



ESPE

UNIVERSIDAD DE LAS FUERZAS ARMADAS
INNOVACIÓN PARA LA EXCELENCIA

DEPARTAMENTO DE ELÉCTRICA, ELECTRÓNICA Y
TELECOMUNICACIONES

CARRERA DE INGENIERÍA EN ELECTRÓNICA, AUTOMATIZACIÓN Y
CONTROL

TRABAJO DE TITULACIÓN, PREVIO A LA OBTENCIÓN DEL TÍTULO
DE INGENIERO EN ELECTRÓNICA, AUTOMATIZACIÓN Y CONTROL

TEMA: DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE
MOVIMIENTO PARA SILLAS DE RUEDAS CONVENCIONAL MEDIANTE
EL USO DE UN CONTROLADOR PI DIFUSO ADAPTATIVO

AUTOR: IGUAGO ZAPATA, LENIN MAURICIO

DIRECTOR: ING. TIPÁN CONDOLO, EDGAR FERNANDO, MSc.

SANGOLQUÍ

2020



ESPE

UNIVERSIDAD DE LAS FUERZAS ARMADAS
INNOVACIÓN PARA LA EXCELENCIA

DEPARTAMENTO DE ELÉCTRICA, ELECTRÓNICA Y TELECOMUNICACIONES

CARRERA DE INGENIERÍA EN ELECTRÓNICA, AUTOMATIZACIÓN Y CONTROL

CERTIFICACIÓN

Certifico que el trabajo de titulación, “DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE MOVIMIENTO PARA SILLAS DE RUEDAS CONVENCIONAL MEDIANTE EL USO DE UN CONTROLADOR PI-DIFUSO ADAPTATIVO” fue realizado por el señor Iguago Zapata, Lenin Mauricio el mismo que ha sido revisado en su totalidad, analizado por la herramienta de verificación de similitud de contenido; por lo tanto cumple con los requisitos teóricos, científicos, técnicos, metodológicos y legales establecidos por la Universidad De Las Fuerzas Armadas ESPE, razón por la cual me permito acreditar y autorizar para que lo sustente públicamente.

Sangolquí, 16 de enero del 2020.

FIRMA

Ing. Tipán Condolo, Edgar Fernando, MSc.

DIRECTOR



ESPE

UNIVERSIDAD DE LAS FUERZAS ARMADAS
INNOVACIÓN PARA LA EXCELENCIA

DEPARTAMENTO DE ELÉCTRICA, ELECTRÓNICA Y TELECOMUNICACIONES

CARRERA DE INGENIERÍA EN ELECTRÓNICA, AUTOMATIZACIÓN Y CONTROL

AUTORÍA DE RESPONSABILIDAD

Yo, Iguago Zapata, Lenin Mauricio, declaro que el contenido, ideas y criterios del trabajo de titulación: "DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE MOVIMIENTO PARA SILLAS DE RUEDAS CONVENCIONAL MEDIANTE EL USO DE UN CONTROLADOR PI DIFUSO ADAPTATIVO", es de mi autoría y responsabilidad, cumpliendo con los requisitos teóricos, científicos, técnicos, metodológicos y legales establecidos por la Universidad de las Fuerzas Armadas ESPE, respetando los derechos intelectuales de terceros y referenciado las citas bibliográficas.

Consecuentemente el contenido de la investigación mencionada es veraz.

Sangolquí, 17 de enero del 2020.

A handwritten signature in blue ink, appearing to read 'Iguago Zapata'.

Iguago Zapata, Lenin Mauricio

C.C.: 1712251063



ESPE

UNIVERSIDAD DE LAS FUERZAS ARMADAS
INNOVACIÓN PARA LA EXCELENCIA

DEPARTAMENTO DE ELÉCTRICA, ELECTRÓNICA Y TELECOMUNICACIONES

CARRERA DE INGENIERÍA EN ELECTRÓNICA, AUTOMATIZACIÓN Y CONTROL

AUTORIZACIÓN

Yo, Iguago Zapata, Lenin Mauricio, autorizo a la Universidad de las Fuerzas Armadas ESPE publicar el trabajo de titulación: "DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE MOVIMIENTO PARA SILLAS DE RUEDAS CONVENCIONAL MEDIANTE EL USO DE UN CONTROLADOR PI DIFUSO ADAPTATIVO" en el Repositorio Institucional, cuyo contenido, ideas y criterios son de mi responsabilidad.

Sangolquí, 17 de enero del 2020.

Iguago Zapata, Lenin Mauricio

C.C.: 1712251063

DEDICATORIA

En especial a Mi Esposa que pese a los problemas persistentes nunca se dio por vencida y me dio fuerzas para seguir adelante.

A Mis Padres y Suegros que me mostraron todo su apoyo en estos últimos momentos.

Y a Nuestros seres queridos que ya no se encuentran con nosotros pero siempre estarán en nuestro corazón

AGRADECIMIENTO

A Dios, Mis Padres y Mis Suegros, ya que con su apoyo y ayuda me permitieron terminar la carrera y sobre todo a Mi Esposa que fue Mí Pilarte en estos últimos años para terminar el proyecto.

ÍNDICE DE CONTENIDO

CARATULA	
CERTIFICADO DE DIRECTOR Y CODIRECTOR	i
AUTORIA DE RESPONSABILIDAD	ii
AUTORIZACIÓN	iii
DEDICATORIA	iv
AGRADECIMIENTO	v
ÍNDICE DE CONTENIDO	vi
ÍNDICE DE TABLAS.....	ix
ÍNDICE DE FIGURAS	x
RESUMEN.....	xiii
ABSTRACT	xiv
CAPÍTULO I.....	1
1. INTRODUCCIÓN.....	1
1.1. Antecedentes.....	1
1.2. Autonomía e Integración de Una Persona Discapacitada Físicamente.....	2
1.2.1. Factor Psicológico y Motriz.....	2
1.2.2. Integración Física	3
1.3. Antecedentes.....	3
1.3.1. Integración de Silla De Ruedas	3
1.3.2. Costos Actuales de Sillas De Ruedas	4
1.3.3. Formas de Abaratamientos de Costos.	4
1.4 Tipos De Control.....	4
1.4.1 Control PID	4
1.4.2 Control Fuzzy	9
1.4.3 Control PI-Fuzzy Adaptativo	13
1.5. Planteamiento del Problema.....	15
1.5.1. Problemas Mecánicos de la Silla de Ruedas.....	15
1.5.2. Integración de Circuitos Electrónicos y de Potencia	16
1.6. Alcance.....	19

	vii
1.6.1. SUBSISTEMA MECÁNICO:	20
1.6.2. SUBSISTEMA DE CONTROL:	20
1.6.3. SUBSISTEMA DE RECARGA Y POTENCIA:	20
1.7. Objetivos.....	21
1.7.1. General.....	21
1.7.2. Específicos	22
CAPÍTULO II.....	23
2. Análisis y diseño mecánico.....	23
2.1. Funcionamiento De La Silla De Ruedas.	23
2.2. Análisis de Estabilidad.....	25
2.3. Evaluación De Soluciones Mecánicas	27
2.3.1. Selección de Materiales y Proceso de Fabricación.	30
2.3.2. Diseño Mecánico.	33
CAPÍTULO III.....	35
3. DISEÑO DEL SISTEMA DE CONTROL.....	35
3.1. Sistema de Alimentación.	35
3.2. Análisis y Selección de Dispositivo de Control.	38
3.2.1. Actuadores	38
3.2.2. Controladores	41
3.2.3. Pre-Actuadores.....	42
3.2.4. Sensores	44
3.3. Control PI Difuso Adaptativo.....	51
3.4. Simulación y Resultados	64
CAPÍTULO IV	66
4. IMPLEMENTACIÓN PARA LA MOVILIZACIÓN DE LA SILLA DE RUEDAS.....	66
4.1. Instalación de Sistemas Mecánicos y Eléctricos.	66
4.1.1. Instalación de Parte Mecánica.....	66
4.1.2. Instalación del Sistema de Control	67
4.2. Pruebas y Análisis de la Silla en Condiciones Alternas.	70
4.2.1. Pruebas en Condiciones Alternas	71
4.2.2. Análisis de Pruebas	72
4.3. Validación de Resultados	72

	viii
CAPÍTULO V	73
5. CONCLUSIONES Y RECOMENDACIONES.....	73
5.1. Conclusiones	73
5.2. Recomendaciones	74
Referencias Bibliográficas	75
Anexos	79
Apéndice 1: Eje Del Motor	79
Apéndice 2: Catalina	80
Apéndice 3: Buje De Catalina.....	81
Apéndice 4: Eje De Bujes	82
Apéndice 5: Base Motor 1	83
Apéndice 6: Base Motor 2	84
Apéndice 7: Perfiles Estructurales Tipo “G”	85
Apéndice 8: Programa De Arduino Nano	86
Apéndice 9: Programa de Arduino Mega.....	90

ÍNDICE DE TABLAS

Tabla 1: <i>Representación Matemática del Control Proporcional</i>	6
Tabla 2: <i>Representación Matemática del Control Integral</i>	6
Tabla 3: <i>Representación Matemática del Control Derivativo</i>	6
Tabla 4: <i>Reglas De Control Fuzzy De la Constante K_p</i>	14
Tabla 5: <i>Reglas De Control Fuzzy De la Constante K_i</i>	14
Tabla 6: <i>Cuadro Comparativo de Motores</i>	26
Tabla 7: <i>Cuadro Comparativo Del Tiempo de Respuesta de Controladores</i>	72

ÍNDICE DE FIGURAS

Figura 1. Registro De Personas Discapacitadas	1
Figura 2. Efectos De Los Términos Independientes Del Controlador PID	7
Figura 3. Estructura Paralela Del Controlador PID	7
Figura 4. Modos de Control del PID.....	8
Figura 5. Acción del Controlador PI.....	9
Figura 6. Estaciones Del Año Consideradas De Forma Binaria Y Analógica	10
Figura 7. Operación Del Controlador De Lógica Difusa.....	10
Figura 8. Forma De Las Funciones Miembro.....	11
Figura 9. Grafico Del Sistema De Lógica Difusa.....	12
Figura 10. Calculo Del Centroide De La Salida De Control	13
Figura 11. Controlador PI-Fuzzy Adaptativo	13
Figura 12. Diagrama de Bloques Del Proceso Mecánico	15
Figura 13. Diagrama De Bloques Del Circuito Electrónico	17
Figura 14. Dimensiones De La Silla De Ruedas.....	23
Figura 15. Motor Limpiaparabrisas	27
Figura 16. Eje de Catalina Asegurada al Eje del Motor	29
Figura 17. Catalina Soldada a un Eje	29
Figura 18. Suelda Entre Bujes.....	31
Figura 19. Buje Con Rosca De Piñón Fijo de 36 radios.....	32
Figura 20. Eje Principal De Bujes	32
Figura 21. Bases De Motor	33
Figura 22. Suelda Del Perfil Estructural G	33
Figura 23. Batería De Motocicleta	36
Figura 24. Fuente de 20A y 12V	36
Figura 25. Step Down 2596S.....	37
Figura 26. Motor De Arranque	39
Figura 27. Motor DC	39
Figura 28: Arduino Mega 2560	41
Figura 29. Arduino Nano V3	42
Figura 30. Puente H Modelo “BTS7960B”	42
Figura 31. Pines De Conexión Puente H.....	43

Figura 32. Sensor De Proximidad “2Y0A21”	45
Figura 33. Compilación Programa Sensor De Pared	45
Figura 34. Parámetros Del Sensor	46
Figura 35. Triángulo Formado Al Ingresar En Una Rampa.....	47
Figura 36. Compilación Programa Sensor De Gradass.....	48
Figura 37. Triángulo Formado Al Bajar La Rampa	49
Figura 38. Giroscopio Analógico GY-61	49
Figura 39. Compilación Programa Sensor De Inclinación Pendiente Positiva	50
Figura 40. Seguidor De Línea.....	51
Figura 41. Programador Arduino Mega Pines Digitales	52
Figura 42. Programador Arduino Mega, Pines PWM.....	53
Figura 43. Programador Arduino Mega Pines De Velocidad	54
Figura 44. Incremento Y Decremento de La Velocidad	55
Figura 45. Programa Arduino Mega Movimiento Para Atrás.....	56
Figura 46. Programa Arduino Mega Movimiento Para La Izquierda	56
Figura 47. Programador Arduino Mega Movimiento Para La Derecha	57
Figura 48. Herramienta de Identificación de la Planta en MatLab	58
Figura 49. Coeficientes de la Función de Transferencia de Segundo Orden	59
Figura 50. Respuesta Temporal De Resultados Experimentales Y Teóricos	59
Figura 51. Ajuste de los Parámetros del Controlador PI.....	60
Figura 52. Herramienta De Diseño Del Control Fuzzy de MatLab	61
Figura 53. Diagrama De Bloques Del Control PI-Difuso Adaptativo Del Motor.....	61
Figura 54. Reglas De Fuzzificación	62
Figura 55. Diagrama Secuencial del Controlador PI-Difuso Adaptativo.....	62
Figura 56. Código De MatLab Equivalente En Arduino	63
Figura 57. Estructura De Control Del PI-Difuso Adaptativo	64
Figura 58. Diagrama de Bloques De Controladores PI y PI-Difuso Adaptativo.....	64
Figura 59. Respuesta Temporal De Los Controladores Y Adaptación De Los Parámetros Kp y Ki.....	65
Figura 60. Esquema De La Ubicación De Los Accesorios Mecánicos	66
Figura 61. Diagrama Electrónico Del Sistema De Control	67
Figura 62. Programación Sensores	68

	xii
Figura 63. Conexiones Del Joystick.....	69
Figura 64. Conexiones De Display Y Botones	69
Figura 65. Caja de Control de Velocidad y Movimiento	70
Figura 66. Silla De Ruedas Terminada	70
Figura 67. Respuesta Temporal Para Peso de 100Kg	71
Figura 68. Respuesta Temporal Para Peso De 75Kg	71
Figura 69. Respuesta Temporal Para Peso De 60Kg	72

RESUMEN

En la teoría de control clásica, técnicas como los controladores PID, son utilizados en la mayoría de procesos industriales. Esto es debido a la simplicidad e implementación de estos controladores. Sin embargo, los nuevos avances tecnológicos permiten implementar reguladores más complejos para sistemas de control no lineales, permitiendo la optimización de energía ante variaciones externas, tales como los controladores de lógica difusa. Por lo anteriormente expuesto, en el Presente Proyecto de Titulación se hablara del controlador PI-Difuso Adaptativo generado para controlar los motores de una silla de ruedas, con lo cual se pretende optimizar la energía evitando sobrecargas al momento del arranque y permitiendo un tiempo de estabilización más rápido. Para la implementación electrónica se utilizó microcontroladores de bajo costo como el arduino mega y el arduino nano que fueron suficiente para la implementación del controlador PI-Difuso, adicionalmente se usó los Puentes H compatibles con Arduino, para controlar motores de gran amperaje. También se observó diferentes opciones para la implementación mecánica del sistema de transmisión de los motores, cambiando a sistemas de piñones a catalinas observando su comportamiento y analizando sus problemas. Gracias a los estudios realizados con este proyecto se comprueba que el control PI-Difuso Adaptativo tiene parámetros más estables que el control PI y el Control Difuso. Finalmente, mediante perturbaciones en el sistema de control PI-Difuso Adaptativo se verifica la robustez del mismo con respecto al tiempo de reacción de los parámetros dados por la lógica PI-Difusa.

PALABRAS CLAVE:

- **Movilidad**
- **Control De Procesos**
- **PI Difuso**
- **PI Fuzzy**

ABSTRACT

In classical control theory, techniques such as PID controllers are used in most industrial processes. This is due to the simplicity and implementation of these drivers. However, with the new technological advances, it is possible to implement more complex regulators for non-linear control systems, allowing energy optimization in the face of external variations, such as fuzzy logic controllers. Due to the above, in the present Degree Project, the adaptive Diffuse PI-controller generated to control the motors of a wheelchair will be discussed, with which it is intended to optimize the energy avoiding overloads at the time of starting and allowing a stabilization time faster. For the electronic implementation, low-cost microcontrollers such as the arduino mega and the arduino nano were used, which were sufficient for the implementation of the PI-Diffuse controller. Additionally, the H Bridges compatible with Arduino were used to control high-amperage motors. Different options were also observed for the mechanical implementation of the engine transmission system, changing from pinion systems to catalysts observing their behavior and analyzing their problems. Thanks to the studies carried out with this project, it is verified that the Adaptive Diffuse PI control has more stable parameters than the PI control and the Diffuse Control. Finally, by means of disturbances in the adaptive PI-Diffuse control system, its robustness is verified with respect to the reaction time of the parameters given by the PI-Diffuse logic.

KEYWORDS

- **Mobility**
- **Process Control**
- **PI Diffuse**
- **PI Fuzzy**

CAPÍTULO I

1. INTRODUCCIÓN

1.1. Antecedentes

En la actualidad y gracias al gobierno del Presidente Lenin Moreno se está tomando más atención a las personas discapacitadas, es por esto que por el PRIMER ESTUDIO BIO-PSICO SOCIAL CLÍNICO Y GENÉTICO DE LAS PERSONAS CON DISCAPACIDAD EN EL ECUADOR – MISIÓN SOLIDARIA MANUELA ESPEJO, se logró establecer que de cada 100 habitantes existen 4 personas con algún tipo de discapacidad, esto da un alcance de 563.515 de las cuales se tienen 447.294 personas registradas en el CONADIS, como se puede observar en la figura 1.

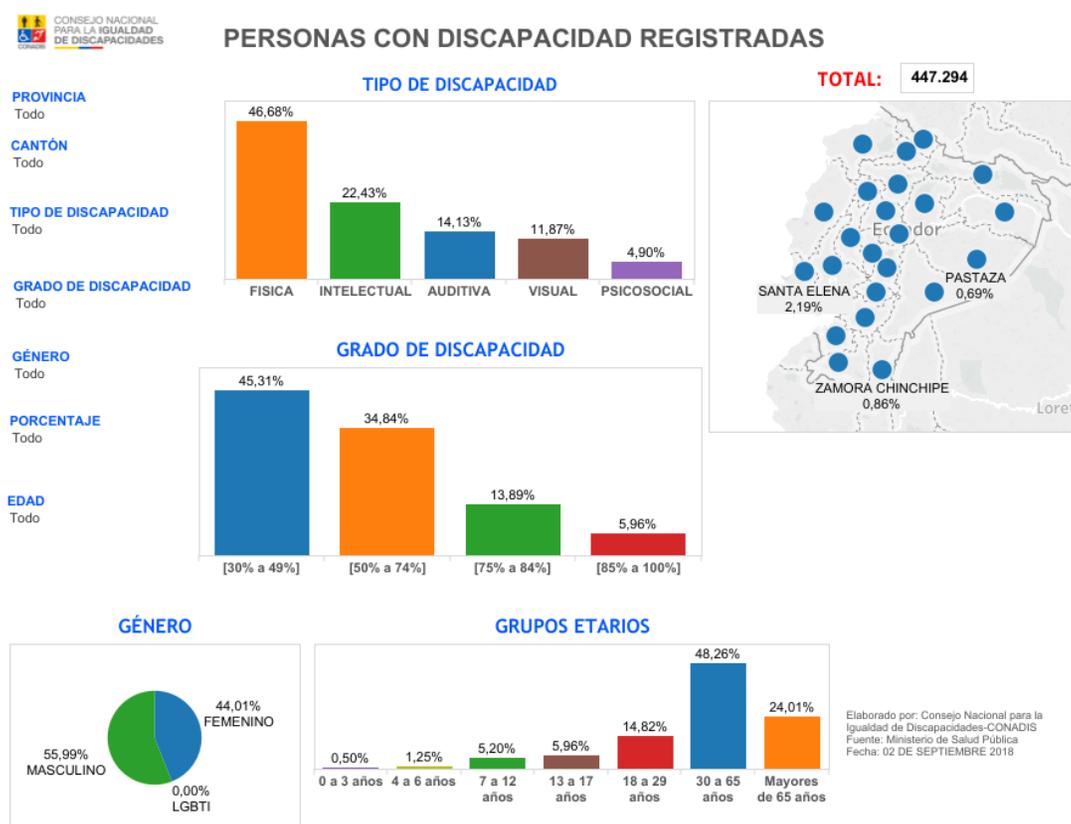


Figura 1. Registro De Personas Discapacitadas

Fuente: Consejo De Discapacidades

De los cuales en la actualidad hay más del 46,68% de personas con algún grado de discapacidad física, lo que equivale a cerca de 208.775 personas y de aquí que muchos de ellos no pueden moverse. Lo que se trata con el presente proyecto es adaptar, a una silla de ruedas convencional, los elementos necesarios para que pueda moverse. Hay que recordar que muchas de estas personas son de escasos recursos, por lo que otro punto que se va a tratar en el presente proyecto es el bajo costo de los elementos a usarse, para que esté al alcance de la mayoría, con esto se logrará que parte de esa población discapacitada físicamente pueda moverse por sus propios medios.

1.2. Autonomía e Integración de Una Persona Discapacitada Físicamente.

En la actualidad una persona discapacitada con problemas de movilidad, especialmente cuando son de la tercera edad, se resignan a no poder moverse ya sea por la edad, dificultades en el manejo de la silla, falta de fuerza en sus extremidades superiores, entre otros inconvenientes, que se pueden presentar al tratar de mover las sillas de ruedas convencionales o por algún otro obstáculo que le impide su libre movilización como los altos costos de las sillas de ruedas eléctricas.

1.2.1. Factor Psicológico y Motriz

Una persona que tiene autonomía en su movilización psicológicamente tendrá una mayor autoestima de la que no, la auto-movilización física no es solo importante es significativa para recuperar esta autoestima para confirmar que son capaces de lograr cosas por ellos mismo.

Al poderse movilizar una persona no cae en depresión se siente más animada e inclusive se siente con ganas de auto superarse.

1.2.2. Integración Física

Una persona con auto-movilización física se integra mejor a la sociedad, busca auto superarse y es más animada con su entorno, es por esto que una persona con discapacidad física se integra mejor con su silla de ruedas.

La integración física que una persona discapacitada tiene hacia su silla de ruedas convencional es paulatina, y a veces demasiado exigente dependiendo de su discapacidad, con una silla de ruedas que se mueve con un simple mando esta persona se integra más rápido a la sociedad y no dependería de su tipo de discapacidad.

1.3. Antecedentes

1.3.1. Integración de Silla De Ruedas

Este proyecto es de gran importancia social puesto que pretende beneficiar a personas con discapacidades y se lo puede acoplar a cualquier tipo de sillas de ruedas que ya se tenga, por otro lado, también se tiene un componente investigativo al tratar de implementar un algoritmo de control moderno para su manipulación lo cual da un valor agregado adicional ya que siempre se pretende mantener el hecho de que sea un sistema de bajo costo que esté al alcance de personas de recursos económicos limitados.

Es importante mencionar que este proyecto busca fomentar en el Ecuador el desarrollo e implementación de tecnologías aplicándolas de forma que permita satisfacer todos los requerimientos básicos y hasta los más ambiciosos de las personas discapacitadas. De esta manera se contribuye al desarrollo del país y se apega a las políticas de cambio de la matriz productiva que hoy en día están surgiendo y alineado a lo establecido en el plan del buen vivir.

1.3.2. Costos Actuales de Sillas De Ruedas

En la actualidad no existe en Ecuador una empresa que cree sillas de ruedas automatizadas, por lo que la mayoría de sillas de ruedas automáticas las traen de importación subiendo sus costos, en el mercado existen sillas con costos mayores a los 1.500 dólares.

Lo que se busca es poder llegar a personas con escasos recursos mejorándoles sus estilos de vida e integrándoles a la sociedad.

1.3.3. Formas de Abaratamientos de Costos.

Gracias al avance de la tecnología en la industria se puede observar que se puede abaratar costos en elementos que antiguamente eran costosos o inaccesibles, es así que actualmente estos elementos se los puede aplicar en el presente proyecto para controlar su potencia, reducir y hacer más eficiente su consumo de energía.

1.4 Tipos De Control

Para el presente Proyecto se usa el Controlador PI-Fuzzy Adaptativo, por lo cual se debe hacer una comprensión de los controladores PI; Fuzzy y la integración en el Controlador PI- Difuso Adaptativo por separado.

1.4.1 Control PID

Para entender completamente al controlador PID se hará una breve explicación del funcionamiento de los procesos integral y derivativo.

Ante todo, hay que recordar que este tipo de controlador maneja un proceso de bucle en el cual su operación principal es mantener la variable del proceso en un valor objetivo deseado. El bucle lee la variable del proceso y lo compara con el valor del punto de ajuste, el controlador lee estos dos valores y determinara cuanta acción correctiva debe tomar. Este tipo de control con bucle de retroalimentación se llama

controlador de circuito cerrado, en el cual el controlador determina el error que existe mediante la resta de la variable en ese momento del proceso y el punto de ajuste a donde queremos que llegue.

En este tipo de controlador no solo depende del tipo de respuesta que da el controlador al proceso, también depende del tiempo de reacción para el mismo, con estos datos se forman funciones de transferencia entre el controlador y el proceso, se recordara que la función de transferencia es la representación matemática de un sistema. Estas funciones de transferencia se las expresa en el dominio de la frecuencia, lo que nos permite usar la transformada de Laplace para crear ecuaciones fáciles de manejar algebraicamente.

Para el presente proyecto se usó una ecuación de Laplace de Segundo orden como se verá posteriormente. Estas ecuaciones de segundo orden tienen 2 tiempos de retraso y se la puede observar en la ecuación siguiente:

$$H_{(s)} = \frac{Out}{In} = \frac{A\omega_n^2}{s^2 + 2\varphi\omega_n s + \omega_n^2}$$

A continuación, se va a explicar el funcionamiento del controlador PID el cual consta de 3 modos, los cuales reaccionan de forma diferente a la entrada del error en el sistema usando la misma función de segundo grado antes definida.

- Modo Proporcional: El controlado está definido por el termino K_p , en el cual la acción de control es proporcional a la señal de error del proceso. La representación para el control proporcional en el tiempo y dominio de Laplace se define como se ve en la tabla 1:

Tabla 1:*Representación Matemática del Control Proporcional*

Dominio del Tiempo	$u_c(t) = k_p e(t)$
Dominio de Laplace	$U_c(s) = k_p E(s)$

- Modo Integral: El modo integral provee una salida cuya razón de cambio es proporcional a la desviación del error. Esto significa que si el error es grande la salida de control cambia de manera rápida, permitiendo eliminar el error en modo estacionario. El controlador está definido por el término K_i , la representación para el control integral en el tiempo y dominio de Laplace se define como se ve en la tabla 2:

Tabla 2:*Representación Matemática del Control Integral*

Dominio del Tiempo	$u_c(t) = k_I \int e(\tau) d\tau$
Dominio de Laplace	$U_c(s) = \left[\frac{k_I}{s} \right] E(s)$

- Modo Derivativo: El modo derivativo provee una salida proporcional a la tasa de cambio del error en el sistema, introduciendo un elemento de predicción en la acción de control. En el control derivativo está definido por el término K_d , la representación para el control derivativo en el tiempo y dominio de Laplace se define como se define en la tabla 3:

Tabla 3:*Representación Matemática del Control Derivativo*

Dominio del Tiempo	$u_c(t) = k_D \frac{de}{dt}$
Dominio de Laplace	$U_c(s) = [k_D s] E(s)$

Para la estructura del controlador PID, el comportamiento de los diferentes términos de control para una referencia y una perturbación en modo transitorio y estado estacionario se definen en la siguiente figura 2.

	Reference tracking tuning Step reference		Disturbance rejection tuning Constant load disturbance	
	Transient	Steady state	Transient	Steady state
P	Increasing $k_p > 0$ speeds up the response	Increasing $k_p > 0$ reduces but does not eliminate steady state offset	Increasing $k_p > 0$ speeds up the response	Increasing $k_p > 0$ reduces but does not eliminate steady state offset
I	Introducing integral action $k_i > 0$ gives a wide range of response types	Introducing integral action $k_i > 0$ eliminates offset in the reference response	Introducing integral action $k_i > 0$ gives a wide range of response types	Introducing integral action $k_i > 0$ eliminates steady state offsets
D	Derivative action $k_D > 0$ gives a wide range of responses and can be used to tune response damping	Derivative action has no effect on steady state offset	Derivative action $k_D > 0$ gives a wide range of responses and can be used to tune response damping	Derivative action has no effect on steady state offset

Figura 2. Efectos De Los Términos Independientes Del Controlador PID

Fuente: PID Control

Los diferentes términos de control mencionados anteriormente se los puede combinar de manera paralela como se muestra en la figura 3.

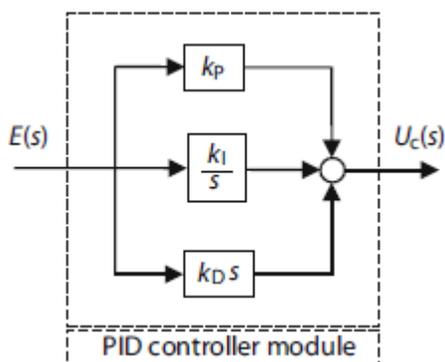


Figura 3. Estructura Paralela Del Controlador PID

Fuente: Arquitectura Paralela del Controlador PID

Esto implica que se pueden existir diferentes combinaciones posibles del controlador PID, tal como se muestra en la figura 4.

Controller Mode		Response	Applications
Proportional	P	CV changes in proportion to E	Systems with small load changes and/or small to moderate lag times
Integral	I	CV changes according to how E changes over time	Processes with small process lags and small capacities
Derivative	D	CV changes according to how fast E changes	Not used alone in applications
Proportional-Integral	PI	CV responds in a combination of P and I actions	Systems with large load changes
Proportional-Derivative	PD	CV responds in a combination of P and D actions	Processes with fast load changes
Proportional-Integral-Derivative	PID	CV responds in a combination of P, I, and D actions	Can be used in practically all process control applications

Figura 4. Modos de Control del PID

Fuente: Modos de Control

En el presente proyecto se utiliza un controlador PI, donde se excluye el control derivativo ya que este considera en el sistema cuán rápido cambia el valor del error en un instante de tiempo, produciendo una gran derivada si el proceso es muy rápido y esto generaría una inestabilidad en todo el proceso de control.

El controlador PI es el controlador más utilizado en la industria, el cual permite variar sus dos parámetros de forma fácil. Entre las ventajas que tiene este tipo de controlador

destacan: una respuesta rápida ante perturbaciones y elimina los errores en estado estacionario, como se ve en la figura 5.

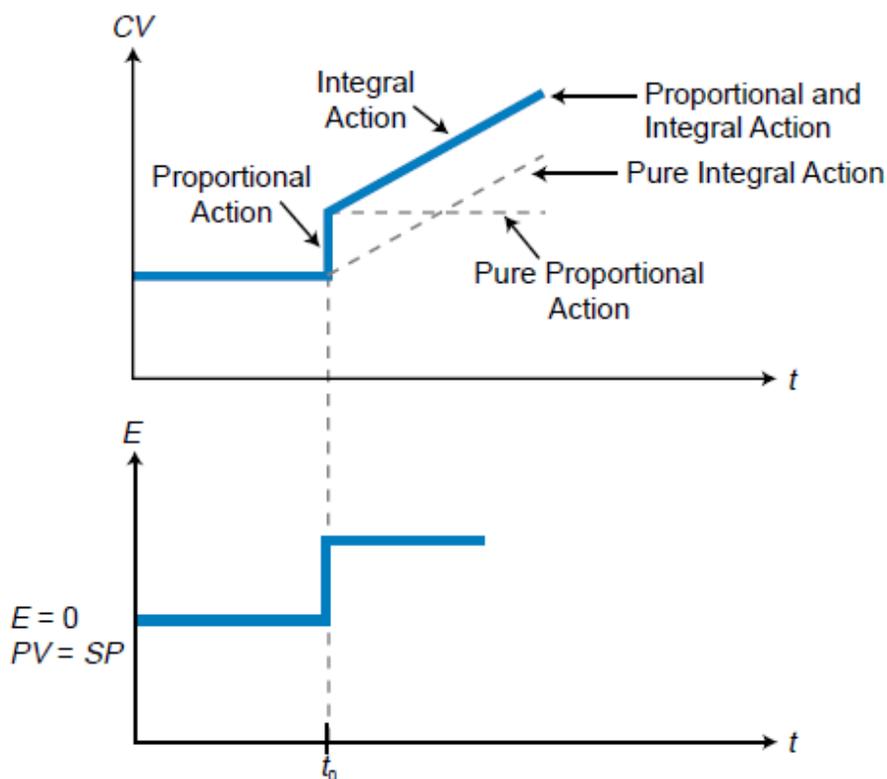


Figura 5. Acción del Controlador PI

Fuente: Acción del controlador Proporcional Integral

Este controlador PI generalmente es bastante simple de calcular e implementar sus parámetros, sin embargo, algunos procesos industriales necesitan controladores más precisos tales como los controladores de lógica difusa

1.4.2 Control Fuzzy

La lógica difusa es una rama de la inteligencia artificial, que se ocupa del razonamiento mediante algoritmos usados para emular el pensamiento humano. Esto se aplica a técnicas de control donde el proceso no puede ser representado en forma binaria, como se lo demuestra en la figura 6 con las estaciones del año.

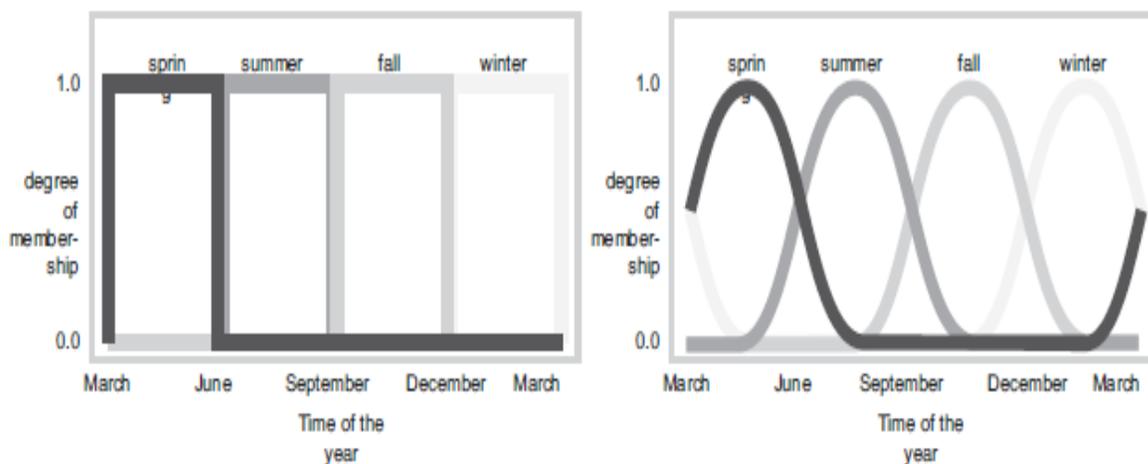


Figura 6. Estaciones Del Año Consideradas De Forma Binaria Y Analógica

Fuente: Diagrama de temperaturas en el Hemisferio Norte

El controlador Fuzzy conlleva 3 acciones principales:

- Fuzzificación.
- Procesamiento Fuzzy
- Defuzzificación

Estas 3 acciones siempre mantienen su orden, como se muestra en la figura 7.

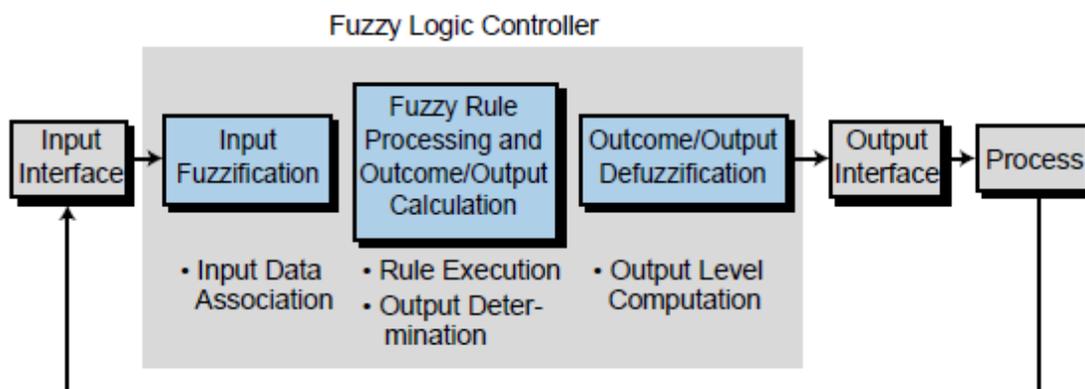


Figura 7. Operación Del Controlador De Lógica Difusa

Fuente: Fuzzy Logic Controller Operation

El controlador difuso recibe los datos de entrada, los cuales traduce a una forma difusa. Este proceso se llama Fuzzificación, este controlador realiza el procesamiento

que involucra la evaluación de la información de entrada de acuerdo a las reglas “IF....THEN”.

Una vez que el controlador difuso termina la etapa de procesamiento de reglas, llega a un resultado, donde comienza el proceso de Defuzzificación. En este último paso el controlador convierte los resultados de salida en datos reales para el actuador.

Las funciones miembro pueden tener diferentes formas dependiendo del conjunto de datos, pudiendo ser tipo S, Z, Λ and Π , representadas en la figura 8.

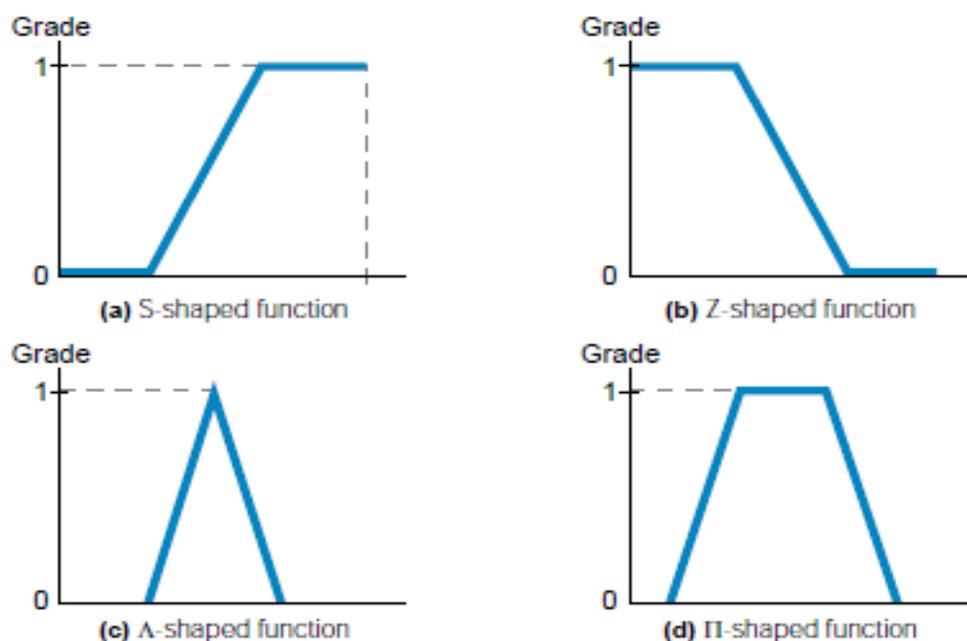


Figura 8. Forma De Las Funciones Miembro

Fuente: Forma de las Funciones Miembro

Cada una de estas funciones miembro tienen niveles que permiten actuar al controlador Difuso, cuyos rangos de datos son: Negativo Grande “NL” (punto mínimo), Negativo Medio “NM”, Negativo Pequeño “NS”, cero “ZR”, Positivo Pequeño “PS”, Positivo Medio “PM” y hasta Positivo Grande “PL” (punto máximo). Donde el número máximo de niveles es 7 y mínimo 3.

La implementación y operación de la lógica de control Fuzzy se muestra en la figura

9.

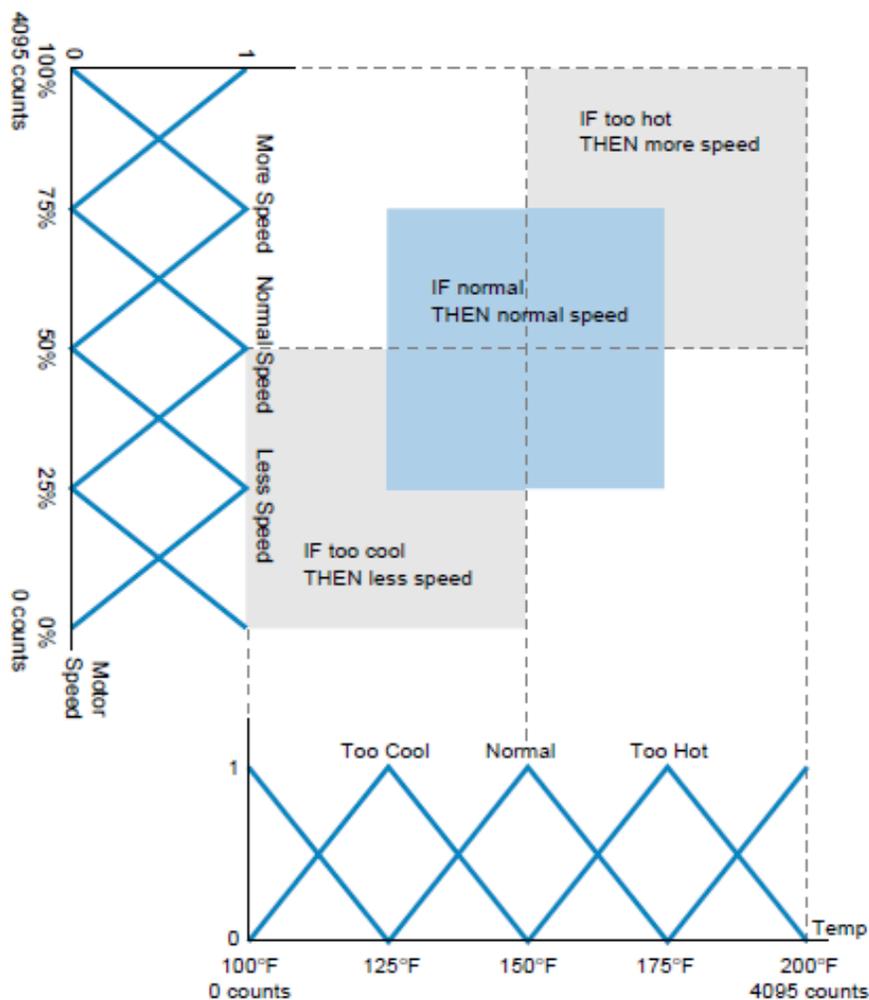


Figura 9. Grafico Del Sistema De Lógica Difusa

Fuente: Sistema de Lógica Difusa con Entrada y Salida

La entrada para el proceso de Defuzzificación es un conjunto difuso con una salida de un solo número, donde existen diferentes métodos tales como: centroide, bisectriz, mayor de los máximos, medio de los máximos y mínimo de los máximos. El método más popular y utilizado es el del cálculo del centroide, el cual retorna el centro del área bajo la curva, como se muestra en la figura 10.

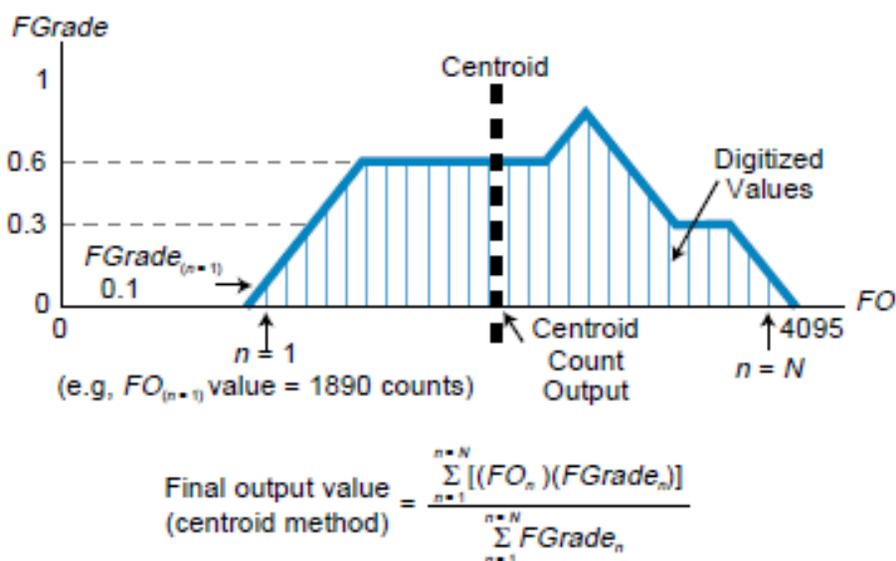


Figura 10. Calculo Del Centroide De La Salida De Control

Fuente: Ubicación del Centroide

Este controlador Fuzzy tiene más parámetros que un controlador PI, por lo que su estabilidad es más difícil de garantizar debido a su complejidad matemática.

1.4.3 Control PI-Fuzzy Adaptativo

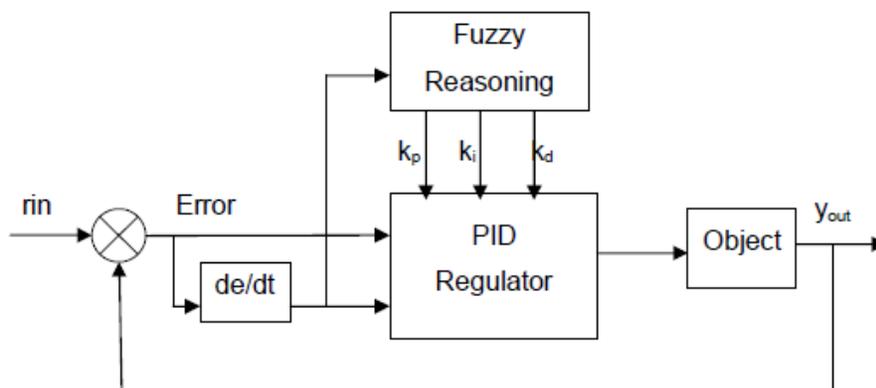


Figura 11. Controlador PI-Fuzzy Adaptativo

Fuente: Diagrama de bloques PI-Fuzzy Adaptativo

Para mejorar la respuesta de un controlador PI, un sistema Fuzzy es usado para ajustar los parámetros del controlador basado en un control de reglas para formar el control adaptativo PI-Fuzzy, se le considera adaptativo por que los parámetros del

controlador PI van cambiando en función de la lógica difusa. Las reglas de control para las constantes K_p y K_i se muestran en las tablas 4 y 5.

Tabla 4:
Reglas De Control Fuzzy De la Constante K_p

$e(t)\dot{e}(t)$	NB	NM	NS	ZO	PS	PM	PB
NB	PB	PB	PM	PM	PS	ZO	ZO
NM	PB	PB	PM	PS	PS	ZO	NS
NS	PM	PM	PM	PS	ZO	NS	NS
ZO	PM	PM	PS	ZO	NS	NM	NM
PS	PS	PS	ZO	NS	NS	NM	NM
PM	PS	ZO	NS	NM	NM	NM	NB
PB	ZO	ZO	NM	NM	NM	NB	NB

Fuente: Reglas de Tabla de Control Fuzzy K_p

Tabla 5:
Reglas De Control Fuzzy De la Constante K_i

$e(t)\dot{e}(t)$	NB	NM	NS	ZO	PS	PM	PB
NB	NB	NB	NM	NM	NS	ZO	ZO
NM	NB	NB	NM	NS	NS	ZO	ZO
NS	NB	NM	PS	NS	ZO	PS	PS
ZO	NM	NM	NS	ZO	PS	PM	PM
PS	NM	NS	ZO	PS	PS	PM	PB
PM	ZO	ZO	PS	PS	PM	PB	PB
PB	ZO	ZO	PS	PM	PM	PB	PB

Fuente: Reglas de Tabla de Control Fuzzy K_i

1.5. Planteamiento del Problema

El Hogar de Ancianos “Dulce María”, ubicado al sur de la Ciudad de Quito, ofrece servicios de cuidado al adulto mayor, atiende aproximadamente a 50 personas, con su alimentación, medicación, terapia y cuidados, de los cuales más del 60% tiene algún tipo de discapacidad física por su edad, por lo que presentan problemas en su movilidad. Si bien es cierto esta institución es de carácter privado, este hogar específicamente está orientado a personas con escasos recursos económicos, del sector sur de la ciudad.

Este Hogar de Ancianos ofrece una oportunidad para mejorar el estatus de comodidad de las personas, esta locación tiene rampas de 15° de inclinación, gradas normalizadas de 18cm x 25cm y barandillas, que son los obstáculos primordiales a superarse.

1.5.1. Problemas Mecánicos de la Silla de Ruedas

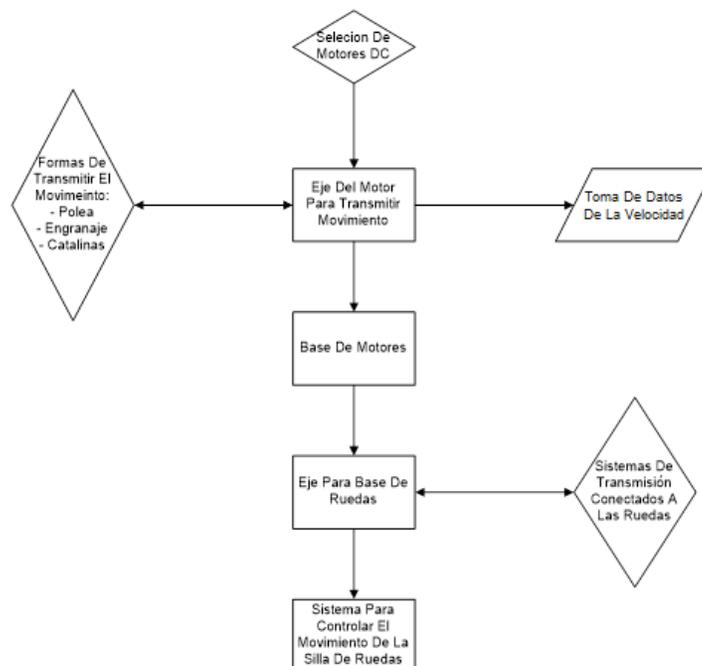


Figura 12. Diagrama de Bloques Del Proceso Mecánico

Según el diagrama de Bloques de la figura 12 se procederá a resolver los problemas mecánicos de la siguiente forma: Su motricidad se la realizó por medio de motores DC, se realizará una selección entre 2 tipos de motores: Motor de Arranque y Motor para Limpiaparabrisas.

Según el motor seleccionado se determinara un eje de transmisión, el cual llevara el movimiento a las ruedas de la silla, se estudiaran 3 formas de transmisión: poleas, engranajes o catalinas. Una vez seleccionado el motor y su forma de transmisión se tomarán datos de la velocidad.

Se soldaran las bases del motor a la carcasa de la silla según datos del motor seleccionado, también se procederá a soldar el eje que servirá como base para las ruedas de la silla. El eje base será determinado según la forma de transmisión que se haya seleccionado para la silla de ruedas.

Al observar sillas de ruedas eléctricas en el mercado se decidió seleccionar un Joystick como la mejor forma para controlar los movimientos de la silla de ruedas del presente proyecto.

1.5.2. Integración de Circuitos Electrónicos y de Potencia

Estos regulan la entrada de corriente hacia los motores y permiten regular las revoluciones que este da para que mantenga una velocidad constante.

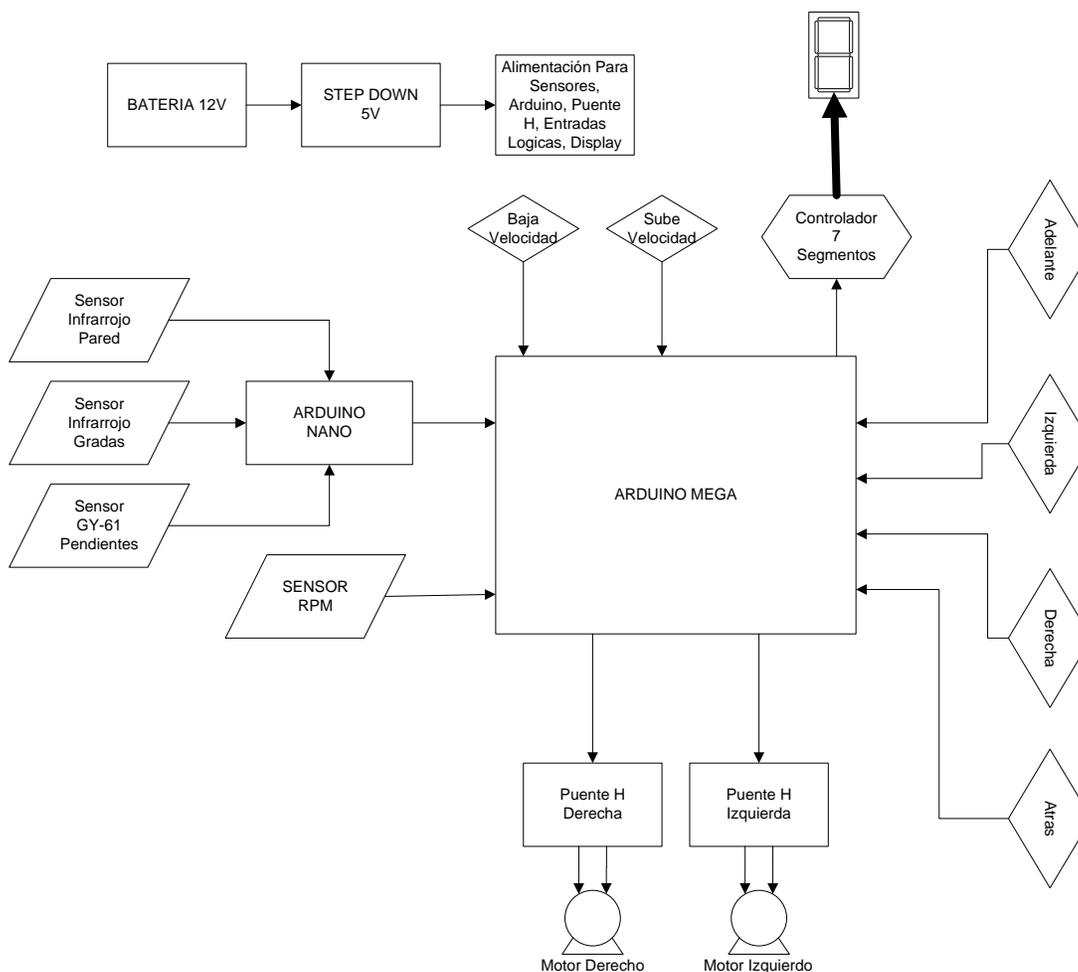


Figura 13. Diagrama De Bloques Del Circuito Electrónico

Desde un inicio se decide trabajar con Arduinos por su bajo costo y fácil manipulación, se usaran 2 arduinos, uno controlara a los motores y otro se encargara exclusivamente de los sensores. Como se observa en la figura 13 el arduino que controla los motores necesita tener varias entradas, ya sea digitales o analógicas y tener varios puntos de salida PWM, por lo que se elige al arduino Mega, en cambio para el arduino que recibe la señal de los sensores, no se necesita mayor complejidad por lo que se decide trabajar con el arduino Nano.

Para movilizar la silla de rueda se necesita motores DC de gran torque lo que implica que se necesitó una gran corriente para su funcionamiento, lo cual nos permitirá trabajar con fuentes DC de gran capacidad y potencia.

Al analizar el tipo de motor DC que se tiene para movilizar la silla de ruedas, este debe de ser de gran Torque, lo que implica que tendrá un gran consumo de energía eléctrica en DC. Al saber que se consume una gran energía eléctrica se podrá limitar la batería para que todo el circuito eléctrico por lo menos tenga un funcionamiento de 2 horas de uso continuo, sin que la silla de ruedas se detenga.

Para alimentar los circuitos de control, sensores y de más componentes electrónicos, se debe trabajar con una fuente de 5V en DC lo que implica que la fuente principal sea del voltaje que tenga la tendremos que reducir. Para que no exista problemas en la alimentación, como se trabaja con las herramientas de arduino se decide trabajar con un Step Down al cual podemos reducirle el voltaje a lo deseado y para no tener limitante de corriente se elige uno de 3A.

Para los sensores infrarrojos se decide trabajar con uno que tenga un alcance no mayor a 80 cm y que nos de datos en forma analógica, lo mismo sucede con el sensor de pendientes, se decidió trabajar con uno que nos de datos fáciles de entender.

Para el sensor que mide las RPM de la salida de los motores, se decide trabajar con un seguidor de línea el cual se lo configurara como un encoger para medir las RPM de las dos ruedas de la silla.

Para visualizar las velocidades a las que se encuentre la silla de ruedas se pondrá un display que estará controlado por el integrado 74HC4052 que recibe las señales del Arduino Mega.

Para los cambios de velocidad se los hace mediante pulsadores y contactores abiertos, esto nos permitirá mandar la señal DC de 5V directamente a las entradas digitales del Arduino Mega.

1.6. Alcance

Para el presente proyecto de titulación se diseñó e implemento la motorización de una Silla De Ruedas a un bajo costo, con un sistema de control fácil de manipular por el usuario, el cual se cambiará de posición junto con sus funciones básicas dependiendo de la movilidad que pueda tener el usuario. Este diseño esta implementado para personas parapléjicas o personas que puedan tener movilidad en una sola mano.

El proyecto está basado en una Silla de Ruedas normal a la cual se le adaptó los mecanismos necesarios para auto-movilizarse por una distancia mínima de 5km, con pendientes máximas de 15° y no se mueve a más de 1,7m/s, que es la velocidad promedio del caminar del ser humano. Se plantea el diseño, instalación y adecuación de todos los componentes, para que funcione con el sistema de control diseñado para el usuario.

El prototipo se desarrolló usando una silla de ruedas donada por una fundación (Free Wheel Chair Misión) al cual se les acopló los elementos necesarios para motorizarlos y tiene un sistema de recarga para poderse conectar fácilmente a cualquier enchufe.

Adicionalmente el circuito eléctrico está protegido contra golpes y daños eléctricos por vibración o derrame de líquidos y su realimentación es relativamente fácil. Se buscaron los mejores costos sin afectar la calidad de los elementos para que el presente proyecto esté al alcance de personas de bajos recursos económicos.

El sistema de control que se implementó para el presente proyecto tiene como fin la autonomía e integración de la persona discapacitada. Para un mejor entendimiento de dicho sistema de movimiento de sillas de ruedas, se lo dividió en tres subsistemas: sistema mecánico, sistema de control y sistema de recarga y potencia de la silla de ruedas. A continuación, se detallan los subsistemas que están implementados:

1.6.1. SUBSISTEMA MECÁNICO:

- Este sistema podrá constar de engranajes o catalinas que reducen la velocidad del motor al ser colocado.
- Adicionalmente consta de una cadena que transmite la motricidad del motor a las ruedas.

1.6.2. SUBSISTEMA DE CONTROL:

- En este sistema están colocados los circuitos eléctricos que permiten controlar los motores de la silla de ruedas.
- Este sistema regula los pulsos de energía que pasan al motor según el mando de la silla de ruedas, regulando la velocidad de desplazamiento.
- No se usan elementos difíciles de encontrar en el mercado, ya que se pretende con esto reducir el costo utilizando elementos de alta disponibilidad en el mercado local.

1.6.3. SUBSISTEMA DE RECARGA Y POTENCIA:

- En este sistema está ubicado los circuitos de alimentación de la batería que alimenta a toda la silla de ruedas y los circuitos de protección de la misma.
- Se seleccionó el motor dc que se va a usar.
- Se analizó diferentes circuitos de carga para la batería y se seleccionó la más adecuada.

El mando de la silla de ruedas es primordial según el tipo de discapacidad que tenga el usuario, en el presente proyecto se lo colocó en el costado derecho del usuario y será tipo joystick.

Para abaratar costos se cotizó cada uno de los elementos de un circuito por separado y se comparó con un circuito equivalentes existentes en el mercado.

Adicional al proyecto se hacen recomendaciones en el mismo al Patrocinador para que vea tiempos de carga de la batería, protección del circuito y conservación de los sistemas motrices.

El costo total de la motorización de la silla de ruedas no superará el valor de 800 dólares para que esté al alcance de los beneficiarios y con un tiempo mínimo de instalación de 5 días.

Se diseñó e implementó una propuesta de sistema de control PI Adaptativo Difuso lo cual representa el componente investigativo del presente proyecto, el cual permite regular la velocidad de la silla de ruedas y se incorporó sensores, controles de velocidad y comandos a través de un panel de configuración y control para establecer posibles mejoras futuras y ofrecer un producto con mayor nivel tecnológico y manteniendo un costo relativamente bajo que se adapte a las necesidades de personas con mayor nivel de discapacidad.

1.7. Objetivos

1.7.1. General

Diseñar e implementar un sistema de movimiento para una silla de ruedas convencional mediante el uso de un controlador PI difuso adaptativo abaratando costos para personas con discapacidad física de escasos recursos económicos.

1.7.2. Específicos

- Investigar e implementar sobre el sistema de control para la movilización de una Silla de Ruedas Convencional con indicadores visuales basados en el control FUZZY-PI Adaptativo.
- Diseñar los circuitos eléctricos de potencia capaces de manejar un motor DC de gran torque, que permita garantizar el desplazamiento de la silla de ruedas.
- Implementar un circuito que genere la potencia necesaria para la recarga de la batería en el menor tiempo posible.
- Documentar todos los procedimientos necesarios para el diseño e implementación del prototipo mecánico y eléctrico.
- Realizar diferentes pruebas de funcionamiento del sistema para garantizar el correcto funcionamiento del mismo.

CAPÍTULO II

2. ANÁLISIS Y DISEÑO MECÁNICO

En el presente capítulo, se analizará las diferentes opciones que se tienen para realizar los movimientos mecánicos y a partir de aquí se diseñará el mecanismo apropiado para cada elemento modificado.

2.1. Funcionamiento De La Silla De Ruedas.

Ante todo se busca que la silla de ruedas motorizada en la presente tesis cumpla con las normas principales de tamaño, esto se observa en la figura 14. Consecuentemente las adaptaciones que se realice no sobrepasaran este tamaño, esto implica que los motores, controles y fuentes de energía de la silla de ruedas estén bien ubicados en lugares que no presenten riesgo al usuario y no sobresalgan del diseño original.

Para el funcionamiento de la silla de ruedas se debe tener bien especificado el espacio necesario para su maniobrabilidad, esto implica el espacio que la silla de ruedas tiene y lo que necesitaría para moverse libremente, como se le indica en la figura 14.

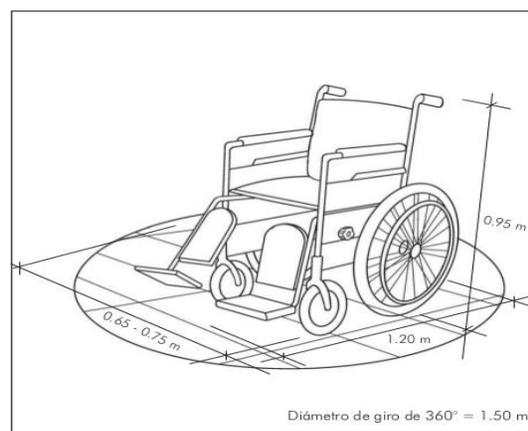


Figura 14. Dimensiones De La Silla De Ruedas

Fuente: Medidas Generales Para El Desplazamiento De Una Silla De Ruedas

Por lo que la silla de ruedas tiene un espacio libre mínimo de 15 cm por el frente antes de toparse con algún objeto o acercarse a unas gradas, en el proyecto actual por cuestión de seguridad y espacio físico en el asilo patrocinador se mantendrá este espacio.

En el caso de corredores menores a 1,50 metros de espacio, se recomendó al patrocinador que la silla de ruedas tenga una movilidad mínima (movimiento solo para adelante y atrás) para evitar que los sensores impidan algún movimiento. En el caso que se necesite ingresar a un corredor y dar alguna vuelta se necesitaría un asistente para que pueda maniobrarla sin dificultad ya que el movimiento para adelante se bloquearía por los sensores y solo serviría los movimientos para atrás y de los costados. Esto se da por que la pared estaría muy próxima y por programación su movilización hacia delante se bloquea.

El actual controlador está ubicado para ser manipulado por un usuario que tenga la movilidad de la mano derecha, pero el mismo se lo puede colocar en el costado izquierdo o inclusive en el manubrio de la silla de ruedas para que un asistente pueda movilizarla.

Por cuestiones de seguridad, como se aclaró antes, la presente silla de ruedas eléctrica no tiene ningún elemento móvil de fácil acceso al operario y se mantendrá los manubrios para que un acompañante le ayude en caso de requerirlo.

Con respecto al funcionamiento de las ruedas estas deben funcionar independientemente, por lo que se trabajó con dos motores para cada rueda para que se movilen al mismo tiempo o separadamente según el caso. Se hace aclaración que se usaran ruedas de bicicleta tipo montaña de radio de 30cm, con lo que se asegura

que sean resistente tanto para el uso como para el peso de la silla de ruedas, las mismas que tienen incorporado un tubo 1,5cm de diámetro que se usaran como borde sobresaliente de la rueda con la cual el usuario podría manipular las ruedas sin necesidad de ser impulsado por los motores en caso de ser necesario.

2.2. Análisis de Estabilidad.

La silla de ruedas donada por la fundación, como se aclaró anteriormente, tiene la característica principal que son ruedas de bicicleta tipo montaña lo que permitirá una mejor estabilidad al momento de movilizarse, esto permitirá que si existen obstáculos estos puedan ser superados.

Toda la silla de ruedas está asentada cerca del centro de las ruedas de mayor diámetro, lo que permite que por ahí pase el centro de gravedad haciéndole más estable pese al peso que se le imponga. Hay que aclarar que la silla de ruedas está diseñada solo para transportar a una persona sentada a la vez.

Por lo anteriormente mencionado y por seguridad del usuario se consideró que la mejor ubicación para aumentar los accesorios y transformarla en silla de ruedas eléctrica sería en la parte inferior de la silla, aquí se centraron los 2 motores D.C. los cuales transmiten su movimiento a las ruedas.

Por el mismo hecho de movilizar a una persona y el peso de la silla de ruedas, se seleccionó un motor de gran potencia para que maneje todo este peso de forma estable, en la actualidad en el mercado no existen motores DC de paso, ni Encoders que cumplan estas características, por lo que se tomó la decisión de usar un motor DC de alto torque y de un consumo de corriente relativamente aceptable.

Para el análisis e estabilidad también se observa el desempeño de los motores, como se dijo en el capítulo 1 se tomara diferentes conceptos para saber con que motor se trabajará.

Tabla 6:
Cuadro Comparativo de Motores

	RPM	Torque	Ruido	Amperaje
Motor de Arranque	1800	10 N*m	Excesivo	200
Motor Limpiaparabrisas	105	2.4 N*m	Normal	43

Por lo que se toma la decisión de trabajar con los motores Limpiaparabrisas.

Para que el motor tenga estabilidad para mover a una persona que pesa 100Kg se harán las conversiones para saber que tipo de motor nos serviría, se sumara el peso de la silla de ruedas 50kg, con lo que el peso total de la silla es de 150kg, como este peso pasa por el centro del eje de las ruedas de la silla, se tomará este valor hacia los extremos para saber la proporción en torque que se necesita para el motor dc.

Se tomará el valor de 0,36m como medida del centro del eje de las ruedas hacia las ruedas, y como se sabe el punto de apoyo para ejercer menos fuerza está determinado por los puntos extremos, por lo que se colocará este punto de apoyo a 0,34m de la rueda la cual ejercerá la fuerza de las ruedas, por lo que para mover 150kg con una distancia uno de 0,34m y las ruedas a una distancia de 0,02m

$$M = \frac{150Kg * 0,02}{0,34} = 8,82Kg$$

Para saber el torque que se tiene se aplica la fórmula:

$$mgh = \frac{1}{2}mV^2 + \frac{1}{2}M\omega R$$

$$m = \frac{\frac{1}{2}M\omega R}{gh - \frac{1}{2}V^2} = \frac{\frac{1}{2} * 8,82 * 2 * 0,3 * \pi * 0,3}{9,8 * 1,7 - \frac{1}{2}(1,7)^2} = 0,195Kg$$

$$T = W * d = 0,195 * 9,8 * 0,6 = 1,15N * m$$

El motor conseguido es de 2,4N*m, por lo que es más que suficiente para movilizar a una persona de 100Kg, este motor por características de placa consume una potencia de 270W a una corriente de 45A y un voltaje de 12V, también tiene un RPM de 1050, pero esta velocidad es del motor sin la caja reductora, las RPM a la salida de la caja reductora es de 105 esto se calculó por medio de un tacómetro manual Electrónico por lo que se deduce que tiene una caja reductora de 1:10. Este motor se lo Puede ver en la figura 15



Figura 15. Motor Limpiaparabrisas

2.3. Evaluación De Soluciones Mecánicas

Esta evaluación de soluciones mecánicas se trata de encontrar la mejor forma de transmitir el movimiento de los motores hacia las ruedas. Se verán tres formas de

transmitir este movimiento las cuales se pondrán en consideración: por bandas, catalinas o piñones.

Bandas. - Al usar bandas se corre el riesgo que estas se resbalen ya sea por el uso o por no estar bien templadas, adicionalmente estas serían muy difíciles de cambiar ya que para instalar alguna de estas se tendría que desarmar toda la silla para luego templarlas.

Piñones. - Con el uso de piñones se evita que se resbalen los accesorios de transmisión, pero se necesita muchos accesorios para mantenerlos fijos, como rodamientos, anillos, ejes, etc. Adicionalmente son muy rígidas y si se tiene algún accidente se puede dañar muy seriamente el mecanismo.

Con el uso de piñones se observa un aumento en el ruido de transmisión, por lo que se prefiere omitir este tipo de transmisión. Adicionalmente el costo de fabricación de piñones es muy alto y en caso de desgaste se tiene que reemplazar todo el mecanismo.

También se observa en autos que hay un excesivo ruido proveniente de los motores de arranque por lo que se toma la decisión de no trabajar con los motores DC tipo motor arranque de autos y se trabajar con motores DC tipo limpiadores de parabrisas de gran tamaño. Esto nos ayudará a reducir el ruido y como son de gran potencia movilizaran fácilmente la silla de ruedas.

Catalinas. – Al hacer el estudio de poner directamente catalinas de bicicleta de gran diámetro a los ejes de los motores, para igualar las catalinas que están ya sujetas a las ruedas de la silla, esto permitirá transmitir directamente la fuerza de los motores sin pérdida por transmisión. Como se usan cadenas para unir los dos juegos de catalinas se las puede cambiar fácilmente en caso de ser necesario sin desarmar toda la silla de

ruedas. Esto reduce el ruido de todo el mecanismo y evita en caso de accidentes que el mecanismo se destruya por completo.

Las catalinas son fáciles de conseguir, sin necesidad de construirlas como en el caso de los piñones. Se debe tener en cuenta el número de dientes de cada catalina para evitar que la una rueda gire más lenta o rápidamente que la otra.

Se tomó la decisión de soldar la catalina a un eje el cual estará sujeto al eje del motor como se observa en la figura 16 y 17. Esto se hará en ambas ruedas de la silla.



Figura 16. Eje de Catalina Asegurada al Eje del Motor



Figura 17. Catalina Soldada a un Eje

2.3.1. Selección de Materiales y Proceso de Fabricación.

Por las recomendaciones de Ing. Mecánicos se decidió que se fabriquen los ejes y bujes de anillo con el acero AISI 4340, esto se decidió por ser un material accesible en el mercado y que resistirá, sin problemas de deformación, la tensión generada por el peso de la silla de ruedas y del operario para movilizarlos.

Adicionalmente, por seguridad, como se dijo antes se usó cadenas de bicicletas de eslabón 3/32 para la transmisión que va a cada una de las ruedas de la silla, esto permitirá que si hay algún material que tensione a las ruedas no rompa las catalinas del eje y rompa primero la cadena, sin dañar estructuralmente a la silla de ruedas.

Para las catalinas que van en el motor se comenzó a usar un eje de 6,5cm de largo y 2,8cm de diámetro al cual se lo maquinará para que quede a las medidas requeridas. A este eje se lo modificará para que entre directamente en el rotor del motor DC, este eje de rotor viene de fábrica con un eje saliente del rotor que tiene un agujero de 1,6cm de diámetro y con una profundidad de 5cm. Al eje que se suelda se lo rebaja en el diámetro hasta que quede de 1,6cm y se lo trabaja en 5cm de su largo total, esto hace que entre en el rotor y se lo asegura con un prisionero de 1/8 que está colocado a un costado del eje del rotor, este prisionero servirá para sujetar el eje de la catalina y que este no se mueva de su sitio. Una vez introducido el eje, se lo señala en donde va a estar el prisionero para hacerle por medio de un taladro un pivote para que el prisionero pueda asegurar de mejor forma al eje de la catalina. El 1,5cm de largo y los 2,8 cm de diámetro que sobra del eje, nos servirán para poner un límite al eje y una base donde se va a asentar la catalina para soldarla como se observa en la figura 17. El eje y la catalina están dibujados en el apéndice 1 y 2.

Las catalinas que están en las ruedas de las sillas, se las sujeta a un buje con anillo, este buje tiene una rosca interna de medidas 1,29pulg con una rosca de 24TPI, esta rosca encaja en la rosca de un buje de bicicleta el cual es de 36 radios para sujetar la rueda. El anillo de buje sujeta a la catalina por medio de 5 pernos esto permite que la transmisión pase directamente a las ruedas, adicionalmente por motivos de seguridad se dieron puntos de suelda en la unión de ambos bujes para evitar que se desenrosque, esto se puede observar en la figura 18, este buje con anillo se dibuja en el apéndice 3.



Figura 18. Suelda Entre Bujes

El buje de bicicleta sujeta a la rueda de 60 cm de diámetro por medio de 36 radios, este tipo de buje es fácil de conseguir en el mercado se le llama buje con rosca de piñón fijo, esto permitirá que la rueda gire al mismo sentido que gire el motor, ya sea para adelante o para atrás. El motivo principal para seleccionar este buje es que es de carácter estándar, lo que significa que tiene una rosca única estandarizada (1,29 x 24TPI), la cual nos permitirá sujetarlo al buje de anillo. Se reemplazará el eje de ambos bujes por un solo eje largo que atraviesa toda la silla de costado a costado, esto nos permitirá centrar las ruedas en un solo eje, adicionalmente dará más fortaleza en sus extremos para soportar el peso que se impone en la silla. El buje de piñón fijo permite

que la rueda gire libremente en ambos sentidos, este buje se lo puede observar en la figura 19.



Figura 19. Buje Con Rosca De Piñón Fijo de 36 radios

El eje que sostiene ambos bujes de bicicleta es del mismo acero AISI 4340, que es el recomendado para no deformarse, este eje esta soldado a la estructura principal de la silla, como se ve en la figura 20 y corresponde al diseño del apéndice 4.



Figura 20. Eje Principal De Bujes

Por cuestión de diseño a los motores se los centralizó y sujeto con 2 placas de acero AISI 4340 en cada motor, como se puede ver en la figura 21, estas placas de acero están diseñadas en el apéndice 5 y 6; hay que recordar que estas placas podrían cambiar según el modelo de motor que se consiga.

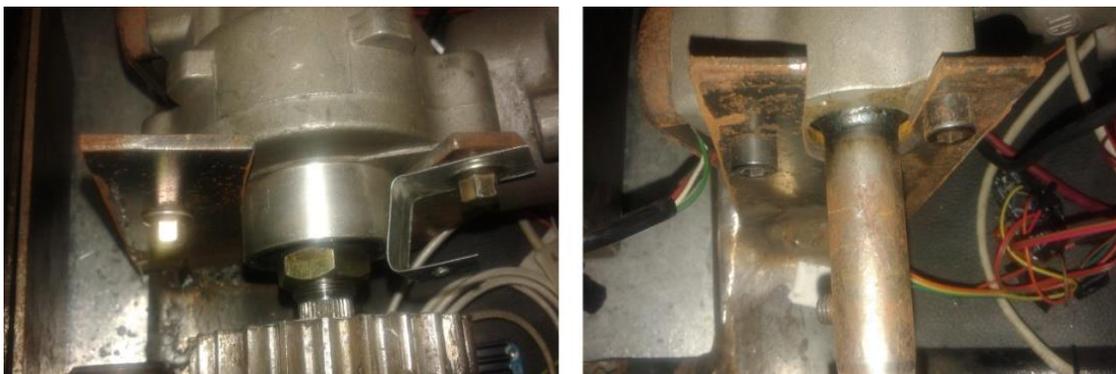


Figura 21. Bases De Motor

A su vez, estas placas, están soldadas a un perfil estructural “G”, el cual está también soldado a los dos extremos fijos de la estructura de la silla como se puede ver en la figura 22. El perfil estructural “G” corresponde a las medidas G 80 x 40 x 15 x 3 y de largo 465mm, se lo puede observar en el apéndice 7.



Figura 22. Suelda Del Perfil Estructural G

2.3.2. Diseño Mecánico.

Con las medidas representadas en planos de AUTOCAD de los apéndices del 1 al 7, se pudo diseñar un mecanismo capaz de mover la silla y que resista el esfuerzo sin deformarse o si fuera el caso sin dañar los componentes principales como las catalinas del motor o de las ruedas.

Adicionalmente, por cuestión de diseño mecánico, se observa que no se debe ejercer la fuerza de los motores directamente en las llantas, ya que esto a corto plazo lo deformaría al eje o desgastaría en forma excesiva los rodamientos internos de los motores, por lo que se usa 2 juegos de catalinas en cada rueda para hacer la transmisión de las fuerzas correspondientes.

Para el diseño de las catalinas, como se seleccionaron en el mercado, se buscaron las de mayor diámetro para transmitir el movimiento, hay que recordar que estas catalinas se necesitan que vengan en pares ya sea para los ejes de los motores o para las ruedas de la silla, esto se da por que deben coincidir en la cantidad de dientes que tienen cada una, si en una se observa una menor cantidad de dientes, esto repercutiría en la transmisión, haciéndole más lento o rápido al movimiento, según sea el caso.

CAPÍTULO III

3. DISEÑO DEL SISTEMA DE CONTROL

En este capítulo no solo se observará el diseño de control de los motores, también se observará la energización a la silla de ruedas y la forma en que se escogió a los sensores.

3.1. Sistema de Alimentación.

Como se trabaja con motores de 12V en corriente directa, se necesita dar energía de este mismo valor y con suficiente corriente que alimente no solo a los 2 motores de gran torque, si no también alimente a los sistemas de control, por lo que se necesita adicionar una corriente de consumo de 3A a esta batería.

Con el consumo de 2 motores de 270W y de los circuitos de control se procede a calcular la energía necesaria para que funcione de manera continua por 2 horas. Se tiene un consumo total de 555W y como se va a trabajar con 12V se tiene una corriente de 46,25A. En el mercado actual existen diferentes tipos de baterías, que dan diferentes valores de energía.

Se decide trabajar con una batería que se ocupa generalmente en motocicletas, modelo "YTX16-BS", la cual por su tamaño es de fácil instalación en la parte inferior de la silla y por sus características de almacenamiento de energía no se descarga la corriente con tanta velocidad, esta batería nos da una carga eléctrica de 14,7Ah, en términos sencillos esto equivale a que se podrá usar la batería con su máxima carga por unas 2 horas continuas. La batería seleccionada se puede ver en la figura 23.



Figura 23. Batería De Motocicleta

Las dimensiones de la batería usada son: 14 x 8 x 16cm. Las conexiones se las realiza por medio de borneras de presión ubicadas en los polos de la batería, que restringen el deslizamiento de los cables de conexión.

Para cargar este tipo de baterías se incorpora en la silla una fuente de corriente continua de gran amperaje, este amperaje debe ser mayor al proporcionado por la batería para que al momento de recargar la batería nos tome menos tiempo en conseguir la carga máxima, si bien es cierto que este tipo de fuentes se las puede construir, se prefiere comprar una que se pueda encontrar fácilmente en el mercado para ahorrar tiempo, es así que se selecciona una fuente AC-DC que transforma 110V o 220V de corriente alterna a 12V de corriente continua con un amperaje de 20A, esta fuente se la puede observar en la figura 24.



Figura 24. Fuente de 20A y 12V

Se recuerda que la fuente de alimentación AC-DC tiene sus propias características de seguridad para evitar sobrecargas lo que nos permite alimentar la batería sin riesgo, una vez visto esto se pone atención a la seguridad que se debe tener en la batería al alimentar los motores DC y los elementos electrónicos, por lo que se le instalará un breaker que limite la carga de corriente proporcionada. Se mide la corriente que necesita el motor DC en vacío y se consigue un valor de 10A en DC, por información proporcionada por el mismo motor se sabe que la corriente máxima del motor es de 22A y como se trata de alimentar los 2 motores y diferentes elementos electrónicos que consumen hasta 3A se decide limitar la corriente con un breaker de máximo 40A en DC, para evitar sobrecargas que dañen los equipos pero sin evitar que la silla de ruedas se movilice fácilmente, este breaker estará conectado a una barra de cobre tipo bornera en donde se conectarán todos los cables de alimentación y fuentes DC-DC.

Una vez seleccionada la fuente para los motores se tiene que ver como se alimenta el sistema de control, hay que recordar que este sistema de control está basado en Arduino para facilitar su programación y su entorno de hardware, por lo que se necesita bajar el voltaje de los 12V de batería a 5V, esto se logra con un "STEP DOWN" el cual puede dar hasta 3 amperios de corriente, esta fuente alimentará un arduino nano, un arduino mega, 2 sensores de proximidad, 1 acelerómetro ADXL335 y 2 puentes H. El Step Down se le puede ver en la figura 25.



Figura 25. Step Down 2596S

Este convertidor DC-DC permite regular el voltaje desde 3.2 a 35V de entrada a un voltaje de salida de 1.25 a 30V, por medio del potenciómetro de precisión, este convertidor da como corriente 3A. Se lo asegurará directamente en la silla por medio de tiras de madera. Este Convertidor tiene fusibles que permiten proteger los elementos electrónicos en el caso de tener sobrecargas.

3.2. Análisis y Selección de Dispositivo de Control.

Para comenzar el análisis y selección de los dispositivos de control se debe de recordar que se maneja un gran peso por lo que los motores son de gran torque, como se ha dicho anteriormente, esto equivale a que los dispositivos de control deben manejar una gran potencia. Adicionalmente se hace un análisis para que los elementos de control y sensores sean ubicados en un lugar que no sea de fácil acceso para el usuario.

3.2.1. Actuadores

Como se ha hablado antes se vio dos tipos de actuadores:

- MOTORES DE ARRANQUE: En un inicio se estudió este tipo de motores para movilizar la silla de ruedas por ser de fácil acceso en el mercado actual, pero los mismos generaban demasiado ruido y consumían una corriente muy alta, por lo que al implementar un dispositivo de control estos podían quemarse por la sobrecarga que ejercían los motores al momento de arranque. También se observó que estos motores producían un ruido innecesario y elevado, con lo que se tomó la decisión de no usar este tipo de motor. Se lo puede ver a este tipo de motor en la figura 26.



Figura 26. Motor De Arranque

- MOTORES DC: Con este tipo de motores se debe tener cuidado ya que son escasos en el mercado actual ya que manejan un gran torque y por este motivo no se manejan motores de estas características, sin embargo, se encontraron motores DC que se usan para limpiaparabrisas de camiones, son de la marca CHILL, se alimentan con 12v y tienen un torque de 2.40 N*m, como se lo ha dicho anteriormente, se lo ve en la figura 27.



Figura 27. Motor DC

Como también se puede observar este motor trabaja con 45A y a 1050RPM, pero ese sería el caso de la máxima velocidad, como se va a trabajar con velocidades

inferiores su amperaje disminuirá, este dato nos servirá mucho para calcular que elementos de pre actuadores que se necesitará posteriormente.

Hay que recordar que las 1050RPM son del motor sin la caja reductora, gracias al tacómetro manual electrónico, se sabe que a la salida de la caja reductora se tiene 105rpm lo que equivale tener una reducción de 1:10, por lo que la silla de ruedas, al darle toda la energía a los motores, alcanzaría una velocidad aproximada de 3,29m/s que es la velocidad máxima que alcanzaría con las llantas de bicicletas colocadas actualmente las cuales tienen un radio de 0,3m. No se necesita alcanzar esta velocidad ya que la velocidad promedio de ser humano al caminar es de 1,7m/s esto implica que hay que reducir la velocidad del motor.

Para saber cuántas RPM necesita la silla para alcanzar la velocidad del ser humano hay que hacer una tabla de conversión:

$$1,7 \frac{m}{s} \left| \frac{60s}{1 \text{ min}} \right| \frac{1 \text{ rev}}{2 * 0.3m * \pi} = 54RPM$$

Estos 54RPM son de la caja reductora, por lo que hay que convertirlo con la escala, por lo que el motor necesita:

$$54RPM * \frac{10}{1} = 540RPM$$

Para que el motor genere 540RPM el programa de arduino mega tiene que mandar una señal PWM de 140, se llegó a esta señal por medio de varias pruebas y simulaciones que se expondrán próximamente en este mismo capítulo. Esta señal es la ideal para alcanzar la velocidad del caminar del ser humano, por lo que esta será la señal referencial para a partir de aquí implementar otras 2 señales adicionales con las cuales la silla de ruedas tendrá 3 tipos de velocidades en total.

3.2.2. Controladores

Los controladores se los hacen con los Arduino Nano y el Arduino Mega:

- ARDUINO MEGA.- Se escogió en un inicio este arduino por tener varias señales tanto digitales como analógicas, esto se vio por la necesidad de un no saber con que tipo de Pre actuador se iba a trabajar; si se trabajaba con mosfet de gran potencia estos necesitarían señales diferentes para poder realizar los diferentes movimientos de la silla de ruedas. Como se decidió trabajar con un Motor DC de 45A se consultara en las herramientas de hardware de Arduino si hay algún elemento que pueda trabajar con esta potencia. Al Arduino Mega se lo puede ver en la Figura 28.

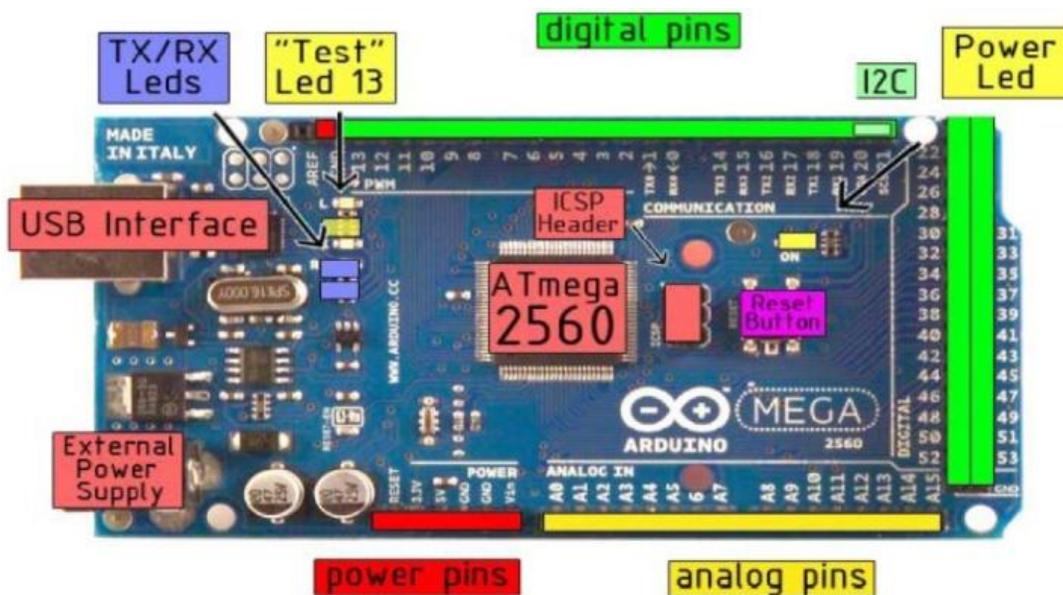


Figura 28: Arduino Mega 2560

Fuente: Diagrama Arduino Mega

- ARDUINO NANO.- Se escogió este arduino solo para trabajar con los sensores, puesto que no se quería saturar la programación del Arduino Mega y provocar retrasos en las respuestas. Se lo puede ver en la Figura 29.

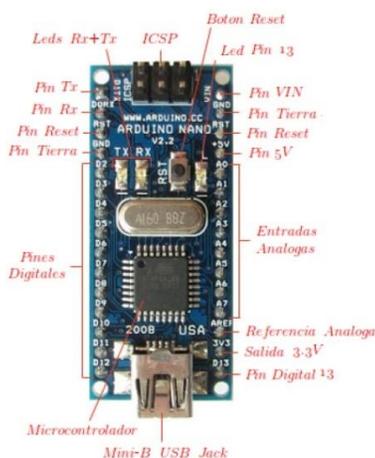


Figura 29. Arduino Nano V3

Fuente: Diagrama Arduino Nano

3.2.3. Pre-Actuadores

Con el dato del Motor que consume 45A se decide trabajar con los denominados Puente H que mediante una señal enviada por el arduino mega nos sirven para poder coordinar las acciones de los motores. Como se habló anteriormente son 2 motores de gran torque, lo que significa que son de gran potencia, por lo que se va a trabajar con puentes H que manejen un gran amperaje. Se va a trabajar con los Puentes H modelo “BTS7960B” de 43A también llamado Módulo de Controlador Doble. Se le puede observar en la figura 30.



Figura 30. Puente H Modelo “BTS7960B”

Se trabajará con 2 Puentes H ya que cada uno manejará un motor, este puente H nos permite cambiar el sentido de giro de los motores DC por tener 2 controladores

internos. Como es una herramienta de arduino los comandos que guían los controladores se manejan con 5V. Cada uno de los pines del puente H tiene su función específica: 1 y 2 es para habilitar el sentido de giro hacia la derecha o izquierda del motor y su velocidad depende de la señal PWM que lance el programa del arduino mega; 3 y 4 habilitan los dos controladores del Puente H para poder cambiar de giro según la necesidad ya sea derecha o izquierda hay que decir que estos solo se los habilitará y permanecerán con VCC; 5 y 6 se usan para sensores de voltaje y corriente que en este caso no se usaran; 7 es para la entrada VCC de 5V; 8 es para GND. Se lo puede ver en la figura 31.

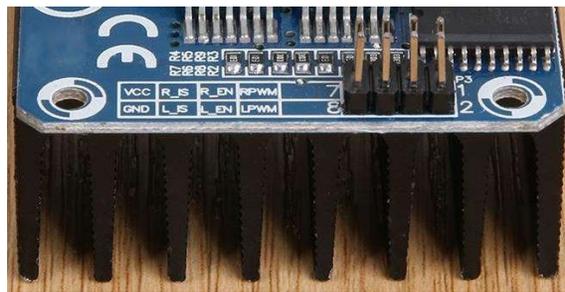


Figura 31. Pines De Conexión Puente H

La programación para manejar los 2 puentes H está en el Arduino Mega que se observará en la sección Control PI-Difuso Adaptativo. Se usan los pines digitales 23 y 25 para habilitar los 2 controladores del puente H y estarán conectados a los pines 3 y 4; los pines 9 y 10 indican el sentido de giro del motor y la velocidad en la que se requiere que este, estos pines parten de los conectores PWM del arduino mega y se los conecta a los pines 1 y 2 del puente H. Lo mismo sucede con el otro Puente H al habilitar los pines 40 y 41 del arduino mega que van conectados a los pines 3 y 4 del puente H y los pines 11 y 12 del arduino mega daríamos el sentido de giro y la

velocidad del motor por poder dar una señal PWM y están conectados a los pines 1 y 2 del puente H.

3.2.4. Sensores

Casi todos los sensores mandan los datos al arduino nano, el cual recibe toda la información y mediante su programación envía una señal al arduino mega para bloquear el movimiento hacia adelante, como se recordará el resto de movimientos si se los podrá realizar. Se hizo de esta forma para evitar sobresaturar la programación del arduino mega con la recepción de los datos analógicos enviados por los sensores y así evitar retrasos en la respuesta del mismo. El único sensor que manda los datos al arduino mega es el sensor que mide las RPM de los motores, mediante el cual se podrá decir si un motor debe de ser más rápido o más lento para llegar a la velocidad deseada.

- **SENSORES DE PROXIMIDAD:** Se coloca 2 tipos de estos sensores; los cuales, como se ve en el mercado actual, pueden llegar a medir grandes distancias, sin embargo se limitaran a medir distancias no mayores a 20cm, en el caso del sensor ubicado para detectar paredes, y no mayores a 18cm en el caso del sensor que detecta si hay gradas o rampas con pendientes negativas mayores a -15° , estos 2 sensores se los coloca desde el punto más extremo de la silla en la parte frontal, que en este caso se trata del apoya pies, se los colocará en la parte inferior del mismo. Los sensores de proximidad que se escogió son los sensores infrarrojos modelo "2Y0A21", se lo puede ver en la figura 32, este tipo de sensor llega a medir hasta 80cm, se escogió este tipo de sensores por su fácil configuración en la programación puesto que dan una señal analógica que ingresa directamente al arduino nano por medio de los pines A2 y A6.

Son de fácil instalación y si se necesita reemplazarlos se los puede conseguir fácilmente en el mercado.



Figura 32. Sensor De Proximidad “2Y0A21”

El primer sensor está regulado para medir la proximidad de una pared ubicada al frente de la silla, este medirá si existe algún obstáculo a una distancia no mayor de 20cm al frente de la silla y a una altura mayor o igual a 11cm, su codificación y compilación se realizó en Arduino y se la puede ver en la figura 33:

```

sketch_jan14a Arduino 1.8.5
Archivo Editar Programa Herramientas Ayuda
sketch_jan14a$
void loop()
{
  valor = digitalRead(Boton1);
  int D1_cm=distanciaA6(20); //lectura de distancia para la entrada para
  int D2_cm=distanciaA2(20); //lectura de distancia para la entrada para
  delay(100);
  if((D1_cm>10 && D1_cm<19)|| (D2_cm>18 && D2_cm<40)|| (Valorsensor3>320)
  {
    bandera=1;
  }
  else
  {
    bandera=0;
  }
  if((millis()-Tiempo_TX)>500)
  {
    Serial.println(bandera);
    Tiempo_TX=millis();
  }
}

float distanciaA6(int n) //programa para el sensor detector de paredes
{
  long suma=0;
  for(int i=0;i<n;i++)
  {
    suma=suma+analogRead(A6); //lectura del sensor infrarrojo al frente
  }
  float adc=suma/n;
  float distancia_cm = 17569.7 * pow(adc, -1.2062);
  return (distancia_cm);
}

```

Figura 33. Compilación Programa Sensor De Pared

Como se ve en la figura 33 se designa el Pin 6 del arduino nano para el ingreso de datos del sensor infrarrojo.

El segundo sensor está regulado para detectar si existen gradas al frente de la silla, por normas de la institución patrocinadora, todas las gradas miden normas estándar (18cm de alto y 25cm de largo), para los cálculos necesarios se tomará la altura del apoya pies que está a 11cm y el largo no superará los 22cm para evitar llegar al filo de las gradas, con estos datos se logra un ángulo de $37,19^\circ$, como se puede ver en la figura 34.

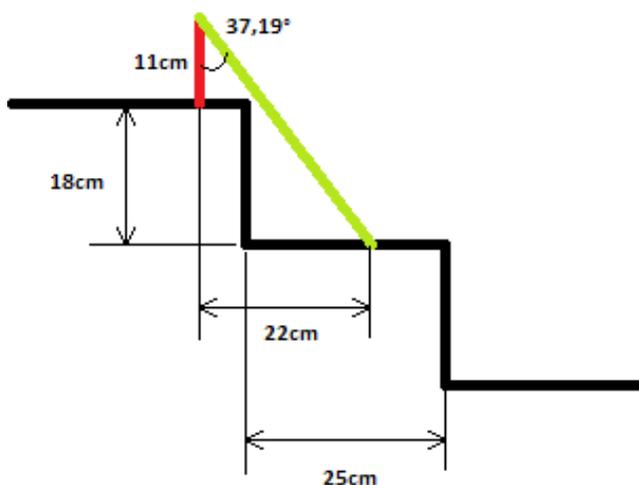


Figura 34. Parámetros Del Sensor

Esto nos generará una hipotenusa de 36,4cm, esta es la medida extrema que el sensor de la silla medirá cuando llegue al filo de unas gradas, pero como se advierte esta medida es el caso extremo. La verdadera distancia máxima que el sensor debe detectar y de la cual no debe de salir, es la que se forma al momento de que la silla salga o entre de una rampa, gracias a que la institución patrocinadora está normalizada, todas las rampas tienen una pendiente positiva de 15° , esto formará un polígono, como se ve en la figura 35.

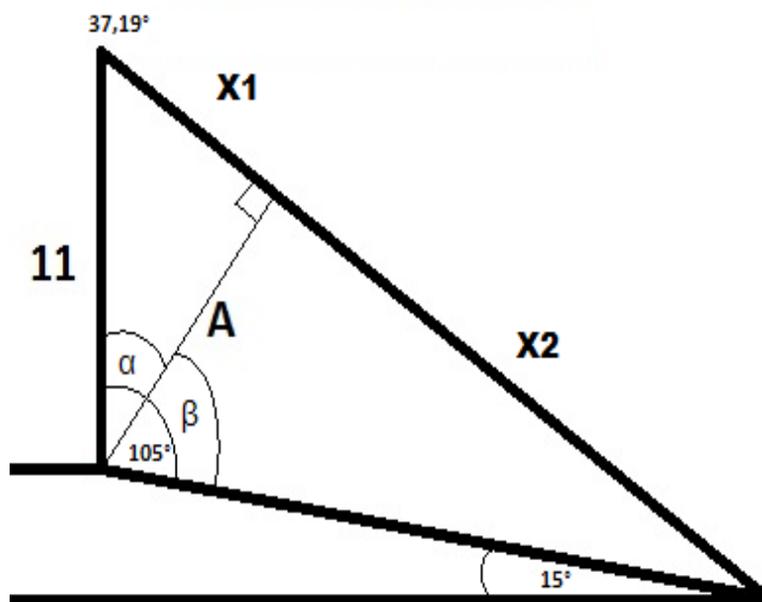


Figura 35. Triángulo Formado Al Ingresar En Una Rampa

Se forma un ángulo de 105° con la rampa, por la suma de 15° y los 90° . Con estos datos se encuentran los ángulos δ y θ y principalmente se encuentra el lado opuesto al Angulo de 105° que es la distancia máxima que debe llegar a medir el sensor:

$$\cos(\delta) = \frac{X_1}{11} \cong X_1 = 11 * \cos(37,19) = 8,764cm$$

$$\text{sen}(\delta) = A/11 \cong A = 11 * \text{sen}(37,19) = 6,648cm$$

$$180 - 105 - 37,19 = 37,81^\circ$$

$$\tan(\theta) = A/X_2 \cong X_2 = A/\tan(37,81) = 8,566cm$$

$$X_1 + X_2 = 8,566 + 8,764 = 17,33cm$$

Este dato sirve para la programación del segundo sensor, es decir que detecta si existe piso hasta 17,33cm al frente del sensor de la silla con una inclinación de $37,19^\circ$ partiendo de la vertical del apoya pies. Su compilación en el programador de Arduino se podrá ver en la figura 36:

```

sketch_jan14a Arduino 1.8.5
Archivo Editar Programa Herramientas Ayuda
sketch_jan14a $
void loop()
{
  valor = digitalRead(Boton1);
  int D1_cm=distanciaA6(20); //lectura de distancia para la entrada para
  int D2_cm=distanciaA2(20); //lectura de distancia para la entrada para
  delay(100);
  if((D1_cm>10 && D1_cm<19) || (D2_cm>18 && D2_cm<40) || (Valorsensor3>320
  {
    bandera=1;
  }
  else
  {
    bandera=0;
  }
  if((millis()-Tiempo_TX)>500)
  {
    Serial.println(bandera);
    Tiempo_TX=millis();
  }
}
float distanciaA2(int n) //programa para el sensor detector de gradas o
{
  long suma=0;
  for(int i=0;i<n;i++)
  {
    suma=suma+analogRead(A2); //lectura del sensor infrarrojo inclinado
  }
  float adc=suma/n;
  float distancia_cm = 17569.7 * pow(adc, -1.2062);
  return(distancia_cm);
}
41 Arduino/Genuino Mega or Mega 2560, ATmega2560 (Mega 2560) en COM5

```

Figura 36. Compilación Programa Sensor De Gradass

Aquí también se observa que se va a usar el Pin 2 del arduino nano para el ingreso de datos del sensor infrarrojo que detecta gradass o pendientes de inclinación negativa.

Esto sirve tanto si entra a una rampa o si sale de la misma. Hay que señalar que en la Institución Patrocinadora no existen gradass ni rampas fuera de las normas ya establecidas. Si la silla baja por una rampa superior a la establecida, esta se bloqueará antes de ingresar porque la pendiente calculada sería superior, esto sucederá a los 8,35cm al frente de la silla por ser la distancia mínima del triángulo que se forma al entrar en la rampa, como se puede observar en la figura 37.

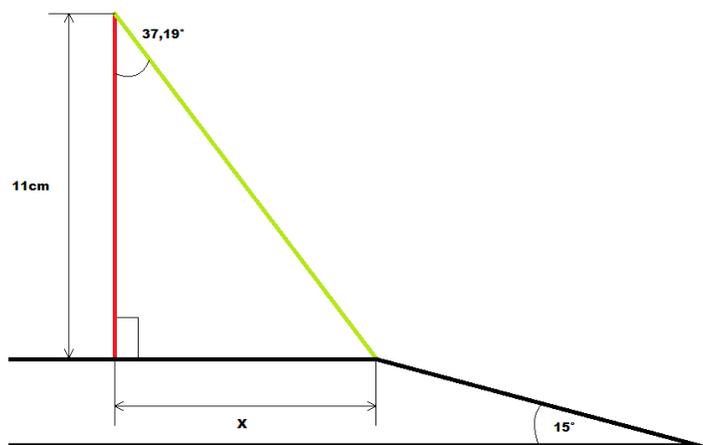


Figura 37. Triángulo Formado Al Bajar La Rampa

El análisis y pruebas de estos sensores se podrá observar en el siguiente capítulo 4.

- SENSOR DE INCLINACIÓN: Como los sensores anteriores aquí también existen varios modelos en el mercado actual que pueden medir un ángulo de inclinación, pero se eligió uno que sea de fácil configuración en su programación y que dé una señal analógica para que lo recepte el arduino nano por medio del pin A1. Se lo coloca en la base de los motores por estar nivelado y así tener la mejor referencia del ángulo a ser medido. El modelo que se escogió es el Sensor Giroscopio Analógico “GY-61”, este medirá un ángulo de 15° de inclinación positiva, esto se lo hace por normalización ya establecida por el patrocinador, como se habló anteriormente, se lo puede observar en la figura 38. Son de fácil instalación y si se necesita reemplazarlos se los puede conseguir fácilmente en el mercado.

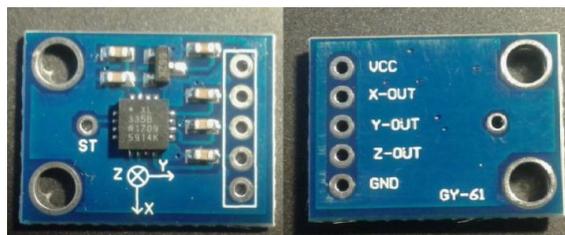
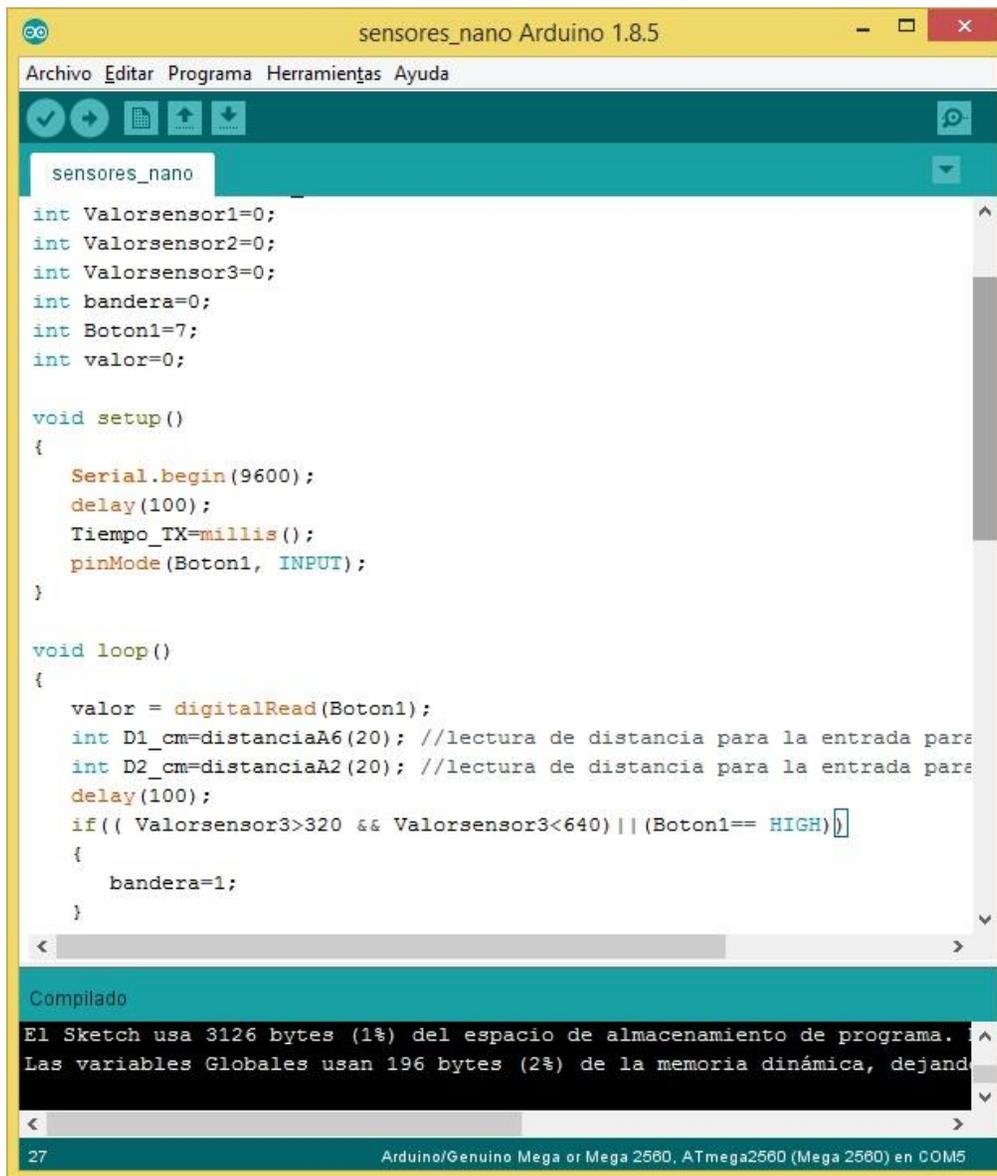


Figura 38. Giroscopio Analógico GY-61

Este sensor mide ángulos de inclinación en los ejes X, Y y Z; para este caso medirá la inclinación en el eje X cuando la silla ya se encuentre en la rampa, si sufre inclinaciones negativas este sensor no bloquea el movimiento hacia adelante, lo bloquearía el sensor de proximidad según las medidas ya establecidas anteriormente. Su compilación en el programador Arduino se ve en la figura 39:



```
sensores_nano Arduino 1.8.5
Archivo Editar Programa Herramientas Ayuda
sensores_nano
int Valorsensor1=0;
int Valorsensor2=0;
int Valorsensor3=0;
int bandera=0;
int Boton1=7;
int valor=0;

void setup()
{
  Serial.begin(9600);
  delay(100);
  Tiempo_TX=millis();
  pinMode(Boton1, INPUT);
}

void loop()
{
  valor = digitalRead(Boton1);
  int D1_cm=distanciaA6(20); //lectura de distancia para la entrada para
  int D2_cm=distanciaA2(20); //lectura de distancia para la entrada para
  delay(100);
  if(( Valorsensor3>320 && Valorsensor3<640) || (Boton1== HIGH))
  {
    bandera=1;
  }
}
```

Compilado

El Sketch usa 3126 bytes (1%) del espacio de almacenamiento de programa.
Las variables Globales usan 196 bytes (2%) de la memoria dinámica, dejand

27 Arduino/Genuino Mega or Mega 2560, ATmega2560 (Mega 2560) en COM5

Figura 39. Compilación Programa Sensor De Inclinación Pendiente Positiva

Mediante esta programación se logra comparar la medida de 15° con pendiente positiva con un valor de 320 que registra el arduino nano en la comunicación serial, por lo que si excede de este valor lanzará la interrupción al arduino mega y bloqueará su movimiento. En este caso el sensor de inclinación se hace por la entrada del Pin 1 del arduino nano para el ingreso de datos analógicos, el resto del programa estará en la sección de este capítulo que habla del Control PI-Difuso Adaptativo y el análisis de las pruebas de este sensor se podrá ver en el capítulo siguiente.

- **SENSORES DE RPM:** Se trabaja con sensores de línea adaptado para medir los engranajes de salida de la caja reductora del motor DC. Este tipo de sensor entra por los pines de PWM 0 y 3 del arduino mega, se lo puede ver en la figura 40.

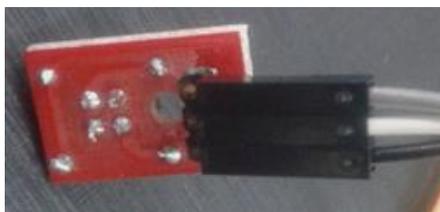


Figura 40. Seguidor De Línea

3.3. Control PI Difuso Adaptativo.

Este es el sistema de control implementado en el presente proyecto. El programa principal que contendrá el sistema de control PI-Difuso está programado en el Arduino Mega. Se generarán 4 movimientos que siempre parten del reposo, pero solo el movimiento hacia adelante implementará el sistema PI-Difuso Adaptativo.

Para comenzar el movimiento hacia adelante se debe recordar que la velocidad que se necesita como máxima es la de 1,7m/s que estarán generados por un PWM de 140 a partir de aquí se toman los datos para generar los límites para hacer el sistema de control difuso. Luego se implementará el sistema PI mediante el error generado y por

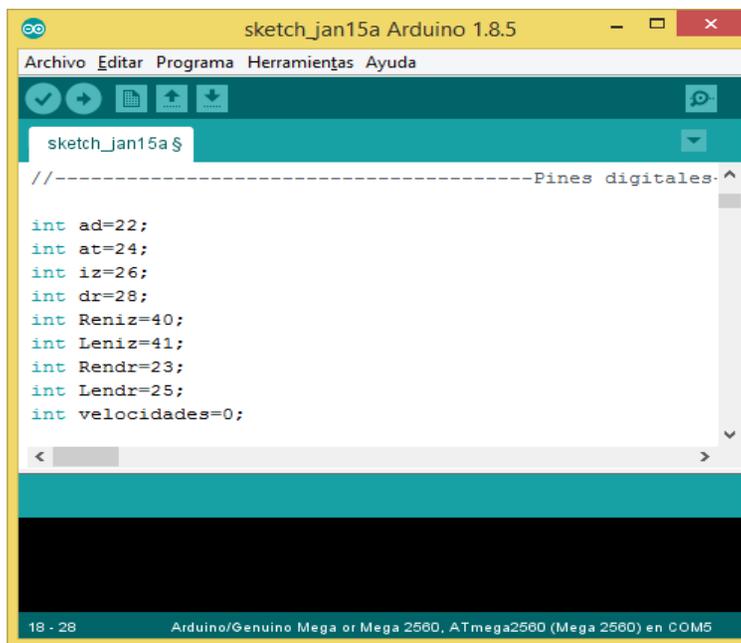
último se realizará el sistema de control Adaptativo mediante los sensores comandados por el controlador Arduino Nano.

Al momento de desplazarse hacia adelante se lo podrá realizar con 3 velocidades las cuales cumplen con el controlador PI-Difuso Adaptativo:

- 1.- La mínima velocidad en el que puede avanzar la silla.
- 2.- Una velocidad intermedio en la que la silla avance más rápidamente, esta velocidad es la promedio entre la velocidad máxima y la velocidad mínima.
- 3.- La velocidad máxima, es decir la velocidad de 1,7m/s. Esta velocidad se eligió como máxima ya que no se necesita mayor desplazamiento para la silla de ruedas.

Para la movilización hacia atrás, para el giro a la izquierda y hacia la derecha no se tomará el control PI-Difuso solo se tomarán pulsaciones fijas dadas.

Como se trabaja con un joystick, se comenzará con entradas digitales que se declaran de la siguiente forma:



```

sketch_jan15a Arduino 1.8.5
Archivo Editar Programa Herramientas Ayuda
sketch_jan15a $
//-----Pines digitales-----^
int ad=22;
int at=24;
int iz=26;
int dr=28;
int Reniz=40;
int Leniz=41;
int Rendr=23;
int Lendr=25;
int velocidades=0;
18 - 28 Arduino/Genuino Mega or Mega 2560, ATmega2560 (Mega 2560) en COM5

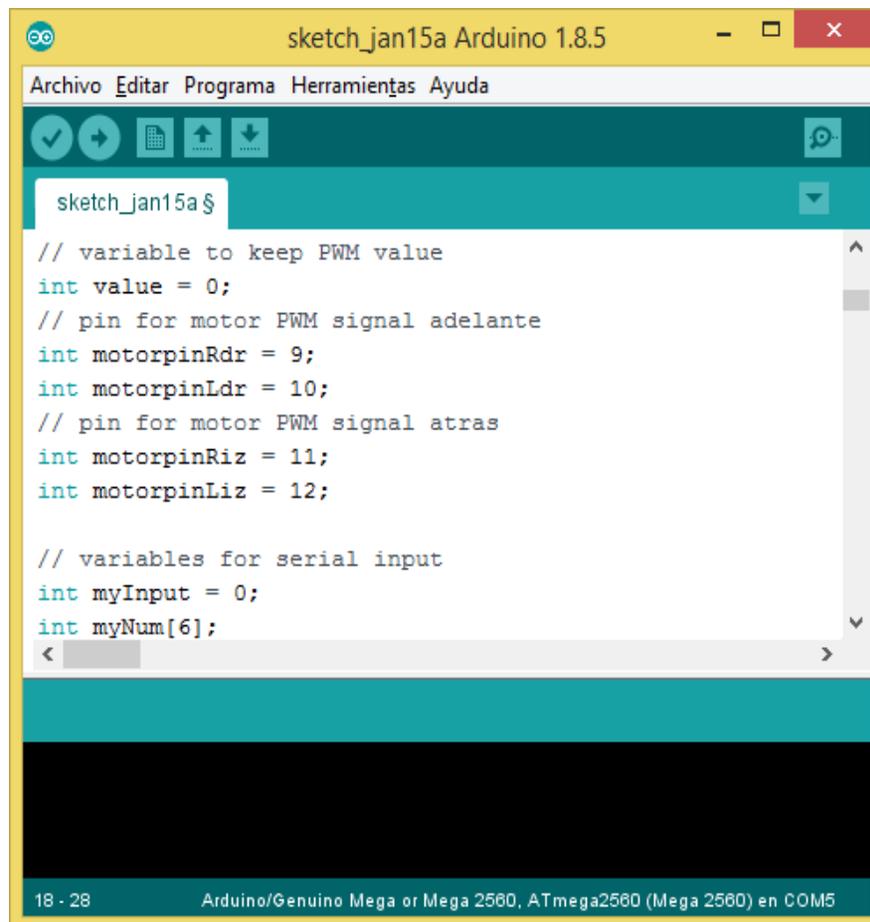
```

Figura 41. Programador Arduino Mega Pines Digitales

Para empezar con los movimientos de los motores también se trabaja con salidas digitales que se mandan a los controladores de los puentes H, también se las observa en la figura 41.

Sin estos controladores no se activaría ningún motor, para tenerlos activados se los debe mandar una señal en alto.

Para poder controlar la velocidad de un motor y si este se mueve hacia atrás o hacia adelante se necesita declarar los pines PWM del arduino mega con los que se va a trabajar, esto se observa en la figura 42:



```
sketch_jan15a $  
  
// variable to keep PWM value  
int value = 0;  
  
// pin for motor PWM signal adelante  
int motorpinRdr = 9;  
int motorpinLdr = 10;  
  
// pin for motor PWM signal atras  
int motorpinRiz = 11;  
int motorpinLiz = 12;  
  
// variables for serial input  
int myInput = 0;  
int myNum[6];  
< >
```

18 - 28 Arduino/Genuino Mega or Mega 2560, ATmega2560 (Mega 2560) en COM5

Figura 42. Programador Arduino Mega, Pines PWM

Para controlar la velocidad hacia adelante se lo hace por medio de 2 botones, uno sube la velocidad y el otro lo baja, la visualización de estas 3 velocidades se lo representará por medio de un display de 7 segmentos se lo declara en la figura 43:

```

Arduino_Mega Arduino 1.8.5
Archivo Editar Programa Herramientas Ayuda
Arduino_Mega$
//-----botones velocidad-----
//int LED = 13;
int Boton1 = 7;
int Boton2=8;
int bandera=0;
int valor = 0;
int contador = 0;
int decremento=0;
int estadoanteriorboton = 0;
int estadoanteriorboton1 = 0;
int pboton=0;
int pboton2=0;
//display de siete segmentos
int A=33;
int B=32;
int C=31;
int D=30;
Guardado.
The sketch name had to be modified. Sketch names can only consist
of ASCII characters and numbers and be less than 64 characters long.
17 Arduino/Genuino Mega or Mega 2560, ATmega2560 (Mega 2560) en COM5

```

Figura 43. Programador Arduino Mega Pines De Velocidad

Hay que aclarar que para el incremento o decremento de la velocidad se da por una suma o resta de 1 consecutiva según el botón que presione, esto se puede ver en la figura 44:

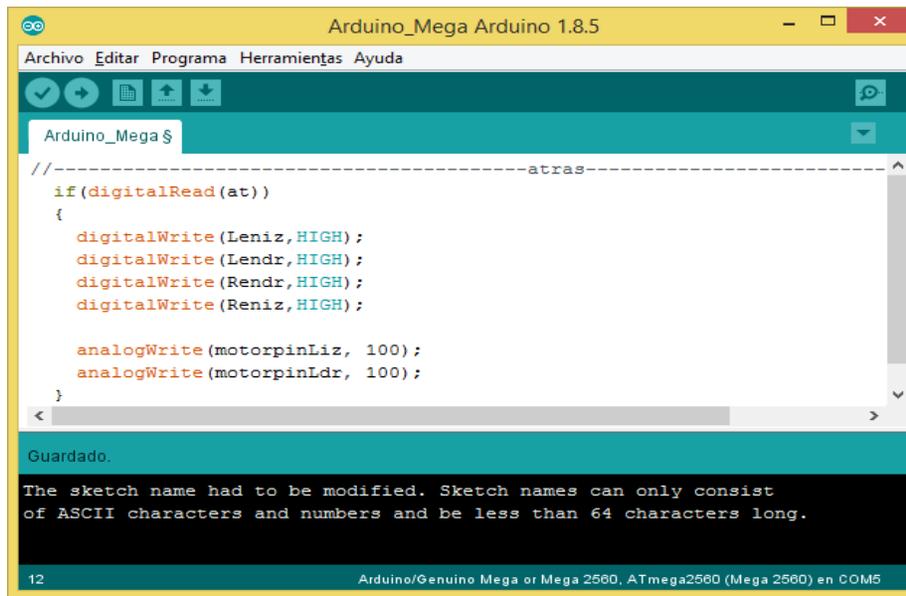
```

Arduino_Mega Arduino 1.8.5
Archivo Editar Programa Herramientas Ayuda
Arduino_Mega $
void botonsubir()
{
  contador=decremento;
  valor = digitalRead(Boton1); // lee el valor de la entrada digital pin 7
  if(digitalRead(Boton1)==LOW)
  {
    pboton=1;
  }
  if(valor != estadoanteriorboton)
  {
    if(valor == 1 && pboton==1 )
    {
      if (contador < 3)
      {
        contador++;
        sietesegmentos(contador);
      }
    }
    pboton=0;
  }
  estadoanteriorboton = valor;
}
void botonbajar()
{
  decremento=contador;
  bandera = digitalRead(Boton2); // lee el valor de la entrada digital pin
  //digitalWrite(LED, valor);
  if(digitalRead(Boton2)==LOW)
  {
    pboton2=1;
  }
  if(bandera != estadoanteriorboton1)
  {
    if(bandera == 1 && pboton2==1)
    {
      if (decremento > 0)
      {
        decremento --;
        sietesegmentos(decremento);
      }
    }
    pboton2=0;
  }
  estadoanteriorboton1 = bandera;
}
}
Guardado.
The sketch name had to be modified. Sketch names can only consist
of ASCII characters and numbers and be less than 64 characters long.
48 Arduino/Genuino Mega or Mega 2560, ATmega2560 (Mega 2560) en COM5

```

Figura 44. Incremento Y Decremento de La Velocidad

Para realizar el movimiento hacia atrás se toma un PWM fijo de 100 y estaría en la figura 45.



```

Arduino_Mega Arduino 1.8.5
Archivo Editar Programa Herramientas Ayuda
Arduino_Mega $
//-----atras-----
if(digitalRead(at))
{
  digitalWrite(Leniz,HIGH);
  digitalWrite(Lendr,HIGH);
  digitalWrite(Rendr,HIGH);
  digitalWrite(Reniz,HIGH);

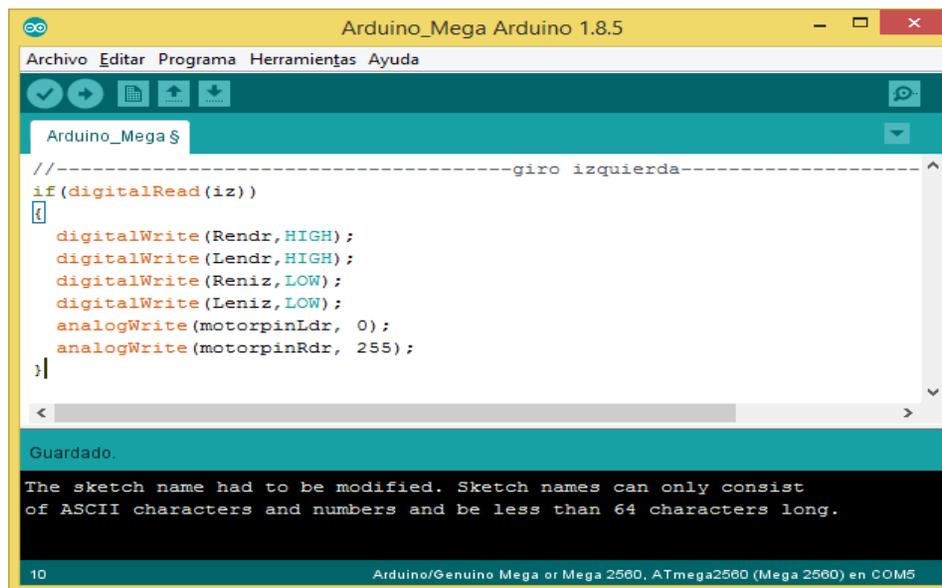
  analogWrite(motorpinLiz, 100);
  analogWrite(motorpinLdr, 100);
}
Guardado.
The sketch name had to be modified. Sketch names can only consist
of ASCII characters and numbers and be less than 64 characters long.
12 Arduino/Genuino Mega or Mega 2560, ATmega2560 (Mega 2560) en COM5

```

Figura 45. Programa Arduino Mega Movimiento Para Atrás

Están activadas los 2 controladores de cada puente H y los motores se activan los Liz y el Ldr con una señal PWM de 100, esta acción hace que motor se vaya para atrás.

Para realizar el movimiento hacia la izquierda se toma un PWM fijo de 255, esto implica que se manda toda la potencia el motor y estaría en la figura 46



```

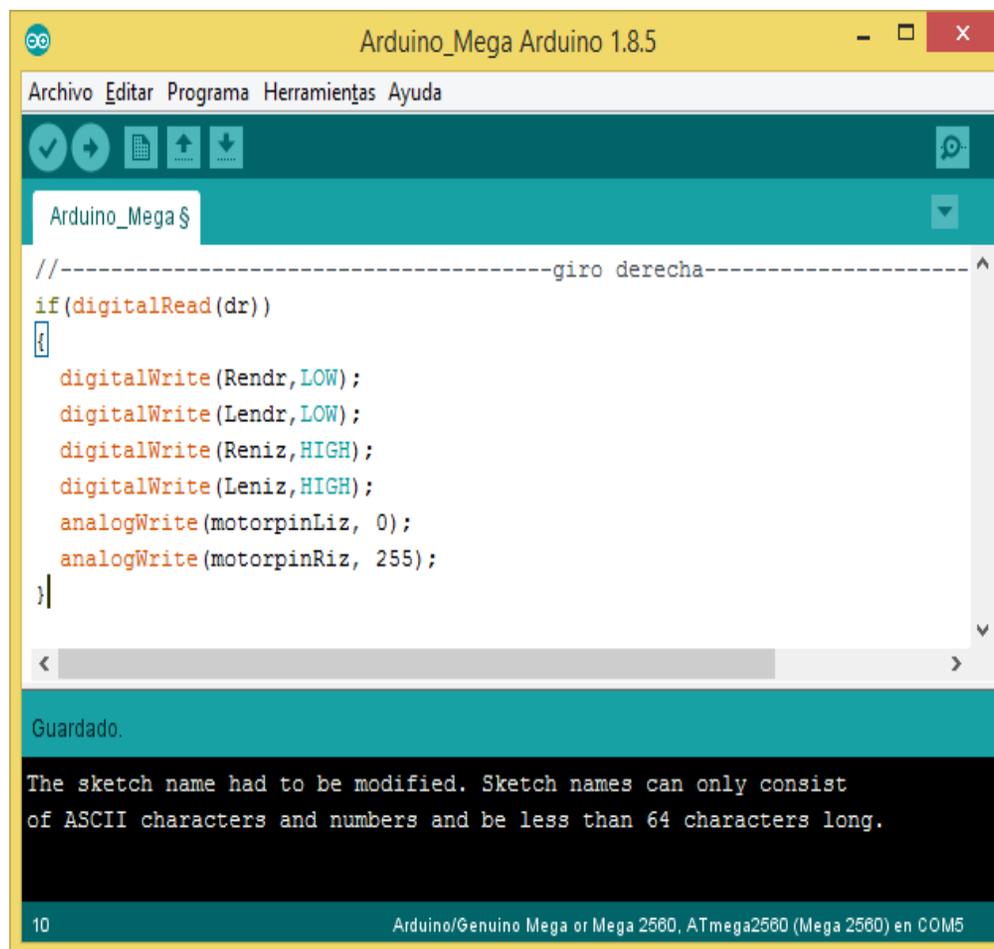
Arduino_Mega Arduino 1.8.5
Archivo Editar Programa Herramientas Ayuda
Arduino_Mega $
//-----giro izquierda-----
if(digitalRead(iz))
{
  digitalWrite(Rendr,HIGH);
  digitalWrite(Lendr,HIGH);
  digitalWrite(Reniz,LOW);
  digitalWrite(Leniz,LOW);
  analogWrite(motorpinLdr, 0);
  analogWrite(motorpinRdr, 255);
}
Guardado.
The sketch name had to be modified. Sketch names can only consist
of ASCII characters and numbers and be less than 64 characters long.
10 Arduino/Genuino Mega or Mega 2560, ATmega2560 (Mega 2560) en COM5

```

Figura 46. Programa Arduino Mega Movimiento Para La Izquierda

Están activadas los 2 controladores del puente H del motor derecho y al mandar la señal PWM de 255 al Rdr el motor girará con toda su potencia hacia adelante.

Para realizar el movimiento hacia la derecha se toma un PWM fijo de 255, esto implica que se manda toda la potencia el motor y estaría en la figura 47.



```

Arduino_Mega Arduino 1.8.5
Archivo Editar Programa Herramientas Ayuda
Arduino_Mega $
//-----giro derecha-----
if(digitalRead(dr))
{
  digitalWrite(Rendr,LOW);
  digitalWrite(Lendr,LOW);
  digitalWrite(Reniz,HIGH);
  digitalWrite(Leniz,HIGH);
  analogWrite(motorpinLiz, 0);
  analogWrite(motorpinRiz, 255);
}
Guardado.
The sketch name had to be modified. Sketch names can only consist
of ASCII characters and numbers and be less than 64 characters long.
10 Arduino/Genuino Mega or Mega 2560, ATmega2560 (Mega 2560) en COM5

```

Figura 47. Programador Arduino Mega Movimiento Para La Derecha

Están activadas los 2 controladores del puente H del motor izquierdo y al mandar la señal PWM de 255 al Riz el motor girará con toda su potencia hacia adelante.

Para mandar hacia adelante primero se comienza identificando el modelo matemático de la planta, que en este caso es el modelo matemático que sale de la onda que se genera en los motores ante una entrada escalón de 12V y una salida en RPM. En la

actualidad y gracias a softwares como MatLab se reduce significativamente el tiempo de análisis para determinar el tipo de onda y la función de transferencia que le acompaña.

Estos datos generados son importados a MatLab desde el puerto serial del arduino Mega, seguidamente mediante la herramienta de System Identification se calcula la función de transferencia como se muestra en la figura 48:

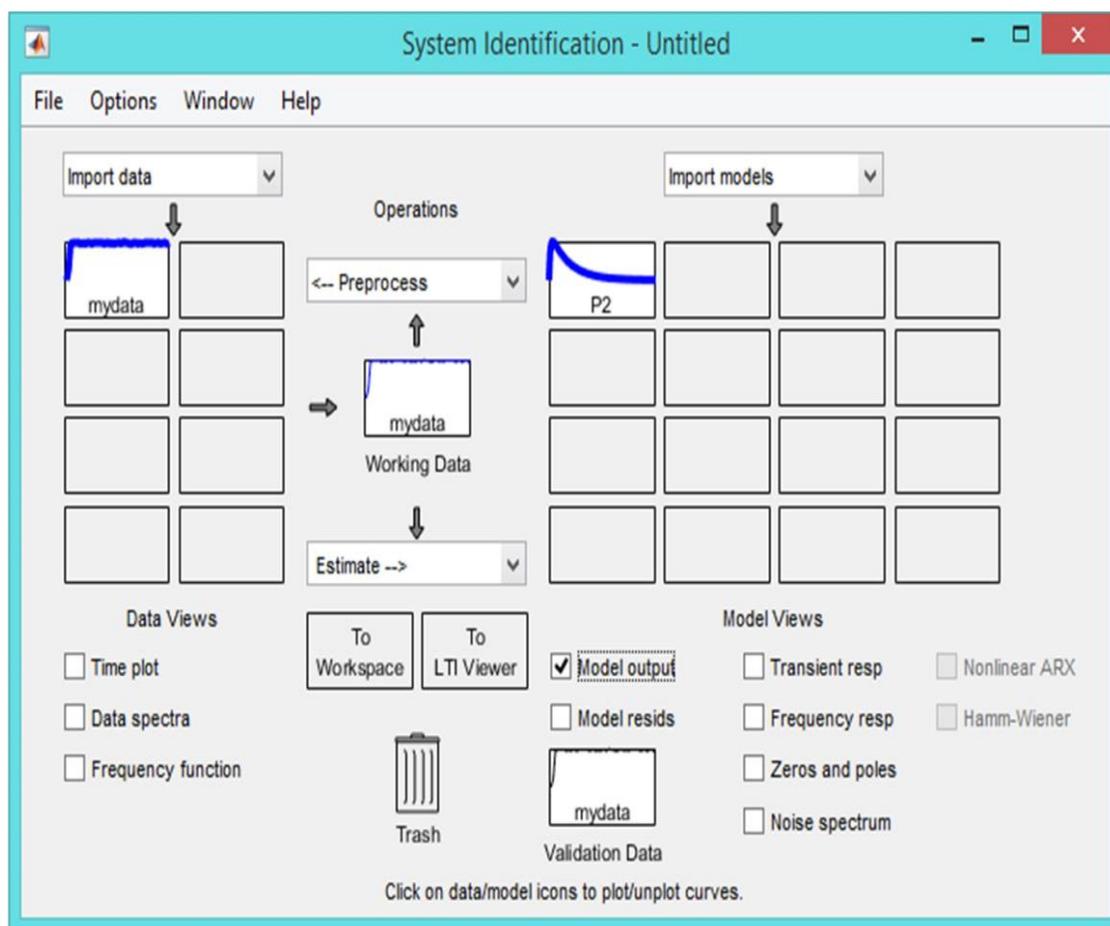


Figura 48. Herramienta de Identificación de la Planta en MatLab

En la figura 49 se muestra los valores de los coeficientes de la función de transferencia en el dominio de Laplace, donde se observa que el mejor ajuste es con una función de transferencia de segundo orden:

Process Models

Transfer Function

$$\frac{K}{(1 + T_{p1} s)(1 + T_{p2} s)}$$

Poles: 2, All real

Zero
 Delay
 Integrator

Par	Known	Value	Initial Guess	Bounds
K	<input type="checkbox"/>	8.9643	Auto	[-Inf Inf]
Tp1	<input type="checkbox"/>	1.9088	Auto	[0 10000]
Tp2	<input type="checkbox"/>	0.12375	Auto	[0 10000]
Tp3	<input type="checkbox"/>	0	0	[0 Inf]
Tz	<input type="checkbox"/>	0	0	[-Inf Inf]
Td	<input type="checkbox"/>	0	0	[0 Inf]

Initial Guess: Auto-selected
 From existing model:
 User-defined: Value-->Initial Guess

Disturbance Model: None
Focus: Simulation

Initial condition: Auto
Covariance: Estimate

Display progress

Name: P2

Buttons: Estimate, Close, Help, Regularization..., Options..., Continue

Figura 49. Coeficientes de la Función de Transferencia de Segundo Orden

A continuación, se muestra en la figura 50 una comparación de la respuesta temporal experimental contra los resultados teóricos del funcionamiento del motor al alimentarle con una fuerza electromotriz de 12V:

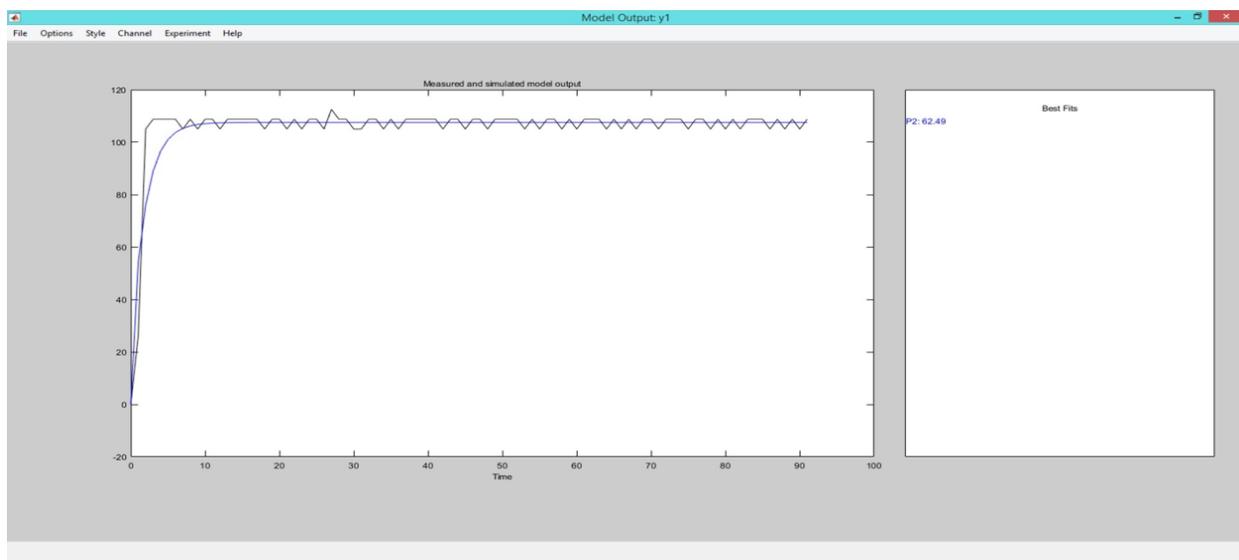


Figura 50. Respuesta Temporal De Resultados Experimentales Y Teóricos

Una vez obtenido la planta se procede al diseño del control PI clásico, cuyos parámetros se obtienen mediante técnica de ajuste de la herramienta Pid Tuner de MatLab y se logra determinar que el valor máximo para $K_p = 2,5$ y para $K_i = 1,2$. Hay que recordar que estos valores obtenidos son teóricos pero se los usará para limitar el funcionamiento del PI-Fuzzy Adaptativo, como se puede ver en la figura 51.

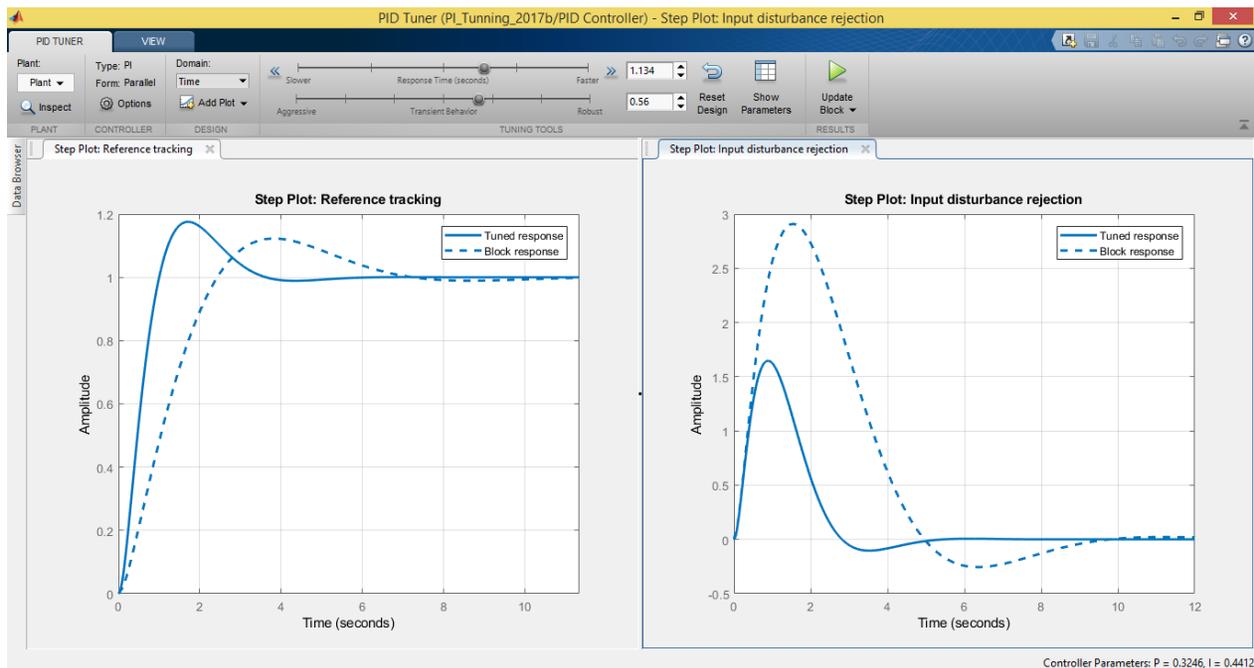


Figura 51. Ajuste de los Parámetros del Controlador PI

Con estos valores teóricos se procede a simular el controlador PI-Fuzzy Adaptativo en la planta del motor anteriormente identificado, por medio de la herramienta Fuzzy Logic Designer de Matlab, el cual se muestra en la figura 52. Esta herramienta nos pedirá los parámetros del control PI-Difuso es decir ingresaremos los límites de K_p y K_i del controlador PI. Una vez dado estos parámetros nos generaran las leyes de Fuzzificación que se adaptaran según el parámetro obtenido de K_p y K_i . Todo este procedimiento se podrá observar en el diagrama de bloques de la figura 53.

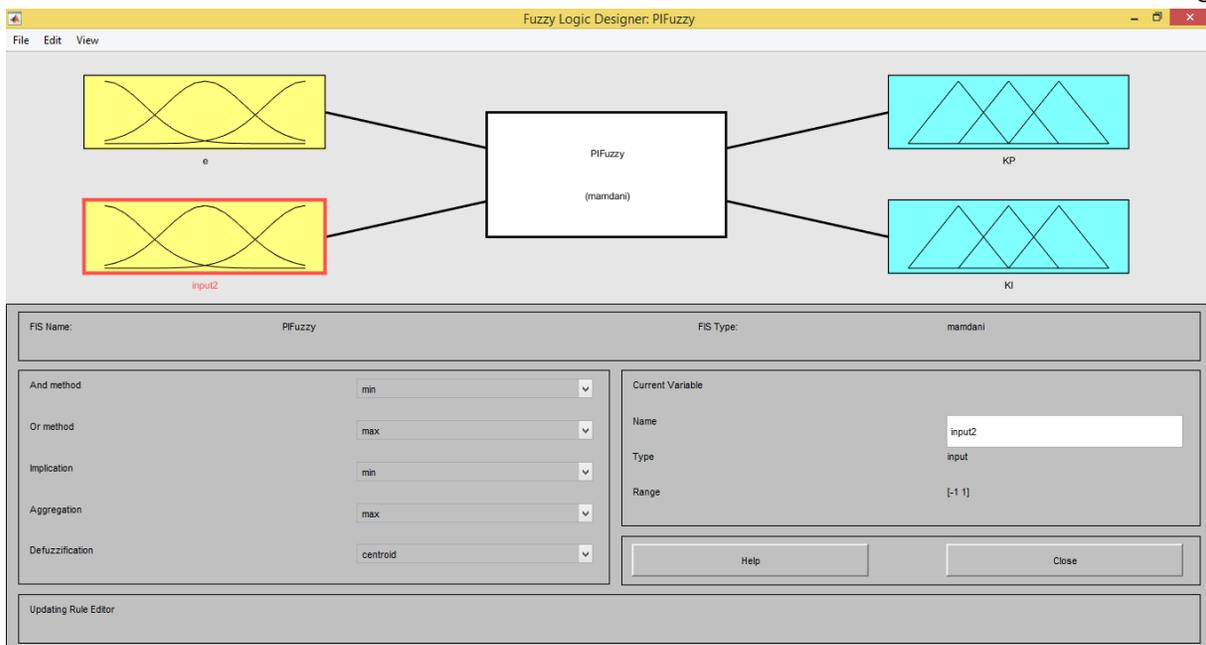


Figura 52. Herramienta De Diseño Del Control Fuzzy de MatLab

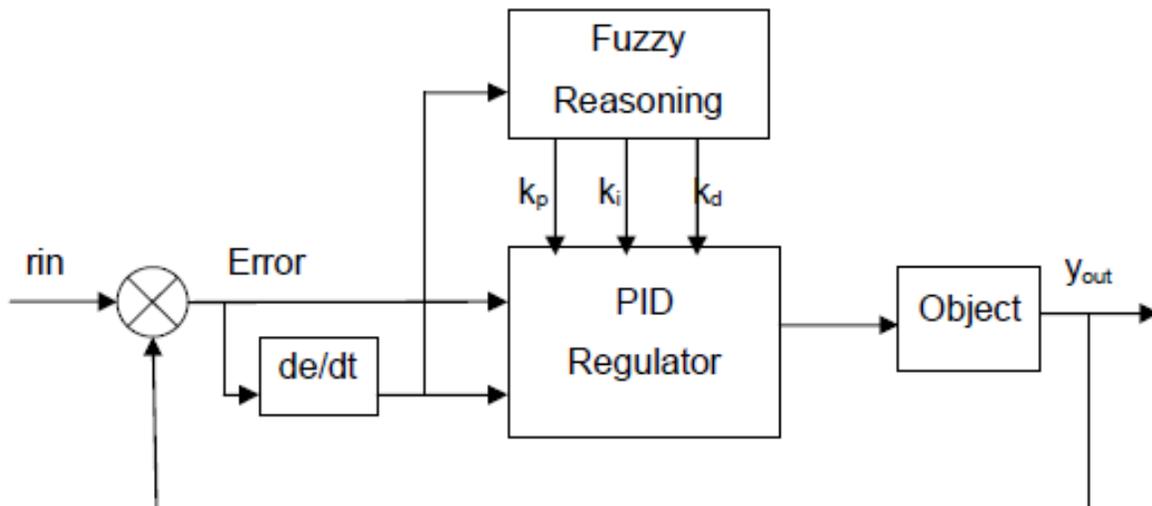


Figura 53. Diagrama De Bloques Del Control PI-Difuso Adaptativo Del Motor

Fuente: Diagrama de bloques PI-Fuzzy Adaptativo

Para crear las reglas de Fuzzificación Se tomó la decisión de trabajar con el máximo grupo de niveles dentro de los valores K_p y K_i , que son 7, lo que nos genera 49 reglas de Fuzzificación, como se observa en la figura 54.

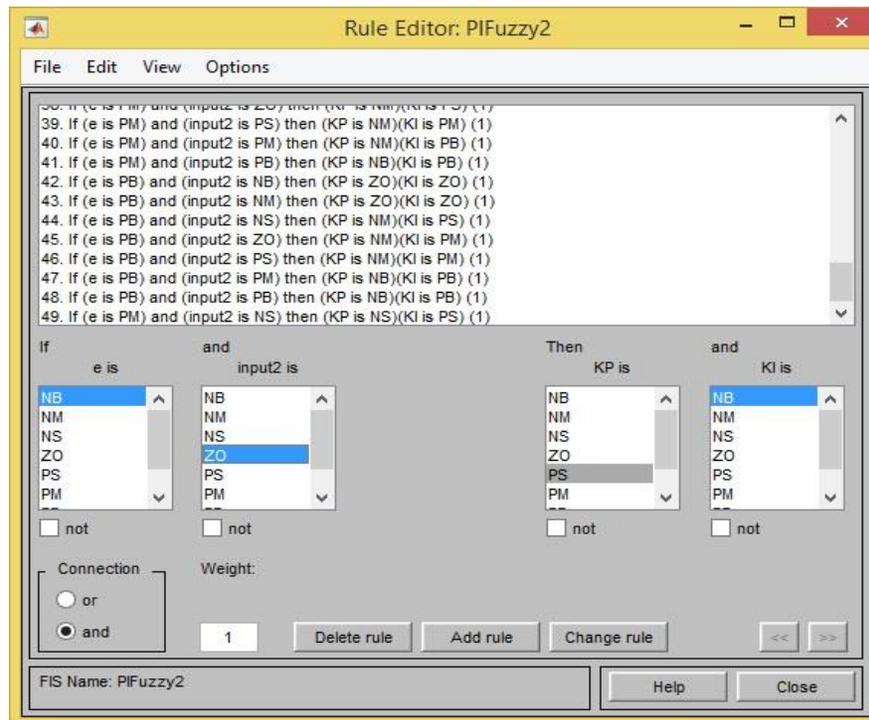


Figura 54. Reglas De Fuzzificación

El diagrama de flujo del controlador PI-Difuso Adaptativo se muestra en la figura 55.

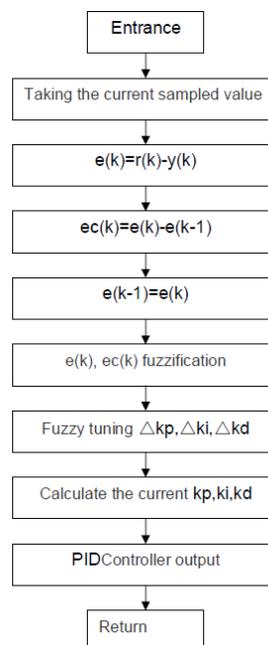


Figura 55. Diagrama Secuencial del Controlador PI-Difuso Adaptativo

Fuente: Diagrama Secuencial PI-Fuzzy Adaptativo

Una vez diseñado el control en MatLab para su posterior implementación se genera el programa de control equivalente para Arduino mediante esta página:

http://www.makeproto.com/projects/fuzzy/matlab_arduino_FIST/index.php

Con lo cual se generó la librería `fis_header.h` la cual nos sirve para generar el código equivalente para arduino, como se muestra en la figura 56:

```

Arduino_Mega Arduino 1.8.5
Archivo Editar Programa Herramientas Ayuda
Arduino_Mega
//-----fuzzy-----

// Number of inputs to the fuzzy inference system
const int fis_gcI = 2;
// Number of outputs to the fuzzy inference system
const int fis_gcO = 2;
// Number of rules to the fuzzy inference system
const int fis_gcR = 49;
FIS_TYPE g_fisInput[fis_gcI];
FIS_TYPE g_fisOutput[fis_gcO];

// Triangular Member Function
FIS_TYPE fis_trimf(FIS_TYPE x, FIS_TYPE* p)
{
    FIS_TYPE a = p[0], b = p[1], c = p[2];
    FIS_TYPE t1 = (x - a) / (b - a);
    FIS_TYPE t2 = (c - x) / (c - b);
    if ((a == b) && (b == c)) return (FIS_TYPE) (x == a);
    if (a == b) return (FIS_TYPE) (t2*(b <= x)*(x <= c));
    if (b == c) return (FIS_TYPE) (t1*(a <= x)*(x <= b));
    t1 = min(t1, t2);
    return (FIS_TYPE) max(t1, 0);
}

FIS_TYPE fis_min(FIS_TYPE a, FIS_TYPE b)
{
    return min(a, b);
}

FIS_TYPE fis_max(FIS_TYPE a, FIS_TYPE b)
{
    return max(a, b);
}

FIS_TYPE fis_array_operation(FIS_TYPE *array, int size, _FIS_ARR_OP pfnOp)
{
    int i;
    FIS_TYPE ret = 0;
    if (size == 0) return ret;
    if (size == 1) return array[0];
    ret = array[0];
    for (i = 1; i < size; i++)
    {
        ret = (*pfnOp)(ret, array[i]);
    }
    return ret;
}
Guardado.
The sketch name had to be modified. Sketch names can only consist
of ASCII characters and numbers and be less than 64 characters long.
-47 Arduino/Genuino Mega or Mega 2560, ATmega2560 (Mega 2560) en COM5

```

Figura 56. Código De MatLab Equivalente En Arduino

3.4. Simulación y Resultados

Una vez conocida la planta del motor y la lógica de Control PI-Difuso Adaptativo se generaran la simulación en SimuLink con el diagrama que se muestra en la figura 57.

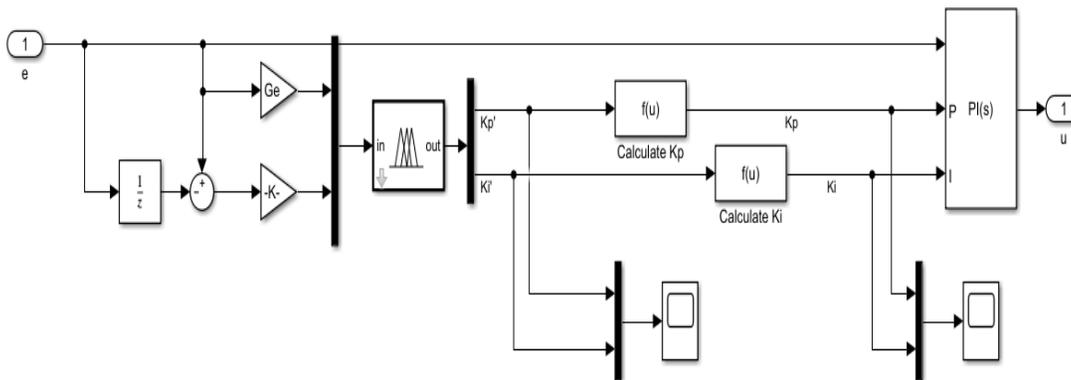


Figura 57. Estructura De Control Del PI-Difuso Adaptativo

A continuación se muestra una representación en diagrama de bloques de SimuLink de un controlador PI versus un controlador PI-Difuso Adaptativo que se observan en las figura 58.

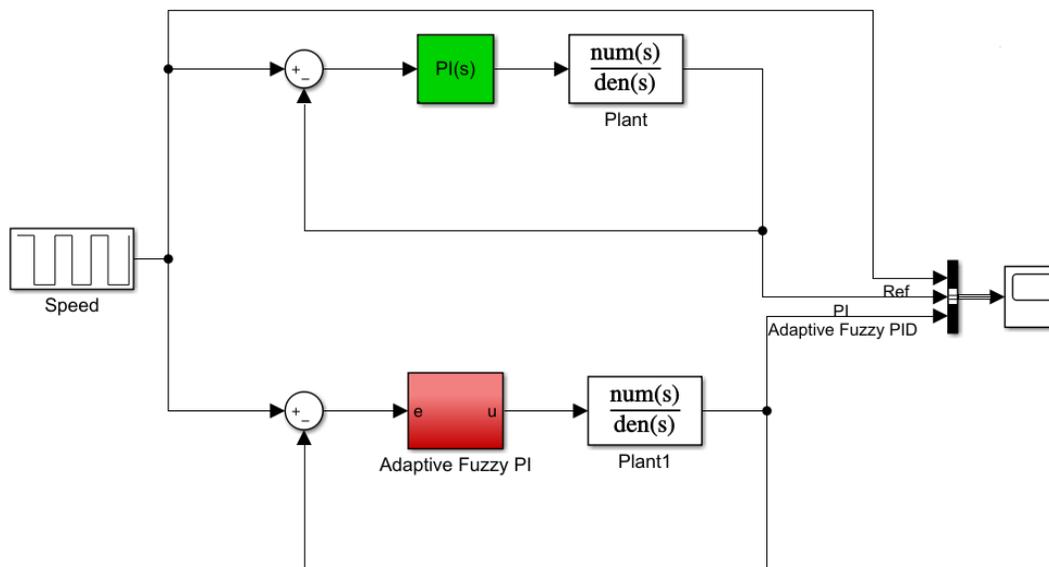


Figura 58. Diagrama de Bloques De Controladores PI y PI-Difuso Adaptativo

La respuesta temporal de cada uno de los controladores se muestra en la figura 59, donde se observa la variación de los parámetros K_p y K_i de manera adaptativa ante una perturbación en la referencia.

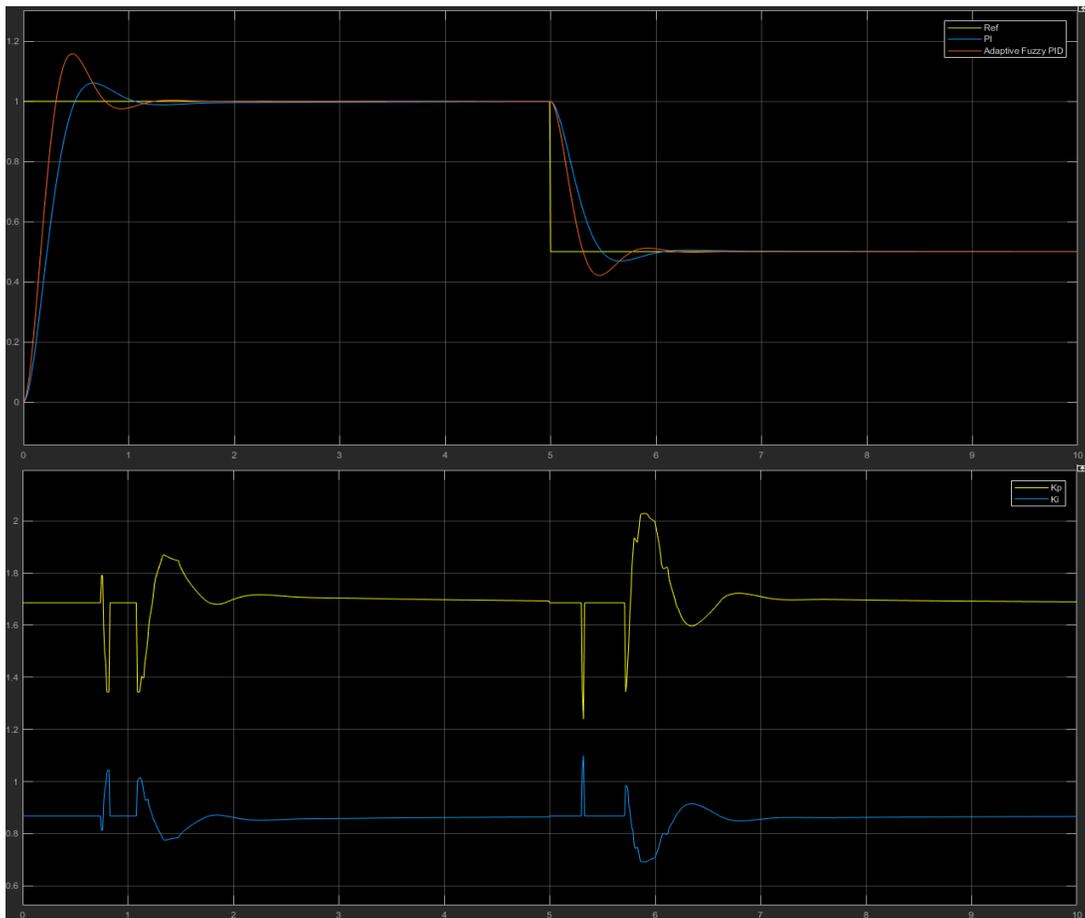


Figura 59. Respuesta Temporal De Los Controladores Y Adaptación De Los Parámetros K_p y K_i

Al simular las dos técnicas de control se logra verificar que hay una mejora en el tiempo de respuesta del Controlador PI-Fuzzy, donde los parámetros del regulador PI se actualizan en el tiempo en función de un cambio de referencia, permitiendo que sea un control Adaptativo, para controlar sistemas no lineales y variantes en el tiempo. En conclusión de manera general se verifica mediante simulación que el sistema de control PI-Difuso Adaptativo sigue la referencia de velocidad ante perturbaciones.

CAPÍTULO IV

4. IMPLEMENTACIÓN PARA LA MOVILIZACIÓN DE LA SILLA DE RUEDAS

En este capítulo se trata la instalación e implementación de los diferentes elementos tanto mecánicos como eléctricos y se hará una tabla de comparación entre controladores PI y el control PI-Difuso para revisar su tiempo de respuesta.

4.1. Instalación de Sistemas Mecánicos y Eléctricos.

Como se habló en capítulos anteriores se tuvo que reemplazar el sistema mecánico que se realizó en un inicio por problemas de sobre corriente que se tuvo y por excesivo ruido. También se cambió el sistema eléctrico ideado en un inicio con Mosfet, por uno que solo trabaja con elementos compatibles con la configuración Arduino.

4.1.1. Instalación de Parte Mecánica

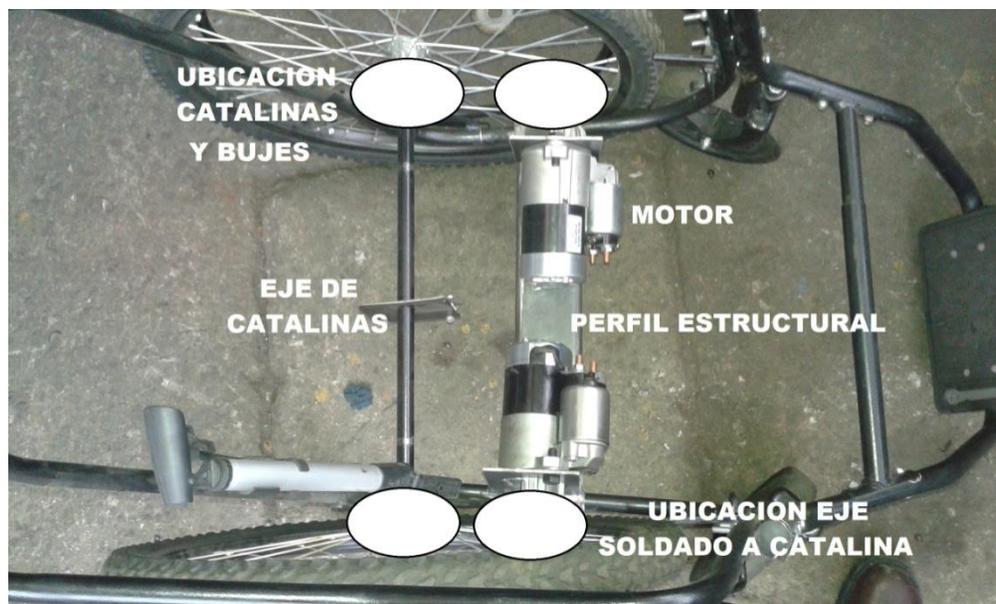


Figura 60. Esquema De La Ubicación De Los Accesorios Mecánicos

Como se puede ver en la figura 60, se puso un perfil estructural para sostener los motores que se puede ver detenidamente en la figura 22, posteriormente se les hicieron las bases de los motores que se les ve en la figura 21, estas bases se las realizo afuera

de la silla para coger las dimensiones de los agujeros que se necesita para atornillar los motores, posteriormente se les soldó al perfil estructural, se generó un eje que pueda contener el eje del motor como se ve en la figura 16, posteriormente se soldó la catalina como lo indica en la figura 17.

A continuación se comenzó soldando el eje posterior el cual sostendrá los bujes de bicicleta y las catalinas como se ve en la figura 19, se tomaron las medidas para hacer el buje que contendrá el buje de bicicleta y la catalina posterior tal como se ve en la figura 18. Por último se instalaron las cadenas de bicicleta para transmisión de catalinas, se instaló una base en donde va a ir la batería y se sujetaron las ruedas.

4.1.2. Instalación del Sistema de Control

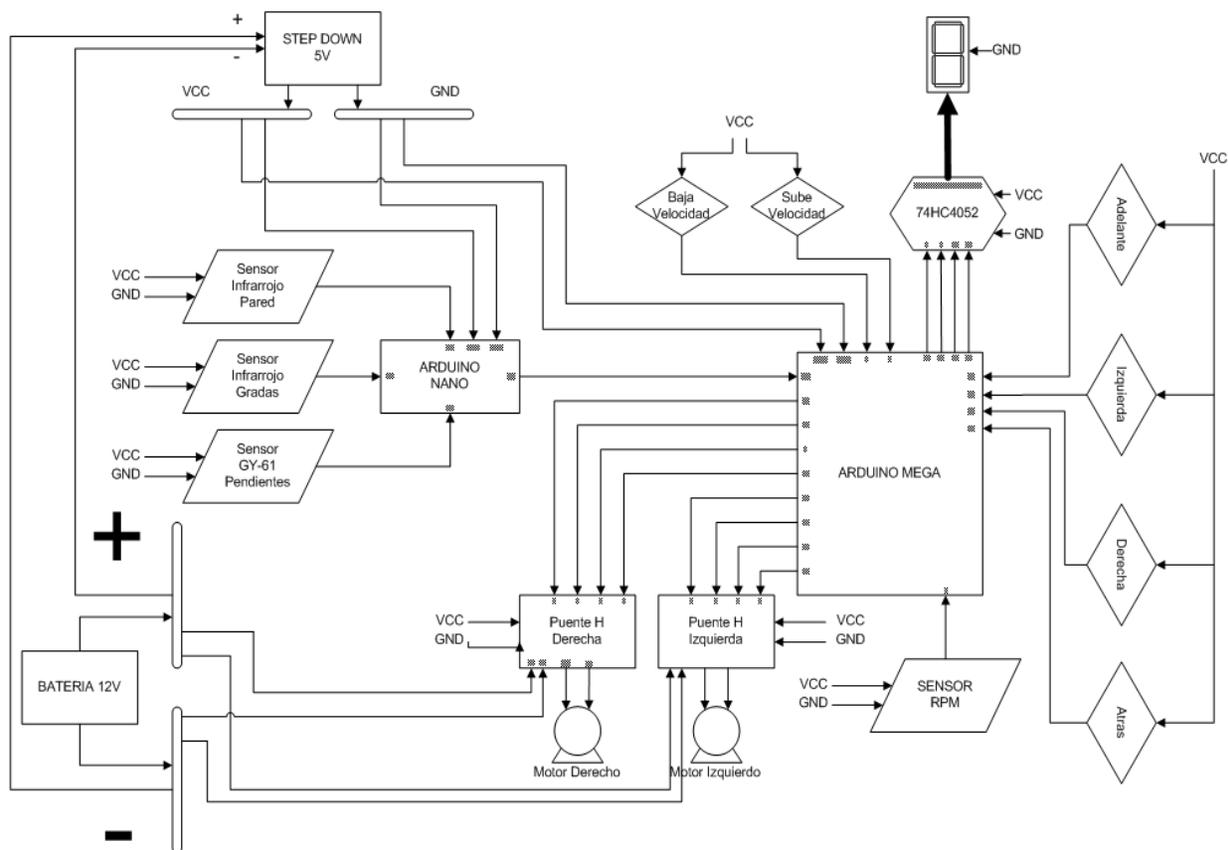


Figura 61. Diagrama Electrónico Del Sistema De Control

Como se observa en la figura 61, se comenzó creando una fuente que alimente al circuito de control esto se logró con un StepDown que nos permitió bajar el voltaje de 12V de la batería a 5V para sensores y arduinos, este integrado entrega una corriente de 3A suficiente para alimentar todo el circuito de control.

Una vez alimentado nuestro circuito se comenzó haciendo pruebas con el arduino nano para limitar los sensores infrarrojos, a los valores requeridos, sin instalarlos en la silla, lo mismo sucedió con el acelerómetro para ver el valor de inclinación para 15° positivos, como se ve en la figura 62, posteriormente se ajustaran estos valores ya instalados en la silla de ruedas.

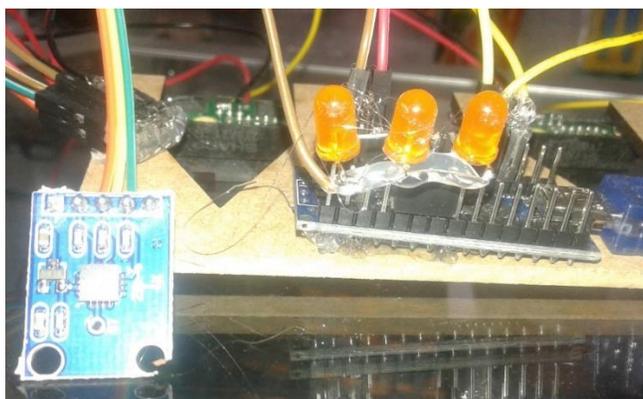


Figura 62. Programación Sensores

El único ajuste que se realiza fue en el sensor de inclinación el cual se ajusta a un valor de 320 que es el valor entendido por el acelerómetro y en el cual estaría inclinado positivamente la silla de ruedas a los 15° requeridos.

A continuación se comenzó haciendo las instalaciones con el arduino mega, igualmente sin instalarlo en la silla de ruedas, se comenzó conectando las variables para adelante, izquierda, derecha y atrás. Hay que recordar que estas entradas son entradas de 5V que van a estar conectadas a un Joystick, como se ve en la figura 63.



Figura 63. Conexiones Del Joystick

Paralelamente a esto se realizó las conexiones para el display, para visualizar en que velocidad se encuentra, junto con los botones que aumentan y disminuyen la velocidad, todo esto se ve en la figura 64.

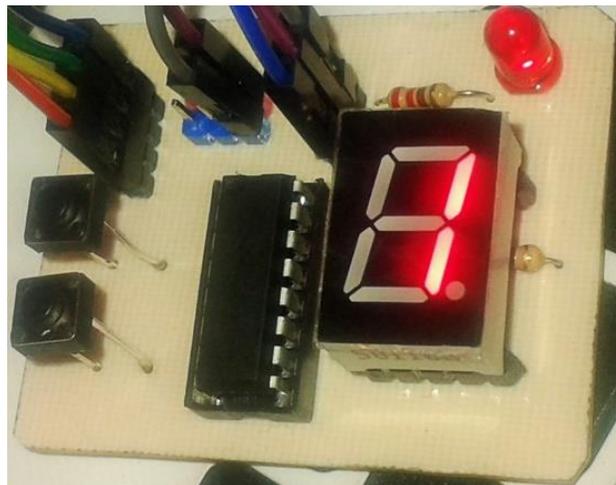


Figura 64. Conexiones De Display Y Botones

Todas estas conexiones al momento de ya instalarlas en la silla de ruedas saldrán de una caja metálica como se ve en la figura 65.



Figura 65. Caja de Control de Velocidad y Movimiento

Seguidamente se hacen pruebas con el Puente H y un motor de 12V de baja potencia para visibilizar la programación. Una vez realizadas estas pruebas e instaladas las conexiones del joystick, botones de velocidad y display en la caja metálica, se procede a hacer las conexiones de los motores antes de instalarlas en la silla de ruedas y procedemos a hacer pruebas con el motor DC que se va a usar para el movimiento de la silla de ruedas.

4.2. Pruebas y Análisis de la Silla en Condiciones Alternas.



Figura 66. Silla De Ruedas Terminada

4.2.1. Pruebas en Condiciones Alternas

Los Resultados Experimentales obtenidos a partir del puerto serial del Arduino Mega, al imponer diferentes pesos en una misma velocidad, son los que se muestran en las figuras 67, figura 68 y figura 69.

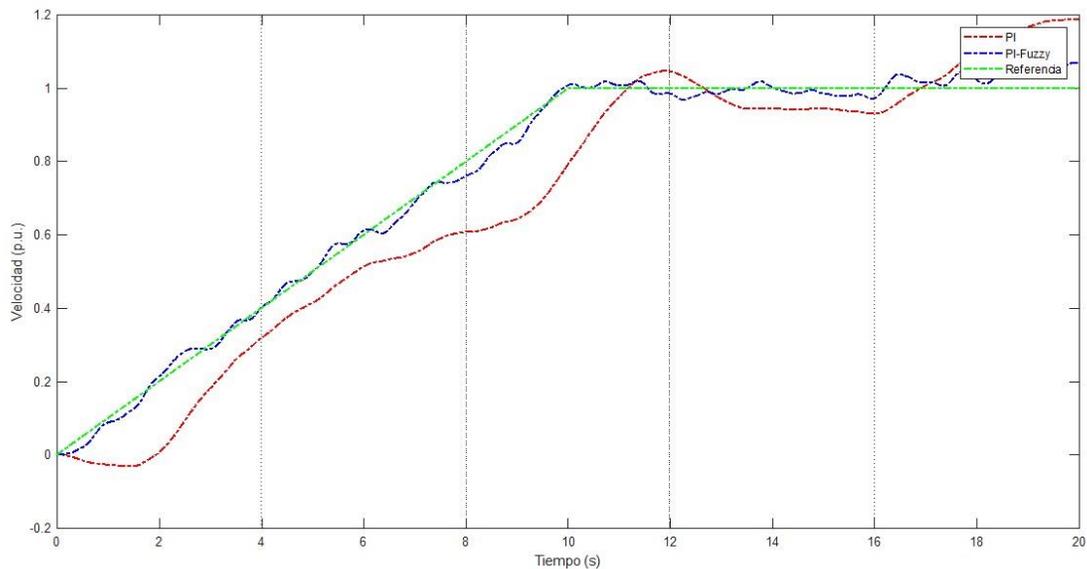


Figura 67. Respuesta Temporal Para Peso de 100Kg

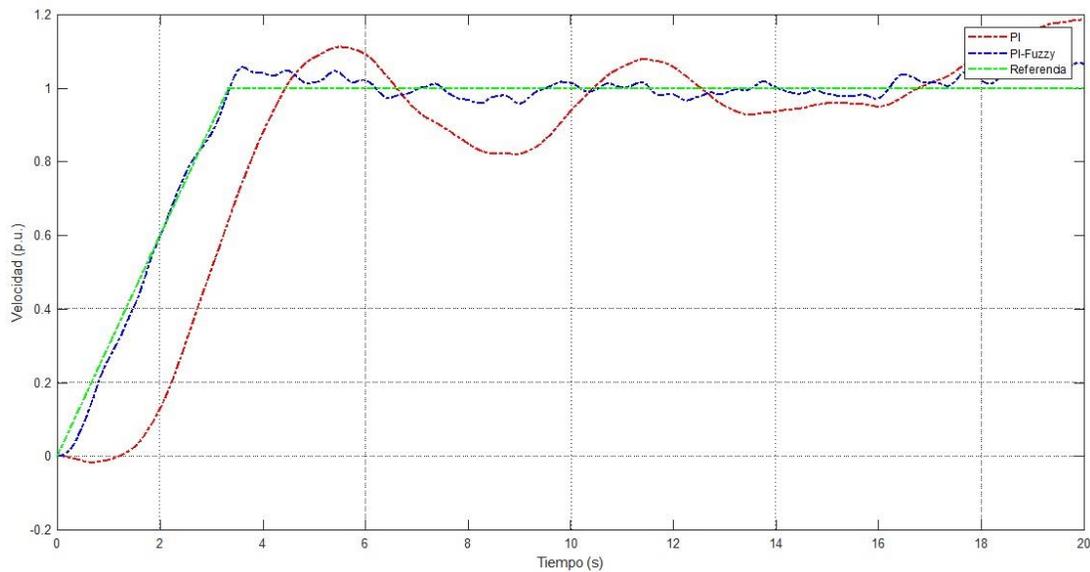


Figura 68. Respuesta Temporal Para Peso De 75Kg

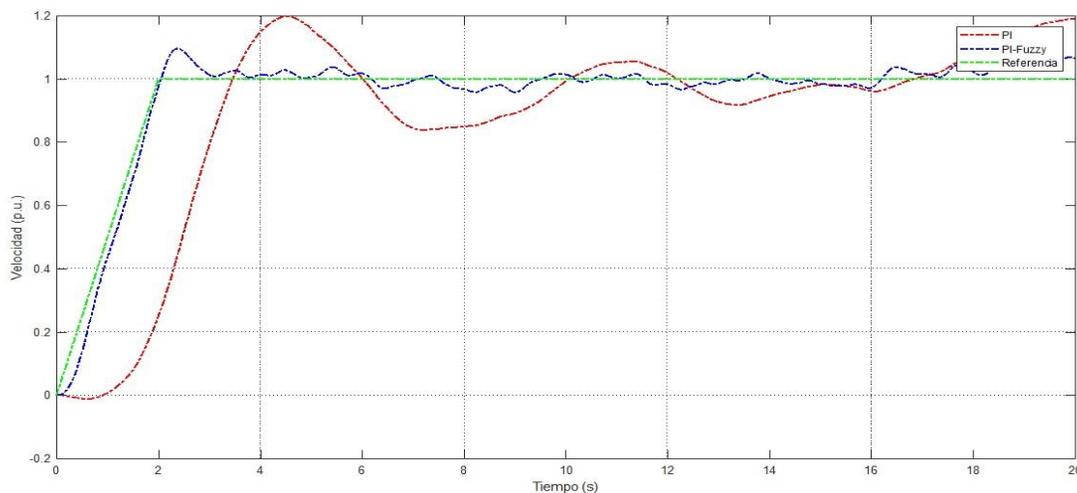


Figura 69. Respuesta Temporal Para Peso De 60Kg

En las figuras anteriores se observa mediante la variación del aumento de peso en cada prueba los diferentes tiempos para alcanzar la velocidad de 1,7m/s.

4.2.2. Análisis de Pruebas

Para un análisis comparativo de las pruebas se implementó un controlador PI el cual se compilo en el arduino mega y se logró obtener los datos de la tabla 6.

Tabla 7:

Cuadro Comparativo Del Tiempo de Respuesta de Controladores

PESO	100Kg	75Kg	60Kg
CONTROL PI	11 seg	5seg	3.5seg
CONTROL PI-FUZZY ADAPTATIVO	10seg	3.5seg	2seg

4.3. Validación de Resultados

Gracias a la comparación de los dos controladores se observa que el tiempo de respuesta para el controlador PI-difuso es más estable y rápido que el controlador PI ante perturbaciones.

CAPÍTULO V

5. CONCLUSIONES Y RECOMENDACIONES

Una vez realizadas las pruebas descritas anteriormente con diferentes pesos y en una misma velocidad se verifico el comportamiento del controlador PI-Difuso y se llegó a las siguientes conclusiones y recomendaciones.

5.1. Conclusiones

- El presente proyecto bajo los costos de una silla de ruedas eléctrica convencional en el mercado, se llegó a gastar 750 dólares con mano de obra y programación.

- El controlador PI-Difuso evita picos excesivos al momento del arranque de los motores debido a la estabilidad presentada por el controlador Difuso.

- La respuesta del controlador PI-Difuso es más rápida para estabilizar el movimiento de los motores

- Gracias al controlador PI-Difuso se corrigen errores en el controlador Difuso, debido a que en su programación el controlador PI-Difuso puede auto-calibrar sus reglas a través del manejo de pruebas y errores.

- Con los datos adquiridos por el software de MatLab se observó un buen tiempo de respuesta del controlador PI-Difuso.

- Gracias al controlador PI-Difuso al evitar picos y tener un menor tiempo para estabilizarse se verá un ahorro de energía y un mayor tiempo de movilización.

- El presente Proyecto de Investigación ayudará a entender de mejor forma la teoría del comportamiento de un controlador PI-Difuso Adaptado a diferentes limitaciones.

- Se podrá mejorar la calidad de vida en los usuarios de la silla de ruedas implementada en el presente proyecto, principalmente para las personas que viven en el Hogar de Ancianos “Dulce María”, patrocinador del presente proyecto.

5.2. Recomendaciones

- Al momento de realizar las pruebas de ambos controladores se deben verificar que estén en las mismas condiciones de peso y velocidad para evitar errores al momento de compararlas.

- Se deben aumentar el uso de software como MatLab y SimuLink para realizar los modelos matemáticos y simulaciones para un ahorro muy significativo en el tiempo al implementar proyectos.

- Tratar de implementar el controlador PI-Difuso para obtener mejores respuestas en las aplicaciones industriales donde se utiliza frecuentemente por separado los controladores PI, PID y Difuso.

- Se recomienda implementar el controlador PI-Difuso en las prácticas de laboratorio para mejorar el tiempo de respuesta en los proyectos de los estudiantes.

Referencias Bibliográficas

1. L. A. Bryan; E. A. Bryan, Programmable controllers theory and implementation, segunda edición, Text Company, 1997.
2. Michael Jhonson, Mohamad Moradi, PID Control, Springer, 2005
3. MatLab, Fuzzy logic toolbox, Math Works, 2019.
4. Saulius Lisauskas, Fuzzy adaptive PID control design, Vilnius University, 2011
5. Muñoz-César. Speed control of a DC brush motor with conventional PID And Fuzzy PI Controllers, Sección de Estudios de Posgrado e Investigación – ESIME – IPN, 2008.
6. Bhushan, B. Adaptive control of DC motor using bacterial foraging algorithm. Applied Soft Computing. pp. 4913-4920, (2011).
7. Herrera Espinosa Javier. Implementación en un DSP de un regulador de velocidad difuso tipo Takagi-Sugeno para un arreglo de motor de corriente directa alternador. Tesis para obtener el grado de Maestro en Ciencias. México D.F. SEPI, (2003).
8. Benalcázar T., J. I., & Verduga F., G. A. Diseño e implementación de un sistema de control pid de posición de un motor de corriente continua por medio de hardware y software matlab - simulink para uso didáctico en la facultad de educación técnica para el desarrollo. (Trabajo de Titulación). Universidad Católica de Santiago de Guayaquil, Guayaquil. Recuperado a partir de <http://repositorio.ucsg.edu.ec/handle/3317/1604>, (2013).
9. Bolton, W. Ingenieria de control. Control Difuso. Obtenido de Universidad Simon Bolivar: prof.usb.ve/montbrun/ps2320/fuzzy/fuzzy.html. pp. 21-23. (2001).

10. Diordiev, A., Ursaru, O., Lucanu, M., & Tigaeru, L. A Hybrid PID-FUZZY controller for dc/dc converters (Vol. 1, pp. 97–100). IEEE. <https://doi.org/10.1109/SCS.2003.1226957>. (2003).

11. H. M. Gerardo Celso, Control difuso de velocidad de un motor de corriente directa. Sepi-esime zacatenco. Tesis para obtener el grado de Maestro en Ciencias, 1997.

12. Saim, Jyoti Ohri, Naveen, Speed control of dc motor using fuzzy logic based on Labview, International Journal of Scientific and Research Publications, Volume 3, Issue 6, 2013.

13. Fattah, A. Design and analysis of speed control using hybrid pid-fuzzy controller for induction motors (Tesis de Maestría). (2015).

14. Kandiban, R., & Arulmozhiyal, R. Speed control of bldc motor using adaptive fuzzy PID controller. Procedia Engineering, 38, 306–313. <https://doi.org/10.1016/j.proeng.2012.06.039>. (2012).

15. Gunt, Hamburg. Conocimientos básicos de control difuso.

16. Herrera Espinosa Javier y Ramírez Trujillo Jorge. Implementación de un control difuso de velocidad a un motor DC. Microcontrolador 68HC11. 4ª semana de Control y Automatización. 2000

17. J. P. Keller, Teaching PID and fuzzy controllers with Labview, Oensingen Institute of Technology, Switzerland, 2000.

- Fuente: Consejo De Discapacidades
<https://www.consejodiscapacidades.gob.ec/estadisticas-de-discapacidad/>

- Fuente: Medidas Generales De Una Silla De Ruedas
<http://www.dspace.uce.edu.ec/bitstream/25000/5266/1/T-UCE-0001-0143.pdf>

- Fuente: Diagrama Arduino Mega
https://www.researchgate.net/profile/Subhi_Zeebaree/publication/299533988_Arduino_Based_Remote_Controlling_for_Home_Power_Saving_Security_and_Protection/links/56fd96d108aee995dde56120.pdf

- Fuente: Diagrama Arduino Nano <https://repositorio.espe.edu.ec/handle/21000/7968>

- Fuente: PID Control; Michael Jhonson, Mohamad Moradi, PID control, Springer, 2005

- Fuente: Arquitectura Paralela del Controlador PID; Michael Jhonson, Mohamad Moradi, PID control, Springer, 2005

- Fuente: Modos de Control; L. A. Bryan; E. A. Bryan, Programmable controllers theory and implementation, segunda edición, Text Company, 1997.

- Fuente: Acción del controlador Proporcional Integral; L. A. Bryan; E. A. Bryan, Programmable controllers theory and implementation, segunda edición, Text Company, 1997.

- Fuente: Diagrama de temperaturas en el Hemisferio Norte; MatLab, Fuzzy logic toolbox, Math Works, 2019.

- Fuente: Fuzzy Logic Controller Operation; L. A. Bryan; E. A. Bryan, Programmable controllers theory and implementation, segunda edición, Text Company, 1997.

- Fuente: Forma de las Funciones Miembro; L. A. Bryan; E. A. Bryan, Programmable controllers theory and implementation, segunda edición, Text Company, 1997.

- Fuente: Sistema de Lógica Difusa con Entrada y Salida; L. A. Bryan; E. A. Bryan, Programmable controllers theory and implementation, segunda edición, Text Company, 1997.

- Fuente: Ubicación del Centroide; L. A. Bryan; E. A. Bryan, Programmable controllers theory and implementation, segunda edición, Text Company, 1997.

- Fuente: Diagrama de bloques PI-Fuzzy Adaptativo; Saulius Lisauskas, Fuzzy adaptive PID control design, Vilnius University, 2011.

- Fuente: Reglas de Tabla de Control Fuzzy Kp; Saulius Lisauskas, Fuzzy adaptive PID control design, Vilnius University, 2011

- Fuente: Reglas de Tabla de Control Fuzzy Ki; Saulius Lisauskas, Fuzzy adaptive PID control design, Vilnius University, 2011

- Fuente: Diagrama de bloques PI-Fuzzy Adaptativo; Saulius Lisauskas, Fuzzy adaptive PID control design, Vilnius University, 2011

- Fuente: Diagrama Secuencial PI-Fuzzy Ad Fuzzy adaptive PID control design, Vilnius University, 2011