una interface de configuración de parámetros del controlador, la cual mediante comunicación usb, permite el seteo de parámetros al controlador.

Se realiza la implementación del sistema controlador sobre los tres pedales de un automóvil convencional de marchas, mediante tres actuadores lineales de precisión

comandados por gatillos, para los cuales se utiliza tres galgas extensiométricas FLEX22 localizadas en tres dedos de las manos indistintamente. Para la medición de posición de los actuadores lineales se utiliza tres potenciómetros lineales. El sistema de control dispone de dos modos de funcionamiento seleccionables, el modo *manual* en el cual cada actuador lineal se lo comanda mediante un gatillo individual; y el modo *semiautomático*, donde se tiene dos gatillos, uno para el control independiente del actuador lineal sobre el freno, y el segundo gatillo que controla los actuadores sobre el acelerador y embrague simultáneamente, para esto, como parámetro de control se requiere conocer la inclinación del automóvil, esto es, en qué ángulo está respecto a la horizontal, esencialmente si está en cuesta o en bajada; con el modo semiautomático se pretende brindar un control más intuitivo y amigable para el conductor.

Para los dos modos de funcionamiento, por razones obvias de seguridad se implementa un control de limitación de velocidad configurable, para esto se dispondrá de un sensor de velocidad a base de un módulo GPS.

Cabe aclarar que el sistema realiza el control de posición sobre los tres pedales del automóvil, mientras que los movimientos de la palanca de cambios y el volante los realiza el conductor.

1.4. Objetivos.

1.4.1. General.

Diseñar e implementar un sistema controlador de posición para los tres pedales de un automóvil de marchas, de mando manual con gatillo para personas con capacidades diferentes de tipo motriz en sus extremidades inferiores, mediante microcontrolador utilizando algoritmos de lógica difusa.

1.4.2. Específicos.

1. Estudiar y analizar la Lógica Difusa para el diseño de controladores inteligentes.
2. Programar el microcontrolador aplicando los criterios de Lógica Difusa.
3. Diseñar la Interface de Configuración de parámetros del controlador en el software Visual Basic 2010 Express, para realizar el seteo de parámetros sobre la marcha.

-
4. Implementar el controlador sobre los tres pedales de un automóvil convencional mediante tres actuadores lineales de precisión.
 5. Probar el desempeño del controlador en su modo de funcionamiento semiautomático.

CAPÍTULO 2

MARCO TEÓRICO

2.1. Introducción a los Sistemas de Control Clásicos.

La mayoría sino todos sistemas de control clásicos pueden ser tratados basándonos en teoría de ecuaciones, lo cual requiere un estudio de la planta por distintos métodos como pueden ser en el dominio de Laplace, en el dominio de la frecuencia, gráficas de Nyquist, técnicas en el lugar geométrico de las raíces (Kuo, 1996), etc.

Todos estos métodos requieren, en el caso de no existir todavía, un complejo desarrollo del modelo matemático del proceso o planta, linealizándolo respecto a un punto de operación para finalmente representar a dicha planta en el dominio de Laplace mediante una función de transferencia (Ogatta, 1998), con lo cual es posible la creación de parámetros para un controlador el que básicamente se encargará de gobernar el comportamiento de la planta, disminuyendo el error en estado estacionario, eliminando o suprimiendo las oscilaciones, y, entre otras cosas, mejorando el tiempo de respuesta.

En el caso de este trabajo en particular, no existe un modelo matemático ya definido que represente a un actuador lineal que ejerza presión sobre los pedales de un automóvil desplazándolos a una posición definida, razón por la cual se hace difícil el utilizar cualquier método analítico mencionado, que permita diseñar un controlador hallando los parámetros correctos para gobernar de forma óptima a la planta en sus varias condiciones de trabajo.

Es posible hallar estos parámetros para ciertas condiciones de trabajo, sin embargo estos no servirán cuando nos hallemos en diferentes condiciones haciendo mucho más complejo y largo el algoritmo de control.

Por esa razón no se utilizarán términos matemáticos para describir el sistema de control, en vez de esto, se usarán las técnicas de control aplicando la lógica difusa.

Los siguientes son algunos de los controladores y reguladores más comunes, en este trabajo se revisarán las características de los primeros 6 controladores en el dominio de Laplace con realimentación.

- Control ON-OFF.
- Control Proporcional (P).
- Control Integrativo (I).
- Control Proporcional - Integrativo (PI).
- Control Derivativo (D).
- Control Proporcional - Derivativo (PD).
- Control Proporcional – Integral – Derivativo (PID).
- Retroalimentación de estado.
- Adelanto/retraso de fase.
- Cancelación de polos y ceros.
- Control robusto.
- Control adaptativo.
- Combinaciones entre controladores.

2.1.1. Control ON-OFF

En el control ON-OFF como su nombre lo dice, la salida del controlador solo puede cambiar entre dos valores, encendido (1) y apagado (0). Este controlador produce en la planta una continua desviación del valor de referencia debido a que al comportarse como un switch no tiene la capacidad para producir un valor variable, diferente de 1 o 0, para producir un valor exacto en la variable controlada, en la figura 2.1 se muestra el esquema de este controlador donde se tienen las señales de entrada o setpoint $R(s)$, el error $E(s)$, la señal de control $C(S)$ y la señal de salida.

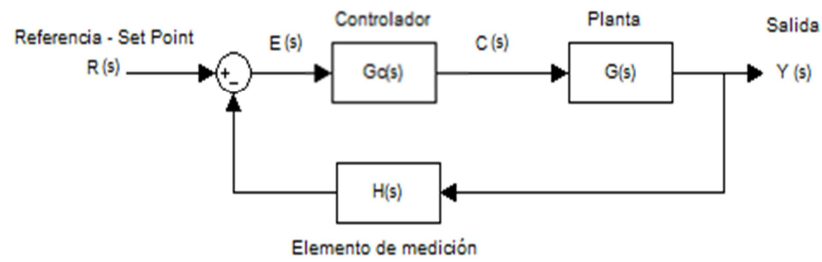


Figura 2. 1 Representación de un sistema realimentado con controlador ON-OFF.

La acción del controlador ON-OFF tiene un simple mecanismo de construcción y activación, por esa razón este tipo de controladores es de los de más amplio uso, y es comúnmente utilizado en sistemas de control de temperatura (Morillo, Técnicas de Control de Procesos, 2011), en la figura No.2.2 se muestra la respuesta típica de una planta de temperatura así como la señal de control aplicada.

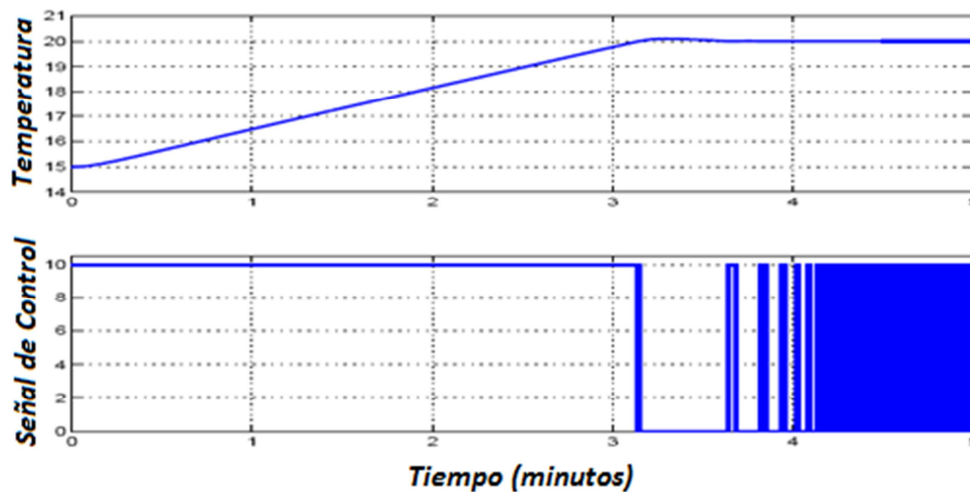


Figura 2. 2 Respuesta en lazo cerrado, Controlador ON-OFF.

Para determinar la regulación del controlador, son importantes los parámetros amplitud y período de tiempo de la oscilación. La oscilación en la respuesta de la planta depende de factores como el período de tiempo de la señal de control y la histéresis propia del sistema.

Sabemos que la mayoría de los sistemas al ser controlados tienen una respuesta que no es la misma al momento de que la variable controlada aumenta y luego disminuye, en este sentido se define al error en histéresis como la diferencia entre los valores medidos obtenidos cuando la cantidad medida se incrementa a cuando se decrementa en un mismo valor. Se expresa en función del valor de la máxima histéresis

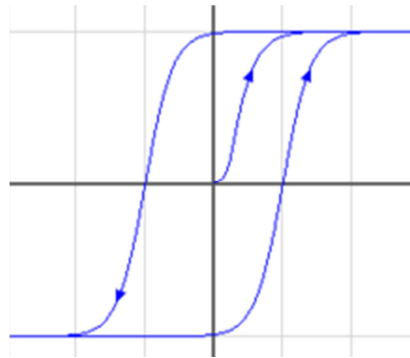


Figura 2. 3 Curva de Histéresis.

2.1.2. Control Proporcional (P)

Los sistemas de control clásicos actúan sobre el error obtenido mediante la realimentación de la salida en la planta, es decir, sobre la diferencia entre la referencia y la variable controlada. Sobre dicho error, el controlador aplica una ganancia y genera una salida. Dicho de otra forma el controlador proporcional genera en su salida, una señal de control que es proporcional al error medido (Spartacus Gomáriz, Diseño de sistemas de control de tiempo continuo y discreto, 1998).

En la figura 2.4 se muestra el esquema del controlador proporcional donde al realimentar la señal de salida $Y(s)$, el controlador $G_c(s)$ le aplica una ganancia K_p obteniendo la señal de control $C(s)$.

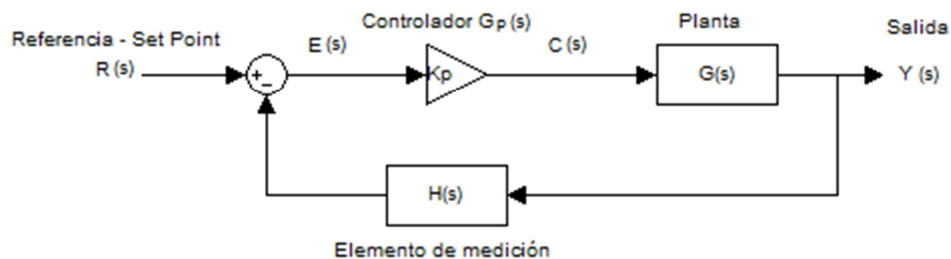


Figura 2. 4 Representación de un Sistema realimentado con Controlador Proporcional.

La acción de control es proporcional al error, teniendo la siguiente ecuación:

$$C(s) = Kp \cdot E(s)$$

Con lo cual la función de transferencia del controlador proporcional es:

$$G_p(s) = \frac{C(s)}{E(s)} = Kp$$

En los controladores, la acción proporcional es la acción de control lineal más importante. Como ventajas se puede mencionar:

- La facilidad y rapidez de comprobar los resultados y por lo tanto la facilidad de sintonizar la ganancia óptima.
- La instantaneidad de la aplicación.

Como desventajas se tiene:

- La alta inmunidad al ruido.
- La imposibilidad de corregir errores en estado estacionario.
- Si la ganancia fijada es demasiado baja, la respuesta de la variable controlada será muy estable en estado estacionario, pero la velocidad de respuesta será lenta ante cambios del set point porque la señal de control no es lo suficientemente agresiva para realizar cambios rápidos en la VC (instrumentacionycontrol, 2009).
- Con control proporcional, la única manera de obtener una respuesta de acción rápida ante cambios de set point o ante perturbaciones en el proceso es fijar una ganancia Kp lo suficientemente alta, hasta que en la señal de salida aparezca un pequeño sobre impulso.
- Se debe tomar en cuenta que el aumentar la ganancia de forma exagerada puede ocasionar que los polos menos dominantes de la FT adquieran importancia y hagan inestable al sistema.

Con lo dicho anteriormente se tiene en la figura 2.5, la respuesta al impulso de un proceso aplicando un control Proporcional.

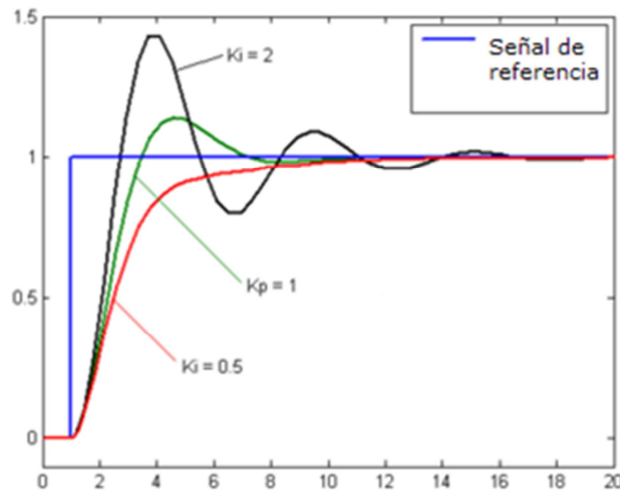


Figura 2. 5 Respuesta al impulso con un controlador Proporcional

Para expresar la sensibilidad del controlador se suele usar la notación en base al inverso de la ganancia, es decir en base a la banda proporcional la cual es el cambio necesario en la variable de proceso para que el respectivo actuador realice el 100% de la carrera o esté activado en su totalidad (Sánchez, 2003). Mientras menor sea la PB, mayor será la sensibilidad del controlador Tenemos así la expresión de la Banda Proporcional:

$$PB = \frac{100}{Kp}$$

La PB es conceptualmente más clara que la ganancia pues se relaciona con el cambio del error que se necesita para recorrer el 100% del actuador.

2.1.3. Control Integrativo (I)

Con lo revisado anteriormente, está claro que la parte proporcional no considera el tiempo, por lo tanto, la mejor manera de solucionar el error en estado estacionario es hacer que el sistema disponga de alguna componente que tenga en cuenta la variación de la

salida respecto al tiempo, la forma de hacerlo es incluyendo las acciones integral y derivativa.

El modo de control Integrativo tiene como propósito disminuir y si está bien configurado, eliminar el error en estado estacionario, provocado por el modo proporcional. La acción integral utiliza la integración como método para promediarlo o sumarlo por un periodo, determinando así la desviación de la señal de salida respecto al set point (Spartacus Gomáriz, Diseño de sistemas de control de tiempo continuo y discreto, 1998). La fórmula integral está dada por:

$$u_i(t) = ki \int_0^t e(\tau) \cdot d\tau$$

Transformando al dominio de Laplace con condiciones iniciales nulas:

$$u_i(s) = \frac{Ki}{s} E(s)$$

Se suelen poner las distintas constantes en función de la ganancia proporcional K_p , con esto, se define la constante K_i como:

$$K_i = \frac{K_p}{T_i}$$

Donde T_i es el tiempo integral, por lo tanto $1/T_i$ representa una frecuencia, la cual se denomina *Frecuencia de Reset* (Morillo, Control Integrativo, 2011), y no es más que la cantidad de veces que se acumula la acción proporcional por la presencia de la acción integral, si el error persiste y es constante. Finalmente la función de transferencia de un controlador Integral es:

$$G_i(s) = \frac{U_i(s)}{E(s)} = \frac{K_p}{T_i \cdot s}$$

En la figura 2.6 se muestra el esquema de un controlador Integral.

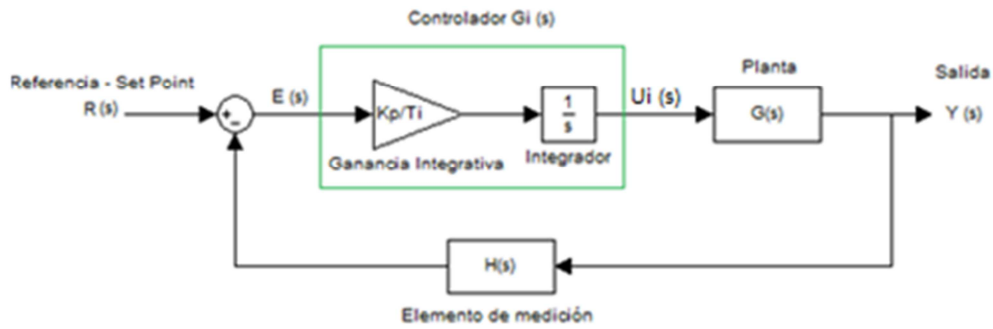


Figura 2. 6 Representación de un Sistema realimentado con Controlador Integral.

La característica más importante de este control es que la acción correctora se efectúa mediante la integral del error, esto nos dice que el control integral proporciona una señal de control que está en función de la historia de la señal de error, es decir, en función de la suma de los valores anteriores durante un periodo de tiempo, lo que permite obtener una señal de control diferente de cero, aunque la señal de error sea cero. Dicho esto, el que el error $e(t) = 0$ no implica que la señal de control $ci(t) = 0$, mas bien $ci(t) = cte$ implica que $e(t) = 0$.

Entre los beneficios del control integral está claramente el de reducir y hasta eliminar el error en estado estacionario en la señal de salida de la planta. Sin embargo, la acción del control integral empeora la estabilidad relativa del sistema, aumentando el sobre impulso inicial durante la respuesta transitoria, pudiendo incluso hacer de un sistema estable, un inestable.

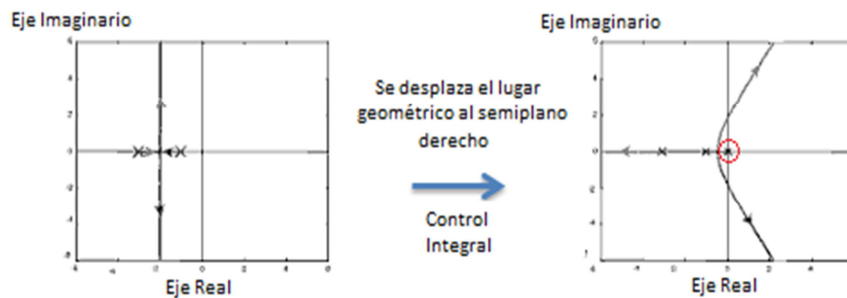


Figura 2. 7 Efecto del Control Integral sobre el lugar geométrico de las raíces del sistema.

Esto se da porque al aplicar el control integral $\frac{K_i}{s}$, se incorpora un polo en el origen a la función de transferencia del sistema, desplazando el lugar geométrico de las raíces hacia el semiplano derecho del plano S como se muestra en la figura 2.6.

2.1.4. Control Proporcional - Integrativo (PI).

La acción de control Proporcional Integral se define mediante:

$$u_{pi}(t) = k_p \cdot e(t) + \frac{k_p}{T_i} \int_0^t e(\tau) \cdot d\tau$$

Donde k_p es la ganancia proporcional y T_i se denomina tiempo integral, este ajusta la acción de control, mientras que si cambiamos el valor de k_p , esta afecta tanto a la parte integral como a la proporcional. El inverso del tiempo integral, se denomina velocidad de ajuste, que no es más que la cantidad de veces por minuto que se duplica la parte proporcional de la acción de control.

Si se aplica la transformada de Laplace con condiciones iniciales nulas:

$$U_{pi}(s) = K_p \cdot E(s) + \frac{K_p}{T_i \cdot s} E(s)$$

De aquí la función de transferencia del controlador Proporcional Integral es:

$$G_{pi}(s) = \frac{U_{pi}(s)}{E(s)} = K_p \left(1 + \frac{1}{T_i \cdot s} \right)$$

En la figura 2.8 se muestra el esquema de un controlador con acción Proporcional Integral.

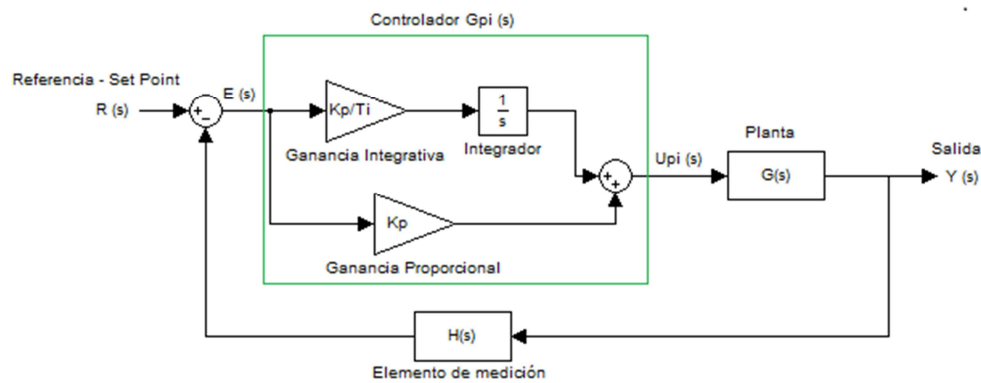


Figura 2. 8 Representación de un Sistema realimentado con Controlador Proporcional Integral.

Como se revisó anteriormente en el control integrativo, este proporciona una señal de control en función de la historia de la señal del error. Con esto, el control Proporcional Integral combina las ventajas de la acción proporcional y la de la acción integral (Antini Manuel, 2001):

La figura 2.9 muestra los efectos del controlador PI.

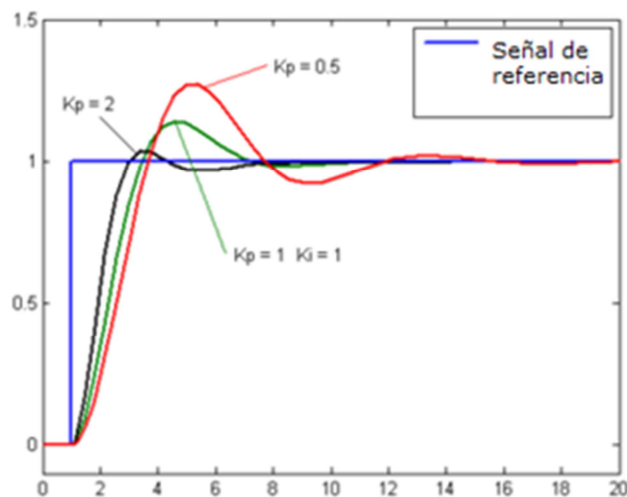


Figura 2. 9 Respuesta al impulso con un controlador Proporcional Integral

- Acción Integral – Reduce el error en estado estacionario.
- Acción Proporcional – Reduce el riesgo de inestabilidad que se añade al sistema debido a la acción Integral.

2.1.5. Control Derivativo (D).

La acción de control derivativa, genera una señal de control proporcional a la derivada de la señal de error, donde se tiene que:

$$u_d(t) = k_d \frac{de(t)}{dt}$$

Siendo $u_d(t)$ la salida de un controlador Derivativo en función del tiempo. De esta manera, con la derivada del error se puede conocer la pendiente de la señal respecto al tiempo y con esto sus características dinámicas, es decir su crecimiento o decrecimiento (Antini Manuel, 2001).

La acción derivativa se manifiesta cuando hay un cambio en el valor absoluto del error, así, si el error es constante, solamente actúan los modos proporcional e integral. La función de la acción derivativa es mantener el error al mínimo corrigiéndolo proporcionalmente con la misma velocidad que se produce; de esta manera evita que el error se incremente.

El control derivativo se caracteriza por el tiempo de acción derivada en minutos de anticipo. Donde también es conveniente expresar la constante k_d en términos de la ganancia proporcional k_p como se ilustra:

$$k_d = k_p \cdot T_d$$

Donde T_d es el tiempo derivativo o de adelanto. Con esta definición y aplicando la transformada de Laplace con condiciones iniciales nulas se obtiene:

$$U_d(s) = K_p \cdot T_d \cdot s \cdot E(s)$$

Donde la función de transferencia de un controlador derivativo es:

$$G_d(s) = \frac{U_d(s)}{E(s)} = K_p T_d \cdot s$$

En la figura 2.10 se muestra el esquema de un controlador con acción Derivativa.

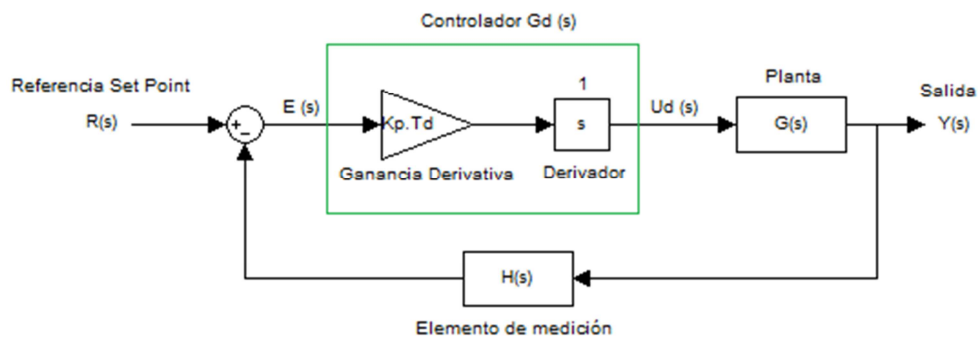


Figura 2. 10 Representación de un Sistema realimentado con Controlador Derivativo.

Cuando el tiempo de acción derivada es grande, hay inestabilidad en el proceso. Cuando el tiempo de acción derivada es pequeño la variable oscila demasiado con relación al punto de consigna.

Con lo cual, del control derivativo se tienen las siguientes ventajas:

- La acción derivativa puede ayudar a disminuir el rebasamiento de la variable durante el arranque del proceso.
- Tiende a estabilizar el sistema, ya que la acción derivativa es anticipativa, es decir, adelanta la acción de control frente a la aparición de una tendencia del error.
- Este puede emplearse en sistemas con tiempo de retardo considerables, porque permite un efecto rápido sobre la variable después de presentarse una perturbación en el proceso.

Sin embargo, se tienen las siguientes desventajas:

- La acción derivativa es prácticamente inaplicable ante la presencia de ruido, este hace que la variable de control tome valores máximos cuando la pendiente del ruido entra como señal de error.
- Es necesario entonces filtrar la señal ruidosa dejando pasar solo las frecuencias de señal que corresponden a la misma, eliminando o atenuando el ruido (Wikipedia, Control PID, conceptos, 2010).

2.1.6. Control Proporcional - Derivativo (PD).

La acción de control proporcional derivativa genera una señal resultante de la combinación de la acción proporcional y la acción derivativa de la señal del error. Así pues un PD en esencia un control anticipatorio. Tenemos.

$$u_{pd}(t) = k_p \cdot e(t) + k_d \frac{de(t)}{dt}$$

Pero la constante k_d se la representa por el tiempo de acción derivada en minutos de anticipo y en términos de la ganancia proporcional k_p como se muestra.

$$k_d = k_p \cdot T_d$$

Con esta definición y aplicando la transformada de Laplace con condiciones iniciales nulas se obtiene:

$$U_{pd}(s) = K_p \cdot (1 + T_d \cdot s) \cdot E(s)$$

Donde la función de transferencia de un controlador proporcional derivativo es:

$$G_{pd}(s) = \frac{U_{pd}(s)}{E(s)} = K_p \cdot (1 + T_d \cdot s)$$

Este tendrá un efecto en el error en estado estable sólo si el error varía con respecto al tiempo, además que es un filtro pasa altas.

Así, en la siguiente figura se muestra el esquema de controlador con acción Proporcional Derivativa.

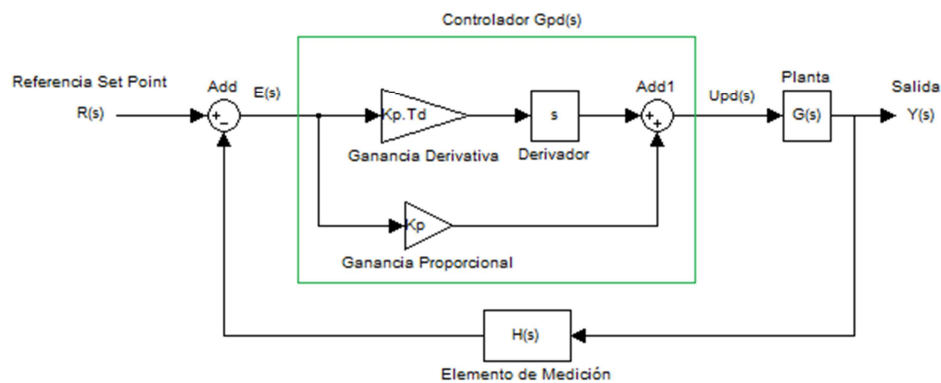


Figura 2. 11 Representación de un Sistema realimentado con Controlador Proporcional Derivativo

El control proporcional derivativo proporciona al sistema una mayor estabilidad relativa, teniendo especialmente en la respuesta transitoria, un menor sobre impulso (Spartacus Gomáriz, Diseño de sistemas de control de tiempo continuo y discreto, 1998), por lo que este controlador es más usado cuando se quiere garantizar la exactitud estacionaria. Sin embargo el control Proporcional derivativo tiene las siguientes desventajas:

- Dado que la respuesta $u(t)$ depende de la primera derivada del error, los ruidos en la señal hacen que $e(t)$ no sea una función suave y por tanto haciendo que $u(t)$ fluctúe considerablemente, saturando al actuador.
- Dado que la señal de control está en función de la derivada del error respecto al tiempo, éste controlador acentúa el ruido a altas frecuencias.

2.1.7. Control Proporcional –Integral - Derivativo (PID).

Los controladores tipo PID, son por mucho los de uso más frecuente en la industria de control de procesos abarcando casi el 95% de las aplicaciones (Moreno, 2003). Debido a su adaptabilidad, pocos requisitos y capacidad de resolución de todo tipo de problemas.

En el control PID, la acción proporcional es en si la más importante, sin embargo esta no corrige el error en estado estacionario. En cambio la acción integral corrige este error, y si es bien configurada se obtiene un error nulo en estado estable. Finalmente la acción derivativa, se tiene un efecto anticipativo lo cual se ve reflejado en la velocidad de reacción (KatsuhikoOgatta, 1998)

En cuanto a la representación matemática, se tiene que:

$$u_{pid}(t) = k_p \cdot e(t) + \frac{k_p}{T_i} \int_0^t e(\tau) \cdot d\tau + k_p \cdot T_d \frac{de(t)}{dt}$$

Con esta definición y aplicando la transformada de Laplace con condiciones iniciales nulas se obtiene:

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

Donde la función de transferencia de un controlador proporcional integral derivativo es:

$$G_{pid}(s) = \frac{U_{pid}(s)}{E(s)} = K_p \cdot \left(1 + \frac{1}{T_i \cdot s} + T_d \cdot s \right)$$

Así, en la siguiente figura se muestra el esquema de este controlador.

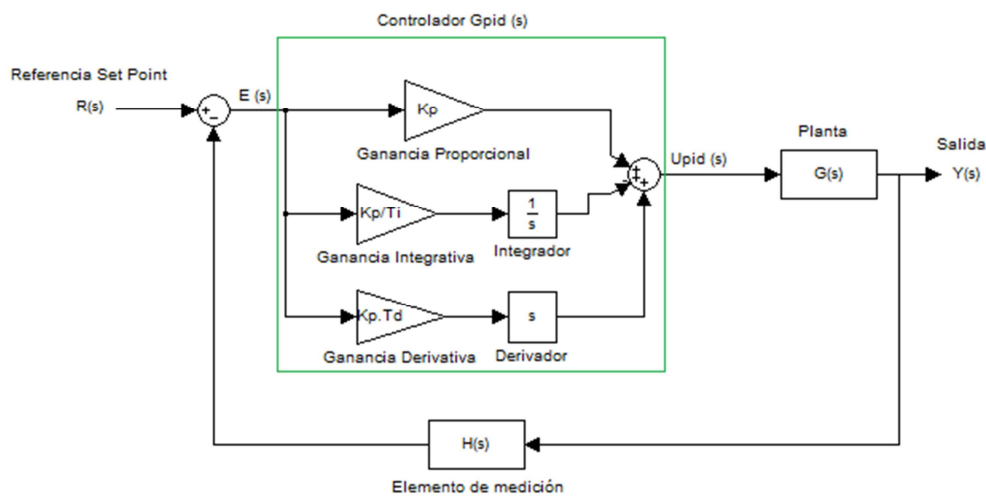


Figura 2. 12 Representación de un Sistema realimentado con Controlador Proporcional Integral Derivativo

En resumen hemos visto diferentes tipos de controladores clásicos utilizados en la mayoría de aplicaciones de control, pero está claro que la nueva tendencia es migrar hacia técnicas de control inteligente, las cuales son mucho más adaptables a sistemas complejos los cuales requieran diferentes consideraciones en cuanto a estados, condiciones de trabajo y número de entradas/salidas, además en el caso de la lógica difusa, esta no necesita de un

modelo matemático base para el diseño del controlador, entonces, por facilidad, versatilidad y eficiencia, este trabajo utilizará conceptos de lógica difusa para el diseño del controlador regulatorio.

2.2. Introducción a los Sistemas de Control Difusos.

La lógica difusa es una técnica de control inteligente que nos permite trabajar sobre un proceso o planta teniendo información no tan precisa, justamente en esto se diferencia de la lógica aplicada en el control convencional (L.A, 1965), el cual trabaja con información bien definida y bastante precisa, dado que haciendo una semejanza con el ser humano, este no necesita información numérica precisa del medio que lo rodea para desarrollar tareas de control como por ejemplo conducir un automóvil o el simple hecho de caminar entre una multitud y no chocarse con alguien.

Si los controladores realimentados convencionales, se pudieran programar para aceptar entradas imprecisas y con ruido, sin duda podrían trabajar de una manera más eficiente y se los podría implementar más fácilmente.

Refiriéndonos a los conjuntos, la lógica de control convencional maneja los conjuntos clásicos, que permiten solo dos opciones, la pertenencia o la no pertenencia de un elemento a dicho conjunto, en cambio la lógica difusa maneja los llamados conjuntos difusos, que permiten procesar la información permitiendo manejar un grado de pertenencia a dichos conjuntos difusos. Por esta razón, la lógica difusa se adapta mejor al mundo real (Kevin M. Passino, 1998), incluso puede funcionar teniendo como principio nuestras expresiones o respuesta ante determinado caso. Estas expresiones conforman una base de reglas en la cual se basará el comportamiento de nuestro controlador.

La lógica difusa es una metodología que proporciona una manera simple de obtener una conclusión a partir de información de entrada vaga, ambigua, imprecisa, con ruido o hasta incompleta, en general la lógica difusa reproduce el comportamiento de una persona al momento de tomar decisiones basadas en información con las características antes mencionadas.

2.2.1. Conjuntos Difusos.

Se puede decir que para entender la lógica difusa se debe primero entender los conjuntos difusos. Para definir estos conjuntos, llamemos a un conjunto difuso A , y llamaremos X a un elemento que puede tener este conjunto, la característica especial de estos es que pueden contener elementos de forma parcial, por lo que la propiedad $X \in A$, está caracterizada por una función de pertenencia que llamaremos $u_A(X)$, tal que $0 \leq u_A(X) \leq 1$, pudiendo u_A tomar valores entre 0 y 1, siendo 0 la no pertenencia y 1 la pertenencia total del elemento al conjunto difuso (Passino, Fuzzy Control: The Basics, 1998).

Se define el llamado universo de discurso U , que no es más que el conjunto de posibles valores que pueden tomar los elementos X pertenecientes a los conjuntos difusos.

A modo de ejemplo, supongamos que se desea representar con conjuntos difusos la variable edad de una persona adulta, en este caso el universo de discurso será el rango de posibles valores de la edad de las personas adultas, se escoge el rango entre 18 años a 78 años, además se entiende que valores fuera de este rango pueden darse pero son escasos, entonces tenemos el universo de discurso $U = [18, 78]$.

Para denominar los conjuntos difusos se suelen utilizar etiquetas lingüísticas, como las que se usan diariamente para representar algo, como por ejemplo:

- Muy joven (MJ).
- Joven (J).
- Media (M).
- Viejo (V).
- Muy viejo (MV).

Definiendo y tabulando los rangos:

Etiqueta	Rango [min, máx]
Lingüística	
MJ	[18, 38]
J	[28, 48]
M	[38, 58]
V	[48, 68]
MV	[58, 78]

Tabla 2. 1 Conjuntos difusos para la edad de una persona adulta

En la figura 2.13 se muestran gráficamente los conjuntos difusos.

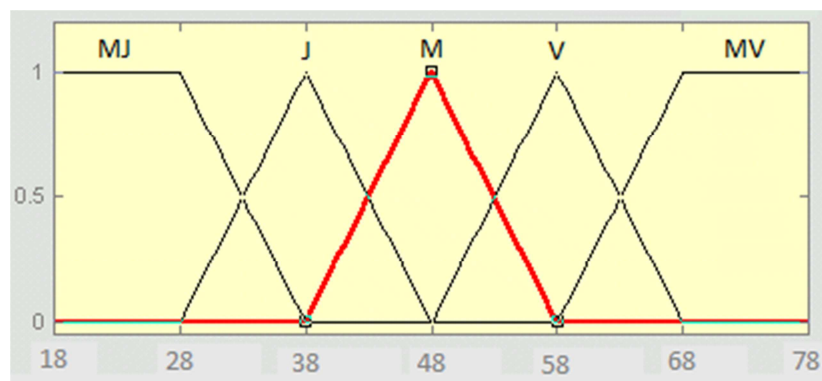


Figura 2. 13 Conjuntos difusos para la edad de una persona adulta

2.2.2. Funciones de Pertenencia.

O también llamada función característica, nos indica el grado en que cada elemento del universo de discurso, pertenece a dicho conjunto. Es decir, la función de pertenencia de un conjunto difuso llamado A sobre un elemento X perteneciente a un universo de discurso llamado U es de la forma $u_A: X \rightarrow [0,1]$, donde $u_A(X) = r$ si r es el grado en que X pertenece a A.

En los conjuntos clásicos, su función de pertenencia tomará dos posibles valores de 0 o 1, mientras que en los conjuntos difusos la función de pertenencia tomará valores dentro del rango $0 \leq u_A(X) \leq 1$.

Al momento de determinar una función de pertenencia, normalmente se eligen funciones sencillas, esto con el objetivo de que los cálculos no sean complicados. En la mayoría de los casos y para diferentes sistemas a ser controlados, se utilizan mayormente las funciones triangulares y las trapezoidales:

A continuación se muestran algunas de las funciones de pertenencia lineales mayormente usadas.

Función Triangular:

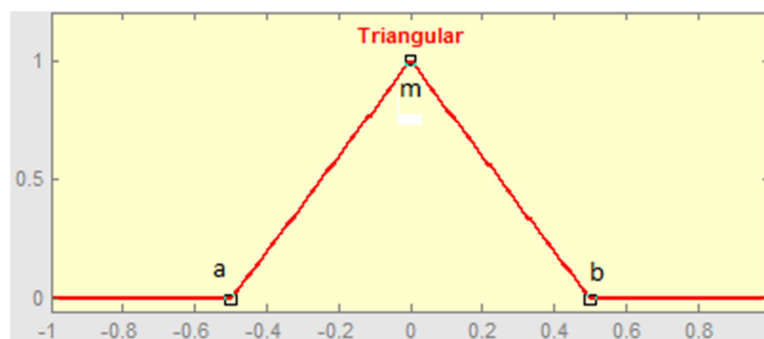


Figura 2. 14 Función de pertenencia triangular

Definida mediante el límite inferior a , el superior b y el valor modal m , tal que $a < m < b$. La función no tiene porqué ser simétrica. La evaluación para cualquier punto comprendido entre a y b se muestra en la siguiente tabla 2.2.

Valor	Condición
0	si $x \leq a$
$(x-a)/(m-a)$	si $a < x \leq m$
$(b-x)/(b-m)$	si $m < x < b$
0	si $x \geq b$

Tabla 2. 2 Valores de Función de pertenencia triangular

Función Trapezoidal:

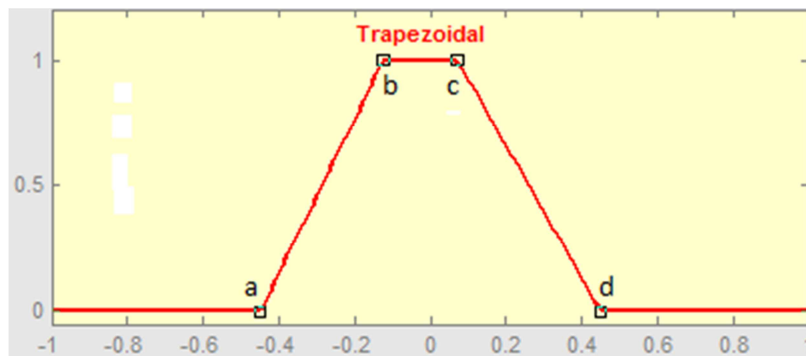


Figura 2. 15 Función de pertenencia trapezoidal

Definida por sus límites inferior a, superior d, y los límites de soporte inferior b y superior c, tal que $a < b < c < d$. En este caso, si los valores de b y c son iguales, se obtiene una función triangular.

Valor	Condición
0	si $(x < a)$ o $(x > d)$
$(x-a)/(b-a)$	si $a \leq x \leq b$
1	si $b \leq x \leq c$
$(d-x)/(d-c)$	si $c \leq x \leq d$

Tabla 2. 3 Valores de Función de pertenencia trapezoidal

A continuación se muestran algunas de las funciones de pertenencia no lineales:

Función Gaussiana:

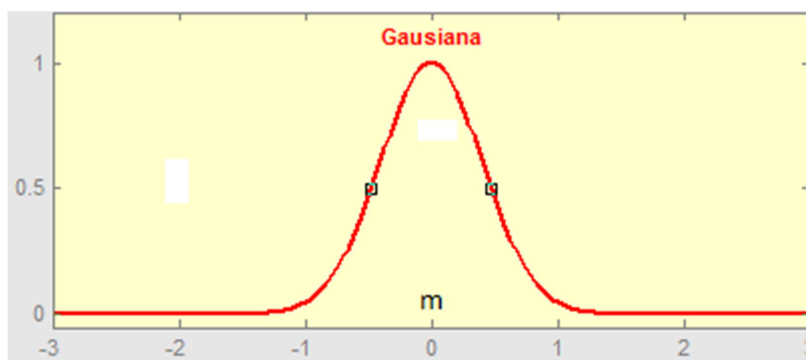


Figura 2. 16 Función de pertenencia Gaussiana

Viene definida por su valor medio m y una desviación estándar $k > 0$. Se cumple que cuanto menor es k , más estrecha es la “campana”

La función de pertenencia toma los valores según la siguiente ecuación:

$$u_A(X) = e^{-\frac{(x-m)^2}{2k^2}}$$

Función Gamma:

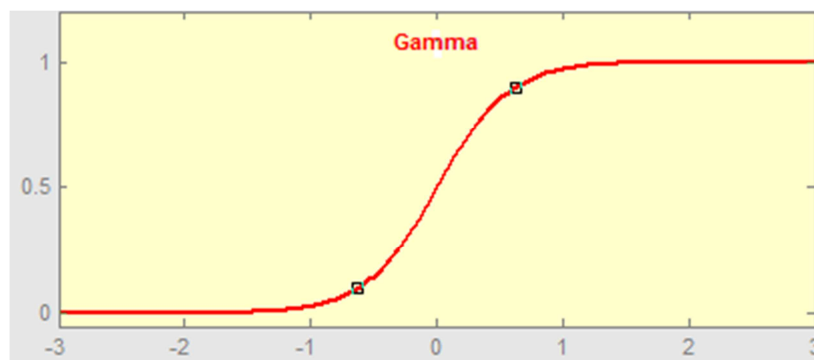
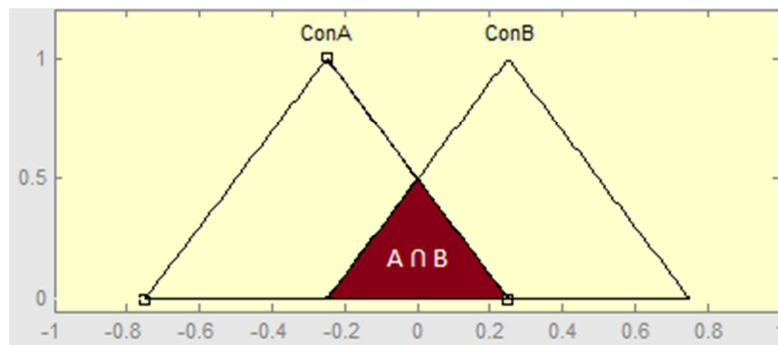


Figura 2. 17 Función de pertenencia Gamma

Definida por su límite inferior a y el valor $k > 0$. Esta función se caracteriza por un rápido crecimiento a partir de a ; cuanto mayor es el valor de k , el crecimiento es más rápido. Se usa para calificar valores lingüísticos extremos. Nunca toma el valor $u_A(X) = 1$, aunque tienen una asíntota horizontal en dicho valor.

2.2.3. Operaciones con conjuntos difusos.

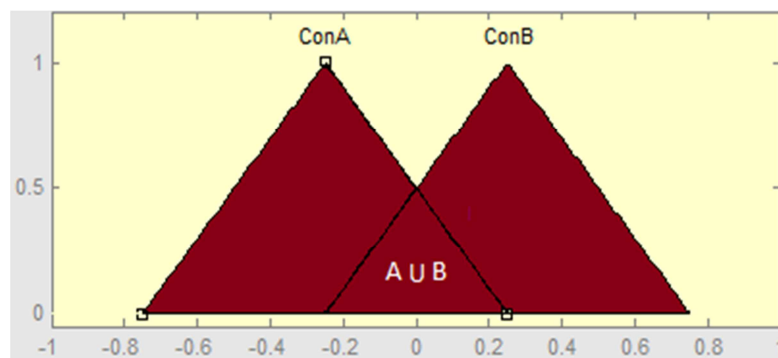
De la misma manera que se realizan operaciones con los conjuntos clásicos, se definen operaciones para los conjuntos difusos, entre las cuales tenemos:

Intersección:**Figura 2. 18 Intersección de Conjuntos**

El conjunto intersección de dos conjuntos A y B, se define como los elementos que están en el conjunto A y en el conjunto B; con esto, se puede decir que la intersección entre conjuntos se puede entender como una operación tipo AND entre los mismos. Esto en los conjuntos clásicos, En los conjuntos difusos la unión se define como el valor mínimo entre los dos valores de pertenencia de los elementos, como se muestra en la siguiente expresión.

$$A \cap B = \{\text{MIN}(p_A(x), p_B(x))\}$$

Donde, $p_A(x)$ y $p_B(x)$ son los grados de pertenencia del elemento x en el conjunto A y B, respectivamente.

Unión:**Figura 2. 19 Unión de Conjuntos**

El conjunto unión de dos conjuntos A y B, se define como los elementos que están en el conjunto A o en el conjunto B. Se puede decir que la unión entre conjuntos se puede entender como una operación tipo OR entre los mismos. En los conjuntos difusos la unión

se define como el valor máximo entre los dos valores de pertenencia de los elementos, como se muestra en la siguiente expresión.

$$A \cap B = \{\text{MAX} (pA(x), pB(x))\}$$

Complemento:

El complemento en los conjuntos difusos tiene que ver con el valor de pertenencia de un elemento a dicho conjunto, se habla del complemento como el conjunto formado por los valores de pertenencias que le permitirían al conjunto obtener el valor máximo de pertenencia posible, siendo 1 el valor máximo de pertenencia que un conjunto difuso puede suministrar, este conjunto se podría formar restando le a 1 los valores de pertenencia del conjunto difuso al que se desea encontrar el complemento.

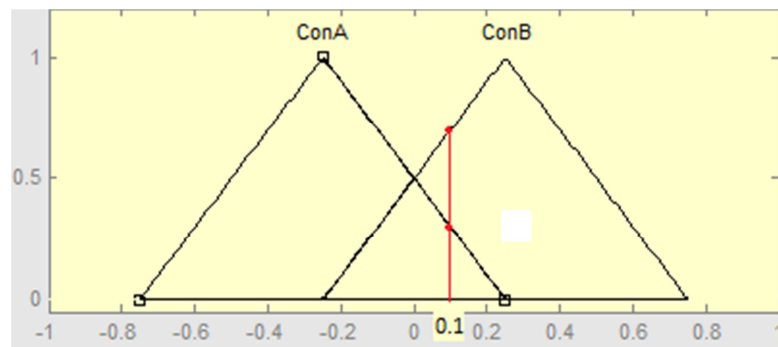


Figura 2. 20 Complemento de Conjuntos

En la gráfica anterior, se establece que el valor de pertenencia de $X = 0.1$ para el conjunto A es de 0.25, por lo que el valor de pertenencia de este valor al complemento del conjunto A es 0.75. Del mismo modo para el conjunto B, el valor de pertenencia de $X = 0.1$ para este conjunto es de 0.75, por lo que el valor de pertenencia de este valor al complemento del conjunto B es 0.25.

2.3. Sistemas Difusos.

En el desarrollo de este trabajo se utilizarán los conceptos de lógica difusa en el desarrollo de un controlador experto para el control regulatorio (refiérase al capítulo 4.1.7.4.1). Los sistemas de lógica difusa tienen una estrecha relación con los conceptos difusos tales como funciones de pertenencia, conjuntos difusos, variables lingüísticas y

demás (Passino, Fuzzy Control: The Basics, 1998) . En general, un sistema difuso cuenta con cuatro módulos como se muestran en la figura 2.21.

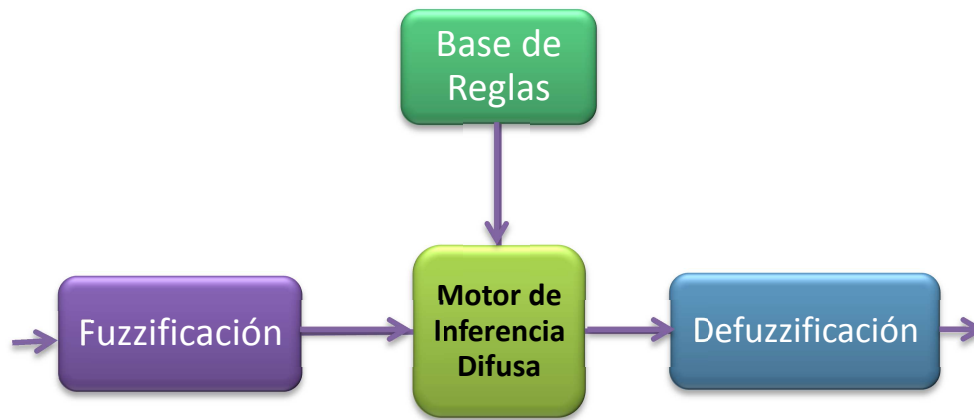


Figura 2. 21 Partes de un sistema difuso

Los más importantes sistemas de lógica difusa son de los siguientes tipos: sistemas difusos tipo Mamdani y sistemas difusos tipo Takagi-Sugeno.

2.3.1. Tipo Mamdani.

El método de Mamdani es el más usado en aplicaciones, dado que tiene una estructura muy simple, aunque el costo computacional es relativamente grande. La configuración básica de un sistema tipo Mamdani es la estructura mostrada en la siguiente figura.

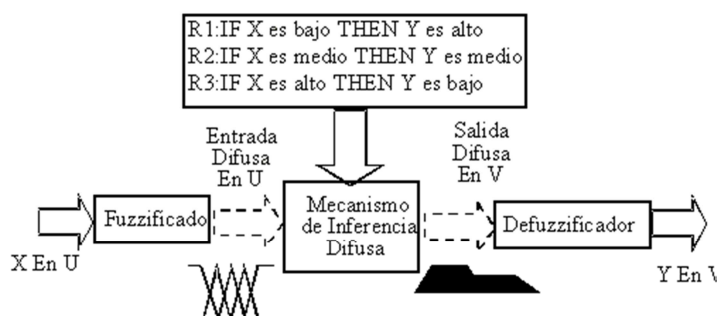


Figura 2. 22 Procesamiento general de Sistema difuso tipo Mamdani

De este modo desglosamos las partes que lo conforman:

Fuzzificador:

Se tiene que la entrada de un sistema de lógica difusa tipo Mamdani es normalmente un valor numérico X perteneciente al universo de discurso U , proveniente por ejemplo, de un sensor. Para que este valor pueda ser procesado por el sistema difuso se hace necesario convertirlo a un lenguaje con el que el motor de inferencia pueda trabajar. Este lenguaje no es más que valores difusos referentes a los niveles de pertenencia de los valores de entrada a los diferentes conjuntos difusos, en los cuales se ha dividido el universo de discurso de las variables de entrada al sistema.

Si a la entrada del sistema se tiene más de una variable, se tiene que aplicar algún operador por ejemplo un OR o un AND, para que con los dos valores de cada variable se obtenga un único valor de pertenencia. Este valor difuso posteriormente es enviado al motor de inferencia difusa para su procesamiento.

Motor de Inferencia Difusa:

El motor de inferencia difusa trabaja en conjunto con la base de reglas, teniendo los diferentes niveles o valores de pertenencia arrojados por el fuzzificador, los mismos deben ser procesados para generar una salida difusa. Es decir, la tarea del motor de inferencia es tomar los niveles de pertenencia y apoyado en la base de reglas, generar la salida del sistema difuso.

La base de reglas es la manera que tiene el sistema difuso de guardar el conocimiento lingüístico del experto que le permite resolver el problema de control. Estas reglas son del tipo IF-THEN.

Base de Reglas:

Almacena las reglas obtenidas de expertos o del aprendizaje. Una regla de la base de reglas o base de conocimiento tiene dos partes, el antecedente y la conclusión como se observa a continuación.

IF *X es baja* THEN *Y es alta*

Donde el antecedente sería la condición de que la entrada X es baja, y la conclusión es la respectiva salida Y alta. En un sistema difuso tipo Mamdani tanto el antecedente como el consecuente de las reglas están dados por expresiones lingüísticas. A continuación se muestra un ejemplo de bases de reglas usando el operador de intersección entre dos variables, el error y el cambio del error en un sistema de péndulo invertido. Primero se tienen las declaraciones de las variables lingüísticas de los conjuntos difusos:

Conjunto difuso	Símbolo
Grande Negativo	GN
Pequeño Negativo	PN
Cero	Z
Pequeño Positivo	PP
Grande Positivo	GP

Tabla 2. 4 Ejemplo de Variables Lingüísticas para los conjuntos difusos.

La base de reglas se muestran en la siguiente tabla, donde las variables lingüísticas de las celdas centrales, denotan el conjunto difuso referente a la variable de salida del motor de inferencia difusa, en función del error y del cambio del error.

		Cambio del error					
		GN	PN	Z	PP	GP	
Error	GN	GP	GP	GP	PP	Z	
	PN	GP	GP	PP	Z	PN	
	Z	GP	PP	Z	PN	GN	
	PP	PP	Z	PN	GN	GN	
	GP	Z	PN	GN	GN	GN	

Tabla 2. 5 Ejemplo de la base de reglas.

Por ejemplo, si el error es pequeño negativo y el cambio del error es pequeño positivo, entonces la salida pertenece al conjunto difuso cero. Además se define el valor de pertenencia de la variable salida a este conjunto.

Defuzzificador:

Convierte el valor de pertenencia del conjunto difuso referente a la salida, obtenido por el motor de inferencia difusa, en un valor numérico o también llamado crisp. Para lograr que la salida del sistema difuso pueda ser interpretada por elementos que solo procesen información numérica, como por ejemplo un microcontrolador, hay que convertir la salida difusa del mecanismo de inferencia; este proceso lo realiza el defuzzificador.

Para generar la salida numérica a partir del valor de pertenencia de un conjunto difuso resultante a un valor numérico clásico análogo o digital, existen varias opciones, entre ellas las más comunes son:

- El Centro del área (CdA).
- Centro de Gravedad (CdG).
- Media del máximo (MdM).
- Bisector del área (BdA).

Sin embargo, el método que requiere menor procesamiento y es ampliamente utilizado es el método de Centro de Gravedad. La siguiente fórmula es la expresión matemática para este método de defuzzificación.

$$Y = \frac{\sum_{i=0}^n u(z)_i \cdot z_i}{\sum_{i=0}^n u(z)_i}$$

Donde,

$u(z)_i$ = valor de pertenencia.

z_i = salida correspondiente al valor de pertenencia de entrada.

La principal desventaja de este modelo difuso es que para sistemas complejos no lineales usualmente se requieren muchas reglas difusas IF-THEN, lo cual hace más compleja su implementación.

2.3.2. Tipo Takagi-Sugeno.

Esta es otra alternativa de procesamiento para sistemas difusos, este sistema difuso funcional fue propuesto por Takagi-Sugeno en 1985. Su estructura básica se muestra a continuación.

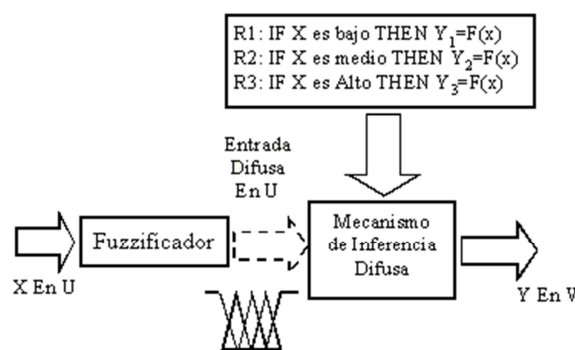


Figura 2. 23 Procesamiento general de Sistema difuso tipo Takagi-Sugeno

El fuzzificador y el motor de inferencia difusa realizan la misma función que en los sistemas tipo Mamdani, sin embargo la base de reglas es diferente. Además se caracteriza porque este no tiene un mecanismo de defuzzificación.

Las reglas de la base de conocimiento de un sistema Takagi-Sugeno son diferentes a las de los sistemas Mamdani, pues el consecuente de estas reglas ya no es una simple etiqueta lingüística, sino que es una función de la entrada que tenga el sistema en ese momento, esto se muestra a continuación.

IF X es *baja* THEN $Y = F(\text{entrada})$

En los sistemas difusos tipo Takagi-Sugeno, los valores que arrojan los consecuentes de las reglas que se han activado en un momento determinado ya son valores numéricos por lo que no se necesita un mecanismo de defuzzificación.

Entonces para calcular la salida numérica del sistema difuso, se tiene que es la media ponderada de todas las salidas de las reglas multiplicada por el peso de cada regla, o también llamado método de centro de gravedad para singletons COGS ((IEC), 1997). Para un sistema con n reglas la salida es expresada de la siguiente forma.

$$Y = \frac{\sum_{i=1}^n w_i \cdot y_i}{\sum_{i=1}^n w_i}$$

Donde n es el número de reglas, y_i son las funciones que permiten calcular el consecuente y w_i son los pesos de cada regla difusa, los que están en función del valor de pertenencia de cada entrada a un conjunto difuso. En la siguiente figura se muestra este procedimiento con mayor claridad.

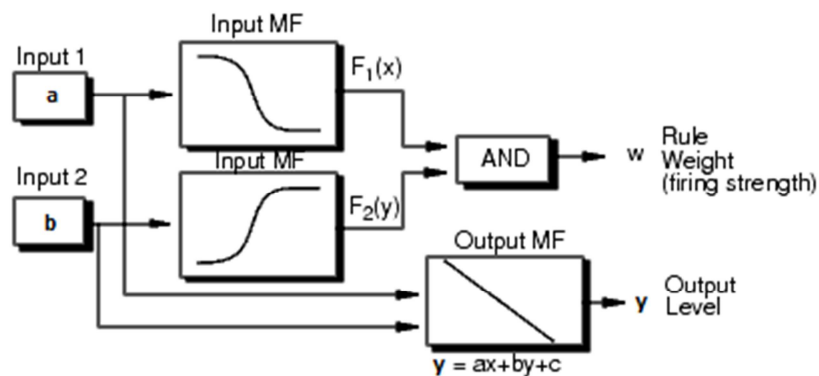


Figura 2. 24 Ejemplo de operación de un sistema Takagi-Sugeno

En este sistema se tienen dos entradas (a y b), en primer lugar se realiza la fuzzificación, donde para cada entrada y mediante la función de membresía se genera un nivel de pertenencia referente a un conjunto difuso, se aplica un operador AND con lo cual se obtiene el peso de la regla (w). Posteriormente se genera la salida numérica (y), la cual está en función de las variables de entrada, en el caso que las funciones de pertenencia de las salidas sean singletons, (y) toma el valor del singleton de salida específico para cada iteración. Finalmente se pondera la multiplicación de estos valores ($w \cdot y$) obteniendo así la variable numérica de salida.

La principal ventaja de este modelo difuso es su poderosa capacidad de representar relaciones complejas y no lineales con un pequeño número de reglas difusas IF-THEN. En resumen, los consecuentes en el modelo de Mamdani, con reglas difusas IF-THEN, son remplazados por consecuentes en forma funciones en base a las entradas.

CAPÍTULO 3

DISEÑO DEL HARDWARE

3.1. Elementos del Sistema de Control.

En este capítulo se revisan los diferentes elementos que conforman el hardware del sistema de control, como se muestra en la figura 3.1, entre los cuales tenemos los actuadores lineales, el microcontrolador, los sensores y finalmente la elaboración de la tarjeta de adquisición y control.

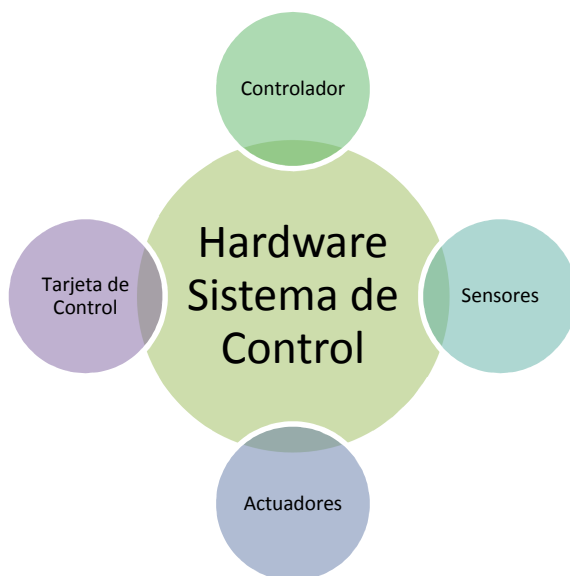


Figura 3. 1 Elementos del Hardware

3.2. Actuadores Lineales.

Como se sabe, el sistema debe empujar los tres pedales del automóvil, para eso se utilizará tres actuadores lineales de precisión, cuyas características se muestran en la tabla 3.1.

PEDALES	CARACTERÍSTICAS	VALORES
Acelerador	Alimentación	12 Vdc
	Velocidad	1,51 cm/seg
y Embrague	Carrera	7,62 cm
	Capacidad de Carga	400 lbs
	Corriente a máxima carga	4,5 Amps
	Temperatura Operacional	-26 a 65 °C
	Protección	IP54
	Nivel de ruido	db<45
Freno	Alimentación	12 Vdc
	Velocidad	2,40 cm/seg
	Carrera	7,62 cm
	Capacidad de Carga	200 lbs
	Corriente a máxima carga	4,5 Amps
	Temperatura Operacional	-26 a 65 °C
	Protección	IP54
	Nivel de ruido	db<45

Tabla 3. 1 Características de Actuadores Lineales

Nótese que para los pedales del acelerador y embrague se utilizarán actuadores lineales con una velocidad nominal de 1,28 cm/seg y una fuerza máxima de 400 lbs, en estos pedales la velocidad no es tan esencial como la precisión y la fuerza. En cambio en el pedal del freno, si resulta indispensable que el actuador lineal tenga la suficiente velocidad para casos súbitos de frenado, en este caso se cuenta con un actuador lineal de casi el doble de velocidad que los del acelerador y embrague (2,40 cm/seg), dispone de la misma carrera de 7,62 cm pero con la mitad de la fuerza (200 lbs). La apariencia física de los actuadores se muestra en la siguiente figura.

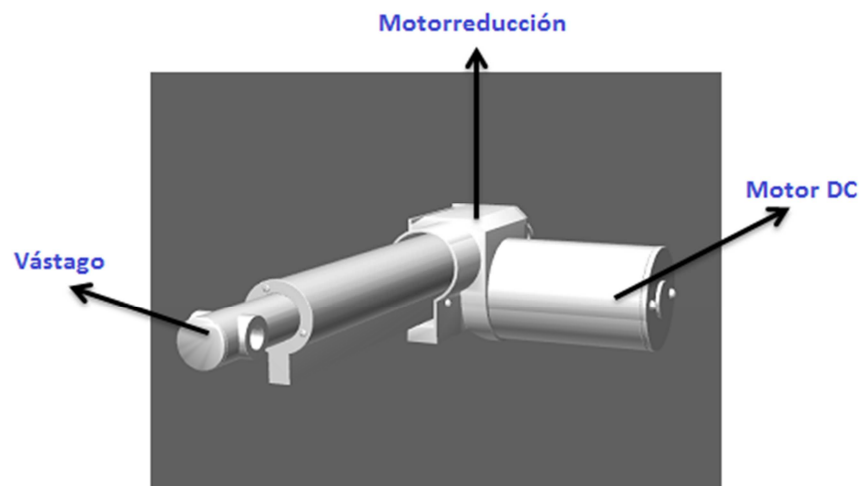


Figura 3. 2 Actuador Lineal de Precisión.

Cada actuador lineal consta básicamente de tres partes:

- **Motor DC:** Es el que convierte la energía eléctrica en energía mecánica, provocando un movimiento rotatorio. Este puede funcionar en dos sentidos.
- **Motorreducción:** Es un juego de engranajes configurados de tal forma que se encargan de disminuir la velocidad (rpm) de entrada, obteniendo a la salida menor velocidad pero con un aumento proporcional de fuerza, es decir, mientras se reduce la velocidad, la fuerza y torque se aumentan. Además transforma el movimiento rotatorio del eje del motor DC en un movimiento lineal aplicado sobre el vástago.
- **Vástago:** Es una barra, en este caso metálica, que transmite el movimiento hacia los pedales.

3.2.1. Pruebas de Funcionamiento.

3.2.1.1. Pruebas de Velocidad.

En primer lugar se realizaron pruebas de velocidad, con una alimentación de 11,3 Vdc, donde a plena carga y sin carga se probaron los dos tipos de actuadores, el de 200 lbs y el de 400 lbs, obteniendo los resultados mostrados en la tabla 3.2.

Tipo de Actuador (lbs)	Sin carga (cm/seg)	Con carga 200 lbs (cm/seg)
200	2,39	1,18
400	1,49	1,16

Tabla 3. 2 Respuesta de velocidad de actuadores lineales.

Se puede ver una ligera variación de la velocidad obtenida experimentalmente de la mostrada en los datos del fabricante, esto se debe a que para las pruebas y por posterior implementación en la tarjeta de control, se alimentó al actuador con un diodo de silicio en serie de 6 amperios, el cual tiene una caída de voltaje de aproximadamente 0,7 voltios con lo cual se reduce el voltaje de entrada a 11,3 voltios y con esto también la potencia.

3.2.1.2. Pruebas de Consumo de Corriente.

Respecto a las pruebas de consumo de corriente en función a la carga, primero se probó el actuador de 200 lbs de fuerza, con los siguientes resultados

Tipo Actuador (Lbs)	Carga (Lbs)	Corriente (A)
200	0	1,01
	25	1,78
	50	2,43
	75	2,93
	100	3,23
	125	3,43
	150	3,67
	175	3,74
	200	3,87

Tabla 3. 3 Respuesta en corriente, actuador 200 lbs.

Tenemos una corriente pico de casi 3,9 amperios a máxima carga, la curva de respuesta se la puede ver en la siguiente gráfica.

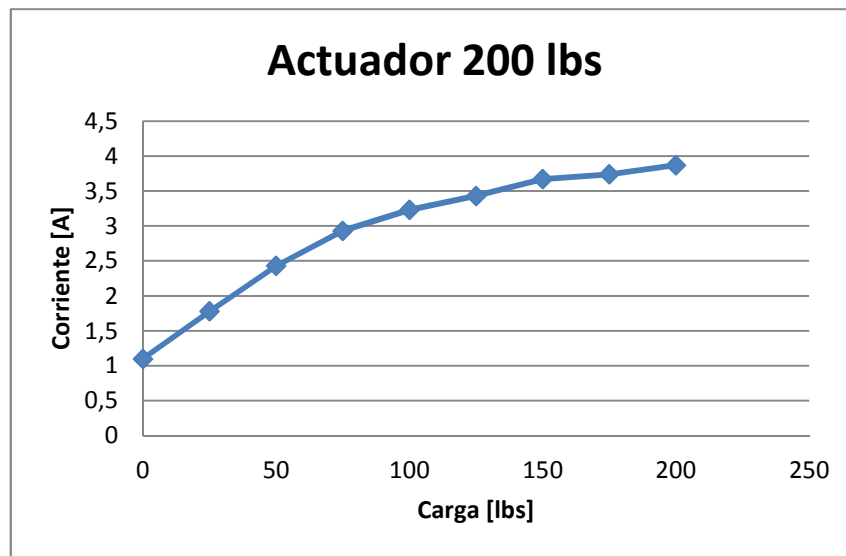


Figura 3. 3 Curva respuesta en corriente, actuador 200 lbs.

Con el actuador de 400 lbs de fuerza se tiene el consumo de corriente:

Tipo Actuador (Lbs)	Carga (Lbs)	Corriente (A)
400	0	0,3
	25	0,47
	50	0,65
	75	0,8
	100	0,91
	125	1,03
	150	1,12
	175	1,23
	200	1,33

Tabla 3. 4 Respuesta en corriente, actuador 400 lbs.

Se puede ver que el actuador de 400 lbs, consume menor corriente que el de 200 lbs, esto se debe a que el bloque de motorreducción aumenta la fuerza y reduce la velocidad, brindando así un menor consumo de corriente respecto a la fuerza de carga, aunque se sacrifica la velocidad.

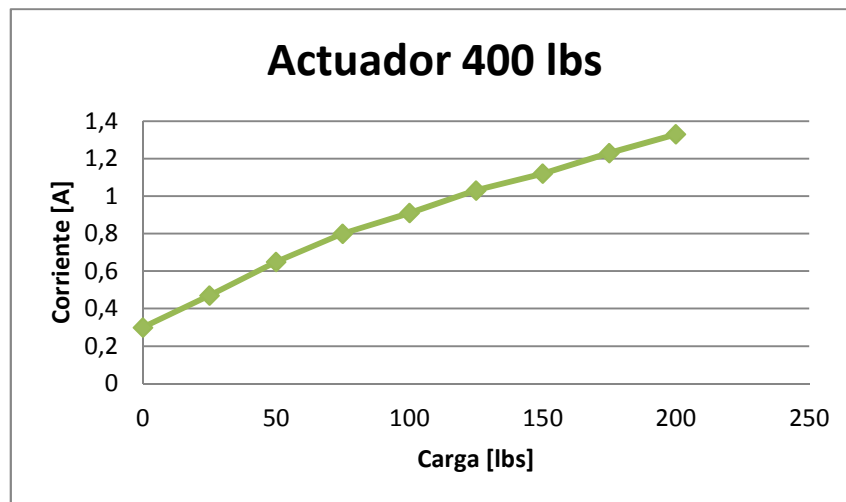


Figura 3. 4 Curva respuesta en corriente, actuador 400 lbs.

3.3. Microcontrolador PIC18F4550.

3.3.1. Generalidades.

El PIC18F4550, es un microcontrolador de 8 bits con gran variedad de número de pines y puertos de E/S, brinda prestaciones de gama media-alta. Es de la familia de los Microcontroladores tipo RISC (Computador de set de instrucciones reducidas), el cual ejecuta una instrucción cada 4 ciclos del oscilador, estos son fabricados por Microchip Technology Inc.

El PIC utiliza un juego de instrucciones tipo RISC, las cuales se clasifican de la siguiente manera:

- Las que realizan operaciones entre el acumulador y una constante.
- Las que realizan operaciones entre el acumulador y una posición de memoria.
- Instrucciones de condicionamiento y de salto/retorno.
- Implementación de Interrupciones.

Microchip facilita un entorno de desarrollo y programación llamado MPLAB, el cual incluye un simulador y un ensamblador el cual genera el código .HEX. Existen otros entornos de programación que utilizan lenguaje C y BASIC.

Uno de los lenguajes más utilizados para programación de PIC's es PICC de CCS, que utiliza lenguaje C. Este tiene un editor de texto y dispone de múltiples bibliotecas que se pueden usar para todos los módulos y funciones que presta el microcontrolador. En el desarrollo de este proyecto en particular, se utilizará el software PICC detallado en el capítulo siguiente.

3.3.2. Arquitectura.

Las dos partes fundamentales del microprocesador son la memoria y la CPU, la primera almacena los datos y el lugar donde estos se encuentran almacenados, la CPU en cambio se encarga de procesarlos. El PIC18F4550 tiene una arquitectura Harvard, esta posee dispositivos de almacenamiento separados físicamente, uno para los datos, y otro para las instrucciones, además de diferentes buses para el acceso a la memoria del programa y a la memoria de datos, típico de la arquitectura Harvard. Esto permite acceder simultáneamente a las dos memorias, es decir, se puede ejecutar una instrucción mientras se lee de la memoria del programa la siguiente instrucción o proceso Pipeline (scribd, Arquitectura Harvard en Microcontroladores, 2010).

3.3.3. Organización de Memoria.

El microcontrolador en estudio dispone de las siguientes memorias mostradas en la tabla 3.5.

Elemento	Función
Memoria de Programa	Memoria Flash de 32,768 bytes, almacena instrucciones, constantes y datos, puede ser escrita o leída externamente mediante un programador.
Memoria RAM de Datos	De tipo SRAM de 2048 bytes, la cual almacena datos de forma temporal mientras se ejecuta el programa, puede ser escrita o leída mediante instrucciones.

Memoria EEPROM de Datos	Es una memoria no volátil de 256 bytes, esta almacena datos aun si no está alimentada, puede ser escrita o leída en tiempo de ejecución mediante registros.
Pila	Es un bloque de 31 palabras de 21 bits cada una, la cual almacena la dirección de la instrucción que se debe ejecutar después del retorno de una interrupción o de una subrutina.
Memoria de Configuración	En esta se guardan los bits de configuración y los registros de identificación.

Tabla 3. 5 Organización de la Memoria PIC18F4550.

3.3.4. Interrupciones.

El microcontrolador tiene 20 diferentes interrupciones, diferenciadas en dos grupos.

- Grupo general de Interrupciones.
- Grupo de Interrupciones de Periféricos.

En la tabla 3.6 se listan las interrupciones típicas del microcontrolador:

Interrupción	Tipo
Timer 0 desbordamiento (RTCC name)	Interrupción General
Timer 0 desbordamiento (TIMER0 name)	Interrupción General
Timer 1 desbordamiento	Interrupción de Periféricos
Timer 2 desbordamiento	Interrupción de Periféricos
Timer 3 desbordamiento	Interrupción de Periféricos
Interrupción Externa #0	Interrupción General
Interrupción Externa #1	Interrupción General
Interrupción Externa #2	Interrupción General
Port B, cualquier cambio en pin B4-B7	Interrupción de Periféricos
Conversión completa de análogo a digital	Interrupción de Periféricos
Dato disponible en puerto RS-232	Interrupción de Periféricos
Buffer de transmisión vacío en puerto RS-	Interrupción de Periféricos

Actividad SPI o I2C	Interrupción de Periféricos
Capturar o Comparar en unidad 1	Interrupción de Periféricos
Capturar o Comparar en unidad 2	Interrupción de Periféricos
Colisión de Bus	Interrupción de Periféricos
Detección de bajo voltaje	Interrupción de Periféricos
Evento de Comparación	Interrupción de Periféricos
Escritura Completa	Interrupción de Periféricos
Error en Oscilador	Interrupción de Periféricos

Tabla 3. 6 Interrupciones en PIC18F4550.

3.3.5. Unidades Funcionales.

El PIC18F4550 dispone de una serie de Unidades Funcionales que le permiten (scribd, Sistema de interrupciones en el PIC18F4550, 2010):

- Realizar tareas específicas como por ejemplo la conversión análoga/digital, transmisión/recepción de datos, generación de señales digitales con temporización a base de los timers.
- Optimizar el rendimiento del microcontrolador, ya que estas unidades trabajan en paralelo y en conjunto a la CPU, permitiendo que ésta se centre en otras tareas como cálculos y procesamiento de datos.

Las unidades funcionales más importantes del PIC son las siguientes: puertos de entrada/salida, temporizadores, conversores ADC, unidad PWM, canales de comunicación, los cuales se revisan a continuación:

3.3.5.1. Puertos de E/S.

El uC dispone un total de cinco puertos digitales que se los puede configurar como entrada o salida en cualquier parte del programa. Como se ve en la tabla 3.7, estos suman 35 pines digitales (Microchip, I/O Ports, 2006).

Nombre de Puerto	Pines de Entrada/Salida	Número de Pines
PORTA	7 Pines E/S	2, 3, 4, 5, 6, 7
PORTB	8 Pines E/S	33, 34, 35, 36, 37, 38, 39, 40
PORTC	6 Pines E/S + 2 Pines de E	15, 16, 17, 18, 25, 26, 27, 23, 24
PORTD	8 Pines E/S	19, 20, 21, 22, 27, 28, 29, 30
PORTE	3 Pines E/S + 1 Pin de E	8, 9, 10, 11

Tabla 3. 7 Distribución de pines de E/S del PIC18F4550

Cada puerto de E/S tiene acceso a tres registros:

- Registro TRIS: Con este se puede configurar cada pin del puerto de entrada salida.
 - Bit = “1”, corresponde a una entrada.
 - Bit = “0”, corresponde a una salida.
- Registro PORT: Con este registro se puede leer el nivel lógico del pin de E/S y también se puede establecer su valor de salida.
- Registro LAT: mediante este registro se puede leer o establecer el valor del Latch de salida.

En la figura 3. Se muestra el diagrama de pines del microcontrolador PIC18F4550.

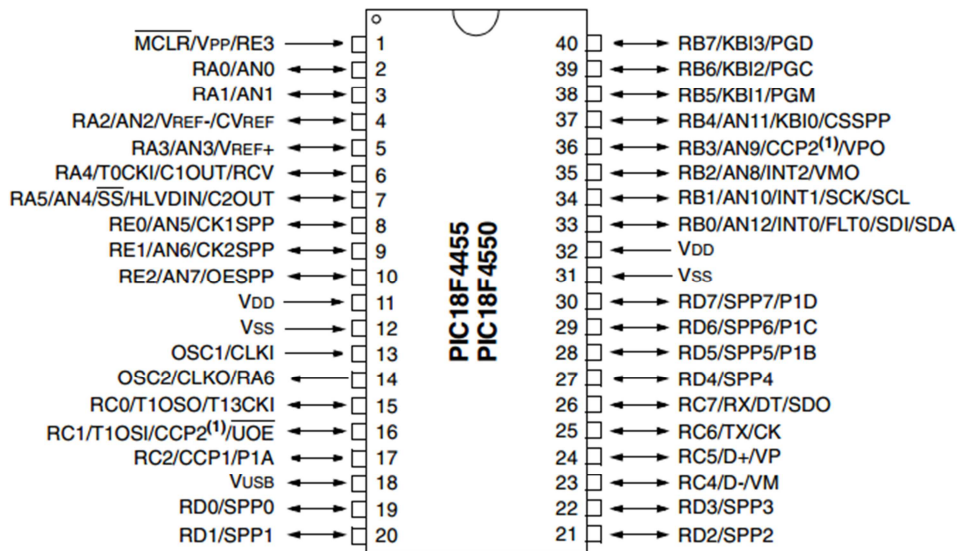


Figura 3. 5 Diagrama de pines PIC18F4550.

3.3.5.2. Temporizadores.

El uP dispone de cuatro temporizadores cuyas características se muestran en la siguiente tabla:

Temporizador	Características
TEMPORIZADOR 0	Configurable como temporizador/contador de 8/16 bits Pre-escalar de 8 bits programable Tiene una Interrupción por desbordamiento
TEMPORIZADOR 1	Configurable como temporizador/contador de 16 bits Dispone de un oscilador propio que puede funcionar como señal de reloj del temporizador 1, señal de reloj del uP en modo de bajo consumo. Pre-escalar de 3 bits programable Tiene una Interrupción por desbordamiento
TEMPORIZADOR 2	Configurable como temporizador/contador de 8 bits Se lo puede utilizar con los módulos CCP para generar señales PWM o con el módulo MSSP como señal de reloj del módulo SPI Pre-escalar de 2 bits programable Tiene una Interrupción por igualdad entre TMR2 y PR2
TEMPORIZADOR 3	Configurable como temporizador/contador de 16 bits

3	Fuente de reloj seleccionable, fuente interna o externa Tiene una Interrupción por desbordamiento
---	--

Tabla 3. 8 Características de temporizadores, PIC18F4550

El temporizador 2, se lo utilizará en conjunto con el módulo PWM para generar la señal cuadrada que servirá para obtener una salida análoga mediante un circuito integrador.

3.3.5.3. Conversores ADC.

El conversor Análogo Digital que dispone el PIC, tiene las siguientes características (Microchip, 10-bit analog-to-digital Converter (a/d) module, 2006):

- Tiene resolución configurable de 8 y 10 bits.
- 13 canales multiplexados.
- Señal de reloj de conversión configurable.
- Posibilidad de establecer el rango de voltaje de conversión mediante voltajes de referencia externos.

El rango de voltaje de conversión del ADC tiene por defecto un rango de 0 a 5 voltios, y dependiendo de la resolución, el valor digital puede tener valores de:

- 0 a 255 – con 8 bits de resolución.
- 0 a 1023 – con 10 bits de resolución.

Los pines del conversor análogo digital son 13, repartidos entre los puertos A, E y B según la siguiente tabla:

Conversor	Puerto	Pin
AN0	PORTA	2
AN1	PORTA	3
AN2	PORTA	4
AN3	PORTA	5
AN4	PORTA	7
AN5	PORTE	8

AN6	PORTE	9
AN7	PORTE	10
AN8	PORTB	35
AN9	PORTB	36
AN10	PORTB	34
AN11	PORTB	37
AN12	PORTB	33

Tabla 3. 9 Distribución de pines del conversor ADC del PIC18F4550.

3.3.5.4. Unidad de Captura, Comparación y PWM (CCP).

Las series de Microcontroladores PIC18F2455/ 2550/ 4455/ 4550, disponen de dos módulos CCP (CCP1, CCP2), cada módulo consta de un registro de 16 bits. El pin asignado al CCP2 puede ser cambiado de acuerdo a la configuración del dispositivo.

El módulo CCP utiliza los temporizadores 1, 2 y 3 dependiendo del módulo seleccionado. Temporizador 1 y 3 están disponibles para los módulos de captura y comparación, mientras que el Temporizador 2 está disponible para el módulo PWM.

En modo de captura, el CCPR (H y L) capturan el valor de los 16 bits del registro TMR1 cuando ocurre un evento en el pin CCP1 o CCP2. Los posibles eventos son:

- Flanco de bajada.
- Flanco de subida.
- 4 flancos de subida.
- 16 flancos de subida.

El modo PWM (modulación de ancho de pulso), entrega en los pines CCP1 (pin RC1) y CCP2 (pin RC2), ambos del puerto C, una señal periódica modificable en el ciclo de trabajo, es decir, se puede variar el tiempo en alto o Ton, modificando el tiempo en bajo o Toff de la señal. La siguiente figura indica la señal típica PWM.

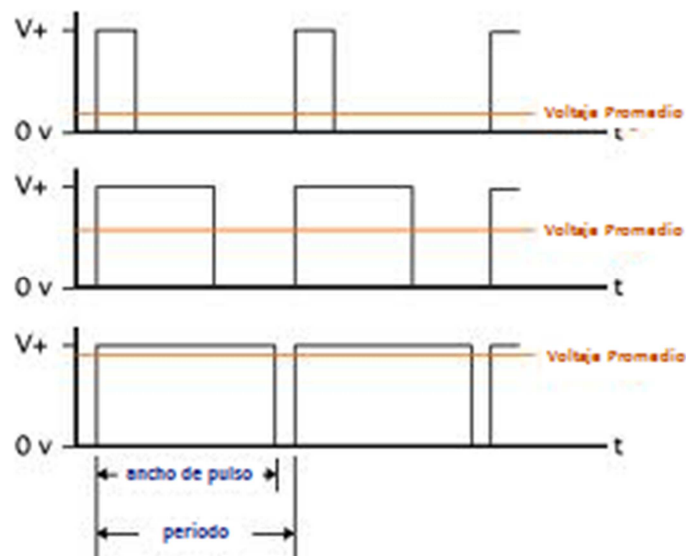


Figura 3. 6 Señal PWM típica.

Estas señales PWM serán utilizadas con ayuda de un circuito integrador y un seguidor de voltaje, con el propósito de obtener una salida de voltaje análoga la cual regulará la potencia de los actuadores lineales en el control regulatorio. Dado que en el PIC se tienen tan solo dos salidas PWM, la tercera salida necesaria se la obtiene de un PIC-18F2550, que mediante comunicación serial recibe el valor del PIC18F4550 referente a la tercera salida análoga (refiérase al capítulo IV, Configuración de Modulo A/D).

3.3.5.5. Canales de Comunicación.

El uC también dispone de varios canales de comunicaciones, como lo son (Microchip, UART, 2006):

- USART.
- Comunicación serie USB.
- Canal de comunicación I2C.
- Canal de transmisión de datos en paralelo.

A continuación se revisaran algunas características de estos módulos, especialmente el USART, USB e I2C.

USART:

El USART es un protocolo de comunicación serial, maneja varios modos de trabajo, entre ellos tenemos: Modo asíncrono de 8bits, asíncrono de 9bits, síncrono maestro, síncrono esclavo. Además tiene la propiedad de detectar automáticamente la velocidad de comunicación o también llamado BaudRate, activación automática por detección de datos recibidos.

Nos centraremos en el modo de transmisión TX/RX asíncrono. Donde los pines RC6 y RC7 son el transmisor Tx y el receptor Rx respectivamente.

En la recepción de la USART incorpora un registro de desplazamiento serie (RSR), donde los datos entran en serie por el pin RC7/RX y son muestreados por un bloque de lectura de bits. Una vez se han recibido todos los bits correspondientes a un byte el valor recibido se pasa en paralelo de RSR al registro RCREG. De esta forma el registro RSR queda listo para recibir un nuevo dato. Si durante el proceso de recepción se produce algún error de formato o de sobre escritura se pondrá a “1” el bit correspondiente (FERR/OERR).

La recepción de datos puede hacerse por dos métodos:

- Polling: En este caso el programa monitorea el estado del bit RCIF, siendo este el estado “1” para poder leer los datos guardados en el registro RCREG.
- Interrupción: Cada vez que se recibe un dato, el Flag se pone a “1”, habilitando la interrupción de recepción del canal USART (RCIE=1) se producirá un interrupción y el microcontrolador saltará a la dirección que corresponda para atender la interrupción de lectura de datos

Para transmitir datos por el canal USART, se lo hace de las siguientes formas:

- Polling: En este caso el programa monitorea el estado del bit TXIF, si está en “1” significa que TXREG está vacío y solo entonces se puede escribir el siguiente dato en el buffer.
- Interrupción: Con la habilitación de esta, se producirá una interrupción por cada vez que el registro TXREG este vacío, entonces se podrá enviar un nuevo dato. Por tanto solo se debe habilitar esta interrupción después de haber enviado el primer dato y debe ser deshabilitada cuando se haya transmitido el último dato.

USB:

Los PIC's de las series PIC18FX455/X550 contienen una interface de comunicación USB de low-speed y hi-peed hasta con 12Mb/s, esto permite la comunicación entre un host USB y cualquier microcontrolador.

El funcionamiento del módulo USB gestionado a través de tres registros de control. Además, un total de 22 registros se utilizan para gestionar las transacciones USB actuales. Los registros son los siguientes:

- Control del registro USB (UCON).
- Configuración del registro USB (UCFG).
- Registro de transferencia de estado USB (USTAT).
- Registro de dirección del dispositivo USB (UADDR).
- Número de registros del frame (UFRMH: UFRML).
- Registro de habilitación externa de 0 a 15 (UEPn).

Este canal de comunicaciones también maneja interrupciones, estas están habilitadas con un conjunto de registros de control y atrapada con un conjunto separado de los registros de bandera. Además posee una USB RAM donde los datos se mueven entre esta y el núcleo del microcontrolador.

Para este trabajo no se utilizará USB puro, sino el llamado USB_CDC (refiérase al capítulo 4.1.5).

I2C:

El módulo Master Synchronous Serial Port (MSSP) es una interfaz serie útil para comunicarse con otro los dispositivos periféricos o microcontroladores. Estos periféricos-dispositivos pueden ser memorias EEPROM seriales, registros de desplazamiento, controladores de pantalla, convertidores A DC, etc.

El módulo MSSP puede funcionar en uno de dos modos siguientes:

-
- Serial Peripheral Interface (SPI)
 - Inter-Integrated Circuits (I2C)
 - Modo Master Full
 - Modo esclavo (con llamada general)

La interfaz I2C soporta los siguientes modos de hardware:

- Modo maestro
- Modo Multi-Maestro
- Modo esclavo

De este módulo MSSP, trabajaremos con el módulo I2C justamente para comunicarnos con la memoria EEPROM.

3.3.6. Dotando de Salida Analógica al Microcontrolador.

En el diseño de nuestro controlador, específicamente en el controlador regulatorio, se requieren tres salidas analógicas de 0 a 5 voltios como entradas a tres puentes H, los cuales comandan cada actuador lineal, regulando su velocidad y transfiriendo su movimiento a los pedales, para ello utilizaremos los dos módulos PWM del PIC principal PIC18F4550 y un módulo PWM del PIC secundario PIC-18F2550, los dos PIC's comparten una comunicación serial asíncrona.

A partir de estas señales PWM se plantea el uso de un Integrador, como se sabe, un integrador obtiene una señal de salida que está en función de la historia de la señal de entrada, con esto se obtiene a su salida un promedio de dicha señal de entrada en un periodo de tiempo.

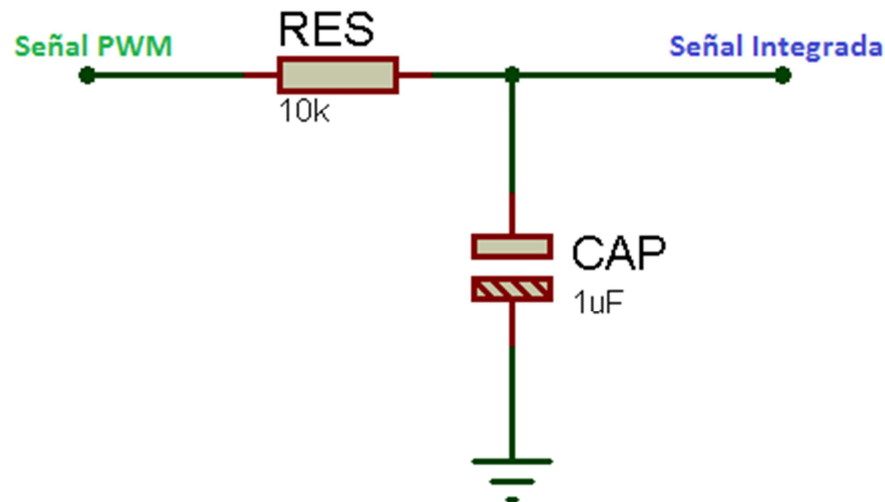


Figura 3. 7 Circuito Integrador.

Como circuito integrador se usará un circuito RC como el de la figura anterior, que no es más que una resistencia seguida de un capacitor electrolítico a tierra, donde la señal de entrada está en el un extremo de la resistencia y la señal integrada de salida se la obtiene entre la resistencia y el capacitor. Luego de esta etapa de integración se envía la señal a un seguidor de voltaje y posteriormente como estrada al puente H.

3.4. Sensores.

Como estradas al controlador, se necesita sensor la posición de los pedales, la posición de los gatillos, la velocidad del automóvil y la inclinación del vehículo. En resumen, se muestra en la siguiente tabla la lista de variables y sensores a utilizar.

Magnitud o Variable	Sensor
Posición de los pedales	Potenciómetro lineal deslizable
Posición de los gatillos	Galga Extensiométrica FLEX22
Inclinación del Automóvil	Acelerómetro MMA7361
Velocidad del Automóvil	Módulo GPS

Tabla 3. 10 Lista de Variables sus respectivos Sensores.

3.4.1. Sensores de Posición

3.4.1.1. Para los pedales.

Para la medición de esta variable se utilizará un potenciómetro lineal deslizante de $10\text{k}\Omega$ que al estar conectado físicamente al pedal como se ve en la figura 3.8, varía respecto a la posición de este, del cual se obtendrá la señal de salida de voltaje respectiva.

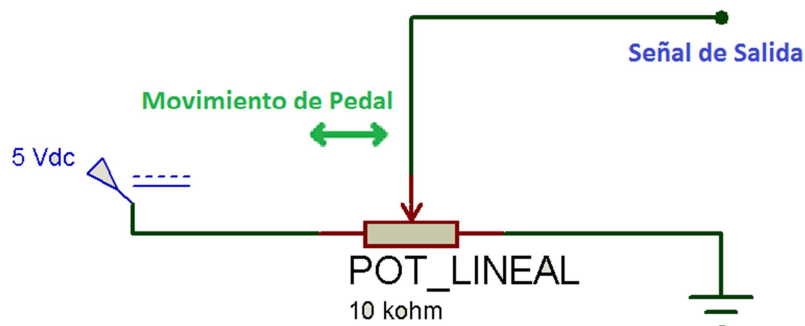


Figura 3. 8 Configuración de Potenciómetro Lineal en Pedales.

En la siguiente tabla se tabulan los resultados de medición de resistencia y voltaje del circuito en función de la distancia recorrida por el potenciómetro lineal.

Distancia [cm]	Resistencia Pot. [kohm]	Voltaje Salida [Vdc]	Valor Digital
0	0,416	0,1	20
0,5	1,762	0,38	78
1	3,16	0,711	146
1,5	4,57	0,946	194
2	5,83	1,162	238
2,5	7,16	1,35	276
3	8,35	1,51	309
3,5	8,78	1,587	325
4	9,15	1,626	333
4,5	9,5	1,671	342
5	9,85	1,713	351
5,5	10,23	1,756	360

Tabla 3. 11 Mediciones de voltaje en función del desplazamiento del potenciómetro lineal.

El voltaje de salida es convertido por el ADC del PIC con resolución de 10 bits, con lo cual se obtienen los datos digitales de la 4ta columna. Para observar gráficamente lo que va a procesar el PIC, se muestran los datos en la siguiente figura.

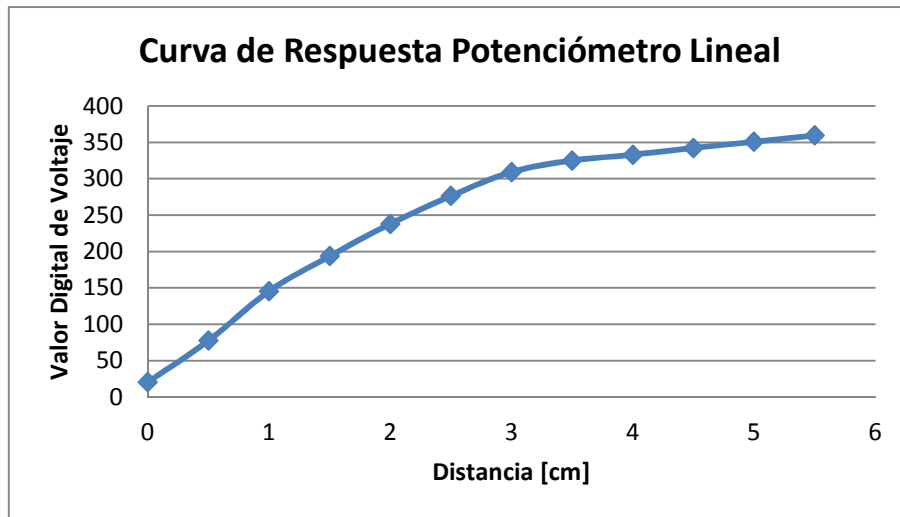


Figura 3. 9 Gráfica de respuesta en voltaje en función del desplazamiento del potenciómetro lineal.

Con esto se tiene un universo de discurso del movimiento del pedal de al menos 340 posibles valores, lo cual es una resolución bastante buena para representar los 5,5 cm de desplazamiento del potenciómetro lineal.

3.4.1.2. Para los gatillos.

Para determinar el set point de los pedales se usarán tres galgas Extensiométricas Flex22 (FS-L-005-253-ST) a manera de gatillos, la cual se muestra en la siguiente figura.

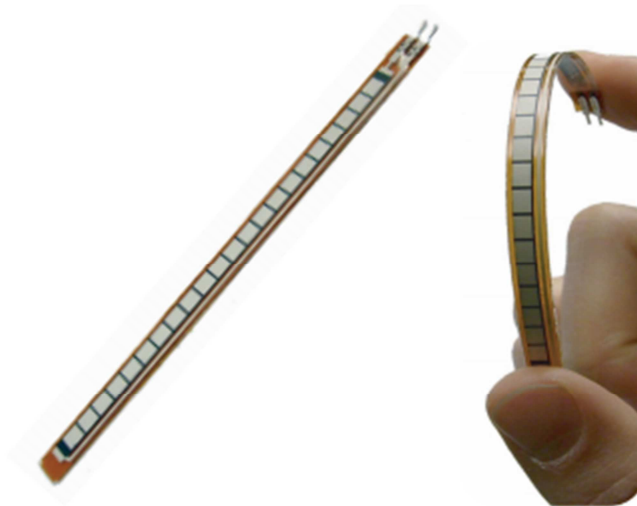


Figura 3. 10 Galga Extensiométrica Flex22 (FS-L-005-253-ST).

Características (Simbol, 2008):

- Longitud 5,53 cm.
- Resistencia en reposo (extendida) 25 k Ω .
- Resistencia a máxima flexión (110 grados) 85 k Ω .

Por facilidad para el usuario, se elaborará dos guantes, el de la mano derecha tendrá en el dedo índice y medio, los sensores (gatillos) para el acelerador y el freno respectivamente, y el de la mano izquierda en el dedo índice tendrá el gatillo del freno.

Cabe aclarar que los tres gatillos se utilizarán en el modo Manual, pero en el modo Semiautomático se usará solo el del acelerador y freno (refiérase al capítulo IV, Modo Manual, Modo Semiautomático).

Al igual que con los potenciómetros lineales de los pedales, las galgas se dispondrán en serie con una resistencia formando un divisor de voltaje del cual se obtendrá la señal de salida respectiva cuyos datos de voltaje respecto al ángulo de flexión se muestra en la siguiente tabla.

Angulo de Flexión [gra]	Voltaje Salida [Vdc]	Valor Digital de Voltaje
0	3,05	625

10	3,17	649
20	3,25	666
30	3,35	686
40	3,47	711
50	3,6	737
60	3,71	760
70	3,84	786
80	3,94	807
90	4,06	831
100	4,13	846
110	4,22	864

Tabla 3. 12 Mediciones de voltaje en función del ángulo de flexión de la galga extensiométrica.

Así mismo, en la siguiente figura se muestran los datos digitales del voltaje respecto al ángulo de flexión, donde el universo de discurso es de 239 posibles valores para el set point.

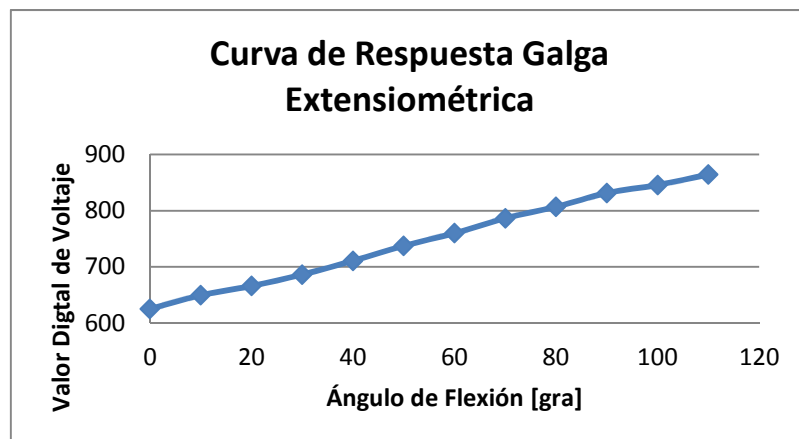


Figura 3. 11 Gráfica de respuesta en voltaje en función del desplazamiento del potenciómetro lineal.

3.4.2. Inclinación del Automóvil

Para la medición de esta variable se usará un sensor acelerómetro, partiendo del principio que la gravedad genera una aceleración de 9.8m/s^2 sobre todos los objetos, este

valor de aceleración se obtiene únicamente cuando el acelerómetro se encuentra perpendicular a la horizontal, entonces al variar su inclinación respecto a determinado eje, la resultante de la aceleración varía permitiendo de cierta forma medir el ángulo en el que está dispuesto el sensor o el objeto en el cual está montado.

El sensor a utilizar es el Mma7361, que es un acelerómetro triaxial, es decir, puede medir la aceleración en los tres ejes, teniendo así para cada eje una salida de voltaje analógica. Las especificaciones generales se muestran en la siguiente tabla:

Parámetro	Valor
No. de ejes	3
Alimentación	2,2 a 3,6 Vdc
Consumo corriente	400 uA
Sensibilidad	800mV/g @ 1,5g
Sensibilidad seleccionable	1,5g o 6g

Tabla 3. 13 Especificaciones generales acelerómetro Mma7361.

El sensor, viene montado en una mini PCB de 10 pines como se muestra en la figura 3.12. De los cuales solo se van a utilizar 4 pines, Vcc que se alimenta con 5Vdc ya que el módulo tiene su propio regulador de 3.3Vdc, también los pines de GND y SLEEP que se activa en bajo, y el pin de salida analógica del eje Y. Además el sensor viene pre configurado para que se lo pueda utilizar con la sensibilidad de 1.5g, con 800mV/g lo cual nos brinda la mayor resolución posible.

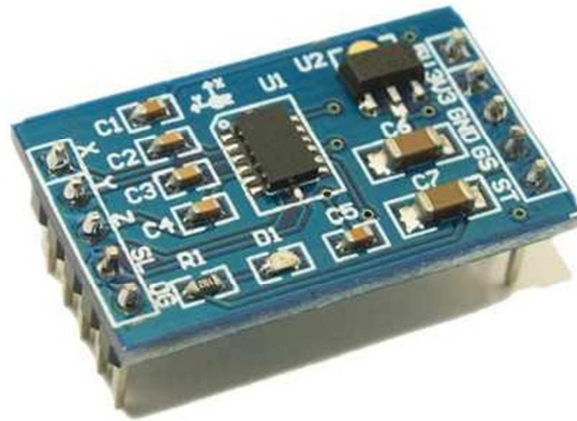


Figura 3. 12 Acelerómetro Mma7361, mini PCB

Con el principio de la aceleración de la gravedad antes mencionado, se procede a convertir el voltaje analógico de salida del sensor, en un valor proporcional al ángulo de inclinación tomando en cuenta la componente de la gravedad presente en el eje Y al someter el sensor a una inclinación. Esto se lo realiza mediante la siguiente ecuación.

$$V_{salida} = V_{offset} + \frac{\Delta V}{\Delta G} \sin \phi$$

Dónde:

V_{salida} : Voltaje analógico de salida del sensor para determinado eje.

V_{offset} : Voltaje de salida del sensor cuando está paralelo a la horizontal (0 grados).

$\frac{\Delta V}{\Delta G}$: Sensibilidad del sensor, parámetro fijo que en este caso es $\approx 800\text{mV/g}$

ϕ : Ángulo de inclinación respecto a la horizontal.

Como nuestro objetivo es hallar el ángulo de inclinación, se despeja ϕ obteniendo la siguiente ecuación:

$$\phi = \text{asin} \frac{V_{salida} - V_{offset}}{\frac{\Delta V}{\Delta G}} \quad [ec. 3.4.2]$$

Después de las respectivas pruebas de funcionamiento, se obtuvieron las constantes que se muestran en la siguiente tabla, con las cuales se obtuvo un error de ± 0.2 grados:

V_{salida}	2.49Vdc (+90 grados), 0.935Vdc (-90 grados)
V_{offset}	1.73 Vdc, al estar horizontal (0 grados)
$\frac{\Delta V}{\Delta G}$	0.775 Vdc / g

Tabla 3. 14 Constantes para cálculo de inclinación

Se ha escogido el rango de -15 a 15 grados, dado que este rango cubre en su mayor parte la posible inclinación de un automóvil dentro de una ciudad como la ciudad de Quito, siendo -15 grados una bajada o pendiente negativa, mientras que 15 grados representa una cuesta o pendiente positiva.

Estos 30 posibles valores de inclinación van a ser utilizados en el diseño del control regulatorio del modo semiautomático, este dato es relativamente importante al momento de arrancar, ya que en situaciones de pendiente positiva los pedales deben salir con menor velocidad. Mientras que en pendiente negativa o en el llano los pedales salen a velocidad normal.

3.4.3. Montaje del GPS para medición de Velocidad.

La velocidad del automóvil se la puede medir directamente con varios métodos como por ejemplo un sensor magnético ubicado en la rueda, con el cual en cada revolución de la misma envíe un pulso al uC indicando que se la rueda ha dado una vuelta, sin embargo, este método es poco práctico por la dificultad de instalación; o también puede hacerse con un encoder, pero se tiene el riesgo de daño físico del sensor, y al ser óptico se puede ver afectado por la suciedad.

Entonces un método simple y preciso de medir la velocidad del automóvil es mediante un GPS. Sabemos que un módulo GPS devuelve las coordenadas geográficas tanto de latitud como de longitud de cualquier lugar en el planeta donde este se encuentre, entonces para medir la velocidad del vehículo, simplemente se tiene que tomar dos coordenadas separadas por un periodo de tiempo entonces se aplica:

$$v = \frac{d}{t}$$

Donde v es la velocidad, d es distancia recorrida y t es el intervalo de tiempo. Entonces mediante trigonometría podemos calcular la distancia recorrida, de la siguiente forma según se muestra en la figura 3.13.

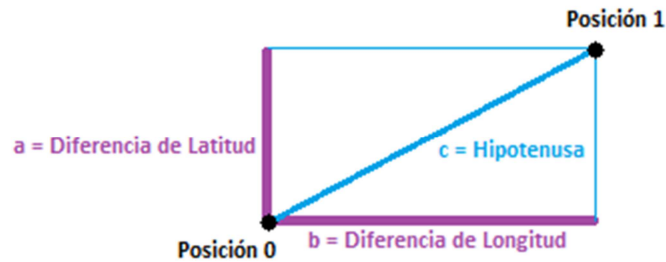


Figura 3. 13 Método de la Hipotenusa.

Posición 0 es el primer dato de posición adquirido desde el GPS en un tiempo 0, cuyos valores de latitud y longitud son lat_0 y lon_0 respectivamente; la posición 1 es un segundo dato adquirido un segundo después, con lat_1 y lon_1 , entonces para calcular la distancia recorrida (c) se tiene la siguiente relación.

$$a = lat_1 - lat_0$$

$$b = lon_1 - lon_0$$

Y aplicando trigonometría se calcula la distancia recorrida c .

$$c = \sqrt{a^2 + b^2}$$

Donde $d = c$, y aplicando en la ecuación de la velocidad

$$v = \frac{d}{t}$$

Con $t = 1$ seg, finalmente se tiene la velocidad del vehículo. Este cálculo es realizado por el uP del GPS. Para este efecto se usará el módulo GPS Click, el cual viene integrado con el chip GPS LEA-6S de la empresa U-Blox (U-Blox, 2011), mostrado en la siguiente figura.



Figura 3. 14 GPS Click.

De este módulo básicamente se usan 3 pines, el transmisor Tx, Vcc y Gnd, que son los pines 8, 9 y 13 respectivamente.

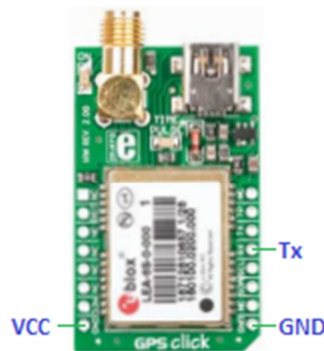


Figura 3. 15 Pines usados del GPS.

Las características del GPS Click se muestran en la tabla a continuación:

Parámetro	Especificación
Tiempo de calentamiento	26 segundos
Sensibilidad	-160 dBm
Frecuencia de actualización	1 Hz
Precisión Horizontal	2 mts
Altitud Máxima	5000 mts
Velocidad Máxima	1800 km/h
Alimentación Vcc	2,7 - 3,3 Vdc

Tabla 3. 15 Características GPS Click.

Donde, frecuencia de actualización significa que el GPS envía una trama de datos cada segundo, entonces podremos tener el valor de la velocidad con una frecuencia de 1Hz.

Estos datos son transmitidos por el pin Tx de módulo, y leídos por el receptor del puerto serial del PIC18F4550, cabe aclarar que el módulo no necesita ningún tipo de comando para enviar la trama ya que lo hace automáticamente a penas se lo alimenta.

El módulo GPS envía la llamada trama NMEA (Gpsinformation, 2010), esta es un estándar de comunicación entre dispositivos de posicionamiento global, está compuesta por una serie de caracteres los cuales contienen la información propia de un GPS, entre los datos de la trama NMEA están los mostrados en la tabla siguiente.

Parámetro	Contenido
\$GPBOD	Rumbo de Origen a Destino
\$GPBWC	Rumbo y distancia al punto de referencia
\$GPGGA	Posicionamiento global de datos
<i>\$GPGLL</i>	<i>Posición geográfica, latitud/longitud</i>
\$GPGSA	Lista de satélites activos
\$GPGSV	Lista de satélites en línea de vista
\$GPHDT	Rumbo varadero
\$GPR00	Lista de los puntos de referencia
\$GPRMA	Recomendaciones mínimas de Loran-c
\$GPRMB	Mínima información de navegación recomendada
\$GPRMC	Datos de transito
\$GPRTE	Rutas
\$GPTRF	Datos sobre el transito - configurados
\$GPSTN	Identificación de datos múltiples
\$GPVBW	Doble fondo / agua
<i>\$GPVTG</i>	<i>Curso sobre la tierra y velocidad de avance</i>
\$GPWPL	Ubicación del punto de referencia
\$GPXTE	Error de rastreador
\$GPZDA	UTC Fecha/Hora y diferencia horaria

Tabla 3. 16 Contenido de la trama NMEA.

Todos estos datos son transmitidos a través del terminal Tx del módulo GPS, pero el dato que nos interesa es el de la sentencia \$GPVTG. A continuación un ejemplo de la sentencia \$ GPVTG.

\$GPVTG,054.7,T,034.4,M,005.5,N,010.2,K

Dónde:

054.7,T Modo de ruta verdadero.

034.4,M Modo de ruta magnético.

005.5,N Velocidad sobre la tierra en “knots” (millas náuticas).

010.2,K Velocidad sobre la tierra en Kilómetros / hora.

Con la información de esta trama, ya podemos programar en el PIC un algoritmo que separe la parte de esta que nos interesa (refiérase al capítulo 4.1.7.2) y prácticamente ya podemos saber la velocidad del automóvil.

3.4.4. Pulsador indicador de cambio de marcha.

Con el propósito de que el sistema sepa que el conductor quiere cambiar de marcha, se instaló un pulsador normalmente abierto como el de la siguiente figura, en el borde de la palanca de cambios, al pulsar este, el conductor le está indicando al sistema que va a realizar un cambio de marcha.



Figura 3. 16 Pulsador indicador de cambio de marcha

3.5. Elaboración de la Tarjeta de Adquisición y Control (TAC).

La tarjeta de adquisición y control tiene el objetivo de recibir la señal de todos los sensores y del Set Point, enviarlas al controlador el cual mediante el conversor análogo digital transforma las señales de voltaje de 0 a 5 voltios, en valores digitales, en este caso se utilizará un ADC con 10 bits de resolución, con lo que el dato digital estará entre 0 y

1023. Una vez gestionados estos datos, las señales de control de tipo PWM, son enviadas a un circuito integrador, después a un seguidor de voltaje, a un demultiplexor de 1 a 2 y finalmente a los Puentes H, el cuales comandan a los actuadores, el esquema básico se muestra en la figura 3.16.

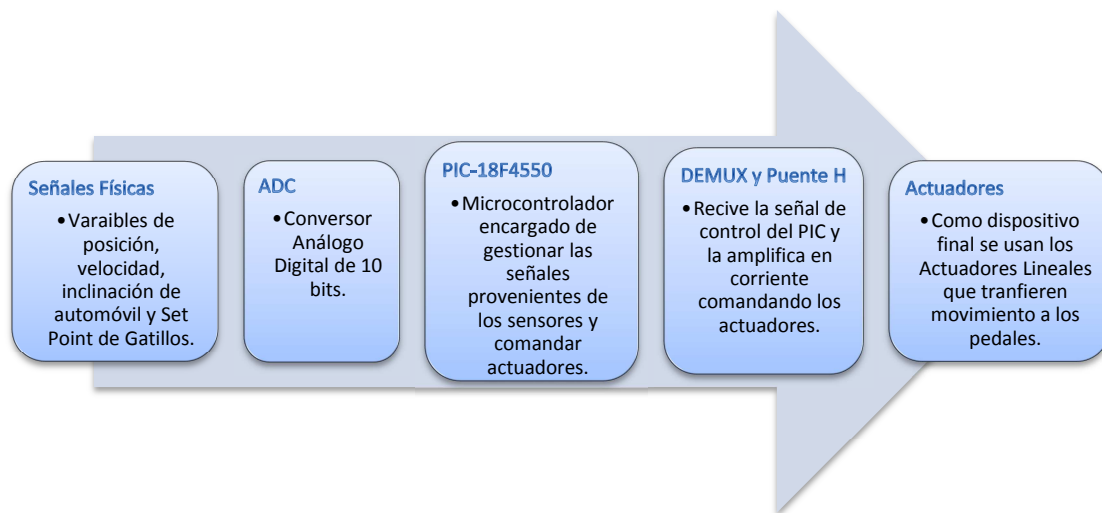


Figura 3. 17 Esquema básico de tarjeta de Control.

Además dispone de una interface de comunicación USB_CDC, mediante la cual se realizará la configuración de parámetros del controlador sobre la marcha, así como la visualización de su desempeño.

3.5.1. Especificaciones de componentes.

Los componentes de la TAC se muestran detalladamente en la siguiente figura.

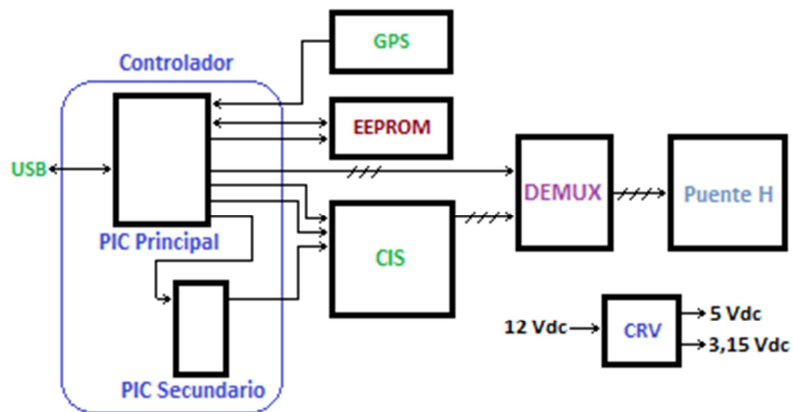


Figura 3. 18 Componentes del Sistema Controlador.

3.5.1.1. Circuito Regulador de Voltaje (CRV)

Dado que nuestra fuente de corriente es la propia batería del vehículo, se debe solucionar el hecho de que es una fuente bastante ruidosa por el hecho que está conectada a diferentes dispositivos electrónicos y sometidos al ruido del propio motor.

Para esto se debe filtrar la señal con reguladores de voltaje, capacitores y diodos, el esquema se muestra en la figura 3.18.

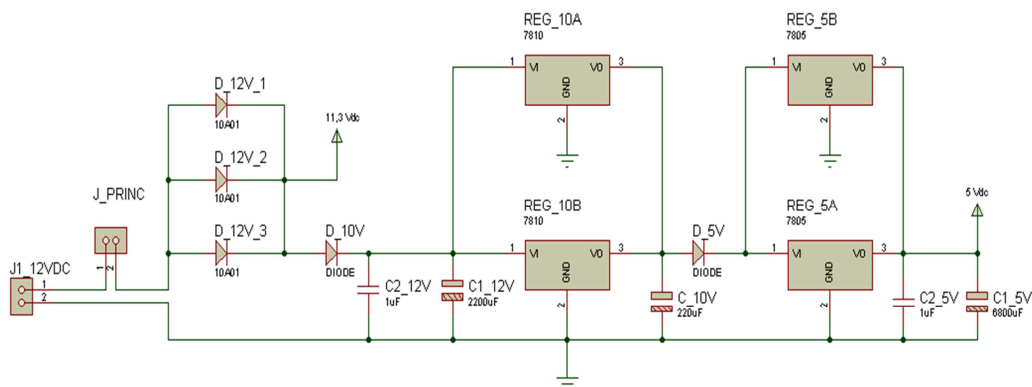


Figura 3. 19 Circuito Regulador de Voltaje.

Se tiene la entrada de voltaje directo de la batería del vehículo J1_12VDC, después de pasar por un interruptor principal J_PRINC, pasa por tres diodos rectificadores de 6 amperios (D_12V_1, D_12V_2 y D_12V_3) en paralelo, de modo que se tiene aproximadamente 11,3 voltios disponibles para alimentar a los actuadores lineales a una

corriente pico de 18 amperios, sabemos que cada actuador consume una corriente a máxima carga de 4 amperios.

Desde los 11,3 voltios se tiene un diodo rectificador de 1 amperio (D_10V) que va a un capacitor electrolítico de 2200uF de 50V en paralelo con un capacitor de tantalio de 1uF, seguido de dos reguladores de 10 voltios 7810 en paralelo, obteniendo así un mayor filtrado.

A su salida tenemos un capacitor electrolítico de 220uF seguido de un diodo rectificador de un amperio (D_10V), con lo que se tiene una entrada prácticamente libre de ruido a los regulador de 5 voltios, los 7805 que también están dispuestos en paralelo. Finalmente se tiene un capacitor electrolítico de 6800uF de 25V en paralelo con un capacitor de tantalio de 1uF.

Para alimentar el GPS se requiere una voltaje de 2.7 a 3.3 Vdc, por lo que se ha configurado un regulador de voltaje variable, el LM317, para que con una alimentación de 5 Vdc, a su salida nos entregue 3.15 Vdc. Las conexiones se muestran a continuación.

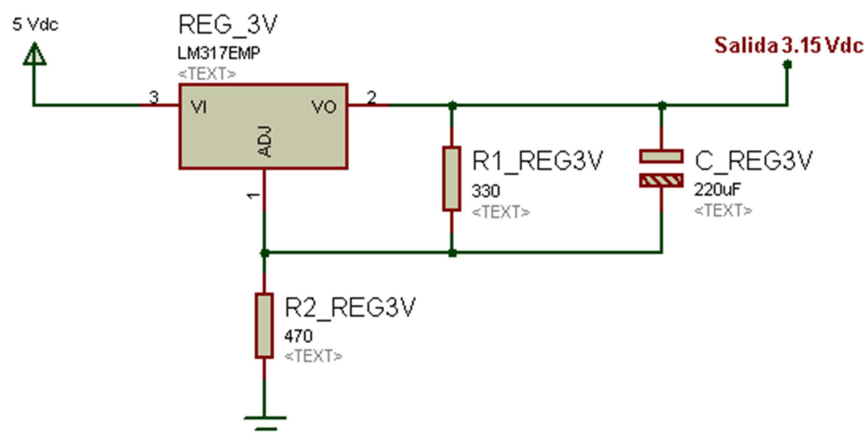


Figura 3. 20 Circuito regulador, 3.15 Vdc

3.5.1.2. Controlador.

Para el controlador, se utiliza el PIC18F4550 como uP principal, este recibe todas las señales análogas de los sensores así como las de los Set Points y genera las salidas de control, tiene dos de las tres salidas análogas de control, una en cada módulo PWM, para los pedales del embrague y freno respectivamente; y el PIC-18F2550 como uP secundario, tiene como función la de proporcionar la tercera salida análoga de control para el pedal del acelerador, este recibe la señal de control del PIC principal mediante comunicación RS-232 y la envía a su módulo PWM.

Las entradas necesarias, así como su respectivo pin asignado en el PIC principal se muestran en la tabla 3.12.

Nombre	Tipo de señal	Descripción	Pin Asignado
EMB_G	Analógica	Señal proveniente del gatillo del Embrague	AN0, pin 2 PIC18F4550
FRE_G	Analógica	Señal proveniente del gatillo del Freno	AN1, pin 3 PIC18F4550
ACE_G	Analógica	Señal proveniente del gatillo del Acelerador	AN2, pin 4 PIC18F4550
EMB_P	Analógica	Señal proveniente del pedal del Embrague	AN3, pin 5 PIC18F4550
FRE_P	Analógica	Señal proveniente del pedal del Freno	AN4, pin 7 PIC18F4550
ACE_P	Analógica	Señal proveniente del pedal del Acelerador	AN5, pin 8 PIC18F4550
INC	Analógica	Señal proveniente del sensor de Inclinación	AN6, pin 9 PIC18F4550
SEL	Digital	Selector de modo, Manual (o lógico), Semiautomático (1 lógico)	RC0, pin 15 PIC18F4550
GPS_D	Datos Seriales RS-232	Trama NMEA proveniente del Módulo GPS, para medición de velocidad de vehículo	RX, pin 26 PIC18F4550

USB_D-	Datos Seriales USB	Datos recibidos desde el Ordenador a través del puerto USB	D-, pin 23 PIC18F4550
EEPROM_SDA	Datos Seriales I2C	Datos recibidos desde la memoria EEPROM	RD2, pin 21 PIC18F4550

Tabla 3. 17 Distribución de entradas al Microcontrolador Principal.

En total se tienen 11 entradas, 7 de ellas son señales analógicas provenientes de los sensores tanto de pedales como de gatillos además del sensor de inclinación, 1 es señal digital proveniente de un interruptor de selección de modo Manual/Semiautomático, y 3 son entradas de datos seriales RS-232 desde el GPS, USB desde el ordenador e I2C desde la memoria EEPROM respectivamente.

Respecto a las salidas del PIC, se muestran en la tabla siguiente.

Nombre	Tipo de señal	Descripción	Pin Asignado
EMB_C	Analógica	Señal de control para el pedal del Embrague	CCP1, pin 17 PIC18F4550
FRE_C	Analógica	Señal de control para el pedal del Freno	CCP2, pin 16 PIC18F4550
A_C	Digital	Comanda la entrada de Control A (pin 11) del Demultiplexor Analógico	RD5, pin 28 PIC18F4550
B_C	Digital	Comanda la entrada de Control B (pin 10) del Demultiplexor Analógico	RD6, pin 29 PIC18F4550
C_C	Digital	Comanda la entrada de Control C (pin 09) del Demultiplexor Analógico	RD7, pin 30 PIC18F4550
LED_USB	Digital	Indica el estado de conexión USB con el ordenador, 0 lógico - desconectado y 1 lógico - conectado	RD0, pin 19 PIC18F4550

LED_AUX	Digital	Indicador Auxiliar	RB0, pin 33 PIC18F4550
J_SPEAK	Digital	Señal de salida a un zumbador el cual genera un sonido indicador para el cambio de marcha	RD1, pin 20 PIC18F4550
ACE_C_TX	Datos Seriales RS-232	Dato serial de la señal de control del Acelerador, enviado al PIC secundario	Tx, pin 25 PIC18F4550
USB_D+	Datos Seriales USB	Datos enviados hacia el Ordenador a través del puerto USB	D+, pin 24 PIC18F4550
EEPROM_SDA	Datos Seriales I2C	Datos enviados hacia la memoria EEPROM	RD2, pin 21 PIC18F4550
EEPROM_SCL	Datos Seriales I2C	Señal de reloj de la memoria EEPROM	RD3, pin 22 PIC18F4550

Tabla 3. 18 Distribución de salidas del Microcontrolador Principal.

Tiene un total de 12 salidas, 2 salidas de control analógicas, 3 salidas digitales de comando hacia el Demultiplexor analógico, 3 salidas digitales para indicadores y 4 salidas de datos seriales.

En el PIC secundario tiene como entrada por el pin de recepción serial RS-232 (RX, pin 26, PIC-18F2550), la señal de control para el acelerador. Y como salida se tiene la misma señal de control pero de tipo PWM (CCP1, pin 13, PIC-18F2550).

3.5.1.3. Circuito Integrador – Seguidor (CIS).

Las tres salidas de control del PIC principal y del PIC secundario son de tipo PWM, entonces es necesario en primer lugar estabilizar la señal y luego aplicar un seguidor de voltaje, el circuito utilizado se muestra en la figura 3.20. Cabe aclarar que para cada señal analógica de control se necesita un circuito CIS.

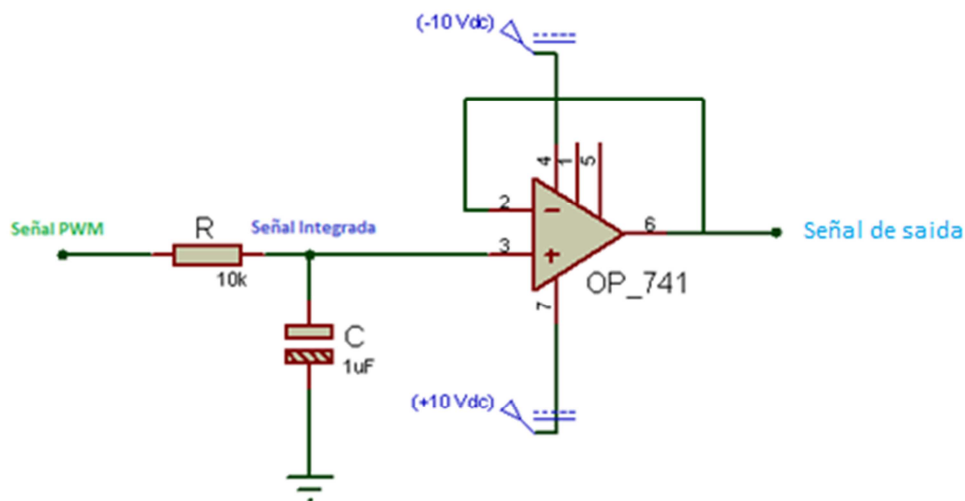


Figura 3. 21 Circuito Integrador - Amplificador de Corriente.

Para estabilizar la señal se usará el Circuito Integrador RC y para hacer un seguimiento de la señal se tiene un circuito seguidor de voltaje a base de amplificadores operacionales (LM741), para usar el amplificador operacional como un seguidor de voltaje se debe realimentar la señal directamente de la salida (pin 6) hacia el terminal inversor (pin 2), la entrada de voltaje tiene que estar en el terminal no inversor (pin 3). Más que amplificar la corriente el objetivo es de separar la señal proveniente del uC para que no esté en conexión directa con los transistores del puente H.

Dado que el amplificador operacional requiere alimentación tanto con voltaje positivo (+10 Vdc) y con voltaje negativo (-10 Vdc), pin 7 y 4 respectivamente. El voltaje positivo se lo obtiene directamente del regulador de 10 voltios, pero fue necesario obtener de algún modo el voltaje negativo a partir de la única alimentación positiva disponible. Para esto, y por ser de fácil adquisición en el mercado local se utilizó un MAX232 configurado según la siguiente figura.

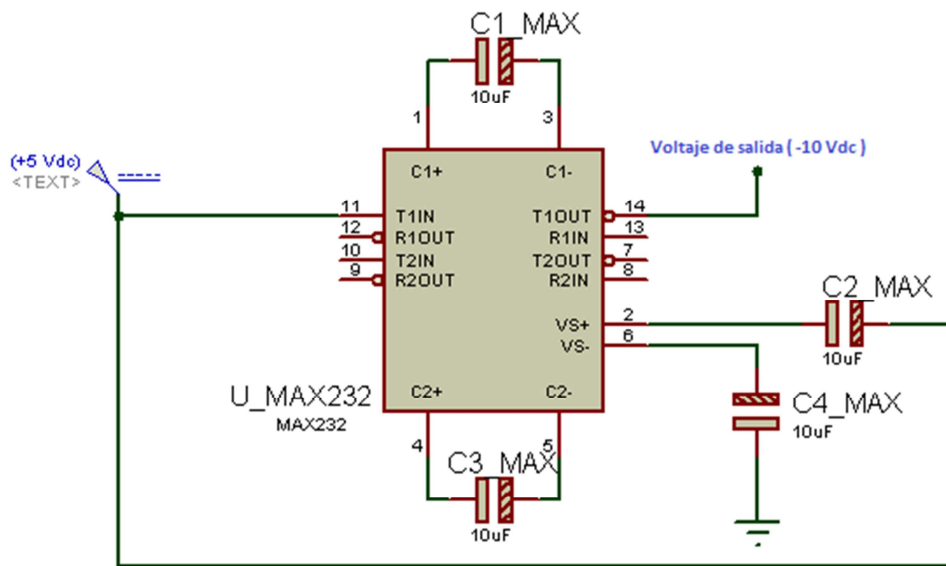


Figura 3. 22 MAX232 usado como fuente negativa de Voltaje.

Sabemos que la función de un MAX232 es la de transformar los niveles de voltaje TTL (0 a 5 Vdc) a los niveles de RS-232 (10 a -10 Vdc) en una comunicación serial entre por ejemplo un microcontrolador y una PC, entonces teniendo en el pin 11 un voltaje de 1 lógico, es decir 5 Vdc, se puede obtener en el pin 14 un voltaje negativo de -10 voltios, mientras que si en el pin 11 tuviésemos un 0 lógico, en el pin 14 tuviésemos +10 Vdc.

3.5.1.4. Demultiplexor (DEMUX).

Los demultiplexores son circuitos combinatoriales que tienen una entrada de datos (In 1) y n entradas de control (Sel) que sirven para seleccionar una de las salidas (Out 1, Out 2), por la que ha de salir el dato que se presente en la entrada (Wikipedia, Circuitos Integrados demultiplexores, 2010). El principio de funcionamiento y su equivalente se muestran en la siguiente figura.

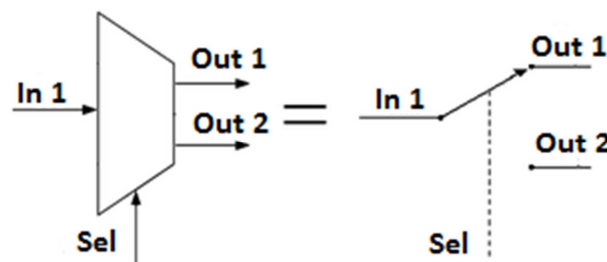


Figura 3. 23 Principio de Funcionamiento del DEMUX.

En este sentido, el DEMUX es un dispositivo base para el control de nuestros actuadores lineales, ya que con este se controla el sentido de giro de los mismos mediante la inversión de la polarización de alimentación.

En la siguiente figura se muestra el diagrama de conexión del DEMUX análogo CD4053, este es un CI interesante ya que puede trabajar como MUX o como DEMUX, dependiendo del sentido de la corriente en los pines de entrada/salida además posee un pin de habilitación general llamado INHIBIT el cual es activado en bajo.

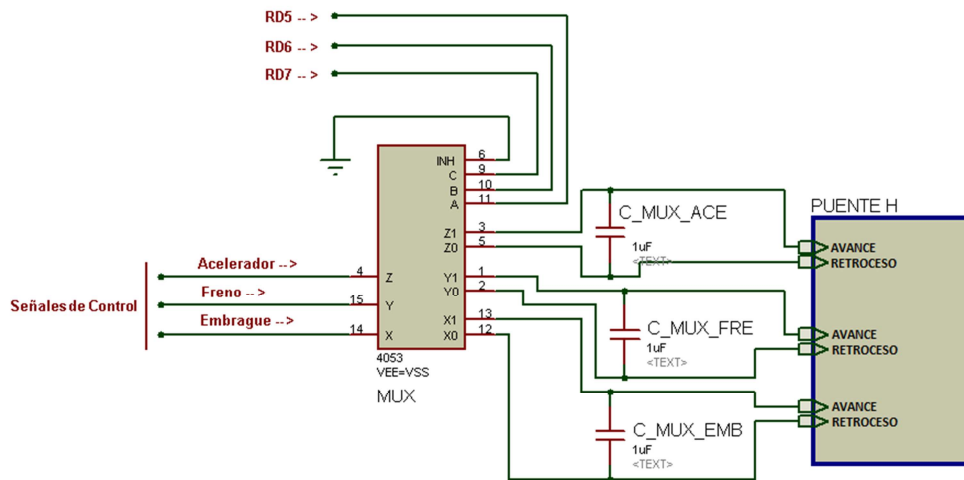


Figura 3. 24 Diagrama de conexión del DEMUX análogo CD4053.

Este recibe las tres señales análogas de control por medio de los pines de entrada 14, 15 y 4, con las señales del embrague, freno y acelerador respectivamente. Mediante los pines de control A, B y C se seleccionan los canales de salida que serán activados, es decir, por qué canales saldrán las señales de entrada. Para entender mejor este proceso es necesario estudiar la tabla de verdad del DEMUX que se muestra a continuación.

INHIBIT	C	B	A	CANALES ACTIVADOS
0	0	0	0	z0, y0, x0
0	0	0	1	z0, y0, x1
0	0	1	0	z0, y1, x0
0	0	1	1	z0, y1, x1
0	1	0	0	z1, y0, x0
0	1	0	1	z1, y0, x1

0	1	1	0	z1, y1, x0
0	1	1	1	z1, y1, x1
1	*	*	*	*

Tabla 3. 19 Tabla de verdad del DEMUX análogo CD4053.

Se puede ver que por ejemplo teniendo el pin A = 1, lo que se tenga como entrada el pin x (pin 14), saldrá por x1 (pin 13), mientras que en la salida x0 (pin 12) tendremos en nivel lógico de 0 Vdc. Caso contrario, si tuviésemos A = 0, lo que se tenga como entrada el pin x, saldrá por x0, mientras que en la salida x1 tendremos en nivel lógico de 0 Vdc. Con esto se asegura que al puente H, en las entradas de avance y retroceso no ingrese el mismo voltaje por ambos pines, es decir si el avance tiene un voltaje diferente de cero, el retroceso tendrá voltaje igual a cero. Y en reposo, es decir cuando las señales análogas de control sean cero, ambos pines tendrán una salida igual a cero voltios, por lo que el actuador no se mueve.

Usando esta configuración, es posible comandar cada salida del DEMUX (Xx, Yx, Zx), con los pines A, B y C, teniendo como entradas las tres señales análogas para comandar los tres pedales y las salidas son enviadas a los 6 puertos de entrada del puente H, dos para cada actuador lineal.

3.5.1.5. Puente H de Potencia.

El puente H es un circuito electrónico a base de transistores, que permite a un motor eléctrico DC girar en ambos sentidos.

Existen en el mercado circuitos integrados como el driver L293B de ST Microelectronics, sin embargo, la potencia de salida de este driver no se ajusta a las necesidades de corriente de nuestros actuadores lineales, como sabemos la corriente máxima de consumo de los actuadores es de 4 amperios, contra 1 amperio por canal que soporta el L293B.

Por esta razón se opta por diseñar un circuito puente H a base de transistores BJT (TIP33, TIP34), usando transistores NPN y sus complementarios PNP, los cuales soportan

hasta 10 amperios de corriente continua. Se utilizó la configuración mostrada a continuación.

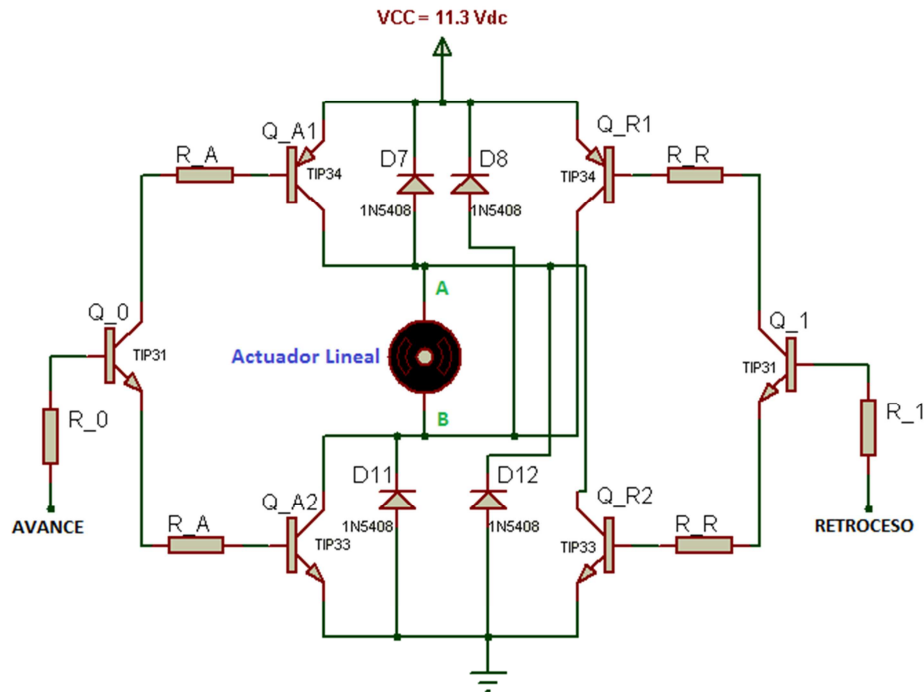


Figura 3. 25 Esquema de conexión del Puente H de Potencia

Para entender el funcionamiento de este circuito tenemos que las señales de control ingresan por los puertos AVANCE y RETROCESO, entonces, cuando se tenga AVANCE $> 0V_{dc}$ y RETROCESO = $0 V_{dc}$, la corriente que ingresa por la base del transistor Q_0 hace que se activen los transistores Q_A1 y Q_A2 haciendo circular corriente de A hacia B, caso contrario cuando se tenga AVANCE = $0V_{dc}$ y RETROCESO $> 0 V_{dc}$, la corriente circulará desde B hacia A, haciendo girar el motor del actuador lineal en sentido contrario.

Para garantizar la protección de los transistores, se debe de agregar diodos rectificadores para evitar corrientes transitorias que puedan quemarlos; estos se conectan inversamente, es decir, el cátodo va al terminal positivo y el ánodo al negativo. El diodo utilizado es el 1N5408 de 3 amperios.

Los transistores utilizados son:

- TIP31: Tipo NPN, amplifica la corriente de la señal de control de avance / retroceso.

- TIP33 (Q_A2, Q_R2): Tipo NPN, corriente colector 10 amperios (12 max).
- TIP34 (Q_A1, Q_R1): Tipo PNP, corriente colector 10 amperios (12 max).

El detalle de cálculos de las resistencias limitadoras de corriente ($R_0=R_1$, $R_A=R_R$), se encuentra en [ANEXOS: 1. Puente H de potencia, cálculo para resistencias limitadoras de corriente.]

Una vez que se han calculado las resistencias limitadoras en el circuito del puente H, aseguramos que la corriente entregada a los actuadores lineales sea la necesaria. Ya se puede controlar la corriente de salida, y por lo tanto la velocidad del actuador al variar el voltaje de control (valor configurable de 2 a 5 voltios) en los puertos de AVANCE y RETROCESO. Finalmente se implementa el circuito de la siguiente figura.

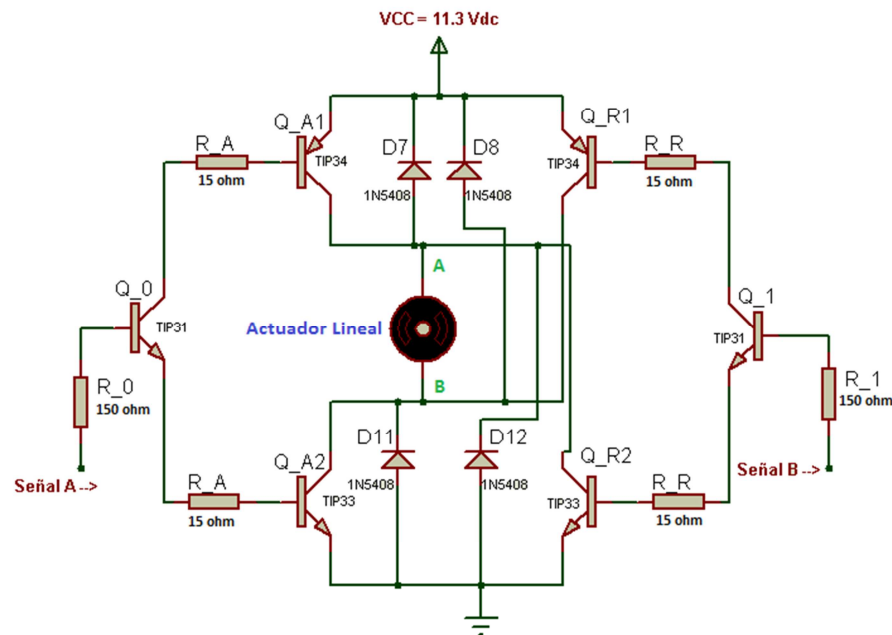


Figura 3. 26 Puente H - Circuito Implementado.

3.5.1.6. Circuito de memoria EEPROM.

Una EEPROM, por sus siglas es una ROM (Memoria de solo lectura) programable y borrable eléctricamente, a diferencia de la EPROM que se borra mediante un dispositivo que emite rayos ultravioletas. Las EEPROM son memorias no volátiles, es decir, es un tipo

de memoria que puede retener información almacenada incluso cuando no recibe alimentación.

Para este trabajo contar con una unidad de memoria resulta de suma importancia, porque al programar el PIC, se pueden almacenar un sin número de constantes pero en el caso que se necesite modificar el valor de esas constantes sobre la marcha lo único que se puede hacer es volver a grabar el PIC, lo cual no es óptimo, por lo que se plantea poder cambiar el valor de estas constantes mediante la interface de configuración en Visual Basic. (refiérase al capítulo 4.3).

En si las constantes que vamos a guardar y a leer de esta memoria son las ganancias del controlador, los valores mínimos y máximos del universo de discurso de las variables de entrada/salida.

Dicho esto se usa la memoria EEPROM ST24C08 cuyas características se muestran a continuación.

CARACTERÍSTICAS	VALORES
Ciclos lectura-escritura	1 millón
Retención de Información	40 años
Alimentación	3 a 5.5 Vdc
Interface de Comunicación	Serial I2C
Protección	Contra Escritura
Memoria	8 kbits
Velocidad comunicación	100 kbits/s

Tabla 3. 20 Características EEPROM ST24C08.

El esquema de conexión se muestra en la siguiente figura, donde para comunicarse con el uC principal (PIC18F4550), necesita de dos hilos, el SCL que es la señal de reloj generada por el uCy el SDA que es la señal de datos, conectados a los pines RD3 (pin 22) y RD2 (pin 21) del microcontrolador.

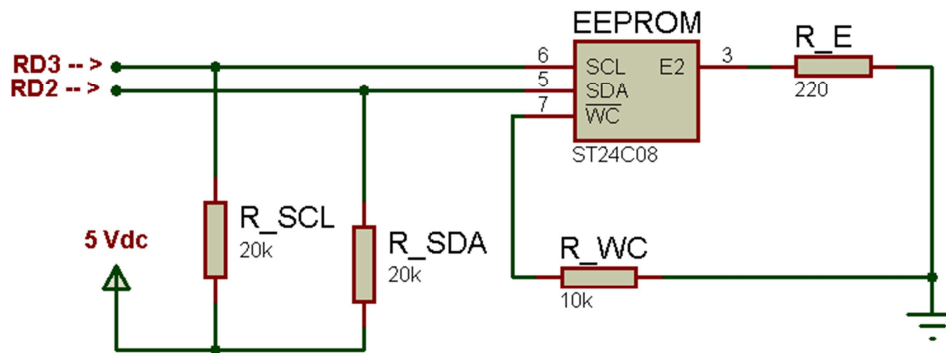


Figura 3. 27 Esquema conexión EEPROM ST24C08.

El PIC actúa como maestro, ya que este genera la señal de reloj y la memoria actúa como esclavo. El Maestro genera la condición de Start, cada palabra puesta en el bus SDA debe tener 8 bits, la primera palabra transferida contiene la dirección del Esclavo seleccionado, en este caso como se tiene un solo esclavo, esta dirección es usada en la EEPROM para definir la dirección de memoria donde se desea leer o escribir los datos. Además se necesita de dos resistencias de pull-up, las cuales mientras el bus esté libre, hace que las dos líneas estén en alto.

3.5.2. Diseño y Vista general de tarjeta.

Para el diseño de la TAC se utilizó el software Proteus 7.7 SP2 que en conjunto con el Ares hace posible la creación de circuitos impresos.

Una vez que en el Proteus hayamos agregado todos los componentes y tengamos el esquema de conexiones entre todos los elementos como se muestra en la figura 3.28, se exporta ese diagrama al Ares, donde luego de posicionar los elementos tal como queremos que quede muestra TAC, se generan las pistas automáticamente.

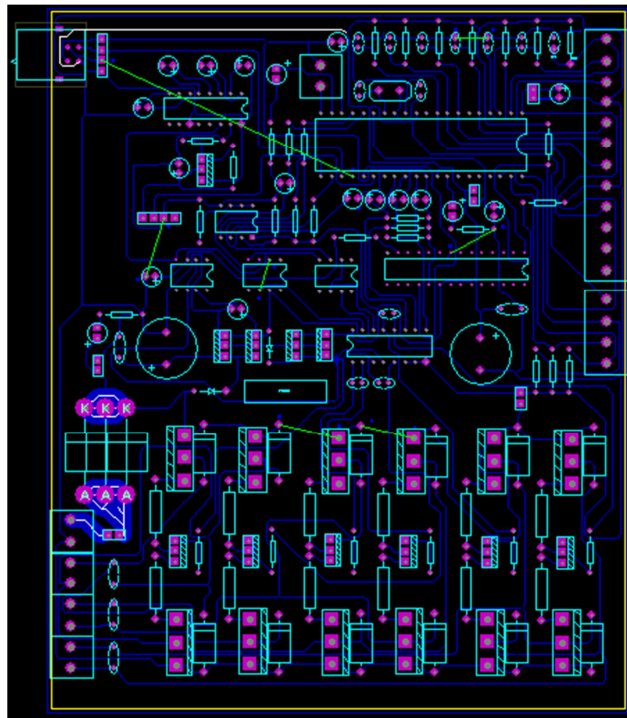


Figura 3. 288 Esquema electrónico de elementos y circuitos en Ares.

Para poder exportar estas pistas, y fabricar la TAC se utiliza la de la barra de menú, la opción Output/ExportGraphics para finalmente escoger Export PDF File. Como se muestra a continuación

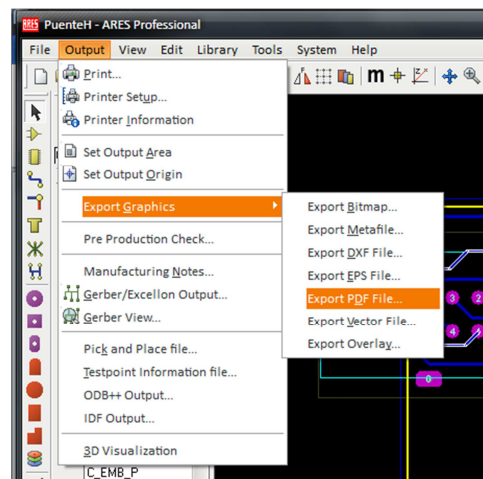


Figura 3. 2929 Exportar pistas en formato PDF.

Con esto se genera un archivo PDF que contienen una imagen de las pistas generadas.

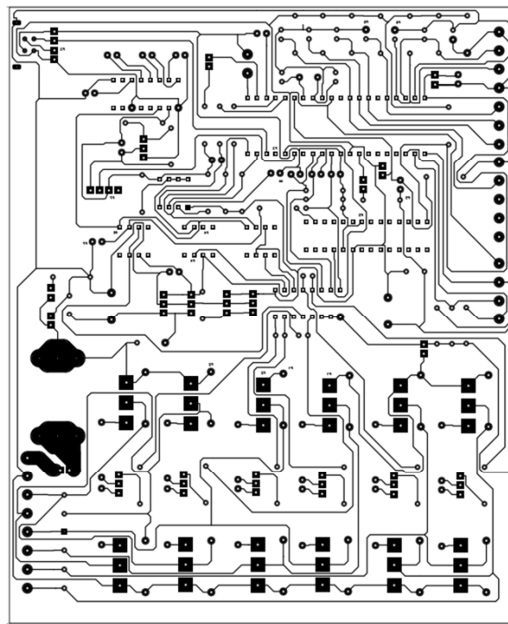


Figura 3. 300 Pistas generadas en formato PDF.

En las figuras siguientes se muestra la TAC en la vista en 3D generada por el software y la vista de la tarjeta elaborada.

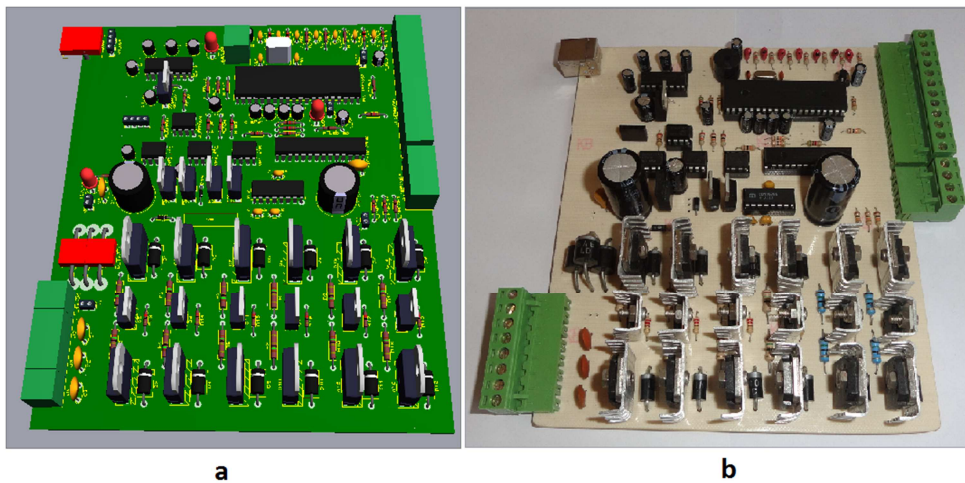


Figura 3. 311 Vista de la TAC, capa superior. a). Vista 3D generada en software ARES. b). Vista real.

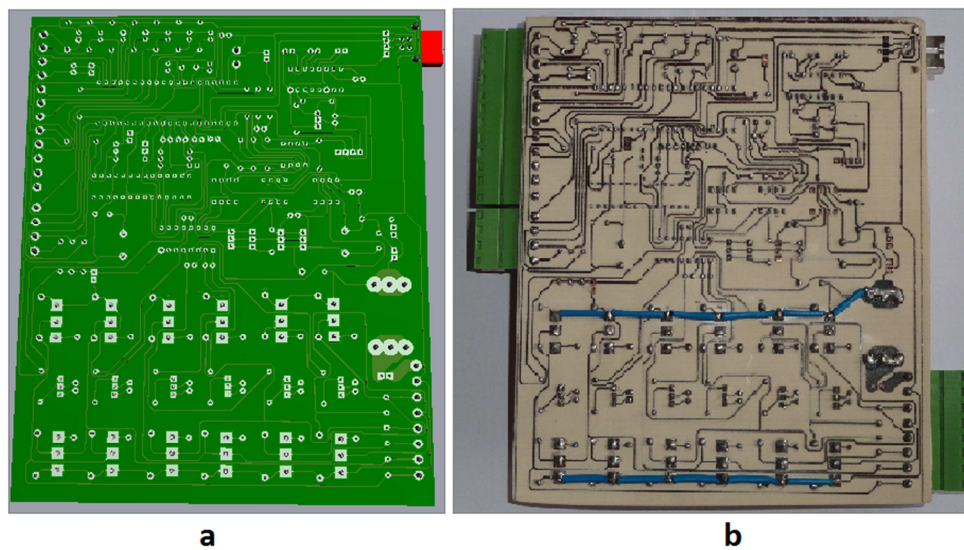


Figura 3. 322 Vista de la TAC, capa inferior. a). Vista 3D generada en software ARES. b). Vista real.

CAPÍTULO 4

DESARROLLO DEL SOFTWARE.

4.1. PICC.

El software PICC de la empresa de desarrollo y servicios computacionales CCS, ofrece una herramienta integrada para desarrollar y depurar aplicaciones embebidas que se ejecutan en Microchip PIC ® y en otros CI programables.

Esta herramienta de programación de microcontroladores contiene los operadores estándar del lenguaje C y funciones incluidas en librerías que son específicas para los registros de PIC, proporcionando a los desarrolladores una herramienta poderosa para acceder a las funciones de dispositivos desde el nivel de lenguaje C (CSS, 2008). A continuación se detalla la configuración, así como las subrutinas aplicadas en el control de los actuadores lineales

4.1.1. Palabras de Control.

Las palabras de control o también llamados fusibles, son comandos iniciales de configuración del compilador los cuales generan el código necesario para instanciar todas las propiedades del uC, ya sea para configurar el oscilador, WatchDogTimer, pin de reset, protección del código, entre otros. Se muestran los fusibles a ser usados en el programa en la tabla 4.1.

Nombre de Fusible	Función
MCLR	Pin Master Clear Activado
NOWDT	No se utiliza el WatchDogTimer
HSPLL	Utilizamos un cristal de alta velocidad \geq a 4MHZ en conjunto con el PLL de nuestro pic.
NOPROTECT	Código no protegido contra lectura
NOLVP	Desactivamos programación en bajo voltaje
NODEBUG	No depuración
USBDIV	Fuente de reloj USB viene de PLL dividido para 2
PLL5	Significa que el PLL prescaler dividirá en 5 la frecuencia del cristal. para HS = $20\text{Mhz}/5 = 4\text{Mhz}$
CPUDIV1	El PLL postscaler divide en 2 la frecuencia de salida del PLL de 96MHZ, si queremos 48MHZ
VREGEN	Habilita el regulador de 3.3 volts que usa el módulo USB

Tabla 4. 1 PICC - Lista de Fusibles utilizados

El uso de los fusibles HSPLL, PLL5 y CPUDIV1, correctamente configurados nos permiten trabajar con una velocidad de procesamiento de 48Mhz, teniendo físicamente conectado al uC, un cristal de 20 Mhz, como se ve en la tabla, al tener PLL5 se divide la frecuencia del cristal para 5, es decir $20/5$, los 4Mhz son necesarios para que el PLL puesta trabajar con 48Mhz y para realizar la comunicación USB.

4.1.2. Librerías.

Las librerías no son más que paquetes de código que contienen variables, parámetros de configuración y funciones preestablecidas, a las cuales se las puede utilizar libremente dentro del código principal.

En la siguiente tabla se muestra la librería para el manejo del puerto USB.

Librería	Descripción	Comandos	Función
usb_cdc.h	Biblioteca para añadir un puerto COM virtual en la PC, a través del puerto USB con el dispositivo estándar de Clase para Comunicación de Dispositivos (CDC).	usb_cdc_init()	Configuramos el puerto virtual.
		usb_init()	Inicializamos el stack USB.
		usb_cdc_kbhit()	Verifica la presencia de datos en el buffer del USB.
		usb_cdc_getc()	Obtiene un carácter del búfer de recepción.
		usb_cdc_putc(char c)	Pone un carácter en el buffer de transmisión.
		usb_cdc_puts(*ptr)	Pone una cadena de caracteres por el buffer de transmisión.

Tabla 4. 2 Librería y Comandos para el manejo del puerto USB.

Así también se facilita el uso de la memoria de datos EEPROM con las siguientes librerías.

Librería	Descripción	Comandos	Función
2408b.c	Librería para la utilización de una memoria de datos EEPROM 24LC08B.	init_ext_eeprom()	Rutina de Inicialización
		ext_eeprom_ready()	Retorna TRUE se la EEPROM está lista para recibir comandos
		write_ext_eeprom(a, d)	Escribe el byte d a la dirección a
		read_ext_eeprom(a)	Lee el byte guardado en la dirección a

external_eeprom.c	Permite escribir datos de diferentes tipos en las direcciones de memoria de la EEPROM	write_int32_ext_eeprom(a, int32 data)	Escribe en la dirección a, un número data de 32 bits
		read_int32_ext_eeprom(a)	Lee de la dirección a, un número de 32 bits
		write_float_ext_eeprom(a, float data)	Escribe en la dirección a, un número data tipo flotante
		read_float_ext_eeprom (a)	Lee de la dirección a, un número tipo flotante

Tabla 4. 3 Librería y Comandos para el manejo de la Memoria EEPROM.

A demás de estas librerías, se utilizó la librería stdlib.h que se muestra a continuación.

Librería	Descripción	Comandos	Función
stdlib.h	Librería para manejo de datos numéricos y vectores	itoa(int32 num, unsigned int base, char * s)	Retorna en el vector s, los caracteres correspondientes al número num de 32 bits con signo

Tabla 4. 4 Librería y Comando para el manejo de vectores.

4.1.3. Configuración del Módulo ADC.

Como sabemos, el PIC18F4550 puede trabajar con resoluciones de 8, 10 y 16 bits para la conversión de las señales analógicas en datos digitales, en este caso se trabaja con la resolución de 10 bits con lo que se obtiene un rango de conversión de voltaje de 0 a 5 voltios, con los correspondientes datos digitales de 0 a 1023.

Para configurar la resolución del ADC del uC se utiliza la siguiente instrucción:

#deviceadc=10.

Además se utilizan los comandos:

setup_adc_ports(AN0_TO_AN5/VSS_VDD), comando con el que se definen los 6 primeros puertos como entradas analógicas.

setup_adc(ADC_CLOCK_DIV_64), con esto se configura al convertor para que al realizar la conversión de la señal analógica haga las comparaciones cada 12,6 us.

set_adc_channel(K), comando para setear el canal con el que se quiere realizar la conversión. En este caso K puede tomar el valor de 0 a 6 (refiérase al capítulo 3.5.12).

read_adc(), retorna un dato tipo int32 referente al valor de la conversión análoga a digital del canal seteado.

Con estos parámetros podemos ver la calidad de la conversión análoga a digital del PIC18F4550, en la siguiente figura se muestra el error en dicha conversión.

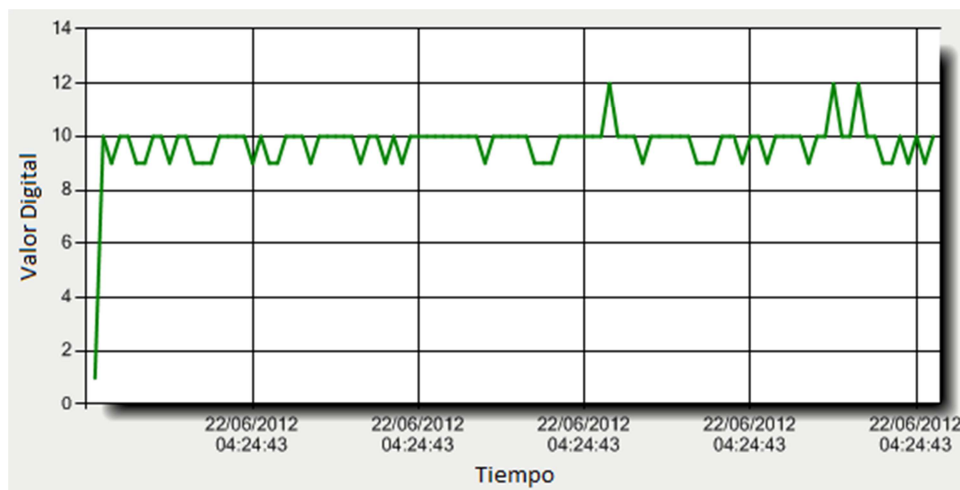


Figura 4. 1 Error de conversión del ADC.

Para esto se aplica en el canal análogo 0 (PIN RA0), el voltaje de 0.049V, cuyo valor digital equivale a 10, se puede ver que hay un pico de 12 y un mínimo de 9, con lo que se tiene un error máximo de ± 2 equivalente a 19mV (0.38% en el rango de 0 – 5Vdc). Se puede decir que el error de conversión es prácticamente despreciable.

4.1.4. Configuración del Módulo PWM.

El módulo PWM es utilizado para generar las salidas análogas de control. En la siguiente figura se setea una frecuencia para la señal PWM de 2 kHz.

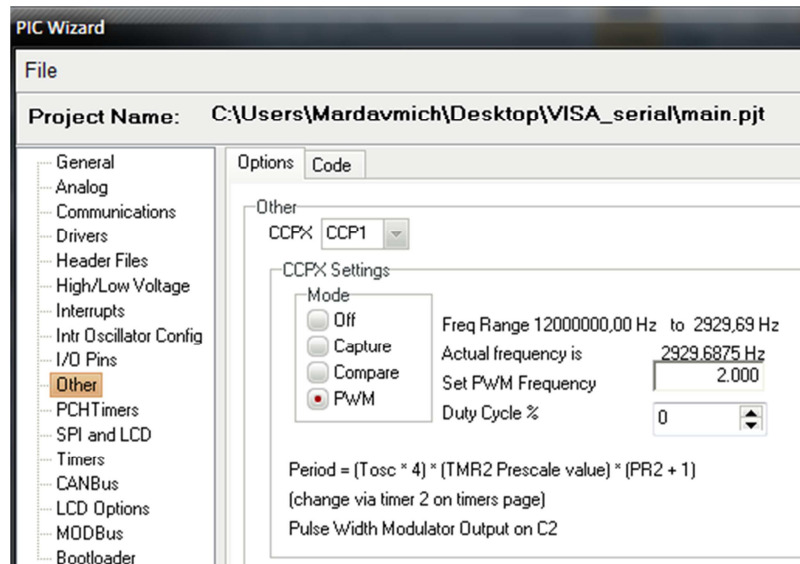


Figura 4. 2 Seteo de frecuencia para la señal PWM.

Con lo que en el wizard, auto configura el timer 2 para un periodo de interrupción de 5ms mediante el siguiente código:

```
setup_timer_2(T2_DIV_BY_16,255,16);
```

Los demás timers quedan deshabilitados. Con esto los comandos utilizados para el manejo del módulo PWM son los siguientes:

- *setup_ccp1(CCP_PWM)*, inicializa el módulo PWM 1.
- *setup_ccp2(CCP_PWM)*, inicializa el módulo PWM 2.
- *set_pwm1_duty(num)*, setea el tiempo en alto de la del módulo PWM 1 (PIN 17), donde *num* puede tomar valores de 0 a 255.
- *set_pwm2_duty(num)*, misma función anterior pero para el módulo PWM 2 (PIN 16).

4.1.5. Comunicaciones.

4.1.5.1. RS-232.

Para agregar comunicación serial RS-232 al uC se utiliza la siguiente instrucción:

```
#use rs232(baud=9600,parity=N,xmit=PIN_C6,rcv=PIN_C7,bits=8)
```

Con esto se setean parámetros como la velocidad (9600 baudios), paridad, pin transmisor (PIN_C6), pin receptor (PIN_C7) y el número de bits.

Para leer y escribir en el puerto serial se utilizan los siguientes comandos:

getc(), recibe un byte a través del pin receptor.

putc(), envía un byte por medio del pin emisor.

A más de esto, para optimizar los recursos del uC se usa la interrupción de datos disponibles en el puerto serial, cuyo código se muestra a continuación:

```
#int_rda  
voidrda_isr(void){  
  if(kbhit()){ dato_serial =getc(); }  
}
```

De modo que cuando el GPS envíe sus tramas, y el uC esté realizando cualquier otra tarea, pueda interrumpir el proceso actual y leer dichas tramas en *dato_serial*,

4.1.5.2. I2C.

La comunicación serial I2C se la usa para leer y escribir datos hacia la memoria EEPROM. Para implementar la comunicación I2C, se usa la siguiente instrucción.

```
#use i2c(Master,Fast,sda=PIN_A2,scl=PIN_A1)
```

Con esto se configure para que el microcontrolador actúe como maestro, el Pin A2 sea el pin de datos, y el Pin A1 sea el pin de señal de reloj. Para leer y escribir sobre la memoria EEPROM se usan los siguientes comandos:

```
write_float_ext_eeprom(address, data);
```

```
read_float_ext_eeprom(address);
```

Con los cuales se pueden leer y escribir las constantes y rangos de variables de nuestro controlador.

4.1.5.3. USB_CDC.

Para contar con la comunicación serial USB_CDC en el controlador, es necesario cargar la siguiente librería.

```
#include <usb_cdc.h>
```

Posterior a esto, en el método main del uC, se llama a las siguientes funciones de configuración inicial:

usb_cdc_init(), con lo que configuramos el puerto virtual COMx.

usb_init(), inicializamos el stack USB.

Para recibir un carácter desde la PC a través del puerto USB, se utiliza la siguiente función:

```
if (usb_enumerated()){  
    usb_task();  
    if (usb_cdc_kbhit()){  
        c_a=usb_cdc_getc();  
    }}
```

Finalmente para enviar un dato a través del puerto se usa la siguiente función:

```
if(usb_enumerated()){  
    usb_cdc_puts(e);  
    }
```

Donde *e* es una cadena de caracteres.

4.1.6. Requerimientos Generales del Sistema de Control.

El sistema de control en general se define en la siguiente figura.

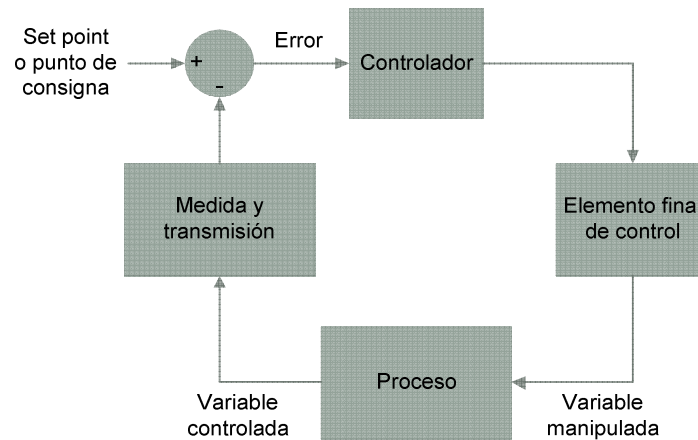


Figura 4. 3 Definición del sistema de control

El bucle de control mostrado, estará formado de los siguientes elementos:

- Elementos de medida o sensores.
- Elemento de control lógico o controlador, el cual compara la variable medida con la deseada (punto de consigna) para determinar el error, y estabilizar el sistema mediante la señal de control adecuada para reducir o eliminar el error.
- Elementos finales de control, estos reciben la señal del controlador y cambian directamente el valor de la variable manipulada afectando a la variable controlada.

Además es un servosistema retroalimentado tipo MIMO (múltiples entradas, múltiples salidas). La especificación de servosistema se debe a que su salida es algún elemento mecánico, en este caso es la posición. Las señales más significativas del sistema de control son las siguientes:

- La señal de referencia o punto de consigna, proveniente de los gatillos, calibrada en función al rango deseado.
- La variable controlada, es la posición de cada uno de los pedales.
- La variable manipulada, es el movimiento obtenido por los actuadores lineales debido a la cantidad de voltaje entregado al puente H el cual comanda los actuadores así como la señal digital del sentido de giro para el multiplexor.

➤ La variable activa o el error, es la diferencia entre el punto de consigna y la variable controlada

En términos generales el controlador está compuesto por los siguientes bloques funcionales.

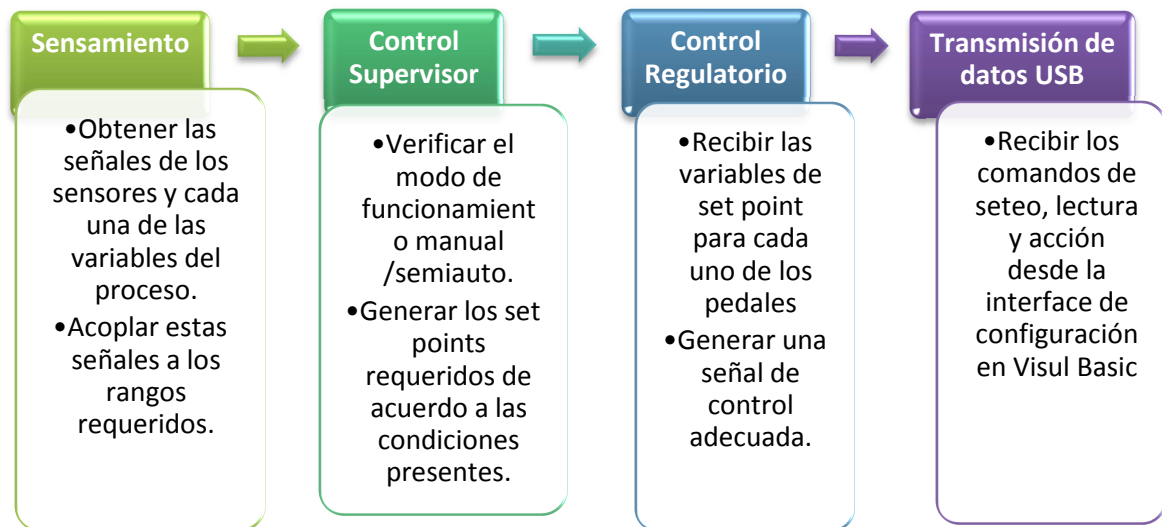


Figura 4. 4 Sistema controlador, bloques funcionales

En primer lugar está la lectura de las variables del proceso, seguido del control supervisor el cual genera los set points, los cuales posteriormente envía al control regulatorio difuso el que genera las señales de control que son enviadas a los actuadores, finalmente se tiene la transmisión de datos USB, esencial para el seteo y lectura de parámetros de control así como de verificación de desempeño del controlador difuso desde el Visual Basic.

4.1.7. Programación del PIC

A continuación se muestra de forma detallada el manejo de variables y los diagramas de flujo y referencias al código en PICC de cada uno de los bloques funcionales.

4.1.7.1. Manejo de variables.

El número de variables que maneja el controlador es relativamente alto, debido a que para el control regulatorio difuso se necesitan los rangos de las funciones de pertenencia

del error y cambio del error de cada pedal, así como los rangos de los universos de discurso entre otras variables.

Para esto se utilizará un vector de variables llamado “var[]” de tipo signed long (entero de 16 bits con signo), se lo definió de la siguiente forma.

```
//Todas las variables

long const nTotalVariables = 280;

signed long var[nTotalVariables];
```

Este almacenara todas las variables involucradas en los diferentes procesos de control, al alimentar la tarjeta, se recorre mediante un bucle las direcciones de la EEPROM y se los asigna en el vector var[]. El código de lectura de las variables guardadas se muestra en [ANEXOS: 2. Lectura de las variables del proceso; d). Leer constantes guardadas en EEPROM].

En la siguiente tabla se muestra los límites de los universos de discurso de los gatillos, pedales y rangos de inclinación, donde el prefijo puede ser (emb, fre, ace) referenciando a cada pedal o gatillo, la letra central (g, p) nos muestra si es gatillo o pedal, finalmente el prefijo (min, max) que indica si es un valor mínimo o un valor máximo.

LÍMITES DE SENSORES			
	Dirección var[]	Dirección EEPROM	Variable
Gatillos	0	0	emb_g_min
	1	2	emb_g_max
	2	4	fre_g_min
	3	6	fre_g_max
	4	8	ace_g_min
	5	10	ace_g_max
Pedales	6	12	emb_p_min
	7	14	emb_p_max
	8	16	fre_p_min
	9	18	fre_p_max

	10	20	ace_p_min
	11	22	ace_p_max
Inclinación	12	24	vOffset
	13	26	inc_min
	14	28	inc_max

Tabla 4. 5 Direcciones para los límites de universos de discurso.

Así mismo para mantener los datos en memoria no volátil, este vector de variables se guarda en la EEPROM, para direccionar datos en esta memoria, los direccionamos siempre a bytes, es decir, la dirección cero es el byte de memoria cero, y así para cada dirección que le asignemos tanto de lectura como de escritura. Dado que el dato signed long tiene un tamaño de memoria de 2 bytes, los datos se graban en la EEPROM cada 2 bytes como se indica en la tabla anterior en la columna Direcciones EEPROM

Para los límites de cada una de las funciones de pertenencia, para cada pedal y gatillo, se muestra en la tabla 4.6. La dirección de la variable correspondiente.

VARIABLES CONTROLADOR									
	Dirección var[]	Dirección EEPROM	Variable	Dirección var[]	Dirección EEPROM	Variable	Dirección var[]	Dirección EEPROM	Variable
	15	30	gn1_e	65	130	gn1_f	115	230	gn1_a
	16	32	gn2_e	66	132	gn2_f	116	232	gn2_a
	17	34	gn3_e	67	134	gn3_f	117	234	gn3_a
	18	36	n1_e	68	136	n1_f	118	236	n1_a
	19	38	n2_e	69	138	n2_f	119	238	n2_a
	20	40	n3_e	70	140	n3_f	120	240	n3_a
	21	42	ne1_e	71	142	ne1_f	121	242	ne1_a
Func.	22	44	ne2_e	72	144	ne2_f	122	244	ne2_a
Pert.	23	46	ne3_e	73	146	ne3_f	123	246	ne3_a
	24	48	z1_e	74	148	z1_f	124	248	z1_a
	25	50	z2_e	75	150	z2_f	125	250	z2_a
	26	52	z3_e	76	152	z3_f	126	252	z3_a
	27	54	z4_e	77	154	z4_f	127	254	z4_a
	28	56	ze1_e	78	156	ze1_f	128	256	ze1_a
	29	58	ze2_e	79	158	ze2_f	129	258	ze2_a
	30	60	ze3_e	80	160	ze3_f	130	260	ze3_a

31	62	ze4_e	81	162	ze4_f	131	262	ze4_a
32	64	p1_e	82	164	p1_f	132	264	p1_a
33	66	p2_e	83	166	p2_f	133	266	p2_a
34	68	p3_e	84	168	p3_f	134	268	p3_a
35	70	gp1_e	85	170	gp1_f	135	270	gp1_a
36	72	gp2_e	86	172	gp2_f	136	272	gp2_a
37	74	gp3_e	87	174	gp3_f	137	274	gp3_a
38	76	pe1_e	88	176	pe1_f	138	276	pe1_a
39	78	pe2_e	89	178	pe2_f	139	278	pe2_a
40	80	pe3_e	90	180	pe3_f	140	280	pe3_a

Tabla 4. 6 Direcciones para los límites de funciones de pertenencia para el error y cambio del error.

Las variables sombreadas son los límites de las funciones de pertenencia del cambio del error. En la tabla anterior el sufijo (e, f, a) hace referencia al pedal, para entender mejor que significa cada variable, en la siguiente figura se muestra los rangos de las funciones de pertenencia del error y cambio del error de donde provienen estas variables.

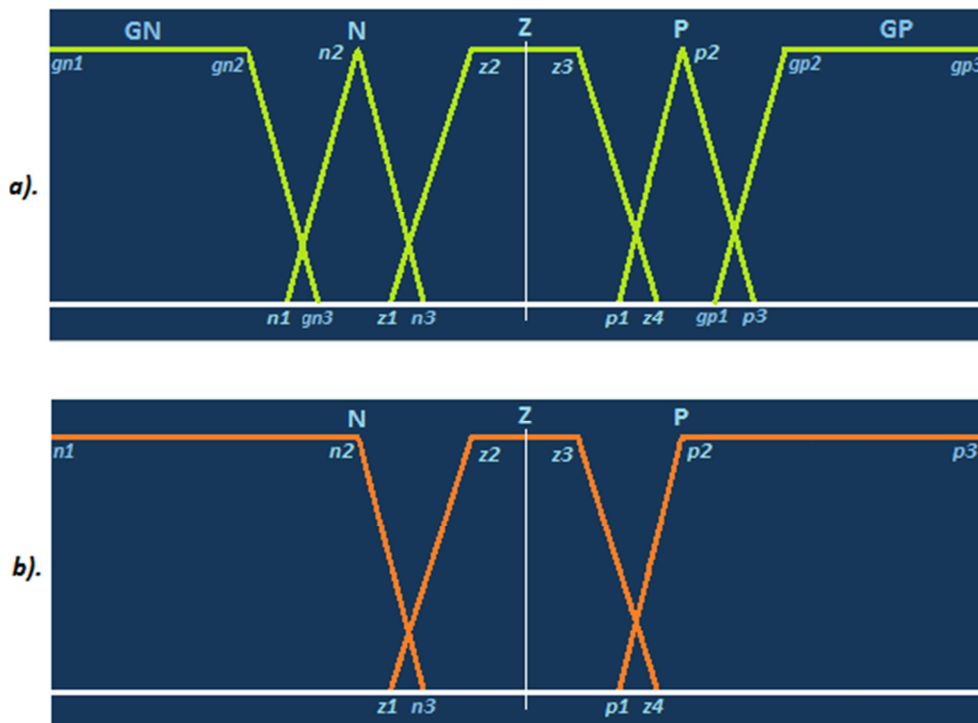


Figura 4. 5 Funciones de pertenencia: a). Error. b). Cambio del error.

Así mismo, los valores de los singletons de salida de la señal de control se muestran a continuación.

Dirección var[]	Dirección EEPROM	Var.	Dirección var[]	Dirección EEPROM	Var.	Dirección var[]	Dirección EEPROM	Var.
41	82	ugn_e	91	182	ugn_f	141	282	ugn_a
42	84	un_e	92	184	un_f	142	284	un_a
43	86	uz_e	93	186	uz_f	143	286	uz_a
44	88	up_e	94	188	up_f	144	288	up_a
45	90	ugp_e	95	190	ugp_f	145	290	ugp_a

Tabla 4. 7 Direcciones para las posiciones de los singletons

El prefijo (u) indica que es un singleton de salida y el sufijo (e, f ,a) muestra el pedal. Cada variable referenciada en la tabla anterior se muestra en la siguiente figura.

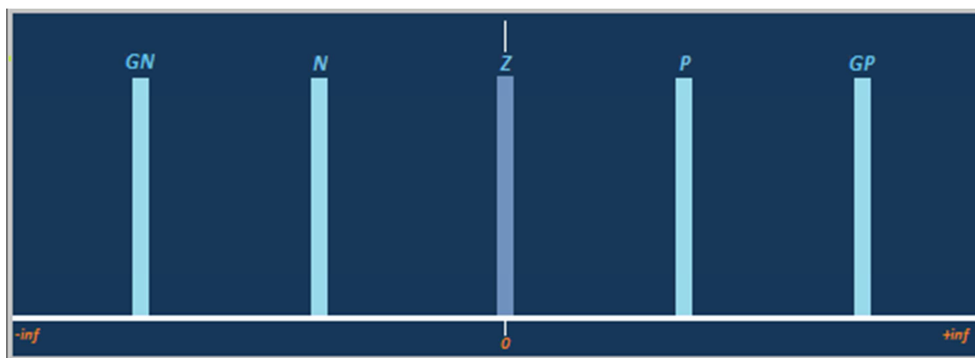


Figura 4. 6 Singletons de salida

Las constantes para la base de reglas se muestran en la tabla 4.8.

	Dirección var[]	Dirección EEPROM	Variable	Dirección var[]	Dirección EEPROM	Variable	Dirección var[]	Dirección EEPROM	Variable
Base de Reglas	49	98	gnn_e	99	198	gnn_f	149	298	gnn_a
	50	100	nn_e	100	200	nn_f	150	300	nn_a
	51	102	zn_e	101	202	zn_f	151	302	zn_a
	52	104	pn_e	102	204	pn_f	152	304	pn_a
	53	106	gpn_e	103	206	gpn_f	153	306	gpn_a
	54	108	gnz_e	104	208	gnz_f	154	308	gnz_a
	55	110	nz_e	105	210	nz_f	155	310	nz_a
	56	112	zz_e	106	212	zz_f	156	312	zz_a
	57	114	pz_e	107	214	pz_f	157	314	pz_a

58	116	gpz_e	108	216	gpz_f	158	316	gpz_a
59	118	gnp_e	109	218	gnp_f	159	318	gnp_a
60	120	np_e	110	220	np_f	160	320	np_a
61	122	zp_e	111	222	zp_f	161	322	zp_a
62	124	pp_e	112	224	pp_f	162	324	pp_a
63	126	gpp_e	113	226	gpp_f	163	326	gpp_a

Tabla 4. 8 Direcciones para las constantes de la base de reglas

En la base de reglas, para cada pedal se tiene un vector que va desde GNN hasta GPP, recorriendo todas las combinaciones del cambio del error en función del error, este vector contiene los valores de los singletons asociados para cada condición, es decir, un número del 0 al 4 que hace referencia al singleton GN, N, Z, P, GP respectivamente.

		ERROR				
		GN	N	Z	P	GP
CAMBIO DEL ERROR	N	GN	P	Z	P	GP
	Z	GN	N	GN N	P	GP
	P	GN	N	Z P GP	N	GP

Figura 4. 7 Base de reglas

También para cada pedal se tienen las constantes de las ganancias Kp, Kd y Ku del controlador, mostradas en la siguiente tabla.

	Dirección var[]	Dirección EEPROM	Variable	Dirección var[]	Dirección EEPROM	Variable	Dirección var[]	Dirección EEPROM	Variable
Ganancias	46	92	kp_e	96	192	kp_f	146	292	kp_a
	47	94	kd_e	97	194	kd_f	147	294	kd_a
	48	96	ku_e	98	196	ku_f	148	296	ku_a

Tabla 4. 9 Direcciones para las ganancias

Finalmente en la siguiente tabla se direccionan las variables del proceso. Estas no se guardan en la EEPROM ya que se miden y recalculan con cada nueva iteración del controlador.

VARIABLES DEL PROCESO		
	Dirección	Variable
	var[]	
Embrague	250	emb_g
	251	emb_p
	252	emb_v
	253	emb_u
	254	emb_e
Freno	255	fre_g
	256	fre_p
	257	fre_v
	258	fre_u
	259	fre_e
Acelerador	260	ace_g
	261	ace_p
	262	ace_v
	263	ace_u
	264	ace_e
Inclinación	265	inc
Velocidad	266	vel

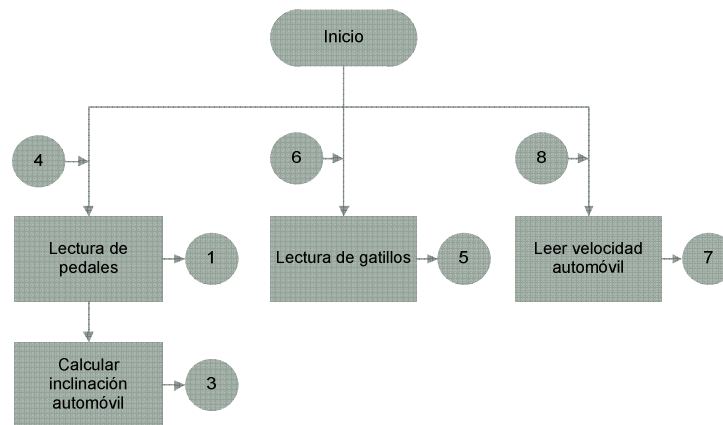
Tabla 4. 10 Direcciones para las variables del proceso

Donde, a parte de las variables de la inclinación y la velocidad del automóvil, se tienen las variables referentes al embrague, freno y acelerador, para estos se define la siguiente descripción de los sufijos.

Sufijo	Variable
_g	gatillo
_p	pedal
_v	cambio del error
_u	señal de control
_e	error

Tabla 4. 11 Descripción de sufijos para variables**4.1.7.2. Sensamiento.**

Para obtener las señales de las variables del proceso de todos los sensores, linealizarlas y guardarlas se plantea la siguiente secuencia lógica.

**Figura 4. 8 Diagrama de flujo de lectura de variables**

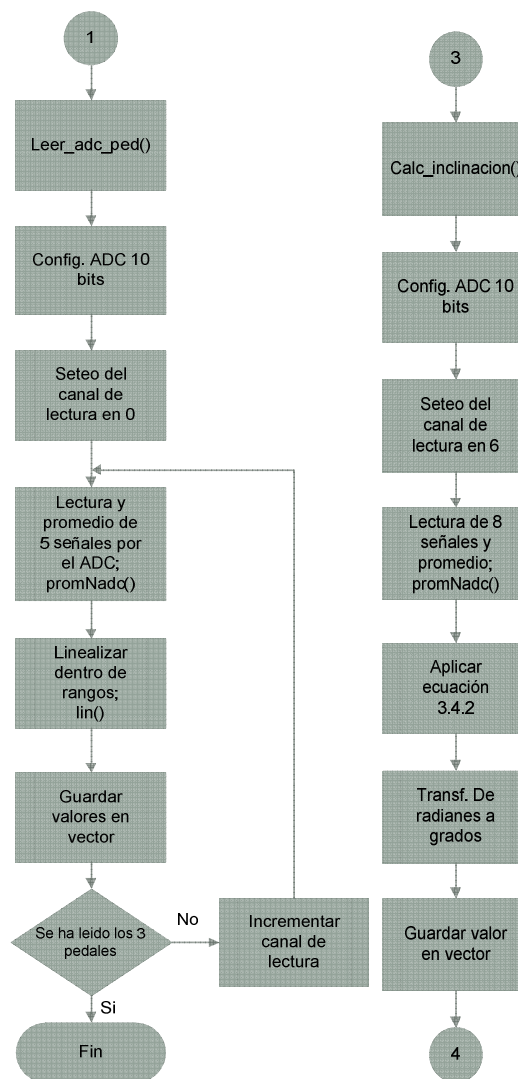


Figura 4. 9 Diagramas de flujo: a). Para la lectura de los pedales. b). Para calcular la inclinación del automóvil.

El código en PIC para leer_adc_ped() y para calc_inclinacion() se muestra en [ANEXOS: 2. Lectura de las variables del proceso; a). Leer las señales de los pedales e inclinación del automóvil].

En los gatillos sin embargo, la secuencia de lectura es un tanto diferente, ya que se utiliza un algoritmo para “suavizar” la lectura proveniente de estos, como se ve en el siguiente diagrama.

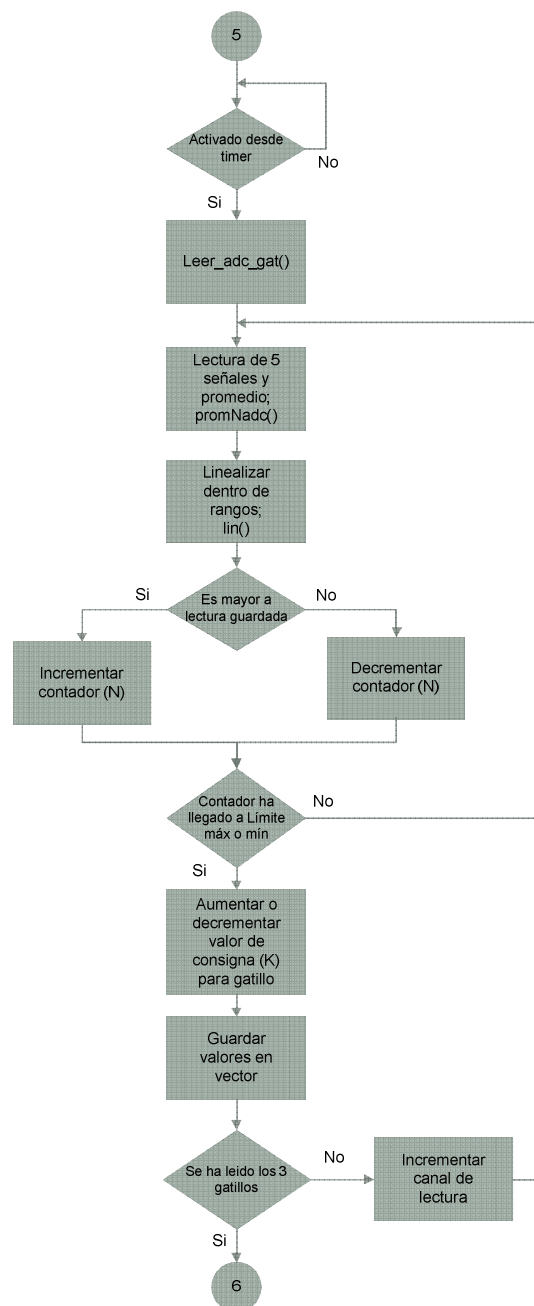


Figura 4. 10 Diagrama de flujo para lectura de gatillos

La lectura se activa cada determinado tiempo por el timer, definido por defecto cada 5ms, el cual al activarse llama a la función leer_adc_gat() detallado en [ANEXOS: 2. Lectura de las variables del proceso; b). Leer señales de gatillos], aquí se tiene un contador para cada pedal el cual aumenta cada que una nueva lectura es mayor a la lectura guardada previamente, caso contrario se decrementa, así, si se ha llegado al valor máximo del contador, se aumenta K (valor pre configurado) a la señal y se la guarda, del mismo modo si se ha llegado al valor mínimo, se resta K a la señal y se la guarda, la variable K es una

variable manipulada por el control supervisor, en la siguiente tabla se indica la dirección de las variables para los tres gatillos.

	Dirección var[]	Dirección EEPROM	Variable
Embrague	164	328	emb_gInc
	165	330	emb_gIncCalc
Freno	166	332	fre_gInc
Acelerador	167	334	ace_gInc

Tabla 4. 12 Variables de incremento de lectura de gatillos

Donde

Para la lectura de la velocidad del automóvil desde el GPS se tiene el siguiente diagrama de flujo.

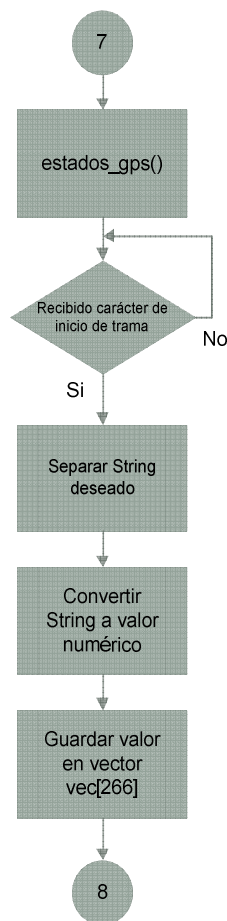


Figura 4. 11 Diagrama de flujo para lectura de la velocidad del automóvil

El código de la función `estados_gps()` se muestra en [ANEXOS: 2. Lectura de las variables del proceso; c). Leer velocidad del automóvil desde GPS], donde se maneja una lógica de estados para ir

4.1.7.3. Sistema de Control Supervisor.

El control supervisor es el encargado de verificar el modo de funcionamiento manual / semiautomático en el cual se va a trabajar, esto se verifica al iniciar “alimentar” la tarjeta de control mediante la entrada digital PIN_C0, siendo 0 lógico para el modo manual y 1 lógico para el modo semiautomático como se muestra en la siguiente gráfica, con esto, genera los set points requeridos de acuerdo a las condiciones presentes.

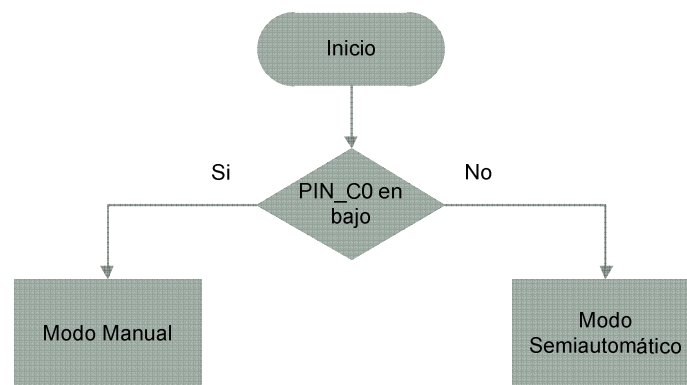


Figura 4. 12 Decisión de modo, control Supervisor

El código general del sistema supervisor está en [ANEXOS: 3. Sistema de control supervisor, a). Estados del control supervisor].

4.1.7.3.1. Modo Manual.

En el modo manual se tiene un control individual de cada actuador lineal, uno con cada gatillo, siendo este un movimiento independiente de un actuador lineal respecto de otro.

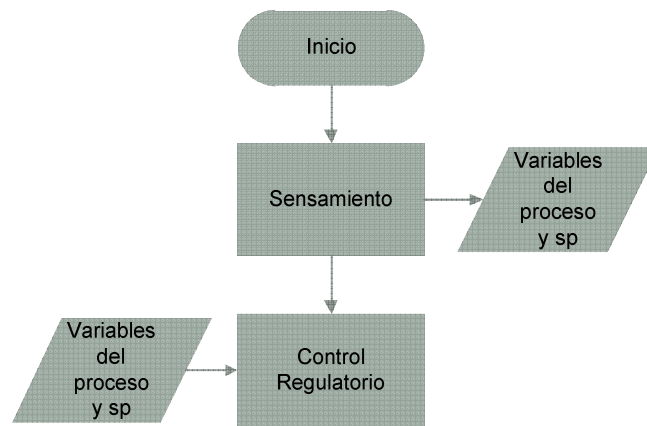


Figura 4. 13 Modo manual

Como se indica en la gráfica anterior en el modo manual en primer lugar se realiza el sensamiento de variables del proceso y de setpoints, enviando estos directamente al control regulatorio el cual genera las señales de control (refiérase al capítulo 4.1.7.4).

4.1.7.3.2. Modo Semiautomático

En el modo semiautomático el usuario solo dispone de dos gatillos, uno para el control independiente del actuador lineal sobre el freno, haciendo básicamente un control manual sobre este, y el segundo gatillo (gatillo de aceleración) que controla los actuadores sobre el acelerador y embrague simultáneamente, para esto, como parámetros de control se requiere conocer la inclinación del automóvil y la señal del gatillo de aceleración especialmente al momento del arranque, esto es para saber en qué ángulo está respecto a la horizontal, este dato es importante para determinar la velocidad de salida del embrague, esencialmente si al está en cuesta la velocidad de salida del actuador sobre el embrague es menor que cuando este en bajada; otro parámetro importante al momento que el auto este en movimiento es el pulsador indicador de cambio de marcha, lo que indica este es que el conductor se dispone a realizar un cambio de marcha, por lo que mientras se mantenga activado este pulsador, se realiza el control de posición del embrague inverso al del acelerador; además se sensará la velocidad del automóvil con el objetivo de limitar la aceleración para una velocidad máxima.

De este modo se pretende brindar un control más amigable para el conductor, a continuación se muestra el diagrama de flujo del modo semiautomático.

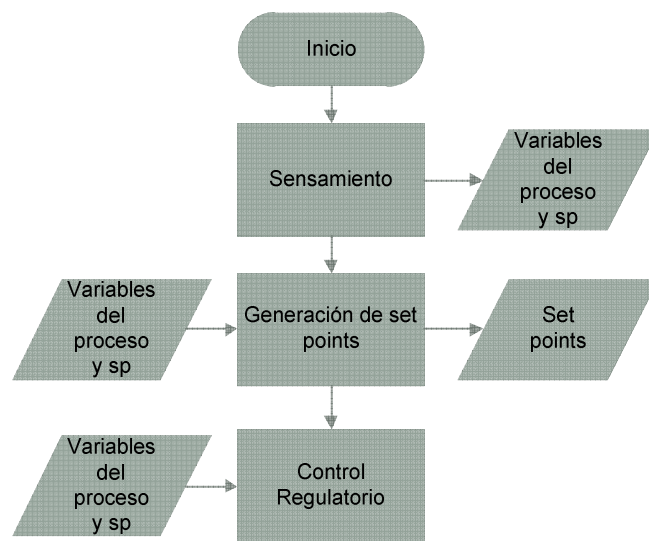


Figura 4. 14 Modo semiautomático

Donde el bloque de Generación de set points se muestra a continuación.

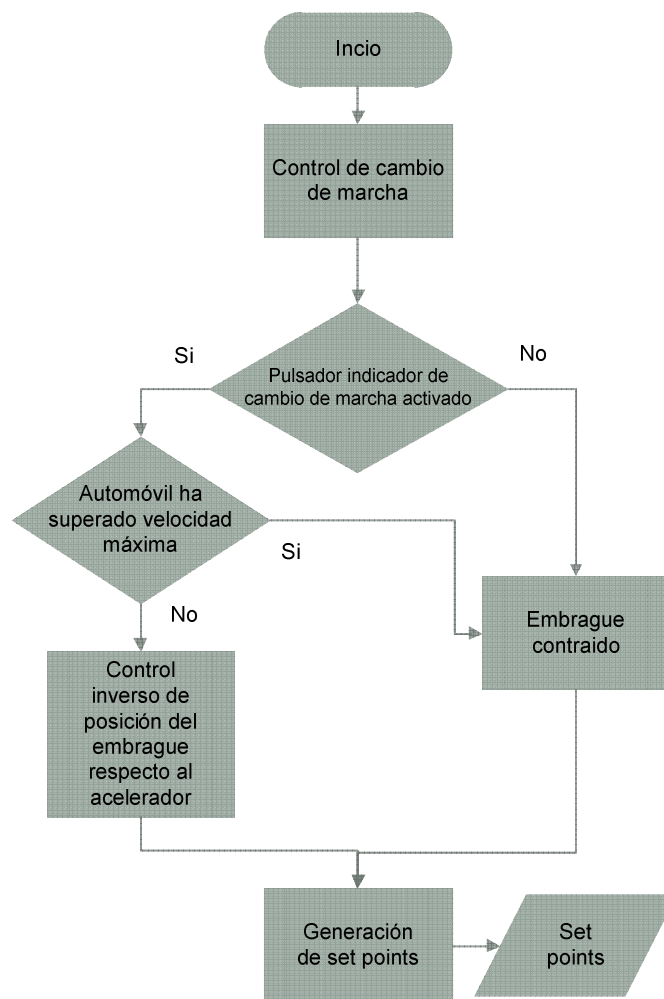


Figura 4. 16 Control de cambio de marcha

Luego, si se supera la velocidad máxima configurada, empieza a limitar la aceleración, retrayendo el acelerador y el embrague; finalmente si se ha entrado de nuevo al rango de velocidad mínima empieza nuevamente el proceso, tanto el código del modo manual como el del modo semiautomático se muestran en [ANEXOS: 3. Sistema de control supervisor, a). Estados del control supervisor].

4.1.7.4. Sistema de Control Regulatorio.

El sistema de control regulatorio tiene como objetivo el recibir las variables del proceso así como el setpoint, y mediante ciertos algoritmos proporcionar la señal de control adecuada de manera que haga cumplir los requerimientos de funcionamiento de la planta.

El sistema regulatorio a utilizar para controlar la posición de los pedales se basa en un controlador PD difuso tipo Takagi-Sugeno, este fue escogido debido a su relativo bajo coste computacional.

Este sistema difuso es más sencillo de implementar, debido a que el cálculo de la salida tiene menor costo computacional que el sistema Mandami ((IEC), 1997). Los pasos de aplicación de un sistema difuso son: fuzzificación de las entradas, evaluar las funciones de pertenencia de cada entrada de acuerdo con el valor fuzzificado, evaluación de la base de reglas y cálculo de la salida. Los pasos de fuzzificación y cálculo de la salida consisten en llevar los valores de entrada y salida a una escala definida en el sistema difuso.

En resumen, para utilizar en conjunto todas las funciones del control regulatorio explicadas a continuación, se ejecuta la función *control_regulatorio(boolean emb, boolean fre, boolean ace)* en [ANEXOS: 4. Sistema de control regulatorio; d). Ejecución del control regulatorio], el proceso se muestra resumido a continuación.

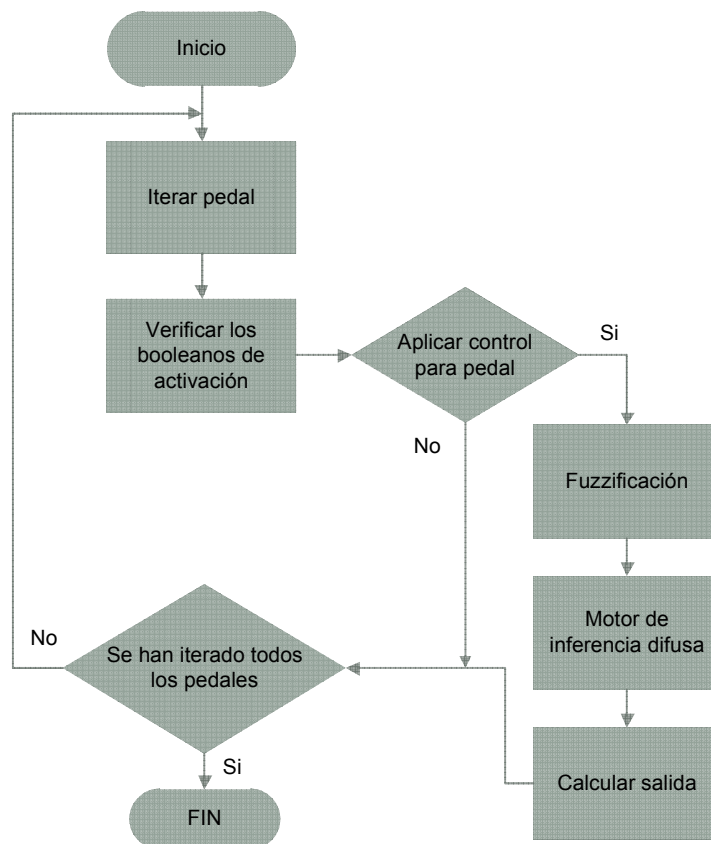


Figura 4. 17 Diagrama de flujo para el control regulatorio

A continuación, se describe la forma de programar cada paso necesario para implementar un sistema difuso tipo Takagi-Sugeno en general, para un sistema microcontrolado como el elaborado para este proyecto.

4.1.7.4.1. *Diseño de un controlador experto usando algoritmos de Lógica Difusa.*

4.1.7.4.1.1. *Definiciones generales.*

Para el diseño de este controlador se ha optado por un sistema difuso PD tipo Takagi-Sugeno, con esto a parte de evaluar el error se toma en cuenta también el cambio del error, con el objetivo de poder definir una acción de frenado, además, el uso del control derivativo “Kd” proporciona un control más fino, reduciendo el sobre pico y el tiempo de establecimiento. El sistema se muestra a continuación.



Figura 4. 18 Siendo la señal de control u en función del error y cambio del error

$$u = K_p e + K_d \dot{e}$$

Donde e es el error y \dot{e} es el cambio del error. Para el sistema micro controlado podemos considerar que

$$e = \text{setPoint} - \text{salida}$$

$$\dot{e} = e_{\text{actual}} - e_{\text{anterior}}$$

Para el cálculo del cambio del error se utilizó la herramienta del timer del microcontrolador, que cada cierto tiempo hace una lectura actual del error y la compara con una lectura anterior obteniendo así el cambio del error (refiérase al capítulo 4.1.7.2). Además se tienen tres ganancias, K_p que es la ganancia del error, K_d es la ganancia para el cambio del error y, K_u que es la ganancia de la salida de control.

En este caso, para cada pedal del automóvil, las entradas y salidas del sistema regulatorio difuso se resumen en la siguiente tabla.

	No.	Desc.	Funciones de pertenencia
Entradas	2	Error	5
		Cambio del error	3
Salidas	1	Salida análoga	5 (singletons)

Tabla 4. 13 Entradas y salidas del sistema regulatorio difuso

4.1.7.4.1.2. *Funciones de pertenencia.*

Definimos las funciones de pertenencia para cada una de las entradas, para estas funciones los valores lingüísticos que se han escogido son los siguientes:

Conjunto difuso	Símbolo
Grande Negativo	GN
Negativo	N
Cero	Z
Positivo	P
Grande Positivo	GP

Tabla 4. 14 Etiquetas lingüísticas para las funciones de salida

En este caso para el error tenemos 5 funciones de pertenencia, dos extremas de tipo trapezoidal, una trapezoidal central y dos triangulares como se muestra en la figura.

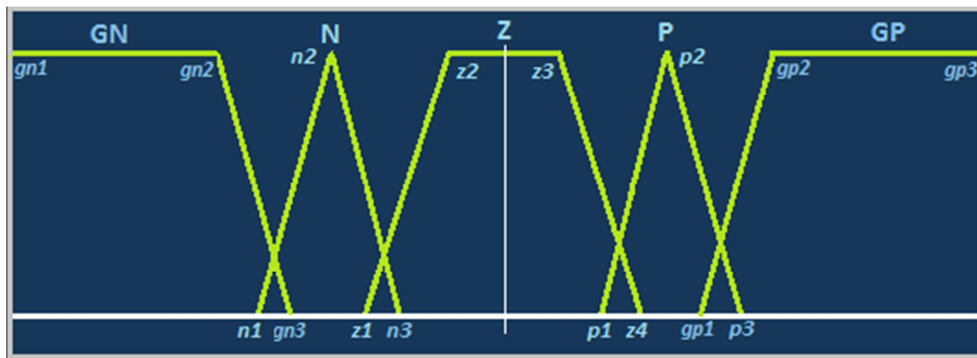


Figura 4. 19 Funciones de pertenencia para el error

Para el cambio del error tenemos tres funciones de pertenencia como se indican a continuación.

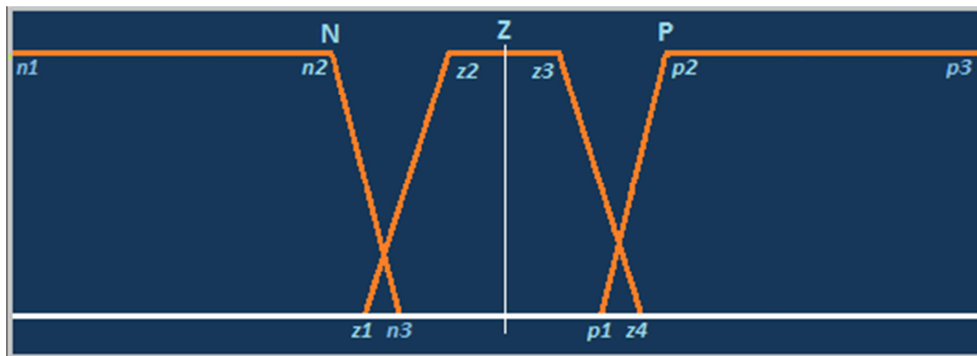


Figura 4. 20 Funciones de pertenencia para el cambio del error

El objetivo de tener una función de pertenencia de tipo trapezoidal para las dos entradas especialmente para el error, es el de definir un rango de error válido, este no debe ser tan grande como para superar el error máximo permitido 5% de error de posición, sin embargo es necesario para no sobrecalentar los actuadores ni la etapa de potencia de la tarjeta de control, ya que si se plantea otro tipo de función de pertenencia, como por ejemplo una triangular donde el error permitido es prácticamente nulo, el controlador intentará a toda costa de mantener el error en cero, invirtiendo la polaridad de los puentes H varias veces por segundo y sobrecalentando los transistores y actuadores.

Para la salida, tenemos 5 funciones de pertenencia que en realidad son constantes “singleton” o conjuntos unitarios, cada una de ellas se activa en base al proceso de inferencia difusa, que tomando la base de reglas y las funciones activas con su respectivo

valor de pertenencia específica la salida de control adecuada. La distribución de estos singletons se muestra a continuación.

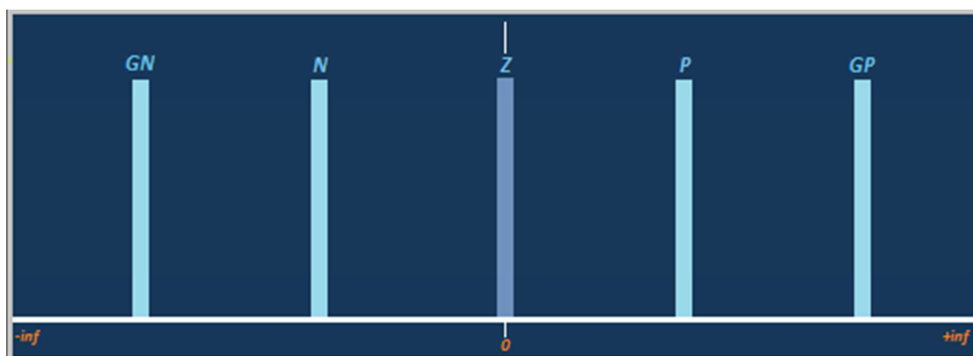


Figura 4. 21 Singletons de salida

Cabe aclarar que la definición de los rangos de cada función de pertenencia, ganancias, los rangos de singletons de salida, el límite de carrera de actuadores y gatillos se los especificará con ayuda de la interface de control (refiérase al capítulo 4.3.3).

La declaración de las variables utilizadas en el programa del controlador regulatorio difuso se presenta en la tabla 4.15.

Variable	Tipo	Rango	Descripción
funActivadas_[2][5]	boolean	(true - false)	Almacena qué funciones de pertenencia se activaron (valor lógico verdadero sí se activó y falso de lo contrario). Este facilita posteriormente la evaluación de las reglas del sistema difuso
funEval_[2][5]	float	(0 a 1)	Almacena los valores de la evaluación de las funciones de pertenencia o la certeza de cada valor lingüístico.
ws_[5]	float	(0 a 1)	Contiene los pesos de cada una de las salidas.

pesoTemp	float	(0 a 1)	Esta variable temporal almacena valores intermedios en la asignación de los valores de ws_.
s_[5]	signed long	(0 a 1023)	Contiene las posiciones de los singletons (constantes) de las funciones de la salida.
num	float	(0 a 1023)	num y den son las variables del numerador y
den	float	(0 a 1)	denominador para el cálculo de la salida
crisp	signed long	(-1023 a 1023)	Devuelve el valor de la salida del controlador difuso

Tabla 4. 15 Declaración de las variables para el controlador regulatorio

Las dimensiones de los vectores funActivadas_ y funEval_ dependen de la cantidad de entradas (filas) y el número de funciones de pertenencia (columnas), en este caso sus dimensiones son de [2][5], 2 para el número de variables de entrada y 5 para el número máximo de funciones de pertenencia, el en caso del cambio del error, se trabajará con los vectores desde el índice [1][1] a [1][3].

4.1.7.4.1.3. Definición y evaluación de las funciones de pertenencia

Las funciones de pertenencia recomendadas son del tipo triangular y trapezoidal, debido a su bajo costo computacional. Para evaluar las funciones de pertenencia de una entrada cualquiera, se emplean estructuras condicionales IF. A continuación se muestra un ejemplo de una función trapezoidal, triangular y una función extrema positiva tipo trapezoidal.

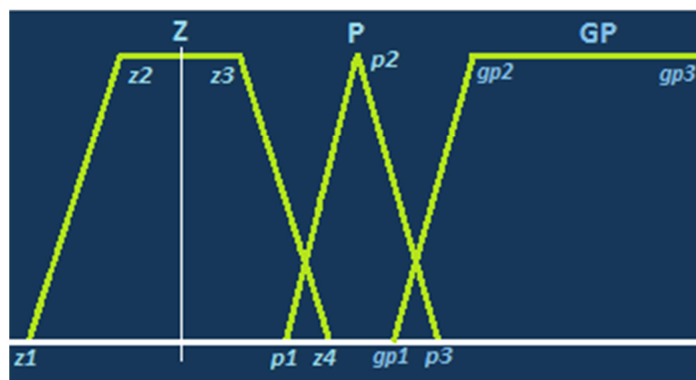


Figura 4. 22 Ejemplos de funciones de pertenencia

El diagrama de flujo para el proceso de fuzzificación se muestra en la siguiente figura.

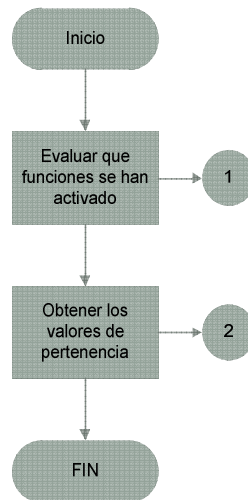


Figura 4. 23 Diagrama de flujo para la fuzzificación

Para una función trapezoidal, en primer lugar evaluamos si está en el rango de definición de la función, entonces se hace verdadero el valor en el vector de activaciones `funActivadas_` para determinada entrada y su función de pertenencia asociada. Luego mediante la ecuación de la recta calculamos el valor de pertenencia a dicha función de pertenencia, el detalle del proceso se muestra a continuación.

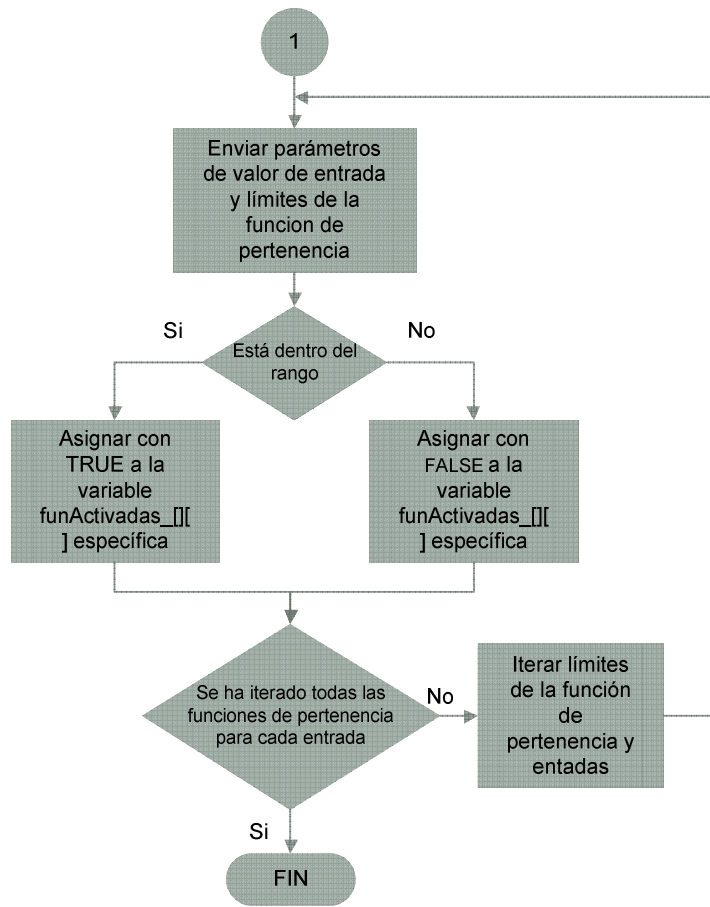


Figura 4. 24 Evaluar funciones de pertenencia activas

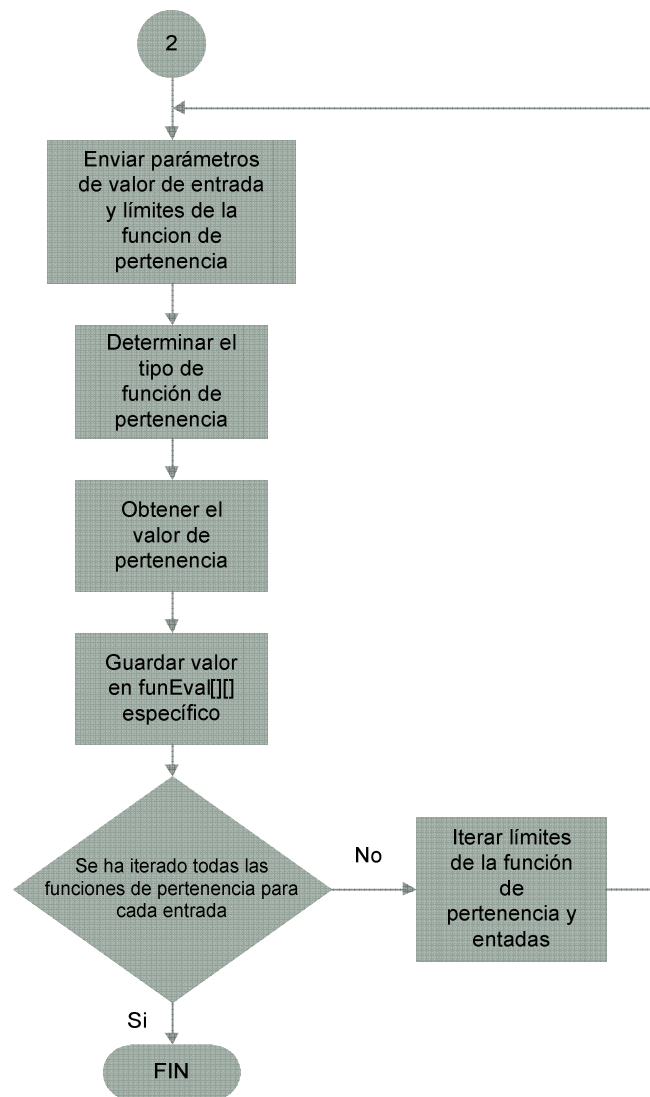


Figura 4. 25 Obtener valores de pertenencia

En [ANEXOS: 4. Sistema de control regulatorio; a). Fuzzificación] se muestra el código para determinar que funciones de pertenencia están activas y el valor de pertenencia de cada entrada para dicha función.

4.1.7.4.1.4. Métodos de implicación

Antes de definir las reglas, se plantea cuáles son los métodos de implicación entre las funciones de pertenencia de las entradas. Los métodos de implicación unen los valores fuzzificados de las entradas para generar las reglas y posteriormente la salida. En general las reglas tienen la siguiente forma:

SI U1 es FP1 Y U2 es FP1 OR U1 es FP2 Y U2 es FP1 ENTONCES U es S [1]

Las uniones “Y” generalmente se aplican seleccionando el mínimo de los valores de pertenencia obtenidos al evaluar las entradas, mientras que en las uniones “O” se aplica el máximo entre los mínimos obtenidos por las uniones “Y”. Las uniones “Y” y “O” se usan para definir las reglas difusas. Al final de las reglas, el consecuente está asociado con una función de pertenencia de la salida.

4.1.7.4.1.5. Definición y evaluación de la base de reglas

Para nuestro sistema se ha definido la siguiente base de reglas, tomando en cuenta cada una de las funciones de pertenencia para las entradas y los singletons de salida (GN, N, Z, P, GP), editable desde la interface de configuración.

		ERROR				
		GN	N	Z	P	GP
CAMBIO DEL ERROR	N	GN ▾	P ▾	Z ▾	P ▾	GP ▾
	Z	GN ▾	N ▾	GN N	P ▾	GP ▾
	P	GN ▾	N ▾	Z P GP	N ▾	GP ▾

Figura 4. 26 Base de reglas

Donde para cada condición tipo AND se define un valor de salida, en este caso se define un singleton de salida específico. Luego de haber definido la base de reglas, para evaluarla se emplea el vector booleano de activación de las funciones de pertenencia `funActivadas_[][]` y el vector que almacena el valor de pertenencia de las entradas a las funciones `funEval_[][]`, el código de la función `eval_base_reglas(int dirBaseReglas0)`, se muestra en [ANEXOS: 4. Sistema de control regulatorio; b). Motor de Inferencia Difusa], este método recibe como parámetro la variable ***dirBaseReglas0*** que indica la posición o el índice de inicio de las constantes de la base de reglas de la variable `var[]` para cada pedal, siendo para el embrague = 49, freno = 99, acelerador = 149.

El proceso del motor de inferencia difusa se muestra en la siguiente figura

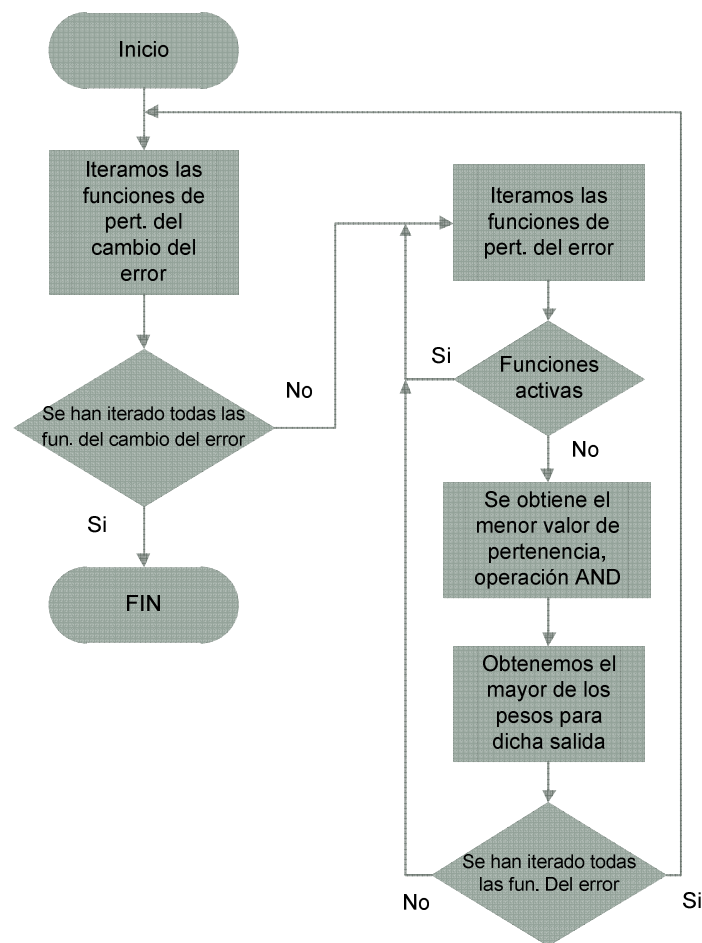


Figura 4. 27 Diagrama de flujo para el Motor de Inferencia Difusa

Cada vez que se evalúa una regla, inicialmente se almacena el mínimo entre los valores de las funciones de pertenencia que hacen parte de la regla en la variable temporal *pesoTemp*. Posteriormente, si existen varias reglas asociadas a una misma salida, entonces se compara el valor obtenido en la regla con el valor del peso almacenado previamente a la salida de dicha regla. Se asigna como peso el mayor entre los dos.

4.1.7.4.1.6. Cálculo de la salida

Como se indicó en el capítulo 2.3.2, la salida del sistema difuso tipo Takagi-Sugeno se calcula de la siguiente forma

$$Y = \frac{\sum_{i=1}^n W_{[i]} \cdot y_{[i]}}{\sum_{i=1}^n W_{[i]}}$$

Donde “ i ” es el número de funciones de pertenencia de la salida, $WS[i]$ contiene el peso máximo entre las reglas que comparten la salida $S[i]$, el código para calcular la salida del sistema difuso se muestra en [ANEXOS: 4. Sistema de control regulatorio; c). Cálculo de la salida].

4.1.7.5. Transmisión de datos USB

Para la transmisión de datos y configuración de parámetros de control desde la interface de configuración, se tiene una base de comandos, los cuales son enviados por la PC mediante la comunicación USB_CDC a la tarjeta de control, el detalle de los comandos se muestra en la siguiente tabla.

	Comando	Sufijo0acc	Sufijo 1 (dirección)	Sufijo 2 (valor)	Acción
No. caracteres	1	1	3	6	x
		x	(0 a 400)	(-32768 a 32767)	Asignar valor en la dirección de indicada
	s	f	(0 a 14)	(0 a 6)	Asignar valor mediante lectura de ADC, por el puerto especificado en Sufijo 2
		c	(0 a 14)	(0 a 186)	Asignar valor en var[Sufijo1] mediante lectura de var[Sufijo 2]
	g	x	(0 a 400)	x	Obtener valor de la dirección indicada
		-	x	x	Iniciar Comunicación
		h	x	x	Actuadores a posición Home
	a	K	x	x	Avanzar actuador del embrague
		L	x	x	Avanzar actuador del freno
		M	x	x	Avanzar actuador del

				acelerador
	k	x	x	Retroceder actuador del embrague
	l	x	x	Retroceder actuador del freno
	m	x	x	Retroceder actuador del acelerador
	s	x	x	Datos seteados y guardados en EEPROM
	n	x	x	Datos no seteados y guardados en EEPROM
	l	x	(0 a 3)	Setpoint por Gatillo
	Q	x	(0 a 1023)	Setpoint por Constante Embrague
	W	x	(0 a 1023)	Setpoint por Constante Freno
	E	x	(0 a 1023)	Setpoint por Constante Acelerador
	Z	x	x	Salir de Desempeño
	i	x	x	Leer y setear vOffset para inclinación

Tabla 4. 16 Comandos de transmisión de datos

Donde los comandos están divididos en categorías dependiendo de su comando principal, las categorías se muestran en la tabla 4.13.

Comando principal	Categoría	Descripción
s	Seteo	Setear la variable en la posición específica con el valor específico
g	Lectura	Leer la variable de la dirección específica

a	Acción	Realizar diferentes acciones dependiendo del sufijo0acc
----------	--------	---

Tabla 4. 17 Categorías de comandos de transmisión

El inicio de la comunicación y la transmisión de datos se describe en el siguiente diagrama de flujo.

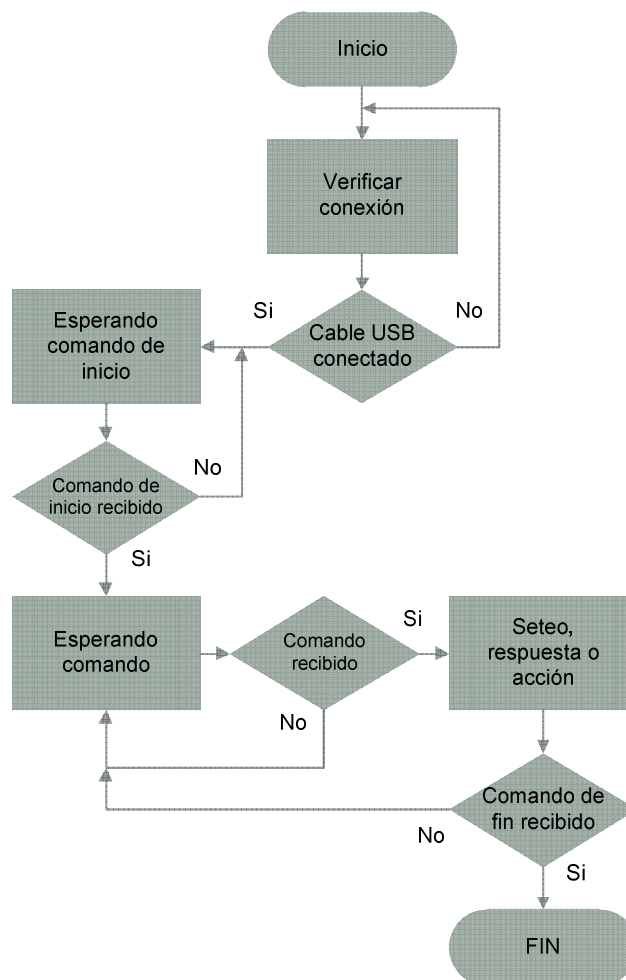


Figura 4. 28 Diagrama de flujo para el inicio y transmisión de datos

Así mismo, el código de recepción, categorización y respuesta de comandos se muestran en [ANEXOS: 5. Transmisión de datos USB. a). Leer comandos USB, b). Categorización y respuesta a comandos].

4.1.8. Generación del (.HEX)

Una vez realizado el algoritmo del sistema controlador, se debe compilar el código del proyecto mediante la herramienta “Compile” en la barra de herramientas del software PICC como se muestra a continuación.

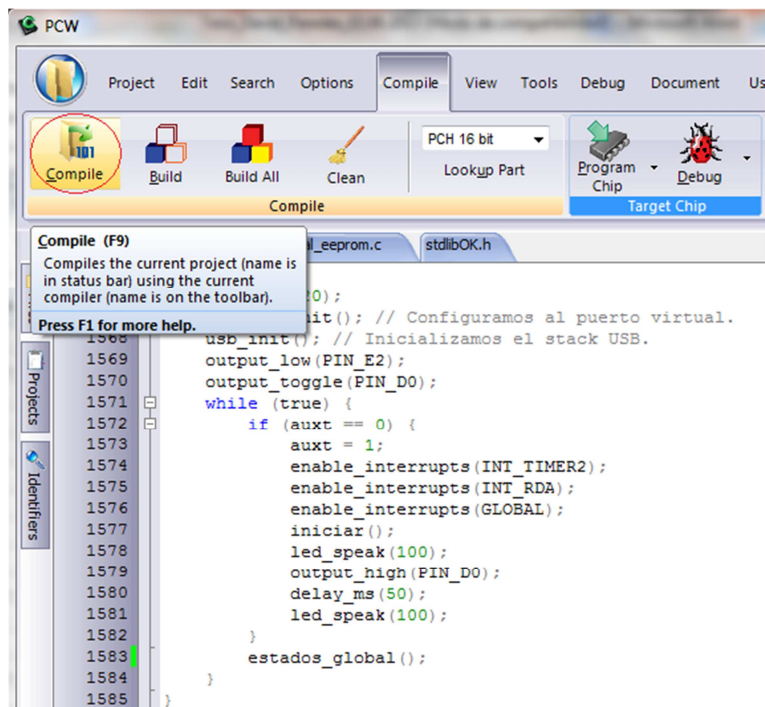


Figura 4. 29 Compilación del código en el PICC

Una vez compilado el código, se muestra una ventana, figura 4.28, la cual nos da un resumen del número de archivos utilizados, el número de funciones, el número de líneas de código, así como el porcentaje de utilización de la memoria RAM y ROM del PIC.

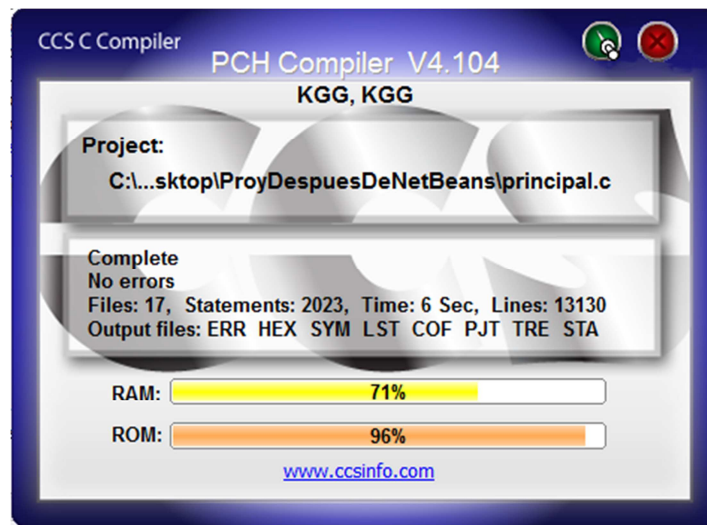


Figura 4. 30 Resumen de compilación

Finalmente como resultado de la compilación, se generan algunos archivos, entre ellos el **.hex**, mostrado en la siguiente figura, el cual se utiliza con el software PICkit 2 para programar el PIC.

Nombre	Tipo	Tamaño
external_eeprom.bak	Archivo BAK	7 KB
external_eepromOK.c	C Source File	7 KB
fuzzy.bak	Archivo BAK	3 KB
fuzzy.c	C Source File	3 KB
math_ok.bak	Archivo BAK	45 KB
MATH_OK.H	C Source File	45 KB
principal.bak	Archivo BAK	53 KB
principal.c	C Source File	53 KB
principal.cof	C Object File	461 KB
principal.err	Archivo ERR	4 KB
principal.esym	Archivo ESYM	89 KB
principal.h	C Source File	3 KB
principal.hex	C Object File	87 KB
principal.lst	C Output File	916 KB
principal.pjt	C Source File	1 KB
principal.sta	C Output File	8 KB
principal.sym	C Output File	21 KB
principal.tre	C Output File	216 KB
stdlibOK.bak	Archivo BAK	52 KB

Figura 4. 31 Generación del .hex en la carpeta del proyecto

4.2. PICKit2.

4.2.1. Tarjeta de Programación.

PICKit es una familia de programadores de microcontroladores PIC hechas por Microchip Technology (Wikipedia, Familia de programadores PICKit, introducción, 2004). Se utilizan para programar microcontroladores, así como algunas EEPROM. La vista general de la tarjeta de programación PICKit 2 se muestra a continuación.



Figura 4. 32 Vista de la tarjeta de programación conectada a la PC

4.2.2. Programación del Microcontrolador.

Una vez descargado e instalado este software (*versión 2.61.00*), se coloca el PIC en el zócalo de 40 pines tal como se muestra en la gráfica anterior, se conecta mediante el cable USB a la PC y se inicia la aplicación, aparece la ventana de trabajo en la cual se tiene el nombre del PIC que se ha insertado y botones como:

- Read: Leer el programa del PIC.
- Write: Escribir el .hex en el PIC
- Verify: Verifica el checksum del .hex.

- Erase: Borra el programa del PIC.

Una vez se haya borrado el programa del PIC en el caso que exista, se siguen los siguientes pasos:

1. En el menú de herramientas, damos clic en Import Hex.
2. Vamos al directorio donde está nuestro proyecto en PICC, seleccionamos el .hex.
3. Abrimos.
4. Damos clic en el botón Write para programar el .hex en el PIC.

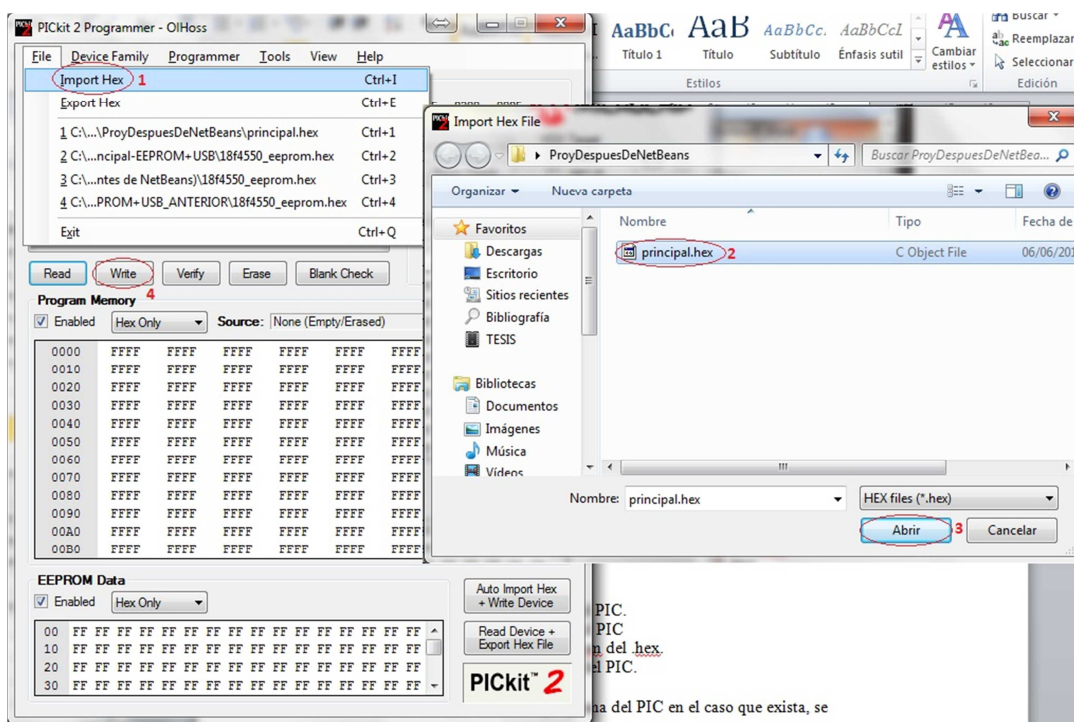


Figura 4. 33 Programación del PIC mediante el PICKit 2

Finalmente cuando se ha programado exitosamente el PIC, aparece el recuadro en verde que se muestra en la figura.

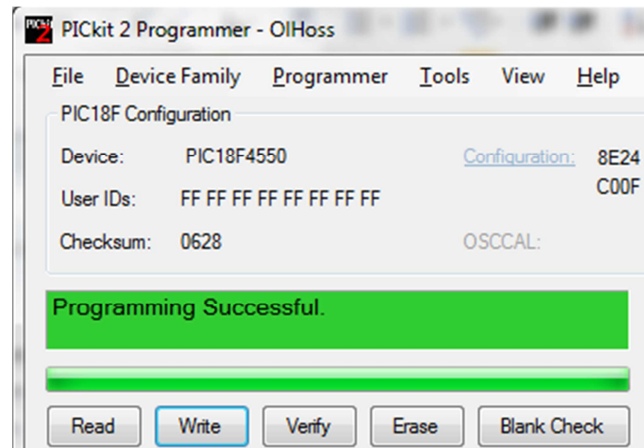


Figura 4. 34 Grabación exitosa del PIC.

4.3. Visual Basic 2010 Express.

Este software utilizado para el desarrollo de la interface de configuración, esencialmente se lo escogió debido a su no tan compleja interface de comunicación serial, así también por su sencillo manejo de gráficas en tiempo real, dos herramientas primordiales para el desarrollo de nuestra interface de configuración.

4.3.1. Módulo para comunicaciones IO.Ports.SerialPort.

La clase IO.Ports.SerialPort es utilizada para controlar un recurso de archivo de puerto serie. Esta clase proporciona acceso a los estados de punto de conexión e interrupción, acceso a las propiedades del controlador serie, además de E/S de datos sincrónica orientada a eventos, específicamente el evento *DataReceived*, es el que nos va a permitir obtener la respuesta del microcontrolador al enviarle un comando.

En la siguiente tabla se muestran los principales parámetros a configurar para el puerto serial. El código para abrir un puerto serial y configurarlo se muestra en [ANEXOS: 5. Visual Basic 2010, a). Instanciación y configuración del puerto serial].

Parámetro	Descripción
PortName	Nombre del puerto a abrir, como ejemplo COM4

BaudRate	Determina la velocidad de comunicación
Parity	Asignar paridad
DataBits	Longitud de bits de datos
StopBits	Bits de parada después de los bits de datos

Tabla 4. 18 Parámetros de configuración del puerto serial

Para el envío y para la lectura de datos desde el microcontrolador se usan las funciones mostradas en [ANEXOS: 5. Visual Basic 2010, b). Envío de comandos, c). Evento de lectura de datos]

4.3.2. Introducción a Windows Forms para creación de gráficas en tiempo real.

En la interface de configuración, quizá una de las herramientas más importantes es la gráfica en tiempo real mediante la cual se visualizan tanto las variables de gatillos, pedales, velocidad, así como la visualización del desempeño del controlador. A continuación se indica como insertar un chart y su configuración para realizar de líneas en tiempo real.

En la pestaña *Cuadro de Herramientas, Datos*, tenemos el ícono de selección de Chart como se indica en la siguiente gráfica.

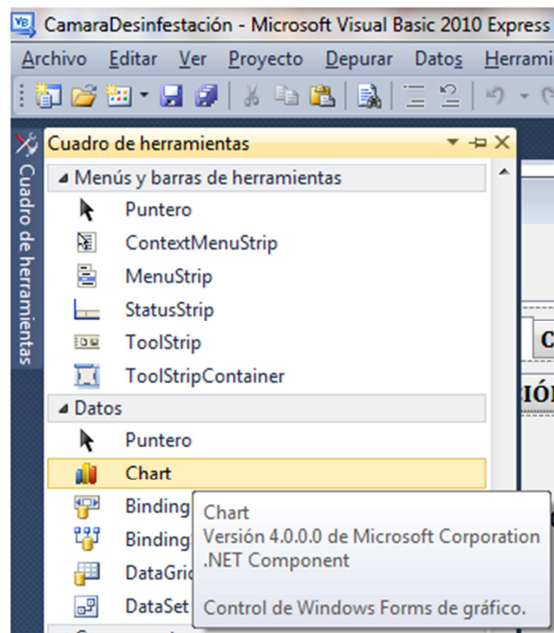


Figura 4. 35 Insertar un Chart en el formulario

Una vez insertada la gráfica, en las propiedades de esta se tiene varias opciones de configuración, entre las más importantes está la opción de **Titles**, en la cual se puede asignar el título de la gráfica. La opción **Series**, en la cual vamos prácticamente a insertar el número de variables a graficar, lo cual se indica en la siguiente figura.

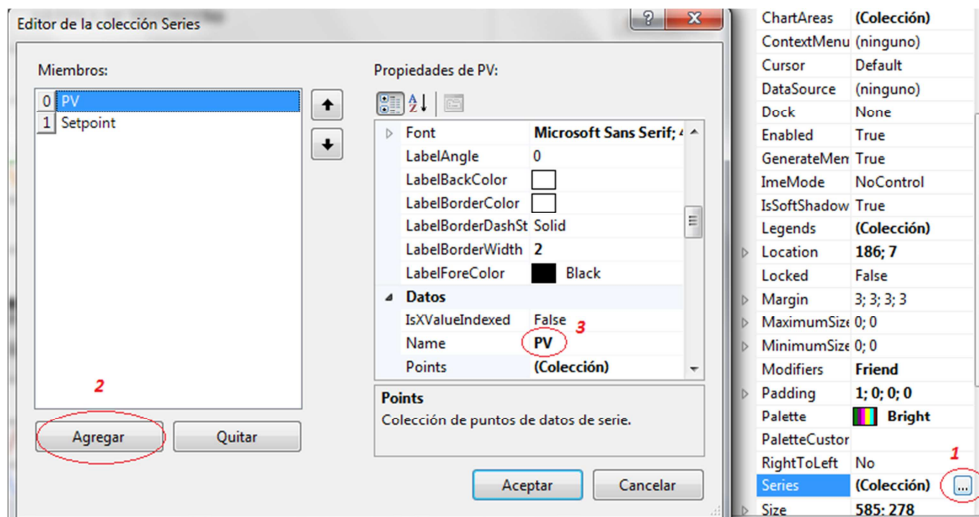


Figura 4. 36 Insertar y configurar series en la gráfica

A más de las series, se debe modificar el tipo de gráfico en el que seamos se muestren los datos, como se muestra en la siguiente figura, seleccionó **FastLine**, ya que es una gráfica de líneas que tiene una respuesta rápida.

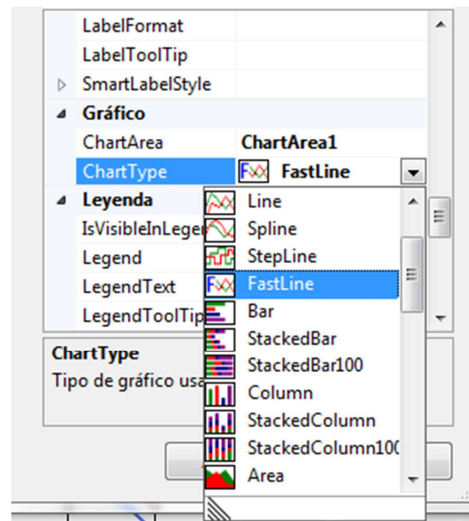


Figura 4. 37 Configuración de tipo de gráfica

Finalmente, para graficar los datos, se tiene como ejemplo el código para graficar los valores obtenidos de los gatillos, mostrado en [ANEXOS: 5. d). Graficar datos.]

4.3.3. Diseño de la Interface de Configuración de parámetros de Control.

La interface de configuración de parámetros de control es en sí una interface de ingeniería, que consta de una serie de pestañas organizadas según el tipo de variables a configurar o editar. El detalle se muestra a en la siguiente tabla.

Pestaña principal	Subpestaña 1	Subpestaña 2	Detalle
Asignar rangos de variables	Gatillos	x	Asignar rangos (universos de discurso) para los gatillos
	Pedales	x	Asignar rangos (universos de discurso) para los pedales
	Inclinación y velocidad	x	Asignar inclinación mín y máx, así como velocidad mín y máx del automóvil
Constantes de Controlador	Entradas	Error	Límites de las funciones de pertenencia del error

		Cambio del error	Límites de las funciones de pertenencia del cambio del error
Ganancias y salidas	x		Valores de las ganancias (Kp, Kd, Ku), valores de singletons de salida
Base de reglas		x	Especificar el comportamiento del sistema mediante la base de reglas
Desempeño	x	x	Visualizar el desempeño del control regulatorio difuso para cada pedal

Tabla 4. 19 Detalle de pestañas de la interface de configuración

Dentro de la pestaña de asignar rangos de variables, como se ve en la siguiente figura, podemos editar los rangos de gatillos, seleccionando el gatillo a configurar, y después haciendo click en el botón asignar mínimo o asignar máximo.

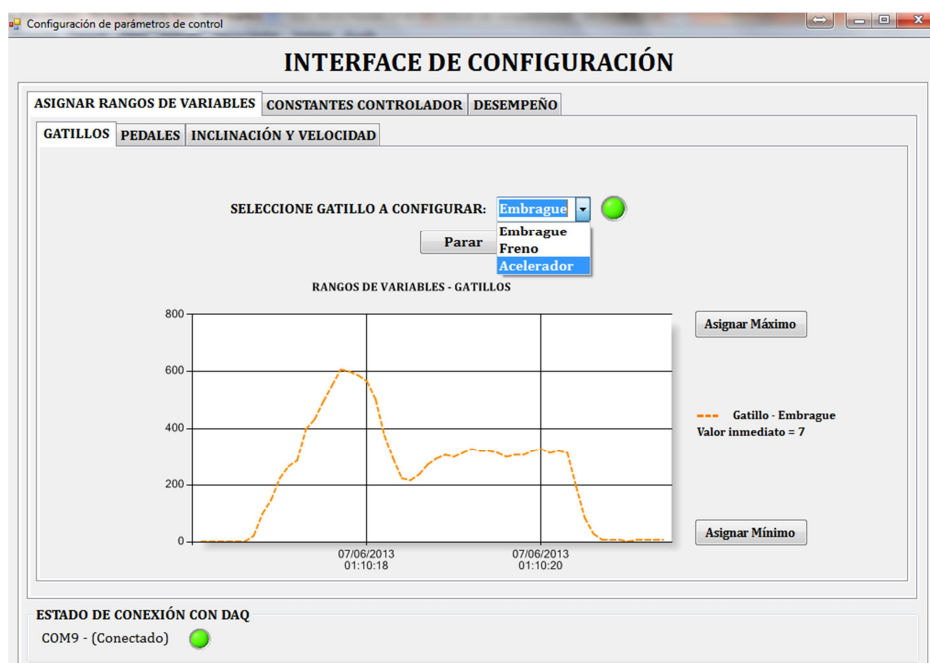


Figura 4. 38 Asignar rangos para los gatillos

Del mismo modo para asignar los rangos de los pedales, hacemos click en la respectiva pestaña y configuramos los rangos, además se tiene un label con el nombre del pedal a configurar y otro con el valor inmediato de la variable a editar, como se muestra en la figura 4.39.

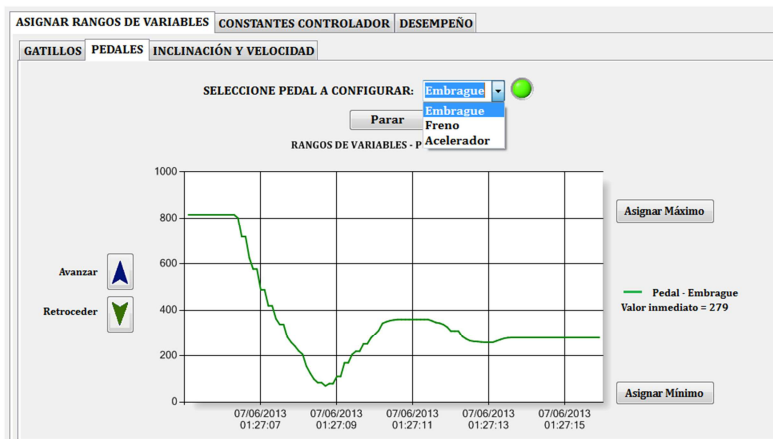


Figura 4. 39 Asignar rangos para los pedales

En la siguiente gráfica se muestra la pestaña para editar los rangos de inclinación y la velocidad mínima y máxima del automóvil, donde además de las variables antes mencionadas se tiene un botón llamado Asignar vOffset, el cual nos permite calibrar el voltaje de salida del sensor de inclinación cuando está paralelo a la horizontal.

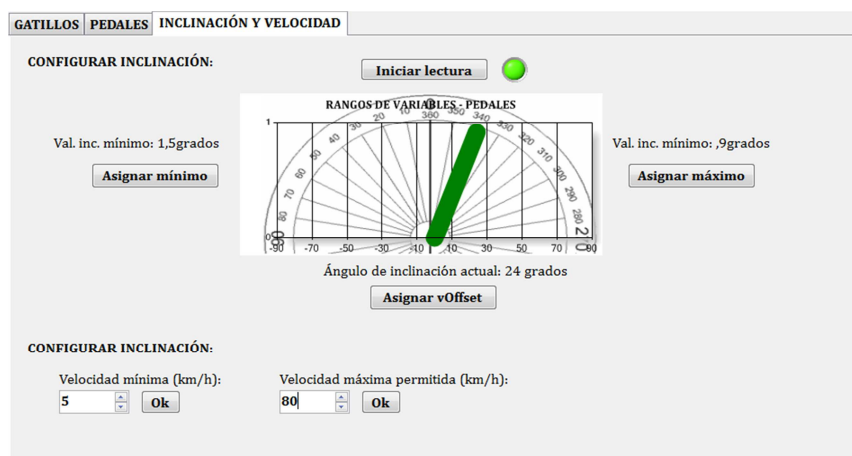


Figura 4. 40 Asignar rangos para inclinación y velocidad

En la subpestaña “Entradas” de la pestaña principal “Constantes del Controlador” podemos configurar los rangos de las funciones de pertenencia para el error y cambio del error de cada pedal como se muestra en las dos figuras a continuación.

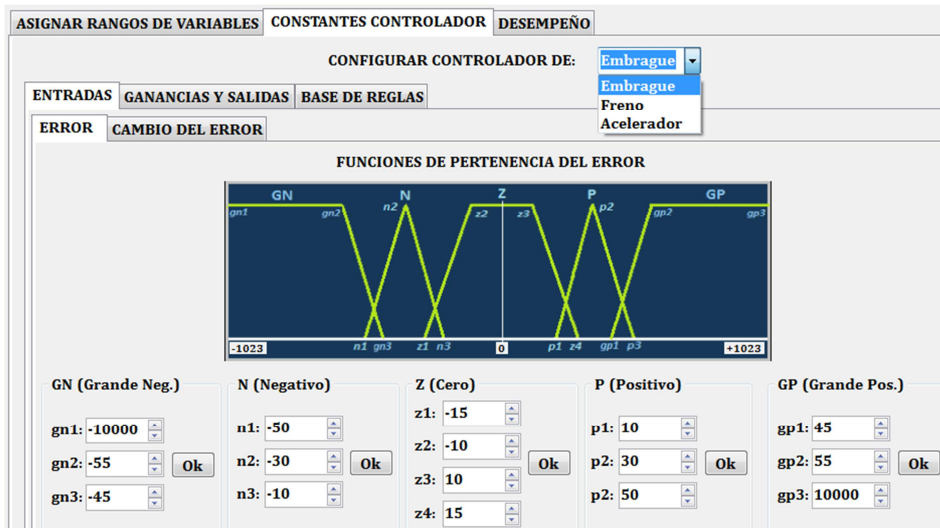


Figura 4. 41 Asignar rangos de funciones de pertenencia para el error

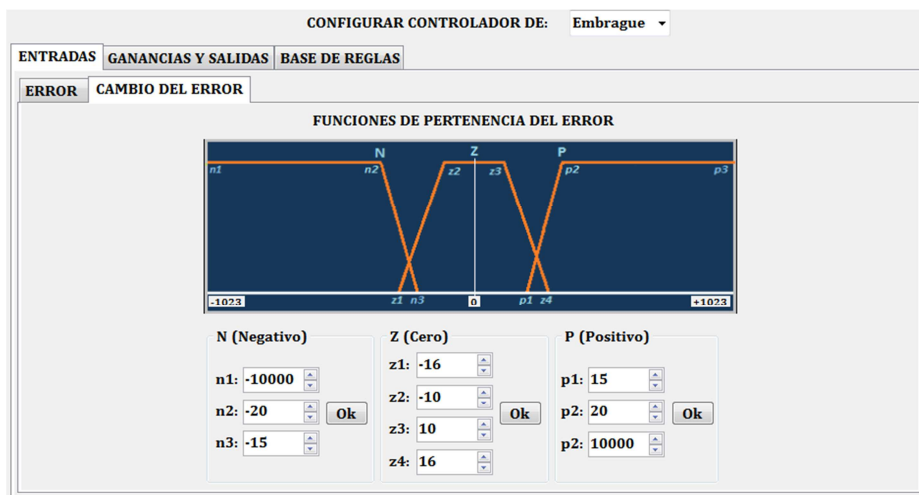


Figura 4. 42 Asignar rangos de funciones de pertenencia para el cambio del error

Luego pasamos a asignar las ganancias y los valores de los singletons de salida, como se muestra en la siguiente gráfica.

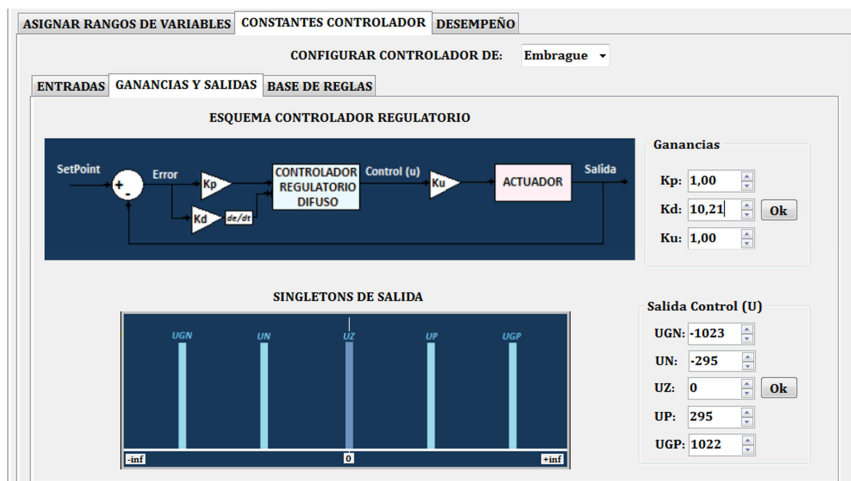


Figura 4. 43 Asignar ganancias y valores de singletons de salida

Para terminar con las constantes del controlador, se asignan los valores de singletons de salida para cada una de las combinaciones mostrada en la base de reglas, se selecciona del combo box el valor deseado como se indica a continuación.

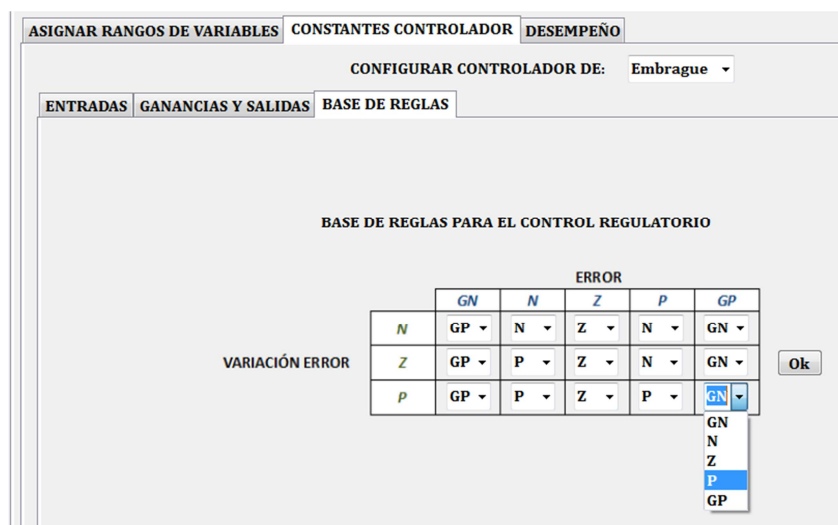


Figura 4. 44 Asignar base de reglas

Finalmente, y siendo tal vez la herramienta más importante de la interface de configuración, está la herramienta de *Desempeño* donde se puede visualizar el desempeño de los controladores regulatorios en los tres modos diferentes como se indica en la tabla a continuación.

Modo	Descripción
Por gatillo	Se visualiza el desempeño del controlador regulatorio del pedal seleccionado, teniendo como set point la medida del gatillo.
Por constante	Tiene como set point la constante ingresada en el campo de texto.
Señal cuadrada	Tiene como set point una señal cuadrada con periodo configurable

Tabla 4. 20 Modos de funcionamiento del Desempeño

En esta pestaña se tiene dos gráficas, la superior muestra la variable del set point de color naranja punteado, y la variable de salida en verde; mientras que en la gráfica inferior se tiene las variables del error en color marrón, la señal de control en color azul y la variable de la velocidad o cambio del error en color gris. Para iniciar el desempeño en primer lugar se selecciona un pedal, por defecto el modo de desempeño por gatillo está pre configurado, luego se da click en el botón iniciar como se muestra en la siguiente gráfica.

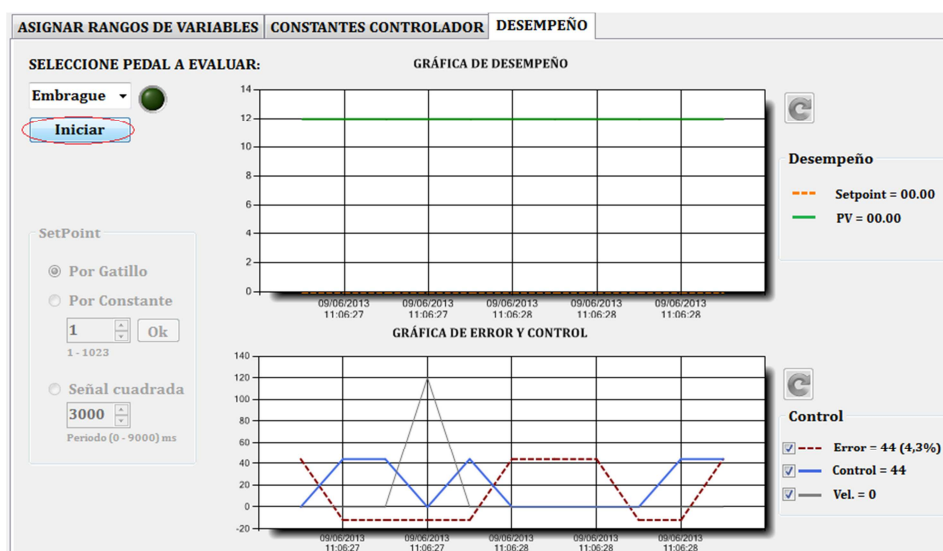


Figura 4. 45 Iniciar al desempeño

Para cambiar el modo de desempeño, simplemente se selecciona uno de los tres botones de radio, uno para cada modo, en el modo por gatillo simplemente se verá la respuesta del pedal respecto al set point enviado a través del gatillo, en el desempeño por constante se tiene que ingresar la constante del set point en el cuadro de texto como se indica a continuación.

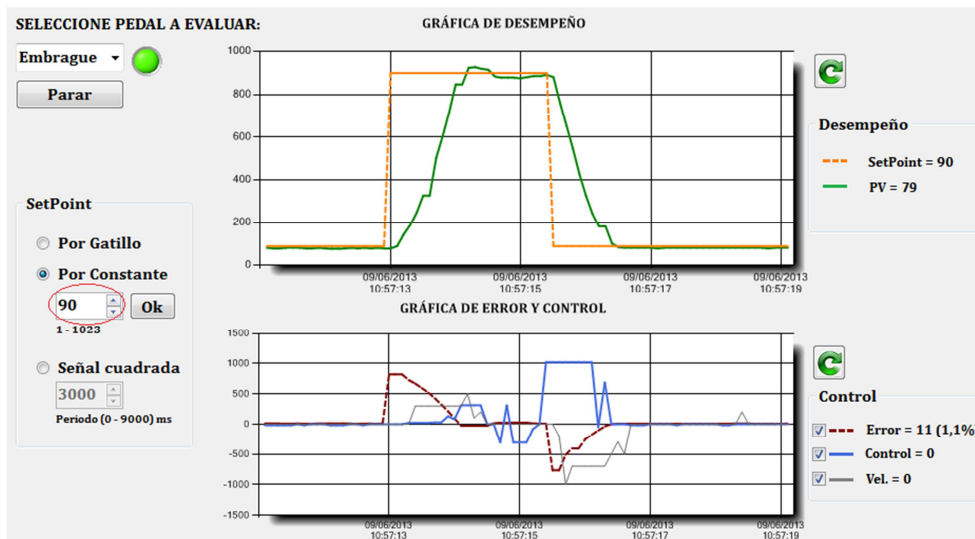


Figura 4. 46 Desempeño por constante

Para el modo de señal cuadrada, tras seleccionar el botón de radio correspondiente, se modifica el valor del periodo como se muestra en la figura 4.46.

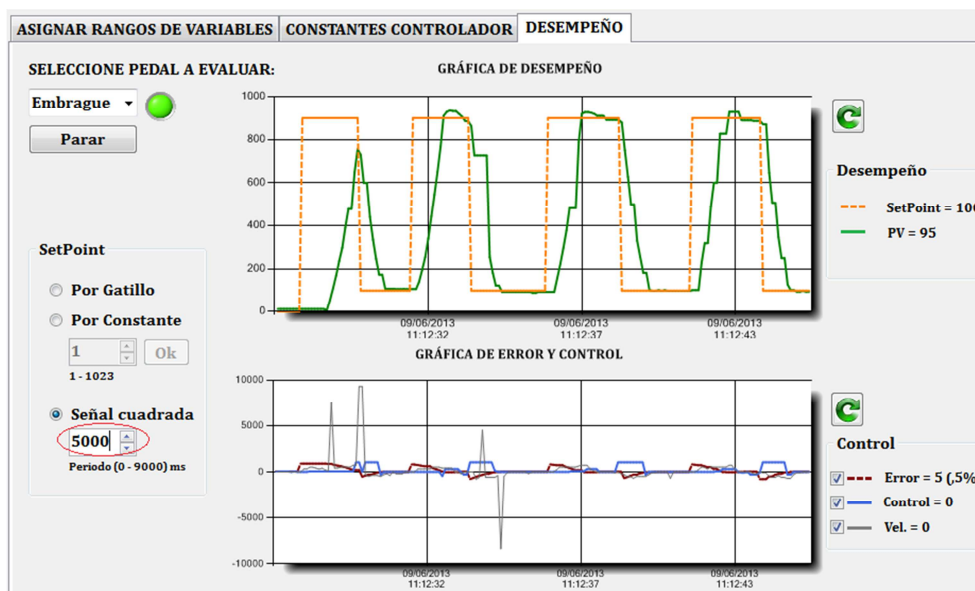


Figura 4. 47 Desempeño por señal cuadrada

CAPÍTULO 5

IMPLEMENTACIÓN DEL SISTEMA CONTROLADOR

5.1. Instalación.

La instalación del sistema controlador de posición de pedales se llevó a cabo en un automóvil convencional Chevrolet Corsa de tres marchas, para montar los actuadores lineales se utilizó la estructura metálica mostrada en la siguiente figura. La estructura se la montó sobre los rieles del asiento del conductor, gracias a que la base de la estructura es un tubo metálico rectangular, siendo de fácil montaje y desmontaje ya que no necesita ningún tipo de sujeción mecánica extra como tornillos



Figura 5. 1 Estructura montada sobre los rieles del asiento del conductor

Como se ve en la figura 5.2, es una estructura sencilla, hecha a medida para que los actuadores queden alineados justo en frente de los pedales, así mismo ubicados para que al estar retraídos estén en contacto con estos.



Figura 5. 2 Estructura metálica soporte para los actuadores

La unión entre los actuadores lineales y la estructura metálica se la hizo mediante pequeñas bisagras, que permiten la movilidad de los actuadores al momento de empujar los pedales tal como se ve en la siguiente figura.



Figura 5. 3 Unión de estructura metálica con actuadores lineales

Del mismo modo, la unión entre los pedales y su respectivo actuador lineal es únicamente con fricción, mediante una platina de hierro en forma de U, forrada en la parte interna con papel lija, como se ve en la figura 5.4, evitando así cualquier tipo de deslizamiento innecesario de los actuadores sobre los pedales.



Figura 5. 4 Unión entre actuadores y pedales

Respecto a los sensores, para la medición de posición de los pedales, se utilizó los ya mencionados potenciómetros lineales deslizantes, montados como se indica en la siguiente figura.

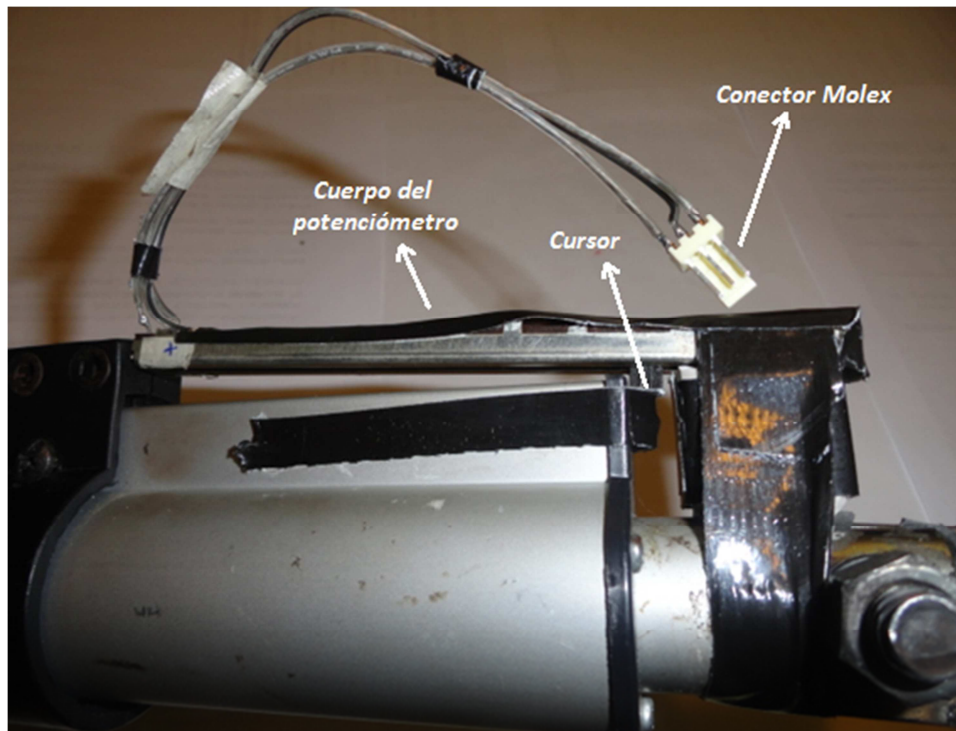


Figura 5. 5 Montaje de sensores sobre actuadores lineales

Donde, sujeto al vástago del actuador lineal se tiene el un extremo del sensor, y el cursor se lo sujeta a la estructura fija del actuador, permitiendo así que mientras se extienda o se retraiga el vástago de este, el cursor se quede fijo mientras el cuerpo del sensor se mueve.

Para conectar este sensor a la tarjeta de adquisición, se utilizó conectores molex de tres pines macho y hembra, de modo que si se requiere desconectar los sensores, solo basta con separar los conectores evitando desconectar los cables de las borneras de la tarjeta de adquisición.

Para facilidad y comodidad del conductor, este puede colocar sus piernas en el espacio disponible entre el asiento y la estructura metálica, además como el cableado está colocado por debajo del tubo derecho de la estructura, no se tiene contacto alguno con estos, lo anterior se muestra en la figura 5.6.



Figura 5. 6 Posición de los pies del conductor entre la estructura

Para los gatillos, se colocaron las galgas extensiométricas en un pequeño compartimento a manera de bolsillo sobre los dedos de dos guantes comunes, cada guante consta de dos compartimentos, uno sobre el dedo índice y el otro sobre el dedo medio, de manera que el conductor pueda ubicar fácilmente las galgas según vea conveniente, en este como se muestra en la figura 5.7 se colocó la galga del freno sobre el dedo índice del guante izquierdo.



Figura 5. 7 Galga del freno en dedo índice de guante izquierdo

Mientras que en el caso de utilizar el sistema en modo manual, la galga del acelerador y del embrague se colocaron sobre los dedos índice y medio del guante de la mano derecha respectivamente, como se muestra en la figura 5.8. Sin embargo para el modo semiautomático, solo se colocó la galga del acelerador sobre el dedo índice de este guante.



Figura 5. 8 Galgas del acelerador y embrague en el guante derecho

El pulsador indicador de cambio de marcha, se colocó en la palanca de cambios, de modo que se pueda pulsar con el dedo pulgar, como se indica en la siguiente figura.

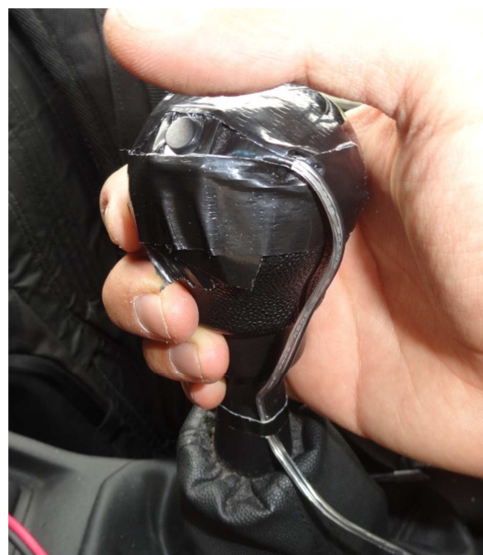


Figura 5. 9 Pulsador indicador de cambio de marcha

Finalmente, para optimizar el espacio útil en el automóvil, el case de la tarjeta de adquisición y control se lo ubicó bajo el asiento del conductor, el case con sus respectivos conectores se muestra en la figura 5.10.

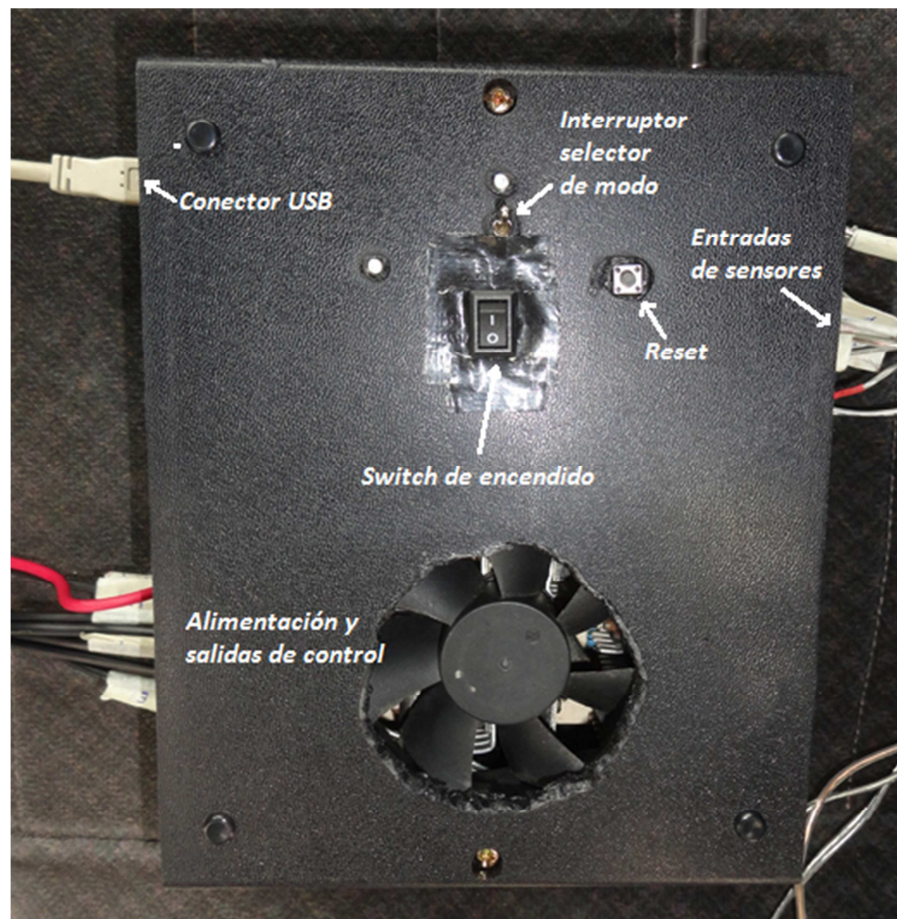


Figura 5. 10 Case de la tarjeta de adquisición y control

5.2. Ajustes y Calibraciones.

Una vez instalado el sistema en el automóvil, por medio de a la interface de configuración, se realizó la calibración de los universos de discurso tanto para los gatillos como para los actuadores lineales. Para los gatillos, por facilidad de manejo y manipulación se tomó el rango mínimo de lectura de los gatillos cuando el dedo está próximo a extenderse mas no cuando este está completamente extendido, esto debido a que es incómodo mantener los dedos completamente extendidos durante periodos relativamente largos (2 horas), así, para los universos de discurso de los gatillos se configuraron los rangos mostrados en la siguiente tabla.

Gatillo	Rango	Variable	Dirección EEPROM	Valor digital	Valor analógico
Embrague	mínimo	emb_g_min	0	601	2,94
	máximo	emb_g_max	2	753	3,68
Freno	mínimo	fre_g_min	4	586	2,86
	máximo	fre_g_max	6	677	3,31
Acelerador	mínimo	ace_g_min	8	664	3,25
	máximo	ace_g_max	10	812	3,97

Tabla 5. 1 Límites universo de discurso para los gatillos

Para los pedales, la configuración de universos de discurso fue más exacta, debido a que cada pedal es diferente entre sí, esto varía según el automóvil, pero en este caso en particular el pedal del embrague requirió casi la totalidad de la carrera del actuador lineal de 400 libras, en cambio para el freno, se requirió aproximadamente el 36% de la carrera del actuador lineal correspondiente, y finalmente el acelerador es el que tiene los límites más cortos, necesitando solo el 23% de la carrera del actuador. Los rangos de los universos de discurso para los pedales se muestran en la tabla 5.2.

Pedal	Rango	Variable	Dirección EEPROM	Valor digital	Valor analógico
Embrague	mínimo	emb_p_min	12	249	1,22
	máximo	emb_p_max	14	1018	4,98
Freno	mínimo	fre_p_min	16	384	1,88
	máximo	fre_p_max	18	755	3,69
Acelerador	mínimo	ace_p_min	20	240	1,17
	máximo	ace_p_max	22	470	2,30

Tabla 5. 2 Límites universo de discurso para los pedales

Para los límites de las variables de inclinación y de velocidad del automóvil se configuró según los datos de la siguiente tabla.

Variable	Rango	Variable	Dirección EEPROM	Valor
Inclinación	mínimo	inc_min	26	1
	máximo	inc_max	28	13
Velocidad	mínimo	vel_min	336	12
	máximo	vel_max	338	55

Tabla 5. 3 Rangos para la inclinación y velocidad

Como parámetros de configuración para los tres controladores regulatorios difusos, se han seteado los rangos de las funciones de pertenencia de las entradas para cada uno de los pedales, los rangos se muestran en la siguiente tabla.

Variable de entrada	Función de pertenencia	Constante	Valor Embrague	Valor Freno	Valor Acelerador
Error	Grande Negativo	gn1	-10000	-10000	-10000
		gn2	-55	-55	-65
		gn3	-45	-45	-45
	Negativo	n1	-50	-50	-50
		n2	-30	-40	-40
		n3	-10	-33	-30
	Cero	z1	-15	-35	-35
		z2	-13	-30	-20
		z3	13	30	20
		z4	15	35	35
	Positivo	p1	10	33	30
		p2	30	40	40
		p3	50	50	50
	Grande Positivo	gp1	45	44	45
		gp2	55	65	65
gp3		10000	10000	10000	
Cambio del Error	Negativo	n1	-10000	-10000	-10000
		n2	-50	-50	-25
		n3	-25	-25	-18
	Cero	z1	-30	-30	-30
		z2	-10	-10	-15
		z3	10	10	15
		z4	30	30	20
	Positivo	p1	25	25	18
		p2	50	50	25
		p3	10000	10000	10000

Tabla 5. 4 Rangos de funciones de pertenencia de las entradas para los tres pedales

Para la configuración de las ganancias y especialmente para los valores de los singletons de salida, se tomó en cuenta la fuerza que requiere cada pedal para moverse, dado que la fuerza efectiva de los actuadores solo se la aplica al momento de empujar los pedales, en el caso del embrague, este actuador es el que mayor fuerza necesita al momento de empujar dicho pedal, especialmente cuando ya está llegando al fondo de su carrera, por lo que se seteó el valor de UP con 400, que es el valor óptimo que el controlador regulatorio aplica

para romper la inercia, este es el mayor valor respecto a los otros pedales para ese singleton de salida. Para el freno, del mismo modo se requiere el valor de 290 para UP de modo que pueda romper la inercia en toda la carrera del pedal. Se puede ver también que el pedal que menor fuerza requiere es el del acelerador. Las ganancias y los valores de los singletons de salida se muestran en la siguiente tabla.

	Constante	Valor Embrague	Valor Freno	Valor Acelerador
Ganancias	Kp	1,0	1,0	1,0
	Kd	10,0	10,0	10,0
	Ku	1,0	1,0	1,0
Funciones de salida	UGN	-1023	-1023	-1023
	UN	-285	-270	-260
	UZ	0	0	0
Singletons	UP	400	290	265
	UGP	1023	1023	1023

Tabla 5. 5 Ganancias y Singletons de Salida

Finalmente la base de reglas, se configuró de modo que cuando el error sea grande negativo o grande positivo, sin importar el valor del cambio del error, se tenga la máxima salida de control obteniendo la máxima velocidad de respuesta de los actuadores optimizando el tiempo de establecimiento.

También se lo configuró para realizar la acción de frenado, en el caso cuando el error se aproxime a cero desde N y el cambio del error tenga valores negativos, y también el caso cuando el error se aproxime a cero desde P y el cambio del error tenga valores positivos, esta acción de frenado es esencial al momento de reducir los sobrepicos. La base de reglas configurada para los tres pedales es mostrada en la tabla 5.6.

		Error				
		<i>GN</i>	<i>N</i>	<i>Z</i>	<i>P</i>	<i>GP</i>
Cambio del error	<i>N</i>	GP	N	Z	N	PN
	<i>Z</i>	GP	P	Z	N	GN
	<i>P</i>	PP	P	Z	P	GN

Tabla 5. 6 Base de reglas para el control sobre el embrague

Con estos parámetros, se procede a observar el desempeño del sistema de control regulatorio.

CAPÍTULO 6

PRUEBAS Y RESULTADOS

6.1. Pruebas de Funcionamiento.

En primer lugar se comprueba el funcionamiento y desempeño del control regulatorio, tomando el embrague para realizar las pruebas ya que este tiene el mayor recorrido de los tres pedales.

6.1.1. Sistema de control regulatorio.

6.1.1.1. Desempeño por gatillo.

A continuación se muestra el desempeño del sistema controlador regulatorio mediante set point por gatillo.

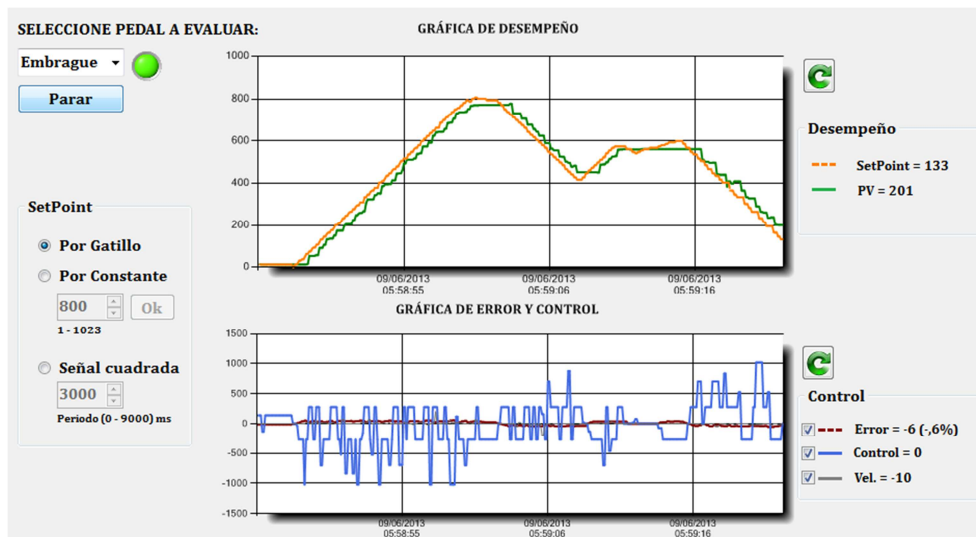


Figura 6. 1 Desempeño por gatillo

En la figura 6.1, en la gráfica superior titulada “Gráfica de Desempeño”, se puede ver la variación del set point mediante el gatillo en color naranja entrecortado, mientras que la señal de respuesta de posición, se muestra en la gráfica verde.

6.1.1.2. Desempeño por constante.

Del mismo modo procedemos al desempeño por constante, la respuesta del sistema tomando a algunos set points consecutivos ingresados por teclado es la siguiente.

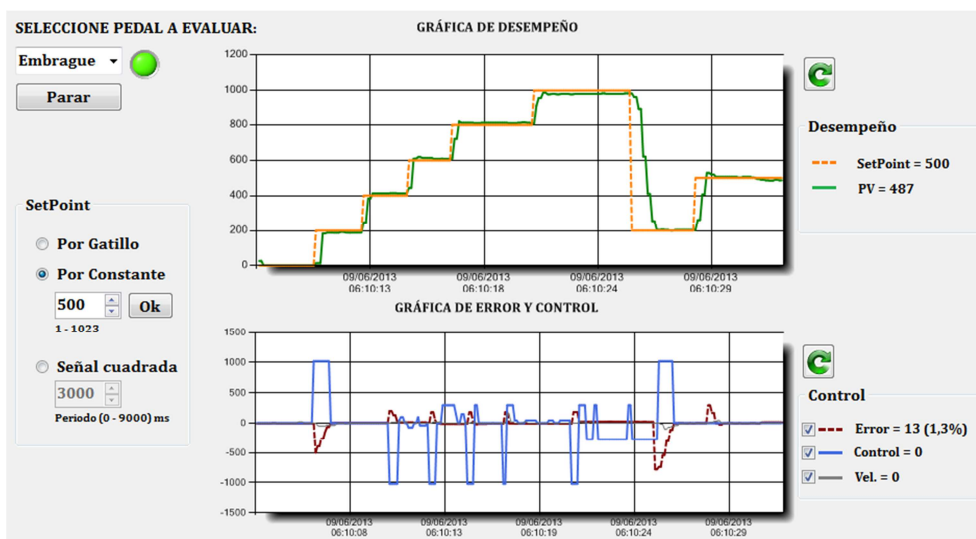


Figura 6. 2 Respuesta a varios set points consecutivos

6.2. Análisis de Resultados.

Tomando en cuenta el tiempo de establecimiento, el máximo pico y el error en estado estacionario, la respuesta del controlador regulatorio se muestra en la siguiente figura.

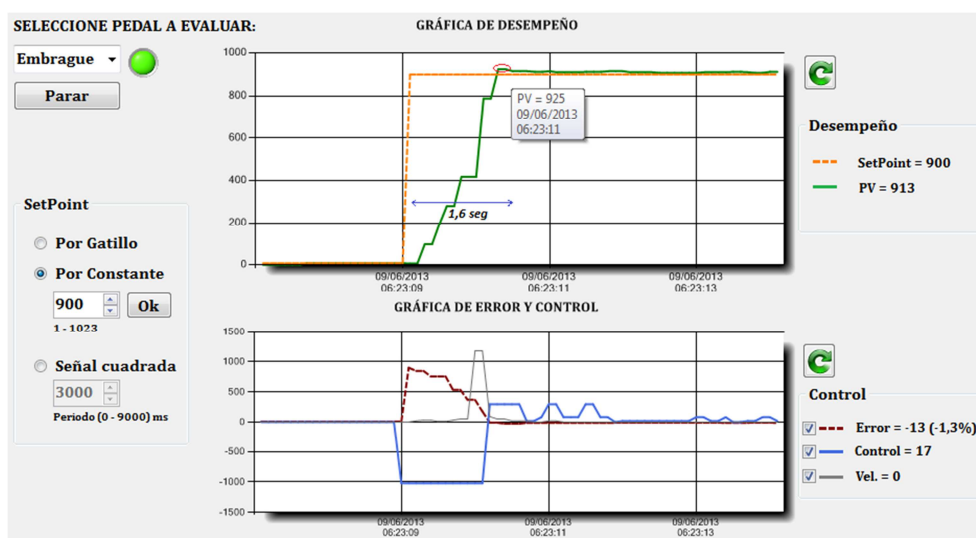


Figura 6. 3 Gráfica de respuesta detallada

De la gráfica anterior, se obtienen los siguientes resultados.

Parámetro	Valor
Tiempo de establecimiento	1,6 seg
Máximo pico	25 (2,44 %)
Error en estado estacionario	13 (1,27 %)

Tabla 6. 1 Resumen de parámetros de respuesta del sistema

En definitiva el funcionamiento del sistema controlador regulatorio difuso es óptimo, respondiendo muy bien a las pruebas de set point por gatillo donde el seguimiento del set point es muy bueno obteniendo un error mínimo.

Así mismo la respuesta a los set points constantes ingresados por teclado muestra una respuesta muy buena, según la tabla 6.1 el tiempo de establecimiento varía entre 1.6 a 1.8 segundos, lo cual es relativamente rápido, tomando en cuenta que la fuerza necesaria para empujar este pedal es alta (mayor a 45 lbs), así mismo el tiempo de establecimiento en los pedales restantes es tan solo del 30% del tiempo para el embrague, esto gracias a que sus límites de carrera son más cortos como se mostró en la tabla 5.2, además que la fuerza necesaria para empujar estos pedales es prácticamente despreciable respecto a la capacidad de empuje de los actuadores.

Analizando el máximo pico y el error en estado estacionario, se puede concluir que el controlador regulatorio difuso, con los ajustes y calibraciones adecuadas dependiendo del tipo de automóvil, la fuerza y los límites de carrera de los pedales, es óptimo en todo sentido. Con estos resultados se puede afirmar que el desarrollo e implementación de un controlador regulatorio difuso sobre un sistema microcontrolado se obtuvo con muy buenos resultados.

Cabe aclarar que la acción de control proporcional de velocidad de salida del embrague en función de la inclinación del automóvil fue obviada, esto debido a que se obtiene un mejor control de manejo con la velocidad de los actuadores al máximo.

CAPÍTULO 7

CONCLUSIONES Y RECOMENDACIONES

7.1. Conclusiones.

- La lógica difusa es una estrategia de diseño de controladores que nos brinda grandes prestaciones. En este trabajo se ha estudiado una metodología de diseño de controladores inteligentes a base de microcontrolador, utilizando algoritmos de lógica difusa, con sistemas tipo Takagi-Sugeno, los cuales nos han permitido literalmente trabajar con un sistema mecánico de posición sin siquiera saber su modelo matemático. Es decir, se facilita el diseño ya que no se requiere el típico proceso de creación de un controlador clásico como por ejemplo el PID, en el cual se requiere un modelo matemático base, para posteriormente aplicar los conceptos de diseño y finalmente sintonizar las ganancias para obtener los resultados de control deseados.
- Generalmente los algoritmos de lógica difusa son algoritmos computacionales, desarrollados en software como Matlab, resulta obvia la diferencia de capacidad de procesamiento entre un computador y un microcontrolador, sin embargo, en este trabajo se implementó el control difuso en un microcontrolador de la familia de los PIC-18F4X5X, con lo que la utilización de un sistema difuso tipo Takagi-Sugeno nos ha permitido trabajar sin problemas, debido a que es una representación más compacta y eficiente que un sistema de tipo Mamdani, computacionalmente hablando, ya que al no tener un defuzzificador se obtiene un bajo coste computacional al momento de calcular la salida. La principal ventaja de este modelo difuso es su poderosa capacidad de representar relaciones complejas y no lineales con un pequeño número de reglas difusas.

-
- El uso de un sistema difuso PD nos ha permitido obtener un control más fino sobre los pedales, reduciendo el sobre pico y tiempo de establecimiento. La reducción del sobre pico se debe a que la acción derivativa permite implementar una acción de frenado con lo que se logra una inversión de polarización en la alimentación de los actuadores para errores pequeños y cambios de error grandes, es decir al momento justo que el error se aproxima a cero. El tiempo de establecimiento también se ve mejorado ya que al contar con la acción de frenado, se puede enviar la señal de control más grande a los actuadores prácticamente sobre todo el trayecto para errores grandes y medianos, obteniendo la máxima velocidad.

 - Se ha diseñado una interface HMI de configuración de parámetros de control en Visual Basic 2010 Express, siendo está de gran ayuda al momento de calibrar el sistema sobre la marcha, además, es de fácil uso ya que es amigable al usuario, ya que mediante la comunicación USB_CDC con la tarjeta de control, permite configurar todos los parámetros del controlador para cada uno de los pedales como son los rangos de variables, límites de funciones de pertenencia para las entradas, ganancias y base de reglas. Además con la herramienta de visualización de desempeño, se puede probar en cualquier momento la performance de los tres controladores y según esto reconfigurar los parámetros para una respuesta óptima.

 - Se ha realizado el montaje y puesta en marcha del sistema en un automóvil convencional Chevrolet Corsa Evolution de tres marchas, mediante una estructura metálica de fácil montaje y desmontaje, además sin ocupar demasiado espacio como para incomodar al conductor, permitiendo ubicar sus piernas en posición normal.

 - En cuanto a las pruebas de funcionamiento, en general se obtuvo muy buenos resultados en cuanto a seguridad, performance de respuesta de actuadores y facilidad de manejo, con lo que el usuario final mediante la práctica y manipulación del sistema puede dominar el manejo del automóvil mediante este en un rango de dos a tres días.

7.2. Recomendaciones

7.2.1. Programación del PIC

➤ Al general el .hex en el software PICC para programar el PIC, si se está trabajando sobre el sistema operativo Windows 7 se recomienda usar la versión CCS 4.104 de este software, la cual es más estable que algunas versiones pasadas e incluso que algunas versiones más actuales, es más estable en el sentido de que al trabajar con muchas de las prestaciones del PIC en el mismo programa como son las comunicaciones seriales, USB_CDC, interrupciones, timers, etc, al generar el .hex no existen bugs que afectan la performance de estas prestaciones, es decir, todo lo que se programa corre sin dificultades de funcionamiento. Además de esto es siempre recomendable que se utilice la herramienta de “Clean” del PICC, para que al compilar de nuevo, no se sobrescriban los datos sobre el .hex anterior, sino generar un .hex nuevo.

7.2.2. Montaje del sistema de control

➤ Se recomienda que al instalar el sistema de control específicamente los actuadores, se lo haga en una estructura metálica, esto con el objetivo de fijarlos eliminando el juego o movimientos innecesarios, además esta estructura debe ser fácil de montar y desmontar, y que deje el suficiente espacio libre en la parte frontal del asiento del conductor de modo que se puedan ubicar las piernas sin tener contacto con el cableado ni los actuadores.

7.2.3. Calibraciones y puesta en marcha

➤ Al momento de configurar los parámetros de control del sistema de posicionamiento de pedales mediante al interface, se recomienda empezar por los rangos de las variables, y cuando se esté configurando los rangos de los pedales es importante probarlos con el automóvil apagado para el embrague y el acelerador, y con el automóvil encendido al configurar el freno, esto es con el objetivo de no generar una sobrepresión en las líneas del líquido de frenos mientras el automóvil está apagado.

-
- La velocidad máxima permitida por defecto es de 80 Km/h, sin embargo, cuando se configure la velocidad máxima del automóvil, es recomendable no exceder los 65 Km/h, por que por razones obvias de seguridad, es mejor trabajar a no tan altas velocidades.

 - Una vez configurados los parámetros de control óptimos para cada pedal, se recomienda usar el sistema en el modo semiautomático, ya al usar solo dos gatillos, uno para el control independiente sobre el freno y el otro para controlar al embrague y al acelerador, lo hace más amigable al conductor el cual tendrá una experiencia de manejo sencilla y divertida a la vez.

 - En ciudades como Quito donde hay muchas cuestas, algunas de ellas bastante pronunciadas, al momento de arrancar, se recomienda hacerlo usando el gatillo de aceleración en conjunto con el del freno, de modo que emule la salida con el freno de mano.

GLOSARIO.

ADC: Conversor Análogo Digital, transforma una señal analógica en un valor digital.

Actuador Lineal: Dispositivo electro-mecánico que tiene como principal función, el de desplazar carga a lo largo de su eje de carrera, en función de la polarización de la alimentación este puede trabajar en dos sentidos.

Banda Proporcional: En un controlador, es el cambio del error que se necesita para que el actuador realice el 100% de la carrera o esté activado en su totalidad.

BaudRate: La tasa de baudios, también conocida como baudaje, es el número de unidades de señal por segundo. Un baudio puede contener varios bits.

CPU: Unidad Central de Proceso.

FT: Función de Transferencia, es un modelo matemático que a través de un cociente relaciona la respuesta de un sistema a una señal de entrada.

EEPROM: Memoria de solo lectura, programable y borrable eléctricamente.

Embebido: Un sistema embebido o empotrado es un sistema de procesamiento diseñado para realizar una o algunas pocas funciones dedicadas, en un sistema de computación en tiempo real.

Error en estado estacionario: Es una medida de la exactitud de un sistema de control para seguir una entrada dada, después de desaparecer la respuesta transitoria (Lara, 2009).

Interrupción: Es una señal recibida por el procesador, indicando que se debe interrumpir el curso de ejecución del programa en curso, y pasar a ejecutar un código específico.

GPS: Sistema de Posicionamiento Global. Módulo que recibe las señales de diferentes satélites, calculando así mediante triangulación su posición en coordenadas latitud y longitud.

NMEA: (National Marine Electronics Association), interface de comunicación entre dispositivos electrónicos, la cual permite enviar información a computadoras o a otros equipos marítimos.

PIC: Controlador de Interface Periférico.

MIMO: Sistema de control cuya característica es tener múltiples entradas y múltiples salidas.

Planta: Se llama planta a cualquier sistema físico que se desea controlar.

PWM: Modulación de ancho de pulso.

Setpoint: El valor de consigna o el valor deseado que un sistema automático debe alcanzar al actuar sobre la planta (Wikipedia, Sistemas de control, conceptos básicos, 2007).

SCL: Señal de reloj en comunicación I2C.

SDA: Señal de Datos en comunicación I2C.

Servosistema: O también llamado servomecanismo a un sistema de control retroalimentado en el que la salida es algún elemento mecánico, sea posición, velocidad o aceleración.

Sistema: Es una combinación de componentes que actúan conjuntamente y cumplen determinado objetivo.

TAC: Tarjeta de Adquisición y Control.

Variable Controlada: La variable controlada es la cantidad o condición que se mide y controla.

Variable Manipulada: La variable manipulada es la cantidad o condición de materia o energía que el controlador modifica para afectar el valor de la variable controlada.

VC: Variable Controlada de un Sistema.

Velocidad de ajuste: Es la cantidad de veces por minuto que se duplica la parte proporcional de la acción de control.

Volátil: Referente a una memoria que es de carácter cambiante o inconstante.

USB: Bus Serial Universal. es un estándar industrial desarrollado en los años 1990 que define los cables, conectores y protocolos usados en un bus para conectar, comunicar y proveer de alimentación eléctrica entre ordenadores y periféricos y dispositivos electrónicos (Wikipedia, Interface USB, conceptos, 2005).

USB_CDC: Interface de comunicación donde al conectar un dispositivo (tarjeta de adquisición) a un host (PC) mediante un cable USB, se emula el dispositivo como un puerto COM o puerto serial.

BIBLIOGRAFÍA.

- (IEC), I. E. (1997). Fuzzy Control Programming. *IEC 1131 Programmable Controllers*.
- Antini Manuel, D. B. (2001). Tipos de Controladores. In D. B. Antini Manuel, *Instrumentación Virtual* (p. 87).
- CSS. (2008). *Detalles compilador CSS para programación de microcontroladores*. Retrieved from Detalles compilador CSS para programación de microcontroladores: <http://www.ccsinfo.com/content.php?page=compiler-details>
- Gpsinformation. (2010). *Información de la sentencia estándar MNEA*. Retrieved from <http://www.gpsinformation.org/dale/nmea.htm>
- instrumentacionycontrol. (2009). *Ventajas y desventajas controladores PID*. Retrieved from <http://www.instrumentacionycontrol.net/es/curso-control-pid-practico/364-el-control-proporcional-definiciones-practicas-y-precisas>
- KatsuhikoOgatta. (1998). Controles PID. In KatsuhikoOgatta, *Ingeniería de Control Moderna* (p. 669).
- Kevin M. Passino, S. Y. (1998). Basics. In S. Y. Kevin M. Passino, *Fuzzy Control* (p. 10).
- Kuo, B. C. (1996). Modelamiento matemático de Sistemas Físicos. In B. C. Kuo, *Sistemas de Control Automático* (p. 134).
- L.A, Z. (1965). Information and Control. *Fuzzy Sets*, 8, 3.
- Lara, M. C. (2009). Error en estado Estacionario.
- Lotfi, Z. (1973). IEEE Transactions on Susters, Man, and Cybernetics. *SMC-3*, 341.
- Mamdani, E. H. (1974). Basic Aplications. *Applications of Fuzzy Algorithm for Control of Simple Dynamic Plant*, 121, 1587.
- Mamdani, E. H. (1975). Intl. J. Man machine study. *A Fuzzy Logic Controller for a Dynamic Plant*, 2.
- Microchip. (2006). 10-bit analog-to-digital Converter (a/d) module. *PIC18F2455/2550/4455/4550 Data Sheet*, 259.
- Microchip. (2006). I/O Ports. *PIC18F2455/2550/4455/4550 Data Sheet*, 111.
- Microchip. (2006). UART. *PIC18F2455/2550/4455/4550 Data Sheet*, 237.
- Moreno, L. (2003). Técnicas de Control. In L. Moreno, *Ingeniería de Control – Modelado y Control de Sistemas Dinámicos* (p. 279).
- Morillo, D. (2011). Control Integrativo. *Contol de Procesos U2-2*.
- Morillo, D. (2011). Técnicas de Control de Procesos. *Contol de Procesos 2*.
- Ogatta, K. (1998). La transformada de Laplace. In K. Ogatta, *Ingeniería de Control Moderna* (p. 13).
- Passino, K. M. (1998). Fuzzy Control: The Basics. In K. M. Passino, *Fuzzy Control* (p. 55).
- Passino, K. M. (1998). Fuzzy Control: The Basics. In K. M. Passino, *Fuzzy Control* (p. 55).
- Profesormolina. (2009). *Transistores BJT, conceptos*. Retrieved from Transistores BJT, conceptos: http://www.profesormolina.com.ar/tutoriales/trans_bipolar.htm
- Sánchez, J. A. (2003). Métodos o Formas de Control. In *Control Avanzado de Procesos* (p. 182).
- scribd. (2010). *Arquitectura Harvard en Microcontroladores*. Retrieved from <http://es.scribd.com/doc/57978444/PIC18F4550>

-
- scribd. (2010). *Sistema de interrupciones en el PIC18F4550*. Retrieved from <http://es.scribd.com/doc/44197510/7/A-3-Sistema-de-interrupciones-de-PIC18F4550>
- Simbol, S. (2008). *Flex Sensor FS Data Sheet*, 2.
- Smith., C. A. (1991). Diseño clásico de un sistema de Control. In C. A. Smith., *Control Automático de Procesos* (p. 65).
- Spartacus Gomáriz, D. B. (1998). Diseño de sistemas de control de tiempo continuo y discreto. In D. B. Spartacus Gomáriz, *Teoría de Control Diseño Electrónico* (p. 132).
- Spartacus Gomáriz, D. B. (1998). Diseño de sistemas de control de tiempo continuo y discreto. In D. B. Spartacus Gomáriz, *Teoría de Control Diseño Electrónico* (p. 143).
- U-Blox. (2011). Functional Description. *LEA-6 series Data Sheet*, 6.
- Wikipedia. (2004). *Familia de programadores PICKit, introducción*. Retrieved from Familia de programadores PICKit, introducción: <http://en.wikipedia.org/wiki/PICKit>
- Wikipedia. (2005). *Interface USB, conceptos*. Retrieved from Interface USB, conceptos: http://es.wikipedia.org/wiki/Universal_Serial_Bus
- Wikipedia. (2007). *Sistemas de control, conceptos básicos*. Retrieved from [http://en.wikipedia.org/wiki/Setpoint_\(control_system\)](http://en.wikipedia.org/wiki/Setpoint_(control_system))
- Wikipedia. (2010). *Circuitos Integrados demultiplexores*. Retrieved from Circuitos Integrados demultiplexores: <http://es.wikipedia.org/wiki/Demultiplexor>
- Wikipedia. (2010). *Control PID, conceptos*. Retrieved from http://es.wikipedia.org/wiki/Proporcional_integral_derivativo

