



BENEMÉRITA UNIVERSIDAD AUTÓNOMA DE PUEBLA



FACULTAD DE CIENCIAS DE LA ELECTRÓNICA

**MAESTRÍA EN CIENCIAS DE LA ELECTRÓNICA
OPCIÓN EN AUTOMATIZACIÓN**

TESIS DE MAESTRÍA EN CIENCIAS:

**CONTROL DE NAVEGACIÓN DE UN
ROBOT MÓVIL**

COMO REQUISITO PARCIAL PARA OBTENER EL
TÍTULO DE:

**MAESTRO EN CIENCIAS DE LA
ELECTRÓNICA, OPCIÓN EN
AUTOMATIZACIÓN**

PRESENTA:

ALFREDO GARCÍA SUÁREZ

ASESORES:

Dra. AMPARO DORA PALOMINO MERINO
Dr. SERGIO VERGARA LIMON

PUEBLA, PUEBLA, MÉXICO

17 DE AGOSTO DEL 2015.

RESUMEN

En el presente trabajo de investigación se realizó la instrumentación de un vehículo a control remoto tipo Ackerman, implementada como robot móvil para ser utilizada en el seguimiento de trayectorias.

El desarrollo de la electrónica del vehículo se implementó en las instalaciones de la Facultad de Ciencias de la Electrónica de la BUAP, realizando la transmisión de datos a través de comunicación inalámbrica, utilizando módulos de radiofrecuencia de la plataforma XBEE.

En un primer acercamiento se realizó la caracterización física del robot móvil, con el fin de obtener a partir de esta los modelos cinemático y dinámico de la planta.

Una vez hecho esto, se estudiaron e implementaron dos estrategias de control: control predictivo basado en el modelo dinámico y control punto a punto basado en el modelo cinemático. Esta implementación se realizó utilizando un algoritmo de control desarrollado en las plataformas del software LABVIEW en conjunto con el software de ARDUINO.

Se utilizaron tres señales de salida del robot móvil HUMMER H2 (posición del vehículo, dirección del eje de las ruedas delanteras y orientación del vehículo) obtenidas cada una a través de la lectura de tres sensores: encoder en cuadratura, potenciómetro y magnetómetro o compás electrónico respectivamente.

Para cerrar el lazo de control se emplearon 2 señales de entrada al vehículo (PWM de tracción y PWM de dirección), las cuales proporcionan el torque a los dos motores actuadores del vehículo por medio de un puente H de 12V a cada uno, con una corriente máxima de 30A cada puente.

Para el procesamiento de datos en el dispositivo remoto (robot móvil HUMMER H2) se utilizó una tarjeta ARDUINO UNO conectada vía serial a un módulo XBEE, y para el procesamiento de datos en el dispositivo local (PC) se utilizó una tarjeta XBIB-U-DEV conectada vía serial al otro módulo XBEE.

Se realizaron las pruebas con las dos estrategias de control propuestas, utilizando el control punto a punto se obtienen resultados satisfactorios dado que muestran una exactitud del robot móvil a la posición deseada con 5 cm de error.

Por otra parte utilizando el control a través de curvas paramétricas también se obtienen resultados satisfactorios dado que muestran que el robot móvil reproduce una trayectoria altamente fiel a la trayectoria de referencia.

Finalmente se presenta un análisis de resultados del desempeño del robot.

AGRADECIMIENTOS

Quiero dedicar este trabajo de manera especial a mi padre José Alfredo García Paredes por ser mi ejemplo a seguir y porque aunque ya no esté conmigo, con sus consejos me sigue llevando de la mano como cuando yo era pequeño.

A Dios por darme la oportunidad de estar en esta vida y poder luchar por lo que quiero.

A mi madre Edilbertha porque con su ejemplo y cariño, me ha motivado y demostrado que no hay barreras que el esfuerzo y la dedicación no puedan superar.

A mi esposa Diana por todo su apoyo, comprensión, esfuerzo y cariño, necesarios para poder superar los momentos difíciles y forjar el camino de nuestros pequeños.

A mi hija Eli por esforzarse cada día en la escuela y motivarme con sus logros y alegrías que cada día no dejan de sorprenderme.

A mi hijo Alfredito por llenar de alegrías todos los momentos que pasamos juntos en familia y recordarme que si existen los ángeles en la tierra.

A mis hermanas Lucia y Guadalupe por todo lo que hemos aprendido juntos y por el apoyo y motivación para ser mejor cada día.

A mis suegros Gilberto y Alejandra por el apoyo y la confianza que me han brindado en este tiempo.

A mi asesora la Dr. Amparo Dora Palomino Merino, por el apoyo, la orientación y la amistad que me ha brindado para poder culminar este trabajo, pero sobre todo por la confianza que depositó en mi para tomar este tema de tesis y trabajar como un equipo.

A mi co-asesor el Dr. Sergio Vergara Limon por todos los conocimientos que me transmitió en el salón de clases y los consejos que me brindo para tomar mejores decisiones en el desarrollo de la tesis.

Al Dr. Fernando Reyes Cortés por todos los conocimientos transmitidos en clase como en sus libros ya que fueron de gran utilidad para desarrollar este trabajo y entender cualquier parte del mismo.

A mis revisores de tesis y jurado Dr. Eduardo Ríos Silva, Dr. Aurora Vargas Treviño y Dr. Moisés Gutiérrez Arias por sus valiosas correcciones que me ayudaron a mejorar este trabajo.

A Francisco, Jesús, Pablo y a todos mis compañeros y amigos de la maestría que con sus comentarios, puntos de vista y consejos me ayudaron a analizar mejor cada etapa de mi trabajo para poder mejorarlo. También les agradezco por todos los momentos vividos y la amistad que me han brindado en estos dos años.

Al CONACYT por el apoyo institucional y económico recibido, y por darme la oportunidad de superarme personal y profesionalmente.

Y a todas aquellas personas que de una u otra manera me ayudaron y motivaron a la culminación de este trabajo tan significativo para mí.

INDICE

RESUMEN	1
AGRADECIMIENTOS	2
INDICE.....	3
INDICE DE FIGURAS	6
INDICE DE TABLAS	11
INTRODUCCIÓN.....	12
PLANTEAMIENTO DEL PROBLEMA	12
JUSTIFICACIÓN	13
PROPUESTA DE SOLUCIÓN DEL PROBLEMA	14
OBJETIVOS	15
OBJETIVO GENERAL.....	15
OBJETIVOS ESPECÍFICOS	15
1 ANTECEDENTES.....	16
1.1 ESTADO DEL ARTE	20
El robot SAAPIN	21
Proyecto CROMAT	21
EYEBOT	22
1.2 ESTRATEGIAS DE CONTROL EN LA ROBÓTICA MÓVIL ACTUAL....	24
2 ROBOTS MÓVILES.....	25
2.1 DEFINICIÓN Y CLASIFICACIÓN	25
Locomoción.....	25
Percepción	26
Razonamiento.....	27
Comunicación	27
2.2 CARACTERISTICAS GENERALES DE UN RMR TIPO ACKERMAN....	29
Ruedas	29
Giro	29
Arreglos de ruedas.....	30
Restricciones cinemáticas	31
Forma del robot	32
3 DESCRIPCIÓN DE LA PLATAFORMA EXPERIMENTAL	33
3.1 INFRAESTRUCTURA DISPONIBLE.....	33
3.2 PLATAFORMA EXPERIMENTAL HUMMER H2.....	34
Actuadores.....	37
3.3 INSTRUMENTACIÓN	38
Sensores.....	38
Batería	40
Aislador de campo magnético	40
Vehículo	41

4	MODELADO MATEMÁTICO DEL VEHÍCULO HUMMER H2	42
4.1	CARACTERIZACIÓN DEL VEHÍCULO	42
	Caracterización del motor de tracción	43
4.2	MODELOS MATEMÁTICOS	47
	Modelo cinemático	47
	Modelo dinámico	50
	Linealización del modelo dinámico	54
	Modelo dinámico Linealizado representado en espacio de estados	55
5	ESTRATEGIAS DE CONTROL	57
5.1	CONTROL PREDICTIVO BASADO EN EL MODELO DINÁMICO	57
	Representación del modelo dinámico en función de transferencia	57
	Propuesta del controlador predictivo MPC	58
	Control predictivo de LABVIEW	60
	Implementación del controlador MPC	62
	Programación del controlador MPC	62
	Desempeño del controlador predictivo MPC	64
5.2	CONTROL PUNTO- PUNTO BASADO EN EL MODELO CINEMÁTICO	66
	Generación de trayectorias	66
	Definición paramétrica de algunas curvas de interés	67
	Algoritmo generador de trayectorias	68
6	INSTRUMENTACIÓN DEL ROBOT MÓVIL HUMMER H2	70
6.1	PLATAFORMA INICIAL	70
6.2	INSTRUMENTACIÓN PARA LA COMUNICACIÓN INALÁMBRICA	71
	Tarjeta de interfaz USB (Dispositivo Local)	71
	Módulos de radiofrecuencia	73
	Cable USB	76
	Adaptador de voltaje (12V)	76
	Computadora portátil	77
	Vista general del dispositivo Local	77
6.3	INSTRUMENTACIÓN PARA EL PROCESAMIENTO DE DATOS	78
	Tarjetas ARDUINO (Dispositivo Remoto)	78
	Vista general del dispositivo Remoto	81
	Vista general del sistema de comunicación inalámbrica	82
6.4	INSTRUMENTACIÓN PARA LA ADQUISICIÓN DE DATOS	82
	Encoder como sensor de Velocidad (v)	82
	Brújula como sensor de orientación (β)	84
	Potenciómetro como sensor de dirección (δ_D)	85
6.5	INSTRUMENTACIÓN PARA LA ETAPA DE POTENCIA	87
	Doble Puente H VNH2SP30-E	87
	Relevadores	88
	Vista general de la instrumentación del robot móvil HUMMER H2	89
7	IMPLEMENTACIÓN DEL ALGORITMO DE CONTROL	90
7.1	ENTORNO DE PROGRAMACIÓN LABVIEW-ARDUINO-XBEE	90

Introducción al entorno de programación LABVIEW-ARDUINO	90
Instalación del entorno de programación LABVIEW-ARDUINO-XBEE	93
Sincronización de la interfaz LABVIEW-ARDUINO-XBEE	95
7.2 PROCESAMIENTO DIGITAL DE SEÑALES SENSORIALES	98
Encoder	98
Magnetómetro	99
Potenciómetro	102
7.3 PROCESAMIENTO DIGITAL DEL ALGORITMO DE CONTROL.....	103
Modelo Cinemático	103
Generador de Trayectorias	105
7.4 PROCESAMIENTO DIGITAL DE LAS SALIDAS DE CONTROL.....	106
Controlador de dirección.....	106
Controlador de tracción.....	107
7.5 DESCRIPCIÓN GENERAL DEL SISTEMA DE CONTROL	109
8 PRUEBAS EXPERIMENTALES	110
8.1 TRAYECTORIAS CON CURVAS PARAMETRIZADAS.....	110
Arco extendido	111
Media circunferencia.....	116
Senoidal.....	121
Circunferencia	125
8.2 TRAYECTORIAS CON CONTROL PUNTO A PUNTO	127
Posición deseada (1, 0).....	127
Posición deseada (1, -1)	128
Posición deseada (3, -1)	129
Posición deseada (3, 2).....	131
Posición deseada (1, 0.5) y (2, -1.5).....	132
8.3 RESULTADOS	133
Arco extendido	134
Media circunferencia.....	136
Senoidal.....	138
Posiciones deseadas	140
CONCLUSIONES	142
APÉNDICES	144
APÉNDICE A. Tarjeta XBIB-U-DEV	144
APÉNDICE B. Características técnicas de los módulos XBEE-PRO.....	144
APÉNDICE C. Especificaciones del adaptador de voltaje.....	147
APÉNDICE D. Hoja de datos del magnetómetro HMC5883L	148
APÉNDICE E. Hoja de datos del puente h VNH2SP30.....	150
PUBLICACIONES.....	152
REFERENCIAS	153

INDICE DE FIGURAS

Figura 1.1. El Roboscrub, para la aspiración y limpieza de pisos.....	17
Figura 1.2. El Robokent, para labores de limpieza.....	18
Figura 1.3. Robot de Seguridad Denning Sentry.....	18
Figura 1.4. El Helpmate o Robot Enfermero.....	19
Figura 1.5. Robot Fletch, de uso militar.....	19
Figura 1.6. Sweeperbot. Un Robot Limpiador de Superficie.....	20
Figura 1.7. Manepa. Un Robot de Limpieza.....	20
Figura 1.8. SAAPIN robot de riego.....	21
Figura 1.9. Robot de exploración CROMAT.....	22
Figura 1.10. Robot EYEBOT, en el que corre el sistema operativo ROBIOS.....	23
Figura 2.1. Clasificación de los robots móviles por locomoción.....	25
Figura 2.2. Clasificación de los robots móviles con ruedas (RMR).....	26
Figura 2.3. Clasificación de los robots móviles con ruedas (RMR).....	27
Figura 2.4. Conexión de un PC con un auto a través de una interfaz.....	28
Figura 2.5. Protocolos de comunicación inalámbrica.....	28
Figura 2.6. Desplazamiento de la rueda.....	29
Figura 2.7. a) CCI ubicado en la intersección de los ejes de las ruedas, debido a trayectoria circular; b) CCI ubicado en el infinito, debido a trayectoria recta.....	29
Figura 2.8. Arreglo Diferencial.....	30
Figura 2.9. Arreglo Síncrono.....	30
Figura 2.10. Arreglo Triciclo.....	31
Figura 2.11. Arreglo Ackerman.....	31
Figura 2.12. a) Restricciones holonómicas, b) Restricciones no holonómicas.....	32
Figura 2.13. a) Robot cilíndrico, b) Robot cuadrado.....	32
Figura 3.1. Vehículo de radio control HUMMER H2.....	33
Figura 3.2. Módulos de desarrollo 2.4 MHZ Digi-Mesh.....	34
Figura 3.3. Carcasa del robot móvil HUMMER H2.....	35
Figura 3.4. Muelle delantero del robot móvil HUMMER H2.....	35
Figura 3.5. Muelle trasero del robot móvil HUMMER H2.....	36
Figura 3.6. Porta batería del robot móvil HUMMER H2.....	36
Figura 3.7. Eje transversal del robot móvil HUMMER H2.....	36
Figura 3.8. Llantas tipo estándar del robot móvil HUMMER H2.....	37
Figura 3.9. Tornillos del robot móvil HUMMER H2.....	37
Figura 3.10. Servomotor de dirección del robot móvil HUMMER H2.....	38
Figura 3.11. Motor de tracción del robot móvil HUMMER H2.....	38
Figura 3.12. Encoder utilizado como sensor de la posición del robot móvil HUMMER H2.....	39
Figura 3.13. Magnetómetro utilizado como sensor de la orientación del robot móvil HUMMER H2.....	39

Figura 3.14. Potenciómetro utilizado como sensor de la dirección del robot móvil HUMMER H2.	40
Figura 3.15. Batería de alimentación del robot móvil HUMMER H2.	40
Figura 3.16. Aislador de campo magnético del magnetómetro.	41
Figura 3.17. Robot móvil HUMMER H2.	41
Figura 4.1. Caracterización del motor de tracción por voltaje.	44
Figura 4.2. Respuesta de la caracterización del motor de tracción por voltaje.	45
Figura 4.3. Respuesta de la caracterización del motor de tracción por ciclo de trabajo a una frecuencia de 480Hz.	46
Figura 4.4. Representación geométrica de la postura de un robot móvil tipo Ackerman.	48
Figura 4.5. Análisis Geométrico del RMR tipo carro.	48
Figura 4.6. Diagrama de fuerzas del vehículo HUMMER H2.	50
Figura 4.7. Graficas obtenidas con la representación del modelo dinámico en espacio de estados.	56
Figura 5.1. Horizontes de control y predicción.	61
Figura 5.2. Algoritmo del controlador MPC.	63
Figura 5.3. Implementación del controlador MPC en LABVIEW.	64
Figura 5.4. Implementación del modelo dinámico del ángulo de dirección (δD) en LABVIEW.	64
Figura 5.5. Implementación del controlador MPC en LABVIEW.	65
Figura 5.6. Desempeño del controlador predictivo utilizando la dinámica del ángulo de dirección (δD).	65
Figura 5.7. Curvas parametrizadas en el tiempo.	68
Figura 6.1. Tarjeta de interfaz USB (XBIB-U-DEV).	71
Figura 6.2. Interfaz física e indicadores de la tarjeta USB.	72
Figura 6.3. Conexiones mínimas requeridas para el módulo XBEE-PRO.	74
Figura 6.4. Sistema de flujo de datos mediante microcontroladores.	74
Figura 6.5. Diagrama de pines Vista Superior.	75
Figura 6.6. Cable USB.	76
Figura 6.7. Adaptador de voltaje 12 VDC.	76
Figura 6.8. Computadora portátil.	77
Figura 6.9. Dispositivo Local.	77
Figura 6.10. Arduino UNO con microcontrolador en formato DIP.	78
Figura 6.11. Arduino UNO con microcontrolador en formato SMD.	79
Figura 6.12. Placa Arduino SHIELD para XBEE.	81
Figura 6.13. Dispositivo Remoto.	81
Figura 6.14. Sistema de comunicación inalámbrica.	82
Figura 6.15. Principio utilizado para la implementación del encoder.	83
Figura 6.16. Encoder implementado en el robot móvil HUMMER H2.	83
Figura 6.17. Conexión de la brújula HMC5883L a Arduino UNO.	84
Figura 6.18. La brújula electrónica HMC5883L implementada en el robot móvil HUMMER H2.	85

Figura 6.19. Sensor del ángulo de dirección de las ruedas delanteras.....	86
Figura 6.20. Potenciómetro empotrado en eje delantero del robot móvil HUMMER H2.	86
Figura 6.21. Doble Puente H VNH2SP30-E	87
Figura 6.22. Implementación del doble Puente H VNH2SP30-E en el robot móvil HUMMER H2.	88
Figura 6.23. Relevadores para la activación de las luces del vehículo.....	89
Figura 6.24. Instrumentación del robot móvil HUMMER H2.	89
Figura 7.1. Interfaz gráfica de usuario.....	91
Figura 7.2. Programación gráfica en bloques (V'S).	91
Figura 7.3. Desarrollo interactivo.....	92
Figura 7.4. Librerías en LABVIEW.	92
Figura 7.5. Entorno de programación LABVIEW 2011.	93
Figura 7.6. VISA-NIMAX 5.5.....	93
Figura 7.7. JKI VI Package Manager (VIPM) 2014.....	94
Figura 7.8. Descarga de toolkit interfaz LABVIEW para Arduino.....	94
Figura 7.9. Arduino IDE 1.0.4.....	95
Figura 7.10. Software X.CTU.	95
Figura 7.11. Archivo LIFA_BASE.	96
Figura 7.12. Configuración de parámetros a través del controlador VISA NI MAX.....	97
Figura 7.13. Configuración de parámetros a través del administrador de dispositivos de Windows 8.1.....	97
Figura 7.14. Programa para la lectura del Encoder en LABVIEW.	98
Figura 7.15. Interfaz de usuario del Encoder.....	99
Figura 7.16. Programa para la lectura del Magnetómetro en LABVIEW.....	100
Figura 7.17. Mapa del valor de la declinación magnética.	101
Figura 7.18. Interfaz de usuario del Magnetómetro.	101
Figura 7.19. Programa para la lectura del Potenciómetro en LABVIEW.	102
Figura 7.20. Interfaz de usuario del Potenciómetro.....	103
Figura 7.21. Plano virtual (x, y) desarrollado con el modelo cinemático del vehículo.104	
Figura 7.22. Plano virtual que representa la magnitud de referencia y del vehículo en los ejes “x” y “y”.....	104
Figura 7.23. Plano virtual que representa el error del vehículo en los ejes “x” y “y”..	105
Figura 7.24. Programa para el generador de trayectorias en LABVIEW.....	105
Figura 7.25. Interfaz de usuario para el generador de trayectorias.....	106
Figura 7.26. Coordenadas almacenadas en cada periodo de muestreo.....	106
Figura 7.27. Implementación de la ley de control de dirección.	107
Figura 7.28. Implementación de la ley de control de tracción.	108
Figura 7.29. Interfaz de usuario para las salidas de control.	108
Figura 7.30. Variables de control mostradas en la interfaz de usuario.....	108
Figura 7.31. Vista a bloques del sistema de control del robot móvil HUMMER H2... 109	
Figura 8.1. Espacio de trabajo para realizar las pruebas del robot móvil.....	110
Figura 8.2. Sistema implementado para el monitoreo del vehículo.	111

Figura 8.3. Trayectoria generada en la primera prueba del arco extendido.	112
Figura 8.4. Interfaz de usuario en la primera prueba del arco extendido.	113
Figura 8.5. Trayectoria generada en la segunda prueba del arco extendido.....	113
Figura 8.6. Interfaz de usuario en la segunda prueba del arco extendido.....	114
Figura 8.7. Trayectoria generada en la tercera prueba del arco extendido.....	114
Figura 8.8. Interfaz de usuario en la tercera prueba del arco extendido.....	115
Figura 8.9. Trayectoria generada en la cuarta prueba del arco extendido.	115
Figura 8.10. Interfaz de usuario en la cuarta prueba del arco extendido.....	116
Figura 8.11. Trayectoria generada en la primera prueba de la mitad de circunferencia.	117
Figura 8.12. Trayectoria generada en la primera prueba de la mitad de circunferencia.	117
Figura 8.13. Trayectoria generada en la segunda prueba de la mitad de circunferencia.	118
Figura 8.14. Trayectoria generada en la segunda prueba de la mitad de circunferencia.	118
Figura 8.15. Trayectoria generada en la tercera prueba de la mitad de circunferencia.	119
Figura 8.16. Trayectoria generada en la tercera prueba de la mitad de circunferencia.	119
Figura 8.17. Trayectoria generada en la cuarta prueba de la mitad de circunferencia.	120
Figura 8.18. Trayectoria generada en la cuarta prueba de la mitad de circunferencia.	120
Figura 8.19. Trayectoria generada en la primera prueba de la curva senoidal.	121
Figura 8.20. Trayectoria generada en la primera prueba de la curva senoidal.	122
Figura 8.21. Trayectoria generada en la segunda prueba de la curva senoidal.	122
Figura 8.22. Trayectoria generada en la segunda prueba de la curva senoidal.	123
Figura 8.23. Trayectoria generada en la tercera prueba de la curva senoidal.....	123
Figura 8.24. Trayectoria generada en la tercera prueba de la curva senoidal.....	124
Figura 8.25. Trayectoria generada en la cuarta prueba de la curva senoidal.....	124
Figura 8.26. Trayectoria generada en la cuarta prueba de la curva senoidal.....	125
Figura 8.27. Trayectoria generada en el seguimiento de una circunferencia.	126
Figura 8.28. Trayectoria generada utilizando una circunferencia como curva paramétrica.	126
Figura 8.29. Trayectoria generada para alcanzar la posición deseada (1, 0).	127
Figura 8.30. Trayectoria graficada para alcanzar la posición deseada (1, 0).	128
Figura 8.31. Trayectoria generada para alcanzar la posición deseada (1, -1).	128
Figura 8.32. Trayectoria graficada para alcanzar la posición deseada (1, -1).	129
Figura 8.33. Trayectoria generada para alcanzar la posición deseada (3, -1).	130
Figura 8.34. Trayectoria graficada para alcanzar la posición deseada (3, -1).	130
Figura 8.35. Trayectoria generada para alcanzar la posición deseada (3, 2).	131
Figura 8.36. Trayectoria graficada para alcanzar la posición deseada (3, 2).	131
Figura 8.37. Trayectoria generada para alcanzar la posición deseada (1, 0.5) y (2, -1.5).	132
Figura 8.38. Trayectoria graficada para alcanzar la posición deseada (1, 0.5) y (2, -1.5).	133

Figura 8.39. Tabla en la interfaz de usuario, utilizada para guardar muestras.	134
Figura 8.40. Gráfica de la prueba 1 del Arco extendido.....	134
Figura 8.41. Gráfica de la prueba 2 del Arco extendido.....	135
Figura 8.42. Gráfica de la prueba 3 del Arco extendido.....	135
Figura 8.43. Gráfica de la prueba 4 del Arco extendido.....	135
Figura 8.44. Gráfica de la prueba 1 de la media circunferencia.	136
Figura 8.45. Gráfica de la prueba 2 de la media circunferencia.	136
Figura 8.46. Gráfica de la prueba 3 de la media circunferencia.	137
Figura 8.47. Gráfica de la prueba 4 de la media circunferencia.	137
Figura 8.48. Gráfica de la prueba 1 de la curva senoidal.	138
Figura 8.49. Gráfica de la prueba 2 de la curva senoidal.	138
Figura 8.50. Gráfica de la prueba 3 de la curva senoidal.	139
Figura 8.51. Gráfica de la prueba 3 de la curva senoidal.	139
Figura 8.52. Gráfica de la prueba con la posición deseada (1, 0).....	140
Figura 8.53. Gráfica de la prueba con la posición deseada (1, -1).	140
Figura 8.54. Gráfica de la prueba con la posición deseada (3, 2).....	141
Figura 8.55. Gráfica de la prueba con la posición deseada (1, 0.5) a (2, -1,5).....	141

INDICE DE TABLAS

Tabla 3.1. Peso de los componentes del robot móvil HUMMER H2.	41
Tabla 4.1. Parámetros intrínsecos del robot móvil HUMMER H2.	43
Tabla 4.2. Caracterización del motor de tracción por voltaje.....	45
Tabla 4.3. Caracterización del motor de tracción por ciclo de trabajo.....	46
Tabla 6.1. Conexión del módulo XBEE-PRO con la tarjeta XBIB-U-DEV.....	73
Tabla 6.2. Características de la tarjeta Arduino UNO.....	80

INTRODUCCIÓN

PLANTEAMIENTO DEL PROBLEMA

A nivel mundial ocurren graves problemas urbanos de destrucción masiva, ya sean causados por la naturaleza o por el hombre, tales como terremotos, inundaciones, guerras e incluso ataques terroristas. Estos sucesos lamentablemente traen consigo un alto número de víctimas, y bajo estas graves circunstancias se requiere la colaboración de los cuerpos de bomberos en el lugar de la catástrofe. Los miembros de estos organismos de rescate realizan una difícil labor, exponiendo sus propias vidas a grandes riesgos para rescatar a las víctimas.

Por otro lado también ocurren graves problemas en la vida cotidiana en los trabajos expuestos en zonas de alto riesgo como en minas y en lugares insalubres para el ser humano como los canales de drenaje.

Gracias a los avances tecnológicos actuales, recientemente en el área de robótica surge una aplicación de gran interés global por su carácter humanitario, en la cual, países como los Estados Unidos y Japón participan activamente.

Consiste en la utilización de robots de rescate en la tarea de detección de víctimas en lugares que se encuentran en condiciones de emergencia, destinados específicamente a la asistencia de los cuerpos de rescate, como los bomberos o protección civil.

También la utilización de robots para detectar minas, intrusos en el caso de vigilancia, o simplemente para realizar un trabajo cotidiano como el riego en un campo de cultivo.

A largo plazo se busca la implementación de un sistema de soporte formado por un conjunto de robots equipados con sensores especializados, como cámaras infrarrojas, sensores de calor, sensores químicos, sistemas de control remoto, entre otros, los cuales realicen una aplicación en conjunto en beneficio de la humanidad.

De este modo, el presente proyecto pretende realizar un aporte a este campo de investigación desarrollando la navegación de un robot móvil a través de comunicación inalámbrica.

JUSTIFICACIÓN

La robótica móvil se considera actualmente un área de la tecnología avanzada manejadora de problemas de alta complejidad. Sus productos se constituyen por aspectos multidisciplinarios como: control, programación, inteligencia artificial, percepción e instrumentación, y sirven de base para el avance en diversos campos de la industria, aportando soluciones tecnológicas innovadoras orientadas al desarrollo de mejores robots y a la ampliación del abanico de aplicaciones disponibles. Así, este campo de la investigación está desarrollándose en todo momento, quedando aún mucho camino por recorrer.

El estudio de la dinámica de los robots, permite conocer el comportamiento del sistema. Con esta información, es posible diseñar y/o seleccionar estrategias de control en base a su desempeño particular.

La investigación y desarrollo de sistemas contribuye a la generación de conocimiento, los cuales brindan información, elementos y partes que permiten crear cualquier sistema, constituyéndose en un apoyo educativo muy fuerte a partir de la utilización potencial del software, piezas mecánicas, sensores, sistemas de desarrollo y demás elementos necesarios para que el estudiante construya sus propios desarrollos asimilando el conocimiento requerido.

El determinar el modelo dinámico de un sistema robótico, permitirá contar con una plataforma experimental didáctica, que permita a los estudiantes de la MCEA, aplicar sus conocimientos de diseño y estrategias aprendidas en los cursos.

PROPUESTA DE SOLUCIÓN DEL PROBLEMA

Luego de hacer un análisis de las necesidades de avance tecnológico en la actualidad, y la demanda de aplicaciones dentro del campo de la robótica móvil para mejorar la eficiencia en las tareas cotidianas y a la vez disminuir el riesgo de trabajo para el ser humano, planteamos una solución factible de comunicación inalámbrica para el control de un robot móvil.

Este robot móvil tiene como característica principal ser un robot con ruedas diseñado para su uso en un espacio específico, controlado por radio frecuencia; en donde su trayectoria será definida así como los giros y movimientos realizados dentro de un área acotada ó espacio de trabajo.

Con este proyecto se impulsa a la implementación de cualquier aplicación a corto plazo tomando las mismas bases para un control inalámbrico al sistema aplicado.

OBJETIVOS

OBJETIVO GENERAL

Controlar la navegación de un robot móvil terrestre, empleando tecnología inalámbrica.

OBJETIVOS ESPECÍFICOS

- 1) Estudiar los conceptos de robots móviles
- 2) Caracterizar el robot móvil
- 3) Obtener el modelo dinámico del robot HUMMER
- 4) Estudio de la instrumentación electrónica y de los módulos de transmisión
- 5) Proponer una estrategia de control para controlar la navegación del robot móvil
- 6) Pruebas experimentales
- 7) Publicación de Resultados
- 8) Escritura de la Tesis

CAPÍTULO

1 ANTECEDENTES

A pesar que pareciera que la robótica no tiene más de un siglo de antigüedad, sus primeras apariciones emergen en el siglo I a. C. En esa época se llevaron a cabo diseños de más de cien máquinas y autómatas distintos, destacando entre sus principales inventores a Herón de Alejandría, creador de “Autómata”: el primer libro de robótica de la historia [1].

Posteriormente fué Leonardo da Vinci en 1495, quien diseñando un autómata humanoide conocido como: “Caballero mecánico” dió inicio a la idea de robot.

La palabra robot proviene del checo “robota” (trabajo forzado) y de “rabota” (servidumbre).

Aunque la robótica tiene sus inicios hace más de quinientos años, no es hasta el año 1966 en que se desarrolla formalmente por la universidad de Stanford la disciplina conocida como “Robótica Móvil”, siendo “Shakey” el primero en su tipo [1].

Un Robot móvil es un sistema electromecánico con gran capacidad de desplazamiento, basado en un carro o plataforma y dotado de un sistema locomotor rodante. Estas maquinas pueden seguir un determinado camino por telemando, o bien guiándose por información recibida de su entorno a través de sensores.

La robótica es una rama de estudio dedicada al diseño, elaboración, control y construcción de los robots.

La robótica combina diversas áreas del conocimiento humano, entre ellas se pueden mencionar a la física, la matemática, la ingeniería, la biología, entre otras [1].

Actualmente la robótica es una disciplina desarrollada masivamente en universidades, industrias y centros de investigación de todo el mundo.

La robótica móvil constituye una valiosa herramienta para el desarrollo de tecnologías para la creación de robots de navegación autónoma. Para que el robot tenga la capacidad necesaria de ejecutar ciertas tareas a través de su movimiento es importante conocer previamente el espacio de trabajo.

La navegación es la técnica de conducir un robot móvil mientras atraviesa un entorno para alcanzar un destino o meta sin chocar con ningún obstáculo.

Las propiedades características de los robots son la versatilidad y la auto-adaptabilidad. La primera se entiende como la potencialidad estructural de ejecutar tareas diversas y/o ejecutar una misma tarea de forma diversa, lo cual implica una estructura mecánica de geometría variable. La auto-adaptabilidad significa que un robot debe, por sí solo, alcanzar su objetivo a pesar de las perturbaciones imprevistas del entorno a lo largo de la ejecución de su tarea; esto supone una consciencia del entorno otorgada por la posesión de sentidos artificiales.

Un robot operacional está constituido por cuatro sistemas relacionados entre sí: mecánico, sensorio, control y alimentación. El sistema mecánico permite el movimiento, utilizando diferentes elementos de acuerdo con el propósito del robot y ubicados estratégicamente. El sistema sensorio brinda información importante para la ubicación de elementos u obstáculos: los sensores propioceptivos permiten determinar posición y velocidad; los sensores exteroceptivos determinan la interacción del entorno con él mismo. El sistema de control es el cerebro del robot, al interpretar y tratar la información; él debe contener un modelo del robot físico (actuadores y sensores), un modelo del entorno o estrategias para conocerlo, y los programas que permiten realizar las tareas (algoritmo de control). El sistema de alimentación suministra la energía a los sensores, actuadores y a todos los dispositivos que conforman del sistema [1].

En 1991 una compañía dedicada a comercializar robots móviles, la Denning Mobile Robots y Windsor Industries, produjo el Roboscrub ilustrado en la Figura 1.1, un robot que aspira y limpia pisos de grandes capacidades [1].



Figura 1.1. El Roboscrub, para la aspiración y limpieza de pisos.

El RoboKent fue otro robot, fabricado por Cleaning Equipments en 1988, como se observa en la Figura 1.2. Este robot fue diseñado para labores de limpieza de espacios grandes; este ha sido tal vez el equipo de este tipo de mayor éxito en Norteamérica. Posee un sistema de detección tipo sonar y un control automático o manual que le indica la ruta establecida para la limpieza; posteriormente, el robot realiza autónomamente la tarea programada [2].



Figura 1.2. El Robokent, para labores de limpieza.

La compañía Demming Mobile Robotics diseñó el Denning Sentry (Figura 1.3), destinado a ser un robot de seguridad. Mediante sensores de detección de intrusos él puede hacer patrullajes de manera infatigable en una instalación, bodega o similares. Cuando la alimentación se agota, el robot retorna automáticamente a su estación de carga y recarga sus baterías sin asistencia de ningún operador; el equipo incorpora un anillo de sonares para detección de obstáculos, sensores de movimiento de tipo infrarrojo y de microondas. También posee una cámara de TV, micrófono y transmisores inalámbricos para enviar la información a la estación de seguridad [2].



Figura 1.3. Robot de Seguridad Denning Sentry.

El Helpmate mostrado en la Figura 1.4 es un robot enfermero fabricado por Helpmate Robotics Inc. Nace de la reducción de la efectividad de los enfermeros y otros

trabajadores de hospitales debido al tiempo que pierden en labores incidentales, y fue diseñado para aliviar el peso de tales tareas. Recorre los corredores de los hospitales combinando el sistema de navegación y la ubicación por sonar. Presionando un botón, el operario puede enviar el robot a determinado lugar del mapa y navegar de una estación programada a otra, para lograr este objetivo el equipo incorpora sonares para evitar obstáculos [3].



Figura 1.4. El Helpmate o Robot Enfermero.

Otro de los robots más interesantes es el Fletch (Ilustrado en la Figura 1.5), diseñado con el propósito de remover las pequeñas bombas y granadas de la munición antitanque que no explotan en el campo de batalla. La munición es soltada desde aeronaves y se calcula que el 25% de ella no explota, lo que genera un grave problema. La tarea de remoción es perfecta para un robot; en la evaluación se demostró que ellos pueden deshacerse incluso de explosivos de artillería. El robot tuvo éxito en su navegación autónoma a un punto, en la búsqueda de munición en espiral, y en su localización, levantamiento y transporte a un punto [4].



Figura 1.5. Robot Fletch, de uso militar.

1.1 ESTADO DEL ARTE

El estudio de Robótica en Colombia ha tomado un nivel de interés muy alto por parte de diferentes universidades. La Universidad de los Andes se encuentra trabajando en sistemas de detección de minas antipersonales y ha desarrollado plataformas posibles para mecanismos. En la Facultad Tecnológica de la Universidad Distrital F.J.C., Los concursos realizados hasta la fecha comprenden: solución de un laberinto, seguidores de línea, limpiadores de superficie, y actualmente en curso luchadores de Sumo [5].

Uno de los diseños tecnológicos aplicados actualmente es el desarrollado por Megacorp, el cual consiste en un robot limpiador de superficie descrito a continuación. Sweeperbot, mostrado en la Figura 1.6, es un robot con un algoritmo autónomo diseñado para limpiar superficies (igual a un carro escoba autónomo), programado en un PIC. Posee un sistema de detección diseñado a partir de sensores ópticos, y un sistema de limpieza desarrollado a partir de un rodillo giratorio tipo escoba; tiene además un sistema de recolección independiente, muy similar a las bolsas de aspiradoras.



Figura 1.6. Sweeperbot. Un Robot Limpiador de Superficie.

Otro diseño tecnológico aplicado por parte de Megacorp es Manepa, el cual es un robot autónomo con un algoritmo de control sencillo implementado en un PLC. Su sistema sensorico se encuentra compuesto por switches y posee un sistema de limpieza tipo aspiradora y un sistema de movilidad tipo triciclo, como se observa en la Figura 1.7. El inconveniente fundamental del diseño es el consumo de energía del robot, que disminuye su tiempo de autonomía.



Figura 1.7. Manepa. Un Robot de Limpieza.

En comparación del desarrollo tecnológico en países de tercer mundo, los avances alcanzados por las potencias internacionales difieren en el alcance económico científico y tecnológico tal es el caso de la robótica móvil en el mundo. Algunas aplicaciones en este ámbito se mencionan a continuación [5].

El robot SAAPIN

El robot SAAPIN (Sistema Autónomo para Agricultura de Precisión e Integrada) ha sido desarrollado en el Centro de Investigación y Tecnología Agroalimentaria de Aragón (Figura 1.8). Su objetivo es crear un vehículo agrícola capaz de desplazarse de forma autónoma por un campo, arrastrando un apero que pueda tomar los datos necesarios para realizar un mapa de salinidad. Así mismo, con otros aperos es capaz de distinguir las plantas de maíz de las malas hierbas cercanas y proceder a su escarda mecánica de forma selectiva [5].

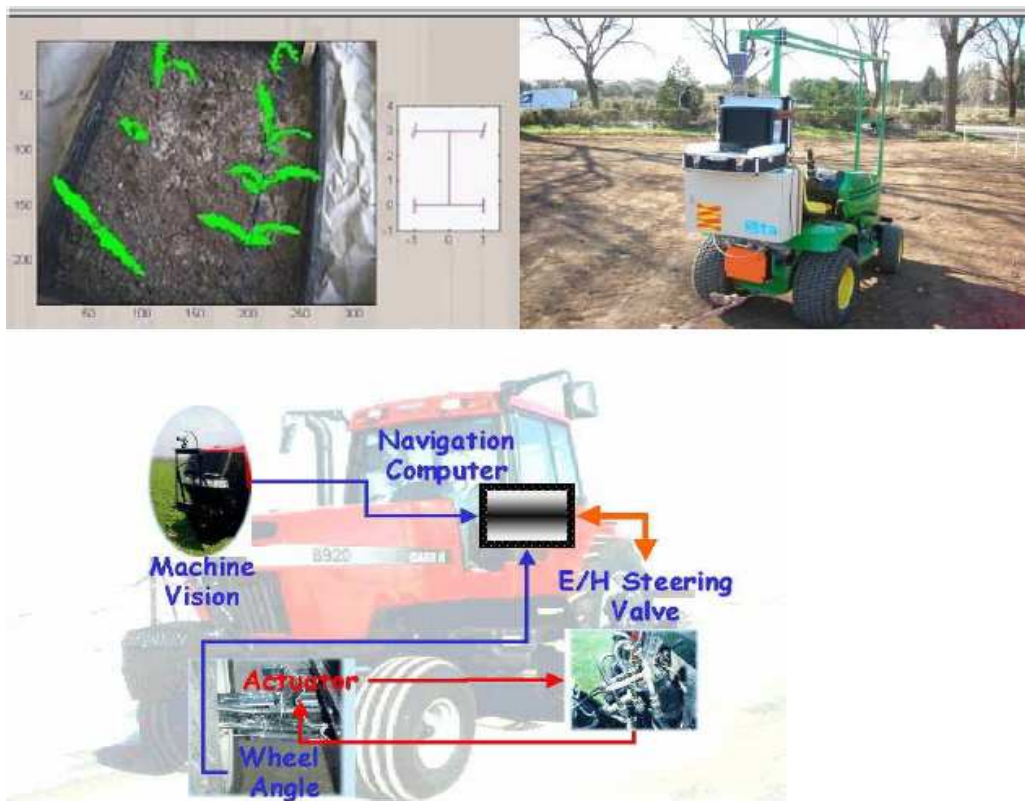


Figura 1.8. SAAPIN robot de riego.

Proyecto CROMAT

Realizado por las universidades de Sevilla, Málaga y Vigo en el año 2003 y 2005, el objetivo del Proyecto CROMAT (Figura 1.9) es generar nuevos métodos y técnicas que permitan la actuación coordinada de robots móviles aéreos y terrestres. Se pretende con ello desarrollar tecnologías de relevancia para aplicaciones tales como inspección de

instalaciones, infraestructuras (puentes, presas) y grandes edificaciones; la detección y monitorización de catástrofes (incendios, inundaciones, erupciones de volcanes, terremotos), exploración; vigilancia, seguridad ciudadana; detección y desactivación de minas anti persona y otras [5].



Figura 1.9. Robot de exploración CROMAT.

EYEBOT

En el mercado de robots móviles programables gran parte de los robots pequeños tienen su propio sistema operativo dedicado, muy ligado al procesador, a los sensores y a los actuadores concretos. El sistema operativo ROBIOs en el robot EyeBot (Figura 1.10), es un ejemplo relevante desarrollado en el Instituto I.E.S. Silverio Lanza. En GETAFE MADRID por el profesor Jerónimo Lobo Núñez y alumnos en el año 2004.

Con la irrupción de las computadoras personales (portátiles o no) como procesadores principales en los robots, los sistemas operativos de propósito general como Linux o MS-Windows han entrado en el mundo de los robots. Estos sistemas, además de los drivers del hardware, incluyen abstracciones y un conjunto muy extenso de bibliotecas genéricas, que también son útiles para la programación de robots. Por ejemplo, bibliotecas e interfaces para la multitarea, las comunicaciones y las interfaces gráficas ROBIOs es una muestra significativa de sistema operativo dedicado. El robot EyeBot es un robot de pequeño tamaño que viene equipado con un microprocesador Motorola 68332 a 35 MHz. Al robot se le conectan dos motores con encoders, tres sensores de infrarrojos y una cámara digital. Su hardware también incluye una pequeña pantalla,

cuatro botones y un radioenlace. El sistema operativo permite la carga de programas a través de un puerto serie y su ejecución pulsando los botones. Como acceso básico al hardware desde los programas, ROBIOS incluye una API (Application Programming Interface) con funciones para recoger las lecturas de los sensores de infrarrojos, para obtener las imágenes de la cámara y para hacer girar los motores a cierta velocidad. En cuanto a comunicaciones, ROBIOS incluye dos funciones que le permiten enviar y recibir bytes a través del radioenlace, a otros EyeBots o un PC que se encuentren dentro del alcance radio de su antena [5].

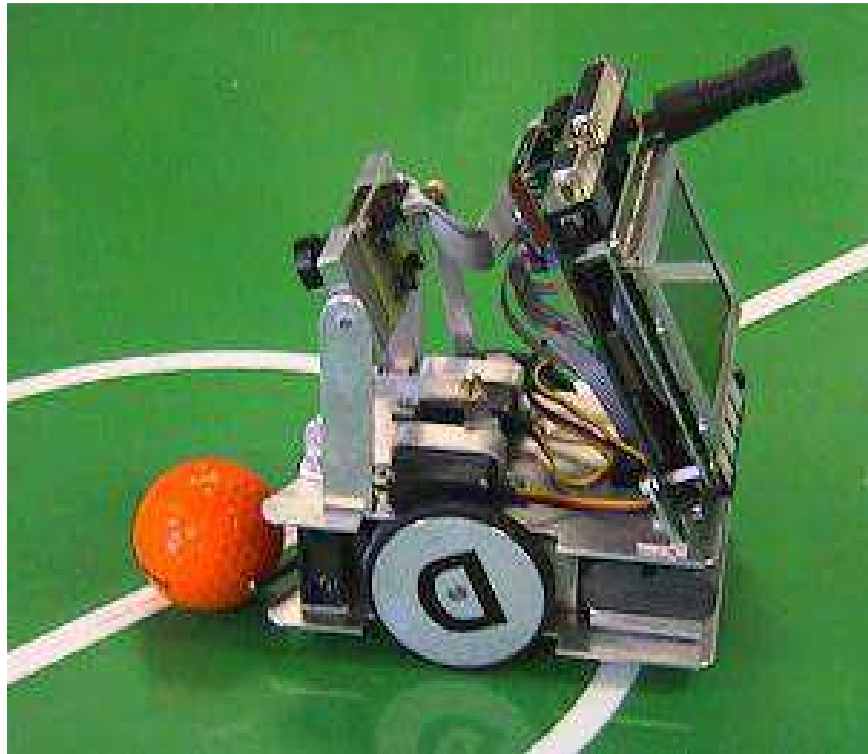


Figura 1.10. Robot EYEBOT, en el que corre el sistema operativo ROBIOS por radiofrecuencia.

1.2 *ESTRATEGIAS DE CONTROL EN LA ROBÓTICA MÓVIL ACTUAL*

En el trabajo presentado por Seyr M. [6] se muestra un método basado en control predictivo para solucionar el problema del control no-holónimo, en el cual se emplea un algoritmo de Gauss-Newton para el control predictivo lo que permite minimizar el error de posición. Por otro lado, GU D. y Hu H. [7] presentan un controlador de modelo predictivo o Receding Horizon (RH). Este tipo de control es frecuentemente utilizado en la optimización de técnicas de control en la industria, se diseña principalmente para tratar el problema de restricciones y se trata de un algoritmo de mejora que predice las salidas del sistema en base a los estados que en ese momento estén presentes y al modelo propio del sistema.

Uno de los métodos más empleados en el seguimiento de trayectoria es el de persecución pura (pure pursuit), en [8]. Este método genera arcos entre el punto de desplazamiento del móvil y los puntos de la trayectoria a seguir; los arcos se generan de 10 a 15 veces por segundo lo que resulta en un seguimiento suave y con muy buenos resultados. Otro método para el seguimiento de trayectorias es conocido como ajuste de polinomios de quinto orden (Quintic Polynomial Fit) [8]. Otra técnica recientemente empleada es mediante linealización de entrada-salida [9] la cual impone que las variables de estado tiendan asintóticamente a la trayectoria deseada, generando una dinámica remanente asociada a una variable de estado, la cual resulta ser estable.

Otro trabajo interesante lo presentan Li X. y Zell A. [10]; ahí se propuso un control lineal por retroalimentación proporcional para lograr que el móvil empujara una pelota sobre una trayectoria preestablecida. Cabe mencionar que los trabajos sobre los RMR que se enfocan al control mediante Internet, i.e., RMR tele operados [11]. Como otra vertiente y por último, un trabajo en donde se utiliza un RMR como plataforma móvil para transportar un brazo robótico se presenta en [12]. Es importante de mencionar que existen muchas técnicas de control clásicas y modernas que se han aplicado a la solución del problema de seguimiento de trayectorias.

CAPÍTULO

2 ROBOTS MÓVILES

2.1 DEFINICIÓN Y CLASIFICACIÓN

Se define a un robot móvil como un dispositivo formado por componentes físicos y computacionales, divididos en cuatro subsistemas: Locomoción, Percepción, Razonamiento y Comunicación [13].

Locomoción

Es la capacidad de los robots para desplazarse de un lugar a otro. Los robots móviles se pueden clasificar según los elementos que emplean para realizar la locomoción en: ruedas, patas, orugas y otros, como se ilustra en la figura 2.1., aunque para efectos del trabajo actual se describirán únicamente los robots móviles con ruedas y los robots móviles con patas.

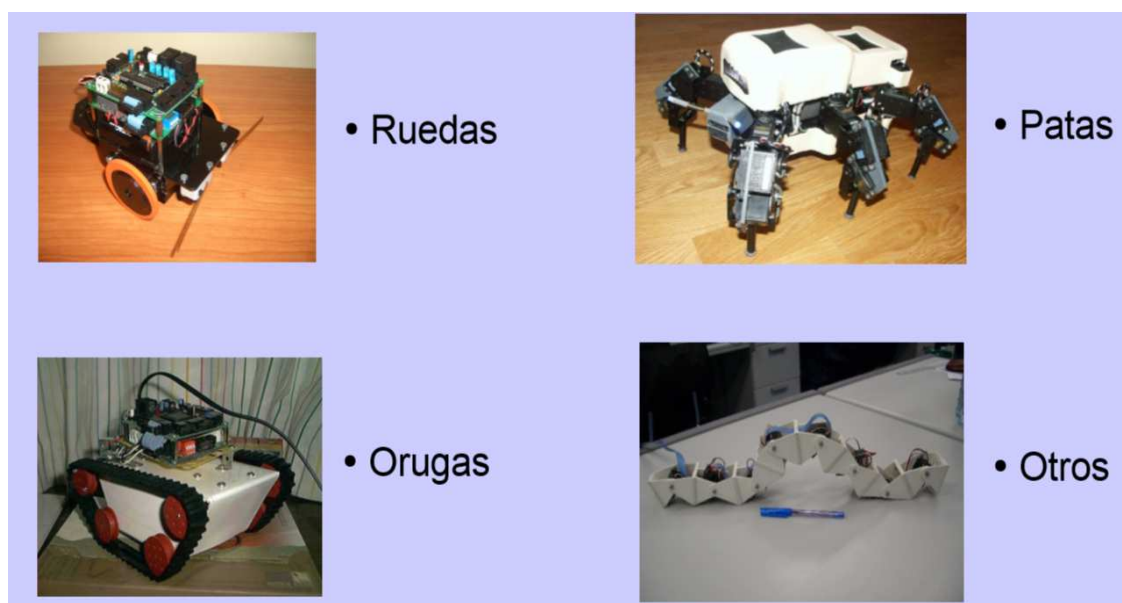


Figura 2.1. Clasificación de los robots móviles por locomoción.

A) ROBOTS MÓVILES CON RUEDAS (RMR). Los robots móviles terrestres poseen diversas aplicaciones en la industria, tales como el bodegaje, inspección y control de producción a distancia. Los robots móviles se han introducido recientemente en la forma de aspiradoras para el hogar. Las diferentes configuraciones que existen de robots móviles con ruedas se describen en la figura 2.3. Los vehículos de ruedas son los más populares por varias razones prácticas, son más sencillos y más fáciles de construir, la carga que pueden transportar es mayor a la que pueden soportar otros tipos de robots móviles. A esto se puede añadir el que se pueden utilizar vehículos de ruedas de radio control para usarlos como bases de robots. La principal desventaja de las ruedas es su empleo en terreno irregular, en el que se comportan bastante mal. Normalmente un vehículo de ruedas podrá sobrepasar un obstáculo que tenga una altura no superior al radio de sus ruedas [13].

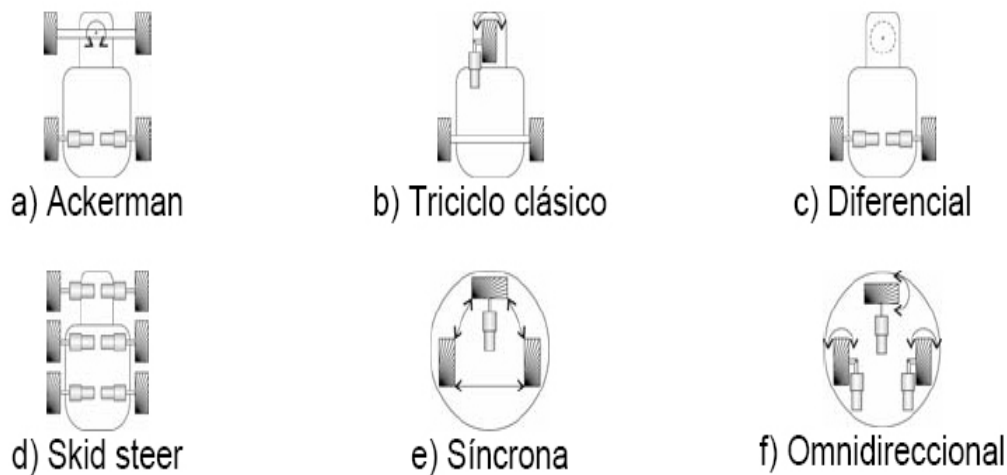


Figura 2.2. Clasificación de los robots móviles con ruedas (RMR).

B) ROBOTS MÓVILES CON PATAS. Estos robots móviles en lugar de utilizar ruedas utilizan patas. El problema principal de este tipo es la estabilidad, lo que implica una mayor complejidad en su control lo que implica también un mayor consumo energía ya que tiene que oponerse a la fuerza de gravedad. Dentro de los Robots móviles con patas destacan los bípedos, cuadrúpedos, hexápodos, octópodos, entre otros, los cuales son mostrados en la figura 2.2 [13].

Percepción

La percepción es un proceso que permite al robot o sistema, a través de los sentidos (sensores), recibir, elaborar e interpretar la información proveniente de su entorno. A través de los sensores se obtiene la magnitud de cada una de las variables físicas que intervienen en el contexto del sistema, dicha información es interpretada y procesada por medio de un algoritmo de control el cual genera la retroalimentación del sistema.

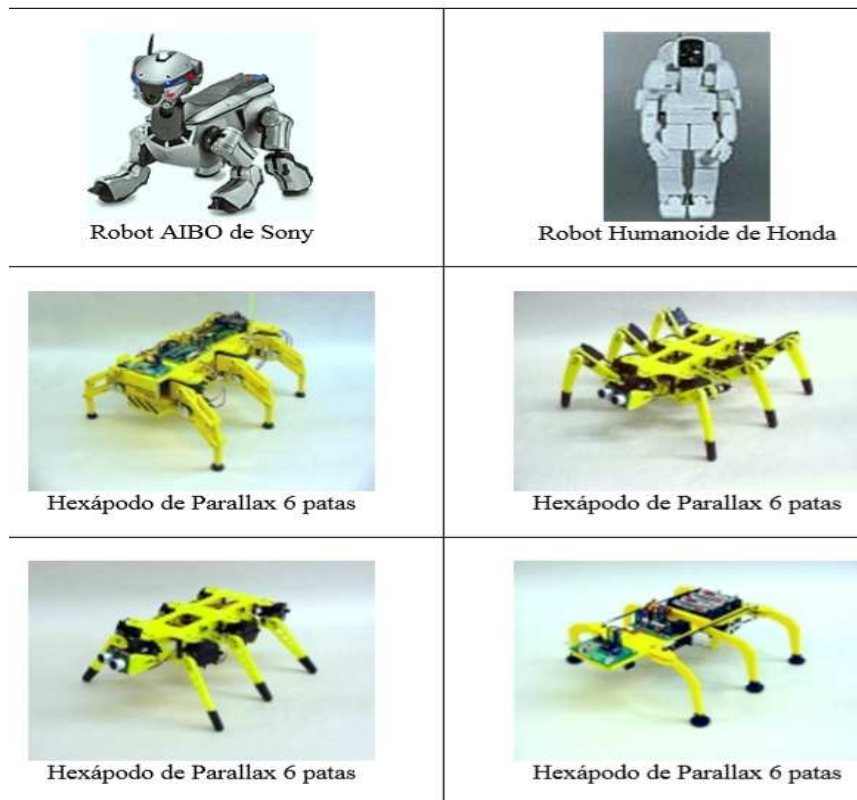


Figura 2.3. Clasificación de los robots móviles con ruedas (RMR).

Razonamiento

El razonamiento es el conjunto de actividades mentales que consiste en la conexión de ideas de acuerdo a ciertas reglas. En el caso del razonamiento en robótica, se trata de la capacidad para razonar con señales lógicas, estableciendo entre ellos algoritmos de control que tengan respuesta aplicada con su entorno.

Comunicación

La comunicación es un proceso importante en la interacción de dos o más sistemas. Para que exista comunicación, se requieren varios elementos; entre ellos están: transmisor o condiciones físicas (envía la información), receptor o sensor (recibe la información), canal o interfaz (medio por el cual se realiza el envío de la información) y mensaje o bits (información o datos enviados). En la figura 2.4 se observa la interfaz o canal de comunicación de una PC con un automóvil el cual retorna señales de información capturadas por los sensores sobre su entorno físico, mismas que a su vez son procesadas a través de un algoritmo de control implementado desde la PC.

La comunicación puede darse de forma alámbrica ó inalámbrica, siendo esta última la más utilizada recientemente por las ventajas tangibles que representa con respecto de la alámbrica.

A) *COMUNICACIÓN INALÁMBRICA*. En las últimas décadas la aparición de la comunicación inalámbrica entre dispositivos se ha impuesto con el uso de teléfonos móviles, computadoras portátiles, PDA's y en general en cualquier dispositivo electrónico. Han aparecido diversos tipos de comunicación como Wi-Fi, Bluetooth o infrarrojos, que aunque hasta el momento no ofrecen prestaciones tan altas como la comunicación por cable, no cabe duda que algún día lo harán. En la figura 2.5 se muestran los principales protocolos de comunicación inalámbrica en su frecuencia y alcance respectivamente.

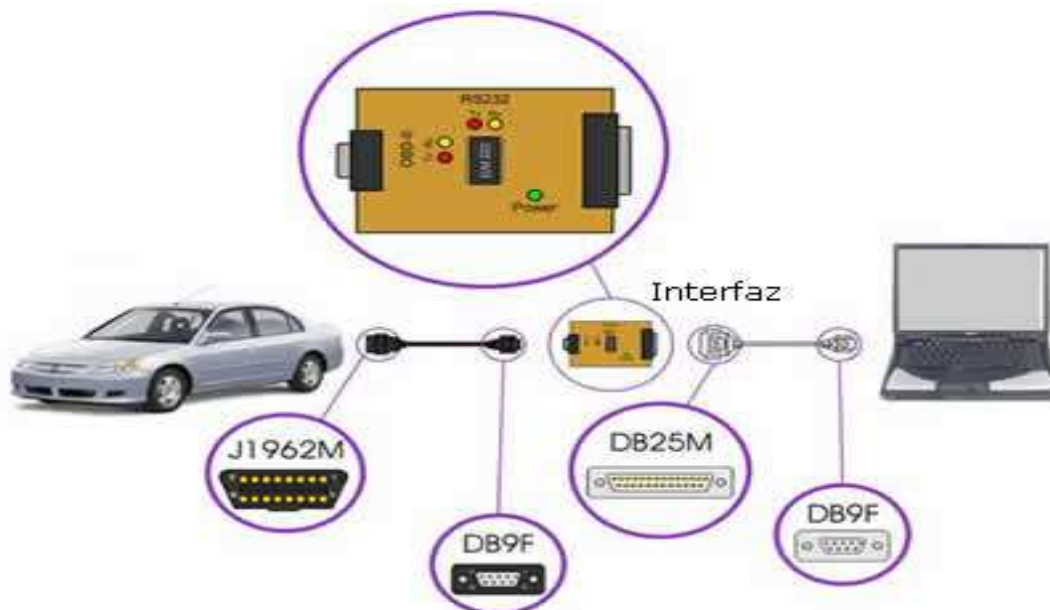


Figura 2.4. Conexión de un PC con un auto a través de una interfaz.

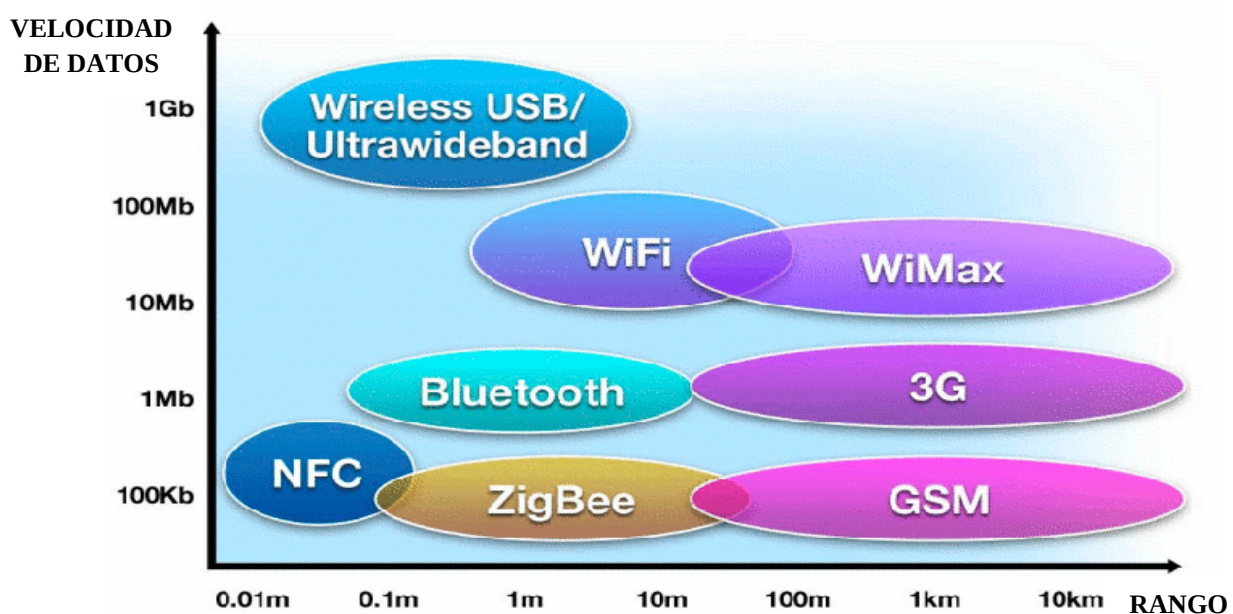


Figura 2.5. Protocolos de comunicación inalámbrica.

2.2 CARACTERÍSTICAS GENERALES DE UN RMR TIPO ACKERMAN

A continuación se describen las características descriptivas para el modelo del RMR, tipo carro o Ackerman de forma general, especificando paso a paso las partes que lo conforman [13].

Ruedas

Al desplazarse el robot móvil, las ruedas giran produciendo un contacto superficial (o fricción) con el suelo, idealmente, se desplaza $2\pi r$ sobre el eje “y” por cada vuelta como se ilustra en la figura 2.6.

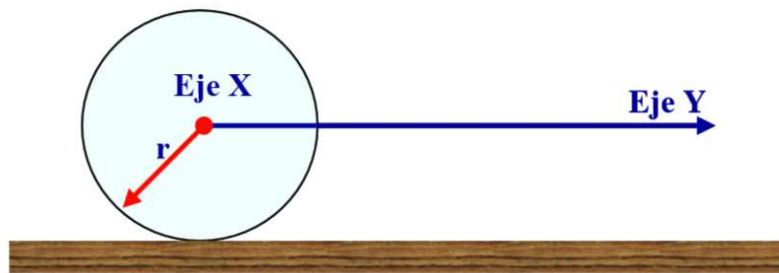


Figura 2.6. Desplazamiento de la rueda.

Dentro de las suposiciones para aproximar la rueda ideal se citan las siguientes:

1. El robot se compone de partes rígidas.
2. No hay deslizamiento en la dirección perpendicular a la rodadura.
3. No hay deslizamiento translacional entre la rueda y el suelo (rodadura pura).
4. Como máximo hay un eje de dirección por rueda.
5. Los ejes de la dirección son perpendiculares al suelo.

Giro

Un robot móvil puede contar con varias ruedas, y existe un punto alrededor del cual cada rueda sigue una trayectoria circular cuyo punto de intersección se llama Centro de Curvatura Instantáneo (CCI) mostrado en la figura 2.7a; en el caso de una trayectoria recta el CCI está en infinito mostrado en la figura 2.7b.

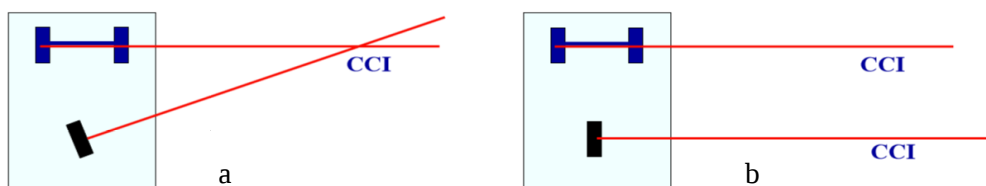


Figura 2.7. a) CCI ubicado en la intersección de los ejes de las ruedas, debido a trayectoria circular; b) CCI ubicado en el infinito, debido a trayectoria recta.

Arreglos de ruedas

Las formas más conocidas de disponer las ruedas en los robots móviles son: diferencial, síncrono, triciclo y carro:

A) *DIFERENCIAL*. Este es uno de los esquemas más sencillos. Básicamente consiste de dos ruedas en un eje común, donde cada rueda se controla independientemente cuyos movimientos son: Línea Recta, en arco y vuelta sobre su propio eje. En la figura 2.8 se muestra el esquema de ruedas y su armado.

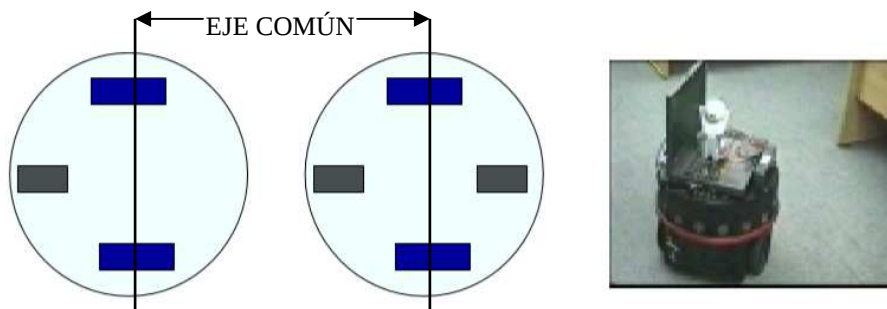


Figura 2.8. Arreglo Diferencial.

B) *SÍNCRONO*. Las ruedas se mueven en forma síncrona, es decir, al mismo instante. El movimiento síncrono es un caso particular del diferencial, donde cada eje se mueve en forma dependiente para dar vuelta y avanzar. Las ruedas están ligadas de forma tal que siempre apuntan en la misma dirección una de la otra y para dar vuelta giran las ruedas sobre el eje vertical descrito en la figura 2.9, por lo que la dirección de la estructura se mantiene y se requiere de un mecanismo adicional para mantener el frente del chasis (estructura del robot) en la dirección de las ruedas (torreta).

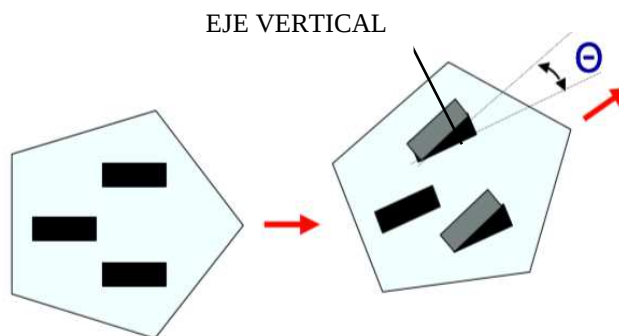


Figura 2.9. Arreglo Síncrono.

C) *TRICICLO*. Los triciclos tienen dos ruedas fijas que le dan tracción, además cuentan con una rueda para la dirección que normalmente no tiene tracción. Estos sistemas tienen buena estabilidad y simplicidad mecánica, tiene facilidad para ir recto y su cinemática es más compleja. Este sistema es ilustrado en la figura 2.10.

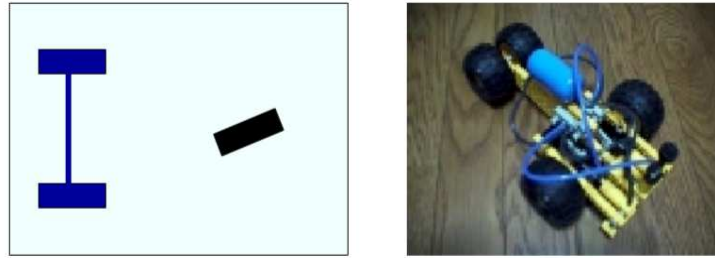


Figura 2.10. Arreglo Triciclo.

D) *CARRO*. Los sistemas de carro son similares al triciclo solo que cuentan con dos ruedas de tracción y dos ruedas para dirección. Tiene una mayor complejidad mecánica que el triciclo por el acoplamiento entre las 2 ruedas de dirección. Sus principales ventajas son buena estabilidad y facilidad de ir derecho. La desventaja es su complejidad cinemática. El arreglo tipo carro, también conocido como tipo Ackerman se ilustra en la figura 2.11.

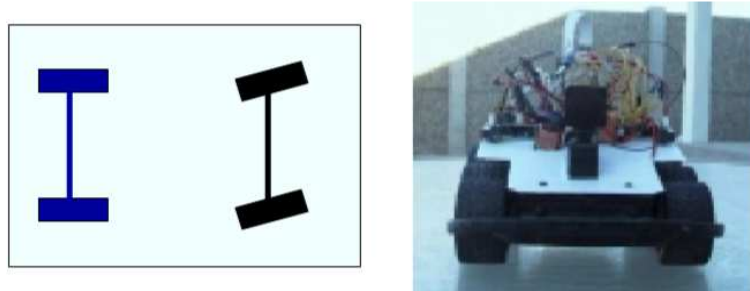


Figura 2.11. Arreglo Ackerman.

Restricciones cinemáticas

Se tienen dos tipos de restricciones que tienen que ver con el acoplamiento de los robots las cuales se conocen como:

A) *HOLONÓMICAS*. Los diferentes grados de libertad están desacoplados. Los robots diferenciales y síncronos: se puede desacoplar la posición de orientación (rotando sobre su eje), ver figura 2.12a.

B) *NO HOLONÓMICAS*. Los grados de libertad están acoplados, por ejemplo los triciclos y carros: para dar vuelta debe moverse hacia el frente o hacia atrás, es más complejo llegar a la posición final deseada, ver figura 2.12b.

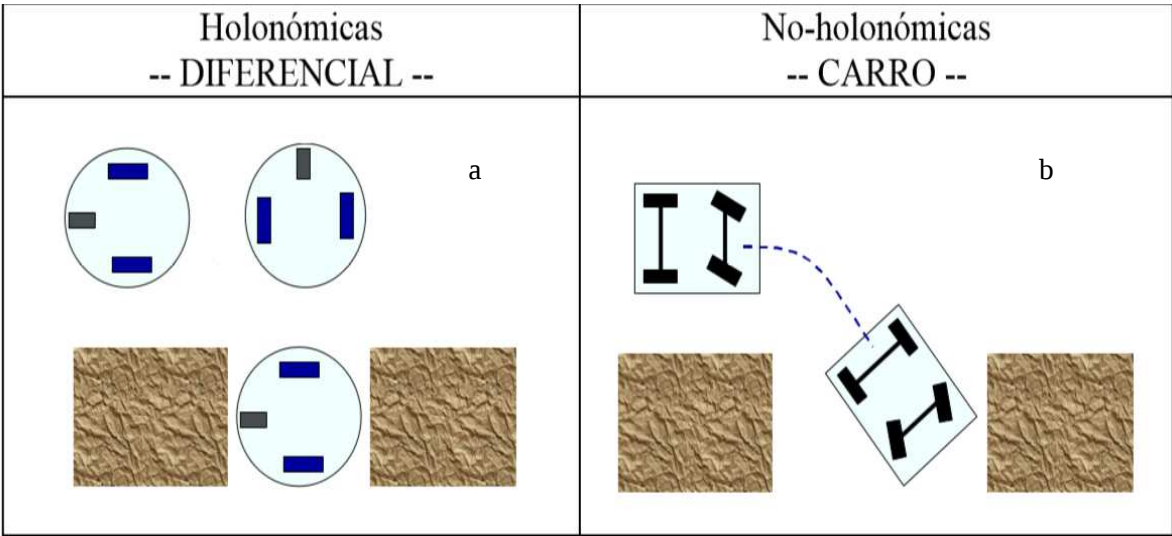


Figura 2.12. a) Restricciones holónicas, b) Restricciones no holónicas.

Forma del robot

La forma del robot móvil tiene un fuerte impacto en su facilidad de navegación, en particular con obstáculos y pasillos angostos por ejemplo:

- A) *ROBOT CILÍNDRICO*. Es más fácil navegar por la simetría del robot (Figura 2.13a).
- B) *ROBOT CUADRADO*. Es más complejo navegar, depende de la orientación del robot (figura 2.13b).

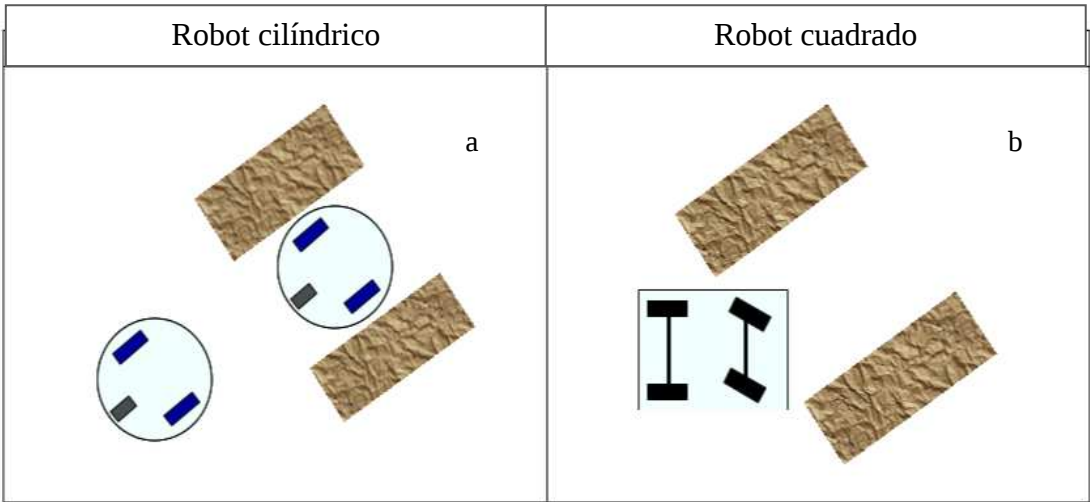


Figura 2.13. a) Robot cilíndrico, b) Robot cuadrado.

CAPÍTULO

3 DESCRIPCIÓN DE LA PLATAFORMA EXPERIMENTAL

3.1 INFRAESTRUCTURA DISPONIBLE

La plataforma experimental se basa en un carro de radio control HUMMER H2, mostrado en la figura 3.1. Este modelo es una reproducción a escala 1:24, el cual es utilizado como la planta a controlar y cuenta con un mando a distancia ó radiocontrol, medidas aproximadas 19 x 7.5 x 8.2cm., funciones completas de movimiento basadas en 4 ruedas tipo transporte (con opción a cambiar por 4 ruedas tipo orugas), luces y un rango de funcionamiento de 5-8m. (establecido por el alcance transmisor-receptor).

Estas características son las comerciales, es decir, las que presenta el fabricante, sin embargo, se tuvieron que hacer algunas modificaciones que permitieran trabajar con la planta y lograr los objetivos planteados. Lo cual se describe más adelante.

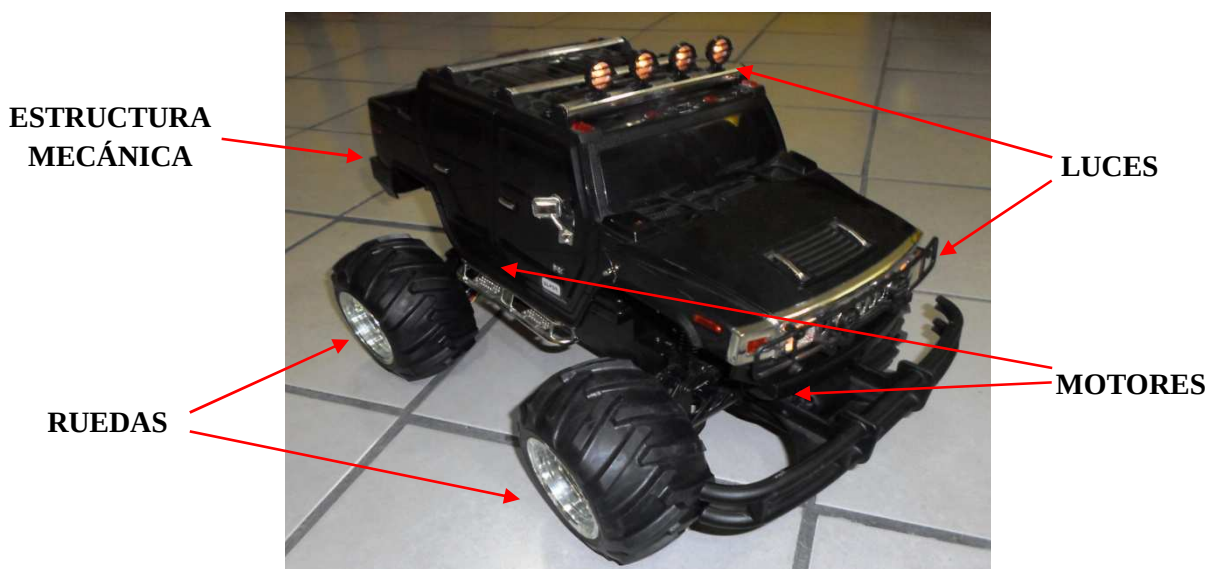


Figura 3.1. Vehículo de radio control HUMMER H2.

Inicialmente también se contó con 2 tarjetas XBIB-U-DEV y 2 módulos XBEE PRO de la marca DIGI (ilustrados en la figura 3.2), los cuales fueron utilizados para implementar la comunicación inalámbrica.

A partir de esta infraestructura disponible, se diseñó y se implementó paso a paso el control del robot móvil HUMMER H2.



Figura 3.2. Módulos de desarrollo 2.4 MHZ Digi-Mesh.

El software utilizado en el procesamiento de señales de control del robot móvil HUMMER H2, es soportado por tres plataformas de programación: XBEE, Arduino y LABVIEW, las cuales inicialmente se instalaron en una computadora personal que forma parte de la infraestructura disponible.

3.2 PLATAFORMA EXPERIMENTAL HUMMER H2

A continuación se describen las características del RMR, HUMMER H2 tipo Ackerman, especificando paso a paso las partes que conforman la estructura mecánica. La estructura del robot móvil HUMMER H2 tiene la configuración tipo Ackerman y está conformada por las partes siguientes:

A) **CARCASA**. Situada en la parte superior del vehículo con luces delanteras y traseras adaptadas como indicadores de control. Su peso es de 460 gramos y es de material de plástico (figura 3.3).



Figura 3.3. Carcasa del robot móvil HUMMER H2.

B) MUELLE DELANTERO. Sujeta las 2 ruedas delanteras, de dirección; tiene un acoplamiento para el eje transversal del vehículo y dentro de su estructura se encuentra el servomotor que se encarga de la dirección del vehículo. Su peso es de 600 gramos y se ilustra en la figura 3.4.

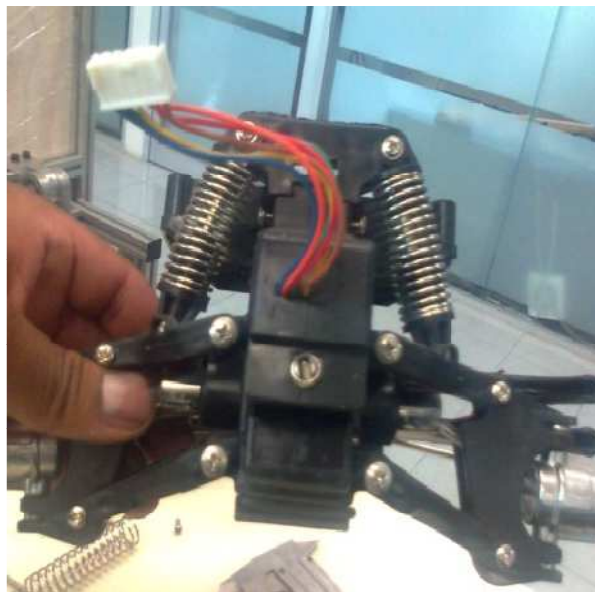


Figura 3.4. Muelle delantero del robot móvil HUMMER H2.

C) MUELLE TRASERO. Sujeta las 2 ruedas traseras, encargadas de la tracción del vehículo. Tiene un acoplamiento para el eje transversal del vehículo y dentro de su estructura se encuentra el motor de corriente directa que genera la tracción del vehículo. Su peso es de 594 gramos (sin considerar el peso del motor) y se ilustra en la figura 3.5.



Figura 3.5. Muelle trasero del robot móvil HUMMER H2.

D) PORTA BATERÍA. Compartimiento dedicado para alojar una batería tipo recargable Ni-mh con dimensiones 13 x 4.8 x 2.5 cm. Su peso es de 236 gramos (sin considerar el peso de la batería) y se ilustra en la figura 3.6.



Figura 3.6. Porta batería del robot móvil HUMMER H2.

E) EJE TRANSVERSAL. Pieza metálica que une al muelle trasero con el muelle delantero, atravesando longitudinalmente el porta batería. Su peso es de 8 gramos y se ilustra en la figura 3.7.



Figura 3.7. Eje transversal del robot móvil HUMMER H2.

F) *LLANTAS*. Posee 4 llantas tipo estándar de hule, cada llanta tiene un diámetro de 0.1305 metros y un perímetro de 0.41 metros. Su peso individual es de 250 gramos y se ilustran en la figura 3.8.



Figura 3.8. Llantas tipo estándar del robot móvil HUMMER H2.

G) *TORNILLOS*. Cada una de las piezas que conforma la estructura mecánica del vehículo esta unida mediante tornillos de diferentes medidas. El peso total de todos los tornillos requeridos para el ajuste de los componentes del vehículo es de 150 gramos y se ilustran en la figura 3.9.



Figura 3.9. Tornillos del robot móvil HUMMER H2.

Actuadores

El robot móvil HUMMER H2, posee 2 motores actuadores, uno se encarga de dirigir la dirección y el otro se encarga de generar la tracción del vehículo.

A) *SERVOMOTOR DE DIRECCIÓN*. Está formado por una caja de engranes metálicos los cuales proporcionan un torque máximo de 15 kilogramos a la salida del servomotor. Su peso es de 250 gramos y se ilustra en la figura 3.10.



Figura 3.10. Servomotor de dirección del robot móvil HUMMER H2.

B) MOTOR DE TRACCIÓN. Es un motor de corriente directa de 12 volts, genera hasta 6000 rpm., produciendo un elevado torque, debido al sistema de engranaje al que se encuentra adaptado. Su peso es de 450 gramos y se ilustra en la figura 3.11.



Figura 3.11. Motor de tracción del robot móvil HUMMER H2.

3.3 INSTRUMENTACIÓN

Sensores

Para obtener la medición de las variables físicas (posición, orientación y dirección) del robot móvil HUMMER H2, necesarias para conocer el comportamiento del vehículo en base a sus modelos cinemático y dinámico, se cuenta con tres sensores los cuales nos proporcionan los datos necesarios para el procesamiento de señales.

A) ENCODER EN CUADRATURA. El encoder es un sensor que mide la velocidad del vehículo a partir de un conteo de pulsos, además de proporcionar la posición del vehículo, también nos otorga la información del sentido de giro del motor de tracción. Tiene una resolución de 80 pulsos/revolución y un peso de 50 gramos, ver figura 3.12.



Figura 3.12. Encoder utilizado como sensor de la posición del robot móvil HUMMER H2.

B) MAGNETÓMETRO O COMPÁS ELECTRÓNICO Este elemento proporciona la orientación del vehículo en unidades de grados, siendo su referencia el norte magnético de la tierra. El protocolo de comunicación con el que trabaja es I2C. El magnetómetro utilizado para este fin es el HMC5883L, el cual posee un peso de 20 gramos y tiene una resolución de $\pm 2^\circ$, ver figura 3.13.

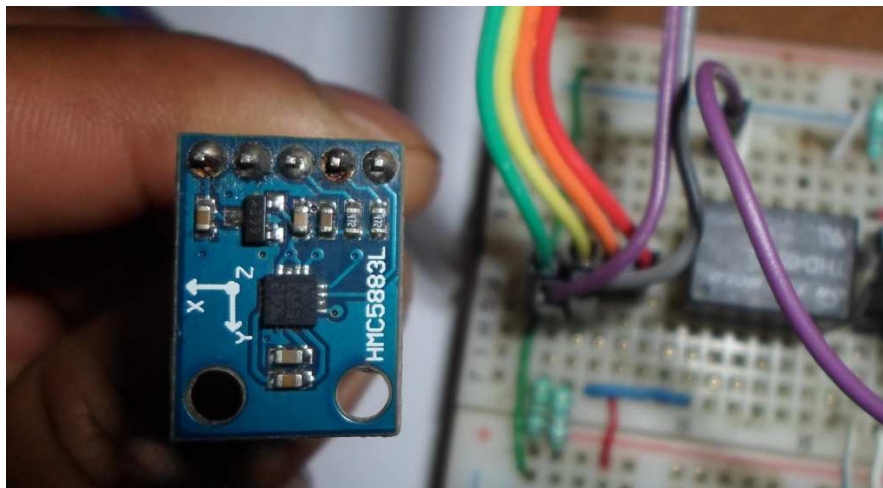


Figura 3.13. Magnetómetro utilizado como sensor de la orientación del robot móvil HUMMER H2.

C) POTENCIÓMETRO. Se encuentra empotrado dentro de la caja de engranes del servomotor de dirección, que forma parte de la infraestructura disponible inicial. Nos otorga una salida variable de voltaje que al procesar se transforma en la señal de información de dirección de las ruedas delanteras del vehículo. Su peso es de 25 gramos y se ilustra en la figura 3.14.



Figura 3.14. Potenciómetro utilizado como sensor de la dirección del robot móvil HUMMER H2.

Batería

Batería recargable sellada, de ácido-plomo, de 12 Volts, 4 Amperes/hora y hasta 1000 ciclos de carga/descarga. No tiene efecto de memoria y se coloca en la parte trasera del vehículo. Posee un peso de 1.603 kilogramos y sus dimensiones son 10.1 x 9 x 7cm. ver figura 3.15.



Figura 3.15. Batería de alimentación del robot móvil HUMMER H2.

Aislador de campo magnético

Para evitar que el campo magnético producido por la inducción de los motores afecte la lectura del compás electrónico, se utiliza un tubo de PVC de 1 pulgada de diámetro y una longitud de 15 cm., para aislar al sensor del resto de la estructura del vehículo. Su peso es de 80 gramos y se ilustra en la figura 3.16.



Figura 3.16. Aislador de campo magnético del magnetómetro.

Vehículo

A continuación se muestra en la figura 3.17 la vista general del robot móvil HUMMER H2 con todos los elementos mencionados. Su peso total, considerando todas las partes que lo componen es de 5.526 kg., el cual se resume en la tabla 3.1.



Figura 3.17. Robot móvil HUMMER H2.

Componentes del vehículo HUMMER H2	
Elemento	Peso (Kilogramos)
CARCASA	0.460
MUELLE DELANTERO	0.600
NUELLE TRASERO	0.594
PORTA BATERIA	0.236
EJE TRANSVERSAL	0.8
LLANTAS (4)	1
TORNILLOS	0.150
SEERVOMOTOR DE DIRECCIÓN	0.250
MOTOR DE TRACCIÓN	0.450
ENCODER	0.50
MAGNETÓMETRO	0.20
POTENCIOMETRO	0.25
BATERIA	1.603
AILADOR DE CAMPO MAGNÉTICO	0.80
PESO TOTAL	5.526

Tabla 3.1. Peso de los componentes del robot móvil HUMMER H2.

CAPÍTULO

4 MODELADO MATEMÁTICO DEL VEHÍCULO HUMMER H2

4.1 CARACTERIZACIÓN DEL VEHÍCULO

Una vez instrumentada la plataforma del vehículo HUMMER H2, se caracteriza en base a sus parámetros intrínsecos físicos, los cuales son necesarios para poder modelarlo matemáticamente.

CARACTERISTICAS

A) El vehículo con tracción en las ruedas traseras, posee una masa de aproximadamente 5.504 kilogramos y está equipado con un motor de corriente directa (ver tabla 3.1).

B) Las ruedas de tracción son fijas y paralelas al eje longitudinal del vehículo.

C) La conducción del vehículo se realiza a través del eje de las ruedas delanteras que posee un ángulo máximo de giro de ≈ 0.5236 radianes (30°).

D) Los parámetros intrínsecos del vehículo HUMMER H2 se presentan en la tabla 4.1.

Los datos obtenidos de la tabla 4.1 son aproximados y dependientes de las medidas realizadas en el robot móvil HUMMER H2, relacionados con los datos de la tabla 3.1. Algunos datos (F_{XD} , F_{XT} , T_M y V_V/a_C) son calculados mediante la simulación y procesamiento del modelo dinámico del vehículo, realizada en el software de MATLAB.

Las dimensiones del robot móvil HUMMER H2 son: 51 centímetros de longitud, 32 centímetros de ancho y 32.5 centímetros de altura.

El tubo de PVC tiene una longitud de 15 centímetros, ya que a partir de esta altura el desempeño del sensor de orientación (magnetómetro) es eficiente.

Datos del vehículo HUMMER H2 analizado	
Masa	$m \approx 5.404\text{kg.}$
Distancia de las ruedas delanteras hasta el centro de masa	$d_D = 14\text{cm.}$
Distancia de las ruedas traseras hasta el centro de masa	$d_T = 14\text{cm.}$
Constante de adherencia del neumático delantero	$c_D = 0.5$
Constante de adherencia del neumático trasero	$c_T = 0.5$
Radio de curvatura	$r_C = 67\text{ cm.}$
Momento de inercia total alrededor del eje vertical z_L del vehículo	$J_Z = 5.607086\text{ Nm.}$
Angulo de deslizamiento lateral del chasis	$\beta = \text{variable}$
Velocidad de rotación en el centro de masa	$\dot{\theta} = \text{variable}$
Velocidad lineal en el centro de masa	$v = \text{variable}$
Fuerza de tracción de rueda delantera	$F_{XD} = 7.5\text{ N}$
Fuerza de tracción de rueda trasera	$F_{XT} = 7.5\text{ N}$
Angulo de direccionamiento de la rueda delantera interna	$\delta_D = 30^\circ$
Constante de tiempo del motor	$T_M = 1.5\text{ seg.}$
Constante de tiempo del vehículo	$T_V = 1\text{ seg.}$
Ganancia entre la fuerza necesaria para desplazar el vehículo y la fricción a que está sometido	$V_V/a_C = 4.1$
Fuerza centrífuga	$F_{CF} = \text{variable}$

Tabla 4.1. Parámetros intrínsecos del robot móvil HUMMER H2.

Caracterización del motor de tracción

En una primera etapa se realiza la caracterización del motor de corriente directa por voltaje, para tal propósito se fija el motor y un dinamómetro a una estructura fija y se conectan 2 multímetros, uno que mida el voltaje y otro que mida la corriente, como se muestra en la figura 4.1.

Posteriormente se aplica una tensión al motor de corriente directa con valores de voltaje desde 0 volts hasta 20.5 volts, con una variación de 0.5 volts. Una vez que se realiza esta serie de pasos se registran los datos de voltaje, corriente y torque resultantes en la tabla 4.2.

Con estos datos se obtiene la grafica de variación de voltaje contra variación del torque (transformado en Newton), mostrada en la figura 4.2.

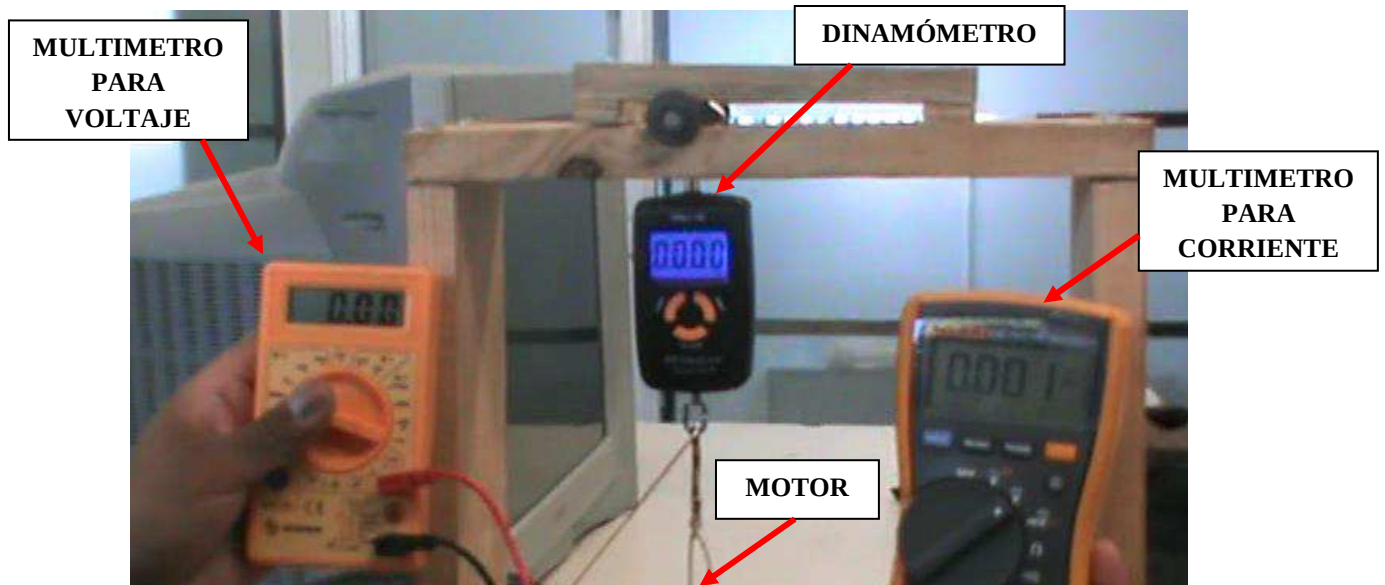


Figura 4.1. Caracterización del motor de tracción por voltaje.

CARACTERIZACION DEL MOTOR DC (Por voltaje)		
VOLTAJE(V)	CORRIENTE(A)	TORQUE(Kg)
0	0.3	0
0.5	0.6	0
1	0.9	0
1.5	0.11	0
2	0.15	0
2.5	0.21	0.0900
3	0.22	0.115
3.5	0.26	0.125
4	0.30	0.160
4.5	0.35	0.185
5	0.41	0.220
5.5	0.45	0.240
6	0.49	0.260
6.5	0.54	0.285
7	0.68	0.345
7.5	0.75	0.375
8	0.80	0.400
8.5	0.88	0.415
9	0.91	0.450
9.5	0.88	0.470
10	0.93	0.495
10.5	0.99	0.520
11	1.03	0.540
11.5	1.07	0.555
12	1.11	0.570
12.5	1.15	0.655

VOLTAJE
SELECCIONADO
12V.

13	1.18	0.660
13.5	1.2	0.680
14	1.24	0.690
14.5	1.27	0.780
15	1.48	0.805
15.5	1.66	0.850
16	1.72	0.885
16.5	1.77	0.920
17	1.81	0.955
17.5	1.85	0.985
18	1.91	1
18.5	1.94	1.020
19	1.97	1.045
19.5	2.03	1.065
20	2.07	1.090
20.5	2.10	1.170

Tabla 4.2. Caracterización del motor de tracción por voltaje.

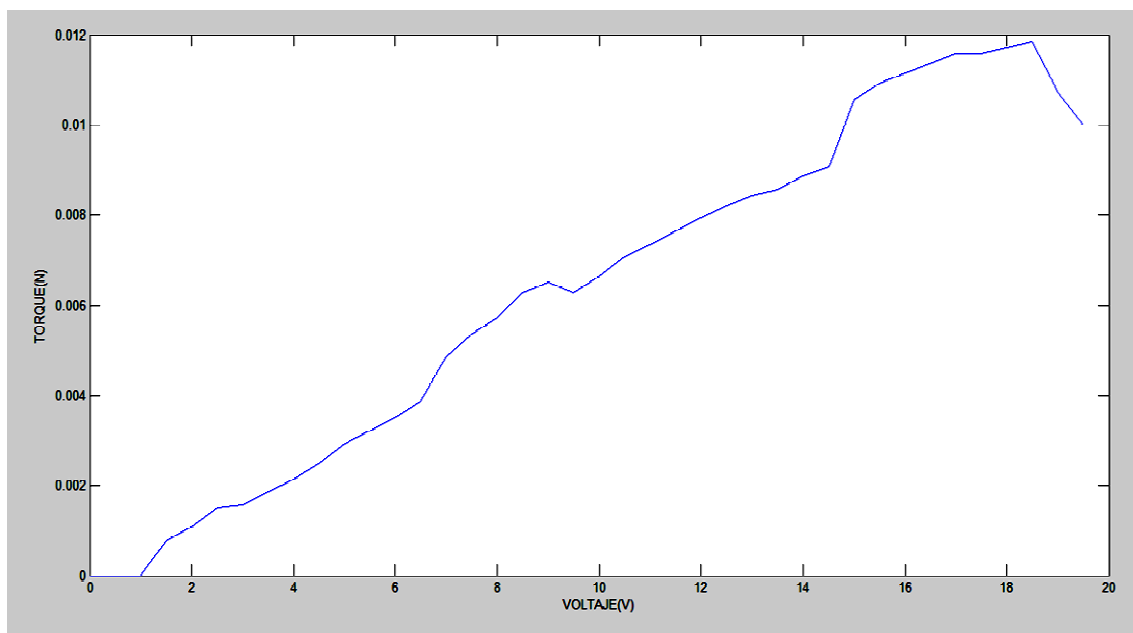


Figura 4.2. Respuesta de la caracterización del motor de tracción por voltaje.

La segunda parte de la caracterización del motor se realiza eligiendo la frecuencia que produce la mitad del torque máximo obtenido en la caracterización por voltaje, este valor es de 0.570 kilogramos con un voltaje de 12 volts y una corriente de 1.11 amperes, dichos valores se encuentran marcados en la tabla 4.2.

La frecuencia obtenida es de 480Hz., ya que el torque máximo tiene un valor de 0.570N, y con 480Hz., se obtiene 0.285N con un ciclo de trabajo del 50%, lo equivalente a la mitad del torque máximo.

Una vez que seleccionada la frecuencia de 480 Hz, se realiza la caracterización por frecuencia, que consiste en variar el ciclo de trabajo de 0% a 100% aumentando en cada paso 5% del ciclo de trabajo generado. Los resultados obtenidos se muestran en la tabla 4.3 y se grafican en la figura 4.3.

CARACTERIZACION DEL MOTOR DC (Por ciclo de trabajo)			
Ciclo Trabajo (%)	TORQUE(N)	Ciclo Trabajo (%)	TORQUE(N)
5	0.025	55	0.300
10	0.035	60	0.330
15	0.070	65	0.350
20	0.105	70	0.375
25	0.135	75	0.400
30	0.170	80	0.425
35	0.200	85	0.450
40	0.225	90	0.480
45	0.260	95	0.505
50	0.285	99	0.490

Tabla 4.3. Caracterización del motor de tracción por ciclo de trabajo.

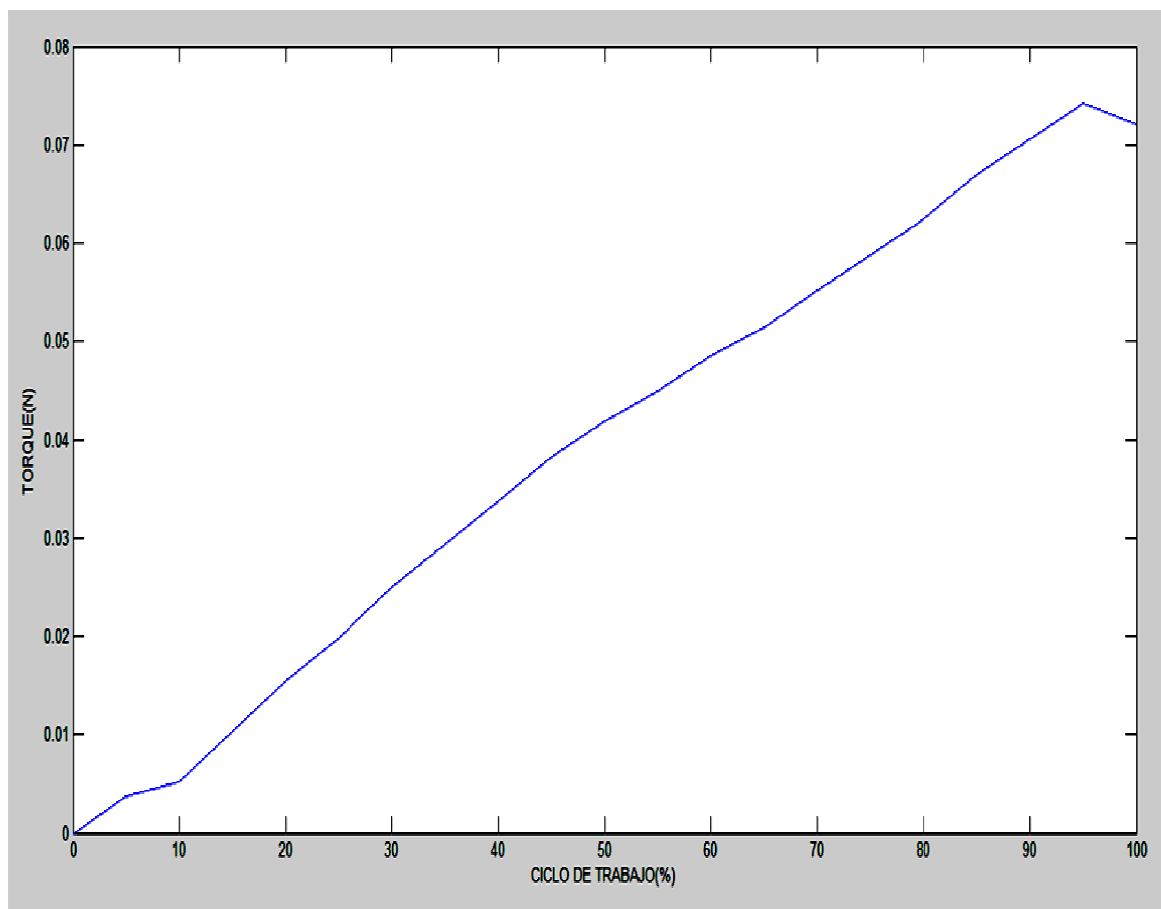


Figura 4.3. Respuesta de la caracterización del motor de tracción por ciclo de trabajo a una frecuencia de 480Hz.

Se observa que la respuesta del motor de tracción de la figura 4.3 es lineal en la frecuencia de 480 Hz, lo cual nos garantiza que el motor de tracción es adecuado para aplicar el control a través de una tarjeta Arduino UNO, la cual nos proporciona su salida PWM a una frecuencia de 490 Hz, con lo que obtendremos una respuesta lineal en el desempeño del motor y un control eficiente, eliminando los sobre impulsos o respuestas no deseadas.

4.2 MODELOS MATEMÁTICOS

La representación matemática de una planta física, que describe su comportamiento mientras el tiempo evoluciona, se representa mediante sus modelos matemáticos: cinemático y dinámico. El modelo cinemático considera el movimiento del robot móvil utilizando las coordenadas de su posición y orientación pero sin tomar en cuenta las causas que originan dicho movimiento. El modelo dinámico describe la evolución en el tiempo del robot móvil en relación con las causas y fuerzas que provocan los cambios de estado de movimiento.

Modelo cinemático

Para la obtención del modelo cinemático, se hicieron las siguientes consideraciones: Dado que un RMR es un robot que debe moverse sobre una superficie mediante la acción de ruedas montadas en él, se asumen lo siguiente:

- El robot móvil se mueve sobre una superficie plana horizontal, es decir la energía potencial es constante.
- Los ejes de guiado son perpendiculares al suelo.
- No existen elementos flexibles en la estructura del robot, incluyendo las ruedas.
- El contacto entre cada rueda y el suelo se reduce a un solo punto.
- No existe deslizamiento.

De acuerdo con estas hipótesis se analiza el caso de movimiento bidimensional del RMR, donde el problema se reduce a encontrar (x, y, Θ) , ver figura 4.4 [14].

Las componentes (x, y) corresponden a la translación, la componente (Θ) a la orientación del RMR y la componente (Ψ) es el ángulo de giro de las ruedas directrices como se aprecia en la figura 4.4.

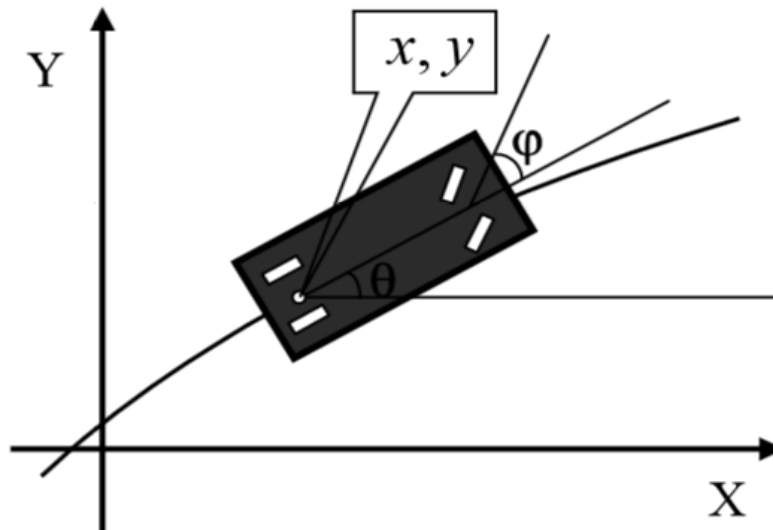


Figura 4.4. Representación geométrica de la postura de un robot móvil tipo Ackerman.

Se analiza la dirección de las 2 ruedas delanteras, especificando que el ángulo de la rueda interior (Θ_i) es mayor que el ángulo de la rueda exterior (Θ_o), para evitar el deslizamiento como se muestra en la figura 4.5. Donde se representan los siguientes parámetros:

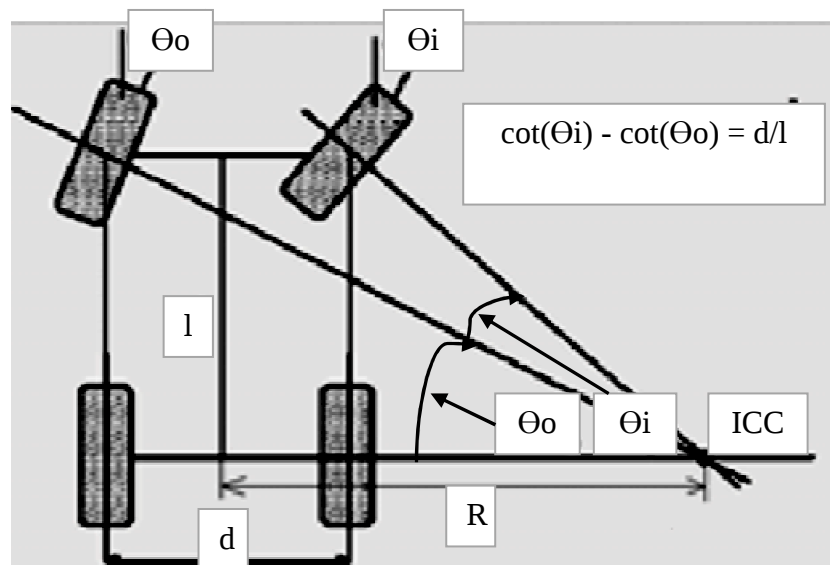


Figura 4.5. Análisis Geométrico del RMR tipo carro.

d = Separación lateral de las ruedas.

l = Separación longitudinal de las ruedas.

Θ_i = Ángulo relativo de dirección de la rueda interior.

Θ_o = Ángulo relativo de dirección de la rueda exterior.

R = Distancia del ICC (CENTRO DE CURVATURA INSTANTANEO) a la línea central del vehículo.

Analizando la representación geométrica del robot móvil tipo Ackerman en la figura 4.4, la ubicación del robot dentro del plano cartesiano puede ser determinada por el vector de la ecuación (4.1) [13] y [14]:

$$\mathbf{x} = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix} \quad (4.1)$$

Mientras la matriz de rotación define la orientación del sistema local (x, y) con respecto al sistema de referencia global (X, Y), en la figura 4.4, tomando en cuenta que los ejes Z de ambos sistemas son paralelos, está dada por la ecuación (4.2) [13]:

$$R(\theta) = \begin{bmatrix} \cos(\theta) & \sin(\theta) & 0 \\ -\sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (4.2)$$

El movimiento de un robot móvil Ackerman esta descrito por el siguiente sistema de ecuaciones (4.3) a (4.6) [13] y [14]:

$$\dot{x} = v \cos(\theta) \quad (4.3)$$

$$\dot{y} = v \sin(\theta) \quad (4.4)$$

$$\dot{\theta} = \frac{v}{l} \tan(\Psi) \quad (4.5)$$

$$\dot{\Psi} = w \quad (4.6)$$

Y la restricción no holónoma por la ecuación (4.7) [13]:

$$\dot{x} \sin(\theta) - \dot{y} \cos(\theta) = 0 \quad (4.7)$$

Donde las entradas de control son:

“v” representa la velocidad del vehículo (ruedas traseras, de tracción).

“w” representa la velocidad angular de las ruedas de dirección.

Modelo dinámico

La mecánica del robot móvil HUMMER H2 presenta una estructura cerrada, donde los grados de libertad de la configuración Ackerman se encuentran acoplados, por este motivo no es posible aplicar la metodología de Euler Lagrange para desarrollar el modelo dinámico, ya que solo se aplica para modelos que poseen una mecánica abierta con grados de libertad desacoplados o independientes [15].

Bajo estas circunstancias se establece como metodología de desarrollo el análisis de las fuerzas que intervienen en cada eje del sistema coordenado, para obtener el modelo dinámico que represente el comportamiento del robot móvil HUMMER H2.

El diagrama de la figura 4.6 representa el análisis de fuerzas que simplifica el modelo tipo Ackerman en el modelo tipo bicicleta para facilitar el desarrollo matemático [16], [17], [18] y [19].

Los valores de los parámetros se describen en la tabla 4.1.

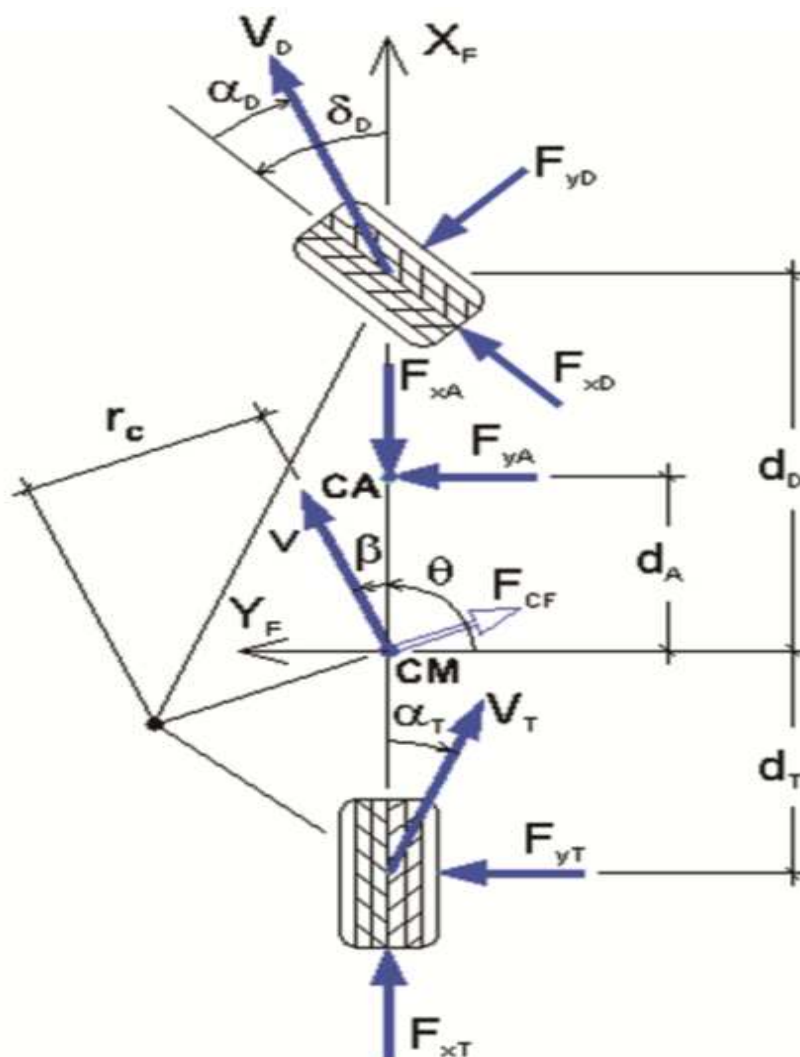


Figura 4.6. Diagrama de fuerzas del vehículo HUMMER H2.

Para el análisis de las dinámicas del vehículo se consideran la masa, la carga y la potencia necesaria para moverlo.

Se ha obtenido el modelo dinámico de un robot móvil del tipo Ackerman, el cual es similar al de un coche convencional, a través de las Leyes de Newton y considerando que el vehículo se mueve en una superficie plana. El modelo comprende los movimientos longitudinal y lateral que son utilizados en el control de velocidad longitudinal y lateral, respectivamente.

Debido a que las variaciones de desplazamiento longitudinal son más lentas que las de desplazamiento lateral, se utiliza una estructura de control descentralizada.

El modelo dinámico se obtiene analizando las fuerzas que ocasionan desplazamiento lateral y longitudinal en el plano $x_f - y_f$ del sistema de coordenadas fijo en el vehículo y movimientos de rotación en torno del eje z_f .

Sin embargo para el control de seguimiento de trayectoria, si se consideran algunas simplificaciones: se asume que el CM (Centro de Masa) está al nivel del suelo, las influencias aerodinámicas están concentradas en un punto CA (Centro Aerodinámico), los movimientos de suspensión son despreciables debido al peso mínimo del robot móvil y a que se fijan los amortiguadores del robot móvil para evitar lectura errónea de información por parte del sensor de orientación.

El peso es igualmente distribuido a lo largo del chasis y las ruedas de un mismo eje son representadas por una rueda posicionada en el centro de este.

Por lo tanto los movimientos de rotación alrededor de los ejes $x_f - y_f$, pueden ser despreciados.

Una vez que se han planteado estas simplificaciones nos llevan al modelo bicicleta bien conocido de un vehículo tipo Ackerman presentado en la figura 4.6.

El análisis del diagrama de fuerzas presentado en la figura 4.6 conduce a las ecuaciones (4.8) a (4.10), las cuales relacionan las componentes de fuerza en los ejes x_f e y_f y los pares alrededor del eje z_f .

*Equilibrio de fuerzas actuantes en el eje x_f :

$$F_{CF}\text{sen}(\beta) + F_{XT} - F_{XA} + F_{XD}\cos(\delta_D) - F_{YD}\text{sen}(\delta_D) = m\dot{v}\cos(\beta) \quad (4.8)$$

*Equilibrio de fuerzas actuantes en el eje y_f :

$$-F_{CF}\cos(\beta) + F_{YT} + F_{YA} + F_{XD}\sin(\delta_D) + F_{YD}\cos(\delta_D) = m\dot{v}\sin(\beta) \quad (4.9)$$

*Equilibrio de fuerzas actuantes en el eje z_f :

$$(F_{YD} \cos(\delta_D) + F_{XD}\sin(\delta_D)) d_D - F_{YT}d_T + F_{YA}d_A = J_z\ddot{\theta} \quad (4.10)$$

Donde:

“ β ” es el ángulo de deslizamiento lateral del chasis.

“ $\ddot{\theta}$ ” es la aceleración de rotación del vehículo en el centro de masa (CM).

“ v ” es la velocidad lineal del vehículo en el centro de masa (CM).

“ α_D ” y “ α_T ” son los ángulos del vector de desplazamiento lateral de la rueda delantera y trasera respectivamente.

“ m ” es la masa total del vehículo.

“ F_{CF} ” es la fuerza centrífuga.

“ F_{XD} ” y “ F_{XT} ” son las fuerzas de tracción en las ruedas delantera y trasera.

“ F_{YD} ” y “ F_{YT} ” son las fuerzas de adherencia en los neumáticos delantero y trasero.

“ F_{XA} ” y “ F_{YA} ” son las fuerzas aerodinámicas.

Los términos de las ecuaciones (4.8), (4.9) y (4.10), se detallan considerando las características de cada fuerza como sigue:

FUERZA CENTRÍFUGA: La fuerza centrífuga puede ser descompuesta en un producto entre la masa del cuerpo y la aceleración centrífuga, dada por la ecuación (4.11):

$$F_{CF} = ma_{CF} \quad (4.11)$$

La aceleración centrífuga escrita en términos de la velocidad lineal del vehículo y del radio de curvatura al cual el sistema está sujeto, se describe en la ecuación (4.12):

$$a_{CF} = v^2/r_C \quad (4.12)$$

Sin embargo, con el objetivo de obtener las ecuaciones dinámicas en función de las variables de interés, se reescribe el radio de curvatura a través de las variables v , $\dot{\beta}$ y $\dot{\theta}$, quedando de la forma expresada en la ecuación (4.13):

$$r_C = \frac{v}{\dot{\beta} + \dot{\theta}} \quad (4.13)$$

De esta manera la fuerza centrífuga se escribe como se expresa en la ecuación (4.14):

$$F_{CF} = mv(\dot{\beta} + \dot{\theta}) \quad (4.14)$$

FUERZA AERODINÁMICA: La fuerza aerodinámica está compuesta por los componentes F_{XA} en el eje x_f y F_{YA} en el eje y_f , y se define por la ecuación (4.15):

$$F_{(x,y)A} = c_a A_C v^2 \frac{r_C}{2} \quad (4.15)$$

Donde c_a es la constante de penetración en el medio (aire) y A_C es el área de contacto del vehículo con el medio. Debe resaltarse que esta fuerza resulta muy difícil de determinar, debido a las variaciones de dirección y velocidad de los vientos.

FUERZA DE ADHERENCIA DE LOS NEUMÁTICOS: Las fuerzas de adherencia lateral en las ruedas, contrarias al deslizamiento del neumático, son determinadas a través del producto entre la constante de adherencia del neumático con el suelo (c_D y c_T) y el ángulo del vector de desplazamiento (α_D y α_T). Así la fuerza de adherencia sobre el neumático delantero está dada por la ecuación (4.16):

$$F_{YD} = c_D \alpha_D = c_D (-\beta + \delta_D - d_D \frac{\dot{\theta}}{v}) \quad (4.16)$$

Y para la rueda trasera está dada por la ecuación (4.17):

$$F_{YT} = c_T \alpha_T = c_T (-\beta + d_T \frac{\dot{\theta}}{v}) \quad (4.17)$$

Basados en las ecuaciones (4.8), (4.9) y (4.10), y en el desarrollo de las fuerzas presentado previamente se obtiene un nuevo conjunto de ecuaciones (4.18), (4.19) y (4.20), describiendo el comportamiento dinámico del vehículo con respecto a las variables deseadas (v , $\dot{\beta}$ y $\dot{\theta}$):

$$\dot{\beta} = \dot{\theta} \left(\frac{c_T d_T - c_D d_D \cos(\delta_D)}{mv^2 \cos(\beta)} - 1 \right) + \frac{c_a A_C v r_C}{2m \cos(\beta)} - \beta \left(\frac{c_T + c_D \cos(\delta_D)}{mv \cos(\beta)} \right) - \frac{\dot{v} \tan(\beta)}{v} + \frac{F_{XD} \sin(\delta_D)}{mv \cos(\beta)} + \frac{c_D \delta_D \cos(\delta_D)}{mv \cos(\beta)} \quad (4.18)$$

$$\ddot{\theta} = \frac{\beta}{J_Z} (c_T d_T - c_D d_D \cos(\delta_D)) + \frac{c_a A_C v^2 r_C d_A}{2J_Z} + \frac{F_{XD} d_D \sin(\delta_D)}{J_Z} + \frac{c_D d_D \delta_D \cos(\delta_D)}{J_Z} - \frac{\dot{\theta}}{J_Z v} (c_T d_T^2 + c_D d_D^2 \cos(\delta_D)) \quad (4.19)$$

$$\dot{v} = v \tan(\beta) (\dot{\theta} + \dot{\beta}) + \frac{F_{XT}}{m \cos(\beta)} - \frac{c_a A_C v^2 r_C}{2 m \cos(\beta)} + \frac{F_{XD} d_D \cos(\delta_D)}{m \cos(\beta)} - \frac{c_D \sin(\delta_D)}{m \cos(\beta)} (-\beta + \delta_D - \frac{d_D \dot{\theta}}{v}) \quad (4.20)$$

Linealización del modelo dinámico

Dado que un controlador predictivo no lineal tendría un elevado coste computacional cuando se calcula la ley de control, se linealiza el modelo dinámico para su aplicación en el Control Predictivo Basado en Modelo lineal (CPBM).

El modelo lineal se obtiene en dos etapas. Primero, resolviendo la expresión de las dinámicas del ángulo de deslizamiento lateral del chasis (β), y de la velocidad angular ($\dot{\theta}$), considerando la velocidad lineal constante [19].

Así, en una segunda etapa se identifica un modelo linealizado de la velocidad del vehículo.

Este procedimiento es necesario porque la dinámica de velocidad es significativamente más lenta que los otros elementos del modelo dinámico. En otras palabras, el periodo de muestreo para el bucle de control de v (T_V), es más grande que para el bucle de β y $\dot{\theta}$ ($T_{\beta, \dot{\theta}}$), y durante $T_V / (T_{\beta, \dot{\theta}})$, periodos de muestreo la velocidad v puede ser considerada constante.

Para las expresiones dinámicas (4.18) y (4.19) las variaciones de los ángulos β y δ_D pueden ser consideradas pequeñas en cada periodo de muestreo. Por lo tanto, pueden considerarse aproximaciones lineales expandiendo $\cos(\beta)$, $\sin(\beta)$, $\cos(\delta_D)$, $\sin(\delta_D)$ usando series de Taylor y truncándolas después del primer término: $\cos(\beta) \approx 1$, $\sin(\beta) \approx \beta$, $\cos(\delta_D) \approx 1$ y $\sin(\delta_D) \approx \delta_D$.

Para el robot móvil HUMMER H2 utilizado en esta tesis, las fuerzas aerodinámicas son muy pequeñas y pueden ser despreciadas. Por ello las expresiones linealizadas para β y $\dot{\theta}$ conducen a las ecuaciones (4.21) y (4.22):

$$\dot{\beta} = \dot{\theta} \left(\frac{c_T d_T - c_D d_D}{mv^2} - 1 \right) - \beta \left(\frac{c_T + c_D}{mv} \right) + \frac{F_{XD} \delta_D}{mv} + \frac{c_D \delta_D}{mv} \quad (4.21)$$

$$\ddot{\theta} = \frac{\beta}{J_Z} (c_T d_T - c_D d_D) + \frac{c_D d_D \delta_D}{J_Z} - \frac{\dot{\theta}}{J_Z v} (c_T d_T^2 + c_D d_D^2) + \frac{F_{XD} d_D \delta_D}{J_Z} \quad (4.22)$$

Con el objetivo de obtener un movimiento sin deslizamiento lateral, β debe tener el mismo signo que el ángulo de dirección δ_D .

El modelo dinámico lineal de la velocidad se define considerando que, en cada periodo de muestreo, la velocidad angular y el ángulo de deslizamiento lateral del chasis son idénticos a los valores de referencia.

Como el vehículo de prueba tiene tracción solo en el eje trasero, la fuerza F_{XD} es despreciable y solo se considera la fuerza F_{XT} , la cual es proporcional al par del motor de corriente directa. Por ello la variable de entrada del modelo puede ser la aceleración angular de la rueda trasera $\dot{w}_{RT}$.

La ecuación diferencial obtenida mediante identificación, que relaciona la velocidad y la aceleración angular de la rueda trasera, teniendo en cuenta las dinámicas de velocidad y del motor de corriente directa es definida como un sistema lineal de segundo orden en la ecuación (4.23):

$$\ddot{V} = -\frac{(T_{Mcd} + T_V)}{T_{Mcd}T_V} \dot{V} - \frac{1}{T_{Mcd}T_V} V + \frac{V_V/a_C}{T_{Mcd}T_V} \dot{w}_{RT} \quad (4.23)$$

Donde:

T_{Mcd} y T_V son las constantes de tiempo del motor y del vehículo respectivamente, y V_V/a_C Representa la ganancia del sistema.

Modelo dinámico Linealizado representado en espacio de estados

Para la variable θ se desglosa las ecuaciones de estado siguientes (4.24), (4.25) y (4.26):

$$\theta = \theta_1 \quad (4.24)$$

$$\theta_2 = \dot{\theta}_1 \quad (4.25)$$

$$\dot{\theta}_2 = \ddot{\theta}_1 = \ddot{\theta} \quad (4.26)$$

A continuación se expresa su representación en espacio de estados en la ecuación (4.27).

$$\begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \end{bmatrix} = \begin{bmatrix} \frac{\beta}{J_z} (c_T d_T - c_D d_D) + \frac{c_D d_D \delta_D}{J_z} - \frac{\theta_2}{J_z v} (c_T d_T^2 + c_D d_D^2) + \frac{F_{XD} d_D \delta_D}{J_z} \\ \theta_2 \end{bmatrix} \quad (4.27)$$

Para la variable β se desglosa las ecuaciones de estado siguientes (4.28) y (4.29):

$$\beta = \beta_1 \quad (4.28)$$

$$\beta_2 = \dot{\beta}_1 = \dot{\beta} \quad (4.29)$$

A continuación se expresa su representación en espacio de estados en la ecuación (4.30).

$$[\dot{\beta}_1] = \left[\dot{\theta} * \left(\frac{c_T * d_T - c_D * d_D}{m * v^2} - 1 \right) - \beta * \left(\frac{c_T + c_D}{m * v} \right) + \frac{F_{XD} * \delta_D}{m * v} + \frac{c_D * \delta_D}{m * v} \right] \quad (4.30)$$

Se desarrolla la implementación del sistema en MATLAB, con los valores determinados en la tabla 4.1, dando como resultado la respuesta del modelo dinámico representado en las gráficas de la figura 4.7.

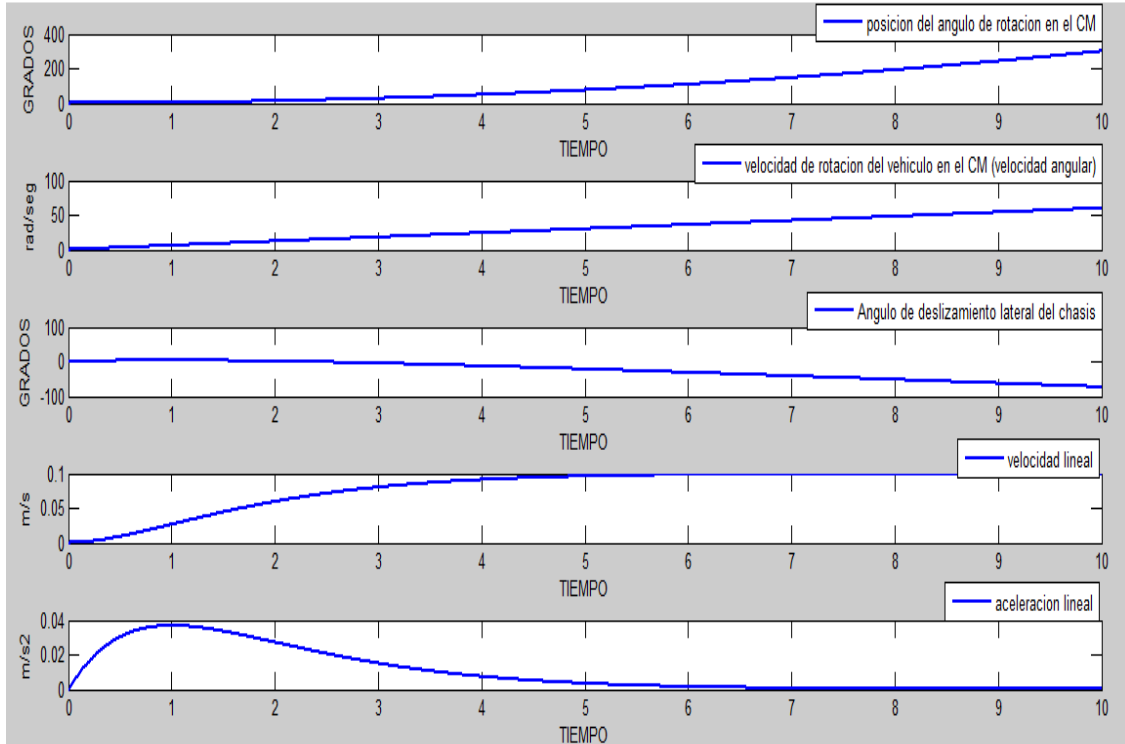


Figura 4.7. Graficas obtenidas con la representación del modelo dinámico en espacio de estados.

CAPÍTULO

5 ESTRATEGIAS DE CONTROL

5.1 CONTROL PREDICTIVO BASADO EN EL MODELO DINÁMICO

Representación del modelo dinámico en función de transferencia

La ecuación (4.20) se utiliza para representar el modelo dinámico de velocidad del vehículo, mientras las ecuaciones (4.18) y (4.19) son utilizadas, para el modelo dinámico del ángulo de deslizamiento lateral del chasis y para la dinámica de la velocidad angular, respectivamente [19].

Para el diseño del controlador de velocidad las otras dos variables se ignoran y para el controlador de las variables β y $\dot{\theta}$ se asume que la velocidad es constante durante un periodo de muestreo. La matriz de transferencia (ecuación 5.1) para el controlador de β y $\dot{\theta}$ se deriva de las ecuaciones (4.21) y (4.22), obteniendo la siguiente matriz:

$$G(s)_{\beta, \dot{\theta}} = \frac{1}{s^2 + (a+d)s + (ad-bc)} \begin{bmatrix} es + de + bf \\ fs + af + ce \end{bmatrix} \quad (5.1)$$

Donde se desglosan las siguientes ecuaciones (5.2), (5.3), (5.4), (5.5), (5.6), (5.7) y (5.8):

$$a = \frac{c_T + c_D}{mv} \quad (5.2)$$

$$b = \frac{c_T d_T - c_D d_D}{mv^2} \quad (5.3)$$

$$c = \frac{c_T d_T - c_D d_D}{J_z} \quad (5.4)$$

$$d = \frac{c_T d_T^2 + c_D d_D^2}{J_z v} \quad (5.5)$$

$$e = \frac{c_D + F_{XD}}{m \cdot v} \quad (5.6)$$

$$f = \frac{d_D (c_D + F_{XD})}{J_z} \quad (5.7)$$

La función de transferencia que relaciona la velocidad tangencial del vehículo con la aceleración angular de la rueda de tracción, obtenida de la ecuación (4.23), está dada por la ecuación (5.8):

$$G(s)_{vel} = \frac{V_V / a_c}{T_{Mcd} T_V s^2 + (T_{Mcd} + T_V) s + 1} \quad (5.8)$$

Propuesta del controlador predictivo MPC

El control predictivo es parte de las metodologías de control avanzado, que ha causado un gran impacto, esto se debe a que es una estrategia de optimización, la cual utiliza un modelo del proceso para obtener la señal de control minimizando la función objetivo.

El MPC o control predictivo basado en modelo presenta estabilidad y robustez, esta estrategia también es conocida como control por horizonte deslizante. El MPC es parte de los controladores óptimos, ya que las acciones responden a un criterio de optimización.

El criterio de optimización se denomina función de costo, y el valor futuro se predice por el modelo dinámico del sistema, el cual se denomina modelo de predicción, el intervalo de tiempo futuro que interviene en la optimización, es denominado horizonte de predicción.

Una de las características principales del MPC es la formulación abierta y flexible que permite la configuración de varios modelos de predicción, para sistemas lineales o no lineales, sistemas SISO, MISO, MIMO y es importante señalar la consideración de restricciones sobre las señales del sistema, esto genera que el control predictivo sea una de las pocas técnicas que permite controlar sistemas, asociando restricciones como parte del diseño del controlador. Para sistemas lineales, en la ausencia de restricciones, el controlador es una ley de control lineal; mientras que cuando las restricciones están presentes aparece un problema de programación cuadrática.

Los controladores predictivos tienen elementos comunes y se pueden elegir diversas opciones, y a diferentes casos dan distintos algoritmos, pero se establece un criterio de optimización definido por [20] y [21]:

- *Modelo de predicción
- *Función Objetivo
- *Obtención de la ley de control

MODELO DE PREDICCIÓN. Es el modelo matemático que describe el comportamiento de la planta a controlarse, la representación y las características del modelo pueden ser lineales, no lineales, en tiempo continuo, o en tiempo discreto, en Variables de Estado o en funciones de entrada y salida. El diseño del modelo debe capturar la máxima dinámica del sistema, para permitir el cálculo de las predicciones y el análisis teórico.

El problema de optimización implicado se resuelve mediante el computador, así como la técnica de horizonte deslizante con la que se aplica la solución, considerando estos dos procesos hace que la mejor opción sea utilizar modelos discretos que modelos continuos.

FUNCIÓN OBJETIVO. Conocida como función de costo, es aquella que indica el criterio a optimizar. Es una función definida positiva que expresa el costo asociado a la evolución del proceso en el intervalo del horizonte predicción N , donde posee la forma expresada en la ecuación (5.9):

$$j(N_1, N_2, N_U) = \sum_{j=1}^{N_2} \delta(j) [\hat{y}(t+j|t) - w(t+j)]^2 + \sum_{j=1}^{N_U} \lambda(j) [\Delta u(t+j-1)]^2 \quad (5.9)$$

Donde los parámetros N_1 y N_2 son los horizontes mínimos y máximos de predicción respectivamente y N_U es el horizonte de control, de esta forma se marcan los límites donde la salida sigue a la referencia. Los coeficientes $\delta(j)$ y $\lambda(j)$ son secuencias que ponderan el comportamiento futuro.

Trayectoria de referencia. Si se conoce aproximadamente la evolución futura de la referencia, el sistema puede reaccionar antes de actuar evitando los efectos de retardo del proceso esto sucede en aplicaciones determinadas como en robótica o servos.

En otras aplicaciones cuando la referencia es constante se puede conseguir una pequeña mejora, y conociendo el instante donde el valor de la respuesta cambia y adelantando la acción de control a esta circunstancia.

Restricciones. Todos los procesos están sujetos a restricciones, los actuadores tienen un campo limitado de acción determinado con cierta velocidad de respuesta, estas restricciones se imponen por razones constructivas, de seguridad o de medioambiente o por los alcances de los sensores. Estas restricciones pueden causar límites en las variables de estado, por esta razón se hace necesaria la introducción de otras funciones a minimizar.

El éxito de la industria, en la implementación del MPC radica en la consideración de las restricciones, ya que se considera los límites en la amplitud, en la velocidad de cambio de la señal de control y los límites en la salida.

OBTENCIÓN DE LA LEY DE CONTROL. Para obtener los valores de la señal de control es necesario minimizar la función de costo J (Ecuación 5.9), pero antes se debe obtener primero las salidas predichas en función de los valores actuales y pasados, tanto de la entrada como de la salida, de igual forma las señales futuras de control, mediante el uso del modelo escogido y reemplazando en la función de costo, obteniendo una expresión que al minimizar da como resultado los valores buscados.

Esta estructura de la ley de control se basa en el uso del concepto del horizonte de control N_U , que consiste en considerar tras un cierto intervalo de tiempo $N_U < N_2$, donde no hay variación en las señales de control.

El desempeño del control predictivo se puede comparar con el de otros controladores si se tiene en cuenta sus ventajas y desventajas.

Las principales **ventajas** que se pueden destacar:

- *Cuando se conoce el comportamiento futuro del sistema de control, el sistema puede reaccionar antes que el cambio se haya presentado en la planta, evitando de este modo retardos en la respuesta del proceso.

- *Es una estrategia de control, que maneja sistemas de varias entradas y salidas, también sistemas lineales y no lineales.

- *Es una estrategia flexible, puesto que su implementación se ha realizado en la industria y puede seguir avanzando en forma intuitiva, a medida que se da el sustento teórico para esta técnica de control.

- *Se toman las restricciones físicas de actuadores, las restricciones ambientales, restricciones de seguridad como parte del diseño del algoritmo de control.

Como **desventajas** se pueden tomar en cuenta:

- *Los algoritmos de control del MPC, requieren un nivel de procesamiento de datos relativamente rápido.

- *Es necesario tener un modelo preciso para obtener una alta fiabilidad del sistema de control, es así que el rendimiento del MPC estará determinado por la relación cercana que exista entre el planta real y el modelo matemático.

- *El proceso de optimización del MPC requiere un nivel matemático complejo, lo cual genera una carga de computación alta.

Control predictivo de LABVIEW

La herramienta computacional LABVIEW en el toolkit Control Design and Simulation, presenta un módulo denominado Control Predictive.

Para construir el controlador MPC, es necesario proveer la siguiente información: Horizonte de Predicción (N_p), es el número de muestras en el futuro en la cual el controlador MPC predice la salida de la planta.

Horizonte de control (N_c), es el número de muestras dentro del horizonte de predicción durante el controlador, este valor puede afectar la acción de control.

En la Figura 5.1, se puede observar que el controlador MPC va a predecir las salidas durante el periodo $k+N_p$, también como se puede observar la acción de control a partir del horizonte de control $k+N_c$ no cambia. Es necesario tomar en cuenta:

- *El horizonte de control es menor que el horizonte de predicción.
- *Si el horizonte de predicción es corto se reduce la longitud de tiempo en la que el controlador MPC puede predecir la señal de salida de la planta. Un horizonte de predicción muy pequeño causa que el controlador MPC opere algo parecido al controlador con realimentación tradicional.
- *Un horizonte de predicción grande incrementa la capacidad de predecir al controlador MPC y observar una restricción potencial y evitarlo. Un horizonte de predicción muy grande decrementa el rendimiento del controlador MPC por el aumento de cálculos extras que realiza el algoritmo de control.
- *Si el horizonte control es corto, y la acción de control es grande podría pasarse del punto de referencia luego que el horizonte de control finalice. Sin embargo como el controlador sigue ejecutándose la salida de la planta va a seguir colocando cerca del punto de referencia.
- *Un horizonte de control grande produce cambios más agresivos en la acción de control. Esos cambios agresivos pueden resultar en oscilaciones o en energía innecesaria.

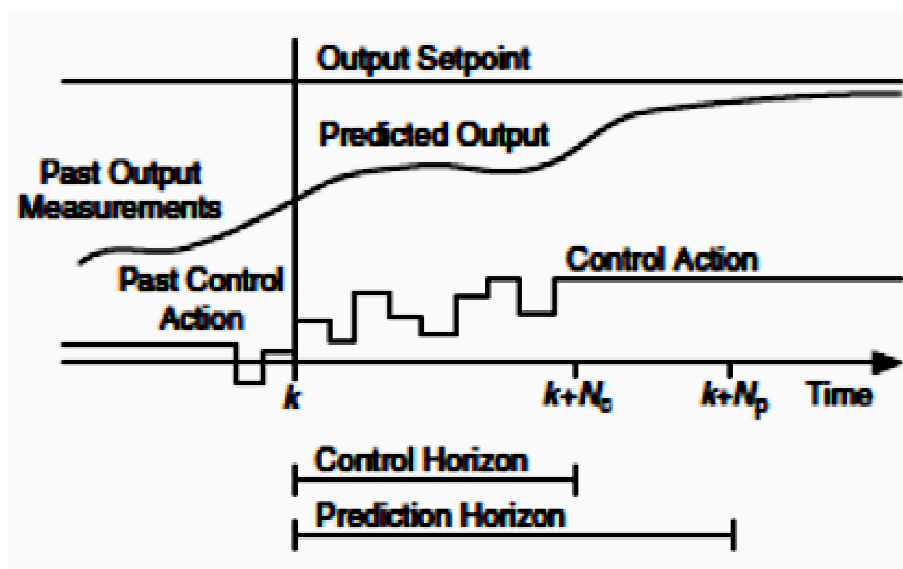


Figura 5.1. Horizontes de control y predicción.

Implementación del controlador MPC

Después de haber creado el controlador MPC, entonces se puede implementar el controlador, ya sea en una simulación o en un escenario en el mundo real [20] y [21].

Se implementa este controlador usando el CD IMPLEMENT MPC CONTROLLER VI con un Timed Loop o Control & Simulation Loop.

Para el diseño del controlador MPC se provee la siguiente información:

*Características de setpoint de entrada, setpoint de salida y/o perturbaciones.

*El valor de salida de la planta.

El CD IMPLEMENT MPC CONTROLLER VI entonces entrega la siguiente información:

*La acción de control que controla el cambio del setpoint de salida.

*La salida predicha de la planta a lo largo del horizonte de predicción.

*La velocidad de cambio en la acción de control.

Se puede proveer las características de setpoint y perturbación mientras avanza la ejecución del controlador o cuando el controlador se ejecuta.

Es necesario el conocimiento de los estados de la planta, para que el controlador MPC tenga un alto rendimiento ya que el modelo del sistema, debe ser lo más cercano a la realidad. Por esta razón el MPC al comienzo de cada muestra, el mismo controlador MPC de LABVIEW estima los estados de la planta actual.

El conocimiento exacto de los estados de la planta, mejora la exactitud de la predicción y a su vez el rendimiento del controlador.

Si todos los estados de las plantas son medibles, la estimación de estados es simple, ya que el controlador solamente requiere la consideración de los efectos de ruido.

En un caso real el trabajo de los estados de la planta pueden ser no medibles, y los estados deben ser estimados a partir de los valores actuales y desde medidas que se encuentren disponibles.

Programación del controlador MPC

En la implementación del MPC en tiempo real, los parámetros de control pueden ser modificados al instante en que la planta se encuentra en proceso de operación, ya que el modelo del sistema puede en algún punto de trabajo encontrarse desactualizado [21]. También se puede actualizar los parámetros de control del MPC que sean necesarios para una mejor sintonización, se puede cambiar los valores de las restricciones, el modelo en Espacio de Estados, los parámetros de control, los valores de ponderación.

En la Figura 5.2 se observa el proceso de programación para implementar el control en el punto de referencia deseado.

A partir de este diagrama de flujo se procede a detallar en secuencia cada VI que es parte de la implementación del MPC en LABVIEW.

1° En el CD CREATE MPC CONTROLLER se sitúan los parámetros del controlador MPC, el modelo en espacio de estados en tiempo discreto, las ponderaciones MPC; las restricciones del MPC, al cambiar estas variables se crea un nuevo perfil de control para la implementación del control predictivo.

2° El CD SET MPC CONTROLLER coloca el perfil del controlador MPC con las mismas variables de control que son repuesta del CD CREATE MPC CONTROLLER, este sub vi al dar clic en Reset configura nuevamente los datos del controlador MPC, ya sean todos los parámetros o uno por uno.

3° UPDATE MPC CONTROLLER se utiliza para la generación de perfiles de referencia.

4° IMPLEMENT MPC CONTROLLER recibe un perfil del controlador MPC, el perfil del punto de referencia y la señal correspondiente a la medida de la variable de salida $y[k]$, mientras que la salida de este sub vi indica la señal de control del conversor $u[k]$.

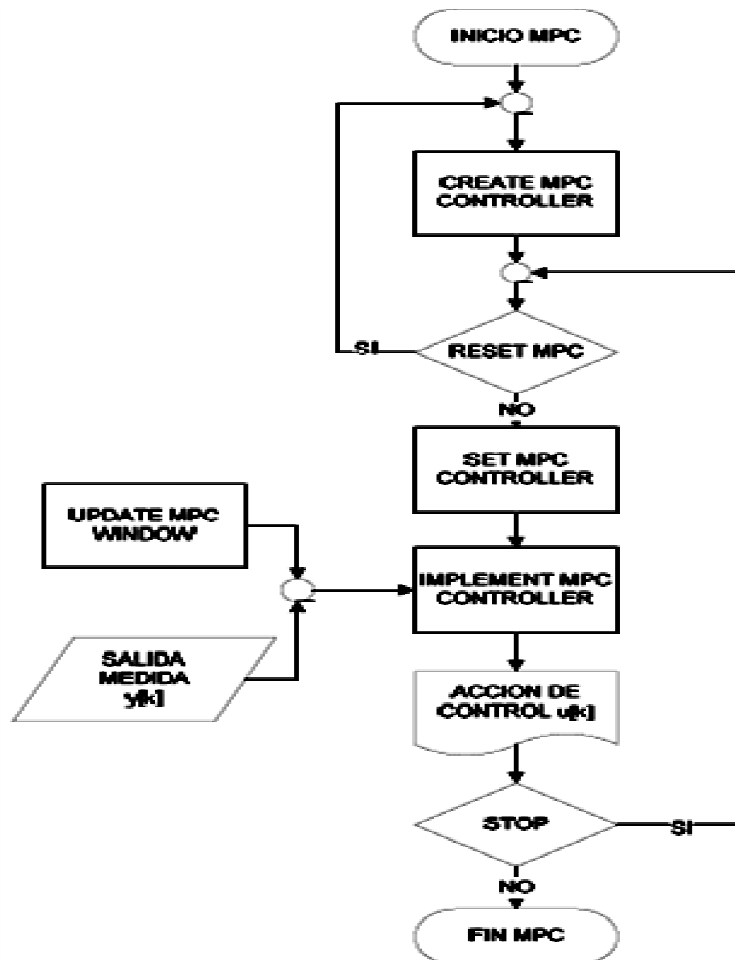


Figura 5.2. Algoritmo del controlador MPC.

Para la implementación del MPC, se utilizan los VI antes mencionados, la forma de conexiones de los bloques para la simulación se detalla en la figura 5.3.

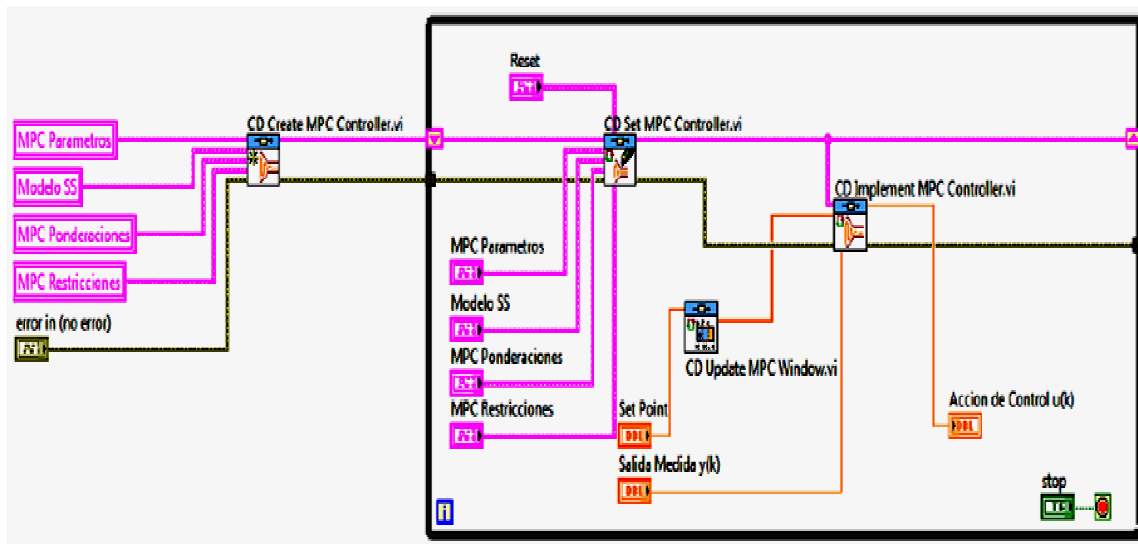


Figura 5.3. Implementación del controlador MPC en LABVIEW.

Para obtener la representación en espacio de estados del modelo dinámico del ángulo de dirección δ_D , se realiza el programa detallado en la figura 5.4, utilizando como complemento la estructura Mathscript, la cual tiene como principal característica la asociación con el entorno programación de MATLAB.

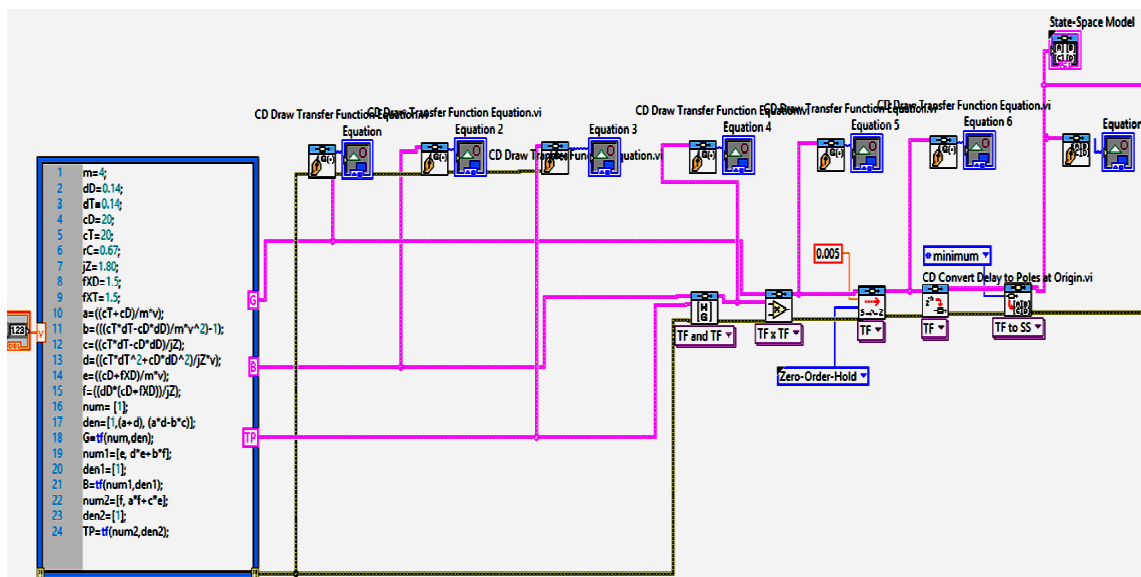


Figura 5.4. Implementación del modelo dinámico del ángulo de dirección (δ_D) en LABVIEW.

Desempeño del controlador predictivo MPC

Al ejecutar el algoritmo del modelo dinámico de dirección (δ_D) implementado en LABVIEW se obtiene la respuesta de la figura 5.5, donde se muestra el proceso que se aplica a la planta, pasando por un retenedor de orden cero hasta convertirla en su

respectiva representación en espacio de estados discreto, la cual es necesaria para enviarla al controlador predictivo.

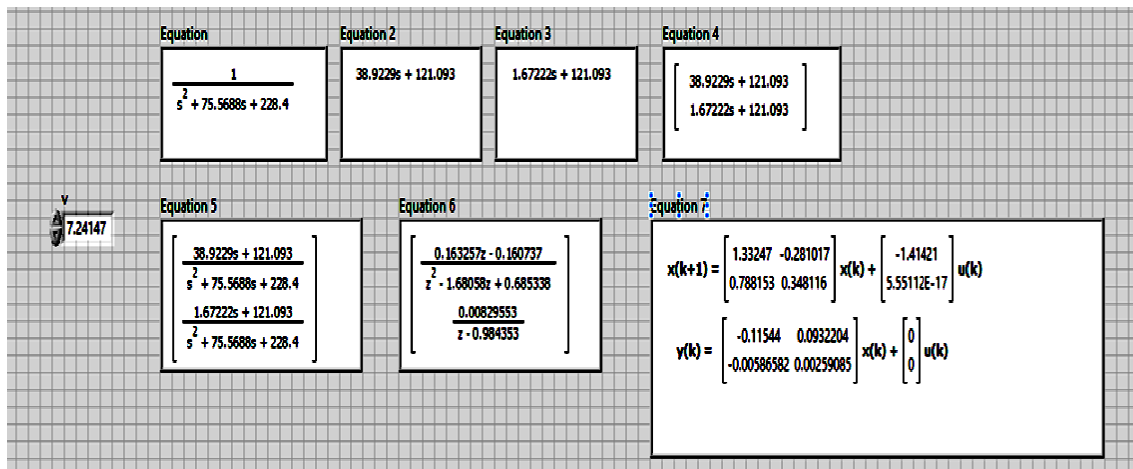


Figura 5.5. Implementación del controlador MPC en LABVIEW.

Una vez obtenido el modelo en espacios de estado de la planta se ingresa al controlador MPC, el cual nos proporciona una salida predicha $y(k)$, y una respuesta de control $u(k)$, la cual será la que proporcione el PWM, una vez normalizada.

Por otro lado el setpoint se utilizará para especificar el ángulo deseado en cada periodo de muestreo.

Otros datos a especificar dentro del controlador MPC, son las restricciones y los horizontes de control y predicción óptimos, así como las ponderaciones de los parámetros.

Un ejemplo es el desempeño del controlador de la dinámica del ángulo de dirección (δ_D), se presenta en la figura 5.6, con un setpoint =1, horizonte de predicción= 5 y horizonte de control = 1.

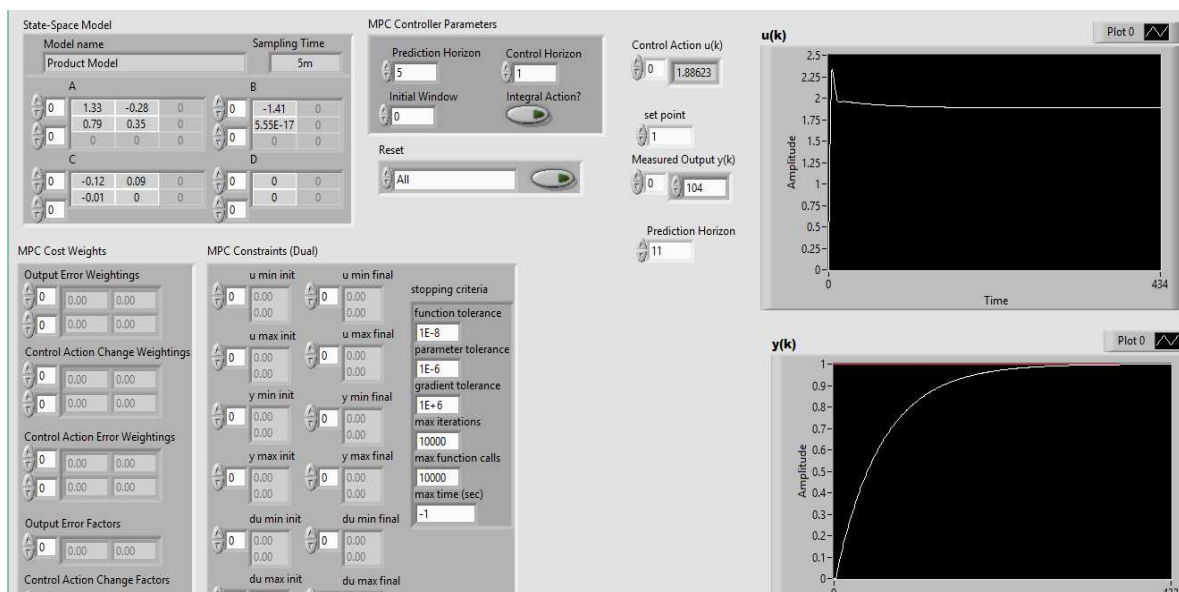


Figura 5.6. Desempeño del controlador predictivo utilizando la dinámica del ángulo de dirección (δ_D).

La simulación mostrada en la figura 5.6 retroalimenta la salida predicha $y(k)$, en el bloque CD implement MPC controller.VI, en el algoritmo en tiempo real la salida que se retroalimenta al bloque CD implement MPC.VI es la de la planta real en el instante k .

5.2 CONTROL PUNTO- PUNTO BASADO EN EL MODELO CINEMÁTICO

Generación de trayectorias

Generalmente cuando se realiza el control de seguimiento de trayectorias de robots móviles esta generalmente es conocida y específica, determinada por una ecuación matemática, sin embargo también es deseable que el móvil siga cualquier otra trayectoria como una definida por puntos en el plano. Se puede ver que existen dos formas básicas para especificar la trayectoria [13], [14], [22] y [23].

- * Suministrando puntos consecutivos e ignorando la trayectoria espacial que describe el robot entre cada dos puntos.
- * Especificando la trayectoria por una curva paramétrica en el tiempo, tal como una recta o una parábola, que debe describir el robot en el espacio de trabajo.

La primera alternativa, denominada tradicionalmente control punto a punto, sólo tiene interés práctico cuando los puntos están suficientemente separados, ya que en caso contrario, sería muy laborioso especificar todos los puntos intermedios.

Su contraparte es que los puntos tampoco puedan estar muy separados ya que existe un alto riesgo de que se generen movimientos imprevisibles o no controlados. En este control punto a punto y en particular para resolver este problema, se propone un algoritmo que en base al modelo cinemático del robot móvil HUMMER H2, calcula el error en cada periodo de muestreo, de esa forma es posible realizar el control del robot móvil para que pase por dichos puntos.

La segunda forma para realizar el seguimiento de trayectorias es utilizando curvas paramétricas en el tiempo como trayectoria deseada que se denomina tradicionalmente control de trayectoria continua, el control debe hacer que el robot móvil reproduzca lo más exacto posible la trayectoria especificada. Para la generación de trayectorias, en ambos casos, es necesario tener en cuenta lo siguiente:

- * Las restricciones cinemáticas y dinámicas involucradas.
- * Las trayectorias generadas deben ser suaves, lo que implica restricciones sobre las derivadas.
- * La generación de trayectorias deben ser computacionalmente eficiente.

Definición paramétrica de algunas curvas de interés

Se puede usar las coordenadas “x” y “y” para describir la ubicación del robot móvil en cualquier punto de su recorrido.

Sin embargo, las coordenadas no incidían en el recorrido cuando el móvil se encuentra en diferente ubicación, por tal motivo en ocasiones, dos variables no son suficientes para describir completamente una situación de grafica. Una alternativa de solución es usar ecuaciones paramétricas para describir las coordenadas “x” y “y” de un punto como funciones de una tercera variable, tiempo, que se denomina parámetro.

Por tanto una curva paramétrica es la que esta definida por ecuaciones paramétricas. A continuación se definen algunas de las curvas paramétricas en el tiempo más comunes en las ecuaciones (5.10), (5.11), (5.12), (5.13), (5.14), (5.15), (5.16) y (5.17).

$$y(t) = mx(t) \quad (5.10)$$

$$x(t) = A \sin(2\pi/P)t \quad (5.11)$$

$$y(t) = x^2(t) \quad (5.12)$$

$$x(t) = A \sin(2\pi/P)t \quad (5.13)$$

$$y(t) = r \cos(2\pi t/P + C) + k \quad (5.14)$$

$$x(t) = r \sin(2\pi t/P + C) + h \quad (5.15)$$

$$y(t) = r \cos(\pi t/P) \quad (5.16)$$

$$x(t) = r \sin(\pi t/P) \quad (5.17)$$

Las ecuaciones (5.10) y (5.11) representan a la recta donde m es la pendiente, A es la amplitud y P es el tiempo en el que la curva completa su recorrido. Un extremo de la recta es el punto (A, mA) y el otro es (-A, -mA).

Las ecuaciones (5.12) y (5.13), representan a la parábola con un extremo en el punto (A, A²) y otro en el punto (-A, A²) y P es el mismo que el usado en la ecuación (5.11).

Las ecuaciones (5.14) y (5.15), representan a la circunferencia con centro en la coordenada (h, k) y radio r. La constante C permite definir la posición de partida de la circunferencia y P es el mismo que el usado en la ecuación (5.11).

Las ecuaciones (5.16) y (5.17), representan una curva en forma de ocho que pasa por el origen y tiene un radio r y P es el mismo que el usado en la ecuación (5.11). En la Figura 5.7 se muestran todas las curvas mencionadas anteriormente [23].

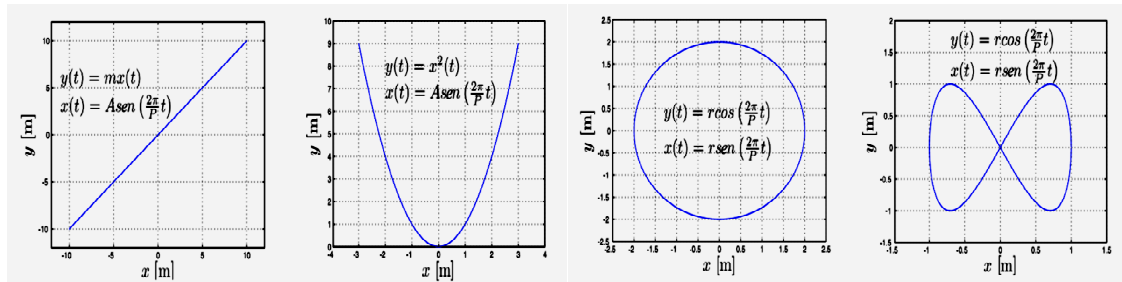


Figura 5.7. Curvas parametrizadas en el tiempo.

Algoritmo generador de trayectorias

OBJETIVO DEL GENERADOR DE TRAYECTORIAS. El generador de trayectorias es el encargado de generar los comandos de velocidad utilizados en el bucle de control de los motores que accionan el eje de tracción y el eje de dirección. La información de la que dispone para realizar su cometido son los puntos que definen los puntos de la trayectoria introducida por el usuario desde la interfaz de supervisión y la medida de la posición actual. Por lo tanto, el generador de trayectorias deberá de implementar un algoritmo que permita la siguiente transformación (5.18):

$$[V_T, V_D] = f(X_D, Y_D, X_A, Y_A) \quad (5.18)$$

Donde X_D, Y_D son las coordenadas de los vértices de la trayectoria deseada, X_A, Y_A la posición actual y V_T, V_D los comandos de velocidad a cada uno de los motores (tracción y dirección). El generador de trayectorias debe de ejecutarse en tiempo real.

IMPLEMENTACION. La implementación del generador de trayectorias se realizará en LABVIEW. El programa deberá de comunicarse con el lazo inferior de control (control de los motores) utilizando el puerto serie. Además, deberá de recibir los puntos de la trayectoria desde la interfaz de usuario. Esta comunicación se implementará mediante conexión inalámbrica. El algoritmo para generar los comandos se basa en el modelo cinemático del robot y en la medida proporcionada por los sensores disponibles en los motores (sensores de velocidad y posición).

DESCRIPCIÓN DEL ALGORITMO. La posición del móvil en el plano $[X, Y]$ (supondremos movimiento restringido a un sólo plano) viene determinada por la terna (X, Y, θ) , donde x es la proyección sobre el eje de abscisas, y sobre el de ordenadas y θ la orientación del robot. La comparación entre la posición comandada (X_D, Y_D) y la estimada (X_A, Y_A) nos permitirá obtener los comandos de velocidad.

Las coordenadas estimadas de la posición actual, (X, Y, θ) , deben de obtenerse a partir de las medidas de la velocidad de los motores que nos suministra cada uno de los pines de la tarjeta ARDUINO UNO.

Una vez se han obtenido las velocidades lineal y angular actuales, es necesario obtener la posición del mismo. Para ello, se aplica el modelo cinemático del vehículo. El modelo cinemático del vehículo, relaciona las velocidades lineales y de rotación del

vehículo con las coordenadas del vehículo en un espacio 2D. Es importante notar que la posición del vehículo en dicho espacio no viene dada sólo por las coordenadas (X, Y), sino que es necesario incorporar una tercera coordenada que indique la orientación del vehículo en cada coordenada, es decir, (X, Y, θ). Las relaciones están expresadas en las ecuaciones (5.19), (5.20) y (5.21) [13], [14], [22] y [23].

$$\theta(t) = \int_0^t w_t dt \quad (5.19)$$

$$x(t) = \int_0^t v(t) \cos \theta_t dt \quad (5.20)$$

$$y(t) = \int_0^t v(t) \sin \theta_t dt \quad (5.21)$$

Una vez estimadas las coordenadas actuales, es necesario generar las nuevas acciones de control para llevar al robot hacia el siguiente punto de la trayectoria. Para ello, es necesario determinar si dada la posición actual y la referencia es necesario aplicar algún cambio de dirección al robot. Se calcula el valor de la orientación deseada θ_D mediante la ecuación (5.22):

$$\theta_D = \text{atan} \frac{Y_D - Y_A}{X_D - X_A} \quad (5.22)$$

Una vez obtenido el comando de orientación, ya es posible obtener los comandos de velocidad lineal (V_T) y de rotación (V_D). Ambos comandos se calcularán como salida de un regulador proporcional (K_D y K_V), según las siguientes expresiones (5.23), (5.24), (5.25) y (5.26) derivadas del modelo cinemático del robot:

$$V_D = K_D \frac{\theta_D - \theta_A}{T_s} \quad (5.23)$$

$$V_{TX} = K_V \frac{X_D - X_A}{T_s} \quad (5.24)$$

$$V_{TY} = K_V \frac{Y_D - Y_A}{T_s} \quad (5.25)$$

$$V_T = \sqrt{V_{TX}^2 + V_{TY}^2} \quad (5.26)$$

Donde T_s es el periodo de muestreo. Una vez obtenidos los comandos V_D y V_T , la generación de los valores de rotación a las ruedas, utilizando la técnica de PWM (Modulación de Ancho de Pulso), a través de un controlador tangente hiperbólica (Tanh).

CAPÍTULO

6 INSTRUMENTACIÓN DEL ROBOT MÓVIL HUMMER H2

6.1 PLATAFORMA INICIAL

La constitución inicial del vehículo de control remoto HUMMER H2, contaba con los siguientes elementos:

MOTOR DE TRACCIÓN. El cual fue caracterizado con el propósito de identificar el valor de la frecuencia en la cual su comportamiento fuera linealmente eficiente para poder controlarlo.

SERVOMOTOR DE DIRECCIÓN. Este actuador se reemplazó debido a que el servomotor original contaba con engranes dentados de plástico, los cuales se barrieron por desgaste. En su lugar se colocó un servomotor con una fuerza de torque máxima de 15 kilogramos, que a su vez posee engranes dentados de metal, con los cuales obtenemos un mejor desempeño en el eje de dirección del robot móvil HUMMER H2.

SENSOR DE DIRECCIÓN. Esta función la realiza un potenciómetro de resistencia variable, el cual se encuentra adaptado dentro de la estructura del servomotor de dirección.

BATERÍA DE LITIO 7.2V. Es la batería con la que originalmente contaba el vehículo a control remoto.

Los elementos antes mencionados y la estructura mecánica del vehículo conforman la plataforma sobre la que se implementó la instrumentación electrónica necesaria para desarrollar el control del vehículo y así convertir las características de un juguete a características de un robot móvil controlado por radiofrecuencia.

6.2 INSTRUMENTACIÓN PARA LA COMUNICACIÓN INALÁMBRICA

Tarjeta de interfaz USB (Dispositivo Local)

La tarjeta de interfaz mediante cable RS-232 para realizar la comunicación inalámbrica se muestra en la figura 6.1.



Figura 6.1. Tarjeta de interfaz USB (XBIB-U-DEV).

La tarjeta de interfaz USB proporciona una conexión directa con dispositivos que usan protocolo USB. Además facilita el acceso a los registros de los módulos de radiofrecuencia XBEE.

Los parámetros de configuración almacenados, permite al usuario personalizar al módulo de acuerdo a la necesidad de su sistema.

Esta tarjeta de interfaz proporciona los medios para conectar el módulo a cualquier dispositivo que tiene disponible puerto USB, como por ejemplo una computadora.

Además esta tarjeta se alimenta por medio de su puerto, el dispositivo al que está conectado le proporciona el voltaje necesario de alimentación.

La tarjeta nos proporciona una interfaz física para visualizar el estado de comunicación y alimentación, la cual se muestra en la figura 6.2

Como se puede observar en la figura 6.2, la tarjeta de interfaz posee un conector tipo B, el cual se conecta a dispositivos que tengan puerto USB.

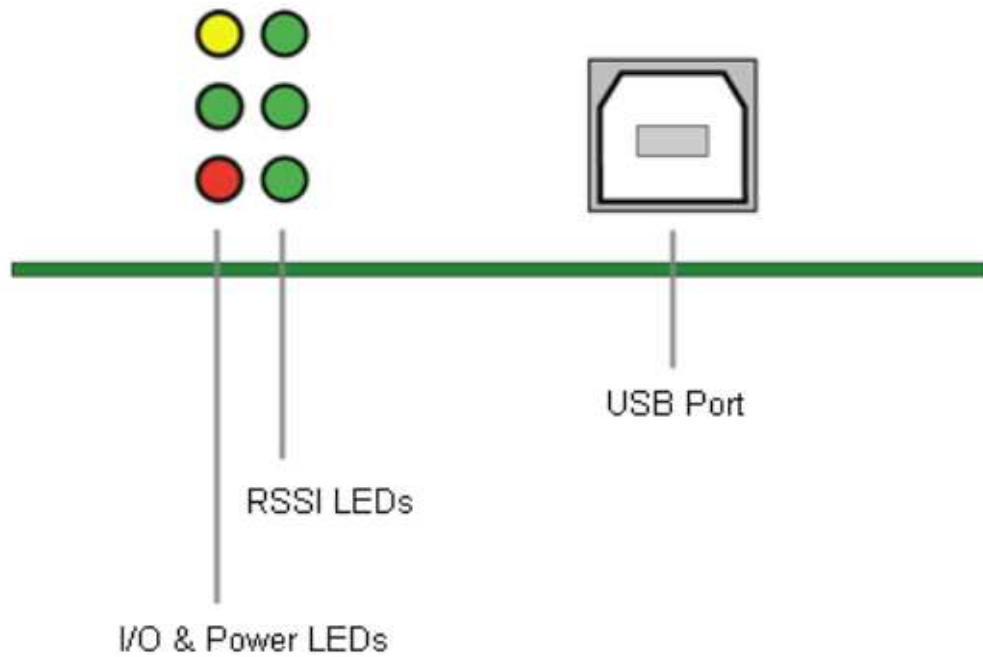


Figura 6.2. Interfaz física e indicadores de la tarjeta USB.

También tiene 3 “I/O Power Leds” los cuales indican la actividad del módulo y se describen a continuación:

- Amarillo (Led superior) = Dato Serial de Salida (hacia el host remoto).
- Verde (Led en medio) = Dato Serial de Entrada (desde el host remoto).
- Rojo (Led inferior) = Alimentación / Indicador de Transmisión (Led se enciende cuando el módulo es alimentado).

También posee 3 “RSSI Leds” los cuales indican la cantidad de margen de desvanecimiento de datos en un enlace inalámbrico activo.

- 3 Leds encendidos = Señal muy fuerte (> 30 dB Fade Margin)
- 2 Leds encendidos = Señal fuerte (> 20 dB Fade Margin)
- 1 Leds encendidos = Señal moderada (> 10 dB Fade Margin)
- 0 Leds encendidos = Señal débil (< 10 dB Fade Margin)

En el apéndice A se anexa la arquitectura de la tarjeta XBIB-U-DEV, la cual posee cuatro pulsadores, 4 leds de configuración y un pulsador de reset.

Los cuatro pulsadores son de configuración de la tarjeta y los cuatros leds son de visualización de estado en que se encuentran, la siguiente tabla describe la conexión pin a pin del modulo XBEE-PRO con respecto a la tarjeta de interfaz.

Pin del Módulo XBEE-PRO	Conexión XBIB-U-DEV
2.- DOUT	RXD – U6 pin 24(USB), J7 pin 2
3.- DIN/DCONFIG	TXD - U6 pin 25 (USB), J7 pin 3
4.- CD/DOUT_EN/DO8	LED 1 & DOUT LED circuit, J7 pin 4
5.- RESET	RESET, J7 pin 5
6.- PWM0/RSSI	RSSI LED circuit input, J7 pin 6
7.- PWM1	LED 2, J7 pin 1
8.- Reservado	J2 pin 1(Programming/Debug header), J7 pin 8
9.- DTR /SLEPP_RQ/DI8	DTR – U6 pin 21 (USB), J5 pin 9
11.- RF_TX/AD4/DIO4	LED 3, J8 pin 11
12.- CTS /DIO7	CTS – U6 pin 22 (USB), J8 pin 10
13.- ON/ SLEEP	J8 pin 9
14.- VREF	J8 pin 7
15.- Associate/AD5/DIO5	LED 4, J8 pin 6
16.- RTS /AD6/DIO6	RTS – U6 pin 23 (USB), J8 pin 5
17.- COORD_SEL/AD3/DIO3	SW1, J8 pin 4
18.- AD2/DIO2	SW2, J8 pin 3
19.- AD1/DIO1	SW3, J8 pin 2
20.- AD0/DIO0	SW4, J8 pin 1

Tabla 6.1. Conexión del módulo XBEE-PRO con la tarjeta XBIB-U-DEV.

Módulos de radiofrecuencia

Los módulos XBEE-PRO son módulos de radiofrecuencia que operan en la banda de 2.4GHz con protocolo de comunicación IEEE_802.15.4. El alcance normal con antena dipolo en línea vista es de 1.6 Km al aire libre y en interiores con obstáculos de aproximadamente unos 100m.

La figura 6.3 muestra las conexiones mínimas que necesita el módulo XBEE-PRO para poder ser utilizado, la cual no permite el uso de Control de Flujo (RTS & CTS), por consiguiente esta opción debe estar desactivada.

El módulo requiere una alimentación desde 2.8 a 3.4 V, la conexión a tierra y las líneas de transmisión de datos por medio del UART (TXD y RXD) para comunicarse con un microcontrolador, o directamente a un puerto serial utilizando algún

conversor adecuado para los niveles de voltaje.

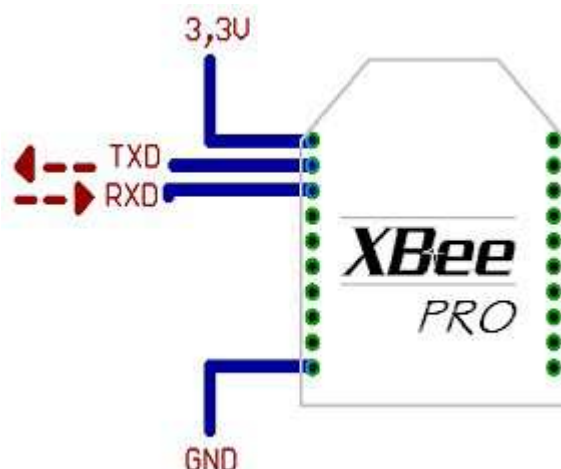


Figura 6.3. Conexiones mínimas requeridas para el módulo XBEE-PRO.

El módulo requiere una alimentación desde 2.8 a 3.4 V, la conexión a tierra y las líneas de transmisión de datos por medio del UART (TXD y RXD) para comunicarse con un microcontrolador, o directamente a un puerto serial utilizando algún conversor adecuado para los niveles de voltaje. Entre las necesidades que satisface el módulo se encuentran:

- *Ultra-bajo consumo de potencia.
- *Uso de bandas de radio libres y sin necesidad de licencias.
- *Instalación barata y simple, redes flexibles.

Todos los dispositivos que disponen de una interfaz UART se pueden conectar directamente a los pines del módulo XBEE-PRO, como se muestra en la figura 6.4.

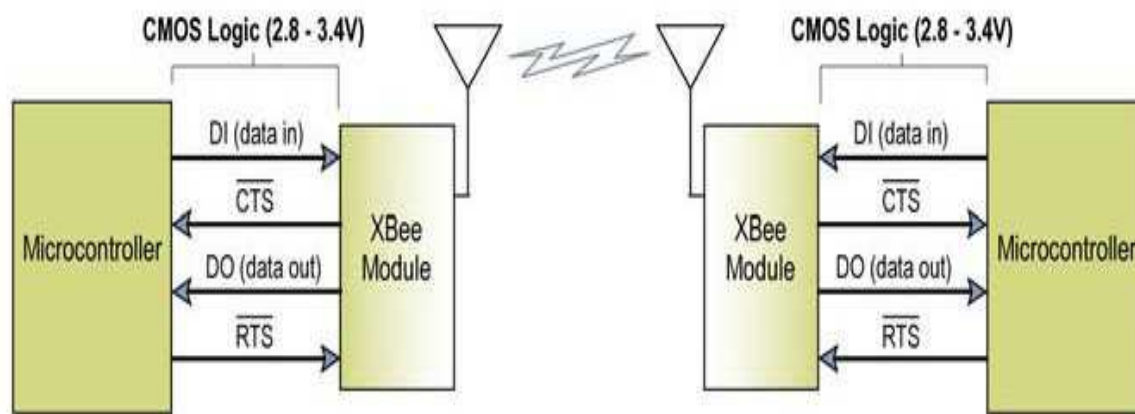


Figura 6.4. Sistema de flujo de datos mediante microcontroladores.

OPERACIÓN EN MODO TRANSPARENTE (AT). En este modo todo lo que ingresa por el pin 3 (Data in), es guardado en el buffer de entrada y luego transmitido y todo lo que ingresa como paquete de radiofrecuencia, es guardado en el buffer de salida y luego enviado por el pin 2 (Data out). El modo Transparente viene por defecto en los módulos XBEE-PRO.

Este modo está destinado principalmente a la comunicación punto a punto, donde no es necesario ningún tipo de control. También se usa para reemplazar alguna conexión serial por cable, ya que es la configuración más sencilla posible y no requiere una mayor configuración.

En este modo, la información es recibida por el pin 3 del módulo XBEE-PRO, y guardada en el buffer de entrada. Dependiendo de cómo se configure el comando RO, se puede transmitir la información apenas llegue un carácter (RO=0) o después de un tiempo dado sin recibir ningún carácter serial por el pin 3.

En ese momento, se toma lo que se tenga en el buffer de entrada, se empaqueta, es decir, se integra a un paquete RF, y se transmite.

A continuación se muestra el diagrama de pines en la figura 6.5 del modulo XBEE-PRO desde una vista superior.

Notas de diseño:

*Conexiones mínimas: VCC, GND, DOUT y DIN

*Conexiones mínimas de descarga el firmware: VCC, GND, DOUT, DIN, RTS y DTR.

*El módulo incluye una resistencia pull-up de 50k Ω para el Reset, es decir se conecta a nivel lógico alto.



Figura 6.5. Diagrama de pines Vista Superior.

En el apéndice B se anexa la tabla descripción de pines de los módulos XBEE-PRO, la tabla de especificaciones eléctricas de los módulos XBEE-PRO, la tabla de detalles eléctricos de las entradas análogas/digitales de los módulos XBEE-PRO, la tabla de las características de rendimiento de los módulos XBEE-PRO y la tabla de las especificaciones técnicas de los módulos XBEE-PRO.

Cable USB

Se utiliza para conectar la tarjeta de interfaz inalámbrica hacia la computadora, y se muestra en la figura 6.6.



Figura 6.6. Cable USB.

Adaptador de voltaje (12V)

Alimenta a la tarjeta de interfaz de comunicación inalámbrica USB y realiza la función de recargar la batería del robot móvil HUMMER H2 (figura 6.7). En el apéndice C se anexa la tabla de especificaciones generales del adaptador de voltaje.



Figura 6.7. Adaptador de voltaje 12 VDC.

Computadora portátil

Almacena y ejecuta el algoritmo de control, recibiendo las señales de los sensores, como señales de entrada y por otro lado enviando las señales de PWM (para aplicar el torque a los motores), como señales de salida. De esta manera se realiza la retroalimentación formando una estructura de control en lazo cerrado. La figura 6.8 ilustra la computadora empleada para esta tarea.



Figura 6.8. Computadora portátil.

Vista general del dispositivo Local

En la figura 6.9 se muestra la conexión utilizada para implementar el dispositivo Local de la comunicación inalámbrica, donde se observa el módulo XBEE-PRO montado en la tarjeta de interfaz XBIB-U-DEV y esta a su vez conectada a través de un cable USB al puerto serie de la computadora portátil.



Figura 6.9. Dispositivo Local.

6.3 INSTRUMENTACIÓN PARA EL PROCESAMIENTO DE DATOS

Tarjetas ARDUINO (Dispositivo Remoto)

Arduino es una plataforma de hardware libre, basada en una placa con un microcontrolador y un entorno de desarrollo, diseñada para facilitar el uso de la electrónica en proyectos multidisciplinarios. Dos de las placas Arduino más utilizadas son las siguientes [24] y [25]:

ARDUINO UNO. Arduino UNO es una placa con un microcontrolador de la marca Atmel y con toda la circuitería de soporte, que incluye, reguladores de tensión, un puerto USB (En los últimos modelos, aunque el original utilizaba un puerto serie) conectado a un módulo adaptador USB-Serie que permite programar el microcontrolador desde cualquier PC de manera cómoda y también hacer pruebas de comunicación con el propio chip.

Un Arduino UNO dispone de 14 pines que pueden configurarse como entrada o salida y a los que puede conectarse cualquier dispositivo que sea capaz de transmitir o recibir señales digitales de 0 y 5 V.

También dispone de entradas y salidas analógicas. Mediante las entradas analógicas podemos obtener datos de sensores en forma de variaciones continuas de un voltaje. Las salidas analógicas suelen utilizarse para enviar señales de control en forma de señales PWM.

Arduino UNO es la última versión de la placa, existen dos variantes, la Arduino UNO convencional y la Arduino UNO SMD. La única diferencia entre ambas es el tipo de microcontrolador que montan.

La primera es un microcontrolador Atmega en formato DIP, mostrada en la figura 6.10.



Figura 6.10. Arduino UNO con microcontrolador en formato DIP.

La segunda dispone de un microcontrolador en formato SMD, mostrada en la figura 6.11.



Figura 6.11. Arduino UNO con microcontrolador en formato SMD.

ENTRADAS Y SALIDAS. Cada uno de los 14 pines digitales se puede usar como entrada o como salida. Funcionan a 5V, cada pin puede suministrar hasta 40 mA. La intensidad máxima de entrada también es de 40 mA.

Cada uno de los pines digitales dispone de una resistencia de pull-up interna de entre 20K Ω y 50 K Ω que está desconectada, salvo que nosotros indiquemos lo contrario.

Arduino también dispone de 6 pines de entrada analógicos que trasladan las señales a un conversor analógico/digital de 10 bits.

PINES ESPECIALES DE ENTRADA Y SALIDA. Los pines 2 y 3 están configurados para generar una interrupción en el atmega. Las interrupciones pueden dispararse cuando se encuentra un valor bajo en estas entradas y con flancos de subida o bajada de la entrada.

RX y TX: Se usan para transmisiones serie de señales TTL.

PWM: Arduino dispone de 6 salidas destinadas a la generación de señales PWM de hasta 8 bits.

SPI: Los pines 10, 11, 12 y 13 pueden utilizarse para llevar a cabo comunicaciones SPI, que permiten trasladar información full dúplex en un entorno Maestro/Esclavo.

I2C: Permite establecer comunicaciones a través de un bus I2C. El bus I2C es un producto de Phillips para interconexión de sistemas embebidos. Actualmente se puede

encontrar una gran diversidad de dispositivos que utilizan esta interfaz, desde pantallas LCD, memorias EEPROM, sensores, entre otros.

ALIMENTACIÓN DEL ARDUINO UNO. Puede alimentarse directamente a través del propio cable USB o mediante una fuente de alimentación externa, como puede ser un pequeño transformador o, por ejemplo una pila de 9V.

Los límites están entre los 6 y los 12 V. Como única restricción hay que saber que si la placa se alimenta con menos de 7V, la salida del regulador de tensión a 5V puede dar menos que este voltaje y si sobrepasamos los 12V, probablemente dañaremos la placa.

La alimentación puede conectarse mediante un conector de 2,1mm con el positivo en el centro o directamente a los pines Vin y GND marcados sobre la placa.

Hay que tener en cuenta que podemos medir el voltaje presente en el jack directamente desde Vin. En el caso de que el Arduino esté siendo alimentado mediante el cable USB, ese voltaje no podrá monitorizarse desde aquí. En la tabla 6.2 se describe un resumen de las características técnicas de la tarjeta Arduino UNO.

Microcontrolador	Atmega328
Voltaje de operación	5V
Voltaje de entrada (Recomendado)	7 – 12V
Voltaje de entrada (Límite)	6 – 20V
Pines para entrada- salida digital.	14 (6 pueden usarse como salida de PWM)
Pines de entrada analógica.	6
Corriente continua por pin IO	40 mA
Corriente continua en el pin 3.3V	50 mA
Memoria Flash	32 KB (0,5 KB ocupados por el bootloader)
SRAM	2 KB
EEPROM	1 KB
Frecuencia de reloj	16 MHz

Tabla 6.2. Características de la tarjeta Arduino UNO.

ARDUINO SHIELD.

La tarjeta Arduino SHIELD (figura 6.12), es una solución de Hardware para implementar dispositivos montados sobre el Arduino. Una particularidad de este tipo de tarjetas es la tarjeta SHIELD para XBEE, donde se puede montar un módulo XBEE o un módulo BLUETOOTH. El SHIELD de XBEE que utilizaremos en este trabajo posee dos opciones para seleccionar el modo de trabajo etiquetados con XBEE y USB. Con el

selector en la posición XBEE, el pin DOUT del módulo XBEE está conectado al pin RX del microcontrolador; y el pin DIN está conectado a TX. Con el selector en la posición USB, el pin DOUT del módulo XBEE está conectado al pin RX del pin del chip FTDI; y el pin DIN está conectado a TX del chip FTDI.

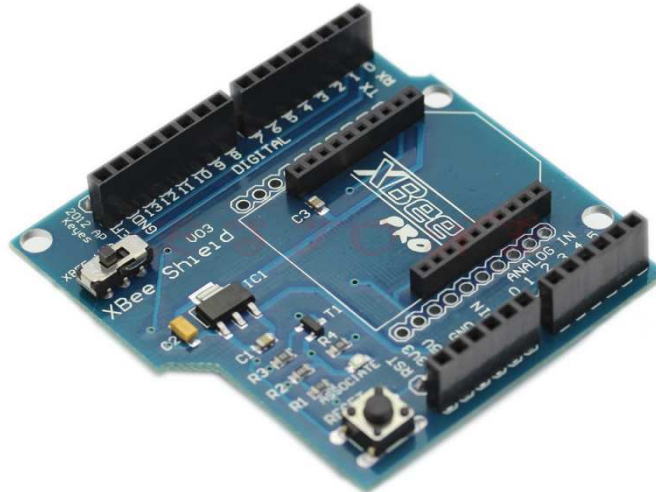


Figura 6.12. Placa Arduino SHIELD para XBEE.

Vista general del dispositivo Remoto

En la figura 6.13 se muestra la conexión utilizada para implementar el dispositivo Remoto de la comunicación inalámbrica, donde se observa el módulo XBEE-PRO montado en la tarjeta ARDUINO SHIELD y esta a su vez montada sobre una tarjeta de adquisición de datos ARDUINO UNO.

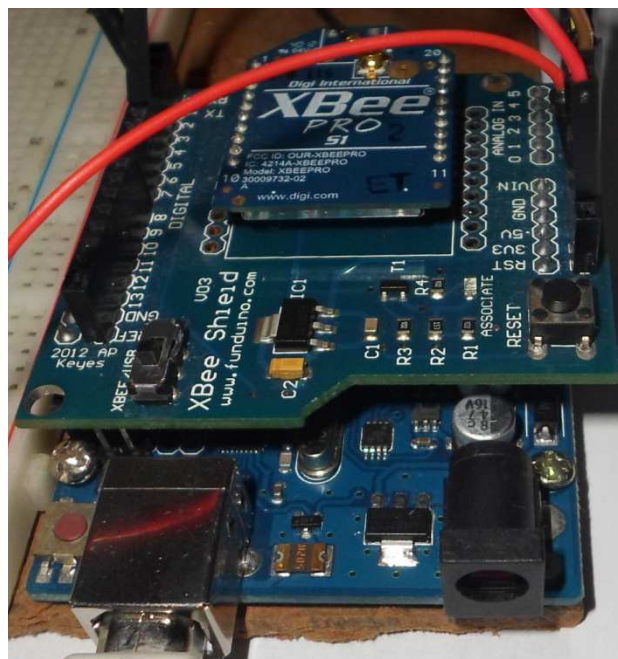


Figura 6.13. Dispositivo Remoto.

Vista general del sistema de comunicación inalámbrica

La interfaz utilizada para la recepción de datos de todos los sensores (potenciómetro, encoder y brújula) y el envío de datos a los actuadores (motores) se ilustra en la figura 6.14, donde la tarjeta ARDUINO UNO funge como tarjeta de adquisición de datos, los módulos XBEE-PRO son el canal por el cual se envían y reciben los datos de forma bidireccional y finalmente la computadora es el elemento que ejecuta el algoritmo de control en lazo cerrado.

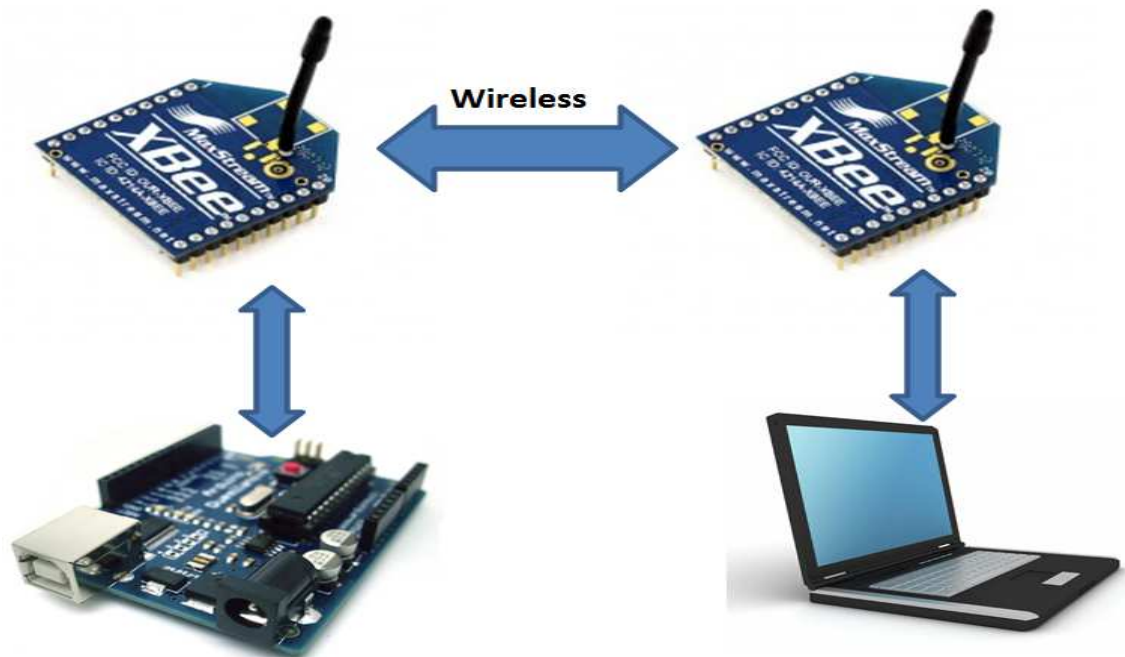


Figura 6.14. Sistema de comunicación inalámbrica.

6.4 INSTRUMENTACIÓN PARA LA ADQUISICIÓN DE DATOS

Encoder como sensor de Velocidad (v)

Una de las variables deseadas que se obtuvieron en el modelo dinámico del robot móvil HUMMER H2 es la velocidad.

La forma ideada para medir dicha magnitud es mediante un disco montado en una de las llantas de tracción del móvil, para obtener en forma de pulsos su distancia recorrida en el tiempo y obtener la medida de la variable física "velocidad".

Para leer los cambios de flanco producidos por las ranuras del disco se utilizan dos sensores de herradura fotoeléctricos, los cuales se acoplan para producir dos señales desfasadas 90° y así formar un encoder en cuadratura, el cual también será montado en

una de las llantas de tracción del robot móvil HUMMER H2, en conjunto con el disco ranurado.

La lectura de los pulsos proveniente del sensor en cuadratura se envía a una compuerta inversor tipo trigger, ya que es un circuito sugerido para limpiar los pulsos enviados por el encoder como se muestra en la figura 6.15.

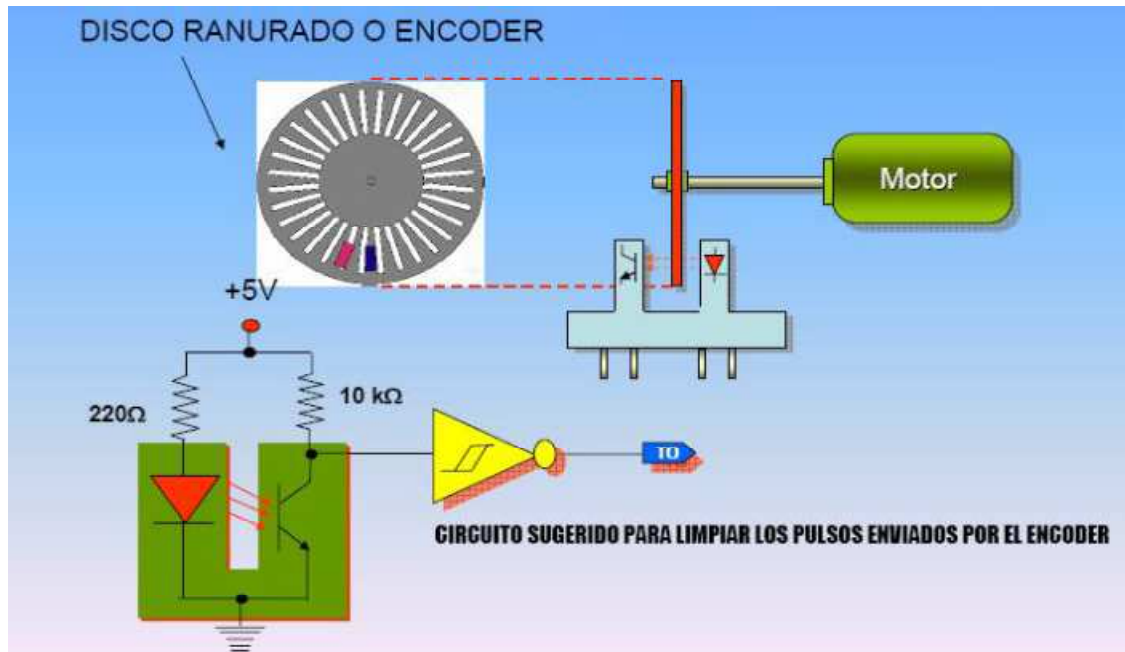


Figura 6.15. Principio utilizado para la implementación del encoder.

Finalmente las 2 señales producidas por el encoder en cuadratura (mostrado en la figura 6.16) , se envían cada una a una entrada digital de la tarjeta Arduino, donde a su vez se transmiten los datos de manera inalámbrica a la computadora a través de un puerto serial virtual, utilizando los módulos XBEE-PRO.

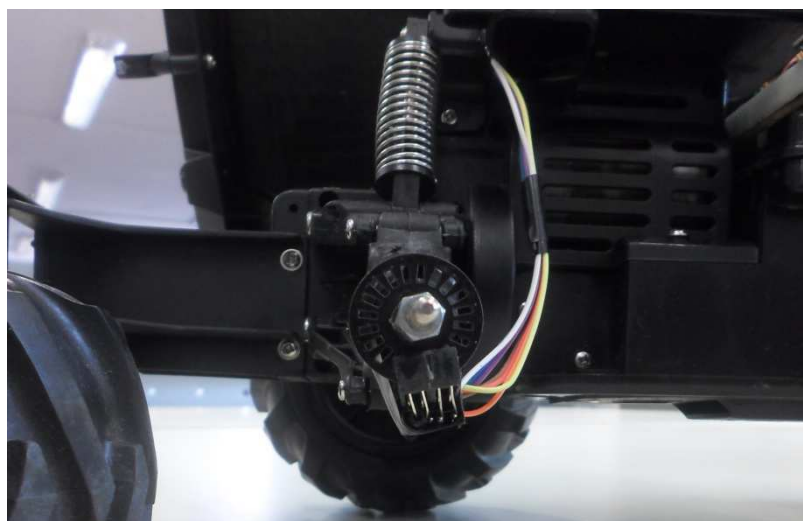


Figura 6.16. Encoder implementado en el robot móvil HUMMER H2.

Brújula como sensor de orientación (β)

Otra de las variables deseadas que se obtuvieron en el modelo dinámico del robot móvil HUMMER H2 es el ángulo de deslizamiento lateral del chasis, el cual es equivalente a la orientación del robot móvil HUMMER H2.

El sensor que nos ofrece de manera directa la medida de esta magnitud con respecto al polo magnético de la tierra es el compás o magnetómetro HMC5883L, el cual es una brújula de tres ejes (x y z).

La comunicación con el HMC5883L es simple y se puede realizar a través de una interfaz I2C. Esta brújula no incorpora un regulador de voltaje, por lo que el voltaje de alimentación debe estar entre 2.16-3.6VDC.

El circuito HMC5883L permite alcanzar la frecuencia de hasta 160 Hz, si se usa un único modo de medición y monitoreo de interrupción DRDY. Una posibilidad interesante que ofrece el sistema es la elección del número de muestras (1, 2, 4 u 8), que se promedian sobre el resultado final.

Para la implementación física del circuito usaremos dos resistencias de 2.2k, tal y como recomienda el datasheet del sensor (anexado en el apéndice D) entre los pines Vcc-SDA y Vcc-SCL. El pin Vcc deberá ir a la entrada de 3.3V de la tarjeta Arduino ya que el sensor necesita un voltaje entre 2.16v a 3.6v.

Como el sensor funciona mediante I2C se conecta la línea de datos SDA y la línea de reloj SCL a los Pines A4 y A5 de la tarjeta Arduino respectivamente. En la figura 6.17 se muestra como se deben conectar los pines del sensor a la tarjeta Arduino UNO.

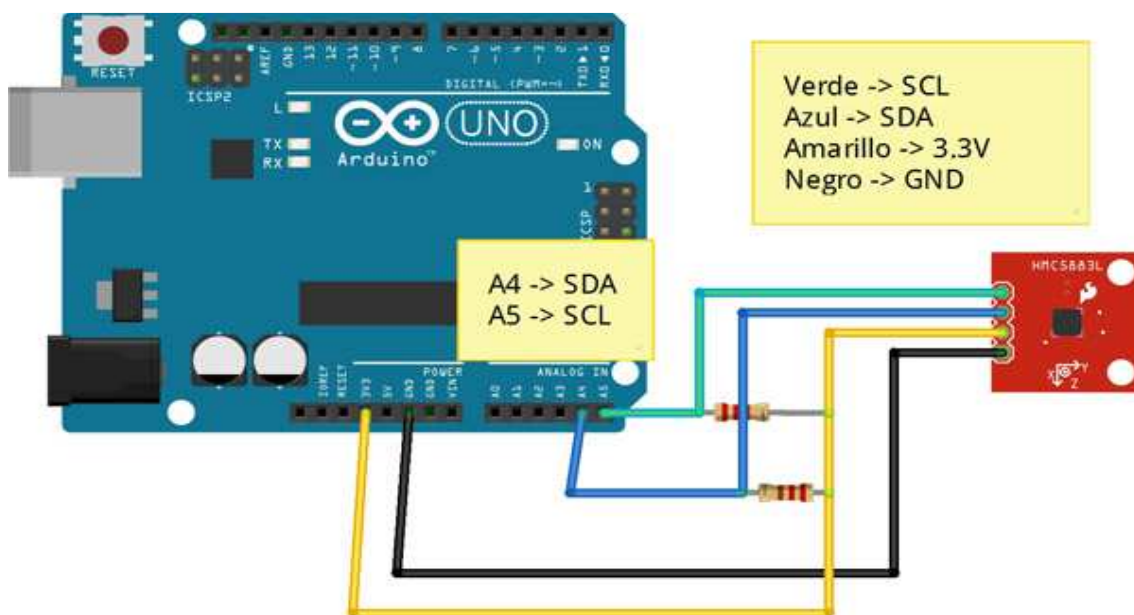


Figura 6.17. Conexión de la brújula HMC5883L a Arduino UNO.

En las pruebas realizadas con el compás HMC5883L se aprecia considerablemente un error de lectura con respecto a los grados de orientación del vehículo, esto es debido al campo magnético producido por los motores.

Para tal inconveniente se implementa un tubo de PVC de 1 pulgada para aislar al sensor lo más lejos posible de la estructura del vehículo. Su peso es de 80 gramos y en la figura 6.18, se ilustra su acoplamiento con el magnetómetro HMC5883L en el robot móvil.



Figura 6.18. La brújula electrónica HMC5883L implementada en el robot móvil HUMMER H2.

Potenciómetro como sensor de dirección (δ_D)

La tercera variable medida es la dirección del eje delantero del robot móvil HUMMER H2, la cual nos da referencia del ángulo de giro en el cual se encuentra eje de dirección del móvil en determinado tiempo.

Esta variable física es medida a través de un potenciómetro montado en el eje delantero del HUMMER H2.

Este sensor está incluido dentro de la infraestructura disponible al comienzo de este trabajo ya que es parte de la mecánica inicial del vehículo.

La señal que se obtiene de este sensor es analógica (0v a 5v) y depende del movimiento del eje delantero del robot, con lo que varía la resistencia y al mismo tiempo el voltaje enviado al puerto serial por medio de la interfaz antes planteada.

En la figura 6.19 se muestra la forma de montaje del sensor de dirección.

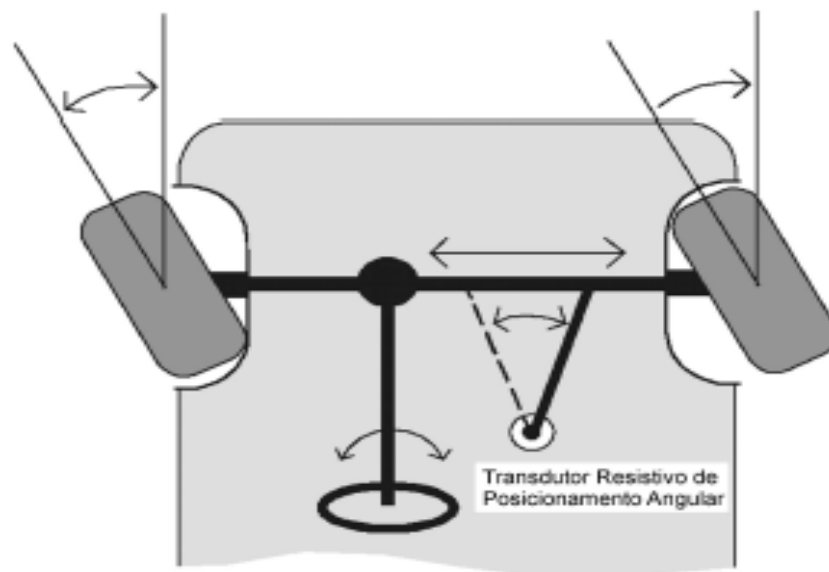


Figura 6.19. Sensor del ángulo de dirección de las ruedas delanteras.

En la figura 6.20 se ilustra el muelle delantero del robot móvil HUMMER H2, donde se pueden observar los cables de conexión del servomotor (azul y amarillo) y del potenciómetro (anaranjado, rojo y café), los cuales se conectan a la tarjeta ARDUINO UNO para poder realizar la lectura del cambio de dirección del eje delantero del vehículo.

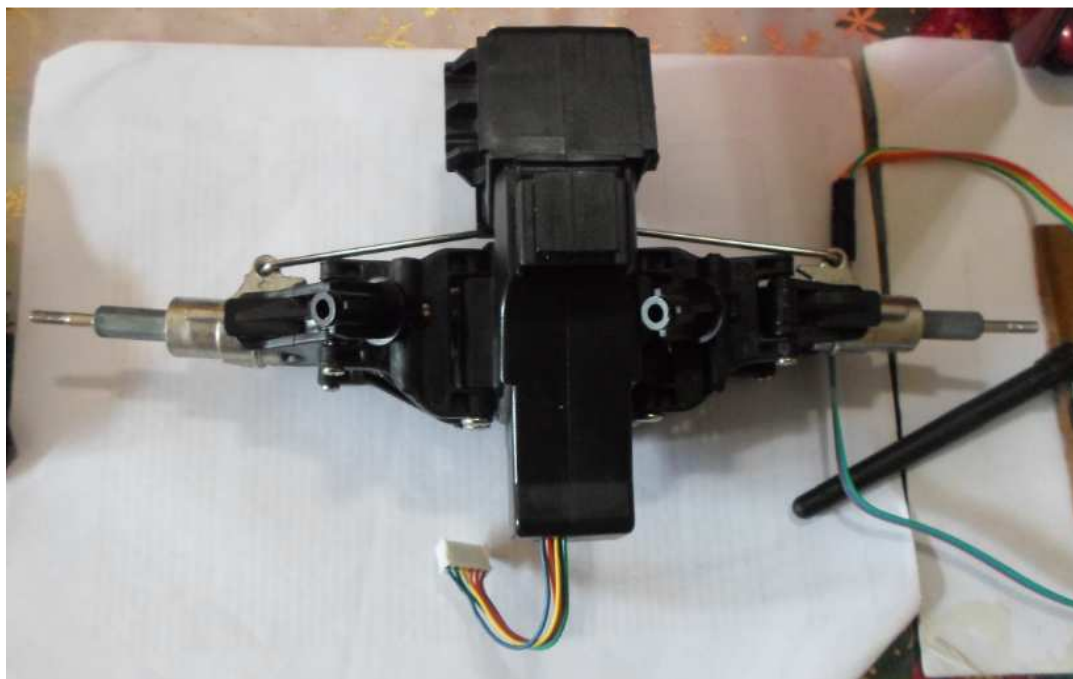


Figura 6.20. Potenciómetro empotrado en eje delantero del robot móvil HUMMER H2.

6.5 INSTRUMENTACIÓN PARA LA ETAPA DE POTENCIA

Doble Puente H VN2SP30-E

El VN2SP30-E (figura 6.21) es un controlador de motor de puente completo, destinado a una amplia gama de aplicaciones de automatización [24].

Contiene 2 puentes H, los cuales se utilizan para cada motor del robot móvil HUMMER H2 (motor de tracción y motor de dirección) de forma independiente.

Las señales de entrada INA e INB pueden interactuar directamente al microcontrolador para seleccionar la dirección del motor y de la condición del freno.

La velocidad del motor puede ser controlada en todas las condiciones posibles por una señal PWM de hasta 20 kHz.

Entre sus características principales se encuentran:

- *Corriente de operación hasta 30.
- *Insumos compatibles nivel lógico 5V.
- *Baja tensión y sobretensión de apagado.
- *Cerrada térmica abajo.
- *La protección cruzada de conducción.
- *Limitador de corriente Lineal.
- *Muy bajo consumo de energía stand-by.
- *PWM operación de hasta 20 kHz.
- *Protección contra la pérdida de tierra y la pérdida de V_{CC} .
- *Corriente de salida proporcional a la corriente de sentido del motor.

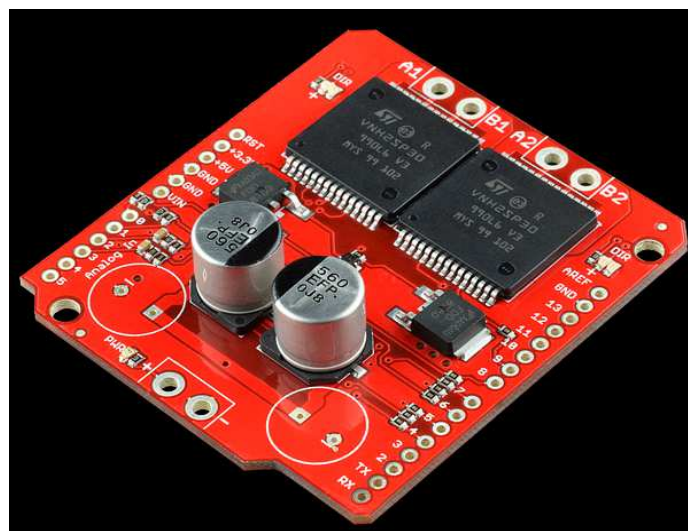


Figura 6.21. Doble Puente H VN2SP30-E

En el apéndice E se anexa la ficha técnica así como también el diagrama de diseño electrónico de la tarjeta VNH2SP30.

A continuación se muestra en la figura 6.22 la implementación de la tarjeta de potencia VNH2SP30 en el robot móvil HUMMER H2, donde se realizan las conexiones correspondientes a la batería de voltaje (fuente de voltaje), al ARDUINO UNO (señales PWM) y a los motores (torque de tracción y torque de dirección).

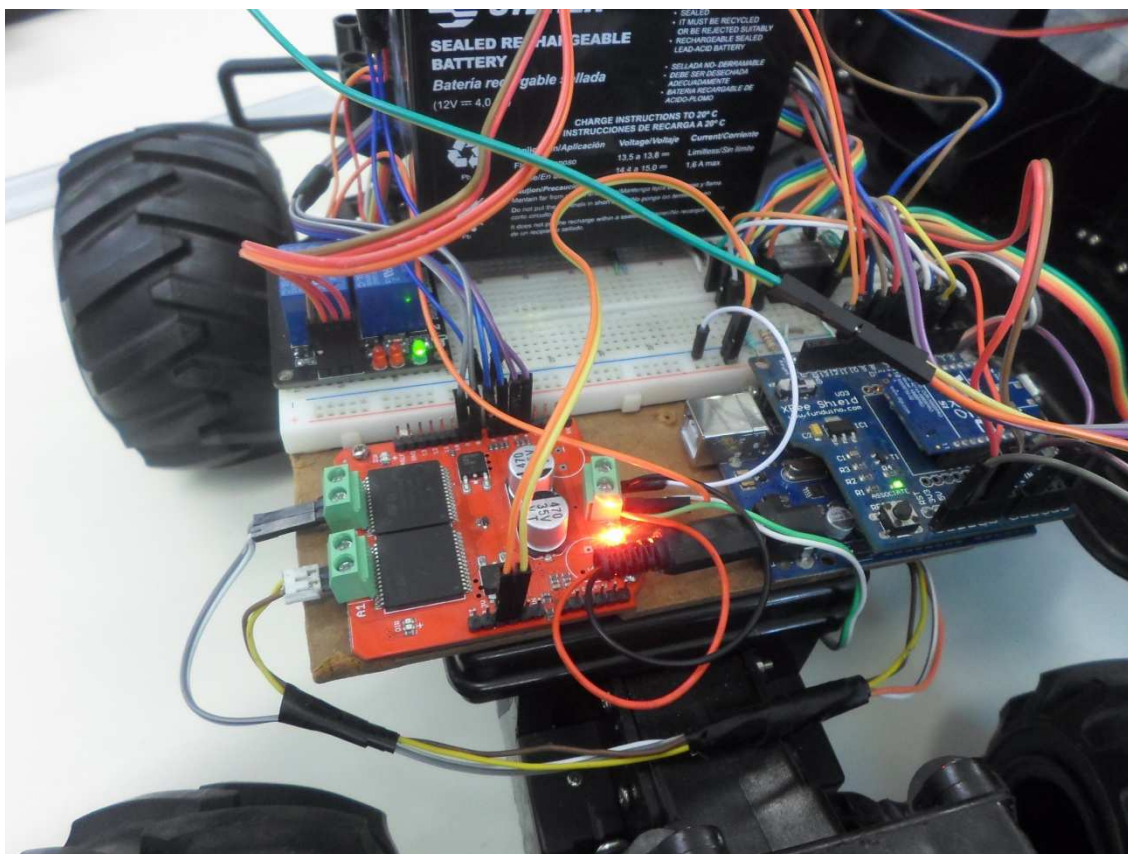


Figura 6.22. Implementación del doble Puente H VNH2SP30-E en el robot móvil HUMMER H2.

Relevadores

Para el control de las luces delanteras y traseras se utiliza una placa de dos relevadores, los cuales se activan con 5V, a la salida se conecta la batería de suministro de energía del vehículo, para energizar las bombillas indicadoras.

En el apéndice F se muestra la ficha técnica de los relevadores, y en el apéndice G se anexa la ficha técnica de la batería de suministro de energía del robot móvil HUMMER H2.

En la figura 6.23 se muestra la tarjeta con los relevadores que activan las luces.

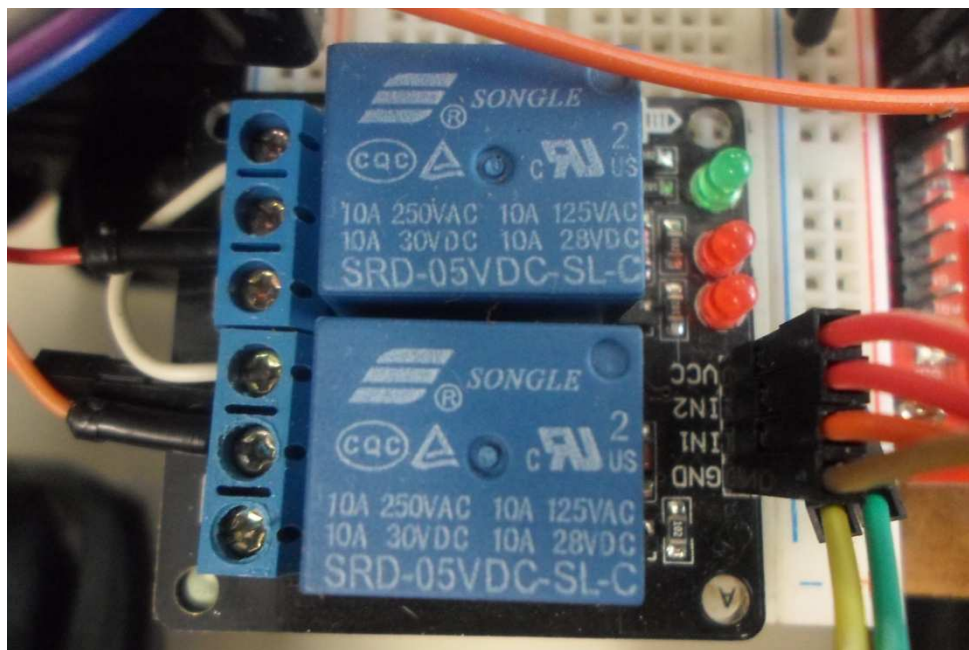


Figura 6.23. Relevadores para la activación de las luces del vehículo.

Vista general de la instrumentación del robot móvil HUMMER H2

En la figura 6.24 se ilustra el sistema de instrumentación electrónica del robot móvil.

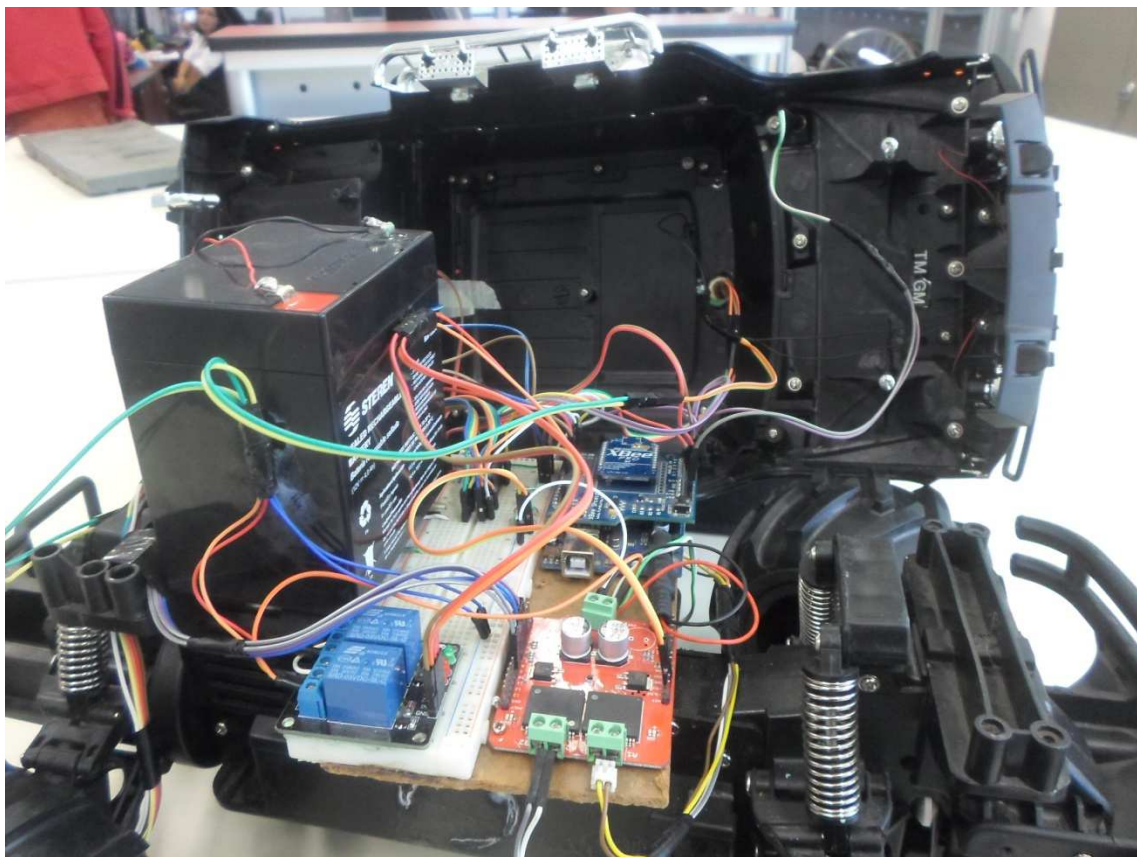


Figura 6.24. Instrumentación del robot móvil HUMMER H2.

CAPÍTULO

7 IMPLEMENTACIÓN DEL ALGORITMO DE CONTROL

7.1 ENTORNO DE PROGRAMACIÓN LABVIEW-ARDUINO-XBEE

Introducción al entorno de programación LABVIEW-ARDUINO

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.

Este trabajo pretende ser una respuesta a la necesidad que existe en el ámbito académico universitario de poder disponer de un entorno de Diseño y Prototipado de Aplicaciones de Medida, Control y Automatización de Procesos y Espacios Físicos. La plataforma Open Hardware Arduino ha demostrado en su corta, pero intensa vida, ser una opción muy interesante para incluir en un Laboratorio de Prototipado. Son muchas sus ventajas, entre las que destacamos su costo, su libre difusión y exención de costos de patentes por desarrollo así como la gran comunidad de usuarios que se ha creado y que está generando una cantidad numerosa de aplicaciones.

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.

Algunas de las razones que se consideraron para utilizar Arduino LABVIEW fueron:

1. *INTERFACE GRÁFICA DE USUARIO (GRAPHICAL USER INTERFACE GUI)*. Ya

que con esto se logra mostrar los datos de los sensores en el monitor del ordenador mediante los paneles frontales de LABVIEW.

Además permite dar al proyecto un toque profesional con los controles del panel frontal de LABVIEW y los indicadores como se muestra en la figura 7.1.

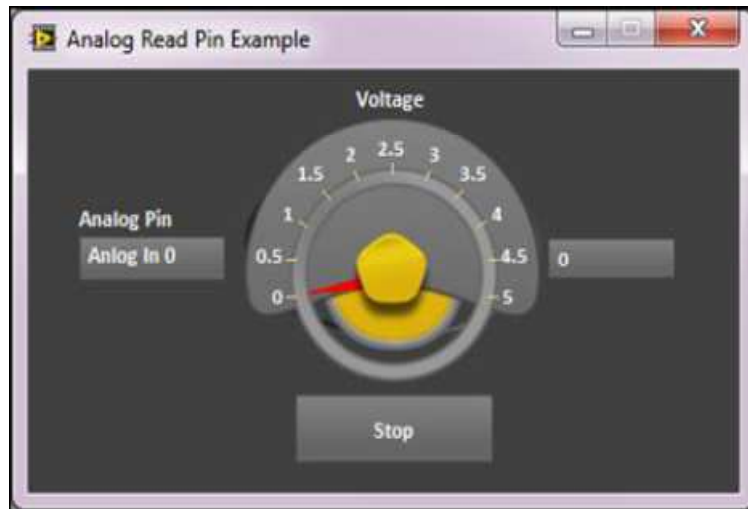


Figura 7.1. Interfaz gráfica de usuario.

2. *PROGRAMACIÓN GRÁFICA*. En lugar de tratar de recordar un nombre de función, se encuentra en la paleta y se coloca en su diagrama de bloques, mostrado en la figura 7.2.

Se pasa el cursor sobre cualquier VI o función y se muestra al instante la documentación con ayuda contextual.

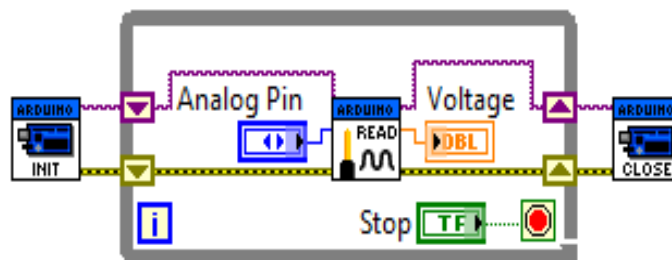


Figura 7.2. Programación gráfica en bloques (VI'S).

3. *DESARROLLO INTERACTIVO*. Se pueden consultar los valores de datos que se transmiten de una función a otra con resaltado de ejecución representada en la figura 7.3.

Se puede aplicar la ejecución de una pausa, y el paso a una subrutina sin necesidad de programación compleja.

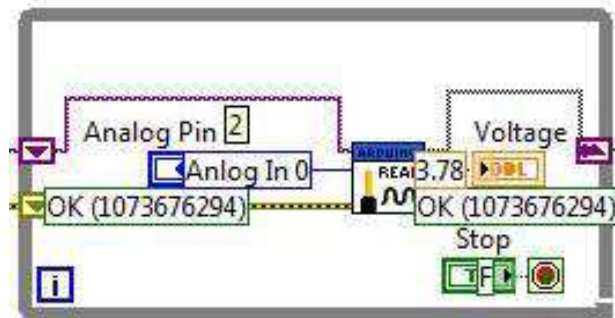


Figura 7.3. Desarrollo interactivo.

4. *UTILIZACIÓN DE LIBRERÍAS.* Permite el uso de cientos de bibliotecas integradas de procesamiento de señales, matemáticas y análisis (figura 7.4).

Al igual que permite el uso de la interfaz con los servicios web, bases de datos, archivos ejecutables y más con funcionalidad integrada en el núcleo de LABVIEW.

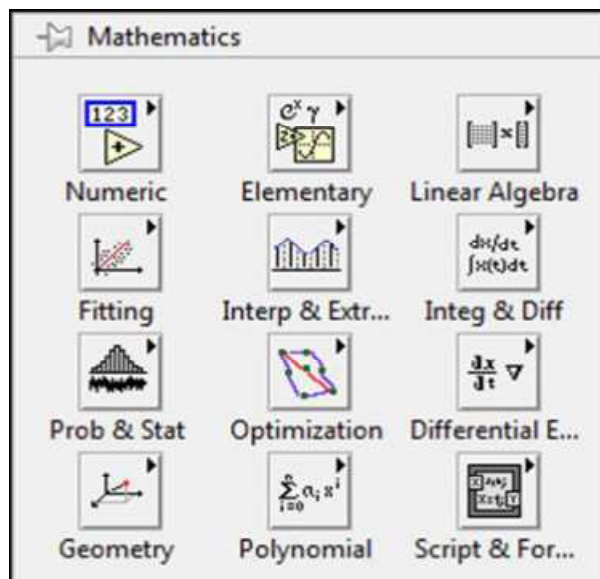


Figura 7.4. Librerías en LABVIEW.

5. *CÓDIGO API ABIERTO.* En el lado de LABVIEW se realiza doble clic en cualquier VI dentro de la caja de herramientas y se muestra cómo funciona.

En el lado de Arduino se incluye la modificación de cualquier Sketch en la caja de herramientas.

Instalación del entorno de programación LABVIEW-ARDUINO-XBEE

La configuración de la Interfaz de LABVIEW para Arduino es un proceso de varios pasos que se describen a continuación:

1. *INSTALACIÓN DEL SOFTWARE LABVIEW 2011.* Se descarga el software LABVIEW 2011 mostrado en la figura 7.5, para implementar la interfaz requerida, utilizando las ventajas y recursos que este programa ofrece. Se elige la versión 2011 debido a la facilidad de encontrar los toolkits gratuitos en la red que complementan las librerías requeridas para este trabajo.



Figura 7.5. Entorno de programación LABVIEW 2011.

2. *INSTALACIÓN DEL CONTROLADOR VISA-NIMAX 5.5.* Se instala el controlador VISA-NIMAX versión 5.5 (figura 7.6), el cual nos permite configurar los parámetros del puerto virtual utilizado en la interfaz como por ejemplo: baud rate, data bits, parity, stop bits y flow control.



Figura 7.6. VISA-NIMAX 5.5.

3. *INSTALACIÓN DEL PAQUETE JKI VI PACKAGE MANAGER (VIPM) 2014.* Posteriormente se instala el paquete JKI VI Package Manager (VIPM) 2014 (figura 7.7), por medio del cual se realizara la descarga e instalación de los toolkits necesarios para la implementación de la interfaz para el control del robot móvil HUMMER H2.



Figura 7.7. JKI VI Package Manager (VIPM) 2014.

4. *INSTALACIÓN DEL TOOLKIT INTERFAZ DE LABVIEW PARA ARDUINO.* Ahora se descarga el toolkit interfaz LABVIEW para Arduino a través del paquete JKI VI Package Manager (VIPM) 2014 ilustrado en la figura 7.8.

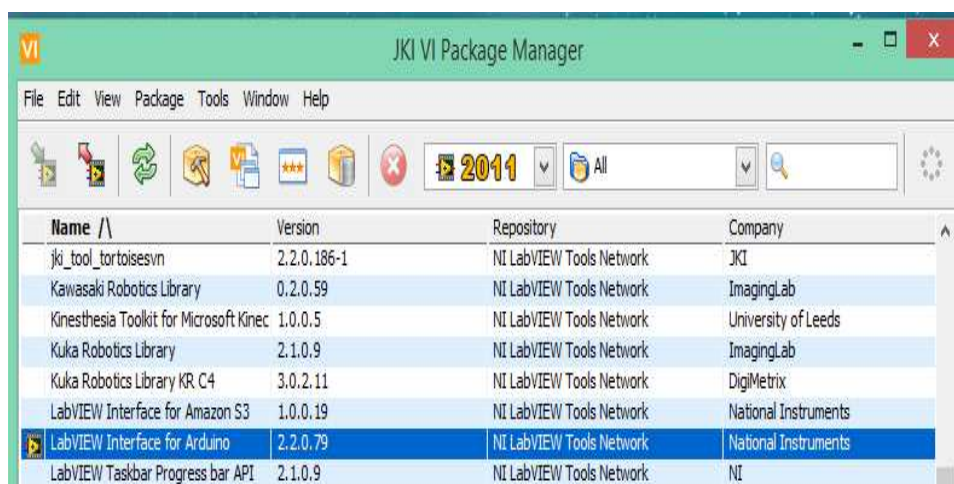


Figura 7.8. Descarga de toolkit interfaz LABVIEW para Arduino.

5. *INSTALACIÓN DEL SOFTWARE ARDUINO IDE 1.0.4.* Se realiza la instalación del entorno de Arduino IDE versión 1.0.4. (Figura 7.9), para realizar la programación de la placa Arduino a través de un puerto serial y para complementar la interfaz Arduino LABVIEW.



Figura 7.9. Arduino IDE 1.0.4.

5. *INSTALACIÓN DEL SOFTWARE X-CTU*. Para realizar la comunicación inalámbrica a través de los dispositivos XBEE, es necesario configurar el firmware de los mismos por medio de este software proporcionado por DIGI y visto en la figura 7.10.

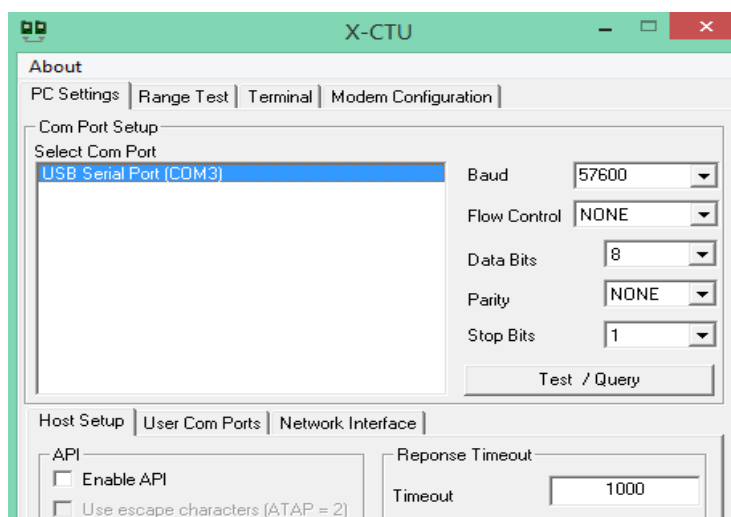


Figura 7.10. Software X.CTU.

Sincronización de la interfaz LABVIEW-ARDUINO-XBEE

Una tarea fundamental para lograr el acoplamiento en la transmisión de datos de las tres plataformas LABVIEW, Arduino y XBEE, es establecer una misma velocidad de transmisión de datos y configurar los parámetros a los que se desea trabajar.

Para lograr este objetivo se requirió conocer a fondo como se realiza la comunicación entre cada una de las plataformas. De esta manera entra a escenario el archivo LIFA_BASE, ubicado en: **C:\Program Files\National**

Instruments\LabVIEW2011\vi.lib\LabVIEWInterfaceforArduino\Firmware\LIFA_Base (Figura 7.11).

Este archivo contiene el firmware entre LABVIEW y Arduino, y está abierto para su libre modificación, dependiendo de la tarea que se desea realizar.

Este es el primer lugar donde se modifica la velocidad de transmisión de datos mejor conocido como "baud rate" (bits/segundo).

También se realiza una mejora en este archivo para sincronizar correctamente la comunicación, ya que inicialmente era necesario utilizar un Reset para su funcionamiento y en ocasiones la comunicación se perdía arrojando un error 5002, 5005 ó 5003.

Estos errores se deben a la falta de sincronización, lo que provoca el desborde de los datos en el buffer, lo que a su vez provoca la pérdida de la transmisión.



```

/*****
**
**  LVFA_Firmware - Provides Basic Arduino Sketch For Interfacing With LabVIEW.
**
**  Written By:    Sam Kristoff - National Instruments
**  Written On:    November 2010
**  Last Updated:  Dec 2011 - Kevin Fort - National Instruments
**
**  This File May Be Modified And Re-Distributed Freely. Original File Content
**  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"

/*****
<

```

Figura 7.11. Archivo LIFA_BASE.

El segundo lugar donde se deben establecer los parámetros deseados es en el puerto virtual a través del controlador VISA como se muestra en la figura 7.12, y también por medio del administrador de dispositivos de Windows 8.1 mostrado en la figura 7.13.

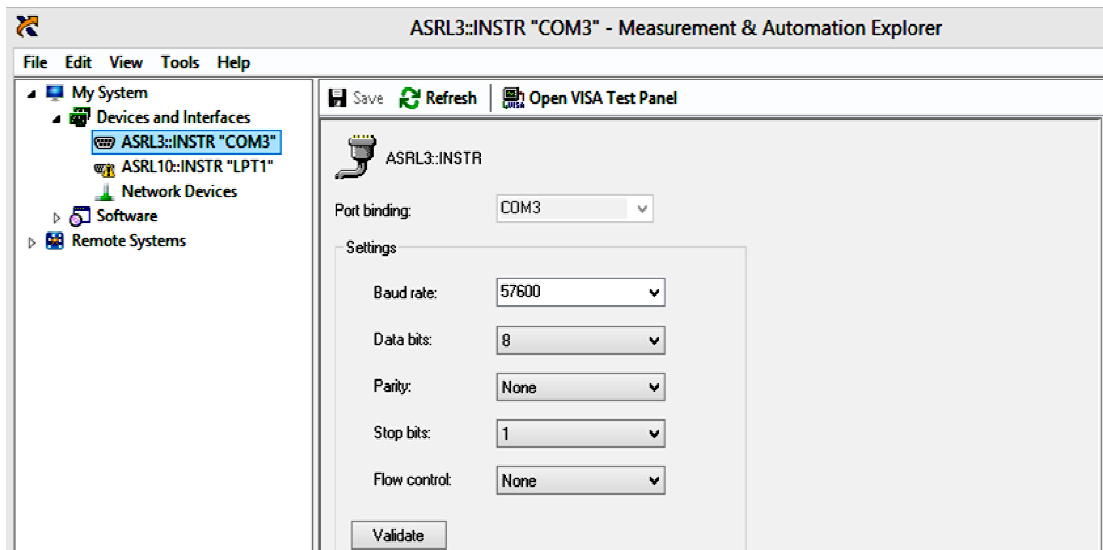


Figura 7.12. Configuración de parámetros a través del controlador VISA NI MAX.

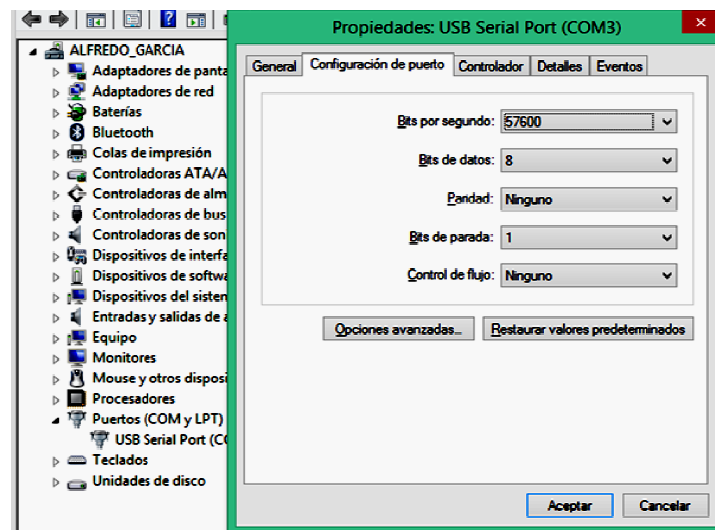


Figura 7.13. Configuración de parámetros a través del administrador de dispositivos de Windows 8.1.

Posteriormente se configuran los parámetros de transmisión en el firmware de los módulos XBEE, por medio del software XCTU como se ilustra en la figura 7.10.

Finalmente el último lugar donde se establecen los parámetros configurados es en el programa realizado en LABVIEW, el cual se describe específicamente más adelante.

7.2 PROCESAMIENTO DIGITAL DE SEÑALES SENSORIALES

Encoder

Para contar el numero de pulsos se desarrolló el siguiente programa en LABVIEW (mostrado en la figura 7.14), utilizando la librería de interfaz para Arduino.

Donde se establece una velocidad de transmisión de 111111 bits/seg., enviando 15 bytes por paquete enviado, a través del puerto virtual 5, hacia una tarjeta Arduino UNO, por medio de una comunicación inalámbrica XBEE.

Todos estos datos se configuran para poder iniciar la comunicación del dispositivo local conectado a una computadora portátil hacia el dispositivo remoto conectado a la tarjeta de datos Arduino UNO.

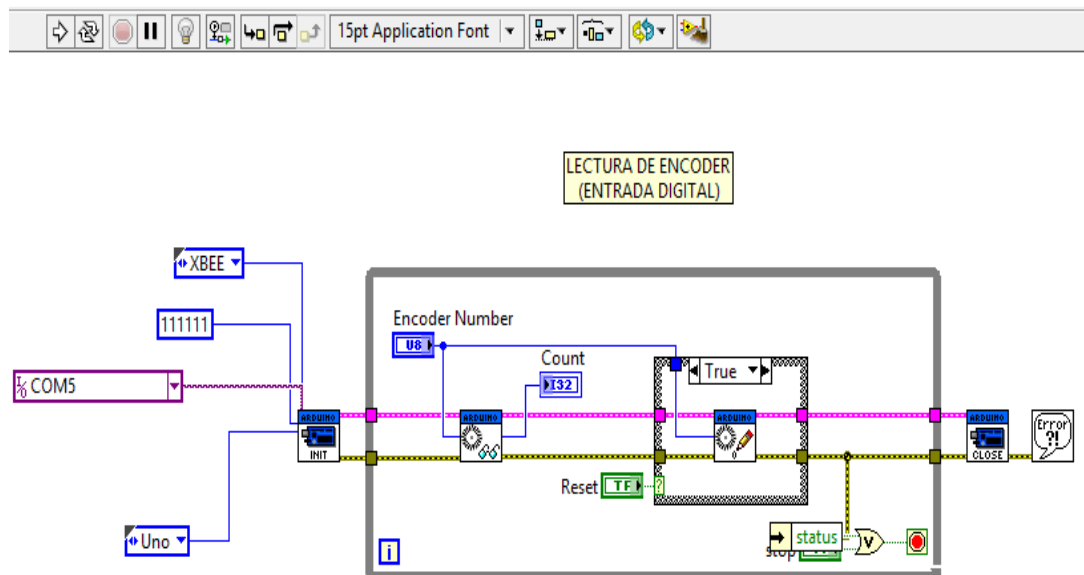


Figura 7.14. Programa para la lectura del Encoder en LABVIEW.

Para obtener la cuenta de los pulsos se utiliza el indicador “count”, mostrado en la figura 7.14. Por otra parte para obtener la velocidad de los pulsos se utilizan los registros de corrimiento del software LABVIEW, los cuales nos permiten mediante una diferencia obtener el numero de pulsos que se han generado por cada ejecución que realiza el ciclo while.

De esta manera cada ejecución está determinada por un temporizador cuyo valor es el tiempo de muestreo de los sensores que equivale a 50 milisegundos.

Una vez que se obtiene la posición en número de pulsos y la velocidad en número de pulsos/periodo de muestreo, se realiza la conversión de pulsos a metros, teniendo en cuenta que el perímetro de cada llanta es de 0.41 metros y que el Encoder nos

proporciona una resolución de 80 revoluciones/vuelta, la relación para obtener la unidad de la posición y velocidad en metros es 1 pulso = 0.005125 metros.

Finalmente los valores de las variables medidas de posición del robot móvil HUMMER H2 se muestran en una interfaz gráfica de usuario, desarrollada en LABVIEW, la cual se ilustra en la figura 7.15.

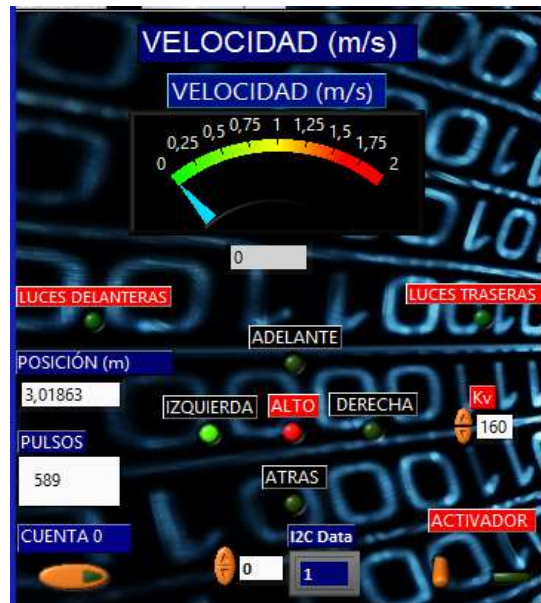


Figura 7.15. Interfaz de usuario del Encoder.

Magnetómetro

Para medir la orientación del móvil en grados se desarrolló el siguiente programa en LABVIEW (mostrado en la figura 7.16), utilizando la librería de interfaz para Arduino con el protocolo de comunicación I2C.

Donde se establece una velocidad de transmisión de 111111 bits/seg, enviando 15 bytes por paquete enviado, a través del puerto virtual 5, hacia una tarjeta Arduino UNO, por medio de una comunicación inalámbrica XBEE.

Todos estos datos se configuran para poder iniciar la comunicación del dispositivo local conectado a una computadora portátil hacia el dispositivo remoto conectado a la tarjeta de datos Arduino UNO.

Es importante señalar que para realizar la interfaz por I2C se configuran 3 registros de inicio (Anexados en el apéndice D) y se selecciona la dirección propia del dispositivo HMC5883L, cuyo valor en este caso es 30.

Una vez que se ha inicializado la comunicación con el magnetómetro, los datos de cada eje (x y z) se reciben en dos paquetes cada uno de 8 bits respectivamente, es decir por los tres ejes se reciben en total 6 paquetes de datos de 8 bits cada uno.

Se inicia el testeo tomando una muestra del magnetómetro utilizando la función $\tan^2(y, x)$, posteriormente se corrigen los valores de los signos de la muestra, después se compensa el error debido a la declinación y finalmente se realiza la conversión de la medida de radianes a grados.

Todo este proceso se muestra en la figura 7.16, desarrollado en el ambiente gráfico de LABVIEW.

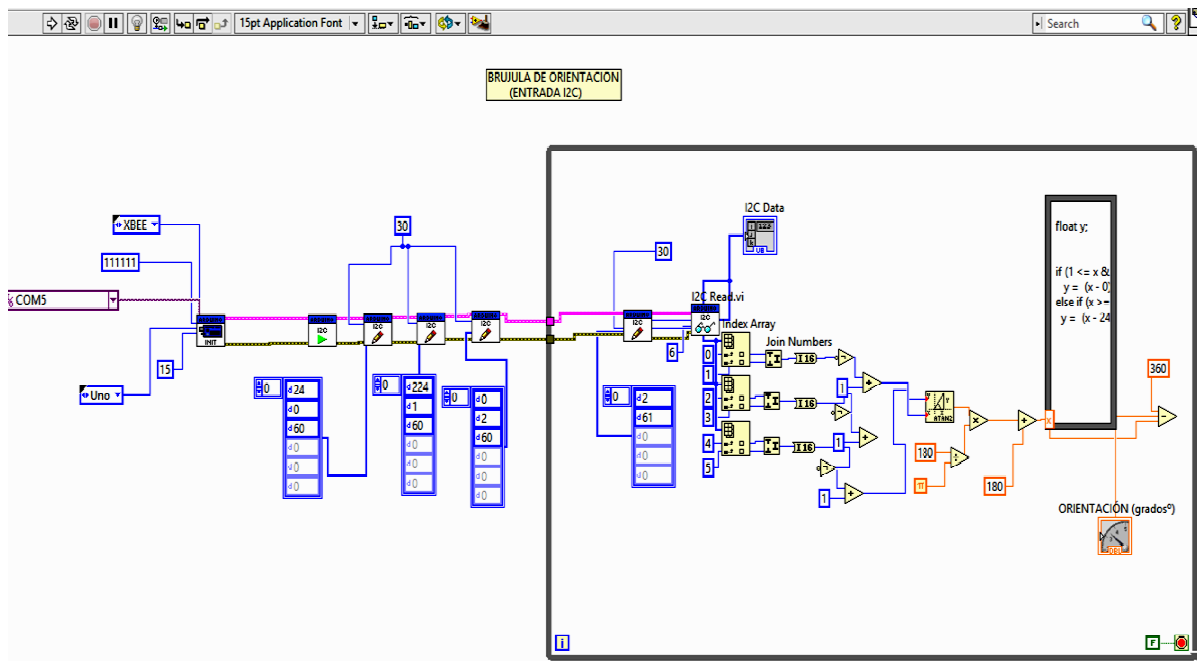


Figura 7.16. Programa para la lectura del Magnetómetro en LABVIEW.

Con el fin de determinar la dirección, también es necesario tener en cuenta el factor de error en la declinación magnética. La declinación magnética se debe tanto a la posición del polo magnético de la tierra en una ubicación diferente que el polo geográfico y diversas condiciones magnéticas en el punto de medición (por ejemplo por la presencia de grandes cantidades de mineral de hierro). Vale la pena señalar que la variación magnética es una variable de parámetros en el tiempo, ya que los polos magnéticos de la Tierra están en constante movimiento.

El valor de la declinación magnética actual se puede encontrar en mapas especiales, mapas magnéticos y de navegación. Actualmente encontramos el mapa correspondiente en la siguiente página de internet: <http://magnetic-declination.com/>, como se observa en la figura 7.17.

La exactitud del magnetómetro HMC5883L es de $\pm 1.5^\circ$, por lo que es un error mínimo para el control de la orientación del robot móvil.



Figura 7.17. Mapa del valor de la declinación magnética.

En resumen, se ingresa a los registros de configuración de la brújula HMC5883L, para modificar los parámetros de respuesta del sensor y a la vez iniciar la comunicación con el mismo. Posteriormente se obtienen los datos en bruto proporcionados por el sensor, los cuales son procesados mediante el algoritmo de la figura 7.16 hasta convertirlos en grados.

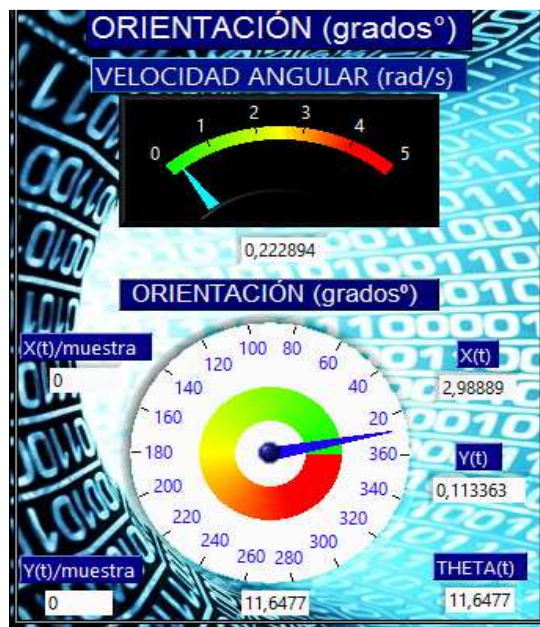


Figura 7.18. Interfaz de usuario del Magnetómetro.

Una vez obtenida la lectura en grados, se suma el valor de la declinación magnética (mostrado en la figura 7.17) correspondiente a la zona del estado de Puebla donde se realiza el trabajo. Para obtener la velocidad angular de orientación (w) se utiliza la

misma técnica que se empleó en la velocidad lineal, utilizando los registros de corrimiento.

Finalmente los valores de las variables medidas de orientación del robot móvil HUMMER H2 se muestran en una interfaz gráfica de usuario, desarrollada en LABVIEW, la cual se ilustra en la figura 7.18.

Potenciómetro

Para medir la dirección del móvil en grados se desarrolló el siguiente programa en LABVIEW (mostrado en la figura 7.19), utilizando la librería de interfaz para Arduino con el protocolo de comunicación serial.

Donde se establece una velocidad de transmisión de 111111 bits/seg., enviando 15 bytes por paquete enviado, a través del puerto virtual 5, hacia una tarjeta Arduino UNO, por medio de una comunicación inalámbrica XBEE.

Todos estos datos se configuran para poder iniciar la comunicación del dispositivo local conectado a una computadora portátil hacia el dispositivo remoto conectado a la tarjeta de datos Arduino UNO.

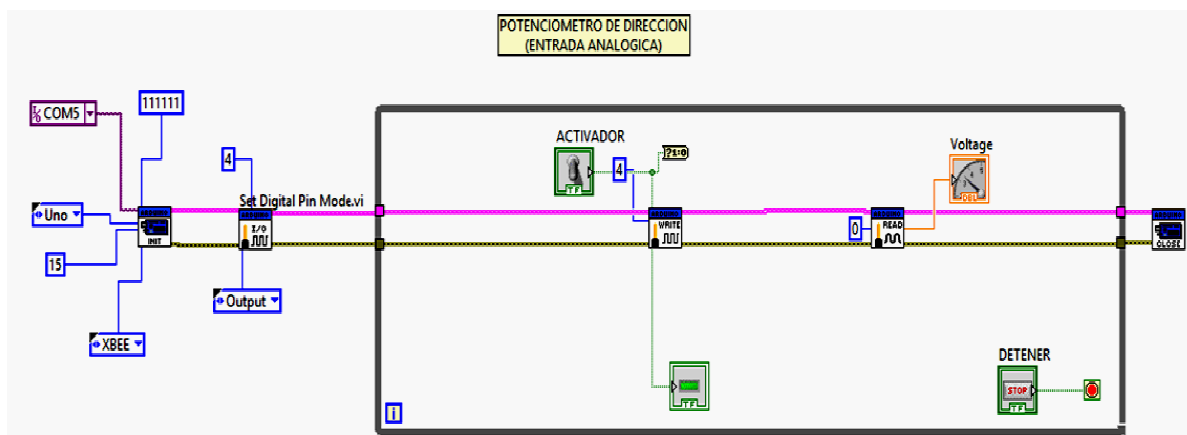


Figura 7.19. Programa para la lectura del Potenciómetro en LABVIEW.

La lectura del voltaje varía de 0.3724V a 0.7252V, con respecto a la posición del eje de las ruedas delanteras, el valor del voltaje variable se lee de forma directa, se mide el valor de voltaje cuando el eje está a centrado (0°), el cual resulta ser de 0.4753V, a partir de este valor se normaliza la escala “Voltaje – Grados”, tanto para el sentido de la izquierda como para el sentido de la derecha, tomando como suposición que el ángulo máximo de giro para ambos sentido es de 30° . Por otro lado también se obtiene la velocidad angular de dirección del eje delantero del robot móvil, utilizando la técnica de diferenciación con los registros de corrimiento de LABVIEW. Finalmente los valores de las variables medidas de dirección de robot móvil HUMMER H2 se muestran en una interfaz gráfica de usuario, desarrollada en LABVIEW, la cual se ilustra en la figura 7.20.

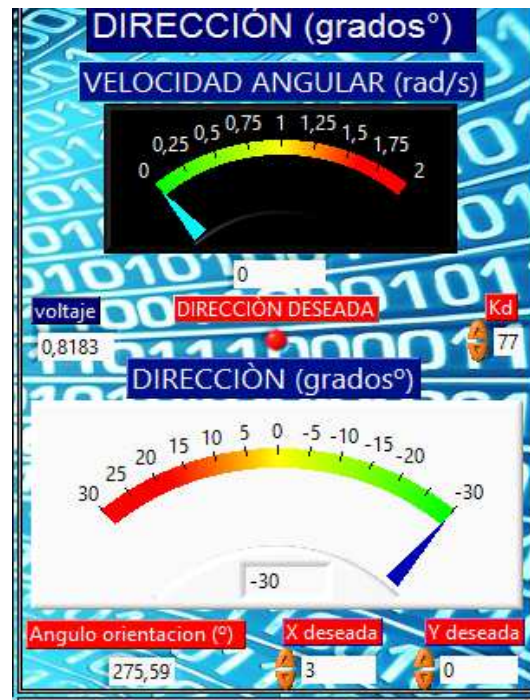


Figura 7.20. Interfaz de usuario del Potenciometro.

7.3 PROCESAMIENTO DIGITAL DEL ALGORITMO DE CONTROL

Modelo Cinemático

Mediante la lectura de las magnitudes físicas: Posición, velocidad lineal, ángulo de orientación, velocidad angular de orientación, ángulo de dirección y velocidad angular de dirección, se implementa el modelo cinemático del robot móvil HUMMER H2 (Ecuaciones 4.3, 4.4, 4.5 y 4.6) en el ambiente de programación de LABVIEW.

En cada periodo de muestreo (cada 50 milisegundos), se calculan las coordenadas del robot móvil HUMMER H2 (x , y , θ) que corresponden a la posición del robot móvil en el eje “x”, la posición del robot móvil en el eje “y” y al ángulo de orientación del robot móvil en el plano (x , y).

A este sistema de coordenadas se agrega una cuarta variable correspondiente a la dirección de las ruedas del robot móvil (δ_D)

Con el modelo cinemático se realiza un plano cartesiano virtual en 2D, el cual nos permite visualizar la trayectoria del robot móvil calculada en cada periodo de muestreo, esto nos permite monitorear la trayectoria del vehículo y también al mismo tiempo la trayectoria de referencia que deseemos que siga el móvil. En la figura 7.21 se ilustra el plano virtual (x , y), donde la línea punteada azul representa la ruta seguida por el robot móvil HUMMER H2, mientras que la línea roja representa la trayectoria de referencia propuesta por el programador.

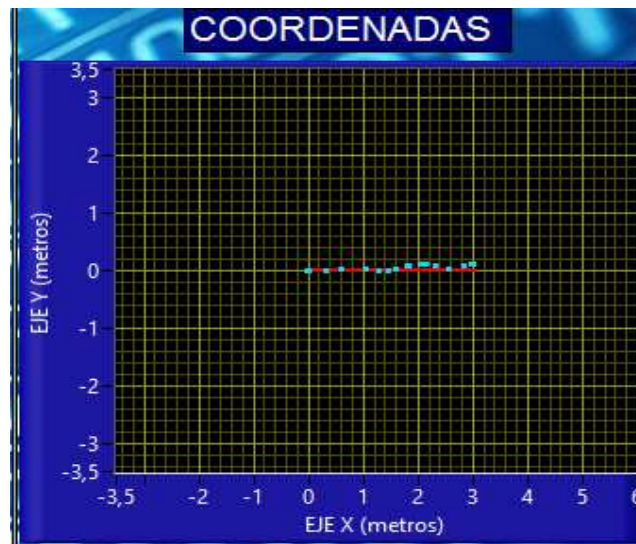


Figura 7.21. Plano virtual (x, y) desarrollado con el modelo cinemático del vehículo.

A la par del plano virtual (x, y), también se desarrollan otros dos planos virtuales de representación numérica.

El plano virtual mostrado en la figura 7.22 representa el valor de cada uno de los ejes coordenados de referencia y el valor de cada uno de los ejes coordenados por los que pasa el robot móvil.

La línea amarilla representa el valor de referencia en el eje “y”, la línea roja representa el valor de referencia en el eje “x”, las dos líneas azules son los valores de trayectoria del robot móvil en cada uno de los ejes.

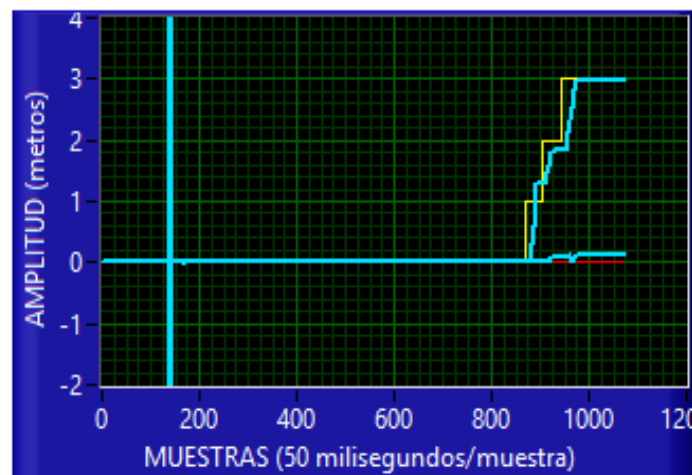


Figura 7.22. Plano virtual que representa la magnitud de referencia y del vehículo en los ejes “x” y “y”.

El plano virtual mostrado en la figura 7.23 representa el error de cada uno de los ejes coordenados del robot móvil, donde la línea amarilla representa el valor en el eje “x” y la línea roja representa el valor en el eje “y”.

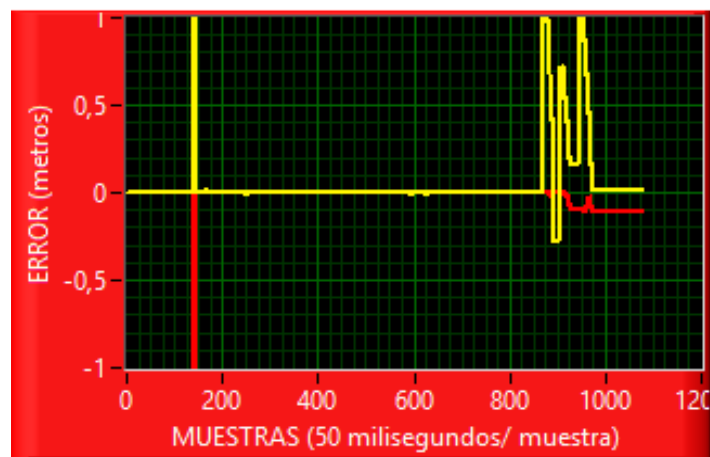


Figura 7.23. Plano virtual que representa el error del vehículo en los ejes “x” y “y”.

Generador de Trayectorias

En el control de seguimiento de trayectoria se implementó el algoritmo para producir el error entre la posición deseada y la posición actual del robot móvil, utilizando las ecuaciones (5.22, 5.23, 5.24, 5.25 y 5.26).

Mediante este principio se proponen dos estrategias de seguimiento de trayectoria.

CONTROL PUNTO A PUNTO. En esta estrategia se define una posición de destino, donde el robot móvil parte del origen del sistema coordenado (0, 0) y tiende a la posición comandada desde la interfaz de usuario implementada en LABVIEW.

CONTROL UTILIZANDO CURVAS PARAMÉTRICAS. En esta estrategia de control interviene una variable adicional denominada tiempo (t), donde se establecen funciones matemáticas dependientes del tiempo las cuales proporcionan una serie de 100 puntos deseados consecutivamente, los cuales va siguiendo el robot móvil como trayectoria de referencia.

En la figura 7.24 se muestra el algoritmo implementado como generador de trayectorias, utilizando el módulo de Mathscript en LABVIEW.

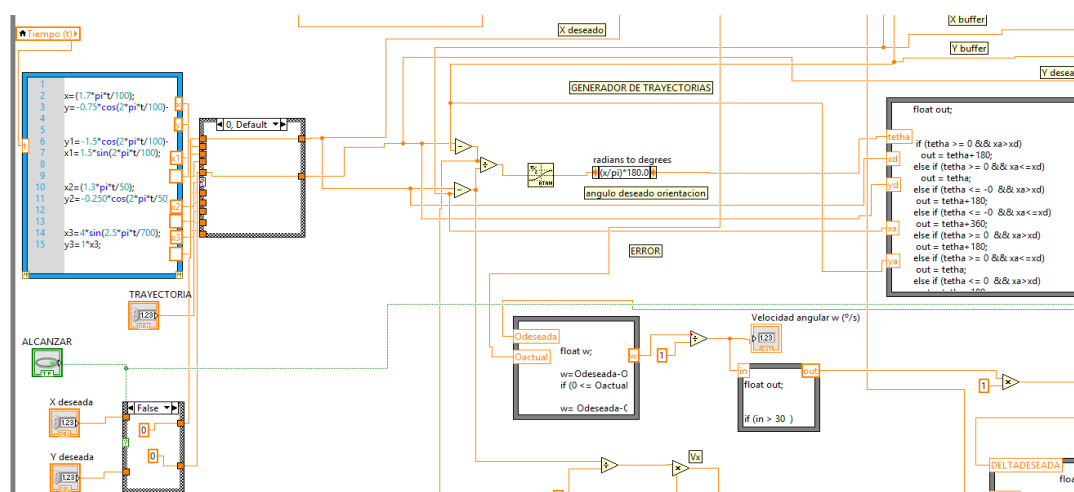


Figura 7.24. Programa para el generador de trayectorias en LABVIEW.

La interfaz de usuario del generador de trayectorias se muestra en la figura 7.25, donde el botón “ALCANZAR” es el activador del tiempo para que se genere la trayectoria deseada.



Figura 7.25. Interfaz de usuario para el generador de trayectorias.

Al ejecutar el algoritmo de control los datos de la trayectoria de referencia y la trayectoria seguida por el robot móvil se capturan en cada periodo de muestreo y se almacenan en la tabla de la figura 7.26, donde posteriormente se genera un archivo automático en Excel con toda la información recabada en cada prueba, para realizar un análisis de su desempeño.

X Robot	Y Robot	X Trayectoria	Y Trayectoria
2.988890	0.113363	3.000000	0.000000
2.988890	0.113363	3.000000	0.000000
2.988890	0.113363	3.000000	0.000000
2.988890	0.113363	3.000000	0.000000
2.988890	0.113363	3.000000	0.000000
2.988890	0.113363	3.000000	0.000000
2.988890	0.113363	3.000000	0.000000
2.988890	0.113363	3.000000	0.000000
2.988890	0.113363	3.000000	0.000000
2.988890	0.113363	3.000000	0.000000
2.988890	0.113363	3.000000	0.000000
2.988890	0.113363	3.000000	0.000000
2.988890	0.113363	3.000000	0.000000
2.988890	0.113363	3.000000	0.000000
2.988890	0.113363	3.000000	0.000000

Figura 7.26. Coordenadas almacenadas en cada periodo de muestreo.

7.4 PROCESAMIENTO DIGITAL DE LAS SALIDAS DE CONTROL

Controlador de dirección

Tomando como referencia la variable del ángulo de dirección del vehículo “ δ_D ”, se obtiene el error mediante la diferencia entre la dirección deseada y la dirección actual. En la ley de control se incluye una variable llamada K_{DELTA} , la cual es la ganancia proporcional del controlador de dirección.

El controlador utilizado es una tangente hiperbólica la cual nos ofrece una respuesta suave en los límites del actuador y se describe en la ecuación (7.1).

$$\text{Controlador} = (K_{\text{DELTA}})\tanh(1*(\text{DELTADESEADA}-\text{DELTA ACTUAL})/60); \quad (7.1)$$

Mientras que la salida transformada en el valor de PWM de dirección está definida por la ecuación 7.2.

$$\text{PWM} = \text{abs}(\text{Controlador}); \quad (7.2)$$

En la figura 7.27 se ilustra la implementación del controlador de dirección en LABVIEW.

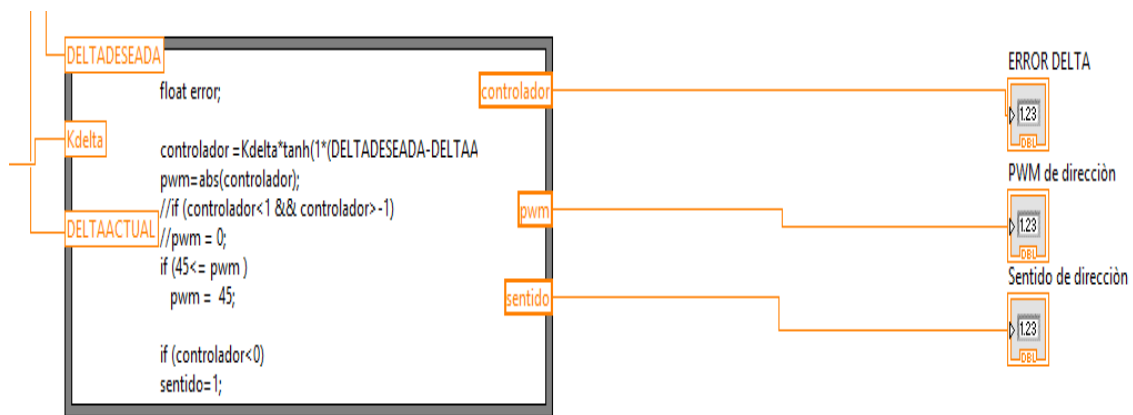


Figura 7.27. Implementación de la ley de control de dirección.

Controlador de tracción

Tomando como referencia la variable de velocidad lineal “v”, se obtiene el error mediante la diferencia entre la velocidad deseada y la velocidad actual.

En la ley de control se incluye una variable llamada K_v , la cual es la ganancia proporcional del controlador de tracción.

El controlador utilizado es una tangente hiperbólica la cual nos ofrece una respuesta suave en los límites del actuador y se describe en la ecuación (7.3).

$$\text{Controlador2} = K_v * \tanh(1*v); \quad (7.3)$$

Mientras que la salida transformada en el valor de PWM de tracción está definida por la ecuación 7.4.

$$\text{PWM} = \text{abs}(\text{Controlador2}); \quad (7.4)$$

En la figura 7.28 se ilustra la implementación del controlador de tracción en LABVIEW.

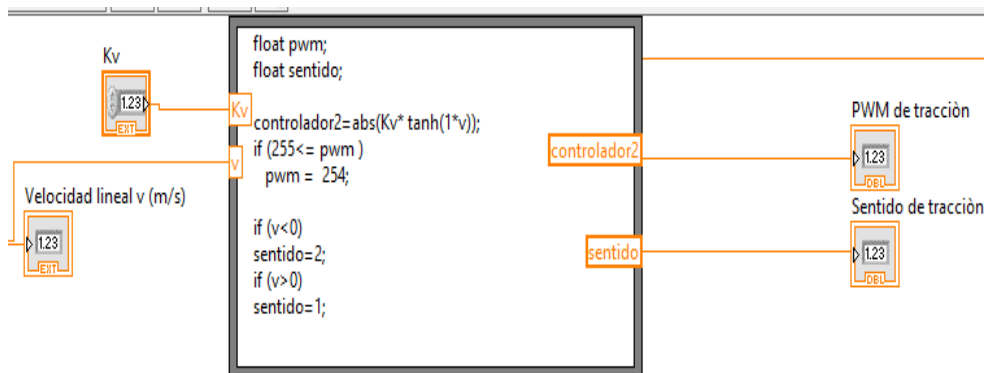


Figura 7.28. Implementación de la ley de control de tracción.

En la figura 7.29 se muestra la interfaz de usuario correspondiente a las salidas de control: PWM de dirección, Sentido de dirección, PWM de tracción y Sentido de tracción.



Figura 7.29. Interfaz de usuario para las salidas de control.

Algunas de las variables utilizadas dentro del algoritmo de control como: velocidad angular, velocidad lineal, ángulo delta, y error delta se muestran en la interfaz de usuario (Figura 7.30) para obtener mayor información del desempeño del robot móvil HUMMER H2.



Figura 7.30. Variables de control mostradas en la interfaz de usuario.

7.5 DESCRIPCIÓN GENERAL DEL SISTEMA DE CONTROL

En la figura 7.31, se ilustra la descripción en bloques de la implementación del sistema de control del robot móvil HUMMER H2.

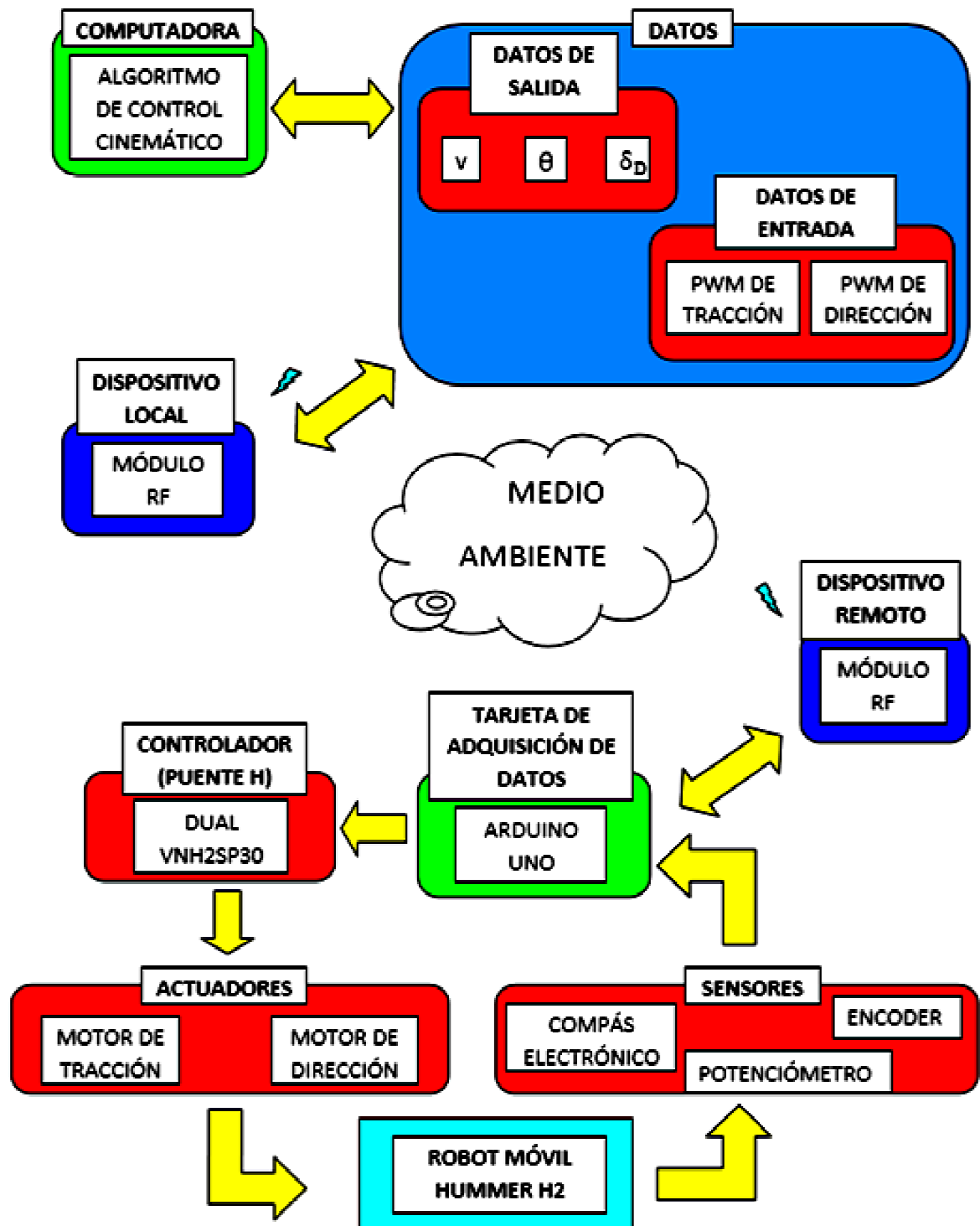


Figura 7.31. Vista a bloques del sistema de control del robot móvil HUMMER H2.

CAPÍTULO

8 PRUEBAS EXPERIMENTALES

8.1 *TRAYECTORIAS CON CURVAS PARAMETRIZADAS*

Las pruebas realizadas con el robot móvil HUMMER H2, se llevaron a cabo en el laboratorio de la MCEA, en la Facultad de Electrónica de la BUAP.

Para realizar dichas pruebas se ocupó un espacio de 5.50 x 3 metros, ilustrado en la figura 8.1, donde el robot móvil seguirá las trayectorias comandadas por el usuario, tomando en cuenta las restricciones del espacio de trabajo.



Figura 8.1. Espacio de trabajo para realizar las pruebas del robot móvil.

Para realizar las pruebas en el espacio de trabajo, se coloca una cámara digital para grabar cada una de las trayectorias seguidas por el robot móvil como se muestra en la figura 8.2, mientras que la computadora portátil se coloca fuera del espacio de trabajo para permitir al usuario monitorear la trayectoria que va generando el robot móvil en el piso y también la trayectoria graficada en la computadora a través de la interfaz de usuario.



Figura 8.2. Sistema implementado para el monitoreo del vehículo.

Arco extendido

La primera curva paramétrica que se aplicará como trayectoria de referencia para el robot móvil HUMMER H2 es un arco extendido, cuyo radio de curvatura es de 1.5 metros.

Esta curva paramétrica está definida por la ecuación (8.1) para el eje “x” y por la ecuación (8.2) para el eje “y”.

$$x = (1.7\pi t/100) \quad (8.1)$$

$$y = -0.75\cos(2\pi t/100) + 0.75 \quad (8.2)$$

La variable “t” representa el tiempo en el cual evoluciona la construcción de la curva paramétrica. Cada segundo el generador de trayectorias comanda 1 punto de la curva paramétrica, el total de puntos generados para formar esta trayectoria es de 100.

Para realizar las pruebas de seguimiento de trayectoria se elige el origen del plano cartesiano dentro del espacio de trabajo, donde el robot móvil iniciara su recorrido, ya que de esto definirá su ubicación inicial.

También se debe definir su posición inicial, para lo cual se emplea la referencia del norte magnético de la tierra, utilizando el magnetómetro HMC5883L, se alinea de modo que la posición del robot móvil este en dirección al eje “x” positivo de las abscisas.

Una vez que se ha definido su ubicación y posición inicial dentro del espacio de trabajo, se inicializa la comunicación inalámbrica desde el dispositivo local, por medio de la interfaz de usuario programada en LABVIEW.

PRIMERA PRUEBA. En la figura 8.3 se muestra la trayectoria generada por el robot móvil HUMMER H2 (línea negra) en la primera prueba, teniendo como trayectoria deseada el arco extendido.



Figura 8.3. Trayectoria generada en la primera prueba del arco extendido.

En la figura 8.4 se ilustra la interfaz de usuario después de haberse realizado la primera prueba, donde se aprecia la trayectoria realizada por el robot móvil (línea punteada azul) comparada con la trayectoria de referencia (línea roja).

También nos proporciona información sobre el recorrido del robot móvil como: la distancia total recorrida por el vehículo, las coordenadas de la ubicación final del robot móvil, el ángulo de orientación final del robot móvil, la gráfica de los valores de referencia y seguidos por el vehículo en cada uno de los ejes coordenados y la gráfica del error en el seguimiento de la trayectoria. La ejecución del algoritmo de control se realiza con un tiempo de muestreo de 50 milisegundos.

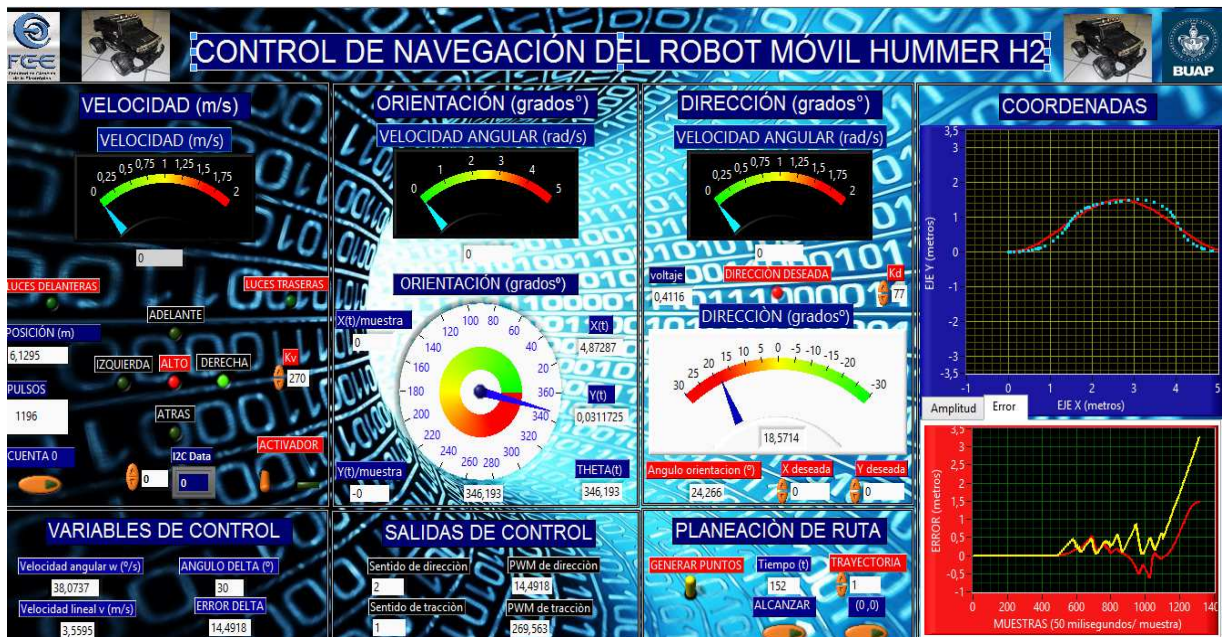


Figura 8.4. Interfaz de usuario en la primera prueba del arco extendido.

SEGUNDA PRUEBA. En la figura 8.5 se muestra la trayectoria generada por el robot móvil HUMMER H2 en la segunda prueba (línea roja), teniendo como trayectoria deseada el arco extendido.



Figura 8.5. Trayectoria generada en la segunda prueba del arco extendido.

En la figura 8.6 se ilustra la interfaz de usuario después de haberse realizado la segunda prueba, donde se aprecia la trayectoria realizada por el robot móvil (línea punteada azul) comparada con la trayectoria de referencia (línea roja).

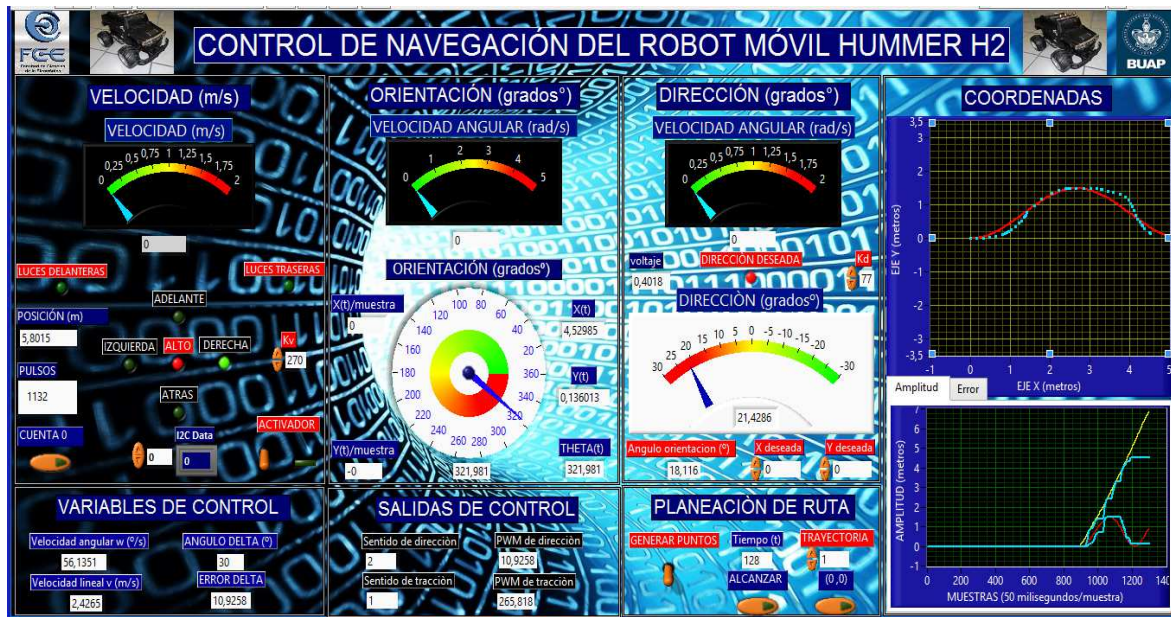


Figura 8.6. Interfaz de usuario en la segunda prueba del arco extendido.

TERCERA PRUEBA. En la figura 8.7 se muestra la trayectoria generada por el robot móvil HUMMER H2 en la tercera prueba (línea verde), teniendo como trayectoria deseada el arco extendido.



Figura 8.7. Trayectoria generada en la tercera prueba del arco extendido.

En la figura 8.8 se ilustra la interfaz de usuario después de haberse realizado la tercera prueba, donde se aprecia la trayectoria realizada por el robot móvil (línea punteada azul) comparada con la trayectoria de referencia (línea roja).

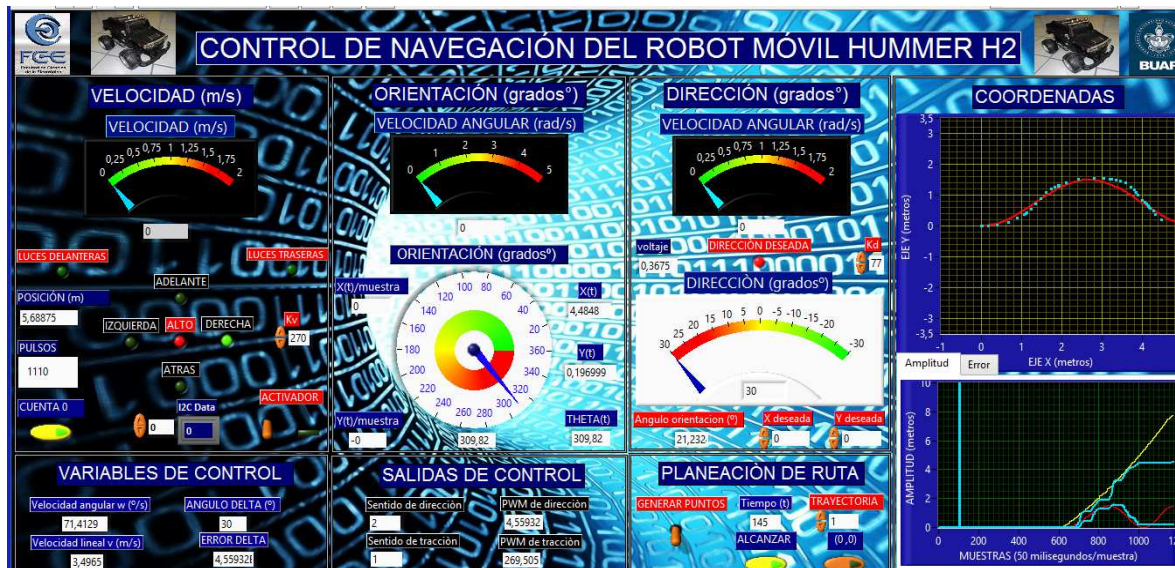


Figura 8.8. Interfaz de usuario en la tercera prueba del arco extendido.

CUARTA PRUEBA. En la figura 8.9 se muestra la trayectoria generada por el robot móvil HUMMER H2 en la cuarta prueba (línea naranja), teniendo como trayectoria deseada el arco extendido.

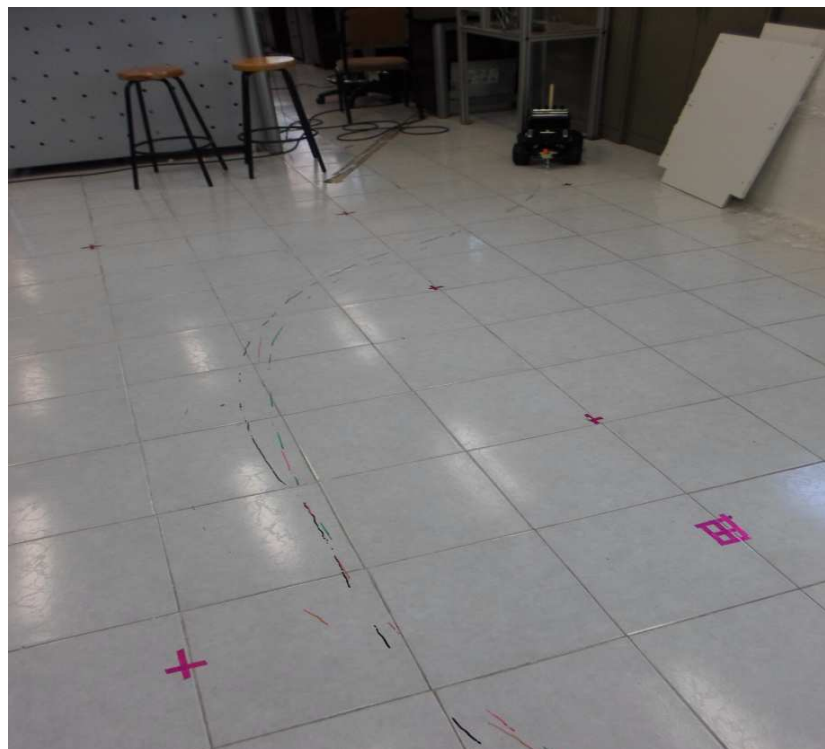


Figura 8.9. Trayectoria generada en la cuarta prueba del arco extendido.

En la figura 8.10 se ilustra la interfaz de usuario después de haberse realizado la cuarta prueba, donde se aprecia la trayectoria realizada por el robot móvil (línea punteada azul) comparada con la trayectoria de referencia (línea roja).

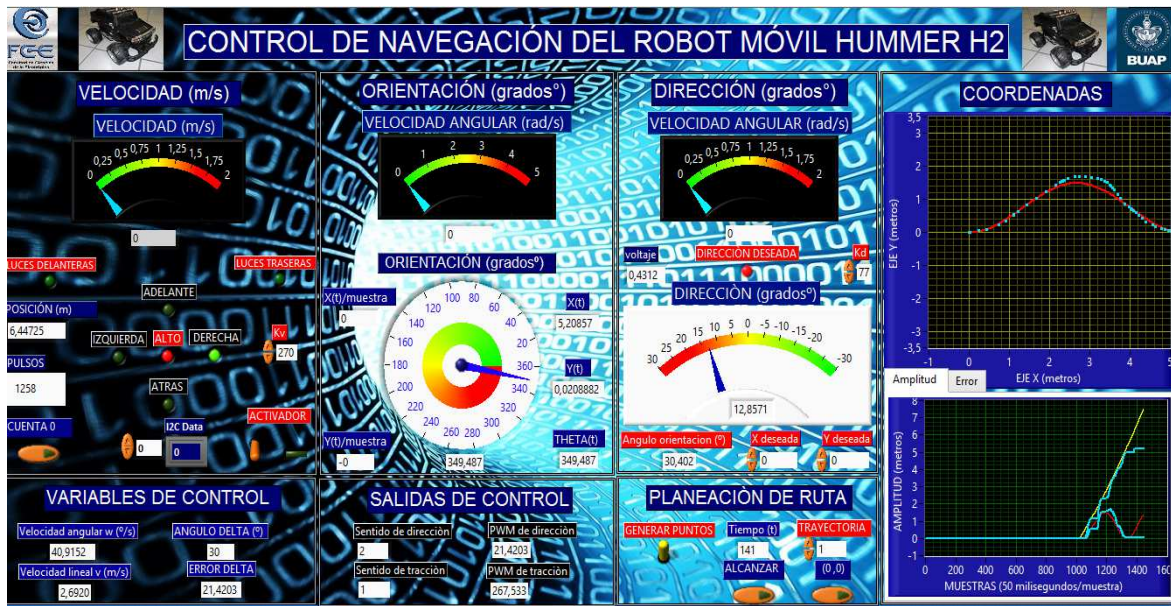


Figura 8.10. Interfaz de usuario en la cuarta prueba del arco extendido.

Media circunferencia

La segunda curva paramétrica que se aplicará como trayectoria de referencia para el robot móvil HUMMER H2 es la mitad de una circunferencia, cuyo radio de curvatura es de 1.5 metros.

Esta curva paramétrica está definida por la ecuación (8.3) para el eje “x” y por la ecuación (8.4) para el eje “y”.

$$x = 1.5 \sin(2\pi t/100) \quad (8.3)$$

$$y = -1.5 \cos(2\pi t/100) + 1.5 \quad (8.4)$$

La variable “t” representa el tiempo en el cual evoluciona la construcción de la curva paramétrica. Cada segundo el generador de trayectorias comanda 1 punto de la curva paramétrica, el total de puntos generados para formar esta trayectoria es de 50.

PRIMERA PRUEBA. En la figura 8.11 se muestra la trayectoria generada por el robot móvil HUMMER H2 (línea verde) en la primera prueba, teniendo como trayectoria deseada la mitad de una circunferencia.

En la figura 8.12 se ilustra la interfaz de usuario después de haberse realizado la primera prueba, donde se aprecia la trayectoria realizada por el robot móvil (línea punteada azul) comparada con la trayectoria de referencia (línea roja).



Figura 8.11. Trayectoria generada en la primera prueba de la mitad de circunferencia.

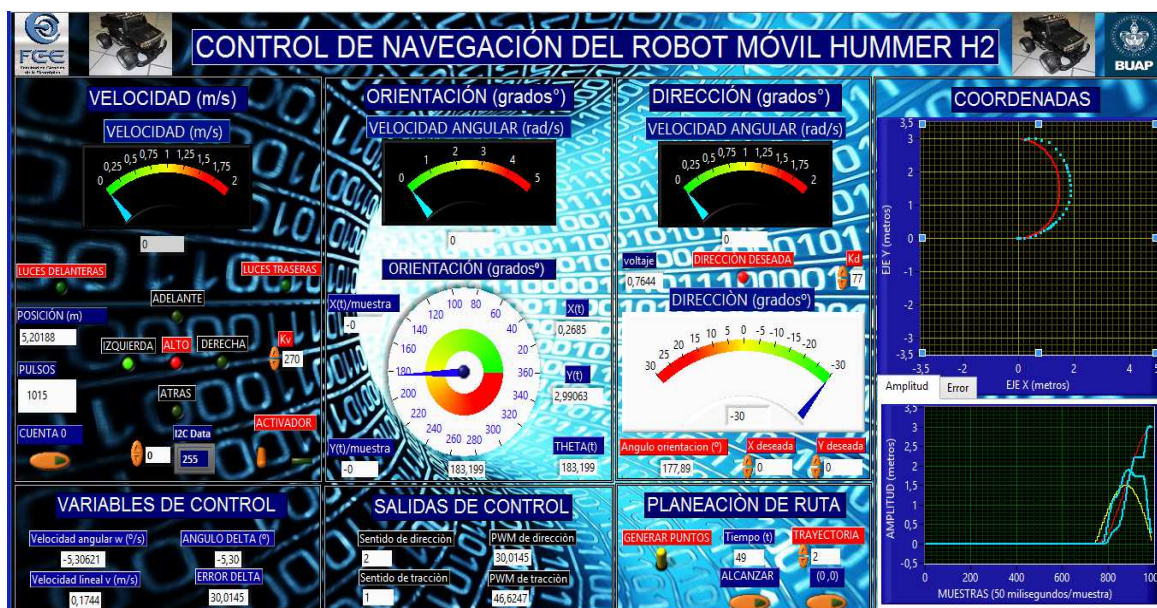


Figura 8.12. Trayectoria generada en la primera prueba de la mitad de circunferencia.

SEGUNDA PRUEBA. En la figura 8.13 se muestra la trayectoria generada por el robot móvil HUMMER H2 (línea roja) en la segunda prueba, teniendo como trayectoria deseada la mitad de una circunferencia.

En la figura 8.14 se ilustra la interfaz de usuario después de haberse realizado la segunda prueba, donde se aprecia la trayectoria realizada por el robot móvil (línea punteada azul) comparada con la trayectoria de referencia (línea roja).

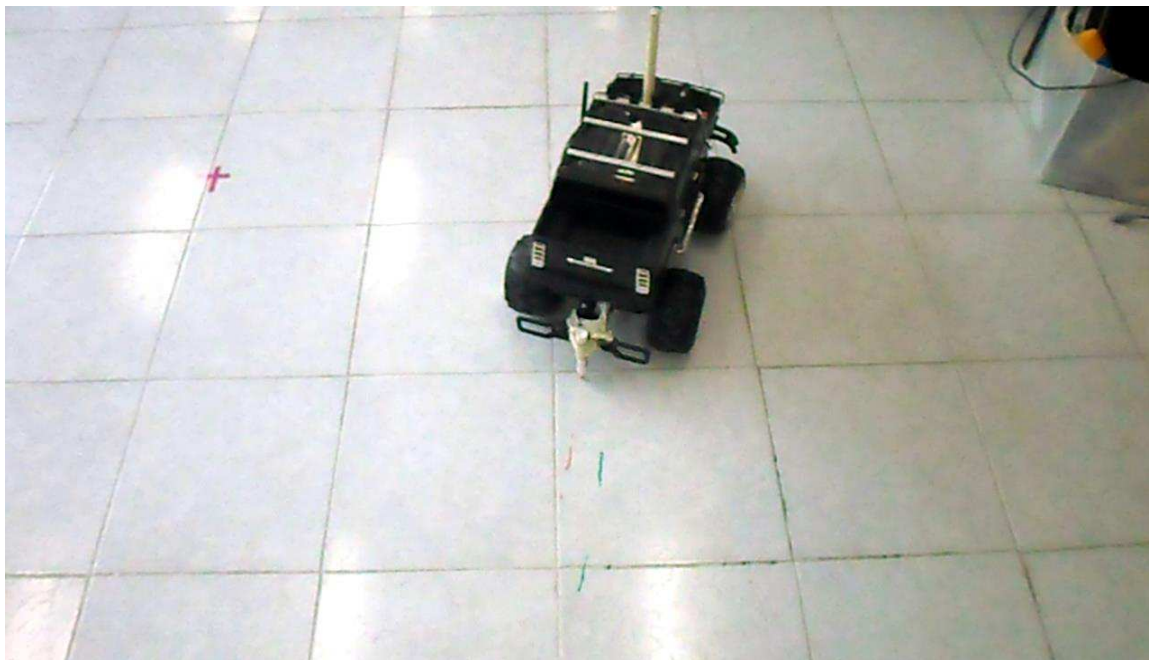


Figura 8.13. Trayectoria generada en la segunda prueba de la mitad de circunferencia.

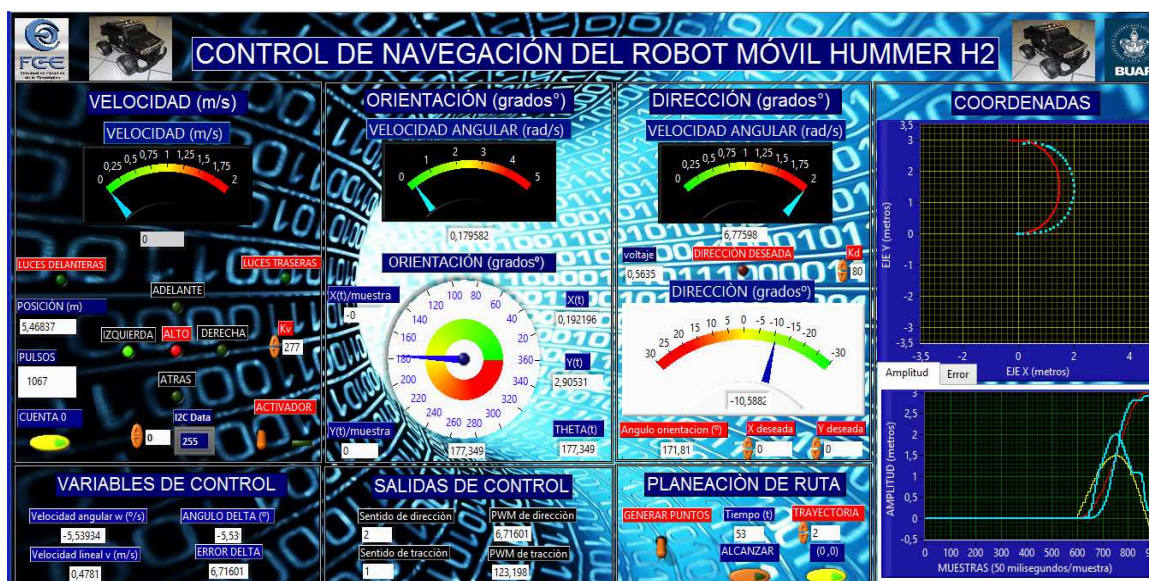


Figura 8.14. Trayectoria generada en la segunda prueba de la mitad de circunferencia.

TERCERA PRUEBA. En la figura 8.15 se muestra la trayectoria generada por el robot móvil HUMMER H2 (línea naranja) en la tercera prueba, teniendo como trayectoria deseada la mitad de una circunferencia.

En la figura 8.16 se ilustra la interfaz de usuario después de haberse realizado la tercera prueba, donde se aprecia la trayectoria realizada por el robot móvil (línea punteada azul) comparada con la trayectoria de referencia (línea roja).



Figura 8.15. Trayectoria generada en la tercera prueba de la mitad de circunferencia.

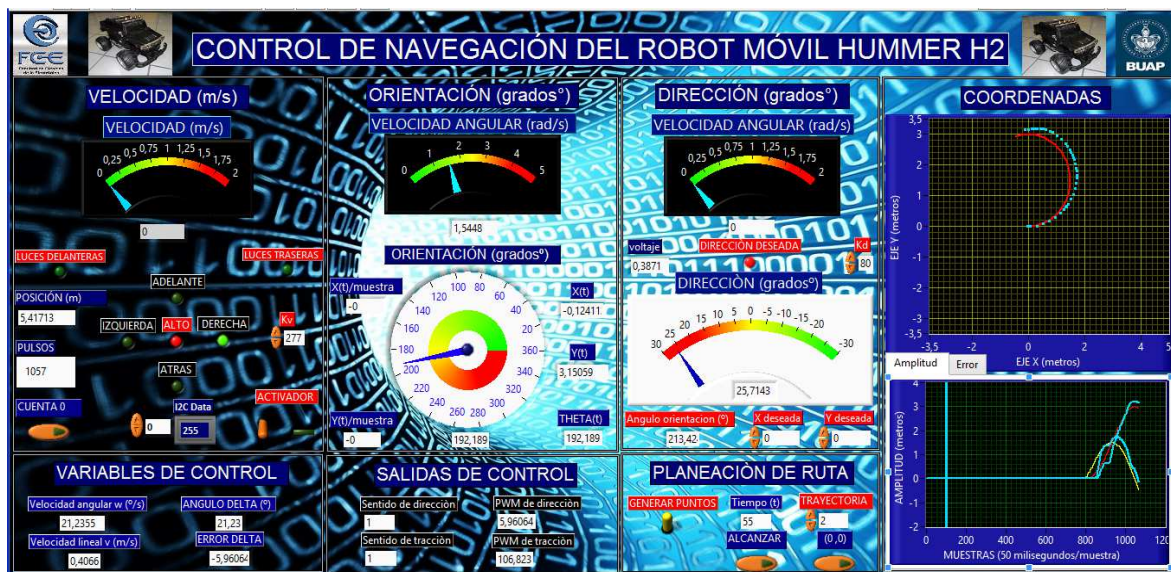


Figura 8.16. Trayectoria generada en la tercera prueba de la mitad de circunferencia.

CUARTA PRUEBA. En la figura 8.17 se muestra la trayectoria generada por el robot móvil HUMMER H2 (línea negra) en la cuarta prueba, teniendo como trayectoria deseada la mitad de una circunferencia.

En la figura 8.18 se ilustra la interfaz de usuario después de haberse realizado la cuarta prueba, donde se aprecia la trayectoria realizada por el robot móvil (línea punteada azul) comparada con la trayectoria de referencia (línea roja).



Figura 8.17. Trayectoria generada en la cuarta prueba de la mitad de circunferencia.

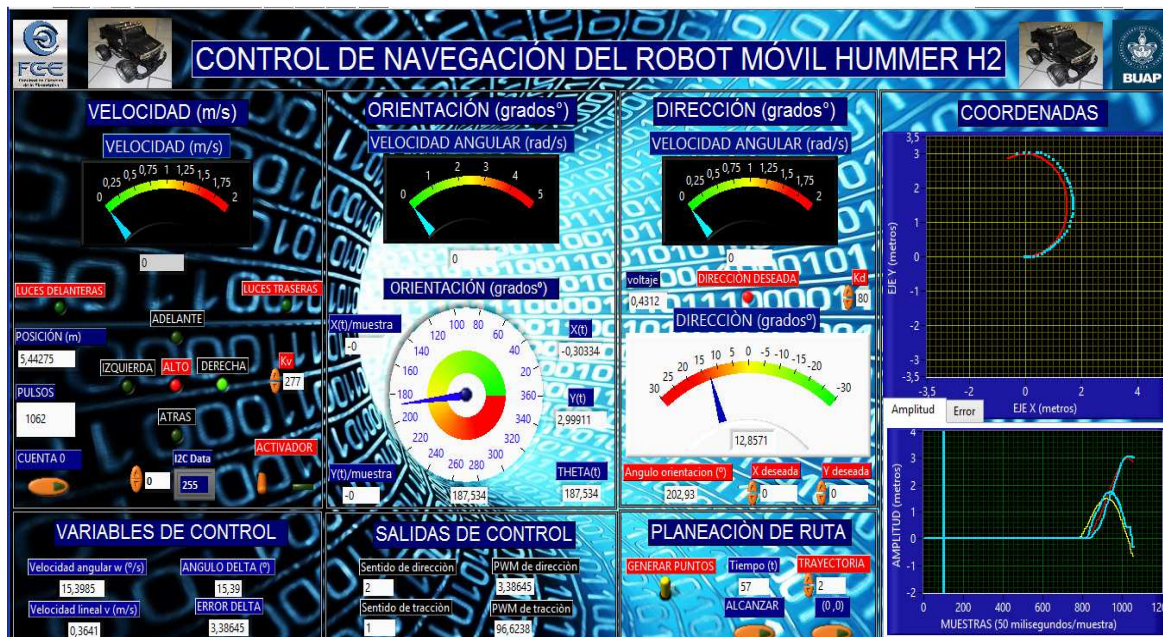


Figura 8.18. Trayectoria generada en la cuarta prueba de la mitad de circunferencia.

Senoidal

La tercera curva paramétrica que se aplicará como trayectoria de referencia para el robot móvil HUMMER H2 es una curva senoidal, cuyo radio de curvatura es de 0.5 metros.

Esta curva paramétrica está definida por la ecuación (8.5) para el eje “x” y por la ecuación (8.6) para el eje “y”.

$$x = (1.3\pi t/50) \quad (8.5)$$

$$y = -0.250\cos(2\pi t/50) + 0.250 \quad (8.6)$$

La variable “t” representa el tiempo en el cual evoluciona la construcción de la curva paramétrica. Cada segundo el generador de trayectorias comanda 1 punto de la curva paramétrica, el total de puntos generados para formar esta trayectoria es de 100.

PRIMERA PRUEBA. En la figura 8.19 se muestra la trayectoria generada por el robot móvil HUMMER H2 (línea verde) en la primera prueba, teniendo como trayectoria deseada la curva senoidal.

En la figura 8.20 se ilustra la interfaz de usuario después de haberse realizado la primera prueba, donde se aprecia la trayectoria realizada por el robot móvil (línea punteada azul) comparada con la trayectoria de referencia (línea roja).



Figura 8.19. Trayectoria generada en la primera prueba de la curva senoidal.

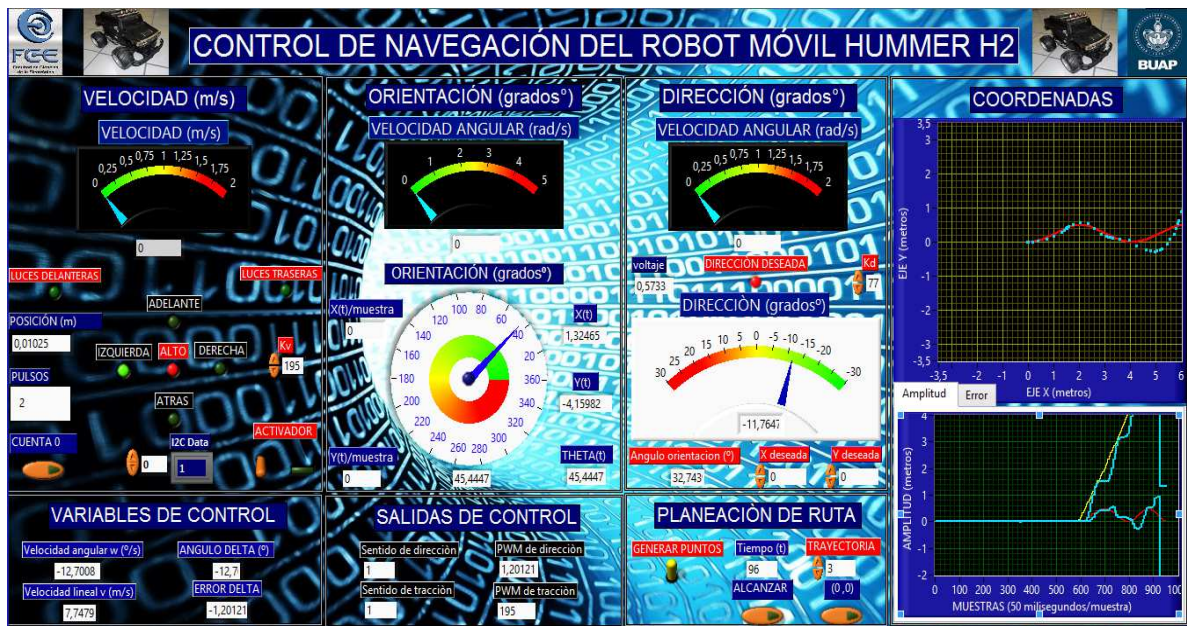


Figura 8.20. Trayectoria generada en la primera prueba de la curva senoidal.

SEGUNDA PRUEBA. En la figura 8.21 se muestra la trayectoria generada por el robot móvil HUMMER H2 (línea roja) en la segunda prueba, teniendo como trayectoria deseada la curva senoidal.

En la figura 8.22 se ilustra la interfaz de usuario después de haberse realizado la segunda prueba, donde se aprecia la trayectoria realizada por el robot móvil (línea punteada azul) comparada con la trayectoria de referencia (línea roja).



Figura 8.21. Trayectoria generada en la segunda prueba de la curva senoidal.

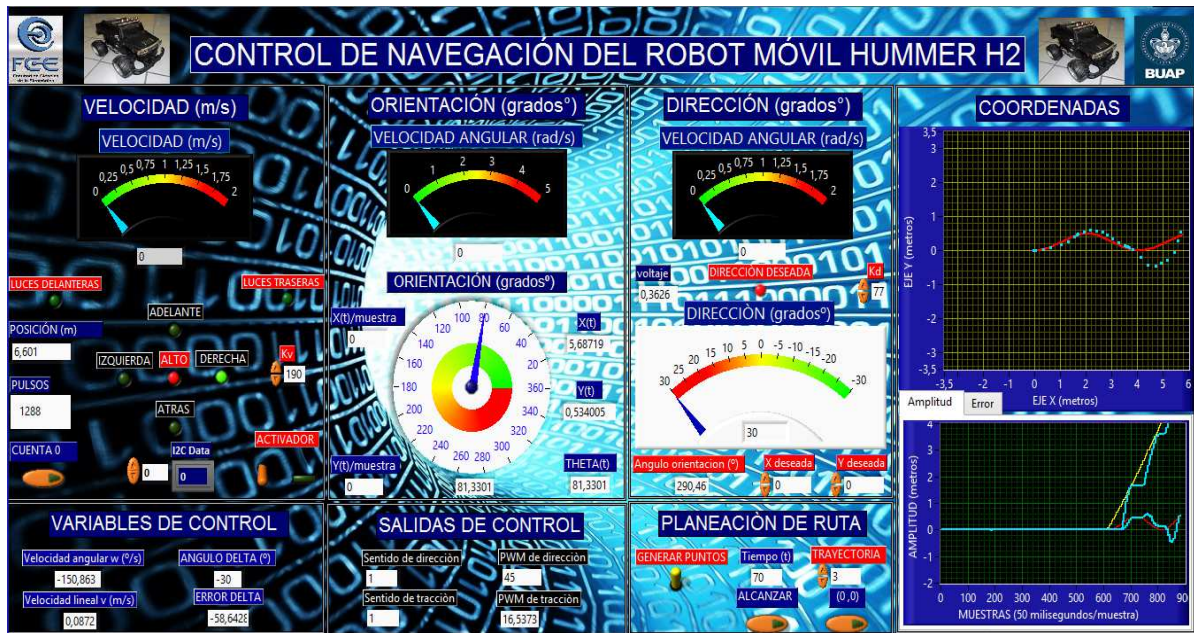


Figura 8.22. Trayectoria generada en la segunda prueba de la curva senoidal.

TERCERA PRUEBA. En la figura 8.23 se muestra la trayectoria generada por el robot móvil HUMMER H2 (línea naranja) en la tercera prueba, teniendo como trayectoria deseada la curva senoidal.



Figura 8.23. Trayectoria generada en la tercera prueba de la curva senoidal.

En la figura 8.24 se ilustra la interfaz de usuario después de haberse realizado la tercera prueba, donde se aprecia la trayectoria realizada por el robot móvil (línea punteada azul) comparada con la trayectoria de referencia (línea roja).

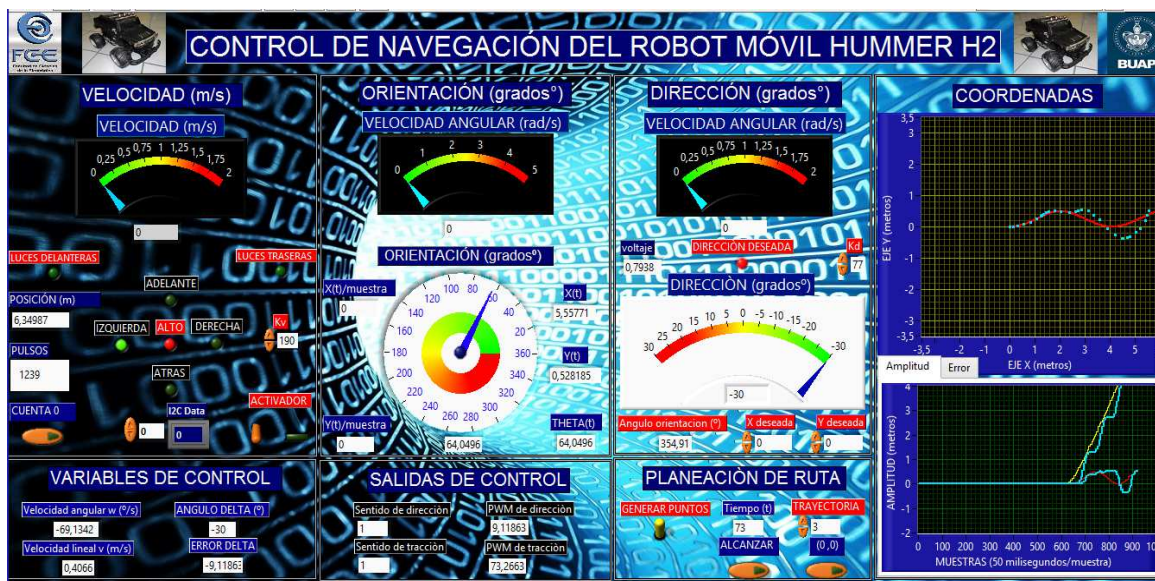


Figura 8.24. Trayectoria generada en la tercera prueba de la curva senoidal.

CUARTA PRUEBA. En la figura 8.25 se muestra la trayectoria generada por el robot móvil HUMMER H2 (línea negra) en la cuarta prueba, teniendo como trayectoria deseada la curva senoidal.



Figura 8.25. Trayectoria generada en la cuarta prueba de la curva senoidal.

En la figura 8.26 se ilustra la interfaz de usuario después de haberse realizado la cuarta prueba, donde se aprecia la trayectoria realizada por el robot móvil (línea punteada azul) comparada con la trayectoria de referencia (línea roja).

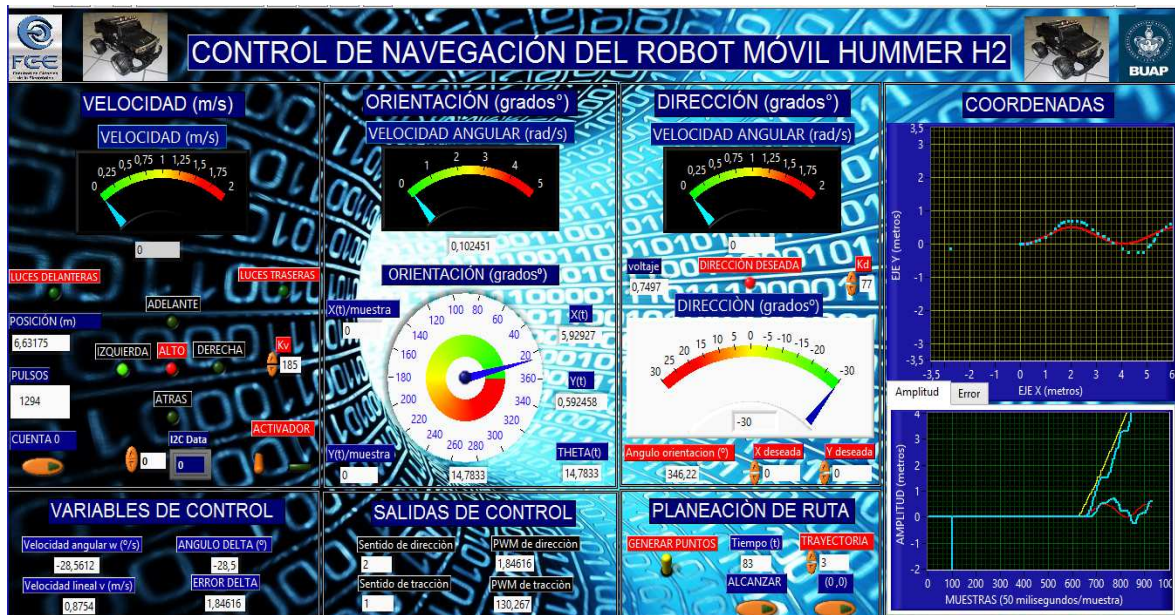


Figura 8.26. Trayectoria generada en la cuarta prueba de la curva senoidal.

Circunferencia

La cuarta curva paramétrica que se aplicará como trayectoria de referencia para el robot móvil HUMMER H2 es una circunferencia, cuyo radio de curvatura es de 1.5 metros.

Esta curva paramétrica está definida por la ecuación (8.3) para el eje “x” y por la ecuación (8.4) para el eje “y”.

Debido a la inclinación en algunas partes del suelo en el espacio de trabajo, las pruebas con esta curva paramétrica presentaban un alto error en el seguimiento de la trayectoria de referencia, por lo que se repitieron consecutivamente, modificando los valores de las ganancias proporcionales de los controladores de dirección y de tracción, también se eligieron diferentes puntos de origen para evitar los lugares con mayor inclinación en el espacio de trabajo hasta obtener un desempeño aceptable.

En la figura 8.27 se muestra la trayectoria dibujada por el robot móvil, teniendo como trayectoria deseada la circunferencia.

En la figura 8.28 se ilustra la interfaz de usuario con la información del desempeño obtenido para el seguimiento de la circunferencia, donde se aprecia la trayectoria realizada por el robot móvil (línea punteada azul) comparada con la trayectoria de referencia (línea roja).



Figura 8.27. Trayectoria generada en el seguimiento de una circunferencia.

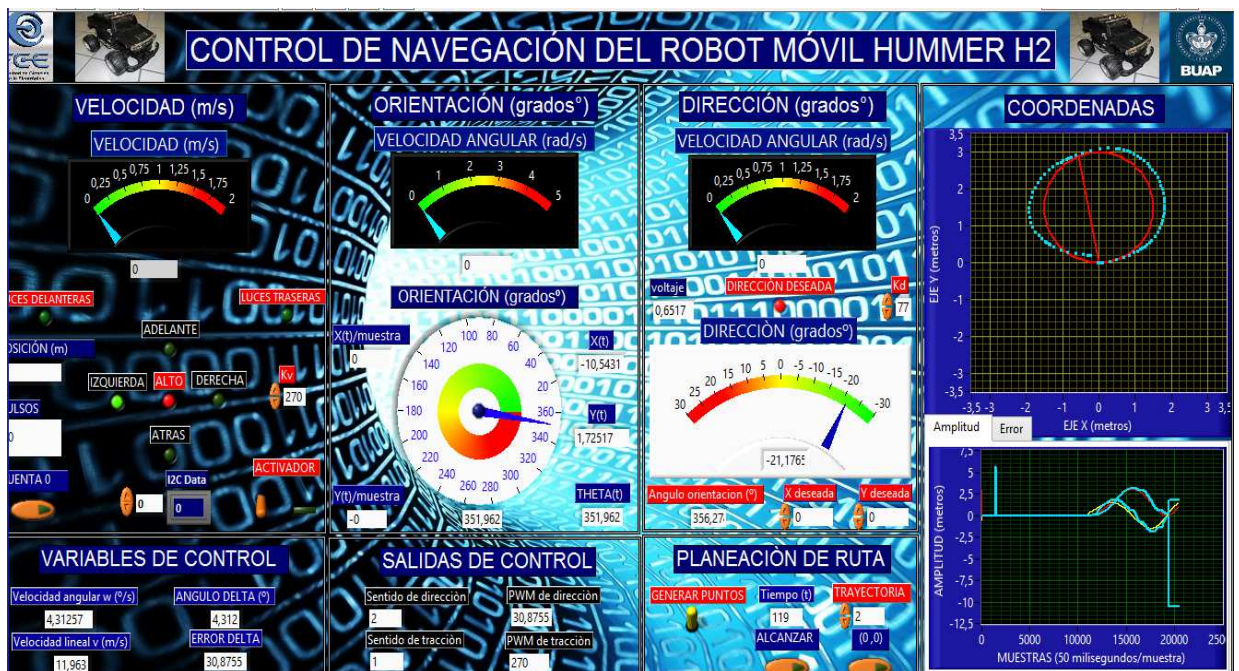


Figura 8.28. Trayectoria generada utilizando una circunferencia como curva paramétrica.

8.2 TRAYECTORIAS CON CONTROL PUNTO A PUNTO

En esta estrategia de control el generador de trayectorias no interviene, el tiempo no influye en la generación de puntos consecutivos, ya que solo se comanda un punto de alcance para el robot móvil HUMMER H2.

El recorrido o la trayectoria que describa el robot móvil para alcanzar la posición deseada, no es considerada en esta estrategia de control, con esto se expresa que el objetivo de esta estrategia de control únicamente depende de que el robot móvil HUMMER H2 alcance la coordenada deseada dentro del plano 2D, independientemente de la trayectoria que se genere.

Una vez que el objetivo sea alcanzado, las luces del robot móvil HUMMER H2 se activarán en forma de parpadeo para indicarnos que el robot móvil ha llegado a la posición deseada.

Posición deseada (1, 0)

La primera coordenada que se comanda desde la interfaz de usuario como posición deseada para ser alcanzada por el robot móvil HUMMER H2 es (1, 0).

En la figura 8.29 se muestra la trayectoria seguida por el robot móvil HUMMER H2 para alcanzar la posición deseada (1, 0).



Figura 8.29. Trayectoria generada para alcanzar la posición deseada (1, 0).

En la figura 8.30 se ilustra la interfaz de usuario con la información del desempeño obtenido para llegar a la posición deseada (1, 0), donde se aprecia la trayectoria realizada por el robot móvil (línea punteada azul) y la línea recta que indica la posición deseada (línea roja).

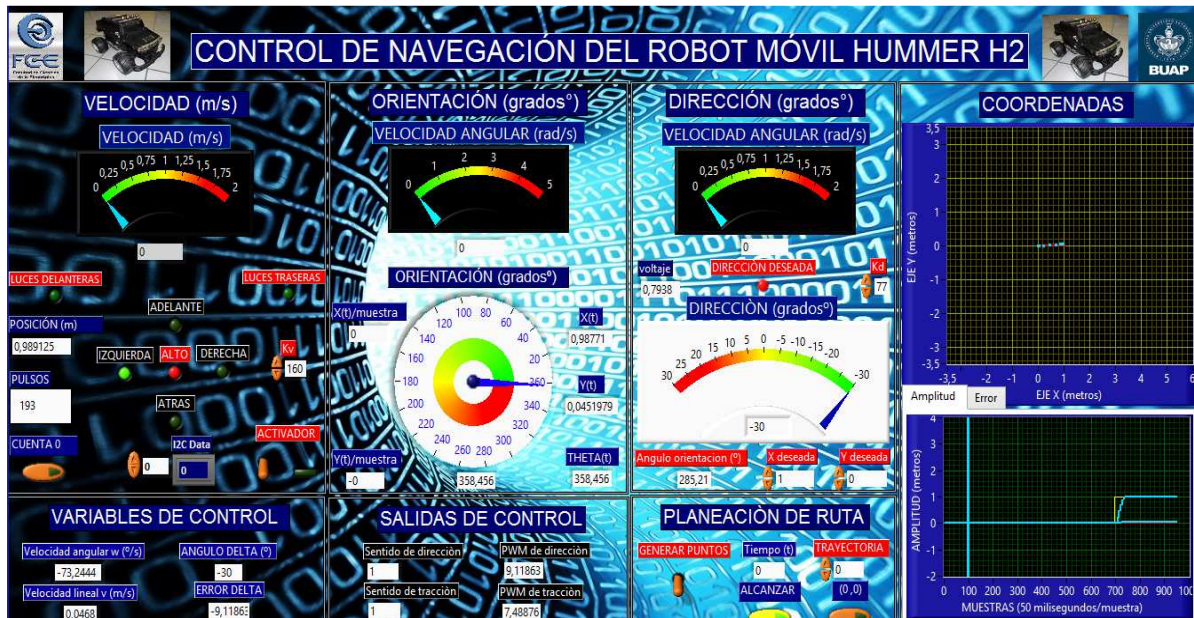


Figura 8.30. Trayectoria graficada para alcanzar la posición deseada (1, 0).

Posición deseada (1, -1)

La segunda coordenada que se comanda desde la interfaz de usuario como posición deseada para ser alcanzada por el robot móvil HUMMER H2 es (1, -1).



Figura 8.31. Trayectoria generada para alcanzar la posición deseada (1, -1).

En la figura 8.31 se muestra la trayectoria seguida por el robot móvil HUMMER H2 para alcanzar la posición deseada (1, -1).

En la figura 8.32 se ilustra la interfaz de usuario con la información del desempeño obtenido para llegar a la posición deseada (1, -1), donde se aprecia la trayectoria realizada por el robot móvil (línea punteada azul) y la línea recta que indica la posición deseada (línea roja).

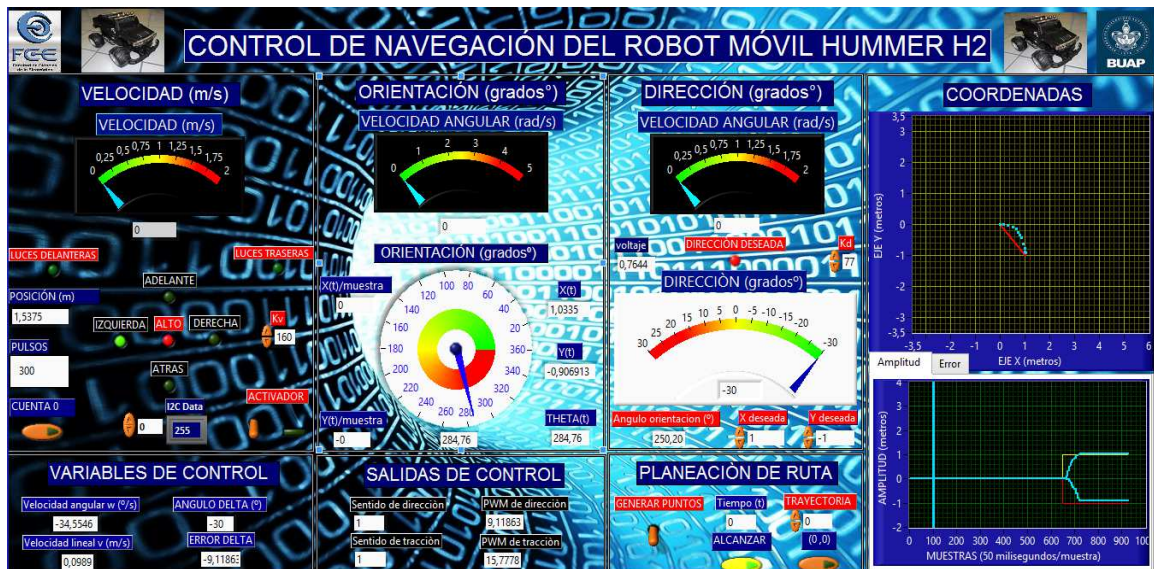


Figura 8.32. Trayectoria graficada para alcanzar la posición deseada (1, -1).

Posición deseada (3, -1)

La tercera coordenada que se comanda desde la interfaz de usuario como posición deseada para ser alcanzada por el robot móvil HUMMER H2 es (3, -1).

En la figura 8.33 se muestra la trayectoria seguida por el robot móvil HUMMER H2 para alcanzar la posición deseada (3, -1).

En la figura 8.34 se ilustra la interfaz de usuario con la información del desempeño obtenido para llegar a la posición deseada (3, -1), donde se aprecia la trayectoria realizada por el robot móvil (línea punteada azul) y la línea recta que indica la posición deseada (línea roja).

Además se coloca un metro para medir la distancia en el eje “y” (figura 8.33), con esto se observa que el error que el desempeño del robot móvil generó en este eje es menor de 5 centímetros.

De igual forma se midió la distancia en el eje “x” donde el error generado también es menor de 1 centímetro de longitud.



Figura 8.33. Trayectoria generada para alcanzar la posición deseada (3, -1).

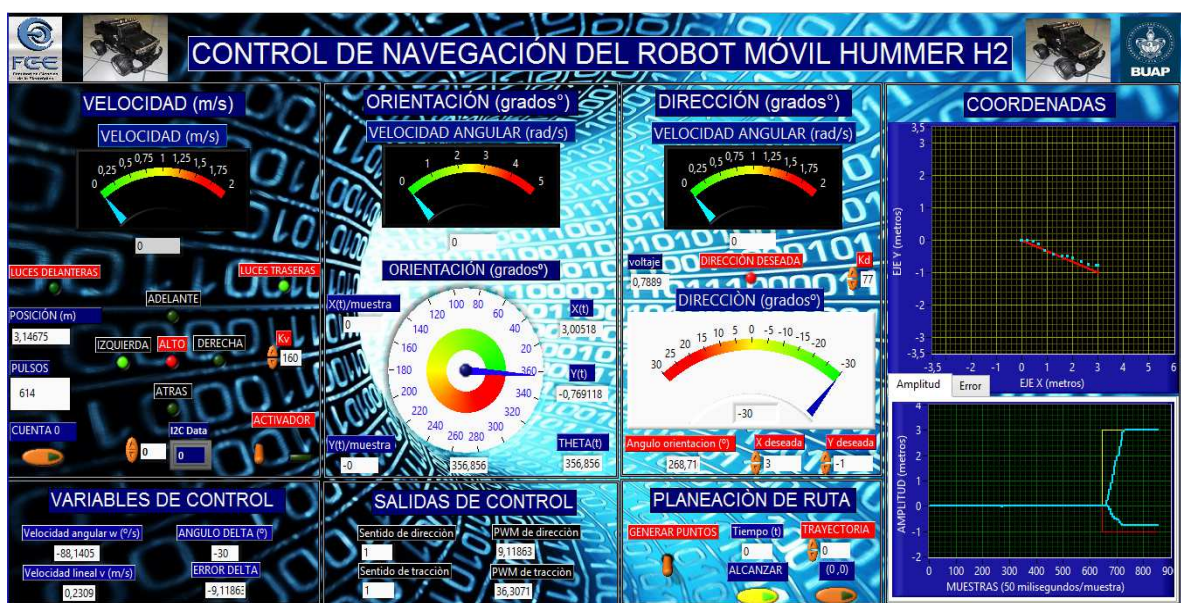


Figura 8.34. Trayectoria graficada para alcanzar la posición deseada (3, -1).

Posición deseada (3, 2)

La cuarta coordenada que se comanda desde la interfaz de usuario como posición deseada para ser alcanzada por el robot móvil HUMMER H2 es (3, 2).



Figura 8.35. Trayectoria generada para alcanzar la posición deseada (3, 2).

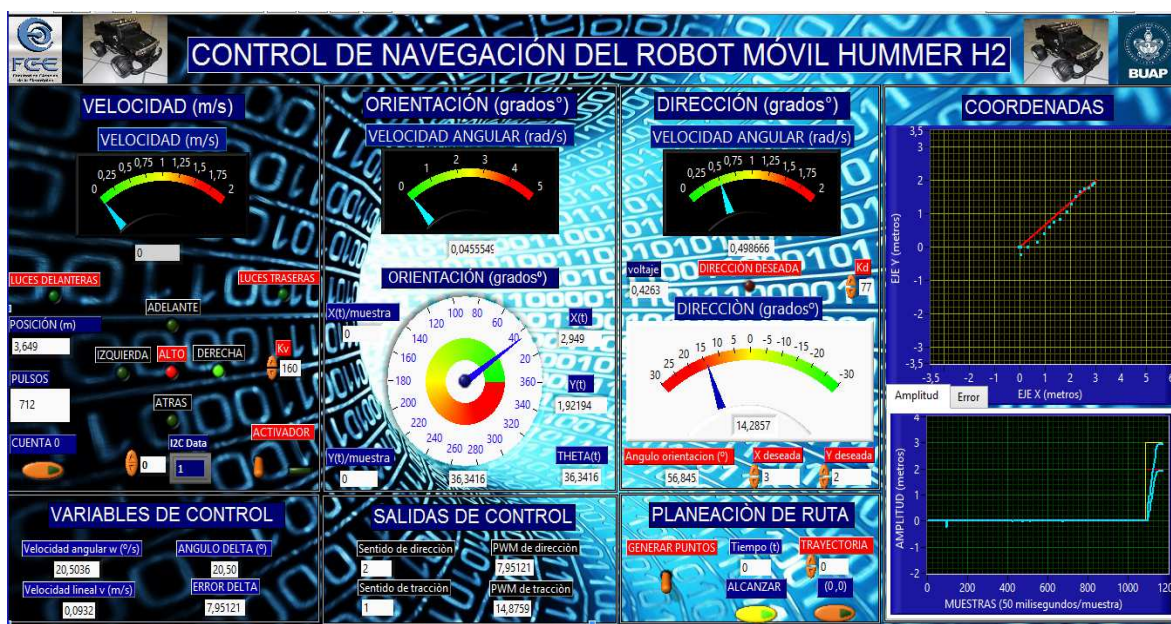


Figura 8.36. Trayectoria graficada para alcanzar la posición deseada (3, 2).

En la figura 8.35 se muestra la trayectoria seguida por el robot móvil HUMMER H2 para alcanzar la posición deseada (3, 2).

En la figura 8.36 se ilustra la interfaz de usuario con la información del desempeño obtenido para llegar a la posición deseada (3, 2), donde se aprecia la trayectoria realizada por el robot móvil (línea punteada azul) y la línea recta que indica la posición deseada (línea roja).

Posición deseada (1, 0.5) y (2, -1.5)

La quinta coordenada que se comanda desde la interfaz de usuario como posición deseada para ser alcanzada por el robot móvil HUMMER H2 es (1, 0.5), una vez que es alcanzada se comanda la sexta coordenada deseada (2, -1.5), esta vez sin posicionar el robot móvil desde el origen sino a partir de la posición que alcanzó anteriormente.

En la figura 8.37 se muestra la trayectoria seguida por el robot móvil HUMMER H2 para alcanzar la posición deseada (1, 0.5) y posteriormente (2, -1.5).

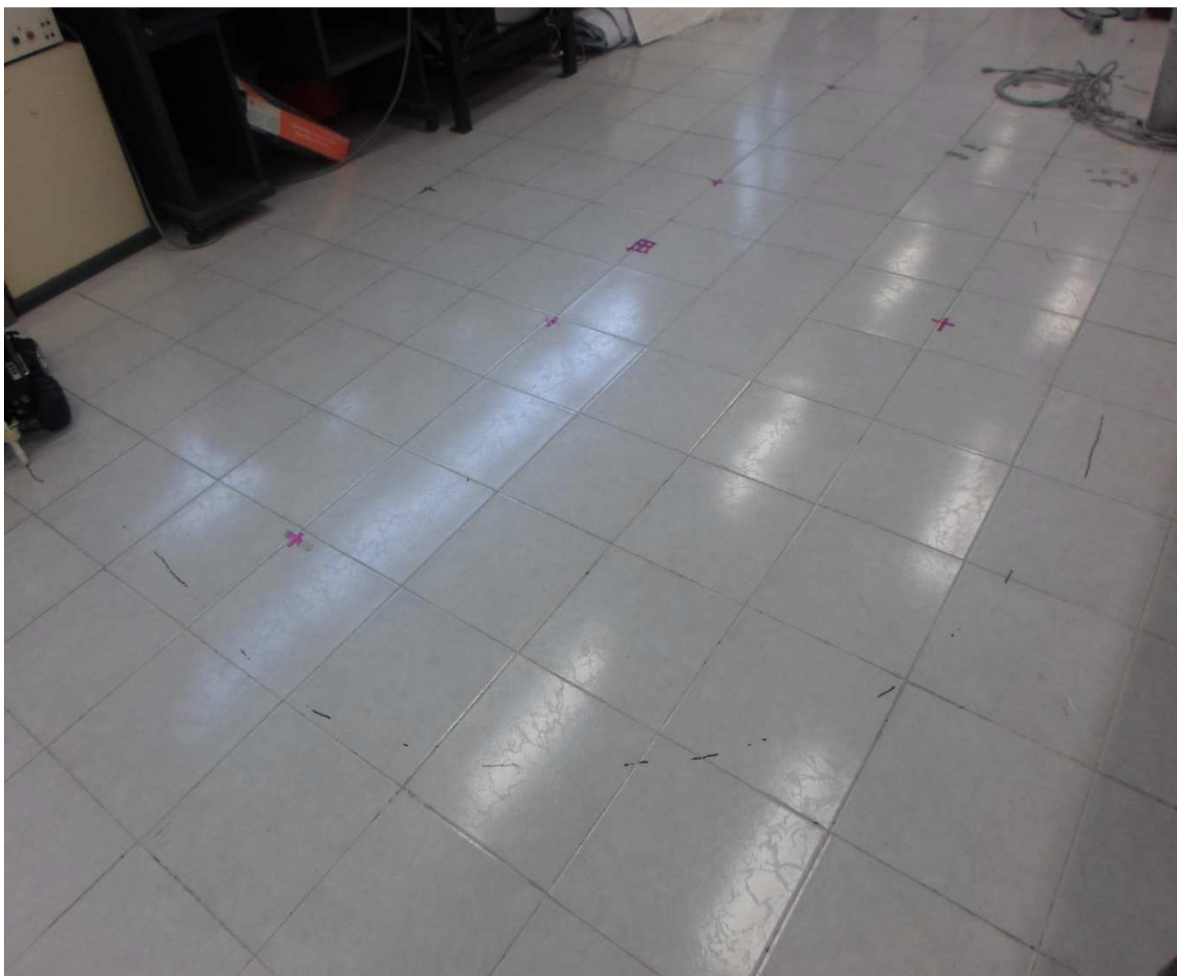


Figura 8.37. Trayectoria generada para alcanzar la posición deseada (1, 0.5) y (2, -1.5).

En la figura 8.38 se ilustra la interfaz de usuario con la información del desempeño obtenido para llegar a la posición deseada final (2, -1.5), donde se aprecia la trayectoria realizada por el robot móvil (línea punteada azul) y la línea recta que indica la posición deseada (línea roja).

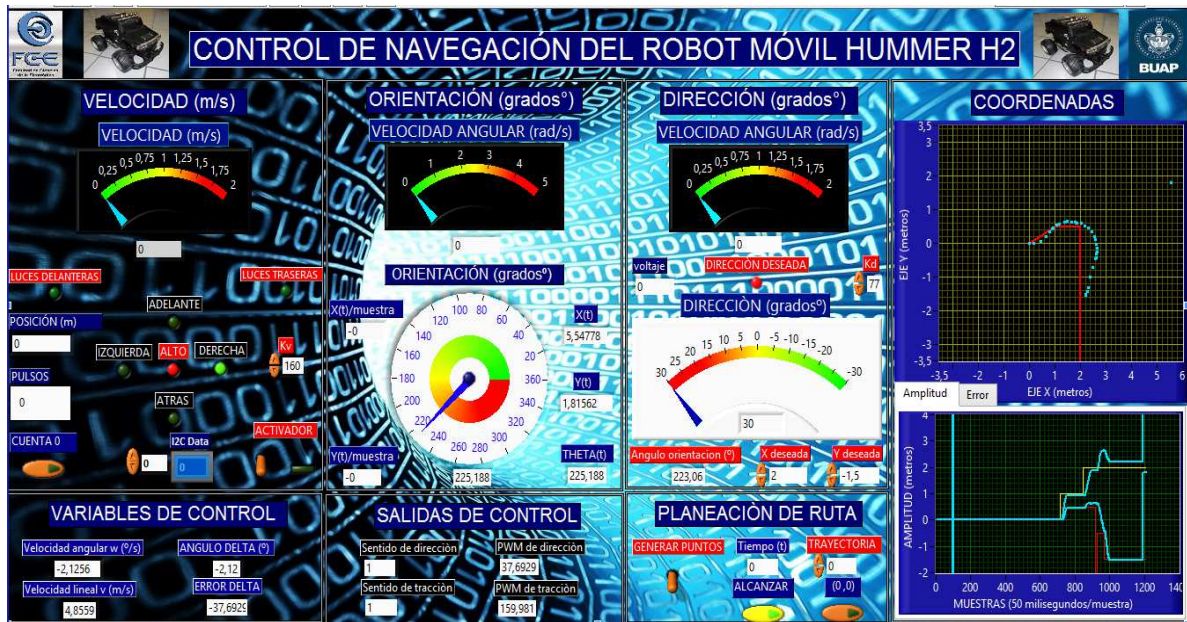


Figura 8.38. Trayectoria graficada para alcanzar la posición deseada (1, 0.5) y (2, -1.5).

8.3 RESULTADOS

Una vez que se han generado cada una de las trayectorias en las pruebas realizadas con el robot móvil HUMMER H2, todas las muestras tomadas en cada uno de los recorridos se guardan en una tabla como se muestra en la figura 8.39, para realizar un análisis sobre el desempeño del vehículo y la mejora con la sintonización de las ganancias proporcionales en cada uno de los controladores.

Al mismo tiempo que se ejecuta el programa del algoritmo de control, se genera un archivo con extensión (.xlsx), el cual contiene las coordenadas de referencia (x, y) como las coordenadas de trayectoria del robot móvil (x, y).

Finalmente los archivos se ejecutan en el programa de Excel donde se realizan las graficas de comparación de datos sobre la trayectoria de referencia.

Se realiza un breve análisis sobre el comportamiento del robot móvil en cada una de las trayectorias de prueba, de igual manera en cada una de las trayectorias descritas utilizando la estrategia de control punto a punto.

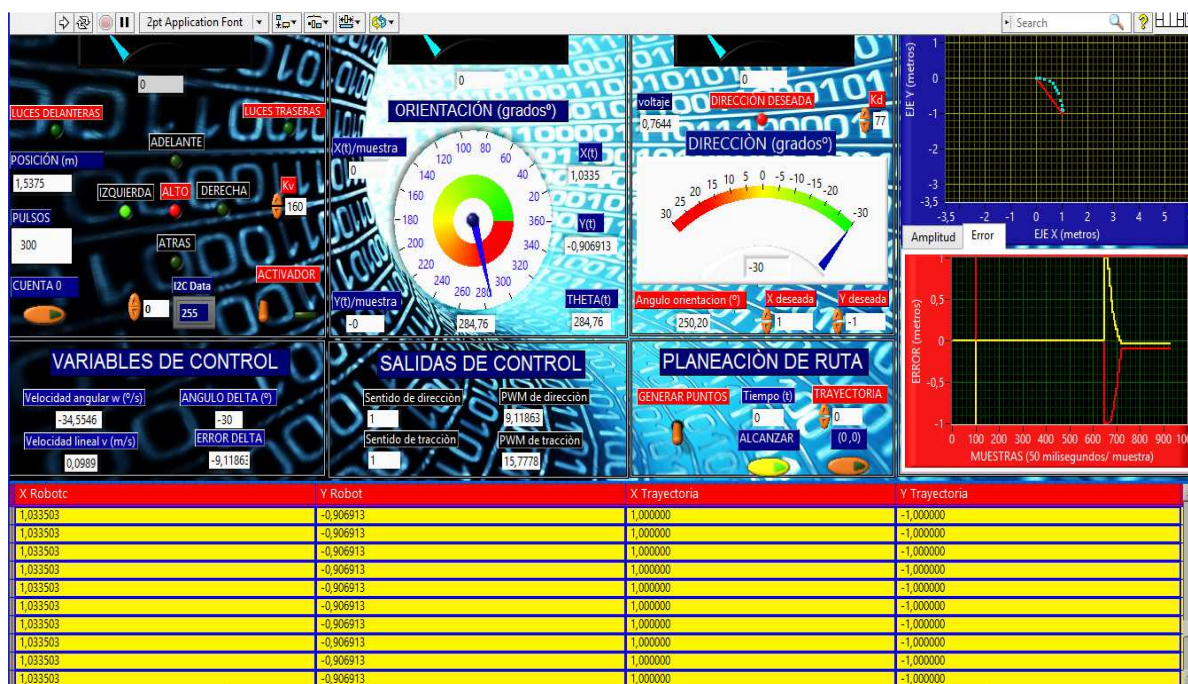


Figura 8.39. Tabla en la interfaz de usuario, utilizada para guardar muestras.

Arco extendido

Con los datos obtenidos en las 4 pruebas de arco extendido, se realizan las graficas de las figuras 8.40 (prueba 1), 8.41 (prueba 2), 8.42 (prueba 3) y 8.43 (prueba 4).

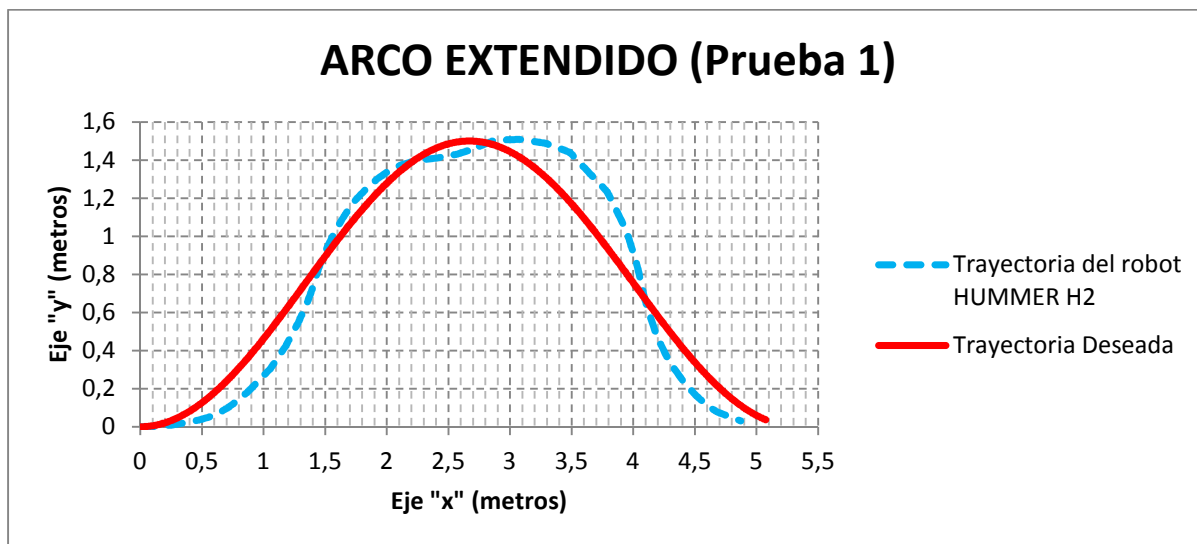


Figura 8.40. Gráfica de la prueba 1 del Arco extendido.

Donde se van sintonizando las ganancias de los controladores K_{DELTA} y K_V hasta obtener una trayectoria más fiel a la trayectoria de referencia.

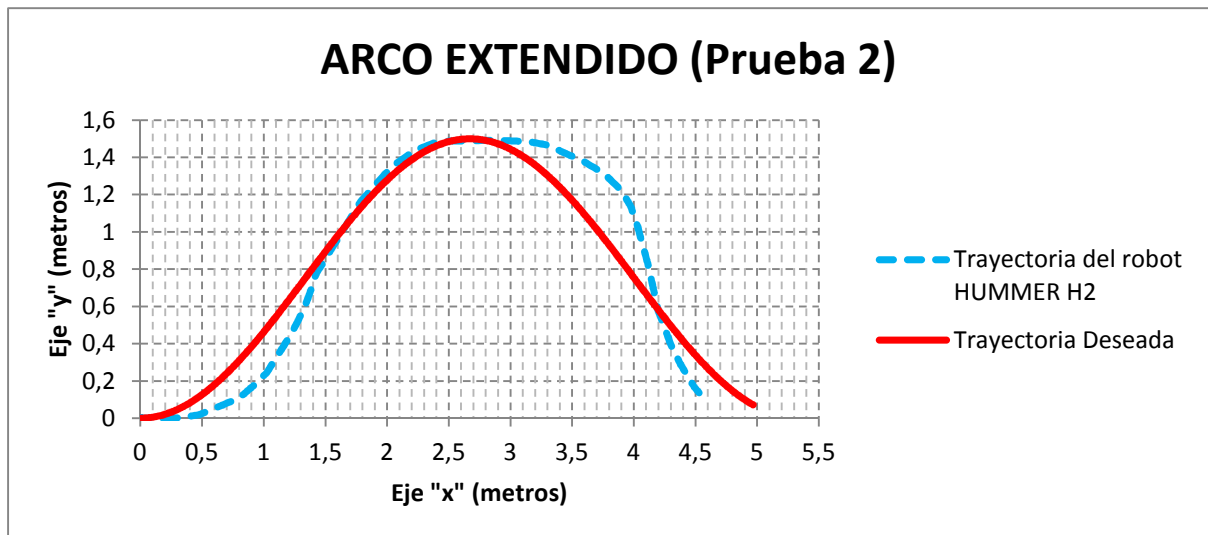


Figura 8.41. Gráfica de la prueba 2 del Arco extendido.

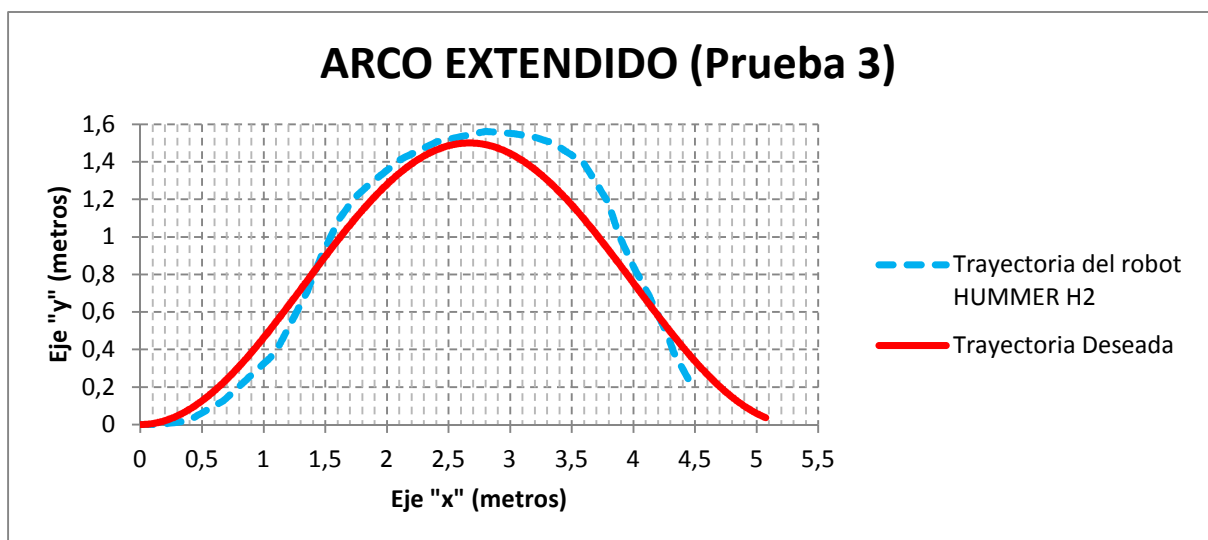


Figura 8.42. Gráfica de la prueba 3 del Arco extendido.

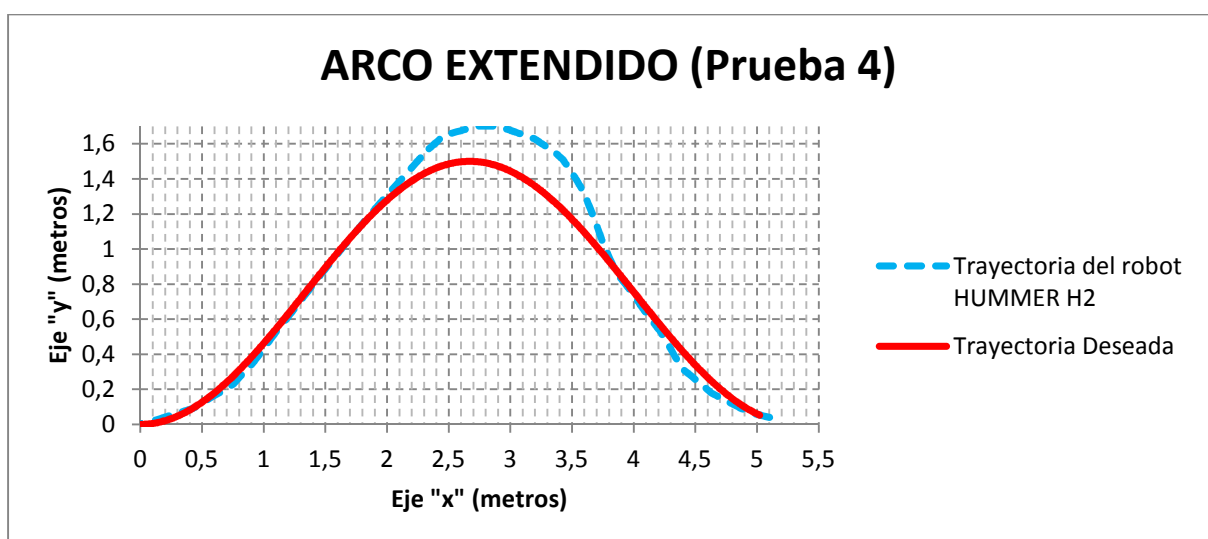


Figura 8.43. Gráfica de la prueba 4 del Arco extendido.

Media circunferencia

Con los datos obtenidos en las 4 pruebas de media circunferencia, se realizan las graficas de las figuras 8.44 (prueba 1), 8.45 (prueba 2), 8.46 (prueba 3) y 8.47 (prueba 4).

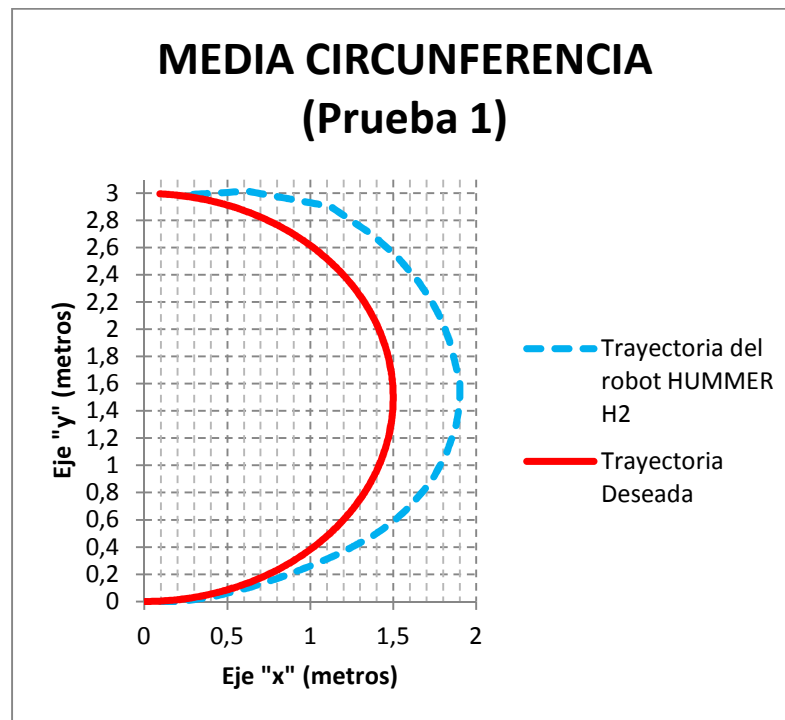


Figura 8.44. Gráfica de la prueba 1 de la media circunferencia.

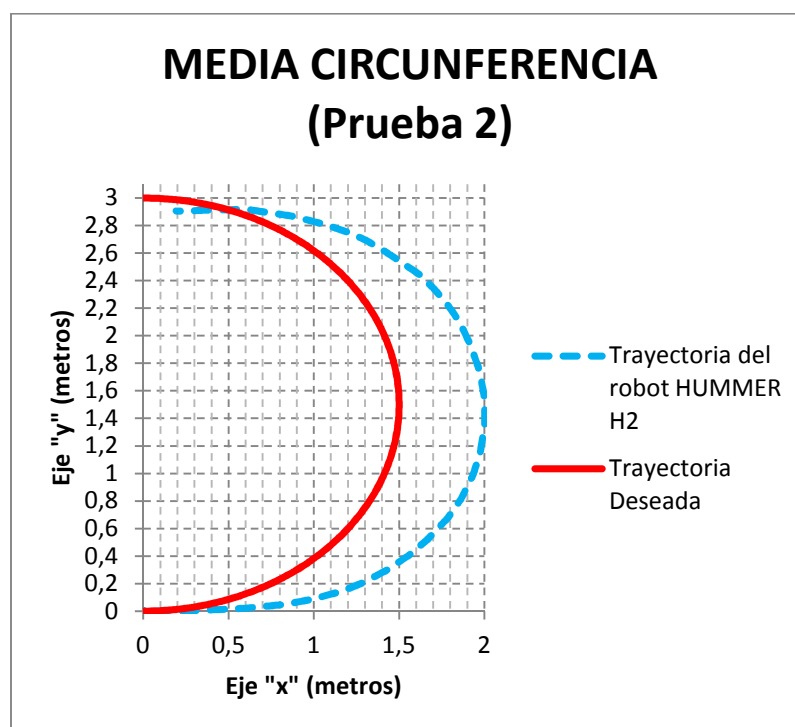


Figura 8.45. Gráfica de la prueba 2 de la media circunferencia.

Donde se van sintonizando las ganancias de los controladores K_{DELTA} y K_V hasta obtener una trayectoria más fiel a la trayectoria de referencia.

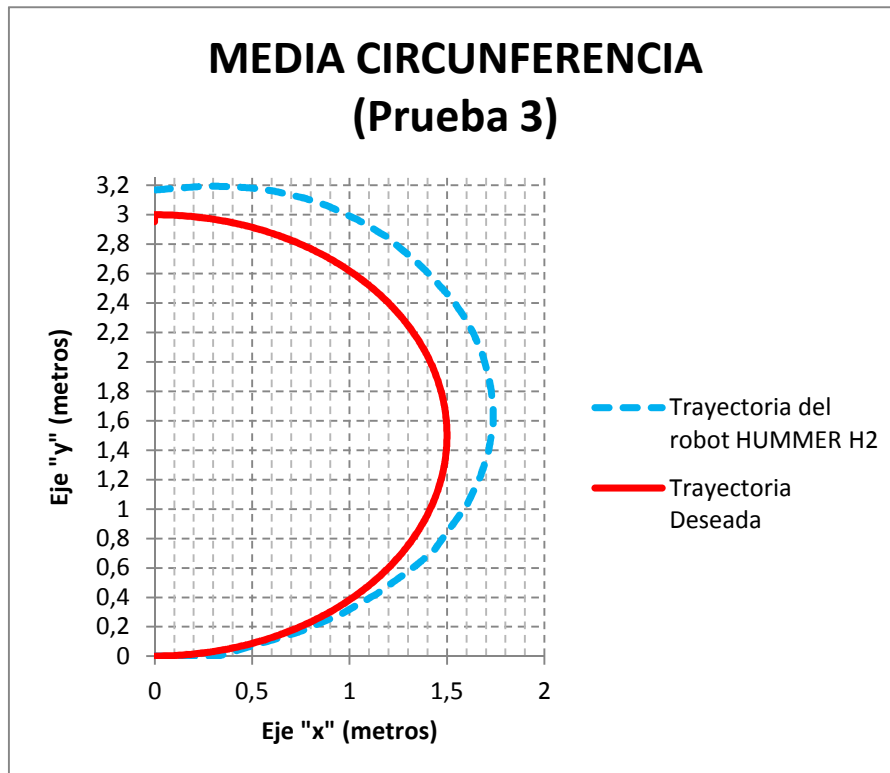


Figura 8.46. Gráfica de la prueba 3 de la media circunferencia.

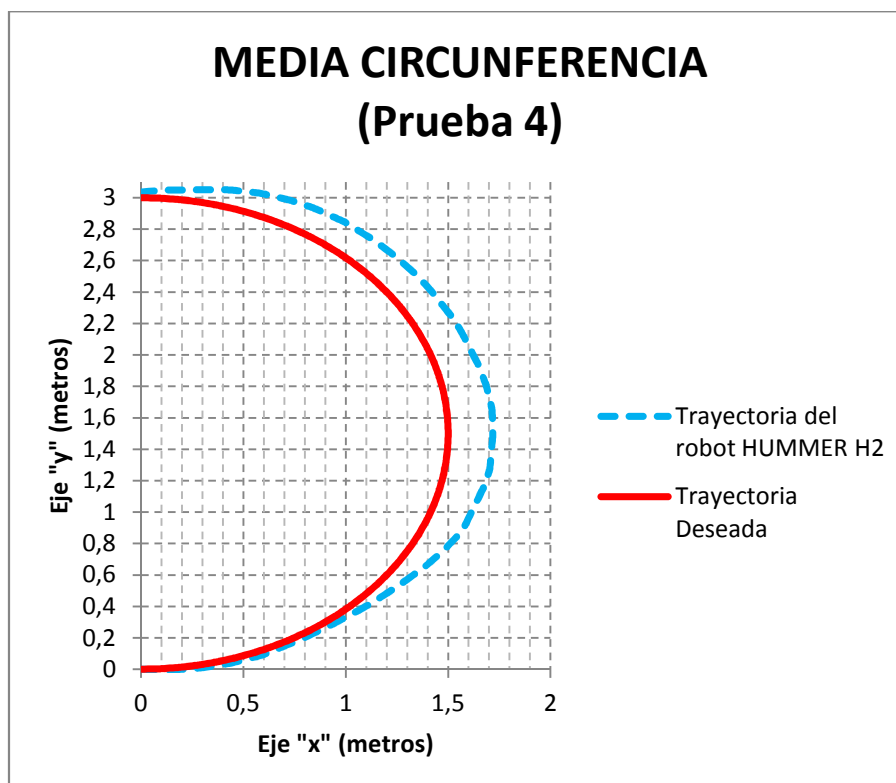


Figura 8.47. Gráfica de la prueba 4 de la media circunferencia.

Senoidal

Con los datos obtenidos en las 4 pruebas de la curva senoidal, se realizan las graficas de las figuras 8.48 (prueba 1), 8.49 (prueba 2), 8.50 (prueba 3) y 8.51 (prueba 4).

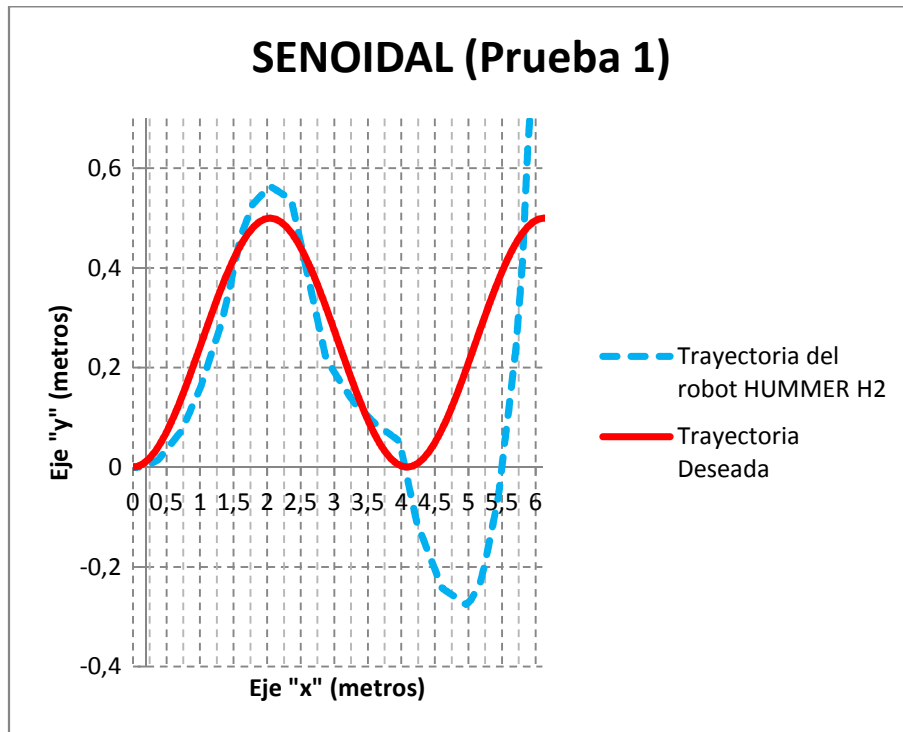


Figura 8.48. Gráfica de la prueba 1 de la curva senoidal.

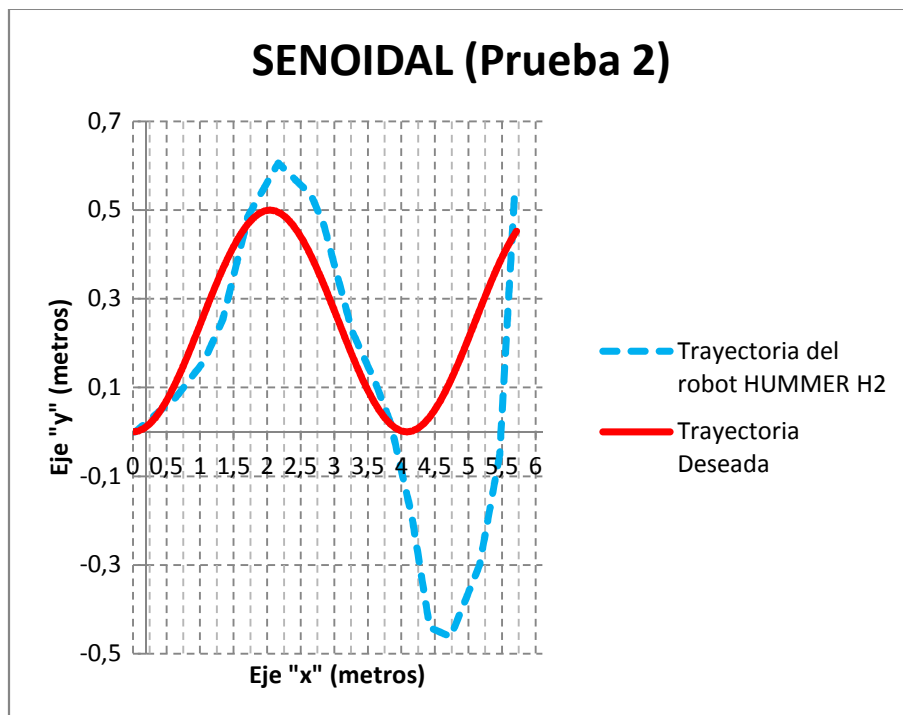


Figura 8.49. Gráfica de la prueba 2 de la curva senoidal.

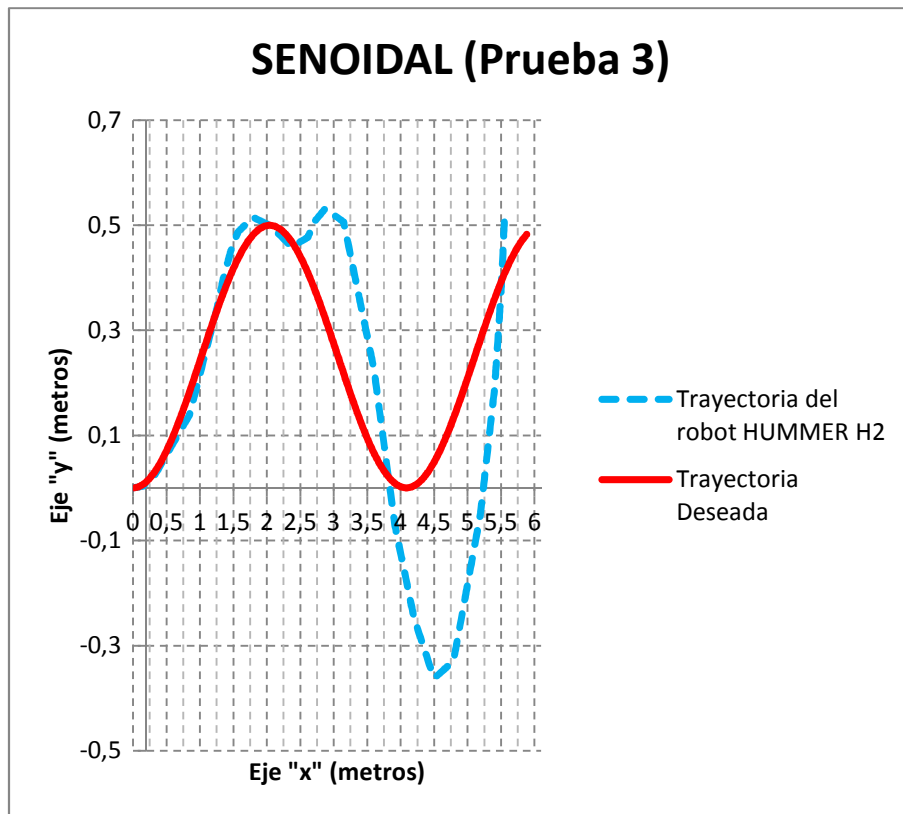


Figura 8.50. Gráfica de la prueba 3 de la curva senoidal.

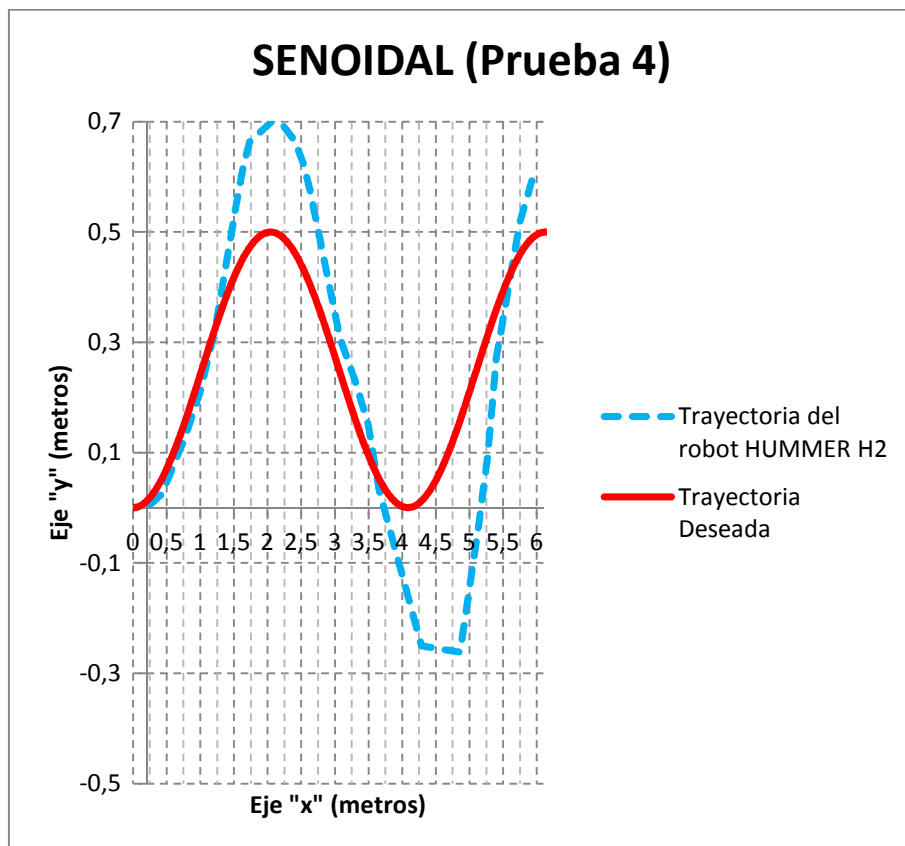


Figura 8.51. Gráfica de la prueba 3 de la curva senoidal.

Posiciones deseadas

Con los datos obtenidos en las pruebas del control punto a punto tomando como posiciones deseadas las coordenadas (1, 0), (1, -1), (3, 2), (1, 0.5) y (2, -1.5) se realizan las gráficas de las figuras 8.52 (1, 0), 8.53 (1, -1), 8.54 (3, 2) y 8.55 (1, 0.5) a (2,-1.5).

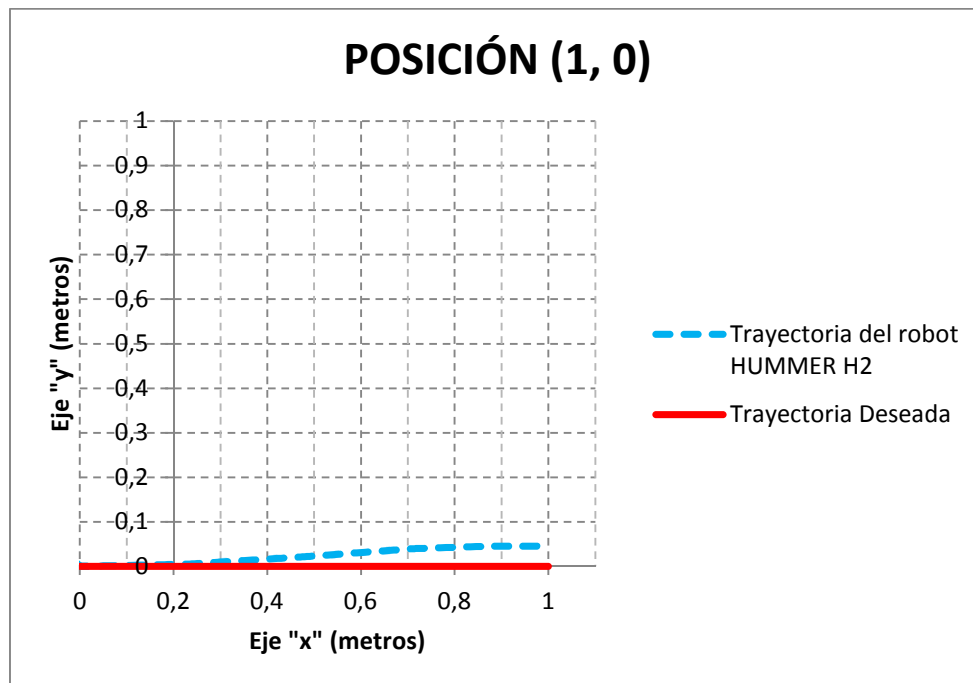


Figura 8.52. Gráfica de la prueba con la posición deseada (1, 0).

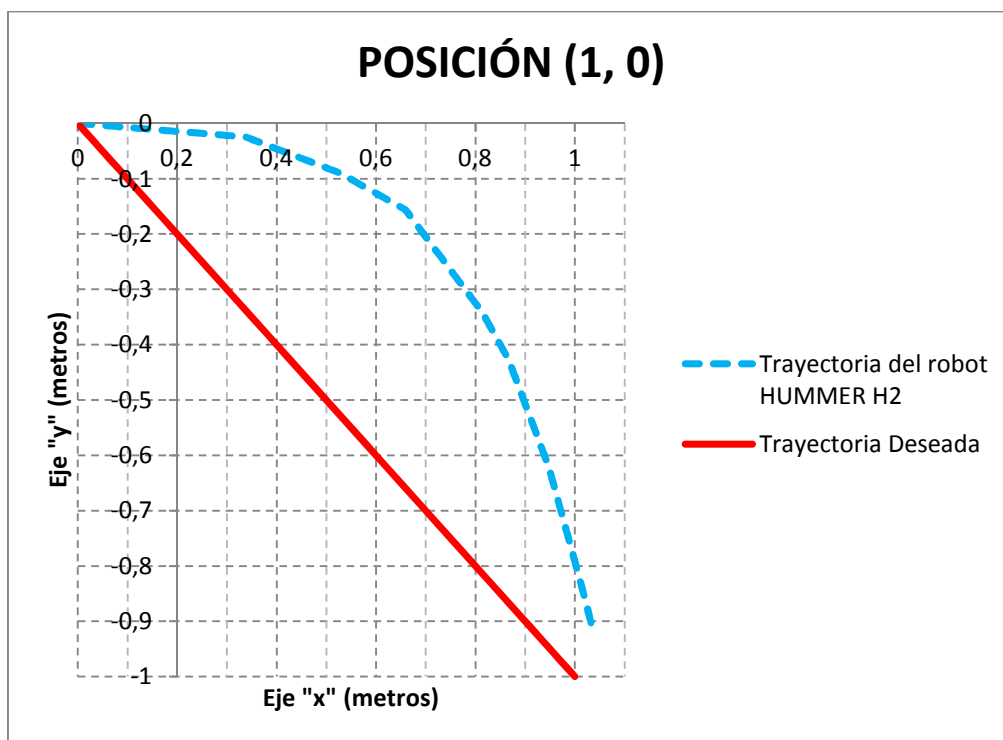


Figura 8.53. Gráfica de la prueba con la posición deseada (1, -1).

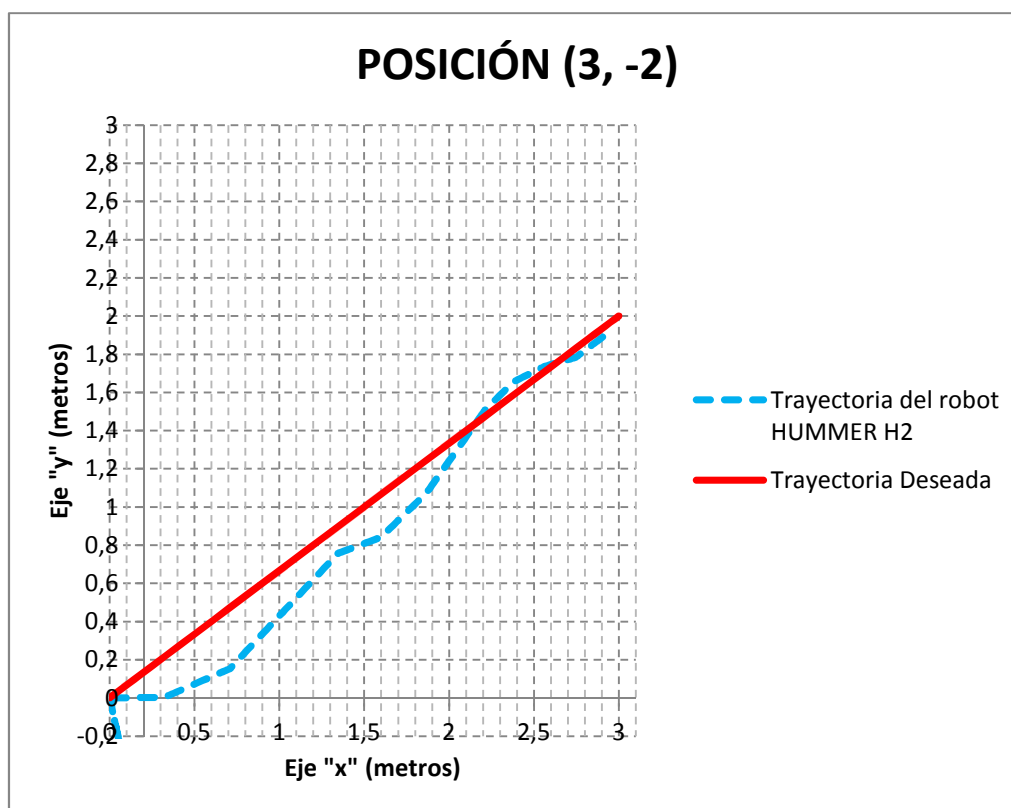


Figura 8.54. Gráfica de la prueba con la posición deseada (3, 2).

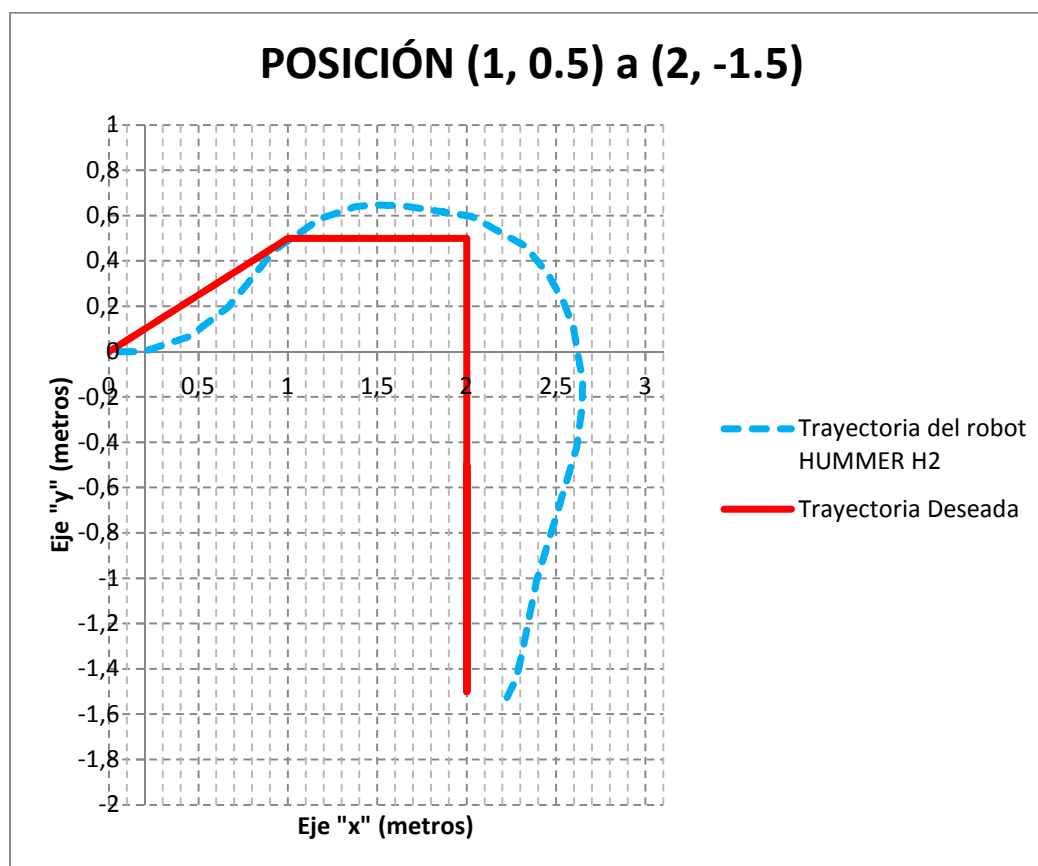


Figura 8.55. Gráfica de la prueba con la posición deseada (1, 0.5) a (2, -1.5)

CONCLUSIONES

En este trabajo de tesis se desarrolló la comunicación inalámbrica entre la computadora portátil y el robot móvil HUMMER. Dicha comunicación inalámbrica se estableció inicialmente en 9600 bits/segundo, dando como resultado una respuesta lenta del controlador hacia la planta. Posteriormente se incrementó la velocidad de transmisión a 57600 bits/segundo lo cual generó una respuesta más rápida que la anterior, pero aún no era eficiente para el control del robot móvil HUMMER H2.

Por tal motivo se procede a aumentar la velocidad de transmisión a 115200 bits/segundo, lo cual generó que la comunicación se interrumpiera después de algunos segundos de transmisión de datos, lo cual era ocasionado porque la velocidad de procesamiento de la computadora es mayor que la velocidad del microcontrolador de la tarjeta ARDUINO UNO. Así que para sincronizar las velocidades de procesamiento, se selecciona el valor de 111111 bits/segundo para la computadora y el valor de 115200 para el ARDUINO UNO, consiguiendo con esto una comunicación estable y una respuesta del controlador hacia la planta eficiente.

Inicialmente se desarrollan dos estrategias de control, para verificar la eficiencia de cada una de ellas utilizando por un lado el control predictivo aplicado al modelo dinámico del robot móvil HUMMER H2, y por otro lado se utilizó el control punto a punto aplicado al modelo cinemático.

En principio y teoría el control predictivo aplicado al modelo dinámico era la estrategia de control con más ventajas y probabilidad de ser aplicada para el control del robot móvil HUMMER H2. La única desventaja de esta estrategia de control es el alto costo computacional que se requiere para poder aplicarla, situación que comprobamos cuando se llevó a cabo la implementación del control predictivo en LABVIEW, por lo que esta opción quedo descartada y se utilizó únicamente en un escenario de simulación del modelo dinámico del robot móvil HUMMER H2.

Una vez descartada la estrategia de control predictivo, se establece el modelo cinemático como centro de la estrategia de control punto a punto. Esta estrategia de control utiliza un costo computacional considerable el cual permite que el algoritmo de control se ejecute a una velocidad aproximada de 50 milisegundos por muestra, lo cual nos otorga un desempeño eficiente en la evaluación del algoritmo en cada periodo de muestreo.

La lectura de los sensores se realiza a través de los pines de entrada del ARDUINO UNO, el potenciómetro a través de una entrada analógica, el encoder por medio de una entrada digital y el magnetómetro mediante los pines de protocolo I2C, la lectura de los

tres sensores es continua en LABVIEW es decir depende de la velocidad del procesador de la computadora.

Una vez que se obtiene la lectura de cada uno de los sensores, esta información es procesada en cada periodo de muestreo por el algoritmo de control en LABVIEW, donde las magnitudes físicas obtenidas se convierten en información del robot móvil, que a su vez es evaluada para poder tomar una acción de control si es requerida para seguir la trayectoria o el punto deseado en ese instante.

En las pruebas experimentales realizadas utilizando la estrategia de control con curvas paramétricas, se consiguió un desempeño favorable a pesar de que intervinieron las restricciones físicas del robot móvil HUMMER H2, como por ejemplo su limitación en el ángulo máximo de giro= 30° . También intervinieron las limitaciones físicas del espacio de trabajo, como por ejemplo la declinación abrupta en algunas partes y por lo tanto la disminución del área eficiente para realizar las pruebas experimentales.

La primera curva paramétrica seleccionada fue una circunferencia con radio= 1.5 metros, al realizar el seguimiento de trayectoria se reflejaron cambios abruptos en la lectura del magnetómetro utilizado como el sensor de orientación, por tal motivo se acoto el área de trabajo eficiente y con ello también se reemplazo la curva paramétrica de la circunferencia completa por la curva paramétrica de la mitad de la circunferencia con radio = 1.5 metros. La segunda trayectoria seleccionada fue un arco de altura = 1.5 metros lo cual permitió que el robot móvil reprodujera la trayectoria sin alcanzar sus límites físicos. Finalmente la tercera trayectoria seleccionada fue una curva senoidal con altura = 0.5 metros, a diferencia de la curva anterior, esta trayectoria forzó al robot móvil a generar su máximo ángulo de giro, lo cual se refleja en el seguimiento con una desviación mayor en las curvas de la trayectoria senoidal.

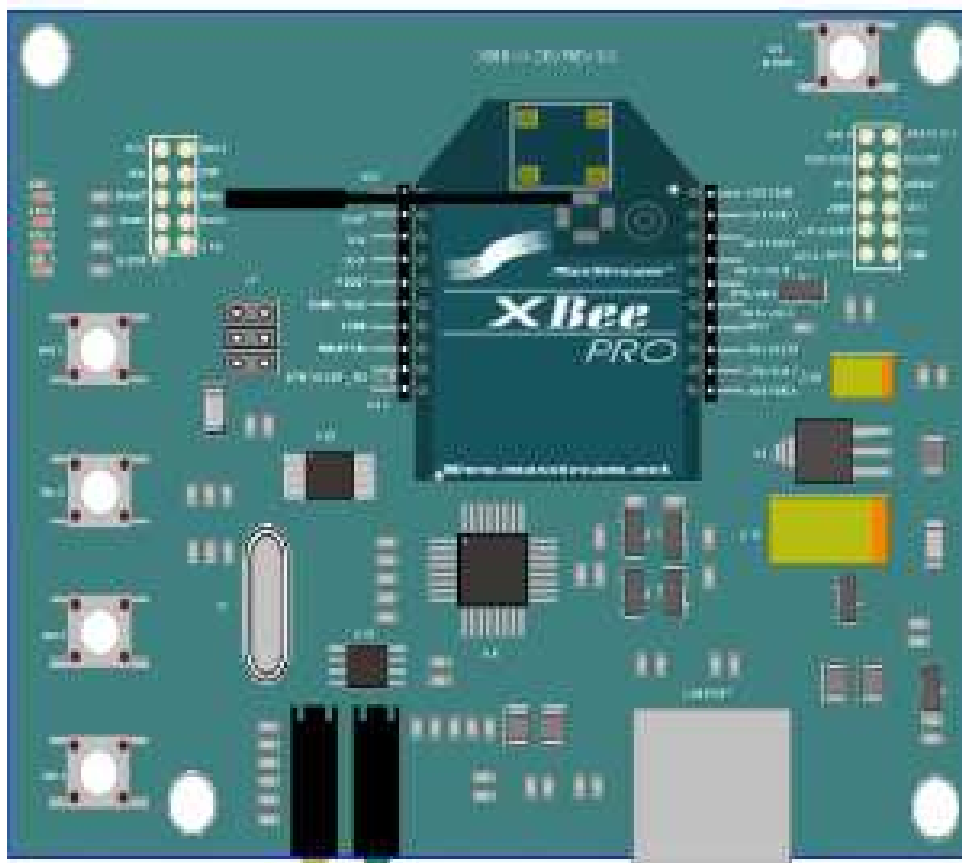
En las pruebas experimentales realizadas utilizando la estrategia de control punto a punto, se seleccionaron diferentes coordenadas dentro del espacio de trabajo, el desempeño del robot móvil fue alto, debido a que el error máximo producido al llegar a la posición deseada no superó los 5 centímetros.

El desempeño del robot móvil HUMMER H2 en general resultó eficiente, adecuándose a las condiciones no ideales de operación y generando un control estable en el seguimiento de trayectorias.

La versatilidad en la implementación de este trabajo de tesis permite la continuación del desarrollo de esta plataforma experimental, llevada a un campo de trabajo en particular puede ser utilizada para realizar alguna aplicación específica. Ya que cada uno de los bloques que la conforman pueden ser reemplazados, sin afectar el resto del sistema de control, con lo que se obtiene continuidad en el trabajo y un avance que será reflejado en el trabajo futuro que de solución a alguna problemática actual.

APÉNDICES

APÉNDICE A. Tarjeta XBIB-U-DEV



ARQUITECTURA DE LA TARJETA DE INTERFAZ XBIB-U-DEV.

APÉNDICE B. Características técnicas de los módulos XBEE-PRO

DESCRIPCIÓN DE PINES DE LOS MÓDULOS XBEE-PRO			
Pin	Nombre	Dirección	Descripción
1	VCC	-	Fuente de Voltaje
2	DOUT	Salida	Dato de Salida UART
3	DIN/DCONFIG	Entrada	Dato de Entrada UART
4	DO8*	Salida	Salida Digital 8
5	RESET	Entrada	Reset del modulo
6	PWM0/RSSI	Salida	Salida PWM 0 / RX Indicador de señal de potencia
7	PWM1	Salida	Salida PWM 1

8	[RESERVED]	-	No conectar
9	DTR/SLEPP_RQ/DI8	Entradas	Pin de línea de control Sleep / Entrada digital 8
10	GND	-	Tierra
11	AD4/DIO4	Cualquiera	Entrada análoga 4 / Salida o Entrada digital 4
12	CTS/DIO7	Cualquiera	Listo para enviar datos / Salida o Entrada Digital 7
13	ON/SLEEP	Salida	Indicador de estado del módulo
14	VREF	Entrada	Voltaje de Referencia para la entradas Análogas/Digitales
15	Associate/AD5/DIO5	Cualquiera	Indicador Asociador / Entrada Análoga 5/ Entrada o Salida digital 6
16	RTS/AD6/DIO6	Cualquiera	Petición para enviar datos / Entrada Análoga 5/Entrada o Salida Digital 6
17	AD3/DIO3	Cualquiera	Entrada análoga 3 / Entrada o Salida digital 3
18	AD2/DIO2	Cualquiera	Entrada análoga 2 / Entrada o Salida digital 2
19	AD1/DIO1	Cualquiera	Entrada análoga 1 / Entrada o Salida digital 1
20	AD0/DIO0	Cualquiera	Entrada análoga 0 / Entrada o Salida digital 0

ESPECIFICACIONES ELÉCTRICAS DE LOS MÓDULOS XBEE-PRO						
Símbolo	Características	Condición	Min	Típico	Max	Unidad
V_{IL}	Entrada de Bajo Voltaje	Todas las entradas digitales	-	-	$0.35 \cdot V_{CC}$	V
V_{IH}	Entrada de Alto Voltaje	Todas las entradas digitales	$0.7 \cdot V_{CC}$	-	-	V
V_{OL}	Salida de Bajo Voltaje	$I_{OL} = 2\text{mA}$, $V_{CC} \geq 2.7\text{V}$	-	-	0.5	V
V_{OH}	Salida de Alto Voltaje	$I_{OH} = -2\text{mA}$, $V_{CC} \geq 2.7\text{V}$	$V_{CC} - 0.5$	-	-	V
I_{IN}	Entrada de fugas de corriente	$V_{IN} = V_{CC}$ or GND, todas las entradas, por pin	-	0.025	1	μA
I_{OZ}	Corriente de fugas con alta impedancia	$V_{IN} = V_{CC}$ or GND, todas las entradas y salidas, por pin	-	0.025	1	μA
TX	Corriente Transmitida	$V_{CC} = 3.3\text{V}$	-	215	-	mA
RX	Corriente Recibida	$V_{CC} = 3.3\text{V}$	-	55	-	mA
PWR-Dwn	Corriente de apagado	Parámetro SM = 1	-	< 10	-	μA

DETALLES ELÉCTRICOS DE LAS ENTRADAS ANALÓGICAS/DIGITALES						
Símbolo	Características	Condición	Min	Típico	Max	Unidad
V_{REFH}	Voltaje de referencia		2.08	-	V_{DDAD}	V
I_{REF}	Corriente de referencia	Habilitado	-	200	-	μA
		Deshabilitado	-	< 0.01	0.02	μA
V_{INDC}	Entrada Análoga de Voltaje		V -0.3	-	V +0.3	V

CARACTERÍSTICAS DE RENDIMIENTO DE LOS MÓDULOS XBEE-PRO.						
Símbolo	Características	Condición	Min	Típico	Max	Unidad
R_{AS}	Impedancia de Entrada		-	-	10	k Ω
V_{AIN}	Entrada Análoga de Voltaje		V_{REFL}	-	V_{REFH}	V
RES	Resolución	$2.08 \leq V_{DDAD} \leq 3.6V$	2.031	-	3.516	mV
DNL	Diferencial – No lineal		-	± 0.5	± 1.0	LSB
INL	Integral – No Lineal		-	± 0.5	± 1.0	LSB
E_{ZS}	Error Escala cero		-	± 0.4	± 1.0	LSB
F_{FS}	Error Full Escala		-	± 0.4	± 1.0	LSB
E_{IL}	Error Entrada de Fuga		-	± 0.05	± 5.0	LSB
E_{TU}	Error total sin Ajustar		-	± 1.1	± 2.5	LSB

ESPECIFICACIONES TÉCNICAS DE LOS MÓDULOS XBEE-	
DETALLES	XBee-PRO 802.15.4
Rendimiento	
Tasa de datos	254 Kbps
Rango recinto interior	100 m
Rango línea de visión	1,6 Km
Potencia de transmisión	100 mW (20dBm)
Sensibilidad del receptor	-100 dBm
Características Generales	
Interfaz de Datos Seriales	3.3 V CMOS UART
Métodos de Configuración	Comandos AT o API, local o sobre aire
Banda de Frecuencia	2.4 Ghz
Inmunidad a la Interferencia	DSSS (Directed Sequence Spread Spectrum)
Tasa de Comunicación serial	1200 – 115200 bps
Conversores ADC	Conversores de 10 bits (hasta 7 disponibles según la versión firmware)

I/O Digitales	Hasta 8 disponibles
PWM (Modulación de ancho del pulso)	Hasta 2 disponibles
Opciones de antena	Chip, Wire Chip, conector U.FL y RPSMA
Redes y Seguridad	
Encriptación	128 bits
Tipos de redes que soporta	Punto a punto, Punto a Multipunto, Peer to Peer y Mesh
Entrega confiable de paquetes	Retries /Acknowledgments (Reintento/ aviso de paquetes)
ID's y canales	PAN ID (personal Área Network), 64 bit IEEE MAC, 16 canales
Requerimientos de poder	
Voltaje de Alimentación	2.8 a 3.4 VDC
Corriente al Transmitir	215 mA @ 3.3 VDC
Corriente Recibir	55 mA @ 3.3 VDC
Corriente Apagado	< 10µA @ 25° C
Propiedades Mecánicas	
Dimensiones	2.438cm x 3.294cm
Temperatura de Operación	Industrial (-40 a 85°C)
Regulaciones Aprobadas	
FCC (USA)	OUR-XBee-PRO
IC (Canadá)	4214A-XBee-PRO
ETSI (Europa)	Si
C-TICK Australia	Si
Telec Japón	Si

APÉNDICE C. Especificaciones del adaptador de voltaje

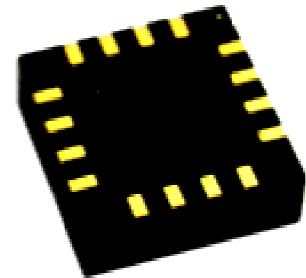
Especificaciones Generales de Adaptador de Voltaje (12V)	
Voltaje de Entrada:	110-240VAC
Voltaje de Salida:	12 VDC
Corriente de Entrada:	600 Ma
Corriente de Salida:	1.5A
Frecuencia de Operación:	50 – 60 Hz.

3-Axis Digital Compass IC HMC5883L

Honeywell

Advanced Information

The Honeywell HMC5883L is a surface-mount, multi-chip module designed for low-field magnetic sensing with a digital interface for applications such as low-cost compassing and magnetometry. The HMC5883L includes our state-of-the-art, high-resolution HMC118X series magneto-resistive sensors plus an ASIC containing amplification, automatic degaussing strap drivers, offset cancellation, and a 12-bit ADC that enables 1° to 2° compass heading accuracy. The I²C serial bus allows for easy interface. The HMC5883L is a 3.0x3.0x0.9mm surface mount 16-pin leadless chip carrier (LCC). Applications for the HMC5883L include Mobile Phones, Netbooks, Consumer Electronics, Auto Navigation Systems, and Personal Navigation Devices.



The HMC5883L utilizes Honeywell's Anisotropic Magnetoresistive (AMR) technology that provides advantages over other magnetic sensor technologies. These anisotropic, directional sensors feature precision in-axis sensitivity and linearity. These sensors' solid-state construction with very low cross-axis sensitivity is designed to measure both the direction and the magnitude of Earth's magnetic fields, from milli-gauss to 8 gauss. Honeywell's Magnetic Sensors are among the most sensitive and reliable low-field sensors in the industry.

FEATURES

BENEFITS

- | | |
|---|--|
| ▶ 3-Axis Magnetoresistive Sensors and ASIC in a 3.0x3.0x0.9mm LCC Surface Mount Package | ▶ Small Size for Highly Integrated Products. Just Add a Micro-Controller Interface, Plus Two External SMT Capacitors Designed for High Volume, Cost Sensitive OEM Designs Easy to Assemble & Compatible with High Speed SMT Assembly |
| ▶ 12-Bit ADC Coupled with Low Noise AMR Sensors Achieves 2 milli-gauss Field Resolution in ± 8 Gauss Fields | ▶ Enables 1° to 2° Degree Compass Heading Accuracy |
| ▶ Built-In Self Test | ▶ Enables Low-Cost Functionality Test after Assembly in Production |
| ▶ Low Voltage Operations (2.16 to 3.6V) and Low Power Consumption (100 μ A) | ▶ Compatible for Battery Powered Applications |
| ▶ Built-In Strap Drive Circuits | ▶ Set/Reset and Offset Strap Drivers for Degaussing, Self Test, and Offset Compensation |
| ▶ I ² C Digital Interface | ▶ Popular Two-Wire Serial Data Interface for Consumer Electronics |
| ▶ Lead Free Package Construction | ▶ RoHS Compliance |
| ▶ Wide Magnetic Field Range (± 8 Oe) | ▶ Sensors Can Be Used in Strong Magnetic Field Environments with a 1° to 2° Degree Compass Heading Accuracy |
| ▶ Software and Algorithm Support Available | ▶ Compassing Heading, Hard Iron, Soft Iron, and Auto Calibration Libraries Available |
| ▶ Fast 180 Hz Maximum Output Rate | ▶ Enables Pedestrian Navigation and LBS Applications |

HMC5883L

SPECIFICATIONS (* Tested at 25°C except stated otherwise.)

Characteristics	Conditions*	Min	Typ	Max	Units
Power Supply					
Supply Voltage	VDD Referenced to AGND	2.16	2.5	3.6	Volts
	VDDIO Referenced to DGND	1.71	1.8	VDD+0.1	Volts
Average Current Draw	Idle Mode	-	2	-	μA
	Measurement Mode (7.5 Hz ODR; No measurement average, MA1:MA0 = 00)	-	100	-	μA
	VDD = 2.5V, VDDIO = 1.8V (Dual Supply)				
	VDD = VDDIO = 2.5V (Single Supply)				

Performance

Field Range	Full scale (FS)	-8		+8	gauss
Mag Dynamic Range	3-bit gain control	±1		±8	gauss
Sensitivity (Gain)	VDD=3.0V, GN=0 to 7, 12-bit ADC	230		1370	LSb/gauss
Digital Resolution	VDD=3.0V, GN=0 to 7, 1-LSb, 12-bit ADC	0.73		4.35	milli-gauss
Noise Floor (Field Resolution)	VDD=3.0V, GN=0, No measurement average, Standard Deviation 100 samples (See typical performance graphs below)		2		milli-gauss
Linearity	±2.0 gauss input range			0.1	±% FS
Hysteresis	±2.0 gauss input range		±25		ppm
Cross-Axis Sensitivity	Test Conditions: Cross field = 0.5 gauss, Applied = ±3 gauss		±0.2%		%FS/gauss
Output Rate (ODR)	Continuous Measurement Mode	0.75		75	Hz
	Single Measurement Mode			160	Hz
Measurement Period	From receiving command to data ready		6		ms
Turn-on Time	Ready for I2C commands		200		μs
	Analog Circuit Ready for Measurements		50		ms
Gain Tolerance	All gain/dynamic range settings		±5		%
I ² C Address	8-bit read address		0x3D		hex
	8-bit write address		0x3C		hex
I ² C Rate	Controlled by I ² C Master			400	kHz
I ² C Hysteresis	Hysteresis of Schmitt trigger inputs on SCL and SDA - Fall (VDDIO=1.8V) Rise (VDDIO=1.8V)		0.2*VDDIO		Volts
			0.8*VDDIO		Volts
Self Test	X & Y Axes		±1.16		gauss
	Z Axis		±1.08		
	X & Y & Z Axes (GN=5) Positive Bias	243		575	LSb
	X & Y & Z Axes (GN=5) Negative Bias	-575		-243	
Sensitivity Tempco	T _A = -40 to 125°C, Uncompensated Output		-0.3		%/°C

General

ESD Voltage	Human Body Model (all pins)			2000	Volts
	Charged Device Model (all pins)			750	
Operating Temperature	Ambient	-30		85	°C
Storage Temperature	Ambient, unbiased	-40		125	°C



VNH2SP30-E

AUTOMOTIVE FULLY INTEGRATED H-BRIDGE MOTOR DRIVER

Table 1. General Features

Type	$R_{DS(on)}$	I_{out}	V_{CCmax}
VNH2SP30-E	19 mΩ max (per leg)	30 A	41 V

- OUTPUT CURRENT: 30A
- 5V LOGIC LEVEL COMPATIBLE INPUTS
- UNDERVOLTAGE AND OVERVOLTAGE SHUT-DOWN
- OVERVOLTAGE CLAMP
- THERMAL SHUT DOWN
- CROSS-CONDUCTION PROTECTION
- LINEAR CURRENT LIMITER
- VERY LOW STAND-BY POWER CONSUMPTION
- PWM OPERATION UP TO 20 KHz
- PROTECTION AGAINST:
 - LOSS OF GROUND AND LOSS OF V_{CC}
- CURRENT SENSE OUTPUT PROPORTIONAL TO MOTOR CURRENT
- IN COMPLIANCE WITH THE 2002/95/EC EUROPEAN DIRECTIVE

DESCRIPTION

The VNH2SP30-E is a full bridge motor driver intended for a wide range of automotive applications. The device incorporates a dual monolithic High-Side drivers and two Low-Side switches. The High-Side driver switch is designed using STMicroelectronics' well known and proven proprietary VIPower™ M0 technology that allows to efficiently integrate on the same die a true Power MOSFET with an intelligent signal/protection circuitry.

Figure 1. Package



The Low-Side switches are vertical MOSFETs manufactured using STMicroelectronics' proprietary EHD ("STripFET™") process. The three dice are assembled in MultiPowerSO-30 package on electrically isolated leadframes. This package, specifically designed for the harsh automotive environment offers improved thermal performance thanks to exposed die pads. Moreover, its fully symmetrical mechanical design allows superior manufacturability at board level. The input signals IN_A and IN_B can directly interface to the microcontroller to select the motor direction and the brake condition. The $DIAG_A/EN_A$ or $DIAG_B/EN_B$, when connected to an external pull-up resistor, enable one leg of the bridge. They also provide a feedback digital diagnostic signal. The normal condition operation is explained in the truth table on page 14. The CS pin allows to monitor the motor current by delivering a current proportional to its value. The PWM, up to 20KHz, lets us to control the speed of the motor in all possible conditions. In all cases, a low level state on the PWM pin will turn off both the LS_A and LS_B switches. When PWM rises to a high level, LS_A or LS_B turn on again depending on the input pin state.

Table 2. Order Codes

Package	Tube	Tape and Reel
MultiPowerSO-30	VNH2SP30-E	VNH2SP30TR-E

Table 3. Pin Definitions And Functions

Pin No	Symbol	Function
1, 25, 30	OUT _A , Heat Slug2	Source of High-Side Switch A / Drain of Low-Side Switch A
2,4,7,12,14,17, 22, 24,29	NC	Not connected
3, 13, 23	VCC, Heat Slug1	Drain of High-Side Switches and Power Supply Voltage
6	EN _A /DIAG _A	Status of High-Side and Low-Side Switches A; Open Drain Output
5	IN _A	Clockwise Input
8	PWM	PWM Input
9	CS	Output of Current sense
11	IN _B	Counter Clockwise Input
10	EN _B /DIAG _B	Status of High-Side and Low-Side Switches B; Open Drain Output
15, 16, 21	OUT _B , Heat Slug3	Source of High-Side Switch B / Drain of Low-Side Switch B
26, 27, 28	GND _A	Source of Low-Side Switch A (*)
18, 19, 20	GND _B	Source of Low-Side Switch B (*)

Note: (*) GND_A and GND_B must be externally connected together

Table 4. Pin Functions Description

Name	Description
VCC	Battery connection.
GND _A GND _B	Power grounds, must always be externally connected together.
OUT _A OUT _B	Power connections to the motor.
IN _A IN _B	Voltage controlled input pins with hysteresis, CMOS compatible. These two pins control the state of the bridge in normal operation according to the truth table (brake to V _{CC} , Brake to GND, clockwise and counterclockwise).
PWM	Voltage controlled input pin with hysteresis, CMOS compatible. Gates of Low-Side FETS get modulated by the PWM signal during their ON phase allowing speed control of the motor
EN _A /DIAG _A EN _B /DIAG _B	Open drain bidirectional logic pins. These pins must be connected to an external pull up resistor. When externally pulled low, they disable half-bridge A or B. In case of fault detection (thermal shutdown of a High-Side FET or excessive ON state voltage drop across a Low-Side FET), these pins are pulled low by the device (see truth table in fault condition).
CS	Analog current sense output. This output sources a current proportional to the motor current. The information can be read back as an analog voltage across an external resistor.

PUBLICACIONES



REFERENCIAS

- [1]. Everett, H.R. "Sensors for Mobile Robots: theory and application". A. K. Peters. USA. 1995.
- [2]. JONES, Joseph. "Mobile Robots: inspirations to implementation". A. K. Peters. USA. 1999.
- [3]. Cogdell, J. R., "Fundamentos de Electrónica" Edit. Prentice Hall. México, 2000.
- [4].ÁLVAREZ, Juan. "Planificación de Movimiento de Vehículos Autónomos Basada en Sensores". Tesis Doctoral. Universidad de Oviedo. Septiembre 1998.
- [5]. TLATELPA JIMENEZ ROLANDO, TITULO DE TESIS: "Control de un robot móvil utilizando tecnología inalámbrica", FECHA: 15 DE DICIEMBRE DE 2011.
- [6]. Seyr M. and Jakubek S. (2005). Mobile robot predictive trajectory tracking. Institute of Mechanics and Mechatronics, Vienna University of Technology.
- [7]. GU D. and Hu H. (2006). Receding horizon tracking control of wheeled mobile robots. IEEE Transactions on control systems technology, vol.14, No.4.
- [8]. AMIDI O. (1990). Integrated Mobile Robot Control. Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, Tech. Rep. CMU-RI-TR-90-17.
- [9]. ISIDORI A. (1995). Nonlinear Control Systems (Communications and Control Engineering). Springer, 3rd edition.
- [10]. LI X. and Zell A. (2005). Path following control for a mobile robot pushing a ball. Wilhelm-Schickard-Institute, University of Tübingen, Tübingen, Germany.
- [11]. YU L., Tsui P.W., Zhou Q., Hu H. (2001). A web-based telerobotic system for research and education at Essex. Presented at IEEE/ASME International Conf. on advanced intelligent mechatronics proceedings, Como, Italy.
- [12]. HOLMBERG R. and Khatib O. (2003). Development and control of a holonomic mobile robot for mobile manipulation tasks. International Journal of Robotics Research, vol. 19, pp. 1066-1074.
- [13]. Ollero Baturone, Anibal, "ROBOTICA, MANIPULADORES Y ROBOTS MÓVILES", Editorial MARCOMBO, Edición 1º Año: 2001, Pag.464.

[14]. Reyes Cortés Fernando, Robótica: "Control de robots manipuladores", Editorial MARCOMBO, Edición 1º Año: 2012, Pág.312.

[15] J. Yang and J. Kim, "Sliding mode control for trajectory tracking of nonholonomic wheeled mobile robots", IEEE Transactions on Robotics and Automation, 15(3): 578-587, 1999, in press.

[16]. Manigel, J. y W. Leonhard (1992), Vehicle Control by Computer Vision. IEEE Transactions on industrial electronics 39(3), 181-188.

[17]. Hunt, K. J., T.A. Johansen, J. Kalkkuhl, H. Fritz y T. Gottsche (2000). Speed Control Design for an Experimental Vehicle Using a Generalized Gain Scheduling Approach. IEEE, Transactions on Control Systems on Control Systems Technology.

[18]. Isermann, R. (2001). Diagnosis Methods for Electronic Controlled Vehicles. Vehicle System Dynamics: International Journal of Vehicle Mechanics and Mobility 36(2-3), 77-117.

[19] Guilherme V. Raffo, Julio E. Normey-Rico, Francisco R. Rubio, "Control predictivo en cascada de un vehículo autónomo", Revista Iberoamericana de Automática e Informática Industrial, vol. 6, Num. 1 pp. 63–74, January 2009, in press.

[20] Camacho, E.F. y C. Bordons, "Model predictive control", Springer-Verlag. New York, 1998, in press.

[21] Raffo, G.V., "Algoritmos de controle predictivo para Seguimiento de Trajetorias de Veiculos Autónomos. Master's Thesis," Universidade Federal de Santa Catarina, Programa de Pós-Graduação em Engenharia Eletrica. Florianopolis, SC. In portuguese, 2005., in press.

[22] Suárez Ramírez Hugo, "Tesis planificación de trayectorias optimales de un robot móvil", March 2007, unpublished.

[23] Ramos Silvestre Edgar Roberto, "Control punto a punto para el seguimietno de trayectorias de un robot móvil de ruedas tipo diferencial", June 2011, in press.

[24] Ortiz Guevara Mauricio Javier, "Diseño y construcción de un conversor DC/DC reductor elevador con control predictivo controlado por una tarjeta FPGA ," Quito, October 2011, unpublished.

[25] Bautista Herrera Mario Samir, "Diseño y desarrollo de PLAYCAT, un centro de juego electrónico interactivo para gatos", July 2014, unpublished.