

UNIVERSIDAD NACIONAL DEL CALLAO

FACULTAD DE INGENIERÍA ELÉCTRICA Y ELECTRÓNICA

ESCUELA PROFESIONAL DE INGENIERÍA ELECTRÓNICA



**“DISEÑO E IMPLEMENTACIÓN DE UN ROBOT
MÓVIL DE DESPLAZAMIENTO AUTÓNOMO
BASADO EN UN CONTROLADOR
PROPORCIONAL DERIVATIVO”**

TESIS PARA OPTAR EL TÍTULO PROFESIONAL DE
INGENIERO ELECTRÓNICO

Autores:

Flores Chacayan, Luis Antonio

Garay Sánchez, Michael David

Asesorado por:

Ing. Julio Cesar Borjas Castañeda

Callao, 2017

PERÚ

UNIVERSIDAD NACIONAL DEL CALLAO

FACULTAD DE INGENIERÍA ELÉCTRICA Y ELECTRÓNICA

ESCUELA PROFESIONAL DE INGENIERÍA ELECTRÓNICA



“DISEÑO E IMPLEMENTACIÓN DE UN ROBOT MÓVIL DE DESPLAZAMIENTO AUTÓNOMO BASADO EN UN CONTROLADOR PROPORCIONAL DERIVATIVO”

TESIS PARA OPTAR EL TÍTULO PROFESIONAL DE
INGENIERO ELECTRÓNICO

INTEGRANTES

Flores Chacayan, Luis Antonio

Garay Sánchez, Michael David

ASESOR

Ing. Julio Cesar Borjas Castañeda

CALIFICACION:

18 DIECIOCHO

ING. JACOB ASTOCONDOR VILLAR
PRESIDENTE DEL JURADO

ING. WILBER CHAVEZ IRAZABAL
SECRETARIO DEL JURADO

RICARDO RODRIGUEZ BUSTINZA
VOCAL DE JURADO

CALLAO – PERU
2017

DEDICATORIA

Dedico este trabajo a dios por iluminarme y guiarme en la vida. También dedico a mis padres, porque ellos siempre están apoyándome en las buenas y en las malas, con su educación, consejos, valores para poder conducirme correctamente en mi camino.

AGRADECIMIENTOS

Agradezco a mis docentes por haberme enseñado la pasión de esta hermosa carrera que es la Ingeniería Electrónica, mis padres por darme el impulso en los momentos importantes de mi vida, a mis compañeros de la universidad y a los integrantes del grupo de investigación de Mecatrónica, por que estuvieron en todo este proceso de formación profesional.

ÍNDICE

Índice de figuras	1
Índice de tablas	7
Resumen	8
Abstract	9
I. PLANTEAMIENTO DE LA INVESTIGACIÓN	10
1.1. Identificación del problema	10
1.2. Formulación del problema	10
1.3. Objetivos de la investigación	11
1.4. Justificación	11
II. MARCO TEÓRICO	12
2.1. Antecedentes del estudio	12
2.2. Fundamento ontológico	14
2.3. Fundamento metodológico	14
2.4. Fundamento epistemológico	14
2.5. Definición de términos básico	15
III. VARIABLES E HIPÓTESIS	16
3.1. Definición de las variables	16
3.2. Operacionalización de variables	16
3.3. Hipótesis general e hipótesis específicas	18
IV. METODOLOGIA	18
4.1. Tipo de investigación	18
4.2. Diseño de la investigación	18
4.2.1. Estudio de las ventajas de la robotización	18
4.2.2. Encoders efecto hall	19
4.2.3. Definición Tacómetro	20
4.2.4. Filtros de Respuesta Infinita (IIR)	21
4.2.5. Interface de Labview para Arduino (LIFA)	22
4.2.6. Arduino como unidad de control y comunicación	35
4.2.7. Toolbox de Matlab para la identificación y control	39
4.2.8. Tipos de locomoción	44
4.2.9. Modelamiento Cinemático Directo e Inverso del Robot móvil	49
4.2.10. Diseño mecánico y elección de motores	55

4.2.11. Diseño de las tarjetas electrónicas	58
4.2.12. Configuración y sincronización del módulo Bluetooth	74
4.2.13. Desarrollo de la interfaz gráfica en Labview	80
4.2.14. Adquisición, identificación y Control para un motor	120
4.2.15. Construcción del robot móvil	147
4.2.16. Control de posición PD para el Robot Móvil	151
4.2.17. Limitaciones del Software y robot Móvil	154
4.2.18. Análisis del funcionamiento del robot	155
4.3. Población y muestra	159
4.4. Técnicas e instrumentos de recolección de datos	159
4.5. Procedimientos de recolección de datos	159
4.6. Procesamiento estadístico y análisis de datos	159
V. RESULTADOS	160
VI. DISCUSIÓN DE LOS RESULTADOS	167
6.1. Contrastación de hipótesis con los resultados	167
6.2. Contrastación de resultados con otros similares	167
VII. CONCLUSIONES	168
VIII. RECOMENDACIONES	169
IX. REFERENCIAS BIBLIOGRÁFICAS	170
ANEXOS	171
Anexo N°1: Matriz de consistencia	171
Anexo N°2: Cronograma de actividades	172
Anexo N°3: Diagrama de Gantt	174
Anexo N°4: Presupuesto de Costo por Tareas	175
Anexo N°5: Presupuesto de Recursos por Tareas	176

ÍNDICE DE FIGURAS

<i>FIGURA N°1 Encoder de efecto hall</i>	19
<i>FIGURA N°2 Diagrama de Tacómetro</i>	20
<i>FIGURA N°3 Circuito Tacómetro analógico</i>	20
<i>FIGURA N°4 Filtro de primer Orden</i>	22
<i>FIGURA N°5 Plataforma LabView</i>	24
<i>FIGURA N°6 VI Panel Frontal</i>	24
<i>FIGURA N°7 VI Diagrama de Bloques.</i>	25
<i>FIGURA N°8 Bloque de herramientas Lifa</i>	27
<i>FIGURA N°9 Bloque Init</i>	27
<i>FIGURA N°10 Bloque Set Digital Pin Mode</i>	27
<i>FIGURA N°11 Bloque Digital Write Pin</i>	28
<i>FIGURA N°12 Bloque PWM Write Pin</i>	28
<i>FIGURA N°13 Bloque I2C Init</i>	28
<i>FIGURA N°14 Bloque I2C Read</i>	28
<i>FIGURA N°15 Bloque I2C Write</i>	29
<i>FIGURA N°16 Bloque Close</i>	29
<i>FIGURA N°17 Bloque While Loop</i>	29
<i>FIGURA N°18 Bloque Case Structure</i>	30
<i>FIGURA N°19 Bloque The waveform chart</i>	30
<i>FIGURA N°20 Bloque Write To Measurement File Express VI</i>	30
<i>FIGURA N°21 Configuración del Bloque Write To Measurement File</i>	31
<i>FIGURA N°22 Bloque Index Array</i>	32
<i>FIGURA N°23 Bloque Merge Signals</i>	33
<i>FIGURA N°24 Bloque Quotient & Remainder Function</i>	33
<i>FIGURA N°25 Bloque Not Equal To 0? Function</i>	33
<i>FIGURA N°26 Bloque Boolean To (0,1)</i>	33
<i>FIGURA N°27 Bloque Multiply</i>	34
<i>FIGURA N°28 Bloque Equal?</i>	34
<i>FIGURA N°29 Bloque Divide</i>	34
<i>FIGURA N°30 Bloque Or</i>	34
<i>FIGURA N°31 Arduino board</i>	35

<i>FIGURA N°32 Tabla de Características (Atmega 328)</i>	36
<i>FIGURA N°33 Tarjeta Arduino</i>	37
<i>FIGURA N°34 Entorno de trabajo de Matlab</i>	40
<i>FIGURA N°35 Tabla de Toolbox Matlab</i>	41
<i>FIGURA N°36 Toolbox de identificación</i>	42
<i>FIGURA N°37 Toolbox de Control</i>	43
<i>FIGURA N°38 Tipos de locomoción</i>	45
<i>FIGURA N°39 Restricción no holonómicas</i>	46
<i>FIGURA N°40 Sistema de Locomoción Tipo Ackerman</i>	46
<i>FIGURA N°41 Sistema de Locomoción Tracción omnidireccional</i>	47
<i>FIGURA N°42 Sistema de Locomoción Tracción diferencial</i>	47
<i>FIGURA N°43 Giros de plataforma diferencial</i>	48
<i>FIGURA N°44 Tipo de Trayectoria Analizada</i>	49
<i>FIGURA N°45 Orientación del robot y sentido de giro</i>	50
<i>FIGURA N°46 Cinemática directa del robot diferencial</i>	51
<i>FIGURA N°47 Diseño del chasis en el Software Autocad</i>	55
<i>FIGURA N°48 Diseño de la carrocería en el Software Autocad</i>	55
<i>FIGURA N°49 Deformación de neumático</i>	57
<i>FIGURA N°50 Esquema del circuito Puente H</i>	58
<i>FIGURA N°51 Layout del circuito Puente H</i>	61
<i>FIGURA N°52 Página calculadora de ancho de pista</i>	62
<i>FIGURA N°53 Esquema del circuito tacómetro digital</i>	63
<i>FIGURA N°54 Señal de salida del encoder</i>	66
<i>FIGURA N°55 Layout del circuito Tacómetro</i>	67
<i>FIGURA N°56 Tarjeta de Conexión de Periféricos</i>	68
<i>FIGURA N°57 Figura Layout de tarjeta de Conexión de Periféricos</i>	73
<i>FIGURA N°58 Esquema de conexión entre Arduino y Bluetooth</i>	75
<i>FIGURA N°59 Código para el ingreso de comando AT</i>	75
<i>FIGURA N°60 Ventana para el ingreso y salida de datos Serial</i>	76
<i>FIGURA N°61 Bluetooth tabla de comandos</i>	76
<i>FIGURA N°62 Confirmación de comando de velocidad</i>	77
<i>FIGURA N°63 Icono del Dispositivo Bluetooth</i>	77
<i>FIGURA N°64 Agregar un nuevo dispositivo Bluetooth</i>	78

<i>FIGURA N°65 Dispositivo Bluetooth Slave HC-06</i>	78
<i>FIGURA N°66 Código de emparejamiento</i>	78
<i>FIGURA N°67 Dispositivo Bluetooth se agregó correctamente</i>	79
<i>FIGURA N°68 Vista Isométrica de un almacén</i>	80
<i>FIGURA N°69 Vista de planta de un almacén y Panel Frontal en Labview</i>	81
<i>FIGURA N°70 Variables de Entrada y selección de la coordenada final</i>	82
<i>FIGURA N°71 Coordenada Inicial en el Origen</i>	82
<i>FIGURA N°72 Coordenada Final en el Punto A1</i>	83
<i>FIGURA N°73 Coordenada Final en el Punto A2</i>	83
<i>FIGURA N°74 Coordenada Final en el Punto B1</i>	84
<i>FIGURA N°75 Coordenada Final en el Punto B2</i>	84
<i>FIGURA N°76 Coordenada Final en el Punto B3</i>	85
<i>FIGURA N°77 Coordenada Final en el Punto B4</i>	85
<i>FIGURA N°78 Coordenada Final en el Punto B5</i>	86
<i>FIGURA N°79 Inicialización de variables y configuración de pines IO</i>	86
<i>FIGURA N°80 Inicialización de variables</i>	87
<i>FIGURA N°81 Diagrama de Flujo para un punto relativo</i>	88
<i>FIGURA N°82 Cinemática Inversa para el primer cuadrante</i>	89
<i>FIGURA N°83 Diagrama de Flujo de un punto en el primer cuadrante</i>	92
<i>FIGURA N°84 Cinemática Inversa para el segundo cuadrante</i>	92
<i>FIGURA N°85 Diagrama de Flujo de un punto en el segundo cuadrante</i>	95
<i>FIGURA N°86 Cinemática Inversa para el tercer cuadrante</i>	96
<i>FIGURA N°87 Diagrama de Flujo de un punto en el tercer cuadrante</i>	99
<i>FIGURA N°88 Cinemática Inversa para el cuarto cuadrante</i>	100
<i>FIGURA N°89 Diagrama de Flujo de un punto en el cuarto cuadrante</i>	103
<i>FIGURA N°90 Cálculo del ángulo del punto 1</i>	104
<i>FIGURA N°91 Desarrollo del cálculo del ángulo en Labview $x \neq 0$</i>	104
<i>FIGURA N°92 Desarrollo del cálculo del ángulo en Labview $x \neq 0$</i>	105
<i>FIGURA N°93 Desarrollo del cálculo del ángulo en Labview $x = 0$</i>	106
<i>FIGURA N°94 Cálculo de la cinemática inversa del punto 1</i>	107
<i>FIGURA N°95 Cálculo de la cinemática inversa en Labview</i>	107
<i>FIGURA N°96 Cálculo de la ecuación de rotación del punto 1 y punto 2</i>	108
<i>FIGURA N°97 Cálculo de la ecuación de rotación en Labview</i>	108

<i>FIGURA N°98 Cálculo del ángulo del punto 2</i>	109
<i>FIGURA N°99 Cálculo del ángulo en Labview</i>	109
<i>FIGURA N°100 Cálculo de la cinemática inversa del punto 2</i>	110
<i>FIGURA N°101 Cálculo de la cinemática inversa en Labview</i>	110
<i>FIGURA N°102 Conversión de Distancia Lineal a pulsos</i>	112
<i>FIGURA N°103 Conversión de Distancia Angular a pulsos</i>	113
<i>FIGURA N°104 Ordenamiento de los datos para mantener la secuencia</i>	114
<i>FIGURA N°105 Diagrama de Flujo de envío de datos al microcontrolador</i>	115
<i>FIGURA N°106 Envío del ángulo en Labview</i>	116
<i>FIGURA N°107 Envío de distancia en Labview</i>	116
<i>FIGURA N°108 Diagrama del sistema de adquisición de datos</i>	120
<i>FIGURA N°109 Componentes de Sistema de Adquisición de Datos</i>	121
<i>FIGURA N°110 Sistema de Adquisición</i>	122
<i>FIGURA N°111 Diagrama de bloques de conexión</i>	122
<i>FIGURA N°112 Lógica del programa tacómetro Digital</i>	123
<i>FIGURA N°113 Segmento de Inicialización de variables</i>	124
<i>FIGURA N°114 Segmento de Configuración</i>	124
<i>FIGURA N°115 Segmento de Lógica</i>	124
<i>FIGURA N°116 Diagrama de flujo del programa tacómetro digital</i>	125
<i>FIGURA N°117 Diagrama de envío de datos</i>	126
<i>FIGURA N°118 Diagrama de transformación de pulsos a velocidad</i>	127
<i>FIGURA N°119 Filtro pasa bajo de 1er orden</i>	128
<i>FIGURA N°120 Filtro del programa arduino</i>	129
<i>FIGURA N°121 Cargando el programa en la tarjeta tacómetro</i>	130
<i>FIGURA N°122 Programa de LIFA_Base</i>	130
<i>FIGURA N°123 Carga de programa Lifa_Base en Arduino</i>	131
<i>FIGURA N°124 Panel de Algoritmo de adquisición de datos</i>	131
<i>FIGURA N°125 Panel de Controles de adquisición de datos</i>	132
<i>FIGURA N°126 Diagrama de Flujo del programa Adquisición de datos</i>	133
<i>FIGURA N°127 Panel de Controles de adquisición de datos</i>	135
<i>FIGURA N°128 Registros de Datos Adquiridos</i>	135
<i>FIGURA N°129 Programa de Identification</i>	136
<i>FIGURA N°130 Segmento de carga de datos</i>	137

<i>FIGURA N°131 Ploteo de Datos</i>	137
<i>FIGURA N°132 Segmento de captura</i>	137
<i>FIGURA N°133 Ploteo de Datos</i>	137
<i>FIGURA N°134 Comando para abrir el toolbox</i>	138
<i>FIGURA N°135 Toolbox de identificación</i>	138
<i>FIGURA N°136 Process Models</i>	139
<i>FIGURA N°137 Ploteo de resultados</i>	139
<i>FIGURA N°138 Función de transferencia Motor velocidad</i>	139
<i>FIGURA N°139 Ploteo de la función de transferencia</i>	139
<i>FIGURA N°140 Función de transferencia Motor posición</i>	140
<i>FIGURA N°141 Ploteo de la función de transferencia</i>	140
<i>FIGURA N°142 Programa de Control</i>	142
<i>FIGURA N°143 Comando pidtool</i>	142
<i>FIGURA N°144 Diseño del controlador</i>	142
<i>FIGURA N°145 Ploteo de la función de transferencia</i>	142
<i>FIGURA N°146 Diseño del controlador</i>	143
<i>FIGURA N°147 Función de transferencia del controlador y del sistema</i>	143
<i>FIGURA N°148 Respuesta del Sistema Controlado en tiempo discreto</i>	143
<i>FIGURA N°149 Diagrama de bloques de un sistema controlado</i>	144
<i>FIGURA N°150 Programa de Aplicación en LabView</i>	146
<i>FIGURA N°151 Programa de Aplicación en LabView Diagrama lógico</i>	146
<i>FIGURA N°152 Señal del Setpoint y Respuesta del Sistema</i>	147
<i>FIGURA N°153 Inserción de los motores</i>	147
<i>FIGURA N°154 Inserción de las llantas</i>	148
<i>FIGURA N°155 Inserción de la Tarjeta de control</i>	148
<i>FIGURA N°156 Inserción de la Tarjeta de potencia</i>	149
<i>FIGURA N°157 Inserción de las baterías</i>	149
<i>FIGURA N°158 Implementación de los componentes básicos</i>	150
<i>FIGURA N°159 Inserción de la carrocería</i>	150
<i>FIGURA N°160 Programa de control para el Motor 1 y Motor 2 respectivamente</i>	151
<i>FIGURA N°161 Diagrama de flujo del controlador PD</i>	152
<i>FIGURA N°162 Carga del programa al microcontroladores</i>	153

<i>FIGURA N°163 Recorrido lineal de 0 a 1 metro</i>	155
<i>FIGURA N°164 Recorrido lineal de 0 a 2 metros</i>	156
<i>FIGURA N°165 Recorrido lineal en reversa</i>	156
<i>FIGURA N°166 Giro de 360°</i>	157
<i>FIGURA N°167 Calculo de la potencia</i>	158
<i>FIGURA N°168 Pruebas de distancia lineales a 1.00 metro</i>	160
<i>FIGURA N°169 Pruebas de distancia lineales a 2.00 metros</i>	160
<i>FIGURA N°170 Panel de Control para Pruebas de distancia lineales</i>	161
<i>FIGURA N°171 Pruebas de distancia Angular a 0 Grados</i>	161
<i>FIGURA N°172 Pruebas de distancia Angular a 90 Grados en sentido</i>	162
<i>Horario</i>	
<i>FIGURA N°173 Panel de Control para Pruebas de distancia Angulares</i>	162
<i>FIGURA N°174 Pruebas de distancia Angular</i>	163
<i>FIGURA N°175 Envió de datos para una trayectoria cuadrada</i>	163
<i>FIGURA N°176 Desplazamiento para una trayectoria cuadrada</i>	164
<i>FIGURA N°177 Desplazamiento para una trayectoria cuadrada</i>	164
<i>FIGURA N°178 Envió de datos para una trayectoria triangular</i>	165
<i>FIGURA N°179 Desplazamiento para una trayectoria triangular</i>	165
<i>FIGURA N°180 Desplazamiento para una trayectoria triangular</i>	165
<i>FIGURA N°181 Desplazamiento para una trayectoria triangular</i>	166
<i>FIGURA N°182 Desplazamiento para una trayectoria triangular</i>	166

ÍNDICE DE TABLAS

<i>TABLA N°1 Características del motor</i>	<i>57</i>
<i>TABLA N°2 Bornes Puente H y Microcontrolador</i>	<i>58</i>
<i>TABLA N°3 Bornes Puente H y Motor</i>	<i>59</i>
<i>TABLA N°4 Especificaciones chip L298</i>	<i>60</i>
<i>TABLA N°5 Comparación L298 y Motor</i>	<i>60</i>
<i>TABLA N°6 Bornes tacómetro - Arduino</i>	<i>63</i>
<i>TABLA N°7 Bornes tacómetro - Puente H</i>	<i>64</i>
<i>TABLA N°8 Bornes tacómetro – Baterías</i>	<i>64</i>
<i>TABLA N°9 Bornes tacómetro – Motor</i>	<i>65</i>
<i>TABLA N°10 Comparación voltaje, frecuencia y Periodo</i>	<i>65</i>
<i>TABLA N°11 Bornes Fuente de control – Batería 12v</i>	<i>68</i>
<i>TABLA N°12 Bornes Fuente de Potencia – Batería 12v</i>	<i>68</i>
<i>TABLA N°13 Bornes Tarjeta de conexión – Arduino</i>	<i>69</i>
<i>TABLA N°14 Bornes Tarjeta de conexión – Arduino Motor 1</i>	<i>69</i>
<i>TABLA N°15 Bornes Tarjeta de conexión – Arduino Motor 2</i>	<i>70</i>
<i>TABLA N°16 Bornes Tarjeta de conexión – Motor 1</i>	<i>70</i>
<i>TABLA N°17 Bornes Tarjeta de conexión – Motor 2</i>	<i>71</i>
<i>TABLA N°18 Bornes Tarjeta de conexión – Bluetooth</i>	<i>71</i>
<i>TABLA N°19 Bornes Tarjeta de conexión – Entrada Puente H</i>	<i>71</i>
<i>TABLA N°20 Bornes Tarjeta de conexión – Salida Puente H</i>	<i>72</i>
<i>TABLA N°21 Bluetooth valores de fábrica</i>	<i>74</i>

RESUMEN

El presente trabajo de tesis consiste principalmente en el diseño e implementación de un robot móvil, controlando su posición y trayectoria en el plano cartesiano, mediante las coordenadas definidas por el usuario y calculada algorítmicamente por el software Labview. El robot móvil tiene la capacidad de desarrollar y desplazarse de una manera segura, óptima y eficiente.

El robot móvil cuenta con dos motores DC en una configuración diferencial, para el desplazamiento y son controlados mediante un controlador proporcional derivativo que miden los pulsos entregados por los econders incrementales. Las señales independientes generadas por cada motor son procesadas por sus respectivos microcontroladores Arduino (Slave) quien ejecuta el controlador proporcional derivativo. Los dos Arduinos (Slave) se comunican por el protocolo I2C al Arduino (Master) que nos permite programar el LIFA y comunicarnos con la PC de manera inalámbrica mediante el modulo Bluetooth.

El robot móvil posee un batería interna y no necesita de cables conectados a la computadora para su alimentación y transferencia de datos. Las tarjetas diseñadas en el software Eagle y fabricadas en fibra de vidrio nos permiten conectar los microcontroladores sin la necesidad de muchos cables, el Puente H servirá para el aislamiento y alimentación de los motores al circuito de control.

Este proyecto servirá para incentivar la robótica, añadir distintos sensores como por ejemplo el sensor ultrasónico que permitiría poner en pause el robot mientras se cruza un obstáculo en su camino y también mejorar las técnicas de control para estudios posteriores.

ABSTRACT

The present thesis consists in the design and implementation of a mobile robot, controlling its position and trajectory in the cartesian plane, through the coordinates defined by the user and calculated algorithmically by Labview software. The mobile robot has the ability to develop and move in a safe, optimal and efficient way.

The mobile robot has two DC motors in a differential configuration, for the displacement and they are controlled through a proportional derivative controller from the pulses delivered by the incremental encoders. The independent signals generated by each motor are processed by their respective Arduino (Slave) microcontrollers that execute the proportional derivative controller. The two Arduinos (Slave) communicate through the I2C protocol to the Arduino (Master) that allow us to program the LIFA and communicate with the PC wirelessly through the Bluetooth module.

The mobile robot has an internal battery and does not need cables connected to the computer for its power and transfer information. Boards designed in Eagle software and made of fiberglass allow us to connect microcontrollers without the need for many cables, the H-Bridge will serve for the insulation and powering of the motors to the control circuit.

This project serves to encourage robotics, adds sensors as the ultrasonic sensors that allow to pause the robot while crossing an obstacle in its path and also improves control techniques for future studies.

CAPÍTULO I

PLANTEAMIENTO DE LA INVESTIGACIÓN

1.1. Identificación del problema

Muchas industrias automatizadas no cuentan con un sistema de transporte autónomo para sus productos, debido a la poca información o porque son sistemas muy complejos y caros, Siendo las mismas personas la que trasladan y distribuyen sus productos. Teniendo que ocupar un personal para cada traslado de producto.

Existen industrias que cuentan con sistemas robotizados de control manual, pero siempre tiene que ser operado por un personal en todo el tiempo que demande su recorrido. La desventaja de estos sistemas es que dependen de la precisión humana, teniendo que entrenar al personal para que sean diestros en el manejo del robot. También teniendo que malgastar un personal por cada unidad robótica que se esté utilizando.

En Ciertos ambientes industriales es complicado implementar un sistema de transporte, ya sea mediante Fajas transportadoras. Ya que requeriría de modificación del área a donde se va implementar, para adecuarlo al sistema autónomo.

1.2. Formulación del problema

El controlador Proporcional Derivativo permitirá el control del desplazamiento de un robot móvil de manera autónoma.

1.3. Objetivos de la investigación

Objetivo general:

- El objetivo del presente trabajo es Diseñar e Implementar un robot móvil para el control del desplazamiento autónomo, utilizando un controlador Proporcional Derivativo.

Objetivos específicos:

- Diseño mecánico e implementación del robot móvil.
- Diseño electrónico y programación de los microcontroladores y elección del bluetooth para la comunicación inalámbrica.
- Diseño y sintonización del controlador PD en el microcontrolador para cada uno de los motores.
- Desarrollo de la interfaz gráfica en el software Labview para la visualización de la trayectoria y el envío del cálculo de distancia y ángulos de rotación al microcontrolador.

1.4. Justificación

- Difundir el desarrollo tecnológico para el servicio del transporte de productos en un ambiente automatizado industrial
- Es necesario implementar un robot móvil autónomo que nos permita optimizar el uso eficiente del espacio para el almacenamiento de productos evitando el desorden y la aglomeración de los productos.
- Disminución de costo y esfuerzo humano.
- En nuestra facultad de ingeniería eléctrica y electrónica no existen suficientes estudios que fomenten al desarrollo de robot móvil autónomos.
- Desarrollar una plataforma donde se podrá aplicar distintas técnicas de control.

2.5. Definición de términos básico

Odometría: La Odometría es el estudio de la estimación de la posición de vehículos con ruedas durante el desplazamiento. Para realizar esta estimación se usa información sobre la rotación de las ruedas para estimar cambios en la posición a lo largo del tiempo. Este término también se usa a veces para referirse a la distancia que ha recorrido uno de estos vehículos (pudiéndose emplear otros sensores para su cálculo, como la Odometría visual).

Encoders: Un haz de luz es periódicamente interrumpido por un disco ranurado o reflejado por una superficie con contrastes brillantes y opacos. Uno o más discos están asociados con cada actuador.

Control Proporcional: Es aquel tipo de control en el cual la salida del controlador es proporcional a la magnitud del error.

Tacómetro: Instrumento para medir la velocidad de rotación de un mecanismo de la máquina al que va acoplado; generalmente, indica la velocidad en revoluciones por minuto.

Magnetómetro: Se llaman magnetómetros a los dispositivos que sirven para cuantificar en fuerza o dirección la señal magnética de una muestra.

Modelo matemático: Un modelo matemático es una descripción, en lenguaje matemático, de un objeto que existe en un universo no-matemático.

Adquisición de datos: La adquisición de datos o adquisición de señales, consiste en la toma de muestras del mundo real (sistema analógico) para generar datos que puedan ser manipulados por un ordenador u otras electrónicas (sistema digital).

Función de transferencia: La función de transferencia de un sistema lineal e invariante en el tiempo (LTI), se define como el cociente entre la transformada de Laplace de la salida y la transformada de Laplace de la entrada, bajo la suposición de que las condiciones iniciales son nulas.

Cinemática: Parte de la mecánica que trata del movimiento en sus condiciones de espacio y tiempo, sin tener en cuenta las causas que lo producen.

Reluctancia: Resistencia que un circuito ofrece al paso del flujo magnético.

CAPÍTULO III

VARIABLES E HIPÓTESIS

3.1. Definición de las variables

VARIABLES INDEPENDIENTES:

- Coordinadas para el cálculo y gráfico de la trayectoria.
- Imperfecciones del terreno de trabajo.
- Voltaje de las baterías.
- Desalineamiento del eje de las ruedas.

VARIABLES DEPENDIENTES:

- Trayectoria generada.
- Velocidad de cada rueda del robot móvil.
- Aceleración del móvil.
- Lectura del encoders del motor.
- Posición actual de robot móvil.

3.2. Operacionalización de variables

Variables Independientes	Dimensión	Indicadores
Coordinadas para el cálculo y gráfico de la trayectoria.	Centímetros	Se ingresara las coordenadas para su posterior conversión en distancias y ángulos.
Imperfecciones del terreno de trabajo.	-	No se consideran irregularidades en el terreno de desplazamiento teniendo como premisa que el producto transportado sea frágil.

Voltaje de las Baterías	Voltios	El descargue de las baterías es algo inevitable, debido a ello se eligió baterías que satisfagan toda la jornada laboral.
Desalineamiento del eje de las ruedas.	Centímetros	Como es un prototipo, no se encuentra milimétricamente alineada, es por ello que el desalineamiento contribuye al error y es bueno tenerlo en cuenta.

Variables Dependientes	Dimensión	Indicadores
Trayectoria generada.	Centímetros	Esta variable es generada por las coordenadas ingresadas y proporciona distancia y ángulos que robot móvil puede ejecutar sin problemas.
Velocidad de cada rueda del robot móvil.	Centímetros por segundo	Esta variable es generada por el voltaje aplicado a cada uno de los motores.
Aceleración del móvil.	Cm/seg^2	Esta variable es generada por la velocidad que se encuentra el motor, pero es obviada debido a que es casi instantánea.
Lectura del encoders del motor	Pulsos	Esta variable es generada por la velocidad de los motores y nos indica velocidad, posición y orientación del motor.
Posición actual de robot móvil.	Centímetros	Esta variable es calculada por la lectura de los encoders y nos permite identificar en el plano cartesiano la ubicación actual del robot móvil.

3.3. Hipótesis general e hipótesis específicas

Hipótesis General:

El robot móvil desarrollara la trayectoria generada por el software Labview de manera óptima y segura controlada mediante un controlador Proporcional Derivativo.

Hipótesis Específica:

Los microcontroladores ejecutara el algoritmo de control en tiempo real sin ninguna interferencia por el desplazamiento del robot móvil.

El software Labview calculara las trayectorias y ángulos mediante la cinemática inversa y nuestro análisis matemático, que a su vez enviara al microprocesador valores que pueda procesar.

CAPÍTULO IV

METODOLOGIA

4.1. Tipo de investigación

En este trabajo de investigación realizado se utilizo el siguiente método:
Cuantitativo Tecnológico

4.2. Diseño de la investigación

4.2.1. Estudio de las ventajas de la robotización

Hoy la mayoría de las empresas están logrando rápidos incrementos en la productividad, aprovechando las tecnologías de automatización. La tecnología del robot es confiable y aceptada hoy por la fuerza laboral, y su empleo avanza en las empresas grandes, medianas y pequeñas [n1].

Se pueden destacar cuatro ventajas principales de los sistemas robotizados: Aumento de la productividad, alta flexibilidad, excelente calidad y mejora de la seguridad.

4.2.2. Encoders efecto hall

Si bien existen numerosos desarrollos de medidores magnéticos de ángulo de giro ("encoders"), no siempre son aptos a condiciones extremas de utilización. Para aquellos requerimientos en donde se exija muy alta precisión en la medición, capacidad de hacerlo en ambos sentidos sin puesta a 0, y condiciones ambientales muy desfavorables (humedad, temperatura y suciedad), no existen medidores de confiabilidad que puedan garantizar los resultados. Frente a esta necesidad se desarrolló un sistema magnético de medición continua de ángulo de rotación que cumple con esas necesidades.

El desarrollo de sensores magnéticos, permite determinar, sin contacto, la velocidad de rotación y las posiciones a partir de la medición del campo magnético generado por un imán multipolar y un sensor magnético de efecto Hall.

FIGURA N°1 Encoder de efecto hall

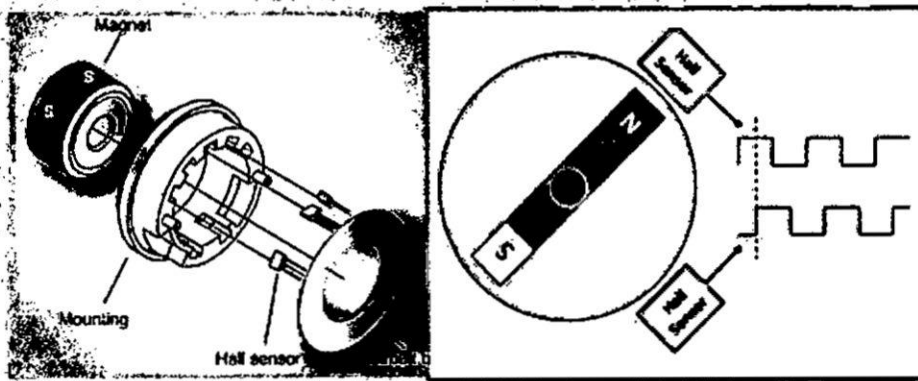


FIGURA N°1 el imán se dispone en una segunda pista que incorpora sin regularidades magnéticas y el sensor dispone de elementos Hall integrados en silicio, dispuestos en cuerpos de forma de pequeñas barritas y asociados en el mismo circuito integrado (chip) a su circuito de acondicionamiento.

La tercera generación de sensores magnéticos que consisten en una resistencia eléctrica sensible al campo magnético, compuesta por dos capas magnéticas conductoras separadas por una barrera aislante de unos nanómetros (milésimas de micra) de espesor. Este último desarrollo presenta

4.2.4. Filtros de Respuesta Infinita (IIR)

Los filtros IIR, también conocidos como Sistemas Auto-regresivos (AR), son llamados de respuesta infinita, porque el proceso de filtrado se realiza por medio de la evaluación de la ecuación de diferencias que regulan el sistema. Como la ecuación de diferencias depende de las salidas anteriores del filtro, existe una dependencia de los infinitos estados anteriores de la variable de salida a la variable de salida actual, por tal razón son llamados de Respuesta al Impulso Infinita.

$$H(z) = \frac{\sum_{k=0}^{N-1} b_k z^{-k}}{1 + \sum_{k=0}^{M-1} a_k z^{-k}} \quad \text{Ec. N}^\circ 1$$

La función de transferencia de los filtros IIR está dada por Ecuación N°1 cómo se puede observar este filtro cuenta con ceros y polos, por lo que la estabilidad del mismo no está garantizada.

De manera similar que en el caso de los filtros FIR, existen diversos métodos para diseñar filtros digitales IIR. Sin embargo, todas las técnicas más popularizadas parten de un filtro análogo que cumplan las condiciones requeridas y luego éste es convertido a filtro digital. Debido a esto discutiremos brevemente algunos aspectos de trascendencia en el diseño de filtros análogos y como éstos afectan al filtro cuando este es pasado al dominio digital.

$$y[n] = b_0 x[n] + b_1 x[n - 1] + b_2 x[n - 2] + \dots + b_M x[n - M] \\ - a_0 y[n] + a_1 y[n - 1] + a_2 y[n - 2] + \dots + a_M y[n - M]$$

Ecuación en Recursión

Mediante la ecuación en recurrencia, este filtro se define por los coeficientes de recursión. La salida en cada instante involucra además de muestras de la entrada, muestras previas de la salida.

Las constantes b_i , $i=1, \dots, M$ y a_j , $j = 1, \dots, N$ se llaman coeficientes del filtro.

El filtro queda completamente especificado con los valores de todos los coeficientes.

Los valores b_i se llaman coeficientes de realimentación (feedforward) y los valores a_j se llaman coeficientes de realimentación (backward).

El filtro es recursivo si tiene algún coeficiente de realimentación no nulo. En ese caso, es un filtro IIR. En caso contrario, no hay realimentación y el filtro es FIR, o equivalentemente, no recursivo.

El retardo máximo usado por la ecuación en recurrencia se llama orden del filtro. El orden es el máximo entre N y M.

FIGURA N°4 Filtro de primer Orden

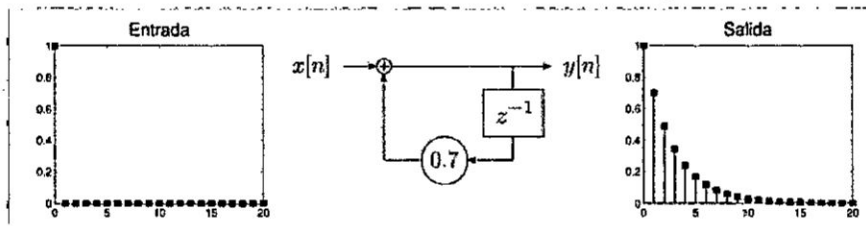


FIGURA N°4 Ejemplo de filtro IIR de primer Orden.

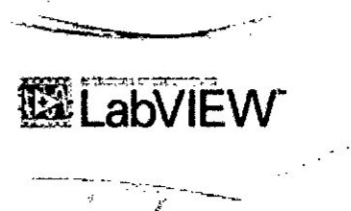
El filtro tiene realimentación entre la entrada y la salida.

En general, la respuesta al impulso de un filtro recursivo es una combinación de exponenciales y sinusoides que decrecen.

4.2.5. Interface de Labview para Arduino (LIFA)

LabView

LabView constituye un revolucionario sistema de programación gráfica para aplicaciones que involucren adquisición, control, análisis y presentación de datos. Las ventajas que proporciona el empleo de LabView se resumen en las siguientes:



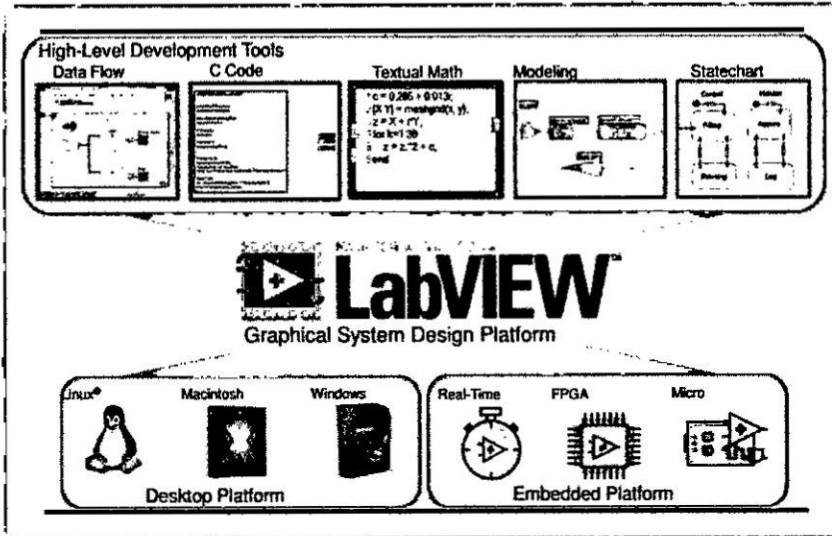
- Se reduce el tiempo de desarrollo de las aplicaciones al menos de 4 a 10 veces, ya que es muy intuitivo y fácil de aprender.
 - Da la posibilidad a los usuarios de crear soluciones completas y complejas.
 - Con un único sistema de desarrollo se integran las funciones de adquisición, análisis y presentación de datos.
 - El sistema está dotado de un compilador gráfico para lograr la máxima velocidad de ejecución posible.
 - Tiene la posibilidad de incorporar aplicaciones escritas en otros lenguajes.
- LabView es un entorno de programación destinado al desarrollo de aplicaciones, similar a los sistemas de desarrollo comerciales que utilizan el lenguaje C o BASIC. Sin embargo, LabView se diferencia de dichos programas en un importante aspecto: los citados lenguajes de programación se basan en líneas de texto para crear el código fuente del programa, mientras que LabView emplea la programación gráfica o lenguaje G para crear programas basados en diagramas de bloques.

Para el empleo de LabView no se requiere gran experiencia en programación, ya que se emplean iconos, términos e ideas familiares a científicos e ingenieros, y se apoya sobre símbolos gráficos en lugar de lenguaje escrito para construir las aplicaciones. Por ello resulta mucho más intuitivo que el resto de lenguajes de programación convencionales.

LabView posee extensas librerías de funciones y subrutinas. Además de las función es básicas de todo lenguaje de programación, LabView incluye librerías específicas para la adquisición de datos, control de instrumentación VXI, GPIB y comunicación serie, análisis presentación y guardado de datos.

Los programas desarrollados mediante LabView se denominan Instrumentos Virtuales (VIs), porque su apariencia y funcionamiento imitan los de un instrumento real. Sin embargo son análogos a las funciones creadas con los lenguajes de programación convencionales. Los VIs tienen una parte interactiva con el usuario y otra parte de código fuente, y aceptan parámetros procedentes de otros VIs.

FIGURA N°5 Plataforma LabView

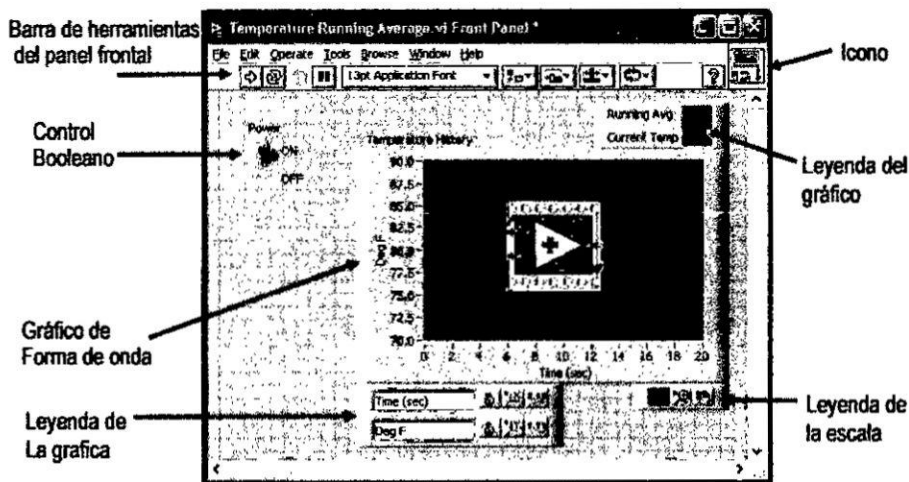


Todos los VIs tienen un panel frontal y un diagrama de bloques. Las paletas contienen las opciones que se emplean para crear y modificar los VIs. A continuación se procederá a realizar una somera descripción de estos conceptos.

Panel Frontal

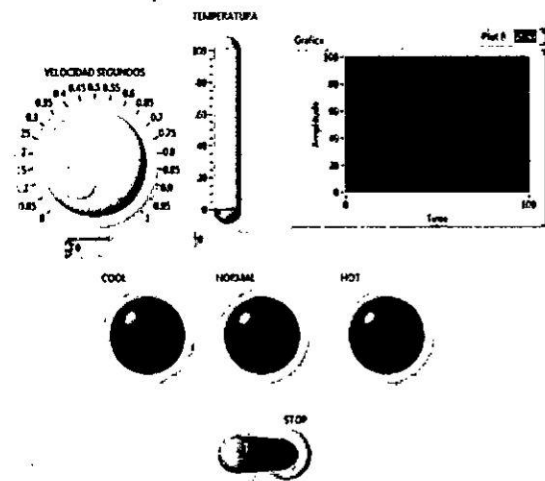
Se trata de la interfaz gráfica del VI con el usuario. Esta interfaz recoge las entradas procedentes del usuario y representa las salidas proporcionadas por el programa. Un panel frontal está formado por una serie de botones, pulsadores, potenciómetros, gráficos, etc.

FIGURA N°6 VI Panel Frontal



Esta interface gráfica que simula el panel de un instrumento real, permite la entrada y salida de datos, puede contener pulsadores, perillas, botones, gráficos y en general controles e indicadores.

Los controles son objetos que sirven para entrar datos al programa y pueden ser manipulados por el usuario. Los controles son variables de entrada.

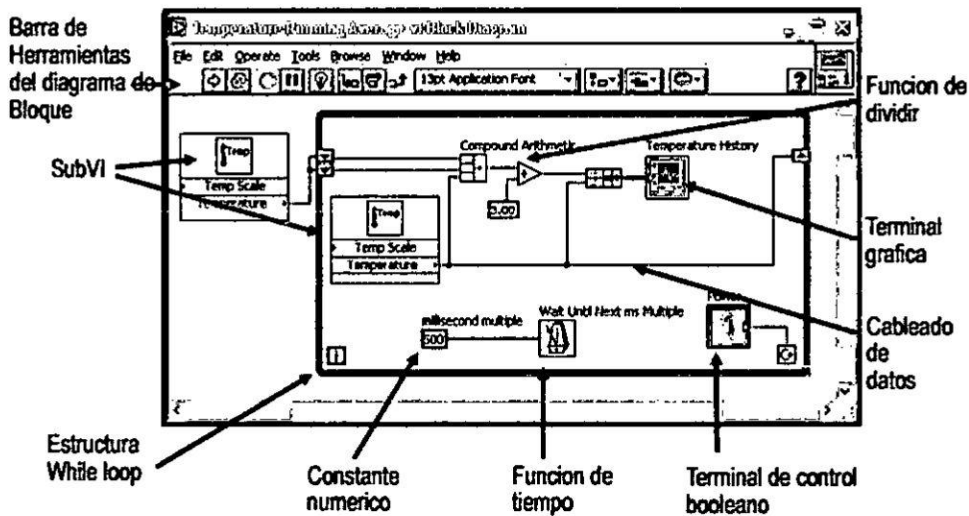


Los indicadores sirven para presentar los resultados entregados por el programa y no pueden ser manipulados por el usuario. Los indicadores son variables de salida.

Diagrama de bloques

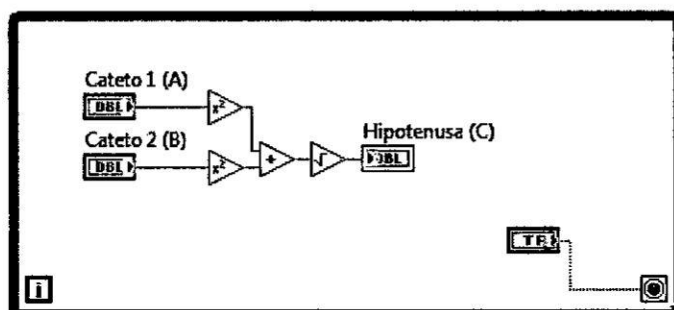
El diagrama de bloques constituye el código fuente del VI.

FIGURA N°7 VI Diagrama de Bloques



En el diagrama de bloques es donde se realiza la implementación del programa del VI para controlar o realizarcualquier procesado de las entradas y salidas que se crearon en el panel frontal.

El diagrama de bloques incluye funciones y estructuras integradas en las librerías que incorpora LabVIEW. En el lenguaje G las funciones y las estructuras son nodos elementales. Son análogos a los operadores o librerías de funciones de los lenguajes convencionales.



Los controles e indicadores que se colocaron previamente en el Panel Frontal, sematerializan en el diagrama de bloques mediante los terminales.

LIFA.

La interfaz de LabView para Arduino (LIFA) Toolkit es una herramienta gratuita que se puede descargar desde el servidor de NI (National Instruments) y que permite a los usuarios de Arduino adquirir datos del microcontrolador Arduino y procesarlos en el entorno de programación gráfica de LabView. Para utilizar la aplicación LIFA no es necesario disponer de una versión comercial de LabVIEW basta con adquirir el software LabView.

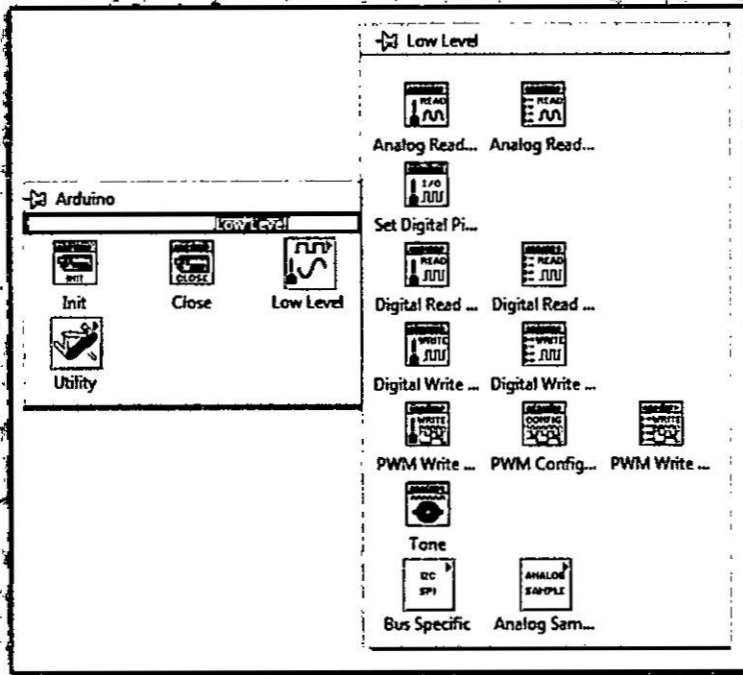
El microcontrolador Arduino es una plataforma de bajo costo de electrónica de prototipos. Con la interfaz de LabView para Arduino LIFA se puede aprovechar la potencia del entorno de programación gráfica de LabView para interactuar con Arduino en una nueva dimensión.

Bloques de herramientas del LIFA.

La paleta de herramientas Arduino VI contiene Grupos de bloques diseñados específicamente para comunicarse con el Arduino.

Estos son los bloques más usados al crear un programa.

FIGURA N°8 Bloque de herramientas Lifa

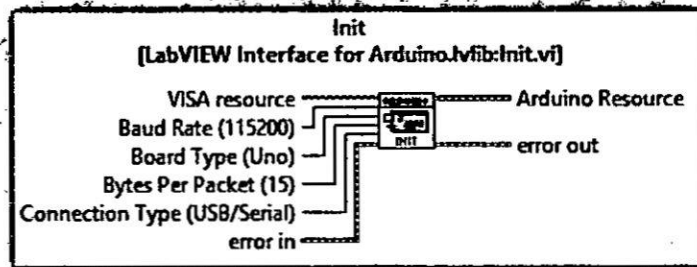


Bloques LIFA y LabView utilizados.

Init:

Inicializa la conexión para un arduino, también se puede configurar el puerto serial de comunicación.

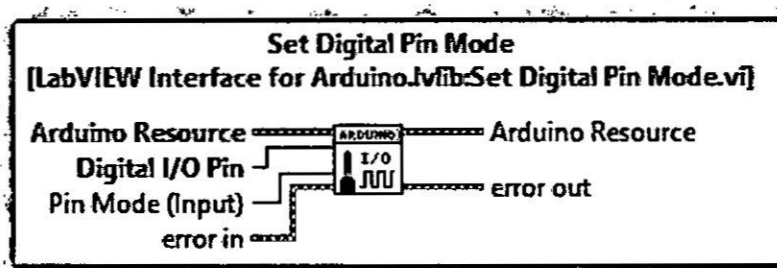
FIGURA N°9 Bloque Init



Set Digital Pin Mode:

Configuramos como entrada o salida el pin de arduino.

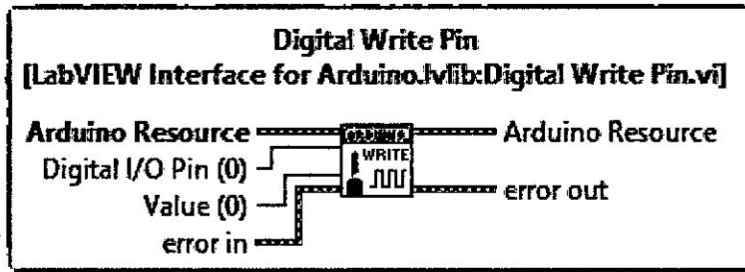
FIGURA N°10 Bloque Set Digital Pin Mode



Digital Write Pin:

Se escribe el valor específico de la salida del Pin.

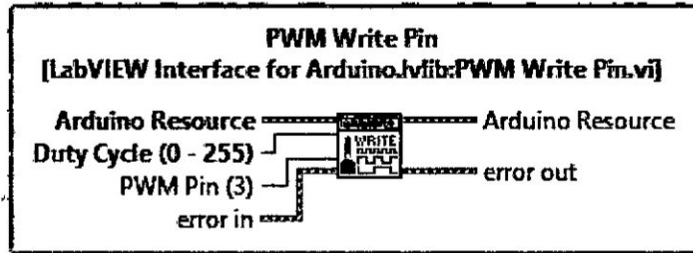
FIGURA N°11 Bloque Digital Write Pin



PWM Write Pin:

Se escribe el valor PWM específico de la salida del Pin.

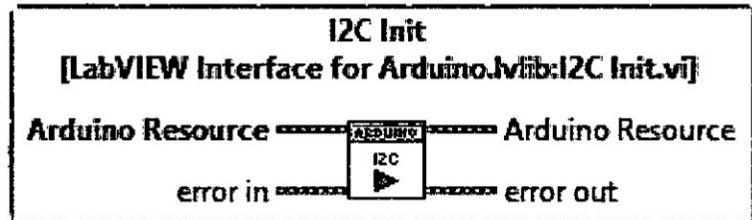
FIGURA N°12 Bloque PWM Write Pin



I2C Init:

Inicializar el bus I2C se establece como maestro.

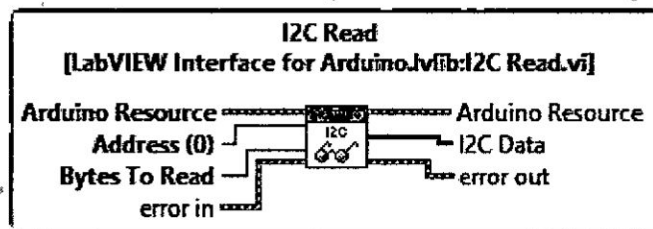
FIGURA N°13 Bloque I2C Init



I2C Read:

Leemos la data I2C desde el arduino.

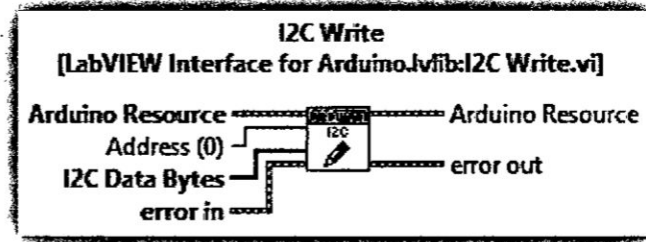
FIGURA N°14 Bloque I2C Read



I2C Write:

Escribimos la data I2c desde el arduino.

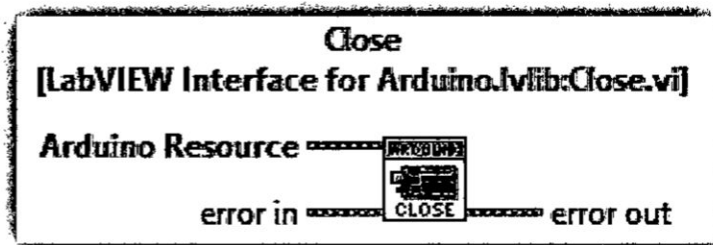
FIGURA N°15 Bloque I2C Write



Close:

Cerramos la conexión activa de arduino.

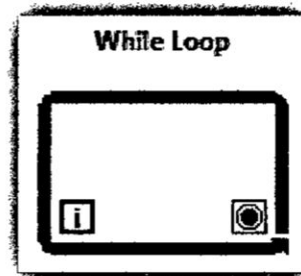
FIGURA N°16 Bloque Close



While Loop:

Repite el sub diagrama en su interior hasta que el terminal condicional, un terminal de entrada, recibe un valor booleano en particular. El valor booleano depende del comportamiento continuación de la Mientras Loop.

FIGURA N°17 Bloque While Loop

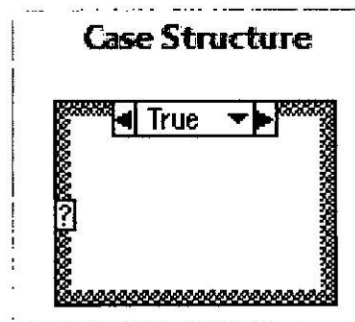


Haga clic con el terminal condicional y seleccione Detener si es verdadero o Continuar si es verdadero en el menú contextual. También puede conectar un cluster de error al terminal condicional, haga clic en el terminal, y seleccione Detener en error o Continuar mientras que el error en el menú contextual. El ciclo Mientras siempre se ejecuta al menos una vez.

Case Structure:

Tiene uno o más subdiagramas, o casos, exactamente uno de los cuales se ejecuta cuando la estructura se ejecuta. El valor por cable al terminal de selección determina cuyo caso ejecución y puede ser booleano, cadena, un entero o tipo enumerado.

FIGURA N°18 Bloque Case Structure



Haga clic en el borde de la estructura para añadir o eliminar los casos. Utilice la herramienta de Etiquetado para introducir el valor (s) en la etiqueta de selección de casos y configurar el valor (s) que maneja cada caso.

The waveform chart:

El gráfico de forma de onda es un tipo especial de indicador numérico que muestra una o más tramas de datos normalmente adquiridos a una velocidad constante. El siguiente panel frontal muestra un ejemplo de un diagrama de forma de onda.

FIGURA N°19 Bloque The waveform chart



Write To Measurement File Express VI:

Escribe los datos en archivos de texto de medición (.lvm) o archivos binarios de medición (.tdm vs .tdms).

FIGURA N°20 Bloque Write To Measurement File Express VI

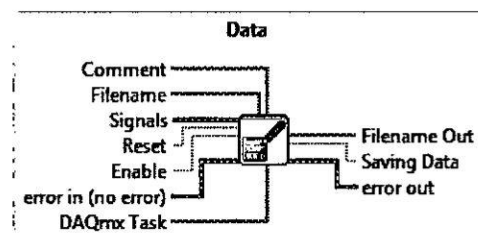
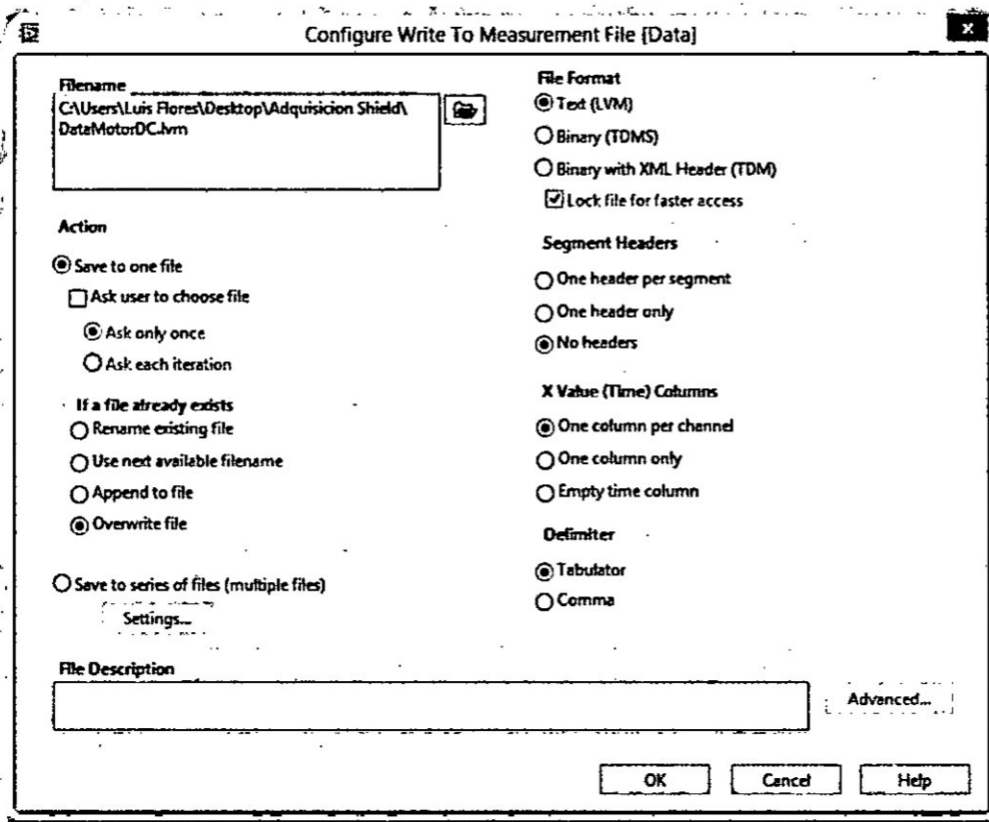


FIGURA N°21 Configuración del Bloque Write To Measurement File



Filename:

Muestra la ruta completa del archivo en el que desea escribir los datos.

Field Format.

Contiene las siguientes opciones:

Texto (LVM): establece el formato de archivo del archivo de medición basado en texto a (.lvm) y la extensión de archivo en el nombre del archivo para .lvm.

Action:

Contiene la siguiente opción:

Save to one field-Guarda todos los datos a un archivo.

If a file already exists

Contiene la siguiente opción:

Overwrite field -Reemplaza en un archivo existente en caso de reinicio es TRUE.

Segment Headers:

Contiene la siguiente opción:

No headers -no crea un encabezado en el archivo en el que LabVIEW escribe los datos. Nota Esta opción sólo está disponible cuando se selecciona la opción Texto (LVM) desde la sección Formato de archivo.

X Value Columns:

Contiene la siguiente opción:

One column per channel: crea una columna separada para cada canal de datos en tiempo genera. Esta opción incluye una columna de valores del eje x para cada columna de valores del eje y. Seleccione esta opción si usted adquiere señales de diferentes tipos o a diferentes velocidades de adquisición.

Delimiter:

Contiene las siguientes opciones:

Tab: utiliza pestañas para delimitar los campos en el archivo de texto.

Utiliza comas para delimitar los campos en el archivo de texto. Nota Estas opciones sólo están disponibles cuando se selecciona la opción Texto (LVM) desde la sección Formato de archivo.

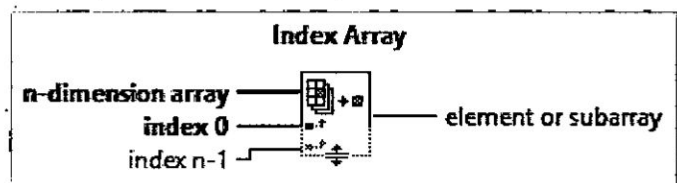
File Description:

Contiene la descripción de la .lvm, .tdm, o archivo .tdms. LabVIEW añade el texto que introduzca en este cuadro de texto para la cabecera del archivo.

Index Array:

Devuelve el elemento de la submatriz de la matriz n-dimensional en el índice.

FIGURA N°22 Bloque Index Array

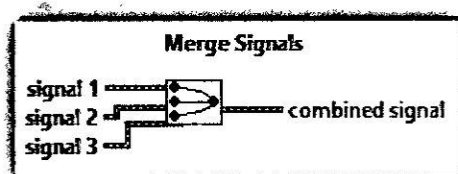


BLOQUES MATEMÁTICOS

Merge Signals;

Combina dos o más señales en una sola salida. Cambiar el tamaño de la función para añadir entradas. Esta función aparece en el diagrama de bloques de forma automática cuando se conecta una señal de salida a la rama de alambre de otra señal.

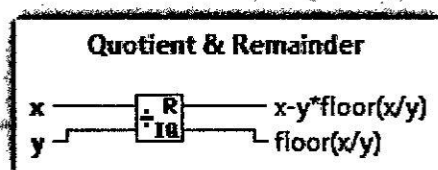
FIGURA N°23 Bloque Merge Signals



Quotient & Remainder Function:

Calcula el cociente entero y el resto de las entradas.

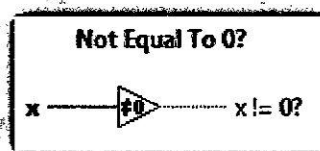
FIGURA N°24 Bloque Quotient & Remainder Function



Not Equal To 0? Function:

Devuelve VERDADERO si x no es igual a 0. De lo contrario, esta función devuelve FALSO.

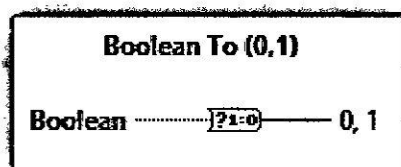
FIGURA N°25 Bloque Not Equal To 0? Function



Boolean To (0,1):

Convierte un FALSE TRUE o el valor a un número entero de 16 bits con un valor de 0 ó 1, respectivamente.

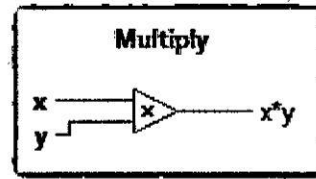
FIGURA N°26 Bloque Boolean To (0,1)



Multiply:

Devuelve el producto de las entradas.

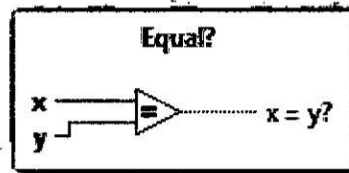
FIGURA N°27 Bloque Multiply



Equal?:

Devuelve VERDADERO si x es igual a y. De lo contrario, esta función devuelve FALSO. Puede cambiar el modo de comparación de esta función.

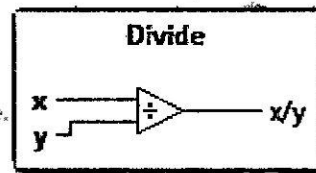
FIGURA N°28 Bloque Equal?



Divide:

Calcula el cociente de las entradas.

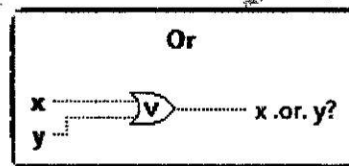
FIGURA N°29 Bloque Divide



Or;

Calcula la lógica OR de las entradas. Ambas entradas deben ser valores booleanos o numéricos. Si ambas entradas son FALSO, la función devuelve FALSO. De lo contrario, devuelve TRUE.

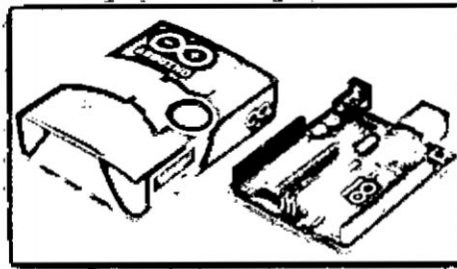
FIGURA N°30 Bloque Or



4.2.6. Arduino como unidad de control y comunicación

Arduino es una plataforma de electrónica libre para la creación de prototipos basada en software y hardware flexibles y fáciles de usar. Gracias a la plataforma Arduino podremos desarrollar proyectos electrónicos propios, con una serie de entradas, como sensores o pulsadores, podremos controlar diferentes dispositivos de salida, ya sean luces, motores o diversos aparatos electrónicos.

FIGURA N°31 Arduino board



En resumen, Arduino puede ser el cerebro de una gran variedad de proyectos electrónicos. Los programas Arduino pueden ser autónomos, si la placa una vez programada no necesita conexión a un ordenador y funciona de manera independiente. En otros casos, la placa Arduino debe estar conectada de forma permanente, vía USB, Ethernet, etc. con otra aplicación con un software específico que permita el intercambio de información entre la placa y la aplicación.

Desde una perspectiva más técnica, Arduino es una placa de hardware libre formado por un microcontrolador programable y un conjunto de pines hembra unidos internamente a las entradas y salidas de dicho microcontrolador. Con el término placa hardware, nos referimos, en concreto, al término PCB, del inglés "PrintedCircuit Board". Una PCB es una superficie fabricada a partir de un material no conductor, sobre la que se laminan pistas de un material conductor, para conectar eléctricamente a los componentes electrónicos soldados en ella. Por lo tanto, podemos describir a Arduino como una placa PCB que implementa un diseño específico de circuitería interna. En la figura 2.14 podemos ver un ejemplo de una placa Arduino.

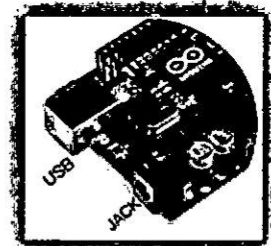
Especificaciones técnicas.

FIGURA N°32 Tabla de Características (Atmega 328)

Microcontroller	ATmega328P
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limit)	6-20V
Digital I/O Pins	14 (of which 6 provide PWM output)
PWM Digital I/O Pins	6
Analog Input Pins	6
DC Current per I/O Pin	20 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB (ATmega328P) of which 0.5 KB used by bootloader
SRAM	2 KB (ATmega328P)
EEPROM	1 KB (ATmega328P)
Clock Speed	16 MHz
LED_BUILTIN	13
Length	68.6 mm
Width	53.4 mm
Weight	25 g

Los pines de alimentación son como sigue:

Vin. El voltaje de entrada a la placa Arduino / Genuino cuando se trata de utilizar una fuente de alimentación externa (en contraposición a 5 voltios de la conexión USB o de otra fuente de alimentación regulada). Se puede suministrar tensión a través de este pin, o, si el suministro de tensión a través de la toma de alimentación, acceder a él a través de este pin.



5V. This pin como salida una 5V regulada del regulador en el tablero. El tablero puede ser alimentado ya sea desde el conector de alimentación de CC (7 - 12 V), el conector USB (5V), o por el pin VIN del tablero (7-12V). El suministro de tensión a través de los pasadores de 5V o 3.3V no pasa por el regulador, y puede dañar la placa. No aconsejamos ella.

3V3. Un suministro de 3,3 voltios generada por el regulador a bordo. Consumo de corriente máximo es de 50 mA.

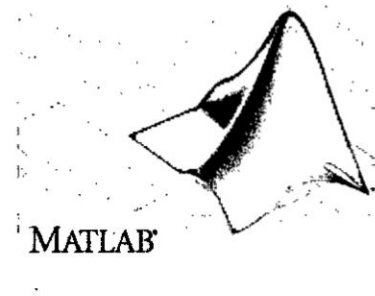
GND. Las patillas de tierra.

I2C y Arduino Arduino cuenta la posibilidad de poder manejar comunicaciones I2C. Según el modelo de Arduino que tengamos, los pines de SDA y SCL, varían, En el caso del Arduino UNO los pines son: para SDA es el pin A4, y el SCL es el pin A5.

4.2.7. Toolbox de Matlab para la identificación y control

Matlab

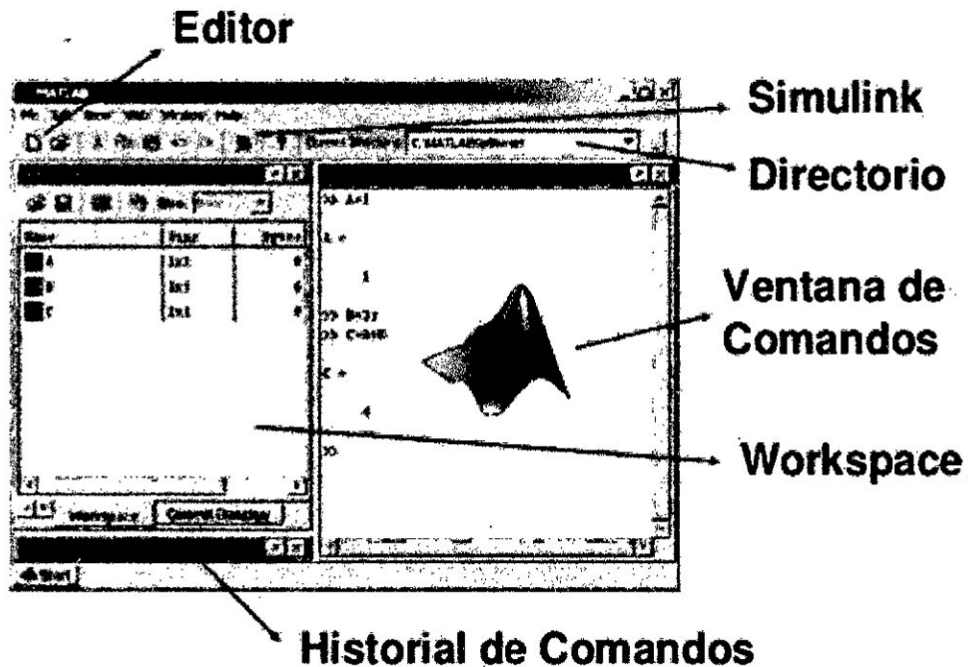
Matlab es un programa interactivo para computación numérica y visualización de datos. Es ampliamente usado por Ingenieros de Control en el análisis y diseño, posee además una extraordinaria versatilidad y capacidad para resolver problemas en matemática aplicada, física, química, ingeniería a, finanzas y muchas otras aplicaciones.



Está basado en un sofisticado software de matrices para el análisis de sistemas de ecuaciones. Permite resolver complicados problemas numéricos sin necesidad de escribir un programa.

MATLAB es un entorno de computación y desarrollo de aplicaciones totalmente integrado orientado para llevar a cabo proyectos en donde se encuentren implicados elevados cálculos matemáticos y la visualización gráfica de los mismos.

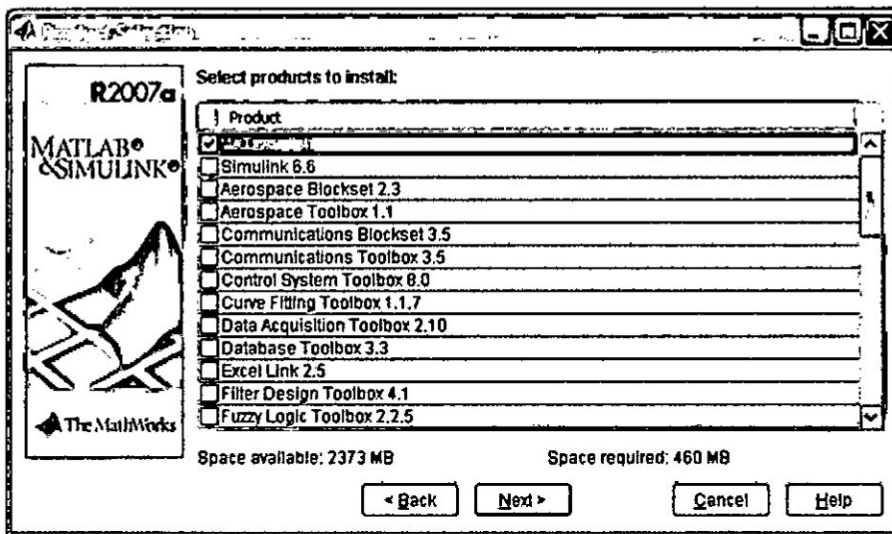
FIGURA N°34 Entorno de trabajo de Matlab



Toolboxes de Matlab.

MATLAB dispone también en la actualidad de un amplio abanico de programas de apoyo especializado, denominado Toolbox, que extienden significativamente el número de funciones incorporadas en el programa principal. Estos Toolbox cubren en la actualidad prácticamente casi todas las áreas principales en el mundo de la ingeniería y la simulación, destacando entre ellos el 'toolbox' de proceso de imágenes, señal, control robusto, estadística, análisis financiero, matemáticas simbólicas, redes neurales, lógica difusa, identificación de sistemas, simulación de sistemas dinámicos, etc.

FIGURA N°35 Tabla de Toolbox Matlab



¿Qué es la identificación del sistema?

Identificación del sistema le permite construir modelos matemáticos de una dinámica sistema basado en datos medidos.

La identificación se desarrolla esencialmente mediante el ajuste de los parámetros dentro de un modelo dado hasta su salida coincide así como es posible con la salida medida.

Modelos matemáticos más comunes, son los que se aplican a modelos muy generales. La mayoría de los modelos comunes son ecuaciones de diferencias, tales como modelos ARX y ARMAX, así como todos los tipos de modelos de espacio de estado lineales.

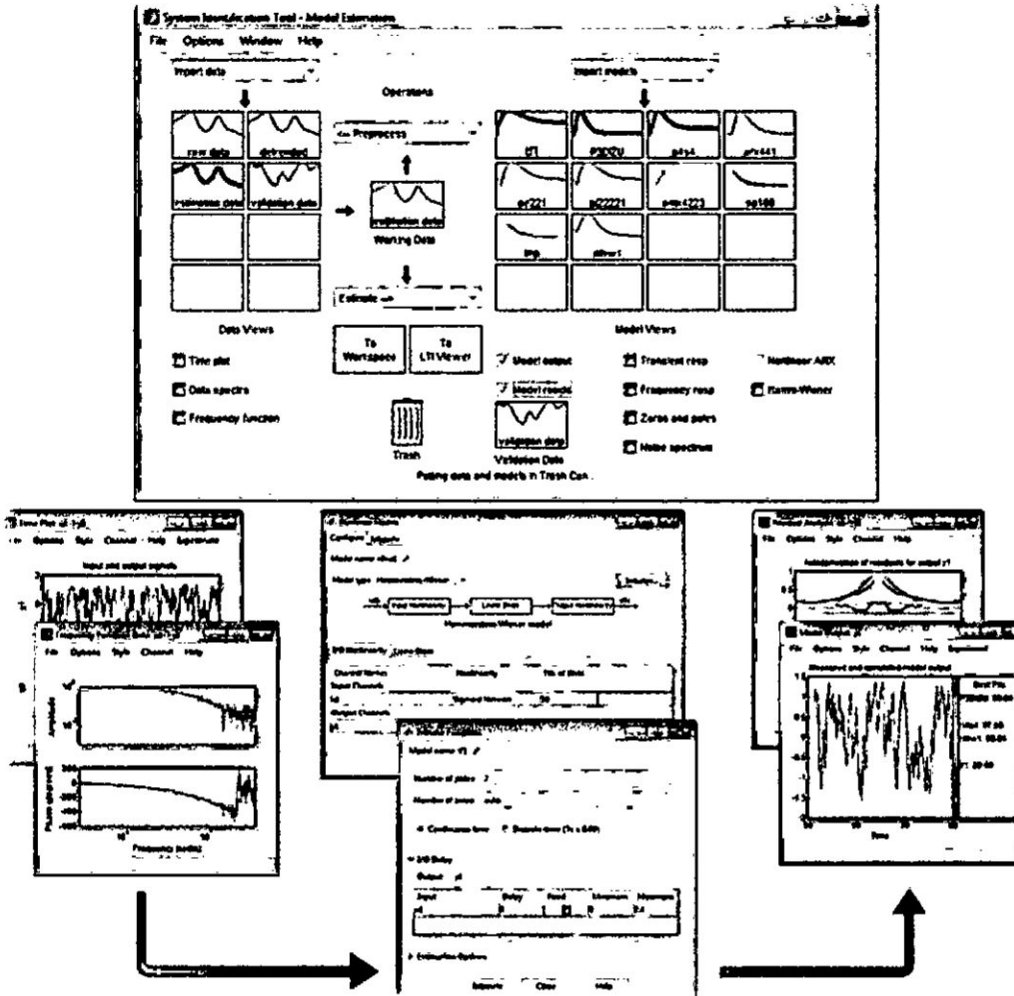
System Identification Toolbox.

El System Identification Toolbox es para la construcción de modelos precisos, simplificados de sistemas complejos a partir de series cronológicas de datos ruidosos.

Proporciona herramientas para crear modelos matemáticos de sistemas dinámicos basados en los datos de entrada / salida observados. La caja de herramientas cuenta con una gráfica de usuario flexible interfaz que ayuda en la organización de los datos y modelos. La identificación técnicas previstas con

esta caja de herramientas son útiles para aplicaciones que van desde el diseño de sistemas de control y procesamiento de señales para análisis de series de tiempo y el análisis de vibraciones.

FIGURA N°36 Toolbox de identificación



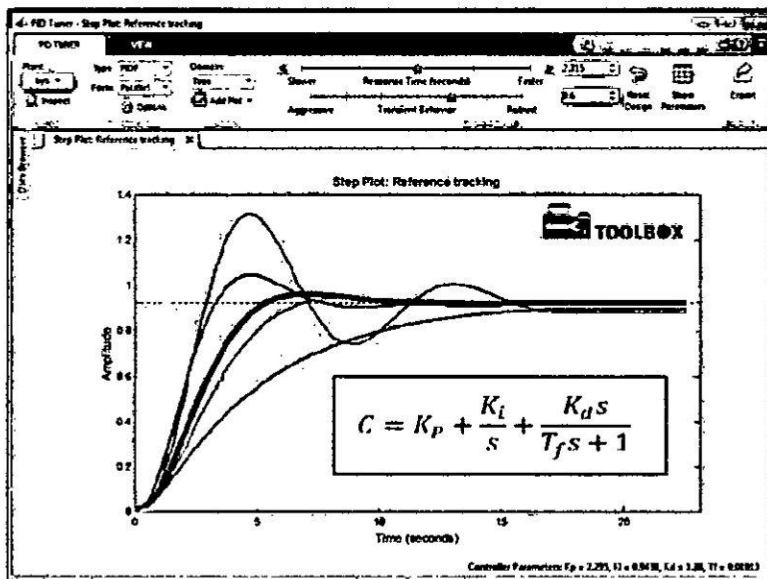
Control System Toolbox.

MATLAB tiene una rica colección de funciones de utilidad inmediata para el ingeniero de control o teórico del sistema. Aritmética compleja, valores propios, búsqueda de raíz, inversión de la matriz, y FFT son sólo algunos ejemplos de importantes herramientas numéricas de MATLAB. Más en general, el álgebra de MATLAB lineal, el cálculo de la matriz, y las capacidades de análisis numérico proporcionan una base fiable para la ingeniería del sistema de control, así como muchas otras disciplinas.

El sistema de control de la caja de herramientas se basa en los fundamentos de MATLAB para proporcionar funciones diseñadas para la ingeniería de control. El sistema de control de la caja de herramientas es una colección de algoritmos, escrita sobre todo como archivos M, que implementa el diseño del sistema de control común, análisis y técnicas de modelado. Interfaces gráficas de usuario convenientes (de interfaz gráfica de usuario) simplificar las tareas típicas de ingeniería de control.

Los sistemas de control pueden ser modelados como funciones de transferencia, en cero polos-ganancia, o en forma de espacio de estado, lo que permite utilizar ambas técnicas de control clásicas y modernas. Se puede manipular tanto los sistemas de tiempo discreto y continuo y; se proporcionan las conversiones entre diferentes representaciones de modelos.

FIGURA N°37 Toolbox de Control



4.2.8. Tipos de locomoción

En cuanto a la locomoción de robots, existen varios modos de moverse sobre una superficie sólida, las más comunes son ruedas, y piernas. Los vehículos con ruedas son la solución más sencilla para conseguir la movilidad en terrenos suficientemente duros, permitiendo velocidades relativamente altas.

Los robots móviles emplean diferentes tipos de locomoción mediante ruedas que les confiere características y propiedades diferentes respecto a la eficiencia energética, dimensiones, cargas útiles y maniobrabilidad.

Las consideraciones que tenemos que tener en cuenta para el diseño son:

- Maniobrabilidad.
- Controlabilidad.
- Tracción.
- Estabilidad.
- Eficiencia.
- Mantenimiento.

Existen varios sistemas de locomoción para robots móviles, a continuación se presentaran los más comunes [n2].

a. Ackerman

b. Tracción omnidireccional

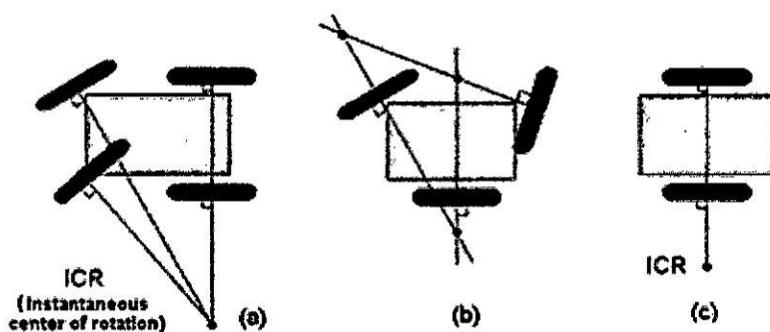
c. Tracción diferencial

Para comprender los tipos de locomoción es necesario conocer algunos conceptos previos:

- Centro Instantáneo de Rotación (CIR) o Centro Instantáneo de Curvatura (CIC); es el punto de intersección de todos los ejes de las ruedas.

Entre los Tipos de Locomoción tenemos:

FIGURA N°38 Tipos de locomoción



- Robots Holonómicos: Son capaces de cambiar de sitio en cualquier dirección y en direcciones aleatorias.
- Robots no Holonómicos: El movimiento se produce solo por dos desplazamientos, y el vehículo posee sólo dos grados de libertad.

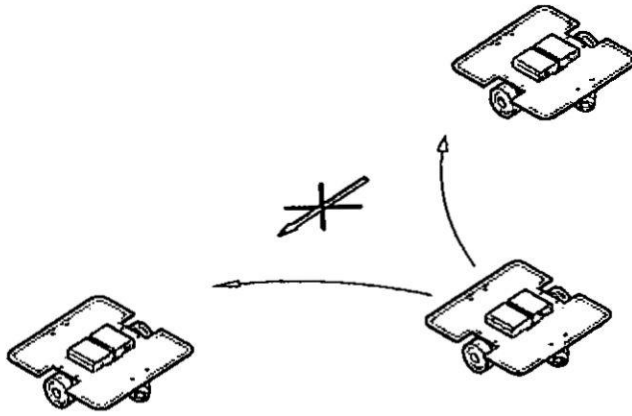
Desde un punto de vista matemático puede considerarse que un vehículo es no Holonómico cuando sus ecuaciones de restricción no son integrables, por lo cual el análisis de la cinemática, dinámica y el control son más complejos.

Debido a la consideración anterior, en sistemas que poseen restricciones no holonómicas, el número de coordenadas que describa su configuración siempre es más grande que el número de grados de libertad que poseen ellos. Para determinar si un robot móvil está bajo condiciones no holonómicas, se debe satisfacer la expresión (Ec. N°2) para la ecuación de restricciones.

$$\cos\theta dy - \sin\theta dx = 0 \quad \text{Ec. N°2}$$

Donde θ es el ángulo de orientación de la plataforma en el espacio, dx y dy las variaciones de la posiciones de la plataforma en x y y, respectivamente [n3].

FIGURA N°39 Restricción no holonómicas



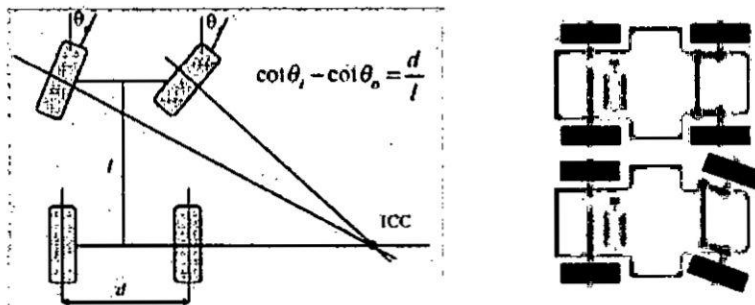
En la FIGURA N°39 se observa que el robot puede moverse instantáneamente adelante o atrás pero no lateralmente por el deslizamiento de las ruedas

a. Ackerman

El modelo Ackerman, o tipo coche es usado en vehículos con cuatro ruedas convencionales, dos de ellas, usadas para la tracción, se montan en la parte posterior en forma paralela al chasis principal del vehículo. Las ruedas delanteras son usadas para el direccionamiento, cada uno presenta dos ángulos de giro distintos, generando problemas cuando se realiza el control. En muchas ocasiones se unifican los ángulos de direccionamiento en uno solo, sin embargo ambas trayectorias no serían paralelas y por lo tanto las ruedas deslizarían en las curvas.

Su mayor problema es la limitación de maniobrabilidad.

FIGURA N°40 Sistema de Locomoción Tipo Ackerman

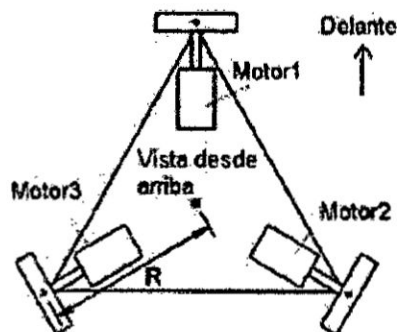


En la FIGURA N°40 se observa un sistema de locomoción tipo Ackerman donde un sistema simple de 4 barras controla la dirección.

b. Tracción omnidireccional

Este sistema de tracción se basa en la utilización de tres ruedas directrices y motrices. Esta configuración posee tres grados de libertad, por lo que puede realizar cualquier movimiento, y posicionarse en cualquier posición en cualquier orientación, no presenta limitaciones cinemáticas.

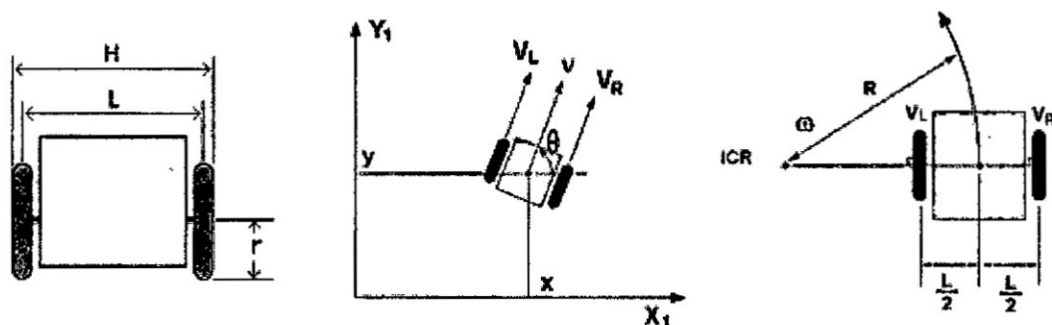
FIGURA N°41 Sistema de Locomoción Tracción omnidireccional



c. Tracción diferencial

Este tipo de direccionamiento viene dado por la diferencia de velocidades de dos ruedas laterales, estas son montadas en un único eje cada una independientemente propulsada y controlada, ambas proporcionando tracción y direccionamiento. Es un sistema muy útil si consideramos su capacidad de movimiento, ya que presenta la posibilidad de cambiar de orientación sin movimientos de traslación (giros sobre su propio eje). Adicionalmente, existen una o más ruedas de soporte, esta configuración es la más frecuente para robots de pequeño tamaño.

FIGURA N°42 Sistema de Locomoción Tracción diferencial



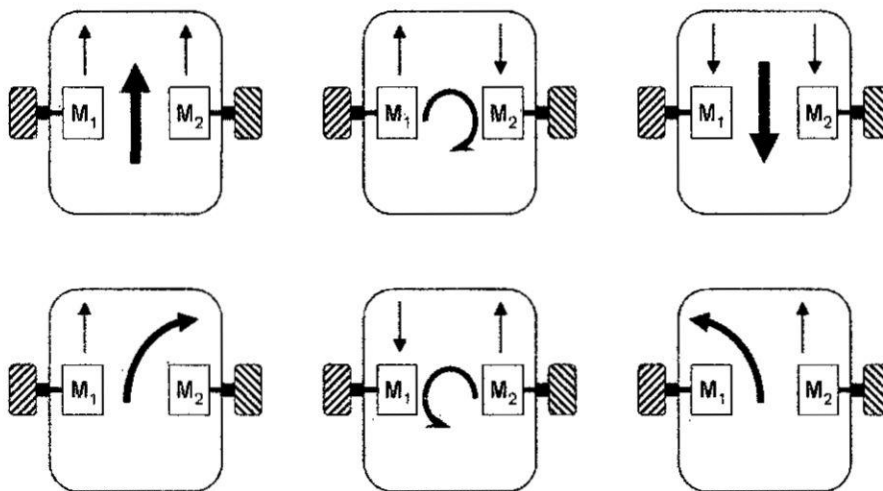
En la *FIGURA N°42* se observa un sistema de tracción diferencial donde el cambio de dirección se realiza modificando la velocidad relativa de las ruedas izquierda y derecha.

ELECCION DEL TIPO DE LOCOMOCIÓN

La rigidez del eje de las ruedas del robot limita el tipo de movimiento que puede efectuar. Se eligió un sistema que sea fácil de implementar, barato y que posea un diseño simple. Características esenciales del sistema de Tracción Diferencial.

El sistema de Tracción Diferencial, apenas son posibles las rotaciones alrededor del punto central del eje de ruedas y las traslaciones en la dirección longitudinal del robot (ver *FIGURA N°43*) Sin embargo, combinando estos movimientos, es posible llevar las rotaciones del robot alrededor de cualquier punto situado en la línea que contiene el eje de ruedas, llamado Centro de Rotación Instantáneo (IRC).

FIGURA N°43 Giros de plataforma diferencial



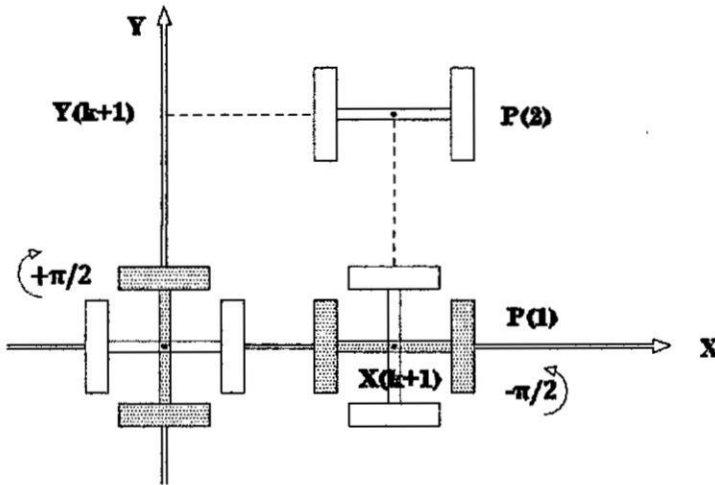
En la *FIGURA N°43* se observa un sistema de tracción diferencial donde es un sistema fácil de implementar, barato y posee un diseño simple.

4.2.9. Modelamiento Cinemático Directo e Inverso del Robot móvil

ELECCIÓN DEL TIPO DE TRAYECTORIA

Se eligió una el tipo de trayectoria que sea utilizado en un entorno cotidiano y controlado, por ello se analizó una trayectoria de un movimiento de rotación seguido por una traslación.

FIGURA N°44 Tipo de Trayectoria Analizada

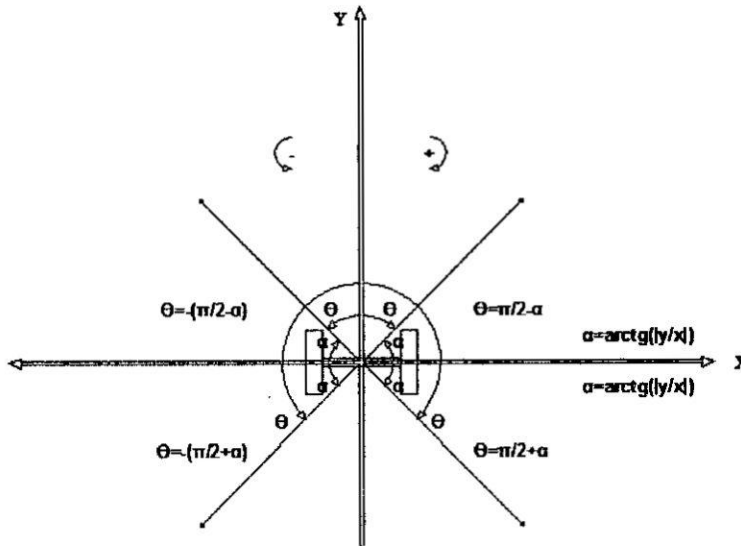


En la FIGURA N°44 se observa un movimiento de rotación de 90° en sentido Horario (Signo positivo) y luego desplazamiento y a continuación un movimiento de rotación de 90° Antihorario (Signo negativo) y luego desplazamiento.

CINEMÁTICA DIRECTA

La cinemática es el estudio de la matemática del movimiento, sin tener en cuenta los efectos de las fuerzas sobre él; trata, por lo tanto, de las relaciones geométricas que gobiernan el sistema, y de la relación entre parámetros de control y comportamiento de sistemas en el espacio estado.

FIGURA N°45 Orientación del robot y sentido de giro



En la *FIGURA N°45* se muestra que el robot en su estado inicial se encuentra frente al eje Y, cuando gira en sentido Horario será tomado como un ángulo de signo positivo, mientras que cuando gira en sentido Antihorario se tomará como un ángulo de signo negativo.

Calculo ángulo:

Donde: $\alpha = \arctg\left(\left|\frac{y}{x}\right|\right)$

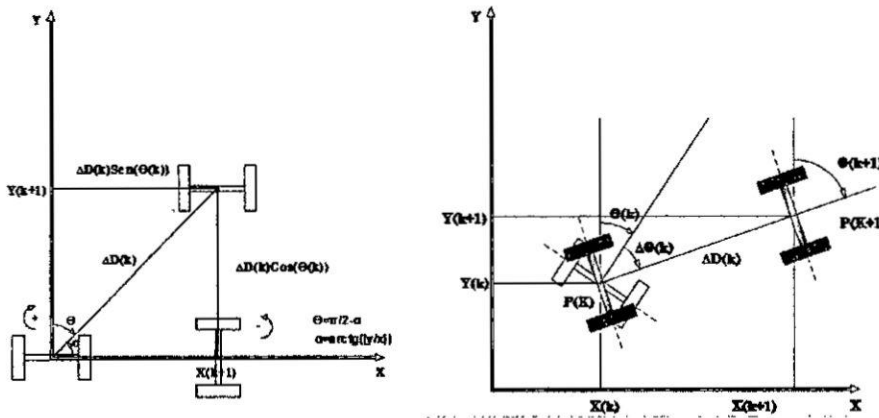
$$y > 0 \begin{cases} x > 0 \rightarrow \theta = +\left(\frac{\pi}{2} - \alpha\right) \\ x < 0 \rightarrow \theta = -\left(\frac{\pi}{2} - \alpha\right) \end{cases} \quad \text{Ec. N°3}$$

$$y < 0 \begin{cases} x > 0 \rightarrow \theta = +\left(\frac{\pi}{2} + \alpha\right) \\ x < 0 \rightarrow \theta = -\left(\frac{\pi}{2} + \alpha\right) \end{cases} \quad \text{Ec. N°4}$$

$$y = 0 \begin{cases} x > 0 \rightarrow \theta = +\frac{\pi}{2} \\ x < 0 \rightarrow \theta = -\frac{\pi}{2} \end{cases} \quad \text{Ec. N°5}$$

$$x = 0 \begin{cases} y \geq 0 \rightarrow \theta = 0 \\ y < 0 \rightarrow \theta = \pi \end{cases} \quad \text{Ec. N°6}$$

FIGURA N°46 Cinemática directa del robot diferencial

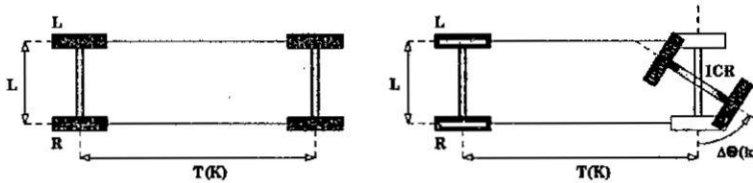


$$\begin{bmatrix} x(k+1) \\ y(k+1) \\ \theta(k+1) \end{bmatrix} = \begin{bmatrix} x(k) + \Delta D(k) \sin(\theta(k) + \Delta\theta(k)) \\ y(k) + \Delta D(k) \cos(\theta(k) + \Delta\theta(k)) \\ \theta(k) + \Delta\theta(k) \end{bmatrix} \quad \text{Ec. N°7}$$

En la ecuación N°7 muestra la ecuación de la cinemática directa de un robot móvil diferencial.

CINEMÁTICA INVERSA

Trayectoria de un movimiento de traslación seguido de una rotación sobre el eje de rotación.



Las distancias recorridas por cada rueda motora en dirección al eje de movimiento están determinada por:

$$\Delta D_L(k) = T(k) - \Delta\theta(k) \frac{L}{2}$$

$$\Delta D_R(k) = T(k) + \Delta\theta(k) \frac{L}{2}$$

De la ecuación anterior se obtiene la distancia recorrida y el ángulo de la orientación del robot en el desplazamiento $\Delta D(k)$ y $\Delta\theta(k)$.

$$\Delta D(k) = \frac{\Delta D_R(k) + \Delta D_L(k)}{2} = T(k)$$

$$\Delta \theta(k) = \frac{\Delta D_R(k) - \Delta D_L(k)}{L} = \Delta \theta(k)$$

Haremos un cambio de variable $\theta(k) + \Delta \theta(k) = \alpha$

Con el fin de no extender la operación

Reemplazando en la ecuación N°7

$$\begin{bmatrix} x(k+1) \\ y(k+1) \\ \theta(k+1) \end{bmatrix} = \begin{bmatrix} x(k) \\ y(k) \\ \theta(k) \end{bmatrix} + \begin{bmatrix} \frac{\Delta D_R(k) \sin(\alpha)}{2} + \frac{\Delta D_L(k) \sin(\alpha)}{2} \\ \frac{\Delta D_R(k) \cos(\alpha)}{2} + \frac{\Delta D_L(k) \cos(\alpha)}{2} \\ \frac{\Delta D_R(k) - \Delta D_L(k)}{L} \end{bmatrix}$$

$$\begin{bmatrix} x(k+1) - x(k) \\ y(k+1) - y(k) \\ \theta(k+1) - \theta(k) \end{bmatrix} = \begin{bmatrix} \frac{\sin(\alpha)}{2} & \frac{\sin(\alpha)}{2} \\ \frac{\cos(\alpha)}{2} & \frac{\cos(\alpha)}{2} \\ \frac{1}{L} & -\frac{1}{L} \end{bmatrix} * \begin{bmatrix} \Delta D_R(k) \\ \Delta D_L(k) \end{bmatrix}$$

Al realizar este paso se encuentra el primer inconveniente, puesto que es necesario determinar la inversa de una matriz no cuadrada. La solución a este problema es encontrar la pseudoinversa de dicha matriz [n4]. Esta matriz se define como:

$$A^+ = (A^T * A)^{-1} * A^T$$

Tenemos:

$$A = \begin{bmatrix} \frac{\sin(\alpha)}{2} & \frac{\sin(\alpha)}{2} \\ \frac{\cos(\alpha)}{2} & \frac{\cos(\alpha)}{2} \\ \frac{1}{L} & -\frac{1}{L} \end{bmatrix}$$

Nos quedaría:

$$\begin{bmatrix} \Delta x \\ \Delta y \\ \Delta \theta \end{bmatrix} = A * \begin{bmatrix} \Delta D_R(k) \\ \Delta D_L(k) \end{bmatrix} \rightarrow \begin{bmatrix} \Delta D_R(k) \\ \Delta D_L(k) \end{bmatrix} = A^+ * \begin{bmatrix} x \\ y \end{bmatrix}$$

$$A^T * A = \begin{bmatrix} \frac{\sin(\alpha)}{2} & \frac{\cos(\alpha)}{2} & \frac{1}{L} \\ \frac{\sin(\alpha)}{2} & \frac{\cos(\alpha)}{2} & -\frac{1}{L} \end{bmatrix} * \begin{bmatrix} \frac{\sin(\alpha)}{2} & \frac{\sin(\alpha)}{2} \\ \frac{\cos(\alpha)}{2} & \frac{\cos(\alpha)}{2} \\ \frac{1}{L} & -\frac{1}{L} \end{bmatrix}$$

$$A^T * A = \begin{bmatrix} \frac{\sin^2(\alpha)}{4} + \frac{\cos^2(\alpha)}{4} + \frac{1}{L^2} & \frac{\sin^2(\alpha)}{4} + \frac{\cos^2(\alpha)}{4} - \frac{1}{L^2} \\ \frac{\sin^2(\alpha)}{4} + \frac{\cos^2(\alpha)}{4} - \frac{1}{L^2} & \frac{\sin^2(\alpha)}{4} + \frac{\cos^2(\alpha)}{4} + \frac{1}{L^2} \end{bmatrix}$$

$$A^T * A = \begin{bmatrix} \frac{1}{4} + \frac{1}{L^2} & \frac{1}{4} - \frac{1}{L^2} \\ \frac{1}{4} - \frac{1}{L^2} & \frac{1}{4} + \frac{1}{L^2} \end{bmatrix} \rightarrow (A^T * A)^{-1}$$

$$= \frac{1}{\left(\frac{1}{4} + \frac{1}{L^2}\right)^2 - \left(\frac{1}{4} - \frac{1}{L^2}\right)^2} * \begin{bmatrix} \frac{1}{4} + \frac{1}{L^2} & \frac{1}{L^2} - \frac{1}{4} \\ \frac{1}{L^2} - \frac{1}{4} & \frac{1}{4} + \frac{1}{L^2} \end{bmatrix}$$

$$(A^T * A)^{-1} = \frac{1}{4 * \frac{1}{4} * \frac{1}{L^2}} * \begin{bmatrix} \frac{1}{4} + \frac{1}{L^2} & \frac{1}{L^2} - \frac{1}{4} \\ \frac{1}{L^2} - \frac{1}{4} & \frac{1}{4} + \frac{1}{L^2} \end{bmatrix} = L^2 * \begin{bmatrix} \frac{1}{4} + \frac{1}{L^2} & \frac{1}{L^2} - \frac{1}{4} \\ \frac{1}{L^2} - \frac{1}{4} & \frac{1}{4} + \frac{1}{L^2} \end{bmatrix}$$

$$(A^T * A)^{-1} * A^T = L^2 * \begin{bmatrix} \frac{1}{4} + \frac{1}{L^2} & \frac{1}{L^2} - \frac{1}{4} \\ \frac{1}{L^2} - \frac{1}{4} & \frac{1}{4} + \frac{1}{L^2} \end{bmatrix} * \begin{bmatrix} \frac{\sin(\alpha)}{2} & \frac{\cos(\alpha)}{2} & \frac{1}{L} \\ \frac{\sin(\alpha)}{2} & \frac{\cos(\alpha)}{2} & -\frac{1}{L} \end{bmatrix}$$

$$(A^T * A)^{-1} * A^T$$

$$= L^2 \begin{bmatrix} \frac{\sin(\alpha)}{8} + \frac{\sin(\alpha)}{2L^2} + \frac{\sin(\alpha)}{2L^2} - \frac{\sin(\alpha)}{8} & \frac{\cos(\alpha)}{8} + \frac{\cos(\alpha)}{2L^2} + \frac{\cos(\alpha)}{2L^2} - \frac{\cos(\alpha)}{8} & \frac{1}{4L} - \frac{1}{L^3} - \frac{1}{L^3} + \frac{1}{4L} \\ \frac{\sin(\alpha)}{2L^2} - \frac{\sin(\alpha)}{8} + \frac{\sin(\alpha)}{8} + \frac{\sin(\alpha)}{2L^2} & \frac{\cos(\alpha)}{2L^2} - \frac{\cos(\alpha)}{8} + \frac{\cos(\alpha)}{8} + \frac{\cos(\alpha)}{2L^2} & \frac{1}{L^3} - \frac{1}{4L} - \frac{1}{4L} - \frac{1}{L^3} \end{bmatrix}$$

$$(A^T * A)^{-1} * A^T = L^2 \begin{bmatrix} \frac{\sin(\alpha)}{L^2} & \frac{\cos(\alpha)}{L^2} & \frac{1}{2L} \\ \frac{\sin(\alpha)}{L^2} & \frac{\cos(\alpha)}{L^2} & -\frac{1}{2L} \end{bmatrix} = \begin{bmatrix} \sin(\alpha) & \cos(\alpha) & \frac{L}{2} \\ \sin(\alpha) & \cos(\alpha) & -\frac{L}{2} \end{bmatrix}$$

La matriz pseudoinversa esta defina por:

$$A^+ = \begin{bmatrix} \sin(\alpha) & \cos(\alpha) & \frac{L}{2} \\ \sin(\alpha) & \cos(\alpha) & -\frac{L}{2} \end{bmatrix}$$

Reemplazamos en la ecuación N°8:

$$\begin{bmatrix} \Delta D_R(k) \\ \Delta D_L(k) \end{bmatrix} = A^+ * \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta \theta \end{bmatrix}$$

$$\begin{bmatrix} \Delta D_R(k) \\ \Delta D_L(k) \end{bmatrix} = \begin{bmatrix} \sin(\alpha) & \cos(\alpha) & \frac{L}{2} \\ \sin(\alpha) & \cos(\alpha) & -\frac{L}{2} \end{bmatrix} * \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta \theta \end{bmatrix}$$

$$\begin{bmatrix} \Delta D_R(k) \\ \Delta D_L(k) \end{bmatrix} = \begin{bmatrix} \sin(\alpha) & \cos(\alpha) & \frac{L}{2} \\ \sin(\alpha) & \cos(\alpha) & -\frac{L}{2} \end{bmatrix} * \begin{bmatrix} x(k+1) - x(k) \\ y(k+1) - y(k) \\ \theta(k+1) - \theta(k) \end{bmatrix}$$

Donde alpha es:

$$\alpha = \theta(k) + \Delta \theta(k)$$

$$\begin{bmatrix} \Delta D_R(k) \\ \Delta D_L(k) \end{bmatrix} = \begin{bmatrix} \sin(\theta(k) + \Delta \theta(k)) & \cos(\theta(k) + \Delta \theta(k)) & \frac{L}{2} \\ \sin(\theta(k) + \Delta \theta(k)) & \cos(\theta(k) + \Delta \theta(k)) & -\frac{L}{2} \end{bmatrix} * \begin{bmatrix} x(k+1) - x(k) \\ y(k+1) - y(k) \\ \theta(k+1) - \theta(k) \end{bmatrix}$$

$$\begin{bmatrix} \Delta D_R(k) \\ \Delta D_L(k) \end{bmatrix} = \begin{bmatrix} \sin(\theta(k) + \Delta \theta(k)) & \cos(\theta(k) + \Delta \theta(k)) & \frac{L}{2} \\ \sin(\theta(k) + \Delta \theta(k)) & \cos(\theta(k) + \Delta \theta(k)) & -\frac{L}{2} \end{bmatrix} * \begin{bmatrix} \Delta x(k) \\ \Delta y(k) \\ \Delta \theta(k) \end{bmatrix}$$

$$\Delta D(k) = \frac{\Delta D_R(k) + \Delta D_L(k)}{2}$$

$$\Delta D(k) = \Delta x(k) \sin(\theta(k) + \Delta \theta(k)) + \Delta y(k) \cos(\theta(k) + \Delta \theta(k))$$

Para nuestro caso $\Delta \theta(k) = 0$

Tenemos:

$$\Delta D(k) = \Delta x(k) \sin(\theta(k)) + \Delta y(k) \cos(\theta(k))$$

Ec. N°9

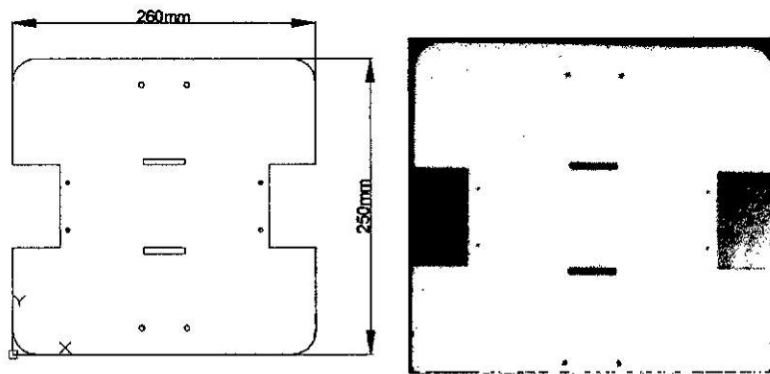
En la ecuación N°9 muestra la ecuación de la cinemática inversa de un robot móvil diferencial.

4.2.10. Diseño mecánico y elección de motores

DISEÑO DEL CHASIS

El material utilizado en la construcción del robot debe ser ligero, se optó por un chasis fabricado en Acrílico de 4 milímetros.

FIGURA N°47 Diseño del chasis en el Software Autocad

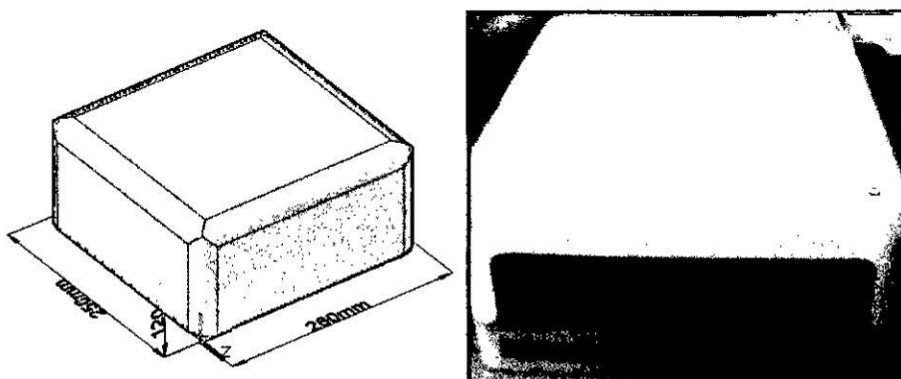


Dimensiones del chasis 250x260x4mm, cortada con una cortadora laser.

DISEÑO DE LA CARROCERÍA

El material utilizado en la construcción del robot debe ser ligero, se optó por una carrocería fabricada en Fibra de Vidrio.

FIGURA N°48 Diseño de la carrocería en el Software Autocad



Dimensiones de la carrocería 250x260x120mm.

TABLA N°1 Características del motor

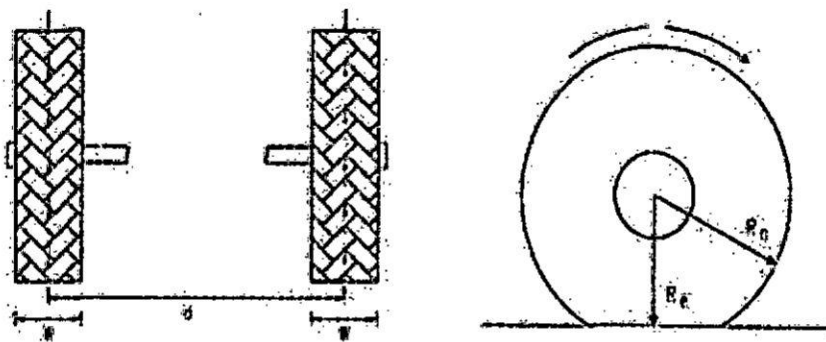
Gear ratio	43.8:1
No-load speed	251 + 10% RPM
No-load current	350 mA
Start Voltage	1.0 V
Stall Torque	18 Kg.com
StallCurrent	7A
Insulationresistance	20 M Ω
EncoderOperatingVoltage	5 V
Encoder type	Hall
Encoder Resolution	700CPR(gearboxshaft)
Weight	205g

ELECCIÓN DE LAS RUEDAS

Por ultimo se comprueba que los valores de velocidad y torque del datasheet estan por encima de los valores de diseño calculados.

Las ruedas deben ser de un material que genere la suficiente fricción para que no resbale en el suelo y que posea rigidez (ver FIGURA N°49) para evitar deformaciones; por lo que se eligieron rueda de silicona.

FIGURA N°49 Deformación de neumático



En la FIGURA N°49 se observa el cambio del diámetro de las ruedas, la cual distorsiona el control de dirección del vehículo.

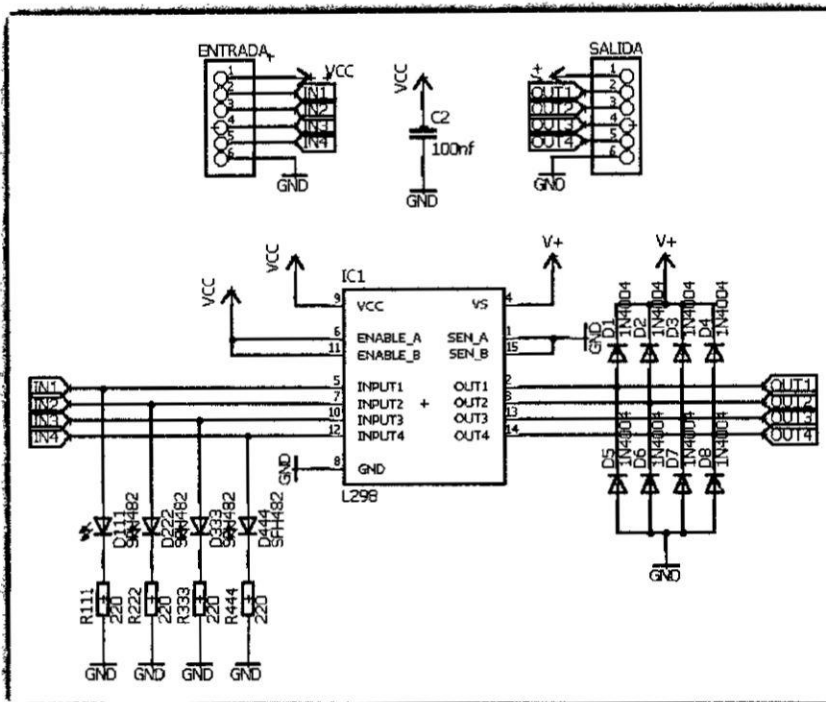
4.2.11. Diseño de las tarjetas electrónicas

DISEÑO DE LA TARJETA DEL PUENTE H

El circuito Puente H nos permite controlar hasta 2 potentes motores de corriente continua de forma muy sencilla y eficaz, también permite controlar el sentido de giro y velocidad mediante señales TTL.

Este esquema esta realizado con el programa Eagle, Eagle es un software editor de diseño de diagramas y pcb.

FIGURA N°50 Esquema del circuito Puente H



Bornes de conexión tipo Molex entre Puente H y Microcontrolador.

TABLA N°2 Bornes Puente H y Microcontrolador

PINES	NOMBRE	DESCRIPCIÓN
1	Vcc	5v DC
2	IN1	PWM 1
3	IN2	PWM 2
4	IN3	PWM 3
5	IN4	PWM 4
6	gnd	0v DC

Los bornes de entrada son de tipo Molex y se conectan al controlador.

IN1 e IN2 son entradas de control y sirve para controlar al Motor1, IN3 y IN4 sirve para controlar al Motor2.

Bornes de conexión tipo Molex entre Puente H y Motor.

TABLA N°3 Bornes Puente H y Motor

PINES	NOMBRE	DESCRIPCIÓN
1	V+	12v DC
2	OUT1	PWM 1 Motor1
3	OUT2	PWM 2 Motor1
4	OUT3	PWM 3 Motor2
5	OUT4	PWM 4 Motor2
6	gnd	0v DC

Los bornes de salida son de tipo Molex y se conectan a los Motores.

OUT1 y OUT2 son salidas que controlan al Motor1, OUT3 y OUT4 salidas que controlan al Motor2.

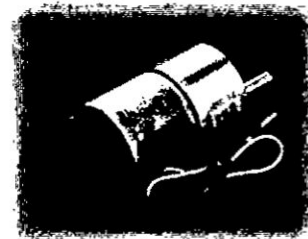
También incluye 4 leds para visualizar las señales de control.

Para el diseño del circuito es necesario analizar las características del motor a controlar.

Motor encoder GB37Y3530-12V-251R.

Especificación.

<i>Gear ratio</i>	43.8:1
<i>No-load speed</i>	251 + 10% RPM
<i>No-load current</i>	350 mA
<i>Start Voltage</i>	1.0 V
<i>Stall Torque</i>	18 Kg.com
<i>Stall Current</i>	7A
<i>Insulation resistance</i>	20 M Ω
<i>Encoder Operating Voltage</i>	5 V
<i>Encoder type</i>	Hall
<i>Encoder Resolution</i>	700CPR(gearboxshaft)
<i>Weight</i>	205g



De sus características obtenemos los parámetros requeridos para el diseño.

- Corriente: 0.4Amperios
- Voltaje: 12voltios
- Potencia: 4.8w

Con los datos de corriente y voltaje podemos elegir el chip para el puente H.

El L298 es un circuito integrado de 15 pines que contiene internamente el 2 circuitos puentes H contruidos a base de transistores y cmos. Las señales de control son de tipo TTL por ende se podría trabajar con el Arduino sin problemas.



Especificaciones

TABLA N°4 Especificaciones chip L298

Symbol	Parameter	Value	Unit
V_s	Power Supply	50	V
V_{SS}	Logic Supply Voltage	7	V
V_i, V_{en}	Input and Enable Voltage	-0.3 to 7	V
I_o	Peak Output Current (each Channel)		
	- Non Repetitive ($t = 100\mu s$)	3	A
	- Repetitive (80% on -20% off; $t_{on} = 10ms$)	2.5	A
	-DC Operation	2	A
V_{sens}	Sensing Voltage	-1 to 2.3	V
P_{tot}	Total Power Dissipation ($T_{case} = 75^\circ C$)	25	W
T_{op}	Junction Operating Temperature	-25 to 130	$^\circ C$
T_{stg}, T_j	Storage and Junction Temperature	-40 to 150	$^\circ C$

Haciendo una comparación entre las especificaciones del chip y el Motor.

TABLA N°5 Comparación L298 y Motor

magnitudes	L298	Motor
Voltaje	50v	12v
Corriente	2A	0.4A
Potencia	25w	4.8w

Notamos que el Chip L298 supera los parámetros de los valores máximos del Motor DC para lo cual garantizamos su buen funcionamiento para la aplicación.

Nótese que se observan dos anchos de las pistas del circuito puente H, esto es debido a que existen 2 etapas la de control y la de potencia.

La etapa de control maneja voltaje TTL de 5 v y corrientes en mA (mili amperios).

La etapa de potencia maneja voltaje de 12v y corriente que es la del motor 0.4A.

Para calcular el ancho de pista de la etapa de potencia tenemos la siguiente página de una empresa (microensamble) colombiana fabricante de circuitos impresos.

Link:http://www.microensamble.com/site/index.php?option=com_content&view=article&id=105:calculadora-de-ancho-de-pista&catid=62&Itemid=197

FIGURA N°52 Página calculadora de ancho de pista

Calculadoras para PCBs: Este utilidad calcula el ancho de una pista en el diseño de circuitos impresos de acuerdo a la curva de la norma IPC 2221. Normalmente Sumaluz (IPC-2221) convierte como base el valor de la corriente en mA a los (mA/1000) así como el espesor del cobre utilizado.

Datos necesarios:

Corriente (0 - 25 A): A (Amperios)

Espesor del cobre (0.5 - 3 Oz): Oz

Datos opcionales:

Aumento de temperatura (10-160°C): °C

Temperatura ambiente: °C

Longitud de pista: mm

Circuito Impreso

Pista

Más sobre pistas

El ancho de pista depende de la corriente que fluye por ella, de la temperatura ambiente, de la longitud de la pista y del espesor del cobre utilizado.

El ancho de pista depende de la corriente que fluye por ella, de la temperatura ambiente, de la longitud de la pista y del espesor del cobre utilizado.

Con esta página es posible calcular el ancho de pista, solo debemos introducir el parámetro de corriente y automáticamente nos calculara el ancho de pista.

Para nuestro caso la corriente que introduciremos debe ser mayor a la corriente nominal del motor

$$I_{pcb} > I_{motor}$$

Donde:

I_{pcb} : corriente del pcb.

I_{motor} : corriente del motor.

Por lo tanto si le damos $I_{pcb} = 1.4$ amperios entonces.

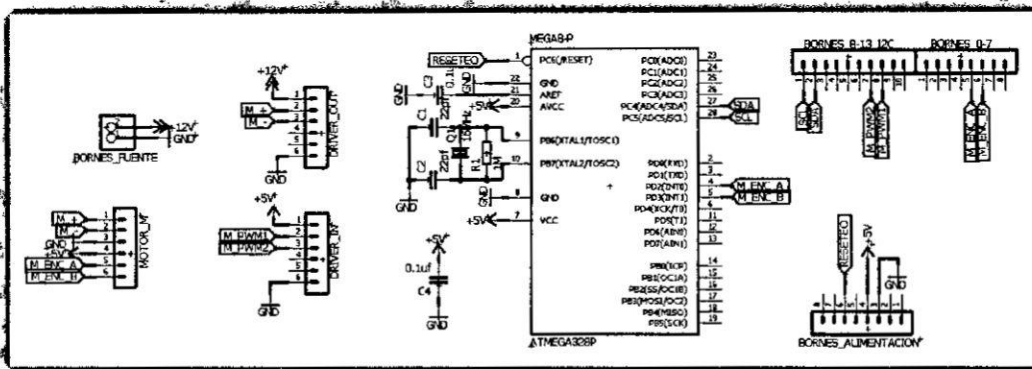
$$1.4A > 0.4A$$

Cumple la condición con dicho valor la calculadora nos arroja un valor de ancho de pista igual. Ancho de PCB = 1.243mm, con este valor ya podemos realizar el pcb sin problemas, garantizando su buen funcionamiento para nuestra aplicación.

DISEÑO DE LA TARJETA DE ADQUISICIÓN DE DATOS

Esta tarjeta se encarga de convertir los pulsos que entrega el Encoder del motor para convertirlo en un valor numérico digital, básicamente la tarjeta de adquisición es la construcción de un **tacómetro digital**.

FIGURA N°53 Esquema del circuito tacómetro digital



Bornes de conexión tipo shield entre tacómetro - Arduino.

TABLA N°6 Bornes tacómetro - Arduino

	PINES	NOMBRE	DESCRIPCIÓN
BORNES DE ALIMENTACIÓN	3	GND	0 voltios
	4	+5V	5 voltios
	6	Reseteo	Reset del Atmega 328
BORNES	5	M_ENC_A	Encoder canal A

0-7	6	M_ENC_B	Encoder canal B
BORNES	8	M_PWM1	Control PWM 1
8-13	7	M_PWM2	Control PWM 2
I2C	1	SCL	I2C Serial Clock
	2	SDA	I2C Serial Data

Bornes de conexión tipo Molex tacómetro – Puente H

TABLA N°7 Bornes tacómetro - Puente H

	PINES	NOMBRE	DESCRIPCIÓN
Drive_IN	1	+5V	5 voltios DC
	2	M_PWM1	Control PWM 1
	3	M_PWM2	Control PWM 2
	6	GND	0 voltios DC
Drive_OUT	1	+12V	12 voltios DC
	2	M_+	PWM 1
	3	M_-	PWM 2
	6	GND	0 voltios DC

Bornes de conexión tipo Molex tacómetro – Baterías.

TABLA N°8 Bornes tacómetro – Baterías

	PINES	NOMBRE	DESCRIPCIÓN
MOTOR_M	1	M_+	PWM 1
	2	M_-	PWM 2
	3	GND	0 voltios DC
	4	+5V	5 voltios DC
	5	M_ENC_A	Encoder canal A
	6	M_ENC_B	Encoder canal B

Bornes de conexión tipo Molex tacómetro – Motor.

TABLA N°9 Bornes tacómetro – Motor

	PINES	NOMBRE	DESCRIPCIÓN
MOTOR_M	1	M_+	PWM 1
	2	M_-	PWM 2
	3	GND	0 voltios DC
	4	+5V	5 voltios DC
	5	M_ENC_A	Encoder canal A
	6	M_ENC_B	Encoder canal B

DISEÑO DEL TACÓMETRO DIGITAL

Para el diseño del tacómetro se analiza la salida del encoder del motor, es decir se examinara la salida del encoder frente a los diversos voltajes de entrada.

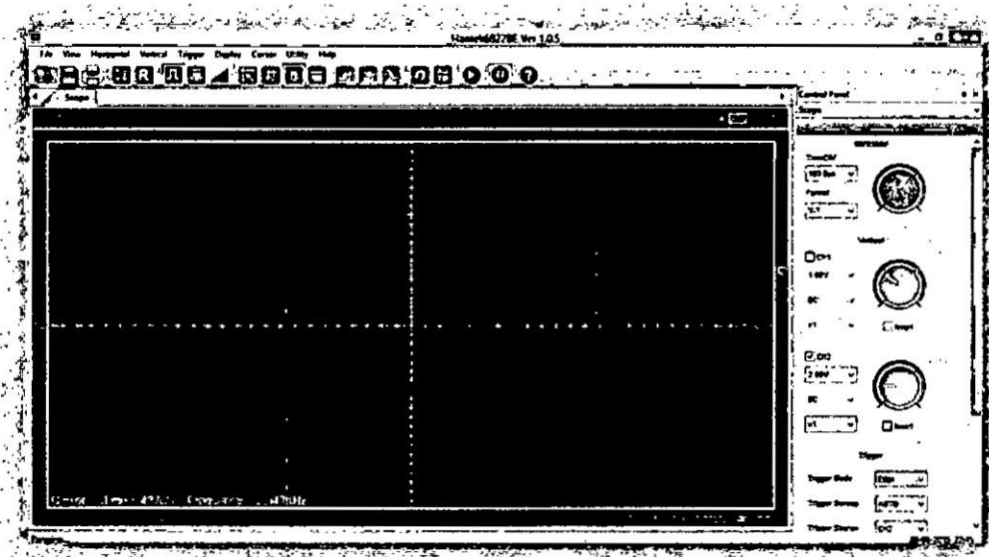
Para ello elaboramos una tabla de voltaje vs frecuencia.

TABLA N°10 Comparación voltaje, frecuencia y Periodo

Voltaje	Frecuencia	Periodo
25	62.18Hz	16.1mS
50	439.5Hz	2.28mS
100	998.6Hz	1.1mS
150	1355Hz	738uS
200	1817Hz	550uS
255	2343HZ	427us

Esta TABLA N°10 se realizó con la ayuda del osciloscopio Hantek.

FIGURA N°54 Señal de salida del encoder



Como se observa en la *FIGURA N°54* la frecuencia máxima es de 2343HZ cuyo periodo es 427us.

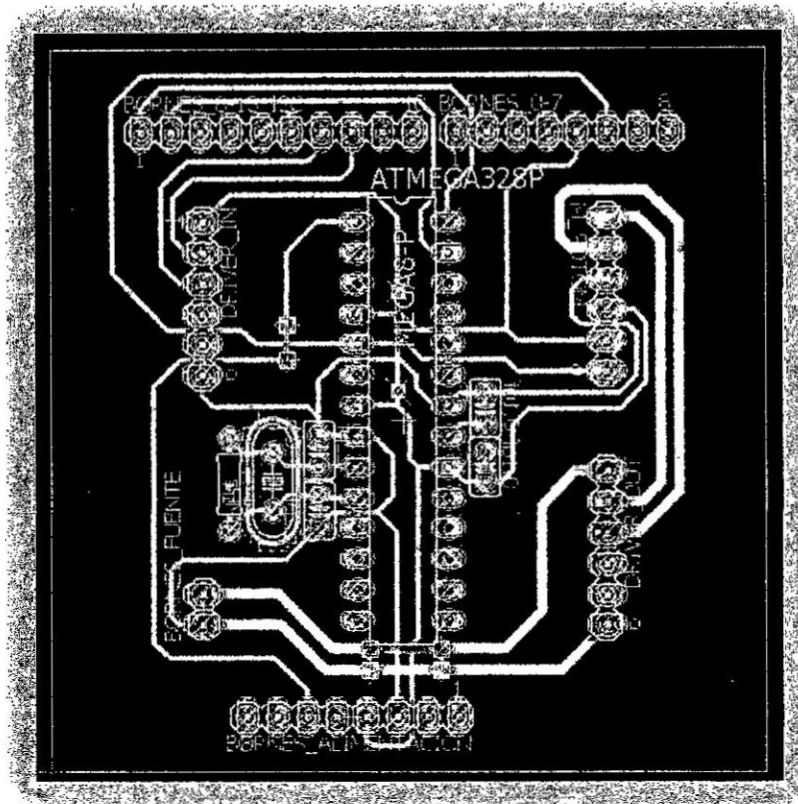
Esta información es importante para la elección del microcontrolador que se usara en el tacómetro. Ya que se tendrá que elegir un microcontrolador que sea capaz de superar este periodo.

El **Atmega328** es capaz de cubrir con esta necesidad, ya que la velocidad de cada instrucción en el lenguaje assembler es de 65nSegundos y en el lenguaje Arduino es de 4uSegundo, es suficientemente veloz para adquirir y mostrar los datos en tiempo real del encoder.

El protocolo a usar para la transmisión de datos del tacómetro es el I2C ya que es un protocolo veloz (100Kbit/s) periodo por byte de 80uS suficiente para la adquisición en tiempo real.

Habiendo terminado con el diseño se procede hacer el diseño layout del PCB.

FIGURA N°55 Layout del circuito Tacómetro

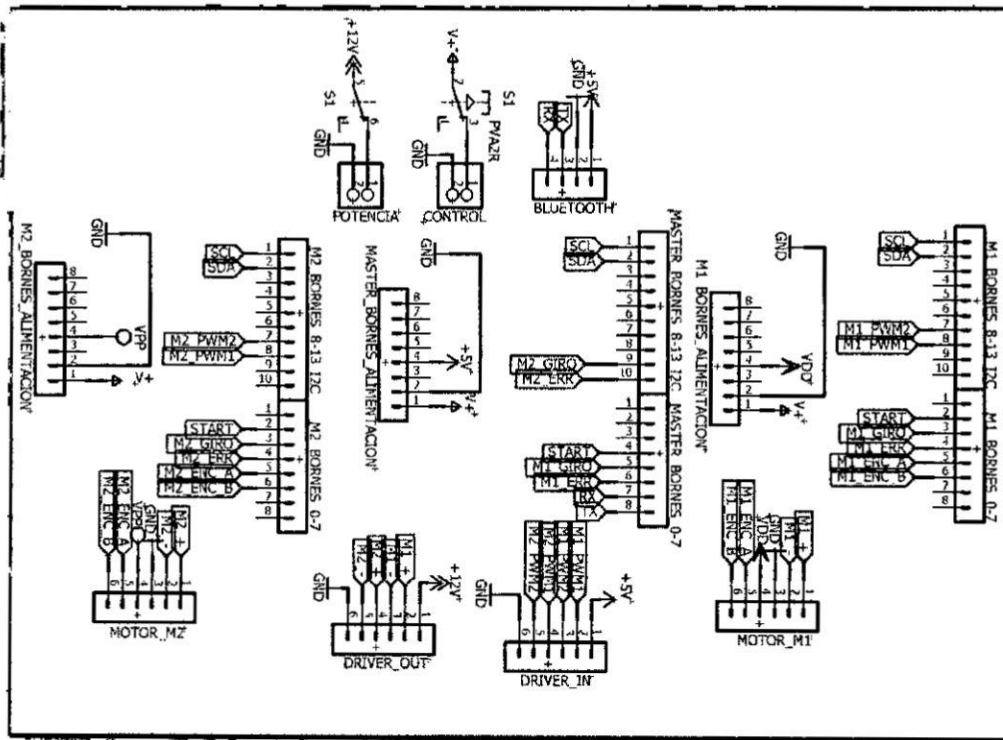


se observa en la figura realizamos la tarjeta del tacómetro en forma de de tal forma que se pueda acoplar el tacómetro con el arduino de manera segura, también tenemos Molex para la conexión del tacómetro con los demás circuitos.

DISEÑO DE LA TARJETA DE CONEXIÓN DE PERIFÉRICOS

Para una conexión segura y práctica hemos construido el siguiente esquema electrónico nos permitirá conectar todos los periféricos del Robot de forma sencilla y ordenada, el esquema ha sido desarrollado y diseñado en Eagle.

FIGURA N°56 Tarjeta de Conexión de Periféricos



A continuación se mostraran las tablas de los bornes de conexión.

Bornes de conexión tipo Molex Fuente de control – Batería 12v

TABLA N°11 Bornes Fuente de control – Batería 12v

Fuente de Control	PINES	NOMBRE	DESCRIPCIÓN
	1	+12V	12 voltios DC
	2	GND	0 voltios DC

Bornes de conexión tipo Molex Fuente de Potencia – Batería 12v

TABLA N°12 Bornes Fuente de Potencia – Batería 12v

Fuente de Potencia	PINES	NOMBRE	DESCRIPCIÓN
	1	+12V	12 voltios DC
	2	GND	0 voltios DC

Bornes de conexión tipo shield entre Tarjeta de conexión – Arduino Master.

TABLA N°13 Bornes Tarjeta de conexión – Arduino

	PINES	NOMBRE	DESCRIPCIÓN
MASTER BORNES DE ALIMENTACIÓN	1	V+	12 voltios
	2	GND	0 voltios
	4	+5V	5 voltios
MASTER BORNES 0-7	4	START	Bit de Inicio
	5	M1_GIRO	Bit de Giro
	6	M1_ERR	Bit de Error
	7	Rx	Transmisor
	8	Tx	Receptor
MASTER BORNES 8-13 I2C	1	SCL	I2C Serial Clock
	2	SDA	I2C Serial Data
	7	M2_GIRO	Bit de Giro
	8	M2_ERR	Bit de Error

Bornes de conexión tipo Molex entre Tarjeta de conexión – Arduino Motor 1.

TABLA N°14 Bornes Tarjeta de conexión – Arduino Motor 1

	PINES	NOMBRE	DESCRIPCIÓN
M1 BORNES DE ALIMENTACIÓN	1	V+	12 voltios
	2	GND	0 voltios
	4	VDD	5 voltios
M1 BORNES 0-7	2	START	Bit de Inicio
	3	M1_GIRO	Bit de Giro
	4	M1_ERR	Bit de Error
	5	M1_ENC_A	Encoder canal A
	6	M1_ENC_B	Encoder canal B
M1 BORNES 8-13 I2C	1	SCL	I2C Serial Clock
	2	SDA	I2C Serial Data
	7	M1_PWM1	Control PWM 1
	8	M1_PWM2	Control PWM 2

Bornes de conexión tipo Molex entre Tarjeta de conexión – Arduino Motor 2.

TABLA N°15 Bornes Tarjeta de conexión – Arduino Motor 2

	PINES	NOMBRE	DESCRIPCIÓN
M2 BORNES DE ALIMENTACIÓN	1	V+	12 voltios
	2	GND	0 voltios
	4	VPP	5 voltios
M2 BORNES 0-7	2	START	Bit de Inicio
	3	M2_GIRO	Bit de Giro
	4	M2_ERR	Bit de Error
	5	M2_ENC_A	Encoder canal A
	6	M2_ENC_B	Encoder canal B
M2 BORNES 8-13 I2C	1	SCL	I2C Serial Clock
	2	SDA	I2C Serial Data
	7	M2_PWM1	Control PWM 3
	8	M2_PWM2	Control PWM 4

Bornes de conexión tipo Molex entre Tarjeta de conexión – Motor 1.

TABLA N°16 Bornes Tarjeta de conexión – Motor 1

	PINES	NOMBRE	DESCRIPCIÓN
MOTOR_M1	1	M1_+	PWM 1
	2	M1_-	PWM 2
	3	GND	0 voltios DC
	4	+5V	5 voltios DC
	5	M1_ENC_A	Encoder canal A
	6	M1_ENC_B	Encoder canal B

Bornes de conexión tipo Molex entre Tarjeta de conexión – Motor 2.

TABLA N°17 Bornes Tarjeta de conexión – Motor 2

	PINES	NOMBRE	DESCRIPCIÓN
MOTOR_M2	1	M2_+	PWM 3
	2	M2_-	PWM 4
	3	GND	0 voltios DC
	4	+5V	5 voltios DC
	5	M2_ENC_A	Encoder canal A
	6	M2_ENC_B	Encoder canal B

Bornes de conexión tipo Molex entre Tarjeta de conexión – Bluetooth.

TABLA N°18 Bornes Tarjeta de conexión – Bluetooth

	PINES	NOMBRE	DESCRIPCIÓN
BLUETOOTH	1	+5V	5 voltios DC
	2	GND	0 voltios DC
	3	TX	Transmisor
	4	RX	Receptor

Bornes de conexión tipo Molex entre Tarjeta de conexión – Puente H.

TABLA N°19 Bornes Tarjeta de conexión – Entrada Puente H

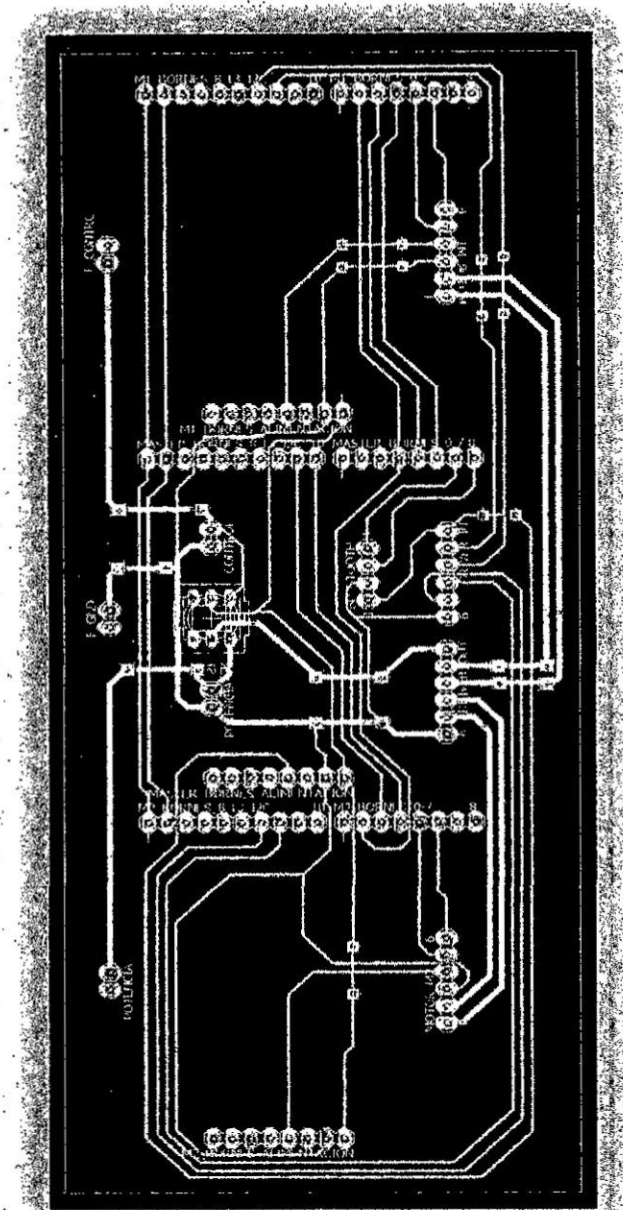
	PINES	NOMBRE	DESCRIPCIÓN
Drive_IN	1	+5V	5 voltios DC
	2	M_PWM1	Control PWM 1
	3	M_PWM2	Control PWM 2
	4	M_PWM3	Control PWM 3
	5	M_PWM4	Control PWM 4
	6	GND	0 voltios DC

TABLA N°20 Bornes Tarjeta de conexión – Salida Puente H

	PINES	NOMBRE	DESCRIPCIÓN
Drive_OUT	1	+12V	12 voltios DC
	2	M1_+	PWM 1
	3	M1_-	PWM 2
	4	M2_+	PWM 3
	5	M2_-	PWM 4
	6	GND	0 voltios DC

Diseño del PCB

FIGURA N°57 Figura Layout de tarjeta de Conexión de Periféricos



Este PCB tiene la siguiente medida 70x170 mm este PCB contendrá los siguientes módulos:

- 3 Arduino.
- 1 Bluetooth.
- 2 Baterías de 12V.
- 2 Motores DC.
- 1 Tarjeta Puente H

Todos los módulos se conectan por medio de la tarjeta de conexión, de esta forma el montaje y desmontaje del equipo es más sencillo y ordenado ya que cada módulo como conectores molex que les permiten tener un solo sentido de conexión.

4.2.12. Configuración y sincronización del módulo Bluetooth

CONFIGURACIÓN DE LA VELOCIDAD DEL MODULO BLUETOOTH

El módulo Bluetooth HC-05 viene configurado de fábrica con los siguientes valores predeterminados.

TABLA N°21 Bluetooth valores de fábrica

Nombre del módulo	HC-05
Pin o contraseña	1234
Modo de trabajo	Modo Esclavo (+ROLE: 0).
Modo de conexión	Se conecta al módulo Bluetooth especificado (+CMOD: 0).
Velocidad de transmisión	38400 bps; Bit de parada: 1 bit; Bit de paridad: Sin paridad

Los valores de fábrica los podemos cambiar en cualquier momento mediante comandos AT. El módulo HC-05 requiere de un proceso específico para entrar en el modo de comandos AT, para que el HC-05 entre en modo comandos AT, necesita que cuando se enciende el modulo, el pin KEY este HIGH.

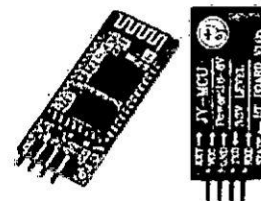
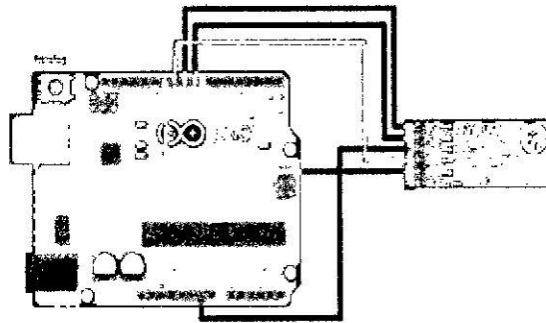


FIGURA N°58 Esquema de conexión entre Arduino y Bluetooth



El consumo del módulo es mínimo y nuestro Arduino es capaz de alimentarlo sin problemas, por eso el modulo bluetooth se alimentara cuando pongamos HIGH en el pin 9.

Al poner en HIGH el pin digital 8 también pondremos en high el pin key del bluetooth, de este modo cuando arranque entrara sin más en el modo de comandos AT.

El resto de los pines Txd y Rxd del bluetooth se deben conectar de forma cruzadas con los pines de comunicación de Arduino, debido a que usaremos la librería software Serial, para añadir otro puerto serial.

Lo siguiente es programar al arduino para que por medio de el ingresemos los comandos AT.

FIGURA N°59 Código para el ingreso de comando AT

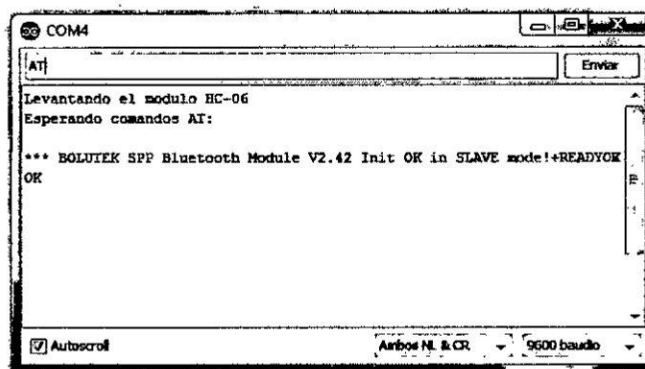
```
1 #include <SoftwareSerial.h>
2 SoftwareSerial BT1(10, 11); // RX | TX
3
4 void setup()
5 {
6   pinMode(8, OUTPUT); // Pin para alimentar el Bluetooth
7   pinMode(9, OUTPUT); // Pin para activar el key
8   digitalWrite(9, HIGH); // Alimentamos Bluetooth
9   delay(500); // Espera antes de encender el modulo
10  Serial.begin(9600);
11  Serial.println("Levantando el modulo HC-05");
12  digitalWrite(8, HIGH); //Al poner en HIGH forzaremos el modo AT
13  Serial.println("Esperando comandos AT.");
14  BT1.begin(38400);
15 }
16
17 void loop()
18 {
19   if (BT1.available())
20     Serial.write(BT1.read());
21   if (Serial.available())
22     BT1.write(Serial.read());
23 }
```

Con este codigo garantizamos que el modulo HC-05 entra solo en modo AT comandos, y que nuestra consola nos va a servir para programarlo.

El HC-05 envía a la consola un mensaje de Status en cuanto arranca en el modo AT, si es nuestro caso deberías ver un mensaje en la consola. Si no es así, o se ve signos raros, suele ser indicador de que la velocidad de comunicación entre Arduino y el modulo es incorrecta.

En el prom de la consola se puede ir probando velocidades hasta que veás algún mensaje se ve correcto. Enviar un AT siempre debería recibir una respuesta de OK.

FIGURA N°60 Ventana para el ingreso y salida de datos Serial



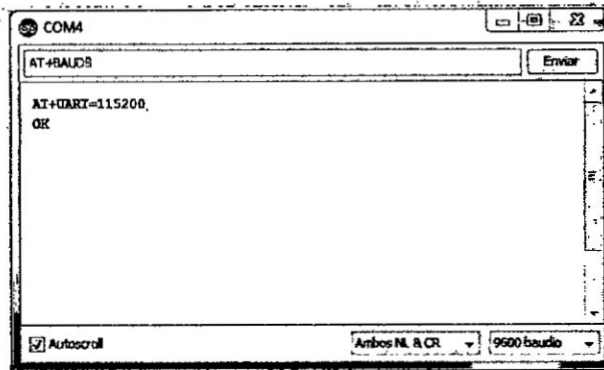
Existe un lista de comando para configurar el modulo bluetooth.

FIGURA N°61 Bluetooth tabla de comandos

Comando AT	Descripción	Respuesta
AT	Test de comunicación.	Responde con un OK.
AT+VERSION	Retorna la versión del módulo	Responde con la versión [OKInvarV1.8].
AT+BAUDx	Configura la velocidad de transmisión del módulo según el valor de "x": 1 = 1200 bps 2 = 2400 bps 3 = 4800 bps 4 = 9600 bps (por defecto) 5 = 19200 bps 6 = 38400 bps 7 = 57600 bps 8 = 115200 bps 9 = 230400 bps A = 460800 bps B = 921600 bps C = 1382400 bps	AT+BAUDA Configura la velocidad a 9600 baud rate. Responde con OK9600.
AT+NAMEx	Configura el nombre con el que se visualizara el módulo. Soporta hasta 20 caracteres	AT+NAMEArduino_HC-06 Configura el nombre del módulo a Arduino_HC-06. Responde con OKsetname.
AT+PINxxx	Configura el Pin de acceso al módulo (password). 1234 por defecto.	AT+PIN1234 Configura el pin a 1234. Responde con OKsetPIN.
Paridad AT+PN, PO, PE	Se configura la paridad de la comunicación en versiones V1.5 o superiores.	AT+PN: Sin paridad. Responde con OK None. AT+PO: Paridad Impar. Responde con OK Odd. AT+PE: Paridad par. Responde con OK Even.

El ultimo paso seria ingresar por consola el comando de velocidad deseada, en nuestro caso sera AT+BAUD8 (velocidad de 115200 bit/seg) que es la velocidad maxima que puede manejar el arduino .

FIGURA N°62 Confirmación de comando de velocidad



SINCRONIZACIÓN BLUETOOTH Y COMPUTADORA

Para lograr conectarse desde la computadora al módulo bluetooth es necesario seguir los siguientes pasos en Windows 7, (pero el procedimiento es similar para W8 y XP)

Si es la primera vez que el modulo bluetooth se va conectar a la PC es necesario primero vincularlo, para esto en la PC hacemos una búsqueda de los dispositivos Bluetooth.

FIGURA N°63 Icono del Dispositivo Bluetooth

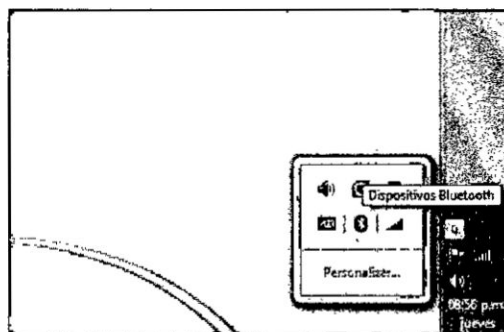


FIGURA N°64 Agregar un nuevo dispositivo Bluetooth

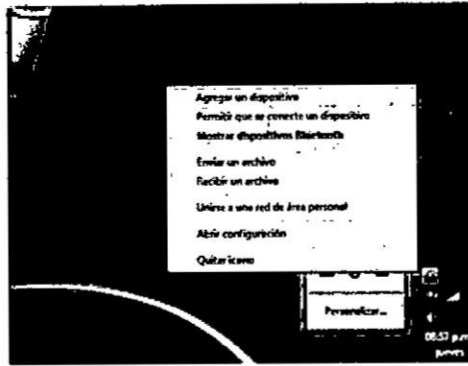
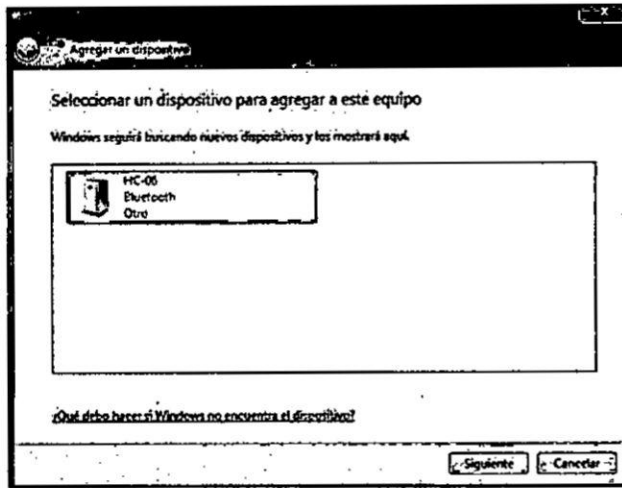
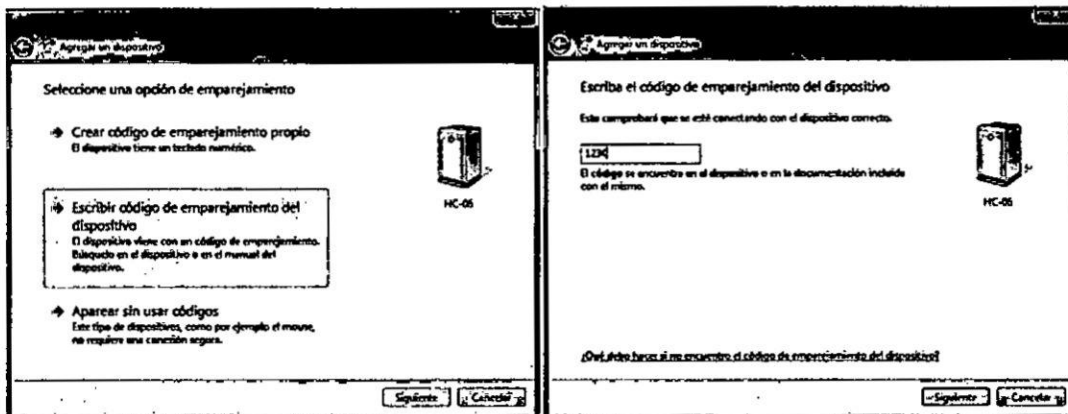


FIGURA N°65 Dispositivo Bluetooth Slave HC-06



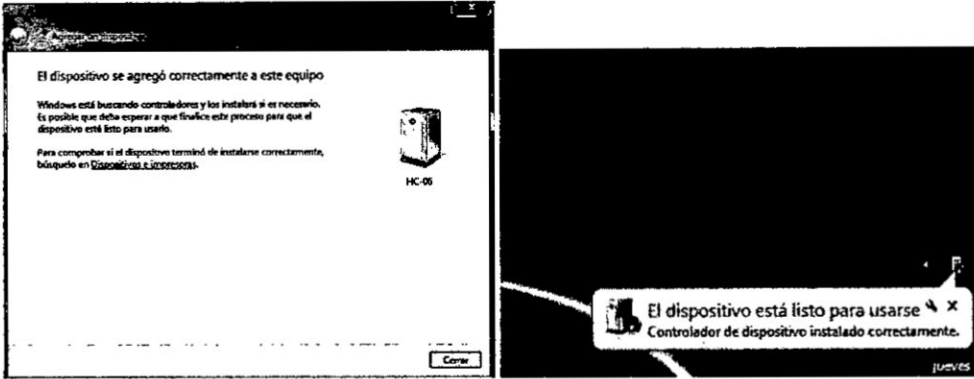
Damos siguiente

FIGURA N°66 Código de emparejamiento



En la búsqueda debería de Aparecer nuestro módulo Bluetooth con el nombre HC-06 o Linvor, Para vincularlo nos pedirá una clave o código de emparejamiento, generalmente es 1234.

FIGURA N°67 Dispositivo Bluetooth se agregó correctamente



El nombre y código de vinculación que mencionamos son los que vienen por defecto al igual que la velocidad de transmisión que viene por defecto de 9600.

4.2.13. Desarrollo de la interfaz gráfica en Labview

DESARROLLO DEL PANEL FRONTAL PARA EL INGRESO DE DATOS

FIGURA N°68 Vista Isométrica de un almacén

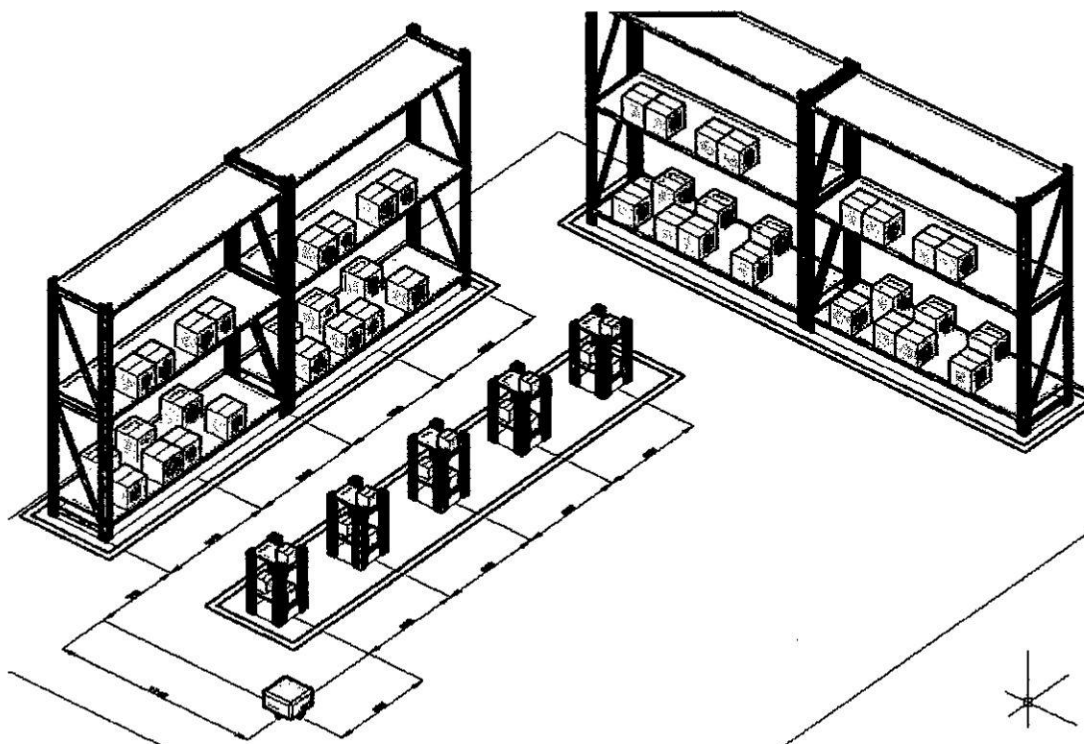
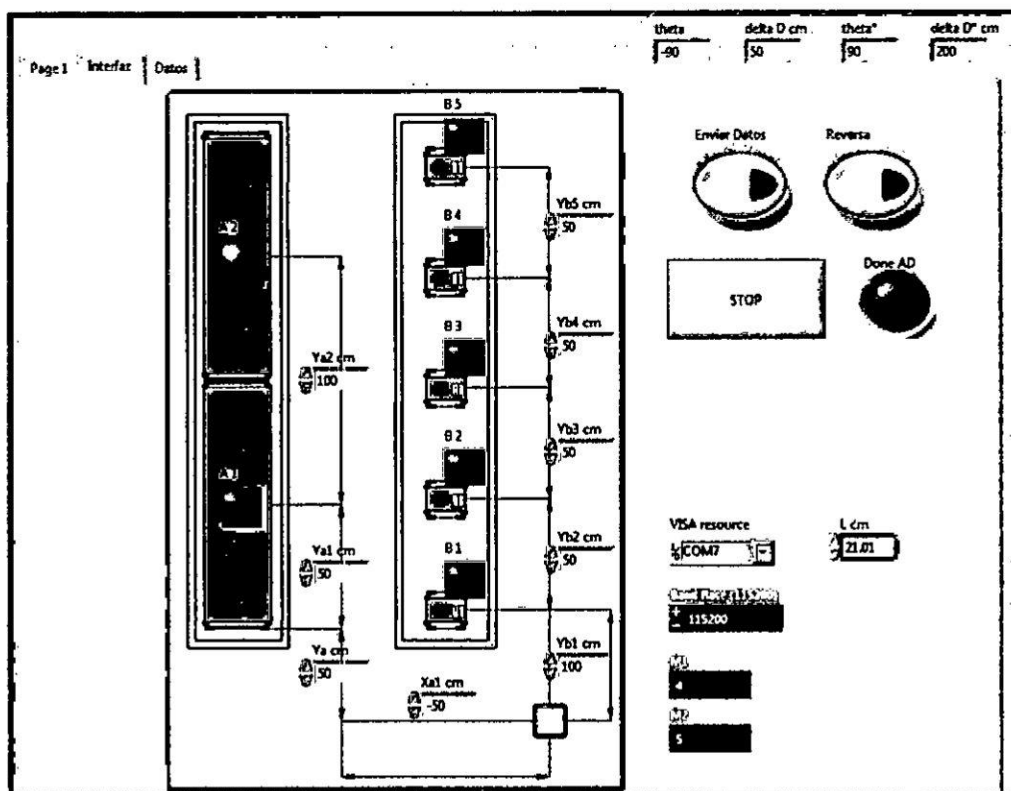


FIGURA N°69 Vista de planta de un almacén y Panel Frontal en Labview



El ingreso de las distancias en centímetros que el robot debe recorrer se ingresara en las variables Xa1, Ya, Ya1, Ya2, Yb1, Yb2, Yb3, Yb4, Yb5.

Visa Resource: Se seleccionara el puerto de la comunicación serial asignado automáticamente por la computadora al momento de la instalación.

L cm: Ingresa la distancia entre las ruedas, se usa esta variable para el control de giro del robot.

El ingreso de las distancias en centímetros que el robot debe recorrer se ingresara en las variables Xa1, Ya, Ya1, Ya2, Yb1, Yb2, Yb3, Yb4, Yb5.

Visa Resource: Se seleccionara el puerto de la comunicación serial asignado automáticamente por la computadora al momento de la instalación.

L cm: Ingresa la distancia entre las ruedas, se usa esta variable para el control de giro del robot.

BaudRate: Se configuro el modulo Bluetooth a la velocidad de 115200 baudios.

M1 y M2: Dirección del Arduino slave por I2C.

A1, A2, B1, B2, B3, B4, B5: Se selecciona la ubicación final donde el robot se dirige; solo se puede elegir uno de ellos.

Enviar Datos: Envía los datos de acuerdo al lugar seleccionado y las distancias ingresadas.

Reversa: El robot regresa a su punto de origen, manteniendo su ubicación inicial.

Stop: Detiene el programa y desconecta la comunicación Serial.

DoneAD: Se enciende completamente cuando termina el recorrido.

FIGURA N°70 Variables de Entrada y selección de la coordenada final

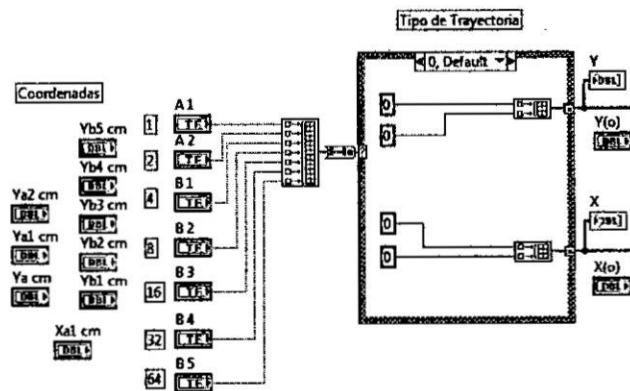
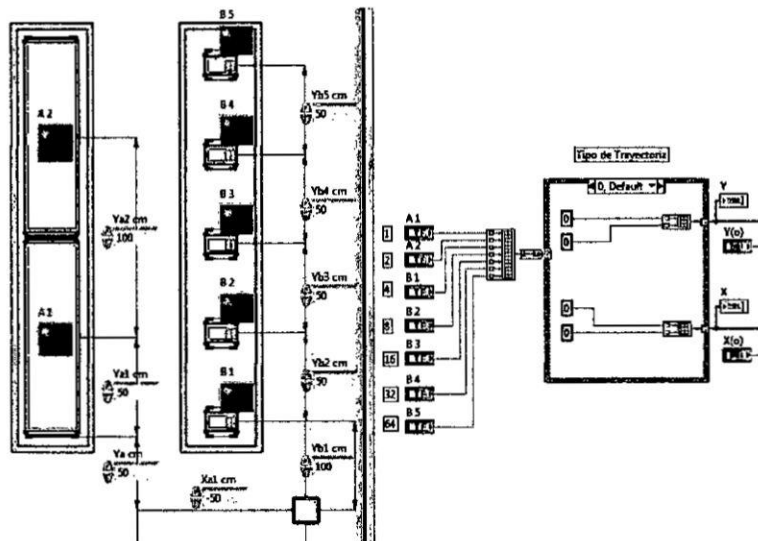
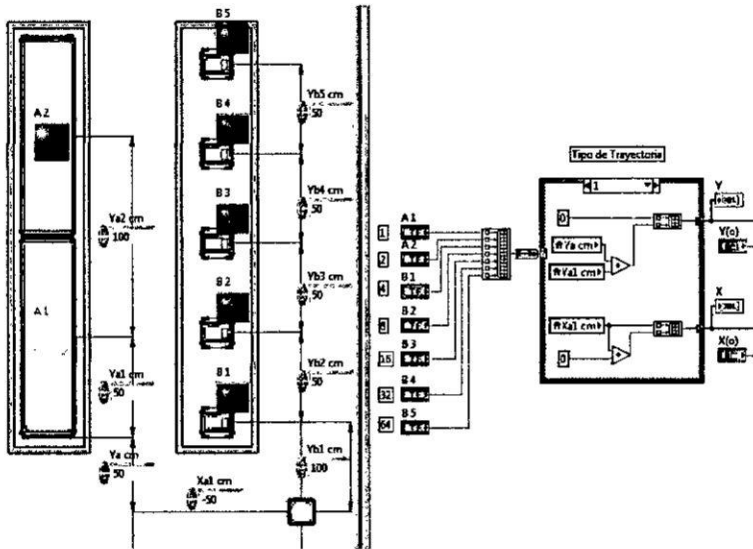


FIGURA N°71 Coordenada Inicial en el Origen



Cuando no se presiona ninguna coordenada final el robot se queda en el origen de coordenadas en el punto (0;0), esperando alguna orden.

FIGURA N°72 Coordenada Final en el Punto A1



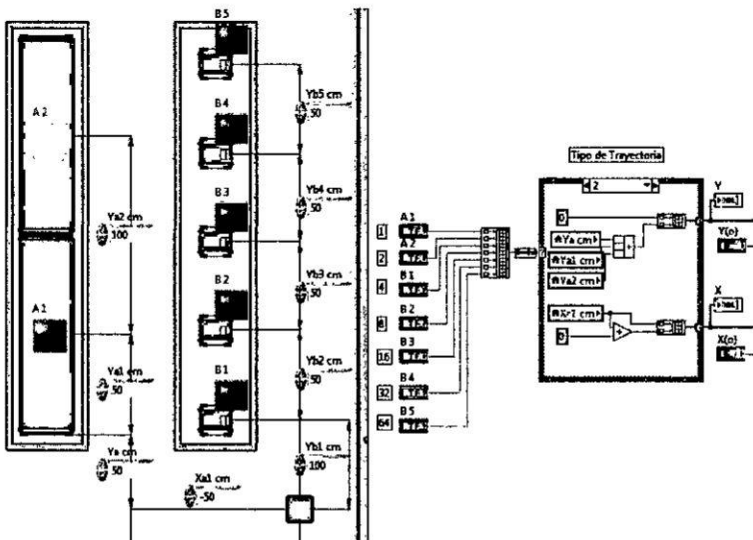
Cuando se presiona A1 el robot se tendrá que desplazar del punto $(0; 0)$ al punto $(X_{a1}; Y_{a1})$.

Teniendo la coordenada inicial y final se genera una trayectoria definida:

Primera trayectoria del punto $(0; 0)$ al punto $(X_{a1}; 0)$.

Segunda trayectoria del punto $(X_{a1}; 0)$ al punto $(X_{a1}; Y_{a1})$.

FIGURA N°73 Coordenada Final en el Punto A2



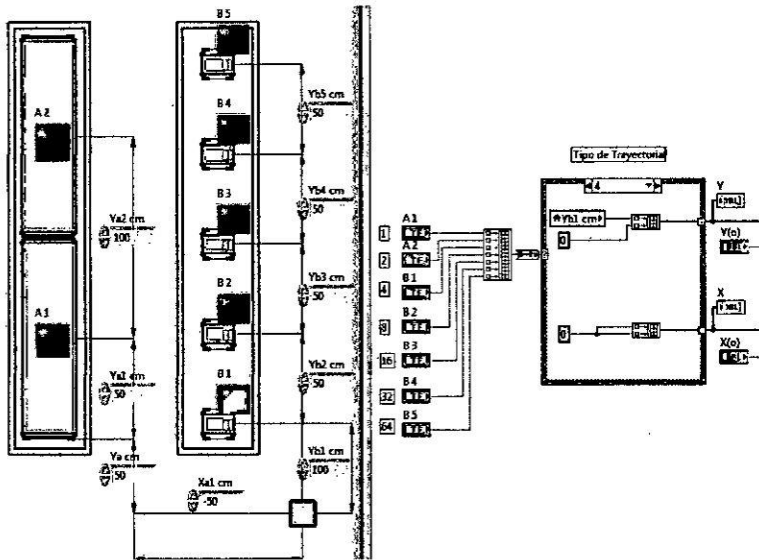
Cuando se presiona A2 el robot se tendrá que desplazar del punto $(0; 0)$ al punto $(X_{a1}; Y_{a1} + Y_{a2})$.

Teniendo la coordenada inicial y final se genera una trayectoria definida:

Primera trayectoria del punto $(0; 0)$ al punto $(X_{a1}; 0)$.

Segunda trayectoria del punto $(X_{a1}; 0)$ al punto $(X_{a1}; Y_{a1} + Y_{a2})$.

FIGURA N°74 Coordinada Final en el Punto B1

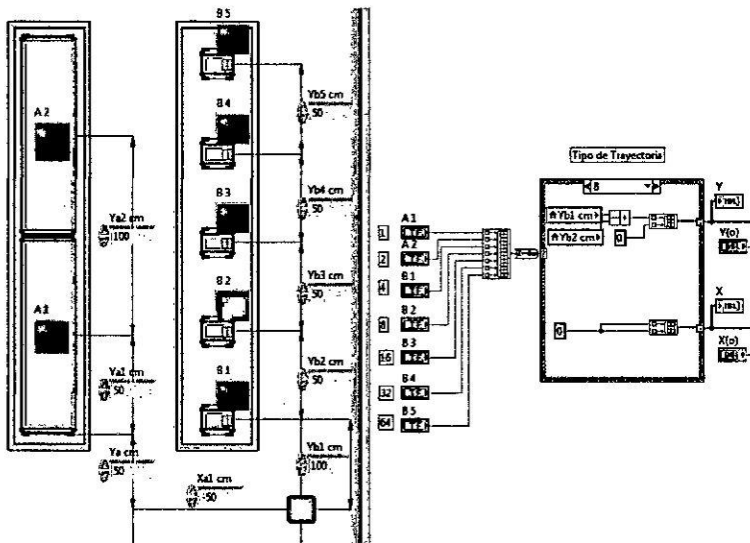


Cuando se presiona B1 el robot se tendrá que desplazar del punto (0; 0) al punto (0; Yb1).

Teniendo la coordenada inicial y final se genera una trayectoria definida:

Primera trayectoria del punto (0; 0) al punto (0; Yb1).

FIGURA N°75 Coordinada Final en el Punto B2

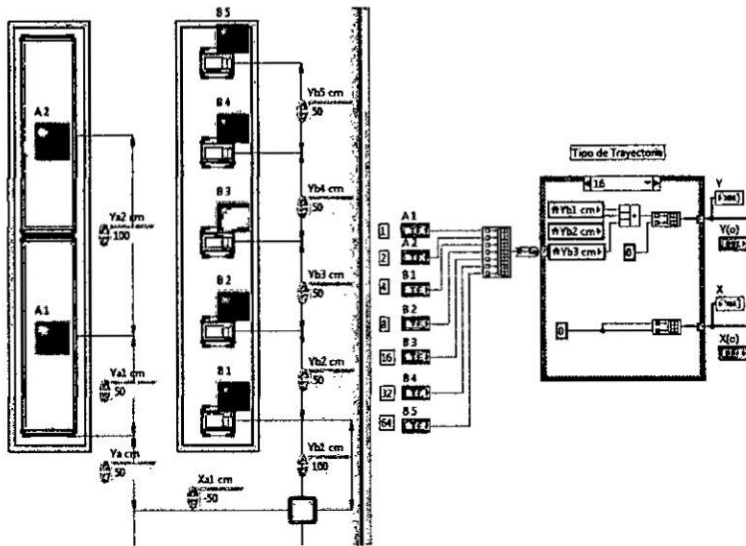


Cuando se presiona B2 el robot se tendrá que desplazar del punto (0; 0) al punto (0; Yb1+Yb2).

Teniendo la coordenada inicial y final se genera una trayectoria definida:

Primera trayectoria del punto (0; 0) al punto (0; Yb1+Yb2).

FIGURA N°76 Coordenada Final en el Punto B3

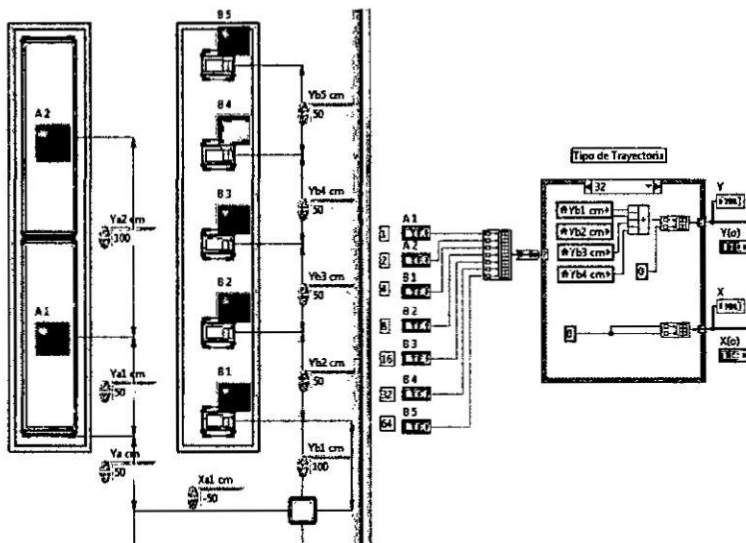


Cuando se presiona B3 el robot se tendrá que desplazar del punto (0; 0) al punto (0; Yb1+Yb2+Yb3).

Teniendo la coordenada inicial y final se genera una trayectoria definida:

Primera trayectoria del punto (0; 0) al punto (0; Yb1+Yb2+Yb3).

FIGURA N°77 Coordenada Final en el Punto B4

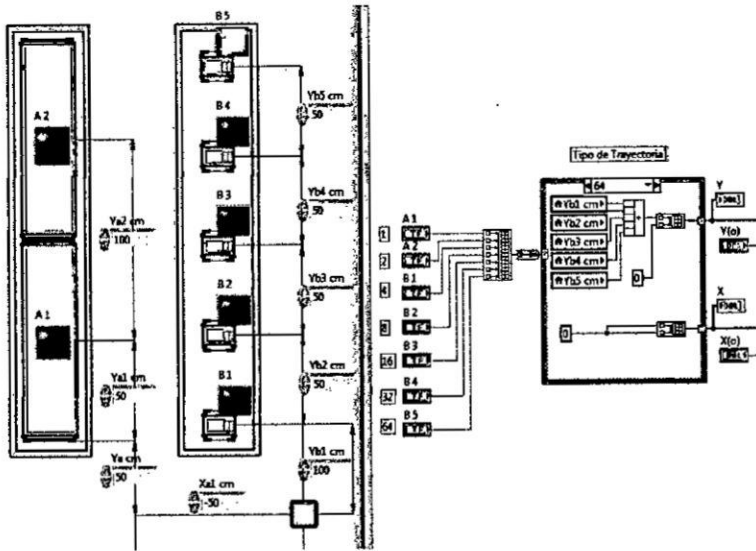


Cuando se presiona B4 el robot se tendrá que desplazar del punto (0; 0) al punto (0; Yb1+Yb2+Yb3+Yb4).

Teniendo la coordenada inicial y final se genera una trayectoria definida:

Primera trayectoria del punto (0; 0) al punto (0; Yb1+Yb2+Yb3+Yb4).

FIGURA N°78 Coordenada Final en el Punto B5



Cuando se presiona B5 el robot se tendrá que desplazar del punto (0; 0) al punto (0; $Yb1+Yb2+Yb3+Yb4+Yb5$).

Teniendo la coordenada inicial y final se genera una trayectoria definida:

Primera trayectoria del punto (0; 0) al punto (0; $Yb1+Yb2+Yb3+Yb4+Yb5$).

INICIALIZACIÓN DE VARIABLES Y CONFIGURACIÓN DE PINES DE ENTRADA Y SALIDA

FIGURA N°79 Inicialización de variables y configuración de pines IO

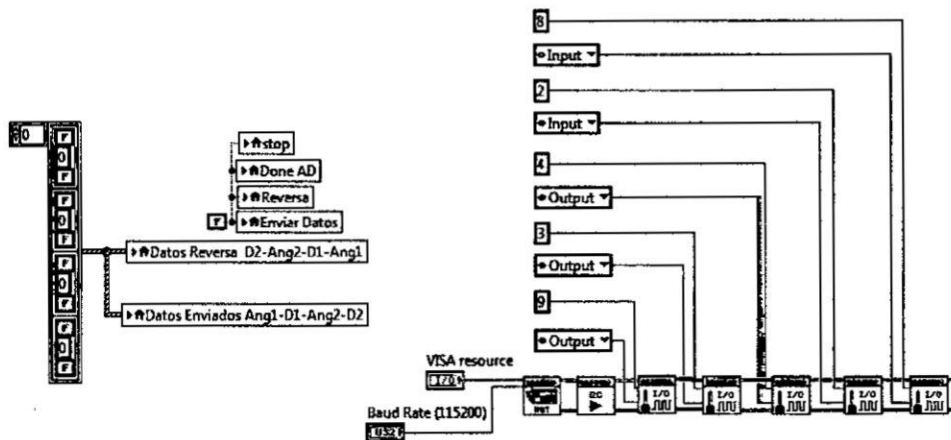
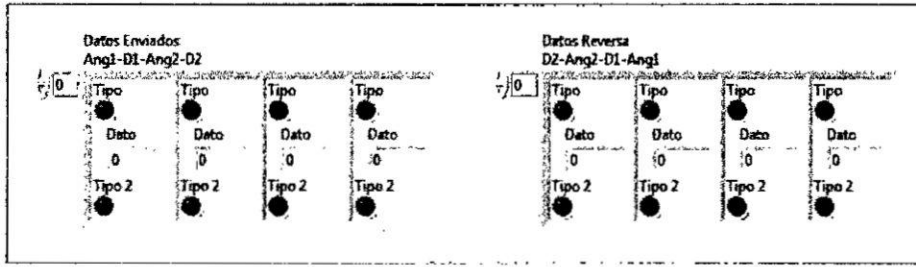


FIGURA N°80 Inicialización de variables



En la *FIGURA N°79* y *FIGURA N°80* se inicializa las variables “Datos Enviados Ang1-D1-Ang2-D2” y “Datos Reversa Ang1-D1-Ang2-D2” en un Array de (False, 0, False).

En la *FIGURA N°79* se configura los pines: PIN 2 (entrada), PIN 3 (salida), PIN 4 (salida), PIN 8 (entrada), PIN 9 (salida). Y se inicializa las variables stop, DoneAD, Reversa, Enviar Datos en Falso.

PIN 2: Lectura cuando el error sea cero.

PIN 8: Lectura cuando el error sea cero.

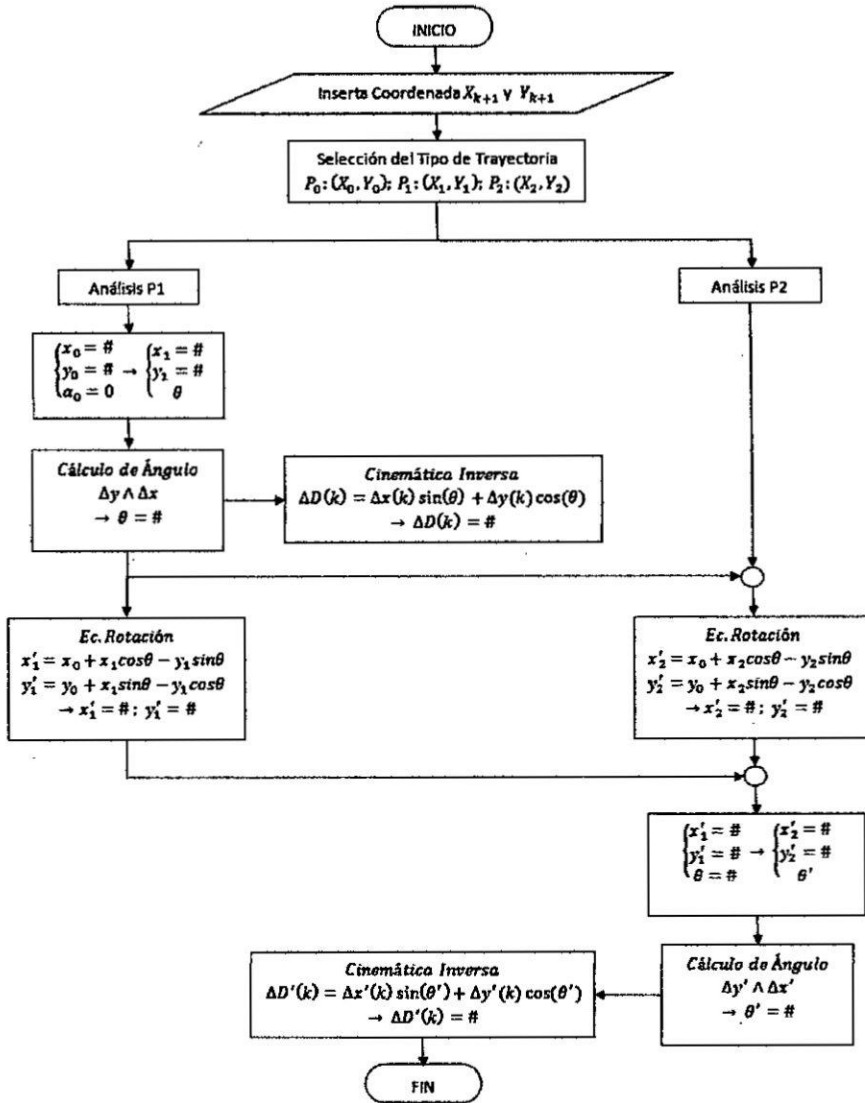
PIN 4: Enviar el Pulso de Start; Después que los módulos Slave almacenen sus datos en la memoria.

PIN 3: Envía el sentido de giro del motor.

PIN 9: Envía el sentido de giro del motor.

DESARROLLO DE LA CINEMÁTICA INVERSA

FIGURA N°81 Diagrama de Flujo para un punto relativo



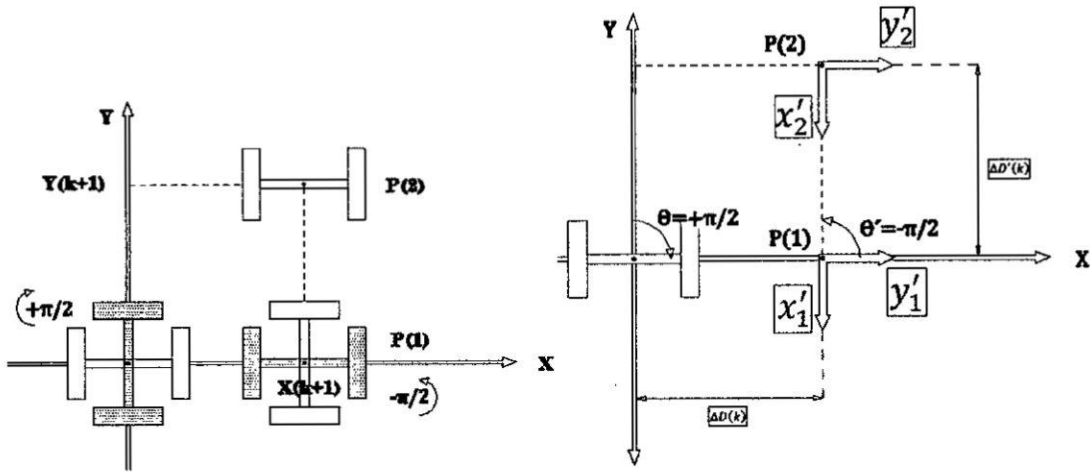
En primer lugar se Inserta las distancias y se selecciona la coordenada.

El análisis se da a través punto a punto el con la condición que una de las variable "x" o "y" se mantengan constante, es decir en dirección del eje "x" o "y". Para poder comprender de manera más didáctica, se analizara a través de cuadrantes.

Con el siguiente Análisis por cuadrante lograremos obtener el valor numérico de la Distancia, Ángulo y sentido de giro para que el robot pueda desplazarse.

Desarrollo de la Cinemática Inversa para Primer Cuadrante

FIGURA N°82 Cinemática Inversa para el primer cuadrante



$$x(k + 1) = 10, \quad y(k + 1) = 10$$

Análisis P1

$$\begin{cases} x_0 = 0 \\ y_0 = 0 \\ \alpha_0 = 0 \end{cases} \rightarrow \begin{cases} x_1 = 10 \\ y_1 = 0 \\ \theta \end{cases}$$

$$\Delta x = x_1 - x_0 = 10$$

$$\Delta y = y_1 - y_0 = 0$$

Calculo del ángulo θ según (Ec. N°5)

Este ángulo se calcula entre P(0) y P(1)

$$\Delta y = 0 \wedge \Delta x > 0 \rightarrow \theta = +\frac{\pi}{2}$$

Cinemática inversa

$$\Delta D(k) = \Delta x(k) \sin(\theta(k)) + \Delta y(k) \cos(\theta(k))$$

$$\Delta D(k) = (10 - 0) \sin\left(\frac{\pi}{2}\right) + (0 - 0) \cos\left(\frac{\pi}{2}\right)$$

$$\Delta D(k) = 10$$

Rotación P1

$$\begin{cases} x_0 = 0 \\ y_0 = 0 \\ \alpha_0 = 0 \end{cases} \rightarrow \begin{cases} x_1 = 10 \\ y_1 = 0 \\ \theta = +\frac{\pi}{2} \end{cases}$$

Ecuación de Rotación

$$x'_1 = x_0 + x_1 \cos \theta - y_1 \sin \theta$$

$$x'_1 = 0 + 10 \cos \left(\frac{\pi}{2} \right) - 0 \sin \left(\frac{\pi}{2} \right)$$

$$x'_1 = 0$$

$$y'_1 = y_0 + x_1 \sin \theta - y_1 \cos \theta$$

$$y'_1 = 0 + 10 \sin \left(\frac{\pi}{2} \right) - 0 \cos \left(\frac{\pi}{2} \right)$$

$$y'_1 = 10$$

Rotación P2

$$\begin{cases} x_0 = 0 \\ y_0 = 0 \\ \alpha_0 = 0 \end{cases} \rightarrow \begin{cases} x_2 = 10 \\ y_2 = 10 \\ \theta = +\frac{\pi}{2} \end{cases}$$

Ecuación de Rotación

$$x'_2 = x_0 + x_2 \cos \theta - y_2 \sin \theta$$

$$x'_2 = 0 + 10 \cos \left(\frac{\pi}{2} \right) - 10 \sin \left(\frac{\pi}{2} \right)$$

$$x'_2 = -10$$

$$y'_2 = y_0 + x_2 \sin \theta - y_2 \cos \theta$$

$$y'_2 = 0 + 10 \sin \left(\frac{\pi}{2} \right) - 10 \cos \left(\frac{\pi}{2} \right)$$

$$y'_2 = 10$$

Tenemos

Análisis P1

$$\begin{cases} x_0 = 0 \\ y_0 = 0 \\ \alpha_0 = 0 \end{cases} \rightarrow \begin{cases} x_1 = -10 \\ y_1 = 0 \\ \theta \end{cases}$$

$$\Delta x = x_1 - x_0 = -10$$

$$\Delta y = y_1 - y_0 = 0$$

Calculo del ángulo θ según (Ec. N°5)

Este ángulo se calcula entre P(0) y P(1)

$$\Delta y = 0 \wedge \Delta x < 0 \rightarrow \theta = -\frac{\pi}{2}$$

Cinemática inversa

$$\Delta D(k) = \Delta x(k) \sin(\theta(k)) + \Delta y(k) \cos(\theta(k))$$

$$\Delta D(k) = (-10 - 0) \sin\left(-\frac{\pi}{2}\right) + (0 - 0) \cos\left(-\frac{\pi}{2}\right)$$

$$\Delta D(k) = 10$$

Rotación P1

$$\begin{cases} x_0 = 0 \\ y_0 = 0 \\ \alpha_0 = 0 \end{cases} \rightarrow \begin{cases} x_1 = -10 \\ y_1 = 0 \\ \theta = -\frac{\pi}{2} \end{cases}$$

Ecuación de Rotación

$$x'_1 = x_0 + x_1 \cos\theta - y_1 \sin\theta$$

$$x'_1 = 0 + (-10) \cos\left(-\frac{\pi}{2}\right) - 0 \sin\left(-\frac{\pi}{2}\right)$$

$$x'_1 = 0$$

$$y'_1 = y_0 + x_1 \sin\theta - y_1 \cos\theta$$

$$y'_1 = 0 + (-10) \sin\left(-\frac{\pi}{2}\right) - 0 \cos\left(-\frac{\pi}{2}\right)$$

$$y'_1 = 10$$

Rotación P2

$$\begin{cases} x_0 = 0 \\ y_0 = 0 \\ \alpha_0 = 0 \end{cases} \rightarrow \begin{cases} x_2 = -10 \\ y_2 = 10 \\ \theta = -\frac{\pi}{2} \end{cases}$$

Ecuación de Rotación

$$x'_2 = x_0 + x_2 \cos \theta - y_2 \sin \theta$$

$$x'_2 = 0 + (-10) \cos \left(-\frac{\pi}{2}\right) - 10 \sin \left(-\frac{\pi}{2}\right)$$

$$x'_2 = 10$$

$$y'_2 = y_0 + x_2 \sin \theta - y_2 \cos \theta$$

$$y'_2 = 0 + (-10) \sin \left(-\frac{\pi}{2}\right) - 10 \cos \left(-\frac{\pi}{2}\right)$$

$$y'_2 = 10$$

Tenemos

$$\begin{cases} x'_1 = 0 \\ y'_1 = 10 \\ \theta = -\frac{\pi}{2} \end{cases} \rightarrow \begin{cases} x'_2 = 10 \\ y'_2 = 10 \\ \theta' \end{cases}$$

$$\Delta x' = x'_2 - x'_1 = 10$$

$$\Delta y' = y'_2 - y'_1 = 0$$

Calculo del ángulo θ' según (Ec. N°5)

Este ángulo se calcula entre P(1) y P(2) con el sistema rotado

$$\Delta y' = 0 \wedge \Delta x' > 0 \rightarrow \theta' = \frac{\pi}{2}$$

Cinemática inversa

$$\Delta D'(k) = \Delta x'(k) \sin(\theta'(k)) + \Delta y'(k) \cos(\theta'(k))$$

$$\Delta D'(k) = (10 - 0) \sin\left(\frac{\pi}{2}\right) + (10 - 10) \cos\left(\frac{\pi}{2}\right)$$

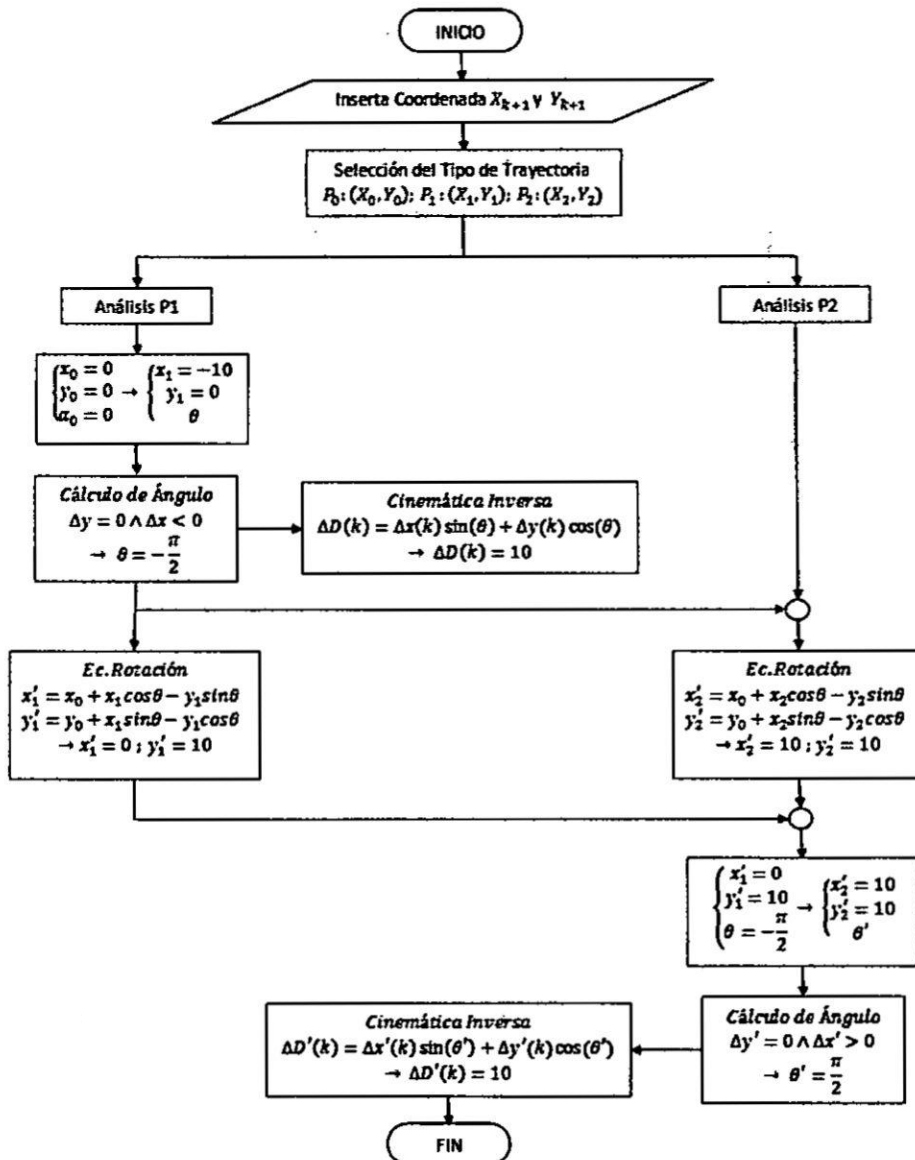
$$\Delta D'(k) = 10$$

Ordenado:

θ	$\Delta D(k)$	θ'	$\Delta D'(k)$
$-\frac{\pi}{2}$	10	$+\frac{\pi}{2}$	10

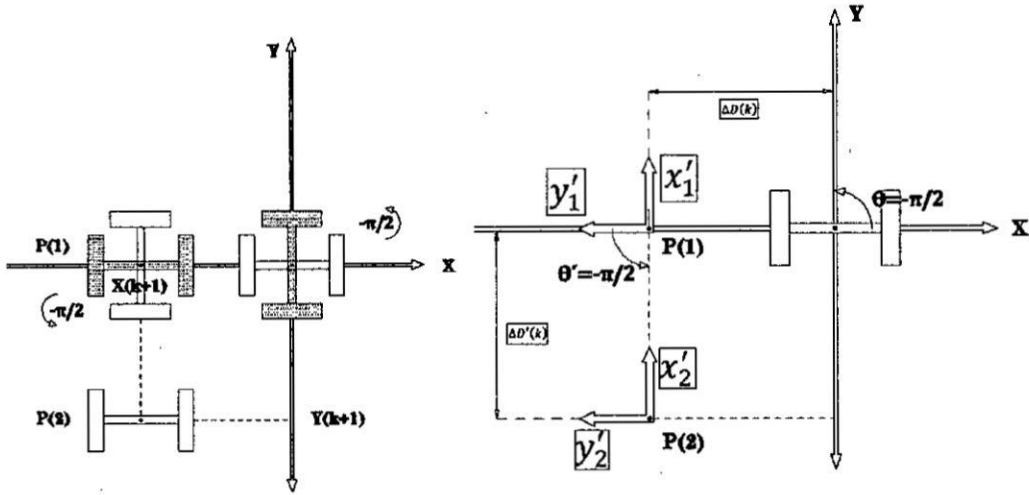
Se obtuvo la secuencia establecida por la cinemática inversa: girar 90° en sentido Antihorario (Signo negativo) y luego desplazarse una distancia de 10cm y a continuación girar 90° Horario (Signo positivo) y luego desplazarse una distancia de 10cm.

FIGURA N°85 Diagrama de Flujo de un punto en el segundo cuadrante



Desarrollo de la Cinemática Inversa para Tercer Cuadrante

FIGURA N°86 Cinemática Inversa para el tercer cuadrante



$$x(k+1) = -10, \quad y(k+1) = -10$$

Análisis P1

$$\begin{cases} x_0 = 0 \\ y_0 = 0 \\ \alpha_0 = 0 \end{cases} \rightarrow \begin{cases} x_1 = -10 \\ y_1 = 0 \\ \theta \end{cases}$$

$$\Delta x = x_1 - x_0 = -10$$

$$\Delta y = y_1 - y_0 = 0$$

Calculo del ángulo θ según (Ec. N°5)

Este ángulo se calcula entre P(0) y P(1)

$$\Delta y = 0 \wedge \Delta x < 0 \rightarrow \theta = -\frac{\pi}{2}$$

Cinemática inversa

$$\Delta D(k) = \Delta x(k) \sin(\theta(k)) + \Delta y(k) \cos(\theta(k))$$

$$\Delta D(k) = (-10 - 0) \sin\left(-\frac{\pi}{2}\right) + (0 - 0) \cos\left(-\frac{\pi}{2}\right)$$

$$\Delta D(k) = 10$$

Rotación P1

$$\begin{cases} x_0 = 0 \\ y_0 = 0 \\ \alpha_0 = 0 \end{cases} \rightarrow \begin{cases} x_1 = -10 \\ y_1 = 0 \\ \theta = -\frac{\pi}{2} \end{cases}$$

Ecuación de Rotación

$$x'_1 = x_0 + x_1 \cos \theta - y_1 \sin \theta$$

$$x'_1 = 0 + (-10) \cos \left(-\frac{\pi}{2}\right) - 0 \sin \left(-\frac{\pi}{2}\right)$$

$$x'_1 = 0$$

$$y'_1 = y_0 + x_1 \sin \theta - y_1 \cos \theta$$

$$y'_1 = 0 + (-10) \sin \left(-\frac{\pi}{2}\right) - 0 \cos \left(-\frac{\pi}{2}\right)$$

$$y'_1 = 10$$

Rotación P2

$$\begin{cases} x_0 = 0 \\ y_0 = 0 \\ \alpha_0 = 0 \end{cases} \rightarrow \begin{cases} x_2 = -10 \\ y_2 = -10 \\ \theta = -\frac{\pi}{2} \end{cases}$$

Ecuación de Rotación

$$x'_2 = x_0 + x_2 \cos \theta - y_2 \sin \theta$$

$$x'_2 = 0 + (-10) \cos \left(-\frac{\pi}{2}\right) - (-10) \sin \left(-\frac{\pi}{2}\right)$$

$$x'_2 = -10$$

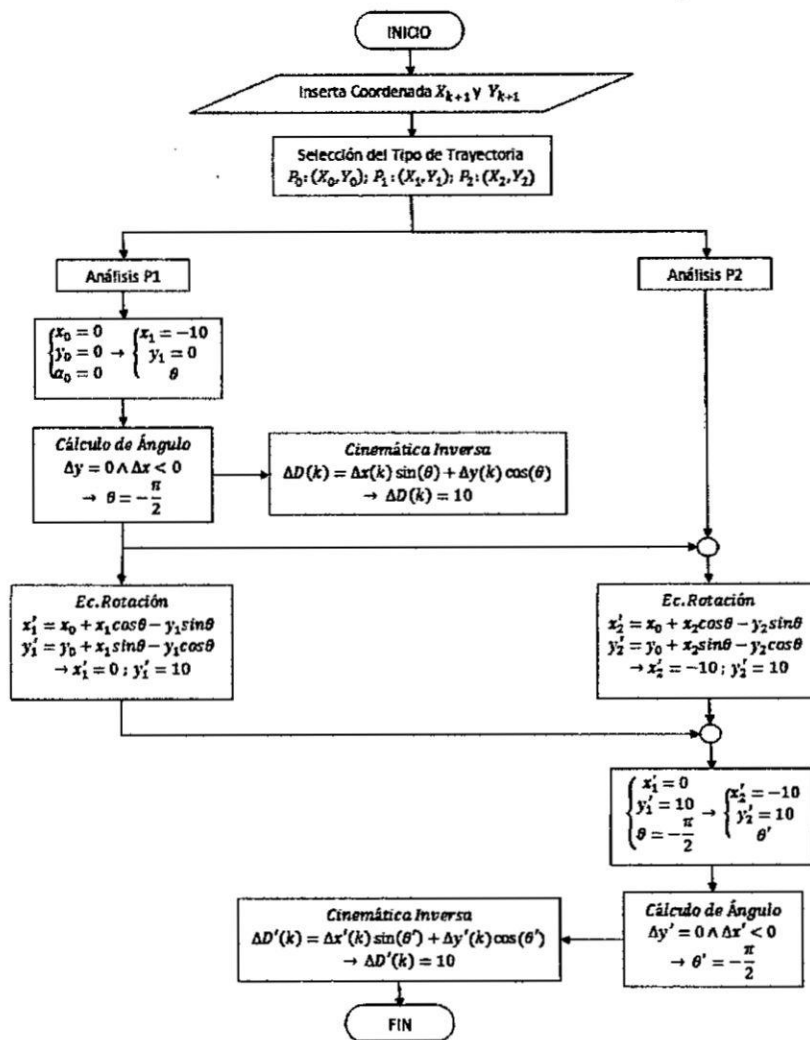
$$y'_2 = y_0 + x_2 \sin \theta - y_2 \cos \theta$$

$$y'_2 = 0 + (-10) \sin \left(-\frac{\pi}{2}\right) - (-10) \cos \left(-\frac{\pi}{2}\right)$$

$$y'_2 = 10$$

Tenemos

FIGURA N°87 Diagrama de Flujo de un punto en el tercer cuadrante



Rotación P1

$$\begin{cases} x_0 = 0 \\ y_0 = 0 \\ \alpha_0 = 0 \end{cases} \rightarrow \begin{cases} x_1 = 10 \\ y_1 = 0 \\ \theta = +\frac{\pi}{2} \end{cases}$$

Ecuación de Rotación

$$x'_1 = x_0 + x_1 \cos \theta - y_1 \sin \theta$$

$$x'_1 = 0 + 10 \cos \left(\frac{\pi}{2} \right) - 0 \sin \left(\frac{\pi}{2} \right)$$

$$x'_1 = 0$$

$$y'_1 = y_0 + x_1 \sin \theta - y_1 \cos \theta$$

$$y'_1 = 0 + 10 \sin \left(\frac{\pi}{2} \right) - 0 \cos \left(\frac{\pi}{2} \right)$$

$$y'_1 = 10$$

Rotación P2

$$\begin{cases} x_0 = 0 \\ y_0 = 0 \\ \alpha_0 = 0 \end{cases} \rightarrow \begin{cases} x_2 = 10 \\ y_2 = -10 \\ \theta = +\frac{\pi}{2} \end{cases}$$

Ecuación de Rotación

$$x'_2 = x_0 + x_2 \cos \theta - y_2 \sin \theta$$

$$x'_2 = 0 + 10 \cos \left(\frac{\pi}{2} \right) - (-10) \sin \left(\frac{\pi}{2} \right)$$

$$x'_2 = 10$$

$$y'_2 = y_0 + x_2 \sin \theta - y_2 \cos \theta$$

$$y'_2 = 0 + 10 \sin \left(\frac{\pi}{2} \right) - (-10) \cos \left(\frac{\pi}{2} \right)$$

$$y'_2 = 10$$

Tenemos

$$\begin{cases} x'_1 = 0 \\ y'_1 = 10 \\ \theta = \frac{\pi}{2} \end{cases} \rightarrow \begin{cases} x'_2 = 10 \\ y'_2 = 10 \\ \theta' \end{cases}$$

$$\Delta x' = x'_2 - x'_1 = 10$$

$$\Delta y' = y'_2 - y'_1 = 0$$

Calculo del ángulo θ' según (Ec. N°5)

Este ángulo se calcula entre P(1) y P(2) con el sistema rotado

$$\Delta y' = 0 \wedge \Delta x' > 0 \rightarrow \theta' = +\frac{\pi}{2}$$

Cinemática inversa

$$\Delta D'(k) = \Delta x'(k) \sin(\theta(k)) + \Delta y'(k) \cos(\theta(k))$$

$$\Delta D'(k) = (10 - 0) \sin\left(\frac{\pi}{2}\right) + (10 - 10) \cos\left(\frac{\pi}{2}\right)$$

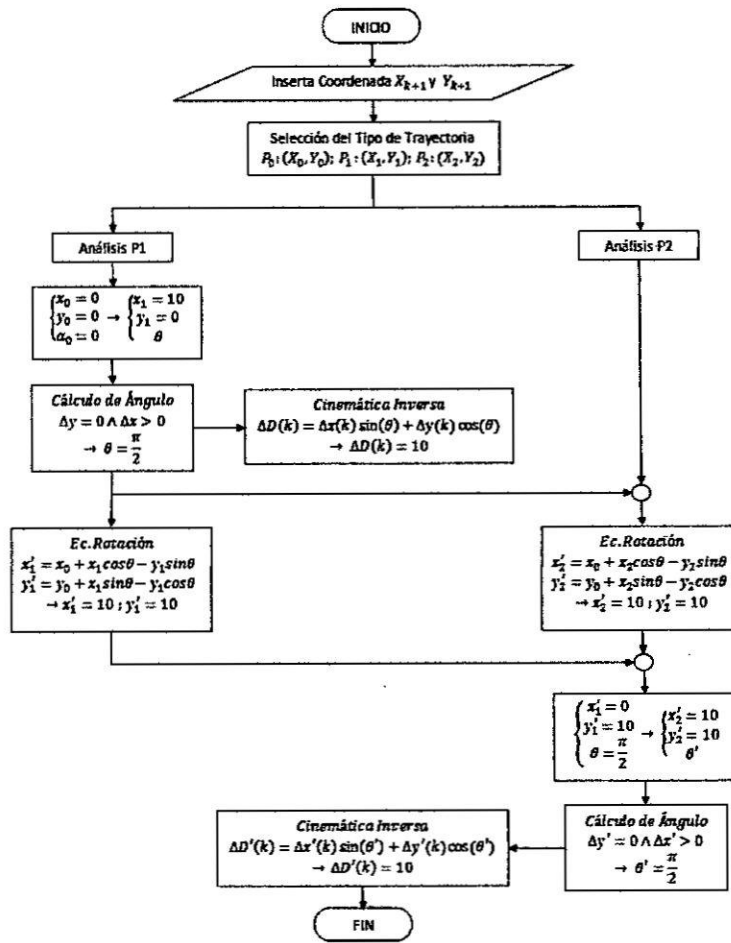
$$\Delta D'(k) = 10$$

Ordenado:

θ	$\Delta D(k)$	θ'	$\Delta D'(k)$
$+\frac{\pi}{2}$	10	$+\frac{\pi}{2}$	10

Se obtuvo la secuencia establecida por la cinemática inversa: girar 90° en sentido Horario (Signo positivo) y luego desplazarse una distancia de 10cm y a continuación girar 90° Horario (Signo positivo) y luego desplazarse una distancia de 10cm.

FIGURA N°89 Diagrama de Flujo de un punto en el cuarto cuadrante



Se analizó la cinemática inversa para los 4 cuadrantes. A continuación nos guiaremos de la FIGURA N°81 “Diagrama de Flujo para un punto relativo” y las ecuaciones Ec. N°3, Ec. N°4, Ec. N°5, Ec. N°6. Para introducir las ecuaciones en Labview.

FIGURA N°90 Cálculo del ángulo del punto 1

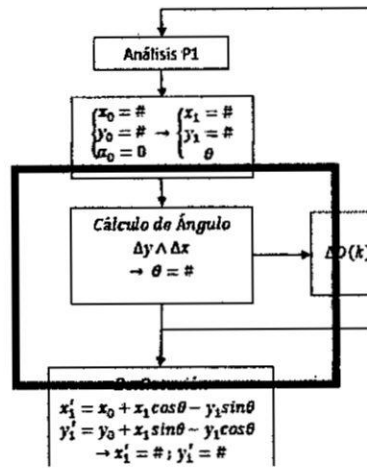


FIGURA N°91 Desarrollo del cálculo del ángulo en Labview $x \neq 0$

$$y > 0 \begin{cases} x > 0 \rightarrow \theta = +\left(\frac{\pi}{2} - \alpha\right) \\ x < 0 \rightarrow \theta = -\left(\frac{\pi}{2} - \alpha\right) \end{cases}$$

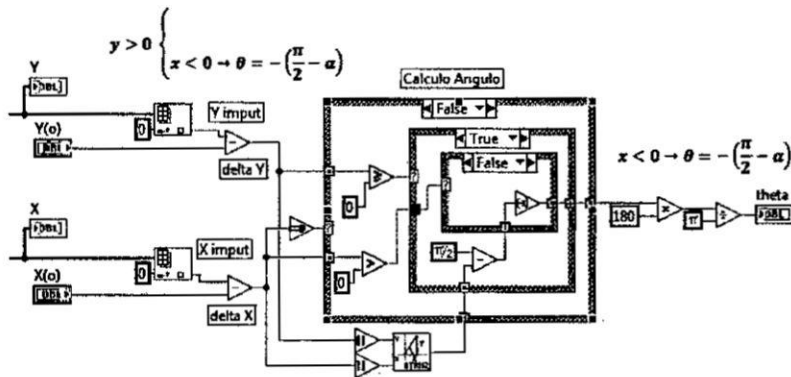
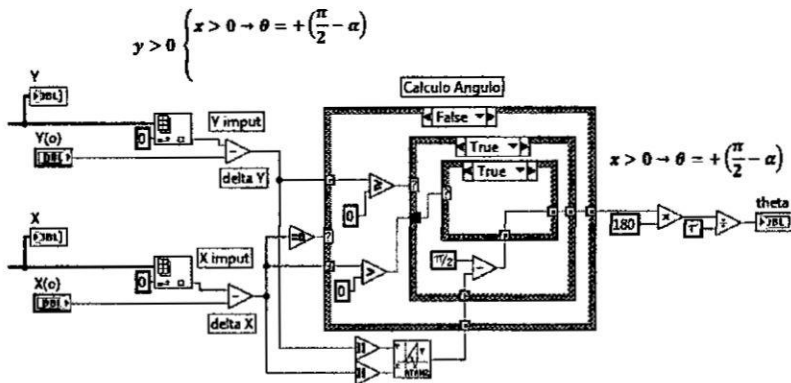


FIGURA N°92 Desarrollo del cálculo del ángulo en Labview $x \neq 0$

$$y < 0 \begin{cases} x > 0 \rightarrow \theta = +\left(\frac{\pi}{2} + \alpha\right) \\ x < 0 \rightarrow \theta = -\left(\frac{\pi}{2} + \alpha\right) \end{cases} \quad y = 0 \begin{cases} x > 0 \rightarrow \theta = +\frac{\pi}{2} \\ x < 0 \rightarrow \theta = -\frac{\pi}{2} \end{cases}$$

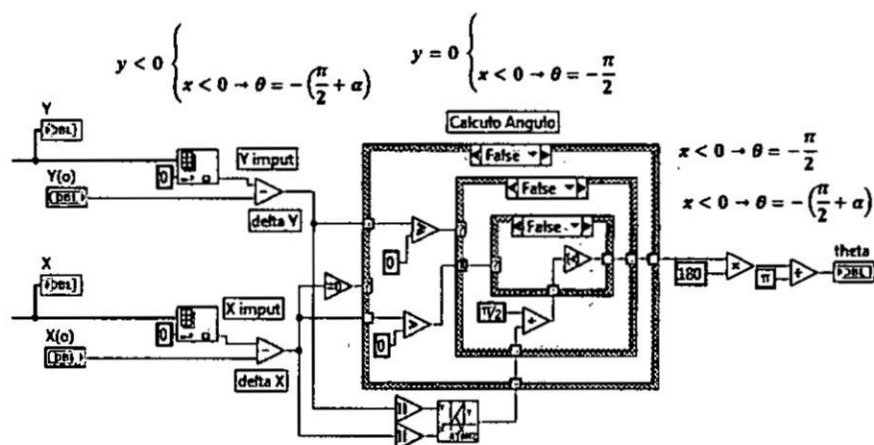
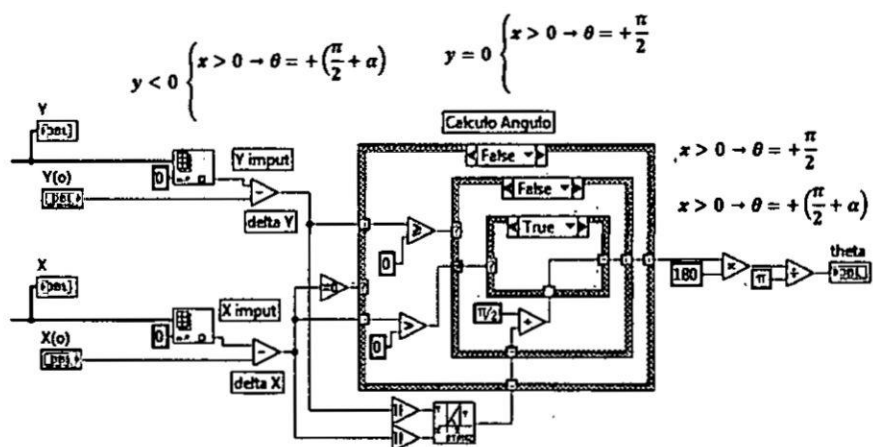


FIGURA N°94 Cálculo de la cinemática inversa del punto 1

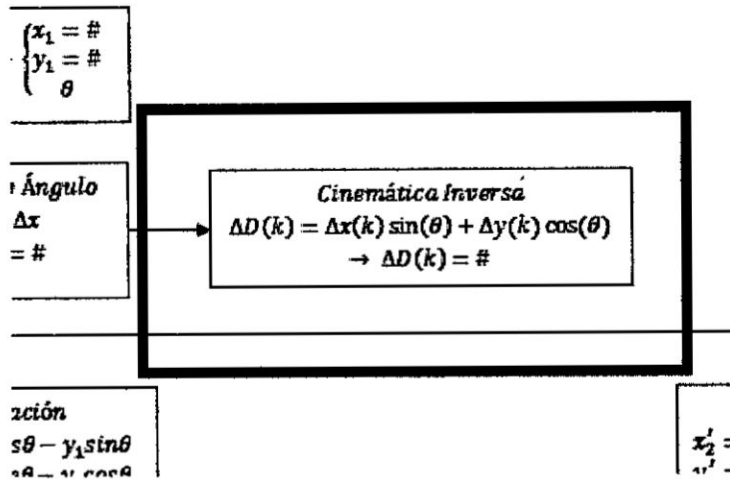


FIGURA N°95 Cálculo de la cinemática inversa en Labview

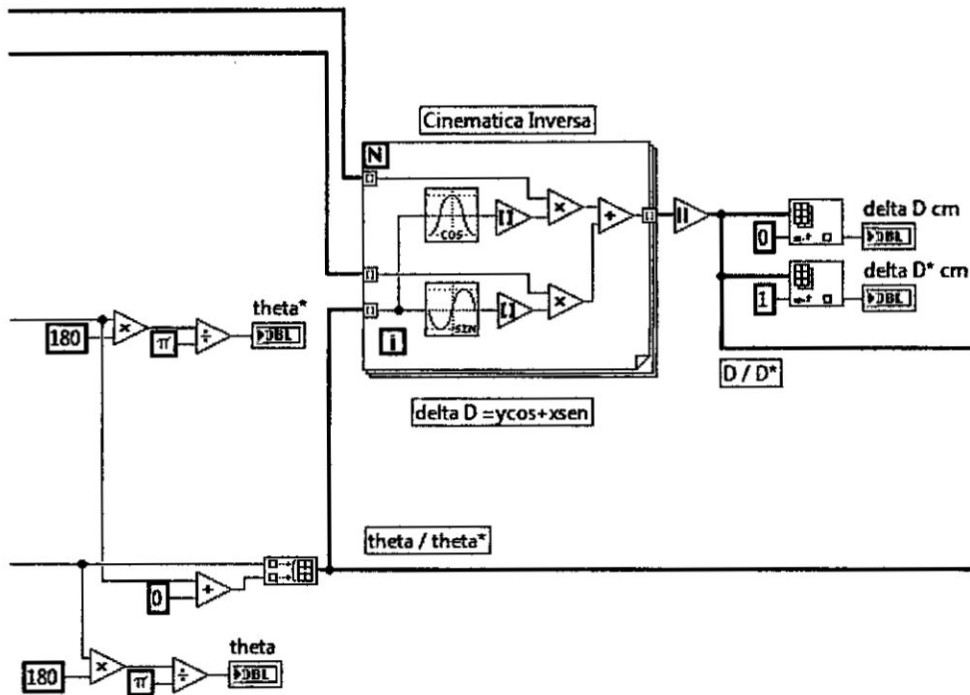


FIGURA N°96 Cálculo de la ecuación de rotación del punto 1 y punto 2

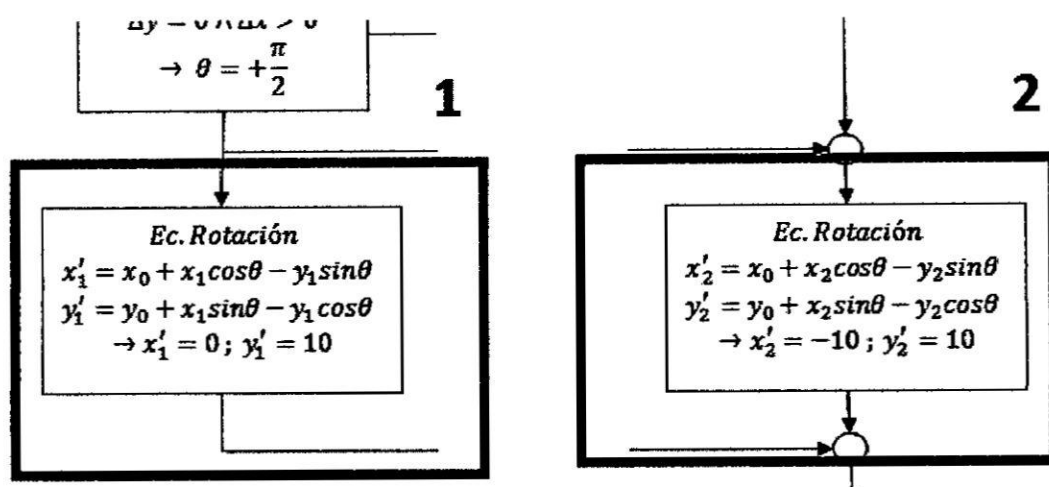


FIGURA N°97 Cálculo de la ecuación de rotación en Labview

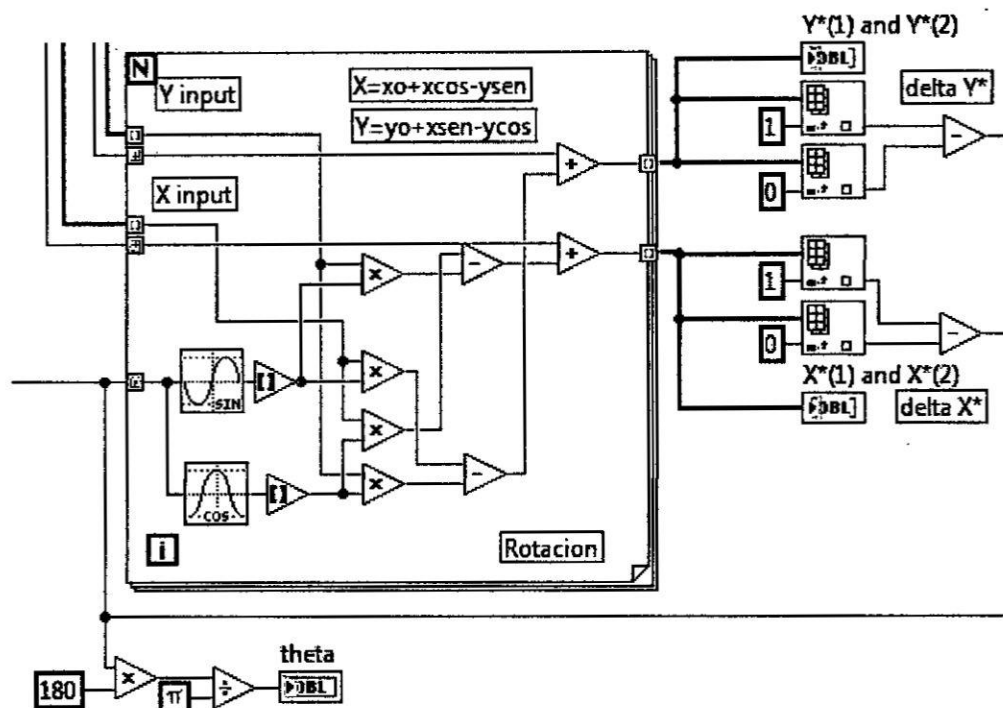


FIGURA N°98 Cálculo del ángulo del punto 2

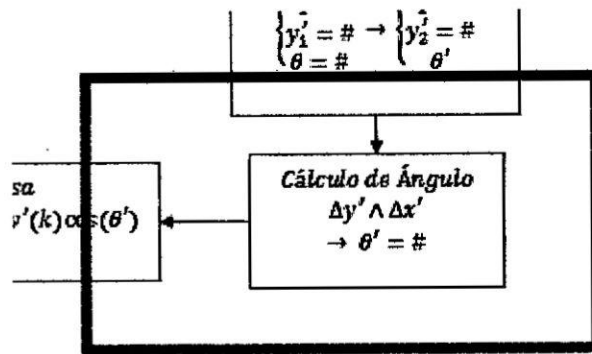


FIGURA N°99 Cálculo del ángulo en Labview

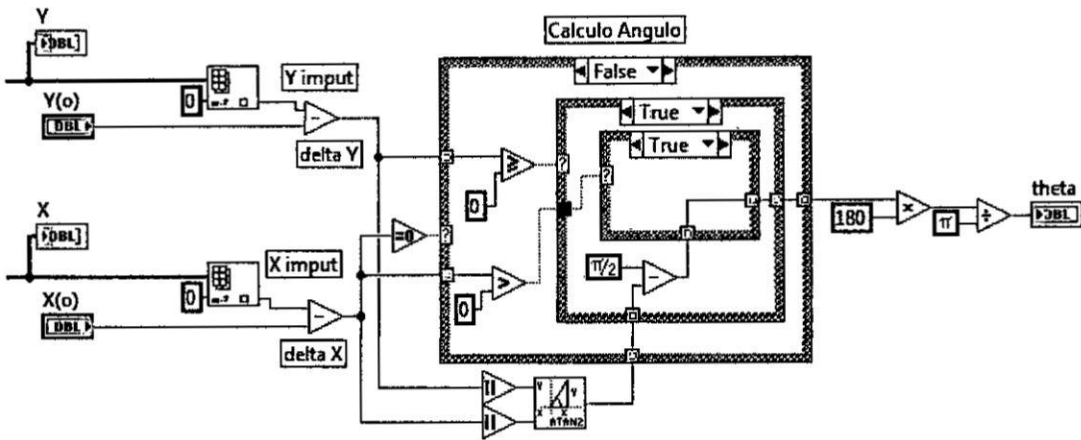


FIGURA N°100 Cálculo de la cinemática inversa del punto 2

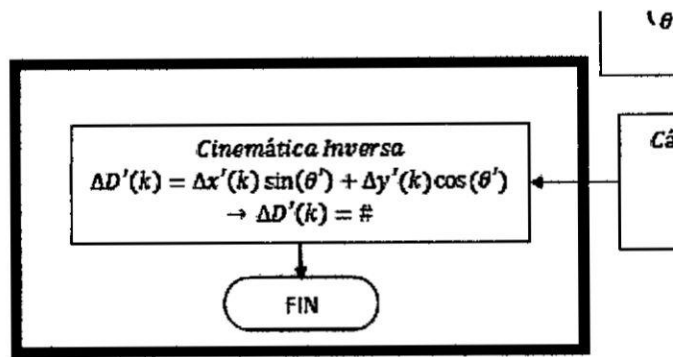
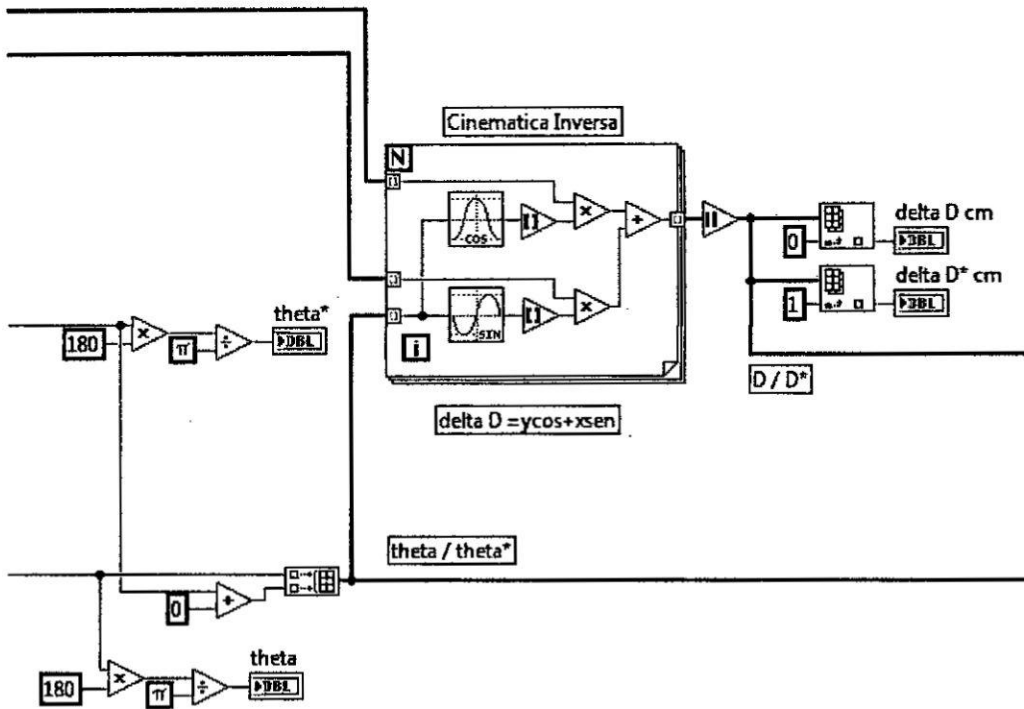
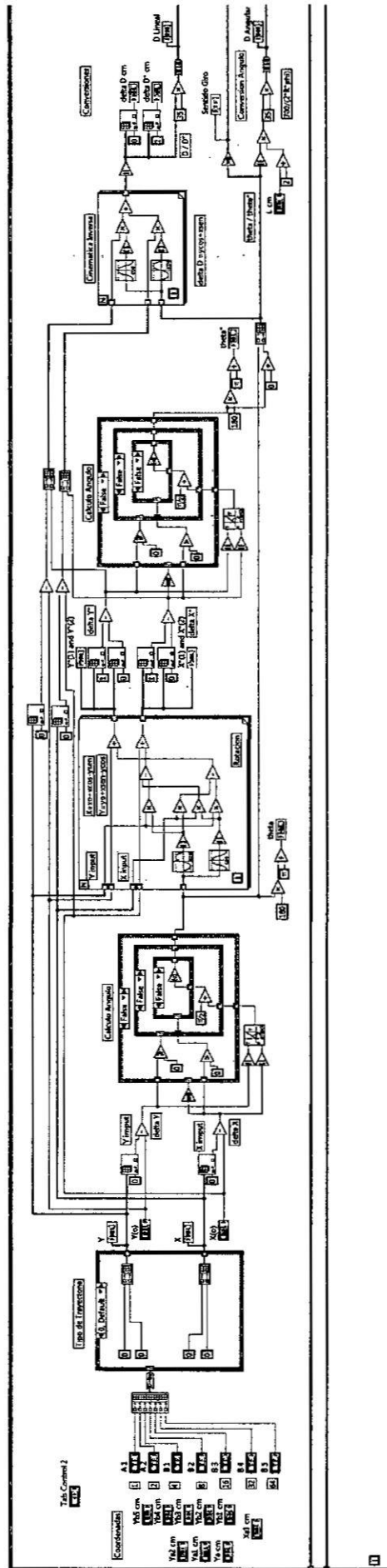


FIGURA N°101 Cálculo de la cinemática inversa en Labview





Conversión de Distancia Angular (cm) a pulsos

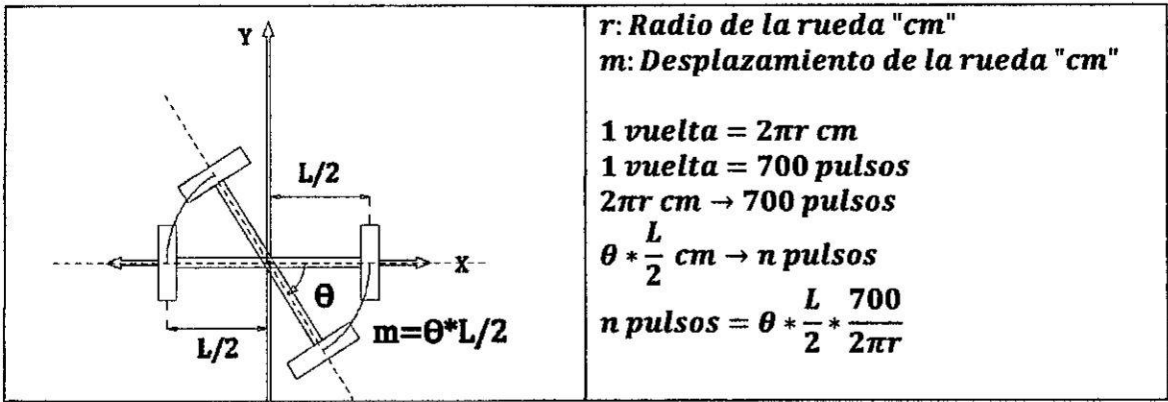
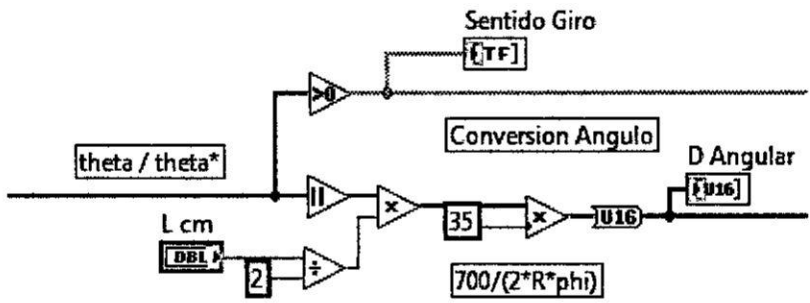


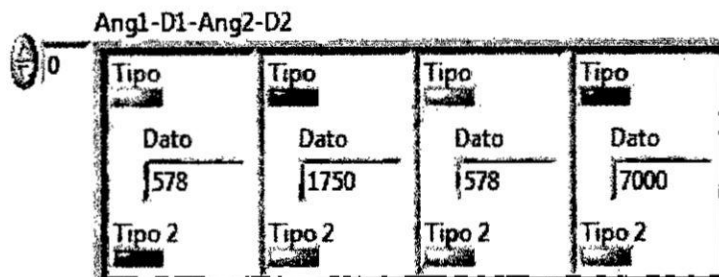
FIGURA N°103 Conversión de Distancia Angular a pulsos



En la FIGURA N°103 se ingresa los datos de los ángulos en radianes y se multiplica por el factor 35 para obtener la cantidad de pulsos que el microcontrolador ejecutara. El sentido se giro se analiza el signo del ángulo de acuerdo a lo analizado en la Cinemática directa.

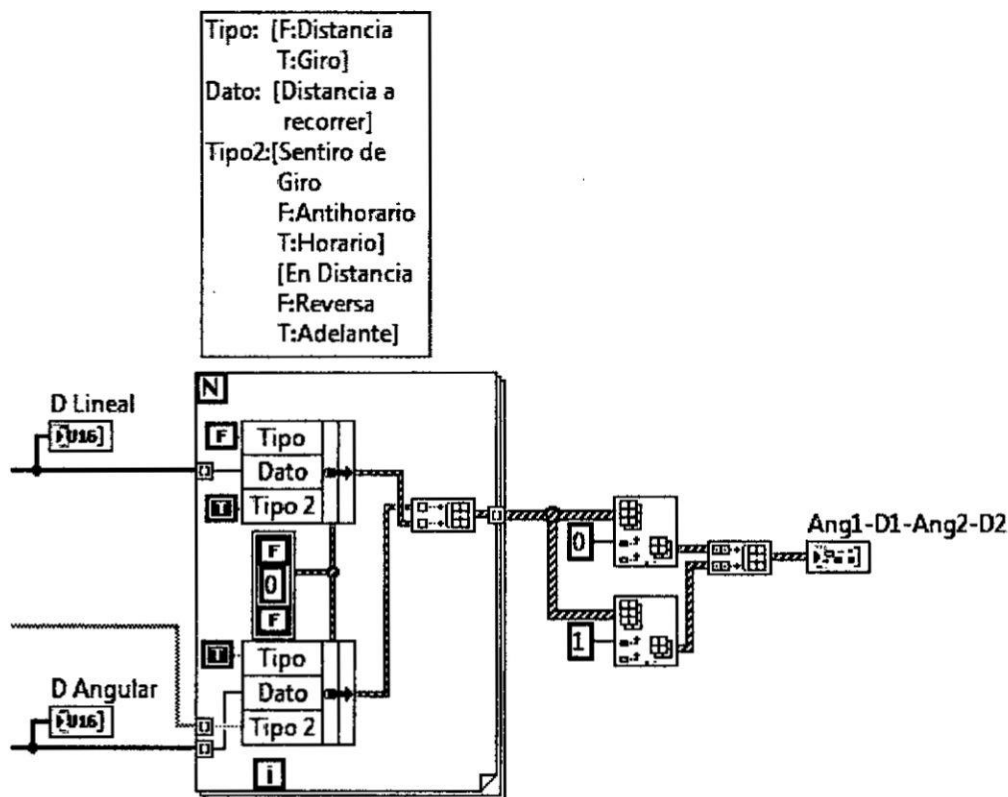
Ordenamiento de los datos.

Para poder enviar los datos al microcontrolador es necesario ordenarlos en un array.



Tipo: { *False: Distancia Lineal* { *Tipo2:* { *False: Reversa*
True: Adelante
True: Giro en sobre su eje { *Tipo2:* { *False: Antihorario*
True: Horario
Dato: { *Distancia a recorrer (Lineal o Angular)*

FIGURA N°104 Ordenamiento de los datos para mantener la secuencia



Desarrollo de envío de datos al microcontrolador.

FIGURA N°105 Diagrama de Flujo de envío de datos al microcontrolador

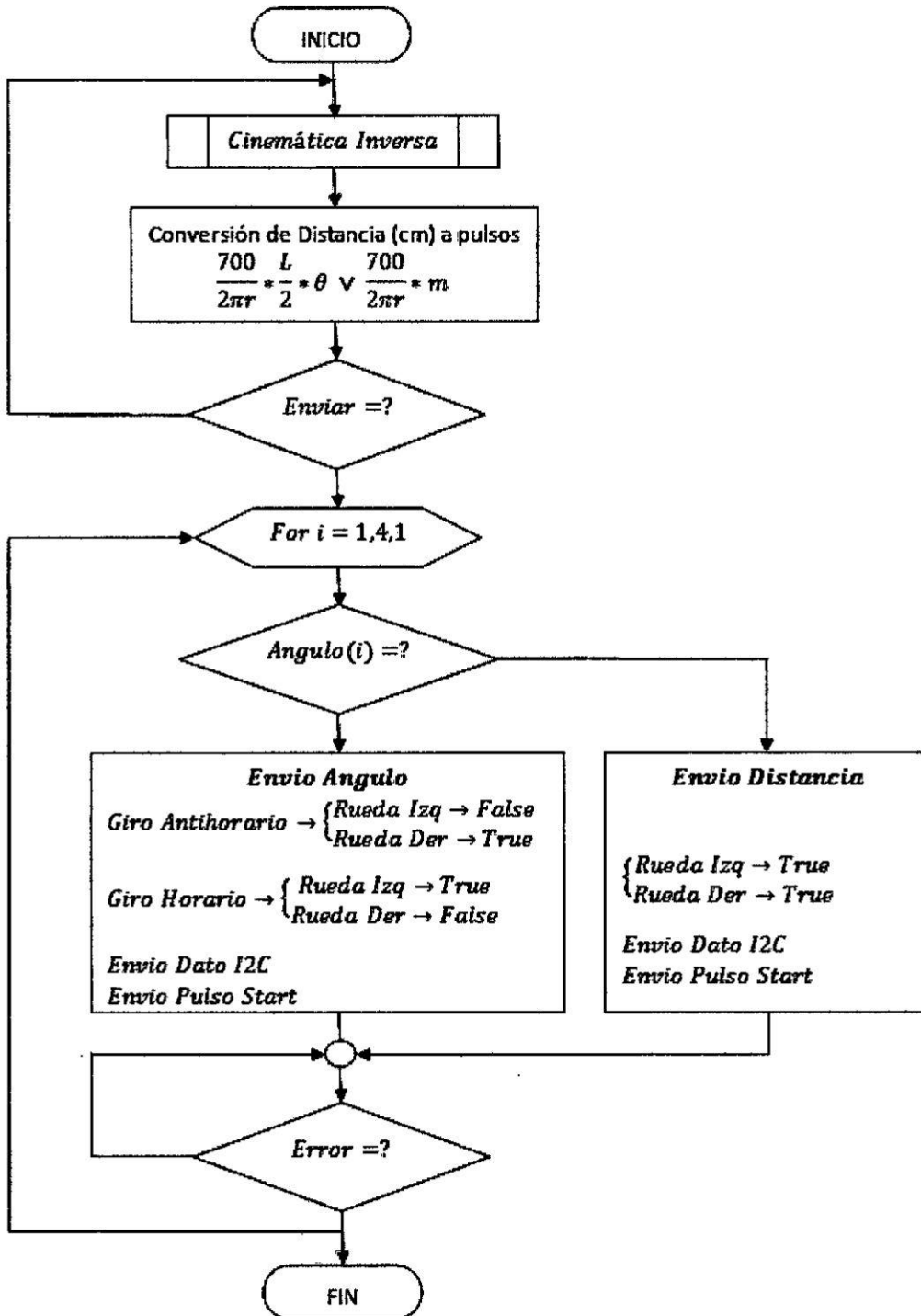


FIGURA N°106 Envió del ángulo en Labview

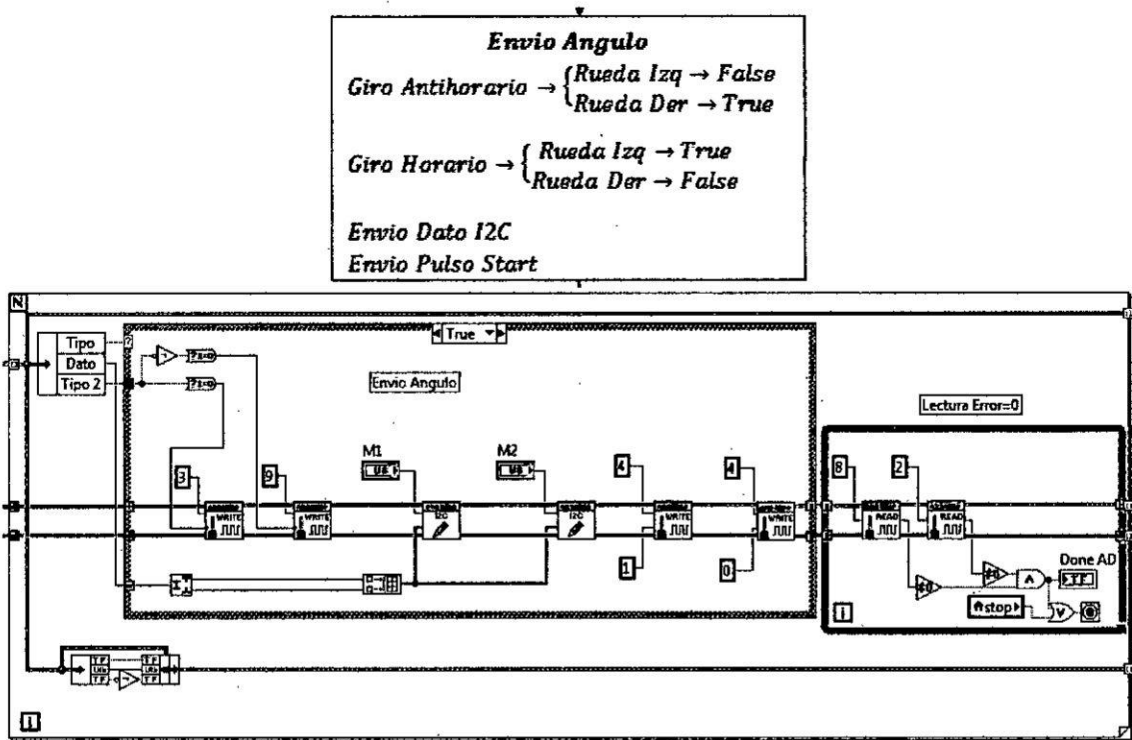
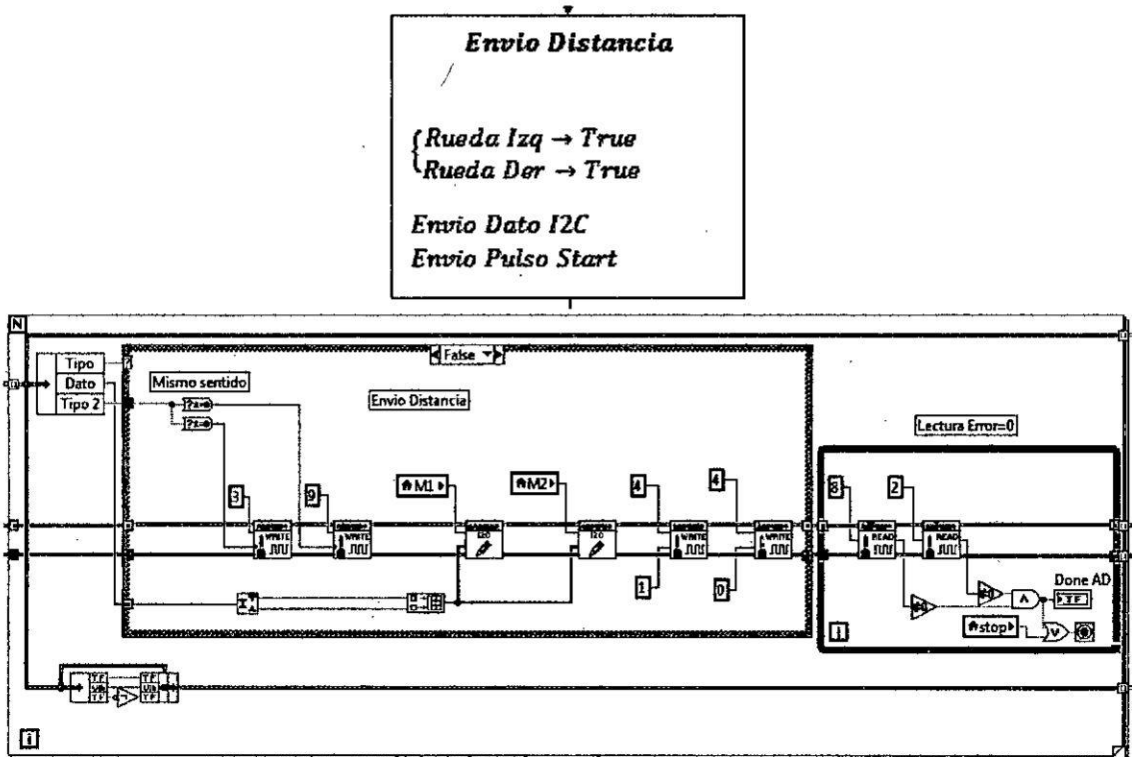
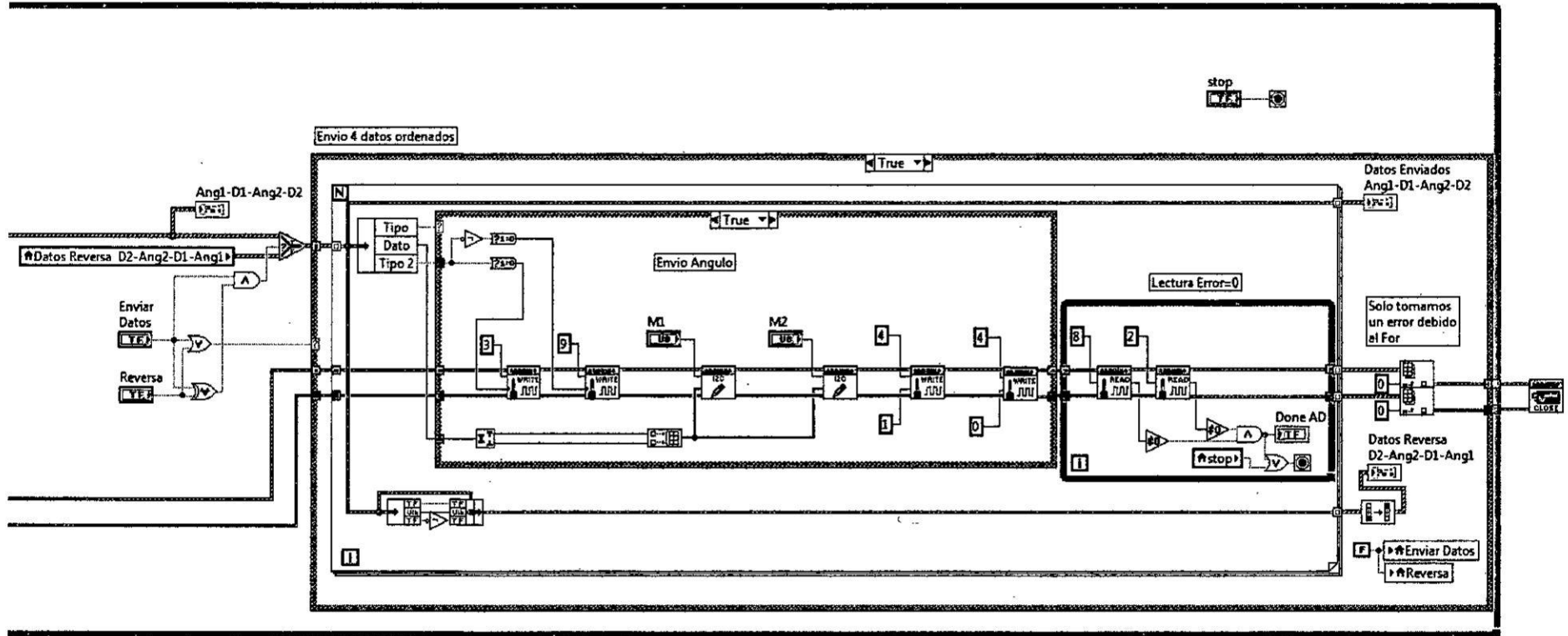
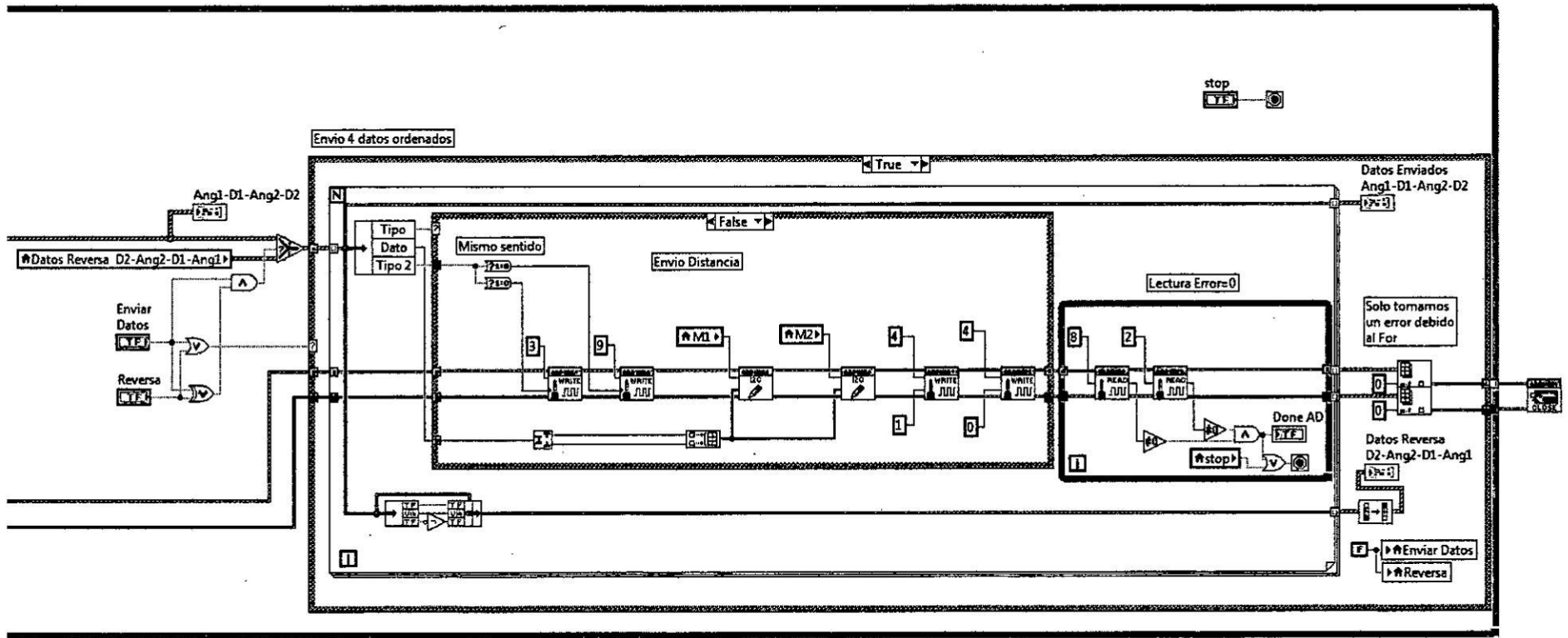
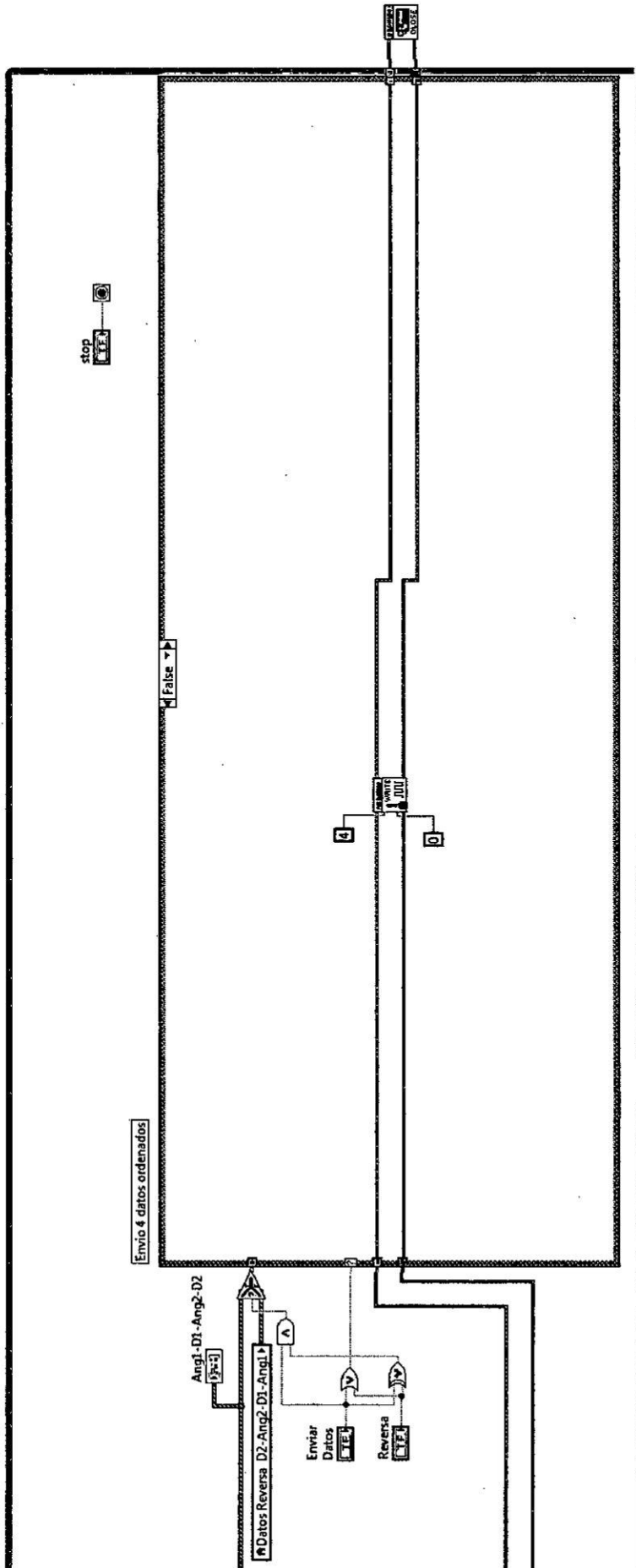


FIGURA N°107 Envió de distancia en Labview







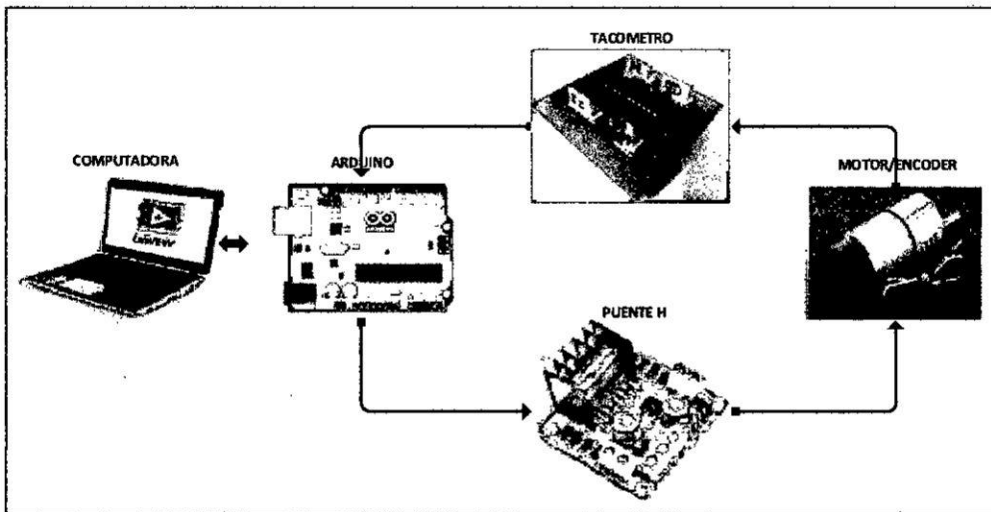


4.2.14. Adquisición, identificación y Control para un motor

SISTEMA DE ADQUISICIÓN DE DATOS

Para la preparación del sistema de adquisición de datos tenemos un diagrama de hardware que nos permitirá entender de manera sencilla todo lo que involucra el hardware de adquisición.

FIGURA N°108 Diagrama del sistema de adquisición de datos



Como se observa en la *FIGURA N°108* muestra el diagrama del hardware comenzando de izquierda tenemos el bloque computadora que se comunica al sistema a través del bloque arduino, luego tenemos el bloque tacómetro que se encarga de leer los datos del encoder, en paralelo tenemos el bloque Puente H que se encarga de acondicionar la señal de escritura (PWM) al motor/Encoder.

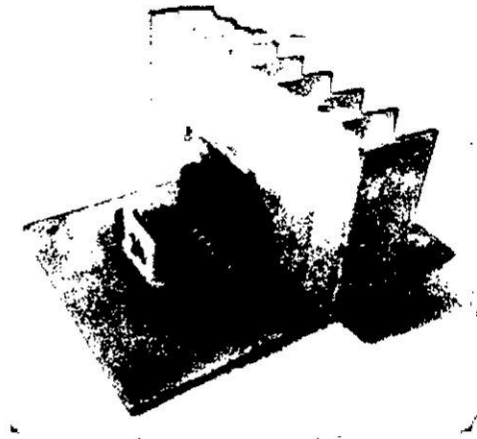
Lista de componentes.

- Computadora.
- Arduino.
- Tacómetro.
- Puente H.
- Motor.
- Batería.

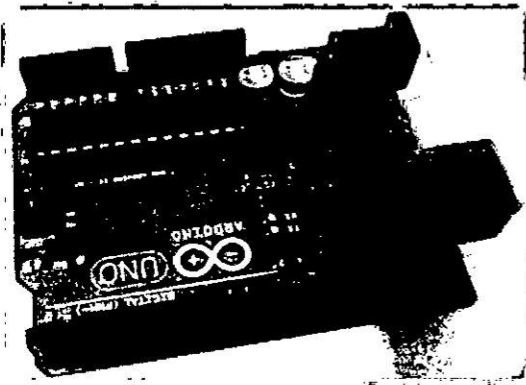
FIGURA N°109 Componentes de Sistema de Adquisición de Datos



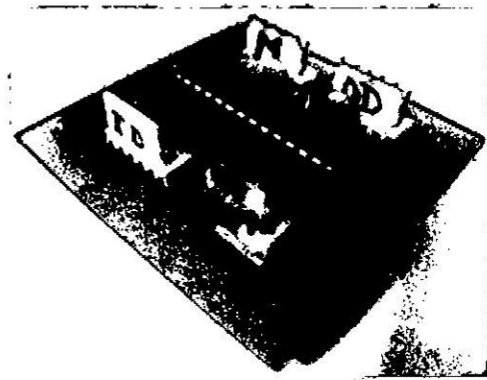
Batería de 12 voltios.



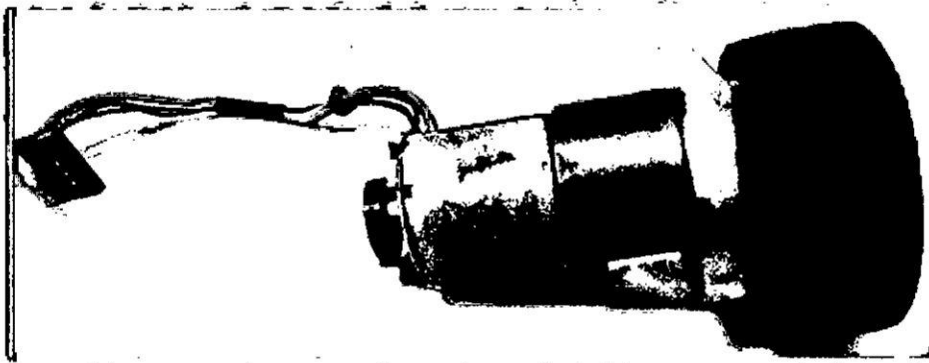
Puente H.



Arduino Uno Versión R3.



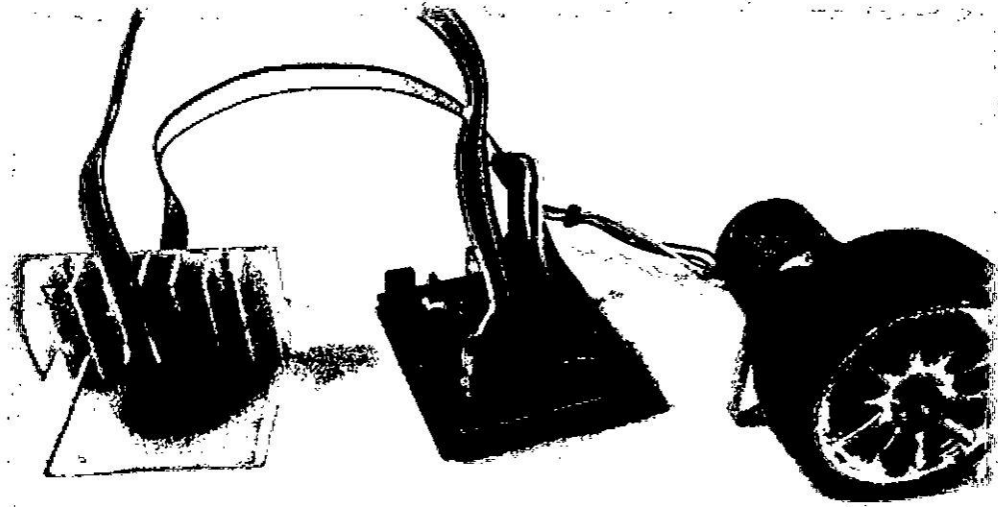
Tacómetro Digital.



Motor/Encoder.

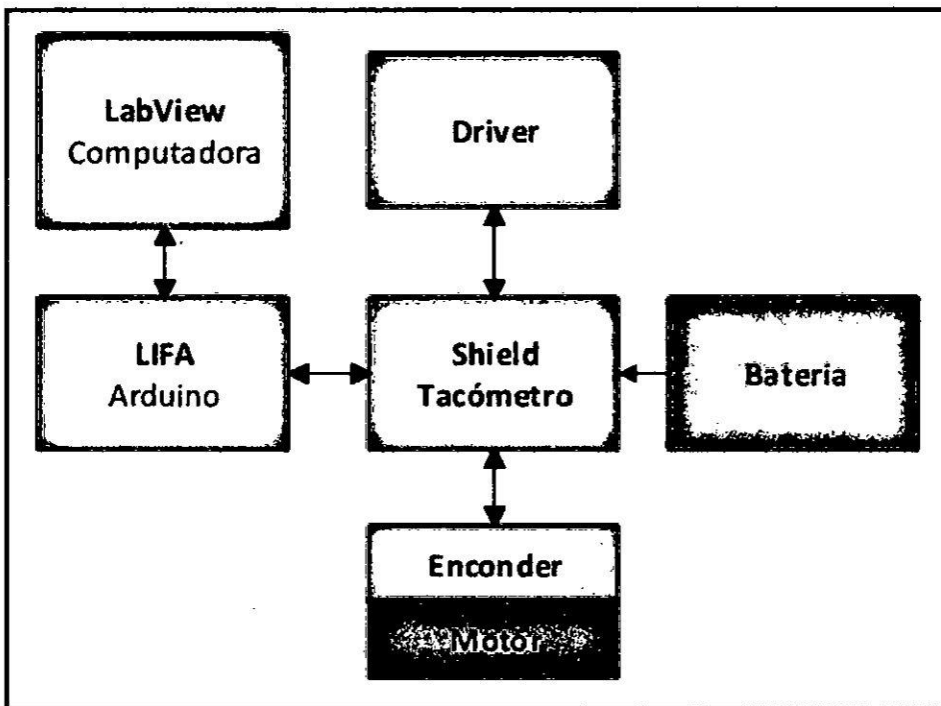
Conexión de los módulos de adquisición.

FIGURA N°110 Sistema de Adquisición



La **FIGURA N°110** nos muestra la conexión de los módulos, la conexión de todos los módulos se da a través de la tarjeta tacómetro que nos permite distribuir todos los conectores de los módulos para que se enlacen entre sí, la conexión se muestra en el siguiente diagrama de bloques.

FIGURA N°111 Diagrama de bloques de conexión




La ejecución consta de cargar todos los programas en sus respectivos módulos y de efectuar la adquisición.

Programación del Shield Tacómetro Digital.

FIGURA N°112 Lógica del programa tacómetro Digital

```
Tacometro_Digital Arduino 1.6.7
Archivo Editar Programa Herramientas Ayuda

Tacometro_Digital$
1. #include <Wire.h> // Iniciamos Libreria I2C
2 // Canal del Encoder
3 int canal_A = 0; //estado pin2
4 // Variables de Tiempos
5 volatile unsigned long time1=0; // Tiempo flanco 1er flanco
6 volatile unsigned long time2=0; // Tiempo flanco 2do flanco
7 volatile unsigned long Periodo=0; // Periodo de del canal A
8 unsigned long time; // Cronometro
9 // frecuencia
10 int f;
11 // Variables Filtro
12 float alpha=0.85; // Polo frecuencia
13 long f_filtrada; // Señal Filtrada
14 long f_filtrada_k_1=0; // Señal Filtrada anterior
15 // Mando
16 int Entrada=11; // Setpoint
17 volatile boolean cambio = false; //variable Booleana
18 void setup()
19 { Wire.begin(4); // Iniciando Direccion I2C
20 Wire.onRequest(requestEvent); // Habilitando Solicitud de Evento
21 attachInterrupt(canal_A, encoderIntA, RISING); // Habilitando Interrupcion Canal A
22 time = micros(); // Iniciando cronometro 0
23 pinMode(Entrada, INPUT); //modo entrada Setpoint
24 }
25 void loop()
26 {
27
28 void encoderIntA() {
29
30 if(cambio==false){
31 time1=micros(); // Asignando dato de reloj a "time1"
32 Periodo=abs(time1-time2); // "time3" Periodo
33 }
34
35 if(cambio==true){
36 time2=micros(); // Asignando dato de reloj a "time1"
37 Periodo=abs(time2-time1); // "time3" Periodo
38 }
39
40 f=27210/Periodo; // frecuencia
41 f_filtrada=(f*(1-alpha))+alpha*f_filtrada_k_1; // Señal filtrada frecuencia
42 f_filtrada_k_1=f_filtrada; // Señal Filtrada anterior
43 if(digitalRead(Entrada)==LOW){ // Estableciendo la salida a 0
44 f_filtrada=f_filtrada-10;
45 if(f_filtrada<0){
46 f_filtrada=0;
47 }
48 }
49 cambio = !cambio;
50 }
51
52 void requestEvent() {
53 Wire.write(f_filtrada); // Escribiendo Señal Filtrada I2C
54 }
55 }
```



Segmentos del programa Tacometro Digital.

Cada segmento del programa esta debidamente detallado como se observan en las siguientes figuras.

FIGURA N°113 Segmento de Inicialización de variables

```
1 #include <Wire.h> // Iniciamos Libreria I2C
2 // Canal del Encoder
3 int canal_A = 0; //estado pin2
4 // Variables de Tiempos
5 volatile unsigned long time1=0; // Tiempo flanco 1er flanco
6 volatile unsigned long time2=0; // Tiempo flanco 2do flanco
7 volatile unsigned long Periodo=0; // Periodo de del canal A
8 unsigned long time; // Cronometro
9 // frecuencia
10 int f;
11 // Variables Filtro
12 float alpha=0.85; // Polo frecuencia
13 long f_filtrada; // Señal Filtrada
14 long f_filtrada_k_1=0; // Señal Filtrada anterior
15 // Mando
16 int Entrada=11; // Setpoint
17 volatile boolean cambio = false; //variable Booleana
```

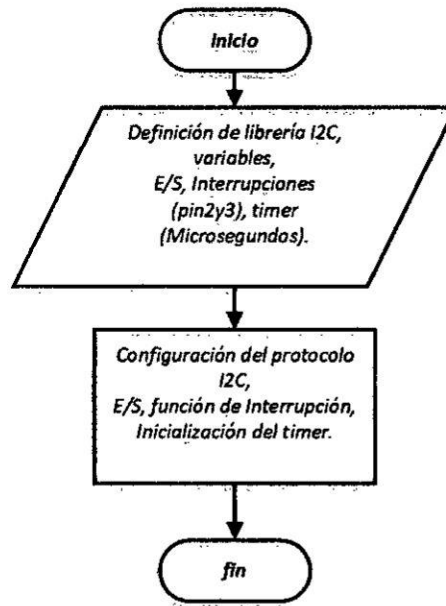
FIGURA N°114 Segmento de Configuración

```
18 void setup()
19 { Wire.begin(4); // Iniciando Direccion I2C
20 Wire.onRequest(requestEvent); // Habilitando Solicitud de Evento
21 attachInterrupt(canal_A, encoderIntA, RISING); // Habilitando Interrupcion Canal A
22 time = micros(); // Iniciando cronometro 0
23 pinMode(Entrada, INPUT); //modo entrada Setpoint
24 }
```

FIGURA N°115 Segmento de Lógica

```
25 void loop()
26 {
27
28 void encoderIntA() {
29
30 if(cambio==false){
31 time1=micros(); // Asignando dato de reloj a "time1"
32 Periodo=abs(time1-time2); // "time3" Periodo
33 }
34
35 if(cambio==true){
36 time2=micros(); // Asignando dato de reloj a "time1"
37 Periodo=abs(time2-time1); // "time3" Periodo
38 }
39
40 f=87210/Periodo; // frecuencia
41 f_filtrada=(f*(1-alpha))+alpha*f_filtrada_k_1; // Señal filtrada frecuencia
42 f_filtrada_k_1=f_filtrada; // Señal Filtrada anterior
43 if(digitalRead(Entrada)==LOW){ // Estableciendo la salida a 0
44 f_filtrada=f_filtrada-10;
45 if(f_filtrada<0){
46 f_filtrada=0;
47 }
48 }
49 cambio = !cambio;
50 }
51
52 void requestEvent() {
53 Wire.write(f_filtrada); // Escribiendo Señal Filtrada I2C
54 }
```

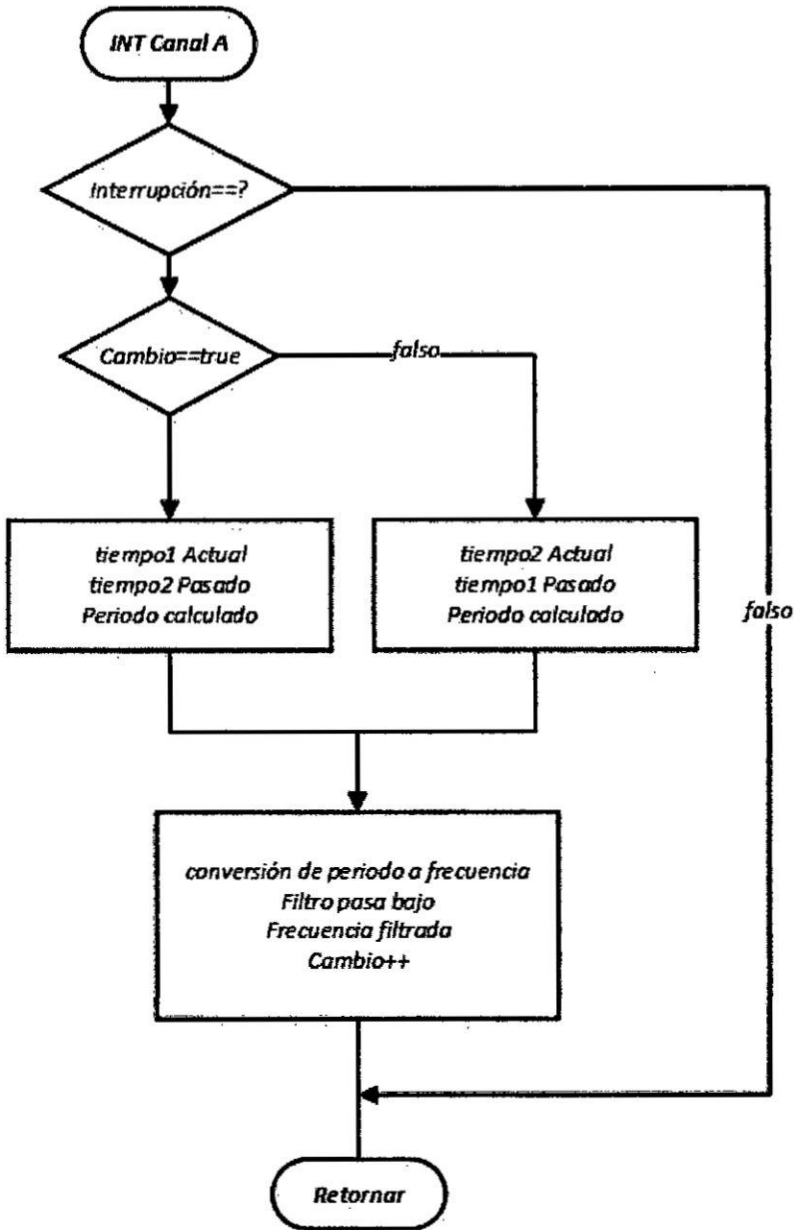
FIGURA N°116 Diagrama de flujo del programa tacómetro digital



- Inicio
- Bloque de Declaración, en este bloque incluimos librerías, definimos las variables he inicializamos todas las variables.
- Bloque de configuración interrupciones, puertos,
- Fin

De esta forma ya seremos capaces de construir la lógica del tacómetro.

FIGURA N°118 Diagrama de transformación de pulsos a velocidad



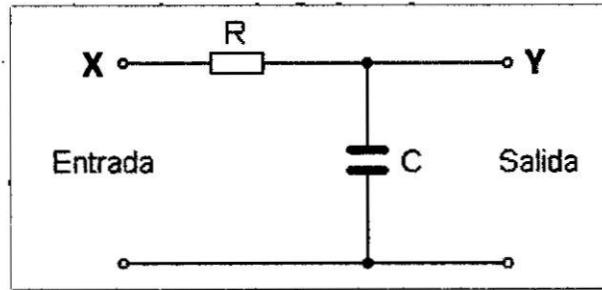
- INT Canal A, inicio de la interrupción.
- Condicional Función de interrupción.
- Condicional cambio, nos permitirá alternar los procesos.
- Bloques procesos Condicional true y false, aquí calcularemos el Periodo tras los flancos de subida.
- Bloque proceso cálculo de la velocidad, aquí determinamos la velocidad del motor, también agregamos el filtro pasa bajos.
- Fin

Filtro Digital (Tacómetro).

Para el filtro digital seleccionamos el filtro Tipo IIR, con este filtro pretendemos atenuar el ruido que nos arroja la señal adquirida por el tacómetro.

Ahora diseñaremos un filtro pasa bajos de 1er orden con el fin de purificar la señal.

FIGURA N°119 Filtro pasa bajo de 1er orden



Como se observa en la FIGURA N°119 tenemos de forma física como es in filtro de pasa bajo ahora hallaremos su ecuación matemática.

Ecuación en Laplace $Y = \left(\frac{1}{sRC+1}\right) X$

si $RC = \beta$

$$Y = \left(\frac{1}{s\beta + 1}\right) X$$

Ec. N°12

Aplicando transformada Z.

Escogiendo la aproximación rectangular. $s = \frac{z-1}{T}$

Reemplazando en la ecuación (Ec. N°12).

$$Y = \left(\frac{1}{\left(\frac{z-1}{T}\right)\beta + 1}\right) X$$

Resolviendo.

$$Y = \left(1 - \frac{T}{\beta}\right) Yz^{-1} + \left(\frac{T}{\beta}\right) X$$

$$\text{Si } \frac{T}{\beta} = \alpha$$

$$Y = (1 - \alpha)Yz^{-1} + \alpha X$$

Ec. N°13

Tiempo discreto (Ec. N°13).

$$Y_k = (1 - \alpha)Y_{k-1} + \alpha X_k$$

Ec. N°14

Como se observa el filtro es una ecuación cuyo performance depende de una única variable que es " α ", es decir el desempeño del filtro dependerá del valor a fijar en la variable α .

Inclusión del filtro en el programa arduino.

FIGURA N°120 Filtro del programa arduino

```
f=87210/Periodo;           // frecuencia
f_filtrada=(f*(1-alpha))+alpha*f_filtrada_k_1; // Señal filtrada frecuencia
f_filtrada k 1=f_filtrada; // Señal Filtrada anterior */
```

FIGURA N°120 el valor de α la hallaremos en la adquisición regulando el valor, por lo pronto ya hemos obtenido nuestro filtro listo para utilizar en el programa del tacómetro.

FIGURA N°121 Cargando el programa en la tarjeta tacómetro

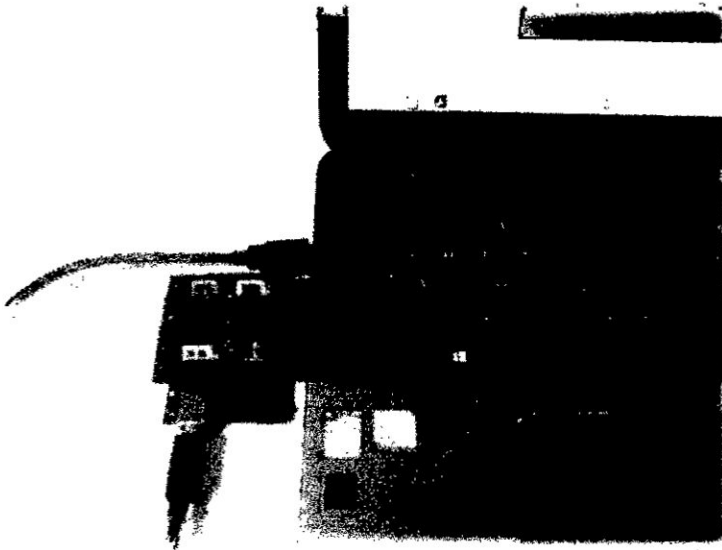
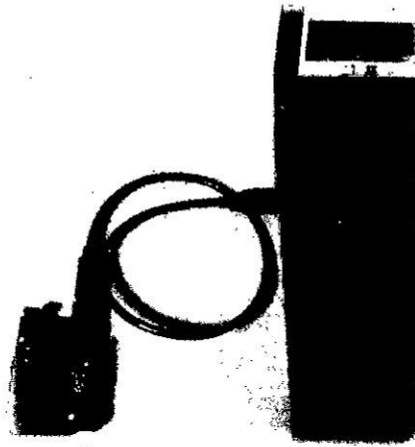


FIGURA N°122 Programa de LIFA_Base

```
LVIFA_Base : Arduino 1.0
File Edit Sketch Tools Help
LVIFA_Base
*****
**
** LVIFA_Firmware - Provides Basic Arduino Sketch For Interfaci
**
** Written By: Sam Kristoff - Applications Engineer - Nation
** Written On: November 2010
** Last Updated: May 2011
**
** This File May Be Modified And Re-Distributed Freely. Origin
** Written By Sam Kristoff And Available At www.ni.com/arduino
**
*****
// *****
**
** Includes.
**
*****
// Standard includes. These should always be included.
#include <Wire.h>
#include <SPI.h>
#include <Servo.h>
#include "LabVIEWInterface.h"
< >
```

Para cargar el sketch "LIFA_Base.ino" lo tendremos que localizar y cargar en nuestra tarjeta Arduino uno. De esta forma seremos capaces de manipular Arduino desde Labview.

FIGURA N°123 Carga de programa Lifa_Base en Arduino



Programa de Adquisicon de Datos.

FIGURA N°124 Panel de Algoritmo de adquisición de datos

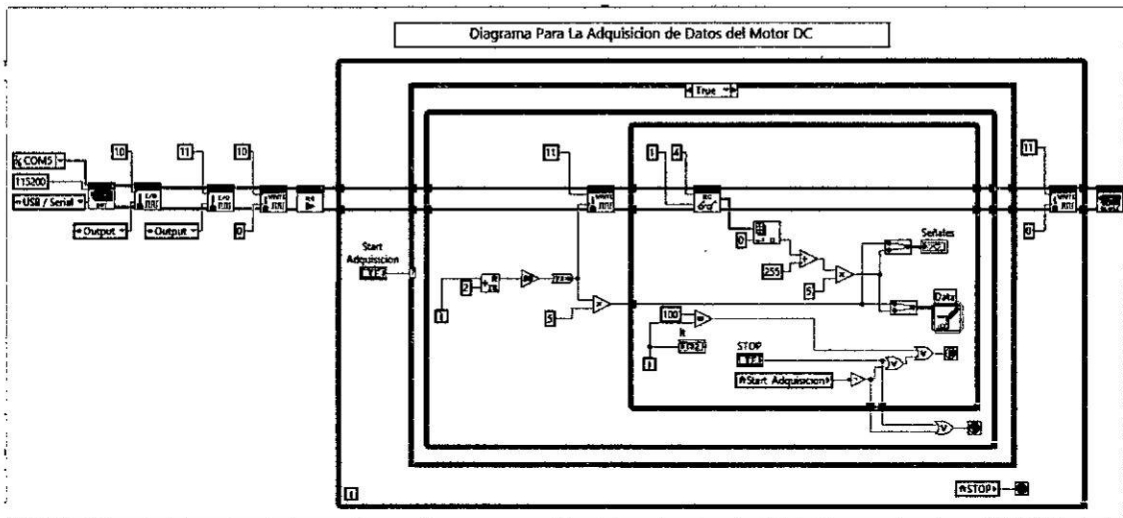


FIGURA N°125 Panel de Controles de adquisición de datos

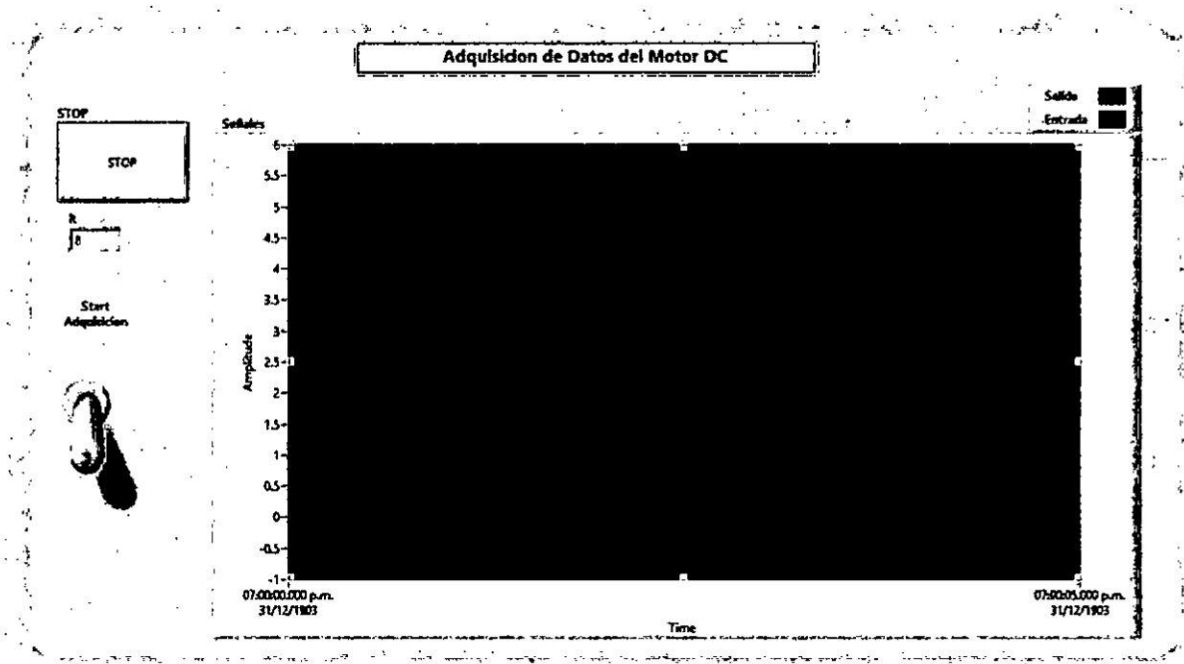
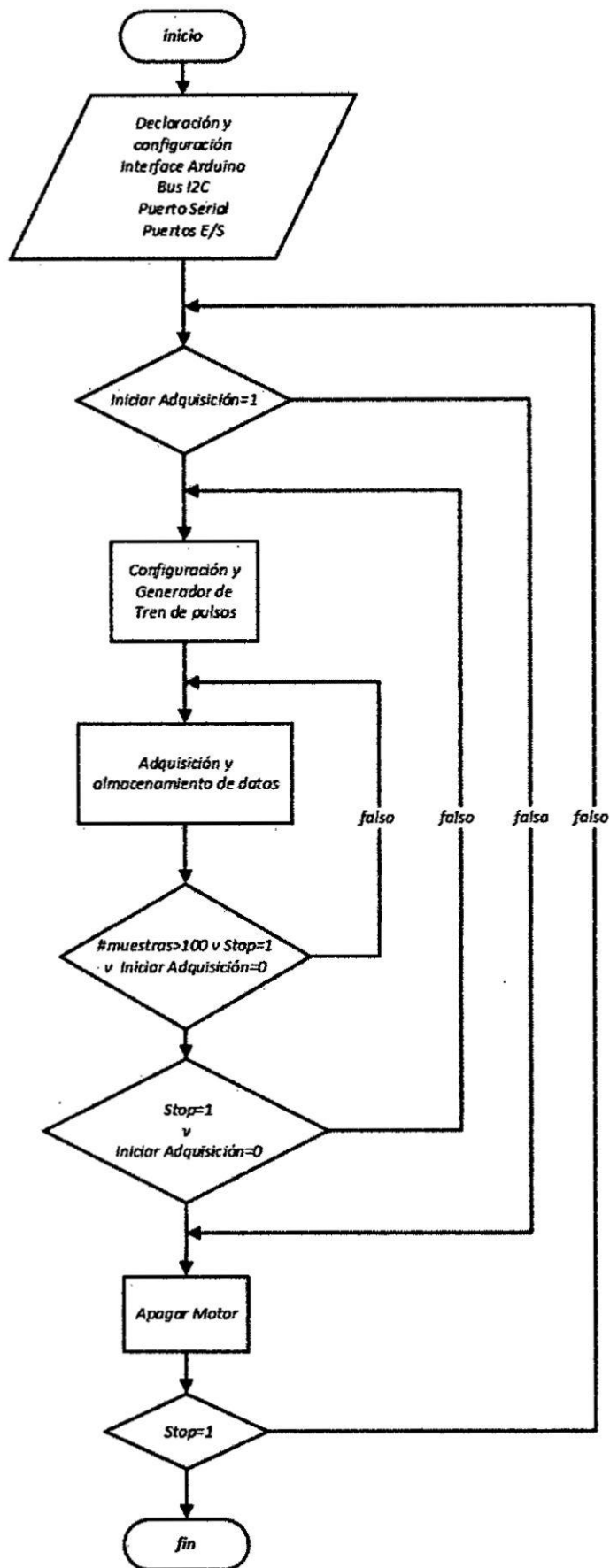


FIGURA N°124 muestra solo la parte lógica del programa de adquisición, mientras la FIGURA N°125 es donde se controla y visualizan las señales.

FIGURA N°126 Diagrama de Flujo del programa Adquisición de datos



Descripción del diagrama de flujo del programa Adquisición de datos.

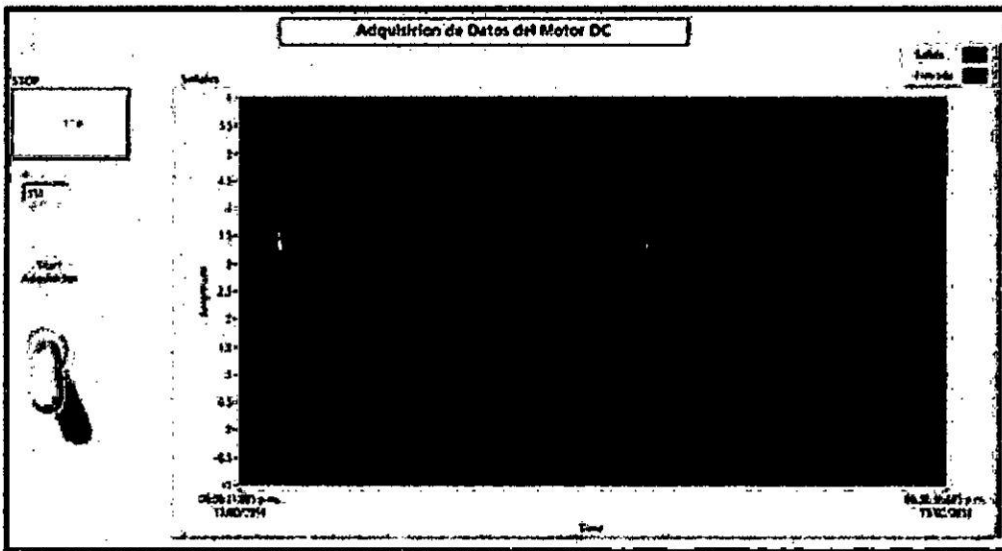
- Inicio
- Bloque de datos, incluimos librerías y declaramos las variables del programa.
- Proceso, polarizando un polo del motor.
- While de programa principal.
- Condicional Start Adquisición, aquí iniciamos la adquisición.
- While adquisición de datos.
- Proceso para la generación de pulsos (estímulos para el motor).
- While muestreo.
- Proceso de almacenamiento de data.
- Condicional FinWhile muestreo.
- Condicional Fin While de adquisición de datos.
- Proceso parada motor Fin
- Condicional Findel programa principal.

Efectuando la Adquisición de Datos.

De la *FIGURA N°127* para dar inicio a la Adquisición debemos Pulsar el Switch de Start Adquisición, este switch dará inicio a la adquisición mandando los estímulos (tren de pulsos) de manera automática, estos estímulos es la entrada del sistema SetPoint (señal color verde), la salida es la respuesta del sistema (señal de color rojo).

Para detener la adquisición debemos presionar otra vez el switch, para detener el programa solo pulsamos el botón de stop.

FIGURA N°127 Panel de Controles de adquisición de datos



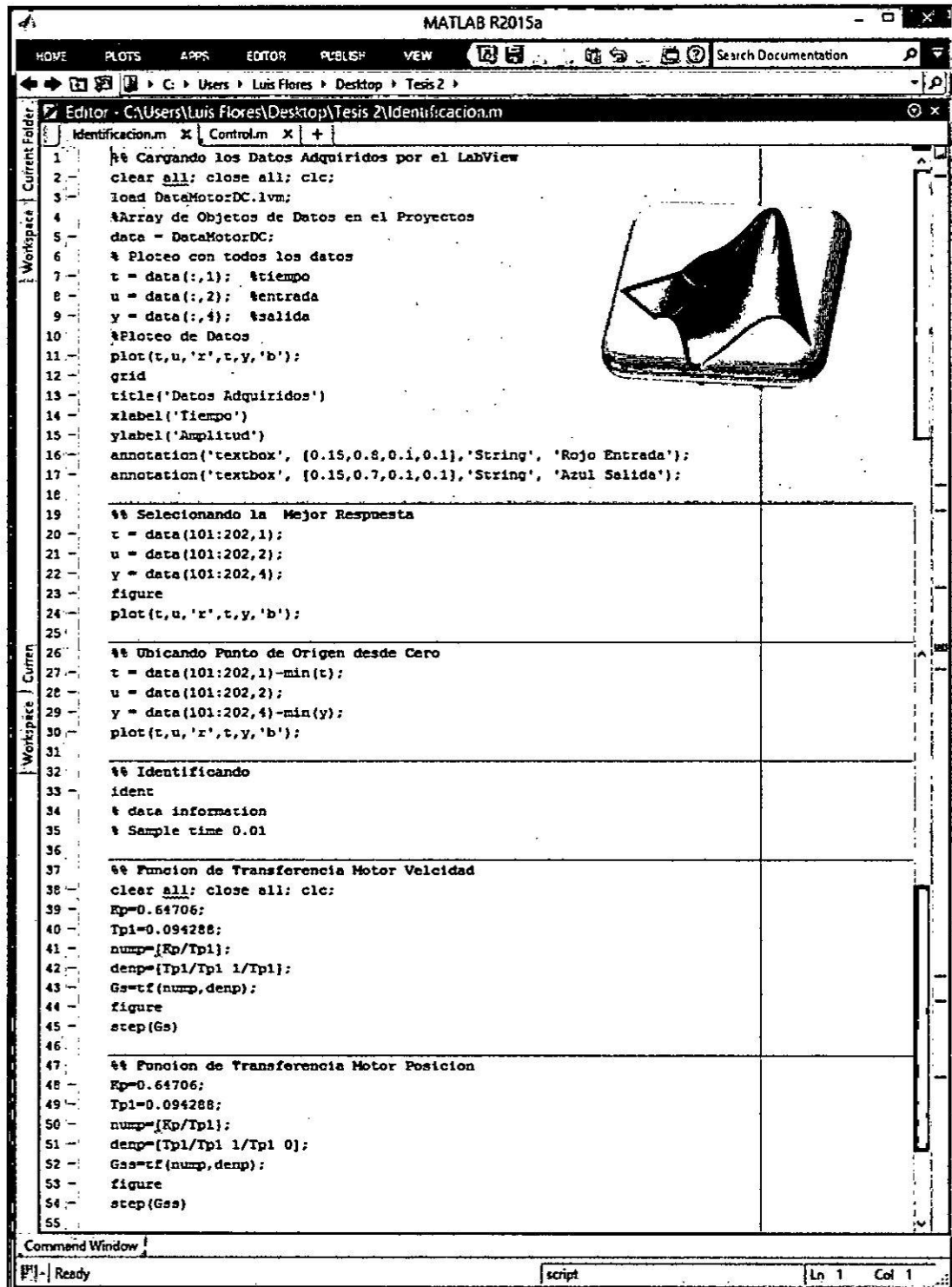
Al terminar de efectuar el programa nos entregara un archivo con los valores adquiridos (formato .lvm) en la FIGURA N°128 se muestra la un ejemplo de los registros adquiridos.

FIGURA N°128 Registros de Datos Adquiridos

DataMotorDC: Bloc de notas			
Archivo	Edición	Formato	Ver Ayuda
0.000000	0.000000	0.000000	0.000000
0.003992	0.000000	0.009116	0.000000
0.012118	0.000000	0.018135	0.000000
0.021137	0.000000	0.025139	0.000000
0.029131	0.000000	0.033145	0.000000
0.037148	0.000000	0.041142	0.000000
0.044145	0.000000	0.045145	0.000000
0.048147	0.000000	0.050195	0.000000
0.053197	0.000000	0.058282	0.000000
0.062289	0.000000	0.067344	0.000000
0.070346	0.000000	0.075348	0.000000
0.079340	0.000000	0.082428	0.000000
0.085431	0.000000	0.090575	0.000000
0.093578	0.000000	0.098580	0.000000
0.101583	0.000000	0.104627	0.000000
0.107630	0.000000	0.110644	0.000000
0.113647	0.000000	0.115648	0.000000
0.119652	0.000000	0.123654	0.000000

REALIZACIÓN DE LA IDENTIFICACIÓN

FIGURA N°129 Programa de Identification



```
1 %% Cargando los Datos Adquiridos por el LabView
2 clear all; close all; clc;
3 load DataMotorDC.lvm;
4 %Array de Objetos de Datos en el Proyectos
5 data = DataMotorDC;
6 % Ploteo con todos los datos
7 t = data(:,1); %tiempo
8 u = data(:,2); %entrada
9 y = data(:,4); %salida
10 %Ploteo de Datos
11 plot(t,u,'r',t,y,'b');
12 grid
13 title('Datos Adquiridos')
14 xlabel('Tiempo')
15 ylabel('Amplitud')
16 annotation('textbox', [0.15,0.8,0.1,0.1],'String', 'Rojo Entrada');
17 annotation('textbox', [0.15,0.7,0.1,0.1],'String', 'Azul Salida');
18
19 %% Seleccionando la Mejor Respuesta
20 t = data(101:202,1);
21 u = data(101:202,2);
22 y = data(101:202,4);
23 figure
24 plot(t,u,'r',t,y,'b');
25
26 %% Ubicando Punto de Origen desde Cero
27 t = data(101:202,1)-min(t);
28 u = data(101:202,2);
29 y = data(101:202,4)-min(y);
30 plot(t,u,'r',t,y,'b');
31
32 %% Identificando
33 ident
34 % data information
35 % Sample time 0.01
36
37 %% Funcion de Transferencia Motor Velocidad
38 clear all; close all; clc;
39 Kp=0.64706;
40 Tpl=0.094288;
41 numz=[Kp/Tpl];
42 denp=[Tpl/Tpl 1/Tpl];
43 Gs=tf(numz,denp);
44 figure
45 step(Gs)
46
47 %% Funcion de Transferencia Motor Posicion
48 Kp=0.64706;
49 Tpl=0.094288;
50 numz=[Kp/Tpl];
51 denp=[Tpl/Tpl 1/Tpl 0];
52 Gss=tf(numz,denp);
53 figure
54 step(Gss)
55
```

Descripción de Identificación.

FIGURA N°130 Segmento de carga de datos

```

%% Cargando los Datos Adquiridos por el LabView
clear all; close all; clc;
load DataMotorDC.lvm;
%Array de Objetos de Datos en el Proyecto
data = DataMotorDC;
% Ploteo con todos los datos
t = data(:,1); %tiempo
u = data(:,2); %entrada
y = data(:,4); %salida
%Ploteo de Datos
plot(t,u,'r',t,y,'b');
grid
title('Datos Adquiridos')
xlabel('Tiempo')
ylabel('Amplitud')
annotation('textbox', [0.15,0.6,0.1,0.1],'String', 'Rojo Entrada');
annotation('textbox', [0.15,0.7,0.1,0.1],'String', 'Azul Salida');
    
```

FIGURA N°131 Ploteo de Datos

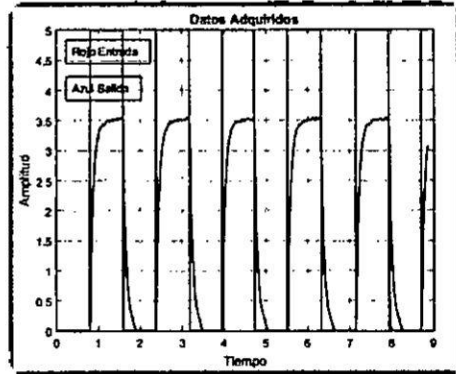


FIGURA N°130 segmento de carga de datos, nos permite importar y capturar los datos almacenados en la adquisición (formato .lvm), estos datos se guardan en el objeto data. FIGURA N°131 ploteo de Datos Adquiridos consta en dibujarlos datos.

FIGURA N°132 Segmento de captura

```

%% Seleccionando la Mejor Respuesta
t = data(101:202,1);
u = data(101:202,2);
y = data(101:202,4);
figure
plot(t,u,'r',t,y,'b');

%% Ubicando Punto de Origen desde Cero
t = data(101:202,1)-min(t);
u = data(101:202,2);
y = data(101:202,4);
figure
plot(t,u,'r',t,y,'b');
    
```

FIGURA N°133 Ploteo de Datos

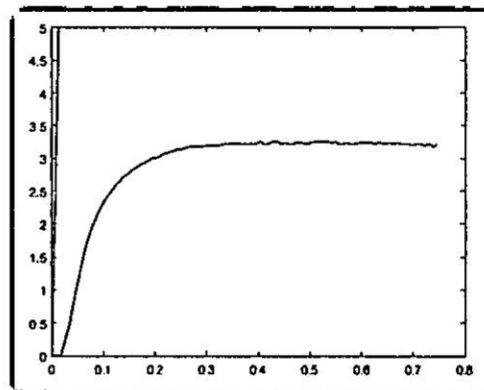


FIGURA N°132 segmento de captura de mejor respuesta y ubicación de punto inicial a cero, FIGURA N°133 nos muestra el dibujo del resultado del segmento de código.

FIGURA N°134 Comando para abrir el toolbox

```
%% Identificando
ident
% data information
% Sample time 0.01
```

FIGURA N°134 segmento comando para abrir el toolbox de Identificación como mostraremos a continuación.

FIGURA N°135 Toolbox de identificación

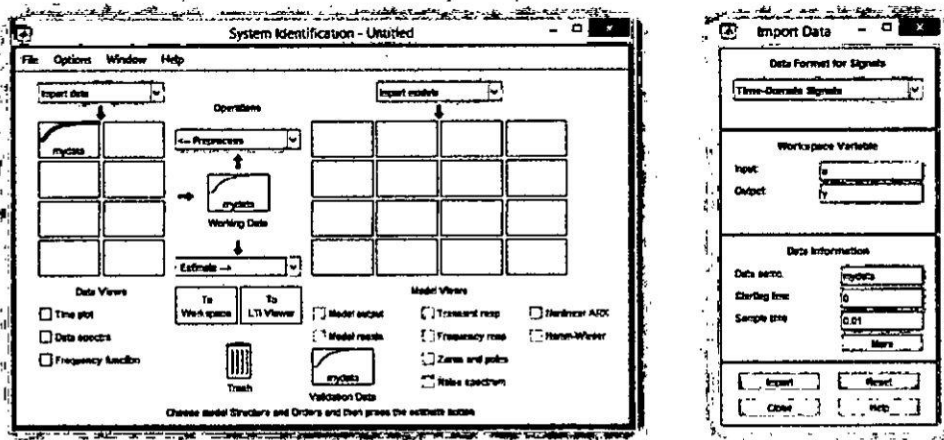


FIGURA N°135 menú principal del toolbox de identificación, aquí importamos nuestro objeto data (objeto data es la captura de los datos de Identification). Y nos permite asignar la entrada "u", salida "y", inicio "0" y tiempo de muestreo "0.01".

FIGURA N°136 Process Models

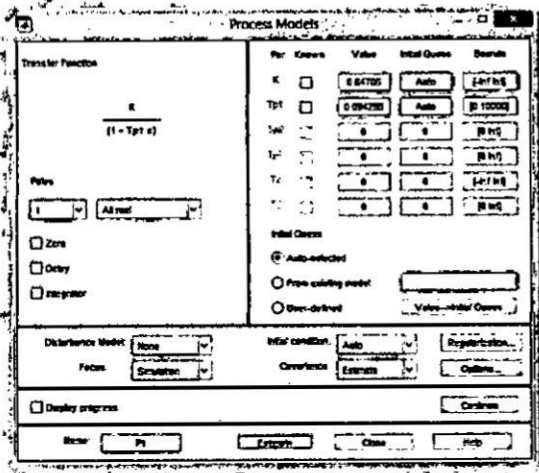


FIGURA N°137 Ploteo de resultados

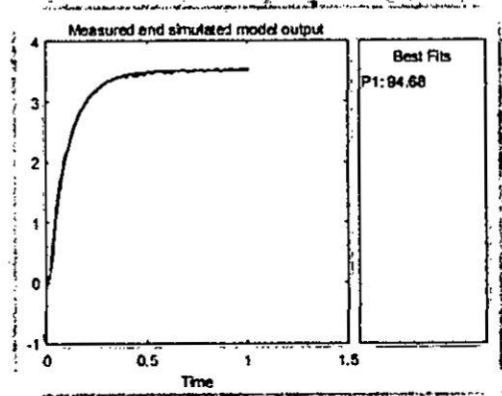


FIGURA N°136 esta ventana nos deja configurar el orden de nuestro modelo matemático entre otras opciones relacionadas, seleccionamos la opción "estimate" para que nos calcule y muestre la función de transferencia de nuestro sistema.

FIGURA N°137 nos muestra el resultado de compatibilidad de nuestro sistema, en nuestro caso la compatibilidad es de 94% es muy buena.

FIGURA N°138 Función de transferencia Motor velocidad

```

%% Funcion de Transferencia Motor Velocidad
clear all; close all; clc;
Kp=0.64706;
Tp1=0.094288;
nump=[Kp/Tp1];
denp=[Tp1/Tp1 1/Tp1];
Gs=tf(nump,denp);
figure
step(Gs)
    
```

FIGURA N°139 Ploteo de la función de transferencia

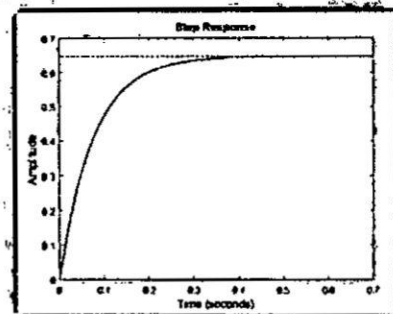
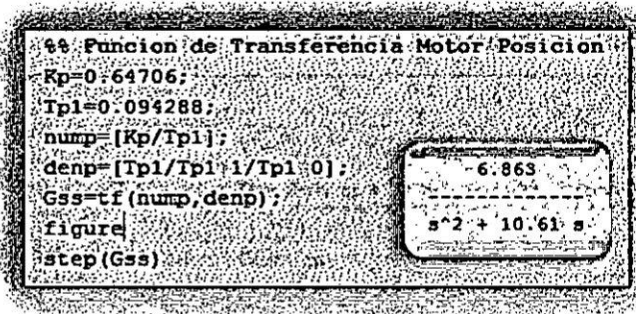


FIGURA N°138 segmento de construcción de la función de transferencia de velocidad del motor, con los parámetros hallados con el toolbox de identificación ya se puede armar la FT. de nuestro sistema.

FIGURA N°139 ploteo de la función de transferencia de velocidad del motor.

**FIGURA N°140 Función de transferencia
Motor posición**



**FIGURA N°141 Ploteo de la
función de transferencia**

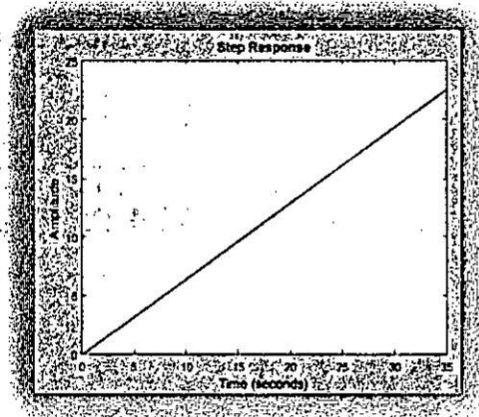


FIGURA N°140 debido a que la función de transferencia de un motor tiene por naturaleza hallarse en función a su velocidad, debimos ingresarle un término integral para lograr tener nuestra función de transferencia en función a la posición, ya que nuestra finalidad es controlar la posición de nuestro motor.

FIGURA N°141 muestra el ploteo de nuestra la función de transferencia de Posición.

FIGURA N°144 Una vez obtenido los parámetros del controlador, podemos agregarlo al sistema para construir la función de transferencia de la planta controlada controlador

FIGURA N°145 Muestra la respuesta de la planta controlada.

FIGURA N°146 Diseño del controlador

```
%% Controlador en Discreto
Ts=0.01;
Gcd=c2d(Gc,Ts,'tustin')

%% Planta y controlador Discreto
Ts=0.01;
Gfd=c2d(Gf,Ts,'tustin')
figure
step(Gfd)
```

FIGURA N°147 Función de transferencia del controlador y del sistema

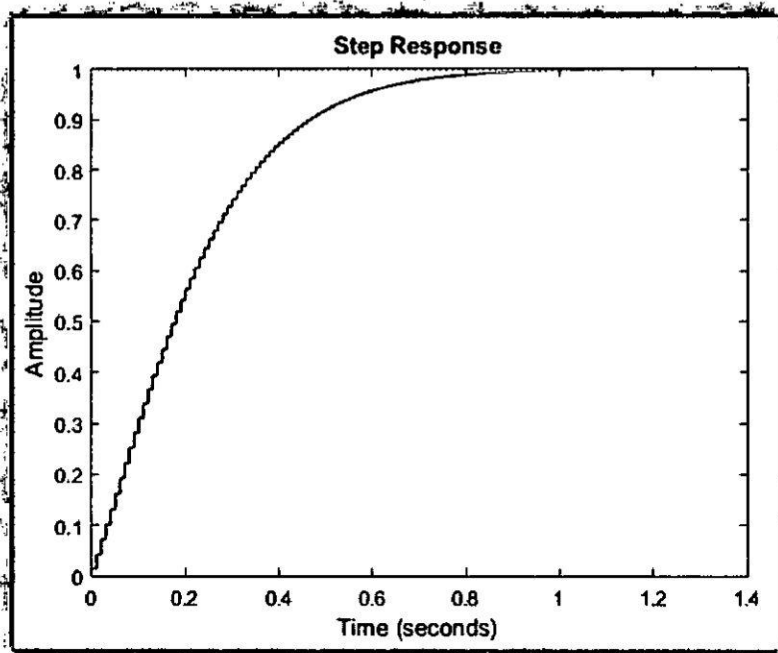
$$\frac{0.01398 z^2 + 0.002249 z - 0.01173}{z^2 - 1.87 z + 0.875}$$

$$\frac{87 z - 73}{z + 1}$$

FIGURA N°146 convertimos el controlador y el sistema (Laplace) a controlador discreto utilizando el método bilineal (tustin).

FIGURA N°147 tablas del controlador y sistema controlado en tiempo discreto.

FIGURA N°148 Respuesta del Sistema Controlado en tiempo discreto

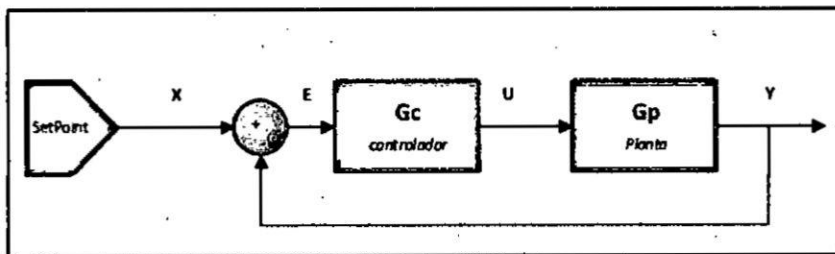


Como se observa en la FIGURA N°148 el sistema ya está totalmente controlado con un tiempo de asentamiento de 1 segundo y sin Sobreelongación.

Acondicionando el controlador PD.

Para introducir el algoritmo de control en el programa arduino debemos darle la forma, esto se hace discretizando la ecuación, para ello procederemos de la siguiente manera.

FIGURA N°149 Diagrama de bloques de un sistema controlado



Ecuación Laplace del controlador.

$$U_s = (Gc_s)E_s \tag{Ec. N°15}$$

Controlador Proporcional derivativo.

$$U_s = (sK_d + K_p)E_s \tag{Ec. N°16}$$

Transformada z

$$Z[Gc_s = sK_d + K_p] \tag{Ec. N°17}$$

Aproximación bilineal

$$s = \frac{2}{T} \left(\frac{z-1}{z+1} \right) \tag{Ec. N°18}$$

$$Gc_z = \left(\frac{2}{T} \left(\frac{z-1}{z+1} \right) \right) K_d + K_p$$

$$Gc_z = \frac{\left(\frac{2K_d + TK_p}{T} \right) z + \left(\frac{TK_p - 2K_d}{T} \right)}{z + 1}$$

$$a = \frac{2K_d + TK_p}{T} \quad y \quad b = \frac{TK_p - 2K_d}{T}$$

Representación en Transformada Z

$$Gc_z = \frac{U_z}{E_z} = \frac{az + b}{z + 1}$$

Controlador hallado de Matlab

$$\frac{87z - 73}{z + 1}$$

$$\frac{U_z}{E_z} = \frac{az + b}{z + 1}$$

Se deduce

$$a=87 \text{ y } b=-73$$

Desarrollando la formula.

$$(z + 1)U_z = (az + b)E_z$$

$$(1 + z^{-1})U_z = (a + bz^{-1})E_z$$

$$U_z = aE_z + bz^{-1}E_z - z^{-1}U_z$$

Tiempo discreto

$$U_z = aE_z + bz^{-1}E_z - z^{-1}U_z$$

$$U_k = aE_k + bE_{k-1} - U_{k-1}$$

Reemplazando los parámetro a y b del controlador.

$$U_k = 87E_k - 73E_{k-1} - U_{k-1}$$

Ec. N°19

EJECUCIÓN DEL CONTROL PROPORCIONAL DERIVATIVO (PD)

FIGURA N°150 Programa de Aplicación en LabView

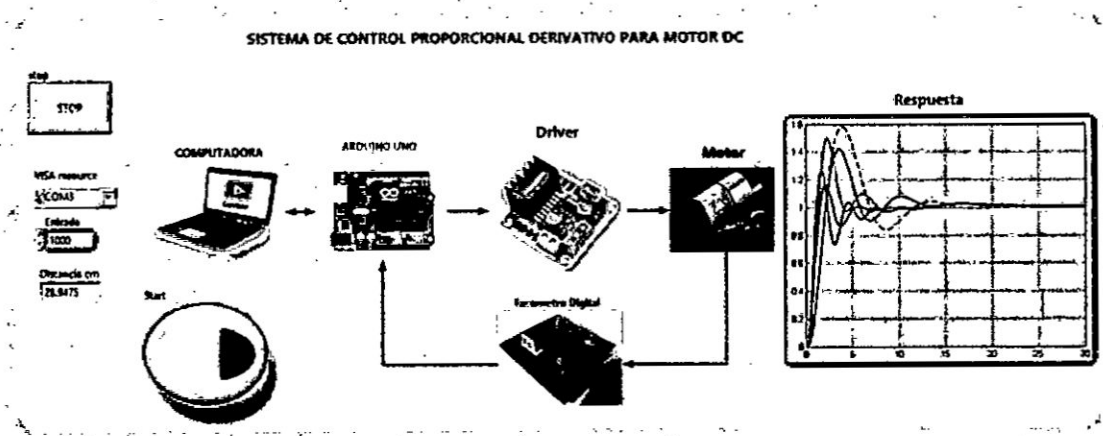


FIGURA N°151 Programa de Aplicación en LabView Diagrama lógico

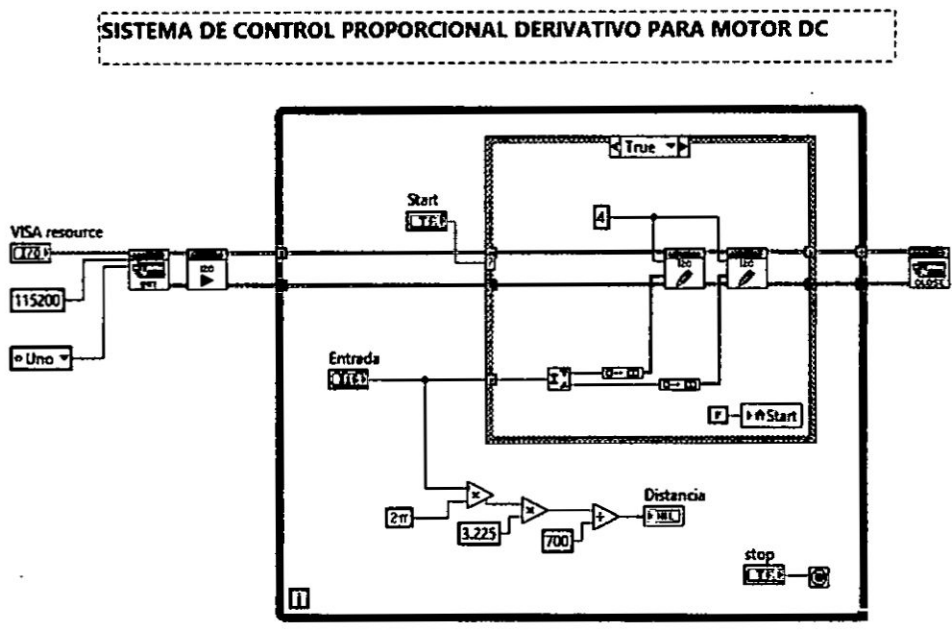


FIGURA N°150 tenemos el diagrama de control, desde aquí se comandara el SetPoint del sistema.

FIGURA N°151 Diagrama lógico de bloques, aquí es donde se arma el algoritmo de mando de SetPoint, la lógica es simple desde aquí se envía 2 bytes previamente escalado, el valor del byte es la posición que deseamos que el motor debe avanzar.

La visualización se realizó con un osciloscopio, esto debido a que si el microcontrolador enviaría la respuesta en cada instante a la aplicación de labview esto ocasionaría un retardo en el procesamiento de control.

FIGURA N°152 Señal del Setpoint y Respuesta del Sistema

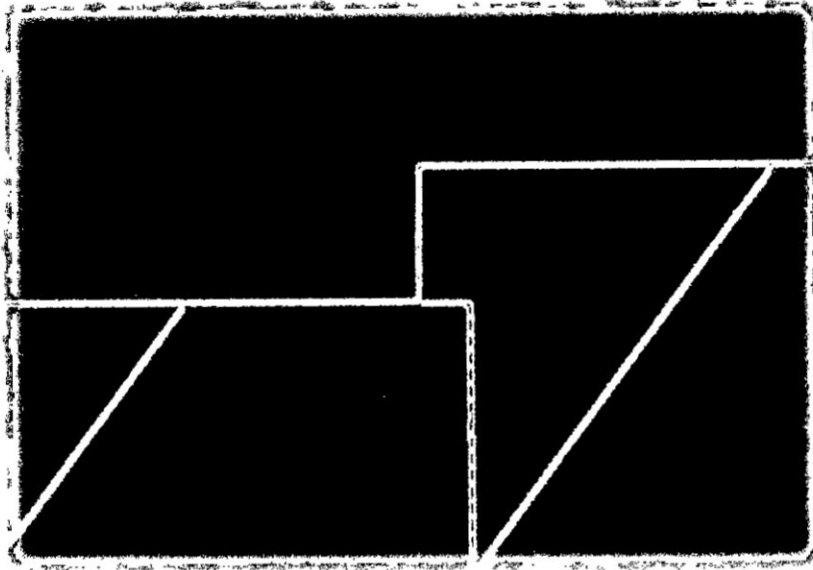
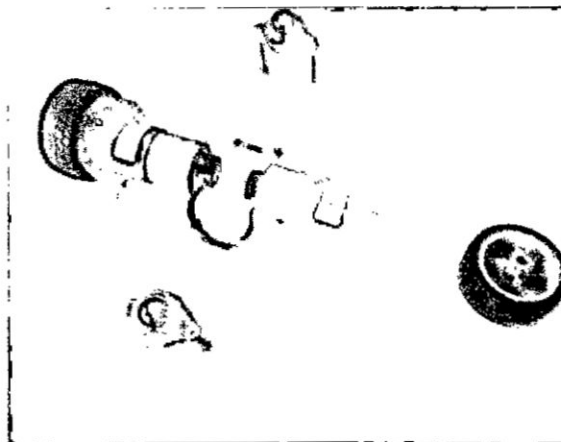


FIGURA N°152 nos muestra 2 señales, la señal de color azul es el valor del SetPoint y la señal de color rojo es la respuesta del sistema, Notamos que el control es sobreamortiguado como se diseñó en el Matlab.

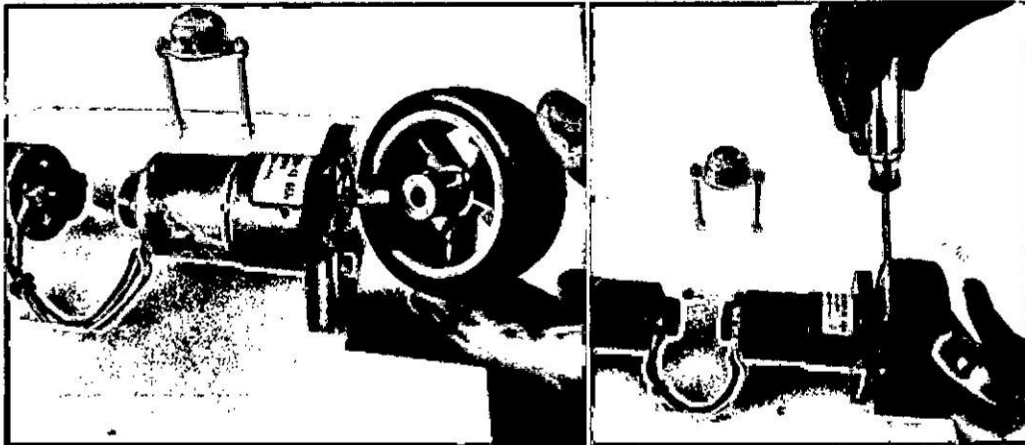
4.2.15. Construcción del robot móvil

FIGURA N°153 Inserción de los motores



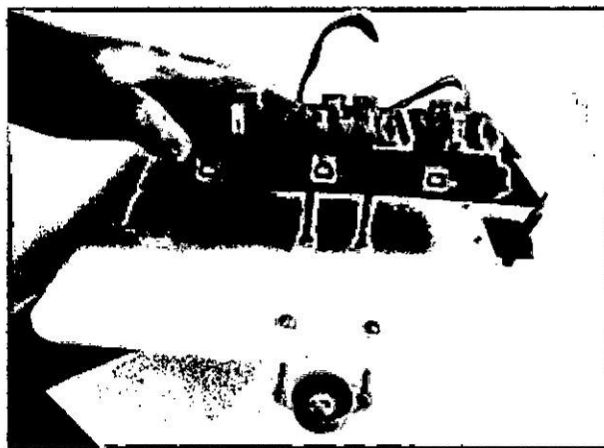
En la *FIGURA N°153* se sujetó el motor con pernos una placa de Acrílico de 2 centímetros de espesor formando un ángulo de 90° con el chasis de 4 milímetros de espesor.

FIGURA N°154 Inserción de las llantas



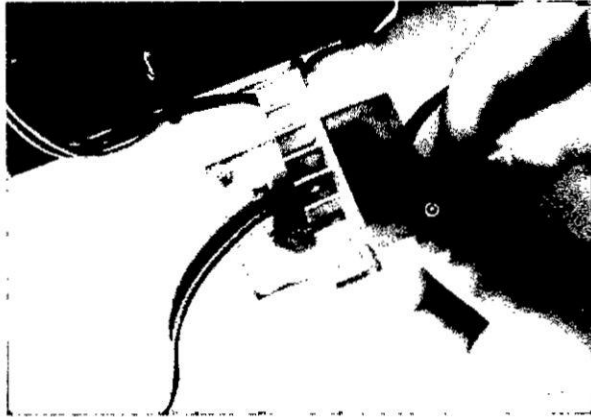
En la *FIGURA N°154* se ajustó con un perno la llanta al eje del motor.

FIGURA N°155 Inserción de la Tarjeta de control



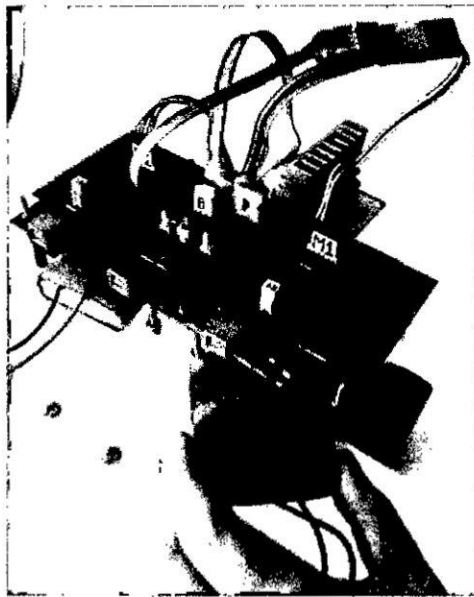
En la *FIGURA N°155* se sujetó al chasis la Tarjeta de control mediante bastones y en el eje del motor.

FIGURA N°156 Inserción de la Tarjeta de potencia



En la *FIGURA N°156* se sujetó al chasis la Tarjeta de potencia mayormente conocida como "Puente H" mediante bastones.

FIGURA N°157 Inserción de las baterías



En la *FIGURA N°157* se insertó las baterías debajo de la tarjeta de control distribuyendo su peso en el eje del motor.

FIGURA N°158 Implementación de los componentes básicos

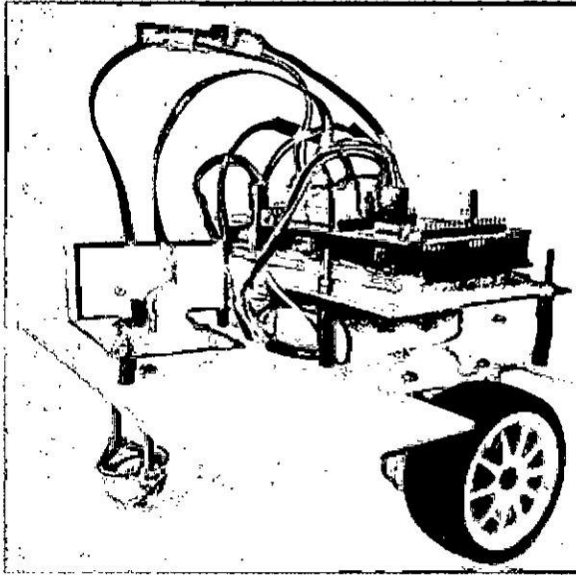
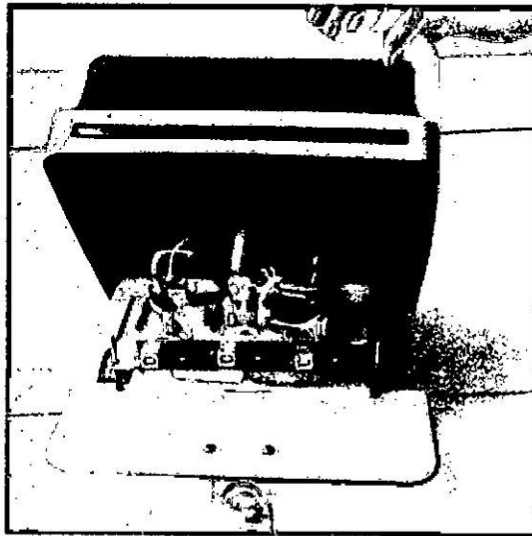
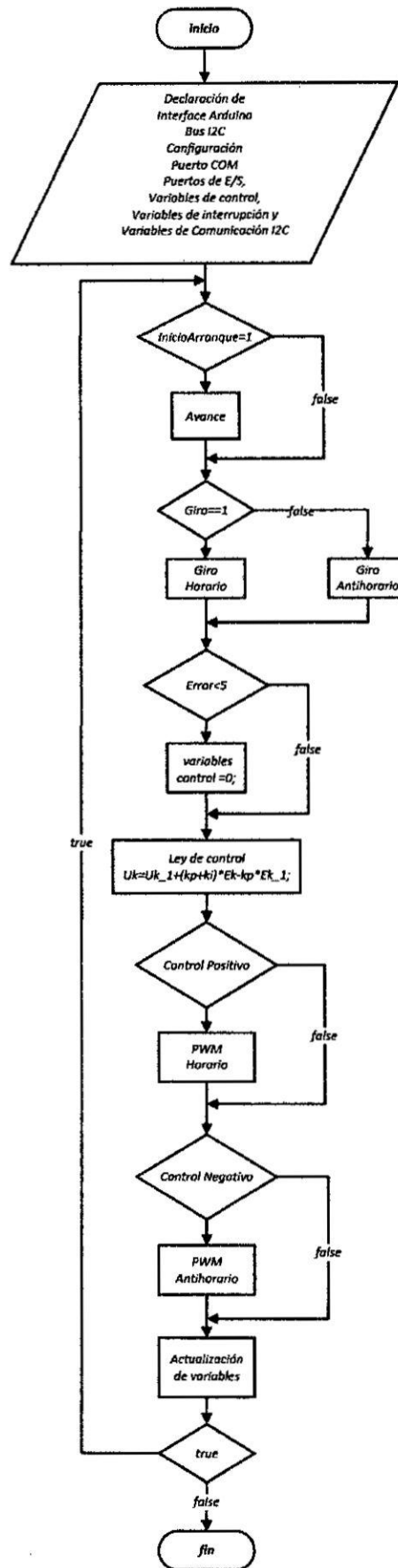


FIGURA N°159 Inserción de la carrocería



En la *FIGURA N°159* una vez que el robot móvil quede implementado se procederá a inserta la carrocería fabricada en fibra de vidrio.

FIGURA N°161 Diagrama de flujo del controlador PD



Descripción del diagrama de flujo del controlador PD.

- Inicio
- Declaración y configuración de variables, aquí se inician las variables, librerías, comunicación, entradas y salidas, etc.
- Condicional Pin Start, pin de inicio del SetPoint, condición true entrega el valor de posición del computador al SetPoint del controlador, variable $M1_Error=1$ sirve para indicar cuándo se ha acabado el control, $M1_Error=0$ inicia el indicador.
- Condicional Pin Giro, cuando es true entregamos un valor positivo al error, cuando es false entregamos el valor negativo, con esto conseguimos el sentido de giro.
- Condicional $Ek \leq 1 \ \&\& Ek \geq -1$, nos permite otorgarle a todas las variables de control los valores de inicio (valor de 0) cuando la condición es true.
- Bloque $Uk = Uk_1 + (kp + ki) * Ek - kp * Ek_1$ ley de control, controlador Proporcional Derivativo.
- Condicional $Uk \leq 255 \ \&\& Uk \geq 0$, nos permite otorgar al pwm el valor del controlador "Uk", condición true hacemos que el motor gire con sentido horario.
- Condicional $Uk \leq 0 \ \&\& Uk \geq -255$, condición true hacemos que el motor gire con sentido antihorario.
- Bloque $Uk_1 = Uk$ y $Ek_1 = Ek$, actualización de datos.
- Condición true loop
- Fin

FIGURA N°162 Carga del programa al microcontroladores



FIGURA N°162 se observa que el móvil tiene un acceso sencillo para la programación, a través de conexión USB entre el computador y el móvil.

La ventaja de tener el controlador de posición PD en cada motor, es que nos permite obtener un control en tiempo real, ya que el procesamiento se ejecuta a alta velocidad.

4.2.17. Limitaciones del Software y robot Móvil

LIMITACIONES DEL SOFTWARE LABVIEW

NI LabView es un lenguaje de programación gráfico diseñado para ingenieros y científicos para desarrollar aplicaciones de pruebas, control y medidas. La naturaleza intuitiva de la programación gráfica de LabView lo hace fácil de usar. Con LabView, los educadores e investigadores pueden usar un enfoque de diseño de sistemas gráficos para diseñar, generar prototipos y desplegar sistemas embebidos. Combina la potencia de la programación gráfica con hardware para simplificar y acelerar drásticamente el desarrollo de diseños.

LabView brinda la flexibilidad de un potente lenguaje de programación sin la complejidad de los entornos de desarrollo tradicionales [n5].

Los tipos de trayectorias que el robot puede realizar, depende de la configuración del robot y del modelamiento matemático en el software Labview.

LIMITACIONES DE LA RESOLUCIÓN ANGULAR DEL MOTOR

Para el cálculo de la resolución del motor analizaremos la rueda y al motor.

La rueda tiene un radio de 3.225cm.

El motor tiene un encoder de resolución=700CPR

Ahora calculamos el valor de la rueda en una revolución.

$$L_{rueda} = 2\pi * R \rightarrow L_{rueda} = 2\pi * 3.225$$

$$L_{rueda} = 20.26cm$$

Dónde:

L_{rueda} : Longitud de la rueda en 1 revolución.

R : Radio de la rueda.

Luego dividimos la medida de la rueda entre su resolución al eje de la caja.

$$R_{angular} = \frac{L_{rueda}}{R_{encoder}}$$

Dónde:

$R_{angular}$: Resolución angular del motor.

$R_{encoder}$: Resolución del encoder.

$$R_{angular} = \frac{20.26cm}{700} = 0.029cm$$
$$\therefore R_{angular} = 0.29mm$$



LIMITACIONES DE LA VELOCIDAD DE ADQUISICIÓN

La velocidad de adquisición de la tarjeta depende de la velocidad del protocolo serial de comunicación entre la PC y la tarjeta de adquisición, velocidad de protocolo I2C entre las tarjetas de adquisición y los algoritmos de la adquisición.

El límite teórico de nuestro método de adquisición es establecido fundamentalmente por el protocolo serial, debido a que la máxima velocidad permitida es 115200 baudios, este valor nos permite una velocidad de adquisición máxima de 69.4uSeg o 14.4KHz.

4.2.18. Análisis del funcionamiento del robot

CALCULO DEL MARGEN DE ERROR

Para el cálculo del margen de error se hicieron pruebas lineales y de rotación.

Error en pruebas lineales.

FIGURA N°163 Recorrido lineal de 0 a 1 metro

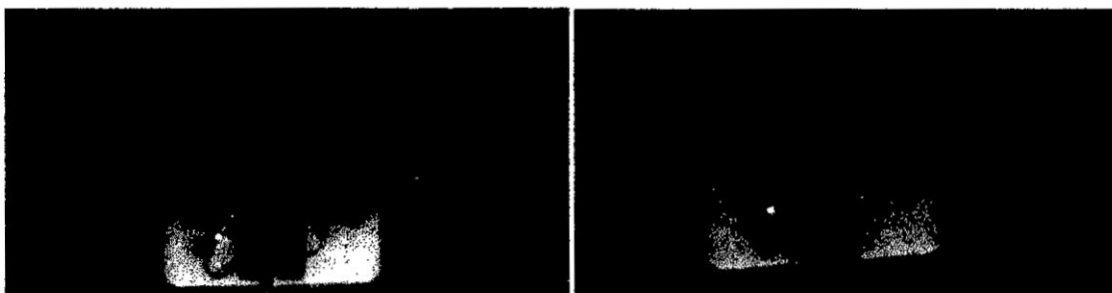
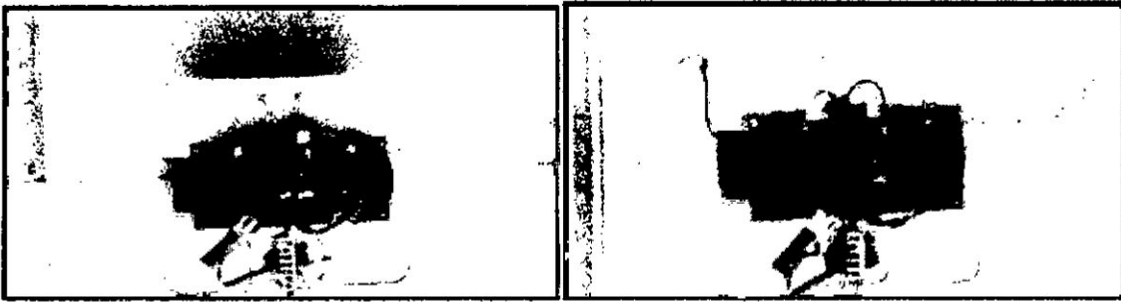


FIGURA N°164 Recorrido lineal de 0 a 2 metros



En la *FIGURA N°164* se muestran el inicio y el final respectivamente que tiene el móvil al recorrer 1 metro, como se observa el resultado nos da un error de desviación visible, este error es aleatorio debido a que se produce al inicio en el arranque, al momento de romper la inercia siempre hay un motor que arranca antes que el otro.

Este es un problema de simultaneidad entre motores al romper la inercia, esto es debido a que la masa del móvil no es uniforme, esto produce que las condiciones dinámicas para cada motor se han distintas, también hay un problema de sincronización entre los motores, pero este problema se ha intentado reducir a su mínima amplitud de forma que solo dependa de la velocidad de arranque del chip TTL (I298), la velocidad de arranque de los chips ttl son medidas en nanosegundos, casi imperceptibles debido a ello la diferencia de tiempos de alimentación del motor es casi nula, pero que de todas formas tomamos en cuenta como error.

FIGURA N°165 Recorrido lineal en reversa



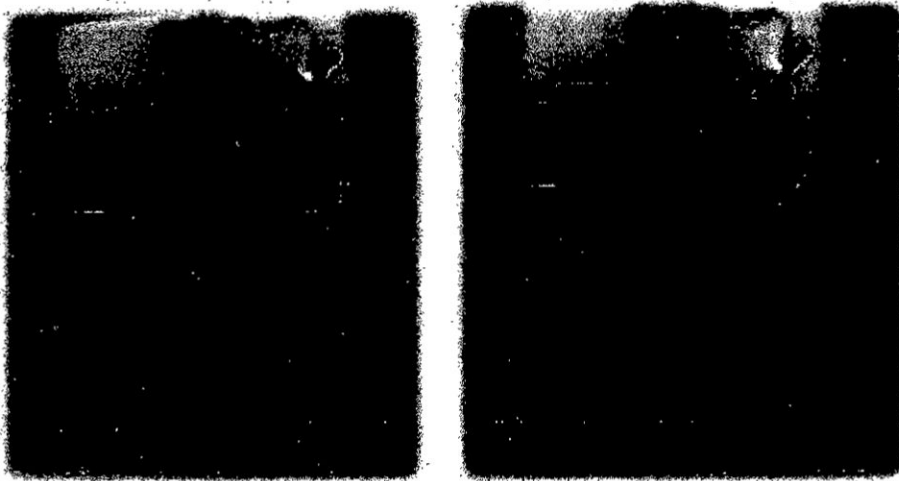
En la *FIGURA N°164* se puede notar que el error de desviación es más visible debido a que le dimos mayor distancia (2 metros), en la *FIGURA N°165*

tenemos el resultado de retroceder al punto de inicio el móvil, esto quiere decir mientras más larga sea la distancia o más puntos de coordenadas recorra la desviación también se incrementara, por lo tanto notamos que el error es acumulativo.

Error en pruebas de rotación.

Las pruebas de rotación también contienen los errores de arranque y sincronización anteriormente mencionada, pero aquí se le adiciona el error de calibración, es decir este error se produce al momento de ingresar los datos de la radio y separación de la rueda, este último dato influye mucho debido que la rotación se hace en torno a un punto centro de rotación.

FIGURA N°166 Giro de 360°



La FIGURA N°166 muestra el punto de inicio y final de la rotación respectivamente, el resultado de hacer un giro de 360 grados, este error se puede disminuir al recalibrar los datos pero siempre existirá el error matemático, error humano etc. que se añadirán al hacer pruebas.

CÁLCULO DE LA POTENCIA Y TIEMPO DE DURACIÓN DEL ROBOT MÓVIL

Para el cálculo de la potencia de nuestro móvil, en nuestro caso tenemos 2 baterías independientes, batería de control y batería de potencia.

Estas pruebas se ejecutan con el móvil en la superficie real afectándole todos los fenómenos físicos que contiene el sistema.

FIGURA N°167 Cálculo de la potencia

Cálculo de Potencia del circuito de control.



Figura de medida del Voltaje.

$$V_{\text{control}} = 11.64V$$



Figura de medida de la corriente.

$$I_{\text{control}} = 190mA$$

$$\text{Potencia de Control } W_{\text{control}} = 2211.6mW$$

Cálculo de Potencia del circuito de Potencia



Figura de medida del Voltaje.

$$V_{\text{potencia}} = 11.90V$$



Figura de medida de la corriente.

$$I_{\text{potencia}} = 660mA$$

$$\text{Potencia de Potencia } W_{\text{potencia}} = 7854mW$$

TABLA N°21 Comparación entre los circuitos de control, potencia y batería.

	Voltaje (V)	Corriente (mA)	Potencia (mW)
Ckto Control	11.64	190	2211.6
Ckto Potencia	11.9	660	7854
Batería	12	2200	26400

Para el calculo de la autonomia del robot movil tan solo debemos dividir la potencia de la batreria entre la potencia del ckto.

$$\textit{Autonomia Control} \quad \frac{W_{bateria}}{W_{control}} = \frac{26400}{2211.6} = 11.93horas$$

$$\textit{Autonomia Potencia} \quad \frac{W_{bateria}}{W_{potencia}} = \frac{26400}{7854} = 3.36horas$$

Debido a que el movil contiene baterias independiente tenemos 2 tiempos de autonomia, la autonomia se logra cuando las 2 baterias estan energizando al movil, entonces con esta razon la autonomia del movil es 3.36 horas.

4.3. Población y muestra

Para la presente investigación no es necesario tener una población y muestra.

4.4. Técnicas e instrumentos de recolección de datos

Para la presente investigación no es necesario técnicas e instrumentos de recolección de datos.

4.5. Procedimientos de recolección de datos

Para la presente investigación no es necesario un procedimiento de recolección de datos.

4.6. Procesamiento estadístico y análisis de datos

Para la presente investigación no es necesario un procedimiento estadístico y análisis de datos.

CAPÍTULO V

RESULTADOS

Prueba de distancias Lineales.

FIGURA N°168 Pruebas de distancia lineales a 1.00 metro

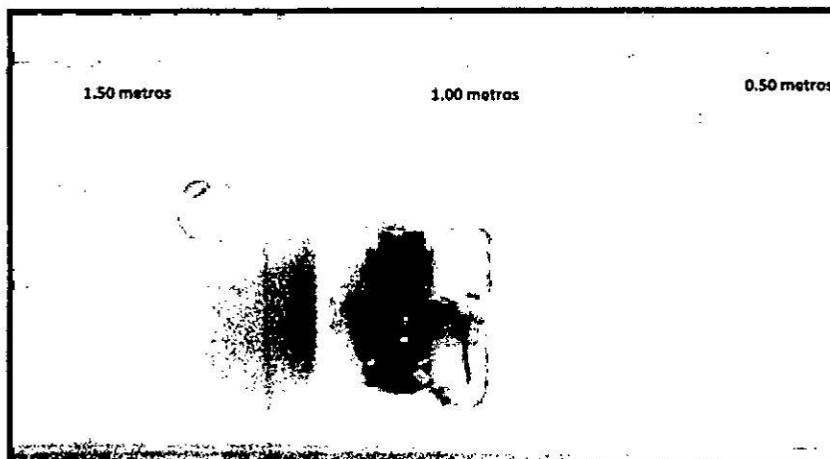


FIGURA N°169 Pruebas de distancia lineales a 2.00 metros

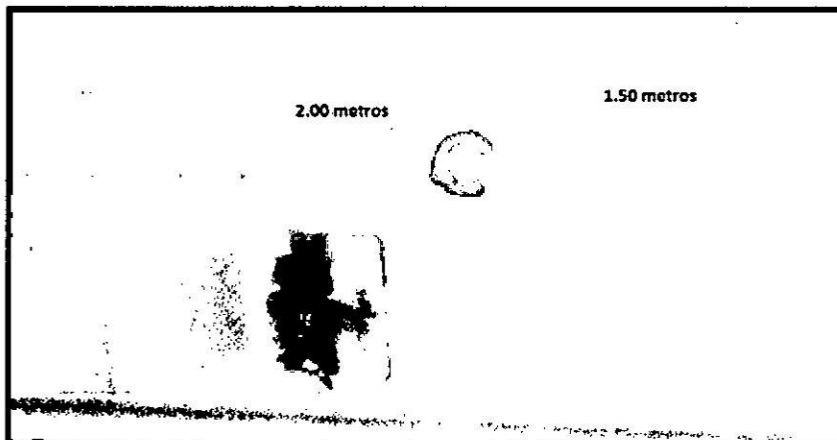
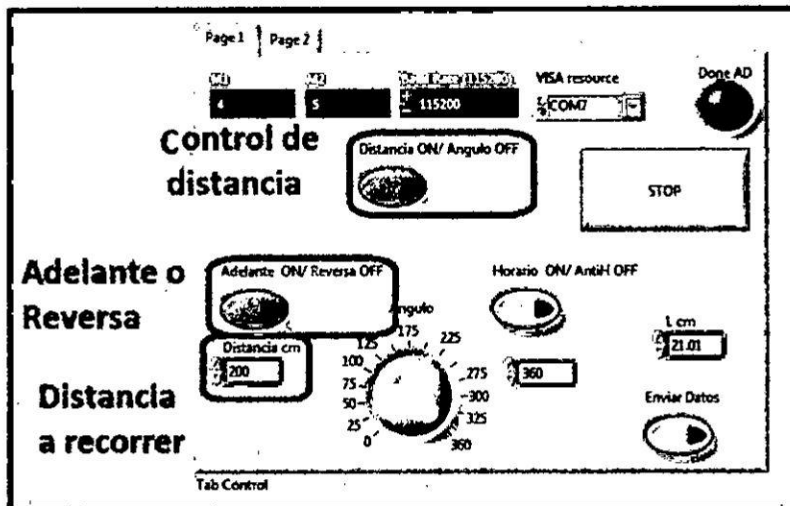


FIGURA N°170 Panel de Control para Pruebas de distancia lineales



Control de Distancia ON:

El robot móvil se desplaza en línea recta.

Control de Adelante ON:

Control se encuentra en ON: El robot móvil se desplazara hacia Adelante.

Control Distancia (cm):

Distancia a recorrer por el robot móvil.

Prueba de distancias Angulares.

FIGURA N°171 Pruebas de distancia Angular a 0 Grados

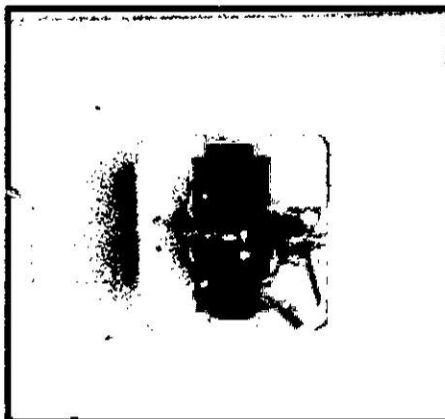
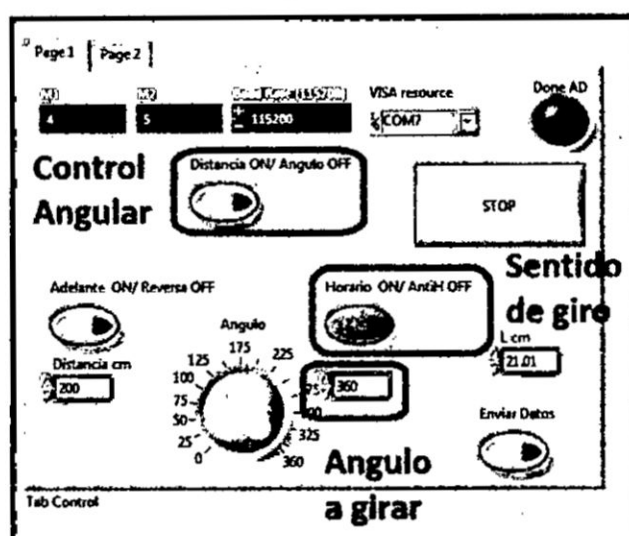


FIGURA N°172 Pruebas de distancia Angular a 90 Grados en sentido Horario



FIGURA N°173 Panel de Control para Pruebas de distancia Angulares



Control de Angulo OFF:

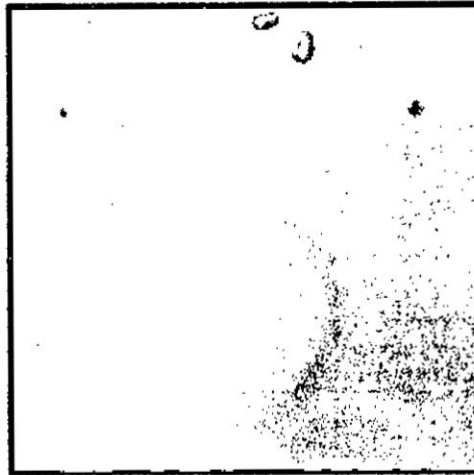
El robot móvil gira en su propio eje de rotación.

Control de Horario ON:

El robot móvil gira en su propio eje de rotación en sentido Horario.

Control Angulo (°): Con este control se ingresa el ángulo a recorrer por el robot móvil.

FIGURA N°174 Pruebas de distancia Angular



Las siguientes Pruebas nos permitirán a poner a prueba la versatilidad del robot, además nos demuestra que el robot puede hacer recorridos secuenciales y repetitivos, esenciales para la industria.

Trayectoria Cuadrada.

FIGURA N°175 Envió de datos para una trayectoria cuadrada

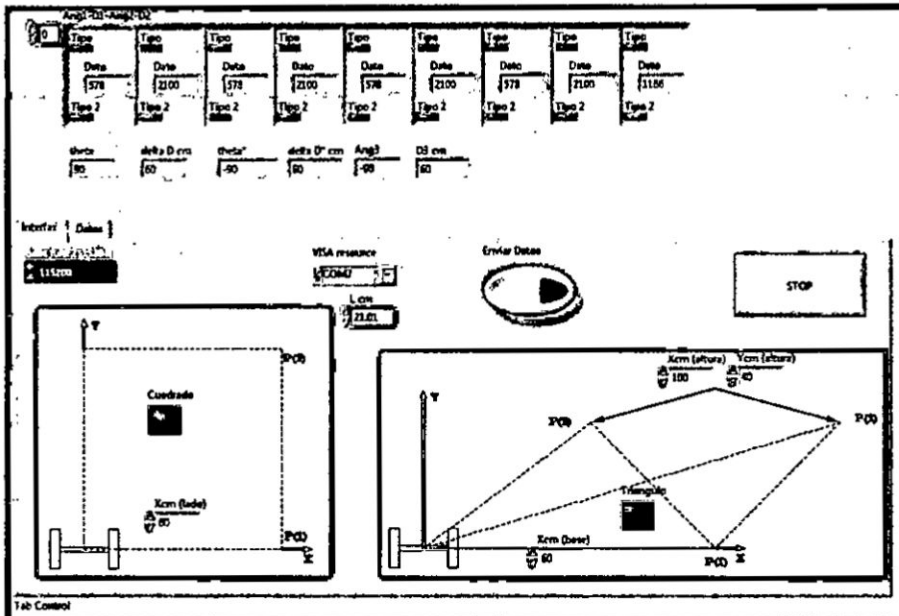


FIGURA N°176 Desplazamiento para una trayectoria cuadrada

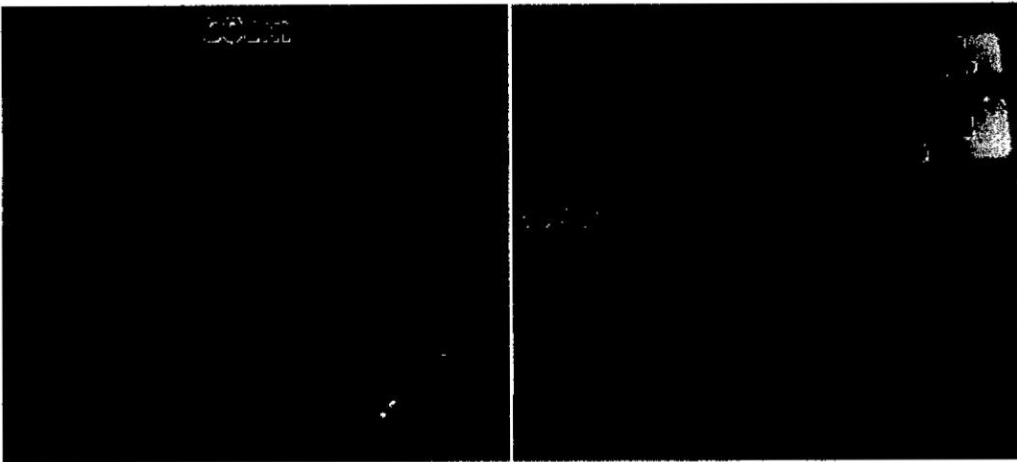
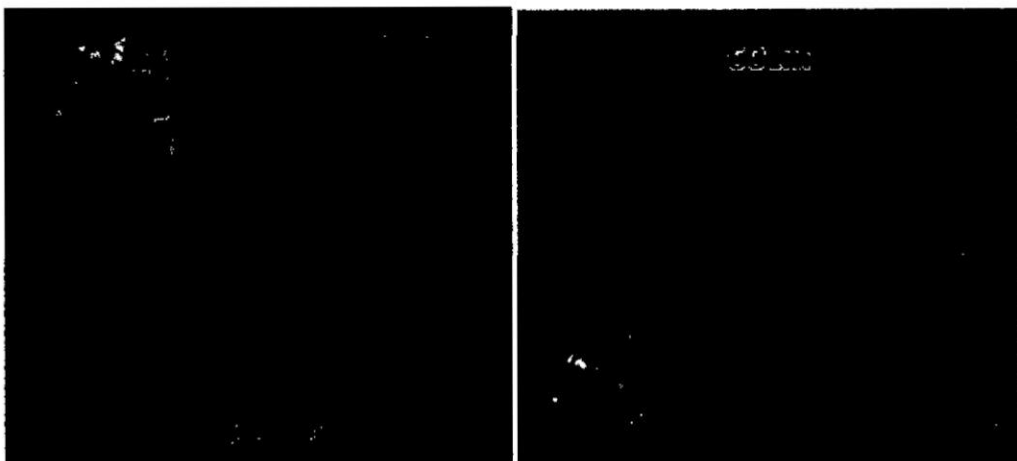


FIGURA N°177 Desplazamiento para una trayectoria cuadrada



En esta aplicación se aprecia el desplazamiento del robot para una trayectoria cuadrada, realizando los ángulos rectángulos y volviendo a su punto de inicial sin el mayor inconveniente. Utilizado el mismo concepto matemático desarrollado en este trabajo.

Trayectoria Triangular.

FIGURA N°178 Envío de datos para una trayectoria triangular

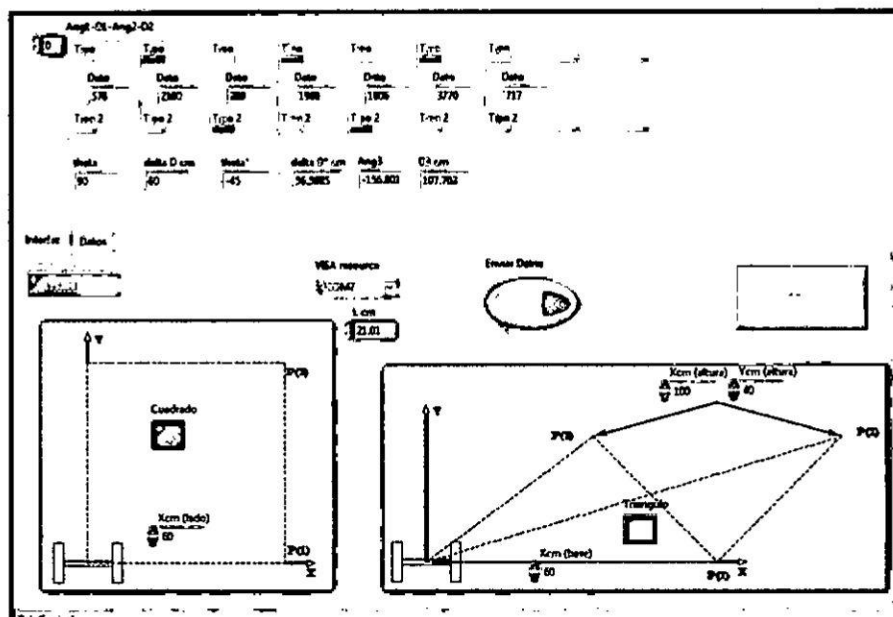


FIGURA N°179 Desplazamiento para una trayectoria triangular

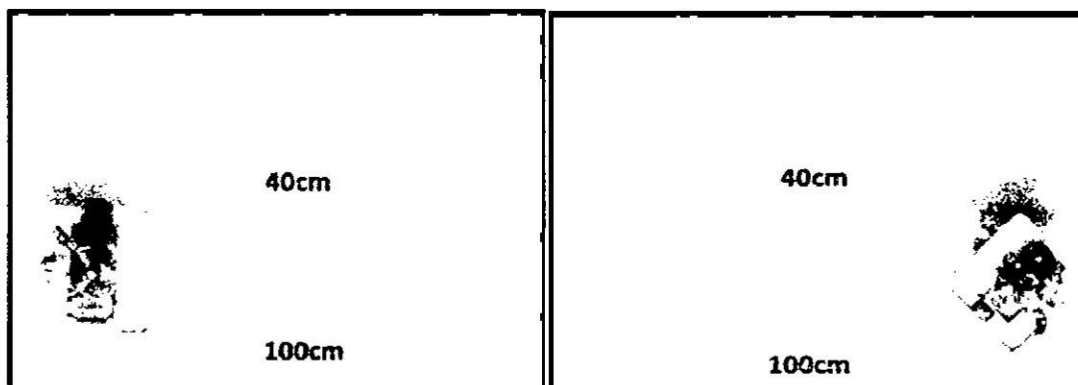
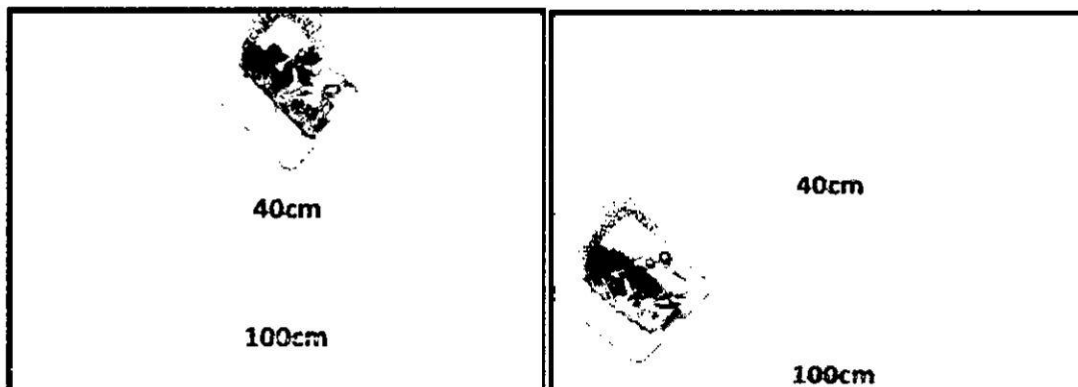


FIGURA N°180 Desplazamiento para una trayectoria triangular



En esta aplicación se aprecia el desplazamiento del robot para una trayectoria cuadrada, realizando los ángulos agudos y volviendo a su punto de inicial sin el

mayor inconveniente. Utilizado el mismo concepto matemático desarrollado en este trabajo.

FIGURA N°181 Desplazamiento para una trayectoria triangular

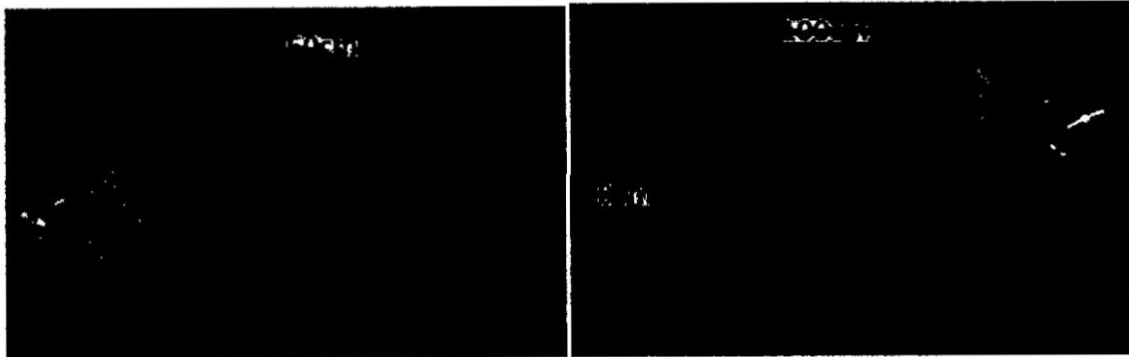


FIGURA N°182 Desplazamiento para una trayectoria triangular



En esta aplicación se aprecia el desplazamiento del robot para una trayectoria cuadrada, realizando los ángulos obtusos y volviendo a su punto de inicial sin el mayor inconveniente. Utilizado el mismo concepto matemático desarrollado en este trabajo.

CAPÍTULO VI

DISCUSIÓN DE LOS RESULTADOS

6.1. Contrastación de hipótesis con los resultados

De acuerdo con el desempeño del robot móvil y contrastado con la hipótesis general y específica, se verifica que el controlador proporcional derivativo cumple los algoritmos de control de rotación y traslación desarrollando toda la trayectoria secuencialmente de manera segura y precisa. Obteniendo una sincronización en el envío de datos del software Labview y el microcontrolador en el momento de ejecutar toda la trayectoria sin la pérdida de datos o interferencias; con un error despreciable en cada prueba realizada.

6.2. Contrastación de resultados con otros similares

El tacómetro convencional con el cual se obtiene de forma análoga la velocidad para luego procesarla en el microcontrolador y obtener la posición este método resulta muy tedioso y adiciona error. Nuestra opción de tacómetro digital nos permite en el mismo tacómetro obtener la posición de manera instantánea sin adicionar hardware.

El uso de una interfaz gráfica elaborada en LabView nos ha permitido un control eficiente con el Arduino (LIFA), esta alternativa nos permite un uso rápido, sencillo y seguro. A diferencia del uso de Matlab, visual Basic, java etc. Estos softwares necesitan desarrollar bloques que permiten vincular y sincronizar con los microcontroladores, esto hace que sea más difícil la elaboración de la interface.

CAPÍTULO VII

CONCLUSIONES

Se desarrolló el modelo cinemático directo e inverso de un robot móvil de tracción diferencial capaz de desplazarse de manera autónoma en una superficie completamente plana. Así como el uso de la ingeniería de control en el desarrollo del controlador Proporcional derivativo.

En la implementación de este proyecto de tesis se ha puesto en práctica los conceptos de diferentes áreas (electrónica, mecánica, control, instrumentación y robótica); obteniendo un robot autónomo con un error máximo de 2cm, aceptable para las aplicaciones que podría desarrollar.

Se logró evitar la adición de un tacómetro analógico, utilizando el Arduino como una tarjeta de adquisición de datos y como un tacómetro digital al mismo tiempo de manera inalámbrica, obteniendo una buena performance con el Arduino. De este modo reducimos el hardware en nuestro sistema.

A cada motor le pertenecen diferentes valores en los parámetros del tacómetro digital. Asimismo es posible hallar el modelo matemático de un motor con respecto a la posición aprovechando la función de transferencia del motor.

En la adquisición de datos la inclusión de un filtro digital de primer orden en el tacómetro digital es suficiente para la atenuación del ruido del encoder al momento adquirir datos del motor.

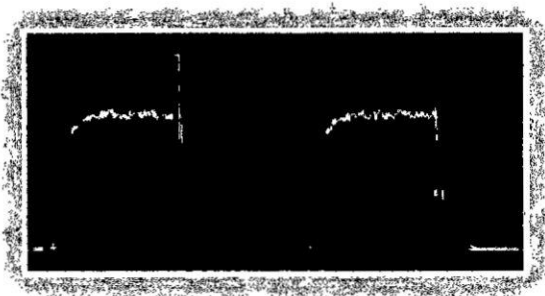


Figura señal sin filtro.

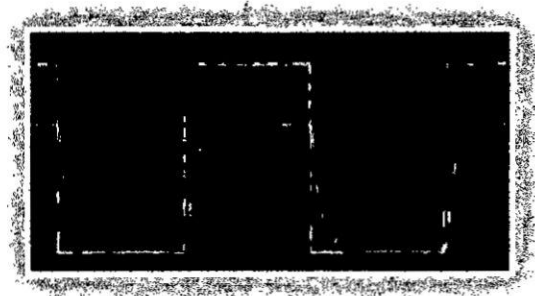


Figura señal con filtro digital.

En el Labview los bloques de Arduino consumen mucho tiempo de procesamiento debido a ello no se puede llegar a la velocidad máxima de adquisición, pero si es posible obtener una datos de adquisición admisibles.

Es posible implementar una Scada del móvil sin que el control en tiempo real se altere, esto depende de la velocidad de transmisión de datos y las líneas de código adicional.

Los bloques de Arduino en el Labview nos han facilitado la elaboración del algoritmo debido a que sus bloques contienen funciones que nos evita realizar diagramas más extensos y seguros.

CAPÍTULO VIII

RECOMENDACIONES

Al construir nuestro puente H tenemos la ventaja de adaptar nuestro hardware a cualquier diseño, ya que podemos disponer de la distribución de nuestro pines de entrada y salida, esto nos facilitara en el momento de la conexión, también podemos añadir a nuestro antojo componentes que nos puedan ayudar en nuestro diseño, en nuestro caso hemos añadido leds para visualizar la intensidad y sentido de giro del móvil.

Para el desarrollo del programa de adquisición se recomienda emigrar todo el algoritmo al lenguaje assembler que nos permitirá tener el control total de los tiempos de ejecución de cada comando.

No se recomienda usar delay en los algoritmos de control, porque debido a ello genera tiempo muerto en el microcontrolador, en lugar de ello se elabora una líneas de código pequeño con la función micros, esto nos permite que el microcontrolador permanezca siempre procesando sin dormirlo en ningún momento.

Es conveniente utilizar las mismas versiones de software entre el Arduino y el LIFA, de esta forma evitaremos problemas de compatibilidad.

Utilizar cables cortos para las conexiones de comunicación (Bluetooth), ya que de no ser así podría afectar la comunicación.

CAPÍTULO IX

REFERENCIAS BIBLIOGRÁFICAS

- ✚ [n1] Urriza Macagno, Roberto. (2000). *“Automatización Industrial Robótica y sus Aplicaciones”*.
- ✚ [n2] Ollero Baturone, Aníbal. (2001). *“Robotica: Manipuladores y robots móviles”*.
- ✚ [n3] Bermúdez Bohórquez, Giovanni. (2003). *“Modelamiento Cinemático Y Odométrico de Robots Móviles”*.
- ✚ [n4] Leonardo Enrique, Manuel Alejandro, Edgar Leonardo. (2014). *“Seguimiento de Trayectorias con un Robot Móvil de Configuración Diferencial”*.
- ✚ [n5] National Instrument. (2016). *“Por que usar Labview”*. [en línea] Disponible en : https://www.ni.com/academic/why_labview/esa/
- ✚ Arduino CC, 2016, *“Referencias estructura, variables (variables y constantes) y funciones”*. [en línea] Disponible en : <https://www.arduino.cc/en/Reference/HomePage>
- ✚ Arellano Zea, Luis Alberto. (2015). *“Diseño e implementación de un robot móvil con Control de trayectoria mediante principios odométricos”*. Perú: Universidad Nacional Mayor de San Marcos.
- ✚ Jordi Alexis Cook Meneses. (2012). *“Diseño e Implementación de un Sistema de Generación de Trayectorias para un Robot Móvil Utilizando Control Odométrico”*. Perú: Pontificia Universidad Católica del Perú.
- ✚ Katsuhiko Ogata. (2003). *“Ingeniería de control moderna”*, 4ta edición. España: Person.
- ✚ Katsuhiko Ogata. (2010). *“Sistema de control en tiempo discreto”*. 2da edición. Mexico: Person.
- ✚ Antonio Barrientos, Luis Felipe Peñin, Carlos Balaguer, Rafael Aracil. (1997). *“Fundamentos de Robótica”*, 1ra edición. España: Mc Graw Hill.
- ✚ Collazo Cuevas, J. Gorrostieta Hurtado, E. Pedraza Ortega, J. Villaseñor Carrillo, U. Romero Torrez, R y González Aguirre, M. (2009). *“Modelación de un robot móvil de dos ruedas con tracción diferencial”*. México: Instituto Tecnológico de Veracruz.

PLANTEAMIENTO DEL PROBLEMA	OBJETIVOS	HIPOTESIS	VARIABLES E INDICADORES
<p>Problema general El controlador Proporcional Derivativo permitirá el control del desplazamiento de un robot móvil de manera autónoma</p>	<p>Objetivo general El objetivo del presente trabajo es Diseñar e Implementar un robot móvil para el control del desplazamiento autónomo, utilizando un controlador Proporcional Derivativo.</p>	<p>Hipótesis General Un robot móvil es capaz de seguir la trayectoria de manera óptima y segura generada mediante un controlador Proporcional Derivativo a pesar que su implementación es la más simplista y el error es aceptable en distancias cortas.</p>	<p>Variables Independientes:</p> <ul style="list-style-type: none"> • Coordenadas para el cálculo y gráfico de la trayectoria. • Imperfecciones del terreno de trabajo. • Voltaje de las baterías. • Desalineamiento del eje de las ruedas.
	<p>Objetivo específico</p> <p>A. Diseño mecánico e implementación del robot móvil.</p> <p>B. Diseño electrónico y programación de los microcontroladores y elección del bluetooth para la comunicación inalámbrica.</p> <p>C. Diseño y sintonización del controlador PD en el microcontrolador para cada uno de los motores.</p> <p>D. Desarrollo de la interfaz gráfica en el software Labview para la visualización de la trayectoria y el envío del cálculo de distancia y ángulos de rotación al microcontrolador.</p>		<p>Variables Dependientes:</p> <ul style="list-style-type: none"> • Trayectoria generada. • Velocidad de cada rueda del robot móvil. • Aceleración del móvil. • Lectura del encoders del motor. • Posición actual de robot móvil.

Anexo N°2: Cronograma de actividades

Id	Hombre de base	Descripción	Comienzo	Fin	Nombres de los recursos
1		INDICE GENERAL	4sb 03/01/15	4sb 23/07/16	
2		CAPITULO I PLANTEAMIENTO DE LA INVESTIGACIÓN	4sb 03/01/15	4sb 21/02/15	
3		Antecedentes.	4sb 03/01/15	4sb 10/01/15	Luis Flores, Michael Garay, Computadora [1]
4		Identificación del problema.	4sb 17/01/15	4sb 17/01/15	Luis Flores, Michael Garay, Computadora [1,3]
5		Formulación del problema.	4sb 24/01/15	4sb 24/01/15	Luis Flores, Michael Garay, Computadora [1]
6		Objetivos de la investigación (general y específicos).	4sb 31/01/15	4sb 31/01/15	Luis Flores, Michael Garay, Computadora [1]
7		Justificación del proyecto de tesis.	4sb 07/02/15	4sb 07/02/15	Luis Flores, Michael Garay, Computadora [1]
8		Importancia.	4sb 14/02/15	4sb 14/02/15	Luis Flores, Michael Garay, Computadora [1]
9		Hipótesis.	4sb 21/02/15	4sb 21/02/15	Luis Flores, Michael Garay, Computadora [1]
10		CAPITULO II MARCO TEÓRICO	4sb 28/02/15	4sb 13/06/15	
11		Estudio de las ventajas de la robotización.	4sb 28/02/15	4sb 28/02/15	Michael Garay, Computadora [1]
12		Encoder de efecto hall.	4sb 14/03/15	4sb 14/03/15	Luis Flores, Computadora [1]
13		Definición Taxómetro.	4sb 21/03/15	4sb 21/03/15	Luis Flores, Computadora [1]
14		Filtros de Respuesta Infinita (IR).	4sb 28/03/15	4sb 28/03/15	Luis Flores, Computadora [1]
15		Interfaz de labview para arduino (LIFA).	4sb 04/04/15	4sb 04/04/15	Michael Garay, Computadora [1]
16		Arduino como unidad de control y comunicación.	4sb 11/04/15	4sb 11/04/15	Luis Flores, Computadora [1]
17		Toolbox de Matlab para la identificación y control.	4sb 18/04/15	4sb 18/04/15	Luis Flores, Computadora [1]
18		Tipos de locomoción.	4sb 25/04/15	4sb 25/04/15	
19		Elección del tipo de locomoción.	4sb 25/04/15	4sb 25/04/15	Michael Garay, Computadora [1]
20		Modelamiento Cinemático Directo e Inverso del Robot móvil.	4sb 02/05/15	4sb 13/06/15	
21		Elección del tipo de trayectoria.	4sb 02/05/15	4sb 02/05/15	Michael Garay, Computadora [1]
22		Cinemática directa.	4sb 16/05/15	4sb 16/05/15	Michael Garay, Computadora [1]
23		Cinemática Inversa.	4sb 30/05/15	4sb 30/05/15	Michael Garay, Computadora [1]
24		Variables de la Investigación.	4sb 13/06/15	4sb 13/06/15	Michael Garay, Computadora [1]
25		CAPITULO III DISEÑO DEL SISTEMA	4sb 27/06/15	4sb 31/10/15	
26		Diseño mecánico.	4sb 27/06/15	4sb 25/07/15	
27		Diseño del chasis.	4sb 27/06/15	4sb 04/07/15	Michael Garay, Computadora [1]
28		Diseño de carrocería.	4sb 11/07/15	4sb 18/07/15	Michael Garay, Computadora [1]
29		Elección de los motores con encoder DC y ruedas.	4sb 25/07/15	4sb 25/07/15	Michael Garay, Computadora [1]
30		Diseño de las tarjetas electrónicas	4sb 01/08/15	4sb 05/09/15	
31		Diseño de la tarjeta del puente H.	4sb 01/08/15	4sb 08/08/15	Luis Flores, Computadora [1]
32		Diseño de la tarjeta de Adquisición de datos.	4sb 15/08/15	4sb 22/08/15	Luis Flores, Computadora [1]
33		Diseño de la tarjeta de conexión de Periféricos.	4sb 29/08/15	4sb 05/09/15	Luis Flores, Computadora [1]
34		Configuración del módulo Bluetooth	4sb 12/09/15	4sb 19/09/15	
35		Configuración de la velocidad módulo Bluetooth.	4sb 12/09/15	4sb 12/09/15	Luis Flores, Computadora [1]
36		Sincronización Bluetooth y PC.	4sb 19/09/15	4sb 19/09/15	Michael Garay, Computadora [1]
37		Desarrollo de la interfaz gráfica en labview.	4sb 26/09/15	4sb 31/10/15	
38		Selección de la cantidad de datos a enviar	4sb 26/09/15	4sb 26/09/15	Michael Garay, Computadora [1]
39		Desarrollo del panel frontal para el ingreso de datos.	4sb 03/10/15	4sb 10/10/15	Michael Garay, Computadora [1]
40		Desarrollo de la Cinemática Inversa.	4sb 17/10/15	4sb 17/10/15	Michael Garay, Computadora [1]
41		Desarrollo de envío de datos al microcontrolador.	4sb 31/10/15	4sb 31/10/15	Michael Garay, Computadora [1]
42		CAPITULO IV IMPLEMENTACIÓN Y RESULTADOS EXPERIMENTALES	4sb 14/11/15	4sb 02/07/16	

Nº	Nombre de tarea	Duración	Comienzo	Fin	Nombre de los recursos
43	Adquisición, identificación y control para un motor.	10 días	sáb 14/11/15	sáb 16/01/16	Luis Flores, Computadora[1], Arduino uno Rev3[1], Motor DC +
44	Ejecución del sistema de Adquisición de Datos	2 días	sáb 14/11/15	sáb 21/11/15	Encoders [1]
45	Ejecución de la identificación.	1 día	sáb 12/12/15	sáb 12/12/15	Luis Flores, Computadora[1]
46	Elaboración del Controlador PD.	1 día	sáb 26/12/15	sáb 26/12/15	Luis Flores, Computadora[1]
47	Ejecución del Control	2 días	sáb 09/01/16	sáb 16/01/16	Luis Flores, Computadora[1]
48	Construcción del robot móvil.	4 días	sáb 23/01/16	sáb 30/02/16	Luis Flores, Michael Garay, Jantitas de goma [2], Motor DC +
49	Insertación de los motores	1 día	sáb 23/01/16	sáb 23/01/16	Encoders [1], Transporte[S/, 10.00]
50	Insertación de las tarjetas electrónicas.	1 día	sáb 30/01/16	sáb 30/01/16	Luis Flores, Michael Garay, Bluetooth[1], Oriver de Motores Rev3[2], Orros[1], Transporte[S/, 10.00]
51	Insertación de la batería.	1 día	sáb 06/02/16	sáb 06/02/16	Luis Flores, Michael Garay, Baterías 12V [2], Transporte[S/, 10.00]
52	Insertación de la carcasa.	1 día	sáb 13/02/16	sáb 13/02/16	Luis Flores, Michael Garay, Chasis del Móvil [1], Sali castor [2], Transporte[S/, 10.00]
53	Prueba de alineamiento del robot en lazo abierto.	3 días	sáb 20/02/16	sáb 05/03/16	Michael Garay, Computadora[1], Transporte[S/, 10.00], Luis Flores
54	Prueba de distancias Lineales.	1 día	sáb 20/02/16	sáb 20/02/16	Michael Garay, Computadora[1], Transporte[S/, 10.00], Luis Flores
55	Prueba de distancias Angulares.	1 día	sáb 05/03/16	sáb 05/03/16	Michael Garay, Computadora[1], Transporte[S/, 10.00], Luis Flores
56	Control de posición para el Robot Móvil.	5 días	sáb 19/03/16	sáb 16/04/16	Luis Flores,
57	Preparación de los circuitos del Robot Móvil.	1 día	sáb 19/03/16	sáb 19/03/16	Luis Flores, Computadora[1]
58	Programa del Controlador Proporcional Derivativo.	2 días	sáb 26/03/16	sáb 02/04/16	Luis Flores, Michael Garay, Computadora[1], Transporte[S/, 10.00]
59	Aplicación en Labview.	2 días	sáb 09/04/16	sáb 16/04/16	Luis Flores, Michael Garay, Computadora[1], Transporte[S/, 10.00]
60	Limitaciones del robot Móvil.	3 días	sáb 23/04/16	sáb 07/05/16	Michael Garay, Arduino uno
61	Limitaciones del software Labview.	1 día	sáb 23/04/16	sáb 23/04/16	Rev3[1], Computadora[1], Transporte[S/, 10.00]
62	Limitaciones de resolución angular del motor.	1 día	sáb 30/04/16	sáb 30/04/16	Michael Garay, Arduino uno
63	Limitación de la velocidad de adquisición.	1 día	sáb 07/05/16	sáb 07/05/16	Rev3[1], Computadora[1], Transporte[S/, 10.00]
64	Análisis del funcionamiento del robot.	8 días	sáb 14/05/16	sáb 02/07/16	Michael Garay, Luis Flores, Computadora[1], Transporte[S/, 10.00]
65	Cálculo del margen de error.	1 día	sáb 14/05/16	sáb 14/05/16	Michael Garay, Computadora[1]
66	Cálculo de la potencia y tiempo de duración del Robot Móvil.	1 día	sáb 21/05/16	sáb 21/05/16	Luis Flores, Computadora[1]
67	Resultados	2 días	sáb 28/05/16	sáb 04/06/16	Luis Flores, Michael Garay, Computadora[1]
68	Conclusiones	2 días	sáb 11/06/16	sáb 18/06/16	Luis Flores, Michael Garay, Computadora[1]
69	Recomendaciones	2 días	sáb 25/06/16	sáb 02/07/16	Luis Flores, Michael Garay, Computadora[1]
70	CAPÍTULO V CRONOGRAMA DE ACTIVIDADES	1 día	sáb 09/07/16	sáb 09/07/16	Luis Flores, Computadora[1]
71	CAPÍTULO VI PRESUPUESTO	1 día	sáb 16/07/16	sáb 16/07/16	Luis Flores, Computadora[1]
72	CAPÍTULO VII REFERENCIAS BIBLIOGRÁFICAS	1 día	sáb 23/07/16	sáb 23/07/16	Luis Flores, Michael Garay, Computadora[1]

Anexo N°4: Presupuesto de Costo por Tareas

Id	Nombre de tarea	Costo total
1	INDICE GENERAL	\$/. 3.090.00
2	CAPÍTULO I PLANTEAMIENTO DE LA INVESTIGACIÓN	\$/. 350.00
3	Antecedentes.	\$/. 50.00
4	Identificación del problema.	\$/. 50.00
5	Formulación del problema.	\$/. 50.00
6	Objetivos de la investigación (general y específicos).	\$/. 50.00
7	Justificación del proyecto de tesis.	\$/. 50.00
8	Importancia.	\$/. 50.00
9	Hipótesis.	\$/. 50.00
10	CAPÍTULO II MARCO TEÓRICO	\$/. 360.00
11	Estudio de las ventajas de la robotización.	\$/. 30.00
12	Encoder de efecto hall.	\$/. 30.00
13	Definición Tacómetro.	\$/. 30.00
14	Filtros de Respuesta infinita (RI).	\$/. 30.00
15	Interface de labview para arduino (LIFA).	\$/. 30.00
16	Arduino como unidad de control y comunicación.	\$/. 30.00
17	Toolbox de Matlab para la identificación y control.	\$/. 30.00
18	Tipos de locomoción.	\$/. 30.00
19	Elección del tipo de locomoción.	\$/. 30.00
20	Modelamiento Cinemático Directo e Inverso del Robot móvil.	\$/. 120.00
21	Elección del tipo de trayectoria.	\$/. 30.00
22	Cinemática directa.	\$/. 30.00
23	Cinemática Inversa.	\$/. 30.00
24	Variables de la Investigación.	\$/. 30.00
25	CAPÍTULO III DISEÑO DEL SISTEMA	\$/. 360.00
26	Diseño mecánico.	\$/. 90.00
27	Diseño del chasis.	\$/. 30.00
28	Diseño de carrocería.	\$/. 30.00
29	Elección de los motores con encoder DC y ruedas.	\$/. 30.00
30	Diseño de las tarjetas electrónicas	\$/. 90.00
31	Diseño de la tarjeta del puente H.	\$/. 30.00
32	Diseño de la tarjeta de Adquisición de datos.	\$/. 30.00
33	Diseño de la tarjeta de conexión de Periféricos.	\$/. 30.00
34	Configuración del módulo bluetooth	\$/. 60.00
35	Configuración de la velocidad módulo bluetooth.	\$/. 30.00
36	Sincronización Bluetooth y PC.	\$/. 30.00
37	Desarrollo de la interfaz gráfica en labview.	\$/. 120.00
38	Selección de la cantidad de datos a enviar	\$/. 30.00
39	Desarrollo del panel frontal para el ingreso de datos.	\$/. 30.00
40	Desarrollo de la Cinemática Inversa.	\$/. 30.00
41	Desarrollo de envío de datos al microcontrolador.	\$/. 30.00
42	CAPÍTULO IV IMPLEMENTACIÓN Y RESULTADOS EXPERIMENTALES	\$/. 1.910.00
43	Adquisición, identificación y Control para un motor.	\$/. 841.00
44	Ejecución del sistema de Adquisición de Datos	\$/. 251.00
45	Ejecución de la identificación.	\$/. 30.00
46	Elaboración del Controlador PD.	\$/. 30.00
47	Ejecución del Control.	\$/. 30.00
48	Construcción del robot móvil.	\$/. 819.00
49	inserción de los motores	\$/. 219.00
50	inserción de las tarjetas electrónicas.	\$/. 400.00
51	inserción de la batería.	\$/. 80.00
52	inserción de la carrocería.	\$/. 120.00
53	Prueba de alineamiento del robot en lazo abierto.	\$/. 120.00
54	Prueba de distancias Lineales.	\$/. 60.00
55	Prueba de distancias Angulares.	\$/. 60.00
56	Control de posición para el Robot Móvil.	\$/. 110.00
57	Preparación de los circuitos del Robot Móvil.	\$/. 20.00
58	Programa del Controlador Proporcional Derivativo.	\$/. 30.00
59	Aplicación en Labview.	\$/. 60.00
60	Limitaciones del robot Móvil.	\$/. 300.00
61	Limitaciones del software Labview.	\$/. 120.00
62	Limitaciones de resolución angular del motor.	\$/. 120.00
63	Limitación de la velocidad de adquisición.	\$/. 60.00
64	Análisis del funcionamiento del robot.	\$/. 220.00
65	Cálculo del margen de error.	\$/. 40.00
66	Cálculo de la potencia y tiempo de duración del Robot Móvil.	\$/. 30.00
67	Resultados	\$/. 50.00
68	Conclusiones	\$/. 50.00
69	Recomendaciones	\$/. 50.00
70	CAPÍTULO V CRONOGRAMA DE ACTIVIDADES	\$/. 30.00
71	CAPÍTULO VI PRESUPUESTO	\$/. 30.00
72	CAPÍTULO VII REFERENCIAS BIBLIOGRÁFICAS	\$/. 50.00

Anexo N°4: Presupuesto de Recursos por Tareas

Id	Nombre de tarea	Costo total
1	INDICE GENERAL	\$/. 3,090.00
2	CAPÍTULO I PLANTEAMIENTO DE LA INVESTIGACIÓN	\$/. 350.00
3	Antecedentes.	\$/. 50.00
	Luis Flores	\$/. 20.00
	Michael Garay	\$/. 20.00
	Computadora	\$/. 10.00
4	Identificación del problema.	\$/. 50.00
	Luis Flores	\$/. 20.00
	Michael Garay	\$/. 20.00
	Computadora	\$/. 10.00
5	Formulación del problema.	\$/. 50.00
	Luis Flores	\$/. 20.00
	Michael Garay	\$/. 20.00
	Computadora	\$/. 10.00
6	Objetivos de la Investigación (general y específicos).	\$/. 50.00
	Luis Flores	\$/. 20.00
	Michael Garay	\$/. 20.00
	Computadora	\$/. 10.00
7	Justificación del proyecto de tesis.	\$/. 50.00
	Luis Flores	\$/. 20.00
	Michael Garay	\$/. 20.00
	Computadora	\$/. 10.00
8	Importancia.	\$/. 50.00
	Luis Flores	\$/. 20.00
	Michael Garay	\$/. 20.00
	Computadora	\$/. 10.00
9	Hipótesis.	\$/. 50.00
	Luis Flores	\$/. 20.00
	Michael Garay	\$/. 20.00
	Computadora	\$/. 10.00
10	CAPÍTULO II MARCO TEÓRICO	\$/. 360.00
11	Estudio de las ventajas de la robotización.	\$/. 30.00
	Michael Garay	\$/. 20.00
	Computadora	\$/. 10.00
12	Encoder de efecto hall.	\$/. 30.00
	Luis Flores	\$/. 20.00
	Computadora	\$/. 10.00
13	Definición Tacómetro.	\$/. 30.00
	Luis Flores	\$/. 20.00
	Computadora	\$/. 10.00
14	Filtros de Respuesta Infinita (IR).	\$/. 30.00
	Luis Flores	\$/. 20.00
	Computadora	\$/. 10.00
15	Interface de labview para arduino (LIFA).	\$/. 30.00
	Michael Garay	\$/. 20.00
	Computadora	\$/. 10.00
16	Arduino como unidad de control y comunicación.	\$/. 30.00
	Luis Flores	\$/. 20.00
	Computadora	\$/. 10.00
17	Toolbox de Matlab para la identificación y control.	\$/. 30.00
	Luis Flores	\$/. 20.00
	Computadora	\$/. 10.00
18	Tipos de locomoción.	\$/. 30.00
19	Elección del tipo de locomoción.	\$/. 30.00
	Michael Garay	\$/. 20.00
	Computadora	\$/. 10.00
20	Modelamiento Cinemático Directo e Inverso del Robot móvil.	\$/. 120.00
21	Elección del tipo de trayectoria.	\$/. 30.00
	Michael Garay	\$/. 20.00
	Computadora	\$/. 10.00
22	Cinemática directa.	\$/. 30.00
	Michael Garay	\$/. 20.00
	Computadora	\$/. 10.00
23	Cinemática Inversa.	\$/. 30.00
	Michael Garay	\$/. 20.00
	Computadora	\$/. 10.00
24	Variables de la Investigación.	\$/. 30.00
	Michael Garay	\$/. 20.00
	Computadora	\$/. 10.00
25	CAPÍTULO III DISEÑO DEL SISTEMA	\$/. 360.00
26	Diseño mecánico.	\$/. 90.00
27	Diseño del chasis.	\$/. 30.00
	Michael Garay	\$/. 20.00
	Computadora	\$/. 10.00
28	Diseño de carrocería.	\$/. 30.00
	Michael Garay	\$/. 20.00
	Computadora	\$/. 10.00
29	Elección de los motores con encoder DC y ruedas.	\$/. 30.00
	Michael Garay	\$/. 20.00
	Computadora	\$/. 10.00

Id	Nombre de tarea	Costo total
30	Diseño de las tarjetas electrónicas	S/. 90.00
31	Diseño de la tarjeta del puente H.	S/. 30.00
	<i>Luis Flores</i>	S/. 20.00
	<i>Computadora</i>	S/. 10.00
32	Diseño de la tarjeta de Adquisición de datos.	S/. 30.00
	<i>Luis Flores</i>	S/. 20.00
	<i>Computadora</i>	S/. 10.00
33	Diseño de la tarjeta de conexión de Periféricos.	S/. 30.00
	<i>Luis Flores</i>	S/. 20.00
	<i>Computadora</i>	S/. 10.00
34	Configuración del módulo bluetooth	S/. 60.00
35	Configuración de la velocidad módulo bluetooth.	S/. 30.00
	<i>Luis Flores</i>	S/. 20.00
	<i>Computadora</i>	S/. 10.00
36	Sincronización Bluetooth y PC.	S/. 30.00
	<i>Michael Garay</i>	S/. 20.00
	<i>Computadora</i>	S/. 10.00
37	Desarrollo de la Interfaz gráfica en labview.	S/. 120.00
38	Selección de la cantidad de datos a enviar	S/. 30.00
	<i>Michael Garay</i>	S/. 20.00
	<i>Computadora</i>	S/. 10.00
39	Desarrollo del panel frontal para el Ingreso de datos.	S/. 30.00
	<i>Michael Garay</i>	S/. 20.00
	<i>Computadora</i>	S/. 10.00
40	Desarrollo de la Cinemática Inversa.	S/. 30.00
	<i>Michael Garay</i>	S/. 20.00
	<i>Computadora</i>	S/. 10.00
41	Desarrollo de envío de datos al microcontrolador.	S/. 30.00
	<i>Michael Garay</i>	S/. 20.00
	<i>Computadora</i>	S/. 10.00
42	CAPÍTULO IV IMPLEMENTACIÓN Y RESULTADOS EXPERIMENTALES	S/. 1,910.00
43	Adquisición, Identificación y Control para un motor.	S/. 341.00
44	Ejecución del sistema de Adquisición de Datos	S/. 251.00
	<i>Luis Flores</i>	S/. 20.00
	<i>Computadora</i>	S/. 10.00
	<i>Arduino uno Rev3</i>	S/. 80.00
	<i>Motor DC + Encoders</i>	S/. 141.00
45	Ejecución de la Identificación.	S/. 30.00
	<i>Luis Flores</i>	S/. 20.00
	<i>Computadora</i>	S/. 10.00
46	Elaboración del Controlador PD.	S/. 30.00
	<i>Luis Flores</i>	S/. 20.00
	<i>Computadora</i>	S/. 10.00
47	Ejecución del Control.	S/. 30.00
	<i>Luis Flores</i>	S/. 20.00
	<i>Computadora</i>	S/. 10.00
48	Construcción del robot móvil.	S/. 819.00
49	Inserción de los motores	S/. 219.00
	<i>Luis Flores</i>	S/. 20.00
	<i>Michael Garay</i>	S/. 20.00
	<i>Transporte</i>	S/. 10.00
	<i>Motor DC + Encoders</i>	S/. 141.00
	<i>Llantas de goma</i>	S/. 28.00
50	Inserción de las tarjetas electrónicas.	S/. 400.00
	<i>Luis Flores</i>	S/. 20.00
	<i>Michael Garay</i>	S/. 20.00
	<i>Transporte</i>	S/. 10.00
	<i>Arduino uno Rev3</i>	S/. 160.00
	<i>Bluetooth</i>	S/. 50.00
	<i>Driver de Motores</i>	S/. 20.00
	<i>Tarjeta de conexión</i>	S/. 40.00
	<i>Otros</i>	S/. 80.00
51	Inserción de la batería.	S/. 80.00
	<i>Luis Flores</i>	S/. 20.00
	<i>Michael Garay</i>	S/. 20.00
	<i>Transporte</i>	S/. 10.00
	<i>Baterías 12V</i>	S/. 30.00
52	Inserción de la carrocería.	S/. 120.00
	<i>Luis Flores</i>	S/. 20.00
	<i>Michael Garay</i>	S/. 20.00
	<i>Transporte</i>	S/. 10.00
	<i>Chasis del Móvil</i>	S/. 50.00
	<i>Ball caster</i>	S/. 20.00
53	Prueba de alineamiento del robot en lazo abierto.	S/. 120.00
54	Prueba de distancias Lineales.	S/. 60.00
	<i>Luis Flores</i>	S/. 20.00
	<i>Michael Garay</i>	S/. 20.00
	<i>Computadora</i>	S/. 10.00
	<i>Transporte</i>	S/. 10.00
55	Prueba de distancias Angulares.	S/. 60.00

Id	Nombre de tarea	Costo total
	Luis Flores	S/. 20.00
	Michael Garay	S/. 20.00
	Computadora	S/. 10.00
	Transporte	S/. 10.00
56	Control de posición para el Robot Móvil.	S/. 110.00
57	Preparación de los circuitos del Robot Móvil.	S/. 20.00
	Luis Flores	S/. 20.00
58	Programa del Controlador Proporcional Derivativo.	S/. 30.00
	Luis Flores	S/. 20.00
	Computadora	S/. 10.00
59	Aplicación en Labview.	S/. 60.00
	Luis Flores	S/. 20.00
	Michael Garay	S/. 20.00
	Computadora	S/. 10.00
	Transporte	S/. 10.00
60	Limitaciones del robot Móvil.	S/. 300.00
61	Limitaciones del software Labview.	S/. 120.00
	Michael Garay	S/. 20.00
	Computadora	S/. 10.00
	Transporte	S/. 10.00
	Arduino uno Rev3	S/. 80.00
62	Limitaciones de resolución angular del motor.	S/. 120.00
	Michael Garay	S/. 20.00
	Computadora	S/. 10.00
	Transporte	S/. 10.00
	Arduino uno Rev3	S/. 80.00
63	Limitación de la velocidad de adquisición.	S/. 60.00
	Luis Flores	S/. 20.00
	Michael Garay	S/. 20.00
	Computadora	S/. 10.00
	Transporte	S/. 10.00
64	Análisis del funcionamiento del robot.	S/. 220.00
65	Cálculo del margen de error.	S/. 40.00
	Michael Garay	S/. 20.00
	Computadora	S/. 10.00
	Transporte	S/. 10.00
66	Cálculo de la potencia y tiempo de duración del Robot Móvil.	S/. 30.00
	Luis Flores	S/. 20.00
	Computadora	S/. 10.00
67	Resultados	S/. 50.00
	Luis Flores	S/. 20.00
	Michael Garay	S/. 20.00
	Computadora	S/. 10.00
68	Conclusiones	S/. 50.00
	Luis Flores	S/. 20.00
	Michael Garay	S/. 20.00
	Computadora	S/. 10.00
69	Recomendaciones	S/. 50.00
	Luis Flores	S/. 20.00
	Michael Garay	S/. 20.00
	Computadora	S/. 10.00
70	CAPÍTULO V CRONOGRAMA DE ACTIVIDADES	S/. 30.00
	Luis Flores	S/. 20.00
	Computadora	S/. 10.00
71	CAPÍTULO VI PRESUPUESTO	S/. 30.00
	Luis Flores	S/. 20.00
	Computadora	S/. 10.00
72	CAPÍTULO VII REFERENCIAS BIBLIOGRÁFICAS	S/. 50.00
	Luis Flores	S/. 20.00
	Michael Garay	S/. 20.00
	Computadora	S/. 10.00