

ESCUELA POLITÉCNICA NACIONAL

FACULTAD DE INGENIERÍA ELÉCTRICA Y ELECTRÓNICA

DISEÑO E IMPLEMENTACIÓN DE UN PROTOTIPO A ESCALA DE ROBOT MÓVIL ACOMPAÑANTE.

PROYECTO PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIERO EN
ELECTRÓNICA Y CONTROL

ALEJANDRO RAMIRO MESÍAS ULLOA

alejandromesiasu@gmail.com

CHRISTIAN RAMIRO LÓPEZ MOLINA

cristianramirolopez@hotmail.com

DIRECTOR: NELSON SOTOMAYOR, MSc.

nelson.sotomayor@epn.edu.ec

Quito, Junio 2012

DECLARACIÓN

Nosotros, Christian Ramiro López Molina y Alejandro Ramiro Mesías Ulloa, declaramos bajo juramento que el trabajo aquí descrito es de nuestra autoría; que no ha sido previamente presentada para ningún grado o calificación profesional; y, que hemos consultado las referencias bibliográficas que se incluyen en este documento.

A través de la presente declaración cedemos nuestros derechos de propiedad intelectual correspondientes a este trabajo, a la Escuela Politécnica Nacional, según lo establecido por la Ley de Propiedad Intelectual, por su Reglamento y por la normatividad institucional vigente.

Christian Ramiro López Molina

Alejandro Ramiro Mesías Ulloa

CERTIFICACIÓN

Certifico que el presente trabajo fue desarrollado por Christian Ramiro López Molina y Alejandro Ramiro Mesías Ulloa, bajo mi supervisión.

Nelson Sotomayor, MSc.
DIRECTOR DEL PROYECTO

AGRADECIMIENTO

Gracias TAITA Diosito, primero por haberme hecho nacer en el seno de mi hermosa familia, por haberme dejado encontrar en mi camino a grandes seres humanos, en fin gracias por todo aquello bueno y más aun, por los retos que me has colocado en el camino para que aprenda, que esta vida se la vive luchando por alcanzar lo que queremos.

Gracias Ángel, gracias Esperancita, gracias Taty, Lizette y mi angelita preciosa Nahomy Lizette, que me han estado acompañando incondicionalmente en mi vida, sin ustedes no sería posible soñar en que es posible un mundo mejor.

Y como último pero no menos importante quiero agradecer a todos mis amigos y amigas, pero principalmente a mis llullinitas Pao y Majito, porque me han acompañado en el camino y han sido una gran ayuda para entender que HAY QUE SER MAS PARA SERVIR MEJOR.

-Christian Ramiro López Molina-

AGRADECIMIENTO

A Angélica Ulloa, mi mamá, y a David, mi hermano, por toda su fe en mí, su cariño y apoyo.

A Oswaldo Mesías, mi papá, quien puso su talento para mejorar este trabajo y por toda su comprensión y soporte durante este proceso.

Al Dr. Vivar y Dra. Dávila por su confianza y ayuda.

A Nico Vivar por su apoyo y aporte para la realización del prototipo.

A la familia Ávila-Vivar, especialmente a Santi por su ayuda desinteresada.

A la familia Herdoiza-Vivar por sus ánimos y asistencia clave.

A don Edwin Rivera por su voluntad y dedicación en su contribución.

A Pao Ramos por su colaboración y compañía.

Al Ing. Nelson Sotomayor y su eficaz asesoría.

Al Club de Robótica de la Escuela Politécnica Nacional, especialmente a Nelson y Estefani.

Al talento humano dentro de organizaciones como el M.I.T., Microsoft o Google, cuyo innovador trabajo hace posible proyectos como este.

A mis amigos de colegio por su confianza y a mis colegas de carrera por todo su aliento, consejos y ayuda, especialmente a Rodrigo.

A María Cecilia Vivar, mi querida y hermosa novia, por todo su tiempo, fuerzas, soporte y mucha ayuda que fueron vitales para este proyecto en todas sus fases.

Y principalmente a Christian López, compañero y amigo, por todo su esfuerzo, paciencia y habilidad con la cuales incansablemente impulsó este proyecto a través de tantas horas de trabajo, decepciones, preocupaciones, expectativas y satisfacciones.

-Alejandro Ramiro Mesías Ulloa-

DEDICATORIA

A ustedes, quienes hacen posible el sueño...

LEV

-Christian Ramiro López Molina-

DEDICATORIA

A mi padre y madre, por quienes soy quien soy.

-Alejandro Ramiro Mesías Ulloa-

CONTENIDO

CONTENIDO	I
INDICE DE FIGURAS	IV
INDICE DE TABLAS	IX
RESUMEN	X
PRESENTACIÓN	XI
CAPÍTULO 1	1
FUNDAMENTOS BÁSICOS	1
1.1 INTRODUCCIÓN	1
1.2 SISTEMAS COMERCIALES	2
1.2.1 SHADOW CADDY	2
1.2.2 GOGO KADDY.....	3
1.3 OTROS SISTEMAS	5
1.3.1 SISTEMAS DE SEGUIMIENTO BASADOS EN ULTRASONIDO.....	5
1.3.1.1 Ultrasonic Signal Follower Robot	5
1.3.1.2 Ultrasonic Source Localizador	5
1.3.2 SISTEMAS DE SEGUIMIENTO BASADOS EN VISIÓN ARTIFICIAL	6
1.3.2.1 Gazebot.....	7
1.3.2.2 Bilibot-Follower.....	7
1.3.2.3 Autonomous Follower Transport (A.F.T).....	8
1.4 SENSORES UTILIZADOS	10
1.4.1 KINECT	10
1.4.1.1 Reseña cronológica del desarrollo de KINECT	10
1.4.1.2 Descripción del Hardware de KINECT	12
1.4.1.3 Descripción del Software de KINECT	15
1.4.2 SENSOR DE GASES	20

1.4.3	SENSOR ULTRASÓNICO	22
CAPÍTULO 2		25
DISEÑO Y CONSTRUCCIÓN DEL PROTOTIPO		25
2.2	DISEÑO MECÁNICO	26
2.2.1	ENSAMBLAJE DEL PROTOTIPO	28
2.3	DISEÑO ELECTRÓNICO Y ELÉCTRICO	31
2.3.1	ALIMENTACIÓN PARA MOTORES	31
2.3.1.1	Diseño de conversor DC/DC	33
2.3.2	ALIMENTACIÓN PARA PLACAS DE CONTROL	35
2.3.2.1	Fuente de 5 voltios	35
2.3.2.2	Fuente de 3,1 voltios	36
2.3.3	MICROCONTROLADOR ATmega164P	37
2.3.3.1	Análisis de recursos del microcontrolador utilizados	38
CAPÍTULO 3		46
DESARROLLO DEL PROGRAMA DE CONTROL		46
3.1	INTRODUCCIÓN	46
3.2	ESTRUCTURA DEL SISTEMA DE CONTROL	46
3.2.1	MODOS GENERALES DE FUNCIONAMIENTO	47
3.2.2	OPERACIÓN GLOBAL DEL ROBOT ACOMPAÑANTE	47
3.3	PROGRAMA DEL MICROCONTROLADOR	50
3.3.1	PROGRAMA PRINCIPAL	50
3.3.1.1	Configuración inicial	50
3.3.1.2	Tarea 1: Lectura de canales analógicos	51
3.3.1.3	Tarea 2: Recepción y clasificación de códigos recibidos de la interfaz del teléfono 52	
3.3.1.4	Tarea 3: Determinación del estado actual del robot	53
3.3.1.5	Tarea 4: Ejecución de órdenes del estado actual del robot	55
3.3.1.6	Tarea 5: Ejecución de la locomoción del robot	56
3.3.2	SUBROUTINA DE INTERRUPCIÓN POR RECEPCIÓN DE DATOS DESDE LA COMPUTADORA	57

3.3.3	SUBROUTINA DE MANEJO	58
3.3.4	SUBROUTINA DE ULTRASÓNICOS.....	59
3.3.5	SUBROUTINA DE COMPENSADORES.....	59
3.3.6	SUBROUTINA DE TRACCIÓN DIFERENCIAL	63
3.4	RESUMEN DE LA CODIFICACIÓN PARA COMUNICACIONES	66
3.5	PROGRAMA DE LA INTERFAZ DEL TELÉFONO.....	66
3.6	PROGRAMA DE LA COMPUTADORA PARA EL SENSOR KINECT	71
3.6.1	CONSIDERACIONES GENERALES EN PROGRAMAS PARA MANEJO DEL SENSOR KINECT	71
3.6.1.1	Características del recurso SkeletalTracking.....	72
3.6.2	DESCRIPCIÓN DEL PROGRAMA DE COMPUTADORA DESARROLLADO PARA EL ROBOT ACOMPAÑANTE	74
3.6.2.1	Programación en el evento SkeletonFrameReady	75
3.6.2.2	Programación en el evento de recepción.....	76
3.6.2.3	Programación en el evento DepthFrameReady.....	76
CAPITULO 4	79
PRUEBAS Y RESULTADOS	79
4.1.	LINEALIZACIÓN DE LA RESPUESTA DE VELOCIDAD DE MOTORES	79
4.1.1.	ADQUISICIÓN DE DATOS DE VELOCIDAD	79
4.1.2.	LINEALIZACIÓN	80
4.2.	EXPLICACIÓN DEL SISTEMA DE COORDENADAS DEL PROTOTIPO	81
4.3.	PROCEDIMIENTO PARA DETERMINAR LOS CONTROLADORES DE GIRO Y DE AVANCE	84
4.3.1.	ADQUISICIÓN DE DATOS DE LA PLANTA DE AVANCE	84
4.3.2.	IDENTIFICACIÓN DE LA PLANTA DE AVANCE.....	87
4.3.3.	ADQUISICIÓN DE DATOS DE LA PLANTA DE GIRO	90
4.3.4.	IDENTIFICACIÓN DE LA PLANTA DE GIRO	94
4.4.	DISEÑO DE COMPENSADORES	96
4.4.1.	COMPENSADOR PARA GIRO	96
4.4.2.	COMPENSADOR DE ALCANCE	100

4.5. PRUEBAS DE DESEMPEÑO.....	104
4.5.1. PRUEBA EN TRAYECTORIA RECTA	105
4.5.2. PRUEBA EN TRAYECTORIAS CURVAS HACIA LA DERECHA E IZQUIERDA.....	107
4.5.3. PRUEBAS DE FUNCIONAMIENTO GLOBAL EN TRAYECTORIAS NO CONTROLADAS DE CORTA DURACIÓN.....	109
4.5.4. PRUEBAS DE FUNCIONAMIENTO GLOBAL EN AMBIENTES REALES	110
4.5.5. PRUEBAS CON PERSONAS AJENAS AL DESARROLLO DEL PROTOTIPO	112
COSTOS DEL PROYECTO.....	112
CAPITULO 5.....	114
CONCLUSIONES Y RECOMENDACIONES.....	114
CONCLUSIONES.....	114
RECOMENDACIONES.....	117
REFERENCIAS BIBLIOGRÁFICAS.....	118
ANEXOS	

INDICE DE FIGURAS

Figura 1. 1 Logotipo de la segunda conferencia internacional sobre HRI de la IEEE en 2007, tomado de [2].	1
Figura 1. 2 Shadow caddy en el campo de golf, tomado de [3].	2
Figura 1. 3 Gogo Kaddy y su transmisor, tomado de [9].	4
Figura 1. 4 Robot con sus receptores y su transmisor, tomado de [14]......	5
Figura 1. 5 Robot Ultrasonic Source Localizator en seguimiento, tomado de [11].	6
Figura 1. 6 Gazebot detectando el color de camiseta de la persona, tomado de [12].	7
Figura 1. 7 Visión de Bilibot durante el seguimiento, tomado de [14].....	8
Figura 1. 8 Diagrama de bloques, tomado de [15]......	8
Figura 1. 9 Comportamiento del vehículo, de acuerdo a la intensidad de señal del señalizador, tomado de [15]	9
Figura 1. 10 Rango de detección, tomado de [15].....	9
Figura 1. 11 Imagen promocional de KINECT para la Xbox 360, tomado de [18]	10
Figura 1. 12 Control con el cuerpo del personaje de un juego, tomado de [18] ..	10
Figura 1. 13 Imagen promocional de KINECT para Windows 7, tomado de [24]	12

Figura 1. 14 Distribución y relación de los elementos del KINECT, tomado de [25]	12
Figura 1. 15 Diagrama de la tecnología patentada por PrimeSense, tomado de [26]	13
Figura 1. 16 Fotografías con cámaras de visión nocturna de la proyección del patrón infrarrojo, que es invisible a la visión humana, tomado de [27]	13
Figura 1. 17 Rango de percepción de profundidad de KINECT, tomado de [28]	14
Figura 1. 18 Tipos de datos entregados por el sensor, tomado de [20]	14
Figura 1. 19 Relación entre el Hardware y Software de KINECT, tomado de [29]	15
Figura 1. 20 Diagrama detallado de los elementos de hardware y software en la utilización de KINECT para la PC, tomado de [29]	15
Figura 1. 21 Imagen de profundidad procesada para representarla en una perspectiva lateral, lo cual enfatiza su carácter tridimensional, tomado de [31]	17
Figura 1. 22 Segmentos en los que se clasifican los pixeles de la imagen de profundidad, tomado de [30]	17
Figura 1. 23 Las cruces son pixeles a clasificarse y los círculos pixeles de comparación. Una misma manera de comparar, dará distintas respuestas de diferencia de profundidad dependiendo de la ubicación del pixel X, tomado de [30]	18
Figura 1. 24 Árboles de comparación, para asignar una probabilidad P de pertenecer a un segmento c de cuerpo, tomado de [30]	18
Figura 1. 25 Imágenes de profundidad y su correspondiente reconocimiento de segmentos, tanto con las imágenes sintéticas del entrenamiento y con imágenes reales de KINECT, tomado de [30]	20
Figura 1. 26 Partes del sensor, tomado de [32]	21
Figura 1. 27 Circuito esquemático del sensor, tomado de [32]	21
Figura 1. 28 Diagrama de conexiones, tomado de [32]	21
Figura 1. 29 Características del haz del sensor MAXSONAR-EZ1, tomado de [33]	22
Figura 1. 30 Sensor MAXSONAR-EZ1, tomado de [33]	23
Figura 1. 31 Circuito interno del sensor LV-MaxSonar-EZ1, tomado de [33]	24
Figura 2. 1 Arquitectura del prototipo	25
Figura 2. 2 Cuerpo principal	26
Figura 2. 3 Gabinete para computadora	26
Figura 2. 4 Gabinete para carga	27
Figura 2. 5 Llantas del prototipo	27
Figura 2. 6 Montaje de motores	28
Figura 2. 7 Sistema de acople y montaje de llantas	29
Figura 2. 8 Montaje de elementos eléctricos y electrónicos	30

Figura 2. 9 Colocación de gabinetes	30
Figura 2. 10 Colocación de equipos	31
Figura 2. 11 Modelo completo	31
Figura 2. 12 Curva, tiempo de descarga vs. Corriente de descarga, tomado de [34]	33
Figura 2. 13 Conversor DC/DC.....	34
Figura 2. 14 Esquema de fuente de 5V	35
Figura 2. 15 Esquema de fuente de 3,1V	36
Figura 2. 16 Microcontrolador ATmega164P, tomado de [36]	38
Figura 2. 17 Circuito de disparo de LEDs indicadores.....	39
Figura 2. 18 Divisor de voltaje para envío de datos.....	41
Figura 2. 19 Puente de transistores.....	42
Figura 2. 20 Circuito lógico, asociado al puente de transistores.....	42
Figura 2. 21 Configuración pull-down	43
Figura 2. 22 Configuración pull-up.....	43
Figura 3. 1 Relaciones entre los componentes sobre los que actúa el sistema de control	47
Figura 3. 2 Luces que indican que el robot está encendido y en el modo de paro.	48
Figura 3. 3 Botón de la interfaz para manejo y luces indicadoras para este modo.	48
Figura 3. 4 Botón de la interfaz para seguimiento en espera y luces indicadoras para este modo.	49
Figura 3. 5 Esquema de posición inicial del usuario en el campo de vista del KINECT y luces indicadoras para el modo de seguimiento listo	49
Figura 3. 6 Esquema del usuario siendo seguido, dentro del campo visual del KINECT y luces indicadoras para este modo.....	50
Figura 3. 7 Diagrama de flujo de Configuración Inicial y la Tarea 1	52
Figura 3. 8 Diagrama de flujo de la Tarea 2	53
Figura 3. 9 Diagrama de flujo de la Tarea 3	54
Figura 3. 10 Diagrama de flujo de la Tarea 4 y 5.....	56
Figura 3. 11 Diagrama de flujo de la subrutina de interrupción por recepción.....	57
Figura 3. 12 Diagrama de flujo de la subrutina de manejo	58
Figura 3. 13 Diagrama de flujo de la subrutina de ultrasónicos	59
Figura 3. 14 Diagrama de flujo de la subrutina de compensadores.....	62
Figura 3. 15 Diagrama de flujo de la subrutina de tracción diferencial	65
Figura 3. 16 Pantalla de programación de la interfaz gráfica.	67
Figura 3. 17 Panel inicial y de paro	68
Figura 3. 18 Panel de manejo.....	69
Figura 3. 19 Panel de seguimiento.	70

Figura 3. 20 Junturas del esqueleto calculadas, tomado de [29].....	72
Figura 3. 21 Sistema de coordenadas de las posiciones espaciales de esqueletos y sus junturas, tomado de [29]	74
Figura 3. 22 Programa para KINECT donde Imagen de profundidad se muestra en escala de grises en estados de no seguimiento.....	77
Figura 3. 23 Programa para KINECT donde Imagen de profundidad se muestra personas azules en estado de seguimiento en espera.	77
Figura 3. 24 Programa para KINECT donde la imagen de profundidad muestra al usuario en verde en estado de seguimiento listo	78
Figura 3. 25 Programa para KINECT donde la imagen de profundidad muestra al usuario seguido en naranja, en estado de seguimiento	78
Figura 4. 1 Modelo de la prueba para adquisición de datos de velocidad	79
Figura 4. 2 Curvas de aproximación para linealización de velocidad de los motores	81
Figura 4. 3 Resultados de linealización	81
Figura 4. 4 Relación espacial entre los ejes del sensor Kinect y los ejes particulares del sistema del robot.....	82
Figura 4. 5 Captura de interfaz, con datos de posición del usuario, en los dos sistemas de coordenadas.....	83
Figura 4. 6 Prueba para la planta de avance en lazo abierto	85
Figura 4. 7 Datos de posición en profundidad durante la prueba para la planta de avance en lazo abierto.	86
Figura 4. 8 Interpretación de la prueba de profundidad como prueba de desplazamiento frontal del robot a lazo abierto	87
Figura 4. 9 Datos de entrada y salida de la prueba de desplazamiento frontal introducidos al <i>System Identification Tool</i>	88
Figura 4. 10 Generación de modelos para la planta de avance en el <i>System Identification Tool</i>	89
Figura 4. 11 Modelo exportado al <i>Workspace</i> y visualizado en forma de polos y ceros	89
Figura 4. 12 Comparación y confirmación del modelo elegido para la planta de avance.....	90
Figura 4. 13 Prueba para la planta de giro en lazo abierto, hacia la izquierda	91
Figura 4. 14 Prueba para la planta de giro en lazo abierto, hacia la derecha.....	91
Figura 4. 15 Datos de posición horizontal durante las pruebas para la planta de giro en lazo abierto.....	92
Figura 4. 16 Interpretación de la prueba de posición horizontal como prueba de desplazamiento lateral del robot a lazo abierto	93
Figura 4. 17 Datos de entrada y salida de la prueba de desplazamiento lateral introducidos al <i>System Identification Tool</i>	94

Figura 4. 18 Generación de modelos para la planta de giro en el <i>System Identification Tool</i>	95
Figura 4. 19 Modelo exportado al <i>Workspace</i> y visualizado en forma de polos y ceros	95
Figura 4. 20 Comparación y confirmación del modelo elegido para la planta de giro	96
Figura 4. 21 LGR y diagrama de Bode de la planta de giro en lazo abierto	97
Figura 4. 22 Datos finales del compensador	98
Figura 4. 23 Gráficas de respuesta con compensador manipulado.....	99
Figura 4. 24 LGR y diagrama de Bode de la planta de alcance en lazo abierto	100
Figura 4. 25 Datos finales del compensador	102
Figura 4. 26 Gráficas de respuesta con compensador manipulado.....	103
Figura 4. 27 Prueba de trayectoria recta	104
Figura 4. 28 Prueba de trayectoria curva a la derecha	104
Figura 4. 29 Prueba de trayectoria curva a la izquierda	104
Figura 4. 30 Alineación del prototipo con la trayectoria	105
Figura 4. 31 Ubicación del usuario para iniciar prueba.....	105
Figura 4. 32 Imágenes de prueba en trayectoria recta	105
Figura 4. 33 Comportamiento en alcance del prototipo, durante seguimiento en trayectoria recta.....	106
Figura 4. 34 Comportamiento en giro del prototipo, durante seguimiento en trayectoria recta.....	106
Figura 4. 35 Imágenes de prueba en trayectoria curva	107
Figura 4. 36 Comportamiento en alcance del prototipo, durante seguimiento en trayectoria curva a la derecha	107
Figura 4. 37 Comportamiento en alcance del prototipo, durante seguimiento en trayectoria curva a la izquierda.....	108
Figura 4. 38 Comportamiento en giro del prototipo, durante seguimiento en trayectoria curva a la derecha	108
Figura 4. 39 Comportamiento en giro del prototipo, durante seguimiento en trayectoria curva a la izquierda.....	109
Figura 4. 40 Trayectoria secante a la curva descrita por el prototipo	109
Figura 4. 41 Ejemplo de seguimiento en una trayectoria curva derecha	110
Figura 4. 42 Ejemplo de seguimiento en una trayectoria curva izquierda	110
Figura 4. 43 Ejemplo de seguimiento en una trayectoria recta.....	110
Figura 4. 44 Ejemplo de recorrido por un supermercado	111
Figura 4. 45 Ejemplo de recorrido por un centro comercial	111
Figura 4. 46 Ejemplo de recorrido por un centro comercial	111
Figura 4. 47 Prueba con un niño de 10 años de edad.....	112
Figura 4. 48 Prueba con una mujer de 24 años de edad.....	112
Figura 4. 49 Prueba con una mujer de 25 años de edad.....	112

INDICE DE TABLAS

Tabla 1. 1 Características de funcionamiento del MAXSONAR-EZ1, tomado de [33]	23
Tabla 2. 1 Características de motor DC	28
Tabla 2. 2 Características mecánicas del prototipo	28
Tabla 2. 3 Características de batería, tomado de [34]	31
Tabla 2. 4 Características del transistor TIP147, tomado de [37]	34
Tabla 2. 5 Características del transistor TIP142, tomado de [38]	34
Tabla 2. 6 Características de batería, tomado de [35]	35
Tabla 2. 7 Asignación de pines para LEDs indicadores	39
Tabla 2. 8 Asignación de pines para entradas A/D	40
Tabla 2. 9 Asignación de pines para comunicación serial	41
Tabla 2. 10 Funcionamiento del motor según combinaciones del circuito lógico. 43	
Tabla 2. 11 Asignación de pines para control de motores	45
Tabla 3. 1 Rangos del byte de comunicación	66
Tabla 4. 1 Detalle de costos del proyecto	112

RESUMEN

Este proyecto presenta el diseño e implementación de un robot móvil acompañante, haciendo énfasis en la interacción robot-humano, y aplicando nuevas herramientas en su control. Dicho robot es de tipo terrestre rodante de tracción diferencial, con un sistema de control basado en microprocesadores.

Como base de su funcionamiento se ha implementado un sistema de seguimiento visual de corto alcance, utilizando el sensor de movimiento "KINECT" para el reconocimiento de la forma humana y de su posición tridimensional relativa al sensor. Por lo tanto, el prototipo es capaz de acompañar a un usuario que solicite el acompañamiento mientras se mantenga en el campo de visión del sensor, siguiéndolo de cerca durante una caminata a ritmo moderado y de trayectoria horizontal sin desniveles. La adquisición y análisis de la posición de la persona se realiza a través del Kit de Desarrollo de Software de KINECT (KINECT SDK), para la plataforma Visual Studio de Microsoft, ejecutándose en una computadora portátil a bordo del robot móvil.

Se implementó una interfaz hombre-máquina para un teléfono inteligente con sistema operativo Android, con la cual el usuario puede activar o desactivar el modo de seguimiento; con el seguimiento desactivado, la interfaz le permite al usuario manejar el robot remota e inalámbricamente en espacios próximos.

El robot cuenta con los sensores adecuados para evitar colisiones durante la navegación; cuenta además con un sensor para detección de gases nocivos inflamables, pudiendo el usuario recibir el estado de este último por medio de la interfaz del teléfono.

Este robot puede convertirse en una ayuda válida para personas con discapacidad, en rehabilitación o de edad avanzada, al llevar los objetos que ellos no puedan cargar, entre otras aplicaciones.

PRESENTACIÓN

El presente proyecto de titulación se encuentra dividido en cinco capítulos. El primer capítulo describe los fundamentos básicos para el desarrollo del prototipo empezando con la descripción de proyectos comerciales y no comerciales similares con sus distintos sistemas de seguimiento. Luego se presenta la descripción teórica de los componentes utilizados en el proyecto.

El segundo capítulo, describe el diseño y construcción del prototipo, en primer lugar se describe el diseño mecánico y ensamblaje del prototipo y a continuación el diseño eléctrico y electrónico.

El tercer capítulo, describe el desarrollo del programa de control. Se presenta primero la estructura del sistema de control, para luego continuar detallando la programación del microcontrolador y las interfaces programadas para la computadora portátil y el teléfono inteligente, involucrados en el proyecto.

En el cuarto capítulo, se muestran las pruebas y resultados experimentales de la implementación del prototipo. Primero las calibraciones necesarias para tener el comportamiento deseado del prototipo y segundo las pruebas de desempeño global del sistema.

El quinto capítulo, contiene las conclusiones y recomendaciones que se han podido determinar al culminar la realización del proyecto. El objetivo es comprobar el cumplimiento de los alcances planteados para el proyecto y presentar posibles ampliaciones a la investigación.

CAPÍTULO 1

FUNDAMENTOS BÁSICOS

1.1 INTRODUCCIÓN

Actualmente, los robots han conquistado el ámbito industrial de modo que pueden ser encontrados en casi cualquier fábrica. En este ambiente, la operación de los robots conlleva un cierto nivel de riesgo para las personas que también laboran con ellos o junto a ellos. Este problema generalmente ha sido solucionado restringiendo el acceso de las personas a la zona de trabajo del robot e incluso colocando jaulas alrededor de ellos [1].

Sin embargo, la expansión de la robótica ha llevado al desarrollo de robots de servicio fuera del ámbito industrial y dentro de ambientes cotidianos, como por ejemplo robots de limpieza, educacionales, asistentes en oficinas, asistentes médicos, de vigilancia, acompañamiento o para tele-presencia. Estos robots de servicio, muchas veces móviles, no pueden trabajar bajo el paradigma de restringir la interacción física con el humano [1]. Por el contrario, dichos robots necesitan una interacción cada vez más profunda, cercana y compleja con las personas. De aquí nace el campo de la robótica llamado Interacción Humano-Robot o HRI por sus siglas en inglés.



Figura 1. 1 Logotipo de la segunda conferencia internacional sobre HRI de la IEEE en 2007, tomado de [2].

HRI es un área de estudio interdisciplinario sobre cómo los humanos interactúan con los robots y acerca de cómo diseñar sistemas capaces de ejecutar tareas

interactivas en ambientes humanos. El desarrollo de HRI abarca desde nuevos sistemas de instrumentación, pasando por inteligencia artificial y llega incluso a la psicología y antropología [1].

Para la HRI, el robot de servicio necesita sensores distintos a los que normalmente se manejan en ambientes industriales para que tengan nuevas capacidades. Por ejemplo construir modelos dinámicos de sus alrededores, reconocer y categorizar objetos, reconocer y localizar humanos o incluso saber su estado emocional [1].

Es en este marco de estudio que este proyecto pretende hacer un aporte al construir un prototipo de robot acompañante, que como tarea principal siga a su usuario humano durante una caminata mediante el empleo de nueva tecnología de reconocimiento visual de humanos.

1.2 SISTEMAS COMERCIALES

1.2.1 SHADOW CADDY

Shadow Caddy es un vehículo robótico de tres ruedas que cumple las funciones de un caddie humano al seguir a su usuario por el campo de golf, cargando la bolsa de palos. A diferencia de vehículos anteriores con similar propósito, Shadow Caddy no requiere de un control remoto manual para ser manejado, sino que se guía por la señal de radiofrecuencia de un transmisor que porta el golfista en su cintura [3].



Figura 1. 2 Shadow caddy en el campo de golf, tomado de [3].

Inventado por los australianos Hubert Novak y Michael Hamilton, el desarrollo del Shadow Caddy tomó 3 años y 3 millones de dólares, saliendo al mercado en el año 2008. Con un precio cercano a los 5000 USD no se comercializa al por menor, sino que se los vende a campos de golf y estos a su vez los alquilan a sus clientes para cada juego [4].

La distancia a la que sigue al golfista es ajustable, pero está alrededor de los 1,8 metros. El vehículo cuenta con 6 sensores infrarrojos distribuidos en 270° alrededor de la nariz del vehículo, que constituyen el sistema de evasión y detención ante obstáculos. El sistema de tracción es capaz de sortear pendientes de hasta 35° [4].

Los detalles del sistema de seguimiento de la señal de RF del transmisor que el golfista debe llevar en la parte de atrás de su cinturón se mantienen en reserva, al igual que del sistema de recepción y procesamiento de señal. Solo se sabe que usa dos antenas receptoras ubicadas en la parte frontal del robot [5].

Lo que sí se conoce es que la señal es muestreada y procesada 50 veces por segundo por un sistema de control equivalente a 3 CPUs de computadora y que a diferencia de lo que se pueda pensar, el sistema no se apoya en receptores de GPS, debido a que la resolución del GPS está en el orden de los metros. Mientras tanto, con el sistema utilizado se puede lograr una resolución de 3cm, según afirma su inventor. Como interfaz de usuario el robot tiene sobre su nariz dos botones: un botón pulsador para elegir la activación o desactivación del seguimiento y otro botón para elegir la marcha en reversa con guía manual. Esta función está pensada para cuando el robot se atasque accidentalmente en lugares estrechos [4].

1.2.2 GOGO KADDY

GoGo Kaddy es también un vehículo robótico que tiene la tarea de seguir a su usuario sobre el campo de golf con sus palos. El usuario debe llevar en su espalda un pequeño dispositivo como el indicado en la Figura 1.3 para posibilitar el seguimiento.

Tiene su origen en una patente de un prototipo italiano desarrollado por la empresa O.M.G. proveedora principalmente de partes automotrices. Dicho prototipo recibió la medalla de oro en el año 2003 del “Salon International des Inventions” de Ginebra, Suiza. Sin embargo es solo en los últimos años que, tras evolucionar el producto, GoGo Kaddy ha comenzado a producirse y comercializarse masivamente, especialmente en Alemania en cooperación con empresas de ese país [6].

A diferencia de Shadow Caddy, el sistema de seguimiento que permite al vehículo rodar manteniendo una distancia de aproximadamente 2 metros no se basa en radiofrecuencia sino en un emisor de ultrasonido que debe ser portado por el golfista en su cinturón [7]. Mayores detalles acerca del sistema de seguimiento no están disponibles al público.



Figura 1. 3 Gogo Kaddy y su transmisor, tomado de [9].

La marcha o interrupción del seguimiento se controla desde el transmisor al prenderlo o apagarlo, pero el robot también cuenta con un panel trasero con controles para su manejo manual en caso de requerirlo así como para notificaciones del estado del sistema como batería, etc. [7].

A diferencia de Shadow Caddy, el GoGo Kaddy se comercializa a los usuarios individuales. Por ello su diseño es compacto y plegable, pensando en que el golfista pueda transportarlo en la cajuela de su auto. Su precio es de 3990 euros [8].

1.3 OTROS SISTEMAS

1.3.1 SISTEMAS DE SEGUIMIENTO BASADOS EN ULTRASONIDO

Un método relativamente común para lograr que un robot móvil sea capaz de reconocer y seguir a un dispositivo es colocar uno o varios emisores de ultrasonido en el dispositivo y hacer que el robot determine la direccionalidad de las ondas emitidas, con sus receptores de ultrasonido.

A continuación se reseñan ejemplos de prototipos que utilizan este tipo de sistema.

1.3.1.1 Ultrasonic Signal Follower Robot

Desarrollado en 2004 por Erkan Tuncali, estudiante de la universidad Hacettepe de Estambul, Turquía [10].



Figura 1. 4 Robot con sus receptores y su transmisor, tomado de [14].

“El emisor transmite una señal ultrasónica en un formato predeterminado” [8]. El robot recibe dicha señal en dos receptores de ultrasonido separados, que asemejan dos antenas y avanza en la dirección por la cual recibe el ultrasonido. El vehículo es de tracción diferencial y el sistema de control tiene dos microcontroladores PIC16F84A [10].

1.3.1.2 Ultrasonic Source Localizer

Es un proyecto de pregrado realizado en 2010 en la Washington University de San Luis. Tiene por enfoque el que un robot móvil pueda seguir a otro robot móvil [11].

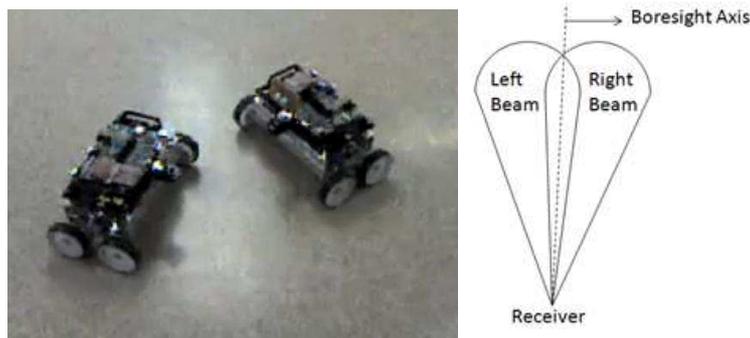


Figura 1. 5 Robot Ultrasonic Source Localizer en seguimiento, tomado de [11].

El primer robot lleva un transmisor de ultrasonido en su parte posterior y el segundo tiene dos receptores ligeramente separados y montados sobre un servomotor. El programa de control compara el valor RMS de la señal de ultrasonido recibida por cada receptor y hace girar el servomotor hacia el que mayor magnitud recibe para que entonces el valor recibido por cada sensor se equipare, indicando así al sistema motor, en qué dirección se encuentra el robot líder [11].

1.3.2 SISTEMAS DE SEGUIMIENTO BASADOS EN VISIÓN ARTIFICIAL

Los robots que utilizan visión artificial para el seguimiento no requieren de un dispositivo activo portado por el usuario. En lugar de ello suelen utilizar características visuales que destacan en el campo visual de la cámara. La característica visual que más comúnmente se utiliza en estos tipos de sistemas es el color. Es decir que la persona o sujeto a seguir debe portar una marca visible de un color específico, o su ropa debe ser de un color fuerte y distintivo.

Menos comunes y más complejos son los sistemas que logran reconocer visualmente la estructura corporal de una persona en lugar de simplemente una marca distintiva. Estos sistemas pueden distinguir a una persona del resto del ambiente. Para lograr este reconocimiento corporal no se utilizan cámaras normales sino de tipo tridimensional, como el sensor visual KINECT recientemente desarrollado por Microsoft para la consola de juegos Xbox 360, pero que está siendo usado de manera experimental en la robótica.

A continuación ejemplos de robots con sistemas visuales.

1.3.2.1 Gazebot

Este es un robot desarrollado por Hemin Shekh Omer en la universidad Nottingham Trent University de Inglaterra en 2008. El principal propósito de Gazebot es ser un dispositivo de tele-presencia de modo que una persona pueda interactuar con otras a distancia, mediante el robot y observar a través de su cámara. Como una funcionalidad importante para a este propósito está el que el robot pueda seguir por si solo a la persona en la que el teleoperador está interesado. Para ello, el diseñador ha desarrollado un programa para que mediante visión artificial, el programa de control ejecutado en la PC del teleoperador indique el color de ropa de la persona de interés y haga que el robot lo siga automáticamente [12].



Figura 1. 6 Gazebot detectando el color de camiseta de la persona, tomado de [12].

1.3.2.2 Bilibot-Follower

No es un robot en particular, sino que se trata de un proyecto de plataforma robótica muy avanzada pero económica para la investigación y desarrollo de múltiples aplicaciones robóticas. Es diseñado en el MIT (Massachusetts Institute of Technology) por Garratt Gallagher [13].

El primer modelo de la plataforma Bilibot está listo desde abril de 2011. Entre otras características, Gallagher le dio la funcionalidad de seguir a personas. Las partes constitutivas del Bilibot, durante su fase de prototipo que es la que se observa en la Figura 1.7, son la plataforma robótica básica iRobot Create (de los fabricantes de la aspiradora robótica Roomba), el sensor KINECT y una

computadora laptop como elemento de control corriendo el sistema operativo ROS, de uso libre, para robots, basado en Linux [14].



Figura 1. 7 Visión de Bilibot durante el seguimiento, tomado de [14]

Las innovaciones que hace posible conocer la ubicación de las personas para seguirlas, son la imagen de profundidad que captura el sensor KINECT, y su software de reconocimiento de la figura humana [14]. Este sensor fue originalmente pensado sólo para el control de videojuegos a través de movimientos corporales, pero su empleo en la robótica se está extendiendo. El funcionamiento de este innovador sensor se detalla más adelante.

1.3.2.3 Autonomous Follower Transport (A.F.T)

Se trata de un vehículo que sigue a un usuario portador de un dispositivo señalizador que emite una señal infrarroja codificada. A continuación se muestra un diagrama de bloques que ilustra el funcionamiento del vehículo.

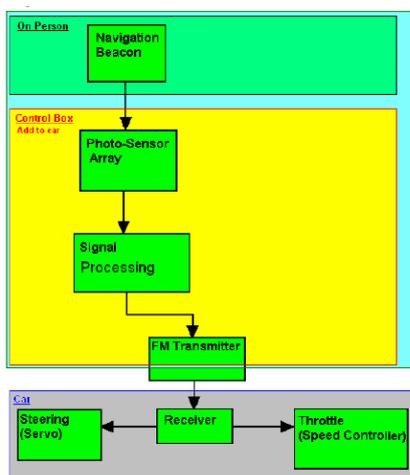


Figura 1. 8 Diagrama de bloques, tomado de [15]

El dispositivo señalizador que porta confortablemente el usuario a quien el vehículo debe seguir, emite una señal pulsante de IR (infrarrojo). El arreglo de foto sensores contiene los sensores necesarios para leer la señal IR emitida desde el dispositivo señalizador para determinar la intensidad de la aceleración y la dirección de seguimiento. El procesamiento de la señal se hace mediante un FPGA con su respectiva interface de conversión análoga/digital. Una vez realizado el algoritmo de control el FPGA se comunica a través a su interfaz digital/análoga para controlar el sistema de aceleración y dirección del vehículo.

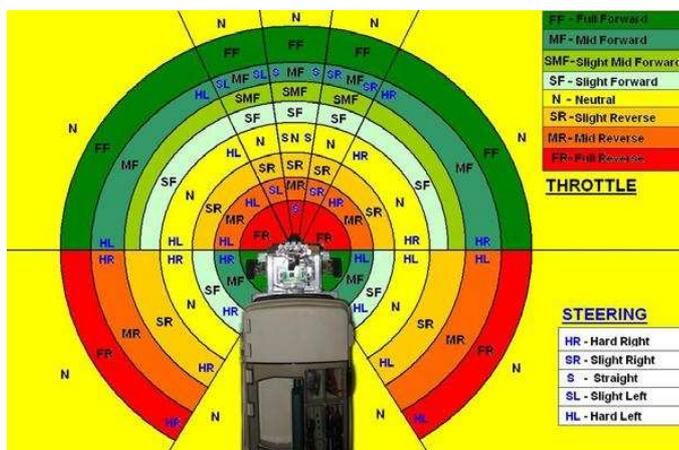


Figura 1. 9 Comportamiento del vehículo, de acuerdo a la intensidad de señal del señalizador, tomado de [15]

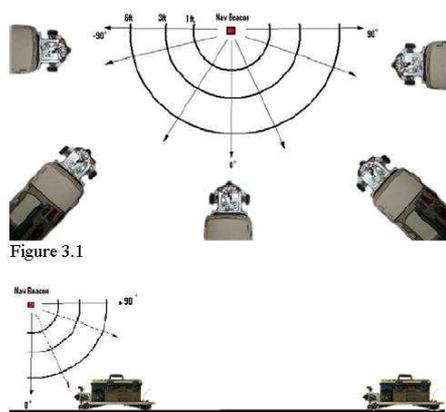


Figura 1. 10 Rango de detección, tomado de [15]

La Figura 1.10, muestra el amplio rango de detección con que cuenta el vehículo respecto del dispositivo señalizador. Además del amplio rango de detección que presenta el vehículo este es capaz de funcionar sin inconvenientes en espacios abiertos como en cerrados, ya que, cuenta con un filtrado del rango infrarrojo de la luz del sol u otras fuentes. Además gracias a que el dispositivo señalizador

emite la señal infrarroja con una frecuencia específica se puede distinguir entre varios dispositivos señalizadores.

1.4 SENSORES UTILIZADOS

1.4.1 KINECT

KINECT es un dispositivo adicional lanzado por Microsoft para su consola de juegos Xbox 360 en noviembre de 2010. Dicho dispositivo es un sensor, que tiene el propósito de reconocer el cuerpo humano, para permitir a los jugadores interactuar con los videojuegos mediante movimientos corporales, gestos o comandos de voz, sin necesidad de manipular controles o portar aparatos, impulsando de esta manera una participación más interactiva, física e intuitiva en los juegos para usuarios expertos pero sobre todo para los novatos [19].



Figura 1. 11 Imagen promocional de KINECT para la Xbox 360, tomado de [18]



Figura 1. 12 Control con el cuerpo del personaje de un juego, tomado de [18]

1.4.1.1 Reseña cronológica del desarrollo de KINECT

La tecnología del sensor KINECT, para reconocer la figura humana y las poses del usuario, es el resultado de una conjunción de sistemas de hardware y software desarrollados por diferentes equipos.

El hardware de KINECT no fue desarrollado por Microsoft directamente sino que se trata de una colaboración y adquisición, en 2008, de una licencia a PrimeSense, una empresa israelí, diseñadora de cámaras tridimensionales, es decir, que pueden reportar de la imagen tanto el color capturado por cada pixel como también la correspondiente información de profundidades en la escena [21].

El Hardware desarrollado por PrimeSense es fundamental, pero no es suficiente para reconocer, en tiempo real, dentro de la imagen tridimensional a la figura humana y sus poses. Por ello, en junio de 2008, Microsoft convocó al grupo *Machine Learning & Perception Group* en la Universidad de Cambridge, con expertos en visión por computadora y aprendizaje de máquina, para el desarrollo de los algoritmos de software necesarios para realizar dicha tarea [22].

En junio de 2009 Microsoft dio a conocer los primeros resultados de esta nueva tecnología bajo el nombre provisional de “Proyecto Natal” mediante una conferencia con juegos demostrativos controlados con gestos corporales [23].

El 4 de noviembre de 2010 se puso ya a la venta el sensor con su nombre definitivo “KINECT” junto a la actualización de software para el reconocimiento de poses humanas en la consola Xbox 360 [30]. Desde su lanzamiento hasta marzo de 2010 se vendieron 10 millones de unidades KINECT a un precio de \$150 en Estados Unidos [17].

A un mes de su lanzamiento, en diciembre de 2010 PrimeSense, diseñadores del Hardware, hicieron públicos los drivers que ellos usan para conectar el sensor KINECT a una PC, y no sólo a la Xbox, y poder recibir vía USB la información del sensor en el computador. Luego difundieron además, en formato de software libre, su propia *Interfaz de Programación de Aplicaciones* (API) que había sido el software prototipo para el reconocimiento de la Figura humana, y que es diferente de la API desarrollada oficialmente por Microsoft para la Xbox360. Con estos elementos en la mano, inventores aficionados e investigadores universitarios comenzaron a desarrollar durante 2011, aplicaciones experimentales del KINECT aparte de los videojuegos, en entornos de software libre [19].

Microsoft en un principio reaccionó con reserva a esta utilización imprevista de su hardware con software no oficial. Sin embargo, en vista de la difusión de esta práctica, publicaron un paquete con software de drivers y APIs oficiales para Windows 7. Este software oficial se trata del paquete *KINECT for Windows SDK Beta* [26][31], publicado en junio de 2011. Este paquete es gratis, pero de licencia limitada a fines de investigación y desarrollo. Microsoft promete una versión de licencia comercial para inicios de 2012 [19].

El *KINECT for Windows SDK Beta* es el software que se ha utilizado en parte del desarrollo de este proyecto de Robot Acompañante.



Figura 1. 13 Imagen promocional de KINECT para Windows 7, tomado de [24]

1.4.1.2 Descripción del Hardware de KINECT

1.4.1.2.1 Funcionamiento del hardware de KINECT.

Como elementos principales, el sensor KINECT cuenta con una cámara a color, una segunda cámara monocromática de luz infrarroja, un proyector láser infrarrojo y el circuito integrado PS1080 de tipo System on Chip. KINECT entrega, imágenes con información por pixel tanto de color como de profundidad (distancia) [25].



Figura 1. 14 Distribución y relación de los elementos del KINECT, tomado de [25]

El principio que utiliza el sensor, para medir la profundidad es el siguiente:

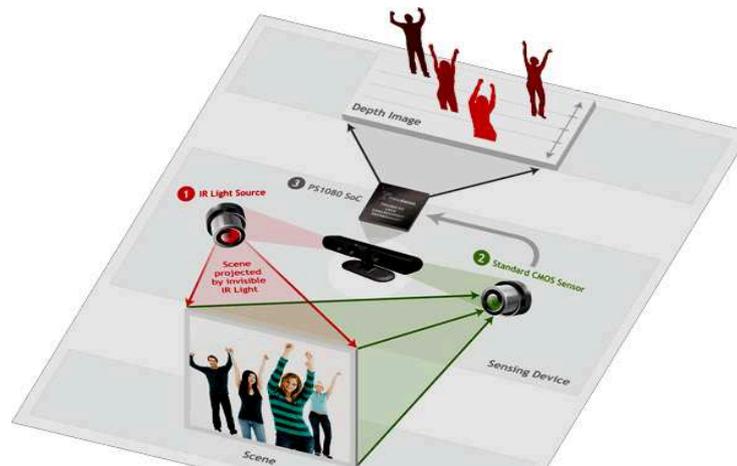


Figura 1. 15 Diagrama de la tecnología patentada por PrimeSense, tomado de [26]

El proyector láser infrarrojo proyecta sobre la escena un patrón conocido de 50000 puntos invisibles al ojo humano, que es el que se observa en la Figura 1.16. Luego de rebotar en los objetos de la escena, el patrón de puntos es captado por la cámara infrarroja que se encuentra a 7,5 cm de separación del proyector. Entonces, el circuito integrado analiza la disparidad, provocada por los objetos de la escena, entre el patrón de puntos proyectados y el patrón de puntos captados. Dicho efecto de *disparidad* es dependiente del ángulo de rebote de los haces de luz infrarroja. Este ángulo depende a su vez de la distancia en que se encuentren los objetos. Es decir que en definitiva, el sensor realiza triangulaciones en base a las disparidades en la imagen del patrón infrarrojo [25].

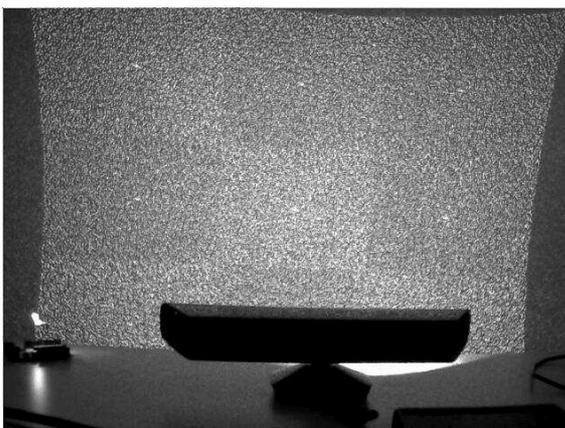


Figura 1. 16 Fotografías con cámaras de visión nocturna de la proyección del patrón infrarrojo, que es invisible a la visión humana, tomado de [27]

1.4.1.2.2 Características técnicas de KINECT.

Su campo de visión tiene una apertura horizontal de 57° y vertical de 43° . Cuenta con un motor en la base con el que se puede ajustar la inclinación vertical del sensor en $+27^\circ$ o -27° y tiene un acelerómetro para estimar su inclinación real. El rango de profundidad absoluto que puede percibir es de 0,8 m a 4 m, siendo el rango recomendado, para posicionar a las personas, de 1,2 m a 3,5 m [26][28]. La profundidad calculada por el chip PS1080 no es de tipo radial, como se podría pensar, sino de tipo cartesiana y ortogonal a la cara del sensor [29].

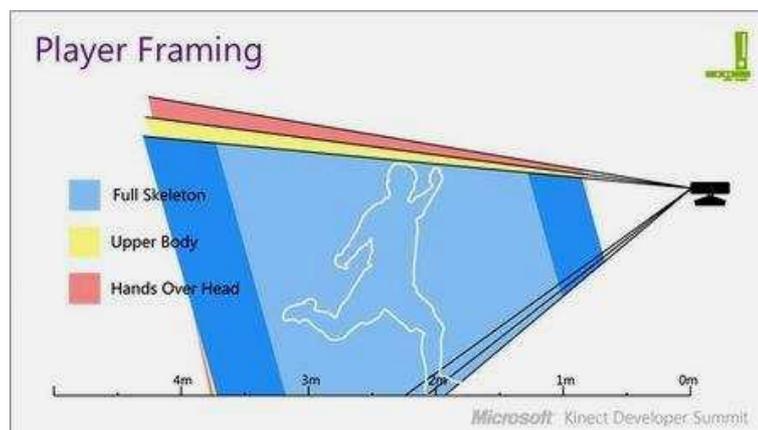


Figura 1. 17 Rango de percepción de profundidad de KINECT, tomado de [28]

Los datos que entrega el sensor a través del enlace USB 2.0 son de tipo flujo de imagen (image stream), a una velocidad de 30 veces cada segundo (30Hz). Un flujo de imagen es RGB de 8bits y resolución de 640 x 480 pixeles; y el otro flujo de imagen es de profundidad de 11bits con resolución de 320 x 240 pixeles. Adicionalmente entrega una señal de audio de 16 bits proveniente de un arreglo de cuatro micrófonos que están distribuidos en el cuerpo del sensor [29].



Figura 1. 18 Tipos de datos entregados por el sensor, tomado de [20]

KINECT utiliza dos voltajes de alimentación: 5V a través de la conexión USB a la PC, pero también 12V desde otro cable, que por medio de un adaptador se conecta a una fuente rectificadora de 110V alternos.

1.4.1.3 Descripción del Software de KINECT

1.4.1.3.1 Funcionamiento del software de KINECT

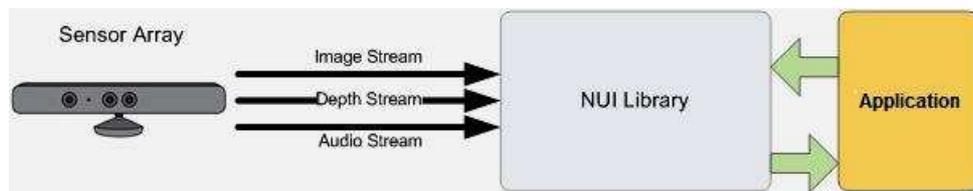


Figura 1. 19 Relación entre el Hardware y Software de KINECT, tomado de [29]

Como ejemplifica la Figura 1.19, una parte importante de la tecnología KINECT no se encuentra a bordo del cuerpo del sensor, sino que se trata de software de *Interfaz de Usuario Natural* (NUI) que corre en la PC, para aprovechar su capacidad computacional y para acoplarse directamente a la aplicación del usuario [29].

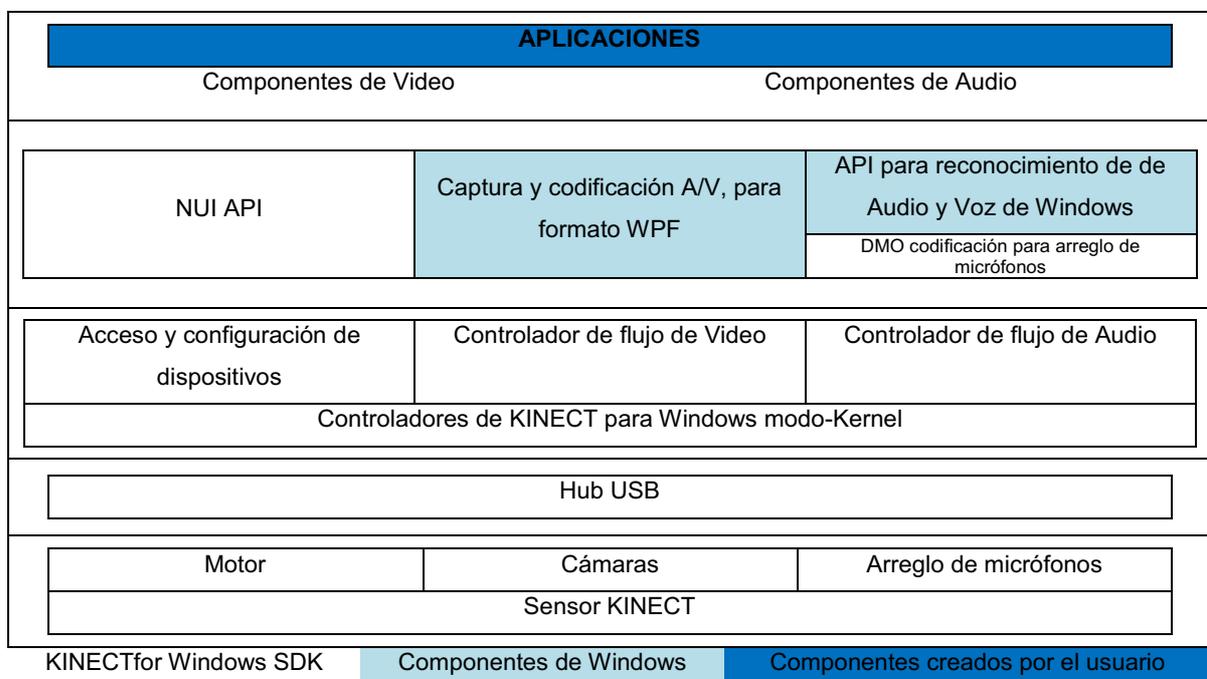


Figura 1. 20 Diagrama detallado de los elementos de hardware y software en la utilización de KINECT para la PC, tomado de [29]

Como se puede observar, la capa 2 de la Figura 1.20 son los drivers que permiten a Windows comunicarse con KINECT y recibir los flujos de imagen de color, profundidad y audio. Por encima de los drivers están las *Interfaces de Programación de Aplicación* (API).

Los drivers y la API de *Interfaz de Usuario Natural* (NUI), que es la que realiza el procesamiento que reconoce las poses de la figura humana, se obtienen al instalar el paquete de software KINECT for Windows SDK.

El paquete KINECT for Windows SDK es proporcionado por Microsoft y se lo descarga desde la dirección <http://kinectforwindows.org/download/>. Para su instalación, el requerimiento mínimo es que el computador tenga el sistema operativo Windows 7. Para su funcionamiento, también se requiere que esté instalado el Framework .NET 4 de Microsoft, el cual se puede obtener automáticamente al instalar el entorno de programación Visual Studio 2010. Como requerimiento de hardware, se recomienda que el computador tenga un procesador de al menos doble núcleo con velocidad de 2.66 GHz, memoria RAM de 2 GB y tarjeta gráfica compatible con DirectX 9 [29].

La forma en que trabaja el algoritmo en el API NUI del software KINECT for Windows SDK (o previamente de la Xbox360) para hacer el reconocimiento de la forma humana, radica en la forma en que fue concebido.

Primeramente, el algoritmo se basa únicamente en las imágenes de profundidad y no en las de color. La razón para ello es que en la imagen de profundidad, las siluetas de los objetos, incluyendo las partes del cuerpo humano, quedan naturalmente separadas del fondo de la imagen y de otros objetos, como se representa en la Figura 1.21, lo que no es posible con una imagen plana de color. Además, en la imagen de profundidad no influye la iluminación de la escena ni la textura de los objetos. Estas condiciones facilitan enormemente el trabajo del algoritmo de la API NUI [30].

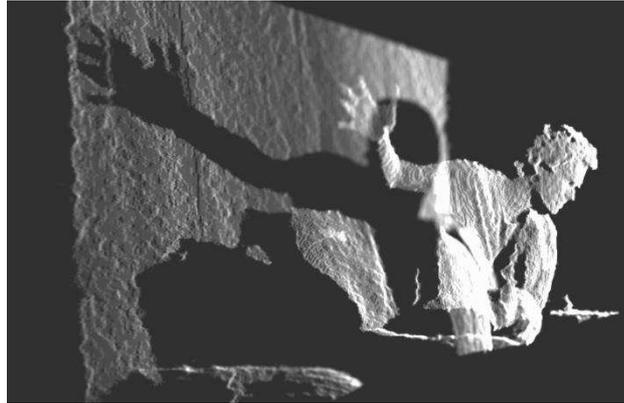


Figura 1. 21 Imagen de profundidad procesada para representarla en una perspectiva lateral, lo cual enfatiza su carácter tridimensional, tomado de [31]

El método desarrollado por el grupo *Microsoft Research Cambridge & Xbox Incubation* en la Universidad de Cambridge, tiene por premisa el encontrar la figura humana y sus partes a partir de cada cuadro de imagen por sí solo; es decir, sin necesidad de tener en cuenta imágenes anteriores, ni información de dónde estuvo o va estar la persona [30]. Al descartar la información temporal, el método no necesita de procedimientos de inicialización, re-inicialización o memoria, ni predicción.

El método tampoco trata de encontrar la silueta completa del humano y luego ir identificando sus partes de forma jerárquica. El algoritmo busca las partes humanas directamente. Es más bien simplificado, iterativo y trabaja a nivel de píxeles para clasificarlos según 31 segmentos del cuerpo humano, que son las áreas coloreadas en la Figura 1.22 [30].



Figura 1. 22 Segmentos en los que se clasifican los píxeles de la imagen de profundidad, tomado de [30]

A breves rasgos, el algoritmo de clasificación trabaja así: Se toma un píxel X de la imagen y una función compara la diferencia de profundidad relativa del píxel X a

otros dos pixeles cercanos, dependiendo del resultado de esta comparación de profundidades, se escogen otros dos pixeles para comparar profundidades nuevamente, y luego se escoge otros dos más y así algunas veces sucesivas. Al final, según la ruta de comparaciones que recorrió el pixel X, se le asigna un valor probabilístico a ese pixel X sobre a cual segmento del cuerpo pertenece. [30]

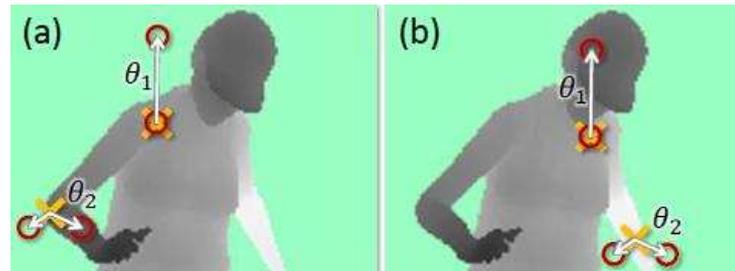


Figura 1. 23 Las cruces son pixeles a clasificarse y los círculos pixeles de comparación. Una misma manera de comparar, dará distintas respuestas de diferencia de profundidad dependiendo de la ubicación del pixel X, tomado de [30]

Las rutas de comparaciones sucesivas para el pixel X constituyen lo que se denomina *un árbol de decisión*. Para robustecer la clasificación del pixel, el algoritmo utiliza más de un árbol de decisión. Así se habla entonces *de bosques de decisión*.

Un bosque es un conjunto de árboles de decisión. Cada árbol tiene nodos divisorios y nodos finales. Se toma el pixel X y se introduce al primer nodo, cada nodo divisorio tiene la función que compara profundidades respecto a distintos pixeles cercanos al pixel X. De este resultado depende si se sigue por la rama derecha o izquierda a los siguientes nodos divisorios, en un esquema como el retratado en la Figura 1.24 [30].

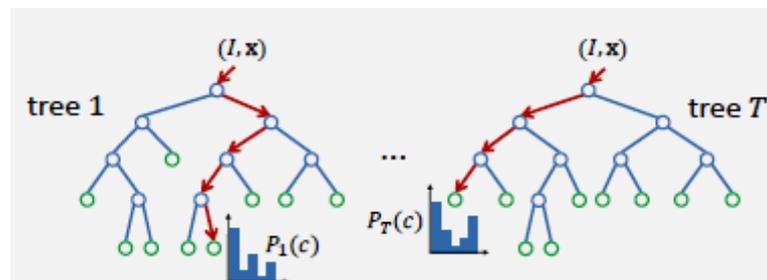


Figura 1. 24 Árboles de comparación, para asignar una probabilidad P de pertenecer a un segmento c de cuerpo, tomado de [30]

Un nodo final es el que asigna al pixel X un valor de probabilidad de pertenecer a cierto segmento del cuerpo. Estos valores de cada árbol se promedian para tener una probabilidad final asignada por el bosque [30].

Hacer este análisis por cada pixel, haciéndolos pasar consecutivamente por ramas de árboles de decisión parecería ser una gran carga computacional, pero en realidad es bastante eficiente pues el cálculo en cada nodo divisorio requiere apenas 5 operaciones matemáticas. Así una imagen de profundidad, con todos sus pixeles a clasificar, puede ser analizada en un tiempo de hasta 2ms en el procesador gráfico de la Xbox360 [30].

1.4.1.3.2 Desarrollo del software de KINECT por el grupo Microsoft Research Cambridge & Xbox Incubation

El éxito del proceso de reconocimiento de las poses humanas radica totalmente en la programación que se haya hecho de los árboles de decisión que operan en el API NUI, durante su desarrollo.

Para los programadores del API habría sido totalmente imposible diseñar los árboles de decisión, mediante programación tradicional, es decir analizando mentalmente lo que sucede con la imagen de profundidad al nivel de los pixeles al considerar todas las posibles poses de un humano con los posibles tipos de cuerpos, desde cualquier ángulo, con poses parciales, etc.

Es en este punto en donde entró el aprendizaje de máquina. Los investigadores de Cambridge no programaron los árboles de decisión en sí, sino que programaron un proceso de aprendizaje para que 3 árboles se formen solos mediante entrenamiento, introduciendo al proceso conjuntos de 1 millón de imágenes de profundidad, con personas de diversos tipos en diversas poses, divididos en aproximadamente 300000 imágenes por árbol [30].

Aún así, tener que grabar el millón de imágenes de profundidad requeridas para el aprendizaje, variando cada vez el tipo de personas y variables de la escena, es también demasiado arduo.

Por eso lo que hicieron en realidad fue grabar algunas escenas con actores realizando actividades usando trajes de captura de movimiento, como los utilizados en la industria del cine. Luego, de estas grabaciones extrajeron 100000 cuadros tridimensionales básicos, con los cuales sintetizaron por computadora el millón de imágenes de profundidad necesarias, modificando por programa las características físicas de los actores, los ángulos de captura, etc. Esto fue mucho más rápido que si hubiesen llamado cientos de distintos actores o cambiado físicamente las condiciones de grabación [30].

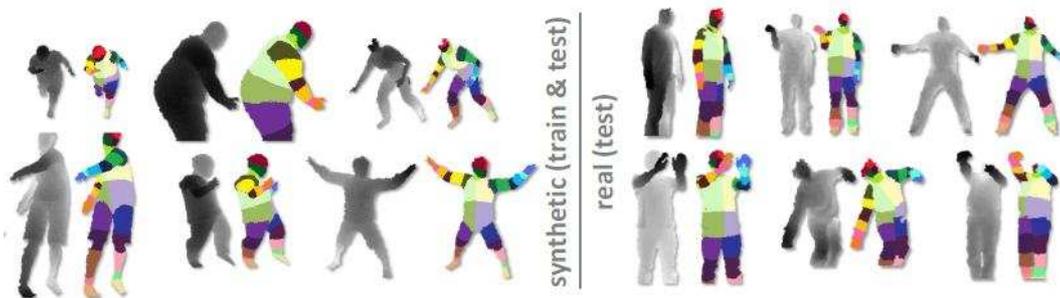


Figura 1. 25 Imágenes de profundidad y su correspondiente reconocimiento de segmentos, tanto con las imágenes sintéticas del entrenamiento y con imágenes reales de KINECT, tomado de [30]

El entrenamiento de estos 3 árboles con un número límite de 20 ramas, a partir de un millón de imágenes de profundidad, corriendo en un cluster de 1000 núcleos de procesadores, tuvo una duración de 1 día completo [30].

1.4.2 SENSOR DE GASES

Se trata de sensores contruidos teniendo como material sensible oxido estánico (SnO_2) además un micro-tubo cerámico de oxido de aluminio (Al_2O_3) como muestra la Figura 1.26. Se usa este material sensible debido a la baja conductividad que presenta en presencia de aire limpio. Cuando existe el gas nocivo en el ambiente, la conductividad del material sensible tiende a elevarse proporcionalmente a la concentración de gas en el ambiente. El sensor contiene además de los elementos sensibles un calefactor que mantiene las condiciones de trabajo necesarias [32].

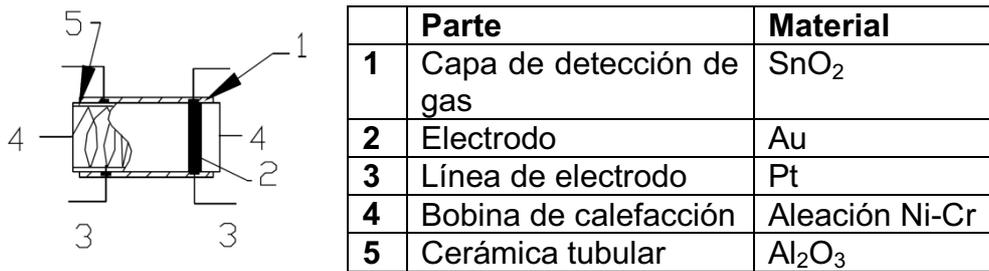


Figura 1. 26 Partes del sensor, tomado de [32].

El circuito estándar (Figura 1.27) de medición consta de dos partes, la primera es el sistema de calefacción que cuenta con un control de tiempos (la variación entre alto y bajo voltaje es cíclica), el voltaje de calefacción V_H y el voltaje de polarización V_C son del mismo valor. La segunda parte es el circuito de señal de salida, esta señal varía de acuerdo a la variación de la resistencia superficial del sensor (R_s) [32].

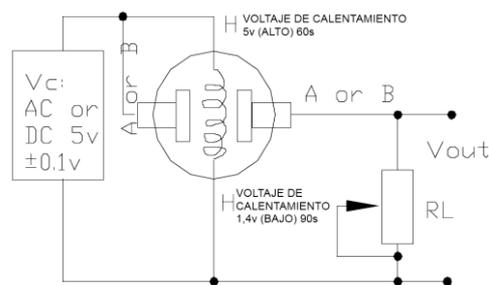


Figura 1. 27 Circuito esquemático del sensor, tomado de [32]

R_s se puede calcular, mediante la siguiente ecuación:

$$R_s \setminus R_L = (V_C - V_{RL}) / V_{RL}$$

Como muestra la expresión anterior, la sensibilidad del sensor se la ajusta variando el valor de la resistencia de carga R_L . La Figura 1.28 muestra el circuito de funcionamiento, donde R_L es una resistencia variable que ayuda a calibrar el rango de señal de salida, de acuerdo a la necesidad [32].

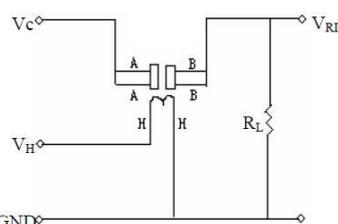


Figura 1. 28 Diagrama de conexiones, tomado de [32]

1.4.3 SENSOR ULTRASÓNICO

El principio de funcionamiento de este tipo de sensores son las ondas ultrasónicas. Este tipo de ondas se caracterizan por estar fuera del rango de audición humana (espectro audible) que está entre 20 Hz y 20000 Hz, toda frecuencia de audio que sobrepase este valor es denominada ultrasónica.

El prototipo diseñado utiliza sensores ultrasónicos MAXSONAR EZ-1, para la prevención de colisiones cuando el prototipo se encuentra en modo de seguimiento.

El sensor ultrasónico mantiene equilibrio entre ancho de haz y sensibilidad de detección. La Figura 1.29 muestra una cuadrícula de 30 cm que contiene información sobre las características de detección:

- (A) Pasador de 6,35 mm de diámetro, notar la forma del haz a corta distancia para objetos pequeños.
- (B) Pasador de 2,54 cm de diámetro, notar lo largo y estrecho del patrón de detección.
- (C) Varilla 8,25 cm de diámetro, notar lo largo y controlado del patrón de detección.
- (D) Tablero de 27,9 cm de ancho, esta prueba es realizada moviendo paralelamente de izquierda a derecha el tablero frente al sensor. Esto muestra la capacidad en rango que posee el sensor.

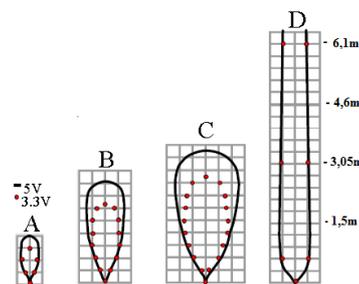


Figura 1. 29 Características del haz del sensor MAXSONAR-EZ1, tomado de [33]

Este sensor es de fácil utilización. Cuenta con dos tipos de salida una analógica y otra en formato serial de 8 bits. Para el prototipo se utiliza la salida analógica del sensor, esta salida tiene un rango entre 0[V] y 5[V] lo que permite que se comunique con el microcontrolador sin ninguna complicación.

A continuación, en la Tabla 1.1, se muestran las características de funcionamiento del sensor:

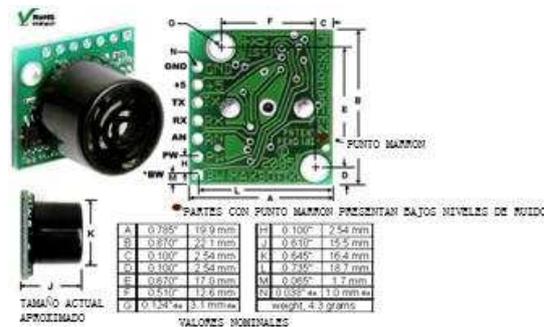


Figura 1. 30 Sensor MAXSONAR-EZ1, tomado de [33]

Tabla 1. 1 Características de funcionamiento del MAXSONAR-EZ1, tomado de [33]

PIN	Descripción
GND	Retorno de la fuente de poder. Para GND (y Vcc) en lo posible usar cable blindado.
+5V	Vcc puede ser entre 2,5 V-5,5 V. Se recomienda una corriente de 3 mA para 5 V y 2 mA para 3 V.
TX	Cuando BW está abierto o en 0L, la salida TX mantiene una salida serial asincrónica con formato RS232, con niveles entre 0-Vcc. El formato de salida es un carácter ASCII mayúsculo "R", seguido por tres caracteres que representan el rango en pulgadas hasta un valor máximo de 255, seguido por un carry (ASCII 13). La velocidad de transmisión es de 9600bps, 8 bits, sin paridad y un bit de parada. Si fuese necesario la salida del sensor puede ser acondicionada a través de un convertidor MAX232 que sirve para elevar los voltajes a los valores estándar del protocolo RS232.
RX	Este pin se encuentra internamente en alto. El sensor EZ1 medirá continuamente si este PIN no se conecta o se mantiene en alto. Si se mantiene este pin en bajo se detendrá la medición. Hay que mantener en alto el pin por lo menos 20 μ s para lograr una buena lectura.
AN	El valor de la salida analógica tiene un factor de escalamiento de (Vcc/512) por pulgada. La salida es almacenada y corresponde al rango de datos más reciente. Una fuente de 5V muestra un rendimiento de ~9.8 mV/in y una de 3.3 V un rendimiento de ~6.4mV/in.
PW	Este pin tiene una salida donde el ancho de pulso representa la medición. Para calcular la distancia se debe usar un factor de escalamiento de 147 μ S por pulgada.
BW	Mantener libre o en 0L para permitir la transmisión de datos a través de TX.

CAPÍTULO 2

DISEÑO Y CONSTRUCCIÓN DEL PROTOTIPO

2.1 INTRODUCCIÓN

El prototipo de robot acompañante cuenta con dos modos de funcionamiento: modo remoto que permite llevar el vehículo hacia donde el usuario lo requiera y modo de seguimiento en el cual el vehículo seguirá al usuario de manera autónoma.

Los dos modos son operados haciendo uso de un teléfono celular que cuente con el sistema operativo ANDROID, que se comunica con el prototipo de manera inalámbrica mediante tecnología Bluetooth.

El modo de seguimiento es posible gracias al sensor de movimiento KINECT, desde el cual se adquieren datos hacia un computador portátil. Este procesa la información y envía los datos de ubicación del usuario mediante comunicación serial, hacia el microcontrolador Atmega164p encargado del control de motores y manejo de periféricos; en la Figura 2.1 se muestra la arquitectura del prototipo.

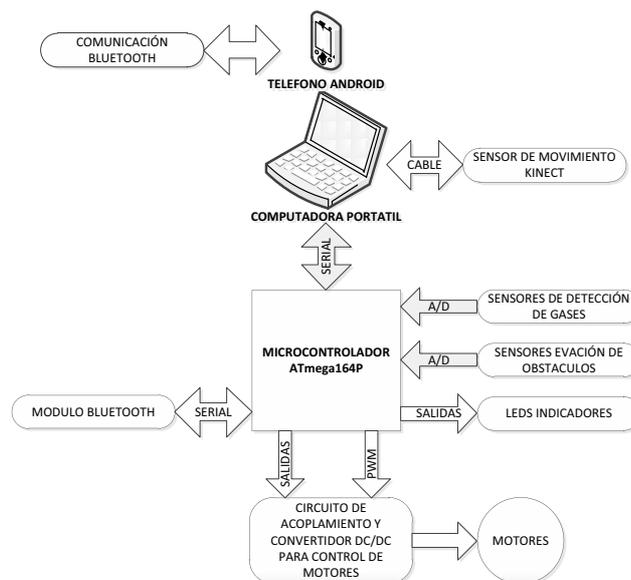


Figura 2. 1 Arquitectura del prototipo

Para la locomoción se utiliza dos motores DC en configuración diferencial, manejados haciendo uso de un conversor DC/DC que permite que el microcontrolador varíe la velocidad y sentido de giro de acuerdo a las necesidades del sistema.

Para brindar ventajas al usuario, el prototipo cuenta con un sensor para la detección de gases nocivos inflamables (GLP, CO y CH₄), con señal de salida analógica, esta información se envía mediante Bluetooth para ser mostrada en la interfaz programada en el celular.

2.2 DISEÑO MECÁNICO

El prototipo cuenta con tres partes principales:

- Cuerpo principal, donde están las tarjetas de control, baterías y motores (Figura 2.2).
- Gabinete para computadora (Figura 2.3).
- Gabinete para carga (Figura 2.4).



Figura 2. 2 Cuerpo principal



Figura 2. 3 Gabinete para computadora



Figura 2. 4 Gabinete para carga

Estas partes han sido construidas con tableros de fibra de media densidad (MDF, por sus siglas en ingles). Se escogió este material por su resistencia mecánica, costo, versatilidad y bajo peso; este último factor es de gran utilidad ya que el vehículo al tener como principal objetivo el ayudar a cargar objetos, no puede desperdiciar su capacidad de carga; además se puede mencionar que si bien el material es sensible a la humedad, esto no se muestra como un problema ya que el prototipo está diseñado para funcionar en ambientes cubiertos como casas y edificios.

Las llantas del vehículo son neumáticos inflables (Figura 2.5), esto por el bajo peso y gran agarre, reduciendo así los deslizamientos y haciéndolo útil para superficies con bajo coeficiente de fricción como baldosas o pisos encerados.



Figura 2. 5 Llantas del prototipo

Se usa motores de corriente continua con caja reductora, que presentan las siguientes características (Tabla 2.1):

Tabla 2. 1 Características de motor DC

Característica	Nominal	Unidades
Torque	3	N.m
Velocidad	90	rpm
Voltaje de funcionamiento	12	V
Corriente	3	A

Las características mencionadas permiten un funcionamiento óptimo, permitiendo que el prototipo tenga las siguientes características (Tabla 2.2):

Tabla 2. 2 Características mecánicas del prototipo

Característica	Nominal	Máxima	Unidades
Velocidad	0,9	1,45	m/s
Peso neto	6	-	Kg
Capacidad de carga.	3	5	Kg

2.2.1 ENSAMBLAJE DEL PROTOTIPO

Para ensamblar el prototipo se debe seguir una secuencia de pasos que se muestran a continuación:

1. *Montaje de motores:* Los motores van sujetos a las paredes laterales del cuerpo principal (Figura 2.6), sujetos con tornillos. Previa a la colocación se acopla platinas a los motores, las que sirven para fijación.

**Figura 2. 6** Montaje de motores

2. *Montaje de llantas:* Las llantas son acopladas a los motores haciendo uso del sistema de acople (Figura 2.7) este sistema está construido de poliuretano para evitar el desgaste del sistema de arrastre del motor.

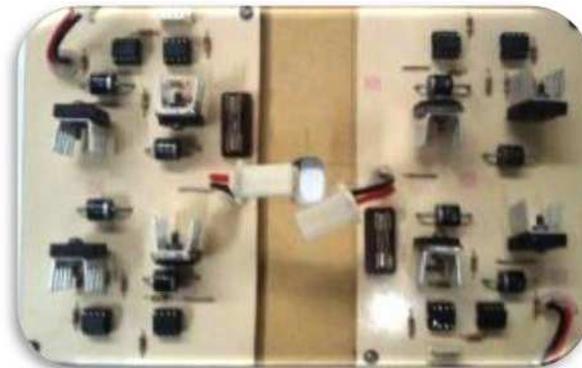


Figura 2. 7 Sistema de acople y montaje de llantas

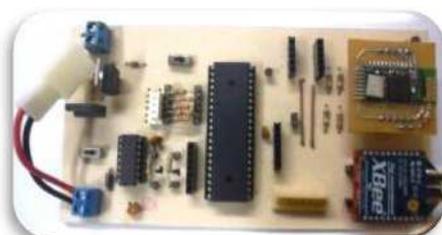
3. *Montaje de elementos electrónicos y eléctricos:* Se lo hace siguiendo la siguiente secuencia:
- Ubicación de baterías (Figura 2.8a).
 - Ubicación de convertidores DC/DC, para control de velocidad (Figura 2.8b).
 - Ubicación de placa de control (Figura 2.8c).



(a)



(b)



(c)

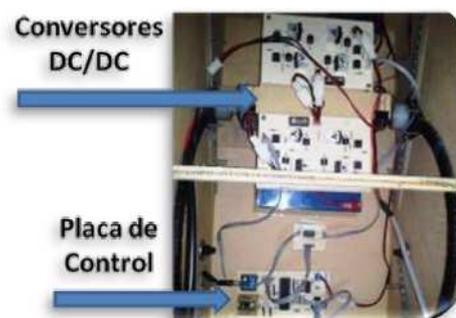




Figura 2. 8 Montaje de elementos eléctricos y electrónicos

La Figura 2.8d muestra el montaje de las placas de potencia y control; estas son fijadas a la carrocería del vehículo mediante tornillos y vinchas de sujeción para evitar daños debido a las vibraciones propias del prototipo.

d. Ubicación de sensores (Figura 2.8e).

4. *Colocación de gabinetes:* Primero se coloca el gabinete para la computadora portátil y se lo asegura, después se coloca el gabinete de carga (Figura 2.9).

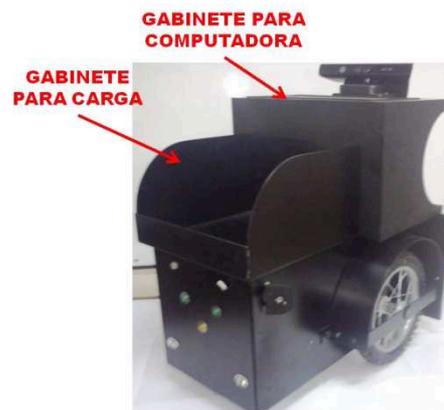


Figura 2. 9 Colocación de gabinetes

5. *Conexiones eléctricas:* Todos los cables se encuentran adecuadamente señalizados, de tal manera que se pueda realizar las conexiones sin ningún tipo de problema.

6. *Colocación de equipos:* Una vez ubicados los gabinetes se coloca el sensor de movimiento KINECT y la computadora como muestra la Figura 2.10.

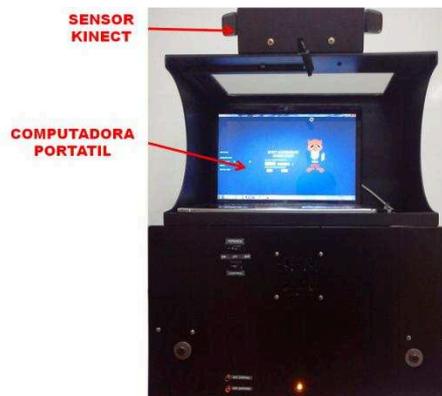


Figura 2. 10 Colocación de equipos

Quedando montado el prototipo, como muestra la Figura 2.11.



Figura 2. 11 Modelo completo

Izquierda: Diseño SolidWorks / Derecha: Prototipo completo

2.3 DISEÑO ELECTRÓNICO Y ELÉCTRICO

Esta sección trata sobre el diseño en detalle de los circuitos de control y de potencia necesarios para el funcionamiento del prototipo; el dimensionamiento de fuentes de alimentación para los diferentes circuitos; y el uso de recursos del microcontrolador encargado del control de funcionamiento del prototipo.

2.3.1 ALIMENTACIÓN PARA MOTORES

Los motores que se van a utilizar son motores de corriente continua con las características mostradas anteriormente (Tabla 2.3); por tal razón se escogió una batería seca de las siguientes características:

Tabla 2. 3 Características de batería, tomado de [34]

CARACTERÍSTICA	VALOR	UNIDAD
Voltaje nominal	12	V
Capacidad nominal	12	12 AH (20h@600mA hasta 10,5V)
Máxima corriente de descarga (7 min)	36	A
Peso aproximado	3,7	Kg

Esta batería da un tiempo de funcionamiento continuo (tfc), a condiciones nominales del motor:

$$tfc_{por\ motor} = \text{Capacidad\ bateria} / I_{n_{motor}}$$

$$tfc_{motor} = 12A \cdot \frac{h}{3A} = 4h$$

Como se tiene dos motores funcionando en condiciones similares:

$$tfc_{total} = \frac{4h}{2} = 2h$$

Este valor podría verse alterado debido a las variaciones de corriente media, que se presentan por el cambio de condiciones de trabajo a las que podría estar sometido el prototipo. Estas variaciones provocan una disminución de tiempo de funcionamiento continuo de hasta 30%, con base a mediciones prácticas realizadas, quedando:

$$tfc_{total} = 2h \pm 30\%$$

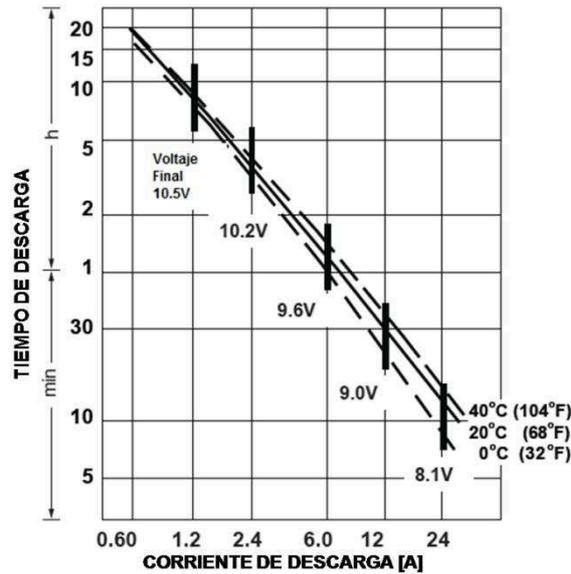


Figura 2. 12 Curva, tiempo de descarga vs. Corriente de descarga, tomado de [34]

2.3.1.1 Diseño de conversor DC/DC

El conversor que se va a utilizar debe ser capaz de soportar los cambios de voltaje y corriente, provocados por la variación de velocidad y cambio de sentido de giro de los motores; por esta razón el dimensionamiento de los transistores que lo conforman, se basara en los parámetros de funcionamiento de los motores.

Como resultado de las pruebas realizadas a los motores, se ha determinado que la relación de trabajo mínima para lograr romper la inercia del vehículo es de 0,35 ($D=0,35$), de esto se tiene que el voltaje medio mínimo que alcanzará el conversor es:

$$V_{out_{min}} = D_{min} \cdot V_{in} = 0,35 \cdot 12 = 4,21V$$

La corriente nominal (I_n) del motor es 3[A], y la corriente máxima del motor es 10[A], de aquí que el transistor deberá soportar una corriente máxima de:

$$I_{Tmax}=10A$$

La frecuencia de la señal PWM de activación de la parte inferior del puente es:

$$f_{sw}=120Hz$$

Por lo mencionado anteriormente se ha seleccionado los siguientes transistores:

- Para la parte superior del puente: TIP147

Tabla 2. 4 Características del transistor TIP147, tomado de [37]

SÍMBOLO	CARACTERÍSTICA	VALOR	UNIDAD
$V_{EBO(MAX)}$	Voltaje base-emisor máximo.	-5	V
I_C	Corriente de colector máxima.	-10	A
I_B	Corriente de base máxima.	-0,5	A
$V_{BE(SAT)}$	Voltaje base-emisor de saturación.	-3,5	V
t_d	Delay time	0,15	μS
t_r	Rise time	0,55	μS
t_{STG}	Storage time	2,5	μS
t_f	Fall time	2,5	μS

- Para la parte inferior del puente: TIP142

Tabla 2. 5 Características del transistor TIP142, tomado de [38]

SÍMBOLO	CARACTERÍSTICA	VALOR	UNIDAD
$V_{EBO(MAX)}$	Voltaje base-emisor máximo.	5	V
I_C	Corriente de colector máxima.	10	A
I_B	Corriente de base máxima.	0,5	A
$V_{BE(SAT)}$	Voltaje base-emisor de saturación.	3,5	V
t_d	Delay time	0,15	μS
t_r	Rise time	0,55	μS
t_{STG}	Storage time	2,5	μS
t_f	Fall time	2,5	μS

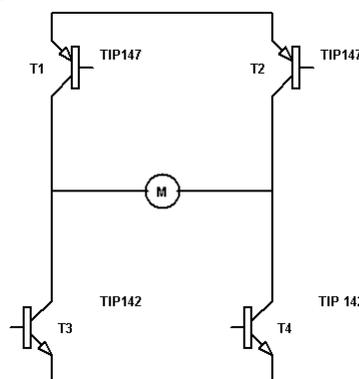


Figura 2. 13 Conversor DC/DC.

2.3.2 ALIMENTACIÓN PARA PLACAS DE CONTROL

Los niveles de voltaje necesarios para el funcionamiento de la placa de control son dos:

- 5 voltios, para el microcontrolador, LEDs indicadores, circuito de acoplamiento y sensores.
- 3,1 voltios, para los módulos de comunicación Xbee y Bluetooth, según las recomendaciones del fabricante.

Para obtener estos dos niveles de voltaje se cuenta con una batería seca recargable de las siguientes características:

Tabla 2. 6 Características de batería, tomado de [35]

CARACTERÍSTICA	VALOR	UNIDAD
Voltaje nominal	12	V
Capacidad nominal	1,4	1,4 AH (20h@70mA hasta 10,5V)
Máxima corriente de descarga (7 min)	4,2	A
Peso aproximado	0,54	Kg

Esta batería sirve además para alimentar el sensor de movimiento KINECT, que requiere 12V@1A para su correcto funcionamiento.

2.3.2.1 Fuente de 5 voltios

El diseño de la fuente de 5 voltios (Figura 2.14) se basa en el uso de un regulador de voltaje LM7805, el cual entrega 5[V]@1[A] máximo, sus características de fabricación indican que para regular sin ningún inconveniente, el voltaje de entrada debe ser mayor a 7[V] y menor a 20[V] [39]. Como se cuenta con una batería de 12V@1,4Ah no se va a tener inconvenientes.

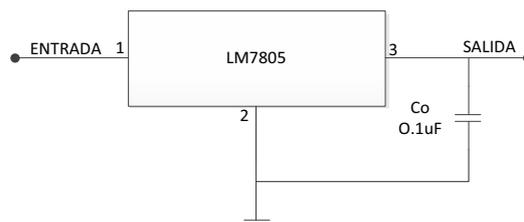


Figura 2. 14 Esquema de fuente de 5V

2.3.2.2 Fuente de 3,1 voltios

Los módulos que van a hacer uso de esta fuente trabajan en un rango de 3,0[V] hasta 3,3[V] como voltaje de alimentación. Para el diseño se va a trabajar con 3,1[V] con el fin de garantizar niveles óptimos de funcionamiento.

Para la obtención del nivel requerido el diseño se basa en un regulador de voltaje LM317 que cuenta con una salida de tensión de entre 1,2[V] y 37[V] con una corriente de hasta 1,5[A]. La Figura 2.15 muestra la conexión del regulador recomendada por el fabricante:

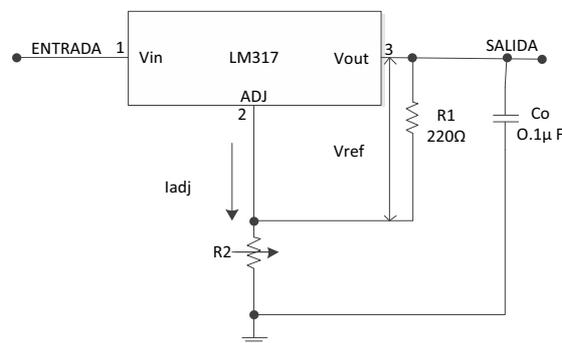


Figura 2. 15 Esquema de fuente de 3,1V

La resistencia $R1=220\Omega$ y $Co=0,1\mu F$ son valores recomendados por el fabricante, mientras que $R2$ es una resistencia variable que ajusta el valor del voltaje de salida. Al variar $R2$, se debe cuidar que I_{adj} no sobrepase los $100\mu A$ con el fin de que el error asociado con esta corriente sea despreciable. El voltaje de referencia (V_{ref}) tiene un valor fijo de 1,25[V].

De la ecuación:

$$V_{out} = V_{ref} \left(1 + \frac{R2}{R1} \right) + I_{adj} \cdot R2$$

Como $I_{adj} \cdot R2 \approx 0$

$$V_{out} = V_{ref} \left(1 + \frac{R2}{R1} \right)$$

$$R2 = \left(\frac{V_{out}}{V_{ref}} - 1 \right) \cdot R1$$

$$R2 = 325,6\Omega$$

$$R2 = 330\Omega$$

$$V_{out} = 3,13V$$

2.3.3 MICROCONTROLADOR ATmega164P

El prototipo diseñado utiliza un microcontrolador ATmega164P para realizar la adquisición de datos provenientes desde la computadora, procesar el algoritmo de control, la conversión análoga/digital de los sensores de detección de gases y prevención de colisiones, la comunicación mediante el módulo Bluetooth con el celular y el control de los motores.

El ATmega164P es un microcontrolador de 8-bits de bajo consumo con tecnología CMOS basado en arquitectura RISC AVR mejorada. Gracias a la ejecución de 3 instrucciones por ciclo de reloj, el ATmega164P logra rendimientos cercanos a 1MIPS por MHz, permitiendo así al diseñador optimizar el consumo de energía versus la velocidad de procesamiento [36].

Entre las características más importantes del microcontrolador se tiene:

- 2 timer/counter de 8-bits con modos de pre-escalamiento y comparación, separados.
- 1 timer/counter de 16-bits con modos de pre-escalamiento, comparación y captura.
- Contador en tiempo real con oscilador individual.
- 6 canales PWM.
- 8 canales ADC de 10-bits.
- 2 puertos USART programables.
- Oscilador interno RC calibrado.

- 32 líneas E/S programables.
- Voltaje de operación entre 2,7V-5,5V.
- Rango de velocidad entre 0-16MHz.
- Consumo de potencia a 8MHz, 5V, 25°C:
 - Modo activo: 8mA.
 - Modo ideal: 2,4mA.
 - Modo de bajo consumo: 8µA.

2.3.3.1 Análisis de recursos del microcontrolador utilizados

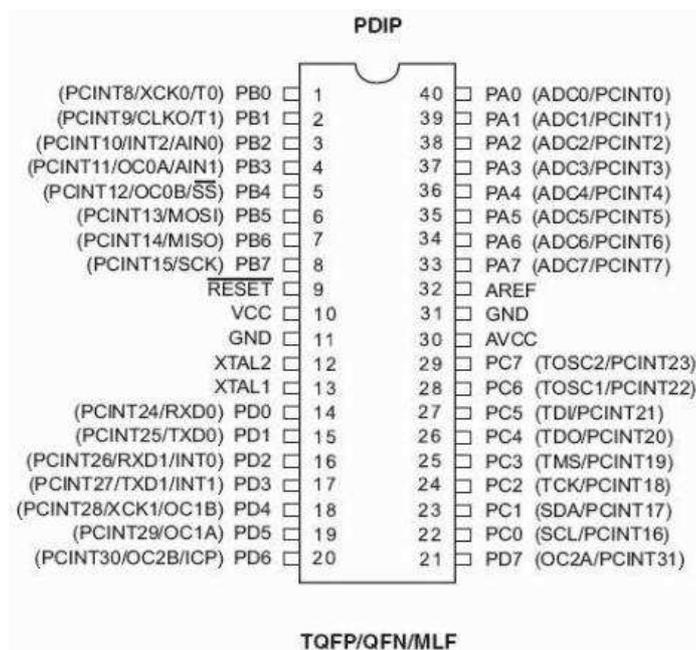


Figura 2. 16 Microcontrolador ATmega164P, tomado de [36]

Los periféricos manejados por el microcontrolador son los siguientes:

a. LEDs indicadores:

Los LEDs indicadores, cumplen con las funciones de:

- Informar al usuario el modo de funcionamiento en el que se encuentra el vehículo.
- Informar cuando el vehículo está listo para iniciar seguimiento.
- Informar cuando el sistema de control está encendido.

Debido a que se usa LEDs que consumen una elevada corriente, se utiliza transistores funcionando en corte/saturación para el encendido y apagado de estos (Figura 2.17).

Considerando como la corriente de disparo para el transistor $I_{base}=5mA$

$$R1 = \frac{V_{disparo}}{I_{base}} = \frac{5V}{5mA} = 1K\Omega$$

$$R1 = \frac{V_{cc} - V_{diodo}}{I_{led}} = \frac{5V - 2V}{20mA} = 150\Omega$$

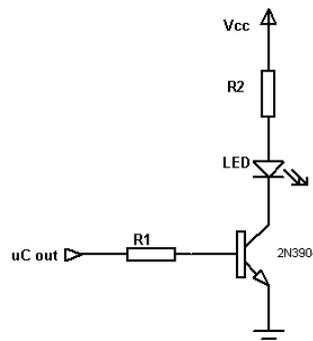


Figura 2. 17 Circuito de disparo de LEDs indicadores

A continuación se muestra la asignación de pines del microcontrolador para LEDs indicadores:

Tabla 2. 7 Asignación de pines para LEDs indicadores

Nombre	PIN	Tipo	Función [1L]
PC7	29	Salida	Indicador de control encendido.
PC6	28	Salida	Indicador de modo seguimiento listo
PC5	27	Salida	Indicador de seguimiento
PC4	26	Salida	Indicador de modo seguimiento en espera
PC3	25	Salida	Indicador de modo remoto.

b. Entradas A/D:

Los sensores para prevención de colisiones y para detección de gases contaminantes, se comunican con el microcontrolador enviando señales analógicas con un rango de entre 0[V] y 5[V].

El módulo de conversión A/D interno del microcontrolador usa como referencia el voltaje de alimentación ($V_{cc}=5V$) y realiza una conversión a $n = 10$ bits, teniendo una resolución de:

$$Resolución = \frac{V_{ref}}{2^n} = \frac{5V}{2^{10}} = \frac{5V}{1024} = 0,005[V]$$

Tabla 2. 8 Asignación de pines para entradas A/D

Nombre	PIN	Tipo	Función
PA1	39	ADC1	Sensor ultrasónico 1.
PA2	38	ADC2	Sensor ultrasónico 2.
PA3	37	ADC3	Sensor de detección de GLP.
PA4	36	ADC4	Sensor de detección de CO.

c. Puerto serial:

El microcontrolador mantiene comunicación por vía serial, con dos periféricos:

- Módulo Bluetooth, a una velocidad de 9600 bps, 8 bits, 1 bit de parada y sin paridad.
- Computadora portátil a través del circuito integrado MAX232, a una velocidad de 9600 bps, 8 bits, 1 bit de parada y sin paridad.

Debido a la diferencia de voltaje de polarización entre el microcontrolador y el módulo Bluetooth de comunicación inalámbrica, se utiliza un divisor de voltaje para el envío de datos (Figura 2.18) desde el microcontrolador hacia el módulo.

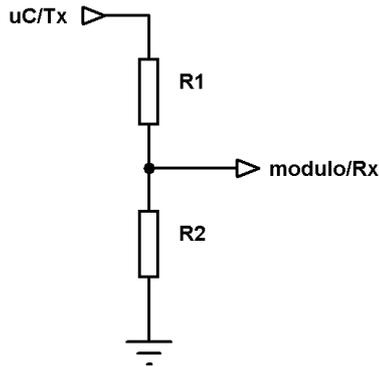


Figura 2. 18 Divisor de voltaje para envío de datos.

$$R2 = \frac{V_{out}}{V_{in}} \cdot R1$$

$$1 - \frac{V_{out}}{V_{in}}$$

El mayor voltaje a la salida del microcontrolador es $V_{in}=5[V]$ y el voltaje deseado $V_{out}=3,3$, si se asume $R1=1K\Omega$, se tiene:

$$R2 = \frac{3,3V}{5V} \cdot 1K\Omega$$

$$1 - \frac{3,3V}{5V}$$

$$R2 = 1,941K\Omega \approx 2K\Omega$$

Este divisor de voltaje no presenta problemas de comunicación con el microcontrolador, ya que, según datos de fábrica del microcontrolador un nivel lógico alto se reconoce desde 2,3[V].

A continuación se muestra la asignación de pines del microcontrolador para la comunicación serial:

Tabla 2. 9 Asignación de pines para comunicación serial

Nombre	PIN	Tipo	Función
PD0	14	Rx	Recepción de datos desde computadora.
PD1	15	Tx	Transmisión de datos hacia computadora.
PD2	16	Rx1	Recepción de datos desde módulo Bluetooth.
PD3	17	Tx1	Transmisión de datos hacia modulo Bluetooth.

d. Control de motores.

El control de velocidad y sentido de giro se lo realiza controlando un puente de 4 transistores (convertor DC/DC, apartado 1.3.1.1), a través de un circuito de opto-acoplamiento.

Para controlar cada motor se utiliza 2 pines de salida y una salida PWM; las dos primeras salidas sirven para controlar el encendido/apagado de la parte superior del puente, y la salida PWM para controlar el encendido/apagado de la parte inferior del puente (Figura 2.19).

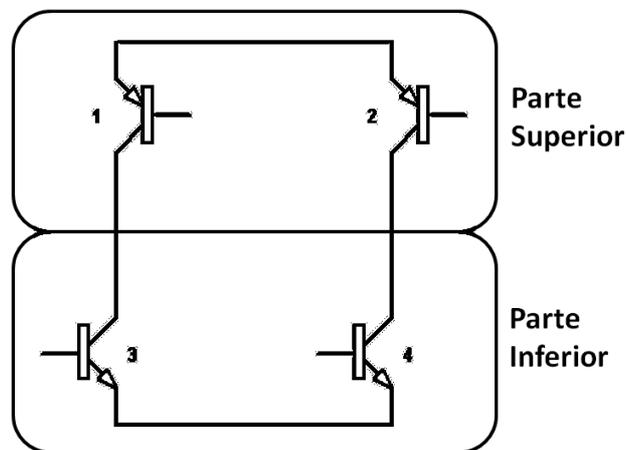


Figura 2. 19 Puente de transistores

Desde el microcontrolador se emiten las señales y antes de llegar al sistema de opto-acoplamiento pasa por un circuito lógico (Figura 2.20) que permite usar la misma señal PWM para encender/apagar la parte inferior del puente.

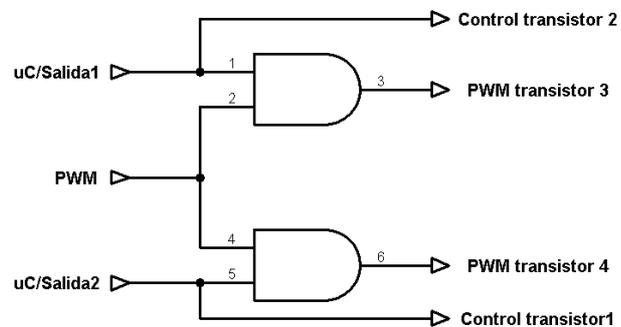


Figura 2. 20 Circuito lógico, asociado al puente de transistores

A continuación se muestra la tabla de funcionamiento del circuito lógico de acuerdo al funcionamiento del motor:

Tabla 2. 10 Funcionamiento del motor según combinaciones del circuito lógico

Salida 1	Salida 2	PWM	Motor
0	0	PWM	Parado
0	1	PWM	Horario
1	0	PWM	Anti horario
1	1	PWM	Prohibida

Las señales que se emiten desde el circuito lógico, llegan a los transistores a través de circuitos de opto-acoplamiento; para los transistores de la parte inferior del puente en configuración pull-down (Figura 2.21), y para la parte superior del puente en configuración pull-up (Figura 2.22).

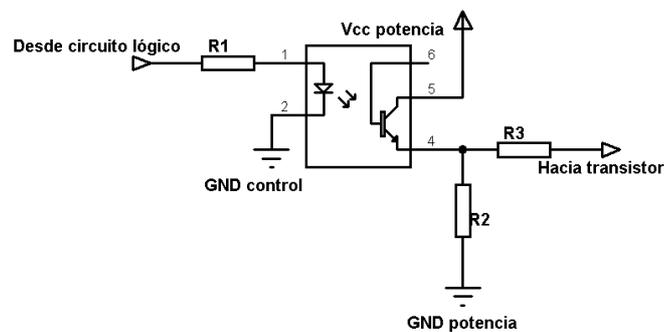


Figura 2. 21 Configuración pull-down

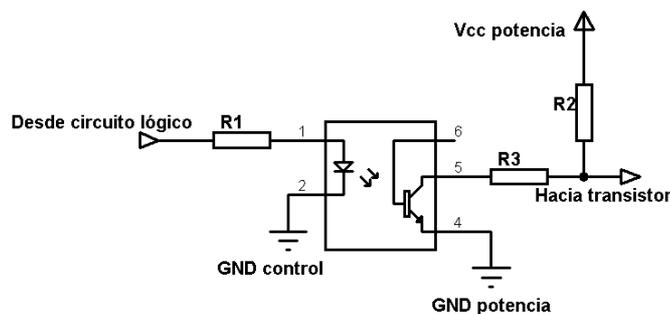


Figura 2. 22 Configuración pull-up

Tomando como base que la corriente de disparo nominal del optoacoplador es 16 mA, se tiene:

$$R1 = \frac{5V - 1,5V}{16mA} = 218,8\Omega$$

Tomando valores estándar de resistencia, se tiene:

$$R1=220\Omega$$

$$I_{\text{disparo}}=15,9\text{mA}$$

Se considera una corriente de salida máxima del optoacoplador de:

$$I_{\text{maxsalida}} = 50\text{mA}$$

Para la configuración pull-down:

Si se considera $R2= 10K\Omega$

$$I_{\text{salida1}} = \frac{12V}{10K\Omega} = 1,2\text{mA}$$

De acuerdo a las recomendaciones del fabricante, se tiene una corriente de base de saturación $I_B= 20 \text{ mA}$, para el TIP142:

$$R3 = \frac{V_{BC}}{I_B} = \frac{12V}{20\text{mA}} = 600\Omega$$

Tomando valores estándar de resistencia, se tiene:

$$R3=510\Omega$$

Para la configuración pull-up:

Si se considera $R2= 10K\Omega$

De acuerdo a las recomendaciones del fabricante, se debe garantizar un voltaje de colector-base $V_{BC} < 2 \text{ V}$ y una corriente de base de saturación $I_B= 10 \text{ mA}$, para el TIP147:

Se asume $V_{R3} = 0,6 \text{ V}$

$$R3 = \frac{R2 * V_{R3}}{V_{CC} - V_{R3}} = 526,3\Omega$$

Tomando valores estándar de resistencia, se tiene:

R3=510Ω

A continuación se muestra la asignación de pines del microcontrolador para el control de los motores:

Tabla 2. 11 Asignación de pines para control de motores

Nombre	PIN	Tipo	Función [1L]
PB1	2	Salida	Salida 1, motor 1
PB2	3	Salida	Salida 2, motor 1
PD5 (OC1A)	19	PWM	Señal PWM para motor 1
PB3	4	Salida	Salida 1, motor 2
PB4	5	Salida	Salida 2, motor 2
PD6 (OC2B)	20	PWM	Señal PWM para motor 2

CAPÍTULO 3

DESARROLLO DEL PROGRAMA DE CONTROL

3.1 INTRODUCCIÓN

En este capítulo se describe la coordinación entre las distintas partes del proyecto. Se explica también el funcionamiento de los programas de los distintos elementos que conforman el sistema de control del Robot Acompañante y se presenta sus diagramas de flujo.

La programación ha sido realizada en tres plataformas diferentes.

El programa para el elemento central de control, el microcontrolador ATmega164, fue desarrollado en la herramienta Bascom-AVR, que es un compilador de lenguaje BASIC con librerías para microcontroladores de ATMEL.

El programa para la utilización del sensor KINECT y la adquisición de la posición de la persona es una aplicación para PC con Windows 7 y fue desarrollada en el entorno Visual Studio, en lenguaje C# (C Sharp), usando las herramientas instaladas con el paquete KINECT for Windows SDK.

El programa para la interfaz de control inalámbrica en el teléfono inteligente es una aplicación para el sistema operativo Android y fue desarrollada con una herramienta web experimental, ofrecida por Google y el Instituto Tecnológico de Massachuset, llamada Appinventor que presenta un lenguaje de programación por bloques de instrucciones.

3.2 ESTRUCTURA DEL SISTEMA DE CONTROL

El elemento central del sistema es el microcontrolador, que ejecuta acciones en base a la información que recibe desde el computador, con datos del sensor KINECT; o desde el teléfono inteligente.

3.2.1 MODOS GENERALES DE FUNCIONAMIENTO

El Robot Acompañante tiene 2 modos de funcionamiento: manejo y seguimiento.

En el modo de manejo, el usuario puede conducir mediante el teléfono inteligente al robot con el mando virtual, cuyo programa le envía al microcontrolador los comandos de dirección de avance, retroceso, giro derecho, giro izquierdo y detención.

En modo de seguimiento, el robot está en capacidad de seguir al usuario mientras camina, gracias al dato de posición que envía el programa del computador.

Existe un modo adicional por defecto, llamado modo de paro, en el cual el robot se mantiene inactivo hasta que el usuario elija un modo de funcionamiento en la interfaz táctil.

3.2.2 OPERACIÓN GLOBAL DEL ROBOT ACOMPAÑANTE

Previamente, deben activarse los interruptores de alimentación para los circuitos de potencia y de control, e iniciarse en el computador la aplicación para el KINECT.

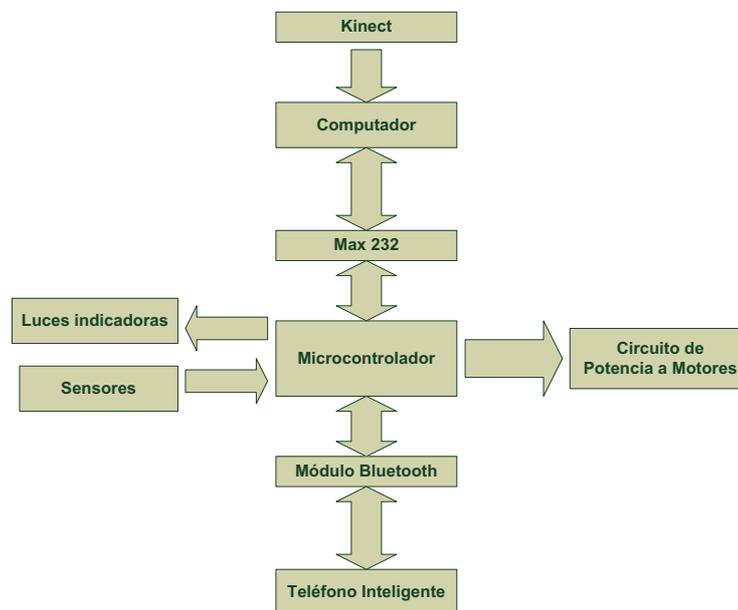


Figura 3. 1 Relaciones entre los componentes sobre los que actúa el sistema de control

Entonces, el usuario puede abrir la aplicación de la interfaz de control en el teléfono inteligente y conectarse al robot. Inmediatamente, el microcontrolador empieza a reportar el estado de los sensores de nivel de gases GLP y CO a la interfaz del teléfono. Dicho reporte es independiente y persiste durante todo el funcionamiento del robot.

En este momento, el robot se encuentra por defecto en el modo de paro. Los motores están detenidos. No existe comunicación con el computador. La forma en que se indica este modo en el robot es encendiendo sólo una de las dos luces de faros.

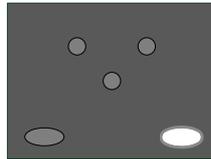


Figura 3. 2 Luces que indican que el robot está encendido y en el modo de paro.

Aquí el usuario puede escoger entre el modo de manejo o el modo de seguimiento.

Para manejar al robot, el usuario debe pulsar el botón “Manejo” de la interfaz del teléfono, lo cual despliega en la pantalla el mando táctil de 5 posiciones. Cuando el usuario mueve el mando, la interfaz envía los códigos de la dirección al microcontrolador, el cual asigna velocidad y sentido de giro a cada motor, para conducir en la dirección deseada.

El modo de manejo se muestra en el robot encendiendo sus dos faros y también se le informa de esta elección al computador.



Figura 3. 3 Botón de la interfaz para manejo y luces indicadoras para este modo.

Para el modo de seguimiento, el usuario debe pulsar el botón “Seguir” en la interfaz del teléfono. Esta acción comunica al microcontrolador que ponga al robot en un estado de *seguimiento en espera*. La razón de este estado preliminar de

seguimiento es que el robot, para poder seguir, necesita que el usuario se ubique en una zona inicial, frente al robot.

El estado de espera es reportado al computador para habilitar dicha zona inicial y es mostrado en el robot encendiendo una primera luz verde.



Figura 3. 4 Botón de la interfaz para seguimiento en espera y luces indicadoras para este modo.

Si el KINECT ya detecta al usuario en la zona inicial, el computador le pide al microcontrolador que ponga al robot en un estado de *seguimiento listo*.

El estado de seguimiento listo es mostrado en el robot encendiendo una segunda luz verde. Además en la interfaz del teléfono se muestran botones de inicio y detención.

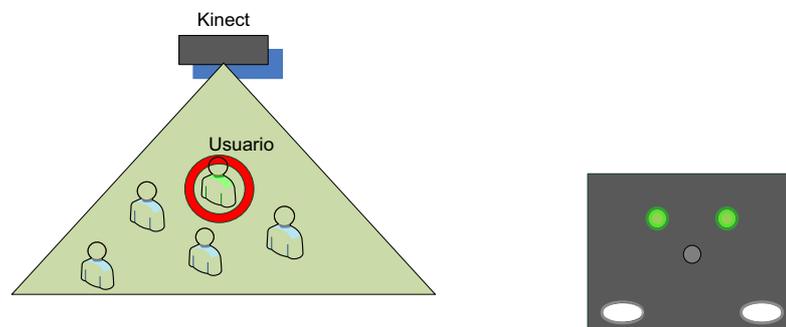


Figura 3. 5 Esquema de posición inicial del usuario en el campo de vista del KINECT y luces indicadoras para el modo de seguimiento listo

En el estado de seguimiento listo, el usuario puede pulsar ya el botón de inicio de la interfaz, ordenando al microcontrolador poner al robot en el estado definitivo de *seguir*. Este es el estado principal del modo de seguimiento, en el cual, el computador le reporta constantemente al microcontrolador la posición del usuario relativa al sensor KINECT, mientras ignora a otras personas que puedan aparecer en el campo visual.

El microcontrolador usa esta posición para realimentar compensadores digitales. Éstos deben mantener constantemente a la persona cerca del robot y en el centro de su campo visual, mediante la correcta regulación del avance y giro de los motores, logrando que el robot siga al usuario.

El estado de seguir se muestra en el robot, apagando las luces verdes y encendiendo una luz amarilla central.

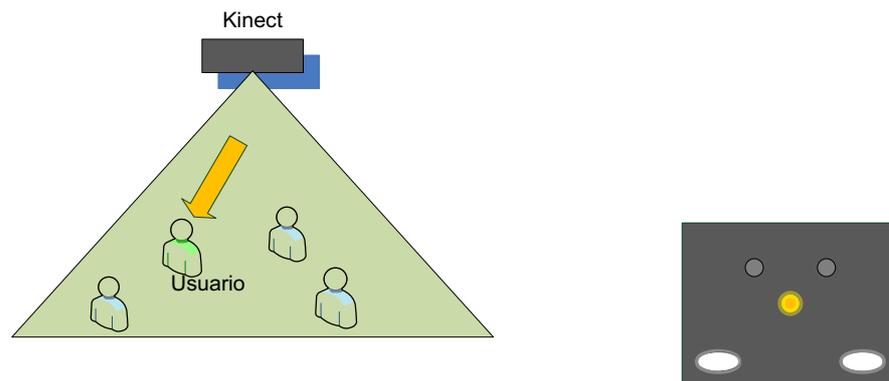


Figura 3. 6 Esquema del usuario siendo seguido, dentro del campo visual del KINECT y luces indicadoras para este modo.

Los sensores que evitan colisiones influyen en este estado del robot, deteniéndolo o desviándolo parcialmente del seguimiento que está realizando.

Si durante la caminata la persona sale del campo visual del sensor o es totalmente obstruida, el robot regresa al estado de seguimiento en espera, activándose de nuevo la zona inicial.

Los modos de manejo, seguimiento o paro pueden ser elegidos y alternados en cualquier instante por el usuario desde la interfaz del teléfono.

3.3 PROGRAMA DEL MICROCONTROLADOR

3.3.1 PROGRAMA PRINCIPAL

3.3.1.1 Configuración inicial

El microcontrolador se encuentra en constante comunicación bidireccional con la computadora y el teléfono inteligente, por lo que se configuran y se abren los 2

puertos seriales UART que posee. Ambos trabajan en modo asíncrono con un bit de parada y ninguno de paridad. El UART #1 corresponde al enlace RS232 para la computadora y el UART #2 está conectado al módulo Bluetooth. Ambos se configuran a una velocidad de 9600 baudios.

El modelo de recepción de la comunicación con el teléfono inteligente es de polling, es decir que revisa periódicamente si ha llegado un dato. Por otro lado, el modelo de recepción de la comunicación desde el computador es por interrupción, es decir que se dispara instantáneamente cada vez que le llega un dato, por ello se habilita la interrupción por recepción (URXC) y todas las interrupciones.

Para la lectura de sensores de gas y ultrasónicos, se habilita el conversor análogo digital en modo no diferencial con voltaje de referencia de la alimentación y demás configuraciones por defecto.

Se configura el puerto C como salida digital para las luces led indicadoras y 4 pines del puerto B también como salidas digitales para las señales de sentido de giro del circuito de potencia.

Finalmente se configura el Timer 1, el cual tiene dos registros comparadores y salidas independientes, en modo PWM de 9 bits y con preescalador de 64, lo cual con un oscilador principal de 8MHz resulta en una frecuencia del PWM de 120Hz.

Con las configuraciones listas se procede al programa en si. Primero se describirá el funcionamiento del programa principal, luego el de la subrutina de interrupción y de las otras subrutinas.

El programa principal del microcontrolador cumple las siguientes tareas secuenciales:

3.3.1.2 Tarea 1: Lectura de canales analógicos

El programa toma la lectura de los sensores analógicos en variables para el sensor de gas, el ultrasónico derecho y el ultrasónico izquierdo.

A continuación se analiza el valor del sensor de gas y se clasifica la lectura dentro de categorías de niveles, nulo, bajo, medio y alto; cada una con su correspondiente código listo para ser comunicado a la interfaz del teléfono.

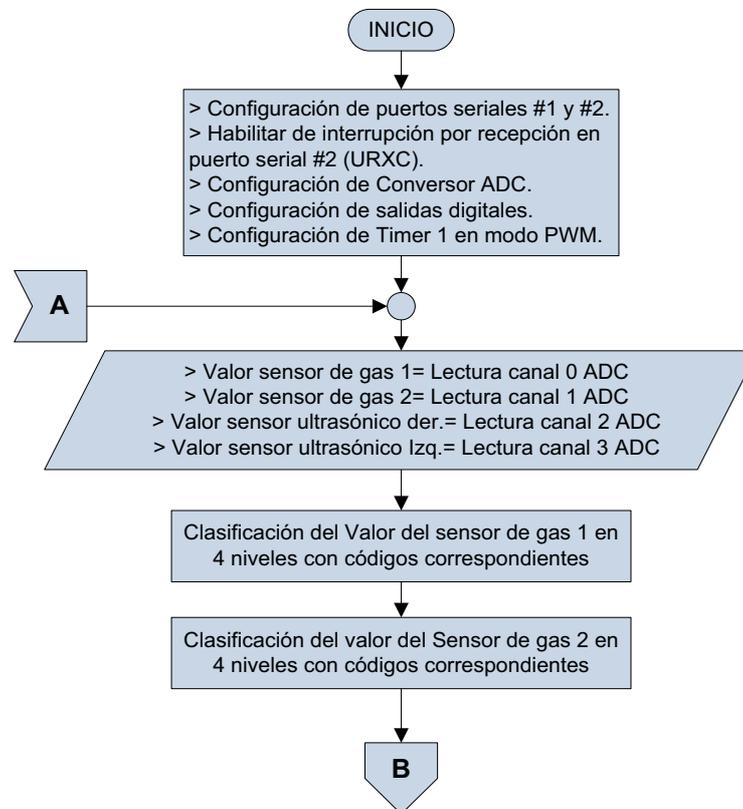


Figura 3. 7 Diagrama de flujo de Configuración Inicial y la Tarea 1

3.3.1.3 Tarea 2: Recepción y clasificación de códigos recibidos de la interfaz del teléfono

El programa revisa si le ha llegado un dato desde el teléfono y si ese es el caso, clasifica el tipo de código recibido. Las tres clasificaciones asignadas para el código son: código de pedido de estado de funcionamiento, código de dirección de manejo manual y código de pedido de respuesta.

El código de pedido de respuesta significa que el teléfono requiere y está listo para que le envíen información. Si ha llegado este código, el microcontrolador inmediatamente le responde con el dato del sensor 1, o del sensor 2, o de algún aviso de cambio de estado.

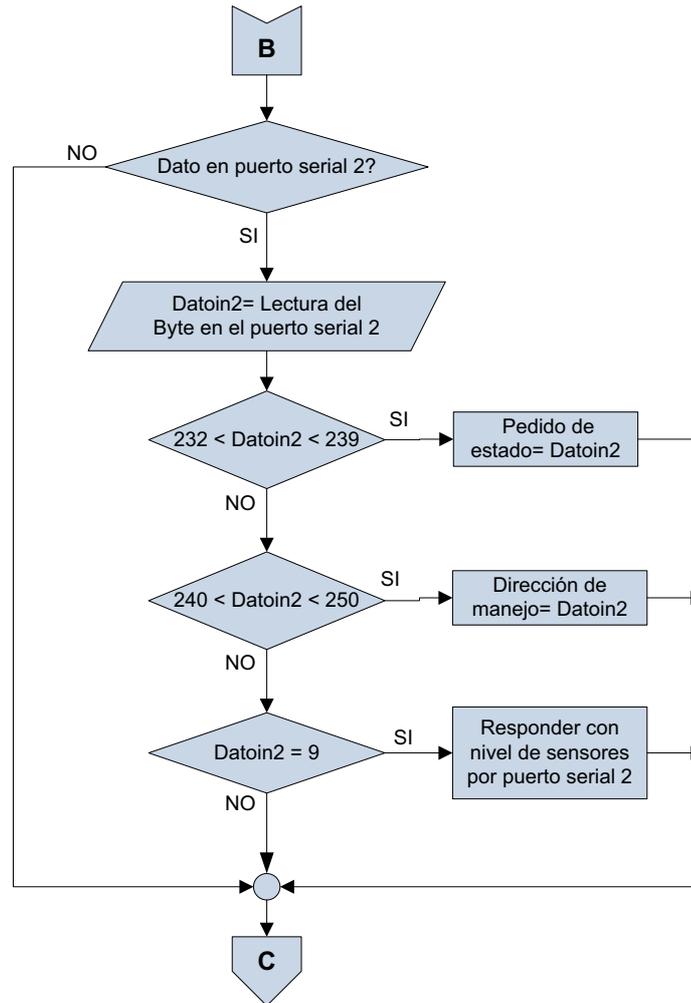


Figura 3. 8 Diagrama de flujo de la Tarea 2

3.3.1.4 Tarea 3: Determinación del estado actual del robot

Los estados posibles del robot son, por diseño, cinco: estado de paro, estado de manejo, estado de seguimiento en espera, estado de seguimiento listo y estado de seguir.

Solo si el programa ha recibido un nuevo código de pedido de estado de funcionamiento, se procede a evaluarlo, para saber si se cambia el estado actual del robot a un nuevo estado; si no existe dicho pedido, esta tarea no se realiza.

Los pedidos, desde la interfaz del teléfono, de paro y de manejo seleccionan directamente los correspondientes estados de paro y manejo. El estado de seguimiento en espera es seleccionado por el pedido de modo de seguimiento,

pero también por el código del botón de pausa de la interfaz y por el aviso desde la computadora de pérdida de la persona durante el seguimiento.

Si el estado actual es de seguimiento en espera y además la computadora ha enviado el código de la detección de un usuario en la zona inicial, se pasa a seleccionar el estado de seguimiento listo.

Si el estado actual es de seguimiento listo y ha llegado el código del botón inicio del teléfono, entonces se pasa a seleccionar al estado principal de seguir.

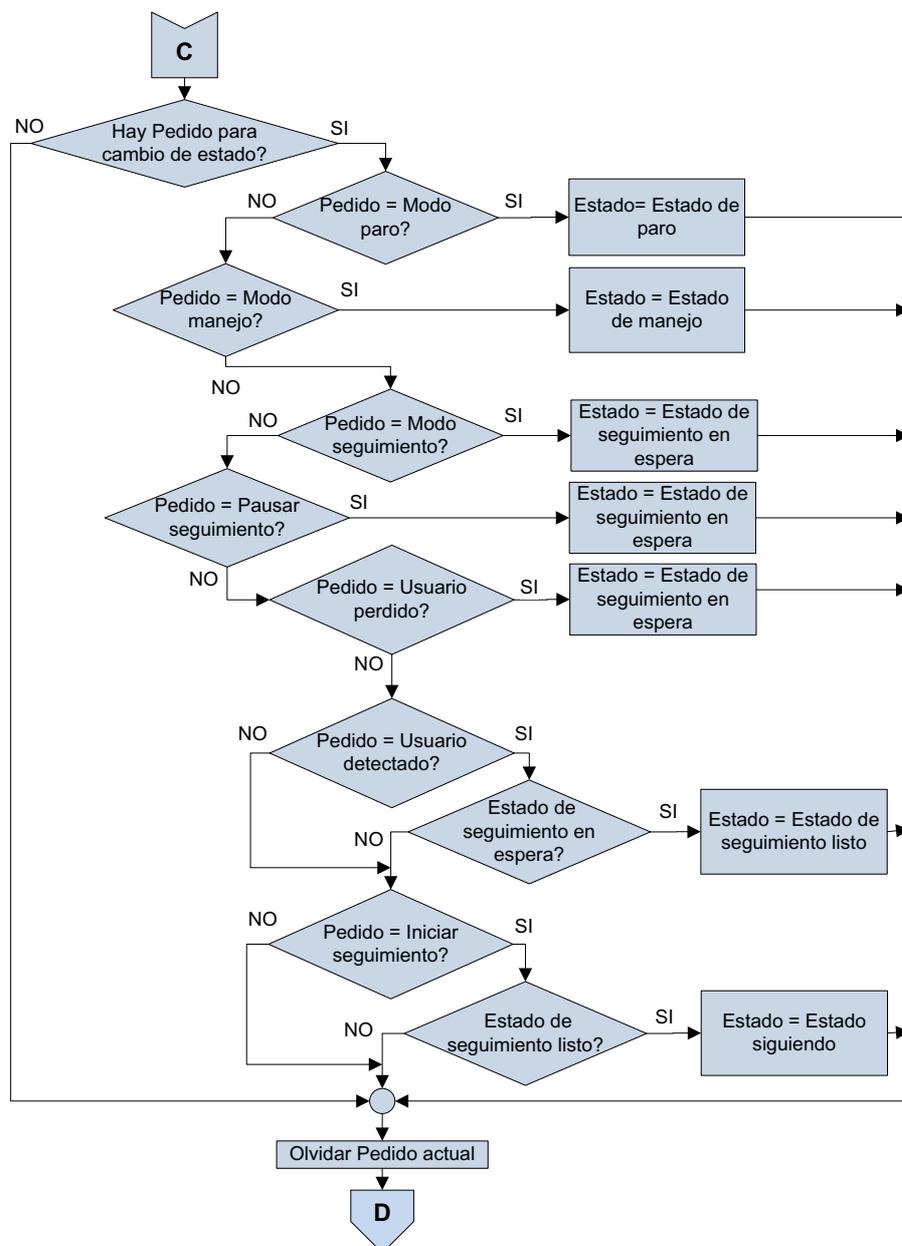


Figura 3. 9 Diagrama de flujo de la Tarea 3

3.3.1.5 Tarea 4: Ejecución de órdenes del estado actual del robot

El programa evalúa cual de los cinco estados posibles del robot está actualmente seleccionado.

Si está seleccionado el estado de paro, se llenan con cero las variables de velocidades de giro y avance del robot (enviando así señales nulas a los motores) y se demuestra este estado apagando todas las luces indicativas del robot, excepto un faro.

Si está seleccionado el estado de manejo, el programa llama a **la subrutina de manejo**, la cual revisa el valor del código de dirección (previamente receptado por la tarea 1), que ha llegado desde la interfaz de teléfono. A cada código de dirección le asigna una correspondiente combinación de valores de velocidad de giro y velocidad de avance. Además el programa indica este estado apagando las otras luces indicadoras y prendiendo los 2 faros del robot.

Si está seleccionado el estado de seguimiento en espera, el programa encera las variables de velocidad de giro y avance (manteniendo al robot detenido) e indica este estado encendiendo una luz indicadora verde, junto con los faros y apagando las luces restantes.

Si está seleccionado el estado de seguimiento listo, las variables de velocidad se mantienen en cero y, para mostrar este estado se enciende, aparte de las otras luces, una segunda luz verde indicadora.

Si está en el estado de seguir, el programa lo muestra apagando las luces indicadoras verdes y encendiendo la luz amarilla central, junto con los faros,

El programa primero llama a la **subrutina de ultrasónicos** para revisar si existe alguna detección de colisión. Si la hay, la subrutina asigna los componentes de velocidad de giro y alcance necesarios para evitar el choque; y el programa continúa a la siguiente tarea con esas velocidades.

Si por el contrario no se ha reportado peligro de colisión, el programa pasa a ejercer su función principal, llamando a la **subrutina de compensadores**

(explicada más adelante), para el cálculo de las velocidades de control de giro y alcance, según la posición del usuario que se receipta desde el computador.

3.3.1.6 Tarea 5: Ejecución de la locomoción del robot

Luego de haber ejecutado los comandos del estado actual del robot, el programa principal llama a la **subrutina de tracción diferencial**. Esta subrutina se encarga de tomar los valores de las componentes de velocidad de giro y de alcance, para traducirlos a velocidades para cada motor y luego a valores de relación de trabajo PWM para los circuitos de potencia de los motores.

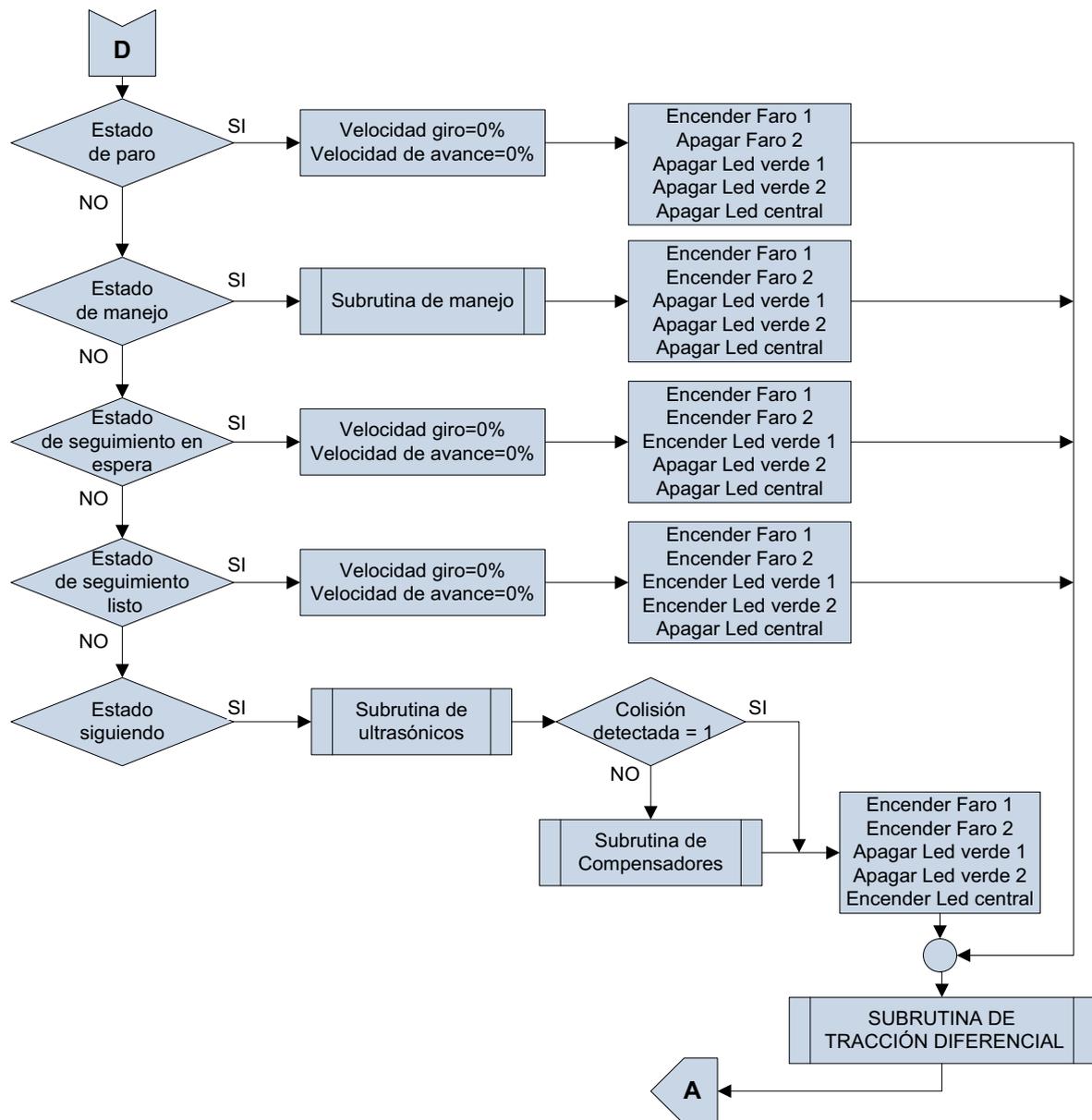


Figura 3. 10 Diagrama de flujo de la Tarea 4 y 5

3.3.2 SUBROUTINA DE INTERRUPCIÓN POR RECEPCIÓN DE DATOS DESDE LA COMPUTADORA

Esta subrutina realiza una función similar a la que realiza la tarea 2 del programa principal. Pero en lugar de recibir datos por polling se ha elegido recepción por interrupción debido a que, cuando el robot se encuentra siguiendo a la persona, de la computadora llegan rápidamente datos consecutivos de su posición, por lo que el poder recibirlos sin fallar tiene la mayor prioridad.

Cuando se recibe un dato por el puerto serial #2, se dispara la interrupción. Dentro de su subrutina, se recibe el dato y se lo clasifica. Las tres clasificaciones asignadas para el código son: código de pedido de estado de funcionamiento, código de posición en profundidad del usuario y código de posición horizontal del usuario. Además se levantan banderas internas correspondientes a cada clase de dato recibido para informar al programa principal acerca del tipo de nuevo dato que ha llegado.

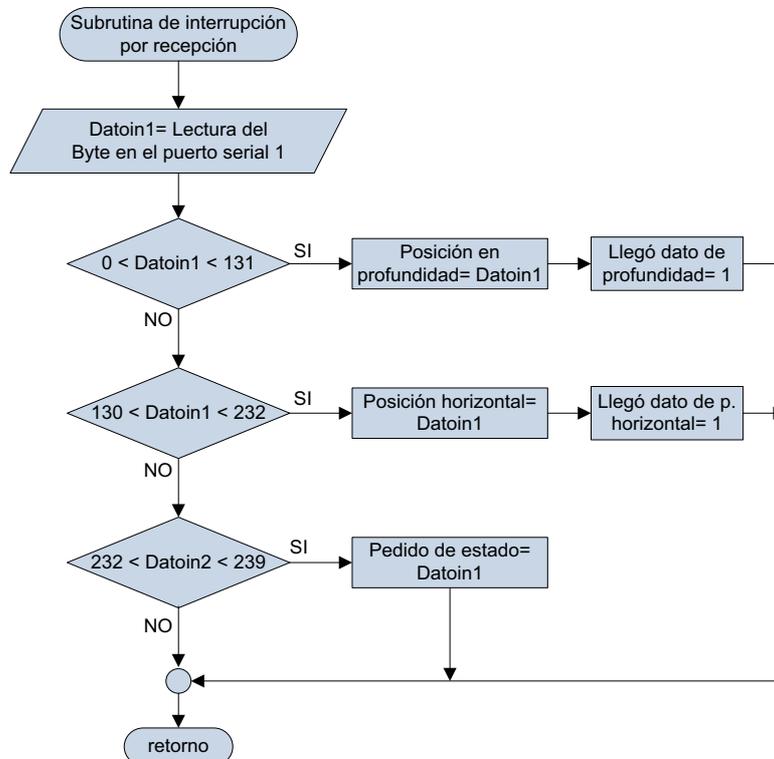


Figura 3. 11 Diagrama de flujo de la subrutina de interrupción por recepción

3.3.3 SUBROUTINA DE MANEJO

Como se había mencionado, esta subrutina revisa el código de dirección enviado por la interfaz y para cada uno se asigna una correspondiente combinación de valores de velocidad de giro y velocidad de avance. Por ejemplo, si el código de dirección corresponde al de retroceso del robot, la variable velocidad de giro será igual a 0% pero la variable velocidad de avance será de - 60% (negativa).

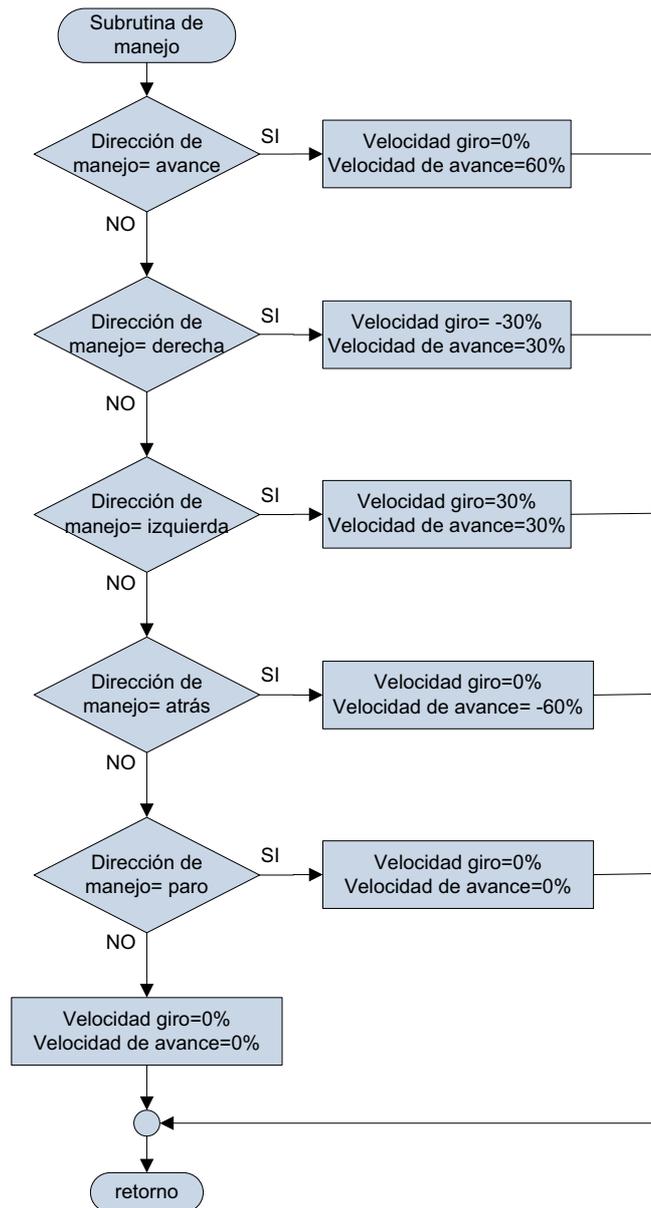


Figura 3. 12 Diagrama de flujo de la subrutina de manejo

3.3.4 SUBROUTINA DE ULTRASÓNICOS

El programa evalúa las lecturas de los sensores ultrasónicos previamente realizadas en la tarea 1; según su nivel de detección se determinan valores de componentes de velocidad de giro y de alcance necesarias para que en lo posible el robot se mueva fuera de la ruta de colisión y se levanta una bandera de colisión. Si no se detecta peligro, no se asigna ningún valor de velocidad, se baja la bandera de colisión y se retorna al programa principal.

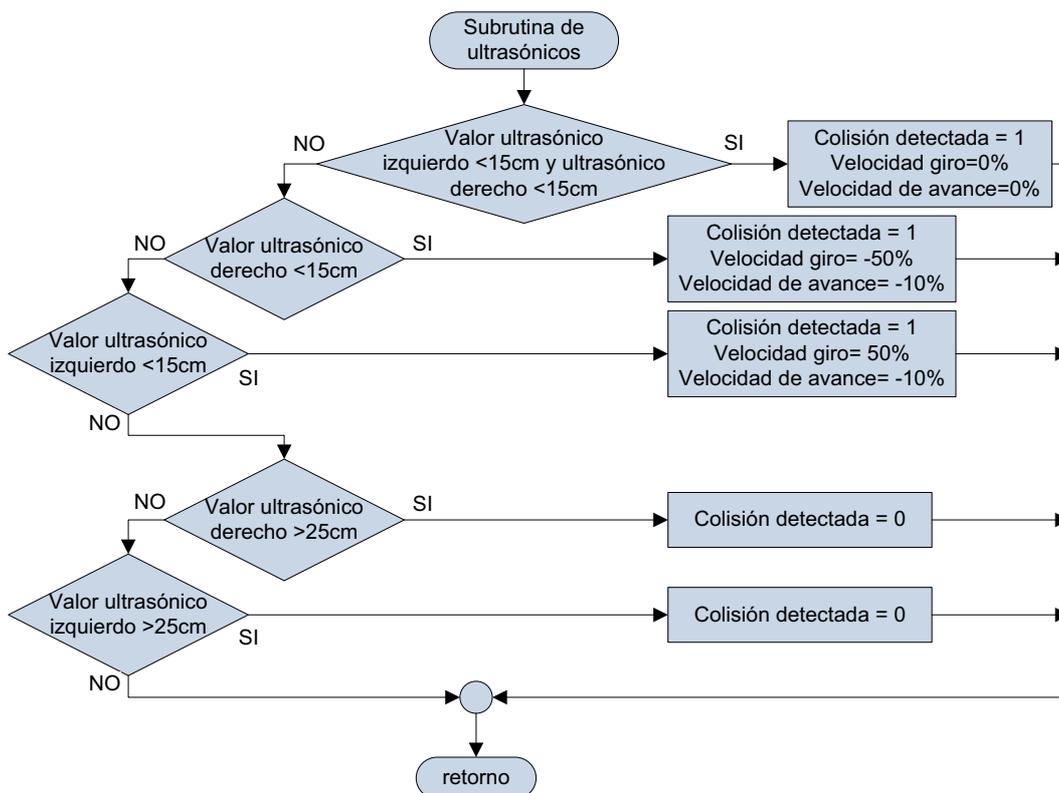


Figura 3. 13 Diagrama de flujo de la subrutina de ultrasónicos

3.3.5 SUBROUTINA DE COMPENSADORES

La rutina de compensadores comprende el cálculo de señales de control de dos compensadores digitales en base a los datos de posición enviados por la computadora. La subrutina revisa si ha llegado un nuevo par de códigos de posiciones horizontal y de profundidad del usuario, de haberlo hecho procede al cálculo de compensadores. Primero procede con el compensador de giro y luego con el de alcance.

Para el compensador de giro, el setpoint de la posición horizontal, es decir la posición relativa en que se desea mantener al usuario, es la posición central del campo visual del KINECT o posición cero.

Desde la posición cero se han definido 50 posiciones positivas a la izquierda y 50 posiciones negativas a la derecha hasta los bordes del campo visual. El error de posición horizontal se obtiene entonces al restar la posición horizontal actual de la posición cero.

El error de posición horizontal entra a la ley de control de un compensador tipo red de adelanto y se obtiene la señal de control para la velocidad de giro del robot expresada en porcentaje. Un valor de control positivo significa girar a la izquierda y un valor negativo a la derecha. La señal de control de velocidad de giro es el componente de velocidad responsable de corregir el rumbo del robot, de forma que se mantenga observando en dirección al usuario.

A la velocidad de giro calculada se la hace pasar por ciertas limitaciones para un mejor comportamiento en la práctica.

Primeramente, la acción amortiguadora de la red de adelanto puede provocar que si el error está disminuyendo rápidamente, la señal de control calculada cambie de signo. Este cambio de signo implica en la práctica un cambio brusco y momentáneo de giro que provoca inestabilidad del robot y estrés mecánico. Es por esta razón que se introduce la restricción de que si la señal de control de giro se ha calculado con signo opuesto al del error actual, se limite su valor a cero como valor mínimo

Otros límites a los que se le somete a la señal de giro calculada son sus valores máximos de 50% y -50% para cuando el usuario se ha alejado del setpoint de profundidad; y de 100% y -100% si el usuario se mantiene cerca del setpoint de profundidad.

Para el compensador de avance, el setpoint de profundidad, es decir, la distancia a la que se desea mantener al usuario, es de 120 cm desde el sensor KINECT.

El error de posición en profundidad se obtiene entonces al restar la profundidad actual del setpoint.

El error de posición en profundidad entra a la ley de control de otro compensador tipo red de adelanto y se obtiene la señal de control para la velocidad de avance del robot expresada en porcentaje. La señal de control para la velocidad de avance es el componente de velocidad responsable de mantener al robot cerca del usuario, mientras se traslada, sin atropellarlo.

La velocidad de avance calculada por el compensador, también debe ser sometida a ciertas limitaciones de valor máximo y mínimo.

Las velocidades de giro y de alcance van a superponerse para determinar la velocidad de cada motor, de manera que para alguno de los dos motores, habrá una suma de las dos componentes de velocidad. El problema es que cada motor naturalmente tiene un tope de velocidad de 100%, sin embargo la suma de velocidades podría sobrepasar ese tope; por ejemplo 30% de giro más 90% de alcance significarían 120% para un motor. Para evitar que esto ocurra, se ha decidido dar prioridad a la velocidad de giro y limitar el valor de la velocidad de avance; para que, cuando se combinen, la máxima suma de sus valores absolutos sea de hasta 100%. Entonces, por ejemplo, si el valor absoluto de la velocidad de giro resultó de 30%, la velocidad de avance será limitada a 70% aunque su valor original haya sido 90%.

Otra limitación para la velocidad de avance es que su valor mínimo sea cero y no llegue a ser negativo. Valores negativos representarían una marcha en reversa. Para ir en reversa el usuario debería acercarse al robot y a la zona muerta de medición del KINECT. Este acercamiento puede provocar medidas erráticas de posición y por lo tanto comportamiento inestable del robot. Por eso es que se deshabilita la marcha reversa del robot mediante una velocidad de avance mínima de cero.

Finalmente, ya con los valores de velocidad de alcance y giro definitivos, la subrutina termina con la memorización de estas velocidades y los errores de

posición calculados como valores anteriores para el próximo ciclo de cálculo de compensadores.

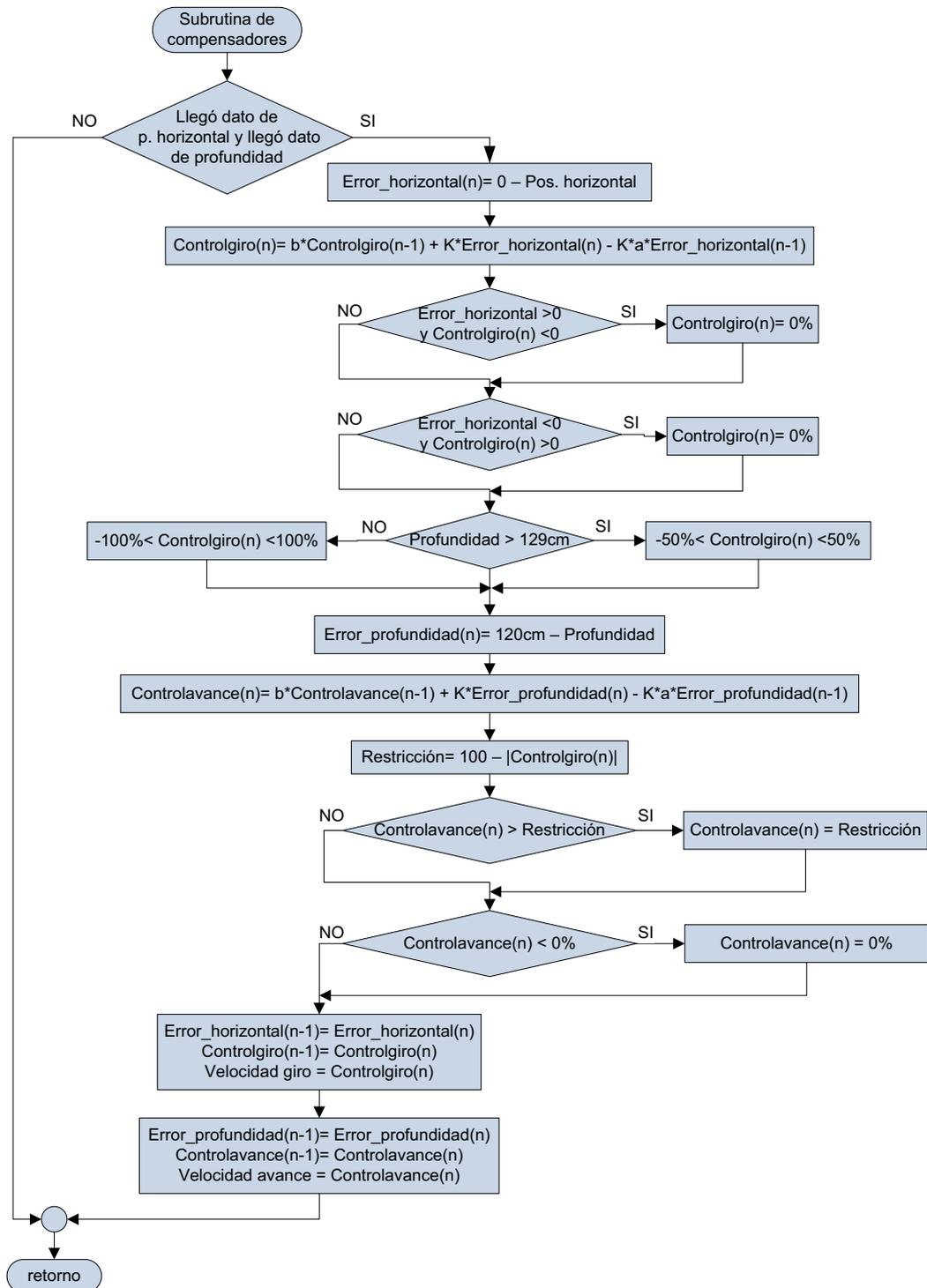


Figura 3. 14 Diagrama de flujo de la subrutina de compensadores

3.3.6 SUBROUTINA DE TRACCIÓN DIFERENCIAL

Antes de llegar a esta subrutina, el programa ha expresado la locomoción como dos componentes de velocidad: velocidad de avance y velocidad de giro porcentuales. Sin embargo, en esta subrutina, dichas componentes de velocidad son transformadas a velocidades para cada motor y luego a valores reales de PWM y sentidos de giro.

La dinámica de un robot con tracción diferencial depende de la velocidad independiente de cada uno de sus motores. Es decir de la velocidad de su motor derecho u_d y de la velocidad de su motor izquierdo u_i .

Si $u_d = u_i > 0$, entonces el robot avanza hacia adelante [16]. Si $u_i = -u_d \neq 0$, entonces el robot rota en sentido horario sobre el centro de su eje, porque sus ruedas estarán girando en direcciones contrarias, y no tendrá movimiento de avance [16]. Cuando estas circunstancias se combinan para formar trayectorias, se tiene que la velocidad de giro u_θ es proporcional a la diferencia de velocidades angulares de las ruedas y que la velocidad de avance u_w depende del promedio de las velocidades angulares de cada rueda [16]. Entonces, las fórmulas para las velocidades de giro y avance son

$$\frac{1}{2}(u_d + u_i) = u_w$$

$$(u_d - u_i) = u_\theta.$$

Haciendo el despeje de estas fórmulas, las expresiones para la velocidad para cada motor a partir de la combinación de componentes de avance u_w y giro u_θ que debe calcular el programa son:

- $u_d = u_w + \frac{1}{2}u_\theta$, para el motor derecho
- $u_i = u_w - \frac{1}{2}u_\theta$, para el motor izquierdo

Sin embargo, cabe anotar que en cuanto a la velocidad de giro, las partes anteriores del programa ya tienen en cuenta su reducción a la mitad y entonces en realidad entregan una velocidad de giro $u_{\theta}' = \frac{1}{2} u_{\theta}$, así que las expresiones que calcula esta subrutina se simplifican a:

- $u_d = u_w + u_{\theta}'$, para el motor derecho
- $u_i = u_w - u_{\theta}'$, para el motor izquierdo

Los resultados de estas velocidades de cada motor son limitados por el programa al rango de -100% a 100%.

Entonces, se procede a separar el módulo de su signo en la velocidad asignada a cada motor. Los valores negativos significan velocidad en sentido inverso al de avance. El programa evalúa si la velocidad es negativa y registra esto en una bandera, llamada “bandera de inversión”.

Luego, el programa toma el valor absoluto de la velocidad porcentual de cada motor y lo ingresa a una tabla de transformación que toma los valores de 0% a 100% en pasos de 1% y entrega como resultado valores correspondientes para el registro comparador de PWM, de modo que el conversor dc-dc entregue el voltaje necesario para una respuesta de velocidad proporcional de motores.

Finalmente el programa actualiza las señales para el circuito de potencia de cada motor. Revisando si la bandera de inversión está levantada o no, se ponen señales digitales para sentido de giro correspondientes y se cargan los registros comparadores del módulo PWM con el valor asignado para la velocidad requerida.

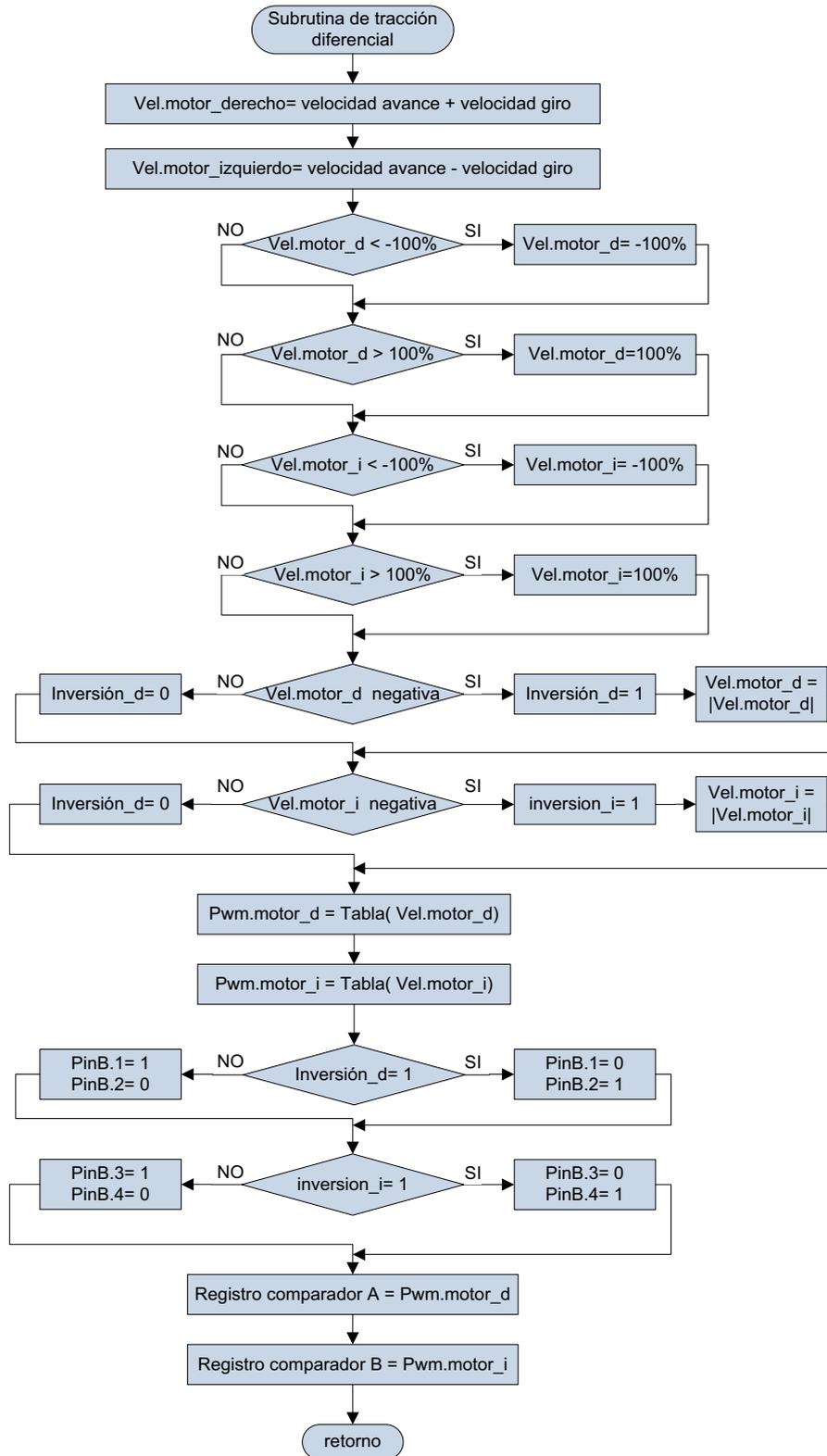


Figura 3. 15 Diagrama de flujo de la subrutina de tracción diferencial

3.4 RESUMEN DE LA CODIFICACIÓN PARA COMUNICACIONES

Debido a que existen múltiples comunicaciones entre los elementos de control y a que la transmisión de datos debe ser bastante rápida, se ha decidido simplificarla encapsulando todos los datos en un solo byte, dividiéndolo en rangos según el tipo de dato comunicado. La Tabla 3.1 resume dichas regiones del byte de comunicación.

Tabla 3. 1 Rangos del byte de comunicación

Tipo de dato	Rango en el byte de comunicación	
Coordenada Z, posición en profundidad (desde la PC al microcontrolador)	1- 130	1-4*
Coordenada X, posición Horizontal (desde la PC al microcontrolador)	131-231	
Códigos de modo de funcionamiento	232-240	
Códigos de dirección de manejo	241-250	

* Bytes utilizados para informar niveles del sensor de gases nocivos e inflamables (hacia el teléfono)

3.5 PROGRAMA DE LA INTERFAZ DEL TELÉFONO

La interfaz se compone de 3 paneles superpuestos intercambiables, correspondientes a los tres modos de funcionamiento: paro, manejo y seguimiento. El modo y panel activos se cambian eligiendo uno de los tres botones de modo. Otros elementos permanentes son los botones y avisos del control de la conexión Bluetooth y los textos que avisan el estado de los sensores de gas.

El programa del teléfono se basa en programación orientada a objetos y su estructura es similar a la de un programa en lenguaje Java, de manera que en lugar de tener una secuencia de instrucciones, el programa está segmentado en eventos que pueden ocurrir aleatoriamente y que afectan a los elementos de la

interfaz. El esquema y aspecto de dichos elementos gráficos de la interfaz son diseñados previa y separadamente de la programación lógica.

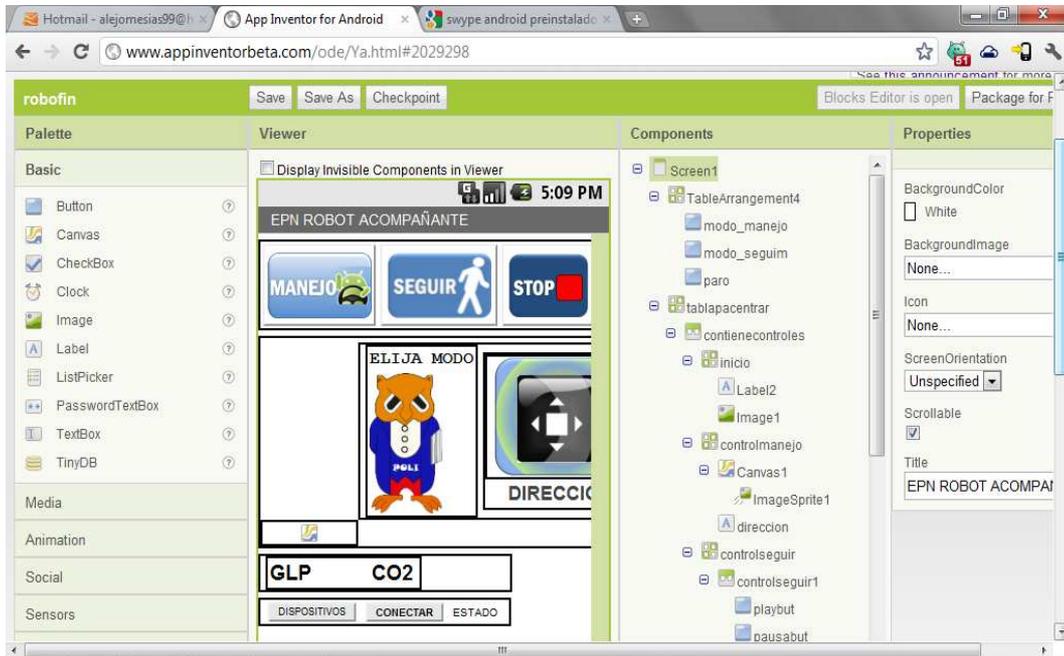


Figura 3. 16 Pantalla de programación de la interfaz gráfica.

Los eventos y su programación general son los siguientes:

a) Inicialización.

Ocurre cuando se abre el programa por primera vez. Aquí se debe definir ciertos aspectos y acciones iniciales de la interfaz. Como el modo de paro es el modo por defecto, en este evento se ha programado mostrar el panel de paro y ocultar los otros paneles.



Figura 3. 17 Panel inicial y de paro

b) Lista de equipos Bluetooth.

Se presenta cuando se pulsa el botón de dispositivos para elegir la dirección del módulo del robot. El programa recoge del sistema las direcciones de los equipos emparejados y los despliega. Cuando el usuario elige la del robot, el programa guarda esta dirección en una variable llamada "equipo".

c) Click en el botón que controla la conexión.

Cuando se pulsa este botón, que inicialmente se llama "Conectar", el programa primero consulta si es que ya existe una conexión. Si no existe, el programa ordena al recurso Bluetooth conectarse con la dirección guardada en la variable "equipo". El recurso le responde al programa si la conexión fue exitosa y si lo fue se despliega un aviso de "conectado", y el nombre del botón es cambiado a "Desconectar". Pero si no hay éxito en la conexión, el programa despliega un aviso de falla. Si por el contrario ya existía una conexión, entonces el botón, que debe estar en modo de "Desconectar", ordena terminar esa conexión, además cambiar el nombre del botón de nuevo a "Conectar" y poner un aviso de "desconectado".

Si en este evento se establece una conexión, además se pone a correr un temporizador que genera un evento cada 100ms. Este evento será el encargado de solicitar constante y regularmente al microcontrolador la información del sensor de gas.

d) Temporizador

Es un evento interno que ha sido configurado para cada 100ms. Aquí el programa envía por Bluetooth un código de pedido de respuesta. Luego revisa si hay un dato recibido en su puerto. Si el dato contiene códigos correspondientes a niveles del sensor de gas, inmediatamente actualiza el texto de aviso de nivel de gas,

según sea el código. Si no había dato en el puerto, el texto del sensor se reemplazara por la frase “No dato”.

e) Click en el botón para modo de manejo

Al pulsar este botón, se hace visible el panel de manejo, junto con sus elementos. Oculta los otros paneles y envía al microcontrolador el código de pedido de modo de manejo.



Figura 3. 18 Panel de manejo.

f) El cursor de mando llegó a los bordes del área de mando.

En el panel de manejo, el usuario debe arrastrar el cursor de mando en la dirección deseada hasta topar cualquier borde del cuadrado de mando que lo encierra. Esto es lo que dispara este evento. Entonces, el programa le consulta al evento cual borde ha sido topado (cada borde está numerado). Según el número de borde topado, el programa envía un correspondiente código de dirección de manejo a través del recurso Bluetooth. Además se hace vibrar al teléfono como respuesta táctil y también se cambia la imagen del cursor de mando a la de un botón de pausa.

g) Click en el cursor de mando.

Ocurre cuando en lugar de arrastrar el cursor de mando, sólo se hace click en él. En este evento se ejecuta la detención del robot enviando por Bluetooth el código

de detención al microcontrolador. Además se hace saltar el cursor de mando al centro del cuadrado y cambia su imagen actual de botón de pausa a su apariencia normal de cursor.

h) Click en el botón de modo de seguimiento

Al pulsar este botón se hace visible el panel de seguimiento, junto con los elementos de control de este modo. Oculta los otros paneles y envía al microcontrolador el código de pedido de modo de seguimiento.



Figura 3. 19 Panel de seguimiento.

i) Click en el botón de inicio de seguimiento

Al pulsar este botón en el panel de seguimiento se envía al microcontrolador el código correspondiente de inicio de seguimiento..

j) Click en el botón de pausa de seguimiento

Al pulsar este botón en el panel de seguimiento se envía al microcontrolador el código correspondiente de pausa de seguimiento.

k) Click en el botón de modo de paro

Al pulsar este botón se oculta los otros paneles superponiendo el panel inicial de paro, que contiene sólo una carátula; y envía al microcontrolador el código de pedido de modo de paro.

3.6 PROGRAMA DE LA COMPUTADORA PARA EL SENSOR KINECT

3.6.1 CONSIDERACIONES GENERALES EN PROGRAMAS PARA MANEJO DEL SENSOR KINECT

La programación para la adquisición de la posición del usuario en el campo visual del KINECT también es de tipo orientada a objetos.

Está realizado en lenguaje C# (C sharp) y en entorno Visual Studio. Cabe notar que para este entorno de programación existen dos formatos de aplicaciones Windows en los que se puede desarrollar. El primero se llama Windows Form (WF) y el segundo es la Windows Presentation Foundation (WPF). Sin embargo, el paquete KINECT for Windows SDK es compatible solo con el formato WPF.

A pesar de no ser necesario para el funcionamiento de este robot, se ha incluido una interfaz gráfica para la pantalla de la computadora, con video procesado a partir del flujo de imagen de profundidad del sensor KINECT, para que de cierta manera las personas puedan visualizar la forma en que el robot “observa” su entorno y el reconocimiento de la Figura humana que internamente ocurre.

Al igual que el programa de la interfaz del teléfono inteligente, el esquema y elementos gráficos de la interfaz de la PC se diseñan previa y separadamente de la programación lógica. La programación lógica tampoco es secuencial sino que se divide en eventos aislados.

Para poder usar en la práctica el API de reconocimiento de Figura humana del KINECT SDK como parte de un nuevo proyecto WPF de Visual Studio, se debe agregar el archivo de librerías llamado *Microsoft.Research.KINECT* a la carpeta de referencias del proyecto. Este archivo es el que contiene los recursos disponibles al programador, producto del reconocimiento de poses humanas del

KINECT SDK. Los recursos usados para este robot acompañante pertenecen específicamente a la librería *Microsoft.Research.KINECT.Nui* y son los siguientes:

- *DepthAndPlayerIndex*, que es el que contiene el flujo de imagen de profundidad junto con códigos por pixel que señalan la persona a la que pertenece dicho pixel.
- *SkeletalTracking*, que contiene la información de la posición de las personas y de la posición de juntas específicas del esqueleto de cada persona. Las características de este recurso se detallan a continuación.

3.6.1.1 Características del recurso *SkeletalTracking*

El API de reconocimiento de Figura humana del KINECT SDK, luego de haber identificado en una persona sus segmentos corporales mencionados en el capítulo 1, calcula y determina en base a ellos una estructura corporal llamada esqueleto, que es un conjunto de 20 puntos clave del cuerpo humano, denominados juntas, las cuales se muestran en la Figura 3.20 [29].

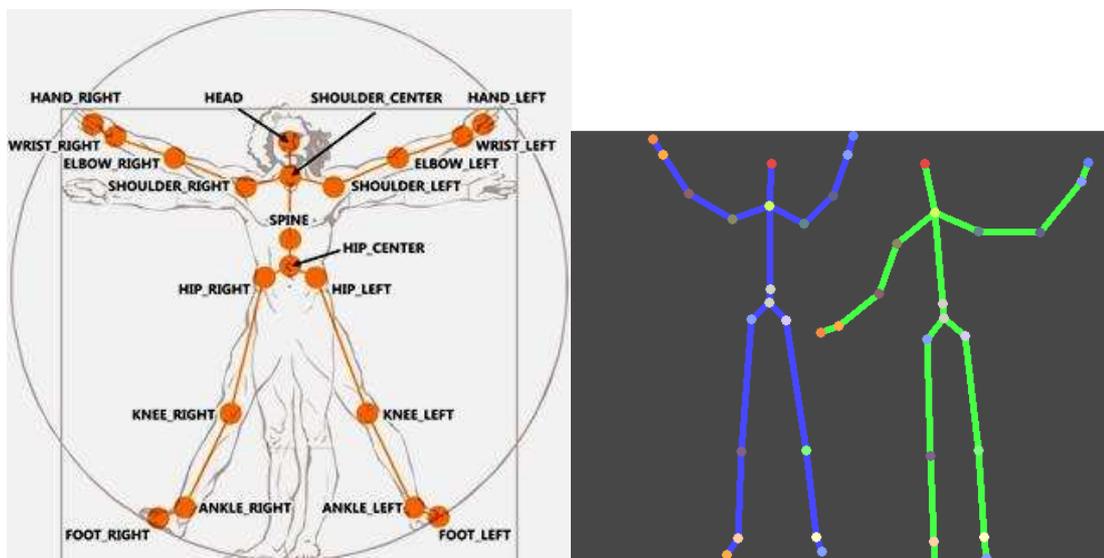


Figura 3. 20 Juntas del esqueleto calculadas, tomado de [29]

El API del KINECT SDK puede reconocer hasta a 6 personas al mismo tiempo en el campo visual del KINECT. A cada una de estas 6 personas le asigna un número para poder diferenciarlas. Este número, cuyo rango va del 0 al 5, se llama *Playerindex* y el KINECT SDK es capaz de asignarlo consistentemente durante todo el flujo de imágenes [29]. De esta manera se obtiene no solo el

reconocimiento de personas en una imagen sino también su rastreo a través del tiempo.

El número *playerindex* de cada persona se mantiene consistente mientras estén dentro de la escena. Si la persona sale, pierde el número que tenía. Cuando otra persona entra o reingresa, los números de *playerindex* disponibles son otorgados aleatoriamente [29]. En resumen, el *playerindex* no expresa la identidad real de las personas, sino una diferenciación relativa en la escena actual.

De las 6 personas que el API KINECT SDK puede rastrear, solo realiza el análisis de esqueleto a máximo dos de ellas. Es decir, que es posible tener información de posición de juntas de esqueleto de hasta dos personas y del resto de personas rastreadas sólo se puede tener el dato de su posición general central [29]. Las dos personas prioritarias que reciben análisis de esqueleto, serán las dos primeras que entraron en escena, sin embargo, si la visibilidad de una de ellas se degrada o se pierde y hay más personas en la escena, el análisis de esqueleto es reasignado aleatoriamente.

En resumen, el recurso *SkeletalTracking* contiene, para uso del programador, los datos de posición general de hasta 6 esqueletos y los datos de posición de juntas de hasta 2 prioritarios de esos 6 esqueletos, con cada esqueleto diferenciado con su propio *playerindex*. Estas posiciones están expresadas en coordenadas cartesianas x , y , z , cuyo origen está en el centro del sensor, como se observa en la Figura 3.21. Los valores de los ejes X y Y , varían de -1 a 1 y son adimensionales. Los valores del eje z van de 0 a 4 metros.

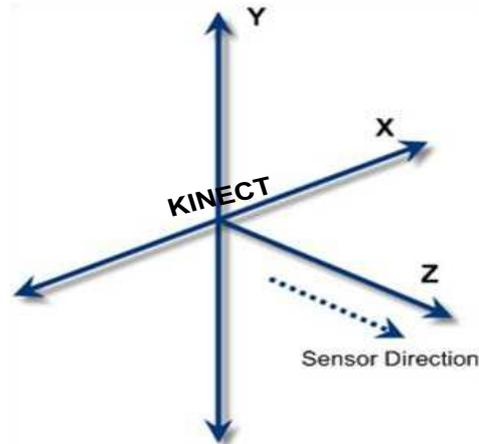


Figura 3. 21 Sistema de coordenadas de las posiciones espaciales de esqueletos y sus juntas, tomado de [29]

3.6.2 DESCRIPCIÓN DEL PROGRAMA DE COMPUTADORA DESARROLLADO PARA EL ROBOT ACOMPAÑANTE

Como se mencionó, este programa se divide y organiza en eventos, que no ocurren secuencialmente, sino a modo de interrupciones cuando algo sucede; y también se organiza con subrutinas, que son llamadas por el código de algún evento para una tarea específica.

El evento inicial ocurre solo cuando se abre el programa. Aquí se habilitan los recursos que se van a usar y los nuevos eventos de librerías que se van a necesitar. En este caso se inicializa los recursos de *DepthStream*, *ColorStream* y *SkeletonStream*. También se crean eventos para cuando está listo un cuadro de imagen de profundidad (*DepthFrameReady*), uno de video (*ColorFrameReady*), y para cuando está listo un cuadro de SkeletalTracking (*SkeletonFrameReady*).

Para el robot acompañante, la aplicación que se ha programado es un sistema de seguimiento del usuario elegido, utilizando los datos de rastreo del KINECT. La mayor parte del código de programación de este sistema se aloja dentro del evento *SkeletonFrameReady* que es el evento principal; ocurre cuando el KINECT SDK tiene listo un cuadro con el conjunto de posiciones de esqueletos y juntas en base a una imagen de profundidad llegada desde el KINECT.

3.6.2.1 Programación en el evento `SkeletonFrameReady`

Cada vez que se dispara este evento, la aplicación primeramente evalúa mediante banderas si el robot está en el estado que se ha definido como *siguiendo* o *seguimiento en espera*. Este estado se relaciona con el estado original de seguimiento del robot reportado por el microcontrolador a través de comunicación serial.

Si el estado actual es de *seguimiento en espera*, la aplicación activa la zona de reconocimiento inicial, que es aproximadamente un cuadrado de 40cm de lado ubicado a 120cm del centro del sensor. Lo que hace es entrar a un lazo que, para cada uno de los 6 posibles esqueletos en el campo visual, evalúa sus coordenadas x, z de su posición general para ver si se encuentran entre aquellos límites. El momento en que esta condición se cumple, el lazo se interrumpe y el programa memoriza el número `PlayerIndex` del esqueleto que se ha ubicado en la zona inicial, escogiendo así el usuario a seguir y además el estado actual es cambiado al estado de *siguiendo*.

Si el estado actual es *siguiendo*, el programa pregunta primero al recurso `SkeletalTracking` si el número de esqueleto memorizado como usuario aún está siendo rastreado. Si la respuesta es negativa, entonces se determina que el usuario ha sido perdido y el estado actual es cambiado a *seguimiento en espera*, Además se le informa de esto al microcontrolador mediante un código por comunicación serial.

Por el contrario, si el esqueleto memorizado aun es rastreado, la aplicación tiene la tarea de enviar la posición del esqueleto del usuario al microcontrolador.

La posición general del usuario es un dato un tanto grueso y ruidoso. Por ello en el momento en que el seguimiento es totalmente activo, se envía un dato más fino que es la posición de un punto específico del cuerpo del usuario. Concretamente se toma la posición de la juntura llamada *Spine* que corresponde al centro del torso de la persona.

Luego, la aplicación toma este dato y lo acondiciona para enviarlo al microcontrolador. Primeramente, de las tres coordenadas de posición se toman

solo la X y la Z. Los valores adquiridos de cada coordenada se codifican a través de ecuaciones lineales a códigos válidos para poder ser enviados dentro de un byte. El rango de los valores Z pasa de ser de entre 0,8 y 3,4 metros a números entre 1 y 130. El rango de X pasa de ser de valores entre -1 a 1 a números entre 131 y 231.

Las coordenadas X, Z codificadas son entonces enviadas consecutivamente a través del puerto serial.

3.6.2.2 Programación en el evento de recepción

Ocurre al recibir datos desde el computador, que son códigos que le informan a la computadora acerca de cambios de estado en el robot o pedidos que hace el usuario en la interfaz gráfica; por ejemplo cambio de estado de manejo a seguimiento; o el haber pulsado botones de inicio o pausa de seguimiento. Este evento simplemente guarda el código en una variable global para ser usado por otros eventos.

3.6.2.3 Programación en el evento DepthFrameReady

Ocurre cuando el KINECT SDK tiene listo un cuadro de imagen de profundidad con sus pixeles etiquetados, si pertenecen al cuerpo de una persona.

La aplicación que se ha programado para el robot acompañante hace uso de este evento para desplegar en la pantalla de la interfaz un flujo de video de imagen de profundidad, modificada con colores para representar visualmente el reconocimiento de la forma humana que está ocurriendo.

Si el robot no se encuentra en estado de seguimiento, todos los pixeles de la imagen de profundidad se colorearan en escala de gris. La escala de gris es proporcional a la información de profundidad del pixel. Pixeles fuera de rango o con información indeterminada se colorean de colores violeta y oliva.



Figura 3. 22 Programa para KINECT donde Imagen de profundidad se muestra en escala de grises en estados de no seguimiento.

Si desde el microcontrolador se informa que el robot está en estado de seguimiento en espera, la aplicación evalúa pixel por pixel si es que tienen etiqueta de persona; los que sí tienen, se colorean en escala de azul, de manera que las personas se vean azules sobre un fondo gris.



Figura 3. 23 Programa para KINECT donde Imagen de profundidad se muestra personas azules en estado de seguimiento en espera.

Si una persona se para en la zona inicial, ésta es reconocida como usuario; su Playerindex es memorizado y los píxeles que corresponden a este Playerindex se colorean de color verde, de manera que el usuario aparezca verde sobre un fondo gris señalizando que el robot se encuentra en estado de *seguimiento listo*. Otras personas diferentes al usuario seguirán siendo coloreadas de azul.



Figura 3. 24 Programa para KINECT donde la imagen de profundidad muestra al usuario en verde en estado de seguimiento listo

Cuando ya se ordena iniciar el seguimiento y si la etiqueta de los pixeles coincide con el número de Playerindex de la persona que está siendo rastreada como usuario, el pixel se colorea en escala de naranja, de manera que se observe un fondo gris con personas azules y sólo el usuario siendo seguido es resaltado en color naranja. Todo esto matizado según la profundidad que detecta el sensor KINECT. Además, la aplicación toma la coordenada X de posición general de todas las personas y las escala al espacio de la imagen de profundidad. Entonces pone en esa posición, sobre la imagen, pequeños círculos que marquen y representen la posición de cada persona que KINECT ve.



Figura 3.18

Figura 3. 25 Programa para KINECT donde la imagen de profundidad muestra al usuario seguido en naranja, en estado de seguimiento

CAPITULO 4

PRUEBAS Y RESULTADOS

En el presente capítulo se muestran los procedimientos y resultados experimentales de la implementación del prototipo. Primero las calibraciones necesarias en cuanto a respuesta de motores y compensadores para tener el comportamiento deseado del prototipo; y segundo las pruebas de desempeño global del sistema.

4.1. LINEALIZACIÓN DE LA RESPUESTA DE VELOCIDAD DE MOTORES

El prototipo funciona con dos motores DC alimentados a través de un conversor DC-DC tipo puente que es controlado mediante señales PWM provenientes de un microcontrolador. La relación de trabajo de la señal PWM es calculada a partir de los resultados de los controladores de avance y giro (descritos más adelante).

Teóricamente la relación de trabajo de la señal PWM debe mantener una relación lineal con la velocidad de cada motor. Mas, debido a las características propias de la construcción de los motores, existen desviaciones de la linealidad, que en la mayoría de casos se pueden despreciar. Pero en el caso del prototipo, el correcto funcionamiento de los controladores depende de la linealidad; por ello es necesario identificar y eliminar dichas desviaciones.

4.1.1. ADQUISICIÓN DE DATOS DE VELOCIDAD

Se realizó ocho pruebas haciendo que el vehículo gire sobre su propio eje; esto porque la velocidad del prototipo es proporcional a la velocidad de sus motores. Cada prueba se la hizo enviando una señal PWM con distinta relación de trabajo y se midió el número de vueltas por segundo del prototipo.

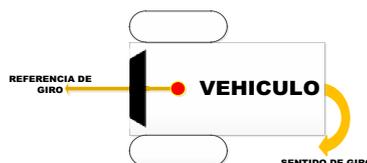


Figura 4. 1 Modelo de la prueba para adquisición de datos de velocidad

4.1.2. LINEALIZACIÓN

Con los datos obtenidos se realizó la grafica %Velocidad de giro (V_g) vs. Relación de trabajo PWM (R_t), notando un comportamiento no lineal. Mediante la herramienta “Agregar línea de tendencia” de Microsoft Excel, se obtuvo una curva de aproximación logarítmica con la siguiente ecuación:

$$V_g = 95,28 \ln(R_t) - 490$$

Debido a que aún persistían diferencias entre la curva real y la aproximación se realizo un ajuste manual, quedando la ecuación:

$$V_g = 95,28 \ln(R_t) - 498$$

Donde:

$$R_t = e^{\frac{V_g + 498}{95,28}}$$

Obteniendo con esto una aproximación final, que se ajusta mucho mejor a la curva real, como se puede apreciar en la Figura 4.2.

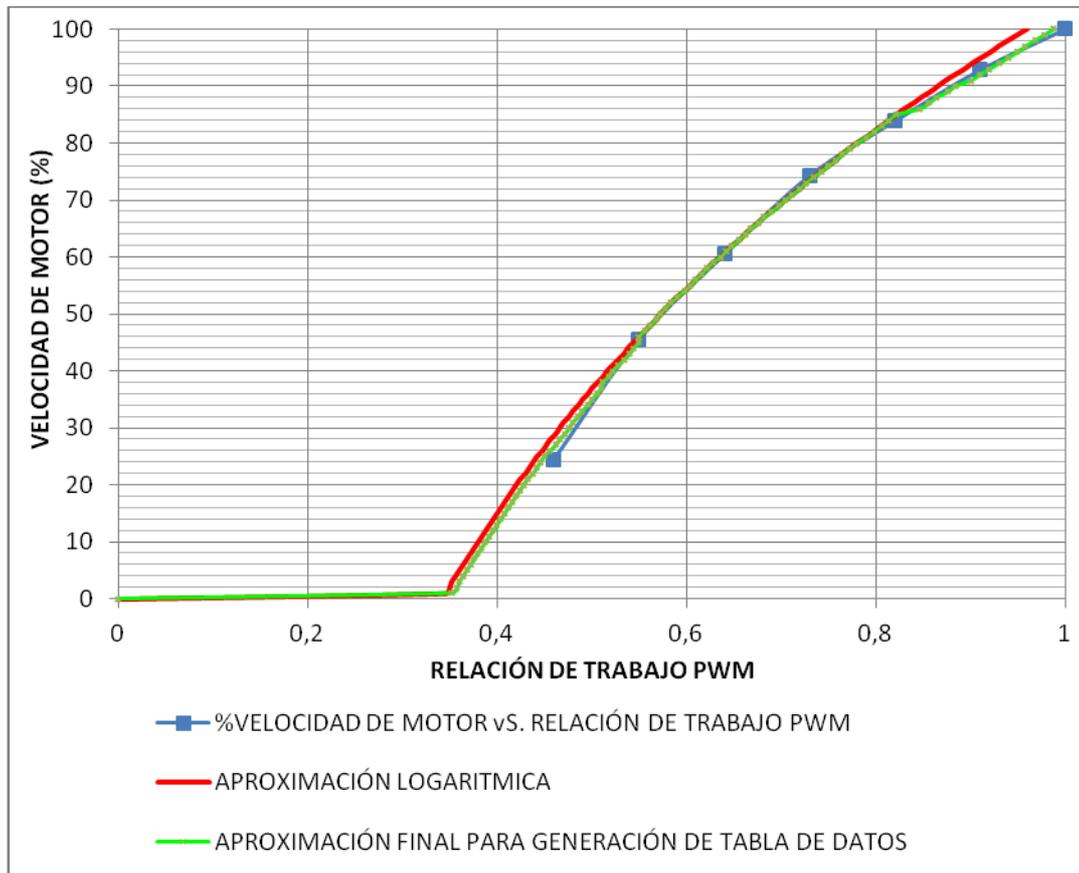


Figura 4. 2 Curvas de aproximación para linealización de velocidad de los motores

Haciendo uso de la ecuación $Rt_{(Vg)}$ de aproximación final, se generó una tabla de valores con la que el microcontrolador asignará las relaciones de trabajo PWM correctas y correspondientes al porcentaje de velocidad requerido; de acuerdo a los resultados obtenidos de los controladores de alcance y giro, logrando con esto la linealidad deseada.

Para comprobar el correcto funcionamiento de la tabla de valores, se replicó la prueba obteniendo la grafica que muestra la Figura 4.3.

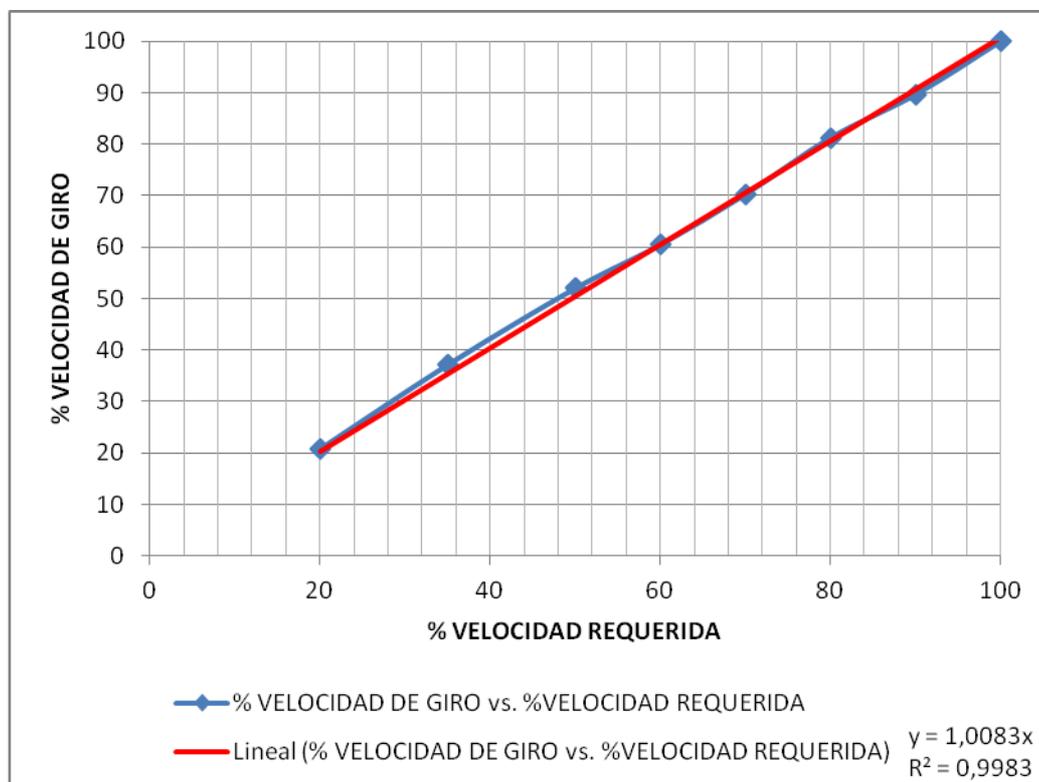


Figura 4. 3 Resultados de linealización

4.2. EXPLICACIÓN DEL SISTEMA DE COORDENADAS DEL PROTOTIPO

La gráfica de la Figura 4.4, muestra los sistemas de coordenadas X-Z involucrados en el funcionamiento del robot. Los ejes de color negro corresponden al sistema coordinado original del software de Kinect.

En el sistema coordenado de KINECT, para la dimensión en profundidad (eje Z), se usa metros, siendo el punto de origen el cuerpo del sensor. Sin embargo, existe una zona de 0 a 0,8m en la cual la confiabilidad de las medidas del Kinect es nula; es decir que aquella es una zona muerta.

En el sistema coordenado de KINECT, para la dimensión horizontal (eje X), se tiene una variación, sin unidades, de -1 a 1 entre los bordes del campo visual de Kinect, estando el cero en el centro de este campo. A pesar de no tener unidades físicas, se ha observado que los valores de 1 y -1 corresponden aproximadamente a 1m a la izquierda y derecha del centro de la visión de Kinect.

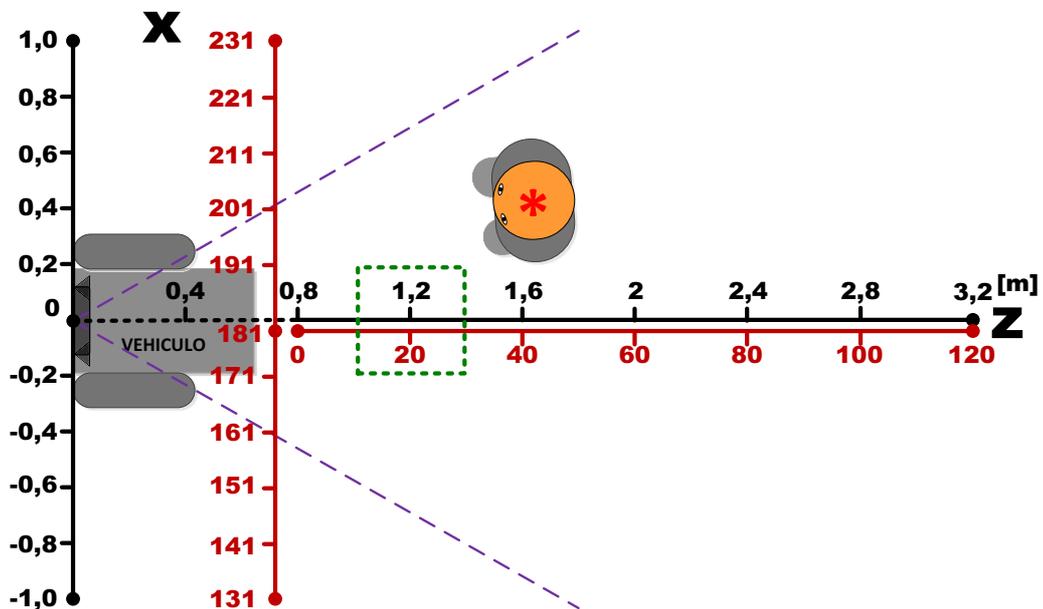


Figura 4. 4 Relación espacial entre los ejes del sensor Kinect y los ejes particulares del sistema del robot



Figura 4. 5 Captura de interfaz, con datos de posición del usuario, en los dos sistemas de coordenadas

Como se mencionó anteriormente, para una mayor facilidad en el manejo de estos datos, especialmente para su transmisión hacia el microprocesador, se ha optado por, a partir del sistema coordenado original, establecer otro con valores y rangos particulares para el sistema del robot acompañante. En la Figura 4.4, los ejes de color rojo corresponden a este sistema coordenado particular. Sus valores no presentan unidades físicas y en adelante serán referidos simplemente como “unidades”, tanto para la dimensión en profundidad como para la dimensión horizontal.

En cuanto al eje Z particular, el punto de origen es en cero unidades. Dicho valor se ubica a los 0,8 metros del sistema coordenado del Kinect, y se extiende hasta las 120 unidades, cuya posición corresponde a los 3,2m desde el sensor. Es decir que este eje empieza en donde termina la zona muerta del sensor.

Para el eje X particular, el punto de origen es en el número 181. Este valor es el que se ubica en el centro del campo visual, correspondiendo al cero del sistema coordenado de Kinect. A partir del número 181, el eje se extiende hacia la derecha hasta las 131 unidades y hacia la izquierda hasta las 231 unidades, coincidiendo respectivamente con el -1 y 1 del eje original.

La relevancia de estos ejes particulares radica en que sus valores son los recibidos por el microcontrolador para hacer los cálculos de errores de posición del usuario.

Además, para poder evaluar las pruebas que se le han realizado al robot, se ha adicionado a la aplicación de la computadora una programación que abre un archivo de texto donde va registrando periódicamente la posición X-Z del usuario. El registro de dicha posición se encuentra expresado en los valores del sistema coordinado particular, por lo que la relación entre sistemas coordinados de la Figura 4.4, sirve de referencia para la presentación de los datos obtenidos en las pruebas.

Como una información adicional, en la Figura 4.4, se muestra la ubicación y extensión de la región inicial en donde se debe ubicar el usuario para el inicio del seguimiento. En profundidad sus límites son 1 y 1,4 metros y en sentido horizontal está entre los valores -0,2 y 0,2, del sistema coordinado del KINECT.

4.3. PROCEDIMIENTO PARA DETERMINAR LOS CONTROLADORES DE GIRO Y DE AVANCE

Como se mencionó anteriormente, la dinámica del robot se ha dividido en 2 elementos de velocidad: la velocidad de giro y la velocidad de avance, cuya superposición configura la trayectoria de seguimiento que realiza el robot.

Cada componente de velocidad es determinado por su correspondiente controlador, el de giro y el de avance.

Para obtener los valores de diseño de estos controladores se han utilizado herramientas computacionales del software MATLAB. Primero, a partir de pruebas experimentales y con la herramienta *System Identification Tool*, se obtuvo funciones de transferencia en lazo abierto del avance y del giro del robot. Es decir que hizo la identificación del robot dividiéndolo en una *planta de avance* y una *planta de giro*. Luego, con la herramienta *SISO Design* se procedió a diseñar los compensadores para cada una de las plantas.

4.3.1. ADQUISICIÓN DE DATOS DE LA PLANTA DE AVANCE

La herramienta de identificación de Matlab requiere datos experimentales de una señal de entrada y datos de la señal de respuesta del sistema. Esta prueba se ha

configurado de tal manera que sea equivalente a la respuesta en lazo abierto de la planta de avance a una señal paso.

Primero, el usuario se ubicó a 50 unidades de profundidad del robot (aunque un poco desviado del centro para evitar ser atropellado). En 20 unidades es el punto de referencia en el cual se desea que se detenga el robot. Entonces al iniciar la prueba, como señal de entrada paso, al robot se le excitó con una velocidad constante de avance de 30%.

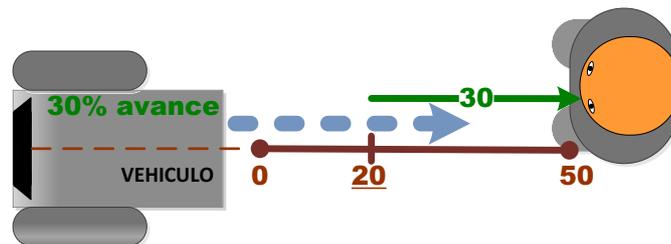


Figura 4. 6 Prueba para la planta de avance en lazo abierto

Es decir que en la prueba, el usuario se paró a una distancia inicial de 30 unidades del punto de referencia; luego el robot recibió una consigna constante de 30% de velocidad de avance, con lo cual empezó a acercarse hacia el usuario.

Para saber la respuesta a esta señal de consigna, durante el acercamiento del robot, se midió con el sensor Kinect el cambio de la posición en profundidad del usuario. Estos datos de posición (junto a su correspondiente tiempo) fueron registrados por programa en el archivo de texto para su posterior análisis.

Los datos adquiridos de esta manera han sido graficados en Figura 4.7.

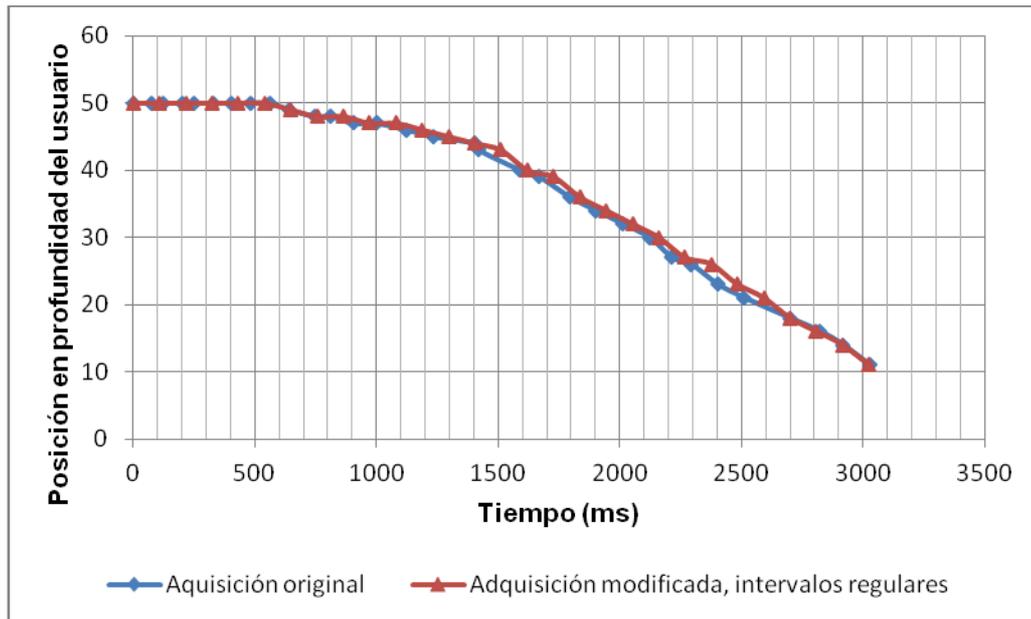


Figura 4. 7 Datos de posición en profundidad durante la prueba para la planta de avance en lazo abierto.

La curva muestra como la posición de profundidad del usuario se reduce progresivamente conforme se acerca el robot, hasta que a las 10 unidades el usuario sale de su campo de visión y la adquisición cesa.

Los marcadores azules corresponden a las muestras originales tal como fueron registradas. Esta adquisición no ocurre con un mismo periodo de muestro. Sin embargo, la herramienta de identificación requiere de datos muestreados con un periodo constante. Para satisfacer este requerimiento, a partir la adquisición original se extrajo un periodo de muestreo medio. En base a dicho periodo se han redistribuido cuidadosamente las muestras para tener una curva modificada similar a la original, pero con muestras a intervalos regulares, la cual está dibujada en la gráfica con marcadores rojos. Por ejemplo, el periodo medio para la adquisición mostrada de la Figura 4.7, es de 108ms.

La curva obtenida hasta este momento corresponde a la posición en profundidad del usuario, la cual es decreciente; pero para proseguir con la identificación, la curva debe interpretarse como una respuesta de desplazamiento creciente frontal del robot. Para ello se hizo una resta entre la posición inicial del usuario (de 50 unidades de profundidad) y los valores de la curva. La gráfica de esta interpretación se muestra en la Figura 4.8, con marcadores rojos.

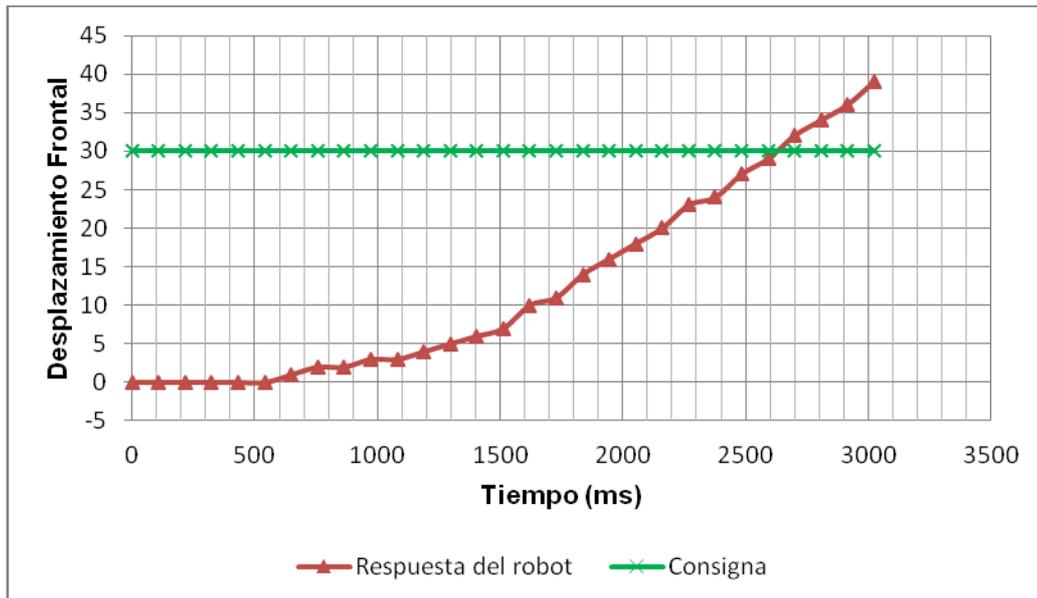


Figura 4. 8 Interpretación de la prueba de profundidad como prueba de desplazamiento frontal del robot a lazo abierto

En definitiva, la Figura 4.8, muestra en verde una señal paso de consigna que le ha ordenado al robot desplazarse frontalmente 30 unidades. En rojo se ve la respuesta de desplazamiento frontal del robot. Al estar en lazo abierto, el desplazamiento simplemente siguió creciendo sin estabilizarse, hasta que la adquisición se interrumpió. Esta prueba en lazo abierto ha tenido una duración de aproximadamente 3 segundos.

Los datos de entrada y salida graficados en la Figura 4.8 son los que se utilizan para la identificación de la planta de avance, ingresándolos como vector de entrada y vector de salida.

4.3.2. IDENTIFICACIÓN DE LA PLANTA DE AVANCE

Tras iniciar la herramienta *System Identification Tool* de MATLAB, se han introducido los vectores de entrada y de salida de dos repeticiones de la prueba de desplazamiento frontal en lazo abierto; se introdujo además el tiempo de muestreo medio anteriormente determinado. Como se puede apreciar, las curvas elaboradas por la herramienta y mostradas en la Figura 4.9, se asemejan a las presentadas en la Figura 4.8.

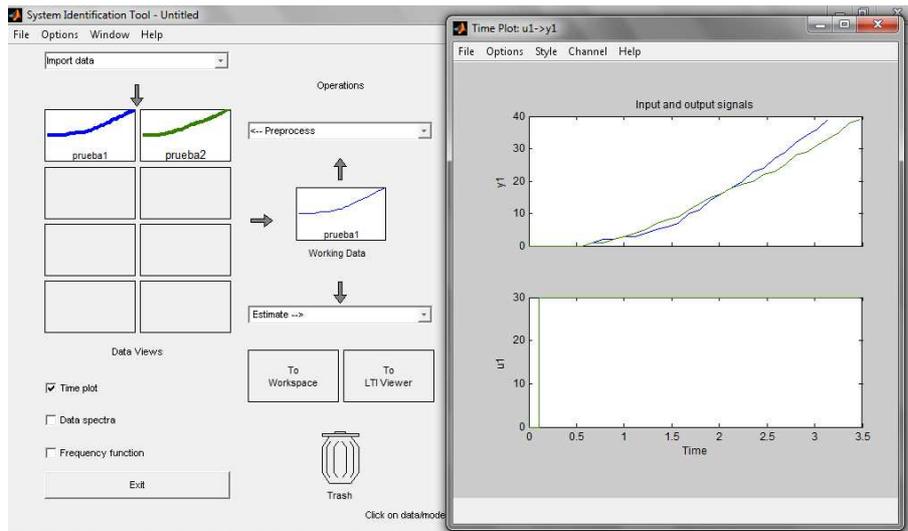


Figura 4. 9 Datos de entrada y salida de la prueba de desplazamiento frontal introducidos al *System Identification Tool*

Para obtener los modelos en sí, primero en la herramienta se elige la forma que se desea que tenga la función de transferencia del modelo a generar. Como la modelación es la base para el posterior diseño del controlador de avance, se debe elegir una forma que se ajuste a los datos, y que a su vez sea bastante sencilla. La forma con dos polos, con uno de ellos en el origen (integrador) resultó la más apropiada.

La herramienta generó modelos en base a cada set de datos introducidos. Como se aprecia en la Figura 4.10, la herramienta además mostró que los modelos se comportan de manera muy similar a los datos experimentales tanto para la señal de entrada ingresada como para una señal paso genérica.

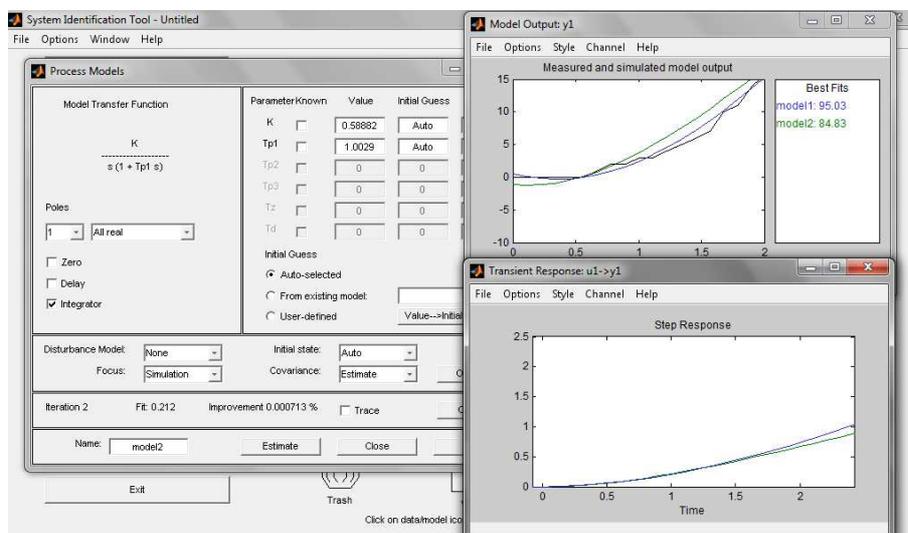


Figura 4. 10 Generación de modelos para la planta de avance en el *System Identification Tool*

Para observar las funciones de transferencia de los modelos obtenidos se los exportó al Workspace de Matlab (Figura 4.11).

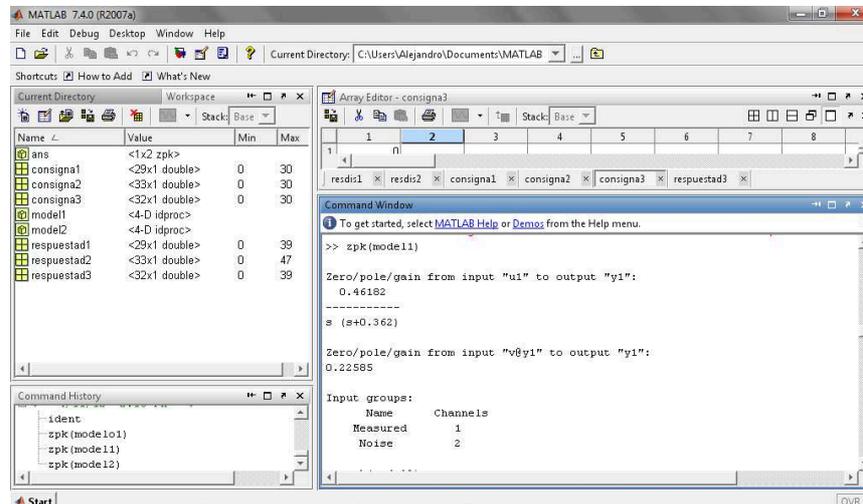


Figura 4. 11 Modelo exportado al *Workspace* y visualizado en forma de polos y ceros

En definitiva las funciones de transferencia generadas para cada repetición de la prueba de desplazamiento frontal fueron: $\frac{0,46182}{s(s+0,362)}$ y $\frac{0,38758}{s(s+0,4434)}$. En lugar de elegir

una de ellas se decidió probar un modelo simplificado similar a los dos obtenidos. El modelo simplificado fue: $\frac{0,4}{s(s+0,4)}$. Se procedió a exportar este modelo

simplificado a la herramienta de identificación para compararlo, y como se aprecia en la Figura 4.12, resultó que también se ajusta razonablemente a los datos experimentales y a los modelos generados, por lo que se tomó dicho modelo simplificado como modelo definitivo de la planta de avance.

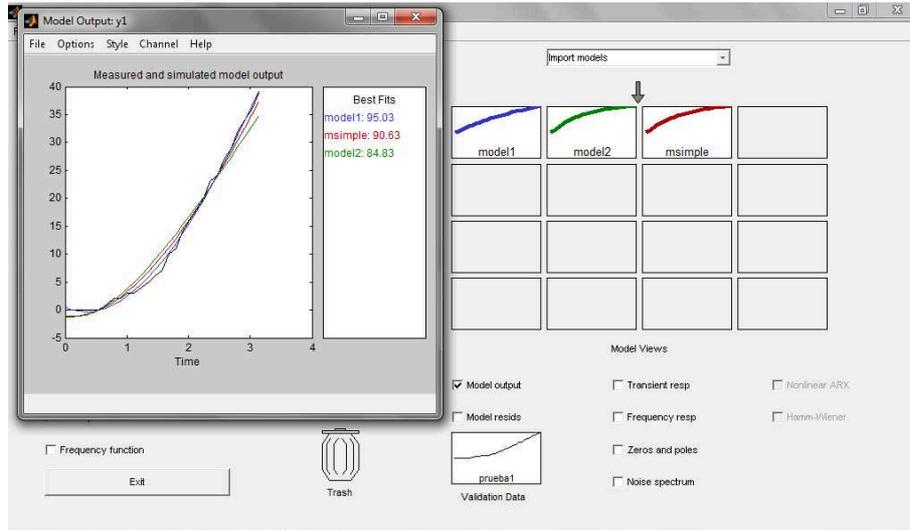


Figura 4. 12 Comparación y confirmación del modelo elegido para la planta de avance

El modelo obtenido es de tiempo continuo; puesto que el sistema de control es digital, a este modelo es necesario discretizarlo. El modelo discretizado correspondiente es: $\frac{0,001974(z+0,9868)}{(z-1)(z-0,9608)}$, con un periodo de muestreo estimado para el sistema del robot de 100ms.

4.3.3. ADQUISICIÓN DE DATOS DE LA PLANTA DE GIRO

De igual manera que con la planta de avance, la prueba para la planta de giro se ha configurado de tal manera que sea equivalente a la respuesta en lazo abierto de esta planta a una señal paso.

Para la versión de giro izquierdo de la prueba, primero el usuario se ubicó en la posición horizontal 212. La posición 181 corresponde a la posición central de referencia. Entonces, al iniciar la prueba, como señal de entrada paso al robot se le excitó con una velocidad constante de giro de 31%.

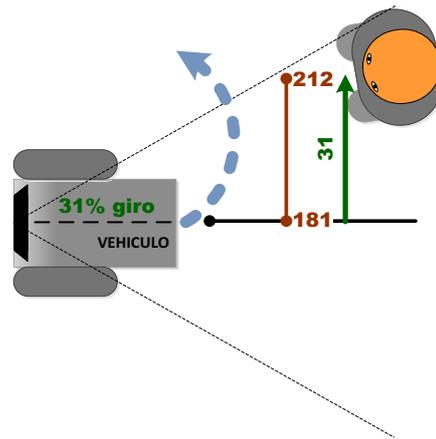


Figura 4. 13 Prueba para la planta de giro en lazo abierto, hacia la izquierda

Es decir que en esta versión de la prueba, el usuario se paró en una posición horizontal a 31 unidades de la posición central (a la izquierda) y luego el robot recibió una consigna constante de 31% de velocidad de giro, con lo cual empezó a girar.

Durante la prueba se midió con el sensor Kinect el cambio de la posición horizontal del usuario. Estos datos, junto a su correspondiente tiempo fueron registrados en el archivo de texto para su análisis posterior.

Luego, para la versión de giro derecho de la prueba, el usuario se ubicó en la posición horizontal 150. Entonces, al iniciar la prueba, como señal de entrada paso al robot se le excitó con una velocidad constante de giro de -31% (negativa).

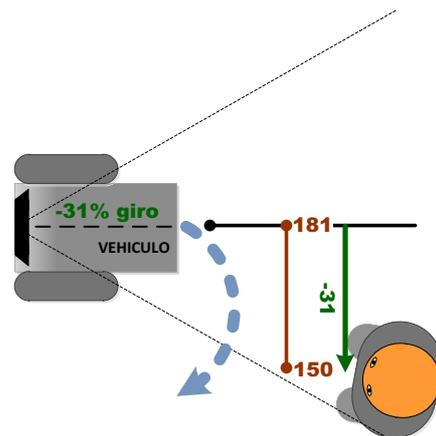


Figura 4. 14 Prueba para la planta de giro en lazo abierto, hacia la derecha

Es decir que en esta otra versión de la prueba, el usuario se paró en una posición horizontal a -31 unidades de la posición central (a la derecha) y luego el robot

recibió una consigna constante de -31% de velocidad de giro, con lo cual empezó a girar.

Durante la prueba se midió con el sensor Kinect el cambio de la posición horizontal del usuario. Estos datos, junto a su correspondiente tiempo se iban registrando en el archivo de texto para su análisis posterior.

Los datos adquiridos de esta manera, para las dos versiones de la prueba, han sido graficados en la Figura 4.15.

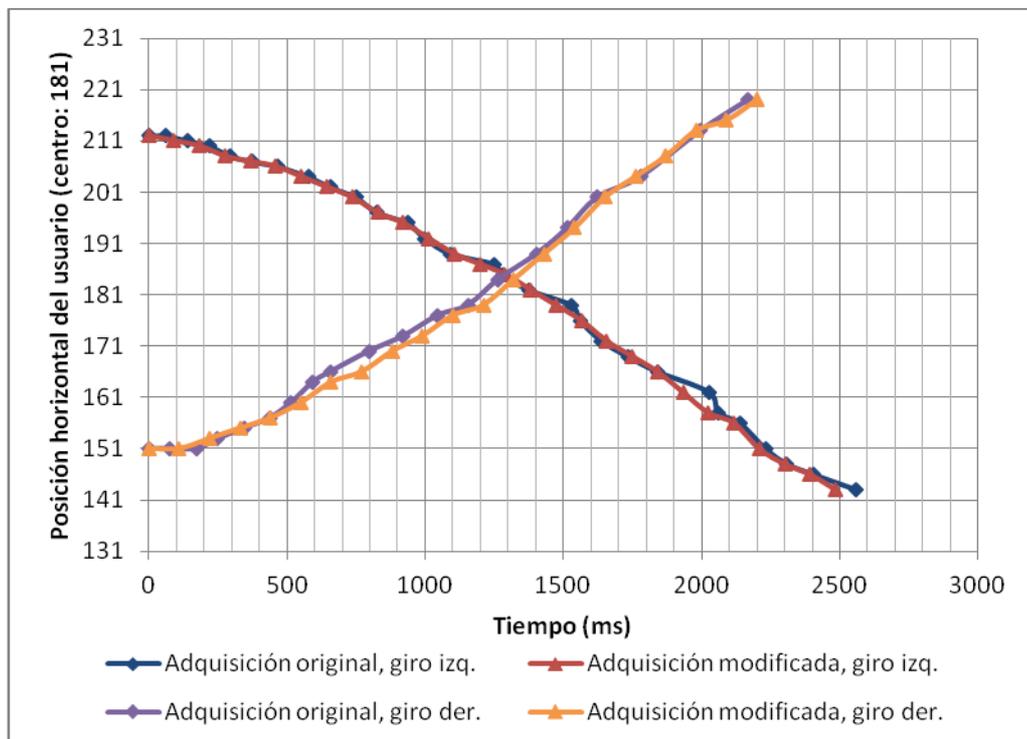


Figura 4. 15 Datos de posición horizontal durante las pruebas para la planta de giro en lazo abierto

Las curvas muestran como, durante el giro izquierdo, la posición horizontal del usuario se reduce progresivamente conforme gira el robot; mientras que, durante el giro derecho, la posición horizontal del usuario aumenta progresivamente. En ambos casos, la adquisición pasa por la posición central, pero como se encuentra en lazo abierto, el robot sigue girando hasta que el usuario sale del campo visual y la adquisición cesa.

Los marcadores azules y violetas corresponden a las muestras originales tal como fueron registradas. Estas adquisiciones no ocurren con un mismo periodo, pero la

herramienta de identificación requiere un periodo de muestreo constante. Por eso, a partir de las adquisiciones originales se extrajeron periodos de muestreo medios, en base a los cuales se han redistribuido las muestras para tener curvas modificadas (de color rojo y naranja) con intervalos regulares. El periodo medio para la adquisición de giro izquierdo fue 92ms y para la de giro derecho fue de 110ms.

Las curvas de la Figura 4.15 describen la posición horizontal del usuario durante las pruebas; pero para proseguir con la identificación, las curvas deben reinterpretarse como respuestas de desplazamiento creciente lateral del robot, desde cero y despreciando el sentido de giro. Para ello se hizo una resta entre las posiciones horizontales iniciales del usuario y los valores de sus correspondientes curvas. Las gráficas de estas interpretaciones se muestran en la Figura 4.16.

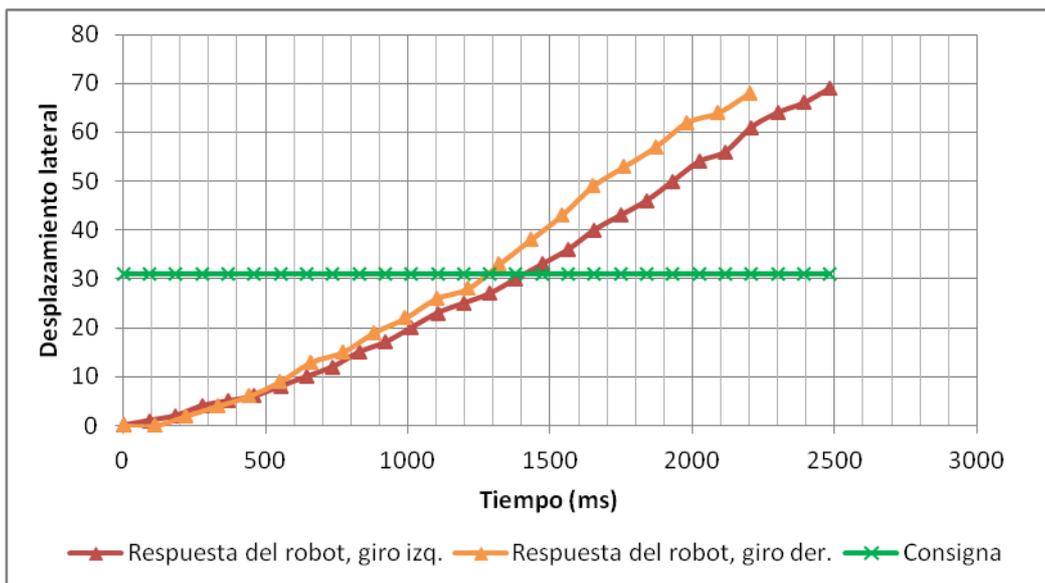


Figura 4. 16 Interpretación de la prueba de posición horizontal como prueba de desplazamiento lateral del robot a lazo abierto

En definitiva, la Figura 4.16 muestra en verde una señal paso de consigna que le ha ordenado al robot desplazarse lateralmente 31 unidades. En rojo y naranja se ven respuestas de desplazamiento lateral del robot. Al estar en lazo abierto, el desplazamiento simplemente siguió creciendo sin estabilizarse, hasta que la adquisición se interrumpió. Estas pruebas en lazo abierto han tenido una duración de aproximadamente 2,3 segundos.

Los datos de entrada y salida graficados en la Figura 4.16 son los que se utilizan para la identificación de la planta de giro, ingresándolos como vectores de entrada y vectores de salida.

4.3.4. IDENTIFICACIÓN DE LA PLANTA DE GIRO

Tras iniciar la herramienta *System Identification Tool*, de MATLAB, se introdujeron los vectores de entrada y de salida de las dos repeticiones de la prueba de desplazamiento lateral en lazo abierto, indicando los tiempos de muestreo medio. Como se puede apreciar, las curvas elaboradas por la herramienta y mostradas en la Figura 4.17 se asemejan a las presentadas en la Figura 4.16.

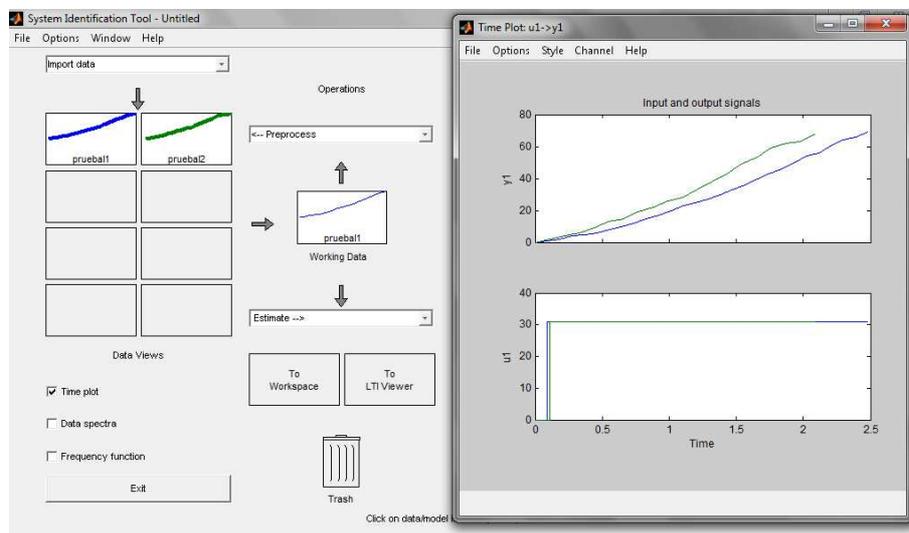


Figura 4. 17 Datos de entrada y salida de la prueba de desplazamiento lateral introducidos al *System Identification Tool*

Para obtener los modelos en sí, primero en la herramienta se elige la forma que se desea que tenga la función de transferencia del modelo a generar. Al igual que con la planta de avance, la forma con dos polos, con uno de ellos en el origen (integrador), resultó la más apropiada.

La herramienta generó entonces modelos en base a cada conjunto de datos introducidos. Como se aprecia en la Figura 4.18, la herramienta muestra que los modelos se comportan de manera muy similar a los datos experimentales, tanto para la señal de entrada ingresada como para una señal paso genérica.

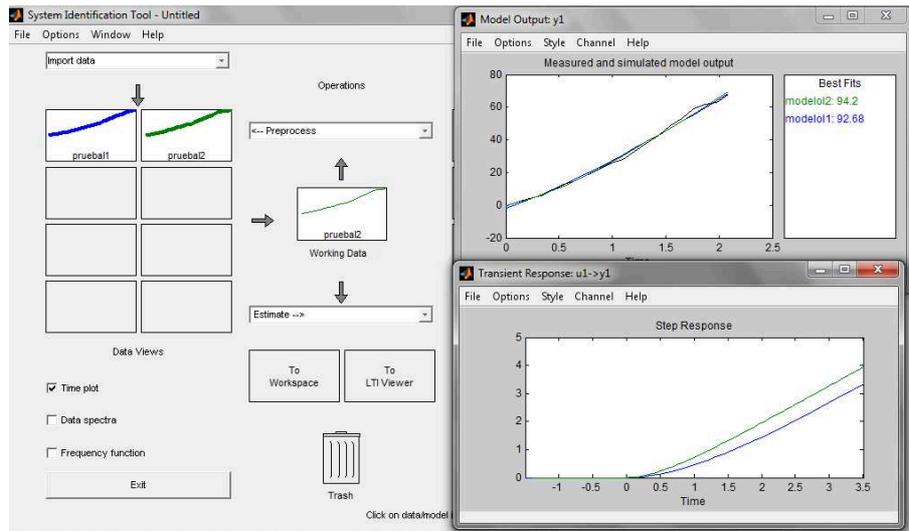


Figura 4. 18 Generación de modelos para la planta de giro en el *System Identification Tool*

Para observar las funciones de transferencia de los modelos obtenidos se los exportó al Workspace de Matlab (Figura 4.19).

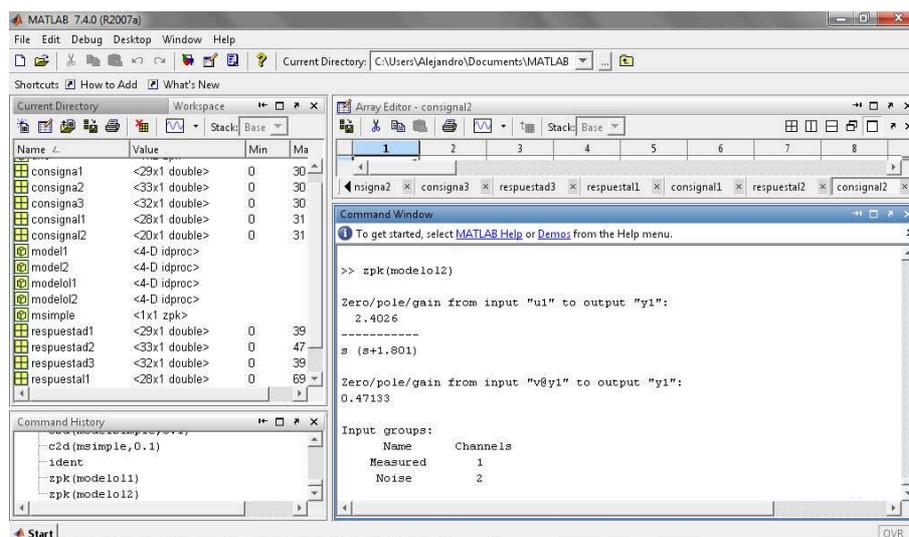


Figura 4. 19 Modelo exportado al *Workspace* y visualizado en forma de polos y ceros

En definitiva las funciones de transferencia generadas para cada repetición de la prueba de desplazamiento lateral fueron: $\frac{2,1691}{s(s+0,9333)}$ y $\frac{2,4026}{s(s+1,801)}$. En lugar de elegir

una de ellas, se decidió probar un modelo simplificado similar a los obtenidos. El modelo simplificado fue: $\frac{2,3}{s(s+1,7)}$. Se procedió a exportar este modelo simplificado a

la herramienta de identificación para compararlo. Como se aprecia en la Figura

4.20, resultó que sí se ajusta razonablemente a los datos experimentales y a los modelos generados; por esta razón se tomó dicho modelo simplificado como modelo definitivo de la planta de giro.

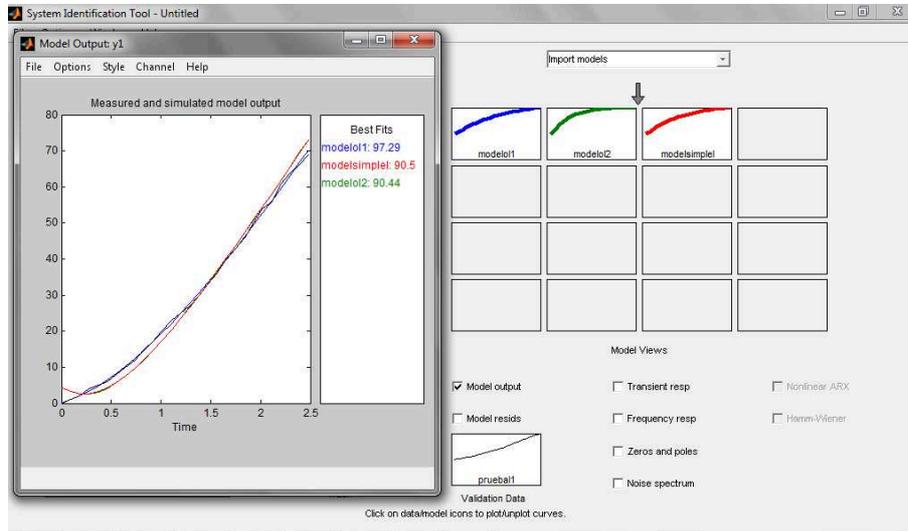


Figura 4. 20 Comparación y confirmación del modelo elegido para la planta de giro

Al igual que sucedió con la planta de avance, este modelo obtenido para la planta de giro es de tiempo continuo; puesto que el sistema de control es digital, a este modelo es necesario discretizarlo. El modelo discretizado correspondiente es: $\frac{0,010875(z+0,9449)}{(z-1)(z-0,8437)}$, con un periodo de muestreo estimado para el sistema del robot de 100ms.

4.4. DISEÑO DE COMPENSADORES

Una vez obtenidos los modelos discretos para la planta de giro y la planta de avance, se procedió a hacer uso de la herramienta SISO Design de MATLAB, que permite diseñar compensadores en base a los modelos obtenidos.

4.4.1. COMPENSADOR PARA GIRO

Para la planta de giro, se optó por un compensador tipo red de adelanto debido a que al graficar el lugar geométrico de las raíces de la planta (Figura 4.21) se nota que este se encuentra al límite de la inestabilidad y es necesario traerlo hacia dentro del círculo unitario. Dicho efecto es logrado gracias a que el cero que

posee este tipo de compensador es dominante. “La compensación en adelanto acelera la respuesta e incrementa la estabilidad del sistema” [44].

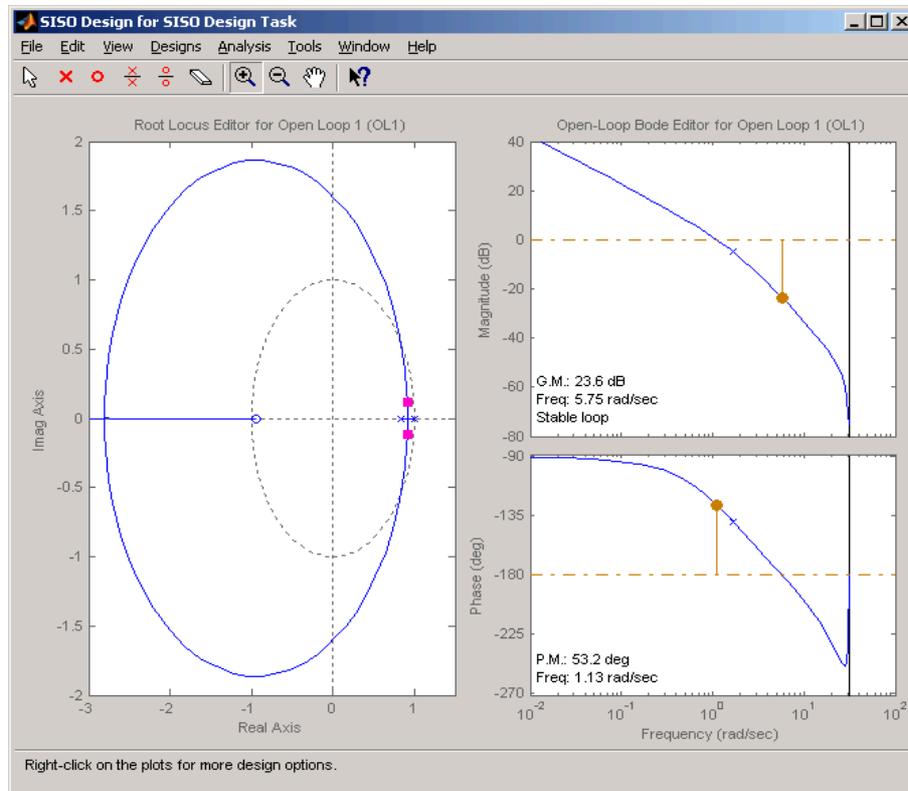


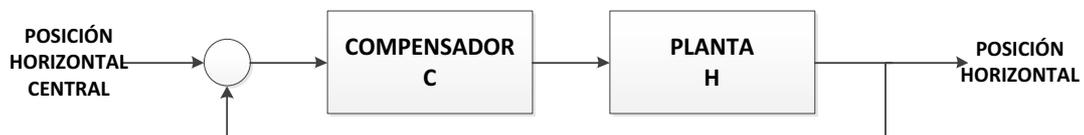
Figura 4. 21 LGR y diagrama de Bode de la planta de giro en lazo abierto

Las especificaciones estimadas para el óptimo funcionamiento del prototipo son:

Sobrepaso máximo: $M_p < 15\%$

Tiempo de crecimiento: $t_r < 2s$

Con la topología:



El compensador inicial que la herramienta SISO Design sugiere al colocar las restricciones de diseño, fue manipulado hasta que se cumplan las especificaciones, obteniendo el modelo:

$$C = \frac{12(z - 0,85)}{(z - 0,1)}$$

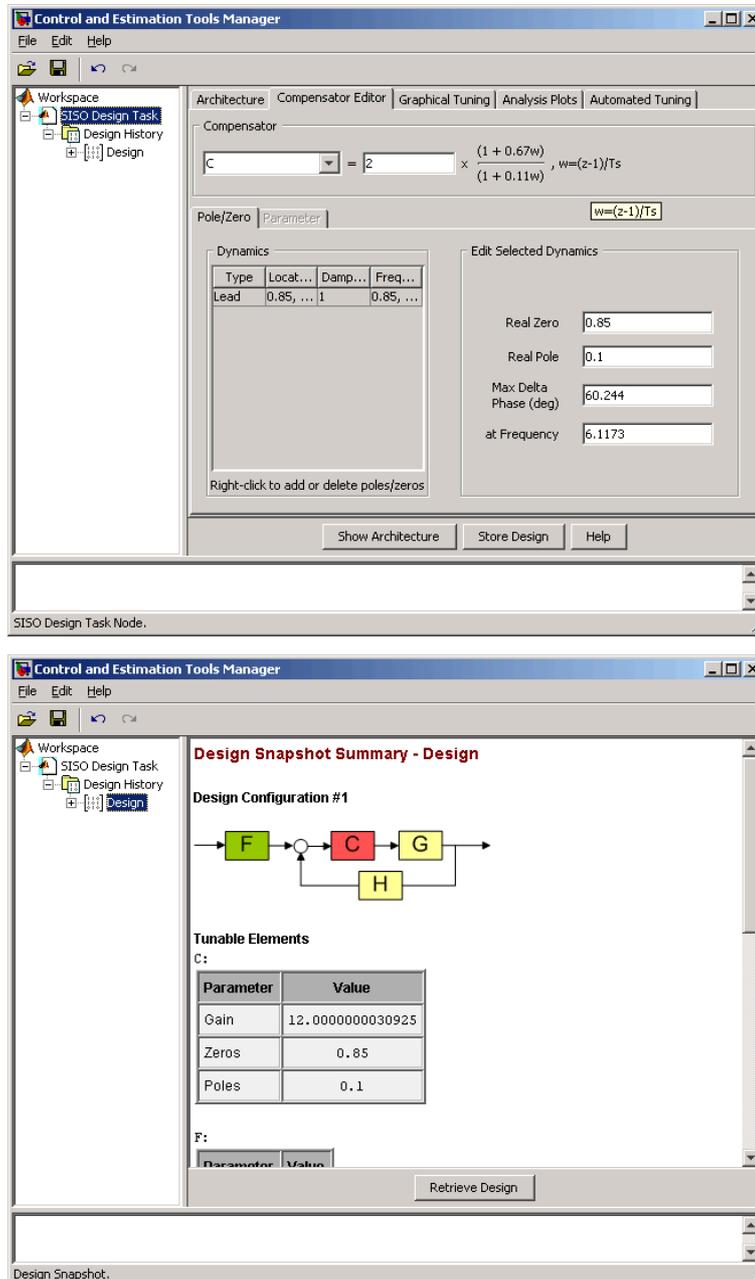
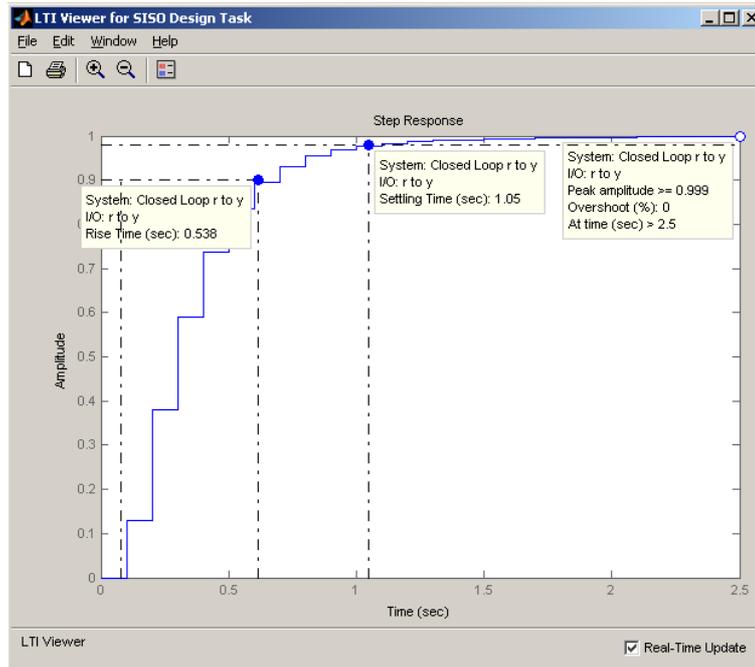
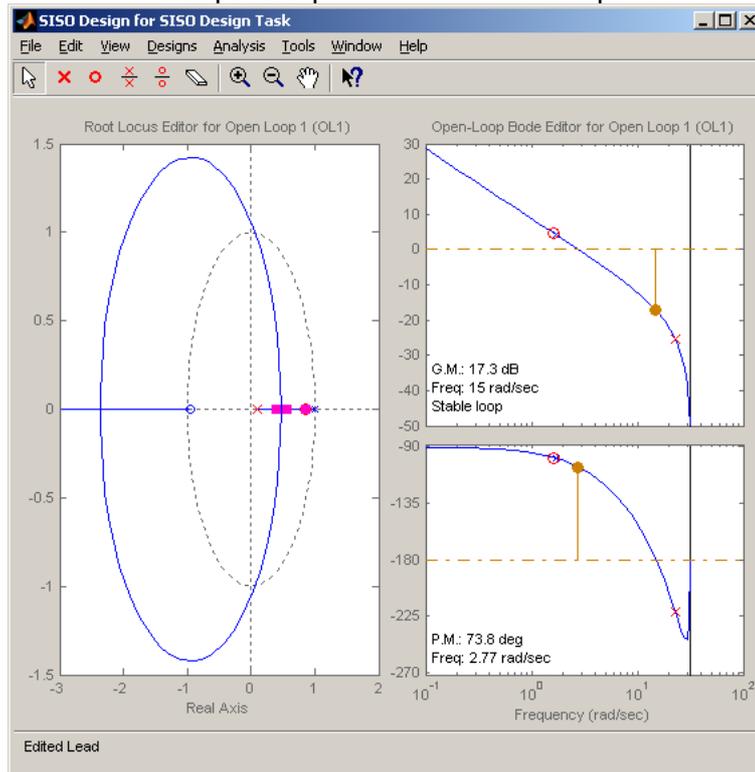


Figura 4. 22 Datos finales del compensador

La Figura 4.23 muestra el comportamiento del sistema compensado luego de haber introducido el controlador manipulado.



Respuesta paso del sistema compensado



LGR y diagramas de BODE del sistema compensado

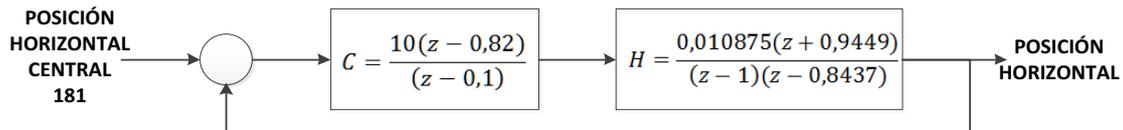
Figura 4. 23 Gráficas de respuesta con compensador manipulado

El compensador obtenido mediante la herramienta de calibración fue ingresado como ley de control en el microcontrolador del prototipo y para su correcto

funcionamiento se realizó re-calibraciones de acuerdo al comportamiento real, usando el método prueba-error. El óptimo funcionamiento del prototipo se obtuvo con el siguiente modelo de compensador:

$$C = \frac{10(z - 0,82)}{(z - 0,1)}$$

Como modelo final del sistema compensado se tiene:



4.4.2. COMPENSADOR DE ALCANCE

Para la planta de alcance, se optó por un compensador tipo red de adelanto debido a que al graficar el lugar geométrico de las raíces de la planta (Figura 4.24) se nota que este se encuentra al límite de la inestabilidad y es necesario traerlo hacia dentro del círculo unitario. Dicho efecto es logrado gracias a que el cero que posee este tipo de compensador es dominante. “La compensación en adelanto acelera la respuesta e incrementa la estabilidad del sistema” [44].

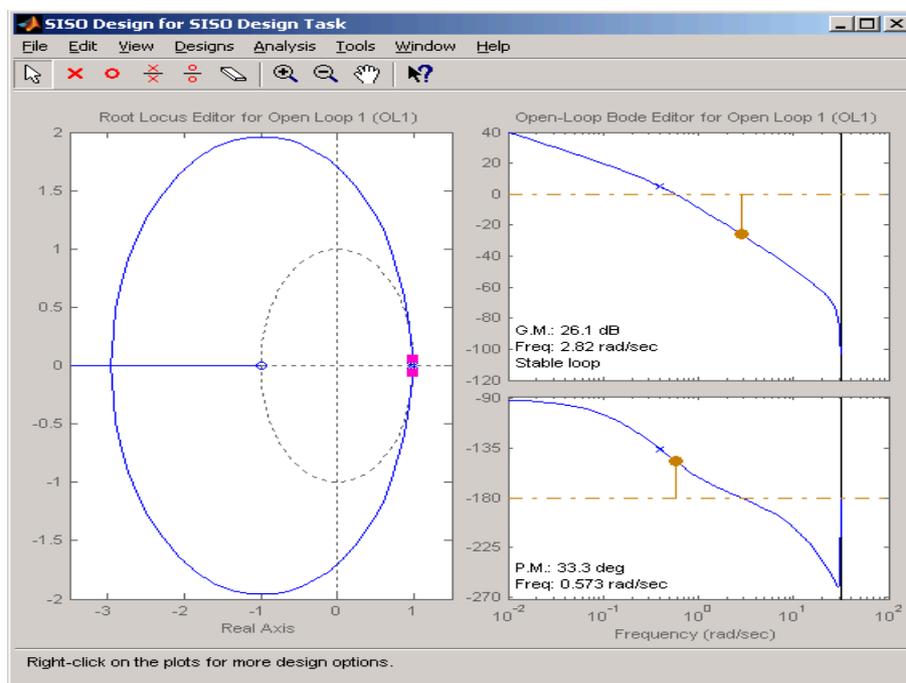


Figura 4. 24 LGR y diagrama de Bode de la planta de alcance en lazo abierto

Las especificaciones estimadas para el óptimo funcionamiento del prototipo son:

Sobrepaso máximo: $M_p < 15\%$

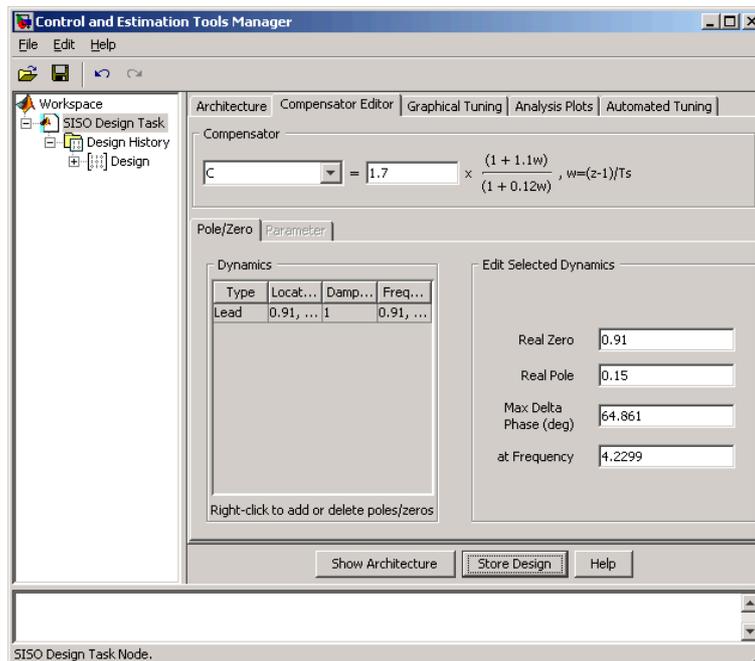
Tiempo de crecimiento: $t_r < 2s$

Con la topología:

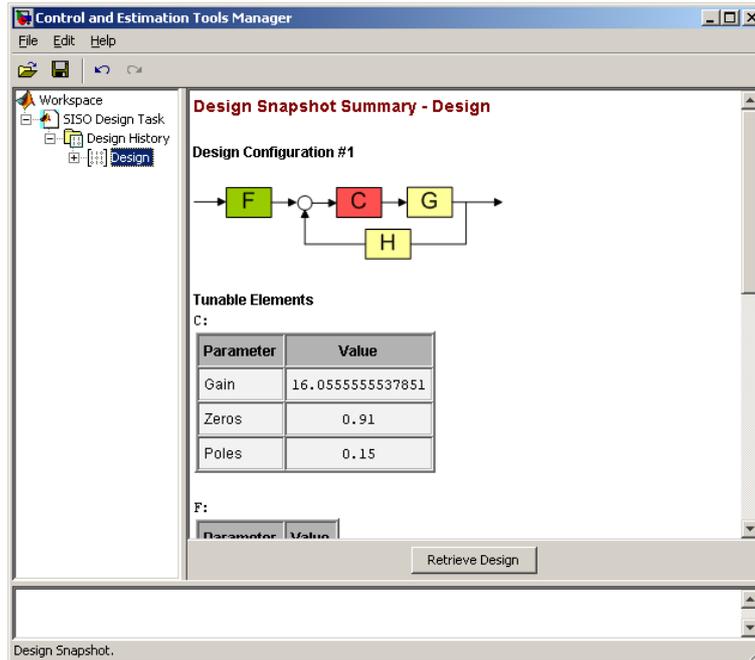


El compensador inicial que la herramienta SISO Design sugiere al colocar las restricciones de diseño fue manipulado hasta que se cumplan las especificaciones, obteniendo el modelo:

$$C = \frac{16,05(z - 0,91)}{(z - 0,15)}$$



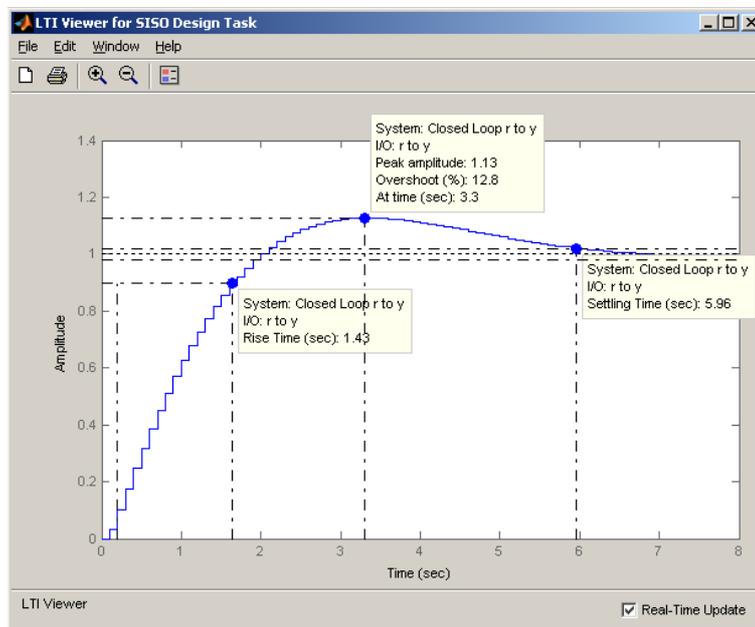
(a)



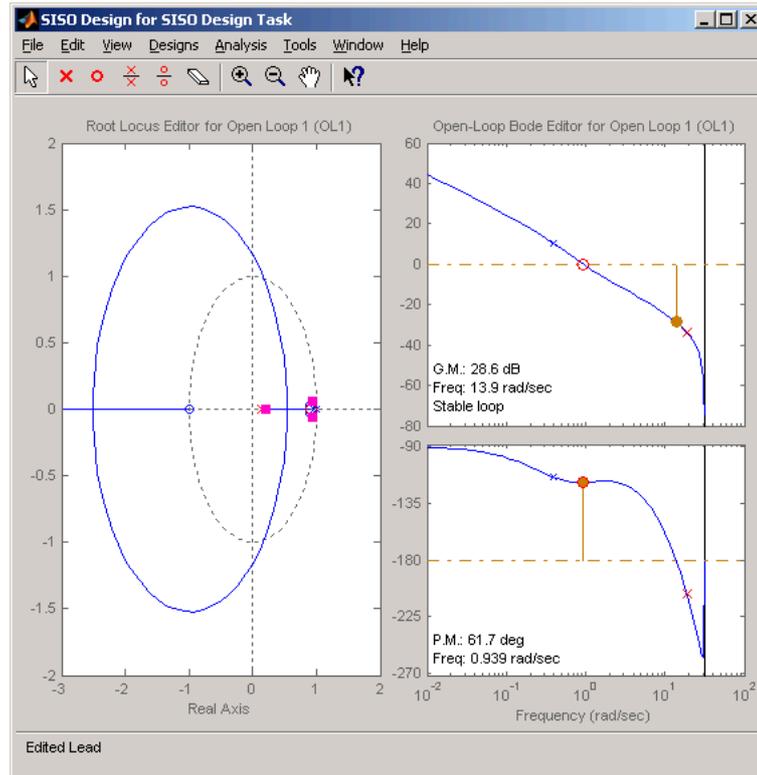
(b)

Figura 4. 25 Datos finales del compensador

La Figura 4.26, muestra el comportamiento del sistema compensado luego de haber introducido el controlador calibrado.



(a) Respuesta paso del sistema compensado



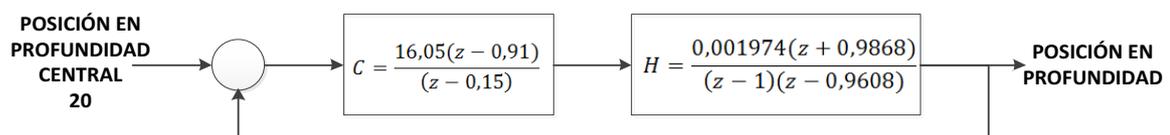
(b) LGR y diagramas de BODE, del sistema compensado

Figura 4. 26 Gráficas de respuesta con compensador manipulado

El compensador obtenido mediante la herramienta de calibración fue ingresado como ley de control en el microcontrolador del prototipo y para su correcto funcionamiento se realizó re-calibraciones de acuerdo al comportamiento real, usando el método prueba-error. El óptimo funcionamiento del prototipo se obtuvo con el siguiente modelo de compensador:

$$C = \frac{16,05(z - 0,91)}{(z - 0,15)}$$

Como modelo final del sistema compensado se tiene:



4.5. PRUEBAS DE DESEMPEÑO

Para comprobar el desempeño del prototipo se lo sometió a tres pruebas:

1) Trayectoria recta, con las siguientes características:

- 300 cm.
- Superficie horizontal.

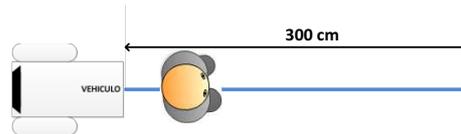


Figura 4. 27 Prueba de trayectoria recta

2) Trayectoria curva hacia la derecha, con las siguientes características:

- Recta 160 cm.
- Curva 200 cm.
- Superficie horizontal.

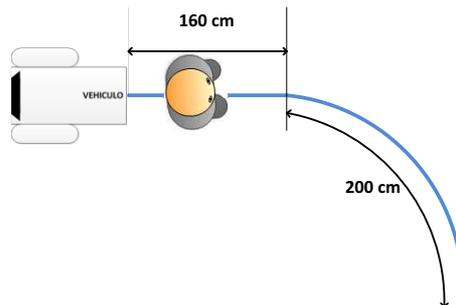


Figura 4. 28 Prueba de trayectoria curva a la derecha

3) Trayectoria curva hacia la izquierda, con las siguientes características:

- Recta 160 cm.
- Curva 200 cm.
- Superficie horizontal.

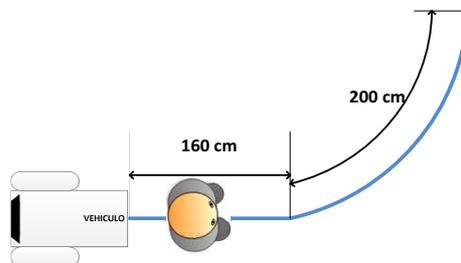


Figura 4. 29 Prueba de trayectoria curva a la izquierda

Las pruebas fueron desarrolladas siguiendo este procedimiento:

1. Ubicación del prototipo, alineado con el centro de la trayectoria marcada.

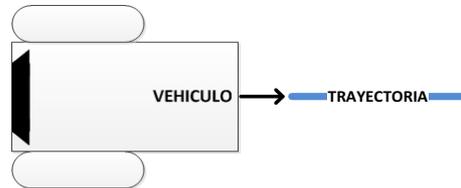


Figura 4. 30 Alineación del prototipo con la trayectoria

2. Ubicación del usuario en posición central para inicio de seguimiento.

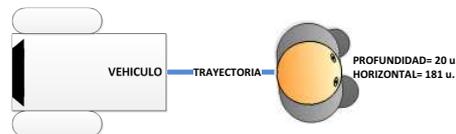


Figura 4. 31 Ubicación del usuario para iniciar prueba

3. Caminata a ritmo moderado del usuario sobre la trayectoria marcada.

En el transcurso de cada prueba se creó un archivo de texto con las posiciones del usuario respecto al sensor, tanto de alcance (profundidad) como de giro (horizontal).

A partir de los datos se construyeron gráficas que muestran el desempeño del prototipo mientras realizaba el seguimiento del usuario.

4.5.1. PRUEBA EN TRAYECTORIA RECTA



Figura 4. 32 Imágenes de prueba en trayectoria recta

En la Figura 4.33 se puede ver los cambios de posición en profundidad del usuario respecto al sensor KINECT; hay que tener en cuenta que la posición de referencia para el alcance es de 20u, por lo que el prototipo responde retornando a esta posición cuando existen cambios en la posición del usuario.

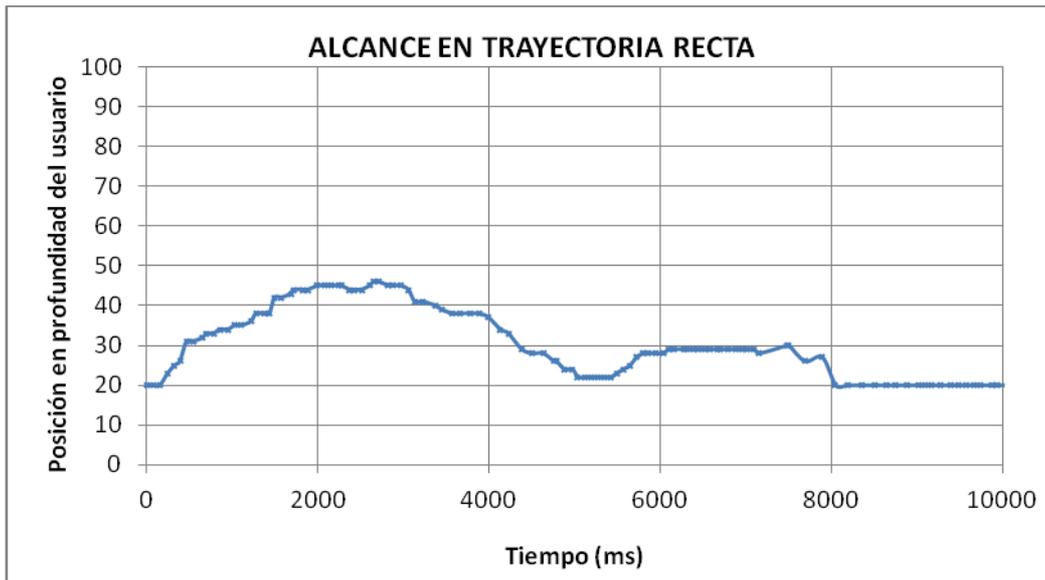


Figura 4. 33 Comportamiento en alcance del prototipo, durante seguimiento en trayectoria recta

En la Figura 4.34 se puede ver los cambios de posición horizontal del usuario respecto al sensor KINECT; hay que tener en cuenta que la posición de referencia para el giro es de 181u, por lo que el prototipo responde retornando a esta posición cuando existen cambios en la posición del usuario. Cabe recalcar que a pesar de tratarse de una trayectoria recta, el sensor detecta cambios ligeros de la posición horizontal que son compensados exitosamente.

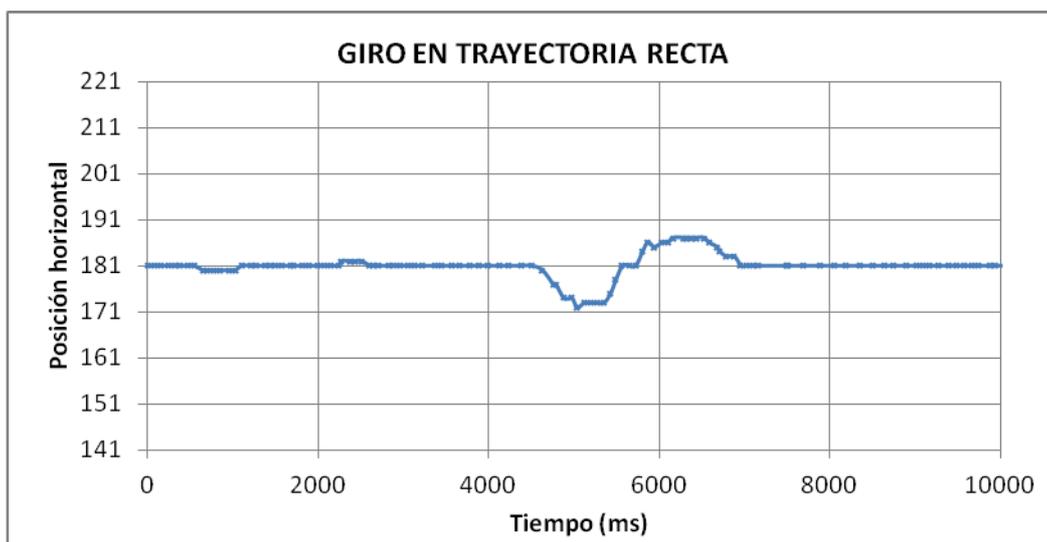


Figura 4. 34 Comportamiento en giro del prototipo, durante seguimiento en trayectoria recta

4.5.2. PRUEBA EN TRAYECTORIAS CURVAS HACIA LA DERECHA E IZQUIERDA



Figura 4. 35 Imágenes de prueba en trayectoria curva

En la Figura 4.36 y Figura 4.37, se puede ver los cambios de posición en profundidad del usuario respecto al sensor KINECT. Hay que tener en cuenta que la posición de referencia para el alcance es de 20u, por lo que el prototipo responde retornando a esta posición cuando existen cambios en la posición del usuario.

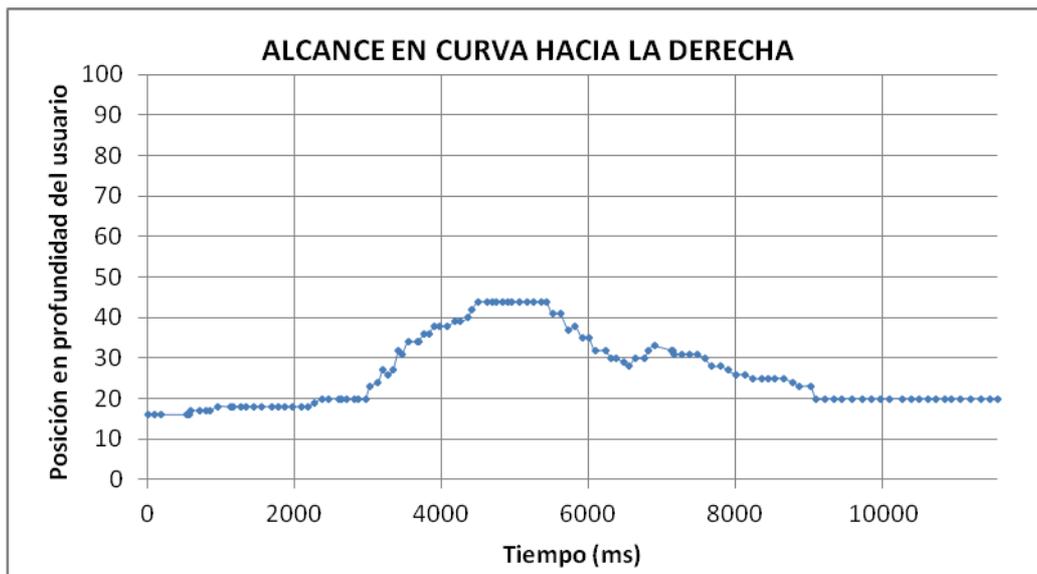


Figura 4. 36 Comportamiento en alcance del prototipo, durante seguimiento en trayectoria curva a la derecha

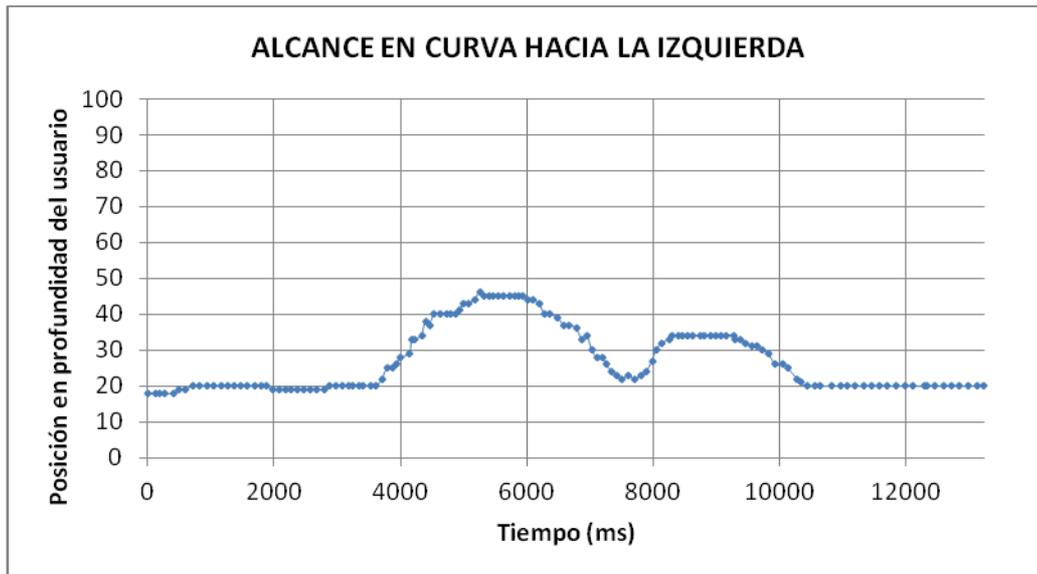


Figura 4. 37 Comportamiento en alcance del prototipo, durante seguimiento en trayectoria curva a la izquierda

En la Figura 4.38 y Figura 4.39, se puede ver los cambios de posición horizontal del usuario respecto al sensor KINECT; hay que tener en cuenta que la posición de referencia para el giro es de 181u, por lo que el prototipo responde retornando a esta posición cuando existen cambios en la posición del usuario.



Figura 4. 38 Comportamiento en giro del prototipo, durante seguimiento en trayectoria curva a la derecha

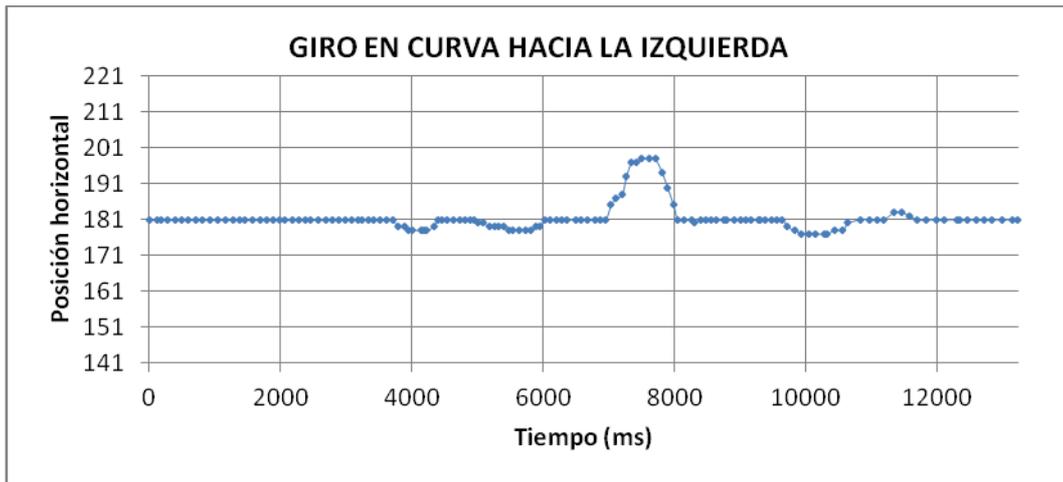


Figura 4. 39 Comportamiento en giro del prototipo, durante seguimiento en trayectoria curva a la izquierda

En este tipo de trayectorias hay que notar que tanto la posición de alcance como la de giro varían simultáneamente, siendo los desplazamientos de alcance menores a los desplazamientos en giro. Esto motiva que el prototipo priorice la compensación horizontal a la de alcance, traduciéndose en que el prototipo siga una trayectoria secante (Figura 4.40) a la curva descrita por el usuario.



Figura 4. 40 Trayectoria secante a la curva descrita por el prototipo

4.5.3. PRUEBAS DE FUNCIONAMIENTO GLOBAL EN TRAYECTORIAS NO CONTROLADAS DE CORTA DURACIÓN

Luego de apreciar un comportamiento de seguimiento razonablemente bueno en laboratorio, se sometió al prototipo a pruebas en ambientes más amplios, no

controlados; específicamente en el hall de la planta baja del edificio de Eléctrica-Química de la Escuela Politécnica Nacional.



Figura 4. 41 Ejemplo de seguimiento en una trayectoria curva derecha



Figura 4. 42 Ejemplo de seguimiento en una trayectoria curva izquierda



Figura 4. 43 Ejemplo de seguimiento en una trayectoria recta

Este tipo de pruebas se repitió un gran número de veces, haciendo varios ajustes tanto mecánicos como de software, especialmente en las constantes de los compensadores, hasta alcanzar los valores mostrados en el punto 4.4.

4.5.4. PRUEBAS DE FUNCIONAMIENTO GLOBAL EN AMBIENTES REALES

Para comprobar el funcionamiento del prototipo se lo introdujo en ambientes cotidianos en los que podría ser utilizado. Para esta prueba en concreto se hicieron recorridos libres a través de los pasillos de un centro comercial y por los pasillos de un supermercado.





Figura 4. 44 Ejemplo de recorrido por un supermercado



Figura 4. 45 Ejemplo de recorrido por un centro comercial



Figura 4. 46 Ejemplo de recorrido por un centro comercial

Estas pruebas fueron desarrolladas exitosamente y permitieron comprobar el correcto desenvolvimiento del prototipo en ambientes cotidianos. Sobretudo demostró que tiene la capacidad de mantenerse siguiendo al usuario a pesar de que distintas personas aparezcan de diversas maneras en el campo visual del robot. Además se pudo comprobar la capacidad de atravesar pasillos evitando colisiones con las diversas estructuras físicas.

Un aspecto a recalcar es que las personas no están familiarizadas a encontrarse en su ambiente con este tipo de vehículos autónomos, y a menudo corren el riesgo de tropezar, si no lo observan con anterioridad.

4.5.5. PRUEBAS CON PERSONAS AJENAS AL DESARROLLO DEL PROTOTIPO

Esta prueba fue realizada con el objetivo de probar el grado de facilidad con el que un usuario que no ha estado vinculado al desarrollo del prototipo, puede hacer uso de él.



Figura 4. 47 Prueba con un niño de 10 años de edad



Figura 4. 48 Prueba con una mujer de 24 años de edad



Figura 4. 49 Prueba con una mujer de 25 años de edad

El prototipo según dijeron quienes lo probaron, es muy fácil de manejar y responde bien a los movimientos que el usuario realiza; además expresaron que el diseño es muy atractivo a la vista.

COSTOS DEL PROYECTO

A continuación se muestra la Tabla 4.1, que detalla los costos de los elementos y de la ingeniería involucrada en el desarrollo del proyecto.

Tabla 4. 1 Detalle de costos del proyecto

Elemento	Unitario \$	Cantidad	Total \$
Par de motores elevadores de vidrios	90	1	90
Par de piezas de acople para motores	80	1	80
Rueda con neumático de 12 pulgadas	11	2	22
Platina para ejes, vara de 7m	10	1	10

Estructura de MDF del prototipo	70	1	70
Placa de acrílico para visor	4	1	4
Rueda garrucha de apoyo	0,5	1	0,5
Batería de 12Ah	70	1	70
Batería de 2Ah	24	1	24
Elementos de cableado del prototipo	12	1	12
Interruptores de 3 posiciones	1,7	2	3,4
Sensor ultrasónico EZ1	42	2	84
Sensor de GLP	17	1	17
Sensor de CO	17	1	17
LEDs indicadores alta intensidad	0,5	7	3,5
Transistores de potencia TIP 142	1	4	4
Transistores de potencia TIP 147	2	4	8
Disipadores	0,4	8	3,2
Diodos de 6A	0,6	8	4,8
Optoacopladores	1,4	6	8,4
Quema y perforación de placas de potencia	12	2	24
Ventilador de 12V	4	1	4
Microcontrolador Atmega 164P	7	1	7
Circuito integrado MAX-232	2	1	2
Circuito integrado AND 74LS08	0,5	1	0,5
Regulador de voltaje LM7805	0,35	1	0,35
Regulador de voltaje LM317	1,1	1	1,1
Módulo Bluetooth	45	1	45
Transistores NPN de señal	0,15	6	0,9
Conector de placa tipo Consil	0,5	32	16
Capacitor de 0.1uF	0,3	5	1,5
Resistencias de 1/4 W	0,02	32	0,64
Quema y perforación de placa de control	18	1	18
Cable convertidor USB-serial	15	1	15
Sensor Kinect de segunda mano	120	1	120
Smartphone Galaxy 5	180	1	180
Computadora portátil DELL Studio 1551	400	1	400
Costo total de elementos			1371,8
Costos de ingeniería			
Actividad	Costo \$/h	Horas	
Investigación, diseño y construcción de prototipo de robot acompañante	20	586	11720
Costo total del proyecto			13091,8

CAPITULO 5

CONCLUSIONES Y RECOMENDACIONES

CONCLUSIONES

Analizando los alcances y objetivos planteados para el proyecto, se determina que fueron cumplidos en su totalidad, concluyendo lo siguiente:

- Se diseñó y construyó un robot móvil tipo terrestre rodante de tracción diferencial, que puede alcanzar una velocidad máxima de 1m/s. El prototipo fue construido en MDF; un material que debido a su bajo peso y resistencia, permite aprovechar el rendimiento del vehículo en beneficio del usuario.
- Se implementó un sistema de seguimiento de personas, donde el señalizador o baliza no es un dispositivo portado por el usuario, sino que se trata de la posición de un punto del cuerpo del usuario determinado por medio de software de reconocimiento de imágenes; en este caso el punto usado es el centro del torso del usuario. El concepto de este sistema de seguimiento permite que se le pueda extender a otros tipos de control de robots; y que se escoja otros puntos del cuerpo, como pueden ser las extremidades o la cabeza.
- El sensor KINECT, a pesar de no haber sido diseñado expresamente para usarse en robótica móvil, ha demostrado ser competente para este tipo de aplicaciones. El flujo de imagen de profundidad para el reconocimiento de la figura humana se mantiene eficaz a pesar de que durante la navegación del robot se presentan vibraciones y constantes cambios de escenas e iluminación.

- El sensor de movimiento KINECT se muestra como una estupenda solución para la identificación y seguimiento de la figura humana, ya que debido a su algoritmo de identificación alcanza velocidades de procesamiento relativamente altas (hasta 30fps). Además se puede destacar la gran cantidad de herramientas de desarrollo gratuitas existentes en la actualidad.
- Para el tratamiento de datos obtenidos desde el sensor de movimiento KINECT, se desarrolló un algoritmo matemático que permite facilitar el control implementado en el microprocesador del prototipo. Los datos, tanto de posición horizontal como de profundidad del usuario, son enviados en formato byte, que de acuerdo con su rango, el sistema los identifica e interpreta como la dirección y sentido del movimiento del usuario.
- Una de las limitaciones del prototipo implementado es la zona muerta de profundidad de 80 cm de extensión del sensor KINECT utilizado. Por esta razón se ha decidido no poner el setpoint de distancia más cerca de 120cm. También por ello se ha optado por no habilitar el desplazamiento en reversa del robot, pues cuando el usuario se acerca o ingresa a esta zona, los datos de posición pierden confiabilidad. Sin embargo, se comprobó que ambientes cotidianos como centros comerciales y supermercados presentan espacios donde traslados en reversa no fueron realmente necesarios.
- Para que los robots puedan tener una mejor interacción con humanos en ambientes cotidianos, son realmente útiles los instrumentos que utilizan visión artificial como lo es el sensor KINECT. Este tipo de instrumentos demandan mayor capacidad de cómputo para el procesamiento informático que la que puede ofrecer un microcontrolador por si solo. Un mayor poder computacional permite elevar la rapidez de reacción del robot y su complejidad, por lo que involucrar computadoras a bordo de robots móviles

es cada vez más necesario si se desea avanzar en la investigación de este campo.

- Para el buen funcionamiento del prototipo se requiere garantizar pasillos mayores a 1,5m de ancho y trayectorias curvas mayores a 0,8m de radio, debido a las características físicas del prototipo.
- La relación de trabajo PWM enviada al conversor DC-DC desde el microcontrolador y la velocidad del prototipo deben tener una relación lineal, porque los compensadores dan como resultado valores que deben corresponder proporcionalmente a la velocidad del prototipo.
- Se ha demostrado que para que un robot con tracción diferencial realice una tarea de seguimiento se puede dividir su dinámica en velocidad de giro y velocidad de avance, con controladores independientes para cada uno de estos comportamientos cuyas señales se pueden superponer y complementar.
- El satisfactorio desempeño de este prototipo en ambientes cotidianos, como es un centro comercial y un supermercado, hace viable pensar en continuar su desarrollo para aplicaciones concretas como carros de compras robóticos, ofreciendo mayor confort durante las compras a clientes regulares pero sobretodo a quienes presenten algún tipo de discapacidad en su movilidad. También se puede incorporar este método de seguimiento a robots de telepresencia, potenciando su utilidad para cuando se desee hacer recorridos guiados de modo que el usuario remoto no tenga que maniobrar al robot y pueda concentrarse sólo en el recorrido. En definitiva, la funcionalidad de este prototipo podría ser incorporada a cualquier tipo de robot móvil como un modo de funcionamiento adicional.

RECOMENDACIONES

- Para llevar este prototipo a una aplicación concreta se recomienda reemplazar los motores y acoples mecánicos de los ejes, por elementos que puedan garantizar una mayor durabilidad a largo plazo. Sobre todo mediante la inclusión de amortiguadores para todas sus ruedas, de modo que el sensor KINECT y la computadora a bordo no sufran vibraciones y sacudones que puedan afectar su vida útil a largo plazo. Aplicando esta recomendación se podría levantar la restricción acerca de desplazamientos por superficies irregulares.
- Se recomienda reemplazar el KINECT utilizado, perteneciente a la consola de juegos Xbox 360, por una nueva variante de KINECT destinado para su uso con PC, cuyo costo es más elevado y que se encuentra disponible recién desde marzo de 2012, pero que ofrece la ventaja de poseer una zona muerta disminuida a 40cm de extensión.
- Para una extensión del estudio de este prototipo o para una aplicación concreta, se recomienda probar reemplazar la computadora portátil por una tablet con sistema operativo Windows 8 de las que saldrán próximamente al mercado, para hacer así posible un diseño mucho más compacto. Sin embargo, para este objetivo, no se recomienda el uso de una portátil tipo netbook, ya que al utilizarla, se demostró que su capacidad de cómputo no era suficiente para procesar la información del KINECT a la velocidad necesaria. Otra opción para un diseño más compacto podría ser despojar a la computadora portátil de sus periféricos y dejarla solo con los componentes necesarios para que corra la aplicación de KINECT.
- Se recomienda ampliar el estudio de este prototipo de robot acompañante, reemplazando la locomoción de tracción diferencial por una locomoción con patas o incluso bípeda, lo cual elevaría la utilidad del método de

seguimiento implementado, haciendo posible su aplicación en muchos más escenarios con obstáculos a nivel del piso.

REFERENCIAS BIBLIOGRÁFICAS

- [1] D. Feil-Seifer, M. J. Matarić. Human-Robot Interaction. Center for Robotics and Embedded Systems, Los Angeles, CA. [Online]. Disponible: http://cres.usc.edu/pubdb_html/files_upload/585.pdf
- [2] HRI2007. (2007, Marzo). 2nd ACM/IEEE International Conference on Human-Robot Interaction. Washington DC, USA [Online]. Disponible: <http://hri2007.org/>
- [3] Shadow Caddy Pty Ltd. (2008) Shadow Caddy-A Better Way to Play Golf. Shadow Caddy Pty Ltd. [Online]. Disponible: <http://www.shadowcaddy.com/shadowcaddy-UK.php>
- [4] D. Blanch. (2008, Julio). Entrevista a Hubert Novak. ABC Radio Australia [Online]. Disponible: <http://www.radioaustralia.net.au/innovations/stories/s2296805.htm>
- [5] ABC Television. (2011) The new Inventors, Shadow Caddy. ABC Television [Online]. Disponible: <http://www.abc.net.au/tv/newinventors/txt/s2296849.htm>
- [6] M.G.M. Marc Gotta Merchandising. Gogo kaddy, Über uns. Gogo-kaddy.de [Online]. Disponible: http://www.gogo-kaddy.de/shop/page/6?shop_param=
- [7] M.G.M. Marc Gotta Merchandising. Gogo kaddy, Funktionsweise. Gogo-kaddy.de [Online]. Disponible: http://www.gogo-kaddy.de/shop/page/7?shop_param=
- [8] M.G.M. Marc Gotta Merchandising. Gogo kaddy Shop. Gogo-kaddy.de [Online]. Disponible: http://www.gogo-kaddy.de/shop/catalog/browse?shop_param=
- [9] O.M.G. S.r.l. (2006). Gogo, Images. O.M.G. S.r.l. [Online]. Available: <http://www.golfcaddygogo.com/en/Images-gogo-electronic-golf-trolley-omg-srl.html>

- [10] E. Tuncali. (2004). Ultrasonic Signal Follower Robot. Cumhuriyet University Faculty of Engineering, Department of Mechanical Engineering. [Online]. Disponible: http://yunus.hacettepe.edu.tr/~b0022968/robot_eng.htm
- [11] A. Wiens. (2010). Ultrasonic Source Localizer. Washington University in St. Louis. [Online]. Disponible: <http://students.cec.wustl.edu/~adw4/ultrasonic/?file=kop1.php>
- [12] Hemin Shekh Omer. (2008) Information Acquisition Using Eye-Gaze Tracking for Person-Following with Mobile Robots. Hemin Shekh Omer [Offline]. Disponible: <http://www.hemin.co.uk/personFollowing.htm>
- [13] Garratt Gallagher. (2009). Garratt Gallagher-an overview. Garratt Gallagher [Online]. Disponible: <http://www.garratt.info/>
- [14] Garratt Gallagher. (2011). Robot Following Demo. ROS Org. [Online]. Disponible: <http://www.ros.org/wiki/mit-ros-pkg/KinectDemos/Follower>
- [15] John Risch. (2006). A.F.T AUTONOMOUS FOLLOWER TRANSPORT. Risch Enterprize Engieneering [Online]. Available: <http://www.riscenterprizes.com/seniorProject.pdf>
- [16] George Schreyer. (2006). Roomba. Schreyer Family Page [Online]. Disponible: http://www.girr.org/random_stuff/roomba.html
- [17] Maria Isabel Ribeiro, Pedro Lima. (2002). Kinematic Models of Mobile Robots.
- [18] Microsoft Corporation. (2011). Kinect para Xbox. Microsoft [Online]. Disponible: <http://www.xbox.com/es-ES/kinect>
- [19] Wikipedia, versión inglés. (2011). Kinect Article. Wikipedia [Online]. Disponible: <http://en.wikipedia.org/wiki/Kinect>
- [20] D. Burnett. (2011). Getting started with Kinect and OpenNI and Xbox Kinect. Don Burnett [Online]. Disponible: <http://www.uxmagic.com/blog/?tag=/NUI>
- [21] M. Rosoff. (2011). The Story Behind Kinect, Microsoft's Billion Dollar Business. Business Insider, Inc. [Online]. Disponible: <http://www.businessinsider.com/the-story-behind-microsofts-hot-selling-kinect-2011-1?op=1>

- [22] Department of Engineering at the University of Cambridge. (2011). Alumnus Dr Jamie Shotton and the development of Kinect for Xbox 360. University of Cambridge [Online]. Disponible: http://www.eng.cam.ac.uk/news/stories/2011/Xbox_Kinect/
- [23] J. Lowensohn. (2011). Timeline: A look back at Kinect's history. CBS Interactive [Online]. Disponible: http://news.cnet.com/8301-10805_3-20035039-75.html
- [24] Microsoft Corporation. (2011) Kinect for Windows. Microsoft [Online]. Disponible: <http://kinectforwindows.org/download/>
- [25] ROS wiki. (2010). Kinect Operation. ROS Org. [Online]. Disponible: http://www.ros.org/wiki/kinect_calibration/technical
- [26] PrimeSense. (2011). Our full 3D Sensing Solution. PrimeSenseBlog [Online]. Disponible: <http://www.primesense.com/en/technology/115-the-primesense-3d-sensing-solution>
- [27] M. Fisher. (2011). Kinect. Matthew Fisher [Online]. Disponible: <http://graphics.stanford.edu/~mdfisher/Kinect.html>
- [28] L. Case. (2011). Kinect in Depth. Tom's Hardware [Online]. Disponible: <http://www.tomshardware.co.uk/game-developers-conference-gdc-2011-world-of-warcraft,review-32131-2.html>
- [29] Microsoft Research, "Programming Guide Getting Started with the Kinect for Windows SDK Beta from Microsoft Research", 2011
- [30] J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, A. Blake, "Real-Time Human Pose Recognition in Parts from a Single Depth Image", IEEE Computer Vision and Pattern Recognition (CVPR), Colorado Springs, 2011
- [31] K. Mc Donald. (2010). Kinect point cloud with depth of field. Yahoo! Inc. [Online]. Disponible: <http://www.flickr.com/photos/kylemcdonald/5174106004/in/photosof-kylemcdonald/>
- [32] Hanwei Electronics Co., MQ-9 Semiconductor Sensor for CO/Combustible Gas. Hanwei Electronics CO., LTD. [Online]. Disponible: http://www.pololu.com/file/download/MQ9.pdf?file_id=0J314

- [33] MaxBotix® Inc. (2011). LV-MaxSonar®-EZ1™ High Performance Sonar Range Finder. MaxBotix Inc. [Online]. Disponible: http://www.maxbotix.com/documents/MB1010_Datasheet.pdf
- [34] POWER SONIC. PS-12100 12 Volt 12.0 AH. Power Sonic Inc. [Online]. Disponible: http://www.power-sonic.com/images/powersonic/sla_batteries/ps_psg_series/12volt/PS-12100_11_Feb_21.pdf
- [35] POWER SONIC. PS-1212 12 Volt 1.4 AH. Power Sonic Inc. [Online]. Disponible: http://www.power-sonic.com/images/powersonic/sla_batteries/ps_psg_series/12volt/PS-1212_11_Feb_21.pdf
- [36] ATMEL. (2009). 8-bit AVR Microcontroller with 16/32/64K Bytes In-System Programmable Flash. Atmel Corporation [Online]. Disponible: http://www.atmel.com/dyn/resources/prod_documents/doc7674.pdf
- [37] FAIRCHILD Semiconductor. (2000). PNP Epitaxial Silicon Darlington Transistor. Fairchild Semiconductor International [Online]. Disponible: <http://www.datasheetcatalog.org/datasheet/fairchild/TIP147.pdf>
- [38] FAIRCHILD Semiconductor. PNP Epitaxial Silicon Darlington Transistor. Fairchild Semiconductor International [Online]. Disponible: <http://www.datasheetcatalog.org/datasheet/fairchild/TIP142.pdf>
- [39] K. Ogata, "Ingeniería de control moderna", 3ra ed., Ed. Mexico: Prentice-Hall, Inc., 1998.

ANEXO A

MANUAL DE USUARIO

1) Requisitos de funcionamiento

- a. La computadora que va a ser utilizada debe cumplir con los siguientes requerimientos:

Sistema operativo

- Windows 7 (x86 o x64)

Requerimientos de hardware

- Computadora con procesador dual-core, 2.66-GHz o mayor.
- Tarjeta de video compatible con Windows 7, que soporte Microsoft® DirectX® 9.0c.
- 2 GB de RAM
- Sensor Kinect para Xbox 360®, con sus cables correspondientes.

Requerimientos de software

- Microsoft Visual Studio® 2010 Express u otra edición Visual Studio 2010.
 - Microsoft .NET Framework 4 (instalado con Visual Studio 2010)
 - DirectX Software Development Kit (DirectX SDK), versión Junio 2010 o posterior.
 - Instalador DirectX End-User Runtime Web.
- b. Instalar el programa “KinectSDK-v1.0”, de la carpeta INSTALADORES contenida en el CD de software, ejecutando el archivo KinectSDK-v1.0-Setup.
- c. Instalar el programa “epn_robot_acompañante”, de la carpeta INSTALADORES contenida en el CD de software, ejecutando el archivo SETUP.
- d. Copiar del CD de software a un directorio del teléfono, vía cable USB, la aplicación “EPNROBOT.apk” e instalarla.

- e. Realizar el emparejamiento Bluetooth del teléfono inteligente con el modulo bluetooth del prototipo. En caso de requerirlo, el código de seguridad para emparejar es 1234.

2) Montaje de compartimientos

El prototipo consta de 3 compartimientos:

El principal donde se ubican motores, tarjetas electrónicas y baterías (Figura A.1).



Figura A.1 Compartimiento principal

El compartimiento para la computadora portátil, donde se encuentran los terminales USB de conexión del sensor y del circuito de control hacia la computadora (Figura A.2).



Figura A.2 Compartimiento para computadora

El compartimiento de carga, donde se colocan los utensilios del usuario (Figura A.3)



Figura A.3 Compartimiento de carga

Estos tres compartimientos se montan de la forma en que muestra la Figura A.4, con la siguiente secuencia:



Figura A.4 Armado general

- a. Montar el compartimiento de la computadora.
- b. Conectar los terminales de alimentación del sensor KINECT.
- c. Conectar el cable convertidor USB-Serial al cable serial proveniente del circuito de control.
- d. Montar el compartimiento de carga.

Los cables se encuentran etiquetados adecuadamente, para impedir equivocaciones.

3) Ubicación de la computadora

La ubicación de la computadora se la hace en el compartimiento destinado para esto, en la siguiente secuencia:

- a. Conectar el cable USB-Serial, en el puerto que se prefiera.



Figura A.5 Conexión cable USB-serial

- b. Conectar el cable USB del sensor KINECT.



Figura A.6 Conexión USB sensor KINECT

- c. Insertar la computadora, en el compartimiento.



Figura A.7 Colocación de la computadora

4) Carga de baterías

- a. Colocar los interruptores del prototipo, en posición de carga.
 - a. Si se desea cargar la batería de potencia, en interruptor de mayor tamaño.
 - b. Si se desea cargar la batería de control, en interruptor de menor tamaño.



Figura A.8 Tablero de interruptores

- b. Conectar el cargador de baterías.



Figura A.9 Conexión de cargador de baterías.

5) Secuencia de encendido

- Abrir el programa “epn_robot_acompañante”, previamente instalado en la computadora.
- Colocar los interruptores del prototipo, en posición de encendido (Figura A.8).
- En el menú desplegable del programa, seleccionar el nombre del puerto serial, en el cual se encuentra el cable USB-Serial.
- Pulsar el botón conectar.

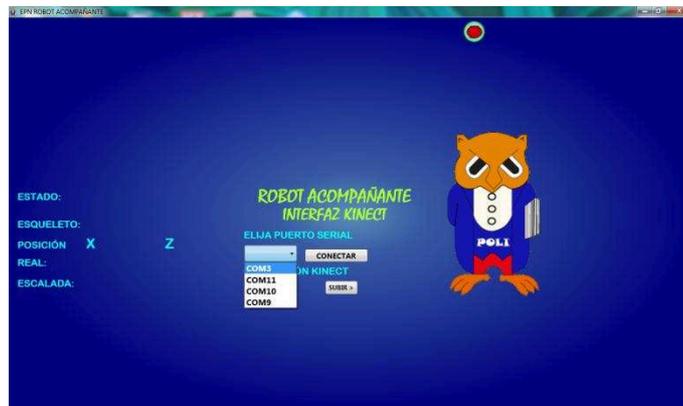


Figura A.10 Selección de puerto serial

- De ser necesario, de acuerdo al usuario, regular el ángulo de visión del sensor KINECT, como valor predeterminado se encuentra 18°.



Figura A.11 Regulación de ángulo

6) Uso del Robot

Una vez realizada la secuencia de encendido, se puede proceder a usar el ROBOT ACOMPAÑANTE.

a) Conexión de la aplicación para teléfono inteligente

En el teléfono inteligente, hay que abrir la aplicación EPNROBOT. En la parte **inferior** de la pantalla inicial (Figura A.12) se presentan dos botones, el primero llamado DISPOSITIVOS y el segundo CONECTAR.

Para continuar, el recurso Bluetooth del teléfono inteligente debe encontrarse activado.

Al pulsar el botón DISPOSITIVOS se desplegará una lista con direcciones de los dispositivos Bluetooth emparejados. Se debe seleccionar la dirección **00:19:5D:EE:40:1C** la cual corresponde al componente Bluetooth del ROBOT ACOMPAÑANTE. Para la siguiente vez que se abra el programa, esta dirección estará preseleccionada y aparecerá en lugar del botón DISPOSITIVOS, por lo que entonces no será necesario abrir la lista de dispositivos nuevamente, ni hacer la selección.



Figura A.12 Pantalla inicial de la aplicación EPNROBOT para el ROBOTACOMPAÑANTE

Entonces se debe pulsar el botón CONECTAR, si la conexión fue exitosa al lado del botón aparecerá la palabra CONECTADO resaltada en verde, de lo contrario se aparecerá la palabra FALLA y se deberá reintentar la conexión.

Con la conexión establecida ahora ya se puede elegir el modo de trabajo del Robot, pulsando los botones superiores del menú de la aplicación del teléfono.

b) Uso del modo de manejo del robot

Para manejar el robot manualmente, en el menú superior se debe pulsar el botón MANEJAR. Se desplegará el mando virtual, con forma de un botón cuadrado dentro de un campo cuadrado.



Figura A.13 Elección del modo de manejo en la aplicación EPNROBOT

Se debe arrastrar este botón hasta que toque los bordes del campo. Tocar los distintos bordes del campo hace que el robot se mueva en las 4 direcciones. Para frenar al robot se debe hacer un “click” sobre el mismo botón cuadrado.



Figura A.14 Mando virtual para manejo en las 4 direcciones.

c) Uso del modo de seguimiento del robot

Para usar el robot en modo de seguimiento, en el menú superior del teléfono se debe pulsar el botón SEGUIR. Se desplegarán los botones de mando para el inicio y detención del seguimiento y el robot encenderá una luz verde.



Figura A.15 Elección del modo de seguimiento en la aplicación EPNROBOT

Entonces el usuario debe pararse en la posición inicial, aproximadamente a un paso del frente del robot, para ser reconocido. Cuando el robot lo reconozca, encenderá sus dos luces verdes y se oirá una melodía. En este momento el robot está listo y el usuario puede ya pulsar el botón de mando “play” y caminar.



Figura A.16 Reconocimiento del usuario en la posición inicial, luces verdes que avisan que está listo y botones de mando para seguimiento

Durante la caminata, el robot mantendrá encendida su luz central amarilla. Si por cualquier motivo el usuario es perdido de vista, el robot se detendrá y sonará una

alarma. De suceder esto, para reanudar la caminata, el usuario debe volver a pararse frente al robot para ser reconocido.



Figura A.17 Luz amarilla central durante seguimiento

Para concluir el seguimiento del robot se debe pulsar el botón “pausa” en el teléfono.

Para concluir el uso en general del robot se debe pulsar el botón STOP en el menú superior de la aplicación, con lo cual además se vuelve a la pantalla inicial.

d) Recomendaciones para uso del robot durante el modo de seguimiento

- No intente recorrer con el robot a través de puertas o pasillos con una extensión menor a 80cm de ancho, pues existe el riesgo inminente de quedar atorado. Para su óptimo desempeño se recomienda su uso a través de pasillos o puertas con un ancho mayor a 150cm.
- El robot no cuenta con marcha reversa por lo que si el usuario se acerca hacia él, no retrocederá sino que se quedará quieto y si se continúa acercando demasiado, probablemente el usuario dejará de ser reconocido.
- La caminata con el robot debe ser de velocidad hasta moderada y en ningún caso se debe trotar o correr.
- A pesar de que el robot cuenta con sensores anticolidión, para cuando hay virajes en el recorrido, se recomienda caminar en las esquinas procurando trazar una curva un tanto amplia y así evitar percances.

- El uso del robot se encuentra indicado para interiores, sin rampas ni irregularidades en el piso y no para exteriores.
- Este prototipo no contempla una carga mayor a 4 Kilogramos en su área de carga.
- Se puede intercambiar el modo de uso entre manejo y seguimiento, a conveniencia e inmediatamente, mediante el menú superior del teléfono.

ANEXO B ESQUEMÁTICOS

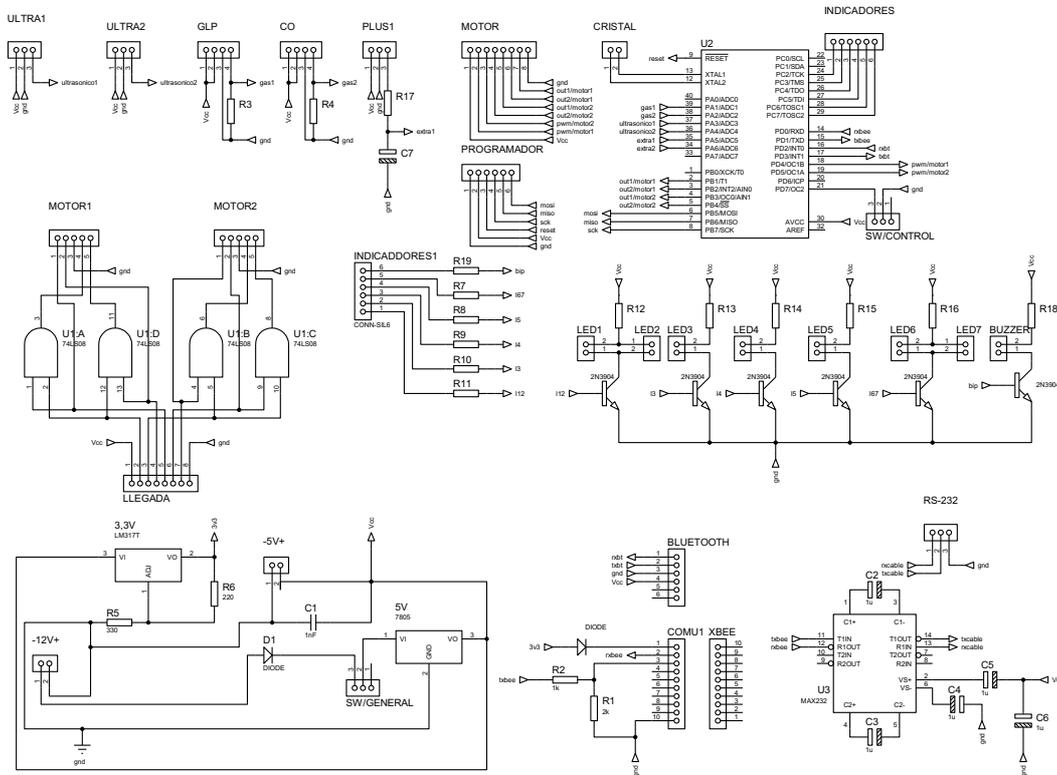


Figura B.1 Esquema de tarjeta de control

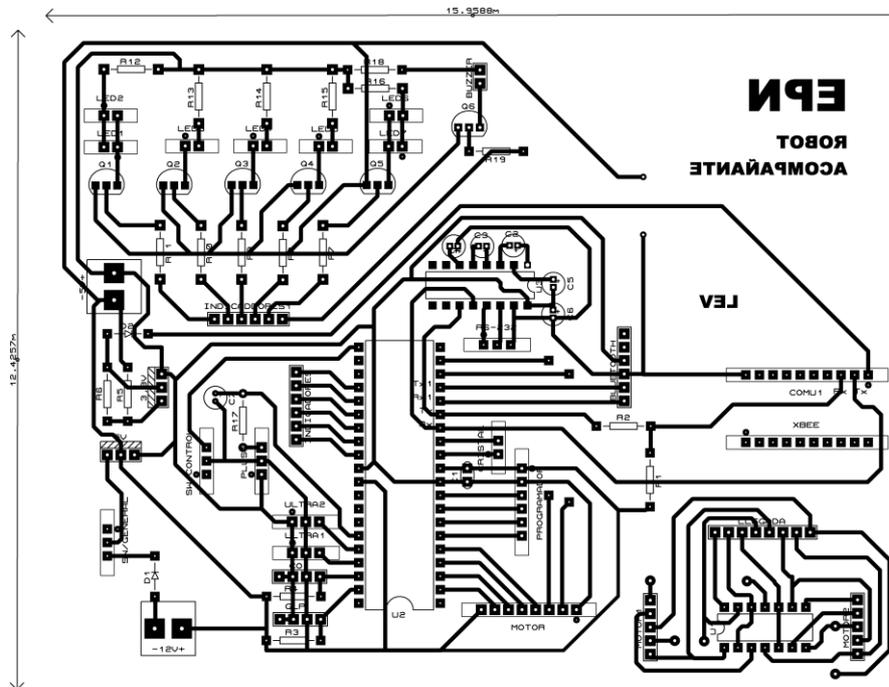


Figura B.2 PBC de tarjeta de control

Descripción	Designación	Cantidad
Compuertas AND 74ls08	U1	1
Microcontrolador Atmega 164p	U2	1
Integrado MAX232	U3	1
Capacitor 100nF	C1	1
Capacitor 1uF	C2,C3,C4,C5,C6	5
Transistores NPN 2N3904	Q1,Q2,Q3,Q4,Q5,Q6	4
Regulador de voltaje 7805	5V	1
Regulador de voltaje LM317	3.3V	1
Diodos de 1A 1N4007	D1,D2	2
Resistencia 10K ohm	R1	1
Resistencia 5,6K ohm	R2	1
Resistencia 510 ohm	R5,R7,R8,R9,R10,R11,R19	7
Resistencia 330 ohm	R6	1
Resistencia 100 ohm	R12,R16	2
Resistencia 150 ohm	R13,R14,R15,R18	4
Conector molex KK, 5 pines	motor1,motor 2	2
Conector molex KK, 8 pines	motor, llegada	2
Conector molex KK, 6 pines	Indicadores,indicadores1	2
Conector molex KK, 2 pines	led1,led2,led3,led4,led5,led6,led7	7
Conector molex KK, 3 pines	rs232,glp,ultra1,ultra2	4

Figura B.3 Tabla de elementos de tarjeta de control

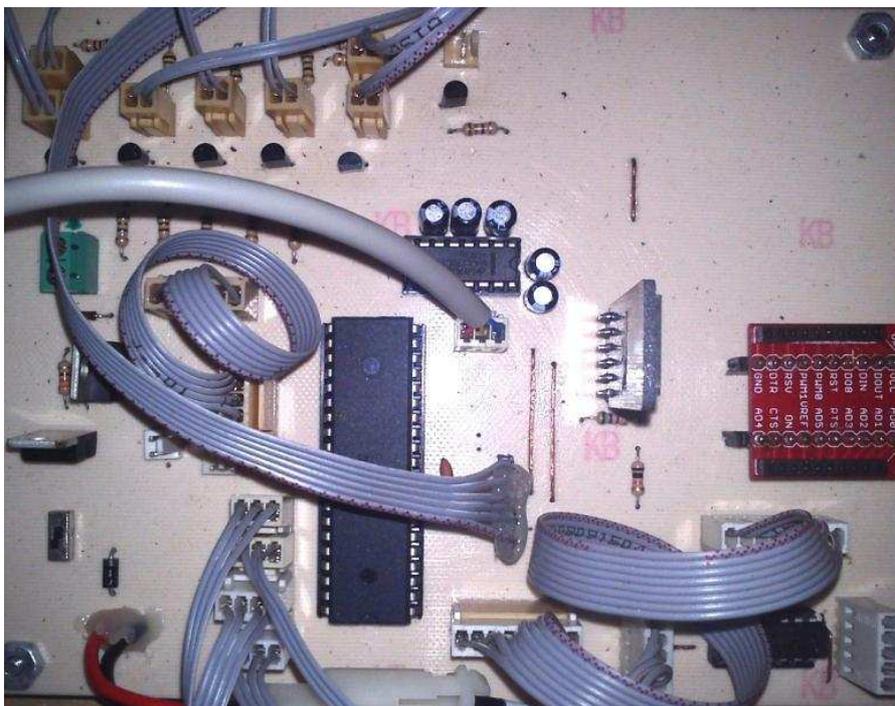


Figura B.4 Fotografía de tarjeta de control, en sitio

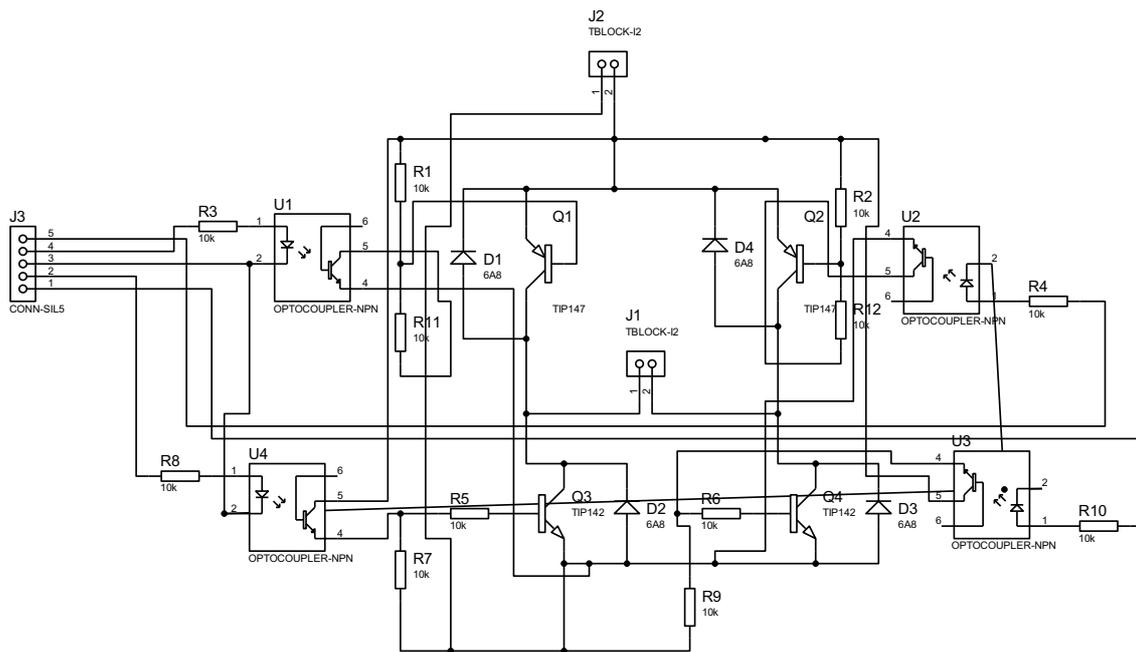


Figura B.5 Esquema de tarjeta controladora de motor

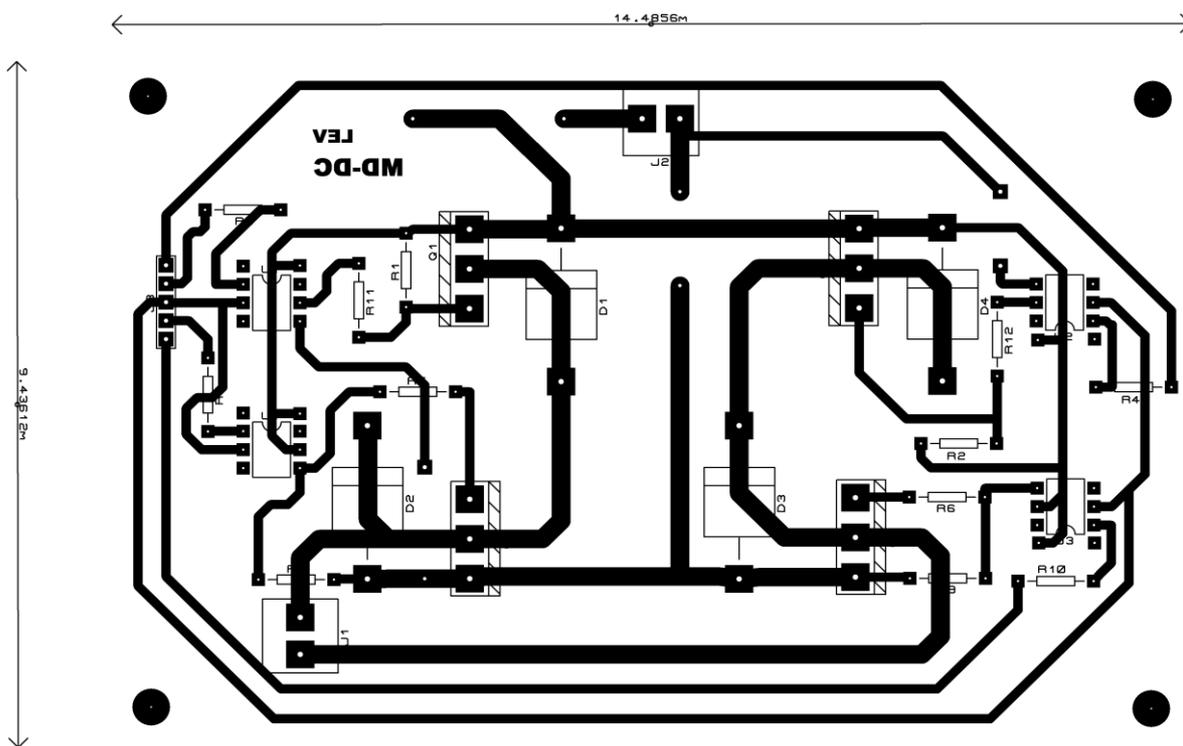


Figura B.6 PBC de tarjeta controladora de motor

Descripción	Designación	Cantidad
Transistor de potencia TIP147	Q1 ,Q2	2
Transistor de potencia TIP142	Q3,Q4	2
Diodo de potencia 6 ^a	D1,D2,D3,D4	4
Optoacoplador 6n137	U1,U2,U3,U4	4
Resistencias de 220 ohm	R3,R4,R8,R10	4
Resistencias de 10K ohm	R1,R2,R7,R9	4
Resistencias de 510 ohm	R5,R6,R11,R12	4
Conector tipo T-block	J1,J2	2
Conector molex KK, 5 pines	J3	1
Portafusible	--	1

Figura B.3 Tabla de elementos de tarjeta de controladora de motor

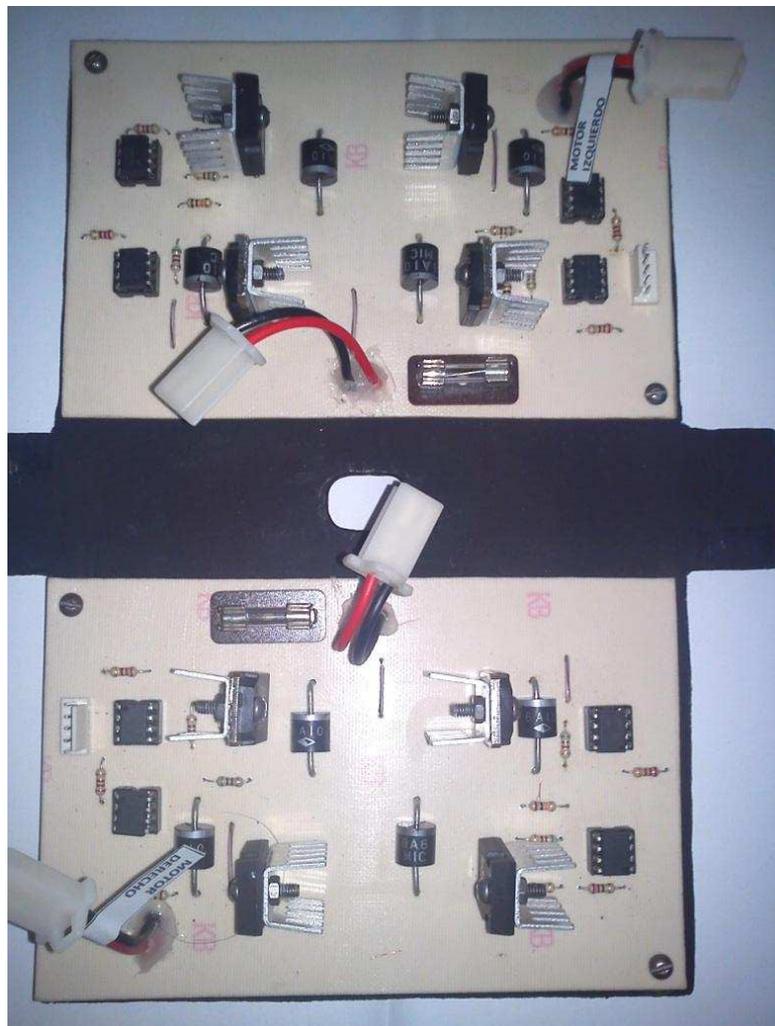
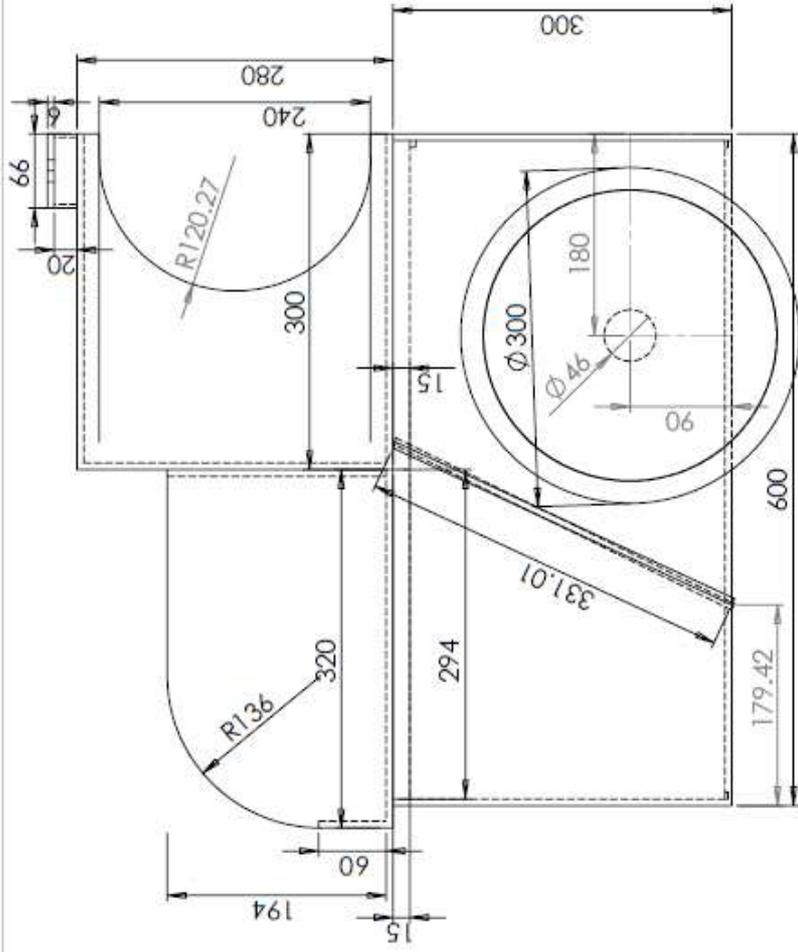


Figura B.8 Fotografía de las tarjetas de controladoras de motor, en sitio

ANEXO C

PLANOS



DIBUJADO		NOMBRES		FECHA	
Christian López		Alejandro Mesa		Mayo/ 2012	
Revisado		Ing. Nelson Sotomayor		Mayo/ 2012	
Aprobado		Ing. Nelson Sotomayor		Mayo/ 2012	
ESCUELA POLITÉCNICA NACIONAL				ESCALA: 1:5	
ELECTRÓNICA Y CONTROL				Plano #3	

TÍTULO: Robot Acompañante Vista Lateral

1 2 3 4 5

ANEXO D

HOJAS DE DATOS

