

# **ESCUELA POLITÉCNICA NACIONAL**

## **FACULTAD DE INGENIERÍA ELÉCTRICA Y ELECTRÓNICA**

### **DISEÑO, CONSTRUCCIÓN Y CONTROL DE UN ROBOT MÓVIL TIPO UNICICLO UTILIZANDO FPGA'S**

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

**LENIN RAFAEL TAPIA JARRÍN**

(lenin\_tapia@ieee.org)

**CHRISTIAN DARÍO SAMPEDRO REA**

(sampeter\_19@hotmail.com)

**DIRECTOR: Dr. ANDRÉS ROSALES ACOSTA**

(androsaco@gmail.com)

**Quito, Marzo 2014**

## DECLARACIÓN

Nosotros, Tapia Jarrín Lenin Rafael y Sampedro Rea Christian Darío, 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.

---

Tapia Jarrín Lenin Rafael

---

Sampedro Rea Christian Darío

## CERTIFICACIÓN

Certifico que el presente trabajo fue desarrollado por Tapia Jarrín Lenin Rafael y Sampedro Rea Christian Darío, bajo mi supervisión.

---

Dr. Jorge Andrés Rosales

DIRECTOR DEL PROYECTO

## **AGRADECIMIENTO**

Agradezco a mi padre Carlos Hernán Tapia Farinango por ser el ejemplo más grande en mi vida y quien me ha direccionado para ser quien hoy soy, a mi madre Fadua del Rocío Jarrín Moncayo por ser siempre mi compañera de estudios y mi gran fortaleza en la vida; a ustedes que con su esfuerzo me han llevado hasta este momento de inmensa felicidad, a mi hermanita Fadua Elizabeth Tapia Jarrín por estar toda la vida a mi lado en los momentos alegres así como también en los tristes, a mi amada esposa Eylem Güngör por todo el apoyo moral en este trayecto, a mi hijo Devrim Hernán por ser la fuente de mi fortaleza en cada día, a mi futur@ hij@ por traerme una nueva alegría a mi vida, a mi tutor el Dr Andrés Rosales Acosta por el apoyo y dirección en el desarrollo del proyecto, finalmente a mis entrañables amigos con quienes he compartido cada momento hasta el día presente.

**Lenin Tapia**

Agradezco a Dios primeramente por permitirme la vida para poder cumplir una meta más en mi vida, a mis padres Carlos Alfredo Sampedro Vargas y Mercedes del Rocío Rea García por darme todo el apoyo, amor, valores y el ejemplo a lo largo de mi vida para ser quien soy, a mi hermano por mostrarme cada día que el esfuerzo tiene su recompensa y que nada es imposible, a mi amigo y compañero de tesis Lenin Tapia por la dedicación y el esfuerzo de cada día para sacar adelante este proyecto, a mi tutor Dr. Andrés Rosales por su apoyo, paciencia y conocimientos brindados a lo largo de este proyecto y a mis amigos que siempre han estado ahí a mi lado luchando a lo largo de la vida estudiantil y universitaria.

**Christian Sampedro**

## DEDICATORIA

Este Proyecto de Titulación va dedicado a mis padres Carlos Hernán y Fadua del Rocío, a mi hermanita Fadua, a mi esposa Eylem, a mi hijo Devrim, a mi pequeñit@ que está por venir así como también a mi tío Pablo, mis abuelitas Carmen, Fabiola y Teresa.

**Lenin Tapia**

Este Proyecto de Titulación va dedicado primeramente a Dios por permitirme cumplir una meta más en mi vida, a mis padres Carlos Alfredo Sampedro Vargas y Mercedes del Rocío Rea García por su esfuerzo, confianza, amor y apoyo incondicional.

**Christian Sampedro**

<b>RESUMEN .....</b>	<b>IX</b>
<b>PRESENTACIÓN .....</b>	<b>X</b>
<b>CAPITULO 1.....</b>	<b>1</b>
<b>MARCO TEÓRICO .....</b>	<b>1</b>
1.1 INTRODUCCIÓN .....	1
1.2 ROBOTS MÓVILES TERRESTRES.....	1
1.2.1 Definición .....	1
1.2.2 Tipos de robots móviles terrestres .....	2
1.2.2.1 Robots con ruedas.....	2
1.2.2.2 Robots tipo oruga.....	2
1.2.2.3 Robots con patas.....	3
1.2.2.4 Robots ápodos .....	3
1.3 APLICACIONES DE LOS ROBOTS MÓVILES [7] .....	4
1.3.1 Aplicaciones Industriales.....	4
1.3.2 Aplicaciones de servicio e innovación.....	4
1.3.3 Aplicaciones especiales y militares .....	5
1.4 SISTEMAS HOLONÓMICOS Y NO HOLONÓMICOS .....	6
1.5 CONFIGURACIONES TÍPICAS PARA ROBOTS MÓVILES CON RUEDAS [10] .....	7
1.5.1 Configuración diferencial [11] .....	8
1.5.2 Configuración triciclo [11] .....	9
1.5.3 Configuración ackerman [11].....	10
1.5.4 Configuración sincronizada [11] .....	10
1.5.5 Configuración omnidireccional .....	12
1.5.6 Configuración con múltiples grados de libertad .....	13
1.6 ROBOT MÓVIL TIPO UNICICLO .....	14
1.7 FPGA ( <i>FIELD PROGRAMMABLE GATE ARRAY</i> ) .....	15
1.7.1 Arquitectura.....	15
1.7.2 Antifuse .....	16
1.7.3 SRAM.....	16
1.7.4 Celdas Lógicas.....	17
1.7.5 Beneficios.....	19
1.7.5.1 Rendimiento.....	19
1.7.5.2 Precio .....	19
1.7.5.3 Fiabilidad .....	19
1.7.6 Criterios para escoger un FPGA .....	20
1.7.7 Programación.....	20

1.7.8	Aplicaciones .....	21
1.7.9	Tecnología de la memoria de programación.....	22
<b>CAPITULO 2.....</b>		<b>23</b>
<b>DISEÑO DEL SISTEMA ROBÓTICO MÓVIL .....</b>		<b>23</b>
2.1	INTRODUCCIÓN .....	23
2.2	MODELO MATEMÁTICO .....	23
2.3	DESCRIPCIÓN Y DISEÑO DE LA ESTRUCTURA.....	25
2.3.1	Estructura mecánica.....	26
2.3.2	Ruedas.....	27
2.3.3	Sensores .....	28
2.3.3.1	Los encoders .....	29
2.3.4	Actuadores .....	33
2.3.4.1	Motor de corriente continua.....	34
2.3.5	Sistema de alimentación con baterías .....	38
2.3.5.1	Baterías Ni-Cd.....	38
2.3.5.2	Baterías Ni-MH .....	39
2.3.5.3	Baterías de Litio.....	40
2.3.5.4	BATERÍAS Tipo Li-Po (polímero de litio) .....	41
2.3.6	SISTEMA INTELIGENTE .....	44
<b>CAPITULO 3.....</b>		<b>45</b>
<b>DESCRIPCIÓN DE LA ESTRUCTURA .....</b>		<b>45</b>
3.1	INTRODUCCIÓN .....	45
3.2	CARACTERÍSTICAS FÍSICAS DEL ROBOT .....	45
3.2.1	DISPOSICIÓN FÍSICA DE LOS COMPONENTES.....	46
3.3	COMPONENTES DEL SISTEMA ROBÓTICO .....	53
3.3.1	SPARTAN 3E FPGA STARTER KIT (500K).....	53
3.3.1.1	Puerto Serial RS-232 del SPARTAN 3E FPGA STARTER KIT.....	54
3.3.1.2	MODULO XBEE SERIES2 Y COMPLEMENTOS .....	56
3.3.1.3	Configuración de los módulos XBEE series 2 .....	58
3.3.1.4	Convertor Digital a Análogo (DAC) del Spartan 3E FPGA Starter Kit .....	61
3.3.1.5	ATMEGA8 .....	63
3.3.1.6	ATMEGA16 .....	65
3.3.1.7	Conectores de expansión del Spartan 3E fpga Starter Kit .....	68
3.3.2	ENCODERS INCREMENTALES .....	70
3.3.3	MOTORES DC .....	71
3.3.4	REGULADOR DE VOLTAJE .....	73

3.3.5	BUFFER CD4050BE .....	74
3.3.6	BATERÍAS Tipo Li-Po (polímero de litio) [36] .....	75

## **CAPITULO 4..... 77**

### **DESARROLLO DEL CONTROL DEL SISTEMA ROBÓTICO MÓVIL MEDIANTE UNA FPGA ..... 77**

4.1	INTRODUCCIÓN .....	77
4.2	DESARROLLO MATEMÁTICO DEL CONTROL [38] .....	78
4.2.1	USO DE LA INTERPOLACIÓN LINEAL PARA EL CONTROL DE SEGUIMIENTO DE TRAYECTORIA .....	78
4.2.1.1	Interpolación Lineal.....	78
4.2.1.2	Planteamiento del Problema.....	79
4.3	DESARROLLO DEL PROGRAMA UTILIZANDO LABVIEW .....	85
4.3.1	Programa para CREACIÓN DE TRAYECTORIAS y valores deseados de la trayectoria.....	85
4.3.1.1	Sub VI de la trayectoria de la recta .....	85
4.3.1.2	Sub VI de la trayectoria de la circunferencia.....	86
4.3.2	Programa ADQUISICIÓN de datos y estimación de Delta s .....	87
4.3.2.1	Sub VI para adquisición de pulsos del encoder .....	88
4.3.2.2	Sub VI para Conversión de pulsos a radianes, estimación de Delta S y para obtener valores actuales.....	89
4.3.2.3	Sub VI Temporizador .....	91
4.3.3	Programa para EL CONTROL DE VELOCIDAD.....	92
4.3.3.1	Sub VI para obtener valores de Velocidad Lineal y Velocidad Angular de las ruedas y para el control de la velocidad de las ruedas .....	93
4.3.4	Programa para Conversión y Transferencia de datos (DAC) .....	94
4.3.4.1	Sub VI de ajuste de la trama y configuración de la DAC.....	95
4.3.4.2	Subvi para transferencia de datos a la computadora.....	96
4.3.5	Programa DE VISUALIZACIÓN DEL SEGUIMIENTO DE LA TRAYECTORIA Y COMPORTAMIENTO DEL ROBOT.....	99

## **CAPITULO 5..... 102**

### **PRUEBAS Y RESULTADOS ..... 102**

5.1	INTRODUCCIÓN .....	102
5.2	RESULTADOS DE LAS SIMULACIONES EN COMPUTADOR.....	102
5.2.1	simulación Trayectoria lineal .....	102
5.2.2	simulación trayectoria parabólica .....	104
5.2.3	simulación trayectoria senoidal .....	106
5.2.4	simulación trayectoria circular .....	109



5.3	PRUEBAS REALES.....	113
5.3.1	TRAYECTORIA RECTA.....	113
5.3.2	TRAYECTORIA CIRCULAR.....	121
5.3.2.1	PRUEBA 1.....	121
5.3.2.2	PRUEBA 2.....	130
<b>CAPITULO 6.....</b>		<b>138</b>
<b>CONCLUSIONES Y RECOMENDACIONES.....</b>		<b>138</b>
6.1	CONCLUSIONES.....	138
6.2	RECOMENDACIONES.....	141
<b>REFERENCIAS BIBLIOGRÁFICAS.....</b>		<b>143</b>
<b>ANEXOS .....</b>		<b>147</b>
ANEXO 1.....		147
ANEXO 2.....		150
ANEXO 3.....		153
ANEXO 4.....		157
ANEXO 5.....		161
ANEXO 6.....		165
ANEXO 7.....		167
ANEXO 8.....		168
ANEXO 9.....		171

## RESUMEN

Este proyecto de titulación trata sobre el diseño y construcción de un robot móvil no holonómico tipo unicycle controlado a través de un FPGA, en este caso de un Spartan 3E Starter Kit de Xilinx, para el seguimiento de trayectorias utilizando la odometría.

Se realiza el diseño de un algoritmo utilizando un control basado en la interpolación lineal para la consecución de los estados deseados a partir de una trayectoria ideal, el mismo permite determinar las acciones correctivas a tomar a través de 2 motores acoplados en las llantas posteriores del robot para conseguir llevar al móvil a la posición deseada en el siguiente período de muestreo.

Se utiliza los periféricos disponibles en el Spartan 3E Starter Kit para la adquisición de datos, el control de actuadores y el envío de datos hacia la PC para la visualización del comportamiento del robot; así como también se implementan una serie de circuitos periféricos adicionales para el tratamiento de datos y acondicionamiento de señales necesarios.

La comunicación entre el robot y el PC se realiza a través de bluetooth con la ayuda del puerto serie en el Spartan 3E y dos módulos XBEE S2, lo cual permite el libre movimiento del robot sin estar físicamente conectado con cables a la PC.

Para la visualización del seguimiento de la trayectoria y del comportamiento del robot se ha elaborado una interfaz en LabVIEW, la misma que está dividida en tres partes: una que permite observar únicamente la trayectoria recorrida por el robot, la segunda que permite una comparación entre la trayectoria seguida por el robot y la trayectoria ideal generada en el FPGA y, finalmente, una tercera en la que se puede observar el desarrollo de las coordenadas  $(x, y)$  así como de las  $(x_d, y_d)$  deseadas para poder observar el error de posición. Es importante señalar que la PC es utilizada únicamente para la visualización de los datos, el control es íntegramente realizado en el FPGA del Spartan 3E Starter Kit.

## PRESENTACIÓN

El Proyecto de Titulación está distribuido de la siguiente manera:

En el Capítulo 1 se presenta el marco teórico relacionado con el proyecto tal como son definiciones sobre robots móviles, sus distintas configuraciones y usos; también se realiza una pequeña introducción a los FPGAS, sus características y bondades.

En el Capítulo 2 se hace una presentación de las diferentes partes que componen un sistema robótico como son: su estructura mecánica, sistemas de accionamiento, sistemas de control, sistemas de sensores; además, una pequeña introducción al tipo de elementos a ser utilizados en la construcción del sistema robótico.

En el Capítulo 3 se encuentran descritos cada uno de los elementos empleados para conformar la estructura del robot, así como también, la disposición física que cada uno de ellos presenta en el mismo y su función dentro del sistema robótico.

En el Capítulo 4 se presenta el desarrollo matemático utilizado para realizar el control del sistema robótico, de igual manera el software desarrollado a través de LabVIEW para la programación del FPGA en el Spartan 3E Starter Kit y la interfaz elaborada para la visualización del seguimiento de trayectorias y comportamiento del robot.

En el Capítulo 5 se muestran los resultados obtenidos con el robot al realizar el seguimiento de diferentes trayectorias, los mismos que se obtuvieron a través de pruebas experimentales y simulaciones en computadora.

En el Capítulo 6 se presentan las conclusiones y recomendaciones obtenidas en base a resultados y a las experiencias dadas en el transcurso de la realización del presente proyecto de titulación.

Como Anexos se tiene: hojas de datos de los componentes utilizados y programas realizados para los microcontroladores.

# CAPITULO 1

## MARCO TEÓRICO

### 1.1 INTRODUCCIÓN

En el presente capítulo se hace una introducción al estudio de los robots móviles, sus diferentes tipos, características y aplicaciones en las cuales pueden ser usados, así como las distintas configuraciones que presenta un robot móvil con ruedas, sus ventajas, desventajas y utilidades.

También se estudia a los FPGA<sup>1</sup>, su arquitectura, aplicaciones y beneficios de su uso en el ámbito de la ingeniería.

### 1.2 ROBOTS MÓVILES TERRESTRES

#### 1.2.1 DEFINICIÓN

“Los robots móviles son dispositivos de transporte automático, es decir, una plataforma mecánica dotada de un sistema de locomoción capaz de navegar a través de un determinado ambiente de trabajo, dotado de cierto nivel de autonomía para su desplazamiento portando cargas.

Sus aplicaciones pueden ser muy variadas y siempre están relacionadas con tareas que normalmente son riesgosas o nocivas para la salud humana, en áreas como la agricultura, en el transporte de cargas peligrosas o en tareas de exploración solitarias o cooperativas junto a otros vehículos no tripulados”. [1]

---

<sup>1</sup>Un FPGA (del inglés *Field Programmable Gate Array*) es un dispositivo semiconductor que contiene bloques de lógica cuya interconexión y funcionalidad puede ser configurada mediante un lenguaje de descripción especializado.

## 1.2.2 TIPOS DE ROBOTS MÓVILES TERRESTRES

### 1.2.2.1 Robots con ruedas

Son los más populares principalmente por la simpleza de su construcción y por la mayor capacidad de carga a transportar comparada con otros robots móviles, pero presentan un desempeño bajo en terrenos irregulares.



Figura 1.1 Robot con Ruedas, tomado de [2]

### 1.2.2.2 Robots tipo oruga

Similares a los de tipo con ruedas, con un desempeño mayor en terrenos irregulares pero con una disminución en velocidad.



Figura 1.2 Tipo Oruga, tomado de [3]

### 1.2.2.3 Robots con patas

Comparado a los anteriores presentan una mayor funcionalidad en terrenos irregulares.



Figura 1.3 Robot con patas, tomado de [4]

### 1.2.2.4 Robots ápodos

Normalmente conocidos como robots gusanos o robots serpientes, presentan reducida velocidad y agilidad, pero por su forma pueden alcanzar sitios inaccesibles para otro tipo de robots. [5]



Figura 1.4 Robot ápodo, tomado de [6]

Para este trabajo se usa un robot móvil con ruedas, específicamente del tipo unicycle en una configuración no holonómica, estas características se describirán posteriormente en el capítulo de diseño.

### **1.3 APLICACIONES DE LOS ROBOTS MÓVILES [7]**

El campo de acción de los robots móviles en la cotidianeidad de la humanidad se ha extendido en gran medida, es así que hoy en día se puede encontrar robots que actúen en todas o casi todas las áreas de trabajo, entretenimiento y servicios que el hombre requiere.

A continuación se detallan algunos sectores en los que la robótica móvil ha tenido un mayor impacto.

#### **1.3.1 APLICACIONES INDUSTRIALES**

Los robots que ayudan en el proceso industrial son muchos, siendo principalmente aquellos que sirven en la interconexión de las cadenas de producción usados en tareas como transporte de cargas, manejo de materiales en general, manejo de materiales peligrosos, transporte de equipos en áreas contaminadas, limpieza de espacios, carga y descarga de materiales.

Estos autómatas han ayudado al desarrollo y sobre todo a la automatización de las empresas, siendo estas herramientas indispensables en la mejora de la productividad y el avance de las mismas, haciendo que las empresas inviertan en estas tecnologías para poder mantener su posición en los mercados y no rezagarse frente a su competencia.

#### **1.3.2 APLICACIONES DE SERVICIO E INNOVACIÓN**

Según la Federación Internacional de Robótica (IFR), un robot de servicio es un robot que opera de forma parcial o totalmente autónoma, para realizar servicios

útiles para el bienestar de los humanos y del equipamiento, excluyendo operaciones de manufactura.

Es así que estos robots pueden a su vez clasificarse en 2 grupos: robots de servicio profesional y robots de servicio personal. Los robots de servicio profesional serán aquellos usados en trabajos, en público, en ambientes arriesgados, en posiciones tal como de alta mar, campos de batalla y espacio, sólo para mencionar unos cuantos.

En adición a las áreas de servicio, como limpieza, vigilancia, inspección y mantenimiento. Los robots personales son los robots de servicio que educan, asisten, o entretienen en casa. Éstos incluyen robots domésticos que pueden realizar tareas diarias, robots de servicio para la gente con discapacidad o adultos mayores, y los robots que pueden servir como compañeros o animales domésticos para entretenimiento.

Es así que se puede observar robots en áreas como la medicina como asistentes en procesos quirúrgicos o de contacto con el paciente, en el hogar en tareas como la limpieza o vigilancia o también aquellos robots para los más pequeños que sirven para estimulación temprana, aprendizaje o simple entretenimiento.

### **1.3.3 APLICACIONES ESPECIALES Y MILITARES**

Éstos son usados con propósitos específicos, por lo tanto necesitan un tratamiento y diseño especial. Se puede ver robots utilizados para la exploración submarina o aviones no tripulados con objetivos de espionaje, patrullajes en terrenos hostiles, a estos podrían ser añadidos armamento no letal para la disuasión en situaciones que conllevan un gran riesgo para las personas en la milicia, se puede destacar también en el uso de robots para desactivación de elementos explosivos y de detección de armamento.

Es por ello que muchos gobiernos del mundo hacen grandes inversiones en el desarrollo y la investigación de la robótica móvil.



## 1.4 SISTEMAS HOLONÓMICOS Y NO HOLONÓMICOS

Se dice que un robot es holonómico cuando el número de grados de libertad que posee es igual al número de grados de libertad que se puede controlar, además se puede expresar su cinemática (posición y velocidad), de cualquier punto del robot mediante una expresión de igualdad que contiene todas las variables del sistema (partes móviles).

Por ejemplo en la Figura 1.5, se muestra un vehículo omnidireccional que no presenta restricciones en su movimiento, ya que puede hacerlo en cualquier dirección.



Figura 1.5 Robot Holonómico, tomado de [8]

Un robot es no holonómico cuando el número de grados de libertad que posee es mayor al número de grados de libertad controlables. Además poseen restricciones las cuales pueden ser expresadas mediante ecuaciones no integrables, por lo cual el análisis de su dinámica, cinemática y control son más complejos; dichas restricciones son originadas debido a dos causas:

La primera es cuando dos superficies al encontrarse en contacto, la una se mueve sobre la otra sin producir deslizamiento, esto sucede en los robot móviles y manos manipuladoras.

La segunda es la conservación del momento angular total del sistema, esto sucede en robots saltadores en su fase de apoyo y robots espaciales.

Si se califica rigurosamente a los robots se podría decir que todos son sistemas no holonómicos, inclusive los brazos manipuladores, se puede afirmar esto, ya que, por ejemplo las articulaciones de los brazos manipuladores están limitadas a realizar movimientos en un cierto rango de valores, o en su defecto las velocidades se encuentran limitadas a valores máximos (límites). Al representar éstas restricciones matemáticamente mediante desigualdades, éstas son no holonómicas.

Un ejemplo más evidente de no holonómico, es el de un automóvil, el cual presenta restricciones de movimiento porque solo se puede desplazar en una dirección a la vez. (Figura 1.6)



Figura 1.6 Robot No holonómico, tomado de [9]

## **1.5 CONFIGURACIONES TÍPICAS PARA ROBOTS MÓVILES CON RUEDAS [10]**

La elección del tipo de configuración es uno de los pasos necesarios para la construcción de un robot móvil. La configuración hace referencia a la forma en que se encuentran distribuidos los principales elementos que lo componen, como

plataformas, motores, ruedas, etc. Existen distintas configuraciones para robots móviles, las cuales se verán a continuación.

### 1.5.1 CONFIGURACIÓN DIFERENCIAL [11]

Es la más fácil de implementar de todas. Está provista de dos ruedas colocadas en el eje perpendicular a la dirección del robot. Cada una de las ruedas es controlada por un motor independiente, así la orientación del robot queda determinado por la diferencia de las velocidades de las ruedas. La Figura 1.7 muestra esta configuración.

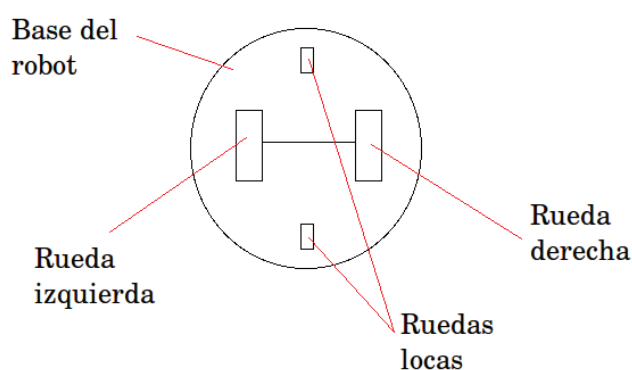


Figura 1.7 Configuración Diferencial, tomado de [11]

Uno de los mayores problemas que tiene éste tipo de configuración (configuración diferencial), es mantener el equilibrio del robot, ya que solo consta de dos ruedas, para corregir esto se le agrega ruedas de libre giro. Dichas ruedas sirven únicamente para mantener al robot de una manera horizontal, ya que no afectan el comportamiento del robot y se acoplan a su movimiento porque giran libremente (ruedas locas).

Esta clase de robots no pueden desenvolverse en terrenos irregulares, su campo de acción se encuentra limitado a superficies planas y sin deformaciones de lo contrario se corre el riesgo de tener pérdidas de tracción y errores en los cálculos de odometría ya que se calcula la distancia desplazada, midiendo el giro de las ruedas. Esto se muestra en la Figura 1.8.

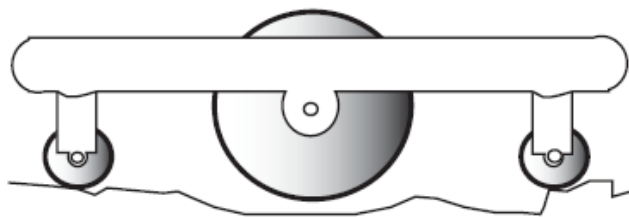


Figura 1.8 Terreno Irregular, tomado de [12]

### 1.5.2 CONFIGURACIÓN TRICICLO [11]

Este tipo de configuración consta de tres ruedas, dos de las cuales son posteriores y sirven únicamente como soporte para la estructura y, una rueda delantera que es la que provee de la dirección, sentido y tracción al robot. En este tipo de configuración no se pueden realizar giros de  $\pm 90^\circ$ .

El radio de curvatura instantáneo del robot, está siempre presente sobre el eje que cruza las dos ruedas posteriores. Un inconveniente común propio de esta configuración, es que en terrenos con inclinación al momento en que el robot se encuentra de subida, el centro de gravedad del mismo, tiende a desplazarse lejos de la rueda delantera lo cual provoca una pérdida de tracción. Ver Figura 1.9.

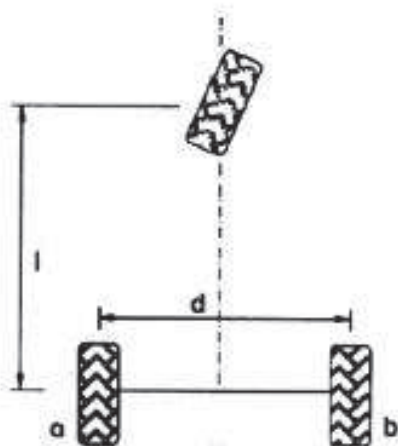


Figura 1.9 Configuración Triciclo, tomado de [12]

### 1.5.3 CONFIGURACIÓN ACKERMAN [11]

Está provista de dos ruedas posteriores con tracción y dos ruedas delanteras para darle la dirección. Esta configuración posee la particularidad de evitar que las ruedas puedan derrapar producto de la irregularidad del terreno (Figura 1.10). Esto es posible gracias a que la rueda delantera interior posee un ángulo de giro  $\theta_i$  un poco mayor que el ángulo de giro de la rueda exterior  $\theta_0$  al momento en que el sistema se encuentra realizando un giro.

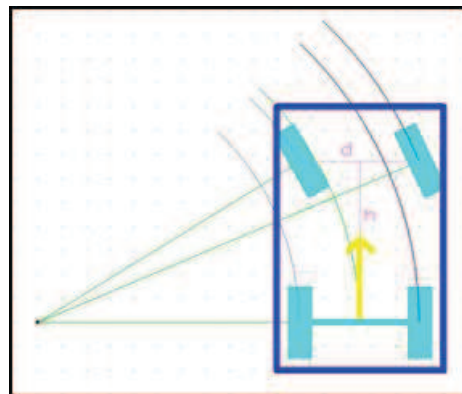


Figura 1.10 Configuración Ackerman tomado de [13]

Se puede concluir que la configuración Ackerman, se ha ganado la reputación de ser un buen sistema que inclusive para terrenos inclinados posee una buena tracción.

### 1.5.4 CONFIGURACIÓN SINCRONIZADA [11]

Cabe mencionar que en este tipo de configuración todas las ruedas se encuentran acopladas mecánicamente y giran a la misma velocidad, es por ello que se la conoce como sincronizada, en el común de los casos puede tener solo tres ruedas para desplazarse, pero también pueden ser más dependiendo de la aplicación. Se puede realizar un control en la rotación de los ejes, por ende en su orientación. (Figura 1.11).



Figura 1.11 Configuración Sincronizada, tomado de [14]

El centro instantáneo de curvatura (ICC<sup>2</sup>), se encuentra en el infinito, lo cual permite un control más simplificado. Al momento de realizar un cambio en la dirección de las ruedas el ICC se ve afectado directamente.

Particularmente este sistema permite mejorar de manera considerable la odometría, ya que al estar acopladas mecánicamente todas las ruedas, éstas generan fuerzas de igual magnitud y paralelas entre sí en cada instante de tiempo, reduciendo de esta manera el deslizamiento de las mismas.

La configuración sincronizada es holonómica, ya que posee dos motores, el primero usado para proveer de movimiento al sistema y el segundo para brindarle una dirección, con esto el robot ya no debe estar girando para cambiar su dirección, solo mover el motor de dirección. Este sistema puede verse en la Figura 1.12.

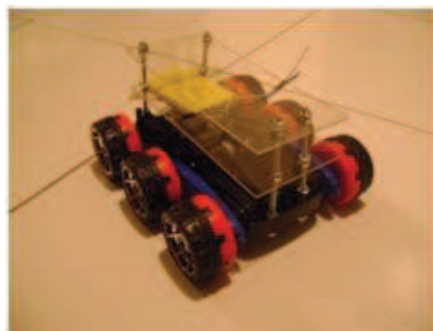


Figura 1.12 Sistema Sincronizado, tomado de [12]

---

<sup>2</sup> Centro Instantáneo de Curvatura: es el punto donde cruzan los ejes de todas las ruedas del robot.

### 1.5.5 CONFIGURACIÓN OMNIDIRECCIONAL

Esta configuración está conformada por un conjunto de ruedas omnidireccionales, (rueda omnidireccional, puede definirse como una rueda común y corriente, a la cual se le ha proporcionado de una corona de rodillos), esto hace que los cálculos de su odometría sean más complicados de realizar, pero a pesar de esto la ventaja es que el robot podrá moverse en cualquier dirección (Figura 1.13).

Los ejes de giro de los rodillos son perpendiculares a la dirección de avance del robot, gracias a esto al aplicarle una fuerza lateral en el sistema, los rodillos giran sobre si mismos permitiendo así que la componente de la velocidad en el eje x no sea nula, dando como resultado que la restricción de no holomicidad sea nula.



Figura 1.13 Configuración Omnidireccional, tomado de [15]

Al estar su estructura definida por una figura de triángulo equilátero y en cuyos vértices se encuentran ruedas de tipo omnidireccionales, su configuración cinemática es más complicada que las demás configuraciones vistas anteriormente. Se debe tener en cuenta que la distancia desde el centro del robot a cualquiera de las ruedas es la misma. El modelo cinemático de esta configuración se muestra en la Figura 1.14.

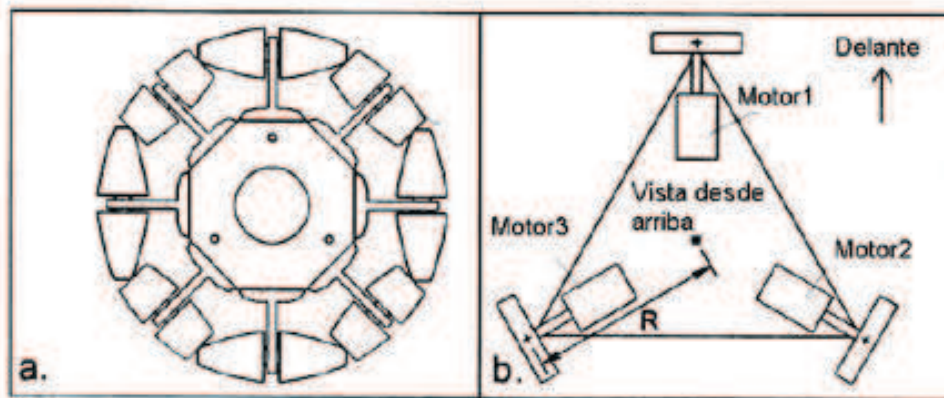


Figura 1.14 Modelo Cinemático, tomado de [12]

### 1.5.6 CONFIGURACIÓN CON MÚLTIPLES GRADOS DE LIBERTAD

Esta configuración está provista de una serie de motores para brindar movimiento y dirección al robot móvil, con esto ayuda mucho a la movilidad del mismo pero empeora el manejo del control. El mayor problema que emerge en este tipo de configuración es el aumento en el deslizamiento de las ruedas, por lo cual se ve afectada la exactitud en su odometría. Otro problema es el lograr controlar de manera coordinada y simultánea a todos los motores. En la Figura 1.15 se muestra este tipo de robot.

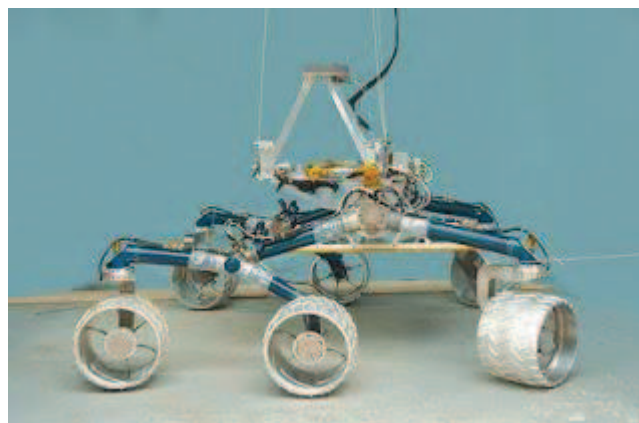


Figura 1.15 Configuración múltiples grados de libertad, tomado de [16]



## 1.6 ROBOT MÓVIL TIPO UNICICLO

El robot tipo unicycle es un sistema móvil con tracción diferencial, es decir, con dos ruedas motoras independientes en el mismo eje y además posee una rueda de giro libre (rueda loca), que le sirve para mantener el equilibrio del sistema. Figura 1.16, el modelo cinemático del sistema es el siguiente:

$$\dot{x} = V * \cos\theta$$

$$\dot{y} = V * \sin\theta \quad \text{Ec 1.1}$$

$$\dot{\theta} = w$$

Donde  $x$  e  $y$  son las coordenadas cartesianas del sistema,  $\theta$  es el ángulo de orientación,  $v$  y  $w$  son las velocidades lineal y angular respectivamente. La restricción no holonómica para este sistema representa la trayectoria tangente a lo largo de cualquier trayectoria real del robot y es representada por:

$$\dot{y} * \cos\theta - \dot{x} * \sin\theta = 0 \quad \text{Ec 1.2}$$

Este tipo de robot es el más elegido al momento de realizar investigaciones para nuevas técnicas de control por la cinemática sencilla que presenta, por su estructura mecánica y su electrónica de control fácil de interpretar. Figura 1.17

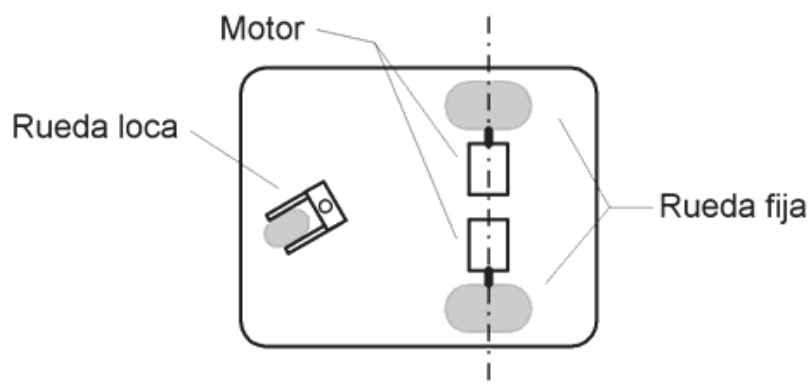


Figura 1.16 Robot Móvil Tipo Unicycle



Figura 1.17 Robot Uniciclo Pioneer, tomado de [17]

El robot uniciclo posee 3 grados de libertad característico del sistema, pero solo se pueden controlar 2 bajo la restricción no holonómica, pero al utilizar el vector  $(x, y, \theta)$ , se demostrará en el capítulo correspondiente al estudio del control que es posible controlar el sistema en posición y orientación.

## 1.7 FPGA (*FIELD PROGRAMMABLE GATE ARRAY*)

### 1.7.1 ARQUITECTURA

La arquitectura de un FPGA (*Field Programmable Gate Array*) consiste en arreglos de varias celdas lógicas las cuales se comunican unas con otras mediante canales de conexiones verticales y horizontales como se muestra en la Figura 1.18.

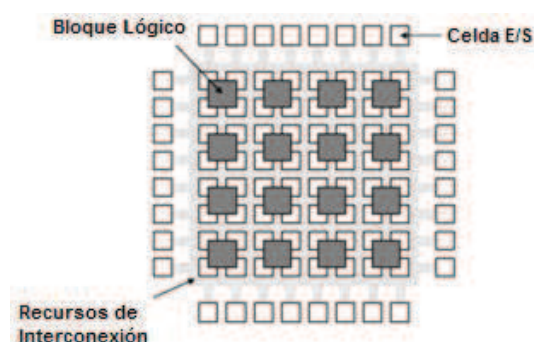


Figura 1.18 Arquitectura FPGA.

Cada celda lógica es similar a los bloques lógicos de un CPLD<sup>3</sup>. La estructura de las celdas lógicas y las formas en que éstas pueden ser interconectadas, tanto salidas como entradas de la celda, varían de acuerdo al fabricante. En general una celda lógica tiene menos funcionalidad que la combinación de sumas de productos y macro-celdas de un CPLD, pero como cada FPGA tienen una gran cantidad de celdas lógicas es posible implementar grandes funciones utilizando varias celdas lógicas en cascada. Además de las celdas lógicas también es importante la tecnología utilizada para crear las conexiones entre los canales; las más importantes son las siguientes:

- Antifuse
- SRAM
- Celdas Lógicas

### 1.7.2 ANTIFUSE

Al igual que la tecnología PROM<sup>4</sup> son programables una sola vez y utilizan algo similar a un fusible para realizar las conexiones, una vez que es programado éste ya no se puede recuperar. Al contrario que un fusible normal, estos antifusibles cuando son programados producen una conexión entre ellos por lo que normalmente se encuentran abiertos, La desventaja es que no son reutilizables pero por el contrario disminuyen considerablemente el tamaño y costo de los dispositivos.

### 1.7.3 SRAM

Celdas SRAM<sup>5</sup> son implementadas como generadores de funciones para simular lógica combinatorial y, además, son usadas para controlar multiplexores e interconectar las celdas lógicas entre sí, es similar a un CPLD.

---

<sup>3</sup> CPLD (del inglés *Complex Programmable Logic Device*) contiene múltiples bloques lógicos, cada uno similar a un PLD (del inglés *Programmable Logic Device*), es decir, un dispositivo que para ser usado debe ser programado de acuerdo a la necesidad.

<sup>4</sup> PROM (del inglés *Programmable Read-Only Memory*) es una memoria solo de lectura.

<sup>5</sup> SRAM (del inglés *Static Random Access Memory*) es una memoria de acceso aleatorio que mantiene los datos mientras está alimentada.

#### 1.7.4 CELDAS LÓGICAS

La estructura de las celdas lógicas se ve fuertemente influida por la tecnología utilizada para fabricar el FPGA. Un FPGA que tiene una gran cantidad de canales de interconexión tiende a tener pequeñas celdas lógicas con muchas entradas y salidas en comparación con el número de compuertas que tiene la celda, este tipo de FPGA's generalmente utilizan tecnología ANTIFUSE<sup>6</sup>.

Un FPGA que tiene una estructura pequeña en canales de interconexión tiende a tener grandes celdas lógicas con pocas entradas y salidas en comparación con el número de compuertas que hay en la celda. Este tipo de FPGA generalmente está hecho con tecnología SRAM.

Una arquitectura con celdas lógicas pequeñas permite utilizar totalmente los recursos del dispositivo. Sin embargo, si las celdas lógicas son demasiado pequeñas entonces se tendrá que utilizar un gran número de éstas en cascada para poder implementar funciones booleanas grandes, lo cual afecta porque cada celda lógica en cascada agrega un tiempo de retardo en la función implementada.

Cuando el tamaño de la celda lógica es grande sucede lo contrario. En este tipo de celdas lógicas es posible utilizar un gran número de compuertas por lo que se puede implementar funciones booleanas de varios términos con pocas celdas lógicas.

El que el tamaño de la celda sea grande no afecta la frecuencia máxima de trabajo porque existe un gran número de compuertas que pueden ser usadas en la función al mismo tiempo, siendo el mismo tiempo de retardo para todas.

En cambio cuando la celda lógica tiene pocas compuertas es necesario utilizar las compuertas de otra celda para poder implementar la misma función y se acumula el tiempo de retardo de las compuertas de la otra celda. Sin embargo, cuando las funciones son pequeñas en comparación con el tamaño de la celda no es

---

<sup>6</sup>ANTIFUSES son elementos que sirven para almacenar información.



## 1.7.5 BENEFICIOS

### 1.7.5.1 Rendimiento

Aprovechando del paralelismo del hardware, los FPGA's exceden la potencia de cómputo de los procesadores digitales de señales (DSPs<sup>7</sup>) rompiendo el paradigma de ejecución secuencial y logrando más en cada ciclo de reloj. El controlar entradas y salidas (E/S) a nivel de hardware ofrece tiempos de respuesta más veloces y funcionalidad especializada que coincide con los requerimientos de una aplicación. [19]

### 1.7.5.2 Precio

El precio de la ingeniería no recurrente de un diseño personalizado ASIC excede considerablemente al de las soluciones de hardware basadas en FPGA.

La misma naturaleza programable del silicio implica que no hay precio de fabricación o largos plazos de ejecución de ensamblado. Los requerimientos de un sistema van cambiando con el tiempo, y el precio de cambiar incrementalmente los diseños FPGA es insignificante al compararlo con el precio de implementar cambios en un ASIC antes de su lanzamiento. [19]

### 1.7.5.3 Fiabilidad

Mientras que las herramientas de software ofrecen un entorno de programación, los circuitos de un FPGA son una implementación segura de la ejecución de un programa. Los sistemas basados en procesadores frecuentemente implican varios niveles de abstracción para auxiliar a programar las tareas y compartir los recursos entre procesos múltiples. Los sistemas basados en procesadores están siempre en riesgo de que sus tareas se obstruyan entre sí. Los FPGAs, que no

---

<sup>7</sup> DSP (del inglés *Digital Signal Processing*) es la manipulación matemática de una señal de información para modificarla o mejorarla en algún sentido.

necesitan sistemas operativos, minimizan estos problemas con ejecución paralela y hardware preciso dedicado a cada tarea. [19]

### 1.7.6 CRITERIOS PARA ESCOGER UN FPGA

Al examinar las especificaciones de un chip FPGA, se observa que generalmente están divididos en bloques de lógica configurables como segmentos o células de lógica, funciones fijas de lógica como multiplicadores, y recursos de memoria como RAM en bloque. El chip FPGA tiene otros componentes, pero éstos son generalmente los más importantes cuando se seleccionan y comparan FPGA's para una aplicación en particular. (Tabla 1.1). [19]

Tabla 1.1 Especificaciones de Recursos FPGA de Varias Familias.

	Virtex-II 1000	Virtex-II 3000	Spartan-3 1000	Spartan-3 2000	Virtex-5 LX30	Virtex-5 LX50	Virtex-5 LX85	Virtex-5 LX110
Compuertas	1 millón	3 millones	1 millón	2 millones	-----	-----	-----	-----
Flip-Flops	10,240	28,672	15,360	40,960	19,200	28,800	51,840	69,120
LUTs	10,240	28,672	15,360	40,960	19,200	28,800	51,840	69,120
Multiplicadores	40	96	24	40	32	48	48	64
RAM en Bloque (kb)	720	1,728	432	720	1,152	1,728	3,456	4,608

### 1.7.7 PROGRAMACIÓN

En el FPGA no se realiza programación tal cual como se realiza en otros dispositivos como DSP, CPLD o microcontroladores. El FPGA tiene celdas que se configuran con una función específica, ya sea como memoria (FLIP-FLOP tipo D), como multiplexor o con una función lógica tipo AND, OR, XOR. La labor del

"programador" es describir el hardware que tendrá el FPGA. Por consiguiente, la tarea del "programador" es definir la función lógica que realizará cada uno de los CLB<sup>8</sup>, seleccionar el modo de trabajo de cada IOB<sup>9</sup> e interconectarlos.

El diseñador cuenta con la ayuda de entornos de desarrollo especializados en el diseño de sistemas a implementarse en un FPGA. Dicho diseño puede ser generado ya sea con un diseño esquemático, o valiéndose de un lenguaje de programación especializado. Estos lenguajes de programación especializados son conocidos como HDL o *Hardware Description Language*. Los HDLs más utilizados son:

- VHDL
- Verilog
- ABEL

En un intento de reducir la complejidad y el tiempo de desarrollo en fases de prototipaje rápido, y para validar un diseño en HDL, existen varias propuestas y niveles de abstracción del diseño. Los niveles de abstracción superior son los funcionales y los niveles de abstracción inferior son los de diseño al nivel de componentes hardware básicos. Entre otras, *National Instruments LabVIEW FPGA* propone un acercamiento de programación gráfica de alto nivel.

[18]

### 1.7.8 APLICACIONES

Cualquier circuito de aplicación específica puede ser implementado en un FPGA, siempre y cuando ésta disponga de los recursos necesarios. Las aplicaciones donde más comúnmente se utilizan los FPGA incluyen a los DSP (procesamiento digital de señales), radio definido por software, sistemas aeroespaciales y de defensa, prototipos de ASICs, sistemas de imágenes para medicina, sistemas de

---

<sup>8</sup>CLB (en inglés *Configurable Logic Block*) proveen acceso a las rutas de conexiones generales y conexiones locales en el FPGA.

<sup>9</sup>IOB (Bloque de entrada-salida) son bloques que permiten su configuración como entrada, salida o ambas para que el FPGA pueda comunicarse.



visión para computadoras, reconocimiento de voz, bioinformática, emulación de hardware de computadora, entre otras. Cabe notar que su uso en otras áreas es cada vez mayor, sobre todo en aquellas aplicaciones que requieren un alto grado de paralelismo.

Existe código fuente disponible (bajo licencia GNU, GPL), de sistemas como microprocesadores, microcontroladores, filtros, módulos de comunicaciones y memorias, entre otros. Estos códigos se llaman cores.

### **1.7.9 TECNOLOGÍA DE LA MEMORIA DE PROGRAMACIÓN**

Los FPGA's utilizan diferentes tecnologías de memoria de programación, las cuales se mencionan a continuación:

- Volátiles: Basadas en RAM. Su programación se pierde al quitar la alimentación. Requieren una memoria externa no volátil para configurarlas al arrancar (antes o durante el reset).
- No Volátiles: Basadas en ROM. Hay de dos tipos, las reprogramables y las no reprogramables.
- Reprogramables: Basadas en EPROM o flash. Éstas se pueden borrar y volver a reprogramar aunque con un límite de unos 10.000 ciclos.
- No Reprogramables: Basadas en fusibles. Solo se pueden programar una vez, lo que las hace poco recomendables para trabajos en laboratorios.

## **CAPITULO 2**

### **DISEÑO DEL SISTEMA ROBÓTICO MÓVIL**

#### **2.1 INTRODUCCIÓN**

En general un robot se halla constituido por una serie de elementos cuyo conjunto permite movimientos para manipular cosas, movimientos rotacionales y desplazamientos en entornos, dicho elementos son los siguientes:

1. Estructura Mecánica
2. Sistemas de Accionamiento
3. Sistemas de Control
4. Sistemas de Sensores
5. Elementos Terminales

En este capítulo se describirán los elementos empleados en un robot y se hará una síntesis de aquellos elementos usados en el diseño del robot de la presente tesis, es decir, se realizará un estudio de la estructura mecánica, haciendo referencia al movimiento que se desea obtener.

Seguidamente se analizarán los sistemas de transmisión y reducción necesarios para permitir el movimiento de cada una de las ruedas del robot generado por los actuadores acoplados a las mismas, después se analizarán los sensores necesarios para la adquisición de los datos del entorno que permitirán guiar y controlar al robot.

#### **2.2 MODELO MATEMÁTICO**

En el presente trabajo se utilizará el modelo cinemático de un robot uniclo, el mismo que a bajas velocidades, de acuerdo a resultados experimentales,

representa una muy buena aproximación del modelo del robot, en donde no es necesario tomar en consideración más factores del modelo dinámico, esto gracias a que el error de seguimiento calculado es muy pequeño.

La representación del modelo cinemático [20] está dada por la Ecuación 2.1:

$$\begin{cases} \dot{x} = v \cos \theta \\ \dot{y} = v \sin \theta \\ \dot{\theta} = w \end{cases} \quad \text{Ec 2.1}$$

Este modelo es un sistema no lineal, multivariable, el mismo que, como ya se ha dicho previamente, presenta una restricción no holonómica en su movimiento.

Esto significa que el robot no resbalará en la dirección ortogonal a la del movimiento y no tendrá deslizamientos entre la rueda y el piso en la dirección del movimiento.

La siguiente figura muestra como es el movimiento en el plano de un robot unicycle (Figura 2.1).

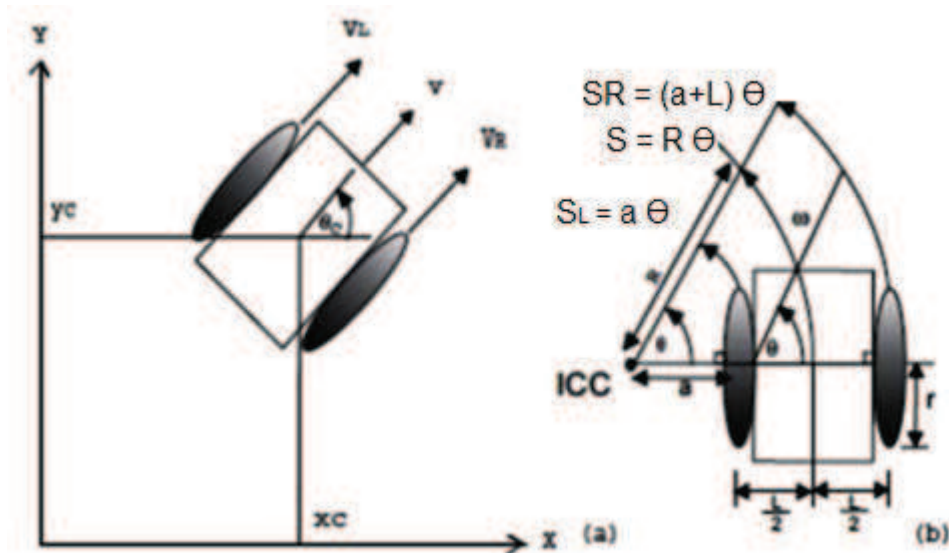


Figura 2.1 Movimiento en el plano robot unicycle, tomado de [20]

### 2.3 DESCRIPCIÓN Y DISEÑO DE LA ESTRUCTURA

En vista que un robot móvil en la mayoría de casos está destinado a simular el comportamiento de seres vivos como personas o animales con un nivel similar de eficiencia, la estructura del robot móvil es bastante semejante a la estructura del ser vivo.

Se puede apreciar una descripción de manera general de la estructura, tanto de un robot móvil como de un ser vivo, en la Figura 2.2.

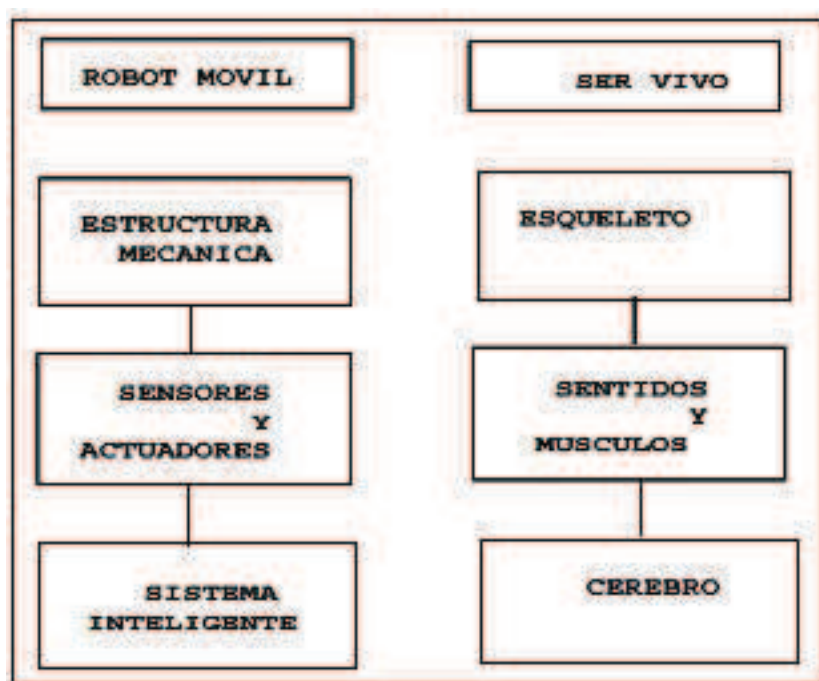


Figura 2.2 Estructura del robot unicycle, tomado de [20]

Se ve entonces que la estructura de un robot, en general, está conformada por diferentes subestructuras, tales como:

**Estructura mecánica:** Pueden ser con ruedas, con patas u orugas.

**Actuadores:** Son elementos que permiten relacionarse con el entorno como por ejemplo motores, brazos, ruedas, luces, etc.

**Sensores:** Son elementos que proporcionan información del medio en el cual está interactuando el robot como por ejemplo encoders, sonares, lázers, cámaras.

**Sistema Inteligente:** Este es el medio mediante el cual el robot interactúa con el medio a partir de toda la información que brinden los sensores, para esto se emplean métodos, algoritmos, etc.

### 2.3.1 ESTRUCTURA MECÁNICA

De lo descrito en el punto anterior, el robot a ser construido para este trabajo, necesita una plataforma a las cuales se acoplarán las ruedas que le darán movilidad al mismo, además al tener estas ruedas un solo punto de contacto con el piso, se necesitará de una rueda castor (rueda de giro libre) que solamente brinde estabilidad al sistema y no interfiera con el comportamiento del mismo.

En la Figura 2.3 se puede observar cómo se vería un sistema de este tipo sin una rueda de estabilidad.



Figura 2.3 Robot unicycle sin rueda estabilizadora, tomado de [20]

Es por ello que se colocará en la parte frontal del robot una rueda que brinde la estabilidad al sistema y se oriente de acuerdo al movimiento mismo del robot. Esto permitirá que el sistema tenga tres puntos de apoyo y no oscile al ponerse en movimiento. Por lo que, se tendría un sistema como el de la Figura 2.4.

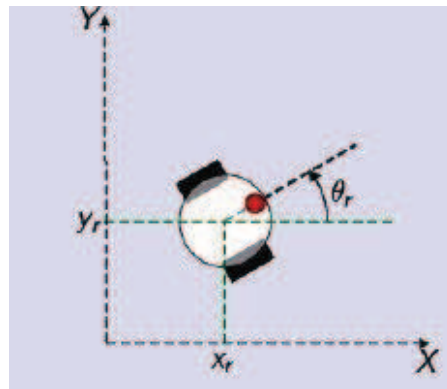


Figura 2.4 Sistema robot uniciclo con rueda de giro libre.

Como la tarjeta que se usa en este trabajo es de un tamaño considerable, se colocará una placa más sobre la que se sostienen las ruedas para poder ubicar el sistema de potencia (baterías, reguladores, etc.) y etapas de control que sean necesarias para complementar el trabajo de la tarjeta principal.

### 2.3.2 RUEDAS

Las ruedas son el elemento que proporciona la capacidad de movilidad en un robot móvil. Es por ello que se debe considerar un tamaño y ubicación adecuada de las ruedas para que éstas no introduzcan perturbaciones en el sistema y simplemente formen parte del sistema motor del robot. (Figura 2.5)



Figura 2.5 Rueda Fija, tomado de [21]

Bajo esta consideración, las llantas serán colocadas en la zona lateral del robot de tal manera que se ubiquen paralelas al avance del robot y, además, que el centro de masa se ubique en el eje que une a las mismas para que no existan saltos al arrancar o detenerse y la trayectoria se realice de manera uniforme.

Se selecciona un tipo de rueda cuya estructura es rígida, para garantizar que no existan perturbaciones (vibraciones) en la recolección de datos de los encoders y evitar deformaciones en la llanta producidas por el peso de la estructura. De esta manera se mantienen constantes los valores de diámetro y distancia entre las ruedas, ya que estos son factores cruciales para realizar el control del sistema.

### 2.3.3 SENSORES

Es el elemento que le informa al robot de la presencia de ciertas características que posee el entorno por el cual se desplaza. Dicho de otra manera, los sensores son dispositivos que le permiten a un robot percibir el medio por el cual se desenvuelven.

Un sensor por definición es un transductor que convierte a un determinado fenómeno físico en señales eléctricas que al ingresar en el microprocesador del robot pueden ser procesadas. Hoy en día hay una gran variedad y muchas clases y su uso depende exclusivamente de la aplicación en la cual se los va a utilizar. (Figura 2.6).



Figura 2.6 Tipos de Sensores, tomado de [22]

En el presente trabajo se desea conocer la distancia que es recorrida por el robot para poder computar y estimar su posición y orientación, por lo que se habla de odometría [23], que es el estudio de la estimación de la posición de vehículos con ruedas durante la navegación utilizando la información sobre la rotación de las ruedas, por lo que se utiliza encoders como medio para obtener la información sobre el giro de las ruedas del robot.

Considerando que el robot va a desenvolverse en un medio conocido, libre de obstáculos y sobre una superficie plana sobre la cual no existirá derrape en las ruedas, los encoders serán los únicos sensores que se utilizará en este trabajo como medio para conocer el entorno y obtener las señales necesarias para el control del robot.

#### **2.3.3.1 Los encoders**

Son elementos electrónicos que sirven para medir posición angular o lineal dependiendo del caso, dependiendo de su tipo pueden ser ópticos o magnéticos. Los primeros constan de un elemento circular dividido en huecos o entradas opacas seguidas de transparentes de la misma dimensión, además de un elemento que les provee de luz como por ejemplo un par de diodos, fotodiodos y un fototransistor que actúa como foto receptor. En el segundo caso el encoder posee un sistema de detección magnética sin contacto, es decir, el disco posee una serie de dientes que generan campos magnéticos y con esto una señal senoidal, la lógica interna del encoder transforma esta señal en cuadrada o triangular lista para ser usada.

El encoder óptico basa su funcionamiento en el giro que realiza, es decir, cada vez que el haz de luz emitido por el fotodiodo choca con un segmento opaco el encoder da una salida o estado de cero lógico por ende cero voltios, en cambio si el haz de luz va por una ranura transparente se marca un uno lógico y se obtiene a la salida el valor de voltaje al cual está alimentado el encoder, éste valor puede ser de 3.3 o 5 voltios comúnmente dependiendo de la aplicación. Al cambio de



estado que se produce se le conoce como pulso, el número de pulsos queda determinado por la velocidad de giro del disco del encoder, es decir, a mayor velocidad de giro, mayor número de pulsos. En la Figura 2.7 se muestra el esquema funcional del encoder.

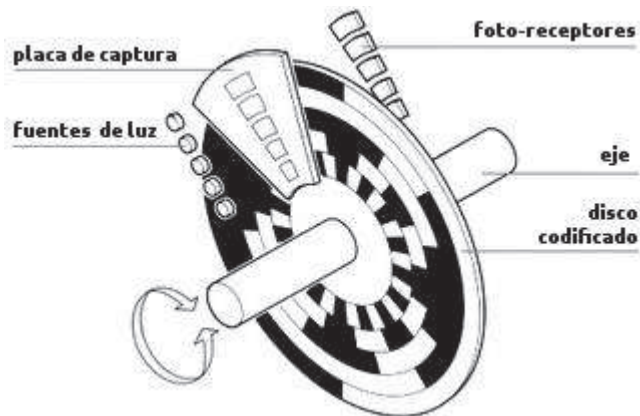


Figura 2.7 Esquema mecánico de un encoder incremental.

El encoder incremental presenta una serie de limitaciones o desventajas como por ejemplo:

- Necesita tener un circuito para acoplar su salida con la entrada de un microcontrolador, esto se lo puede realizar mediante circuitos o programas.
- No pueden tener desajustes mecánicos, ya que al ser elementos de precisión si existe algún tipo de vibración o movimientos bruscos, los valores que provee serán erróneos y se verá afectado el sistema en el cual opera.
- El mismo hecho de que es un elemento de precisión, no permite cambios en su estructura y su precisión queda limitada por el número de franjas o hendiduras que posee. Esto se lo puede apreciar de mejor manera en la Figura 2.8.

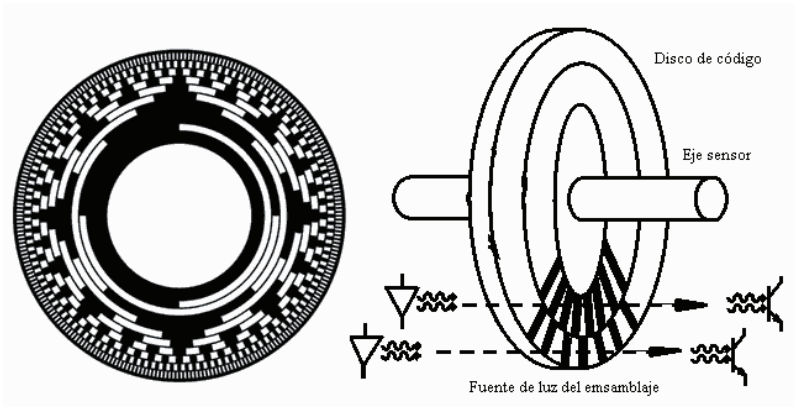


Figura 2.8 Funcionamiento de un encoder.

Dicha precisión está afectada por una serie de factores mecánicos y eléctricos entre los cuales se encuentran:

- El error de división del retículo
- La excentricidad del disco
- La excentricidad de los rodamientos
- El error introducido por la electrónica de lectura
- Imprecisiones de tipo óptico

La unidad de medida para definirla precisión de un encoder es el grado eléctrico.

El grado eléctrico determina la división de un impulso generado por el encoder, los 360 grados eléctricos corresponden a la rotación mecánica del eje, necesaria para hacer que se realice un ciclo o impulso completo de la señal de salida.

Un factor primordial a la hora de elegir el encoder es el número de pulsos que presenta, ya que, un mayor número brindará un control más fino del sistema, evitando de ésta manera errores de posición.

La señal que brinda el encoder es de tipo senoidal, con una amplitud de 1.3 voltios, lo cual representa un problema, ya que la tarjeta de control trabaja con voltajes de 3.3 voltios. Para solucionar este inconveniente se implementa un circuito adicional presentado en la Figura 2.9.

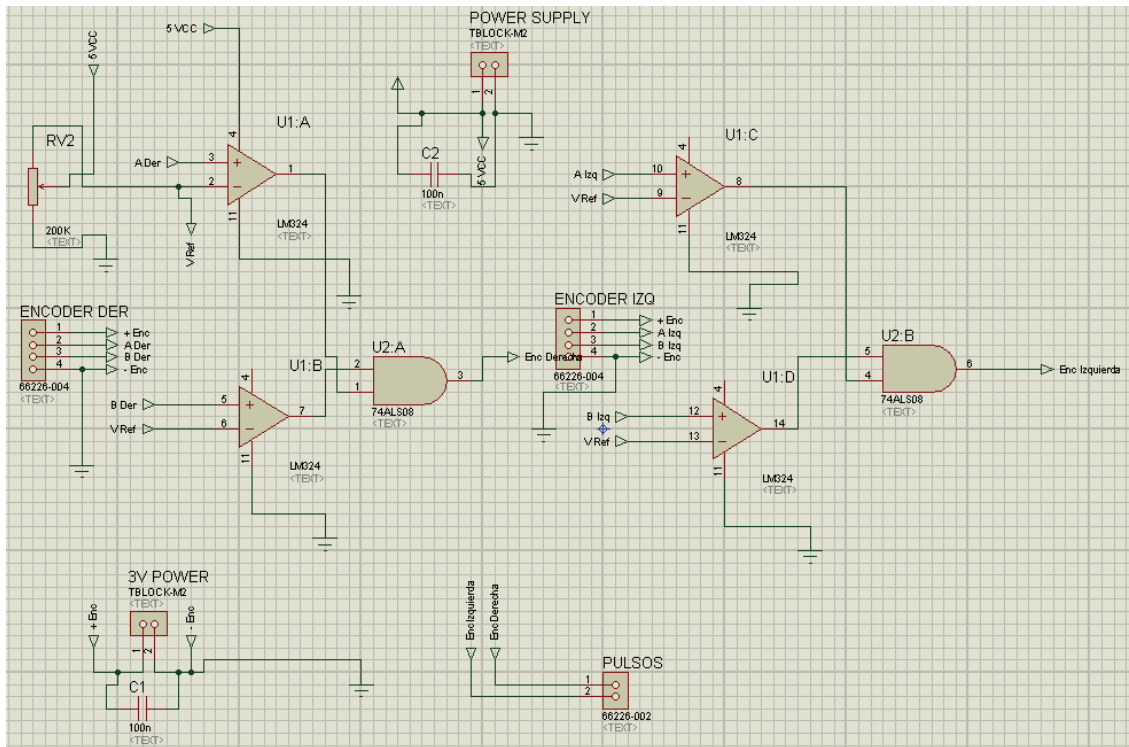


Figura 2.9 Circuito comparador 1v para mejorar los pulsos de los encoders

El cual consta de:

- Amplificador operacional Lm324, el cual compara las señales de los canales A y B del encoder con 1 voltio, obteniendo a la salida una señal cuadrada de amplitud 3.3 voltios y actúa también como filtro para el ruido.
- Compuerta And 74ls08, compara las señales obtenidas en el operacional para obtener un pulso siempre que éstas sean un 1 lógico.
- Zener 3.3V (1N2748A), para regular voltaje.
- Resistencia de 100  $\Omega$ , para limitar la corriente que pasa por el zener.

Como se mencionó anteriormente el encoder presenta baja tolerancia a las vibraciones, por lo que se debe garantizar que los pulsos recolectados sean lo más exactos, por lo que se vio la necesidad de implementar un circuito que procese dichos datos antes de ocuparlos en la lógica de control Figura 2.10.

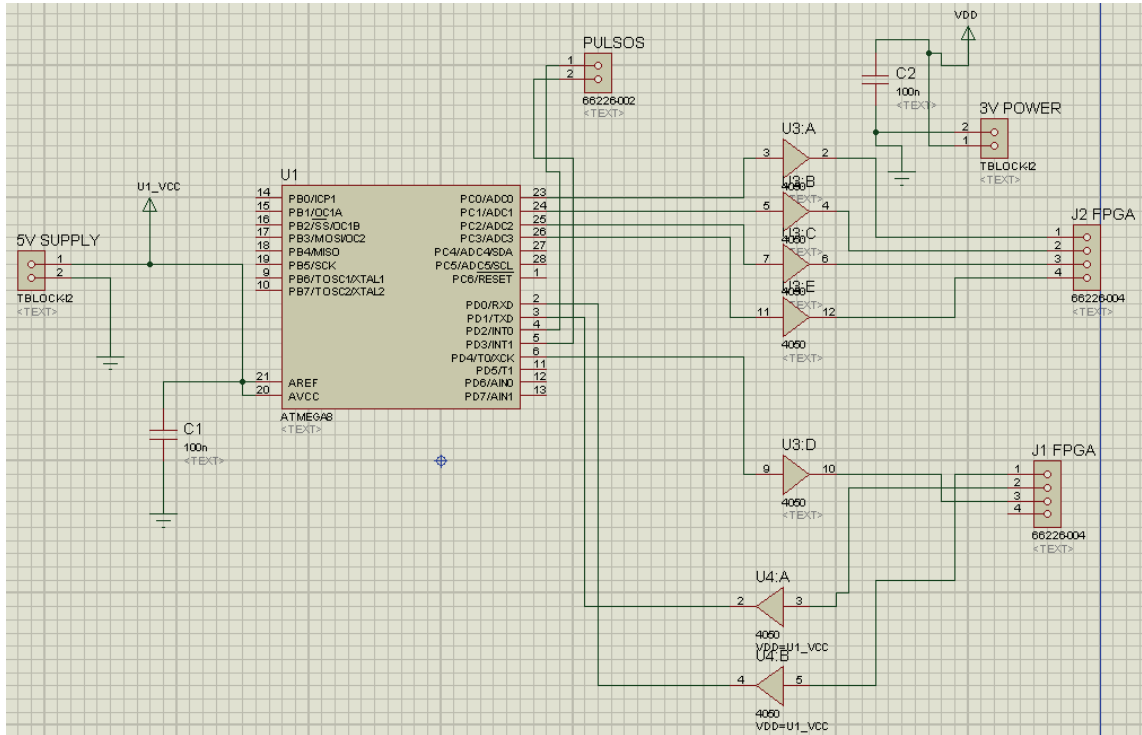


Figura 2.10 Circuito para la recolección de pulsos de los encoders

El circuito consta de los siguientes elementos:

- Microcontrolador atmega16, el cual realiza el proceso de recolección y conteo de los pulsos de los encoders para garantizar una buena toma de datos para ser procesados por la FPGA.
- Compuerta And 74ls08, compara las señales obtenidas en el operacional para obtener un pulso siempre que éstas sean un 1 lógico.
- Diodo 1N4148, para regular voltaje de 5V a 3.3V.
- Resistencias de 4.7 K $\Omega$ , para limitar la corriente que pasa por los diodos.
- Driver CD4050BE, permite aumentar la señal de 3.3 a 5 voltios para acoplarla con el microcontrolador.

### 2.3.4 ACTUADORES

Los actuadores son los elementos que accionan los mecanismos en el robot al ser estimulados dependiendo de la respuesta dada por el sistema ante las entradas de los sensores. Para los robots móviles los actuadores son fundamentalmente debido al medio de locomoción que tienen. (Figura 2.11)

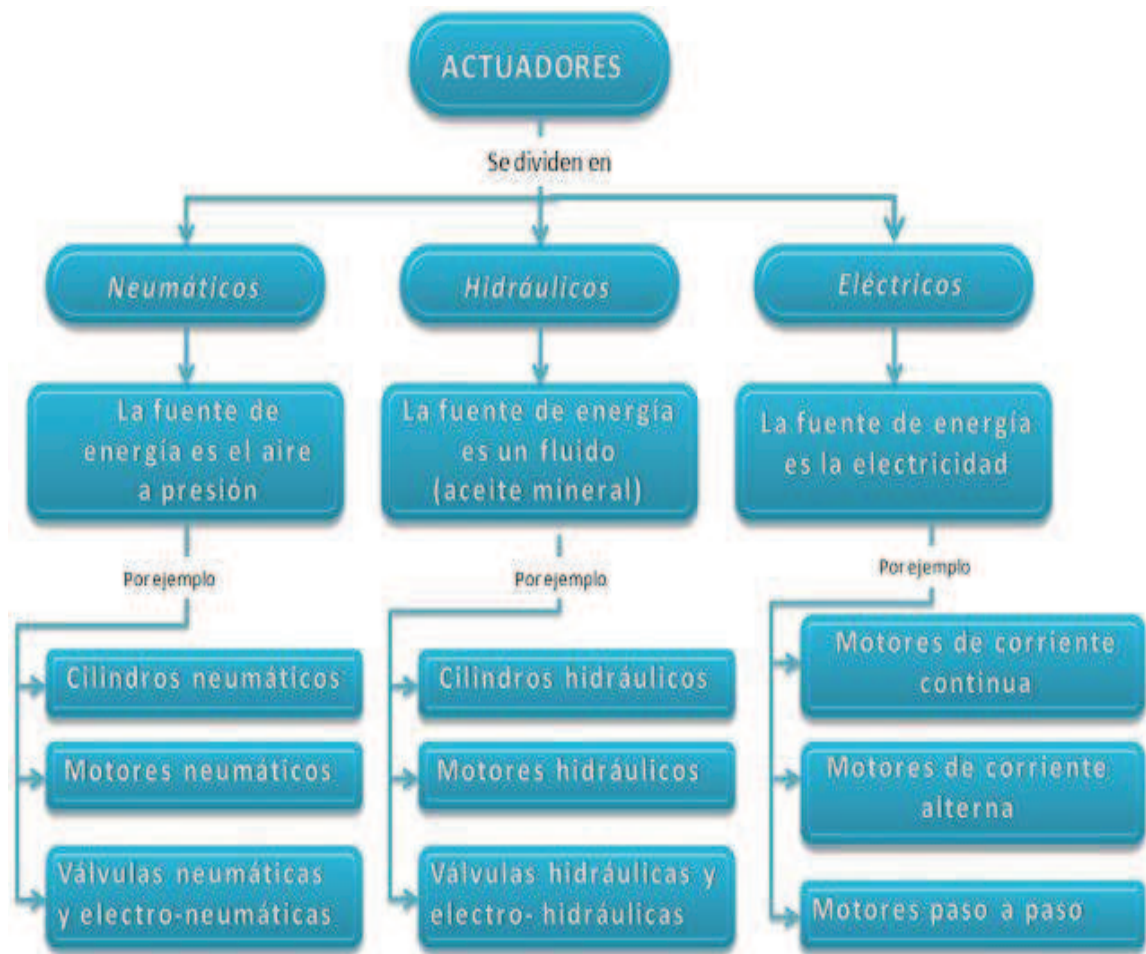


Figura 2.11 Clasificación de los actuadores, tomado de [24]

En este caso, los motores son los actuadores que permiten el giro de las ruedas acopladas a los mismos.

En el presente trabajo se utilizarán motores de corriente continua con caja reductora para permitir el aumento de torque y la reducción de la velocidad.

#### 2.3.4.1 Motor de corriente continua

Son implementados por ser fáciles de usar y son relativamente baratos, más en aplicaciones de velocidad y donde se requiera un control de posición. Están provistos de dos imanes permanentes fijados en la carcasa y una serie de

bobinados de cobre ubicados en el eje del motor, que habitualmente suelen ser tres. Figura 2.12

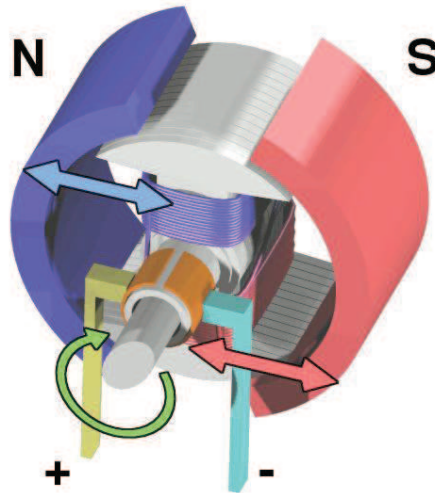


Figura 2.12 Motor de Corriente Continua, tomado de [25]

Son eficientes para girar con poco par y gran velocidad; añadiendo una caja reductora se consigue más par aunque menos velocidad. Son los más simples de los motores y de uso comercial, por lo que resultan económicos. Para controlarlos se suele usar la técnica de puente en H. En la Figura 2.13, puede observarse este tipo de motor.



Figura 2.13 Motor de DC con caja reductora interna, tomado de [21]

En vista que el DAC de la FPGA tiene un voltaje de salida de 2.5 voltios y los motores trabajan a 6 voltios, es necesario implementar dos circuitos adicionales, el primero de los cuales permitirá aumentar dicho valor de voltaje de 2.5 a 5 voltios para la generación de un pwm para cada motor Figura 2.13; el segundo circuito proveerá el giro de los motores mediante un puente H. Figura 2.14

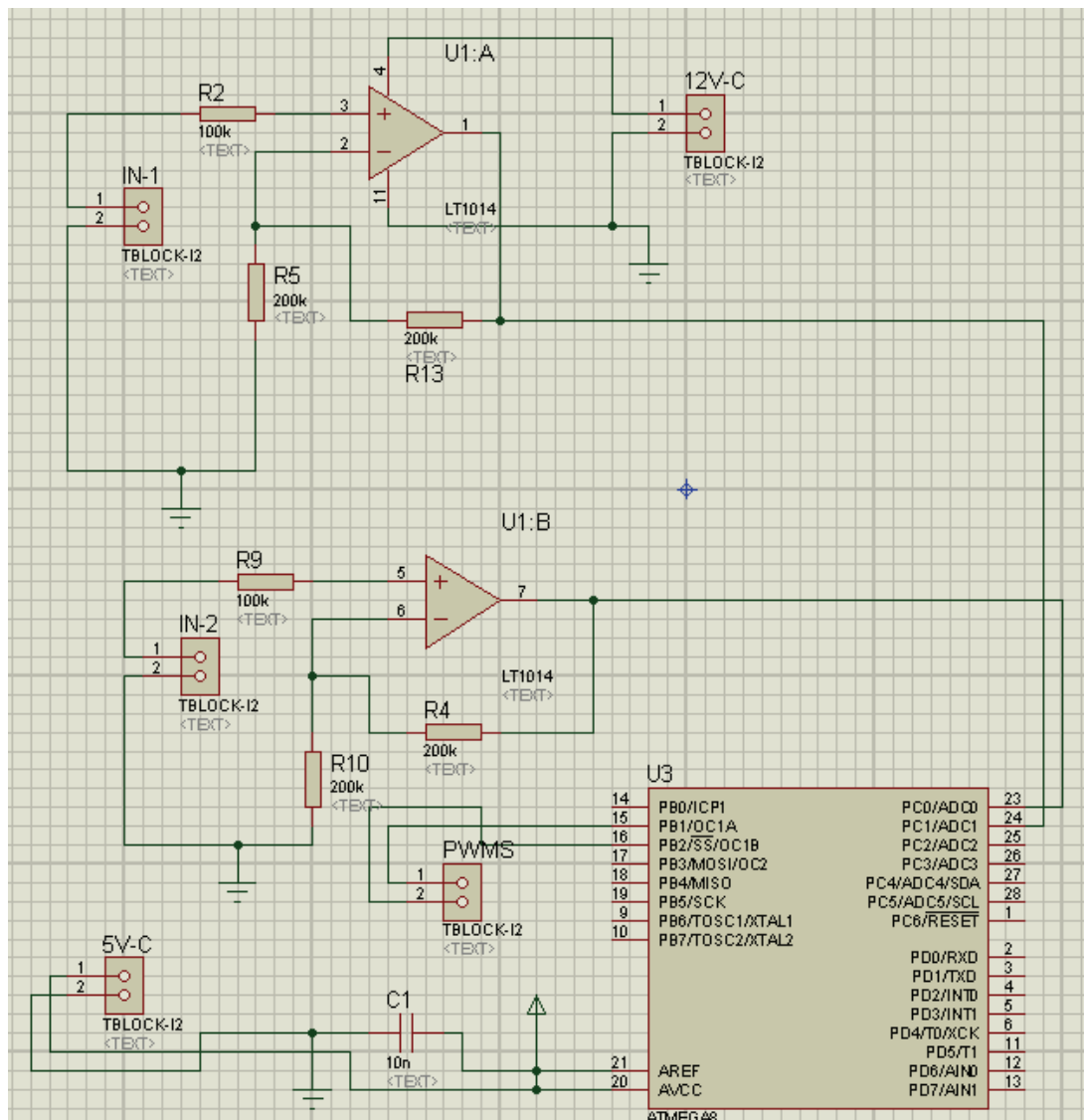


Figura 2.14 Circuito para controlar los motores

Este circuito consta de:

- Amplificador operacional LM324, permitirá realizar el aumento de voltaje mediante una ganancia de valor 2, como se puede ver en la siguiente ecuación:

$$Ganancia = \frac{R2}{R1} + 1 = 2$$

Ec 2.2

- Microcontrolador Atmega8, genera las señales pwm para alimentar a los motores.





### 2.3.5 SISTEMA DE ALIMENTACIÓN CON BATERÍAS

Permite energizar el sistema para que cumpla con las tareas específicas de cada componente del robot, sea movilizarse, recoger los datos generados, procesar los datos, entregar una respuesta al sistema. Poseen en su estructura dos polos, uno positivo y otro negativo, al momento de energizar un sistema, los electrones van del polo negativo al positivo y así se consume su energía.

Se debe diferenciar entre pila y batería, La pila es un elemento que brinda energía mediante un proceso de óxido reducción y su vida útil es muy corta; la batería es un conjunto de pilas que brindan un voltaje por mayor tiempo, es decir, tiene un tiempo de descarga mayor que la pila.

Existen tres tipos comunes de baterías:

- Ni-Cd
- Ni-MH
- Litio

#### 2.3.5.1 Baterías Ni-Cd

Este tipo de baterías están formadas por una placa positiva de hidróxido de níquel y una placa negativa de hidróxido de cadmio, dichas placas están separadas por un electrolito compuesto por una solución acuosa de potasio cáustico, contenido dentro de una estructura porosa, su tiempo de vida útil decae rápidamente y son las más comunes en el uso diario.

Los principales tamaños en baterías de Ni-Cd son:

- AA en capacidades entre 500 y 1200 mA/H, para emisoras y receptores.
- SC en capacidades entre 1300 y 2000 m/AH para tracción en coches eléctricos, ordenadores personales, cámaras de video, teléfonos móviles.
- C, D y E en chispómetros.

La Figura 2.16, muestra las diferentes formas de este tipo de baterías.



Figura 2.16 Baterías Ni-Cd, tomado de [26]

### 2.3.5.2 Baterías Ni-MH

Este tipo de baterías son semejantes a las de níquel cadmio, sólo que en su interior no poseen cadmio, ya que es un elemento caro y a la vez dañino para el medio ambiente. Tienen mayor tiempo de vida útil, almacenan un 30% más de energía que las Ni-Cd, además son recargables.

Para ser recargadas necesitan de un cargador específico, ya que de no tenerlo se dañan con facilidad o su tiempo de vida útil se reduce súbitamente.

A continuación las Figuras 2.17 y 2.18, muestran las diferentes formas de las baterías Ni-MH.



Figura 2.17 Baterías Ni-MH tipo pila recargable, tomado de [27]



Figura 2.18 Baterías Ni-MH tipo pila normal, tomado de [28]

### 2.3.5.3 Baterías de Litio

Este tipo de baterías poseen un elevado grado de densidad de energía, ya que acumulan mayor cantidad de energía por unidad de volumen, esto las hace relativamente más livianas que las de los otros tipos de baterías de las mismas dimensiones, tienen por lo general formas rectangulares y son delgadas.

Presentan una descarga lineal, es decir, que el voltaje que ofrece no varía durante la descarga, por esta razón no necesitan de circuitos reguladores de voltaje. Presentan una descarga paulatina y lenta con relación a las pilas anteriores, es recomendable tenerlas en ambientes secos y a temperaturas no mayores a 15 grados centígrados, no es recomendable descargarlas totalmente, ya que sus celdas se dañan, necesitan de un cargador específico para recargarlas.

Las Figuras 2.19, 2.20 y 2.21, muestran este tipo de baterías.



Figura 2.19 Batería de Litio tipo disco, tomado de [29]



Figura 2.20 Batería de Litio tipo pila cilíndrica, tomado de [30]



Figura 2.21 Batería de Litio, tomado de [31]

#### 2.3.5.4 BATERÍAS Tipo Li-Po (polímero de litio)

Son una variación de las Baterías Litio-Ion (Li-ion).

Son baterías que presentan un tiempo de descarga mayor al de las baterías vistas anteriormente. Tienen una densidad de energía de entre 5 y 12 veces las de Ni-Cd o las de Ni-MH. Son baterías mucho más ligeras y que pueden adoptar cualquier forma. No poseen efecto memoria. El voltaje de cada celda es el mismo y varía dependiendo del voltaje de la batería. (Figura 2.22).



Figura 2.22 Baterías LI-PO.

Debido a los distintos niveles de voltaje que utiliza cada componente del sistema, se debe utilizar reguladores de voltaje para asegurar que la energía suministrada a los mismos sea la adecuada, evitando así daños por sobrecargas eléctricas, por ejemplo la tarjeta SPARTAN 3E FPGA STARTER KIT trabaja con un voltaje de 3.3V o de 2.5V.

Los encoders a su vez, utilizan 3.3V por lo que se debe buscar la manera de acoplar estos voltajes, además las baterías más comúnmente utilizadas entregan un voltaje de alrededor de 12V por lo que se debe regular el voltaje a niveles que no afecten el funcionamiento de los componentes. A continuación se presentan los elementos empleados en los diferentes circuitos acompañados del valor del voltaje de alimentación y operación.

- Tarjeta FPGA Spartan 3E Started Kid, 5V y 3.3V
- Atmega8, 5V
- Atmega16, 5V
- Antenas Xbee, 5V
- Lm324, 5V
- Lm298, 5V
- CD4050BE, 5V y 3.3V
- 74LS08, 5V
- Motores, hasta 12V
- Encoders, 3.3V

En la figura 2.23 se muestra el circuito de alimentación de la parte de potencia, el cual es un regulador de voltaje de 6 voltios.

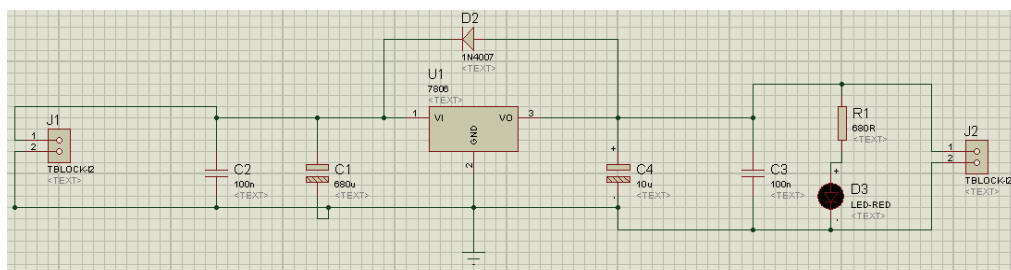


Figura 2.23 Circuito Regulador 6V

En la Figura 2.24 se muestra el circuito de alimentación para la parte de control, el cual consta de cuatro reguladores de voltaje tres de los cuales son a 5 voltios y uno es a 3.3 voltios.

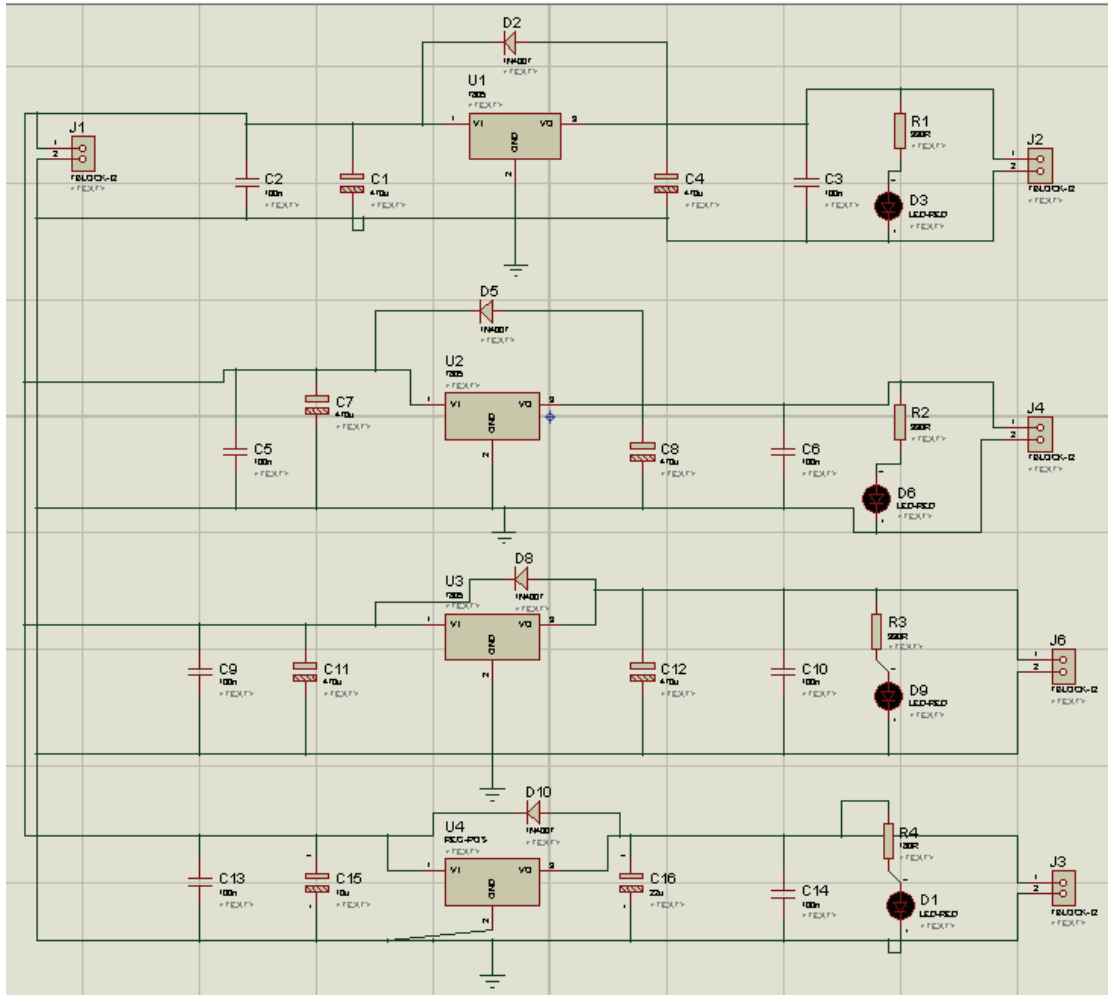


Figura 2.24 Circuito Regulador 5V y 3.3V

La diferencia que presentan estos circuitos es que el primero utiliza el integrado 7806 para regular a 6 voltios y en cambio el segundo el integrado 7805 para regular a 5 voltios y el CJP1117 para regular a 3.3 voltios.

### 2.3.6 SISTEMA INTELIGENTE

En base a lo mencionado en el Capítulo 1, Sección 1.7.6; la elección de la tarjeta FPGA depende de ciertos factores, por lo que para el presente trabajo se emplea como cerebro del robot la tarjeta SPARTAN 3E FPGA STARTER KIT (Figura 2.25), la cual cuenta con todos los componentes necesarios para la adquisición de las señales, el procesamiento de los datos y la respuesta del sistema hacia los motores para generar el movimiento.

Cuenta también con periféricos adicionales que permitirán realizar tareas extras tales como el envío de datos hacia el computador para la visualización del estado del robot y todas las variables. Estas características serán explicadas en detalle en el Capítulo 3.

Además de lo mencionado anteriormente, al ser ésta tarjeta la única que cuenta con el driver para ser conectada con el software LabVIEW 2010, este resultó ser el factor decisivo para la utilización de la misma, ya que la programación de la lógica de control se desarrolló integralmente en el mismo.

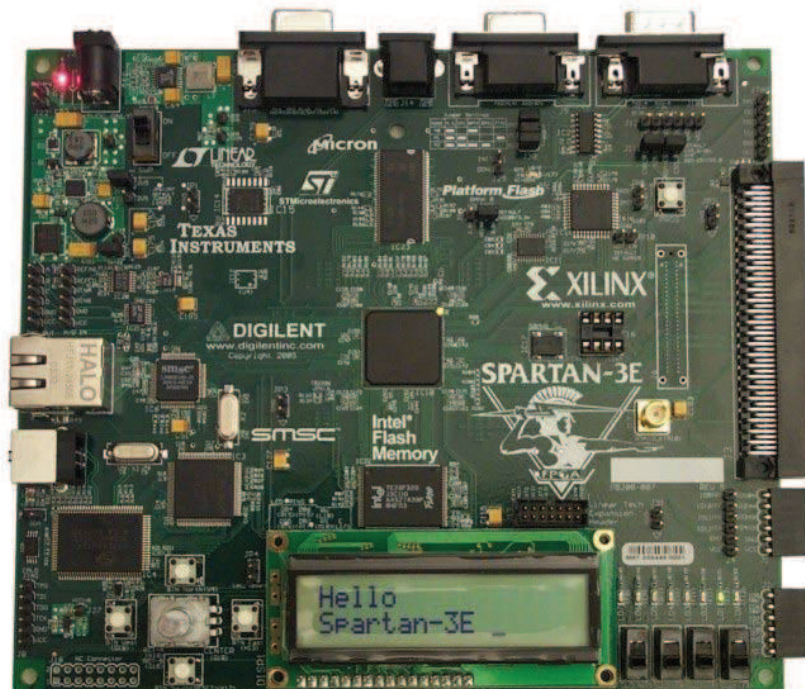


Figura 2.25 Tarjeta 3E-Spartan (500K) FPGA, tomado de [32]

## **CAPITULO 3**

### **DESCRIPCIÓN DE LA ESTRUCTURA**

#### **3.1 INTRODUCCIÓN**

En este capítulo se tratará los temas relacionados con las características físicas del robot, los componentes que conforman su estructura y se realizará una descripción de su funcionamiento, con esto se pretende tener un concepto más claro de cómo opera cada una de sus partes constitutivas, su ubicación y así tener una mejor idea de la manera en que funciona su lógica de control.

#### **3.2 CARACTERÍSTICAS FÍSICAS DEL ROBOT**

Todas las piezas del robot están montadas en una serie de placas de acrílico de 2,5mm de espesor, 17cm de ancho y 20cm de largo, dispuestas de tal manera que forman una estructura de dos pisos cuya altura total es de 16cm; estas placas se encuentran separadas por soportes de aluminio.

El robot móvil presenta una cinemática sencilla, su estructura consta de un par de ruedas fijas alineadas a la misma altura y a la misma distancia del centro de la placa, las mismas tienen un diámetro de 90mm, su estructura es rígida para evitar deformaciones por el peso, estas se encuentran controladas de forma independiente con un motor de DC y un encoder para adquisición de datos por cada una, también posee una rueda loca tipo roll-on para proporcionar estabilidad al sistema.

Este tipo de configuración se la conoce como unicyclo y es la más sencilla de todas las configuraciones de robots móviles.

Cada rueda es controlada por un motor, de tal forma que el giro del robot queda determinado por la diferencia de velocidad de las ruedas. Así, para girar a la



izquierda, hay que darle una velocidad mayor a la rueda derecha. Para girar a la derecha hay que darle una velocidad mayor a la rueda izquierda.

Uno de los problemas que tiene esta configuración es mantener el equilibrio del robot, ya que consta de dos ruedas, por lo cual se le agregó una rueda de libre giro; esta rueda sirve para mantener horizontal al robot, por lo que gira libremente según el movimiento, orientándose hacia la dirección del robot, se agregó una sola rueda de giro libre (rueda loca) para garantizar así la estabilidad del sistema, pero en el caso de ser necesario se pueden agregar más de una. (Figura 3.1).



Figura 3.1 Rueda de giro libre.

Como se señaló antes cada una de las ruedas posee un encoder acoplado a su eje, el cual permitirá conocer la distancia recorrida por cada rueda en determinado intervalo para realizar una corrección y control del recorrido del robot a lo largo de la trayectoria previamente definida en el FPGA y así poder garantizar un desempeño óptimo del sistema y que la trayectoria real recorrida por el robot sea lo más semejante posible a la ya definida.

### 3.2.1 DISPOSICIÓN FÍSICA DE LOS COMPONENTES

En la placa inferior se encuentran acopladas las ruedas, cada una con su respectivo motor de DC con caja reductora y su encoder. Estos elementos se encuentran alineados al mismo nivel del eje de la rueda, es decir, que tanto el eje del motor y el eje del encoder girarán al mismo tiempo conforme al movimiento de la rueda y así se garantizará que no se pierdan datos para determinar la distancia

real recorrida por cada una de las ruedas, ya que es indispensable para el control del desplazamiento del robot móvil. (Figura 3.2.)



Figura 3.2 Disposición física de los encoders. [21]

Para garantizar que los motores permanezcan fijos sobre la placa y su alineación con las ruedas no se altere se emplearon placas de acrílico, tuercas y tornillos que mantienen inmóviles a los motores sin importar la superficie sobre la cual se mueva el robot; Figura 3.3a y la Figura 3.3b.

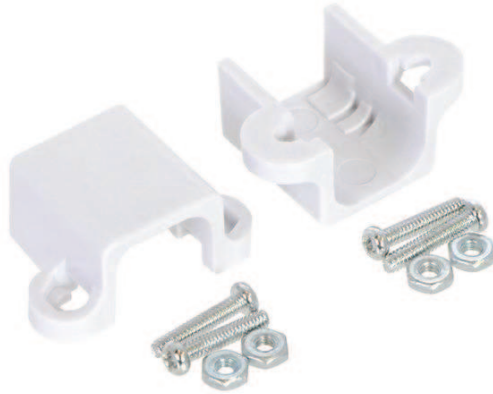


Figura 3.3a Estructuras para el montaje del motor en la placa. [21]



Figura 3.3b Vista frontal del montaje del motor placa inferior.

Además sobre esta placa se encuentra acoplada la rueda de giro libre, para garantizar que el robot esté al mismo nivel en toda su estructura con respecto al piso, se colocó una placa de plástico como base para esta rueda. (Figura 3.4).

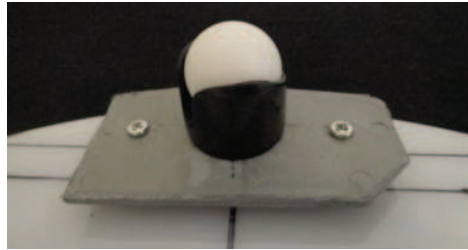


Figura 3.4 Montaje de la rueda de giro libre.

En la Figura 3.5 se puede apreciar la disposición de las ruedas sobre la placa inferior.

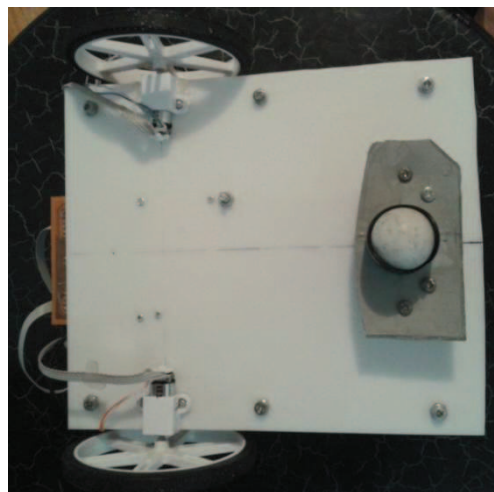


Figura 3.5 Vista posterior de la placa inferior.

Se puede apreciar la estructura completa de la placa inferior en la Figura 3.6, la cual consta del sistema de alimentación (baterías), el regulador de voltaje de 5V y de 3.3V, la placa para la toma de pulsos de los encoders y acondicionamiento de voltaje y las llantas acopladas a su correspondiente motor y encoder.

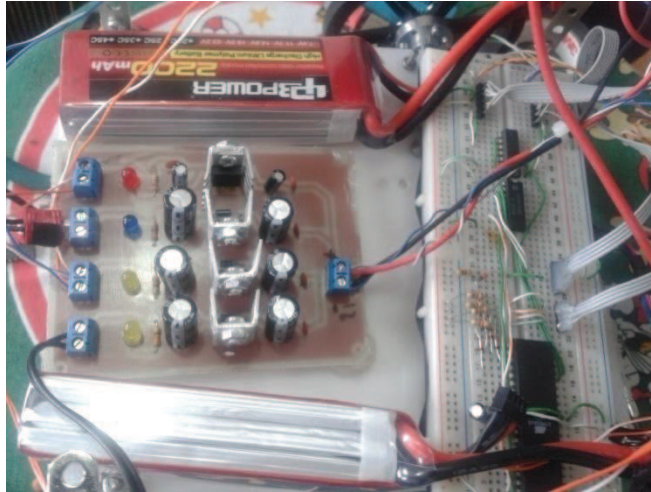


Figura 3.6 Estructura completa placa inferior.

En la placa superior se encuentra localizada la tarjeta SPARTAN 3E FPGA STARTER KIT (500K), la cual permite procesar todos los datos obtenidos por los dispositivos del robot y ejecutar el control del mismo mediante el software y hardware configurado. (Figura 3.7).



Figura 3.7 Disposición física de la tarjeta Spartan 3E placa superior.

También se localiza en esta misma placa una antena Xbee S2, que permite enviar los datos serial y remotamente hacia la computadora para poder procesar sus valores y así poder obtener las representaciones gráficas de las trayectorias realizadas por el robot unicycle (Figura 3.8).

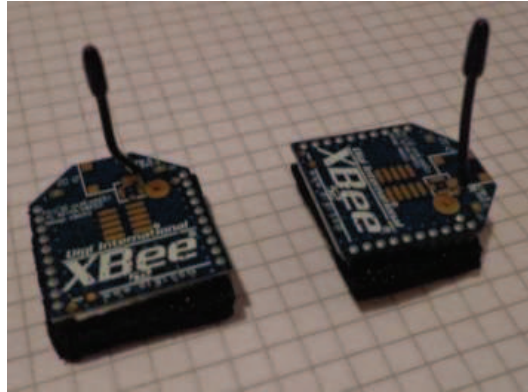


Figura 3.8 Antenas Xbee S2.

Además se cuenta con la presencia de tres placas adicionales, la primera de las cuales permite procesar, acondicionar y aumentar de 0 a 5V el valor que sale del conversor análogo digital del FPGA que tan solo es de 2.5V, dicho valor corresponde a la velocidad de cada rueda del robot, este circuito consta de un Atmega8, el cual permite procesar los datos (Figura 3.9), que hace referencia al circuito de la Figura 2.14.

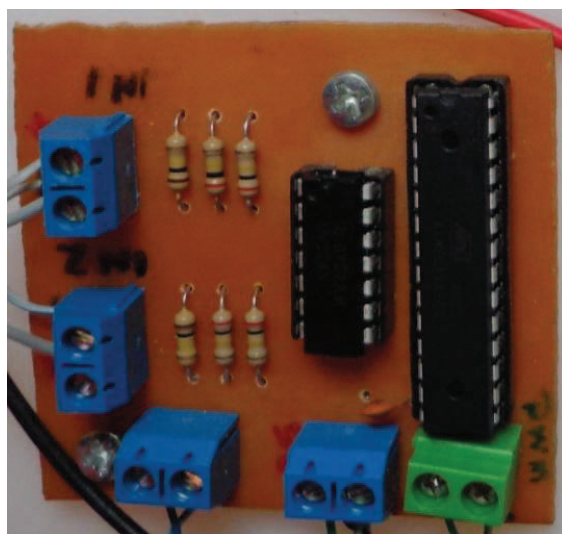


Figura 3.9 Placa acondicionadora de señal.

La segunda placa utiliza el integrado L298, que es un driver que permite acoplar el circuito anterior con las ruedas para variar su velocidad asegurándose que la corriente que necesitan los motores sea la suficiente (Figura 3.10), que hace referencia al circuito de la Figura 2.15.

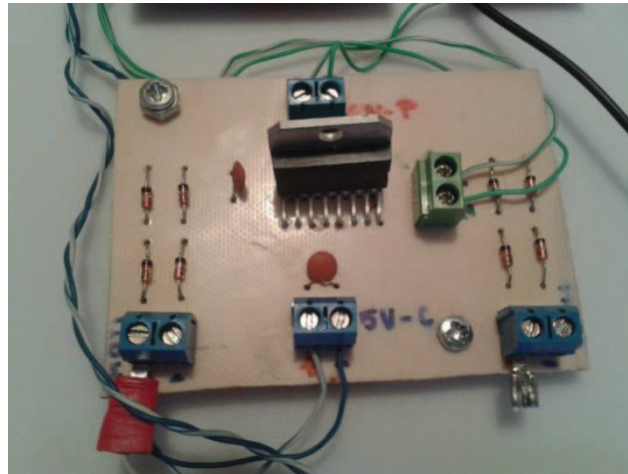


Figura 3.10 Placa de alimentación de motores.

La tercera de las placas es un regulador de voltaje de 6V necesario para el funcionamiento de la placa anterior (Figura 3.11), que hace referencia al circuito de la Figura 2.23.



Figura 3.11 Regulador de voltaje de 6V.

En las Figuras 3.12 y 3.13 se puede apreciar la placa superior completamente armada, la primera figura corresponde a la parte superior de la placa donde se encuentran la tarjeta FPGA y la antena Xbee.

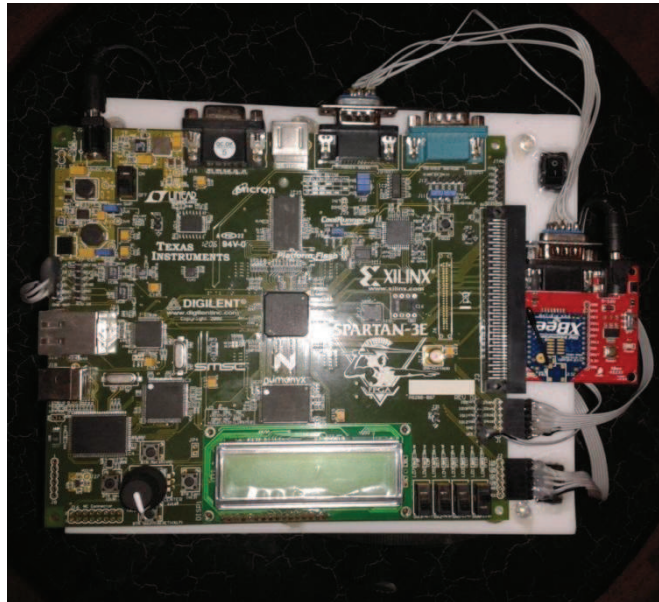


Figura 3.12 Vista superior de la segunda placa.

En la segunda figura se encuentran los tres circuitos explicados anteriormente que se localizan en la parte inferior de la palca.

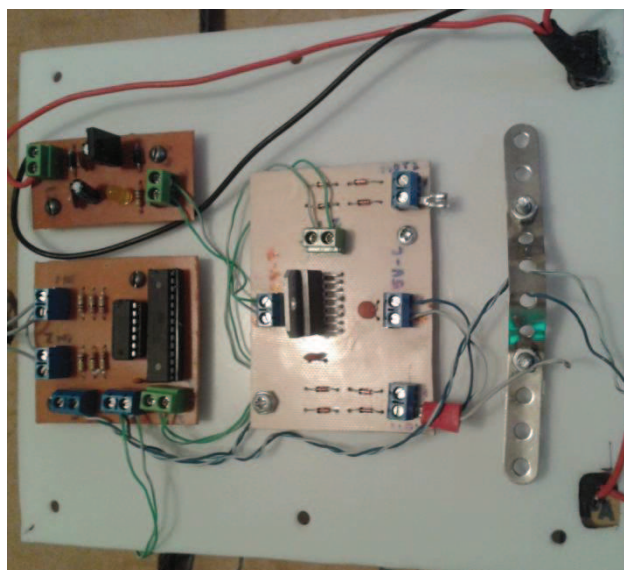


Figura 3.13 Vista inferior de la placa superior.

### 3.3 COMPONENTES DEL SISTEMA ROBÓTICO

#### 3.3.1 SPARTAN 3E FPGA STARTER KIT (500K)

Permiten realizar la adquisición y procesamiento de datos, el control del robot y la comunicación con la computadora para la visualización del comportamiento del sistema. (Figura 3.14).

A continuación se presentan las características y componentes principales del módulo Spartan 3e FPGA starter kit.

Esta tarjeta contiene varios componentes que facilitan el diseño de proyectos, estas características y componentes son [32] :

- Xilinx XC3S500E Spartan-3E FPGA
- Up to 232 user-I/O pins
- 320-pin FBGA package
- Over 10,000 logic cells
- Xilinx 4 Mbit Platform Flash configuration PROM
- Xilinx 64-macrocell XC2C64A CoolRunner™ CPLD
- 64 MByte (512 Mbit) of DDR SDRAM, x16 data interface, 100+ MHz
- 16 MByte (128 Mbit) of parallel NOR Flash (Intel StrataFlash)
- FPGA configuration storage
- MicroBlaze code storage/shadowing
- 16 Mbits of SPI serial Flash (STMicro)
- FPGA configuration storage
- MicroBlaze code shadowing
- 2-line, 16-character LCD screen
- PS/2 mouse or keyboard port
- VGA display port
- 10/100 Ethernet PHY (requires Ethernet MAC in FPGA)
- Two 9-pin RS-232 ports (DTE- and DCE-style)
- On-board USB-based FPGA/CPLD download/debug interface
- 50 MHz clock oscillator



- SHA-1 1-wire serial EEPROM for bitstream copy protection
- Hirose FX2 expansion connector
- Three Digilent 6-pin expansion connectors
- Four-output, SPI-based Digital-to-Analog Converter (DAC)
- Two-input, SPI-based Analog-to-Digital Converter (ADC) with programmable-gain pre-amplifier
- ChipScope™ SoftTouch debugging port
- Rotary-encoder with push-button shaft
- Eight discrete LEDs
- Four slide switches
- Four push-button switches
- SMA clock input
- 8-pin DIP socket for auxiliary clock oscillator



Figura 3.14 Tarjeta Spartan 3E (500K).

### 3.3.1.1 Puerto Serial RS-232 del SPARTAN 3E FPGA STARTER KIT

Como se puede ver en la Figura 3.15 la tarjeta SPARTAN 3E tiene 2 puertos seriales RS-232. En este caso solamente se utilizará el puerto DCE que usa un conector DB9 hembra, el mismo se puede conectar directamente a un

computador ya que la tarjeta incorpora un integrado MAXIM que se encarga de traducir los voltajes TTL o CMOS de la tarjeta a un voltaje apropiado para la comunicación con la computadora.

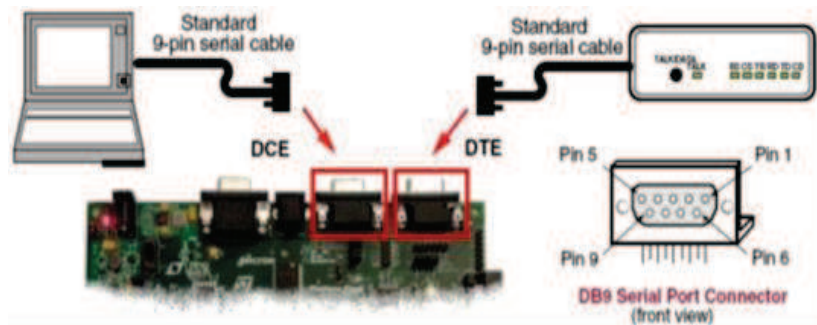


Figura 3.15 Puerto Serial Spartan 3E, tomado de [32]

La tarjeta incorpora resistencias en serie con las salidas del MAXIM como protección ante posibles conflictos de niveles lógicos.

Es importante señalar que no es posible realizar un control de flujo por Hardware ya que como se muestra en la Figura 3.15 los terminales DCD, DTR y DSR están cortocircuitados, de igual manera los terminales RTS y CTS. (Figura 3.16).

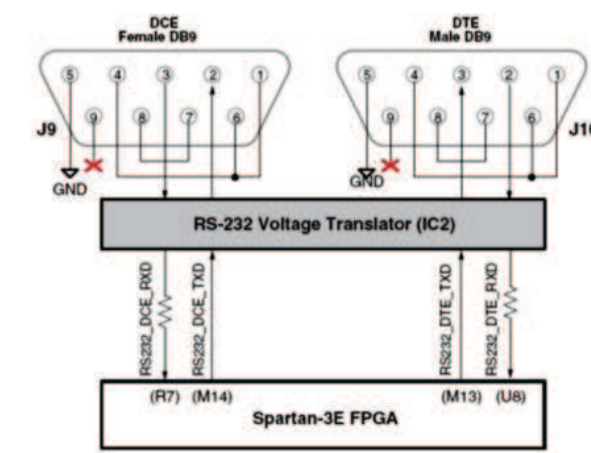


Figura 3.16 Terminales del puerto serial RS232, tomado de [32]

Este puerto serial permite realizar el envío de datos hacia la computadora para visualizar el comportamiento del sistema en todas las instancias.

Permitiendo así conocer la trayectoria ideal o deseada y contrastarla con la trayectoria real que está realizando el robot, al igual que conocer los errores del sistema y la corrección que se realiza a través de la señal de control.

### 3.3.1.2 MODULO XBEE SERIES2 Y COMPLEMENTOS

El módulo Xbee S2 (Figura 3.17), permite realizar la comunicación serial entre la tarjeta Spartan 3e FPGA starter kit (500k) y el computador, esto sirve para poder procesar los datos de la posición del robot unicycle y así graficarlos en el computador con la ayuda del software LabVIEW.

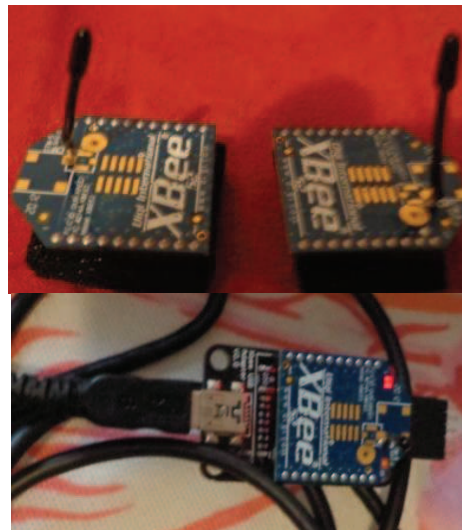


Figura 3.17 Antenas Xbee S2.

Algunas de sus características más importantes se presentan a continuación.

- 3.3V @ 40mA
- 250kbps Max data rate
- Default UART interface: 9600, 8-N-1
- 2mW output (+3dBm)

- 400ft (120m) range
- Built-in antenna
- 6 10-bit ADC input pins
- 8 digital IO pins
- 128-bit encryption
- AT or API command set.

Para lograr su conexión con la tarjeta Spartan 3e FPGA STARTER KIT (500k), se utiliza un conector serial propio para el Xbee S2 llamado Xbee Explorer Serial (Figura 3.18), que permite el acoplamiento y funcionamiento del módulo, ya que al poseer un conector serial al igual que la tarjeta Spartan 3e solo es necesario un cable serial db9 macho-macho para lograr su comunicación. (Figura 3.19).



Figura 3.18 Módulo Xbee Explorer Serial.

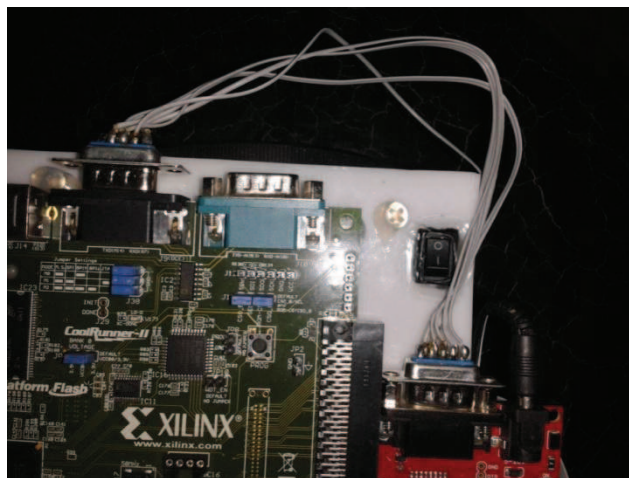


Figura 3.19 Conexión con el puerto serial de la FPGA.

Para lograr transferir los datos desde la tarjeta Spartan 3E a la computadora se requiere que la antena Xbee S2 esté conectada a ésta última también, por lo que se empleó otra antena Xbee con su propio módulo de conexión usb llamado Xbee Explorer USB (Figura 3.20).

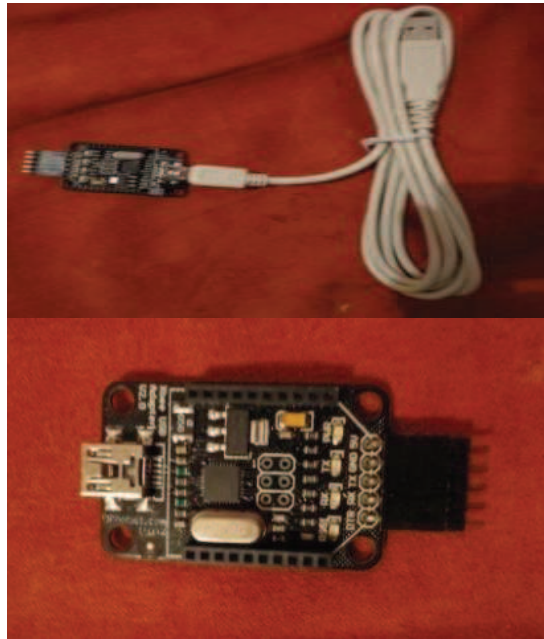


Figura 3.20 Módulo Xbee Explorer USB.

Para lograr la comunicación entre las dos antenas es necesario configurarlas, para ello se emplea el software de DIGI llamado X-CTU. Este software permite mediante comandos AT o en una pantalla seleccionar directamente los valores de las configuraciones para su comunicación.

### 3.3.1.3 Configuración de los módulos XBEE series 2

Haciendo uso del software ya mencionado, es necesario dar a cada antena los parámetros necesarios para la comunicación entre sí. Lo primero que se debe definir es cuál de las dos antenas actuará como coordinador de la comunicación,

en este caso será la antena que se conecta al FPGA. Para ello en la pantalla “*Modem Configuration*” se coloca la opción de “*COORDINATOR AT*”.

La antena que se conecta al computador debe actuar como un router, así en la misma pantalla se la programa como “*ROUTER/END DEVICE*” (Figura 3.21).

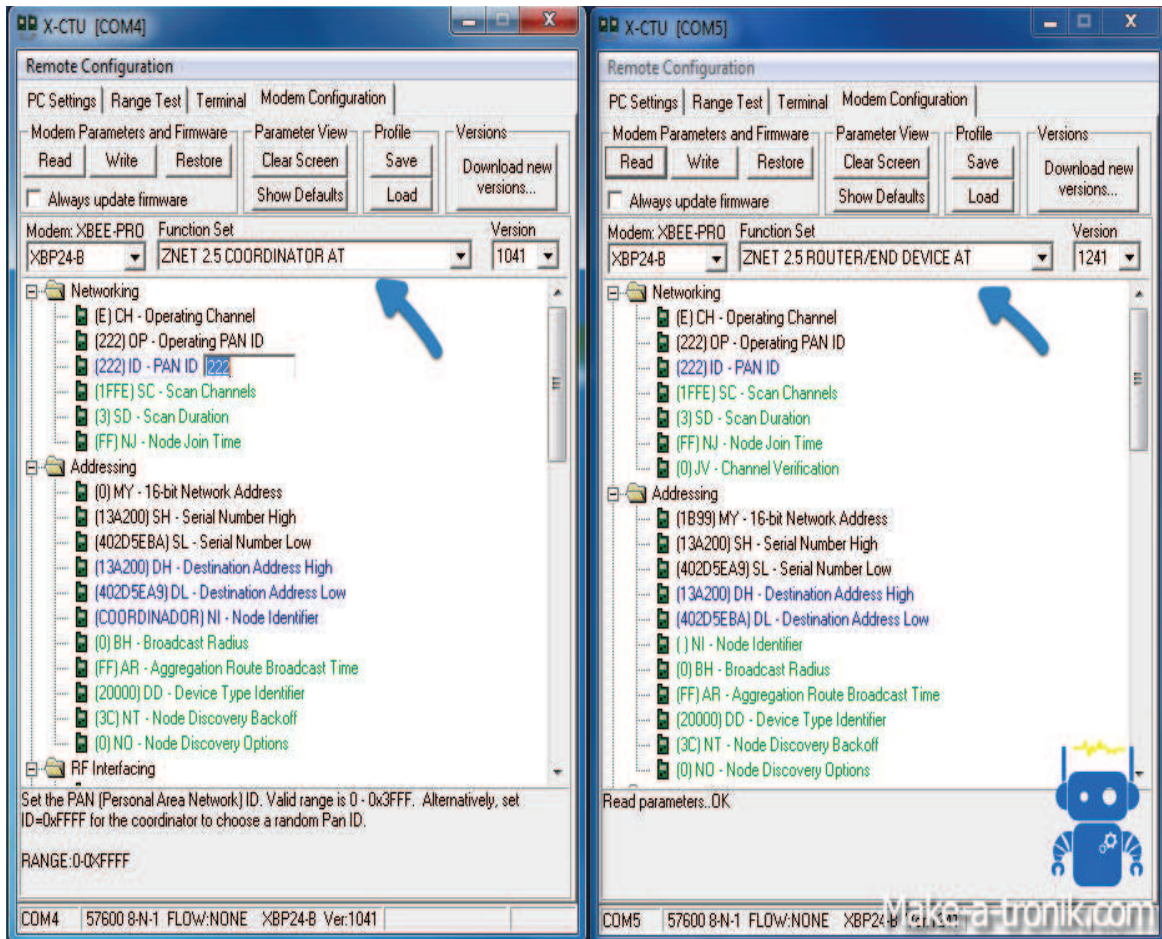


Figura 3.21 *Router/EndDevice*.

Después se debe configurar los direccionamientos, ambos dispositivos deben tener el mismo PAN ID, pero deben apuntarse el uno al otro, para ello en “*Destination Address*” (DH y DL) se escribe el “*Serial Number*” del otro dispositivo (SH y SL) según corresponda, es decir en DH y DL del coordinador se ingresa SH y SL del router y viceversa (Figura 3.22).

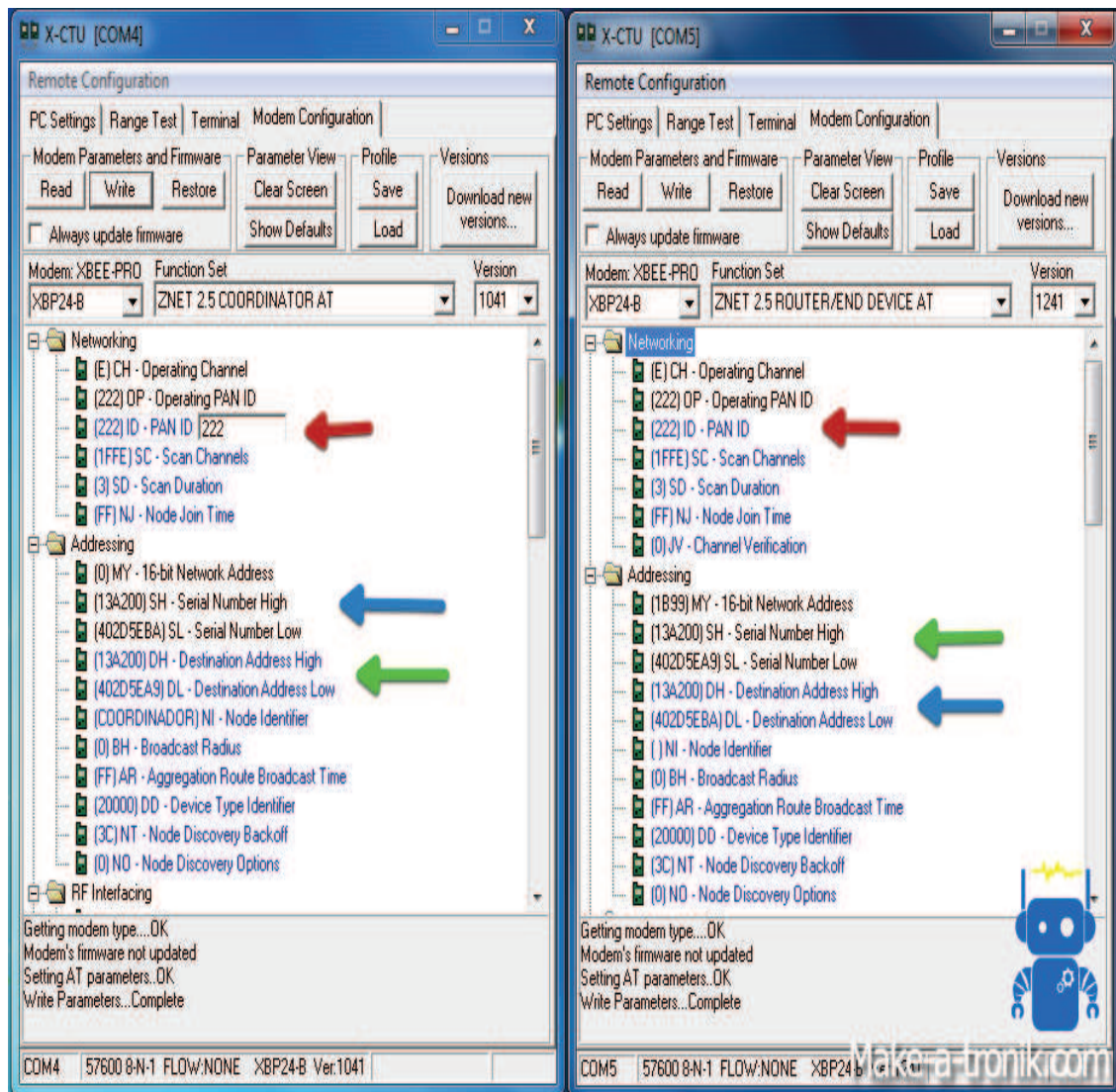


Figura 3.22 Configuración de las antenas Xbee S2.

Finalmente, se da click en *WRITE* para grabar la configuración y las antenas estarán listas para comunicarse entre sí. Para comprobar que las antenas están comunicándose entre sí se puede ir a la pantalla "Terminal" y escribir algo, de esta manera se puede observar que los datos se están enviando y recibiendo de forma correcta.

### 3.3.1.4 Conversor Digital a Análogo (DAC) del Spartan 3E FPGA Starter Kit

La tarjeta Spartan 3E FPGA Starter Kit 3E tiene un convertidor de Digital a Análogo con 4 canales de salida, (A, B, C, D) que se usa para la aplicación del voltaje de control a los motores del robot.

Este DAC se encuentra ubicado en la tarjeta junto al puerto Ethernet como se puede observar en la Figura 3.23, ubicado como conector J5.

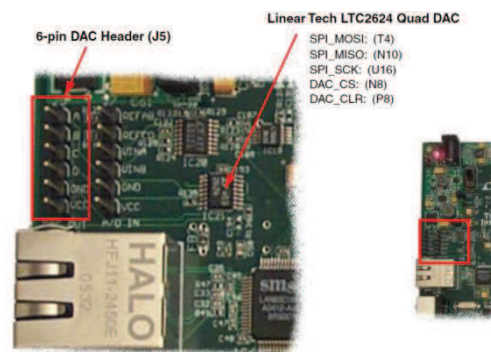


Figura 3.23 Conversor Análogo Digital Tarjeta Spartan 3E, tomado de [32]

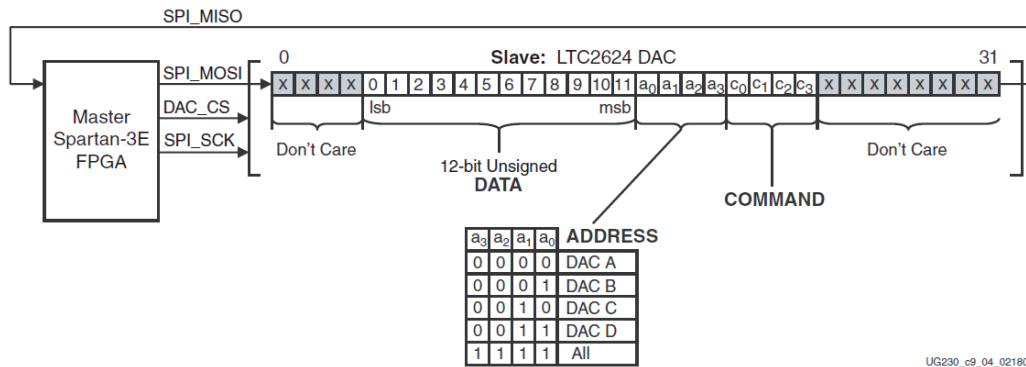
El DAC utiliza la comunicación SPI, misma que es compartida por otros dispositivos en la tarjeta por lo que, para su correcta utilización se debe asegurar que al momento de actualizar datos en el DAC los demás dispositivos que comparten el bus SPI estén desactivados, un cuadro de los dispositivos que se deben desactivar se muestra a continuación. (Tabla 3.1).

Tabla 3.1 Configuración de la DAC Spartan 3E, tomado de [32]

Signal	Disabled Device	Disable Value
SPI_SS_B	SPI serial Flash	1
AMP_CS	Programmable pre-amplifier	1
AD_CONV	Analog-to-Digital Converter (ADC)	0
SF_CE0	StrataFlash Parallel Flash PROM	1
FPGA_INIT_B	Platform Flash PROM	0



Los datos que se deben transmitir para la comunicación con el DAC deben ser de 32 bits, primero el FPGA envía 8 bits “dummy”, seguidos por un comando de 4 bits, que por lo general es el comando [3:0] “0011” que implica que el DAC se actualiza inmediatamente al valor cargado en el mismo, después se envía una dirección (4 bits) que sirve para apuntar al canal que se desea se cargue con el dato, seguido por los 12 bits de datos propiamente dichos y finalmente 4 bits “dummy”. A continuación se muestra una gráfica en donde se puede ver más claramente esta distribución. (Figura 3.24).



UG230\_c9\_04\_021806

Figura 3.24 Trama de datos para transferencia DAC, tomado de [32]

El DAC tiene 2 niveles de salida de voltaje, 3.3V para los canales A y B y 2.5V para los canales C y D. En el presente trabajo se utilizan los canales C y D para el control de los motores, teniendo una relación para la salida del voltaje como se muestra a continuación en la Ecuación 3.1.

$$V_{out\ C,D} = \frac{D[11:0]}{4096} * 2.5\ V \quad \text{Ec 3.1}$$

Donde D[11:0] representa el valor digital de 12 bits cargado en el DAC, el voltaje de referencia tiene una tolerancia de  $\pm 5\%$ , por lo que se podría esperar una variación similar en los voltajes de salida del DAC.

Junto con el DAC de la tarjeta SPARTAN 3E es necesario utilizar una etapa de amplificación externa para conseguir la variación del voltaje aplicado a los motores en el rango que estos necesitan, es decir, de 0 a 6 V.

### 3.3.1.5 ATMEGA8

#### **CARACTERÍSTICAS GENERALES DEL ATMEGA8/8L (Anexo 1) [33]**

Microcontrolador AVR de 8 bit de alto rendimiento y bajo consumo.

##### **Arquitectura RISC avanzada.**

- 130 instrucciones. La mayoría de un simple ciclo de clock de ejecución.
- 32 x 8 registros de trabajo de propósito general.
- Capacidad de procesamiento de unos 16 MIPS a 16 MHz.
- Funcionamiento estático total.
- Multiplicador On-Chip de 2 ciclos

##### **Memorias de programa y de datos no volátiles.**

- 8K bytes de FLASH auto programable en sistema.
  - ⊞ Resistencia: 1.000 ciclos de escritura / borrado.
- Sección de código añadida opcional con bits de bloqueo independientes.
  - ⊞ Programación en sistema con el programa añadido On-Chip.
  - ⊞ Operación de lectura durante la escritura.
- 512 bytes de EEPROM.
  - ⊞ Resistencia: 100.000 ciclos de escritura / borrado.
- 1K bytes de SRAM interna.

- Bloqueo (cerradura) programable para la seguridad del software.

### **Características de los periféricos.**

- Dos Timer/Contadores de 8 bits con prescaler separado y modo comparación.
- Un Timer/Contador de 16 bits con prescaler separado, modo comparación y modo de captura.
- Comparador analógico On-Chip.
- Timer watchdog programable con oscilador separado On-Chip.
- Interfaz serie SPI maestro/esclavo.
- USART serie programable.
- Contador en tiempo real con oscilador separado.
- ADC de 8 canales en los encapsulados TQFP y MLF.
  - 6 canales de 10 bits de precisión.
  - 2 canales de 8 bits de precisión.
- ADC de 6 canales en el encapsulado PDIP.
  - 4 canales de 10 bits de precisión.
  - 2 canales de 8 bits de precisión.
- 3 canales de PWM.
- Interfaz serie de dos hilos orientada a byte.

### **Características especiales del microcontrolador**

- Reset de Power-on y detección de Brown-out programable.
- Oscilador RC interno calibrado.

- Fuentes de interrupción externas e internas.
- 5 modos de descanso: Idle, reducción de ruido ADC, Power-save, Power-down y Standby.

### **I/O y encapsulados**

- 23 líneas de I/O programables.
- PDIP de 28 pines, TQFP y MLF de 32 pines.



Figura 3.25 Integrado Atmega8, tomado de [34]

### **3.3.1.6 ATMEGA16**

#### **CARACTERÍSTICAS GENERALES DEL ATMEGA16 (Anexo 2 ) [35]**

Alto rendimiento, Bajo consumo, microcontrolador de 8 bits

#### **Advanced RISC Architecture**

- 131 Powerful Instructions – Most Single-clock Cycle Execution
- 32 x 8 General Purpose Working Registers
- Fully Static Operation
- Up to 16 MIPS Throughput at 16 MHz
- On-chip 2-cycle Multiplier

### **Nonvolatile Program and Data Memories**

- 16K Bytes of In-System Self-Programmable Flash
- Endurance: 10,000 Write/Erase Cycles
- Optional Boot Code Section with Independent Lock Bits
- In-System Programming by On-chip Boot Program
- True Read-While-Write Operation
- 512 Bytes EEPROM
- Endurance: 100,000 Write/Erase Cycles
- 1K Byte Internal SRAM
- Programming Lock for Software Security

### **JTAG (IEEE std. 1149.1 Compliant) Interface**

- Boundary-scan Capabilities According to the JTAG Standard
- Extensive On-chip Debug Support
- Programming of Flash, EEPROM, Fuses, and Lock Bits through the JTAG Interface

### **Peripheral Features**

- Two 8-bit Timer/Counters with Separate Prescalers and Compare Modes
- One 16-bit Timer/Counter with Separate Prescaler, Compare Mode, and Capture
- Mode
- Real Time Counter with Separate Oscillator
- Four PWM Channels

- 8-channel, 10-bit ADC
  - 8 Single-ended Channels
  - 7 Differential Channels in TQFP Package Only
  - 2 Differential Channels with Programmable Gain at 1x, 10x, or 200x
- Byte-oriented Two-wire Serial Interface
- Programmable Serial USART
- Master/Slave SPI Serial Interface
- Programmable Watchdog Timer with Separate On-chip Oscillator
- On-chip Analog Comparator

### **Special Microcontroller Features**

- Power-on Reset and Programmable Brown-out Detection
- Internal Calibrated RC Oscillator
- External and Internal Interrupt Sources
- Six Sleep Modes: Idle, ADC Noise Reduction, Power-save, Power-down, Standby
- and Extended Standby

### **I/O and Packages**

- 32 Programmable I/O Lines
- 40-pin PDIP, 44-lead TQFP, and 44-pad MLF

### Operating Voltages

- 2.7 - 5.5V for ATmega16L
- 4.5 - 5.5V for ATmega16

### Speed Grades

- 0 - 8 MHz for ATmega16L
- 0 - 16 MHz for ATmega16

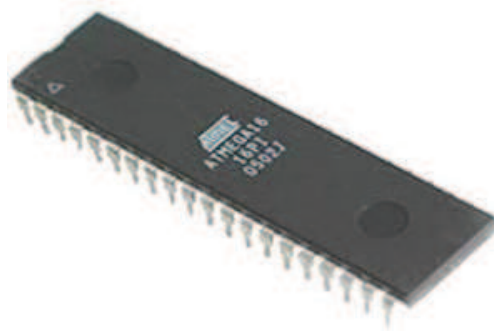


Figura 3.26 Integrado Atmega16, tomado de [35]

#### 3.3.1.7 Conectores de expansión del Spartan 3E fpga Starter Kit

Muy aparte de la versatilidad y gran funcionalidad que presenta la tarjeta SPARTAN 3E FPGA STARTER KIT, la misma permite incorporar componentes externos a través de sus módulos de expansión.

La tarjeta tiene 5 conectores de expansión los cuales se puede apreciar en la Figura 3.27, de estos interesan los conectores de expansión J1 y J2 de 6 pines mediante los cuales se va a incorporar la medición de la distancia recorrida por las llantas del robot a través de encoders incrementales acoplados al giro del motor a través del eje de las llantas.

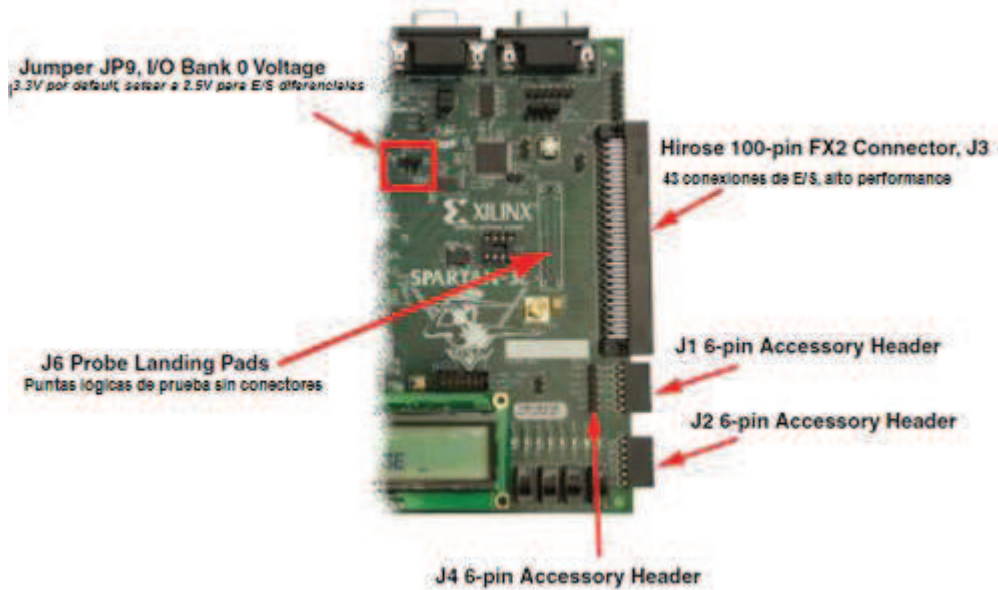


Figura 3.27 Conectores de expansión, tomado de [32]

Los conectores de expansión J1 y J2 de la tarjeta tienen 4 pines I/O más un pin de GND y un PIN de VCC (3,3V). Una distribución de los pines de este conector se muestra a continuación en la Figura 3.28.

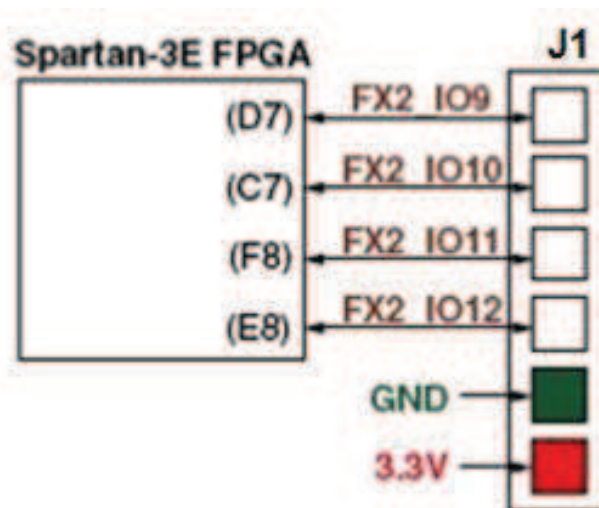


Figura 3.28 Conector de Expansión J1, tomado de [32]



### 3.3.2 ENCODERS INCREMENTALES

Los encoders (Figura 3.29) montados uno en cada rueda del robot son empleados para determinar la distancia recorrida por cada una de las mismas, para de ésta manera conocer la posición en la que se encuentra el robot expresada en coordenadas cartesianas  $(x,y)$ , y la orientación del mismo expresada en un ángulo  $\theta$  medido de forma anti horaria con respecto del eje X.

Una vez obtenido estos datos se da paso al software para que realice los cálculos necesarios para la corrección de posibles errores y la aplicación de la acción de control.

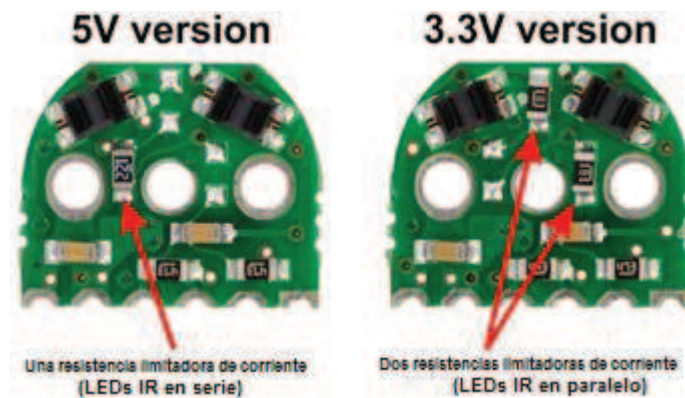


Figura 3.29 Encoder Incremental, tomado de [21]

El encoder presenta las siguientes características [21] :

- Alimentación de 3.3V
- Consumo de corriente: 20 mA máximo
- Frecuencia máx: 100kHz
- Número de pulsos 64.

Al trabajar con encoders se corre el riesgo de tener la presencia de ruido, que puede ser ocasionado por los demás componentes del robot como motores o la alimentación, por eso es indispensable realizar un filtro en base al número de pulsos que tenga el encoder. El filtro puede apreciarse en la Figura 3.30, que hace referencia al circuito de la Figura 2.9 .

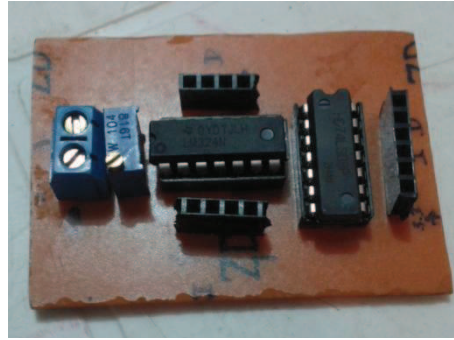


Figura 3.30 Conexión del Encoder.

### 3.3.3 MOTORES DC

Se utilizaron dos motores de corriente continua, uno para cada llanta, por la facilidad de control y de conexión que presentan este tipo de motores, además los motores empleados presentan la particularidad de poseer una caja de reducción; estas cajas de reducción son un sistema de engranajes conectados al eje del motor (Figura 3.31).

Se escogió esta condición para aumentar el torque de arranque y reducir así el esfuerzo inicial (la velocidad de giro del motor), para producir el movimiento del robot, ya que, dependiendo del peso se necesitará más o menos torque y esto afecta directamente a los motores produciendo aumentos en la corriente empleada por los mismos y que pueden ocasionar daños e inclusive la recolección de datos erróneos que afectarán directamente al seguimiento de la trayectoria.

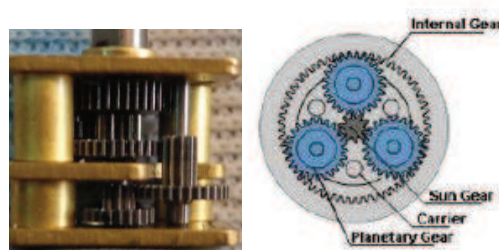


Figura 3.31 Figura de una caja reductora.

Para efectuar el control de posición y orientación en la trayectoria se varía la velocidad de los motores, a través de la DAC de la tarjeta SPARTAN 3E y la etapa de amplificación externa variando el voltaje aplicado a los terminales de cada motor.

Para escoger los motores adecuados para el robot, se tomó en cuenta criterios como el voltaje de alimentación, el torque, el tamaño del motor y el peso de la estructura completamente armada. El motor escogido se puede apreciar en la Figura 3.32.



Figura 3.32 Motor DC con caja reductora y eje extendido, tomado de [21]

Un criterio muy importante de tomar en cuenta al momento de escoger el motor es que los encoders deben estar completamente acoplados con las ruedas y el motor, ya que cualquier vibración por más pequeña que sea produce datos erróneos en el número de pulsos y por ende en el tratamiento de los datos y en el resultado final obtenido, por esta razón los motores utilizados tienen la peculiaridad de tener ya acoplados los encoders como se aprecia en la Figura 3.32.

Características del motor con caja reductora [21]:

- Dimensiones: 0.94" x 0.39" x 0.47", 4.5 x 1 mm eje extendido
- Ratio de la reductora: 298:1
- Diámetro del eje: 3,0 mm (con ranura de bloqueo)
- Voltaje nominal: 6Vcc (puede funcionar entre 3 a 12Vcc)
- Velocidad de giro sin carga: 45rpm

- Consumo sin carga: 30mA free rotor, 360mA max
- Torque: 2.9 kg-cm (máximo)
- Peso: 10 gramos

### 3.3.4 REGULADOR DE VOLTAJE

En vista que las baterías empleadas para la alimentación del sistema proporcionan un voltaje de 11.1V se presentó la necesidad de emplear tres reguladores de voltaje para asegurar que el voltaje que llegue a cada uno de los componentes del robot sea el adecuado, es decir, un máximo de 5V de DC y así evitar fallas o daños en los equipos y también lecturas de datos erróneas que afecten al desarrollo del control en el robot.

Este regulador está compuesto por un integrado L7805CV que permite limitar el voltaje de salida a un máximo de 5V DC a partir de un voltaje de 12V, también consta de tres diodos LN4007 que sirven para prevenir daños en el sistema, ya que, evitan el regreso de la corriente en el mismo, dos capacitores uno cerámico de 0,1 nano faradios (nF) y el otro electrolítico de 470 micro faradios ( $\mu$ F), empleados para tener un voltaje estable, finalmente una resistencia de 330 ohmios ( $\Omega$ ) para descargar los capacitores en el momento en que el sistema se quede sin energía. Figura 3.33, que hace referencia al circuito de la Figura 2.24.

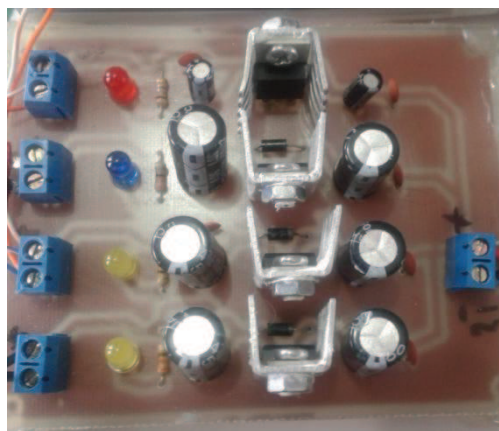


Figura 3.33 Regulador de voltaje a 5V y de 3.3V.

Se utilizó los mismo valores para el otro regulador de voltaje, la única diferencia es que el segundo es de 6V por los cual consta del integrado L7806, para cumplir con dicho voltaje. Figura 3.34, que hace referencia al circuito de la Figura 2.23.



Figura 3.34 Regulador de voltaje a 6 voltios.

### 3.3.5 BUFFER CD4050BE

Dado que la tarjeta SPARTAN 3E trabaja con un voltaje de 3.3V es necesario acoplar los encoders de forma que no existan conflictos en los voltajes utilizados, para cuidar la integridad física de la tarjeta. Es por ello que se usan estos buffers ya que permiten acoplar el voltaje de los encoders (5V) a la tarjeta (3.3V), esto se hace ya que si se conecta los pines del conector J1 de la tarjeta a este integrado, al recibir un 1 lógico por parte del encoder es decir 5V a su salida (que está conectada a la tarjeta) se tiene un 1 lógico pero al nivel de Vref es decir 3.3V. El circuito de este elemento se presenta en la Figura 3.35, que hace referencia al circuito de la Figura 2.10.

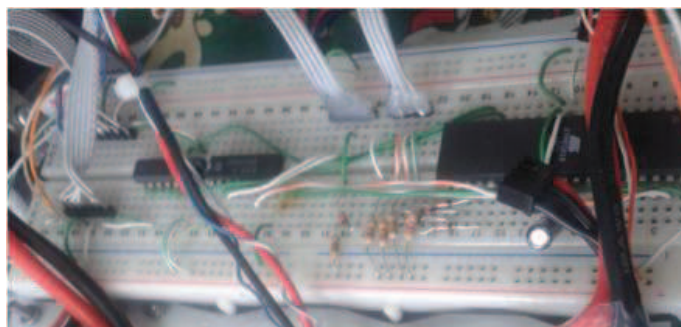


Figura 3.35 Buffer CD5040BE

### 3.3.6 BATERÍAS TIPO LI-PO (POLÍMERO DE LITIO) [36]

Son una variación de las Baterías Litio-Ion (Li-ion).

Las baterías de polímero de litio utilizan un polímero que les permite ser fabricadas en una mayor variedad de formas y tamaños que las baterías de ión de litio. Figura 3.36.

Emplean un electrolito gelificado para aumentar así la conductividad, a través del electrolito no circulan electrones sino iones. Esta batería es la única que reemplaza el separador poroso (separador de los electrodos) por un electrolito sólido, el gel solo se añade para el aumento de la conductividad y disminuir así la resistencia interna.



Figura 3.36 Baterías LI-PO.

Tienen una densidad de energía de entre 5 y 12 veces las de Ni-Cd o las de Ni-MH. Son baterías mucho más ligeras y que pueden adoptar cualquier forma. No poseen efecto memoria. El voltaje de cada elemento es de 3,7V.

La desventaja es que necesitan una carga mucho más lenta que las de Ni-Mh y además la carga de las baterías de Li-Po no produce el pico de tensión característico de las de NiCd o NiMH al alcanzar la carga máxima por lo que se deben emplear cargadores especiales.

Nunca se deben emplear cargadores diseñados para las baterías de Ni-Cd o Ni-Mh. Son también más caras. Tienen un voltaje mínimo de descarga más alto, si se descargan demasiado se pueden dañar.

Tienen poca resistencia interna por lo que su capacidad de descarga es elevada (se aprovecha casi el 100% de la energía), cercana a las de Ni-Cd.

Suministran un voltaje más alto que además se mantiene casi constante durante su descarga (un voltaje más alto por célula indica que se debe usar un menor número de ellas para alcanzar un determinado valor por lo que se ahorra en peso).

Para almacenarlas se recomienda dejarlas a media carga. Con un uso adecuado el tiempo de vida de estas baterías es superior al de las otras. Estas baterías también se degradan con el tiempo.

Este tipo de baterías son ideales para alimentar a motores muy potentes. Se debe cuidar de que el consumo máximo del motor sea menor que la descarga máxima de la batería para evitar que la vida de ésta se acorte demasiado.

Su reducido tamaño y ligereza también las hace ideales para aeromodelos y robótica, es por ello que se usan en el presente trabajo.

## CAPITULO 4

# DESARROLLO DEL CONTROL DEL SISTEMA ROBÓTICO MÓVIL MEDIANTE UNA FPGA

### 4.1 INTRODUCCIÓN

En este capítulo se indicará la teoría de control implementada así como también toda la programación realizada en la tarjeta Spartan 3E Starter Kit (500K) para poder realizar el control del sistema robótico, dicha programación está compuesta por una serie de etapas para facilitar la comprensión del proceso, entre las cuales se verán la adquisición de datos, conversión de los datos para ser procesados en el FPGA, algoritmo de control, conversión D/A, comunicación inalámbrica con la computadora y una interfaz gráfica en el computador que permita ver el comportamiento de los datos.

Además, se indican los circuitos implementados en el hardware del robot necesarios para manejar las velocidades de los motores, la adquisición de datos y el envío de los mismos vía inalámbrica hacia el computador.

Todos los procesos de programación fueron realizados en el software LABVIEW [37], estos procesos al interactuar entre sí, permiten que el FPGA controle a todo el sistema, los mismos constan en el programa como un SUB VI.

Como el FPGA presenta la peculiaridad de no poder usar más de 18 operadores en el mismo programa, se procedió a usar una serie de subvi que cumplan con el objetivo de la programación y así evitar el error de compilación que se obtiene al rebasar el número de dichos operadores.



## 4.2 DESARROLLO MATEMÁTICO DEL CONTROL [38]

### 4.2.1 USO DE LA INTERPOLACIÓN LINEAL PARA EL CONTROL DE SEGUIMIENTO DE TRAYECTORIA

#### 4.2.1.1 Interpolación Lineal

La idea de interpolación es poder estimar un valor  $f(x)$  para un valor  $x$  arbitrario a partir de la construcción de una curva que pasa por puntos en donde se ha realizado mediciones o que se conocen previamente, asumiendo que este valor  $x$  esta dentro de los límites de los puntos conocidos ya que de lo contrario se estaría hablando de una extrapolación.

La interpolación lineal es el método más sencillo de interpolar, ya que existen también interpolaciones cuadráticas y polinomiales. En la interpolación lineal (Figura 4.1) se trata de conectar 2 puntos conocidos  $(x_1, y_1)$  y  $(x_2, y_2)$  y para encontrar los valores deseados  $x$  entre estos dos puntos se debe seguir la ecuación de la recta:

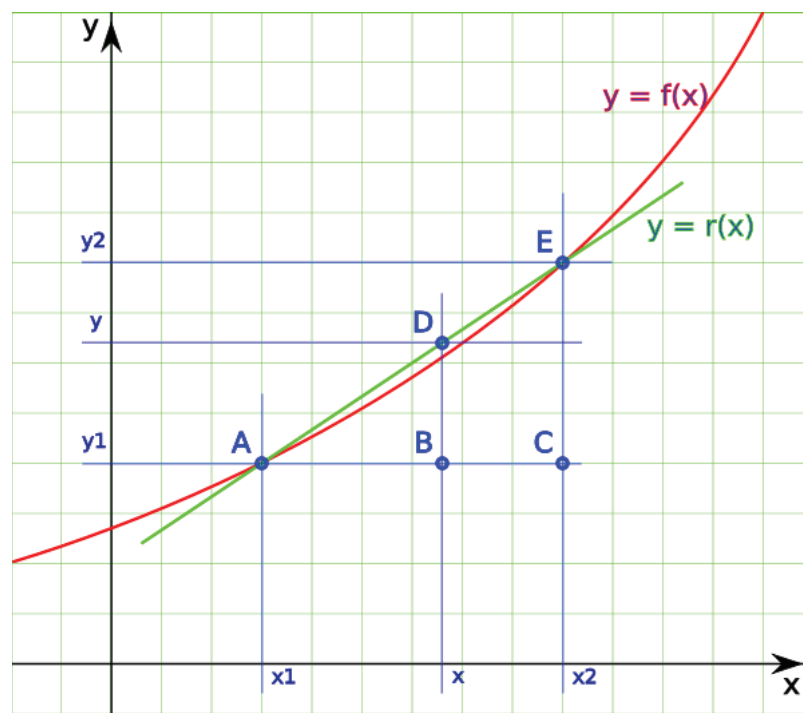


Figura 4.1 Interpolación Lineal.

$$\frac{y-y_1}{y_2-y_1} = \frac{x-x_1}{x_2-x_1} \quad \text{Ec 4.1}$$

En la Ecuación el único valor no conocido sería  $Y$  por lo que se determina la ecuación:

$$y = y_1 + (x - x_1) \frac{y_2 - y_1}{x_2 - x_1} \quad \text{Ec 4.2}$$

En donde se asume que  $x_1 < x < x_2$ , de lo contrario como ya se señaló se estaría ante una extrapolación.

#### 4.2.1.2 Planteamiento del Problema

Para aplicar este método en la resolución del problema, se debe asumir que la evolución de la salida del sistema puede ser aproximada mediante la interpolación lineal en cada periodo de muestreo de los datos. Ante esta suposición y, además, conociendo el estado deseado, es decir, la trayectoria a seguir con los parámetros  $x_d$ ,  $y_d$ ,  $\theta_d$  es posible encontrar la acción de control necesaria para llevar al sistema desde el estado actual al estado deseado.

Si bien el realizar el control de seguimiento mediante odometría implica un error de seguimiento, gracias a este método de control este error puede ser reducido simplemente decrementando el tiempo de muestreo ya que de esta forma la línea obtenida por interpolación será mucho más cercana a la trayectoria real.

La interpolación lineal es la base para muchos métodos numéricos, por ejemplo si esta aproximación es integrada, se obtiene el método llamado integración trapezoidal.

La Figura 4.2 representa la aproximación de la función  $y=g(x)$  para el intervalo  $[a, b]$ , donde  $g(a)$  y  $g(b)$  son valores conocidos de  $g(X)$  en  $x=a$  y  $x=b$  respectivamente.

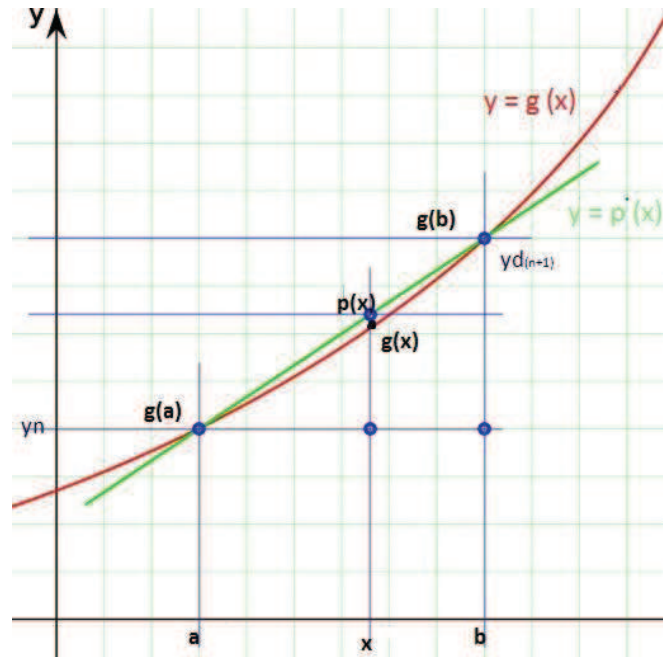


Figura 4.2 Función  $y = g(x)$ .

Es así que el error de interpolación:

$$e(x) = g(x) - p(x) \quad \text{Ec 4.3}$$

Es limitado y está determinado por:

$$|e(x)| \leq \frac{1}{2} |(x - a)(x - b)| \max_{a \leq x \leq b} |g''(x)| \quad \text{Ec 4.4}$$

Donde el máximo error se da aproximadamente en el punto medio  $x_m = 0,5(a + b)$ ; y si la segunda derivada de la función  $g(x)$  en el intervalo se mantiene aproximadamente constante entonces puede ser aproximada a  $g''(x_m)$ .

Visto esto se puede utilizar esta metodología para la obtención de la señal de control de forma que el sistema siga la trayectoria de referencia.

Si se tiene el sistema:

$$y' = f(y, t, u) ; y(0) = y_0 \quad \text{Ec 4.5}$$

Donde  $y$  sería la salida del sistema a ser controlada;  $u$  la acción de control;  $t$  el tiempo. Los valores de  $y(t)$  en tiempo discreto  $t = nT_0$  donde  $T_0$  es el tiempo de muestreo y  $n \in \{0, 1, 2, 3, \dots\}$  serán notados como  $y_n$ .

Si se desea calcular el valor de  $y_{n+1}$  conociendo previamente el valor de  $y_n$  la Ecuación 4.5, puede ser integrada dentro de un intervalo de tiempo  $nT_0 \leq t \leq (n+1)T_0$  así:

$$y_{n+1} = y_n + \int_{nT_0}^{(n+1)T_0} f(y, t, u) dt \quad \text{Ec 4.6}$$

Donde la acción de control  $u$  se mantiene constante en el intervalo  $nT_0 \leq t \leq (n+1)T_0$ . Por lo tanto si se conoce la trayectoria deseada a ser seguida por  $y(t)$  y se nota como  $y_d(t)$ , entonces  $y_{n+1}$  puede ser sustituida como  $y_{d_{n+1}}$  en la ecuación anterior considerando que el sistema se desarrolla de acuerdo a la aproximación lineal  $p(t)$ , entonces es factible calcular  $u_n$  que representa la acción de control que se requiere para ir del estado actual  $y_n$  al estado deseado  $y_{d_{n+1}}$ .

Entonces la aproximación lineal:

$$P(t) = y_n + \frac{y_{d_{n+1}} - y_n}{T_0} (t - nT_0) \quad \text{Ec 4.7}$$

Puede ser sustituida en la Ecuación 4.5

$$y_{n+1} \approx y_n + \int_{nT_0}^{(n+1)T_0} f(P(t), t, u) dt \quad \text{Ec 4.8}$$

Considerando el modelo cinemático del robot unicyclo:

$$\begin{cases} \dot{x} = v \cos \theta \\ \dot{y} = v \sin \theta \\ \dot{\theta} = w \end{cases} \quad \text{Ec 4.9}$$

Donde  $v$ ,  $w$  son las velocidades lineal y angular del robot móvil respectivamente;  $(x,y)$  son las coordenadas cartesianas de la posición,  $\theta$  es la orientación del robot con respecto al eje  $x$ .

El objetivo se centra en encontrar los valores de  $v$ ,  $w$  que hagan que el robot siga una trayectoria predeterminada con un error mínimo.

De la Ecuación 4.9, (modelo cinemático) se desprende:

$$\begin{cases} x_{n+1} = x_n + \int_{nT_0}^{(n+1)T_0} v \cos \theta dt \\ y_{n+1} = y_n + \int_{nT_0}^{(n+1)T_0} v \sin \theta dt \\ \theta_{n+1} = \theta_n + \int_{nT_0}^{(n+1)T_0} w dt \end{cases} \quad \text{Ec 4.10}$$

$v, w$  se mantienen constantes en el intervalo  $nT_0 \leq t \leq (n+1)T_0$  e iguales a  $v_n, w_n$ , entonces:

$$\begin{aligned} \theta(t) &= \theta_n + w_n(t - nT_0) \\ \theta_{n+1} &= \theta_n + w_n T_0 \end{aligned} \quad \text{Ec 4.11}$$

Sustituyendo en  $x_{n+1}$  sería:

$$x_{n+1} = x_n + \int_{nT_0}^{(n+1)T_0} v \cos[\theta_n + w_n(t - nT_0)] dt \quad \text{Ec 4.12}$$

Donde:

$$\int_{nT_0}^{(n+1)T_0} \cos[\theta_n + w_n(t - nT_0)] dt = \frac{1}{w_n} \{\sin \theta_{n+1} - \sin \theta_n\} \quad \text{Ec 4.13}$$

Sustituyendo:

$$x_{n+1} = x_n + v_n \frac{1}{w_n} \{\sin \theta_{n+1} - \sin \theta_n\} \quad \text{Ec 4.14}$$

Análogamente para  $y$ :

$$y_{n+1} = y_n + v_n \frac{1}{w_n} \{\cos \theta_n - \cos \theta_{n+1}\} \quad \text{Ec 4.15}$$

Por lo que se obtiene el siguiente sistema:

$$\begin{bmatrix} x_{n+1} \\ y_{n+1} \\ \theta_{n+1} \end{bmatrix} = \begin{bmatrix} x_n \\ y_n \\ \theta_n \end{bmatrix} + \begin{bmatrix} \frac{1}{w_n} (\sin \theta_{n+1} - \sin \theta_n) & 0 \\ \frac{1}{w_n} (\cos \theta_n - \cos \theta_{n+1}) & 0 \\ 0 & T_0 \end{bmatrix} \begin{bmatrix} v_n \\ w_n \end{bmatrix} \quad \text{Ec 4.16}$$

Como se conoce la trayectoria a seguir por el robot  $x_{n+1}$ ,  $y_{n+1}$ ,  $\theta_{n+1}$ , la única incógnita del sistema serían las velocidades  $v_n$  y  $w_n$ .

Entonces, de la Ecuación 4.11 se obtiene:

$$w_n = \frac{\theta_{n+1} - \theta_n}{T_0} = \frac{\Delta\theta}{T_0} \quad \text{Ec 4.17}$$

Y si de:

$$\begin{bmatrix} x_{n+1} \\ y_{n+1} \\ \theta_{n+1} \end{bmatrix} = \begin{bmatrix} x_n \\ y_n \\ \theta_n \end{bmatrix} + \begin{bmatrix} \frac{1}{w_n} (\sin \theta_{n+1} - \sin \theta_n) & 0 \\ \frac{1}{w_n} (\cos \theta_n - \cos \theta_{n+1}) & 0 \\ 0 & T_0 \end{bmatrix} \begin{bmatrix} v_n \\ w_n \end{bmatrix} \quad \text{Ec 4.18}$$

Se Toma:

$$\begin{bmatrix} x_{n+1} \\ y_{n+1} \end{bmatrix} = \begin{bmatrix} x_n \\ y_n \end{bmatrix} + \begin{bmatrix} \frac{1}{w_n} (\sin \theta_{n+1} - \sin \theta_n) \\ \frac{1}{w_n} (\cos \theta_n - \cos \theta_{n+1}) \end{bmatrix} [v_n] \quad \text{Ec 4.19}$$

Se Puede definir:

$$\begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} = \begin{bmatrix} x_{n+1} - x_n \\ y_{n+1} - y_n \end{bmatrix} \quad \text{Ec 4.20}$$

$$B = \begin{bmatrix} \frac{1}{w_n} (\sin \theta_{n+1} - \sin \theta_n) \\ \frac{1}{w_n} (\cos \theta_n - \cos \theta_{n+1}) \end{bmatrix} \quad \text{Ec 4.21}$$

Donde:

$$B v_n = \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} \quad \text{Ec 4.22}$$

La Ecuación 4.22 es un sistema de 2 ecuaciones con 1 incógnita, obtenida de un estudio realizado en el artículo de la referencia [39] y cuya solución está dado por:

$$B^T B v_n = B^T \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} \quad \text{Ec 4.23}$$

De donde se tiene:

$$v_n = \frac{\Delta \theta}{(\sin \theta_{n+1} - \sin \theta_n)^2 + (\cos \theta_n - \cos \theta_{n+1})^2} \left[ \frac{\Delta x}{T_0} (\sin \theta_{n+1} - \sin \theta_n) + \frac{\Delta y}{T_0} (\cos \theta_n - \cos \theta_{n+1}) \right] \quad \text{Ec 4.24}$$

De tal manera que el controlador propuesto para el sistema robótico sería:

$$\begin{bmatrix} v_n \\ w_n \end{bmatrix} = \begin{bmatrix} k_v \left\{ \frac{\Delta \theta}{a^2 + b^2} \left[ \frac{\Delta x}{T_0} a + \frac{\Delta y}{T_0} b \right] \right\} \\ k_w \frac{\Delta \theta}{T_0} \end{bmatrix} \quad \text{Ec 4.25}$$

Donde  $a = (\sin \theta_{n+1} - \sin \theta_n)$  y  $b = (\cos \theta_n - \cos \theta_{n+1})$ , además  $k_v$  y  $k_w$  son constantes positivas  $0 \leq k_v, k_w \leq 1$  que permiten ajustar la función de control propuesta para este sistema.

### 4.3 DESARROLLO DEL PROGRAMA UTILIZANDO LABVIEW

#### 4.3.1 PROGRAMA PARA CREACIÓN DE TRAYECTORIAS Y VALORES DESEADOS DE LA TRAYECTORIA

Para el desarrollo de las trayectorias se utilizó la forma paramétrica de las funciones empleadas, es decir, las variables de dichas funciones van a depender explícitamente del tiempo, esto se puede apreciar en las ecuaciones vistas a continuación.

$$x = f(t) \quad \text{Ec 4.26}$$

$$y = f(x) = f(t) \quad \text{Ec 4.27}$$

Cada una de las trayectorias fue programada en un subvi diferente y así se obtuvo la trayectoria de la recta, la parábola, la circunferencia y de una senoidal, (en el caso de querer otra trayectoria diferente solo se debe parametrizar su función). Además este subvi permite obtener los valores deseados de la trayectoria que son  $x_d \ n+1$ ,  $y_d \ n+1$  y  $\theta_d \ n+1$ , como se verá a continuación para cada trayectoria.

##### 4.3.1.1 Sub VI de la trayectoria de la recta

La función normal de la recta consta de las variables  $x$  e  $y$ , las cuales pueden tener cualquier valor perteneciente al conjunto de los números reales, al transformarla a una función paramétrica se obtiene:

$$x = t \quad \text{Ec 4.28}$$

$$y = x \quad \text{Ec 4.29}$$



Estas ecuaciones al ser programadas dan como resultado el siguiente Sub VI (Figura 4.4) y cuyo símbolo dentro del programa final se muestra en la Figura 4.3



Figura 4.3 Icono del Sub VI para la generación de la trayectoria recta.

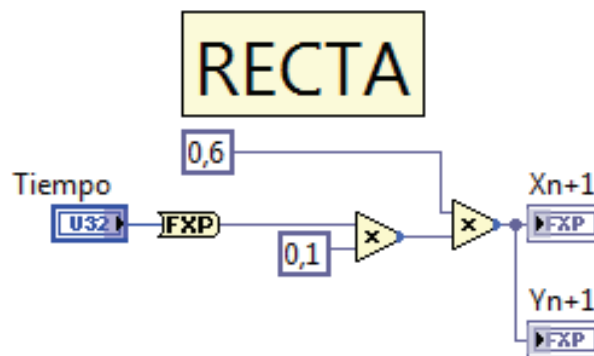


Figura 4.4 Sub VI para la generación de la trayectoria recta.

#### 4.3.1.2 Sub VI de la trayectoria de la circunferencia

La función normal de la circunferencia está representada en la ecuación siguiente:

$$X^2 + Y^2 = r^2 \quad \text{Ec 4.30}$$

y sus variables pueden ser expresadas por:

$$x = r \cos \alpha \quad \text{Ec 4.31}$$

$$y = r \sin \alpha \quad \text{Ec 4.32}$$

Al transformarlas a una función paramétrica se obtiene:

$$x = r \cos(t) \quad \text{Ec 4.33}$$

$$y = r \sin(t) \quad \text{Ec 4.34}$$

Estas ecuaciones al ser programadas se obtiene el siguiente Sub VI (Figura 4.6) y cuyo símbolo dentro del programa final se muestra en la Figura 4.5.



Figura 4.5 Icono del Sub VI para la generación de la trayectoria circular.

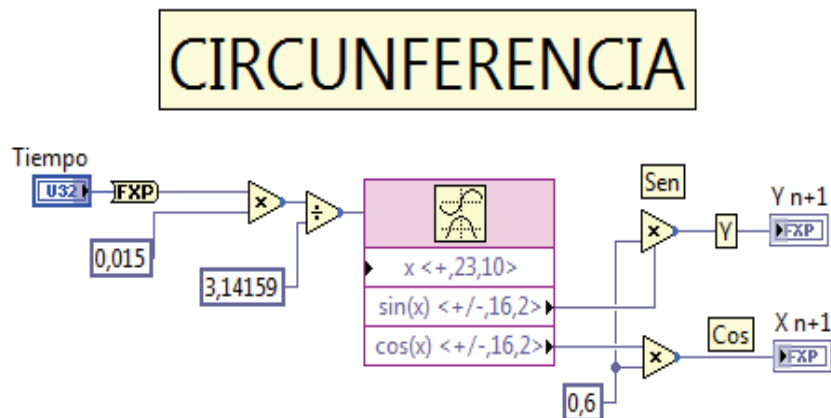


Figura 4.6 Sub VI para la generación de la trayectoria circular.

Se debe aclarar que por la capacidad de memoria de la tarjeta Spartan 3E Starter kit (FPGA), 128 + 16 Mbit, solo se puede presentar una trayectoria a la vez. Esto ya que el programa con cada trayectoria ocupa el 100% de los componentes de memoria de la Tarjeta

#### 4.3.2 PROGRAMA ADQUISICIÓN DE DATOS Y ESTIMACIÓN DE DELTA S

Para saber la posición en la que se encuentra el robot se usan los encoders incrementales. Estos encoders tienen la característica de que envían pulsos, los

mismos que mediante un Sub VI se procesan para transformarlos en distancia y así poder saber que distancia recorrió cada una de las ruedas. Como se vió en el Capítulo 3 los encoders funcionan a 3,3V pero se debe acondicionar su señal para evitar errores en los datos y a su vez el FPGA tan solo a 3,3V, por lo que se usó un circuito adicional para acoplarlos, éste circuito se lo puede observar en la Capítulo 3 en la Figura 3.32.

#### 4.3.2.1 Sub VI para adquisición de pulsos del encoder

Permite recolectar los datos de cada uno de los encoders, para su posterior procesamiento y así obtener los valores de la distancia que recorre cada una de las ruedas, para corregir la trayectoria que describe el sistema.

La programación empleada para lograr este propósito se presenta en las Figuras 4.7 y 4.8 y el ícono de este Sub VI se muestra en la Figura 4.9.

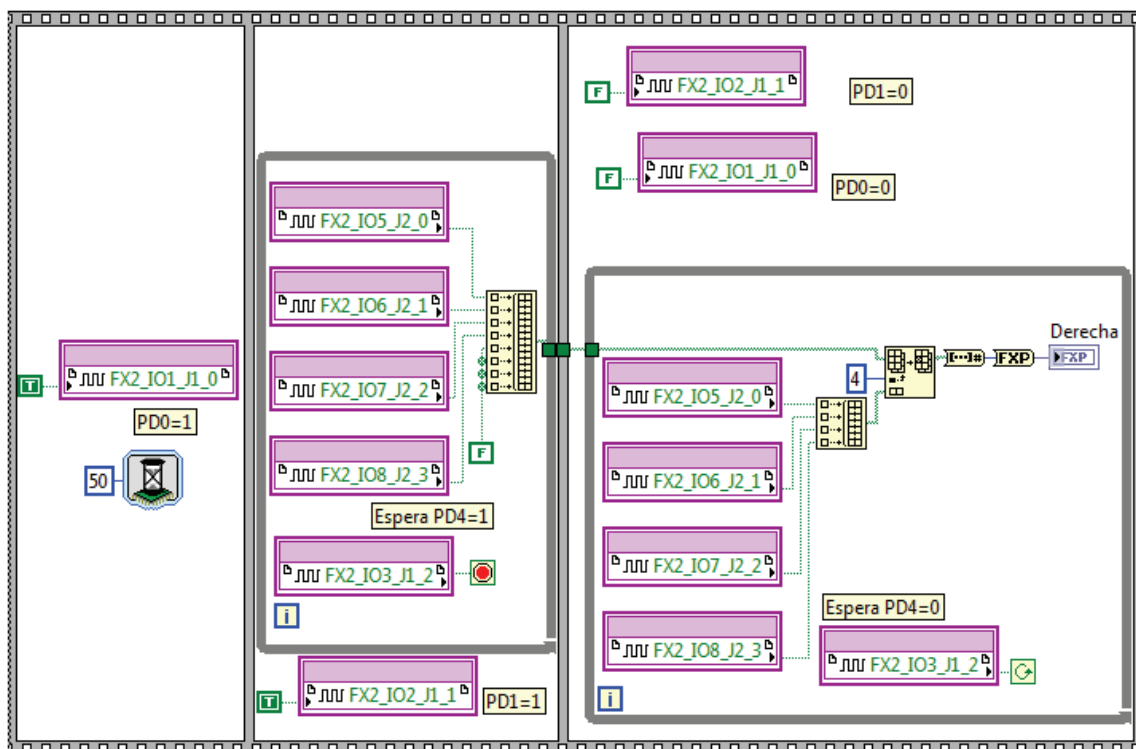


Figura 4.7 Sub VI para la recolección de datos de los encoders (Encoder Derecho).

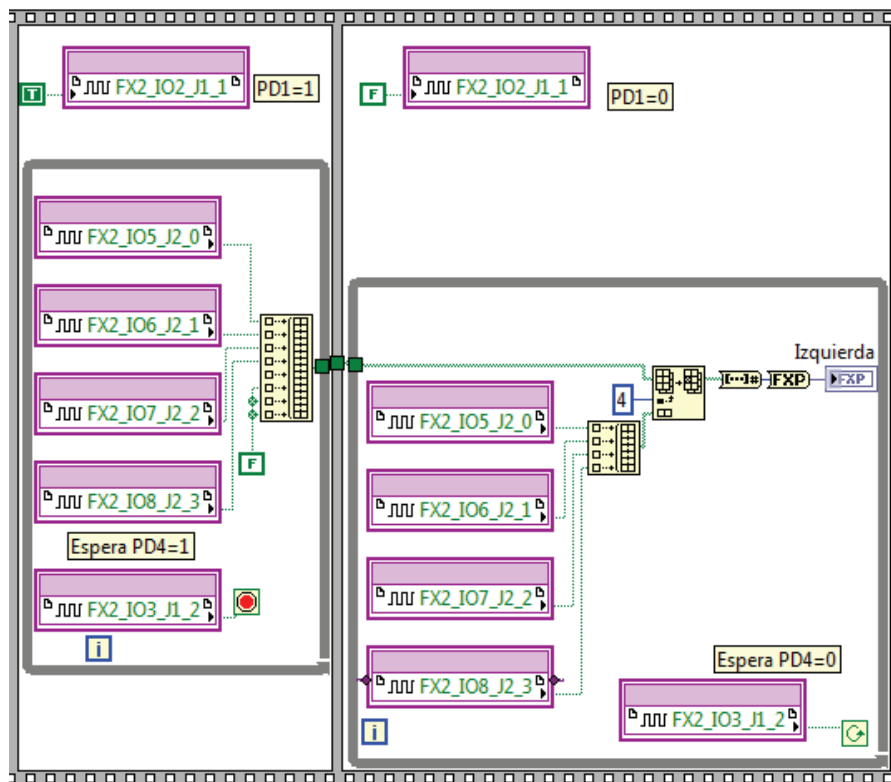


Figura 4.8 Continuación del Sub VI para la recolección de datos de los encoders (Encoder Izquierdo).



Figura 4.9 Icono del Sub VI para la recolección de datos de los encoders.

#### 4.3.2.2 Sub VI para Conversión de pulsos a radianes, estimación de Delta S y para obtener valores actuales

Este subvi permite mediante una serie de operaciones matemáticas (suma, multiplicación y división), transformar los pulsos captados por el anterior Sub VI y dar como resultado valores de distancia recorrida por cada rueda. Al aplicar la

Ecuación 4.35 se obtiene el valor de delta S, necesario para hallar las velocidades de cada rueda.

$$\Delta S = \frac{S_d + S_i}{2} \quad \text{Ec 4.35}$$

Donde  $S_d$  y  $S_i$ , son las variaciones del desplazamiento realizado por cada rueda y cuyo valor se obtiene mediante la ecuación presentada a continuación:

$$S = \frac{R * 2\pi}{PPV} * PR \quad \text{Ec 4.36}$$

Donde:

- $R$  es el radio de la rueda.
- $PPV$  son los pulsos máximos por rueda que da el encoder, en este caso son 20 pulsos/vuelta.
- $PR$  son los pulsos recorridos por cada encoder (dependen del tiempo de recolección de los datos)

Una vez obtenido el valor de delta S, este Sub VI permite calcular los valores actuales de la trayectoria los cuales son  $X_n$ ,  $Y_n$  y  $\Theta_n$  empleando las siguientes ecuaciones:

$$\theta_n = \frac{S_d - S_i}{b} + \theta_{n-1} \quad \text{Ec 4.37}$$

Donde:

$b$  es la distancia entre las ruedas traseras.

$\theta_{n-1}$  es el valor de  $\theta_n$  anterior.

Ahora es posible obtener los valores de  $X_n$  e  $Y_n$ , ya que estos dependen del valor de  $\theta_n$  como se puede observar en la siguiente Ecuación 4.38.

$$x_n = (\Delta S * \cos(\theta_n)) + x_{n-1}$$

$$y_n = (\Delta S * \text{sen}(\Theta n)) + y_{n-1} \tag{Ec 4.38}$$

Donde:

$x_{n-1}$  es el valor anterior en la coordenada de X,  $y_{n-1}$  es el valor anterior en la coordenada de Y.

La programación de este subvi se presenta en la Figura 4.11 y el ícono en la Figura 4.10, mostrados a continuación.



Figura 4.10 Icono del Sub VI para la generación de los valores n de X, Y y  $\Theta$ .

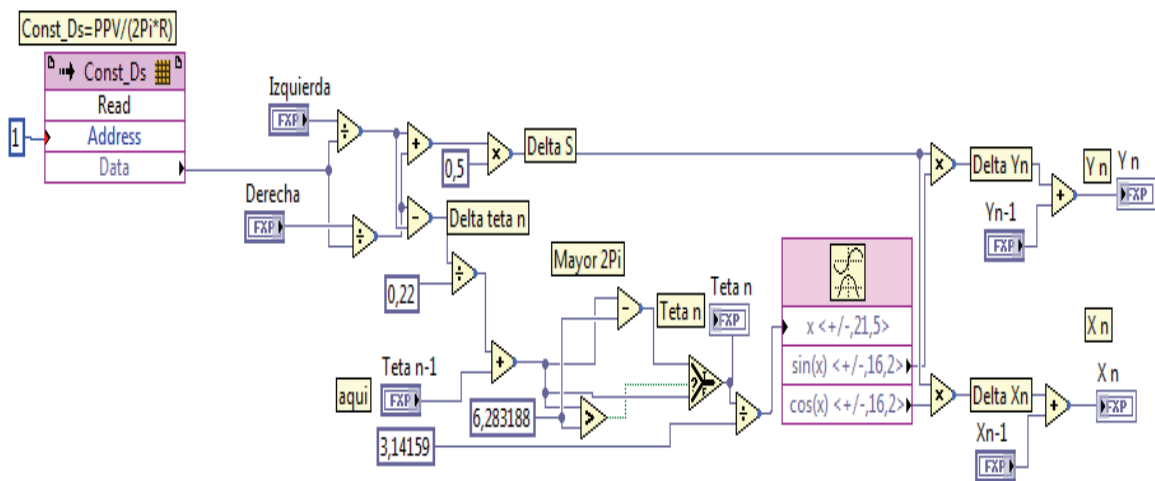


Figura 4.11 Sub VI para la generación de los valores n de X, Y y  $\Theta$ .

### 4.3.2.3 Sub VI Temporizador

Permite actualizar el dato obtenido de los encoders cada 100ms, esto sirve para realizar un control más fino de los datos, ya que las funciones paramétricas son dependientes del tiempo, y mientras menor sea el valor de éste, mejor será el control de las velocidades de las ruedas.

La programación de este Sub VI se presenta en la Figura 4.13 y el ícono en la Figura 4.12, mostrados a continuación.



Figura 4.12 Icono del Sub VI para encerrar los valores de los encoders cada 0,1seg

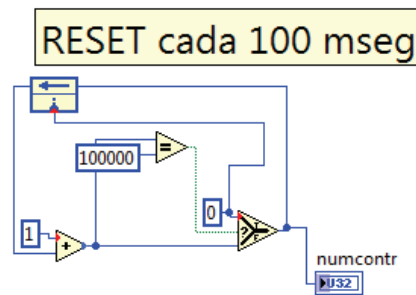


Figura 4.13 Sub VI para encerrar los valores de los encoders cada 0,1seg.

Al unir todos estos Sub VI se obtiene un módulo para la adquisición y procesamiento de datos provenientes de los encoders. Los valores finales corresponden a los valores actuales de la trayectoria  $X_n$ ,  $Y_n$  y  $\Theta_n$ , los cuales son necesarios para determinar las velocidades  $V_n$  y  $W_n$  de las ruedas, lo cual se explicará posteriormente.

### 4.3.3 PROGRAMA PARA EL CONTROL DE VELOCIDAD

Para realizar el control de la velocidad que debe tener cada rueda es necesario tener los valores de  $X_n$ ,  $Y_n$ ,  $\Theta_n$ , hallados anteriormente, así como también  $x_{n+1}$ ,  $y_{n+1}$  y  $\theta_{n+1}$ , que son los valores deseados de la trayectoria; estos últimos ya se obtuvieron con los Sub VI anteriormente presentados.

Ahora se analizará cómo obtener dichos valores y el control de velocidad propiamente dicho.

#### 4.3.3.1 Sub VI para obtener valores de Velocidad Lineal y Velocidad Angular de las ruedas y para el control de la velocidad de las ruedas

Las variables de velocidad lineal y velocidad angular permitirán obtener los valores de las velocidades angulares de las ruedas es decir  $w_d$  y  $w_i$  [10], estos valores son cruciales para el control del sistema, ya que determinan la velocidad que se le dará a cada rueda para que el sistema siga la trayectoria deseada con el menor error posible.

Para obtener las variables de la velocidad lineal y angular  $v_n$  y  $w_n$  respectivamente, se empleó la Ecuación 4.25, mostrada anteriormente en el desarrollo matemático del control.

Ahora, como se mencionó anteriormente, las ruedas se controlan en base a las velocidades angulares  $w_d$  y  $w_i$ , obtenidas a partir de  $v_n$  y  $w_n$ , como se puede apreciar en las siguientes ecuaciones.

$$w_d = \frac{v_n}{R} + \frac{w_n * b}{2R} \quad \text{Ec 4.39}$$

$$w_i = \frac{v_n}{R} - \frac{w_n * b}{2R} \quad \text{Ec 4.40}$$

Donde:

$w_d$  y  $w_i$  son las velocidades de las ruedas ya corregidas por el control del sistema. La programación de este Sub VI se presenta en la Figura 4.15 y el ícono en la Figura 4.14, mostrados a continuación.



Figura 4.14 Icono del Sub VI para obtener los valores de velocidades lineal  $v_n$  y angular  $w_n$  del sistema.



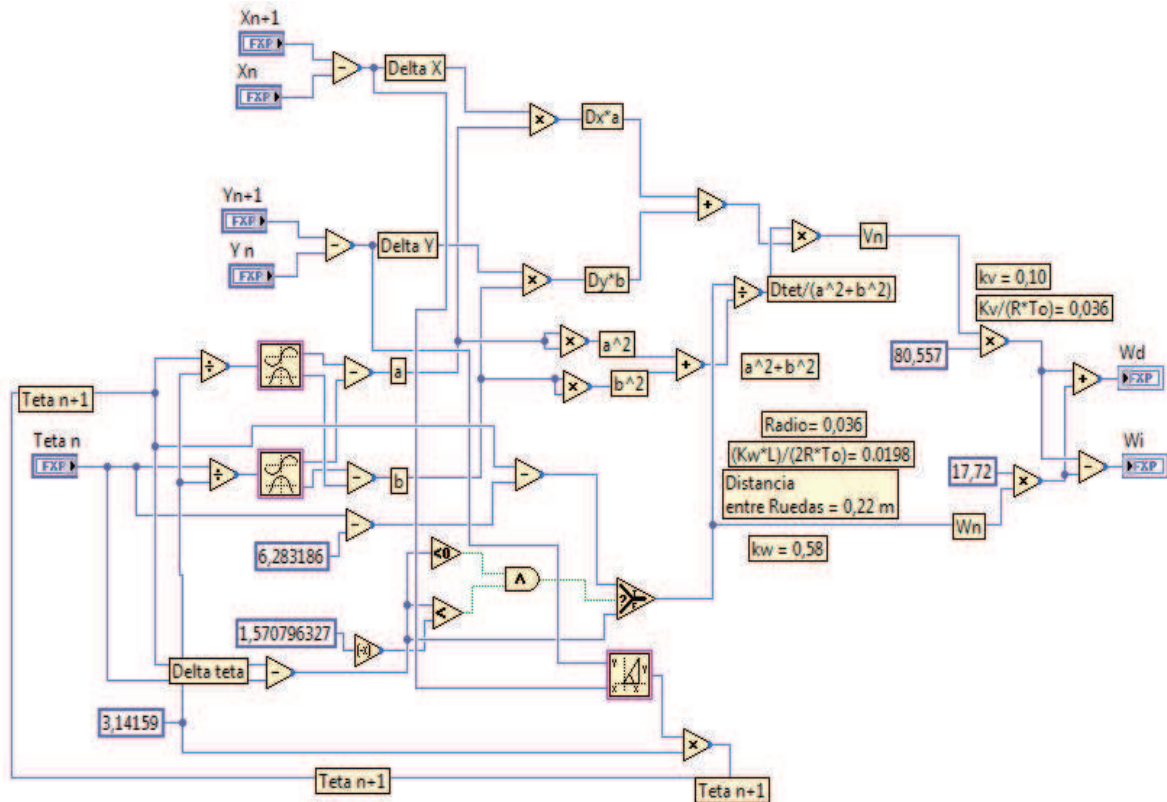


Figura 4.15 Sub VI para obtener velocidades lineal  $V_n$  y angular  $W_n$  del sistema.

Al unir todos estos Sub VI se obtiene los valores de  $W_d$  y  $W_i$ , que son necesarios para regular la velocidad de las ruedas, este control se realizará enviando dichos valores a los motores ensamblados a las ruedas mediante el conversor Digita-Análogo que posee la tarjeta Spartan 3E (500K), dicha transferencia se verá posteriormente.

#### 4.3.4 PROGRAMA PARA CONVERSIÓN Y TRANSFERENCIA DE DATOS (DAC)

Previo a indicar como se realizó la programación de este Sub VI para la conversión de datos y su transferencia, es necesario tener en cuenta cómo funciona la DAC y como debe estar integrada la trama de transferencia de la tarjeta Spartan 3E (500K). Este estudio fue realizado en el Capítulo 3 en la Sección 3.3.1.4, en dónde se explica claramente estas características.

#### 4.3.4.1 Sub VI de ajuste de la trama y configuración de la DAC

Este Sub VI permite convertir los datos de  $Wd$  y  $Wi$  obtenidos durante el proceso de la tarjeta Spartan 3E Started kid, en un valor de voltaje con variación de 0 a 2,5V, dichos valores serán captados por el amplificador de voltaje y acondicionados por el circuito siguiente para aumentar su rango de 0 a 6V necesarios para la alimentación de los motores, esto es posible gracias al Atmega8 y al integrado L298 como se vió en el Capítulo 3 Sección 2.1 .

En base a lo mencionado al inicio de la Sección 4.3.4, se pudo obtener el siguiente Sub VI (Figura 4.16), en donde además de realizar el ajuste en la trama de la DAC [40], de deshabilitan los demás dispositivos que comparten el puerto SPI de la tarjeta para evitar errores en los demás elementos que posee la placa del FPGA. El ícono de este Sub VI se muestra en la Figura 4.17.

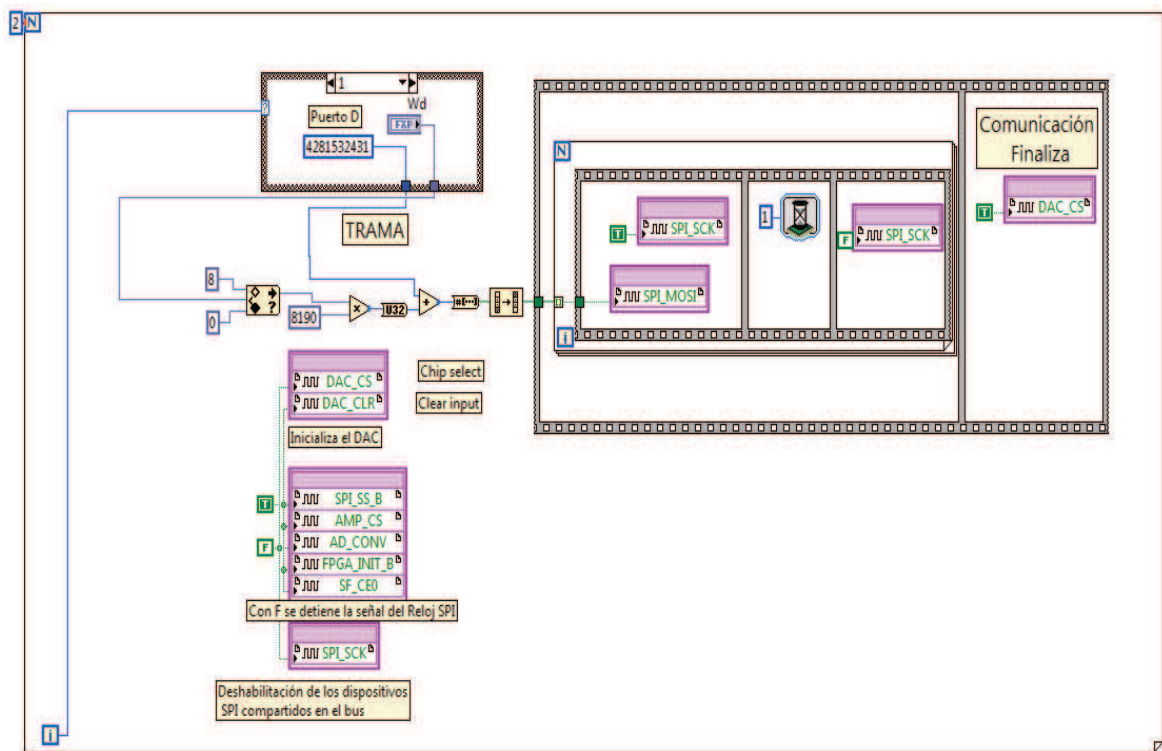


Figura 4.16 Sub VI para la configuración de la DAC de la tarjeta Spartan Started Kit 3E (FPGA).



Figura 4.17 Icono del Sub VI para la configuración de la DAC de la tarjeta Spartan Started Kit 3E (FPGA).

#### 4.3.4.2 Subvi para transferencia de datos a la computadora

En este subvi se realiza el tratamiento del dato para su posterior transmisión hacia el PC vía comunicación serial [41] por bluetooth, los datos transmitidos son  $X_n$ ,  $Y_n$ ,  $X_{n+1}$  y  $Y_{n+1}$ . Estos datos son transmitidos uno por uno para permitir la visualización del seguimiento de la trayectoria y el comportamiento del robot en el VI del PC. El tratamiento que se le da al dato es el siguiente: primero se determina si el número es positivo o negativo, inmediatamente después se separa el dato de las decenas y las unidades, después uno por uno se determinan los tres primeros números decimales después de la coma para ser transmitidos. Una vez separados cada número desde la decena hasta el tercer decimal después de la coma se procede a su transmisión, para ello se cuenta con una memoria auxiliar en el FPGA (`serial_aux`) que permite ubicar de forma temporal 9 elementos que conforman la cadena de datos a transmitirse, estos 9 datos son:

- Ubicación: que corresponde al código ASCII de x ( $X_n$ ), y ( $Y_n$ ), X ( $X_{n+1}$ ), Y ( $Y_{n+1}$ ).
- Signo del número.
- Decena.
- Unidad.
- Código ASCII del signo coma (,).
- Primer decimal.
- Segundo decimal.
- Tercer decimal.
- Código ASCII del Carácter de terminación (`\n`), necesario para cerrar la comunicación en el subVI Visa de LabVIEW.

Una vez que todos los datos están almacenados en la memoria auxiliar (serial\_aux), se utiliza otra memoria auxiliar (Conver\_serial) para transformar los números almacenados en la primera a su correspondiente código ASCII y transmitir los 8 bits que conforman cada uno de los nueve datos.

La transmisión serial se la realiza con la ayuda del terminal serial DCE con el que cuenta el Spartan 3E Starter Kit; en este caso se ha programado el terminal de tal manera que la velocidad de transmisión sea 9600 baudios, para ello se debe jugar con el tiempo y el estado en el que se encuentra el terminal DCE\_TXD, es por ello que en el Sub VI cada bit debe permanecer 104  $\mu$ s en el terminal. Por lo demás, es una transmisión serial típica con 1 bit de inicio, 8 bits de datos y 1 bit de parada, sin paridad. Esta configuración debe corresponderse con lo programado en el VI de la PC.

La programación de este Sub VI se muestra en las Figuras 4.18a y 4.18b, su correspondiente icono se muestra en la Figura 4.19.

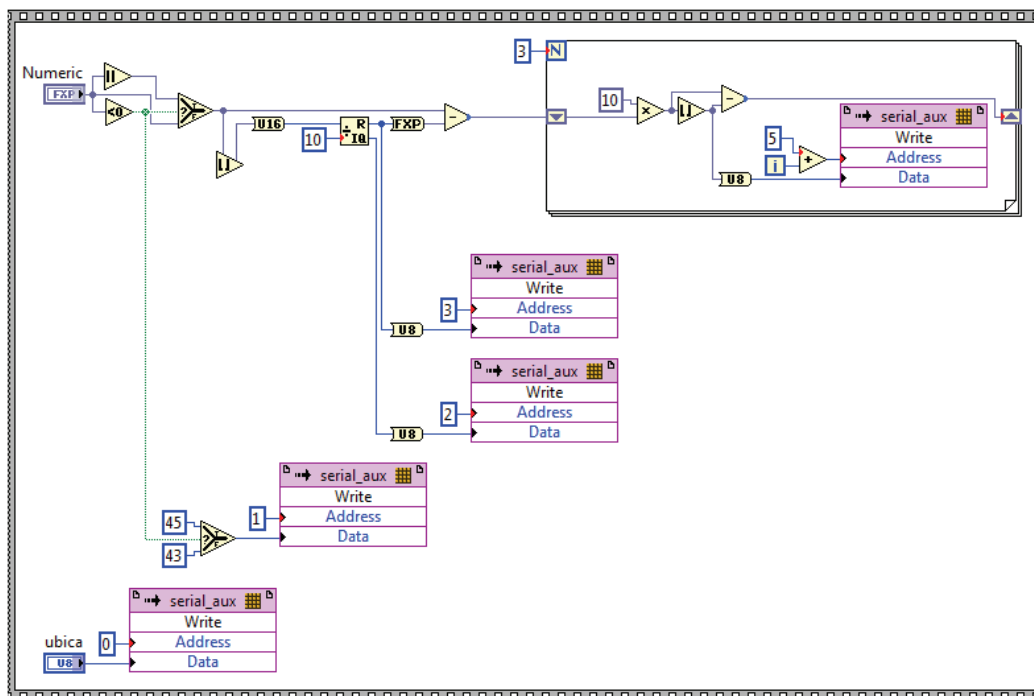


Figura 4.18a Sub VI para la transferencia de datos hacia la computadora.

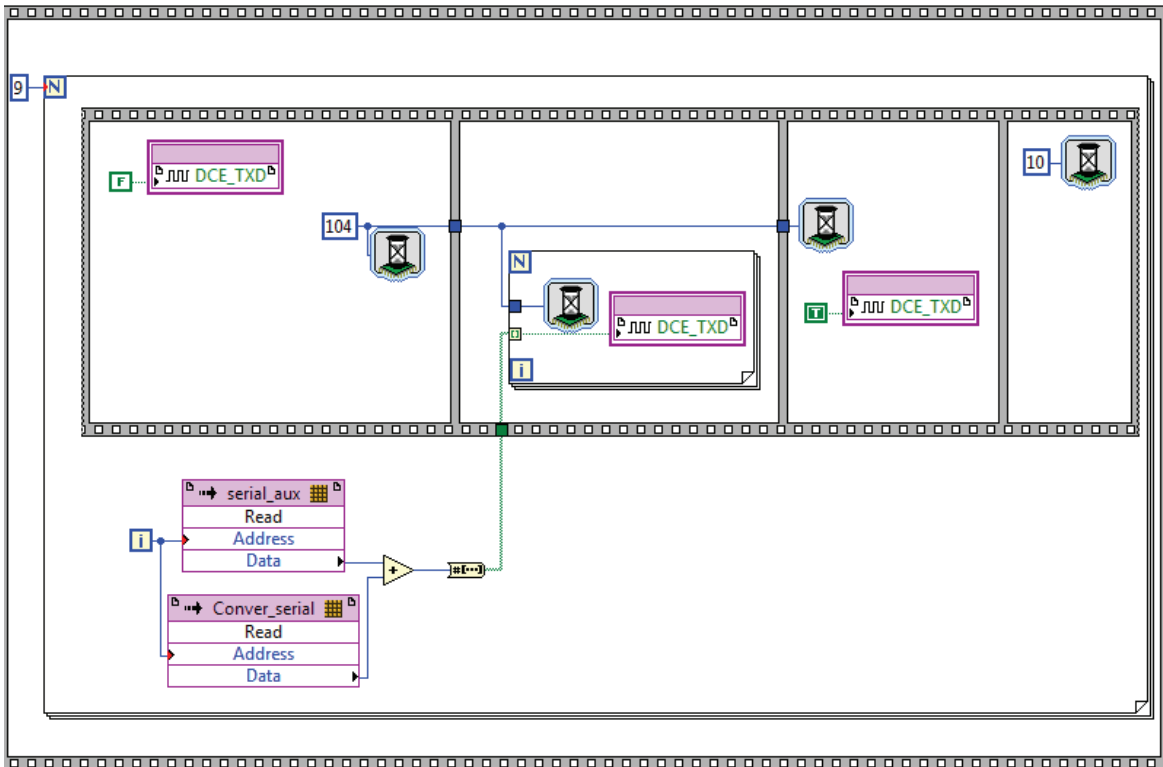


Figura 4.18b Sub VI para la transferencia de datos hacia la computadora.



Figura 4.19 Icono del Sub VI para la transferencia de datos hacia la computadora.

A continuación se presenta la pantalla del VI final empleado para realizar el control en la tarjeta Spartan 3E 500K Started Kid. Figura 4.20.

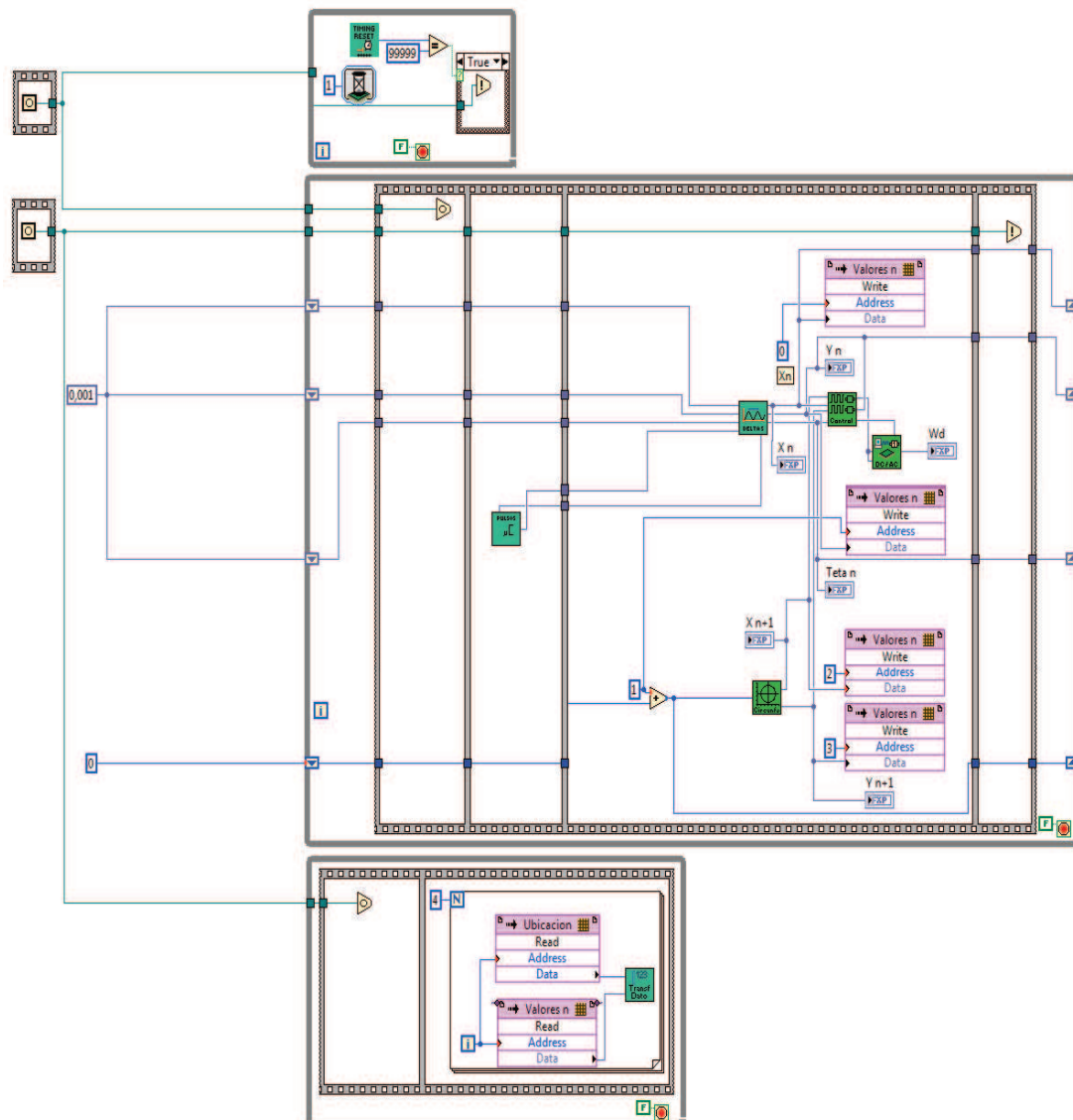


Figura 4.20 Pantalla principal del control.

#### 4.3.5 PROGRAMA DE VISUALIZACIÓN DEL SEGUIMIENTO DE LA TRAYECTORIA Y COMPORTAMIENTO DEL ROBOT.

Este es el VI desarrollado para la visualización del seguimiento de la trayectoria y del comportamiento del robot en el PC, el mismo consta de una pantalla principal con 3 pestañas para visualizar diferentes formas de onda.

En la primera pestaña (Figura 4.21), es posible visualizar únicamente la trayectoria realizada por el robot; en la segunda pestaña (Figura 4.22), es posible

realizar un contraste entre la trayectoria realizada por el robot y la trayectoria ideal generada en el FPGA; finalmente en la tercera pestaña (Figura 4.23), se encuentran la visualización del comportamiento en el eje X y en el eje Y, en las cuales se puede visualizar el error de seguimiento que tiene el robot. Su correspondiente diagrama de bloques se presenta en la Figura 4.24.

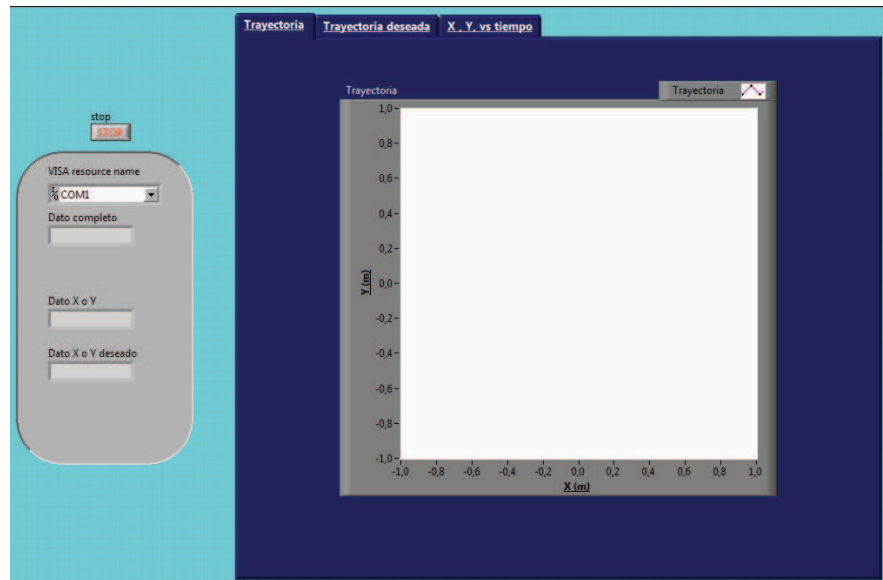


Figura 4.21 Primera pantalla trayectoria real.

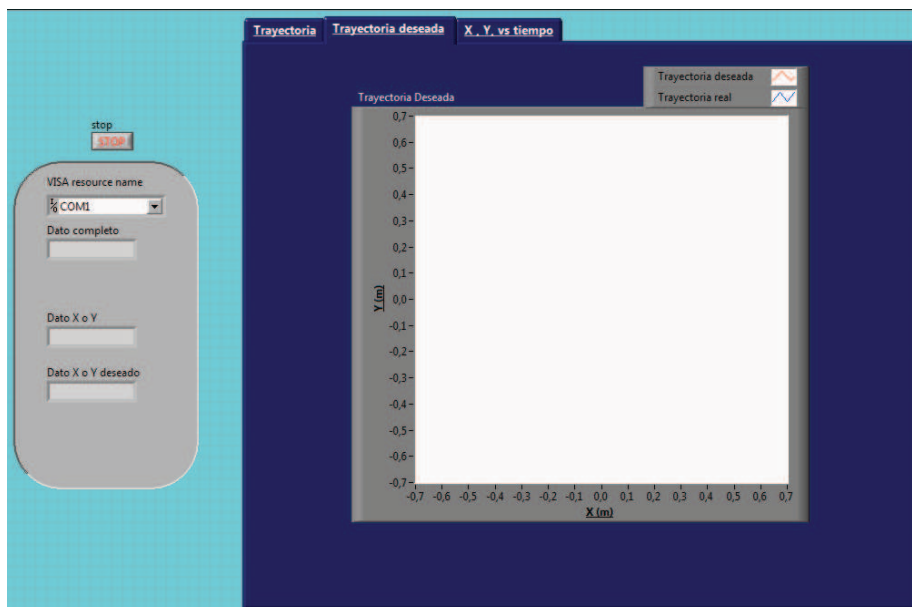


Figura 4.22 Segunda pantalla contraste de la trayectoria real y la trayectoria deseada.

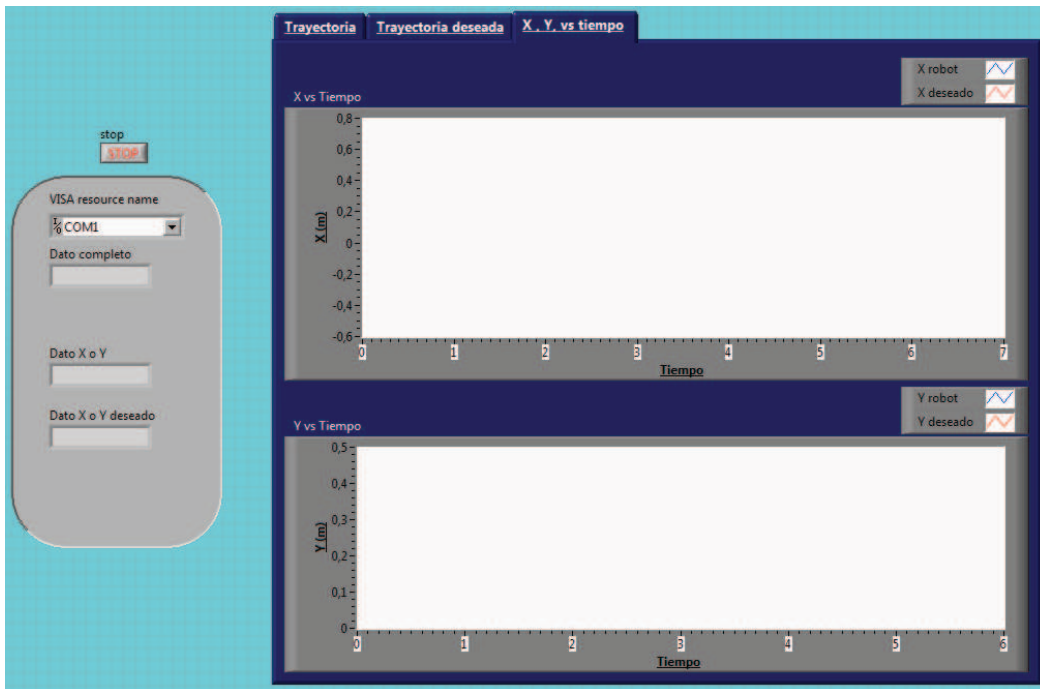


Figura 4.23 Tercera pantalla valores de  $X_n$ ,  $Y_n$ ,  $X_{n+1}$ ,  $Y_{n+1}$  vs tiempo.

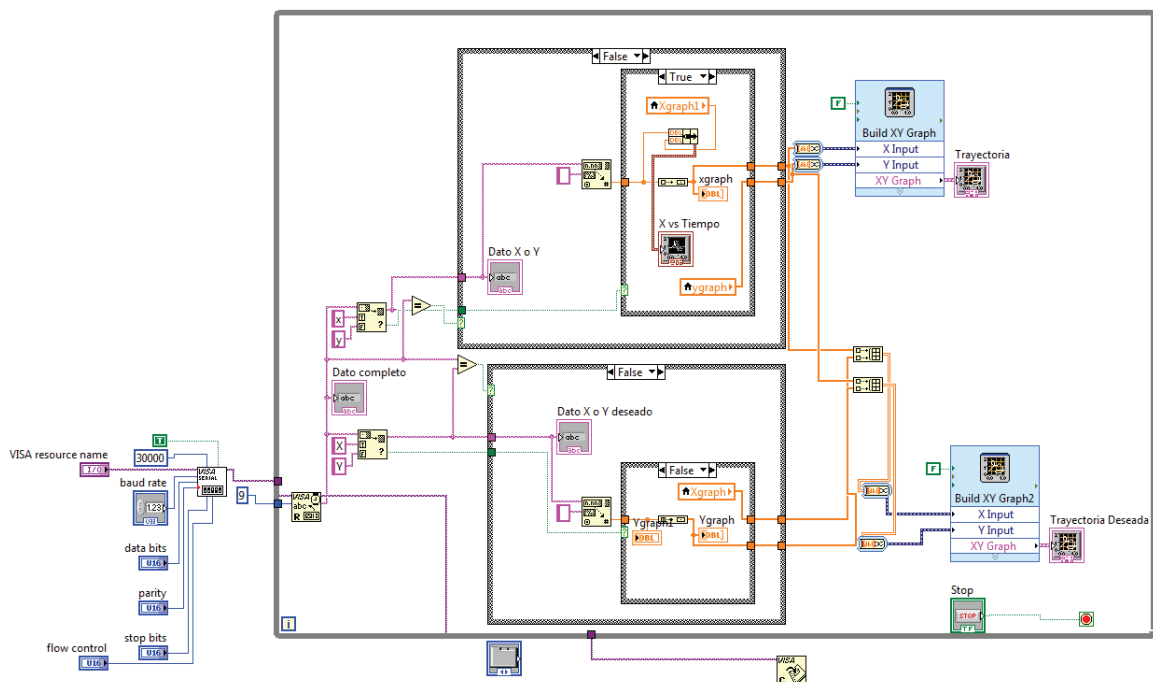


Figura 4.24 Diagrama de Bloques del programa de visualización de los datos en la PC.



# CAPITULO 5

## PRUEBAS Y RESULTADOS

### 5.1 INTRODUCCIÓN

En el presente capítulo se mostrarán los resultados obtenidos tanto de las simulaciones realizadas en el computador, así como también, de los resultados reales producidos por el robot a diferentes valores de  $K_v$  y  $K_w$ .

Las gráficas de las trayectorias y valores vs tiempo serán mostrados independientemente para dar un mejor tratamiento a los mismos y con el fin de ser lo más claros posible en el comportamiento que presenta el robot.

Además después de presentar las gráficas de cada trayectoria, se realizará una breve explicación del resultado obtenido así como también se aclararán los valores empleados para obtener los resultados indicados en cada pantalla.

### 5.2 RESULTADOS DE LAS SIMULACIONES EN COMPUTADOR

#### 5.2.1 SIMULACIÓN TRAYECTORIA LINEAL

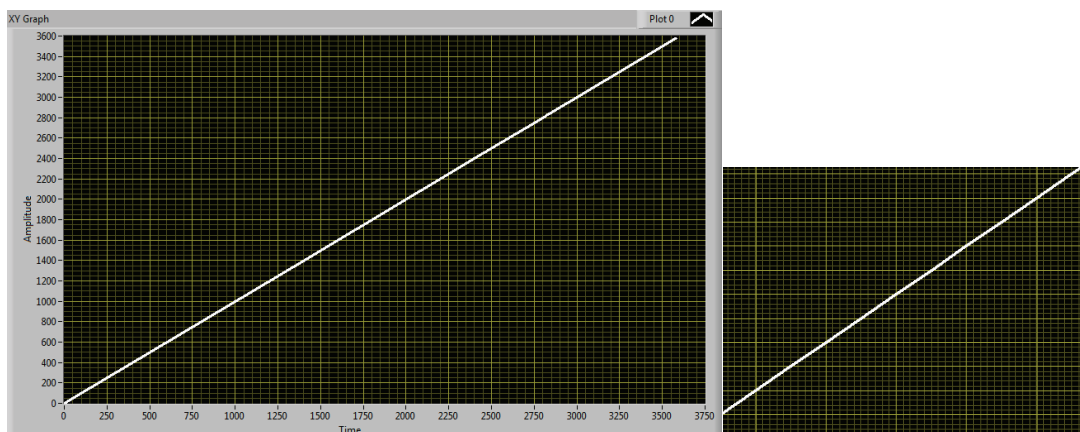


Figura 5.1 Simulación de la trayectoria del robot siguiendo una trayectoria recta.

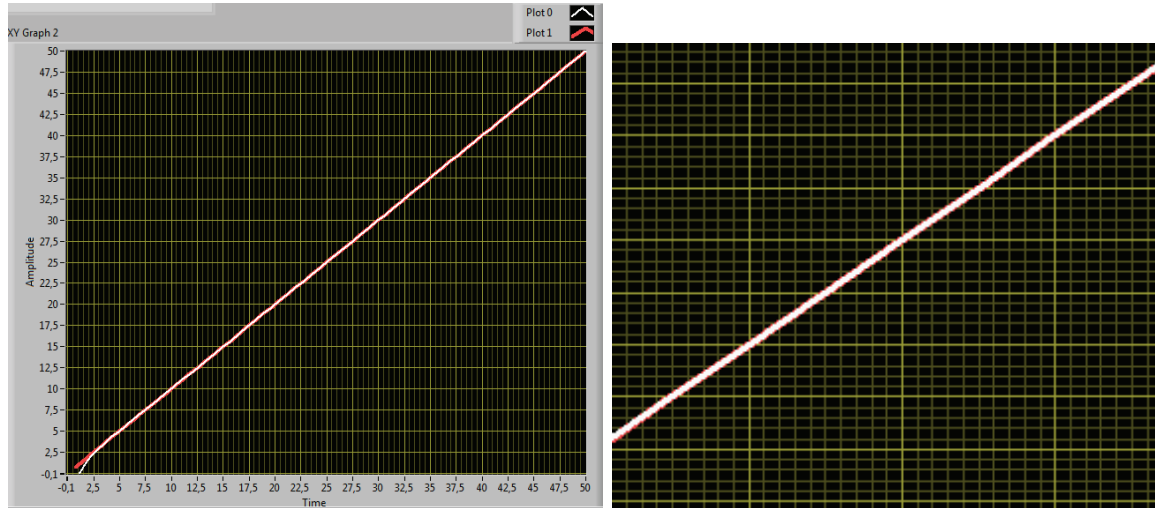


Figura 5.2 Comportamiento del robot siguiendo una trayectoria recta en base a la simulación.

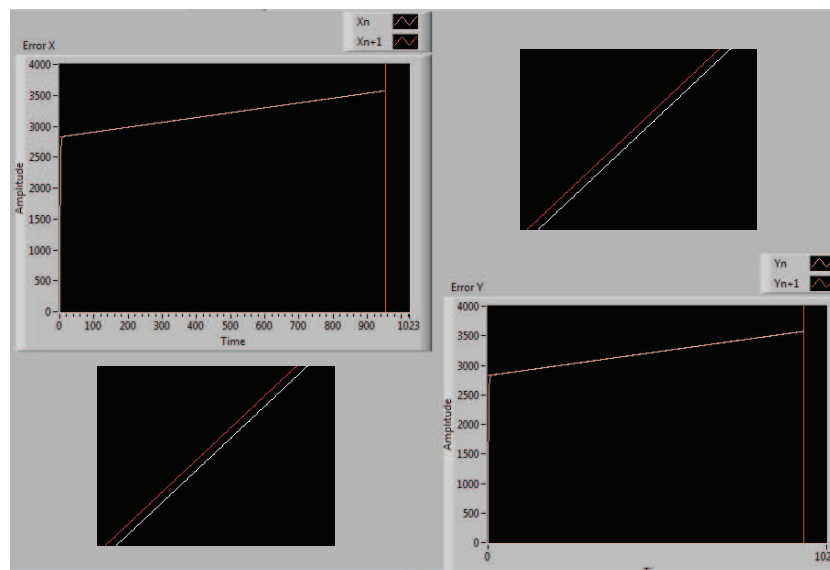


Figura 5.3 Error de seguimiento de las coordenadas X e Y del robot en base a la simulación.

En base a los resultados se puede decir que, primero el robot realiza un recorrido con la forma de la trayectoria deseada (Figura 5.1) y en segundo lugar que el error del robot al seguir la trayectoria recta es casi nulo (Figura 5.2), ya que a cada instante de tiempo está sobre ella y a ningún momento la llega a perder. Como se ve en la Figura 5.3, el robot sigue los valores de las coordenadas X e Y, cumpliendo así el objetivo del seguimiento, estos resultados se obtuvieron en

base a valores en las constantes  $K_v$  y  $K_w$  de 0,25 y 0,5 respectivamente y a un periodo de muestreo de 100mseg.

### 5.2.2 SIMULACIÓN TRAYECTORIA PARABÓLICA

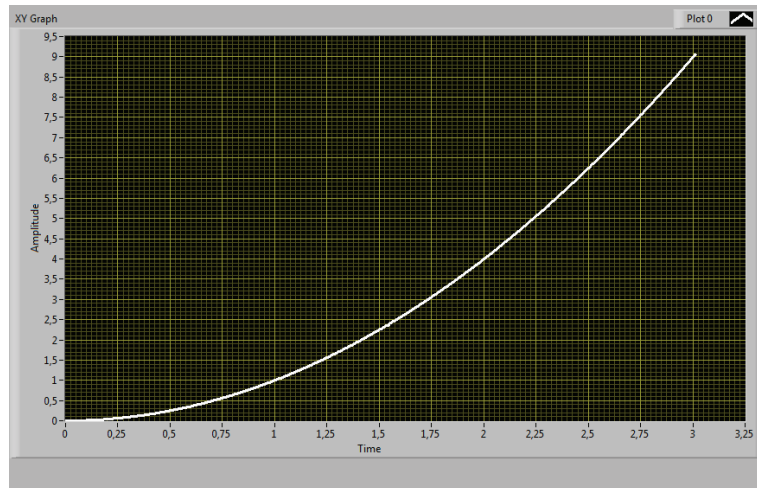


Figura 5.4 Simulación de la trayectoria del robot siguiendo una trayectoria parabólica.

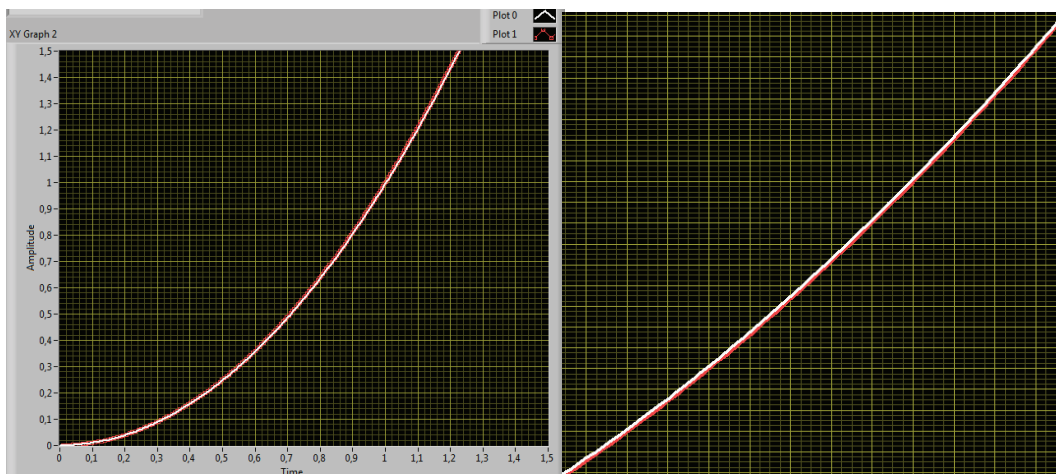


Figura 5.5 Comportamiento del robot siguiendo una trayectoria parabólica en base a la simulación.

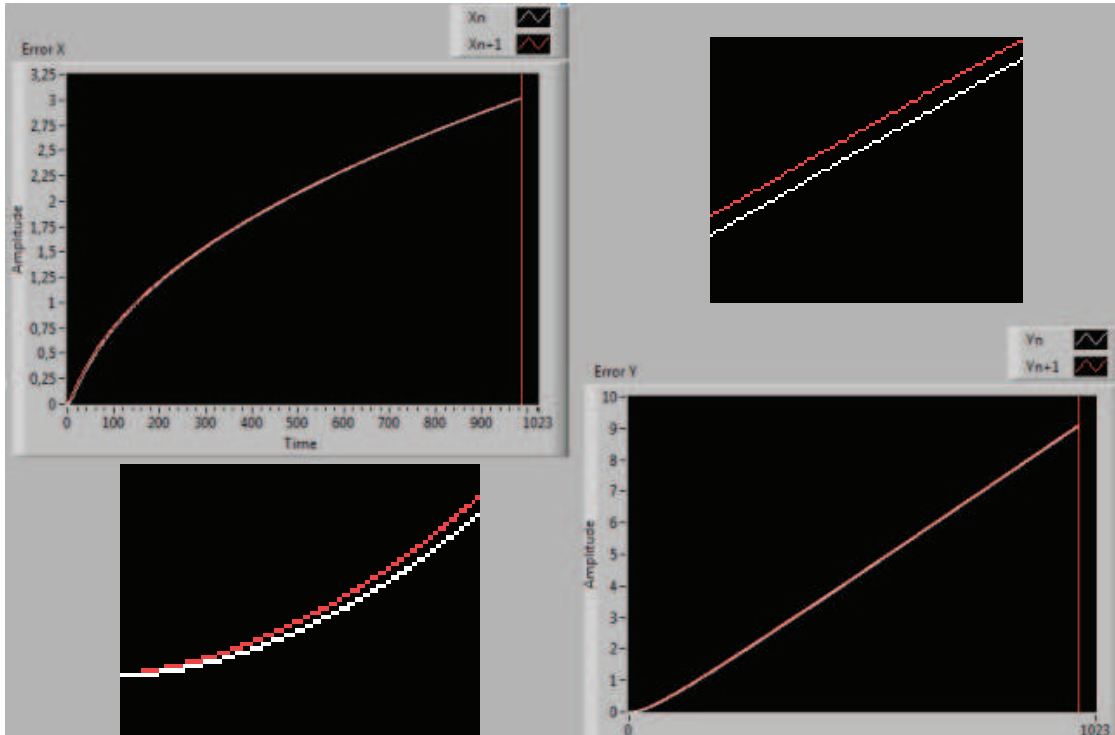


Figura 5.6 Error de seguimiento de las coordenadas X e Y del robot en base a la simulación.

En base a los resultados se puede decir que, primero el robot realiza un recorrido con la forma de la trayectoria deseada (Figura 5.4), en segundo lugar que el error del robot al seguir la trayectoria parabólica tiende a cero (Figura 5.5), ya que a ningún momento la pierde y está sobre ella a cada instante de tiempo.

Finalmente, para confirmar esto se puede ver en la Figura 5.6 que el robot sigue los valores de las coordenadas X e Y, cumpliendo así el objetivo del seguimiento, estos resultados se obtuvieron en base a valores en las constantes  $K_v$  y  $K_w$  de 0,15 y 0,3 respectivamente y a un periodo de muestreo de 100mseg.

### 5.2.3 SIMULACIÓN TRAYECTORIA SENOIDAL

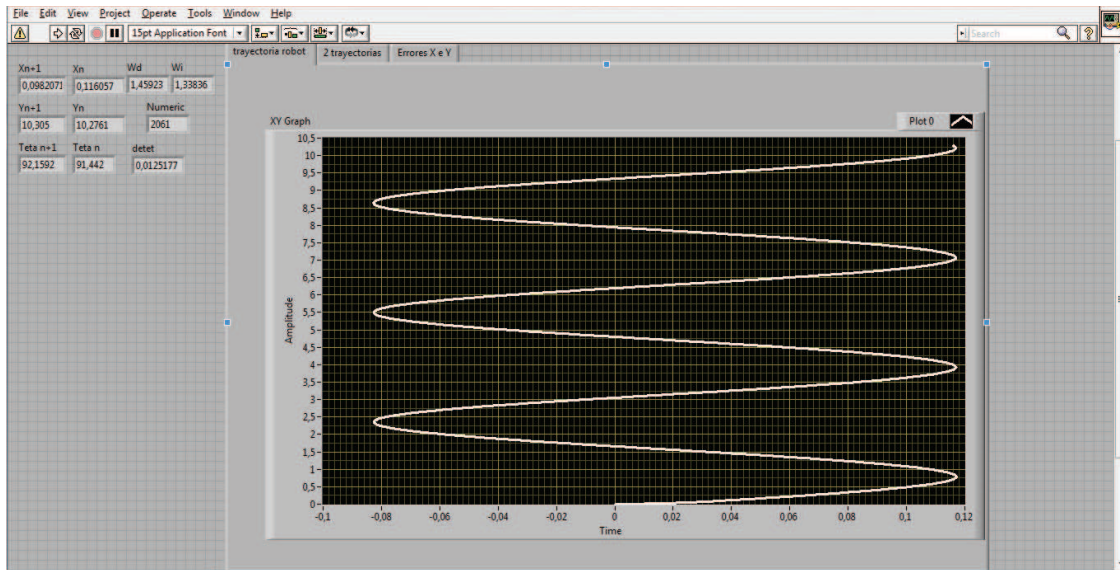


Figura 5.7 Simulación de la trayectoria del robot siguiendo una trayectoria senoidal.

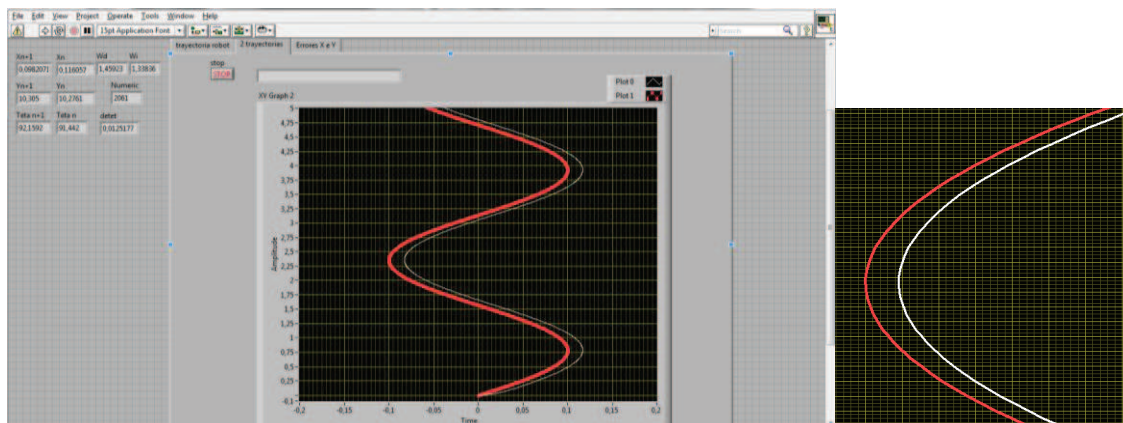


Figura 5.8 Comportamiento del robot siguiendo una trayectoria senoidal en base a la simulación.

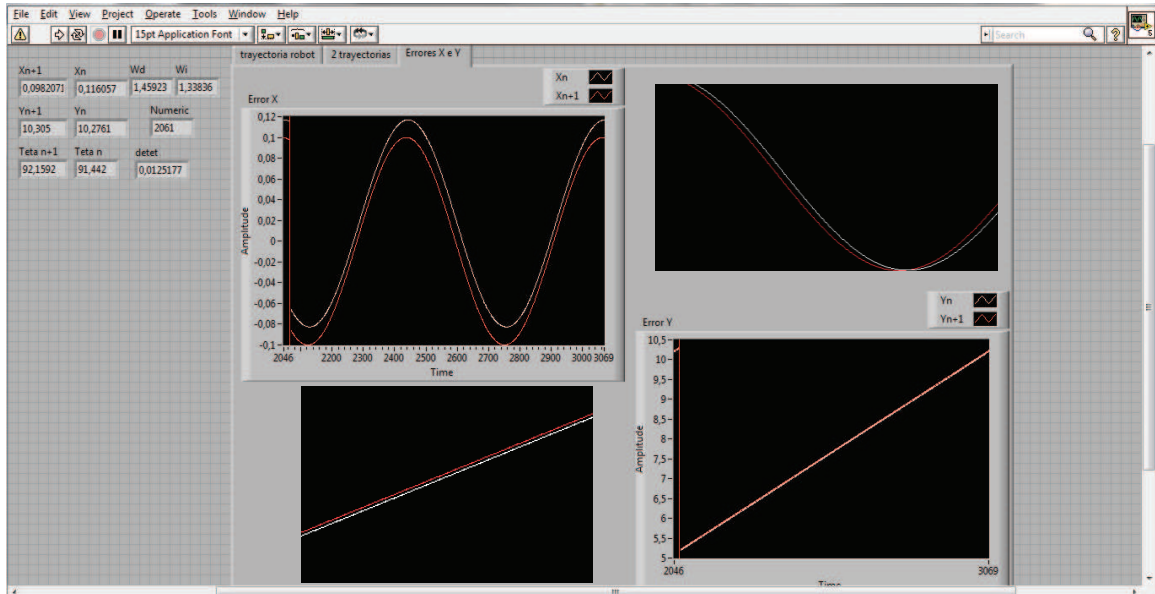


Figura 5.9 Error de seguimiento de las coordenadas X e Y del robot en base a la simulación.

En base a los resultados se puede decir que a pesar que el robot realiza una trayectoria senoidal (Figura 5.7), el error del robot al seguir la misma es notorio (Figura 5.8), ya que se puede apreciar que en momentos no alcanza la trayectoria, sino que simplemente se mueve paralelamente a ella.

El error radica en el seguimiento de la coordenada en Y como se aprecia en la Figura 5.9, estos resultados se obtuvieron en base a valores en las constantes  $K_v$  y  $K_w$  de 0,5 y 0,2 respectivamente y a un periodo de muestreo de 100mseg.

A continuación se muestra la misma simulación con otros valores de  $K_v$  y  $K_w$ , para así poder apreciar con claridad la diferencia que existe al variar dichos valores. Figuras 5.10 y 5.11.

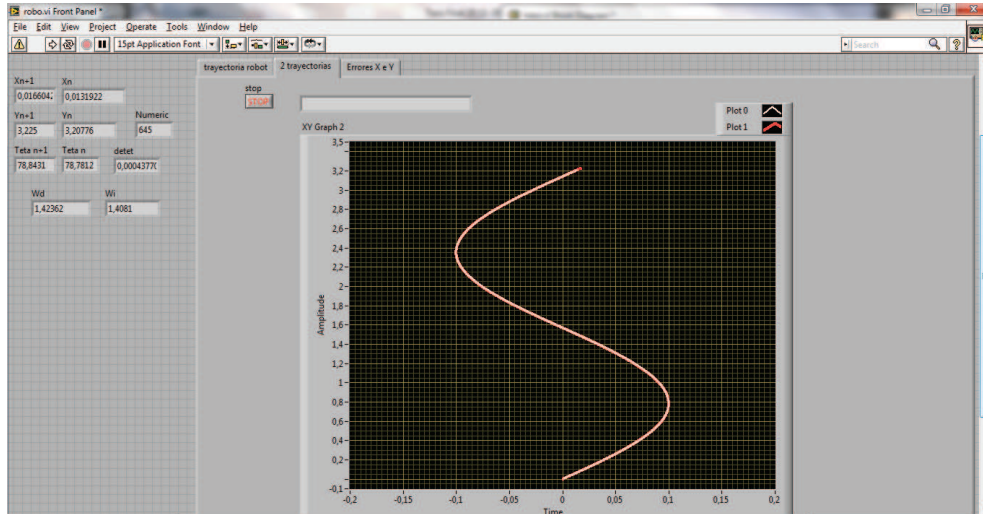


Figura 5.10 Comportamiento del robot siguiendo una trayectoria senoidal en base a la simulación.

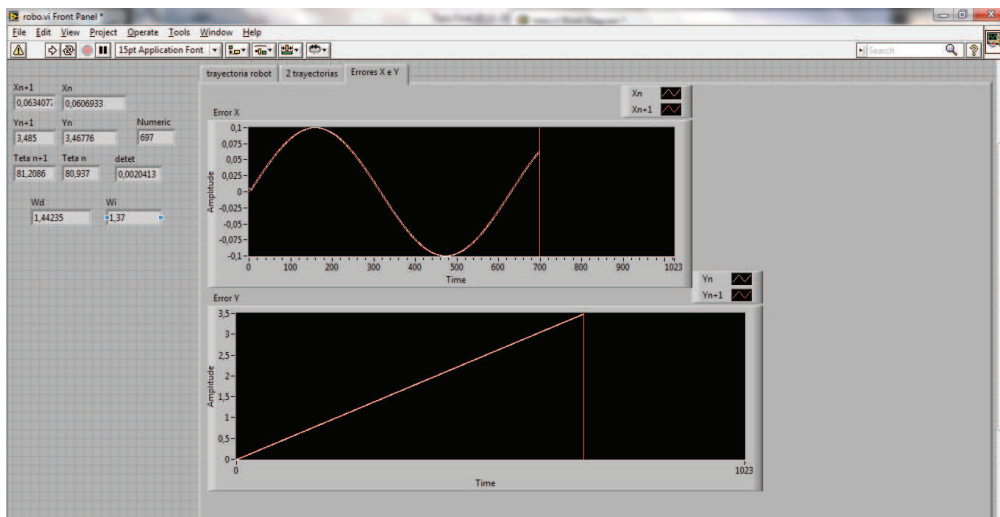


Figura 5.11 Error de seguimiento de las coordenadas X e Y del robot en base a la simulación.

Como se puede apreciar en la Figura 5.10, al variar los valores de las constantes  $K_v$  y  $K_w$  a 0,29 y 0,58 respectivamente, se obtiene una mejor respuesta del sistema, el error de seguimiento se redujo a prácticamente cero y así el robot a cada instante de tiempo está sobre la trayectoria, esto se puede comprobar al observar la Figura 5.11, en la cual los errores de posición con respecto a las coordenadas X e Y, son prácticamente cero.

## 5.2.4 SIMULACIÓN TRAYECTORIA CIRCULAR

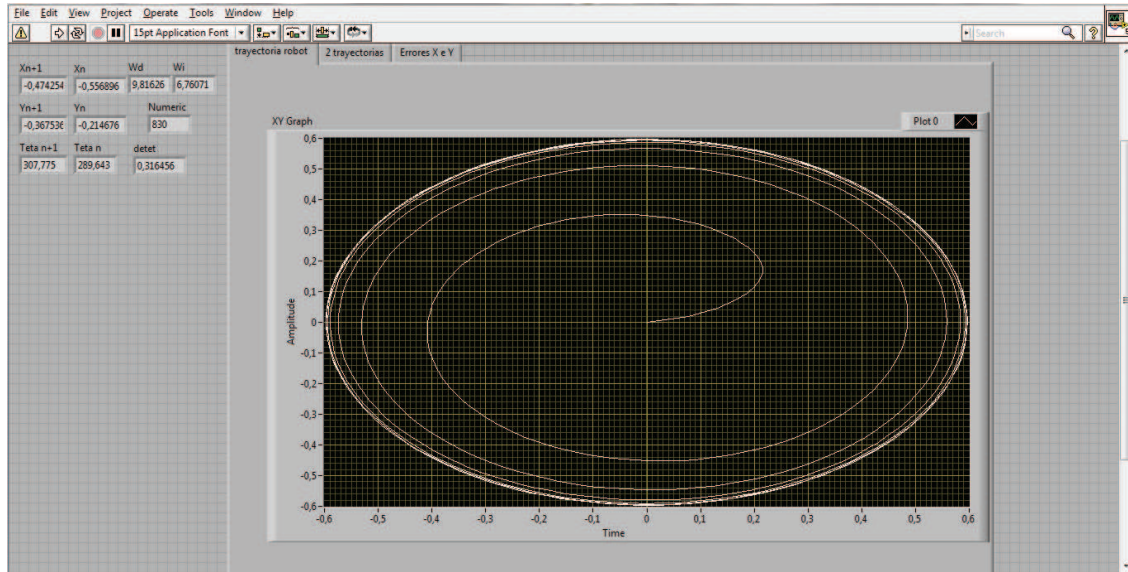


Figura 5.12 Simulación de la trayectoria del robot siguiendo una trayectoria circular.

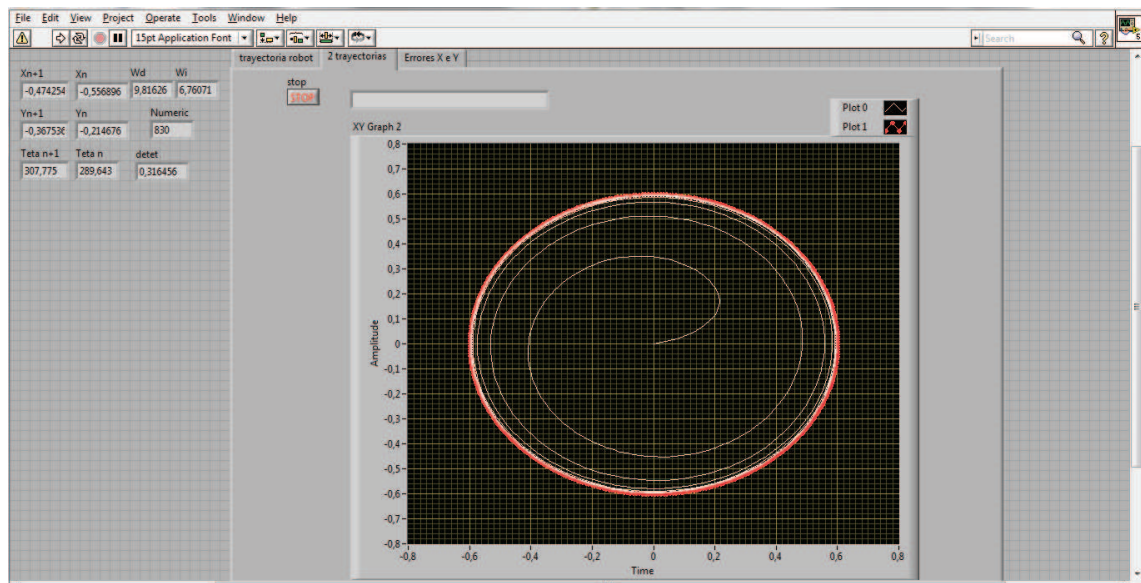


Figura 5.13 Comportamiento del robot siguiendo una trayectoria circular en base a la simulación.



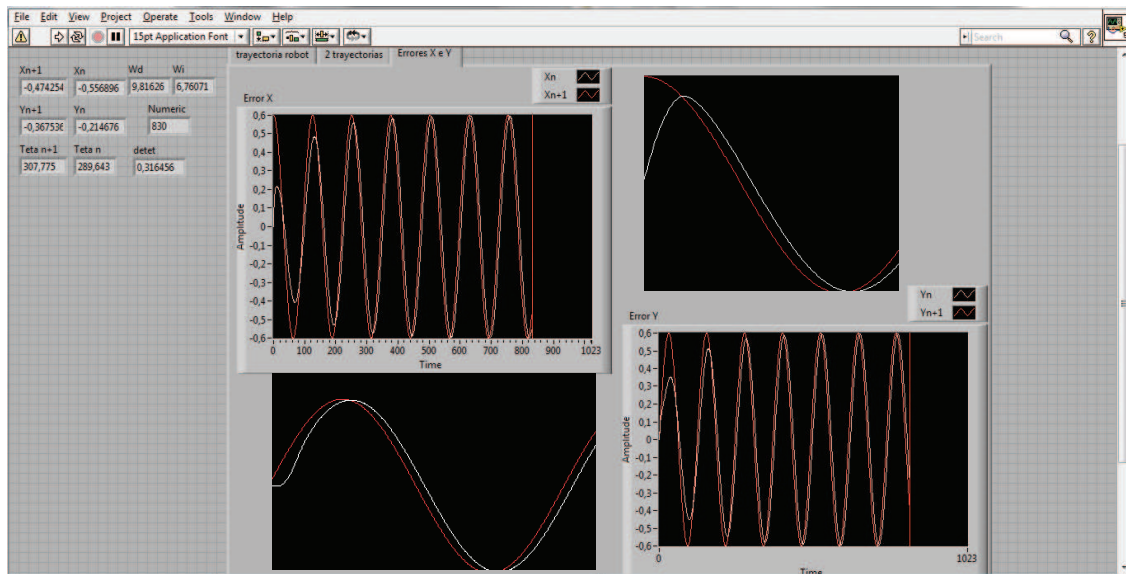


Figura 5.14 Error de seguimiento de las coordenadas X e Y del robot en base a la simulación.

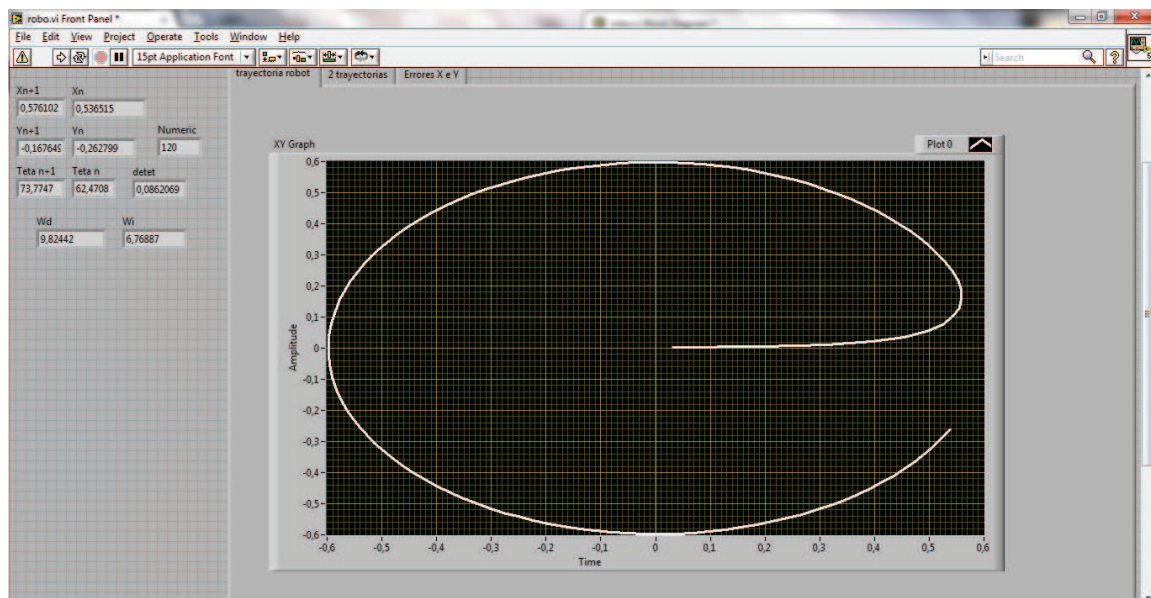


Figura 5.15 Simulación de la trayectoria del robot siguiendo una trayectoria circular.

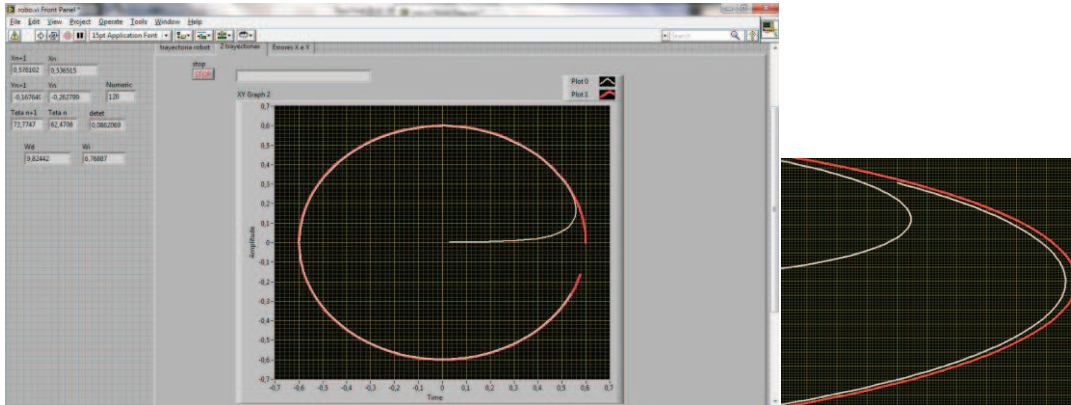


Figura 5.16 Comportamiento del robot siguiendo una trayectoria circular en base a la simulación.

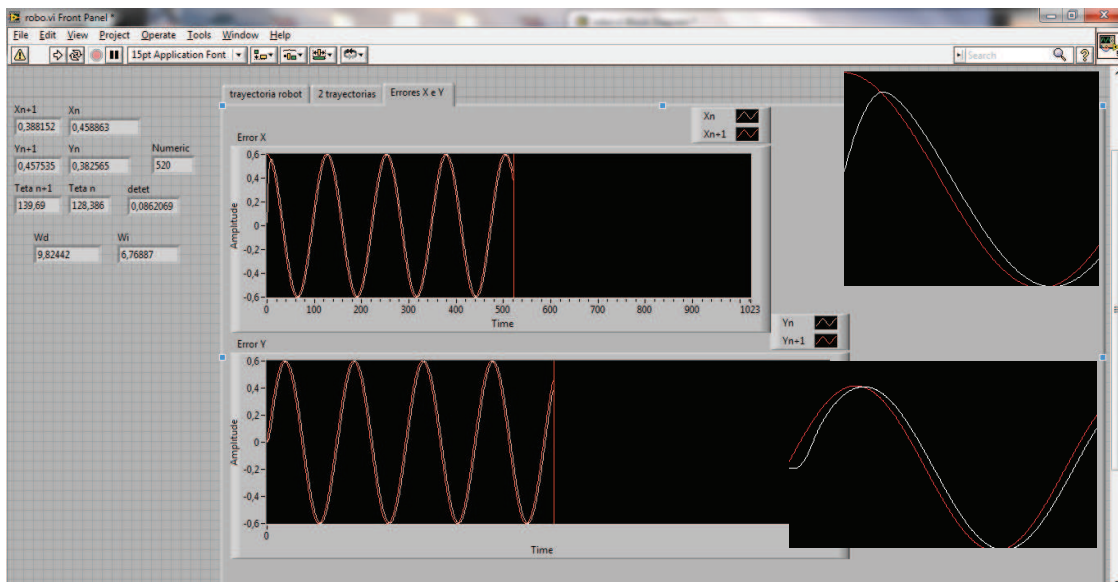


Figura 5.17 Error de seguimiento de las coordenadas  $X$  e  $Y$  del robot en base a la simulación.

Se puede mencionar en base a los resultados obtenidos en las gráficas de las Figuras 5.12 a 5.14, que el robot se demora alrededor de cuatro vueltas en alcanzar la trayectoria deseada, es decir, que en el momento en que la trayectoria circular está realizando su cuarta vuelta en el programa, el robot termina de alcanzarla, provocando que el error de seguimiento de la misma sea alto tanto en el desenvolvimiento del sistema (Figura 5.13), así como en el error de posición con respecto a las coordenadas  $X$  e  $Y$ . El error del sistema radica en la mala

interpretación al momento de usar la variable  $\theta_{n+1}$ , que no es lo mismo que  $\theta_{dn+1}$ , la primera corresponde a la dirección que tiene la trayectoria en el instante  $n+1$  y la segunda se refiere a la inclinación que debe poseer el robot en el instante  $n+1$ , si se emplea la primera se obtiene el error que se presentó en la Figura 5.13, además del error de posición propio del valor de  $K_v$  y  $K_w$ , si se emplea la segunda se obtiene el resultado de la Figura 5.16, que es lo que se pretende obtener como comportamiento, es decir, que el robot apenas arranque trate de alcanzar la trayectoria y así poder obtener un seguimiento casi instantáneo de la misma, reduciendo con esto también el error en la posición con respecto a las coordenadas  $X$  e  $Y$  (Figura 5.17).

Para apreciar la diferencia notoria en base al error de interpretación antes mencionado ambas simulaciones se realizaron con los mismos valores en las constantes  $K_v$  y  $K_w$  de 0,29 y 0,58 respectivamente y a un periodo de muestreo de 100mseg.

Con lo antes mencionado y con lo que se aprecia en las Figuras 5.15 a 5.17, se puede mencionar que se realiza la trayectoria circular, el error de seguimiento de la trayectoria por parte del robot es prácticamente de cero, así como también el error en la posición con respecto a las coordenadas  $X$  e  $Y$ .

A continuación se realizaran las pruebas en el robot real, para así poder ver las diferencias existentes entre las simulaciones y el comportamiento real del robot, y poder tener una mejor comprensión del comportamiento del sistema y en sí del seguimiento de trayectorias mediante la interpolación lineal.

## 5.3 PRUEBAS REALES

### 5.3.1 TRAYECTORIA RECTA

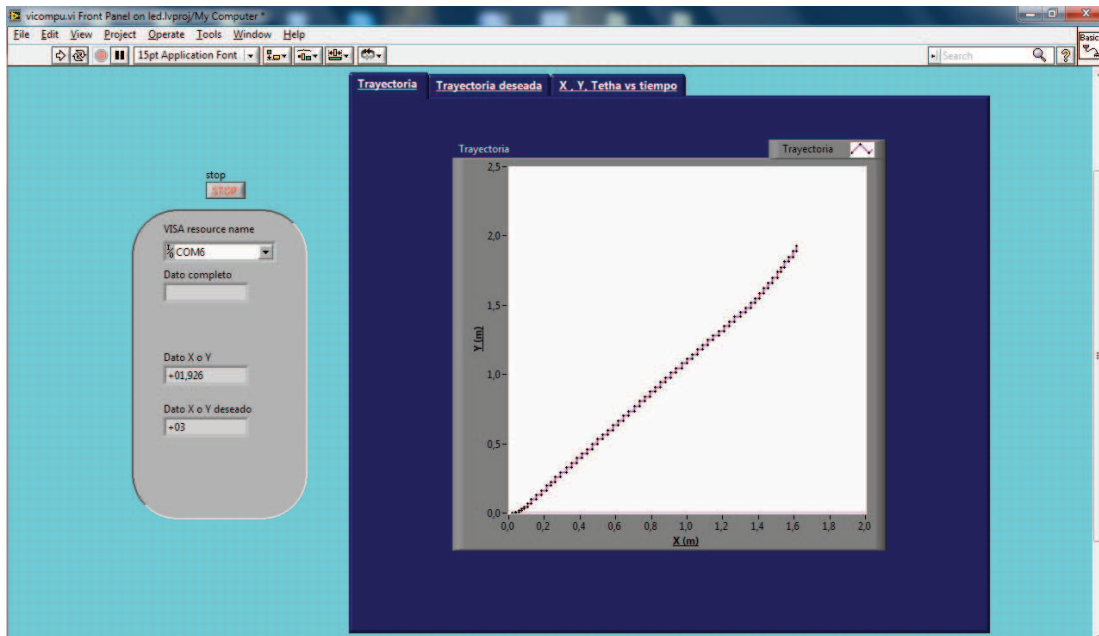


Figura 5.18 Comportamiento real del robot siguiendo una trayectoria recta.

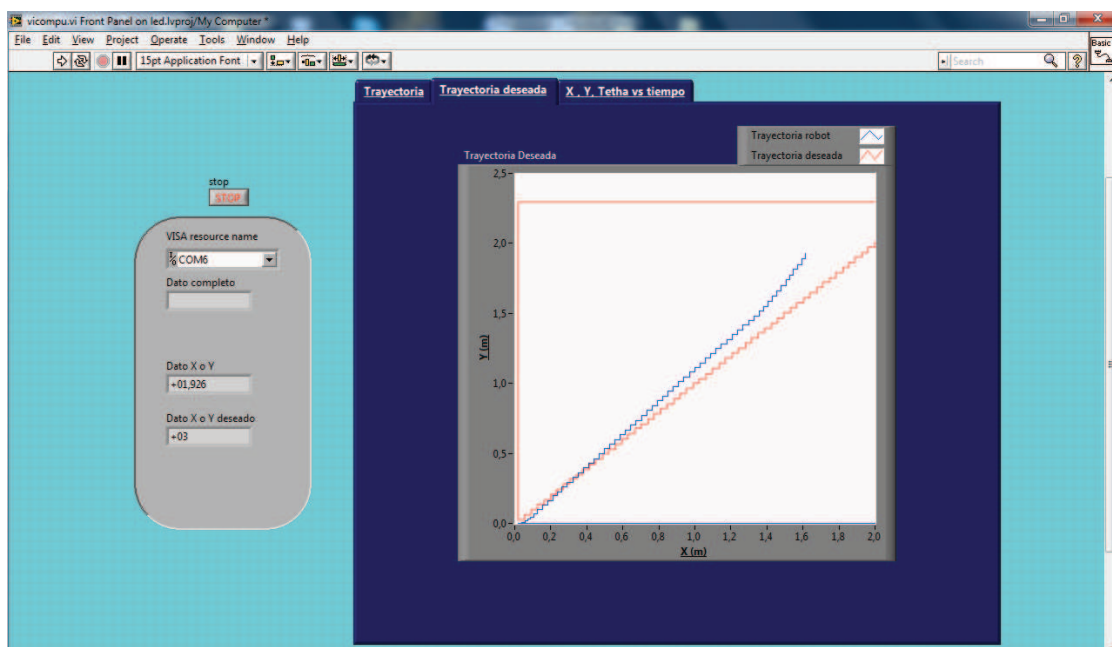


Figura 5.19 Comparación de las trayectorias real y deseada del robot siguiendo una trayectoria recta.

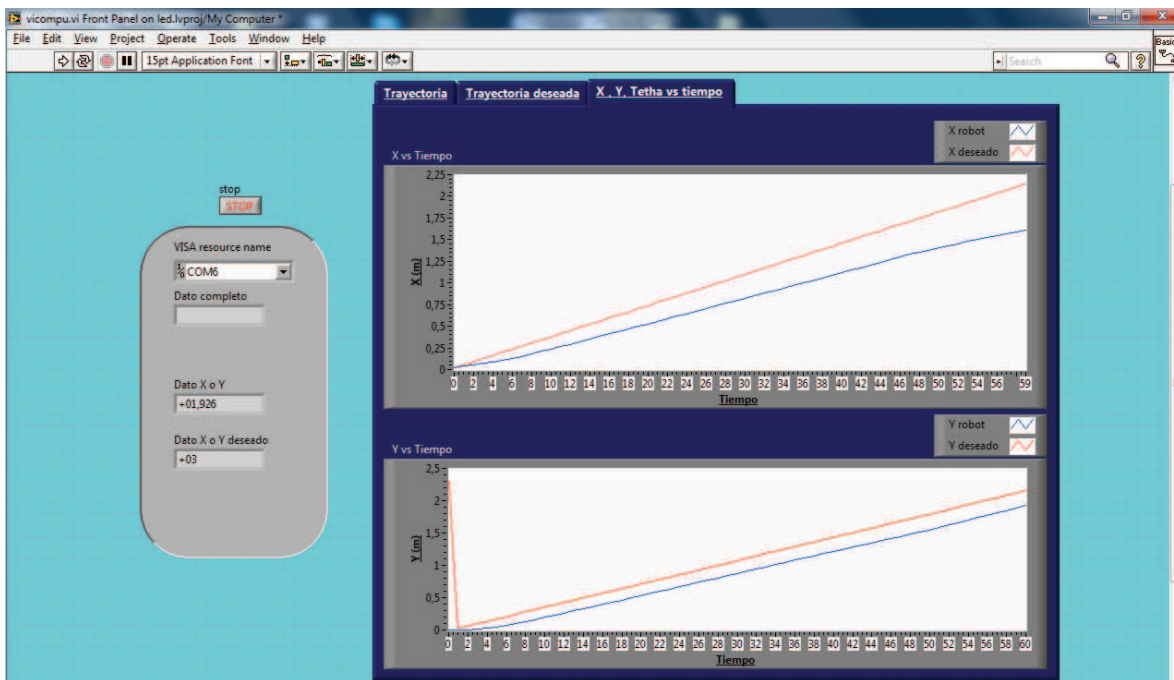


Figura 5.20 Presentación de los errores de seguimiento de las coordenadas X e Y con respecto al tiempo de la trayectoria real en relación a la trayectoria deseada del robot siguiendo una trayectoria recta.

En base a los resultados se puede decir que el robot realiza un recorrido con la forma de la trayectoria deseada (Figura 5.18) y que a cada instante de tiempo trata de estar sobre ella y a ningún momento se aleja demasiado como para llegar a perderla (Figura 5.19). Como se ve en la Figura 5.20, el robot sigue los valores de las coordenadas X e Y, cumpliendo así el objetivo del seguimiento, el error que se tiene en la coordenada en X se puede apreciar en la Tabla 5.1 y Figura 5.21; y el error en Y en la Tabla 5.2 y Figura 5.22, estos resultados se obtuvieron en base a valores en las constantes  $K_v$  y  $K_w$  de 0,1 y 0,58 respectivamente y a un periodo de muestreo de 100mseg.

Tabla 5.1 Error de posición en la coordenada X

Tiempo - X robot	X (m) - Trayectoria Robot	X (m) - Trayectoria deseada	Error (real - deseado)
0	0	0	0
1	0	0	0
2	0	0	0
3	0	0	0

4	0	0,1	-0,1
5	0	0,1	-0,1
6	0	0,1	-0,1
7	0	0,1	-0,1
8	0	0,1	-0,1
9	0	0,1	-0,1
10	0	0,1	-0,1
11	0	0,1	-0,1
12	0	0,2	-0,2
13	0	0,2	-0,2
14	0	0,2	-0,2
15	0	0,2	-0,2
16	0	0,2	-0,2
17	0	0,2	-0,2
18	0	0,2	-0,2
19	0	0,2	-0,2
20	0	0,3	-0,3
21	0	0,3	-0,3
22	0,1	0,3	-0,2
23	0,1	0,3	-0,2
24	0,1	0,3	-0,2
25	0,1	0,3	-0,2
26	0,1	0,3	-0,2
27	0,1	0,3	-0,2
28	0,1	0,4	-0,3
29	0,1	0,4	-0,3
30	0,1	0,4	-0,3
31	0,1	0,4	-0,3
32	0,1	0,4	-0,3
33	0,1	0,4	-0,3
34	0,1	0,4	-0,3
35	0,1	0,4	-0,3
36	0,1	0,5	-0,4
37	0,1	0,5	-0,4
38	0,1	0,5	-0,4
39	0,1	0,5	-0,4
40	0,1	0,5	-0,4
41	0,1	0,5	-0,4
42	0,2	0,5	-0,3
43	0,2	0,5	-0,3
44	0,2	0,6	-0,4
45	0,2	0,6	-0,4
46	0,2	0,6	-0,4
47	0,2	0,6	-0,4

48	0,2	0,6	-0,4
49	0,2	0,6	-0,4
50	0,2	0,6	-0,4
51	0,2	0,6	-0,4
52	0,2	0,6	-0,4
53	0,2	0,6	-0,4
54	0,3	0,6	-0,3
55	0,3	0,6	-0,3
56	0,3	0,7	-0,4
57	0,3	0,7	-0,4
58	0,3	0,7	-0,4
59	0,3	0,7	-0,4
60	0,3	0,7	-0,4
61	0,3	0,7	-0,4
62	0,4	0,7	-0,3
63	0,4	0,7	-0,3
64	0,4	0,8	-0,4
65	0,4	0,8	-0,4
66	0,4	0,8	-0,4
67	0,4	0,8	-0,4
68	0,4	0,8	-0,4
69	0,4	0,8	-0,4
70	0,4	0,8	-0,4
71	0,4	0,8	-0,4
72	0,4	0,9	-0,5
73	0,4	0,9	-0,5
74	0,5	0,9	-0,4
75	0,5	0,9	-0,4
76	0,5	0,9	-0,4
77	0,5	0,9	-0,4
78	0,5	0,9	-0,4
79	0,5	0,9	-0,4
80	0,5	1	-0,5
81	0,5	1	-0,5
82	0,6	1	-0,4
83	0,6	1	-0,4
84	0,6	1	-0,4
85	0,6	1	-0,4
86	0,6	1	-0,4
87	0,6	1	-0,4
88	0,6	1,1	-0,5
89	0,6	1,1	-0,5
90	0,7	1,1	-0,4
91	0,7	1,1	-0,4

92	0,7	1,1	-0,4
93	0,7	1,1	-0,4
94	0,7	1,1	-0,4
95	0,7	1,1	-0,4
96	0,7	1,2	-0,5
97	0,7	1,2	-0,5
98	0,7	1,2	-0,5

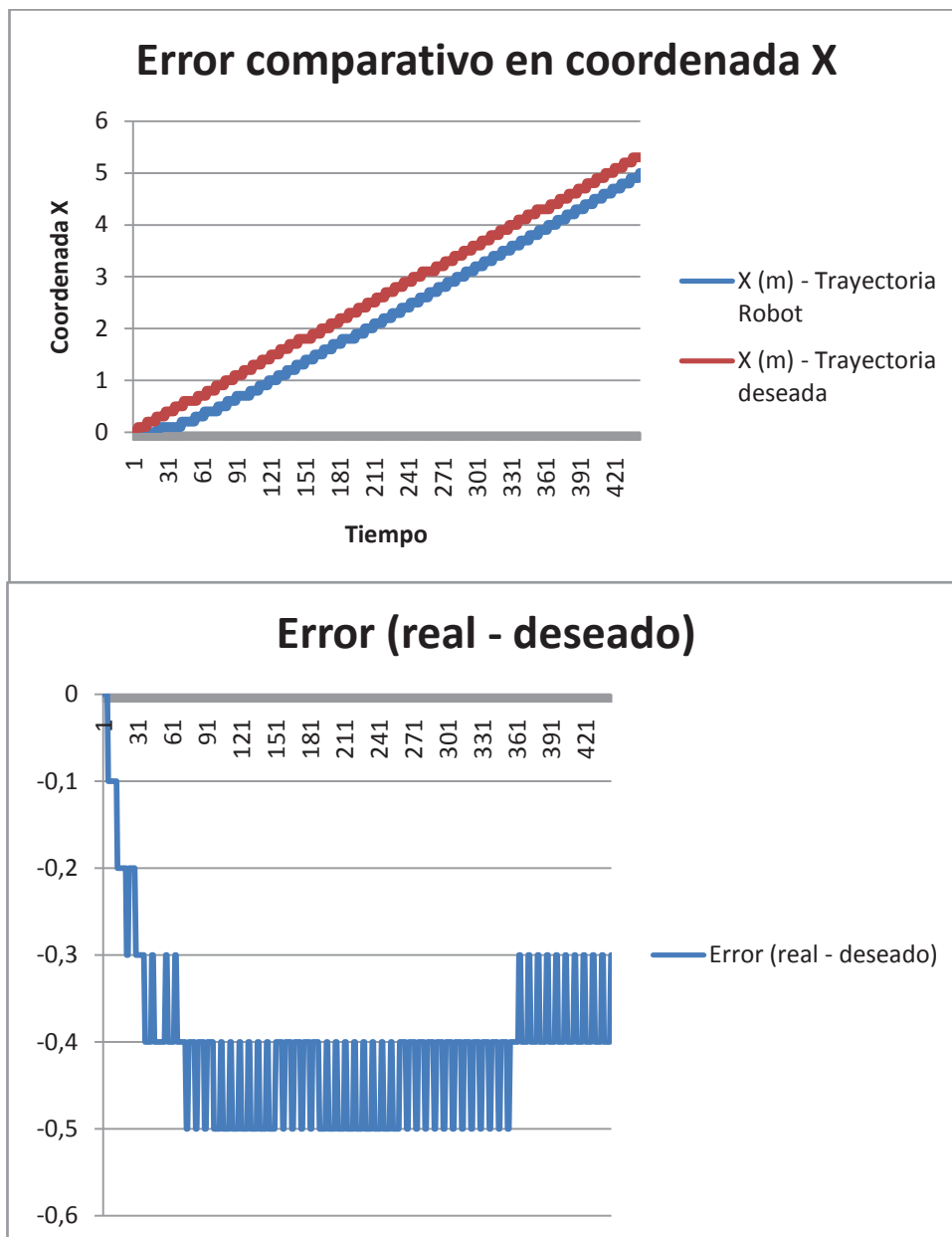


Figura 5.21 Error de posición de la coordenada en X



Tabla 5.2 Error de posición en la coordenada Y

Tiempo - Y robot	Y (m) - Trayectoria Robot	Y (m) - Trayectoria deseada	Error (real - deseado)
0	0	0	0
1	0	0	0
2	0	0	0
3	0	0	0
4	0	0	0
5	0	0	0
6	0	0,1	-0,1
7	0	0,1	-0,1
8	0	0,1	-0,1
9	0	0,1	-0,1
10	0	0,1	-0,1
11	0	0,1	-0,1
12	0	0,1	-0,1
13	0	0,1	-0,1
14	0	0,2	-0,2
15	0	0,2	-0,2
16	0	0,2	-0,2
17	0	0,2	-0,2
18	0	0,2	-0,2
19	0	0,2	-0,2
20	0	0,2	-0,2
21	0	0,2	-0,2
22	0	0,3	-0,3
23	0	0,3	-0,3
24	0	0,3	-0,3
25	0	0,3	-0,3
26	0	0,3	-0,3
27	0	0,3	-0,3
28	0	0,3	-0,3
29	0	0,3	-0,3
30	0	0,4	-0,4
31	0	0,4	-0,4
32	0,1	0,4	-0,3
33	0,1	0,4	-0,3
34	0,1	0,4	-0,3
35	0,1	0,4	-0,3
36	0,1	0,4	-0,3
37	0,1	0,4	-0,3
38	0,1	0,5	-0,4
39	0,1	0,5	-0,4
40	0,1	0,5	-0,4

41	0,1	0,5	-0,4
42	0,1	0,5	-0,4
43	0,1	0,5	-0,4
44	0,2	0,5	-0,3
45	0,2	0,5	-0,3
46	0,2	0,6	-0,4
47	0,2	0,6	-0,4
48	0,2	0,6	-0,4
49	0,2	0,6	-0,4
50	0,2	0,6	-0,4
51	0,2	0,6	-0,4
52	0,3	0,6	-0,3
53	0,3	0,6	-0,3
54	0,3	0,7	-0,4
55	0,3	0,7	-0,4
56	0,3	0,7	-0,4
57	0,3	0,7	-0,4
58	0,3	0,7	-0,4
59	0,3	0,7	-0,4
60	0,3	0,7	-0,4
61	0,3	0,7	-0,4
62	0,3	0,8	-0,5
63	0,3	0,8	-0,5
64	0,4	0,8	-0,4
65	0,4	0,8	-0,4
66	0,4	0,8	-0,4
67	0,4	0,8	-0,4
68	0,4	0,8	-0,4
69	0,4	0,8	-0,4
70	0,4	0,9	-0,5
71	0,4	0,9	-0,5
72	0,5	0,9	-0,4
73	0,5	0,9	-0,4
74	0,5	0,9	-0,4
75	0,5	0,9	-0,4
76	0,5	0,9	-0,4
77	0,5	0,9	-0,4
78	0,5	0,9	-0,4
79	0,5	0,9	-0,4
80	0,6	0,9	-0,3
81	0,6	0,9	-0,3
82	0,6	1	-0,4
83	0,6	1	-0,4
84	0,6	1	-0,4

85	0,6	1	-0,4
86	0,6	1	-0,4
87	0,6	1	-0,4
88	0,7	1	-0,3
89	0,7	1	-0,3
90	0,7	1,1	-0,4
91	0,7	1,1	-0,4
92	0,7	1,1	-0,4
93	0,7	1,1	-0,4
94	0,7	1,1	-0,4
95	0,7	1,1	-0,4
96	0,8	1,1	-0,3
97	0,8	1,1	-0,3
98	0,8	1,2	-0,4
99	0,8	1,2	-0,4
100	0,8	1,2	-0,4

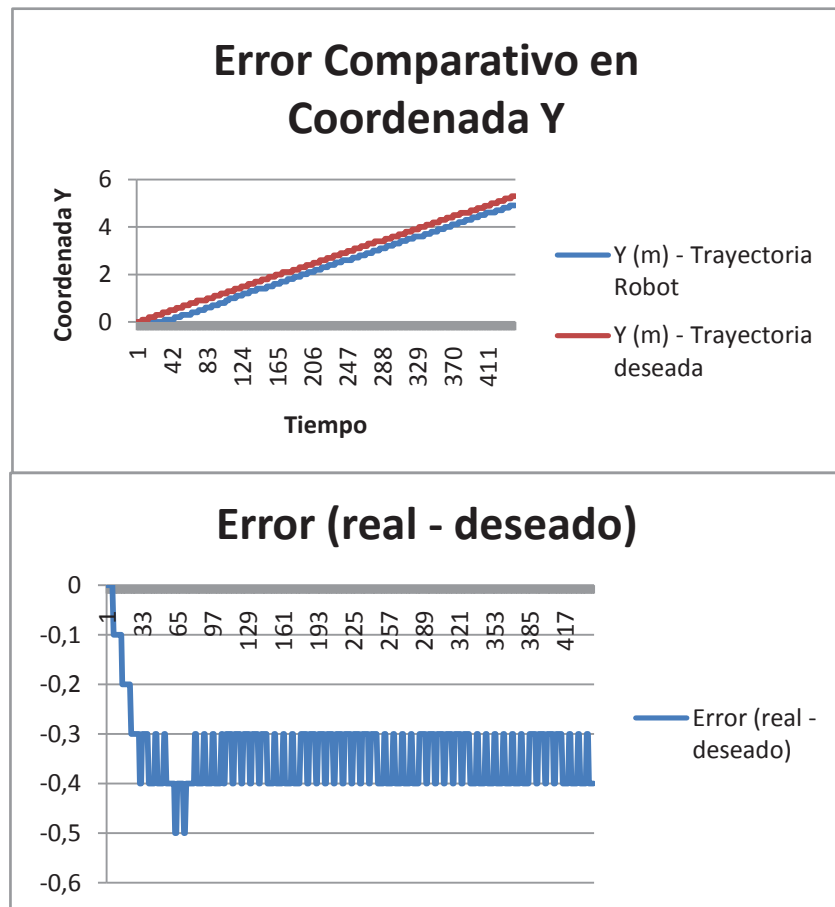


Figura 5.22 Error de posición de la coordenada en X

## 5.3.2 TRAYECTORIA CIRCULAR

### 5.3.2.1 PRUEBA 1

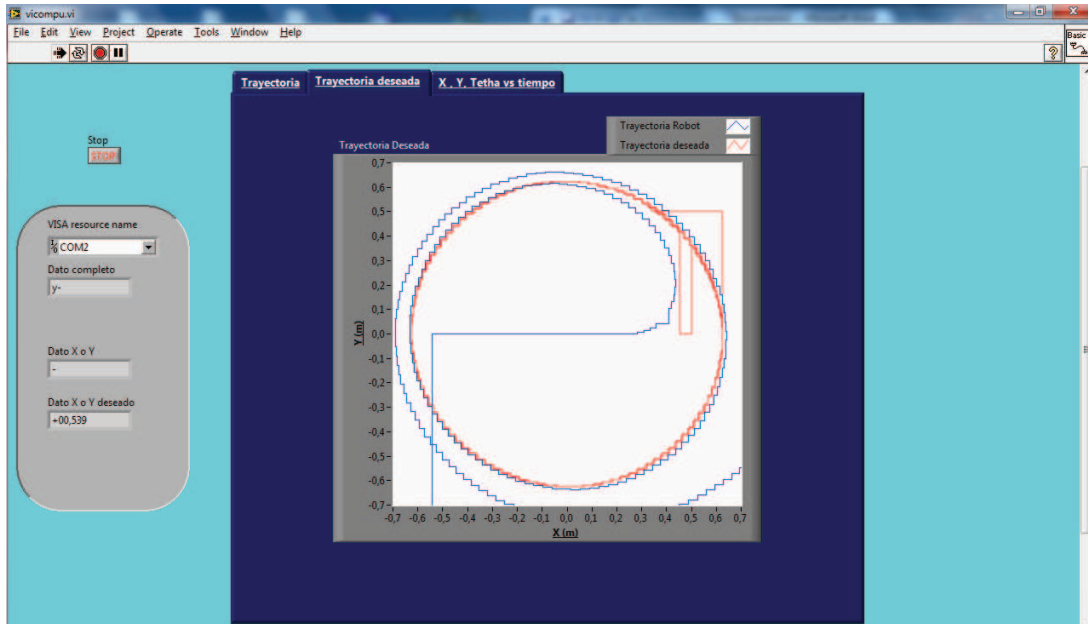


Figura 5.23 Comportamiento real del robot siguiendo una trayectoria circular.

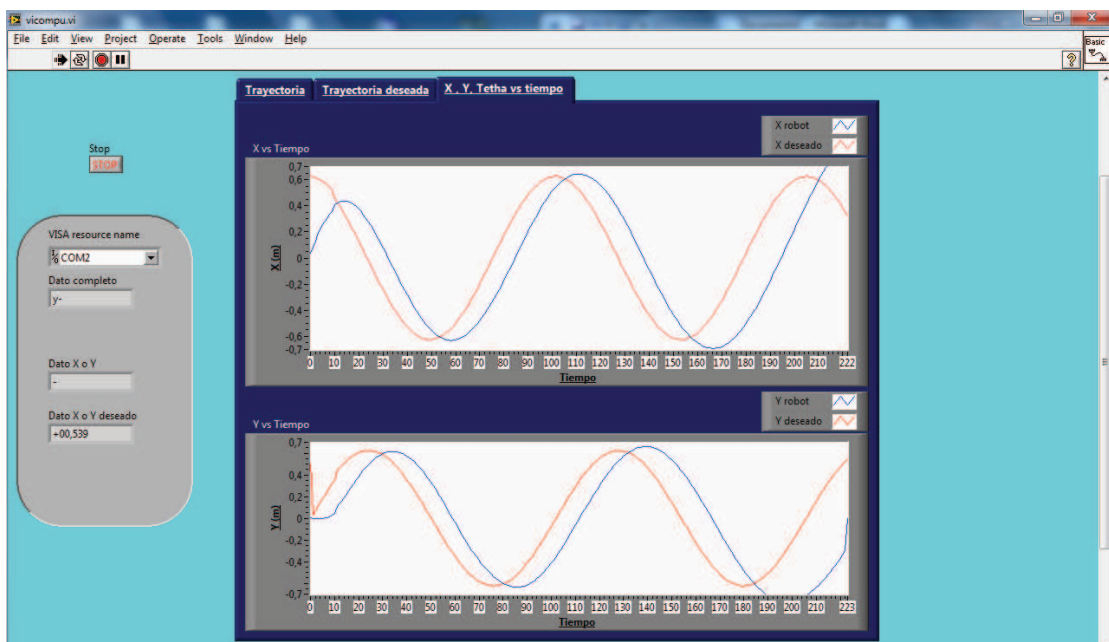


Figura 5.24 Comparación de las trayectorias real y deseada del robot siguiendo una trayectoria circular.

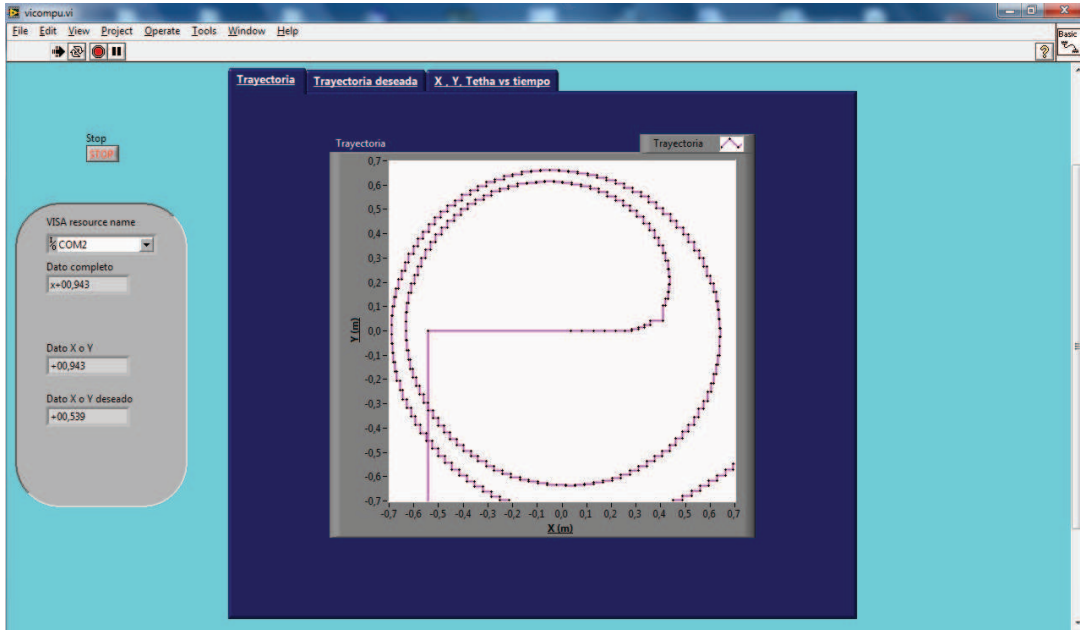


Figura 5.25 Presentación de los errores de seguimiento de las coordenadas X e Y con respecto al tiempo de la trayectoria real en relación a la trayectoria deseada del robot siguiendo una trayectoria circular.

En base a los resultados se puede decir que a pesar que el robot realiza una trayectoria circular (Figura 5.23), no sigue la trayectoria en cada instante de tiempo (Figura 5.24), ya que, se puede apreciar que después de alcanzarla, el robot no logra mantenerse sobre el recorrido de la misma, en su lugar simplemente se mueve paralelamente a ella realizando una espiral.

El error radica en el seguimiento de ambas coordenada en X e Y como se aprecia en la Figura 5.25, estos resultados se obtuvieron en base a valores en las constantes  $K_v$  y  $K_w$  de 0,2 y 0,4 respectivamente y a un periodo de muestreo de 100mseg. El error que se tiene en la coordenada en X se puede apreciar en la Tabla 5.3 y Figura 5.26; y el error en Y en la Tabla 5.4 y Figura 5.27

Tabla 5.3 Error de posición en la coordenada X

Tiempo - X robot	X (m) - Trayectoria Robot	Y (m) - Trayectoria Robot	Error (real - deseado)
0	0,2	0	0,2
1	0,2	0	0,2
2	0,2	0	0,2
3	0,2	0	0,2

4	0,2	0	0,2
5	0,2	0	0,2
6	0,2	0	0,2
7	0,2	0	0,2
8	0,2	0	0,2
9	0,2	0	0,2
10	0,2	0	0,2
11	0,2	0	0,2
12	0,2	0	0,2
13	0,2	0	0,2
14	0,2	0	0,2
15	0,2	0	0,2
16	0,2	0	0,2
17	0,2	0	0,2
18	0,2	0	0,2
19	0,2	0	0,2
20	0,2	0	0,2
21	0,2	0	0,2
22	0,2	0	0,2
23	0,2	0	0,2
24	0,2	0	0,2
25	0,2	0	0,2
26	0,2	0	0,2
27	0,2	0	0,2
28	0,2	0	0,2
29	0,2	0	0,2
30	0,2	0	0,2
31	0,2	0	0,2
32	0,2	0	0,2
33	0,2	0	0,2
34	0,2	0	0,2
35	0,2	0	0,2
36	0,2	0	0,2
37	0,2	0	0,2
38	0,2	0	0,2
39	0,2	0	0,2
40	0,2	0	0,2
41	0,2	0	0,2
42	0,2	0	0,2
43	0	0	0
44	0	0	0
45	0	0	0
46	0	0	0
47	0,1	0	0,1

48	0,1	0	0,1
49	0,1	0	0,1
50	0,1	0	0,1
51	0,1	0	0,1
52	0,1	0	0,1
53	0,1	0	0,1
54	0,1	0	0,1
55	0,2	0	0,2
56	0,2	0	0,2
57	0,2	0	0,2
58	0,2	0	0,2
59	0,2	0	0,2
60	0,2	0	0,2
61	0,2	0	0,2
62	0,2	0	0,2
63	0,2	0	0,2
64	0,2	0	0,2
65	0,2	0	0,2
66	0,2	0	0,2
67	0,3	0	0,3
68	0,3	0	0,3
69	0,3	0	0,3
70	0,3	0	0,3
71	0,3	0	0,3
72	0,3	0	0,3
73	0,3	0	0,3
74	0,3	0	0,3
75	0,3	0	0,3
76	0,3	0	0,3
77	0,3	0	0,3
78	0,3	0	0,3
79	0,3	0	0,3
80	0,3	0	0,3
81	0,3	0	0,3
82	0,3	0	0,3
83	0,3	0	0,3
84	0,3	0,1	0,2
85	0,3	0,1	0,2
86	0,3	0,1	0,2
87	0,4	0,1	0,3
88	0,4	0,1	0,3
89	0,4	0,1	0,3
90	0,4	0,1	0,3
91	0,4	0,1	0,3

92	0,4	0,1	0,3
93	0,4	0,1	0,3
94	0,4	0,1	0,3
95	0,4	0,1	0,3
96	0,4	0,1	0,3
97	0,4	0,1	0,3
98	0,4	0,1	0,3
99	0,4	0,1	0,3
100	0,4	0,1	0,3

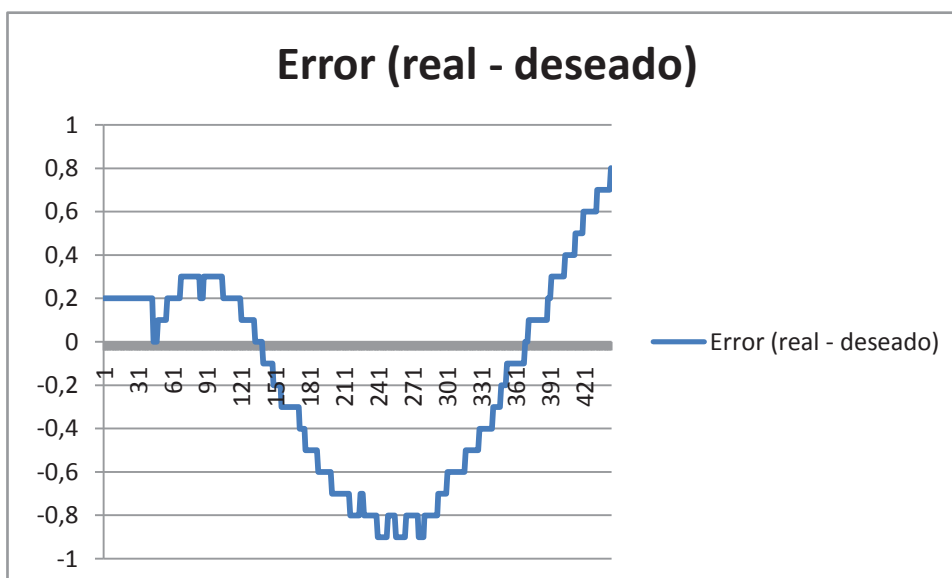
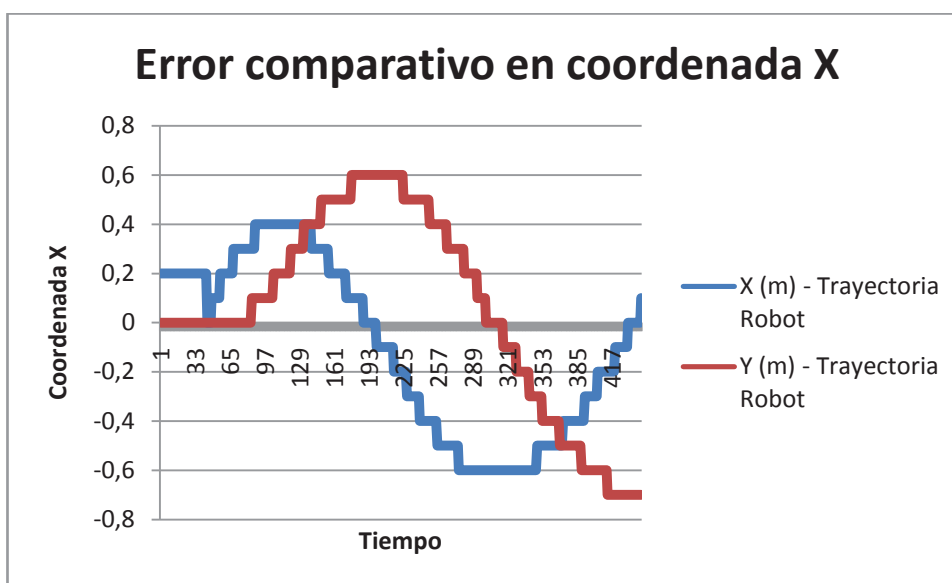


Figura 5.26 Error de posición de la coordenada en X



Tabla 5.4 Error de posición en la coordenada Y

Tiempo - X robot	X (m) - Trayectoria deseada	Y (m) - Trayectoria deseada	Error (real - deseado)
0	-0,5	-0,3	0,2
1	-0,5	-0,3	0,2
2	-0,5	-0,4	0,1
3	-0,5	-0,4	0,1
4	-0,5	-0,4	0,1
5	-0,5	-0,4	0,1
6	-0,5	-0,4	0,1
7	-0,5	-0,4	0,1
8	-0,5	-0,4	0,1
9	-0,5	-0,4	0,1
10	-0,5	-0,4	0,1
11	-0,5	-0,4	0,1
12	-0,5	-0,4	0,1
13	-0,5	-0,4	0,1
14	-0,5	-0,4	0,1
15	-0,5	-0,4	0,1
16	-0,5	-0,4	0,1
17	-0,4	-0,4	0
18	-0,4	-0,5	-0,1
19	-0,4	-0,5	-0,1
20	-0,4	-0,5	-0,1
21	-0,4	-0,5	-0,1
22	-0,4	-0,5	-0,1
23	-0,4	-0,5	-0,1
24	-0,4	-0,5	-0,1
25	-0,4	-0,5	-0,1
26	-0,4	-0,5	-0,1
27	-0,4	-0,5	-0,1
28	-0,4	-0,5	-0,1
29	-0,3	-0,5	-0,2
30	-0,3	-0,5	-0,2
31	-0,3	-0,5	-0,2
32	-0,3	-0,5	-0,2
33	-0,3	-0,5	-0,2
34	-0,3	-0,6	-0,3
35	-0,3	-0,6	-0,3
36	-0,3	-0,6	-0,3
37	-0,3	-0,6	-0,3
38	-0,3	0	0,3
39	-0,3	0	0,3
40	-0,3	0	0,3

41	0,6	0	-0,6
42	0,6	0	-0,6
43	0,6	0	-0,6
44	0,6	0	-0,6
45	0,6	0	-0,6
46	0,6	0,1	-0,5
47	0,6	0,1	-0,5
48	0,6	0,1	-0,5
49	0,6	0,1	-0,5
50	0,6	0,1	-0,5
51	0,6	0,1	-0,5
52	0,6	0,1	-0,5
53	0,6	0,1	-0,5
54	0,6	0,1	-0,5
55	0,6	0,1	-0,5
56	0,6	0,1	-0,5
57	0,6	0,1	-0,5
58	0,6	0,2	-0,4
59	0,6	0,2	-0,4
60	0,6	0,2	-0,4
61	0,6	0,2	-0,4
62	0,6	0,2	-0,4
63	0,6	0,2	-0,4
64	0,6	0,2	-0,4
65	0,6	0,2	-0,4
66	0,6	0,2	-0,4
67	0,6	0,2	-0,4
68	0,6	0,2	-0,4
69	0,6	0,2	-0,4
70	0,6	0,3	-0,3
71	0,6	0,3	-0,3
72	0,6	0,3	-0,3
73	0,5	0,3	-0,2
74	0,5	0,3	-0,2
75	0,5	0,3	-0,2
76	0,5	0,3	-0,2
77	0,5	0,3	-0,2
78	0,5	0,3	-0,2
79	0,5	0,3	-0,2
80	0,5	0,3	-0,2
81	0,5	0,3	-0,2
82	0,5	0,4	-0,1
83	0,5	0,4	-0,1
84	0,5	0,4	-0,1

85	0,5	0,4	-0,1
86	0,5	0,4	-0,1
87	0,5	0,4	-0,1
88	0,5	0,4	-0,1
89	0,5	0,4	-0,1
90	0,5	0,4	-0,1
91	0,5	0,4	-0,1
92	0,5	0,4	-0,1
93	0,4	0,4	0
94	0,4	0,5	0,1
95	0,4	0,5	0,1
96	0,4	0,5	0,1
97	0,4	0,5	0,1
98	0,4	0,5	0,1
99	0,4	0,5	0,1
100	0,4	0,5	0,1
101	0,4	0,5	0,1
102	0,4	0,5	0,1
103	0,4	0,5	0,1
104	0,4	0,5	0,1
105	0,3	0,5	0,2
106	0,3	0,5	0,2
107	0,3	0,5	0,2
108	0,3	0,5	0,2
109	0,3	0,5	0,2
110	0,3	0,5	0,2
111	0,3	0,5	0,2
112	0,3	0,5	0,2
113	0,3	0,5	0,2
114	0,3	0,6	0,3
115	0,3	0,6	0,3
116	0,3	0,6	0,3
117	0,2	0,6	0,4
118	0,2	0,6	0,4
119	0,2	0,6	0,4
120	0,2	0,6	0,4
121	0,2	0,6	0,4
122	0,2	0,6	0,4
123	0,2	0,6	0,4
124	0,2	0,6	0,4
125	0,2	0,6	0,4
126	0,2	0,6	0,4
127	0,2	0,6	0,4
128	0,2	0,6	0,4

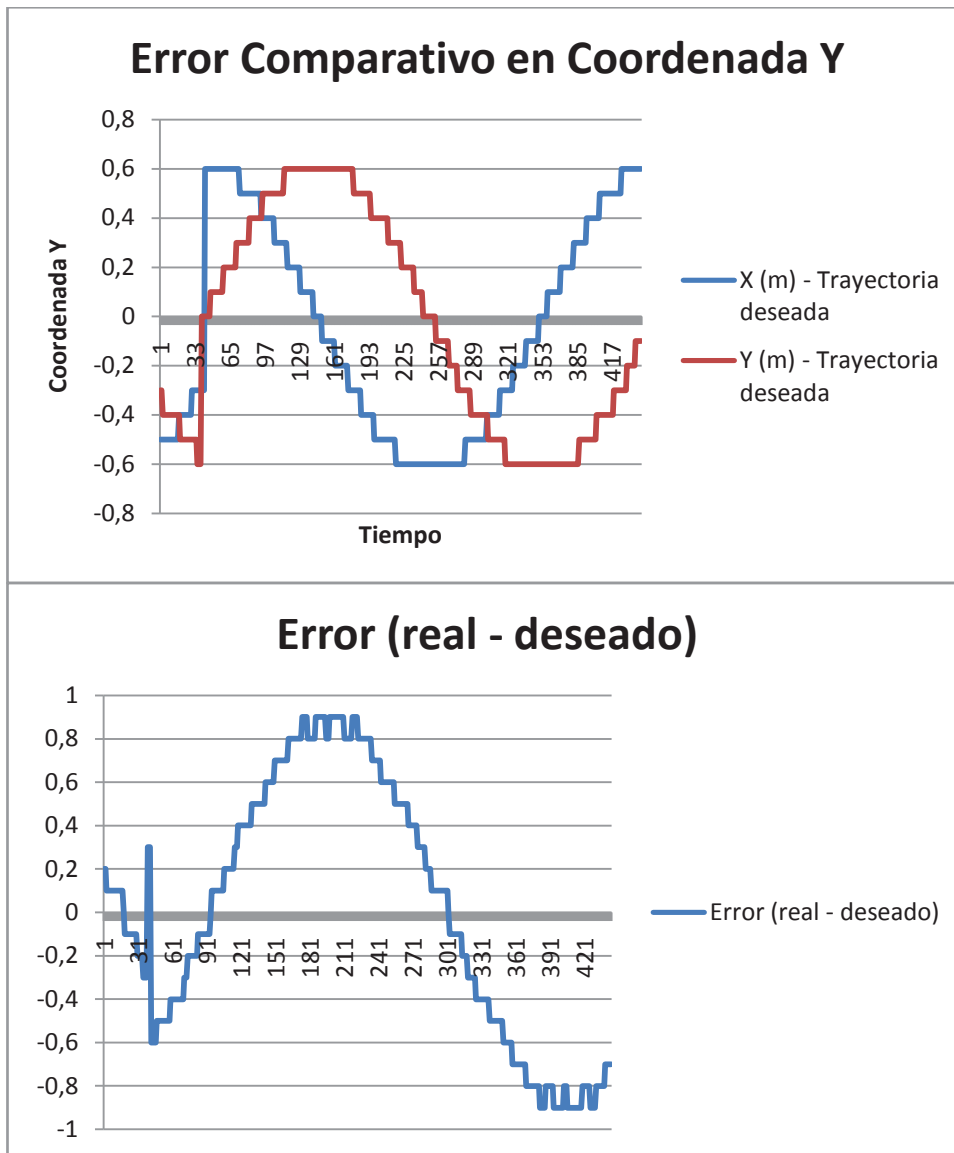


Figura 5.27 Error de posición de la coordenada en Y

### 5.3.2.2 PRUEBA 2

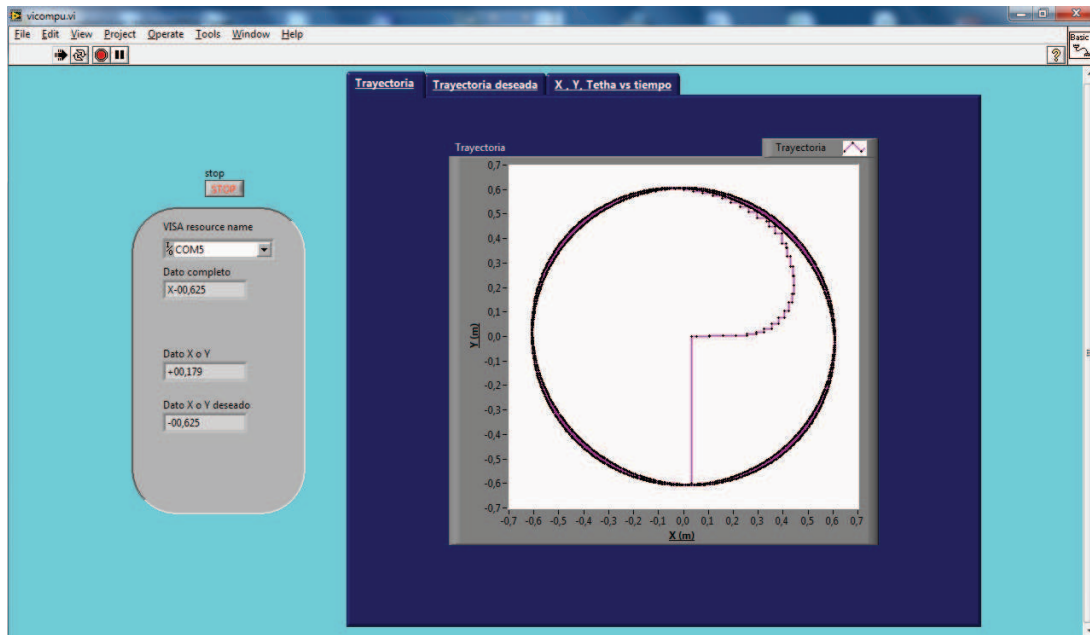


Figura 5.28 Comportamiento real del robot siguiendo una trayectoria circular.

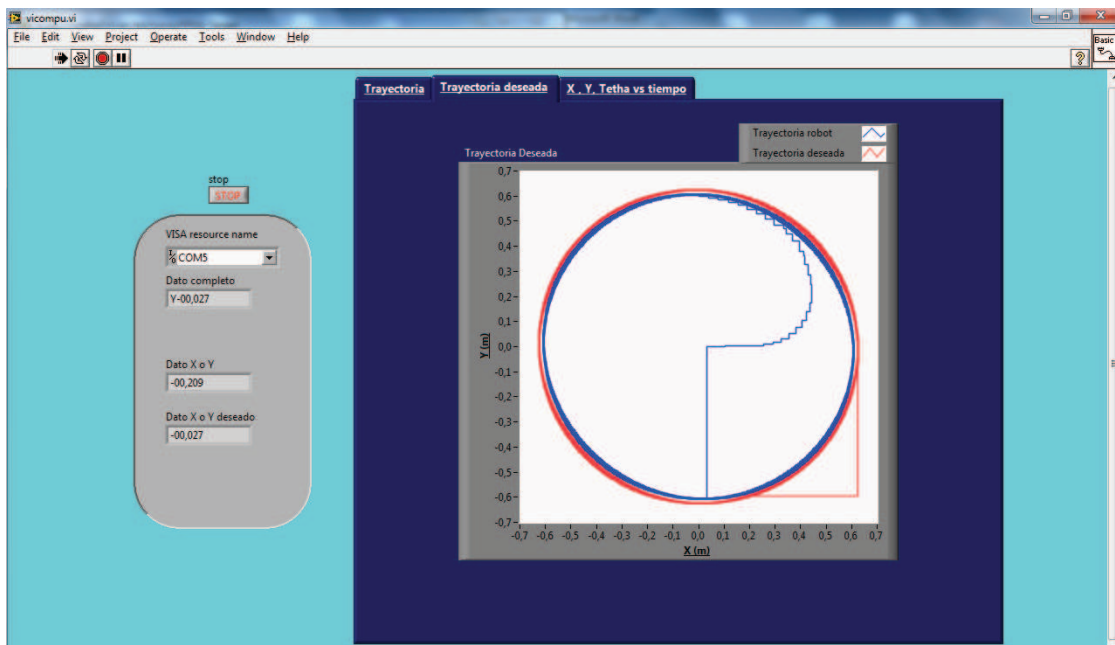


Figura 5.29 Comparación de las trayectorias real y deseada del robot siguiendo una trayectoria circular.

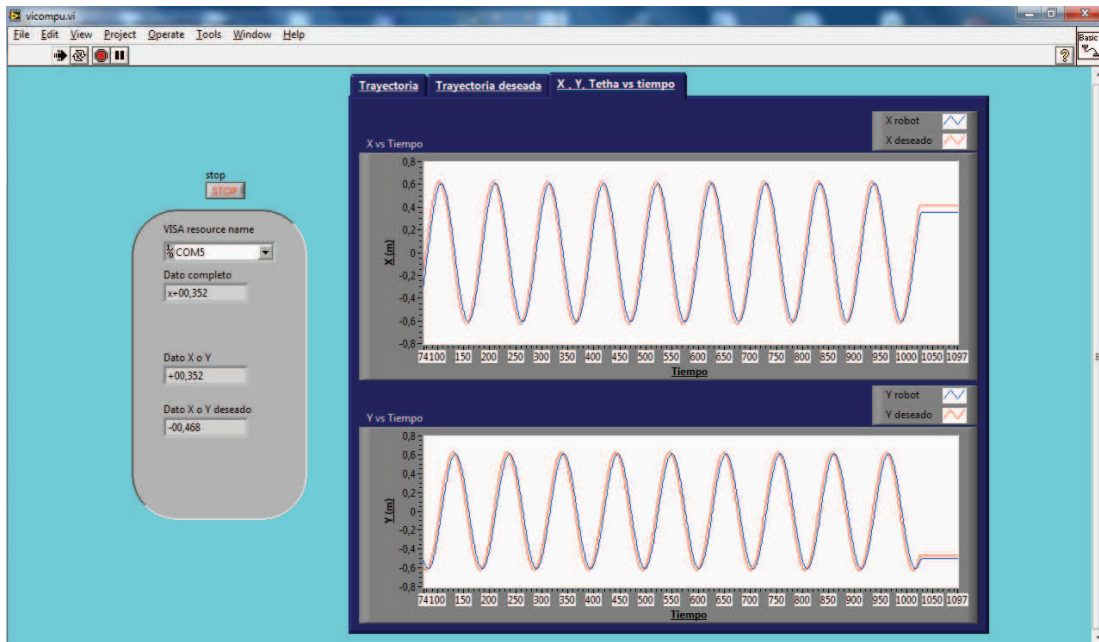


Figura 5.30 Presentación de los errores de seguimiento de las coordenadas  $X$  e  $Y$  con respecto al tiempo de la trayectoria real en relación a la trayectoria deseada del robot siguiendo una trayectoria circular.

En base a lo mencionado como resultados de la Prueba 1 y como se puede apreciar en la Figura 5.29, al variar los valores de las constantes  $K_v$  y  $K_w$  a 0,07 y 0,58 respectivamente, se obtiene una mejor respuesta del sistema, el error que se tiene en la coordenada en  $X$  se puede apreciar en la Tabla 5.5 y Figura 5.31; y el error en  $Y$  en la Tabla 5.6 y Figura 5.32, y así el robot a cada instante de tiempo está siguiendo la trayectoria, esto se puede comprobar al observar la Figura 5.30, en la cual los errores de posición con respecto a las coordenadas  $X$  e  $Y$ , son prácticamente cero; con esto se cumplió satisfactoriamente el objetivo del seguimiento, que radica en que el robot lleve a cabo la trayectoria fijada con el menor error de posición posible.

Tabla 5.5 Error de posición en la coordenada X

Tiempo - X robot	X (m) - X robot	X (m) - X deseado	error (real - deseado)
0	0,05	0,624	-0,574
1	0,102	0,622	-0,52
2	0,155	0,617	-0,462
3	0,207	0,61	-0,403
4	0,259	0,601	-0,342
5	0,303	0,59	-0,287
6	0,335	0,577	-0,242
7	0,364	0,561	-0,197
8	0,393	0,543	-0,15
9	0,419	0,524	-0,105
10	0,441	0,502	-0,061
11	0,459	0,479	-0,02
12	0,473	0,454	0,019
13	0,482	0,427	0,055
14	0,486	0,399	0,087
15	0,486	0,369	0,117
16	0,481	0,338	0,143
17	0,471	0,305	0,166
18	0,457	0,272	0,185
19	0,439	0,237	0,202
20	0,418	0,202	0,216
21	0,392	0,166	0,226
22	0,364	0,129	0,235
23	0,333	0,092	0,241
24	0,299	0,054	0,245
25	0,262	0,016	0,246
26	0,224	-0,021	0,245
27	0,184	-0,058	0,242
28	0,142	-0,096	0,238
29	0,1	-0,133	0,233
30	0,056	-0,17	0,226
31	0,013	-0,206	0,219
32	-0,03	-0,241	0,211
33	-0,074	-0,276	0,202
34	-0,118	-0,309	0,191
35	-0,161	-0,341	0,18
36	-0,204	-0,372	0,168
37	-0,245	-0,402	0,157
38	-0,285	-0,43	0,145
39	-0,324	-0,457	0,133

40	-0,361	-0,482	0,121
41	-0,397	-0,505	0,108
42	-0,43	-0,526	0,096
43	-0,461	-0,546	0,085
44	-0,49	-0,563	0,073
45	-0,516	-0,579	0,063
46	-0,54	-0,592	0,052
47	-0,561	-0,603	0,042
48	-0,579	-0,611	0,032
49	-0,594	-0,618	0,024
50	-0,606	-0,622	0,016
51	-0,616	-0,624	0,008
52	-0,622	-0,624	0,002
53	-0,625	-0,621	-0,004
54	-0,625	-0,617	-0,008
55	-0,622	-0,609	-0,013
56	-0,617	-0,6	-0,017
57	-0,608	-0,589	-0,019
58	-0,598	-0,575	-0,023
59	-0,587	-0,559	-0,028
60	-0,573	-0,541	-0,032
61	-0,557	-0,521	-0,036
62	-0,539	-0,5	-0,039
63	-0,519	-0,476	-0,043
64	-0,497	-0,451	-0,046
65	-0,474	-0,424	-0,05
66	-0,449	-0,395	-0,054
67	-0,422	-0,365	-0,057
68	-0,393	-0,334	-0,059
69	-0,364	-0,302	-0,062
70	-0,332	-0,268	-0,064
71	-0,3	-0,233	-0,067
72	-0,267	-0,198	-0,069
73	-0,232	-0,161	-0,071
74	-0,197	-0,125	-0,072
75	-0,16	-0,087	-0,073
76	-0,124	-0,05	-0,074
77	-0,087	-0,012	-0,075
78	-0,05	0,025	-0,075
79	-0,012	0,063	-0,075
80	0,024	0,1	-0,076



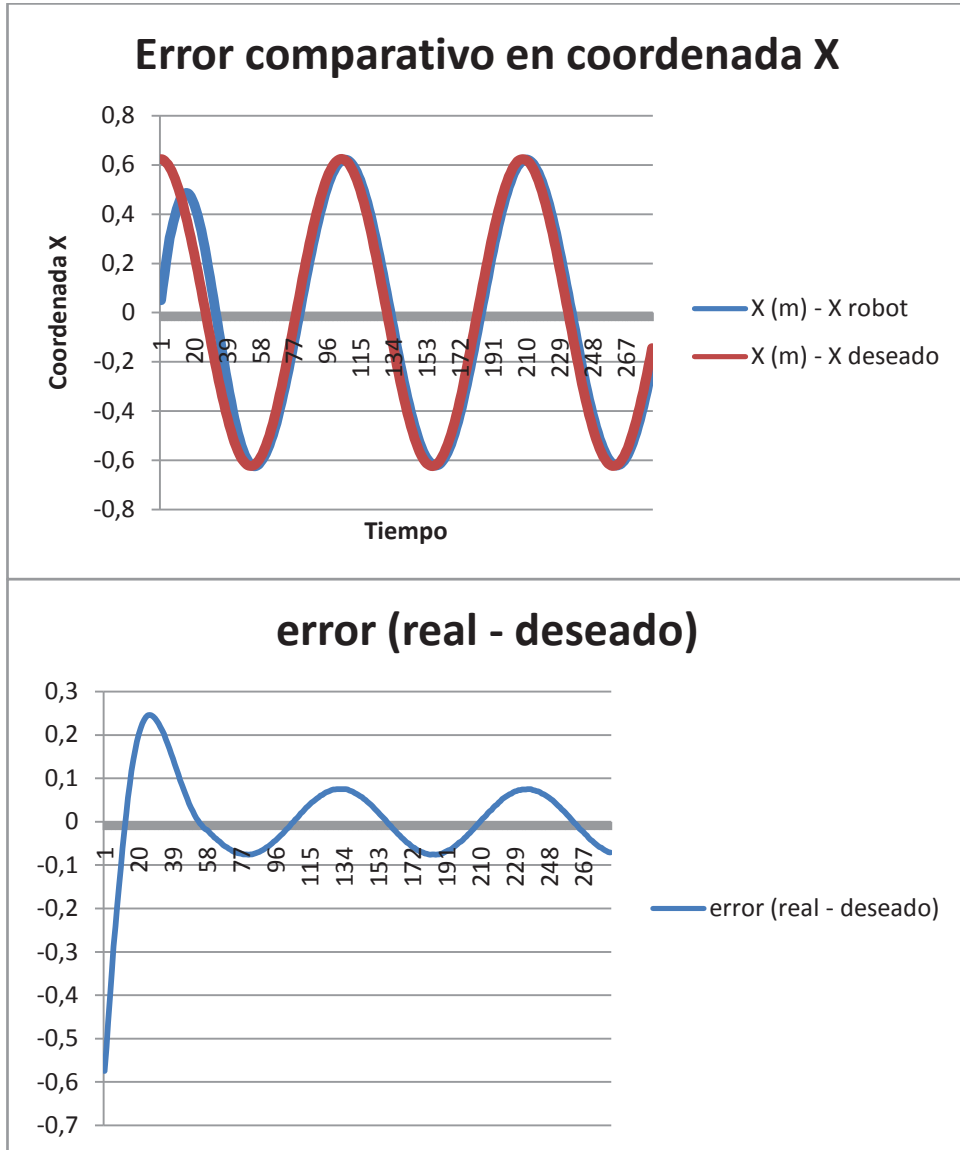


Figura 5.31 Error de posición de la coordenada en X

Tabla 5.6 Error de posición en la coordenada Y

Tiempo - Y robot	Y (m) - Y robot	Y (m) - Y deseado	error (real-deseado)
0	0,001	0	0,001
1	0,001	0,028	-0,027
2	0,001	0,066	-0,065
3	0,001	0,103	-0,102
4	0,001	0,14	-0,139
5	0,001	0,177	-0,176
6	0,003	0,213	-0,21
7	0,009	0,248	-0,239
8	0,02	0,282	-0,262
9	0,038	0,315	-0,277

10	0,06	0,347	-0,287
11	0,087	0,378	-0,291
12	0,118	0,408	-0,29
13	0,151	0,436	-0,285
14	0,187	0,462	-0,275
15	0,224	0,486	-0,262
16	0,262	0,509	-0,247
17	0,301	0,53	-0,229
18	0,339	0,549	-0,21
19	0,377	0,566	-0,189
20	0,413	0,581	-0,168
21	0,448	0,594	-0,146
22	0,481	0,604	-0,123
23	0,512	0,613	-0,101
24	0,54	0,619	-0,079
25	0,565	0,623	-0,058
26	0,588	0,624	-0,036
27	0,607	0,624	-0,017
28	0,624	0,621	0,003
29	0,637	0,616	0,021
30	0,647	0,608	0,039
31	0,653	0,598	0,055
32	0,657	0,586	0,071
33	0,656	0,572	0,084
34	0,653	0,556	0,097
35	0,646	0,538	0,108
36	0,636	0,517	0,119
37	0,623	0,495	0,128
38	0,607	0,472	0,135
39	0,588	0,446	0,142
40	0,565	0,418	0,147
41	0,541	0,39	0,151
42	0,513	0,359	0,154
43	0,483	0,328	0,155
44	0,451	0,295	0,156
45	0,417	0,261	0,156
46	0,381	0,226	0,155
47	0,343	0,191	0,152
48	0,303	0,155	0,148
49	0,262	0,117	0,145
50	0,22	0,08	0,14
51	0,177	0,043	0,134
52	0,133	0,005	0,128
53	0,089	-0,032	0,121

54	0,044	-0,07	0,114
55	0	-0,107	0,107
56	-0,044	-0,144	0,1
57	-0,083	-0,181	0,098
58	-0,124	-0,217	0,093
59	-0,165	-0,252	0,087
60	-0,203	-0,286	0,083
61	-0,24	-0,319	0,079
62	-0,275	-0,351	0,076
63	-0,309	-0,382	0,073
64	-0,341	-0,411	0,07
65	-0,372	-0,439	0,067
66	-0,402	-0,465	0,063
67	-0,43	-0,489	0,059
68	-0,456	-0,512	0,056
69	-0,481	-0,532	0,051
70	-0,504	-0,551	0,047
71	-0,525	-0,568	0,043
72	-0,544	-0,583	0,039
73	-0,561	-0,595	0,034
74	-0,576	-0,606	0,03
75	-0,59	-0,614	0,024
76	-0,6	-0,62	0,02
77	-0,609	-0,623	0,014
78	-0,615	-0,624	0,009
79	-0,619	-0,623	0,004
80	-0,621	-0,62	-0,001
81	-0,621	-0,615	-0,006
82	-0,618	-0,607	-0,011
83	-0,614	-0,597	-0,017
84	-0,606	-0,585	-0,021
85	-0,597	-0,571	-0,026
86	-0,585	-0,554	-0,031
87	-0,571	-0,536	-0,035
88	-0,556	-0,515	-0,041
89	-0,538	-0,493	-0,045
90	-0,518	-0,469	-0,049
91	-0,496	-0,443	-0,053
92	-0,473	-0,415	-0,058
93	-0,448	-0,386	-0,062
94	-0,421	-0,356	-0,065
95	-0,392	-0,324	-0,068
96	-0,362	-0,291	-0,071
97	-0,331	-0,257	-0,074

98	-0,298	-0,223	-0,075
99	-0,265	-0,186	-0,079
100	-0,23	-0,15	-0,08
101	-0,195	-0,113	-0,082
102	-0,159	-0,076	-0,083
103	-0,122	-0,038	-0,084
104	-0,085	0	-0,085
105	-0,048	0,036	-0,084
106	-0,01	0,074	-0,084
107	0,026	0,111	-0,085
108	0,064	0,149	-0,085
109	0,101	0,185	-0,084

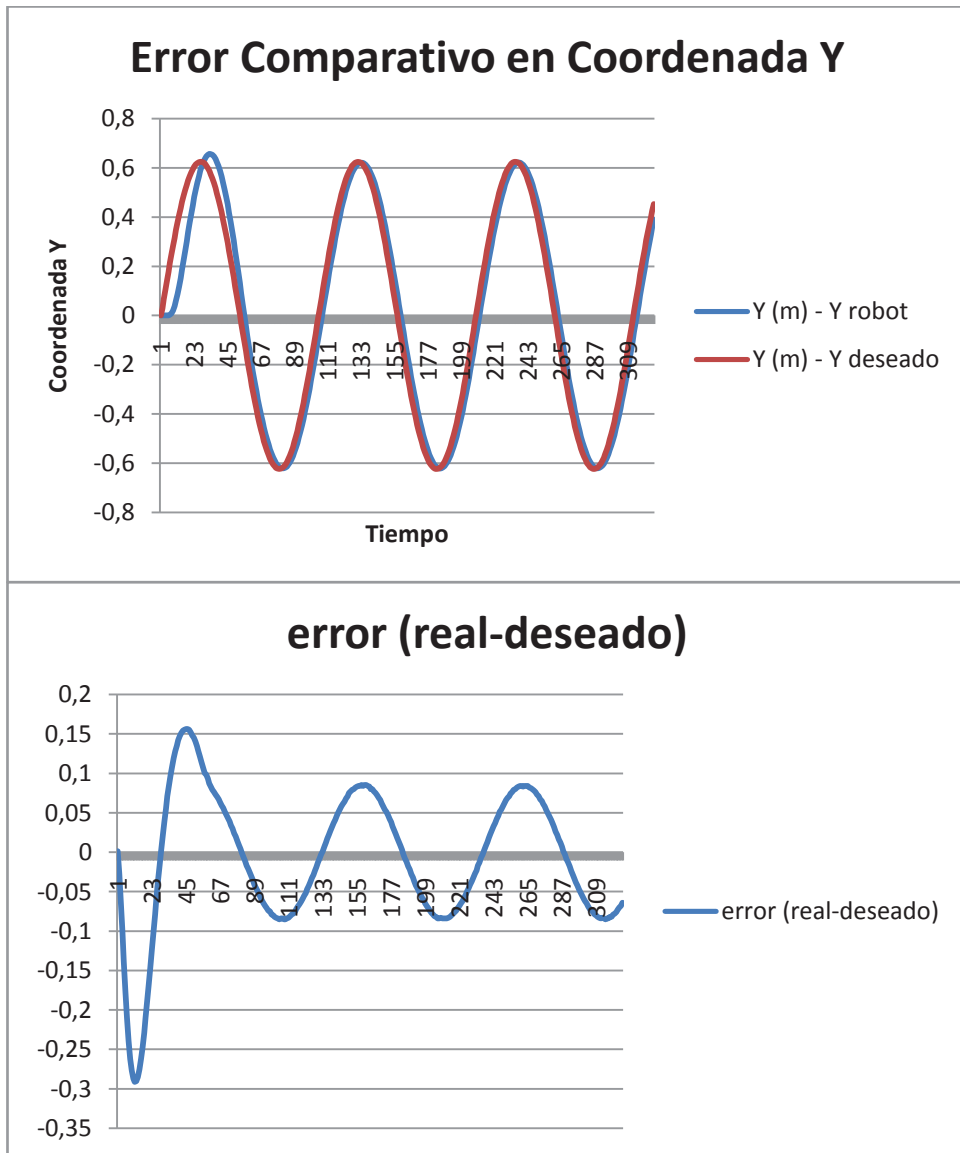


Figura 5.32 Error de posición de la coordenada en Y

## CAPITULO 6

### CONCLUSIONES Y RECOMENDACIONES

#### 6.1 CONCLUSIONES

- El avance de la robótica móvil ha permitido el aprovechamiento de los sistemas robóticos autónomos para tareas repetitivas dentro de procesos industriales, ayudando así a mejorar la eficiencia y productividad; ahí radica la importancia de este tipo de estudios que contribuyen a la generación de procesos y tecnologías en el campo industrial del país que permitan progresivamente el desarrollo de la industria y al mejoramiento económico de la sociedad.
- Si bien la planificación y generación de trayectorias es uno de los problemas que más concierne al estudio de la robótica móvil, el problema del seguimiento de las mismas es la culminación de estos estudios, donde se materializa la acción propuesta. En el presente proyecto de titulación se utiliza la interpolación lineal como base para la solución al problema del seguimiento. Esta herramienta permite una resolución relativamente sencilla al sistema no lineal multivariable que representa un robot en configuración unicyclo con restricciones no holonómicas, realizando una interpolación de las variables de control entre dos tiempos de muestreo y que en base de los resultados tanto de simulaciones como reales, permiten observar que tiene un desempeño bastante elevado, con un error de seguimiento muy pequeño.
- El uso de tecnologías como los FPGA, que se encuentran en pleno desarrollo dentro del país, permite realizar un contraste entre tecnologías tradicionales como los microcontroladores y proveen alternativas para dar soluciones tecnológicas a los distintos retos que presenta la realidad productiva del país.

- Una de las principales ventajas de la utilización de FPGA's es la programación en paralelo que permite realizar tareas de forma más eficiente y con mayor velocidad de procesamiento de datos, a pesar de ello en los sistemas de desarrollo educativos como la utilizada en el presente proyecto, Spartan 3E Starter Kit, limitan un poco la funcionalidad del FPGA cuando por ejemplo se requieren mayores capacidades de memoria para el desarrollo de programas. Este fue uno de los principales problemas que se tuvo que enfrentar en el presente proyecto, por ello es muy importante dimensionar correctamente el dispositivo o la plataforma adecuada para los proyectos específicos a los que se quiera dar solución.
- LabVIEW ofrece una plataforma de programación muchísimo más sencilla en comparación con los lenguajes VHDL, especialmente para quienes no se encuentran familiarizados con ese tipo de lenguajes de programación, ya que toda la programación es gráfica; sin embargo esto implica que al ser un lenguaje de alto nivel no se tenga un control estricto en las configuraciones básicas y manejo de tiempos, esto también ha representado un reto al tener que aprender a manejar adecuadamente los largos de palabra y las cantidades de bits de datos en las variables y operadores utilizadas en la programación para evitar errores de tiempos en la compilación del programa en el dispositivo.
- En referencia a los encoders que representan el sistema sensorial de relación con el entorno en el presente proyecto, es importante tener en consideración las interferencias que pueden presentar los motores y el entorno mismo en su desempeño, ya que al ser la odometría el método de cálculo de las posiciones del robot, errores en la adquisición de datos afectarán el comportamiento final del robot. Para ello es importante verificar como se tiene los datos directos del encoder y utilizar en caso de ser necesario circuitos adicionales que permitan el acondicionamiento de la señal para una correcta interpretación de los datos obtenidos.

- Si bien el Spartan 3E Starter Kit cuenta con periféricos que permiten la comunicación serial, Ethernet, conexión con dispositivos PS2, pulsadores, switches y demás para poder utilizarlos, es necesario programar paso a paso cada una de las características del protocolo correspondiente a cada periférico, es así que para poder realizar una correcta transmisión serial de los datos se tuvo que generar un programa en el cual se cumpla con el estándar de comunicación RS-232, es decir, programar el tiempo que dura un bit en el puerto para que pueda ser leído por el correspondiente equipo receptor, números de bits de datos, de parada e inicio, etc; esto implica que si bien se tiene los periféricos listos para ser usados es importante conocer los estándares para la utilización de los mismos y conseguir de manera exitosa los fines propuestos.
- La utilización de los módulos XBEE S2 brinda una mayor libertad para el desempeño y movilidad del robot al evitar la utilización de medios físicos de conexión para la transmisión y posterior visualización de datos, ofreciendo una comunicación bastante estable a través de bluetooth, que sin embargo como se comentó anteriormente está sujeta a la correcta programación de la transmisión serial.
- Si bien en el presente proyecto sería difícil agregar más utilidades al programa por la capacidad de memoria, existen plataformas similares especialmente de National Instruments con mayores capacidades de memoria y que acoplándolos a un sistema similar al del proyecto se puede generar una retro alimentación desde la PC que en este proyecto sirve únicamente para la visualización del comportamiento del robot que permitiría la interacción del usuario y la manipulación de las variables de control durante la ejecución del programa.
- Como se pudo apreciar en los resultados obtenidos en las gráficas de las simulaciones de la trayectoria circular, el robot se demora alrededor de cuatro vueltas en alcanzar la trayectoria deseada, provocando que el error de seguimiento de la misma sea alto tanto en el desenvolvimiento del

sistema, así como en el error de posición con respecto a las coordenadas  $X$  e  $Y$ , este error se dio como resultado de la mala interpretación al momento de usar la variable  $\theta_{n+1}$ , que no es lo mismo que  $\theta_{dn+1}$ . Como se mencionó en el capítulo de pruebas, la primera corresponde a la dirección que tiene la trayectoria en el instante  $n+1$  y la segunda se refiere a la inclinación que debe poseer el robot en el instante  $n+1$ , si se emplea la primera se obtiene el error que se presentó en la Figura 5.13, además del error de posición propio del valor de  $K_v$  y  $K_w$ ; si se emplea la segunda se obtiene el comportamiento deseado del sistema, que consiste en un seguimiento inmediato del robot con relación a la trayectoria, además con esto se reduce el error en la posición con respecto a las coordenadas  $X$  e  $Y$ .

## 6.2 RECOMENDACIONES

- Se debe poner énfasis en tener niveles de alimentación de voltaje adecuados e independientes en las etapas de control y de potencia del sistema, si se descuida esto se producen errores en la toma de datos de los encoders, en la velocidad de los motores, en la transferencia de datos hacia la computadora vía bluetooth, en el comportamiento de la FPGA e inclusive mal funcionamiento o destrucción de los elementos empleados en el robot, por ende el sistema no funcionará adecuadamente y el objetivo principal que es el seguimiento de trayectorias no se llevará a cabo. Como un dato en el presente trabajo los niveles de voltaje empleados para el funcionamiento de los componentes varían desde 3,3V utilizados por los encoders, 5V para la FPGA y la antena XBEE y 6V para la alimentación de los motores.
- Al momento de realizar el diseño del robot se debe tener muy en cuenta la cinemática del sistema, ya que de ésta dependerá el comportamiento del mismo, así como también el peso de la estructura totalmente armada, ya que el control del sistema depende directamente de la distancia recorrida



por los motores y si la capacidad de los mismos a bajas velocidades no abastece el suficiente torque para poder mover la estructura con facilidad dependiendo de las demandas del control, éste se verá afectado y no se podrá cumplir con las necesidades del sistema para corregir la trayectoria a la cual se está haciendo el seguimiento.

- Cuando se usan motores con cajas reductoras se debe tener en cuenta la relación que poseen los mismos, ya que, al estar los encoders acoplados al eje del motor sus pulsos se ven afectados por dicha relación y por consiguiente el valor del recorrido que realiza cada rueda ( $\Delta S$ ), si no se tiene presente este cambio, al momento de realizar la toma de datos de los pulsos provenientes de los encoders, éstos tendrán errores graves y por consiguiente el control se verá seriamente afectado al punto de que el robot no siga la trayectoria.
- Al momento de utilizar la FPGA Spartan 3E Started Kit se debe tener presente que ésta posee un número máximo de multiplicadores disponibles para ser usados, éstos vienen en un número de 20, al no pasar esta cifra se pueden evitar errores en la compilación, ya que de no ser así la compilación se detiene y no es posible realizar las pruebas deseadas en la FPGA.
- Como uno de los posibles trabajos futuros sobre este módulo se podría implementar un lazo interno de control sobre los motores que son comandados a través del convertidor de Digital a Análogo (DAC) del Spartan 3E Starter Kit, que asegure que la señal enviada hacia los motores sea la señal que éstos reciben y de lo contrario se corrija, el mismo podría ser un control de tipo PID por ejemplo, esto con el fin de asegurar que la orden de control se cumpla en el actuador en este caso el motor y se verifique el cumplimiento de la trayectoria.

## REFERENCIAS BIBLIOGRÁFICAS

- [1] Ollero Aníbal Gonzáles Javier, *Estimación de la posición de un robot móvil*. España, 1996.
- [2] Venator Libris Loci. (2009, Noviembre) Letraherido. [Online]. <http://www.letraherido.com/19040302robots.htm>
- [3] Slin Castro. (2010, Octubre) Blogspot. [Online]. <http://m-team.blogspot.com/>
- [4] Oscar Gonzales. (2008, Marzo) BricoGeek.com. [Online]. <http://blog.bricogeek.com/noticias/diy/video-hexapod-phoenix-el-robot-arana/>
- [5] Patricio Cruz, Cristóbal Villegas, Ing. Nelson Sotomayor, *Diseño y Construcción de un Robot Articulado que Emula el Movimiento de un Gusano*. Quito, Ecuador, 2004.
- [6] Fernando Herrero. MatangaWeb.com. [Online]. <http://www.matangaweb.info/crean-robot-que-se-mueve-como-un-gusano.html>
- [7] Magdalena Andaluz Ortiz Gabriela and Dr Rosales Acosta Andrés, *Modelación, identificación y control de robots móviles*. Quito, Ecuador, 2011.
- [8] Robot House. (2011, December) Blogger. [Online]. <http://myrobothouse.blogspot.com/2011/12/tribot-is-autonomous-robot-from-wowwee.html>
- [9] Super Robotica. (2013, Julio) Super Robótica. [Online]. <http://www.superrobotica.com/S300250.htm>
- [10] Universidad Politécnica Salesiana. (2007) Repositorio Digital UPS. [Online]. <http://dspace.ups.edu.ec/bitstream/123456789/175/2/Capitulo%201.pdf>
- [11] Cuentos Cuanticos. (2011, Diciembre) CuentosCuanticos. [Online]. <http://cuentos-cuanticos.com/2011/12/15/robotica-estimacion-de-posicion-por-odometria/>

- [12 MuchoTrasto. (2000, Septiembre) Mucho Trasto MT. [Online].  
] <http://www.muchotrasto.com/TiposDePlataformas.php>
- [13 CSCazorla. (2011, Abril) Xatakaciencia. [Online].  
] <http://www.xatakaciencia.com/robotica/robots-moviles-i>
- [14 Juan Carlos Espinosa Ceniceros. (2012, Abril) Embedded Computing & Mobile  
] Devices Engineering. [Online]. <http://jcecembmob.blogspot.com/2012/04/embed-comb-lab-robots.html>
- [15 Rovio TM. (2010, Diciembre) Wowwee. [Online].  
] <http://www.wowwee.com/en/support/rovio>
- [16 Daniel Marín. (2012, Julio ) EUREKA. [Online]. [http://astronomia-fisica-misiones-espaciales.blogspot.com/2012\\_07\\_01\\_archive.html](http://astronomia-fisica-misiones-espaciales.blogspot.com/2012_07_01_archive.html)
- [17 Universidad de Bologna. (2000) LAR DEIS. [Online]. <http://www-lar.deis.unibo.it/equipments/p2dx/>
- [18 Wikipedia. (2010, Mayo) Wikipedia.com. [Online].  
] [http://es.wikipedia.org/wiki/Field\\_Programmable\\_Gate\\_Array](http://es.wikipedia.org/wiki/Field_Programmable_Gate_Array)
- [19 National Instruments. (2011, Diciembre) National Instruments NI. [Online].  
] <http://www.ni.com/white-paper/6984/es/>
- [20 Tomas salgado y Martín Velazco Eduardo Aranda, "Control No Lineal  
] Discontinuo de un Robot Móvil," Mexico D.F., 2012.
- [21 Pololu. (2000) Pololu Robotics and Electronics. [Online]. <http://www.pololu.com>  
]
- [22 Jaime Rueda Rueda. (2012, Junio) Blogspot. [Online].  
] <http://jaimeruedayvuelvearodar.blogspot.com/2012/06/sensores.html>
- [23 H.R. Everett, L. Feng J. Borenstein, ""Where am I!" Sensors and Methods for

- ] Mobile Robot Potitioning," Michigan - USA, 1996.
- [24 Robotica al descubierto. (2011, Agosto) Blogspot. [Online].  
] <http://solorobotica.blogspot.com/2011/08/actuadores-en-robotica.html>
- [25 Electrónica Básica. Electrónica Básica. [Online]. <http://www.electronica-basica.com/images/motor-electrico.jpg>
- [26 Direct Industry. sitio Web de SIEMENS. [Online].  
] <http://www.directindustry.es/prod/power-sonic/baterias-ni-cd-19906-438153.html>
- [27 Usuario Tirithel - Wikipedia. (2009, Marzo) Wikipedia. [Online].  
] [http://es.wikipedia.org/wiki/Bater%C3%ADa\\_de\\_n%C3%ADquel\\_e\\_hidruro\\_metal%C3%A1lico](http://es.wikipedia.org/wiki/Bater%C3%ADa_de_n%C3%ADquel_e_hidruro_metal%C3%A1lico)
- [28 Yuribas. (2011, Julio) Blogspot. [Online].  
] [http://yuribass.blogspot.com/2011\\_07\\_01\\_archive.html](http://yuribass.blogspot.com/2011_07_01_archive.html)
- [29 Usuario 77.226.1.14 - Wikipedia. (2011, Enero) Wikipedia. [Online].  
] [http://es.wikipedia.org/wiki/Historia\\_de\\_la\\_pila](http://es.wikipedia.org/wiki/Historia_de_la_pila)
- [30 Guangdong. (2013, Septiembre) Suppliers, China. [Online]. [http://es.made-in-china.com/co\\_yjpower/product\\_3-7V-Cylindrical-Li-ion-Battery-18650-with-2600mAh\\_euiyuyoyg.html](http://es.made-in-china.com/co_yjpower/product_3-7V-Cylindrical-Li-ion-Battery-18650-with-2600mAh_euiyuyoyg.html)
- [31 Direct Industry. (2000, Enero) Virtual EXPO. [Online].  
] <http://www.directindustry.es/prod/ultralife/baterias-li-ion-recargables-31268-172252.html>
- [32 XILINX, *Spartan-3E FPGA Etarted Kit Board User Guide*, XILINX, Ed. Estados Unidos, 2011.
- [33 Atmel. (2013, Febrero) Atmel. [Online]. [http://www.atmel.com/images/atmel-2486-8-bit-avr-microcontroller-atmega8\\_l\\_summary.pdf](http://www.atmel.com/images/atmel-2486-8-bit-avr-microcontroller-atmega8_l_summary.pdf)
- [34 OpenLab. (2011, Mayo) MIDI sense parts. [Online].

- ] <http://www.ladyada.net/make/midisense/makerespairs.html>
- [35 Atmel. (2002, Octubre) Atmel (Atmega 16). [Online].  
] <http://pdf1.alldatasheet.com/datasheet-pdf/view/78532/ATMEL/ATMEGA16.html>
- [36 Jacobo Soto. (2013, Octubre) Open RC. [Online].  
] <http://openrcecuador.com/conceptos-basicos-en-las-baterias-lipo-tensioncapacidad-descarga-y-conexion-de-elementos/>
- [37 National Instruments, "LabVIEW FPGA module User Manual ," Austin Texas,  
] 2004.
- [38 Usuario Hi Te - Wikipedia. (2007, Mayo) Wikipedia la enciclopedia libre.  
] [Online].  
[http://commons.wikimedia.org/wiki/File:Interpolaci%C3%B3n\\_lineal.svg](http://commons.wikimedia.org/wiki/File:Interpolaci%C3%B3n_lineal.svg)
- [39 Gustavo Scaglia, Vicente Mut, Lucia Quintero Andres Rosales, "Tracking  
] Control of a Mobile Robot using Linear Interpolation," *Consejo Nacional de Investigaciones Científicas y Técnicas CONICET Argentina*, p. 1 a 5, 2010.
- [40 Patricio Chico Rodrigo López, "Control del Módulo Pendubot utilizando una  
] FPGA," Quito-Ecuador, 2011.
- [41 Vincent Claes, "Serial Communication," Belgium, 2008.  
]

## ANEXOS


A continuación se presentan las características más importantes de los elementos empleados en el robot provenientes de sus respectivos manuales de usuario (*datasheet*), los programas realizados para captar los pulsos provenientes de los encoders y para generar las salidas pwm que van a los motores.

### ANEXO 1

#### ATMEGA 8

**Features**

- High-performance, Low-power Atmel® AVR® 8-bit Microcontroller
- Advanced RISC Architecture
  - 130 Powerful Instructions – Most Single-clock Cycle Execution
  - 32 × 8 General Purpose Working Registers
  - Fully Static Operation
  - Up to 16MIPS Throughput at 16MHz
  - On-chip 2-cycle Multiplier
- High Endurance Non-volatile Memory segments
  - 8Kbytes of In-System Self-programmable Flash program memory
  - 512Bytes EEPROM
  - 1Kbyte Internal SRAM
  - Write/Erase Cycles: 10,000 Flash/100,000 EEPROM
  - Data retention: 20 years at 85°C/100 years at 25°C<sup>(1)</sup>
  - Optional Boot Code Section with Independent Lock Bits
  - In-System Programming by On-chip Boot Program
  - True Read-While-Write Operation
  - Programming Lock for Software Security
- Peripheral Features
  - Two 8-bit Timer/Counters with Separate Prescaler, one Compare Mode
  - One 16-bit Timer/Counter with Separate Prescaler, Compare Mode, and Capture Mode
  - Real Time Counter with Separate Oscillator
  - Three PWM Channels
  - 8-channel ADC in TQFP and QFN/MLF package
    - Eight Channels 10-bit Accuracy
  - 8-channel ADC in PDIP package
    - Six Channels 10-bit Accuracy
  - Byte-oriented Two-wire Serial Interface
  - Programmable Serial USART
  - Master/Slave SPI Serial Interface
  - Programmable Watchdog Timer with Separate On-chip Oscillator
  - On-chip Analog Comparator
- Special Microcontroller Features
  - Power-on Reset and Programmable Brown-out Detection
  - Internal Calibrated RC Oscillator
  - External and Internal Interrupt Sources
  - Five Sleep Modes: Idle, ADC Noise Reduction, Power-save, Power-down, and Standby
- I/O and Packages
  - 23 Programmable I/O Lines
  - 28-lead PDIP, 32-lead TQFP, and 32-pad QFN/MLF
- Operating Voltages
  - 2.7V - 5.5V (ATmega8L)
  - 4.5V - 5.5V (ATmega8)
- Speed Grades
  - 0 - 8MHz (ATmega8L)
  - 0 - 16MHz (ATmega8)
- Power Consumption at 4Mhz, 3V, 25°C
  - Active: 3.6mA
  - Idle Mode: 1.0mA
  - Power-down Mode: 0.5µA




---


**8-bit Atmel with  
8KBytes In-  
System  
Programmable  
Flash**

---

**ATmega8  
ATmega8L**

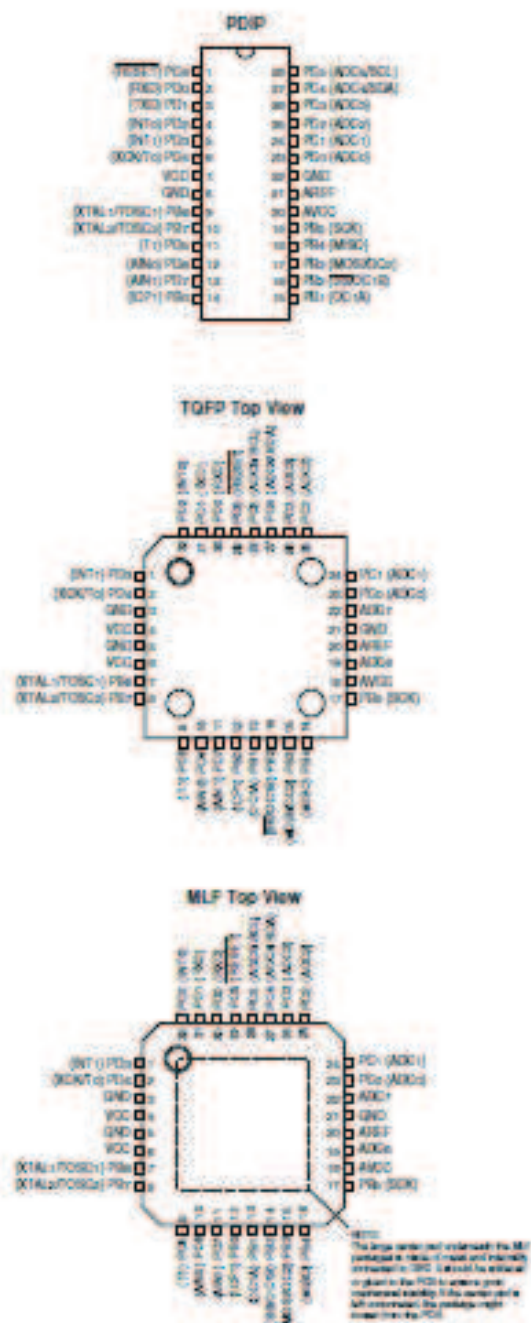
---

16u-2406AA-AVR-020013



## ATmega8(L)

### Pin Configurations



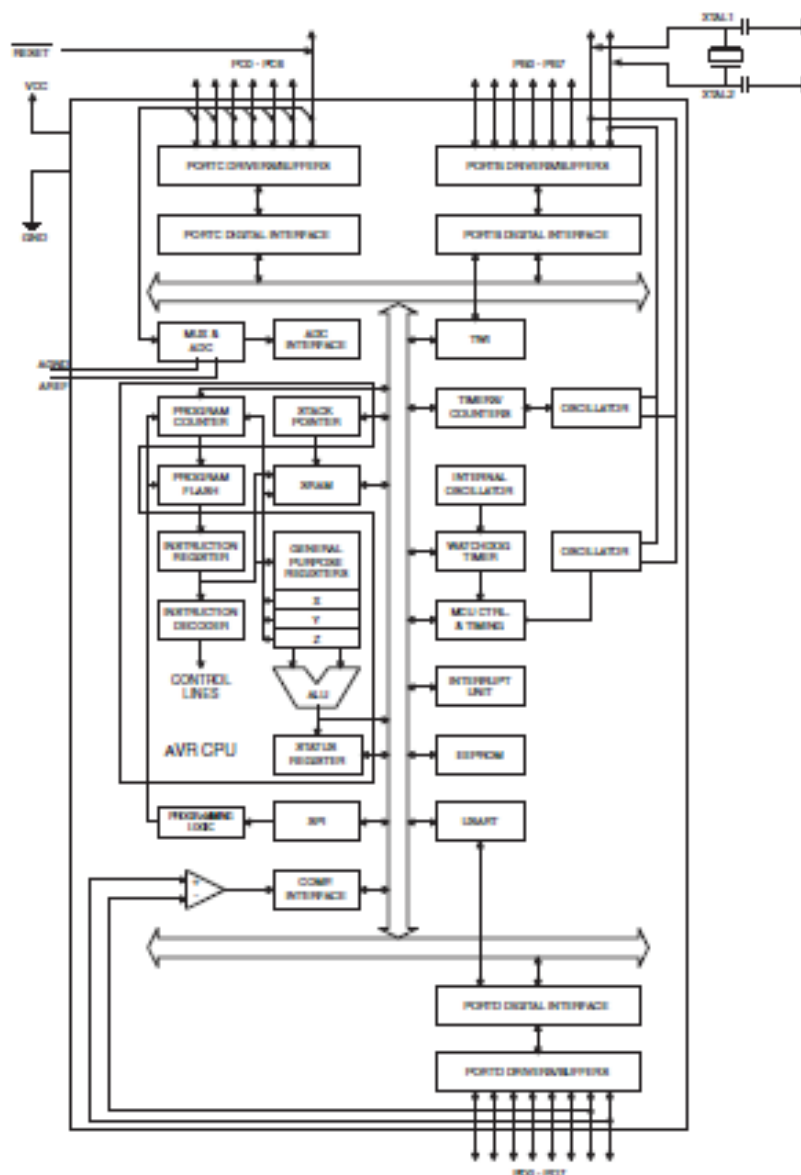
## ATmega8(L)

### Overview

The Atmel®AVR® ATmega8 is a low-power CMOS 8-bit microcontroller based on the AVR RISC architecture. By executing powerful instructions in a single clock cycle, the ATmega8 achieves throughputs approaching 1MIPS per MHz, allowing the system designer to optimize power consumption versus processing speed.

### Block Diagram

Figure 1. Block Diagram





## ANEXO 2

## ATMEGA 16

---

**Features**

- High-performance, Low-power AVR<sup>®</sup> 8-bit Microcontroller
- Advanced RISC Architecture
  - 131 Powerful Instructions – Most Single-clock Cycle Execution
  - 32 x 8 General Purpose Working Registers
  - Fully Static Operation
  - Up to 16 MIPS Throughput at 16 MHz
  - On-chip 2-cycle Multiplier
- Nonvolatile Program and Data Memories
  - 16K Bytes of In-System Self-Programmable Flash
    - Endurance: 10,000 Write/Erase Cycles
  - Optional Boot Code Section with Independent Lock Bits
    - In-System Programming by On-chip Boot Program
    - True Read-While-Write Operation
  - 512 Bytes EEPROM
    - Endurance: 100,000 Write/Erase Cycles
  - 1K Byte Internal SRAM
  - Programming Lock for Software Security
- JTAG (IEEE std. 1149.1 Compliant) Interface
  - Boundary-scan Capabilities According to the JTAG Standard
  - Extensive On-chip Debug Support
  - Programming of Flash, EEPROM, Fuses, and Lock Bits through the JTAG Interface
- Peripheral Features
  - Two 8-bit Timer/Counters with Separate Prescalers and Compare Modes
  - One 16-bit Timer/Counter with Separate Prescaler, Compare Mode, and Capture Mode
  - Real Time Counter with Separate Oscillator
  - Four PWM Channels
  - 8-channel, 10-bit ADC
    - 5 Single-ended Channels
    - 7 Differential Channels in TQFP Package Only
    - 2 Differential Channels with Programmable Gain at 1x, 10x, or 200x
  - Byte-oriented Two-wire Serial Interface
  - Programmable Serial USART
  - Master/Slave SPI Serial Interface
  - Programmable Watchdog Timer with Separate On-chip Oscillator
  - On-chip Analog Comparator
- Special Microcontroller Features
  - Power-on Reset and Programmable Brown-out Detection
  - Internal Calibrated RC Oscillator
  - External and Internal Interrupt Sources
  - Six Sleep Modes: Idle, ADC Noise Reduction, Power-save, Power-down, Standby and Extended Standby
- I/O and Packages
  - 32 Programmable I/O Lines
  - 40-pin PDIP, 44-lead TQFP, and 44-pin MLF
- Operating Voltages
  - 2.7 - 5.5V for ATmega16L
  - 4.5 - 5.5V for ATmega16
- Speed Grades
  - 0 - 8 MHz for ATmega16L
  - 0 - 16 MHz for ATmega16




---

8-bit **AVR<sup>®</sup>**  
 Microcontroller  
 with 16K Bytes  
 In-System  
 Programmable  
 Flash

---

ATmega16  
 ATmega16L

Preliminary

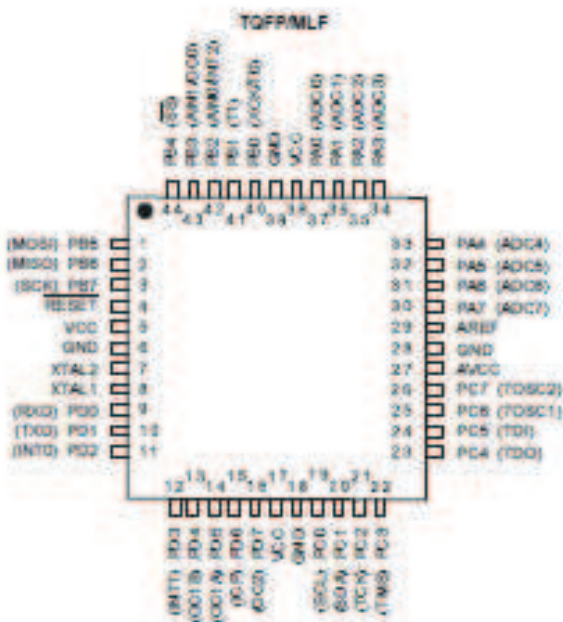
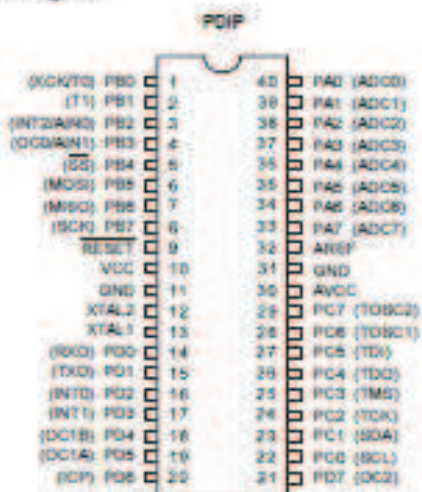
Rev. 2492C-AVR-10/02





**Pin Configurations**

Figure 1. Pinouts ATmega16



**Disclaimer**

Typical values contained in this data sheet are based on simulations and characterization of other AVR microcontrollers manufactured on the same process technology. Min and Max values will be available after the device is characterized.

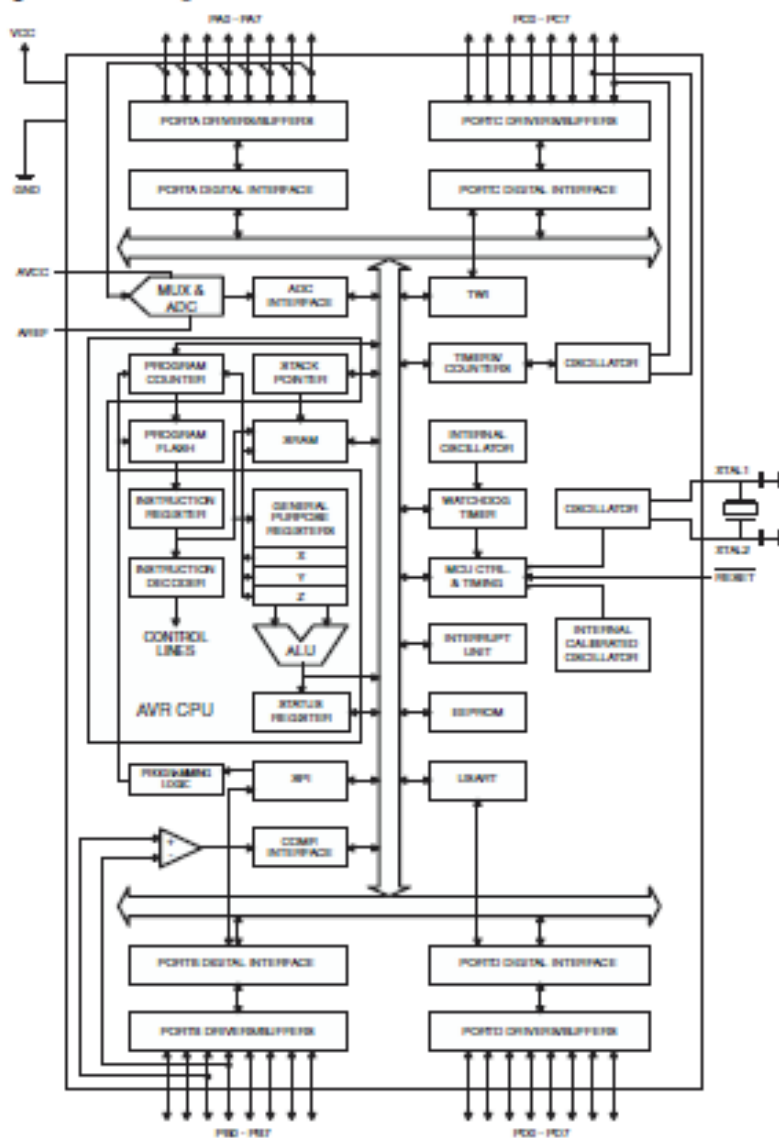
## ATmega16(L)

### Overview

The ATmega16 is a low-power CMOS 8-bit microcontroller based on the AVR enhanced RISC architecture. By executing powerful instructions in a single clock cycle, the ATmega16 achieves throughputs approaching 1 MIPS per MHz allowing the system designer to optimize power consumption versus processing speed.

### Block Diagram

Figure 2. Block Diagram



## ANEXO 3

## MANUAL CD4050BE



Data sheet acquired from Harris Semiconductor  
5010048

## CD4049UB, CD4050B

August 1986 - Revised May 2004

## CMOS Hex Buffer/Converters

The CD4049UB and CD4050B devices are inverting and non-inverting hex buffers, respectively, and feature logic-level conversion using only one supply voltage ( $V_{DD}$ ). The input-signal high level ( $V_{IH}$ ) can exceed the  $V_{DD}$  supply voltage when these devices are used for logic-level conversions. These devices are intended for use as CMOS to DTL/TTL converters and can drive directly two DTL/TTL loads. ( $V_{DD} = 5V$ ,  $V_{OL} < 0.4V$ , and  $I_{OL} > 3.3mA$ .)

The CD4049UB and CD4050B are designated as replacements for CD4009UB and CD4010B, respectively. Because the CD4049UB and CD4050B require only one power supply, they are preferred over the CD4009UB and CD4010B and should be used in place of the CD4009UB and CD4010B in all inverter, current driver, or logic-level conversion applications. In these applications the CD4049UB and CD4050B are pin compatible with the CD4009UB and CD4010B respectively, and can be substituted for these devices in existing as well as in new designs. Terminal No. 16 is not connected internally on the CD4049UB or CD4050B, therefore, connection to this terminal is of no consequence to circuit operation. For applications not requiring high sink-current or voltage conversion, the CD4050B Hex inverter is recommended.

## Features

- CD4049UB Inverting
- CD4050B Non-Inverting
- High Sink Current for Driving 2 TTL Loads
- High-To-Low Level Logic Conversion
- 100% Tested for Quiescent Current at 20V
- Maximum Input Current of 1 $\mu$ A at 18V Over Full Package Temperature Range; 100nA at 18V and 25°C
- 5V, 10V and 15V Parametric Ratings

## Applications

- CMOS to DTL/TTL Hex Converter
- CMOS Current "Sink" or "Source" Driver
- CMOS High-To-Low Logic Level Converter

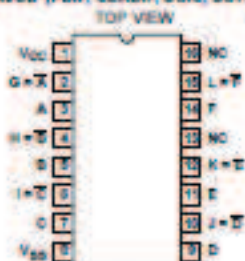
## Ordering Information

PART NUMBER	TEMP. RANGE (°C)	PACKAGE
CD4049UBF3A	-55 to 125	16 Ld CERDIP
CD4050BF3A	-55 to 125	16 Ld CERDIP
CD4049UBD	-55 to 125	16 Ld SOIC
CD4049UBDR	-55 to 125	16 Ld SOIC
CD4049UBDT	-55 to 125	16 Ld SOIC
CD4049UBDW	-55 to 125	16 Ld SOIC
CD4049UBDWR	-55 to 125	16 Ld SOIC
CD4049UBE	-55 to 125	16 Ld PDIP
CD4049UBNSR	-55 to 125	16 Ld SOP
CD4049UBPW	-55 to 125	16 Ld TSSOP
CD4049UBPWR	-55 to 125	16 Ld TSSOP
CD4050B	-55 to 125	16 Ld SOIC
CD4050BDR	-55 to 125	16 Ld SOIC
CD4050BDT	-55 to 125	16 Ld SOIC
CD4050BDW	-55 to 125	16 Ld SOIC
CD4050BDWR	-55 to 125	16 Ld SOIC
CD4050BE	-55 to 125	16 Ld PDIP
CD4050BNSR	-55 to 125	16 Ld SOP
CD4050BPW	-55 to 125	16 Ld TSSOP
CD4050BPWR	-55 to 125	16 Ld TSSOP

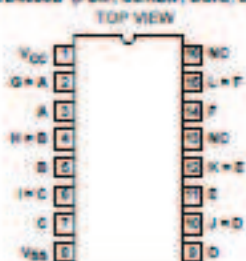
NOTE: When ordering, use the entire part number. The suffix R denotes tape and reel. The suffix T denotes a small-quantity reel of 250.

## Pinouts

CD4049UB (PDIP, CERDIP, SOIC, SOP, TSSOP)



CD4050B (PDIP, CERDIP, SOIC, SOP)

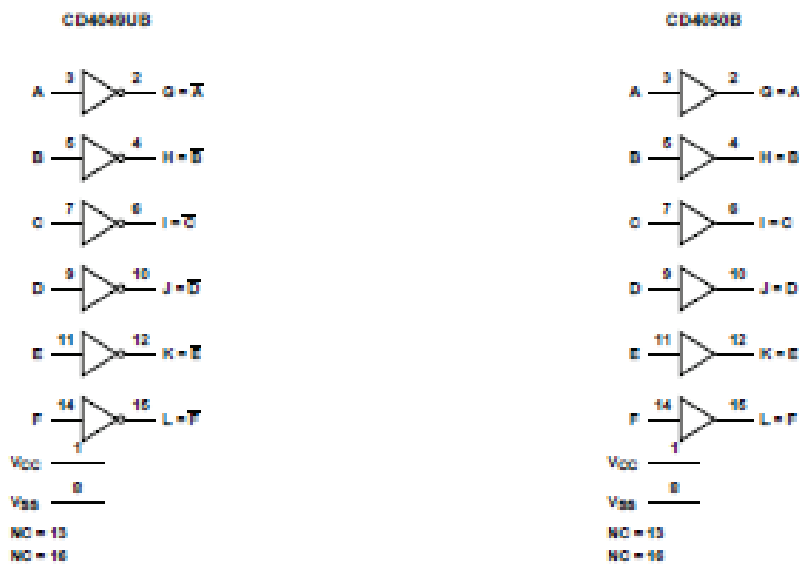


†

CAUTION: These devices are sensitive to electrostatic discharge; follow proper IC Handling Procedures.  
Copyright © 2004, Texas Instruments Incorporated

## CD4049UB, CD4050B

## Functional Block Diagrams



## Schematic Diagrams

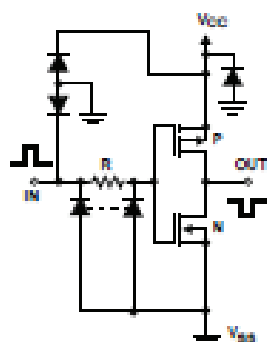


FIGURE 1A. SCHEMATIC DIAGRAM OF CD4049UB, 1 OF 8 IDENTICAL UNITS

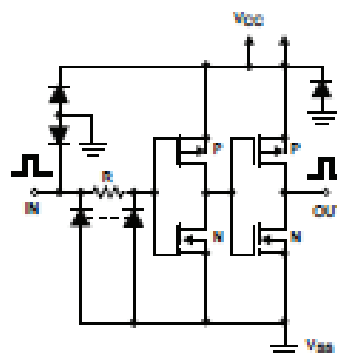


FIGURE 1B. SCHEMATIC DIAGRAM OF CD4050B, 1 OF 8 IDENTICAL UNITS

## CD4049UB, CD4050B

## Absolute Maximum Ratings

Supply Voltage ( $V_V$ to $V_O$ )	-0.5V to 20V
DC Input Current, Any One Input	±10mA

## Operating Conditions

Temperature Range	-55°C to 125°C
-------------------	----------------

## Thermal Information

Package Thermal Impedance, $\theta_{JA}$ (see Note 1):	
E (PDP) Package	87°C/W
D (SOIC) Package	75°C/W
DN (SOIC) Package	57°C/W
NS (SOP) Package	84°C/W
PW (TSSOP) Package	108°C/W
Maximum Junction Temperature (Plastic Package)	150°C
Maximum Storage Temperature Range	-65°C to 150°C
Maximum Lead Temperature (Soldering Iron)	265°C
SOIC - Lead Tip Only	

CAUTION: Stresses above those listed in "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress only rating and operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied.

## NOTE

- The package thermal impedance is calculated in accordance with JEDEC 51-7.

## DC Electrical Specifications

PARAMETER	TEST CONDITIONS			LIMITS AT INDICATED TEMPERATURE (°C)							UNITS
	$V_O$ (V)	$V_{IH}$ (V)	$V_{DC}$ (V)	-55	-40	25	125	25			
								MIN	TYP	MAX	
Quiescent Device Current ( $I_{DD}$ (Max))	-	0.5	5	1	1	30	30	-	0.02	1	$\mu$ A
	-	0.10	10	2	2	60	60	-	0.02	2	$\mu$ A
	-	0.15	15	4	4	120	120	-	0.02	4	$\mu$ A
	-	0.20	20	20	20	600	600	-	0.04	20	$\mu$ A
Output Low (Sink) Current ( $I_{OL}$ (Min))	0.4	0.5	4.5	3.5	3.1	2.1	1.8	2.8	5.2	-	mA
	0.4	0.5	5	4	3.5	2.9	2.4	3.2	6.4	-	mA
	0.5	0.10	10	10	9.5	6.5	5.5	8	18	-	mA
	1.5	0.15	15	28	25	20	18	24	48	-	mA
Output High (Source) Current ( $I_{OH}$ (Min))	4.8	0.5	5	-0.81	-0.73	-0.58	-0.48	-0.58	-1.2	-	mA
	2.5	0.5	5	-2.8	-2.4	-1.9	-1.55	-2.1	-3.9	-	mA
	9.5	0.10	10	-2.0	-1.8	-1.35	-1.18	-1.65	-3.0	-	mA
	13.5	0.15	15	-5.2	-4.8	-3.5	-3.1	-4.3	-8.0	-	mA
Out Voltage Low Level $V_{OL}$ (Max)	-	0.5	5	0.05	0.05	0.05	0.05	-	0	0.05	V
	-	0.10	10	0.05	0.05	0.05	0.05	-	0	0.05	V
	-	0.15	15	0.05	0.05	0.05	0.05	-	0	0.05	V
Output Voltage High Level $V_{OH}$ (Min)	-	0.5	5	4.95	4.95	4.95	4.95	4.95	5	-	V
	-	0.10	10	9.95	9.95	9.95	9.95	9.95	10	-	V
	-	0.15	15	14.95	14.95	14.95	14.95	14.95	15	-	V
Input Low Voltage, $V_{IL}$ (Max) CD4049UB	4.5	-	5	1	1	1	1	-	-	1	V
	9	-	10	2	2	2	2	-	-	2	V
	13.5	-	15	2.5	2.5	2.5	2.5	-	-	2.5	V
Input Low Voltage, $V_{IL}$ (Max) CD4050B	0.5	-	5	1.5	1.5	1.5	1.5	-	-	1.5	V
	1	-	10	3	3	3	3	-	-	3	V
	1.5	-	15	4	4	4	4	-	-	4	V

## CD4049UB, CD4050B

## DC Electrical Specifications (Continued)

PARAMETER	TEST CONDITIONS			LIMITS AT INDICATED TEMPERATURE (°C)							UNITS
	V <sub>O</sub> (V)	V <sub>IN</sub> (V)	V <sub>CC</sub> (V)	-55	-40	25	125	25			
								MIN	TYP	MAX	
Input High Voltage, V <sub>IH</sub> Min CD4049UB	0.5	-	5	4	4	4	4	4	-	-	V
	1	-	10	8	8	8	8	8	-	-	V
	1.5	-	15	12.5	12.5	12.5	12.5	12.5	-	-	V
Input High Voltage, V <sub>IH</sub> Min CD4050B	4.5	-	5	3.5	3.5	3.5	3.5	3.5	-	-	V
	9	-	10	7	7	7	7	7	-	-	V
	13.5	-	15	11	11	11	11	11	-	-	V
Input Current, I <sub>II</sub> Max	-	0.18	15	±0.1	±0.1	±1	±1	-	±10 <sup>-5</sup>	±0.1	µA

AC Electrical Specifications T<sub>A</sub> = 25°C, Input t<sub>r</sub> = 20ns, C<sub>L</sub> = 50pF, R<sub>L</sub> = 200kΩ

PARAMETER	TEST CONDITIONS		LIMITS (ALL PACKAGES)		UNITS
	V <sub>IN</sub>	V <sub>CC</sub>	TYP	MAX	
Propagation Delay Time Low to High, t <sub>PLH</sub> CD4049UB	5	5	60	120	ns
	10	10	32	65	ns
	10	5	45	90	ns
	15	15	25	50	ns
	15	5	45	90	ns
Propagation Delay Time Low to High, t <sub>PLH</sub> CD4050B	5	5	70	140	ns
	10	10	40	80	ns
	10	5	45	90	ns
	15	15	30	60	ns
	15	5	40	80	ns
Propagation Delay Time High to Low, t <sub>PHL</sub> CD4049UB	5	5	32	65	ns
	10	10	20	40	ns
	10	5	15	30	ns
	15	15	15	30	ns
	15	5	10	20	ns
Propagation Delay Time High to Low, t <sub>PHL</sub> CD4050B	5	5	55	110	ns
	10	10	22	45	ns
	10	5	50	100	ns
	15	15	15	30	ns
	15	5	50	100	ns
Transition Time, Low to High, t <sub>TLH</sub>	5	5	60	120	ns
	10	10	40	80	ns
	15	15	30	60	ns
Transition Time, High to Low, t <sub>THL</sub>	5	5	30	60	ns
	10	10	20	40	ns
	15	15	15	30	ns

## ANEXO 4

## MANUAL L298



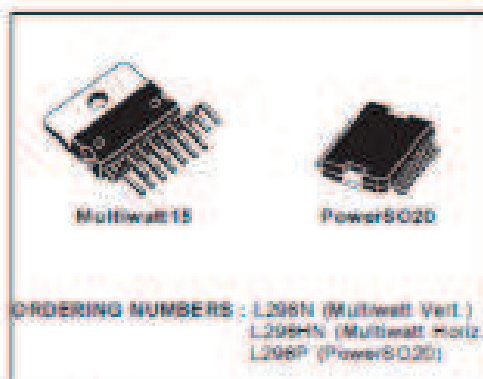
L298

## DUAL FULL-BRIDGE DRIVER

- OPERATING SUPPLY VOLTAGE UP TO 46 V
- TOTAL DC CURRENT UP TO 4 A
- LOW SATURATION VOLTAGE
- OVERTEMPERATURE PROTECTION
- LOGICAL '0' INPUT VOLTAGE UP TO 1.5 V (HIGH NOISE IMMUNITY)

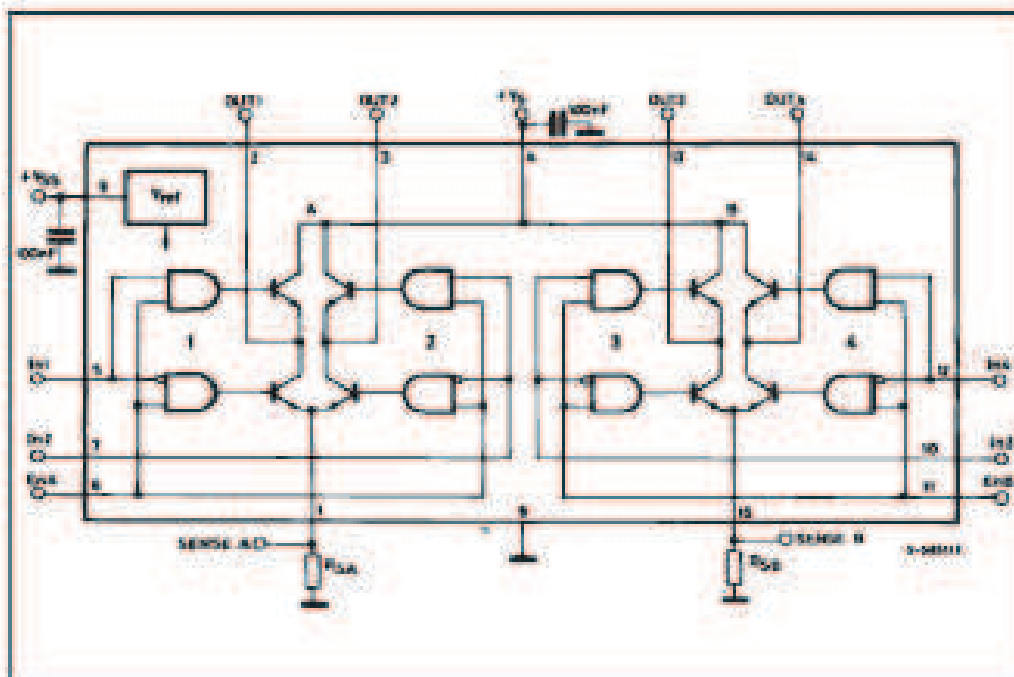
## DESCRIPTION

The L298 is an integrated monolithic circuit in a 15-lead Multiwatt and PowerSO20 packages. It is a high voltage, high current dual full-bridge driver designed to accept standard TTL logic levels and drive inductive loads such as relays, solenoids, DC and stepping motors. Two enable inputs are provided to enable or disable the device independently of the input signals. The emitters of the lower transistors of each bridge are connected together and the corresponding external terminal can be used for the connection of an external sensing resistor. An additional supply input is provided so that the logic works at a lower voltage.



nection of an external sensing resistor. An additional supply input is provided so that the logic works at a lower voltage.

## BLOCK DIAGRAM



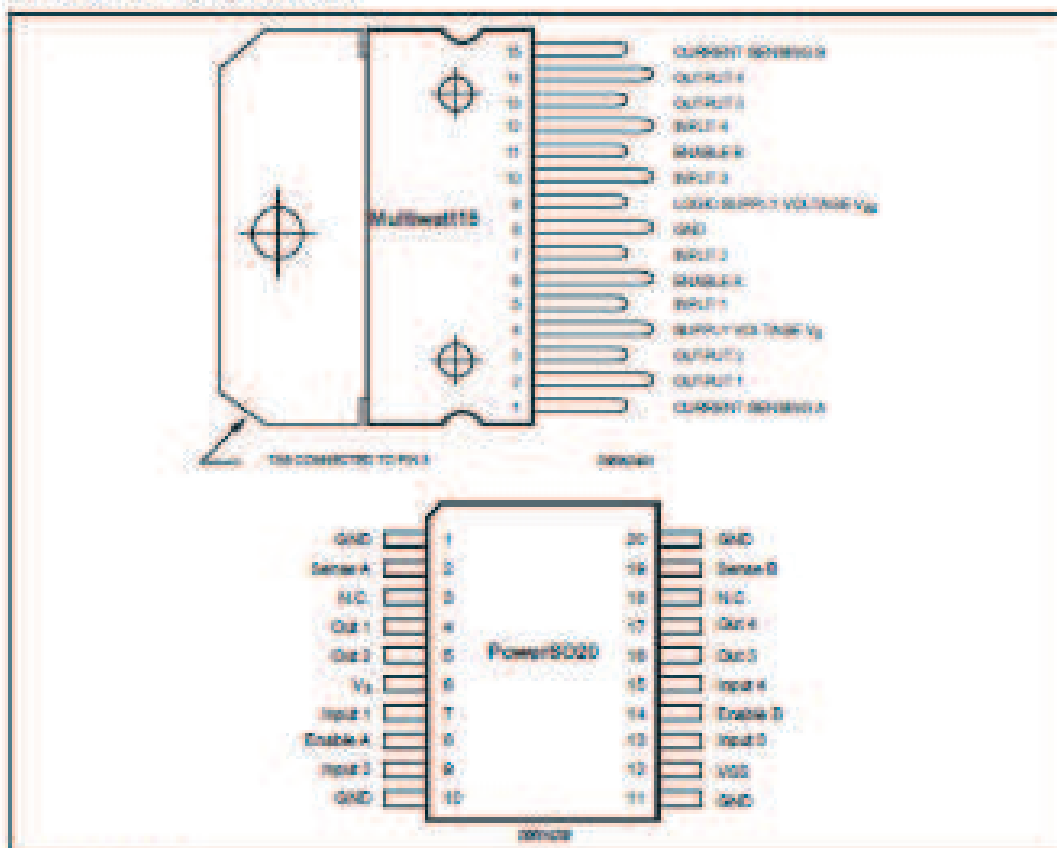


## L298

## ABSOLUTE MAXIMUM RATINGS

Symbol	Parameter	Value	Unit
$V_{CC}$	Power Supply	50	V
$V_{EE}$	Logic Supply Voltage	7	V
$V_{IN}, V_{EN}$	Input and Enable Voltage	-0.5 to 7	V
$I_O$	Peak Output Current (each Channel)	8	A
	- Non Repetitive ( $t = 100\mu s$ )	2.5	A
	- Repetitive (50% on -20% off; $t_{ON} = 10ms$ )	2	A
	-DC Operation	2	A
$V_{SEN}$	Sensing Voltage	-1 to 2.5	V
$P_{tot}$	Total Power Dissipation ( $T_{amb} = 75^{\circ}C$ )	25	W
$T_{op}$	Junction Operating Temperature	-25 to 150	$^{\circ}C$
$T_{stg}, T_j$	Storage and Junction Temperature	-40 to 150	$^{\circ}C$

## PIN CONNECTIONS (top view)



## THERMAL DATA

Symbol	Parameter		PowerSO20	Multiwatt18	Unit
$R_{\theta(jc)}$	Thermal Resistance Junction-case	Max.	-	3	$^{\circ}C/W$
$R_{\theta(ja)}$	Thermal Resistance Junction-ambient	Max.	13 (*)	35	$^{\circ}C/W$

(\*) Mounted on aluminum substrate

## PIN FUNCTIONS (refer to the block diagram)

MW, 15	PowerSO	Name	Function
1,15	2,10	Sense A; Sense B	Between this pin and ground is connected the sense resistor to control the current of the load.
2,3	4,5	Out 1, Out 2	Outputs of the Bridge A. The current that flows through the load connected between these two pins is monitored at pin 1.
4	6	V <sub>S</sub>	Supply Voltage for the Power Output Stages. A non-inductive 100nF capacitor must be connected between this pin and ground.
5,7	7,9	Input 1, Input 2	TTL Compatible Inputs of the Bridge A.
6,11	8,14	Enable A; Enable B	TTL Compatible Enable Input. The L state disables the bridge A (enable A) and/or the bridge B (enable B).
8	1,10,11,20	GND	Ground.
9	12	V <sub>SS</sub>	Supply Voltage for the Logic Blocks. A100nF capacitor must be connected between this pin and ground.
10,12	13,15	Input 3, Input 4	TTL Compatible Inputs of the Bridge B.
13,14	16,17	Out 3, Out 4	Outputs of the Bridge B. The current that flows through the load connected between these two pins is monitored at pin 15.
-	3,18	N.C.	Not Connected

ELECTRICAL CHARACTERISTICS (V<sub>S</sub> = 4.2V; V<sub>SS</sub> = 5V; T<sub>J</sub> = 25°C; unless otherwise specified)

Symbol	Parameter	Test Conditions	Min.	Typ.	Max.	Unit
V <sub>S</sub>	Supply Voltage (pin 4)	Operative Condition	V <sub>S</sub> + 2.5		49	V
V <sub>SS</sub>	Logic Supply Voltage (pin 9)		4.5	5	7	V
I <sub>S</sub>	Quiescent Supply Current (pin 4)	V <sub>IN</sub> = H, I <sub>L</sub> = 0		19	22	mA
		V <sub>IN</sub> = L		50	70	mA
		V <sub>IN</sub> = X			4	mA
I <sub>SS</sub>	Quiescent Current from V <sub>SS</sub> (pin 9)	V <sub>IN</sub> = H, I <sub>L</sub> = 0		24	39	mA
		V <sub>IN</sub> = L		7	12	mA
		V <sub>IN</sub> = X			6	mA
V <sub>L</sub>	Input Low Voltage (pins 5, 7, 10, 12)		-0.3		1.5	V
V <sub>H</sub>	Input High Voltage (pins 5, 7, 10, 12)		2.5		V <sub>SS</sub>	V
I <sub>L</sub>	Low Voltage Input Current (pins 5, 7, 10, 12)	V <sub>IN</sub> = L			-10	μA
I <sub>H</sub>	High Voltage Input Current (pins 5, 7, 10, 12)	V <sub>IN</sub> = H; V <sub>SS</sub> = 0.6V		30	100	μA
V <sub>EN</sub> = L	Enable Low Voltage (pins 6, 11)		-0.3		1.5	V
V <sub>EN</sub> = H	Enable High Voltage (pins 6, 11)		2.5		V <sub>SS</sub>	V
I <sub>EN</sub> = L	Low Voltage Enable Current (pins 6, 11)	V <sub>EN</sub> = L			-10	μA
I <sub>EN</sub> = H	High Voltage Enable Current (pins 6, 11)	V <sub>EN</sub> = H; V <sub>SS</sub> = 0.6V		30	100	μA
V <sub>OS(sat)</sub>	Source Saturation Voltage	I <sub>L</sub> = 1A I <sub>S</sub> = 2A	0.95	1.55	1.7	V
				2	2.7	V
V <sub>OS(sink)</sub>	Sink Saturation Voltage	I <sub>L</sub> = 1A (S) I <sub>S</sub> = 2A (S)	0.85	1.2	1.6	V
				1.7	2.5	V
V <sub>OD(sat)</sub>	Total Drop	I <sub>L</sub> = 1A (S) I <sub>S</sub> = 2A (S)	1.60		3.2	V
					4.9	V
V <sub>SEN</sub>	Sensing Voltage (pins 1, 15)		-1 (1)		2	V

## L298

## ELECTRICAL CHARACTERISTICS (continued)

Symbol	Parameter	Test Conditions	Min.	Typ.	Max.	Unit
$T_1$ (N)	Source Current Turn-off Delay	0.5 V <sub>I</sub> to 0.9 I <sub>L</sub> (2); (4)		1.5		μs
$T_2$ (N)	Source Current Fall Time	0.9 I <sub>L</sub> to 0.1 I <sub>L</sub> (2); (4)		0.2		μs
$T_3$ (N)	Source Current Turn-on Delay	0.5 V <sub>I</sub> to 0.1 I <sub>L</sub> (2); (4)		2		μs
$T_4$ (N)	Source Current Rise Time	0.1 I <sub>L</sub> to 0.9 I <sub>L</sub> (2); (4)		0.7		μs
$T_5$ (N)	Sink Current Turn-off Delay	0.5 V <sub>I</sub> to 0.9 I <sub>L</sub> (3); (4)		0.7		μs
$T_6$ (N)	Sink Current Fall Time	0.9 I <sub>L</sub> to 0.1 I <sub>L</sub> (3); (4)		0.25		μs
$T_7$ (N)	Sink Current Turn-on Delay	0.5 V <sub>I</sub> to 0.9 I <sub>L</sub> (3); (4)		1.5		μs
$T_8$ (N)	Sink Current Rise Time	0.1 I <sub>L</sub> to 0.9 I <sub>L</sub> (3); (4)		0.2		μs
$f_c$ (N)	Commutation Frequency	I <sub>L</sub> = 2A		25	40	kHz
$T_1$ (N <sub>sat</sub> )	Source Current Turn-off Delay	0.5 V <sub>sat</sub> to 0.9 I <sub>L</sub> (2); (4)		5		μs
$T_2$ (N <sub>sat</sub> )	Source Current Fall Time	0.9 I <sub>L</sub> to 0.1 I <sub>L</sub> (2); (4)		1		μs
$T_3$ (N <sub>sat</sub> )	Source Current Turn-on Delay	0.5 V <sub>sat</sub> to 0.1 I <sub>L</sub> (2); (4)		0.9		μs
$T_4$ (N <sub>sat</sub> )	Source Current Rise Time	0.1 I <sub>L</sub> to 0.9 I <sub>L</sub> (2); (4)		0.4		μs
$T_5$ (N <sub>sat</sub> )	Sink Current Turn-off Delay	0.5 V <sub>sat</sub> to 0.9 I <sub>L</sub> (3); (4)		2.2		μs
$T_6$ (N <sub>sat</sub> )	Sink Current Fall Time	0.9 I <sub>L</sub> to 0.1 I <sub>L</sub> (3); (4)		0.55		μs
$T_7$ (N <sub>sat</sub> )	Sink Current Turn-on Delay	0.5 V <sub>sat</sub> to 0.9 I <sub>L</sub> (3); (4)		0.25		μs
$T_8$ (N <sub>sat</sub> )	Sink Current Rise Time	0.1 I <sub>L</sub> to 0.9 I <sub>L</sub> (3); (4)		0.1		μs

1) 1) Gate voltage can be -1 V for t<sub>off</sub> < 50 μsec; in steady state V<sub>gate</sub> min > -0.5 V

2) See fig. 2

3) See fig. 4

4) The load must be a pure resistor.

Figure 1 : Typical Saturation Voltage vs. Output Current.

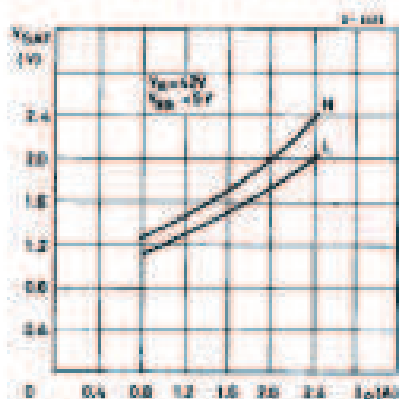
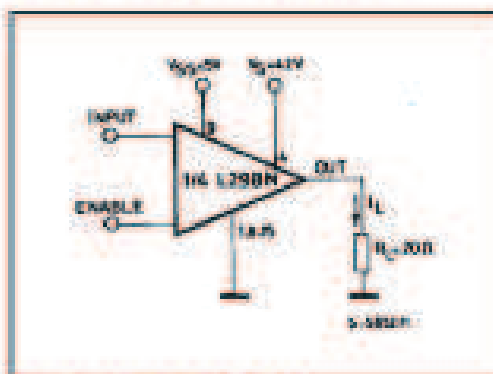


Figure 2 : Switching Times Test Circuits.



Note: For INPUT Switching see (N = I)  
For ENABLE Switching see (N = II)

## ANEXO 5

## MANUAL LM324

Philips Semiconductors	Product specification
<b>Low power quad op amps</b>	<b>LM124/224/324/324A/ SA534/LM2902</b>

## DESCRIPTION

The LM124/SA534/LM2902 series consists of four independent, high-gain, internally frequency-compensated operational amplifiers designed specifically to operate from a single power supply over a wide range of voltages.

## UNIQUE FEATURES

In the linear mode, the input common-mode voltage range includes ground and the output voltage can also swing to ground, even though operated from only a single power supply voltage.

The unity gain crossover frequency and the input bias current are temperature-compensated.

## FEATURES

- Internally frequency-compensated for unity gain
- Large DC voltage gain: 100dB
- Wide bandwidth (unity gain): 1MHz (temperature-compensated)
- Wide power supply range: Single supply: 3V<sub>DC</sub> to 30V<sub>DC</sub> or dual supplies: ±1.5V<sub>DC</sub> to ±15V<sub>DC</sub>
- Very low supply current drain: essentially independent of supply voltage (1mW/op amp at +5V<sub>DC</sub>)
- Low input biasing current: 45nA<sub>DC</sub> (temperature-compensated)
- Low input offset voltage: 2mV<sub>DC</sub> and offset current: 5nA<sub>DC</sub>
- Differential input voltage range equal to the power supply voltage
- Large output voltage: 0V<sub>DC</sub> to V<sub>DC</sub>-1.5V<sub>DC</sub> swing

## PIN CONFIGURATION

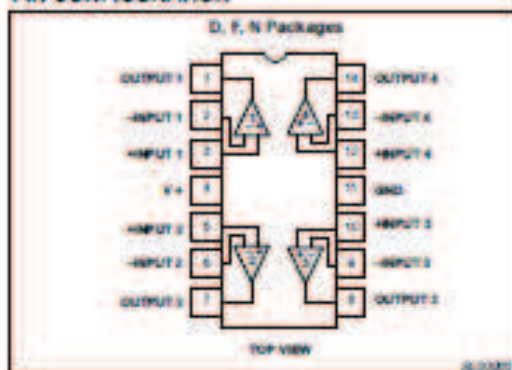


Figure 1. Pin Configuration

## ORDERING INFORMATION

DESCRIPTION	TEMPERATURE RANGE	ORDER CODE	DWG #
14-Pin Plastic Dual In-Line Package (DIP)	-55°C to +125°C	LM124N	SOT127-1
14-Pin Ceramic Dual In-Line Package (CERDIP)	-55°C to +125°C	LM124F	D561B
14-Pin Plastic Dual In-Line Package (DIP)	-25°C to +85°C	LM224N	SOT127-1
14-Pin Ceramic Dual In-Line Package (CERDIP)	-25°C to +85°C	LM224F	D561B
14-Pin Plastic Small Outline (SO) Package	-25°C to +85°C	LM224D	SOT108-1
14-Pin Plastic Dual In-Line Package (DIP)	0°C to +75°C	LM324N	SOT127-1
14-Pin Ceramic Dual In-Line Package (CERDIP)	0°C to +75°C	LM324F	D561B
14-Pin Plastic Small Outline (SO) Package	0°C to +75°C	LM324D	SOT108-1
14-Pin Plastic Dual In-Line Package (DIP)	0°C to +75°C	LM324AN	SOT127-1
14-Pin Plastic Small Outline (SO) Package	0°C to +75°C	LM324AD	SOT108-1
14-Pin Plastic Dual In-Line Package (DIP)	-40°C to +85°C	SA534N	SOT127-1
14-Pin Ceramic Dual In-Line Package (CERDIP)	-40°C to +85°C	SA534F	D561B
14-Pin Plastic Small Outline (SO) Package	-40°C to +85°C	SA534D	SOT108-1
14-Pin Plastic Small Outline (SO) Package	-40°C to +125°C	LM2902D	SOT108-1
14-Pin Plastic Dual In-Line Package (DIP)	-40°C to +125°C	LM2902N	SOT127-1

## Low power quad op amps

LM124/224/324/324A/  
SA534/LM2902

## ABSOLUTE MAXIMUM RATINGS

SYMBOL	PARAMETER	RATING	UNIT
$V_{CC}$	Supply voltage	32 or ±16	V <sub>DC</sub>
$V_{IN}$	Differential input voltage	32	V <sub>DC</sub>
$V_{IN}$	Input voltage	-0.3 to +32	V <sub>DC</sub>
$P_D$	Maximum power dissipation, $T_A=25^\circ\text{C}$ (M0-air) <sup>1</sup>		
	N package	1420	mW
	F package	1190	mW
	D package	1040	mW
	Output short-circuit to GND one amplifier <sup>2</sup> $V_{CC}=15V_{DC}$ and $T_A=25^\circ\text{C}$	Continuous	
$I_{IN}$	Input current ( $V_{IN}=0.3V$ ) <sup>3</sup>	50	mA
$T_A$	Operating ambient temperature range		
	LM304A	0 to +70	°C
	LM224	-25 to +85	°C
	SA534	-40 to +85	°C
	LM2902	-40 to +125	°C
LM124	-55 to +125	°C	
$T_{STG}$	Storage temperature range	-65 to +150	°C
$T_{SOL}$	Lead soldering temperature (10sec max)	300	°C

## NOTES:

- Derate above 25°C at the following rates:  
F package at 0.5mW/°C  
N package at 11.4mW/°C  
D package at 8.3mW/°C
- Short-circuits from the output to  $V_{CC}$  can cause excessive heating and eventual destruction. The maximum output current is approximately 40mA, independent of the magnitude of  $V_{CC}$ . At values of supply voltage in excess of +15V<sub>DC</sub>, continuous short-circuits can exceed the power dissipation ratings and cause eventual destruction.
- This input current will only exist when the voltage at any of the input leads is driven negative. It is due to the collector-base junction of the input PNP transistors becoming forward biased and thereby acting as input bias diodes. In addition, there is also lateral NPN parasitic transistor action on the IC chip. This action can cause the output voltages of the op amps to go to the  $V^+$  rail (or to ground for a large overdrive) during the time that the input is driven negative.

## Low power quad op amps

LM124/224/324/324A/  
SA534/LM2902

## DC ELECTRICAL CHARACTERISTICS

V<sub>CC</sub>=5V, T<sub>a</sub>=25°C unless otherwise specified

SYMBOL	PARAMETER	TEST CONDITIONS	LM124/LM224			LM324/SA534/LM2902			UNIT
			Min	Typ	Max	Min	Typ	Max	
V <sub>OS</sub>	Offset voltage <sup>1</sup>	R <sub>F</sub> =0Ω		±2	±5		±2	±7	mV
		R <sub>F</sub> =0Ω, over temp.			±7			±8	
ΔV <sub>OS</sub> /ΔT	Temperature drift	R <sub>F</sub> =0Ω, over temp.		7			7	μV/°C	
I <sub>BIAS</sub>	Input current <sup>2</sup>	(I <sub>B1</sub> ) or (I <sub>B2</sub> )		45	150		45	250	nA
		(I <sub>B1</sub> ) or (I <sub>B2</sub> ), over temp.		40	300		40	500	
ΔI <sub>BIAS</sub> /ΔT	Temperature drift	Over temp.		50			50	pA/°C	
I <sub>OS</sub>	Offset current	(I <sub>B1</sub> -I <sub>B2</sub> )		±5	±50		±5	±50	nA
		(I <sub>B1</sub> -I <sub>B2</sub> ), over temp.			±100			±150	
ΔI <sub>OS</sub> /ΔT	Temperature drift	Over temp.		10			10	pA/°C	
V <sub>CM</sub>	Common-mode voltage range <sup>3</sup>	V <sub>CC</sub> =30V	0		V <sub>CC</sub> -1.5	0		V <sub>CC</sub> -1.5	V
		V <sub>CC</sub> =30V, over temp.	0		V <sub>CC</sub> -2	0		V <sub>CC</sub> -2	
CMRR	Common-mode rejection ratio	V <sub>CC</sub> =30V	70	85		65	70	dB	
V <sub>OUT</sub>	Output voltage swing	R <sub>L</sub> =2kΩ, V <sub>CC</sub> =30V, over temp.	26			26		V	
V <sub>OH</sub>	Output voltage high	R <sub>L</sub> =10kΩ, V <sub>CC</sub> =30V, over temp.	27	28		27	28	V	
V <sub>OL</sub>	Output voltage low	R <sub>L</sub> =10kΩ, over temp.		5	20		5	20	mV
I <sub>CC</sub>	Supply current	R <sub>L</sub> =∞, V <sub>CC</sub> =30V, over temp.		1.5	3		1.5	3	mA
		R <sub>L</sub> =∞, over temp.		0.7	1.2		0.7	1.2	
A <sub>VOL</sub>	Large-signal voltage gain	V <sub>CC</sub> =15V (for large V <sub>O</sub> swing), R <sub>L</sub> =2kΩ	50	100		25	100	V/mV	
		V <sub>CC</sub> =15V (for large V <sub>O</sub> swing), R <sub>L</sub> =2kΩ, over temp.	25			15			
	Amplifier-to-amplifier coupling <sup>4</sup>	5-10kHz to 20kHz, input referred		-120			+120	dB	
PSRR	Power supply rejection ratio	R <sub>F</sub> =0Ω	65	100		65	100	dB	
I <sub>OUT</sub>	Output current source	V <sub>IN1</sub> =+1V, V <sub>IN2</sub> =0V, V <sub>CC</sub> =15V	20	40		20	40	mA	
		V <sub>IN1</sub> =+1V, V <sub>IN2</sub> =0V, V <sub>CC</sub> =15V, over temp.	10	20		10	20		
	Output current sink	V <sub>IN1</sub> =1V, V <sub>IN2</sub> =0V, V <sub>CC</sub> =15V	10	20		10	20		
		V <sub>IN1</sub> =1V, V <sub>IN2</sub> =0V, V <sub>CC</sub> =15V, over temp.	5	8		5	8		
		V <sub>IN1</sub> =1V, V <sub>IN2</sub> =0V, V <sub>OL</sub> =200mV	12	50		12	50	μA	
I <sub>SC</sub>	Short-circuit current <sup>5</sup>		10	40	60	10	40	60	mA
GBW	Unity gain bandwidth			1			1	MHz	
SR	Slew rate			0.3			0.3	V/μs	
V <sub>IN(rms)</sub>	Input noise voltage	f=1kHz		40			40	nV/√Hz	
V <sub>DIFF</sub>	Differential input voltage <sup>6</sup>				V <sub>CC</sub>		V <sub>CC</sub>	V	

## Low power quad op amps

LM124/224/324/324A/  
SA534/LM2902

## DC ELECTRICAL CHARACTERISTICS (Continued)

 $V_{CC}=5V$ ,  $T_A=25^{\circ}C$  unless otherwise specified.

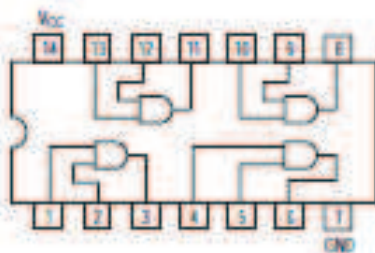
SYMBOL	PARAMETER	TEST CONDITIONS	LM124A			UNIT
			Min	Typ	Max	
$V_{OS}$	Offset voltage <sup>1</sup>	$R_G=0\Omega$		±2	±3	mV
		$R_G=0\Omega$ , over temp.			±5	
$\Delta V_{OS}/\Delta T$	Temperature drift	$R_G=0\Omega$ , over temp.		7	30	$\mu V/^{\circ}C$
$I_{BIAS}$	Input current <sup>2</sup>	$I_{IN}(+) \text{ or } I_{IN}(-)$		45	100	nA
		$I_{IN}(+) \text{ or } I_{IN}(-)$ , over temp.		40	200	
$\Delta I_{BIAS}/\Delta T$	Temperature drift	Over temp.		50		$\mu A/^{\circ}C$
$I_{OC}$	Offset current	$I_{IN}(+) + I_{IN}(-)$		±5	±10	nA
		$I_{IN}(+) + I_{IN}(-)$ , over temp.			±15	
$\Delta I_{OC}/\Delta T$	Temperature drift	Over temp.		10	300	$\mu A/^{\circ}C$
$V_{CM}$	Common-mode voltage range <sup>3</sup>	$V_{CC}=30V$	0		$V_{CC}-1.5$	V
		$V_{CC}=30V$ , over temp.	0		$V_{CC}-2$	V
CMRR	Common-mode rejection ratio	$V_{CC}=30V$	85	85		dB
$V_{OUT}$	Output voltage swing	$R_L=2k\Omega$ , $V_{CC}=30V$ , over temp.	26			V
$V_{OH}$	Output voltage high	$R_L < 10k\Omega$ , $V_{CC}=30V$ , over temp.	27	28		V
$V_{OL}$	Output voltage low	$R_L < 10k\Omega$ , over temp.		5	20	mV
$I_{CC}$	Supply current	$R_L=0$ , $V_{CC}=30V$ , over temp.		1.5	2	mA
		$R_L=0$ , over temp.		0.7	1.2	mA
$A_{VOL}$	Large-signal voltage gain	$V_{CC}=15V$ (for large $V_O$ swing), $R_L=2k\Omega$	25	100		V/mV
		$V_{CC}=15V$ (for large $V_O$ swing), $R_L=2k\Omega$ , over temp.	15			V/mV
	Amplifier-to-amplifier coupling <sup>5</sup>	5-10kHz to 20kHz, input-referred		-120		dB
PSRR	Power supply rejection ratio	$R_G=0\Omega$	85	100		dB
$I_{OUP}$	Output current source	$V_{IN}=+1V$ , $V_{IN}=0V$ , $V_{CC}=15V$	20	40		mA
		$V_{IN}=+1V$ , $V_{IN}=0V$ , $V_{CC}=15V$ , over temp.	10	20		mA
	Output current sink	$V_{IN}=+1V$ , $V_{IN}=0V$ , $V_{CC}=15V$	10	20		mA
		$V_{IN}=+1V$ , $V_{IN}=0V$ , $V_{CC}=15V$ , over temp.	5	5		mA
		$V_{IN}=+1V$ , $V_{IN}=0V$ , $V_O=200mV$	12	30		$\mu A$
$I_{SC}$	Short-circuit current <sup>4</sup>		10	40	80	mA
$V_{DIP}$	Differential input voltage <sup>7</sup>			$V_{CC}$		V
GBW	Unity gain bandwidth			1		MHz
SR	Slew rate			0.3		V/ $\mu s$
$V_{IN(RMS)}$	Input noise voltage	5-10kHz		40		nV/Hz

## NOTES:

- $V_O = 1/3 V_{CC}$ ,  $R_G=0\Omega$  with  $V_{CC}$  from 5V to 30V and over full input common-mode range ( $\Delta V_{CC}$  is  $V_{CC} - 1.5V$ ).
- The direction of the input current is out of the IC due to the PNP input stage. This current is essentially constant, independent of the state of the output so no loading change exists on the input lines.
- The input common-mode voltage or either input signal voltage should not be allowed to go negative by more than 0.5V. The upper end of the common-mode voltage range is  $V_{CC} - 1.5$ , but either or both inputs can go to +30V without damage.
- Short-circuits from the output to  $V_{CC}$  can cause excessive heating and eventual destruction. The maximum output current is approximately 40mA independent of the magnitude of  $V_{CC}$ . At values of supply voltage in excess of  $\approx 15V_{CC}$ , continuous short-circuits can exceed the power dissipation ratings and cause eventual destruction. Destructive dissipation can result from simultaneous shorts on all amplifiers.
- Due to proximity of external components, insure that coupling is not originating via stray capacitance between these external parts. This typically can be detected as the type of coupling increases at higher frequencies.

## ANEXO 6

## MANUAL 74LS08

**SN74LS08****Quad 2-Input AND Gate**

## GUARANTEED OPERATING RANGES

Symbol	Parameter	Min	Typ	Max	Unit
$V_{CC}$	Supply Voltage	4.75	5.0	5.25	V
$T_A$	Operating Ambient Temperature Range	0	25	70	$^{\circ}\text{C}$
$I_{OH}$	Output Current – High			-0.4	mA
$I_{OL}$	Output Current – Low			8.0	mA

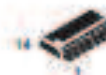


**ON Semiconductor**  
 Formerly a Division of Motorola  
<http://onsemi.com>

**LOW  
POWER  
SCHOTTKY**



PLASTIC  
N SUFFIX  
CASE 646



SOIC  
D SUFFIX  
CASE 751A

## ORDERING INFORMATION

Device	Package	Shipping
SN74LS08N	14 Pin DIP	2000 Units/Box
SN74LS08D	14 Pin	2500/Tape & Reel



## SN74LS08

## DC CHARACTERISTICS OVER OPERATING TEMPERATURE RANGE (unless otherwise specified)

Symbol	Parameter	Limits			Unit	Test Conditions
		Min	Typ	Max		
$V_{IH}$	Input HIGH Voltage	2.0			V	Guaranteed Input HIGH Voltage for All Inputs
$V_{IL}$	Input LOW Voltage			0.8	V	Guaranteed Input LOW Voltage for All Inputs
$V_{IK}$	Input Clamp Diode Voltage		-0.85	-1.5	V	$V_{CC} = \text{MIN}$ , $I_{IK} = -18 \text{ mA}$
$V_{OH}$	Output HIGH Voltage	2.7	3.5		V	$V_{CC} = \text{MIN}$ , $I_{OH} = \text{MAX}$ , $V_{IN} = V_{IH}$ or $V_{IL}$ per Truth Table
$V_{OL}$	Output LOW Voltage		0.25	0.4	V	$I_{OL} = 4.0 \text{ mA}$
			0.35	0.5	V	$I_{OL} = 8.0 \text{ mA}$
$I_{IH}$	Input HIGH Current			20	$\mu\text{A}$	$V_{CC} = \text{MAX}$ , $V_{IN} = 2.7 \text{ V}$
				0.1	mA	$V_{CC} = \text{MAX}$ , $V_{IN} = 7.0 \text{ V}$
$I_{IL}$	Input LOW Current			-0.4	mA	$V_{CC} = \text{MAX}$ , $V_{IN} = 0.4 \text{ V}$
$I_{OS}$	Short Circuit Current (Note 1)	-20		-100	mA	$V_{CC} = \text{MAX}$
$I_{CC}$	Power Supply Current Total, Output HIGH			4.8	mA	$V_{CC} = \text{MAX}$
				8.8		
	Total, Output LOW					

Note 1: Not more than one output should be shorted at a time, nor for more than 1 second.

AC CHARACTERISTICS ( $T_A = 25^\circ\text{C}$ )

Symbol	Parameter	Limits			Unit	Test Conditions
		Min	Typ	Max		
$t_{PLH}$	Turn-Off Delay, Input to Output		8.0	15	ns	$V_{CC} = 5.0 \text{ V}$ $C_L = 15 \text{ pF}$
$t_{PHL}$	Turn-On Delay, Input to Output		10	20	ns	

## ANEXO 7

## MANUAL 7805

## 7805 • THREE-TERMINAL POSITIVE VOLTAGE REGULATOR IC

## FEATURES:

- OUTPUT CURRENT IN EXCESS OF 1A;
- NO EXTERNAL COMPONENTS REQUIRED;
- INTERNAL SHORT CIRCUIT CURRENT LIMITING;
- INTERNAL THERMAL OVERLOAD PROTECTION;
- OUTPUT TRANSISTOR SAFE-AREA COMPENSATION;
- OUTPUT VOLTAGE OFFERED IN 4% TOLERANCE.

ABSOLUTE MAXIMUM RATINGS ( $T_a = 25^\circ\text{C}$ )

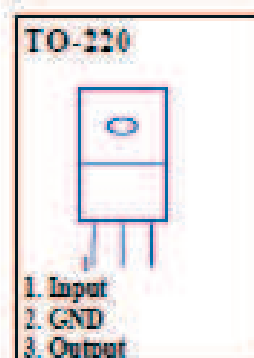
Characteristic	Symbol	Norm	Unit
Input Voltage	$V_{in}$	V	35
Maximum Dissipated Power (with heat sink)	$P_{tot(max)}$	W	15
Maximum Dissipated Power (without heat sink)	$P_{tot(max)}$	W	1.5
Thermal Resistance Junction to Case	$\theta_{jC}$	$^\circ\text{C/W}$	5.0
Thermal Resistance Junction to Air	$\theta_{jA}$	$^\circ\text{C/W}$	65
Junction Temperature	$T_j$	150	$^\circ\text{C}$

 $T_c = -45 \text{ to } +70^\circ\text{C}$ 

## ELECTRICAL CHARACTERISTICS

 $(V_{in} = 10\text{V}, I_o = 0.5\text{A}, C_i = 0.33\text{mF}, C_o = 0.1\text{mF}, T_j = 0 \text{ to } 125^\circ\text{C}, \text{ unless otherwise noted.})$ 

Characteristic	Symbol	Norm.			Unit
		Min	TYP	Max	
Output Voltage ( $T_j = 25^\circ\text{C}$ )	$V_o$	4.8		5.2	V
Output Voltage ( $5.0\text{mA} \leq I_o \leq 1.0\text{A}, P_o \leq 15\text{W}$ ) $7.0\text{V} \leq V_{in} \leq 20\text{V}$	$V_o$	4.75		5.25	V
Line Regulation ( $T_j = +25^\circ\text{C}$ ) $7.0\text{V} \leq V_{in} \leq 25\text{V}$ $8.0\text{V} \leq V_{in} \leq 12\text{V}$	$\Delta V_v$			100 50	mV
Load Regulation ( $T_j = +25^\circ\text{C}$ ) $5.0\text{mA} \leq I_o \leq 1.5\text{A}$ $0.25\text{A} \leq I_o \leq 0.75\text{A}$	$\Delta V_l$			100 50	mV
Quiescent Current ( $T_j = +25^\circ\text{C}$ )	$I_b$			8.0	mA
Quiescent Current Change $7.0\text{V} \leq V_{in} \leq 25\text{V}$ $5.0\text{mA} \leq I_o \leq 1.0\text{A}$	$\Delta I_b$			1.3 0.5	mA
Dropout Voltage ( $I_o = 1.0\text{A}, T_j = +25^\circ\text{C}$ )	$V_i - V_o$		2.0		V
Short Circuit Current Limit ( $T_a = +25^\circ\text{C}, V_{in} = 35\text{V}$ )	$I_{sc}$		0.4		A
Peak Output Current ( $T_j = +25^\circ\text{C}$ )	$I_{max}$		2.2		A
Average Temperature Coefficient of Output Voltage	$TCV_o$		0.3		mV/ $^\circ\text{C}$



## ANEXO 8

### MANUAL CJP 1117

#### GENERAL DESCRIPTION

The CJ1117 is a series of low dropout three-terminal regulators with a dropout of 1.15V at 1A output current.

The CJ1117 series provides current limiting and thermal shutdown. Its circuit includes a trimmed bandgap reference to assure output voltage accuracy to be within 1%. Current limit is trimmed to ensure specified output current and controlled short-circuit current. On-chip thermal shutdown provides protection against any combination of overload and ambient temperature that would create excessive junction temperature.

The CJ1117 has an adjustable version, that can provide the output voltage from 1.25V to 12V with only 2 external resistors.

The CJ1117 series is available in the industry standard SOT-223, SOT-89, TO-220, TO-252 and TO-263 power packages.

#### FEATURES

- Low Dropout Voltage: 1.15V at 1A Output Current
- Trimmed Current Limit
- On-Chip Thermal Shutdown
- Three-Terminal Adjustable or Fixed 1.5V, 1.8V, 2.5V, 3.3V, 5V
- Operation Junction Temperature: 0 °C to 125°C

#### APPLICATIONS

- PC Motherboard
- LCD Monitor
- Graphic Card
- DVD-Video player
- NIC/Switch
- Telecom Modem
- ADSL Modem
- Printer and other peripheral Equipment



Figure 1. Package Types of CJ1117

## PIN CONFIGURATIONS

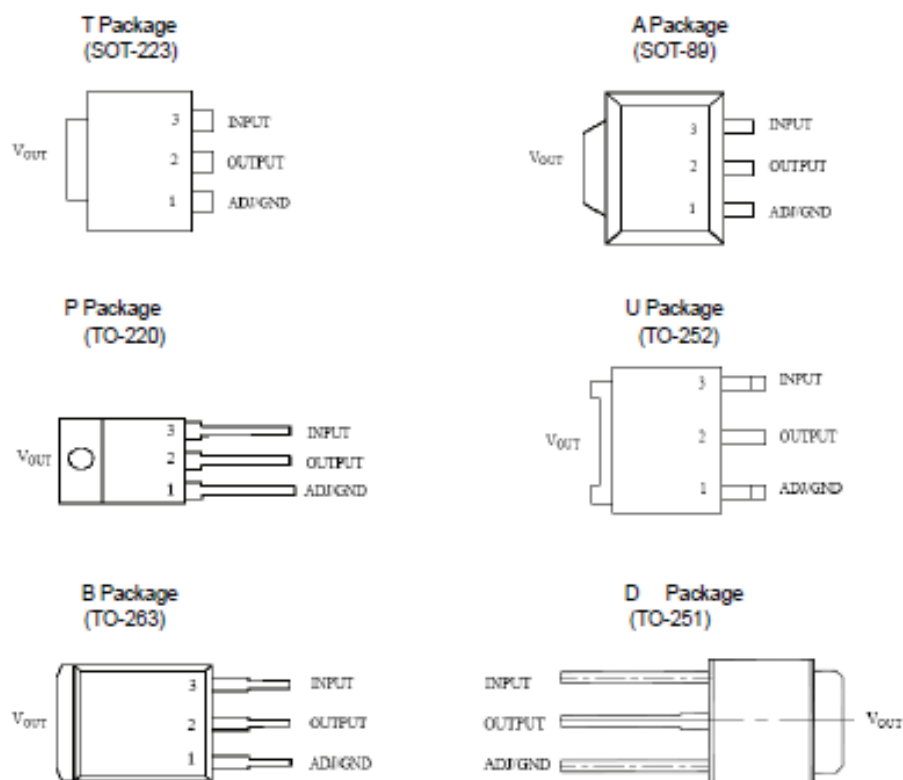
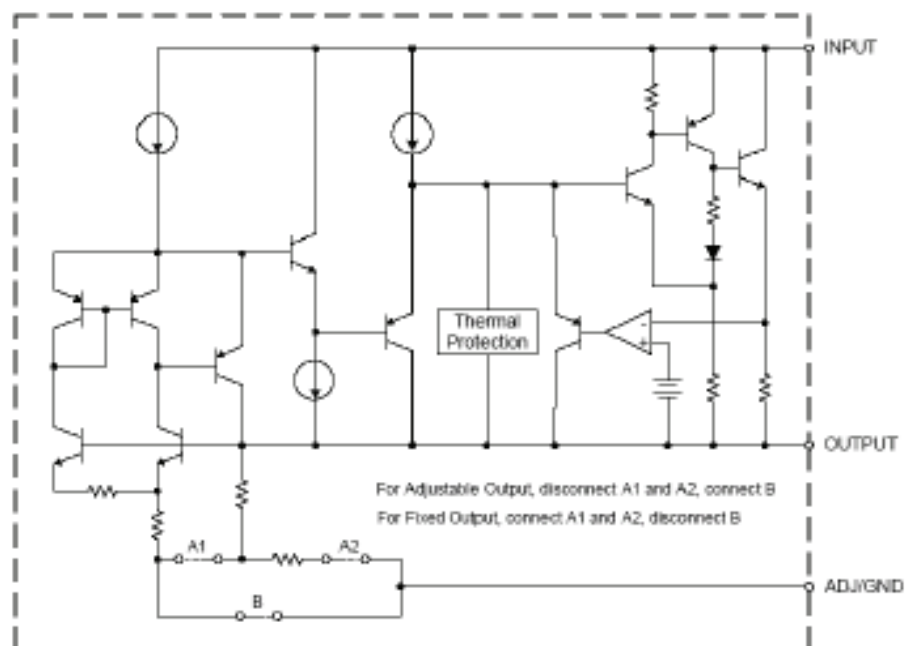


Figure 2. Pin Configurations of CJ1117

## FUNCTIONAL BLOCK DIAGRAM



## ELECTRICAL CHARACTERISTICS

Operating Conditions:  $V_{IN}=10V$ ,  $T_A=25^{\circ}C$  unless otherwise specified.

Parameter	Conditions	Min	Typ	Max	Unit	
Reference Voltage	CJ1117-ADJ $I_{OUT}=10mA$ , $V_{IN}=2V$ , $10mA \leq I_{OUT} \leq 1A$ , $1.4V \leq V_{IN} - V_{OUT} \leq 8V$ , $P \leq$ Maximum power Dissipation	1.238 1.225	1.250 1.250	1.262 1.270	V	
	CJ1117-1.5 $I_{OUT}=10mA$ , $V_{IN}=3.5V$ $10mA \leq I_{OUT} \leq 1A$ , $3.0V \leq V_{IN} \leq 10V$	1.485 1.470	1.5 1.5	1.515 1.530	V	
Output Voltage	CJ1117-1.8 $I_{OUT}=10mA$ , $V_{IN}=3.8V$ $10mA \leq I_{OUT} \leq 1A$ , $3.2V \leq V_{IN} \leq 10V$	1.782 1.746	1.8 1.8	1.818 1.854	V	
	CJ1117-2.5 $I_{OUT}=10mA$ , $V_{IN}=4.5V$ $10mA \leq I_{OUT} \leq 1A$ , $3.9V \leq V_{IN} \leq 10V$	2.475 2.450	2.5 2.5	2.525 2.550	V	
	CJ1117-3.3 $I_{OUT}=10mA$ , $V_{IN}=5.0V$ $10mA \leq I_{OUT} \leq 1A$ , $4.75V \leq V_{IN} \leq 10V$	3.267 3.235	3.3 3.3	3.333 3.365	V	
	CJ1117-5.0 $I_{OUT}=10mA$ , $V_{IN}=7.0V$ $10mA \leq I_{OUT} \leq 1A$ , $6.5V \leq V_{IN} \leq 12V$	4.950 4.900	5.0 5.0	5.050 5.100	V	
	Line Regulation	CJ1117-ADJ $I_{OUT}=10mA$ , $1.5V \leq V_{IN} - V_{OUT} \leq 10V$		0.035	0.2	%
		CJ1117-1.5 $I_{OUT}=10mA$ , $1.5V \leq V_{IN} - V_{OUT} \leq 10V$		1	5	mV
		CJ1117-1.8 $I_{OUT}=10mA$ , $1.5V \leq V_{IN} - V_{OUT} \leq 10V$		1	5	mV
		CJ1117-2.5 $I_{OUT}=10mA$ , $1.5V \leq V_{IN} - V_{OUT} \leq 10V$		1	5	mV
CJ1117-3.3 $I_{OUT}=10mA$ , $1.5V \leq V_{IN} - V_{OUT} \leq 10V$			1	5	mV	
CJ1117-5.0 $I_{OUT}=10mA$ , $1.5V \leq V_{IN} - V_{OUT} \leq 10V$			1	5	mV	
Load Regulation		CJ1117-ADJ $V_{IN}-V_{OUT}=2V$ , $10mA \leq I_{OUT} \leq 1A$		0.2	0.4	%
		CJ1117-1.5 $V_{IN}-V_{OUT}=2V$ , $10mA \leq I_{OUT} \leq 1A$		1	10	mV
	CJ1117-1.8 $V_{IN}-V_{OUT}=2V$ , $10mA \leq I_{OUT} \leq 1A$		1	10	mV	
	CJ1117-2.5 $V_{IN}-V_{OUT}=2V$ , $10mA \leq I_{OUT} \leq 1A$		1	10	mV	
	CJ1117-3.3 $V_{IN}-V_{OUT}=2V$ , $10mA \leq I_{OUT} \leq 1A$		1	10	mV	
	CJ1117-5.0 $V_{IN}-V_{OUT}=2V$ , $10mA \leq I_{OUT} \leq 1A$		1	15	mV	
	Dropout Voltage	$\Delta V_{REG}=1\%$ , $I_{OUT}=0.1A$		1.00	1.1	V
		$\Delta V_{REG}=1\%$ , $I_{OUT}=0.5A$		1.08	1.18	V
$\Delta V_{REG}=1\%$ , $I_{OUT}=1.0A$			1.15	1.25	V	
Current Limit	$V_{IN}-V_{OUT}=2V$	1.25	1.35		A	
Adjust Pin Current			60	120	$\mu A$	
Adjust Pin Current Change	$1.4V \leq V_{IN} - V_{OUT} \leq 10V$ , $10mA \leq I_{OUT} \leq 1A$		0.2	5	$\mu A$	
Minimum Load Current (ADJ)	$1.5V \leq V_{IN} - V_{OUT} \leq 10V$ (ADJ only)		1.7	5	mA	
Quiescent Current	$V_{IN} \leq V_{OUT} + 1.25V$		5	10	mA	
Ripple Rejection	$f=120Hz$ , $C_{OUT}=22\mu F$ Tantalum, $V_{IN}-V_{OUT}=3V$ , $I_{OUT}=1A$	60	75		dB	
Temperature Stability			0.5		%	
Long-Term Stability	$T_A=125^{\circ}C$ , 1000hrs		0.3		%	
RMS Output Noise (% of $V_{OUT}$ )	$T_A=25^{\circ}C$ , $10Hz \leq f \leq 10KHz$		0.003		%	
Thermal Resistance, Junction to Case	DOT-223		15		$^{\circ}C/W$	
	TO-252/251		10			
	TO-220		4.5			
	TO-263		4			
Thermal Shutdown	Junction Temperature		150		$^{\circ}C$	
Thermal Shutdown Hysteresis			25		$^{\circ}C$	

## ANEXO 9

### PROGRAMAS ADICIONALES

#### PROGRAMA PARA LA RECOLECCION DE PULSOS EN BASCOM

```

$regfile = "m164pdef.dat"           ' specify the used micro

$crystal = 8000000                  ' used crystal frequency

$hwstack = 32                       ' default use 32 for the hardware stack

$swstack = 10                       ' default use 10 for the SW stack

$framesize = 40

Config Pina.0 = Input                'desde fpga pin J1_0 a pina.0

Config Pina.1 = Input                'desde fpga pin J1_1 a pina.1

Config Pina.2 = Output               'desde pina.2 a fpga pin J1_0

Config Pinc.0 = Output               'config pins for derecha as output

Config Pinc.1 = Output

Config Pinc.2 = Output

Config Pinc.3 = Output

On Int0 Contar_puls_der              'we enable the int0 interrupt

Enable Int0                          'we configure the INT0 interrupt to
trigger when a rising edge is detected

Config Int0 = Rising                 'finally we enabled interrupts

On Int1 Contar_puls_izq              'we enable the int0 interrupt

Enable Int1                          'we configure the INT0 interrupt to
trigger when a rising edge is detected

```

Config Int1 = Rising

'finally we enabled interrupts

Enable Interrupts

Dim Contder As Byte

Dim Contizq As Byte

Dim Aux1 As Byte

Dim Aux2 As Byte

Dim Auxsacar1 As Byte

Dim Auxsacar2 As Byte

Dim Auxnumer1 As Byte

Dim Auxnumer2 As Byte

Contder = 0

Contizq = 0

Aux1 = 0

Aux2 = 0

Auxnumer1 = 0

Auxnumer2 = 0

Porta.2 = 0

Do

If Pina.0 = 1 Then

Goto Sacarnum

End If

Loop

End

Contar\_puls\_der:

Disable Interrupts

Disable Int0

Aux1 = Aux1 + 1

Auxnumer1 = Aux1

Enable Int0

Enable Interrupts

Return

Contar\_puls\_izq:

Disable Interrupts

Disable Int1

Aux2 = Aux2 + 1

Auxnumer2 = Aux2

Enable Int1

Enable Interrupts

Return

Sacarnum:

Contder = Auxnumer1

Contizq = Auxnumer2

Aux1 = 0



Aux2 = 0

Auxnumer1 = 0

Auxnumer2 = 0

Auxsacar1 = Contder

Auxsacar1 = Auxsacar1 And 15

Auxsacar2 = Contder

Contder = 0

Rotate Auxsacar2 , Left , 4

Auxsacar2 = Auxsacar2 And 15

Do

Loop Until Pina.1 = 1

Portc = Auxsacar1

Porta.2 = 1  
dato

'le decimos a fpga q se saco al puerto el

'esperamos q el fpga ponga el pin en 0 y lea

MSB de

Do

LoopUntil Pina.1 = 0

Portc = Auxsacar2

Porta.2 = 0  
dato

'le decimos al fpga q se saco al puerto el

Auxsacar1 = Contizq

Auxsacar1 = Auxsacar1 And 15

Auxsacar2 = Contizq

Contizq = 0

```
RotateAuxsacar2 ,Left , 4
```

```
Auxsacar2 = Auxsacar2 And 15
```

```
LSB izq
```

```
'esperamos q el fpga ponga el pin en 1 y lea
```

```
Do
```

```
LoopUntil Pina.1 = 1
```

```
Portc = Auxsacar1
```

```
Porta.2 = 1  
dato
```

```
'le decimos a fpga q se saco al puerto el
```

```
MSB der
```

```
'esperamos q el fpga ponga el pin en 0 y lea
```

```
Do
```

```
LoopUntil Pina.1 = 0
```

```
Portc = Auxsacar2
```

```
Porta.2 = 0  
dato
```

```
'le decimos a fpga q se saco al puerto el
```

```
'aqui el fpga debe poner el bit de aviso en 0 y cuando sea 100ms volver a poner  
en 1 y tambien poner el bit de pedido en 0 por si acaso
```

```
Return
```

## **PROGRAMA PARA LAS SALIDAS PWM A LOS MOTORES**

```
$regfile = "m8def.dat"
```

```
' specify the used micro
```

```
$crystal = 8000000
```

```
' used crystal frequency
```

```
$hwstack = 32
```

```
' default use 32 for the hardware stack
```

```
$swstack = 10
```

```
' default use 10 for the SW stack
```

```
$framesize = 40
```

Config Timer1 = Pwm ,Pwm = 10 , Compare A Pwm = Clear Down , Compare B  
Pwm = Clear Down , Prescale = 1

Config Pinc.0 = Input

Config Pinc.1 = Input

Config Pinb.1 = Output

Config Pinb.2 = Output

Dim VoltajeAs Single

Dim Voltaje1 As Byte

ConfigAdc = Single ,Prescaler = Auto , Reference = Avcc

Start Adc

Enable Interrupts

Dim Mot1 As Word

Dim Mot2 As Word

Mot1 = 0

Mot2 = 0

Do

Mot1 = Getadc(0)

Mot2 = Getadc(1)

Pwm1a = Mot1

Pwm1b = Mot2

Loop

End                    'Aquí el micro controlador termina de sacar los dos pwm para  
que los motores giren a la velocidad que solicita el control dado por el FPGA.