



**UNIVERSIDAD NACIONAL AUTÓNOMA
DE MÉXICO**

FACULTAD DE ESTUDIOS SUPERIORES ARAGÓN

**CONTROL DE UN ROBOT MÓVIL
PARA EL SEGUIMIENTO DE
TRAYECTORIAS**

T E S I S

QUE PARA OBTENER EL TÍTULO DE:

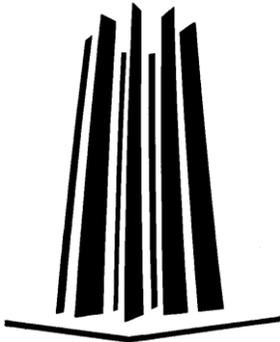
**INGENIERO MECÁNICO
ELECTRICISTA**

P R E S E N T A:

JONATHAN MARTINEZ ROMERO

ASESOR: ING. BENITO ZÚÑIGA VILLEGAS

MÉXICO 2010





Universidad Nacional
Autónoma de México



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

Agradecimientos

A mis padres. Guadalupe Romero, quién me ha dado todo su apoyo, cariño, enseñanzas, etcétera. Aquiles Martínez, ya disfrutando del paraíso.

A Araceli Martínez, por su comprensión, respaldo y sobre todo por ser mi compañera.

A Vanessa Valerio, Elyoenai Martínez y Gaspar Valerio, por esos buenos momentos que hemos pasado.

A Socorro Romero, por su invaluable ayuda durante el estudio de la carrera, y por todas esa comida tan rica que cocina.

A mis profesores, quienes me han dado las bases para poder llevar a buen término esta meta.

A la Universidad Nacional Autónoma de México, por permitirme ingresar y llevar a cabo uno de mis sueños.

A mi asesor, Ing. Benito Zúñiga, por creer en mi proyecto, por su disciplina y consejos.

A los profesores que con sus revisiones mejoraron en gran medida esta tesis: M. en I. Alejandro Vega, Ing. Martín Hernández, M. en C. Rafael Márquez, Ing. Ramón Patiño.

A Sergio Esteves, por su valiosa asesoría y ayuda desinteresada.

Al Centro de Investigación y Estudios Avanzados, donde he podido enriquecer mis conocimientos sobre robótica.

A mis compañeros en el Universum, de lo mejor que me ha pasado.

A quienes me faltaron, por la distracción del mundial, jajaja, es broma.

Gracias, muchas gracias, ¿qué más les puedo decir?

Jonathan Martínez Romero

:o)

Tabla de contenidos

Introducción.....	v
Resumen.....	vii
Capítulo 1: Antecedentes, tipos de robots móviles y planteamiento del problema.	
1.1 Definición de robot móvil y antecedentes históricos.....	1
1.2 Comparación entre robots manipuladores y robots móviles.....	12
1.3 Tipos de ruedas y configuraciones principales de robots móviles con ruedas.....	13
1.4 Planteamiento del problema.....	17
Capítulo 2: Modelos matemáticos del robot.	
2.1 Velocidades lineales y angulares del robot y sus ruedas.....	19
2.2 Modelo cinemático del robot móvil.....	24
2.3 Modelo Jacobiano y restricción no holonómica.....	26
2.4 Modelo odométrico para el robot diferencial.....	31
Capítulo 3: Esquemas de control para el seguimiento de trayectorias.	
3.1 Control usando el método de persecución pura.....	35
3.2 Control no lineal usando seguimiento de postura.....	37
3.3 Control lineal usando seguimiento de postura.....	40
3.4 Control usando modos deslizantes.....	43
Capítulo 4: Simulaciones numéricas.	
4.1 Simulación usando el esquema de persecución pura.....	47
4.2 Simulación usando control no lineal.....	51
4.3 Simulación usando control lineal.....	56
4.4 Simulación usando control por modos deslizantes.....	61
4.5 Animación del robot móvil en el módulo de realidad virtual de Simulink.....	69

Capítulo 5: Diseño de los circuitos electrónicos y programación.

5.1 Descripción del robot móvil utilizado en el trabajo.....	71
5.2 Circuitos de puente H del robot.....	78
5.3 Tarjeta de conexión de los motores y encoders.....	82
5.4 Modulación por ancho de pulsos (PWM), tarjeta con microcontrolador PIC16F887 y puentes H.....	83
5.5 Programa en el compilador mikroC para el PIC16F887.....	89
5.6 Tarjeta de adquisición de datos Sensoray 626.....	90
5.7 Conexión entre la tarjeta de adquisición de datos y el robot.....	92

Capítulo 6: Implementación y pruebas.

6.1 Implementación de sistema de visión para la localización del robot.....	95
6.2 Pruebas con los motores del robot.....	100
6.3 Descripción del sistema de control implementado.....	104
6.4 Pruebas en tiempo real.....	112
Conclusiones y trabajo futuro.....	124
Referencias.....	126

Introducción

La investigación y desarrollo de la robótica en México, así como de otras áreas de la tecnología, se encuentra muy por debajo respecto a otras naciones. Qué importante es para el desarrollo de un país no depender tecnológicamente de otros, crear y proveer en lugar de comprar. Si bien estamos lejos de llegar al nivel de los países llamados del primer mundo, partiendo de bases sólidas en el estudio de los robots móviles poco a poco seremos capaces de generar nuevas aplicaciones y conocimientos.

El campo de la robótica móvil tiene aplicaciones inmediatas, como labores de vigilancia, limpieza y asistencia a enfermos. Por lo que la investigación y desarrollo en esta área cobra una gran relevancia.

Por estos motivos y por razones de interés personal se optó por la elección de este tema, el cual además es muy interesante debido a su carácter multidisciplinario, reflejando el conocimiento de muchas de las asignaturas cursadas durante la carrera.

Este trabajo beneficiará a aquellos estudiantes e investigadores interesados en el tema de la robótica móvil, específicamente en el control de los mismos. Cabe señalar que la bibliografía del tema en nuestro idioma está en desarrollo, por lo que resulta importante contribuir a su difusión.

La investigación en robótica móvil da pie a varios temas: diseño mecánico, localización, sistemas de control para robots individuales o grupos de robots, evasión de obstáculos, etc. Abarcar todos estos temas en un trabajo de este tipo sería algo muy extenso, así que se decidió enfocarlo a la implementación de un control para el seguimiento de trayectorias de un robot móvil.

El trabajo experimental se llevó a cabo en el Centro de Investigación y Estudios Avanzados del Instituto Politécnico Nacional. En dicho centro se cuenta con un robot móvil diferencial de la marca Acroname (robot García). Dicho robot tal como lo entregan de fábrica está diseñado para realizar tareas simples mediante instrucciones de avance y giro, además del uso de sus sensores óptico reflexivos. Este tipo de tareas no es muy eficiente cuando se quiere implementar un controlador como los que se describen en este trabajo, ya que ni siquiera existe una instrucción para obtener la localización del robot.

Para poder emplear el robot en tareas más útiles se necesitan conocimientos avanzados de C++ y de sus microcontroladores. Pero la documentación para programar el robot es escasa, o no corresponde a los nuevos códigos que el fabricante está implementando.

Debido a lo anterior se optó por realizar la implementación del controlador en una computadora, usando el programa MATLAB / Simulink; haciendo la interface con el robot mediante una tarjeta de adquisición de datos y aprovechando la etapa de

potencia incluida con el robot. Una implementación de este tipo permite un control directo de los motores del robot, al mismo tiempo que se consigue una mayor flexibilidad para realizar cambios en el controlador. Así como la ventaja de contar con las herramientas y librerías de MATLAB, tales como las de creación de graficas, procesamiento de imágenes, filtrado y funciones matemáticas, entre otras.

Otro problema es obtener la localización del robot en el plano X-Y para todo instante. Para resolver esto se pueden usar los sensores ópticos conectados a los motores, pero problemas como la baja resolución de los sensores, deslizamiento de las ruedas o irregularidades en el piso, impiden usar esta técnica eficientemente.

Una solución más adecuada al problema es la implementación de un sistema de visión, el cual es un sistema de localización absoluta, que no requiere el conocimiento de donde inicia el robot ni con que orientación, además de que no se ve afectado por los problemas mencionados en el párrafo anterior. Por lo tanto se optó por implementar un sistema de este tipo.

Resumen

Este trabajo se enfoca en la implementación de un sistema de control, con el objetivo de que un robot móvil sea capaz de seguir trayectorias. Se listan diferentes hechos históricos de la robótica móvil. Se obtienen modelos matemáticos que describen al robot. Se presentan diferentes esquemas de control, así como resultados de simulaciones numéricas de los mismos.

La implementación se realiza utilizando un robot comercial, al que se le hicieron modificaciones, y con la ayuda de una tarjeta de adquisición de datos. Se desarrollaron tarjetas electrónicas, para crear una interface entre la computadora de control y el robot. Se implementó un sistema de localización del robot basado en visión. Finalmente, se muestran los resultados experimentales del sistema.

CAPÍTULO 1

Antecedentes, tipos de robots móviles y planteamiento del problema

1.1 Definición de robot móvil y antecedentes históricos.

El ser humano a lo largo de su historia, ha tratado siempre de ir más allá de sus límites físicos. Diversos sueños, muchos ya realizados, originados por necesidades, se han forjado en la mente de muchos hombres. Sueños tales como volar, transportarse de un lugar a otro de manera eficiente, establecer una comunicación entre lugares alejados. En fin se podrían mencionar un gran número de ejemplos como estos.

Lo interesante aquí es ese sueño de crear una máquina semejante a un ser vivo (humano, animal ó vegetal), y que desempeñe labores en beneficio de nuestra especie, es decir, un robot.

La Real Academia Española de la Lengua define a un robot como: una máquina o ingenio electrónico programable, capaz de manipular objetos y realizar operaciones, antes reservadas solo a las personas.

El estándar ISO 8373:1994 define a un robot industrial como un manipulador programable en tres o más ejes, multipropósito, controlado automáticamente y reprogramable.

Una definición más completa de un robot es la siguiente [2]:

“Una máquina diseñada para mimetizar los comportamientos de humanos y animales, con frecuencia en un esfuerzo para hacer tareas que son indeseables, peligrosas o inaccesibles a los humanos, y con frecuencia logradas a través del uso de sensores, actuadores y programas de control mediante computadora”

Entre las aplicaciones peligrosas, monótonas e indeseables que pueden realizar robots móviles, y que ya están en comercialización, se encuentran: limpieza de pisos, poda de pasto y desactivación de bombas.

Es este apartado conoceremos algunos de los hechos más significativos en la historia y evolución de la robótica móvil.

El término robot se utilizó por primera vez en 1920 en la obra teatral R.U.R. (Rossum's Universal Robots – Los robots universales de Rossum), escrita por el checo Karel Capek (figura 1.1). El término robot se deriva de la palabra *robota*, que en checo significa fuerza de trabajo o servidumbre. En dicha obra, los robots que aparecen son de tipo humanoide y son fabricados en una isla en grandes cantidades. Seres apenas distinguibles de los seres humanos, pero con más fuerza e inteligencia y capacitados para realizar cualquier trabajo de modo más eficiente, además de un nivel más elevado de conciencia y sentimientos, y que acaban sublevándose y aniquilando a la raza humana.



Fig. 1.1 Karel Capek y su obra R. U. R.

Robots de este tipo aparecen en obras literarias y películas de ciencia ficción, robots semejantes a los humanos, robots inteligentes. Pero la realidad es que con los avances en inteligencia artificial, y con las computadoras actuales, sólo es posible ir un poco más allá del nivel de procesamiento de un insecto, de acuerdo a las investigaciones de Hans Moravec¹, quién sostiene que algún día la tecnología reemplazará a la biología como vehículo de evolución.

Cabe señalar que antes de que se utilizará el término robot, se empleaba la palabra autómeta para referirse a máquinas automáticas puramente mecánicas, que tuvieron su auge en figuras con movimiento creadas por relojeros, principalmente en Francia y Alemania. Entre estos autómetas destaca uno que era capaz de dibujar y escribir, creado por el suizo Henri Maillardet, y un pato mecánico construido por el francés Jacques de Vaucanson (figura 1.2).

Los robots móviles surgen durante la segunda guerra mundial en la forma de bombas voladoras inteligentes, que detonaban solo a una cierta distancia del objetivo.

¹ Hans Moravec es investigador del Instituto de Robótica de la Universidad Carnegie Mellon en Pittsburgh, Pennsylvania, E.U.A.

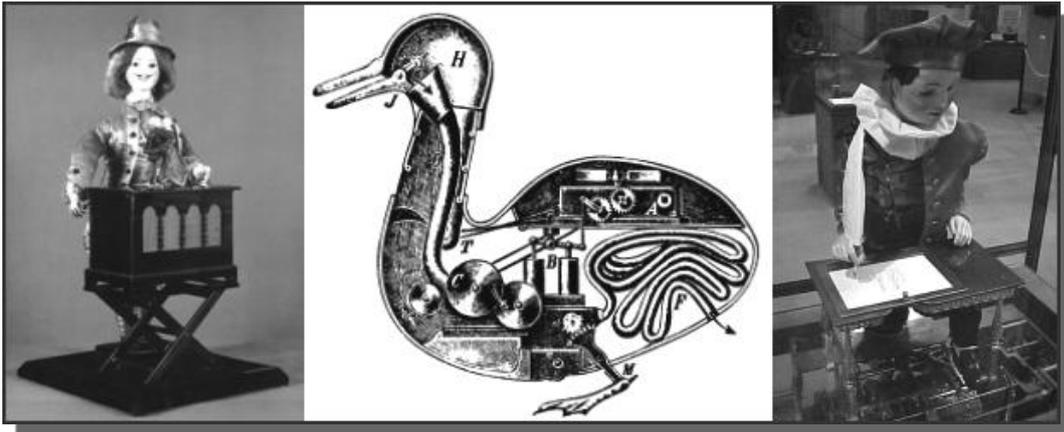


Fig. 1.2 Autómatas mecánicos

En los años 40 se introdujo la palabra “Automatización” en la Ford Motor Company, siendo la contracción de las palabras Automatic Motivation (en Inglés Automation), o motivación² automática.

Entre 1948 y 1949, William Grey Walter (figura 1.3) construye dos robots autónomos parecidos a tortugas, a los que llamó Elsie y Elmer. Estos robots eran capaces de detectar y seguir una fuente de luz, al mismo tiempo que evitaban obstáculos.

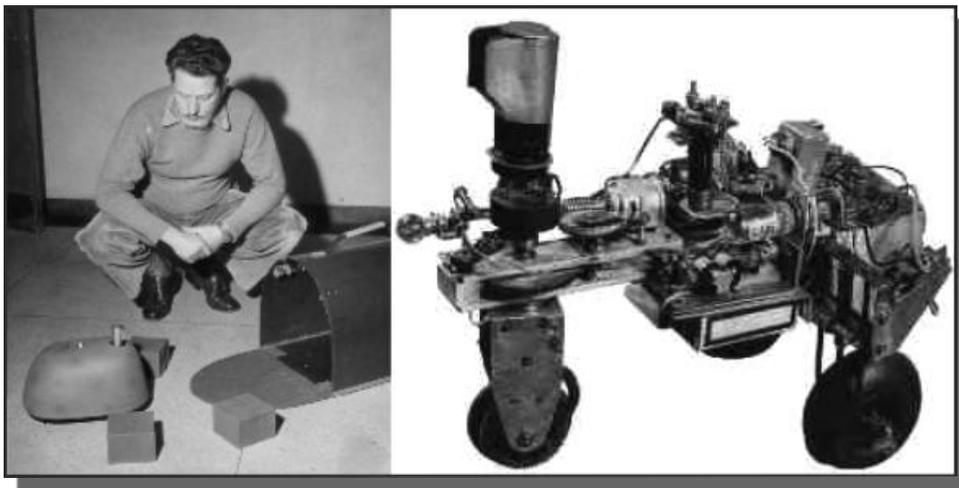


Fig. 1.3 W. G. Walter y su robot móvil

A principios de los 60 se desarrolló un robot móvil en el laboratorio de física aplicada de la universidad Johns Hopkins; se le dio el nombre de Johns Hopkins Beast (figura 1.4). Este robot navegaba por el centro de investigación vía sonar, y si la carga de su batería disminuía a un cierto nivel, era capaz de detectar, mediante optoelectrónica, una toma de corriente negra que contrastaba con las paredes blancas.

El primer robot móvil capaz de podar pasto se creó en 1969, llamado Mowbot (figura 1.5).

²La palabra motivación proviene del latín *motivus*, que significa mover.

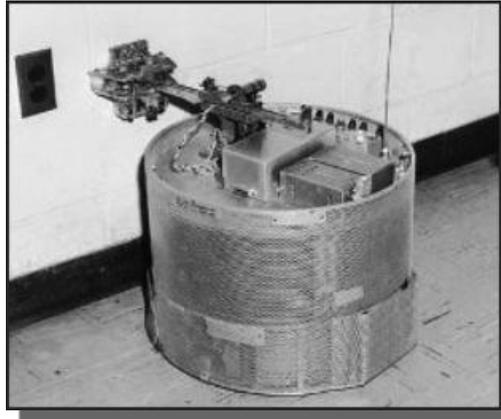


Fig. 1.4 Robot Johns Hopkins Beast



Fig. 1.5 Mowbot, robot podador de pasto

En 1970 se crea el carro Stanford, capaz de seguir una línea blanca usando una cámara. Por ese entonces se trabaja en el desarrollo y construcción del robot Shakey en el Instituto Stanford, este último podía recibir una serie de comandos para realizar una tarea determinada (figura 1.6).

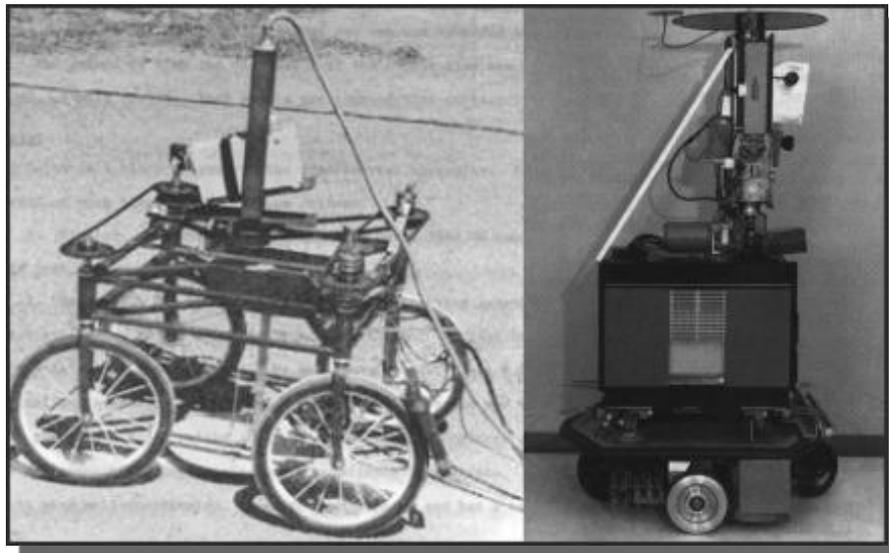


Fig. 1.6 Stanford cart (izquierda), Shakey (derecha)

A finales de 1970 los rusos mandan un robot para explorar la superficie lunar; Lunokhod 1 (figura 1.7), un robot de 2.3 metros de largo, que contaba con antenas, cámaras de televisión, baterías recargables y equipo especializado para análisis.

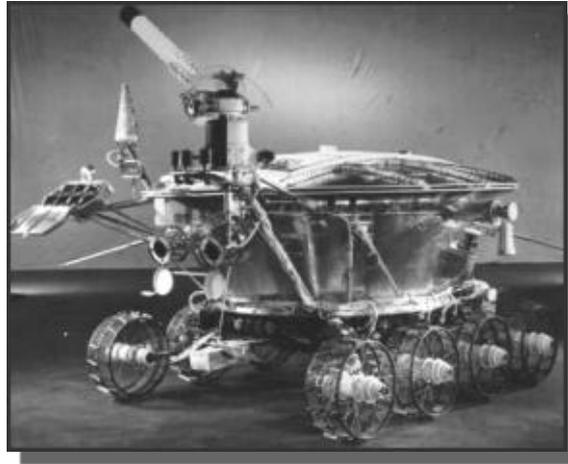


Fig. 1.7 Robot Lunokhod 1, para exploración lunar

El microprocesador, un componente indispensable, no sólo para la robótica sino para muchas áreas de la electrónica, aparece en 1971 y es desarrollado por Intel.

En 1976 la NASA manda dos naves espaciales no tripuladas a Marte, vía el programa Viking.

Para 1980 el carro Stanford es capaz de navegar a través de caminos con obstáculos y hacer mapas de su entorno.

Por ese tiempo se crea el primer carro robot en la universidad Bundeswehr de Munich, este vehículo alcanzaba una velocidad de 88 Km / h. Surgen también robots comerciales para uso en el hogar, con propósitos educativos y de diversión, como el robot RB5X ó los robots HERO (figura 1.8).

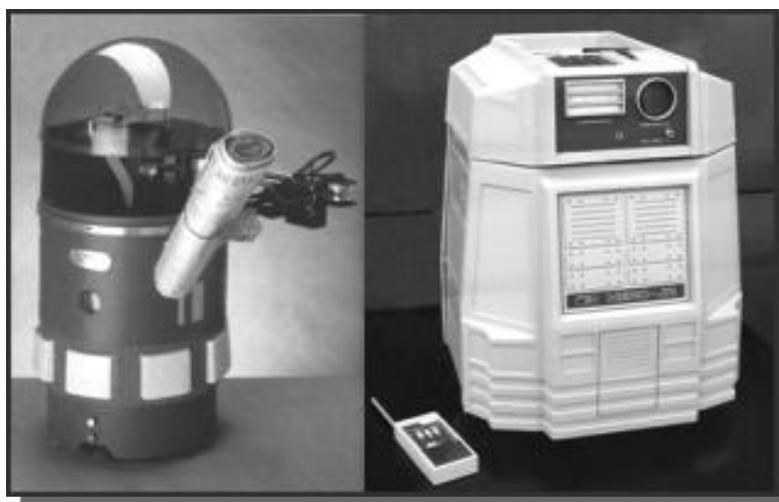


Fig. 1.8 Robot RB5X (izquierda) y robot HERO (derecha)

El desarrollo de robots para uso en hospitales inicia en 1984 cuando Joseph Engelberger crea la empresa Transition Research Corp (llamada después Helpmate Robotics Inc.)

El primer Robot HelpMate (Compañero) es introducido en 1988 en el Hospital Danbury en Connecticut (figura 1.9); las funciones de estos robots es repartir medicamentos, alimentos o documentos entre los pacientes con base en trayectorias programadas, de acuerdo a las necesidades.



Fig. 1.9 Robot HelpMate para hospitales

La utilización de robots en el ámbito de la asistencia también está presente en la ayuda de personas con alguna discapacidad ó en rehabilitación, por lo que se les conoce como robots asistenciales. En la década de los 90 han surgido robots tales como el MIT – Manus, creado por el Instituto Tecnológico de Massachussets y el Centro de Investigación TNO de Holanda. Este robot consta básicamente de una silla de ruedas motorizada con un brazo robótico, que entre otras cosas puede servir para dar de comer al paciente (figura 1.10).



Fig. 1.10 Robot Asistencial MIT-Manus

En 1991 Franzini, Guignard y Mondana desarrollan el robot Khepera (figura 1.11), un robot móvil pequeño y autónomo para propósitos de investigación.

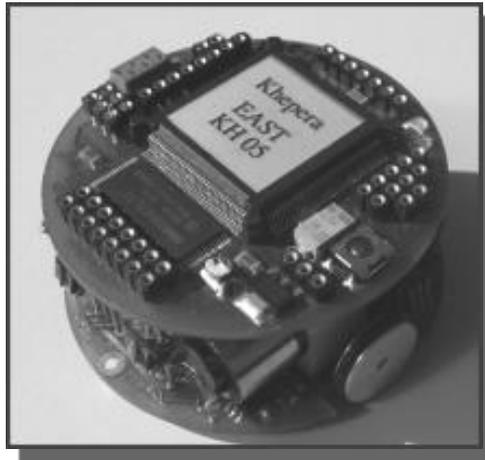


Fig. 1.11 Robot Khepera

En 1993 y 1994 se desarrollan los robots DANTE I y DANTE II (figura 1.12) por la universidad Carnegie Mellon. Se trata de robots caminadores para explorar volcanes activos.



Fig. 1.12 Robot Dante II

En 1994 los carros robots gemelos VaMP y VITA-2 recorren más de 1000 Km con tráfico normal, en una autopista de tres carriles en París, con velocidades superiores a 130 Km / h, utilizando técnicas de visión.

En 1995 se lanza a la venta el robot programable Pioneer (figura 1.13), posibilitando una mayor investigación en el área de la robótica móvil.



Fig. 1.13 Robot Pioneer

Entre 1996 y 1997 la NASA hace llegar a Marte al robot Sojourner (figura 1.14), equipado con cámaras, sensores y dispositivos especiales para realización de análisis y experimentos. Se trata de un robot de 65 cm de longitud con 6 ruedas y una velocidad máxima de un centímetro por segundo.

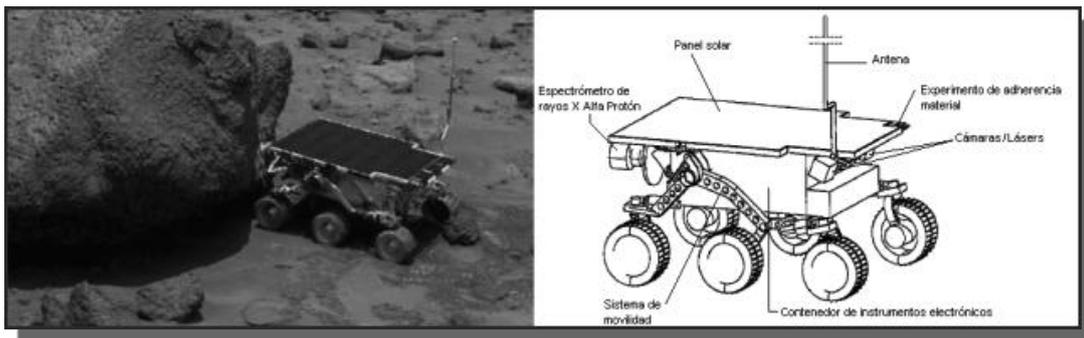


Fig. 1.14 Robot Sojourner

En 1999 se introduce al mercado un perro robótico llamado AIBO (**Artificial Intelligence roBOT** - robot con inteligencia artificial), capaz de caminar, “ver” su entorno gracias a una cámara y reconocer comandos hablados (figura 1.15).



Fig. 1.15 Robot AIBO

En el año 2000 la empresa Honda crea el robot ASIMO (Advanced Step in Innovative Mobility – paso avanzado en movilidad innovadora). Se trata de un robot humanoide capaz de moverse e interactuar con los humanos (figura 1.16).



Fig. 1.16 Robot ASIMO

En 2001 se inician proyectos de enjambres de robots. Estos enjambres consisten de un gran número de robots simples, que en conjunto, pueden ejecutar una tarea sofisticada.

En 2002 aparece un robot móvil autónomo para limpieza de pisos, llamado Roomba (figura 1.17).



Fig. 1.17 Robot Roomba

En 2003 la empresa Acroname introduce el robot Garcia (figura 1.18), un robot móvil de dos ruedas, como el utilizado en este trabajo de investigación. Se describirá más adelante.

En este mismo año, la NASA lanza la misión MER (Mars Exploration Rover – Robot de exploración de Marte), que incluye el envío de los robots Spirit (Espíritu) y Opportunity (Oportunidad) a la superficie de Marte (figura 1.19).



Fig. 1.18 Robot Garcia



Fig. 1.19 Robot de la misión MER

En 2004 se pone a la venta un robot humanoide, el Robosapien (figura 1.20), creado por Mark Tilden. En este año se lleva a cabo la primera competencia de carros robot llamada DARPA Grand Challenge, que consiste en una carrera en la que vehículos autónomos deben cruzar un desierto.



Fig. 1.20 Robosapien

En 2005 la empresa Boston Dynamics crea un robot cuadrúpedo, para llevar cargas pesadas a través de terrenos complicados para vehículos convencionales, a este robot le llamaron BigDog (figura 1.21).

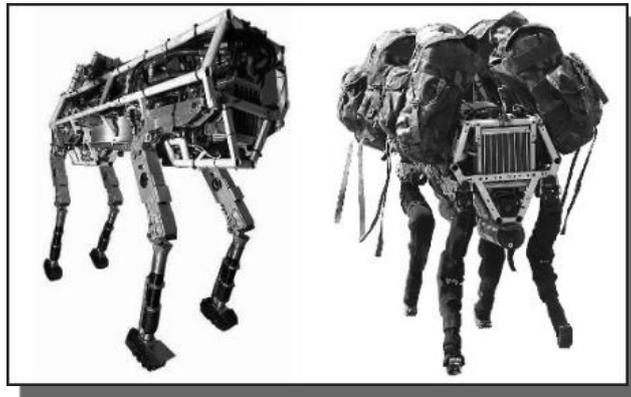


Fig. 1.21 BigDog

En 2006 se deja de producir el robot AIBO. En este mismo año se pone a la venta el robot TALON – SWORD, es el primer robot comercial con lanzagranadas y otras armas integradas opcionales, este se controla por un operador de manera remota (figura 1.22).



Fig. 1.22 TALON SWORD

Para 2007 se lleva a cabo la competencia de la DARPA en una base de la fuerza aérea norteamericana en California, incluyéndose obstáculos y tráfico, en la figura 1.23 se puede apreciar un vehículo de esta competencia.



Fig. 1.23 Vehículo para la competencia de la DARPA 2007

Una nueva versión de BigDog se lanza en 2008, es capaz ahora de caminar en nieve y de recuperar el equilibrio después de ser pateado en un costado. En 2009 el robot ASIMO puede ser controlado por una persona mediante un dispositivo ICC (Interface Computadora Cerebro), con un 90.6 % de aciertos.

Algunas aplicaciones de los robots móviles, derivadas de la investigación y desarrollo, son: Exploración espacial, limpieza y pulido de pisos, vigilancia, poda de pasto, transporte y manipulación, asistencia a enfermos o a personas de la tercera edad, espionaje, recepción y guiado en museos y centros turísticos, desactivación de bombas, manejo de residuos peligrosos, competencia y entretenimiento, Inspección y limpieza de tuberías.

Como puede verse, el campo de la robótica móvil tiene una gran importancia en diversas áreas, sólo se mencionaron algunos robots móviles y se hizo una descripción muy breve de ellos. Cabe señalar también que el área de la robótica que más se ha desarrollado es la de los robots manipuladores, dado el uso que tienen en las industrias. En el siguiente apartado se hace una comparación entre los robots manipuladores y los robots móviles.

1.2 Comparación entre robots manipuladores y robots móviles.

Los robots manipuladores tienen la desventaja de estar confinados a un área de trabajo fija, que para muchos procesos industriales funciona bien. Pero si se quieren realizar otras tareas o dotar a un manipulador de un área de trabajo, que pueda moverse de un sitio a otro, se debe recurrir a los robots móviles.

La tabla 1.1 compara los robots manipuladores y los móviles:

Manipuladores	Móviles
Su espacio de trabajo define el rango de posibles posiciones que pueden ser alcanzadas por su efector final.	El espacio de trabajo define el rango de posibles posiciones que el robot puede alcanzar.
La controlabilidad define la manera en la cual se pueden usar los motores para alcanzar las diferentes posiciones.	La controlabilidad define los posibles caminos y trayectorias en su espacio de trabajo.
La dinámica del robot origina restricciones adicionales en el espacio de trabajo y trayectoria, debido a consideraciones de masa y fuerza.	También está limitado por dinámica, por ejemplo, un centro de gravedad alto limita el radio práctico para dar vueltas.
Un manipulador tiene un extremo fijo al ambiente, medir la posición del efector final se basa en un entendimiento de la cinemática y una medición de la posición de las articulaciones intermedias.	No hay una forma directa de medir la posición instantáneamente, existen imprecisiones en la estimación del movimiento debido a resbalamiento, la medición precisa de la posición es una tarea extremadamente desafiante.

Tabla 1.1 Comparación entre los robots manipuladores y los móviles

1.3 Tipos de ruedas y configuraciones principales de robots móviles con ruedas

Existen muchos tipos de robots móviles que se pueden clasificar de acuerdo a la forma en que se desplazan: caminan, ruedan, saltan, se arrastran, nadan, vuelan, etc.

Las ruedas actuadas son un invento del hombre que tiene una muy alta eficiencia en superficies planas y duras. Un robot móvil con ruedas es un vehículo que es capaz de realizar un movimiento autónomo (sin que actúe un humano de forma externa) porque está equipado para su movimiento con una computadora que controla los actuadores.

El balance no es usualmente un problema en los diseños de robots con ruedas, estos se diseñan para que todas las ruedas estén en contacto con el piso todo el tiempo. Tres ruedas son suficientes para garantizar un balance estable, cuando se usan más de tres ruedas se requiere un sistema de suspensión para permitir que todas las ruedas mantengan contacto con la tierra, cuando el robot encuentra un terreno disparejo.

Hay cuatro tipos principales de ruedas que se ilustran en la figura 1.24:

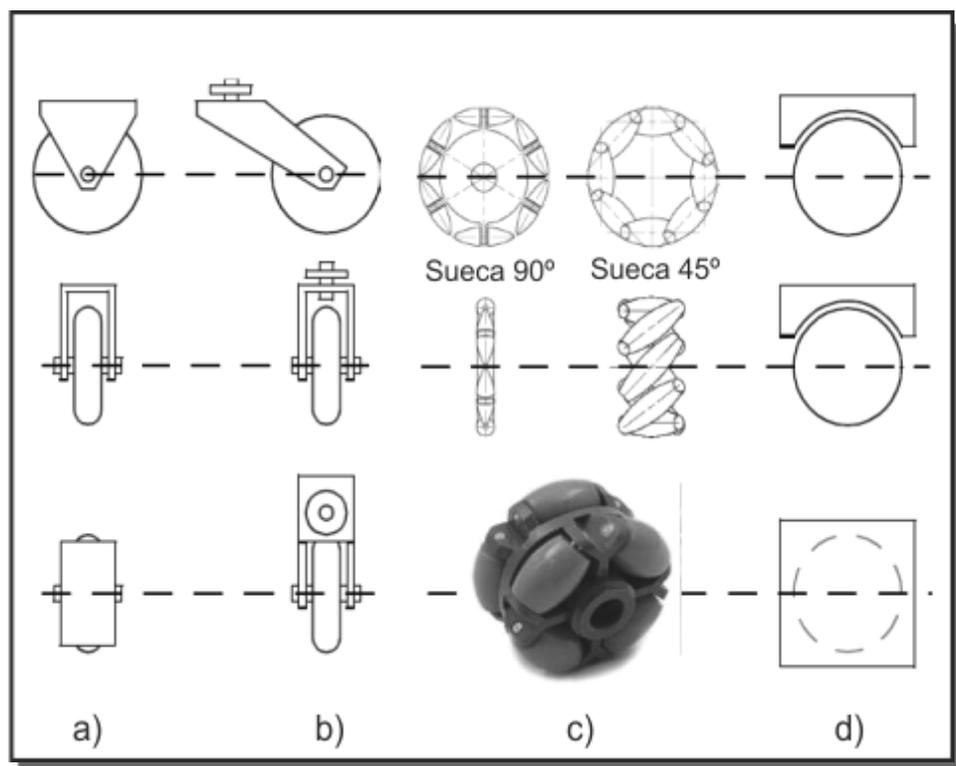


Fig. 1.24 Tipos de ruedas

- (a) Rueda Estándar – 2 DOF (Deegres of Freedom – Grados de Libertad), rotación alrededor del eje (motorizado) de la rueda y el punto de contacto.

- (b) Rueda Castor – 2 DOF, rotación alrededor de una articulación de dirección con un offset.
- (c) Rueda Sueca – 3 DOF, rotación alrededor del eje (motorizado) de la rueda, alrededor de los cilindros y del punto de contacto.
- (d) Rueda de bola o esférica – verdaderamente omnidireccional, realización técnicamente difícil.

Configuraciones principales de los robots móviles³

- **Configuración síncrona ó synchro-drive.** En esta configuración existen transmisiones que permiten orientar las tres ruedas simultáneamente con una velocidad angular, ω , y hacer que el vehículo se desplace con una velocidad lineal, v . En esta configuración se tiene una restricción no holónomica (ver capítulo 2) y dos grados de libertad. La orientación es no controlable (figura 1.25).

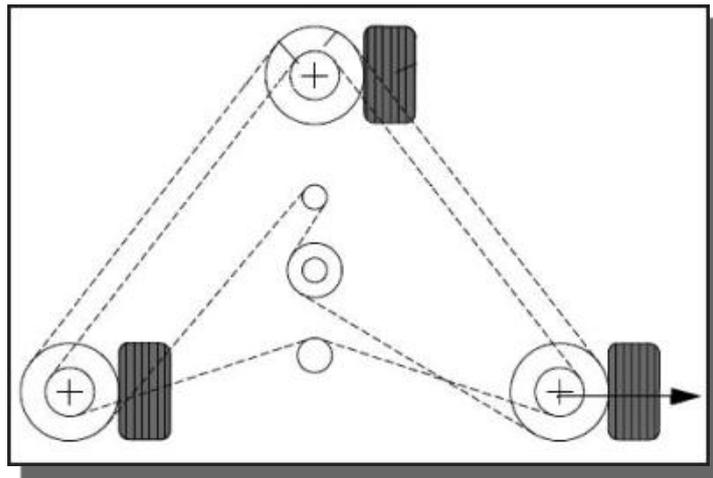


Fig. 1.25 Configuración síncrona

- **Configuración con Guiado Diferencial.** Las variables de control son las velocidades de las ruedas laterales. El centro de guiado del vehículo (origen del sistema de referencia local) se considera en el punto medio de las ruedas traseras. Está configuración en la mayoría de los casos cuenta con una rueda delantera no actuada que sirve como apoyo para lograr la estabilidad del vehículo (figura 1.26).

³ Una clasificación muy completa aparece en [11]

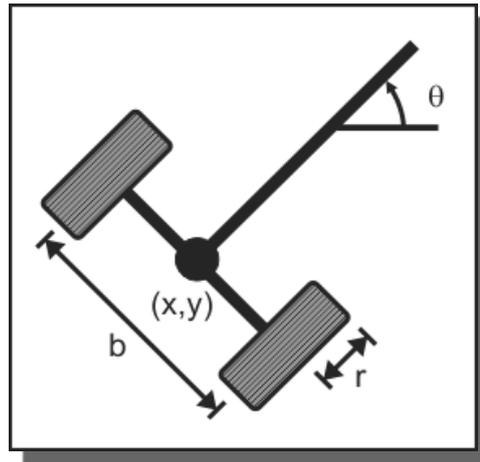


Fig. 1.26 Configuración diferencial

- **Configuración de Triciclo Clásico.** La rueda delantera se utiliza tanto para la orientación como para la tracción. En este caso las variables de control suelen tomarse como el ángulo de dirección de la rueda delantera (ó su velocidad angular) y la velocidad de giro de la misma rueda (ó su velocidad lineal correspondiente). Se supone que el punto guía (x, y) está en el centro del eje trasero (figura 1.27).

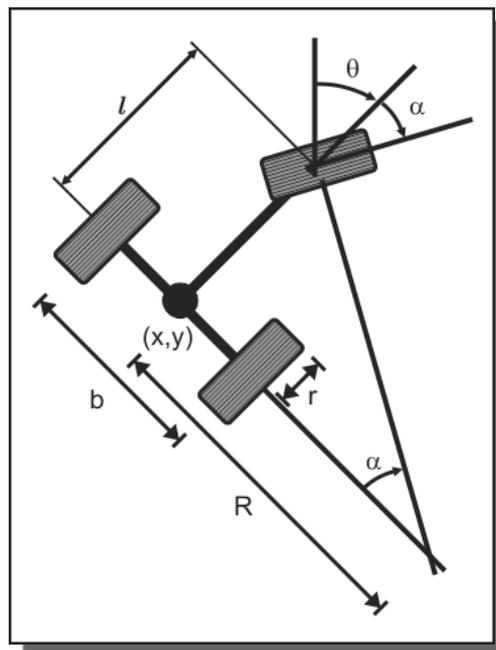


Fig. 1.27 Configuración de triciclo clásico

- **Configuración Ackerman.** Es la configuración que tienen las ruedas del automóvil. Se supone que el centro de guiado del vehículo está situado a la mitad del eje de las ruedas de tracción (ruedas traseras). Pocos robots usan esta configuración debido a su pobre maniobrabilidad (figura 1.28).

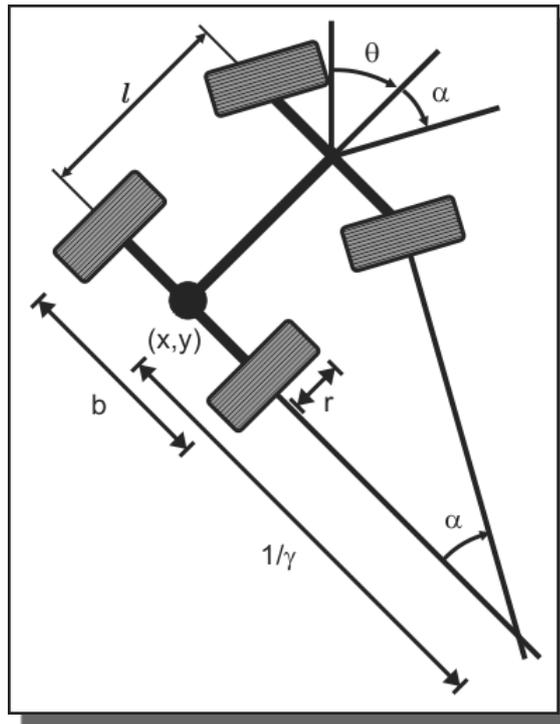


Fig. 1.28 Configuración Ackerman

La elección del tipo y geometría de las ruedas afecta tres características básicas de los robots móviles: maniobrabilidad, controlabilidad y estabilidad.

Estabilidad

Un robot de dos ruedas diferenciales puede lograr estabilidad si el centro de masa está debajo del eje de las ruedas.

La estabilidad estática requiere un mínimo de tres ruedas, además de que el centro de gravedad debe de estar contenido dentro del triángulo formado por los puntos de contacto de las ruedas.

Maniobrabilidad

Los robots omnidireccionales son capaces de moverse en cualquier instante y en cualquier dirección a lo largo del plano (x, y) , a pesar de la orientación del robot alrededor de su eje vertical. Este nivel de maniobrabilidad requiere ruedas suecas ó esféricas.

La configuración de cuatro ruedas castor, en la cual cada rueda es activamente dirigida y activamente trasladada, ofrece una solución al problema de navegación omnidireccional, sin el problema de la distancia al suelo que se presenta por las restricciones mecánicas de las ruedas suecas y esféricas.

Otra clase de robots con una alta maniobrabilidad tiene un chasis circular y un eje de rotación en el centro del robot, tal robot puede girar sin cambiar su espacio ocupado.

La dirección Ackerman tiene una limitada maniobrabilidad, pero su direccionalidad y geometría de dirección hacen que se tenga muy buena estabilidad lateral en vueltas a alta velocidad.

Controlabilidad

Hay una correlación inversa entre controlabilidad y maniobrabilidad. Los diseños omnidireccionales requieren procesamiento significativo para convertir las velocidades traslacionales y rotacionales a comandos individuales para cada rueda, además de que cada una de estas últimas presenta múltiples grados de libertad.

1.4 Planteamiento del problema

El problema consiste en que dada una trayectoria predefinida, es decir un camino que se quiere que el robot siga, deben obtenerse e implementarse leyes de control para las velocidades de las ruedas del móvil para conseguir el seguimiento.

Esto conlleva a cumplir con los siguientes objetivos:

- Obtener un modelo cinemático para el robot móvil.
- Presentar diferentes esquemas para el seguimiento de trayectorias.
- Seleccionar los dispositivos adecuados para el control del vehículo.
- Implementar un método para localizar al robot en el plano X-Y.
- Programar el controlador para el seguimiento de trayectorias en la computadora.
- Construir e implementar los circuitos electrónicos para la interface entre el robot y la computadora con el programa de control.
- Realizar pruebas con el sistema de control implementado.
- Analizar y discutir los resultados, presentar las conclusiones finales y proponer opciones de trabajo a futuro.

Previo a la implementación del sistema de control, resulta adecuado simular el funcionamiento del robot mediante una computadora, esto permite ver que desempeño tiene el robot para diferentes ganancias de las leyes de control, generándose valores seguros para evitar daños en los motores y electrónica asociada.

Limitantes:

- Sólo se desarrollará e implementará un esquema de control, dándose las características generales de diversos tipos de control aplicables al problema.
- Se tratará con un modelo cinemático del robot móvil y se considera que no existe deslizamiento en las ruedas.
- Se asume que todas las partes del robot son rígidas.

CAPÍTULO 2

Modelos matemáticos del robot

En este capítulo se obtienen diferentes modelos que van a ser de gran utilidad cuando se implemente el sistema de control en el robot.

2.1 Velocidades lineales y angulares del robot y sus ruedas

Partiendo de la figura 2.1, es posible derivar ecuaciones para las velocidades lineal (v) y angular (ω) del robot, de estas dos últimas se pueden conocer las velocidades angulares (ω_i y ω_d) de las ruedas del robot.

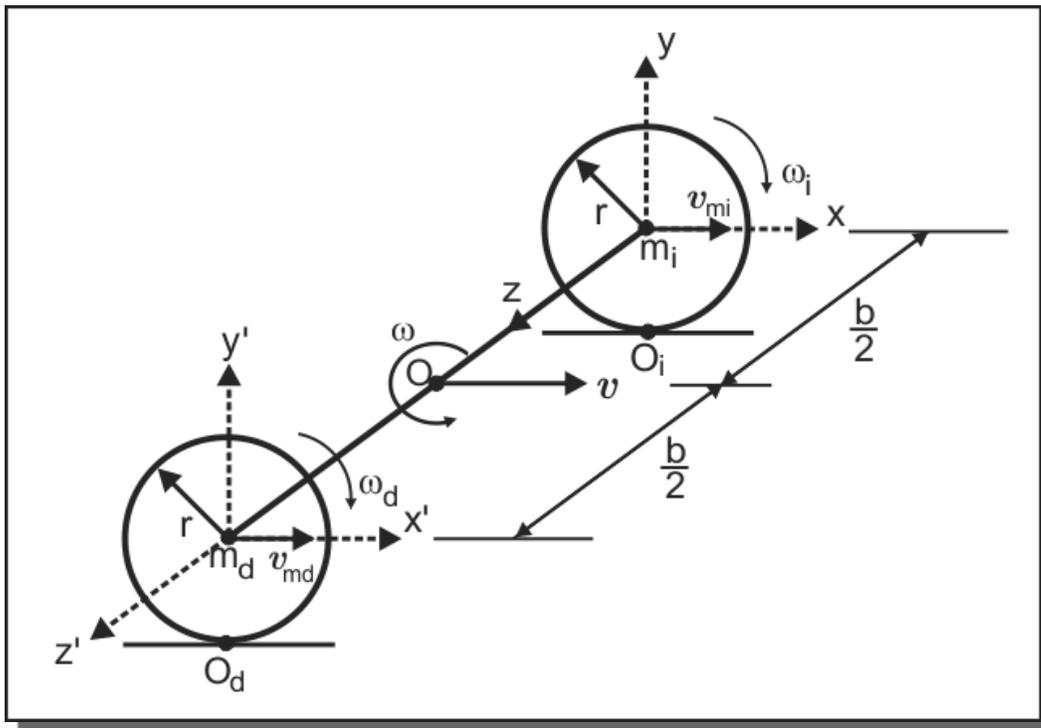


Fig. 2.1 Ruedas del robot móvil y velocidades

m_i – centro de la rueda izquierda

m_d – centro de la rueda derecha

b – distancia entre el centro de las dos ruedas

$\vec{\omega}_i$ – velocidad angular de la rueda izquierda

$\vec{\omega}_d$ – velocidad angular de la rueda derecha

\vec{v}_{mi} – velocidad lineal del centro de la rueda izquierda

\vec{v}_{md} – velocidad lineal del centro de la rueda derecha

\vec{v}_{oi} – velocidad lineal del punto de contacto de la rueda izquierda

\vec{v}_{od} – velocidad lineal del punto de contacto de la rueda derecha

$\vec{r}_{md/oi}$ – radio de la rueda izquierda (vector de m_i a O_i)

$\vec{r}_{md/od}$ – radio de la rueda derecha (vector de m_d a O_d)

\vec{v} – velocidad lineal del robot

$\vec{\omega}$ – velocidad angular del robot

La velocidad lineal del centro de la rueda izquierda es:

$$\vec{v}_{mi} = \vec{v}_{oi} + \vec{\omega}_i \times \vec{r}_{mi/oi} \quad (2.1)$$

La velocidad lineal del centro de la rueda derecha es:

$$\vec{v}_{md} = \vec{v}_{od} + \vec{\omega}_d \times \vec{r}_{md/od} \quad (2.2)$$

Si se considera que no existe deslizamiento, $\vec{v}_{oi} = \vec{v}_{od} = 0$

$$\vec{v}_{mi} = (-\omega_i \hat{k}) \times (r \hat{j}) = \omega_i r \hat{i} \quad (2.3)$$

$$\vec{v}_{md} = (-\omega_d \hat{k}) \times (r \hat{j}) = \omega_d r \hat{i} \quad (2.4)$$

Donde:

\hat{i} – vector unitario en la dirección del eje x

\hat{j} – vector unitario en la dirección del eje y

\hat{k} – vector unitario en la dirección del eje z

La velocidad del centro de cada rueda está relacionada con ω y v de acuerdo a lo siguiente:

$$\vec{v}_{mi} = \vec{v} + \vec{\omega} \times \vec{d}_{mi/o} \quad (2.5)$$

$$\vec{v}_{md} = \vec{v} + \vec{\omega} \times \vec{d}_{md/o} \quad (2.6)$$

Dónde:

$\vec{d}_{mi/o}$ – vector de m_i a O , su magnitud es $b/2$

$\vec{d}_{md/o}$ – vector de m_d a O , su magnitud es $b/2$

$$\omega_i r \hat{i} = \vec{v} \hat{i} + (\vec{\omega} \hat{j}) \times \left(-\frac{b}{2} \hat{k} \right) = \vec{v} \hat{i} - \frac{b}{2} \omega \hat{i} \quad (2.7)$$

$$\omega_d r \hat{i} = \vec{v} \hat{i} + (\vec{\omega} \hat{j}) \times \left(\frac{b}{2} \hat{k} \right) = \vec{v} \hat{i} + \frac{b}{2} \omega \hat{i} \quad (2.8)$$

Por lo tanto:

$$\omega_i r = v - \frac{b}{2} \omega \quad (2.9)$$

$$\omega_d r = v + \frac{b}{2} \omega \quad (2.10)$$

$$v = \omega_i r + \frac{b}{2} \omega$$

$$\omega_d r = \omega_i r + \frac{b}{2} \omega + \frac{b}{2} \omega$$

$$(\omega_d - \omega_i) r = b \omega$$

$$\omega = \frac{(\omega_d - \omega_i) r}{b} \quad (2.11)$$

$$v = \omega_i r + \frac{b}{2} \frac{(\omega_d - \omega_i) r}{b}$$

$$v = \omega_i r + \frac{r}{2} \omega_d - \frac{r}{2} \omega_i$$

$$v = \frac{r}{2} \omega_i + \frac{r}{2} \omega_d$$

$$v = \frac{(\omega_d + \omega_i) r}{2} \quad (2.12)$$

En forma matricial:

$$\begin{bmatrix} v \\ \omega \end{bmatrix} = \frac{r}{2} \begin{bmatrix} 1 & 1 \\ \frac{2}{b} & -\frac{2}{b} \end{bmatrix} \begin{bmatrix} \omega_d \\ \omega_i \end{bmatrix} \quad (2.13)$$

b es la distancia que separa a las dos ruedas traseras del robot móvil.

Las variables de control son las velocidades de las ruedas laterales ω_i (velocidad de giro de la rueda izquierda) y ω_d (velocidad de giro de la rueda derecha).

Si el radio de la rueda es r , las velocidades lineales de las ruedas son:

$$v_i = \omega_i r \quad (2.14)$$

$$v_d = \omega_d r \quad (2.15)$$

Si se especifica la velocidad lineal v y la velocidad angular ω del vehículo, es posible encontrar, a partir de (2.13), las velocidades de giro que hay que aplicar a las ruedas izquierda y derecha.

De (2.12):

$$\frac{2v}{r} = \omega_d + \omega_i$$

$$\omega_d = \frac{2v}{r} - \omega_i \quad (2.16)$$

Sustituyendo (2.16) en (2.11):

$$\omega = \frac{\left(\frac{2v}{r} - \omega_i - \omega_i\right) r}{b}$$

$$\frac{b\omega}{r} = \frac{2v}{r} - 2\omega_i$$

$$2\omega_i = \frac{2v - b\omega}{r}$$

$$\omega_i = \frac{v - \frac{b}{2}\omega}{r} \quad (2.17)$$

Sustituyendo (2.17) en (2.16):

$$\omega_d = \frac{2v}{r} - \frac{v - \frac{b}{2}\omega}{r}$$

$$\omega_d = \frac{v + \frac{b}{2}\omega}{r} \quad (2.18)$$

$$\begin{bmatrix} \omega_i \\ \omega_d \end{bmatrix} = \frac{1}{r} \begin{bmatrix} 1 & -\frac{b}{2} \\ 1 & \frac{b}{2} \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} \quad (2.19)$$

2.2 Modelo cinemático del robot móvil

Se considera que el vehículo se desplaza una longitud de arco Δs (del punto F al punto G), ver figura 2.2.

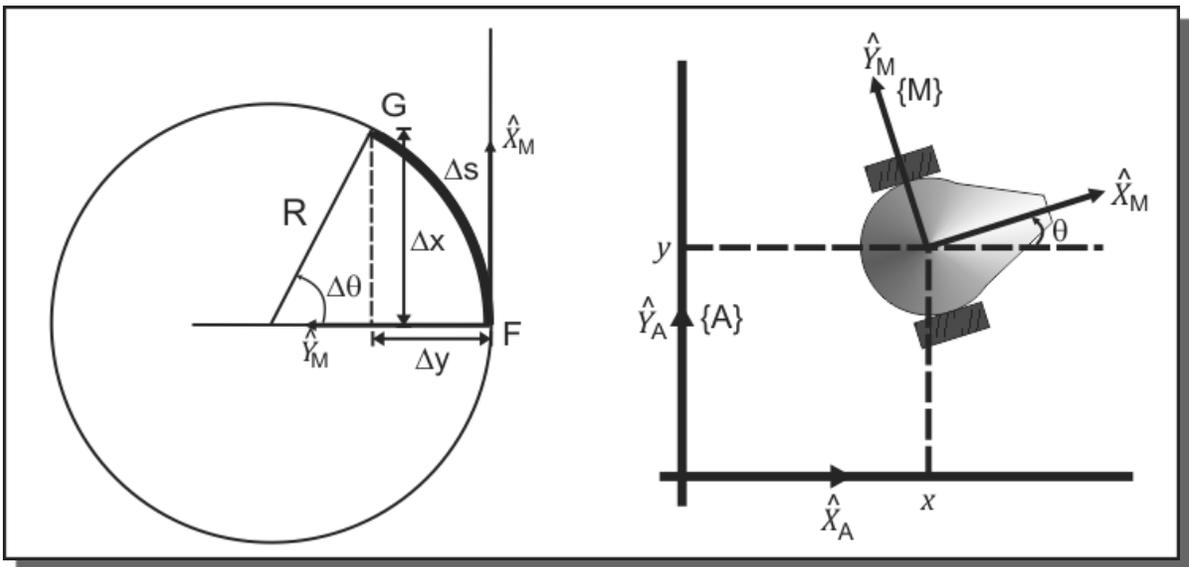


Fig. 2.2 Coordenadas del robot móvil; {A} es un marco de referencia absoluto, y {M} un marco de referencia móvil situado en el robot

Las velocidades lineal y angular del móvil son:

$$v = \frac{\Delta s}{\Delta t} \quad (2.20)$$

$$\omega = \frac{\Delta \theta}{\Delta t} \quad (2.21)$$

La longitud de arco se puede expresar de la siguiente forma:

$$\Delta s = R \Delta \theta \quad (2.22)$$

R – radio de giro ó radio de la circunferencia, que describe el punto de guiado (situado en este caso en medio de las dos ruedas traseras).

Se puede definir la curvatura γ como la inversa del radio de giro:

$$\gamma = \frac{1}{R} = \frac{\Delta \theta}{\Delta s} \quad (2.23)$$

Las ecuaciones de movimiento en $\{M\}$ son:

$${}^M(\Delta x) = R \operatorname{sen}(\Delta \theta) \quad (2.24)$$

$${}^M(\Delta y) = R - R \cos(\Delta \theta) = R[1 - \cos(\Delta \theta)] \quad (2.25)$$

Para reflejar estas ecuaciones al marco de referencia absoluto $\{A\}$, es necesario encontrar una matriz de rotación ${}^A R_M$:

$${}^A R_M = \begin{bmatrix} {}^A \hat{X}_M & {}^A \hat{Y}_M \end{bmatrix} = \begin{bmatrix} \hat{X}_M \cdot \hat{X}_A & \hat{Y}_M \cdot \hat{X}_A \\ \hat{X}_M \cdot \hat{Y}_A & \hat{Y}_M \cdot \hat{Y}_A \end{bmatrix} = \begin{bmatrix} \cos \theta & -\operatorname{sen} \theta \\ \operatorname{sen} \theta & \cos \theta \end{bmatrix} \quad (2.26)$$

$$\begin{bmatrix} {}^A(\Delta x) \\ {}^A(\Delta y) \end{bmatrix} = {}^A R_M \begin{bmatrix} {}^M(\Delta x) \\ {}^M(\Delta y) \end{bmatrix} = \begin{bmatrix} \cos \theta & -\operatorname{sen} \theta \\ \operatorname{sen} \theta & \cos \theta \end{bmatrix} \begin{bmatrix} {}^M(\Delta x) \\ {}^M(\Delta y) \end{bmatrix}$$

$$\Delta x = {}^A(\Delta x) = R \operatorname{sen}(\Delta \theta) \cos \theta - R[1 - \cos(\Delta \theta)] \operatorname{sen} \theta \quad (2.27)$$

$$\Delta y = {}^A(\Delta y) = R \operatorname{sen}(\Delta \theta) \operatorname{sen} \theta + R[1 - \cos(\Delta \theta)] \cos \theta \quad (2.28)$$

Si se considera que $\Delta\theta$ es muy pequeña, se pueden hacer las siguientes aproximaciones:

$$\cos(\Delta\theta) \cong 1$$

$$\text{sen}(\Delta\theta) \cong \Delta\theta$$

Quedando Δx y Δy como:

$$\Delta x = R\Delta\theta \cos\theta$$

$$\Delta y = R\Delta\theta \text{sen}\theta$$

Considerando la ecuación (2.22):

$$\Delta x = \Delta s \cos\theta \tag{2.29}$$

$$\Delta y = \Delta s \text{sen}\theta \tag{2.30}$$

Dividiendo (2.29) y (2.30) entre Δt , utilizando (2.20), y haciendo tender Δt a cero:

$$\dot{x} = v \cos\theta \tag{2.31}$$

$$\dot{y} = v \text{sen}\theta \tag{2.32}$$

Haciendo tender Δt a cero en (2.21):

$$\dot{\theta} = \omega \tag{2.33}$$

(2.31), (2.32) y (2.33) constituyen el modelo cinemático del robot móvil.

2.3 Modelo Jacobiano y restricción no holonómica.

El Jacobiano es una forma de mapear las velocidades del espacio articular al espacio cartesiano, se representa como $J(p)$.

Donde el vector $p = [p_1 \ p_2 \ \dots \ p_n]^T$ representa las variables necesarias para determinar completamente la posición y orientación del robot.

Las restricciones holonómicas son aquellas en las que no intervienen las velocidades, y tienen la forma $G_k(p, t) = 0$; $k = 1, 2, \dots$

Las restricciones no holonómicas dependen de las velocidades y, además, son no integrables, es decir, que no se pueden deducir por derivación total con respecto al tiempo de una restricción holonómica, estas tienen la forma $G_k(p, \dot{p}, t) = 0; k = 1, 2, \dots$

Si q es el vector de m variables de actuación, se puede definir el vector \dot{p} como:

$$\dot{p} = J(p)\dot{q} \quad (2.34)$$

Para el robot diferencial el vector p es $p = [x \ y \ \theta]$. Se cumplen además las relaciones (2.31) – (2.33), que en forma matricial quedan:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos\theta & 0 \\ \text{sen}\theta & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} \quad (2.35)$$

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos\theta \\ \text{sen}\theta \\ 0 \end{bmatrix} v + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \omega$$

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos\theta \\ \text{sen}\theta \\ 0 \end{bmatrix} \frac{(\omega_d + \omega_i)r}{2} + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \frac{(\omega_d - \omega_i)r}{b}$$

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \frac{r \cos\theta}{2} \\ r \text{sen}\theta \\ \frac{2}{r} \\ -\frac{1}{b} \end{bmatrix} \omega_i + \begin{bmatrix} \frac{r \cos\theta}{2} \\ r \text{sen}\theta \\ \frac{2}{r} \\ \frac{1}{b} \end{bmatrix} \omega_d$$

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \frac{r \cos\theta}{2} & \frac{r \cos\theta}{2} \\ r \text{sen}\theta & r \text{sen}\theta \\ \frac{2}{r} & \frac{2}{r} \\ -\frac{1}{b} & \frac{1}{b} \end{bmatrix} \begin{bmatrix} \omega_i \\ \omega_d \end{bmatrix} \quad (2.36)$$

La ecuación (2.36) corresponde al modelo cinemático directo del robot móvil. Donde se puede identificar al Jacobiano, $J(p)$, como:

$$J(p) = \begin{bmatrix} \frac{r \cos\theta}{2} & \frac{r \cos\theta}{2} \\ \frac{r \operatorname{sen}\theta}{2} & \frac{r \operatorname{sen}\theta}{2} \\ r & r \\ -\frac{r}{b} & \frac{r}{b} \end{bmatrix} \quad (2.37)$$

El modelo inverso involucra la inversa del Jacobiano. Para resolverlo de una forma general sería necesario invertir el Jacobiano. Cuando el Jacobiano no es cuadrado, como en este caso, se debe emplear la pseudoinversa. Multiplicando ambos miembros de (2.35) por J^T y despejando \dot{q} :

$$\begin{aligned} [J(p)]^T \dot{p} &= \{[J(p)]^T J(p)\} \dot{q} \\ \dot{q} &= \{[J(p)]^T J(p)\}^{-1} [J(p)]^T \dot{p} \end{aligned} \quad (2.38)$$

Utilizando el Jacobiano (2.37), se tiene para el robot móvil diferencial:

$$\begin{aligned} [\omega_d] &= \left\{ \begin{bmatrix} \frac{r \cos\theta}{2} & \frac{r \operatorname{sen}\theta}{2} & -\frac{r}{b} \\ \frac{r \cos\theta}{2} & \frac{r \operatorname{sen}\theta}{2} & \frac{r}{b} \\ r & r & r \\ -\frac{r}{b} & \frac{r}{b} & r \end{bmatrix} \begin{bmatrix} \frac{r \cos\theta}{2} & \frac{r \cos\theta}{2} \\ \frac{r \operatorname{sen}\theta}{2} & \frac{r \operatorname{sen}\theta}{2} \\ -\frac{r}{b} & \frac{r}{b} \end{bmatrix} \right\}^{-1} \begin{bmatrix} \frac{r \cos\theta}{2} & \frac{r \operatorname{sen}\theta}{2} & -\frac{r}{b} \\ \frac{r \cos\theta}{2} & \frac{r \operatorname{sen}\theta}{2} & \frac{r}{b} \\ r & r & r \\ -\frac{r}{b} & \frac{r}{b} & r \end{bmatrix} \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} \\ [\omega_d] &= \begin{bmatrix} \frac{r^2 \cos^2\theta}{4} + \frac{r^2 \operatorname{sen}^2\theta}{4} + \frac{r^2}{b^2} & \frac{r^2 \cos^2\theta}{4} + \frac{r^2 \operatorname{sen}^2\theta}{4} - \frac{r^2}{b^2} \\ \frac{r^2 \cos^2\theta}{4} + \frac{r^2 \operatorname{sen}^2\theta}{4} - \frac{r^2}{b^2} & \frac{r^2 \cos^2\theta}{4} + \frac{r^2 \operatorname{sen}^2\theta}{4} + \frac{r^2}{b^2} \\ r^2 & r^2 \\ -\frac{r^2}{b} & \frac{r^2}{b} \end{bmatrix}^{-1} \begin{bmatrix} \frac{r \cos\theta}{2} & \frac{r \operatorname{sen}\theta}{2} & -\frac{r}{b} \\ \frac{r \cos\theta}{2} & \frac{r \operatorname{sen}\theta}{2} & \frac{r}{b} \\ r & r & r \\ -\frac{r}{b} & \frac{r}{b} & r \end{bmatrix} \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} \\ [\omega_d] &= \begin{bmatrix} \frac{r^2}{4} + \frac{r^2}{b^2} & \frac{r^2}{4} - \frac{r^2}{b^2} \\ \frac{r^2}{4} - \frac{r^2}{b^2} & \frac{r^2}{4} + \frac{r^2}{b^2} \\ r^2 & r^2 \\ -\frac{r^2}{b} & \frac{r^2}{b} \end{bmatrix}^{-1} \begin{bmatrix} \frac{r \cos\theta}{2} & \frac{r \operatorname{sen}\theta}{2} & -\frac{r}{b} \\ \frac{r \cos\theta}{2} & \frac{r \operatorname{sen}\theta}{2} & \frac{r}{b} \\ r & r & r \\ -\frac{r}{b} & \frac{r}{b} & r \end{bmatrix} \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} \end{aligned}$$

$$[\omega_i] = \frac{b^2}{r^4} \begin{bmatrix} \frac{r^2}{4} + \frac{r^2}{b^2} & \frac{r^2}{b^2} - \frac{r^2}{4} \\ \frac{r^2}{b^2} - \frac{r^2}{4} & \frac{r^2}{4} + \frac{r^2}{b^2} \end{bmatrix} \begin{bmatrix} \frac{r \cos\theta}{2} & \frac{r \sen\theta}{2} & -\frac{r}{b} \\ \frac{r \cos\theta}{2} & \frac{r \sen\theta}{2} & \frac{r}{b} \end{bmatrix} \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix}$$

$$[\omega_i] = \begin{bmatrix} \frac{b^2 + 4}{4 r^2} & \frac{4 - b^2}{4 r^2} \\ \frac{4 - b^2}{4 r^2} & \frac{b^2 + 4}{4 r^2} \end{bmatrix} \begin{bmatrix} \frac{r \cos\theta}{2} & \frac{r \sen\theta}{2} & -\frac{r}{b} \\ \frac{r \cos\theta}{2} & \frac{r \sen\theta}{2} & \frac{r}{b} \end{bmatrix} \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix}$$

$$[\omega_i] = \begin{bmatrix} \frac{b^2 + 4}{8 r} \cos\theta + \frac{4 - b^2}{8 r} \cos\theta & \frac{b^2 + 4}{8 r} \sen\theta + \frac{4 - b^2}{8 r} \sen\theta & -\frac{b^2 + 4}{4 br} + \frac{4 - b^2}{4 br} \\ \frac{4 - b^2}{8 r} \cos\theta + \frac{b^2 + 4}{8 r} \cos\theta & \frac{4 - b^2}{8 r} \sen\theta + \frac{b^2 + 4}{8 r} \sen\theta & -\frac{4 - b^2}{4 br} + \frac{b^2 + 4}{4 br} \end{bmatrix} \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix}$$

$$[\omega_i] = \begin{bmatrix} \cos\theta \frac{b^2 + 4 + 4 - b^2}{8 r} & \sen\theta \frac{b^2 + 4 + 4 - b^2}{8 r} & \frac{-b^2 - 4 + 4 - b^2}{4 br} \\ \cos\theta \frac{4 - b^2 + b^2 + 4}{8 r} & \sen\theta \frac{4 - b^2 + b^2 + 4}{8 r} & \frac{-4 + b^2 + b^2 + 4}{4 br} \end{bmatrix} \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix}$$

$$[\omega_i] = \begin{bmatrix} \frac{1}{r} \cos\theta & \frac{1}{r} \sen\theta & -\frac{b}{2 r} \\ \frac{1}{r} \cos\theta & \frac{1}{r} \sen\theta & \frac{b}{2 r} \end{bmatrix} \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix}$$

Finalmente:

$$[\omega_i] = \frac{1}{r} \begin{bmatrix} \cos\theta & \sen\theta & -\frac{b}{2} \\ \cos\theta & \sen\theta & \frac{b}{2} \end{bmatrix} \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} \quad (2.39)$$

La ecuación (2.39) corresponde al modelo cinemático inverso del robot móvil. Como ya se observó anteriormente, ω_i y ω_d están en función de v y ω , así que también es posible encontrar un modelo inverso con $\dot{q} = [v \ \omega]^T$. Esto se hace más abajo, antes se deriva la restricción no holonómica.

Multiplicando ambos lados de (2.31) por $-\text{sen}\theta$:

$$-\dot{x} \text{sen}\theta = -v \text{sen}\theta \cos\theta \quad (2.40)$$

Multiplicando ambos lados de (2.32) por $\cos\theta$:

$$\dot{y} \cos\theta = v \text{sen}\theta \cos\theta \quad (2.41)$$

Sumando (2.40) y (2.41) se obtiene la restricción no holonómica:

$$\dot{y} \cos\theta - \dot{x} \text{sen}\theta = 0 \quad (2.42)$$

A partir de (2.42) se puede obtener una forma de calcular θ :

$$\dot{x} \text{sen}\theta = \dot{y} \cos\theta$$

$$\tan\theta = \frac{\dot{y}}{\dot{x}}$$

$$\theta = \tan^{-1}\left(\frac{\dot{y}}{\dot{x}}\right) \quad (2.43)$$

De la parte derecha de (2.35) se define la matriz $J(p)$:

$$J(p) = \begin{bmatrix} \cos\theta & 0 \\ \text{sen}\theta & 0 \\ 0 & 1 \end{bmatrix} \quad (2.44)$$

Deduciendo el modelo inverso a partir de (2.44) en (2.38):

$$\begin{bmatrix} v \\ \omega \end{bmatrix} = \left\{ \begin{bmatrix} \cos\theta & \text{sen}\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\theta & 0 \\ \text{sen}\theta & 0 \\ 0 & 1 \end{bmatrix} \right\}^{-1} \begin{bmatrix} \cos\theta & \text{sen}\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix}$$

$$\begin{bmatrix} v \\ \omega \end{bmatrix} = \begin{bmatrix} \cos^2\theta + \text{sen}^2\theta & 0 \\ 0 & 1 \end{bmatrix}^{-1} \begin{bmatrix} \cos\theta & \text{sen}\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix}$$

$$\begin{bmatrix} v \\ \omega \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}^{-1} \begin{bmatrix} \cos\theta & \text{sen}\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix}$$

$$\begin{bmatrix} v \\ \omega \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\theta & \text{sen}\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix}$$

$$\begin{bmatrix} v \\ \omega \end{bmatrix} = \begin{bmatrix} \cos\theta & \text{sen}\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix}$$

2.4 Modelo odométrico para el robot diferencial.

La posición (x, y) y orientación (θ) del robot móvil se pueden obtener a partir de la medición de los encoders (codificadores) ópticos conectados a los motores del robot.

El desplazamiento total del robot está dado por:

$$d_T = \frac{d_i + d_d}{2} \quad (2.45)$$

Dónde:

d_i – desplazamiento de la rueda izquierda

d_d – desplazamiento de la rueda derecha

Los segmentos de arco d_i y d_d forman parte de una circunferencia de radio $b + l$, cuyo perímetro está dado por:

$$P_d = 2\pi(b + l) \quad (2.46)$$

Dónde:

P_d – perímetro de la circunferencia trazada por la rueda derecha

b – distancia entre las ruedas del robot

l – radio de giro respecto al eje del robot

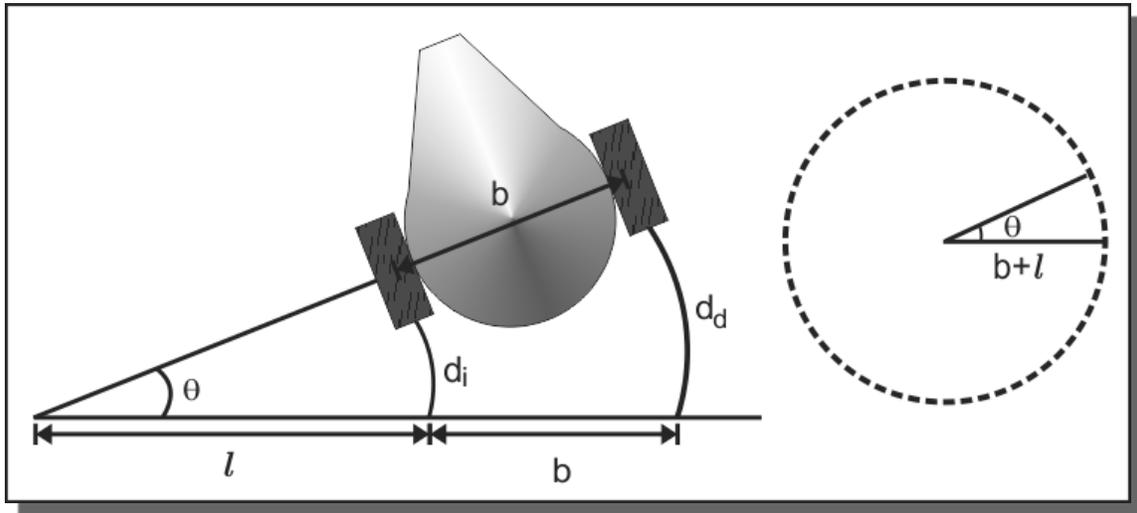


Fig. 2.3 Desplazamiento de las ruedas izquierda y derecha

A partir de la geometría del círculo, es válida la siguiente relación:

$$\frac{d_d}{P_d} = \frac{\theta}{2\pi} \quad (2.47)$$

Despejando θ :

$$\theta = \frac{2\pi d_d}{P_d} \quad (2.48)$$

Sustituyendo (2.46) en (2.48):

$$\theta = \frac{d_d}{b+l} \quad (2.49)$$

Así mismo, el arco d_i es una porción de la circunferencia de radio l con perímetro:

$$P_i = 2\pi l \quad (2.50)$$

Se cumple la siguiente relación:

$$\frac{d_i}{P_i} = \frac{\theta}{2\pi} \quad (2.51)$$

Sustituyendo (2.50) en (2.51) y despejando l :

$$(2.52)$$

$$l = \frac{d_i}{\theta}$$

Sustituyendo (2.52) en (2.49) y despejando θ :

$$\theta = \frac{d_d}{b + \frac{d_i}{\theta}}$$

$$\theta b + d_i = d_d$$

$$\theta = \frac{d_d - d_i}{b} \quad (2.53)$$

La expresión (2.53) proporciona la orientación del robot en función de los desplazamientos de las ruedas izquierda y derecha.

Si se expresan d_i y d_d en función de los pulsos del encoder:

$$d_i = 2\pi r \frac{N_i}{N_T} \quad (2.54)$$

$$d_d = 2\pi r \frac{N_d}{N_T} \quad (2.55)$$

Dónde:

r – radio de las ruedas (se asume que es el mismo para las dos ruedas)

N_i – número de pulsos contados en el encoder de la izquierda

N_d – número de pulsos contados en el encoder de la derecha

N_T – número total de pulsos contados en una revolución del encoder.

Sustituyendo en (2.53):

$$\theta = \frac{2\pi r \frac{N_d}{N_T} - 2\pi r \frac{N_i}{N_T}}{b}$$

$$\theta = \frac{2\pi r}{N_T b} (N_d - N_i)$$

Para la obtención de la posición se considera el desplazamiento del robot d_T , a partir de sus componentes en los planos x y y .

$$x = d_T \cos\theta \quad (2.57)$$

$$y = d_T \sen\theta \quad (2.58)$$

Utilizando (2.45), (2.54) y (2.55):

$$x = \frac{2\pi r \frac{N_i}{N_T} + 2\pi r \frac{N_d}{N_T}}{2} \cos\theta$$

$$x = \frac{\pi r}{N_T} (N_i + N_d) \cos\theta \quad (2.59)$$

$$y = \frac{2\pi r \frac{N_i}{N_T} + 2\pi r \frac{N_d}{N_T}}{2} \sen\theta$$

$$y = \frac{\pi r}{N_T} (N_i + N_d) \sen\theta \quad (2.60)$$

(2.59) y (2.60) permiten estimar la posición del robot a partir de las mediciones de los encoders.

CAPÍTULO 3

Esquemas de control para el seguimiento de trayectorias

3.1 Control usando el método de persecución pura

Este método de control está basado en conceptos geométricos. Tiene su origen en el problema de la persecución de un objetivo por parte de un misil [10], cuyo vector de velocidad se dirige siempre hacia la posición objetivo instantánea. El nombre de persecución pura viene de la idea de que el vehículo persigue un punto en el camino que se encuentra a una cierta distancia delante de él.

Se considera que en el intervalo de control la curvatura es constante, por lo que el robot móvil describe un arco de circunferencia, como se ilustra en la figura 3.1:

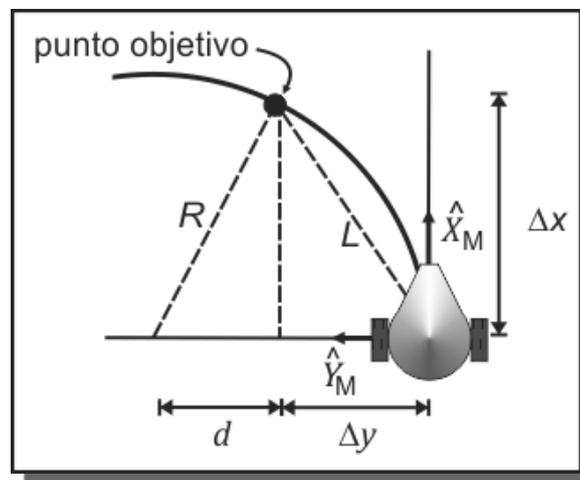


Fig. 3.1 Robot móvil y punto objetivo de la trayectoria

De la figura se pueden obtener las siguientes relaciones:

$$R = \Delta y + d \quad (3.1)$$

$$d^2 + (\Delta x)^2 = R^2 \quad (3.2)$$

$$L^2 = (\Delta x)^2 + (\Delta y)^2 \quad (3.3)$$

R – radio de curvatura del robot

L – distancia a la que se encuentra el punto objetivo

De (3.1):

$$d = R - \Delta y \quad (3.4)$$

Sustituyendo (3.4) en (3.2):

$$(R - \Delta y)^2 + (\Delta x)^2 = R^2$$

Despejando R :

$$R^2 - 2R\Delta y + (\Delta y)^2 + (\Delta x)^2 = R^2$$

$$R = \frac{(\Delta x)^2 + (\Delta y)^2}{2\Delta y}$$

Considerando (3.3):

$$R = \frac{L^2}{2\Delta y} \quad (3.5)$$

De la definición del radio de curvatura, ver ecuación (2.23):

$$\gamma = \frac{1}{R} = \frac{\Delta\theta}{\Delta s}$$

Considerando (2.20) y (2.21):

$$\gamma = \frac{1}{R} = \frac{\omega}{v}$$

Por lo tanto, al robot móvil se le debe suministrar la siguiente curvatura:

$$\gamma = \frac{2\Delta y}{L^2} \quad (3.6)$$

La ecuación (3.6) corresponde a la ley de control del método de persecución pura, se trata de una ley de control proporcional al error Δy con respecto al punto objetivo. La ganancia puede considerarse como $2/L^2$.

La razón de la distancia L es que los robots móviles no holonómicos no pueden corregir errores con respecto al punto más cercano del camino. Si se tiene una

distancia L más pequeña se pueden reducir los errores de seguimiento, pero se incrementan los comandos de direccionamiento, por lo que el movimiento del robot se puede volver inestable.

Para determinar Δy es necesario tomar en cuenta la figura 3.2.

Si el robot está en el punto (x, y) del marco de referencia absoluto, y el punto objetivo (x_{obj}, y_{obj}) está expresado en este mismo marco de referencia, se tiene lo siguiente:

$$\Delta y = -(x_{obj} - x)\text{sen}\theta + (y_{obj} - y)\text{cos}\theta \quad (3.7)$$

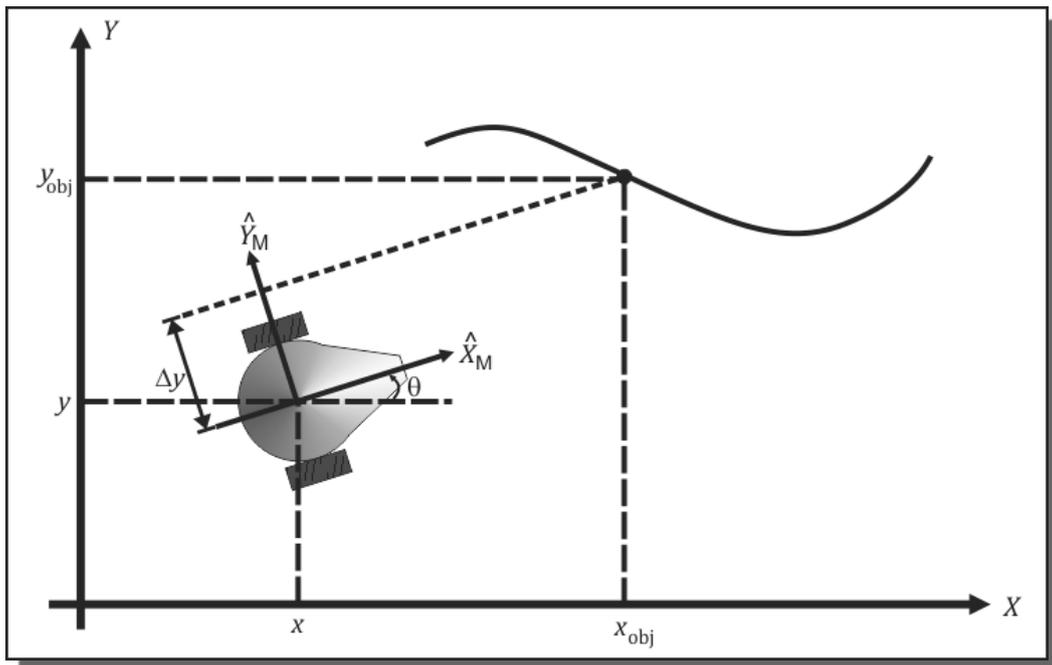


Fig. 3.2 Coordenadas del robot móvil y del punto objetivo

Entre las ventajas de este método de control se encuentran: una fácil sintonización de la distancia L , simplicidad computacional y la ausencia de términos derivativos.

3.2 Control no lineal usando seguimiento de postura

La postura del robot móvil se define como la configuración (x, y, θ) en un instante de tiempo determinado; en el enfoque de control de este apartado el robot sigue una postura de referencia (x_r, y_r, θ_r) dada por el modelo de un robot móvil diferencial de referencia (figura 3.3).

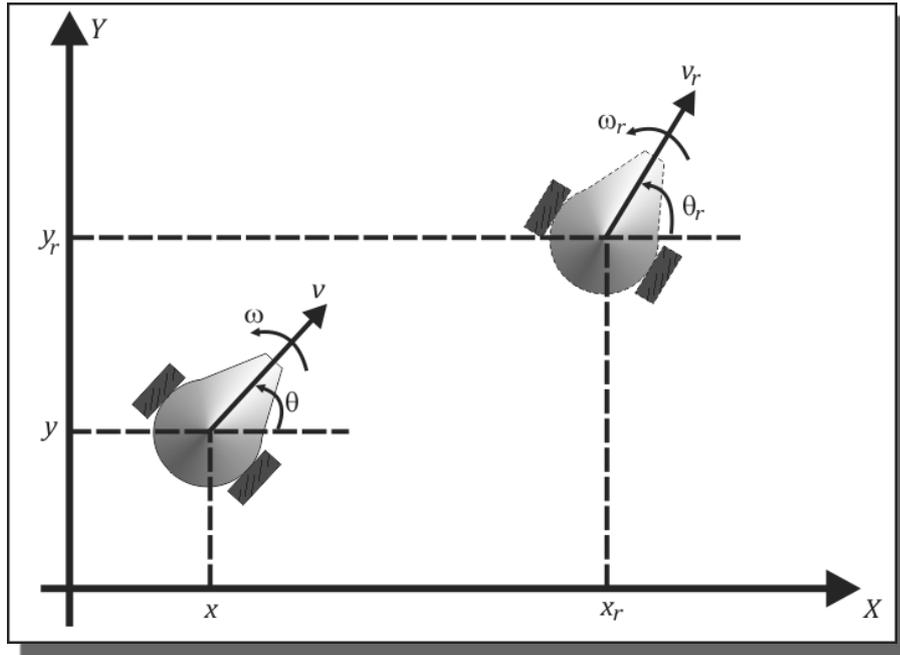


Fig. 3.3 Seguimiento de postura

En el capítulo anterior se obtuvo el modelo cinemático del robot móvil diferencial, el cual se repite aquí:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos\theta & 0 \\ \sin\theta & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} \quad (3.8)$$

Mientras que la trayectoria que se pretende seguir viene dada por el modelo de referencia siguiente:

$$\begin{bmatrix} \dot{x}_r \\ \dot{y}_r \\ \dot{\theta}_r \end{bmatrix} = \begin{bmatrix} \cos\theta_r & 0 \\ \sin\theta_r & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v_r \\ \omega_r \end{bmatrix} \quad (3.9)$$

Se supone que v_r y ω_r , así como sus derivadas, están acotadas y que se cumplen las condiciones siguientes:

$$\lim_{t \rightarrow \infty} v_r(t) \neq 0 \quad (3.10)$$

$$\lim_{t \rightarrow \infty} \omega_r(t) \neq 0$$

Estas condiciones implican que se quiere seguir una trayectoria variante en el tiempo, caso contrario al de alcanzar una postura fija (regulación).

Para cuantificar el objetivo de control de seguimiento se definen las siguientes variables:

$$\begin{aligned}\tilde{x} &= x - x_r \\ \tilde{y} &= y - y_r \\ \tilde{\theta} &= \theta - \theta_r\end{aligned}\tag{3.11}$$

Para facilitar el manejo del sistema en lazo cerrado se define la siguiente transformación [3]:

$$\begin{bmatrix} e_1 \\ e_2 \\ e_3 \end{bmatrix} = \begin{bmatrix} \cos\theta & \text{sen}\theta & 0 \\ -\text{sen}\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \tilde{x} \\ \tilde{y} \\ \tilde{\theta} \end{bmatrix}\tag{3.12}$$

El sistema en lazo abierto se obtiene derivando (3.12) respecto al tiempo. Utilizando (3.8), (3.9), (3.12) y las identidades trigonométricas para el seno y coseno de la resta de ángulos, se tiene para e_1 :

$$\begin{aligned}e_1 &= \tilde{x}\cos\theta + \tilde{y}\text{sen}\theta \\ e_1 &= (x - x_r)\cos\theta + (y - y_r)\text{sen}\theta \\ \dot{e}_1 &= (\dot{x} - \dot{x}_r)\cos\theta - (x - x_r)\dot{\theta}\text{sen}\theta + (\dot{y} - \dot{y}_r)\text{sen}\theta + (y - y_r)\dot{\theta}\cos\theta \\ \dot{e}_1 &= v + \omega e_2 - v_r \cos e_3\end{aligned}$$

Para e_2 :

$$\begin{aligned}e_2 &= -\tilde{x}\text{sen}\theta + \tilde{y}\cos\theta \\ e_2 &= -(x - x_r)\text{sen}\theta + (y - y_r)\cos\theta \\ \dot{e}_2 &= -(\dot{x} - \dot{x}_r)\text{sen}\theta - (x - x_r)\dot{\theta}\cos\theta + (\dot{y} - \dot{y}_r)\cos\theta + (y - y_r)\dot{\theta}\text{sen}\theta \\ \dot{e}_2 &= -\omega e_1 + v_r \text{sen} e_3\end{aligned}$$

Mientras que para e_3 :

$$\begin{aligned}e_3 &= \tilde{\theta} \\ e_3 &= \theta - \theta_r \\ \dot{e}_3 &= \dot{\theta} - \dot{\theta}_r \\ \dot{e}_3 &= \omega - \omega_r\end{aligned}$$

El sistema de error en lazo abierto queda:

$$\begin{bmatrix} \dot{e}_1 \\ \dot{e}_2 \\ \dot{e}_3 \end{bmatrix} = \begin{bmatrix} v + \omega e_2 - v_r \cos e_3 \\ -\omega e_1 + v_r \sin e_3 \\ \omega - \omega_r \end{bmatrix} \quad (3.13)$$

Con base en el objetivo de control del seguimiento de la postura de referencia, se propone el controlador diferenciable variante en el tiempo siguiente (ver [4]):

$$\begin{bmatrix} v \\ \omega \end{bmatrix} = \begin{bmatrix} -k_1 e_1 + v_r \cos e_3 \\ -v_r \frac{\sin e_3}{e_3} e_2 - k_2 e_3 + \omega_r \end{bmatrix} \quad (3.14)$$

Donde k_1 y k_2 son las ganancias de control, reales y positivas, y se cumple lo siguiente:

$$\lim_{e_3 \rightarrow 0} \frac{\sin e_3}{e_3} = 1$$

Sustituyendo (3.14) en (3.13) se obtiene el sistema en lazo cerrado del error:

$$\begin{bmatrix} \dot{e}_1 \\ \dot{e}_2 \\ \dot{e}_3 \end{bmatrix} = \begin{bmatrix} \omega e_2 - k_1 e_1 \\ -\omega e_1 + v_r \sin e_3 \\ -v_r \frac{\sin e_3}{e_3} e_2 - k_2 e_3 \end{bmatrix} \quad (3.15)$$

3.3 Control lineal usando seguimiento de postura

Para este esquema de control se parte del sistema (3.13), considerando el siguiente cambio de variables:

$$v - v_r \cos e_3 = u_1$$

$$\omega - \omega_r = u_2$$

El sistema (3.13) queda:

$$\begin{bmatrix} \dot{e}_1 \\ \dot{e}_2 \\ \dot{e}_3 \end{bmatrix} = \begin{bmatrix} \omega e_2 + u_1 \\ -\omega e_1 + v_r \sin e_3 \\ u_2 \end{bmatrix}$$

$$\begin{bmatrix} \dot{e}_1 \\ \dot{e}_2 \\ \dot{e}_3 \end{bmatrix} = \begin{bmatrix} 0 & \omega & 0 \\ -\omega & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} e_1 \\ e_2 \\ e_3 \end{bmatrix} + \begin{bmatrix} 0 \\ \sin e_3 \\ 0 \end{bmatrix} v_r + \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \quad (3.16)$$

En el punto de equilibrio se cumple lo siguiente:

$$v = v_r, \quad \omega = \omega_r, \quad sene_3 \cong e_3$$

Por lo tanto el sistema linealizado es:

$$\begin{bmatrix} \dot{e}_1 \\ \dot{e}_2 \\ \dot{e}_3 \end{bmatrix} = \begin{bmatrix} 0 & \omega_r & 0 \\ -\omega_r & 0 & v_r \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} e_1 \\ e_2 \\ e_3 \end{bmatrix} + \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \quad (3.17)$$

Para analizar la controlabilidad del sistema se consideran las matrices:

$$A = \begin{bmatrix} 0 & \omega_r & 0 \\ -\omega_r & 0 & v_r \\ 0 & 0 & 0 \end{bmatrix}$$

$$B = \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix}$$

Y de acuerdo a [8], se consideran los siguientes cálculos:

$$AB = \begin{bmatrix} 0 & \omega_r & 0 \\ -\omega_r & 0 & v_r \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ -\omega_r & v_r \\ 0 & 0 \end{bmatrix}$$

$$A^2 = \begin{bmatrix} 0 & \omega_r & 0 \\ -\omega_r & 0 & v_r \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 & \omega_r & 0 \\ -\omega_r & 0 & v_r \\ 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} -\omega_r^2 & 0 & v_r\omega_r \\ 0 & -\omega_r^2 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$A^2B = \begin{bmatrix} -\omega_r^2 & 0 & v_r\omega_r \\ 0 & -\omega_r^2 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} -\omega_r^2 & v_r\omega_r \\ 0 & 0 \\ 0 & 0 \end{bmatrix}$$

La matriz de controlabilidad es:

$$C = [B \quad AB \quad A^2B]$$

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 & -\omega_r^2 & v_r\omega_r \\ 0 & 0 & -\omega_r & v_r & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (3.18)$$

Si la matriz \mathcal{C} es de rango completo el sistema es controlable. Por lo tanto, si $v_r = \omega_r = 0$ se pierde controlabilidad del modelo linealizado.

Si ahora se considera la realimentación lineal siguiente (ver [3]):

$$u_1 = -k_1 e_1 \quad (3.19)$$

$$u_2 = -k_2 \text{sign}(v_r) e_2 - k_3 e_3 \quad (3.20)$$

Donde:

$$\text{sign}(v_r) = \begin{cases} -1, & v_r < 0 \\ 1, & v_r > 0 \end{cases}$$

Esta ley se escoge para hacer que los polos del sistema en lazo cerrado sean iguales a las raíces de la ecuación polinomial característica:

$$(s + 2\xi a)(s^2 + 2\xi a s + a^2) = 0 \quad (3.21)$$

El sistema en lazo cerrado resultante es:

$$\begin{aligned} \begin{bmatrix} \dot{e}_1 \\ \dot{e}_2 \\ \dot{e}_3 \end{bmatrix} &= \begin{bmatrix} 0 & \omega_r & 0 \\ -\omega_r & 0 & v_r \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} e_1 \\ e_2 \\ e_3 \end{bmatrix} + \begin{bmatrix} -k_1 e_1 \\ 0 \\ -k_2 \text{sign}(v_r) e_2 - k_3 e_3 \end{bmatrix} \\ \begin{bmatrix} \dot{e}_1 \\ \dot{e}_2 \\ \dot{e}_3 \end{bmatrix} &= \begin{bmatrix} -k_1 & \omega_r & 0 \\ -\omega_r & 0 & v_r \\ 0 & -k_2 \text{sign}(v_r) & -k_3 \end{bmatrix} \begin{bmatrix} e_1 \\ e_2 \\ e_3 \end{bmatrix} \end{aligned} \quad (3.22)$$

Encontrando el polinomio característico $|SI - \bar{A}| = 0$:

$$\begin{aligned} SI - \bar{A} &= \begin{bmatrix} s & 0 & 0 \\ 0 & s & 0 \\ 0 & 0 & s \end{bmatrix} - \begin{bmatrix} -k_1 & \omega_r & 0 \\ -\omega_r & 0 & v_r \\ 0 & -k_2 \text{sign}(v_r) & -k_3 \end{bmatrix} \\ SI - \bar{A} &= \begin{bmatrix} s + k_1 & -\omega_r & 0 \\ \omega_r & s & -v_r \\ 0 & k_2 \text{sign}(v_r) & s + k_3 \end{bmatrix} \end{aligned}$$

$$|SI - \bar{A}| = (s + k_1)(s)(s + k_3) + \omega_r^2(s + k_3) + v_r(s + k_1)k_2 \text{sign}(v_r)$$

El polinomio característico es:

$$(s + k_1)(s)(s + k_3) + \omega_r^2(s + k_3) + v_r(s + k_1)k_2 \text{sign}(v_r) = 0 \quad (3.23)$$

Igualando (3.21) y (3.23) se obtienen las ganancias k_1 , k_2 y k_3 siguientes:

$$\begin{aligned} k_1 &= 2\xi a \\ k_2 &= \frac{a^2 - \omega_r^2}{|v_r|} \\ k_3 &= 2\xi a \end{aligned} \quad (3.24)$$

Con una estrategia de asignación de polos fija (a y ξ constantes), la ganancia k_2 se incrementa sin límite cuando ω_r tiende a cero.

La regularización del controlador es posible si se hace que los polos en lazo cerrado dependan de los valores de v_r y ω_r . Este procedimiento se conoce como escalamiento de velocidad.

Si se escoge $a = (\omega_r^2 + bv_r^2)^{\frac{1}{2}}$ con $b > 0$, las ganancias del control son:

$$\begin{aligned} k_1 &= 2\xi(\omega_r^2 + bv_r^2)^{\frac{1}{2}} \\ k_2 &= b|v_r| \\ k_3 &= 2\xi(\omega_r^2 + bv_r^2)^{\frac{1}{2}} \end{aligned} \quad (3.25)$$

El control resultante está ahora definido para todos los valores de v_r y ω_r . Los valores $v_r = \omega_r = 0$ para los cuales el sistema es no controlable simplemente originan cero acción de control.

3.4 Control usando modos deslizantes

El ángulo de orientación del robot se mide de forma indicada en la figura 3.4:

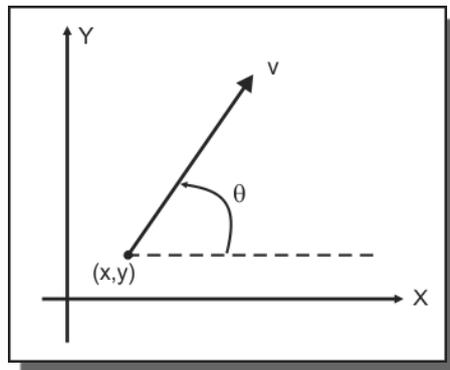


Fig. 3.4 Forma de medir el ángulo de orientación del robot

Partiendo de esto se definen las dos primeras ecuaciones de (3.26):

$$\begin{aligned}\dot{x} &= v \cos\theta \\ \dot{y} &= v \operatorname{sen}\theta \\ \dot{v} &= \frac{F}{m} \\ \dot{\theta} &= \frac{M}{J}\end{aligned}\quad (3.26)$$

Las últimas dos ecuaciones de (3.26) se originan de la segunda ley de Newton, siendo m la masa del vehículo, J su momento de inercia, F la fuerza y M el torque. Con estas dos ecuaciones se realiza el control del vehículo, mediante el cambio de aceleraciones lineales y angulares.

Se pretende que el robot se mueva hacia un punto objetivo con coordenadas $(x_o(t), y_o(t))$ sobre una curva en el plano $X - Y$, y que continúe su movimiento sobre dicha curva (figura 3.5).

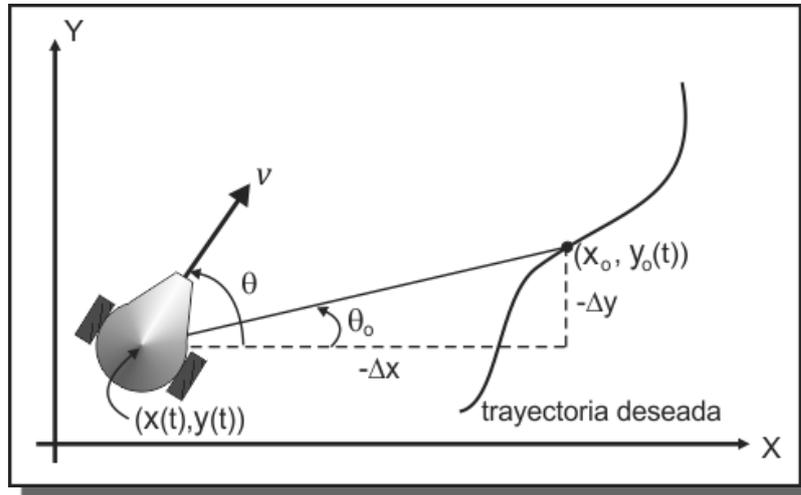


Fig. 3.5 Coordenadas del robot y del punto objetivo

Se define el error en la posición del robot móvil respecto al punto objetivo de la siguiente forma:

$$\begin{aligned}\Delta x(t) &= x(t) - x_o(t) \\ \Delta y(t) &= y(t) - y_o(t) \\ \Delta\theta(t) &= \theta(t) - \theta_o(t)\end{aligned}\quad (3.27)$$

El ángulo $\theta_o(t)$ se define como sigue:

$$\theta_o(t) \begin{cases} \operatorname{Arctan}\left(\frac{-\Delta y}{-\Delta x}\right) & \text{para } \Delta x^2 + \Delta y^2 > \varepsilon \\ \operatorname{Arctan}\left(\frac{-\dot{y}_o}{-\dot{x}_o}\right) & \text{para } \Delta x^2 + \Delta y^2 \leq \varepsilon \text{ y } \dot{x}_o^2 + \dot{y}_o^2 \neq 0 \\ \theta_{o_{ref}} & \text{para } \Delta x^2 + \Delta y^2 \leq \varepsilon \text{ y } \dot{x}_o^2 + \dot{y}_o^2 = 0 \end{cases}\quad (3.28)$$

El ángulo θ_{o_ref} define la orientación del robot en caso de que la posición objetivo (x_o, y_o) sea estacionaria, y ε especifica una vecindad del punto (x_o, y_o) .

Se define la trayectoria deseada en el espacio de las variables de estado $x, \dot{x}, y, \dot{y}, \theta$ y $\dot{\theta}$ por medio de las ecuaciones

$$\begin{aligned} \dot{x} + k_1 \Delta x &= 0 \\ \dot{y} + k_1 \Delta y &= 0 \\ \dot{\theta} + k_2 \Delta \theta &= 0 \end{aligned} \quad (3.29)$$

En el caso de una posición objetivo estacionaria, el cumplimiento de las condiciones (3.29) asegura un decrecimiento exponencial de los errores de posición del robot. Cuando la posición objetivo se está moviendo, los errores Δx , Δy y $\Delta \theta$ se descartan con una incertidumbre dinámica, que se define por medio de las derivadas $\dot{x}_o(t)$, $\dot{y}_o(t)$ y la movilidad del vehículo.

Desafortunadamente, no es posible asegurar cumplimiento simultaneo de (3.29) por medio de los parámetros F y M , que no influyen directamente en $\dot{x}_o(t)$, $\dot{y}_o(t)$ y $\dot{\theta}_o(t)$.

Sin embargo, se puede establecer un movimiento forzado del sistema en el modo deslizante (ver [5], [12], [14] y la sección 6.3 de este trabajo), a lo largo de las trayectorias definidas por estas ecuaciones.

Se define la desviación de la trayectoria (3.29) por medio de las funciones:

$$\begin{aligned} S_1(x, \dot{x}) &= \dot{x} + k_1 \Delta x \\ S_2(y, \dot{y}) &= \dot{y} + k_1 \Delta y \\ S_3(\theta, \dot{\theta}) &= \dot{\theta} + k_2 \Delta \theta \\ S(x, \dot{x}, y, \dot{y}, \theta, \dot{\theta}) &= \frac{1}{2} (S_1^2 + S_2^2 + S_3^2) \end{aligned} \quad (3.30)$$

Los parámetros de control F y M se eligen de la condición

$$\frac{dS}{dt} \leq 0 \quad (3.31)$$

El signo igual en (3.31) ocurre sólo si $S = 0$, lo que asegura que se satisfagan las condiciones (3.29). Si $S \neq 0$, la condición $\dot{S} < 0$ fuerza el movimiento del sistema, en la vecindad de la trayectoria definida por las ecuaciones (3.29).

Calculando

$$\begin{aligned} \dot{S} &= S_1 \dot{S}_1 + S_2 \dot{S}_2 + S_3 \dot{S}_3 \\ \dot{S} &= S_1 (\ddot{x} + k_1 \dot{\Delta x}) + S_2 (\ddot{y} + k_1 \dot{\Delta y}) + S_3 (\ddot{\theta} + k_2 \dot{\Delta \theta}) \end{aligned}$$

$$\dot{S} = S_1(-v\text{sen}\theta + \dot{v}\text{cos}\theta + k_1\dot{\Delta x}) + S_2(v\text{cos}\theta + \dot{v}\text{sen}\theta + k_1\dot{\Delta y}) + S_3\left(\frac{M}{J} + k_2\dot{\Delta\theta}\right)$$

$$\begin{aligned} \dot{S} = S_1\left(\frac{F}{m}\right)\text{cos}\theta + S_2\left(\frac{F}{m}\right)\text{sen}\theta + S_3\left(\frac{M}{J}\right) - S_1v\text{sen}\theta + S_1k_1\dot{\Delta x} + S_2v\text{cos}\theta \\ + S_2k_1\dot{\Delta y} + S_3k_2\dot{\Delta\theta} \end{aligned}$$

$$\dot{S} = S_1\left(\frac{F}{m}\right)\text{cos}\theta + S_2\left(\frac{F}{m}\right)\text{sen}\theta + S_3\left(\frac{M}{J}\right) + \dots \quad (3.32)$$

Definiendo la variable u de la siguiente forma:

$$u = -(S_1\text{cos}\theta + S_2\text{sen}\theta) \quad (3.33)$$

En ausencia de restricciones en los parámetros de control F y M , y tomando en cuenta (3.31) y $v \geq 0$, se define el parámetro de control F/m como:

$$\frac{F}{m} = \begin{cases} c_1 u & \text{para } v > 0 \text{ ó } u \geq 0 \\ 0 & \text{para } v = 0 \text{ y } u < 0 \end{cases} \quad (3.34)$$

c_1 es el factor de ganancia en el canal de control de velocidad.

De forma análoga

$$\frac{M}{J} = -c_2 S_3 \quad (3.35)$$

Cuando c_1 y c_2 son suficientemente grandes es posible satisfacer (3.13), si u y S_3 no son exactamente iguales a cero. Las igualdades $u \equiv 0$ y $S_3 \equiv 0$ son equivalentes al cumplimiento de las condiciones (3.29), es decir, al movimiento exacto del sistema a lo largo de la trayectoria deseada.

Entre las ventajas de este tipo de control se encuentra la robustez que tiene ante incertidumbres en el sistema. Dichas incertidumbres se pueden deber a parámetros no modelados del sistema ó a ruido de los sensores involucrados. En teoría, este controlador debería tener un mejor desempeño que los considerados en los apartados anteriores.

CAPÍTULO 4

Simulaciones numéricas

La simulación de la dinámica de sistemas físicos por medios computacionales es una herramienta que permite entre otras cosas, evaluar el desempeño de múltiples esquemas de control aplicados a una planta, bajo diferentes circunstancias; se pueden variar parámetros del sistema, condiciones iniciales, ganancias de control, etc. Las simulaciones facilitan la implementación y permiten, entre otras cosas, generar valores seguros para los actuadores.

Un programa de cómputo de uso extendido y que cuenta con utilidades diversas para la simulación de sistemas es MATLAB / Simulink; teniéndose además la opción de poder realizar animaciones con su módulo de realidad virtual.

En este capítulo se presentan los resultados de la simulación de los esquemas de control mostrados en el capítulo anterior.

4.1 Simulación usando el esquema de persecución pura

La figura 4.1 muestra el programa a bloques en Simulink. Se incluye el modelo cinemático del robot móvil diferencial encontrado en la sección 2.2, el esquema de control y un bloque que genera una animación en el módulo de realidad virtual, que se describirá en 4.5.

Mientras que la figura 4.2 corresponde al bloque del algoritmo de control por persecución pura, de acuerdo a las ecuaciones del apartado 3.1.

La trayectoria propuesta es una línea recta que va del punto $(-0.7, -0.7)$ al punto $(0.7, 0.7)$; se maneja una velocidad $v = 0.1$ m/s para el vehículo y se varía el parámetro L para ver cómo afecta al seguimiento de la trayectoria.

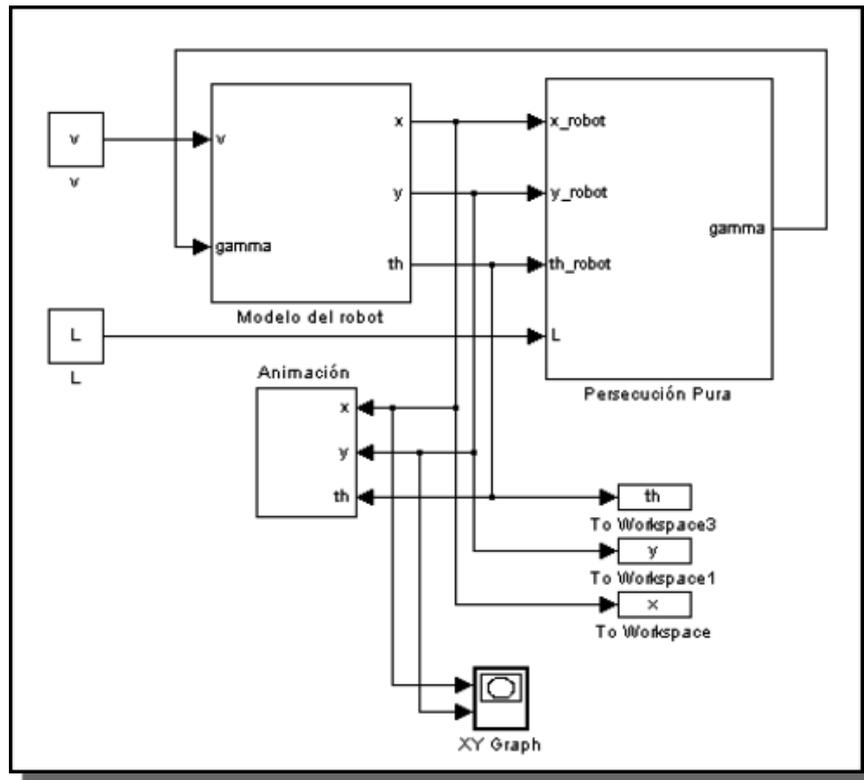


Fig. 4.1 Programa en Simulink del sistema en lazo cerrado

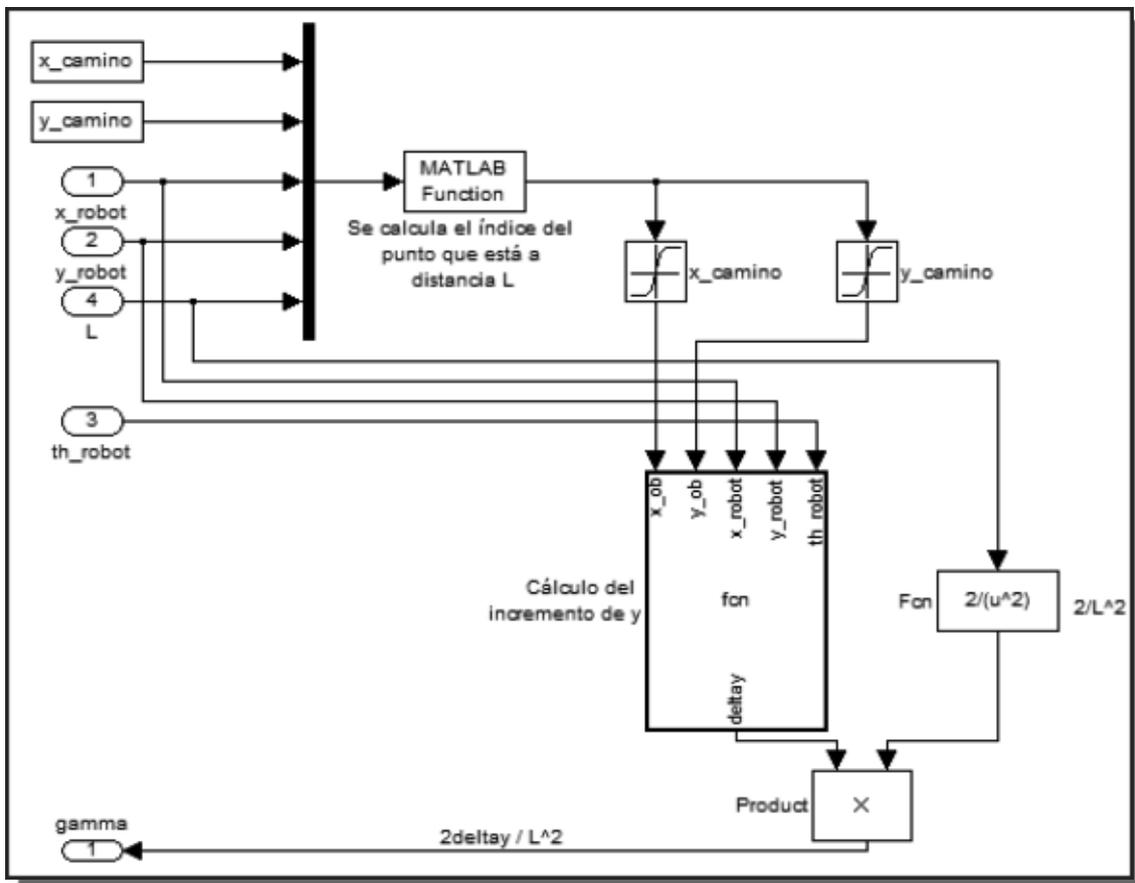


Fig. 4.2 Programa en Simulink del control por persecución pura

La trayectoria se guarda en el espacio de trabajo (workspace) de MATLAB, con los siguientes comandos:

```
x0 = -0.7:0.01:0.7;
```

```
y0 = x0;
```

Los valores utilizados de L fueron de 0.5, 0.05 y 0.005.

El robot móvil se inicia en el punto (-0.8, -0.6), con una orientación de $-\pi/2$.

El tiempo de la simulación es de 20 segundos.

Los resultados del seguimiento de la trayectoria en el plano XY se muestran para los tres valores de L propuestos (figuras 4.3 – 4.5).

Trayectoria en el plano XY para L = 0.5:

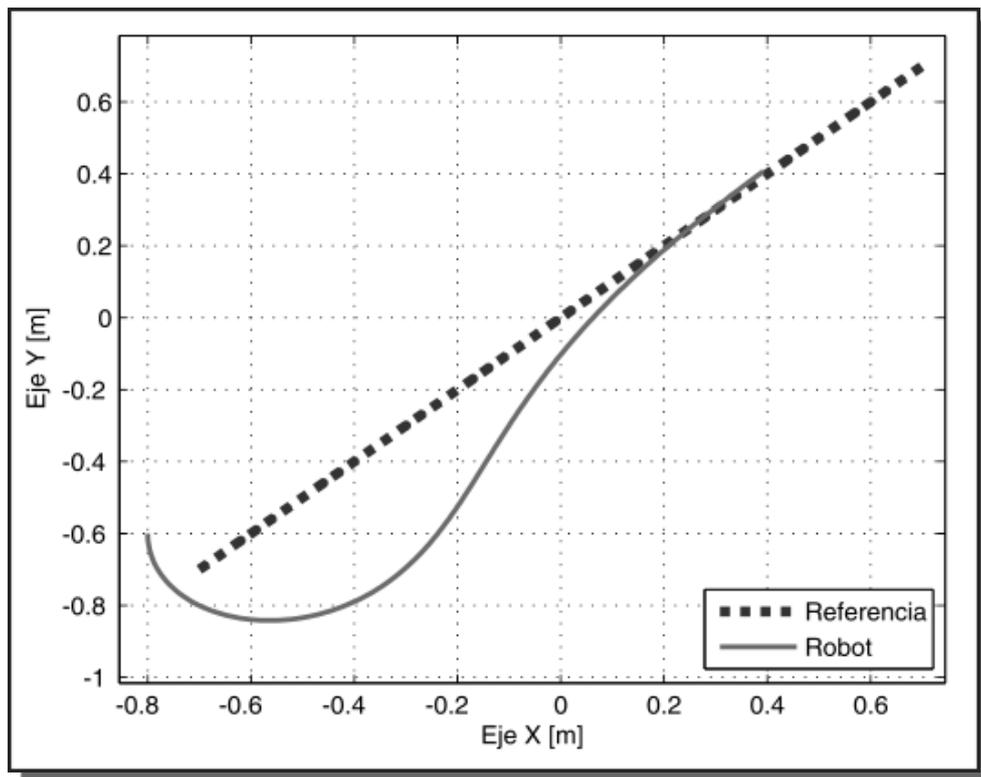


Fig. 4.3 Trayectoria en el plano XY para L = 0.5

Trayectoria en el plano XY para $L = 0.05$:

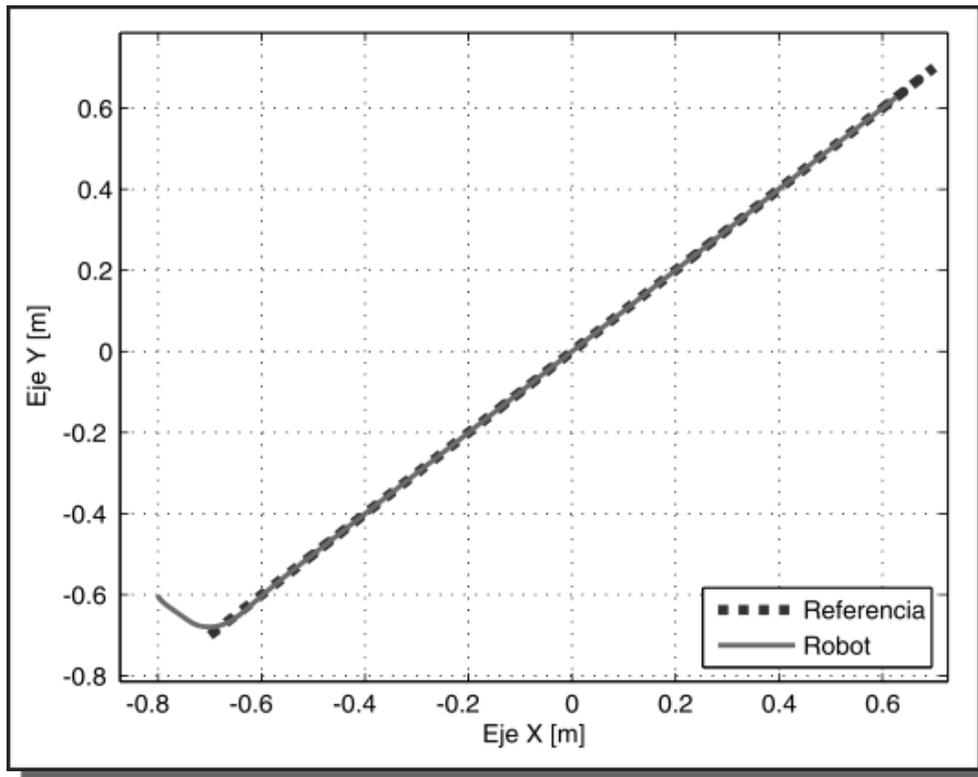


Fig. 4.4 Trayectoria en el plano XY para $L = 0.05$

Trayectoria en el plano XY para $L = 0.005$:

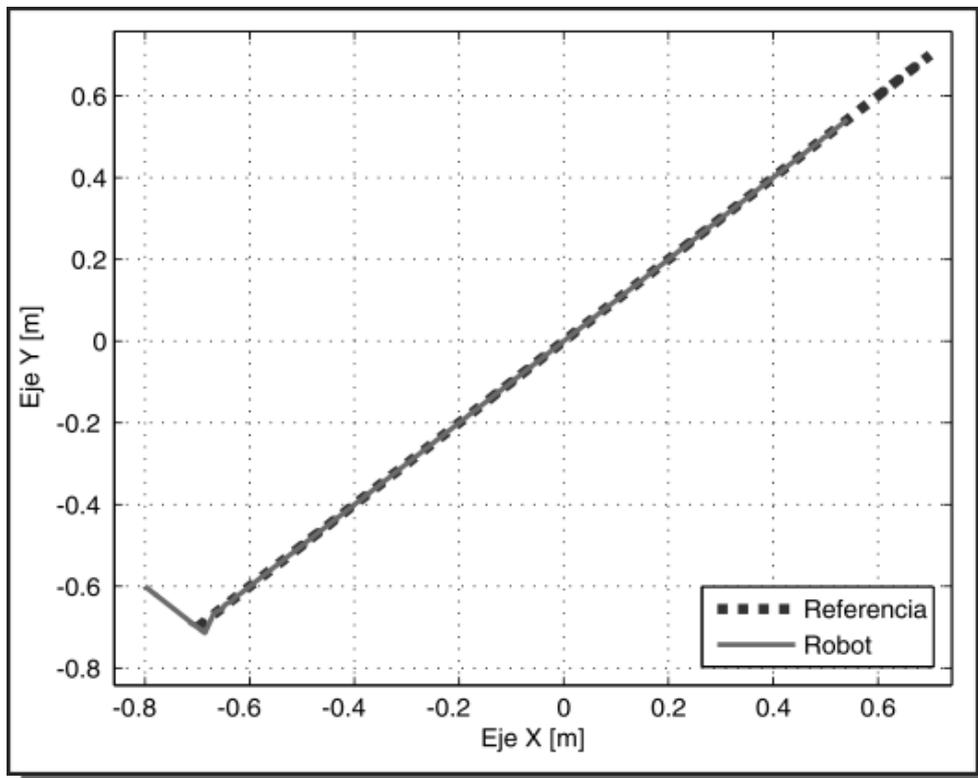


Fig. 4.5 Trayectoria en el plano XY para $L = 0.005$

4.2 Simulación usando control no lineal

En este caso se simula el esquema de control de la sección 3.2 para una trayectoria circular. Las velocidades para generar este círculo, ingresadas al modelo de referencia, son $v_r = 0.05 \text{ m/s}$ y $\omega_r = 0.1 \text{ rad/s}$. La figura 4.6 muestra el programa de simulación.

El cambio de variables de la ecuación (3.12) se hace en el bloque “cambio de variables” de la misma figura:

```
e1 = (x - xr)*cos(th) + (y - yr)*sin(th);
e2 = -(x - xr)*sin(th) + (y - yr)*cos(th);
e3 = th - thr;
```

El bloque “controles” (figura 4.6) contiene el siguiente código:

```
function [v,w] = fcn(e1,e2,e3,vr,wr,k1,k2)

v = -k1*e1 + vr*cos(e3);

if e3 == 0
    y = 1;
else
    y = sin(e3)/e3;
end

w = -(vr)*(y)*e2 - k2*e3 + wr;
```

Las ganancias utilizadas para este código fueron:

$$k_1 = 16$$

$$k_2 = 20$$

El robot móvil a ser controlado arranca en $(-0.1, -0.3)$, con una orientación de $-\pi/4$.

El tiempo de simulación es de 60 segundos.

Los resultados de la simulación se muestran en las figuras 4.7 – 4.12.

En dichas figuras se puede observar un muy buen seguimiento de la trayectoria, pero esto a costa de la generación de señales de control muy grandes, que llevan a los errores de seguimiento hacia cero en un intervalo muy corto de tiempo.

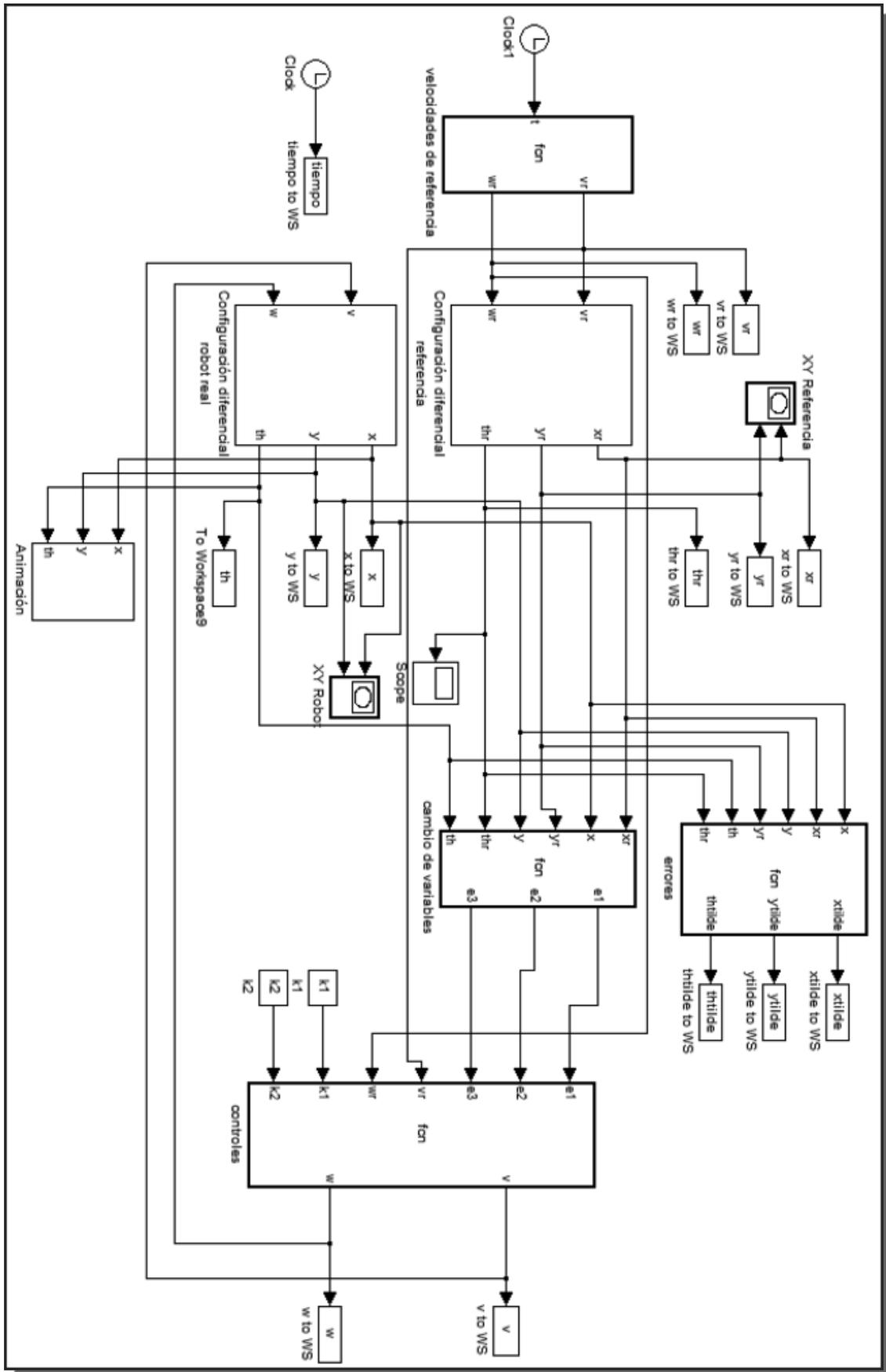


Fig. 4.6 Programa en Simulink del sistema en lazo cerrado

Trayectoria del robot en el plano XY (figura 4.7):

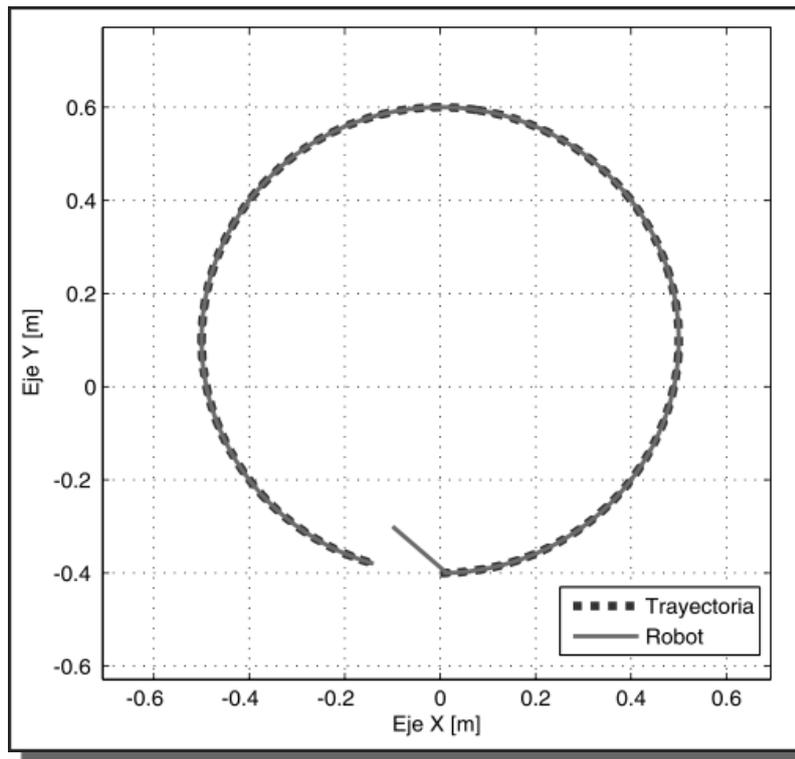


Fig. 4.7 Trayectoria en el plano XY

Variable \tilde{x} (figura 4.8):

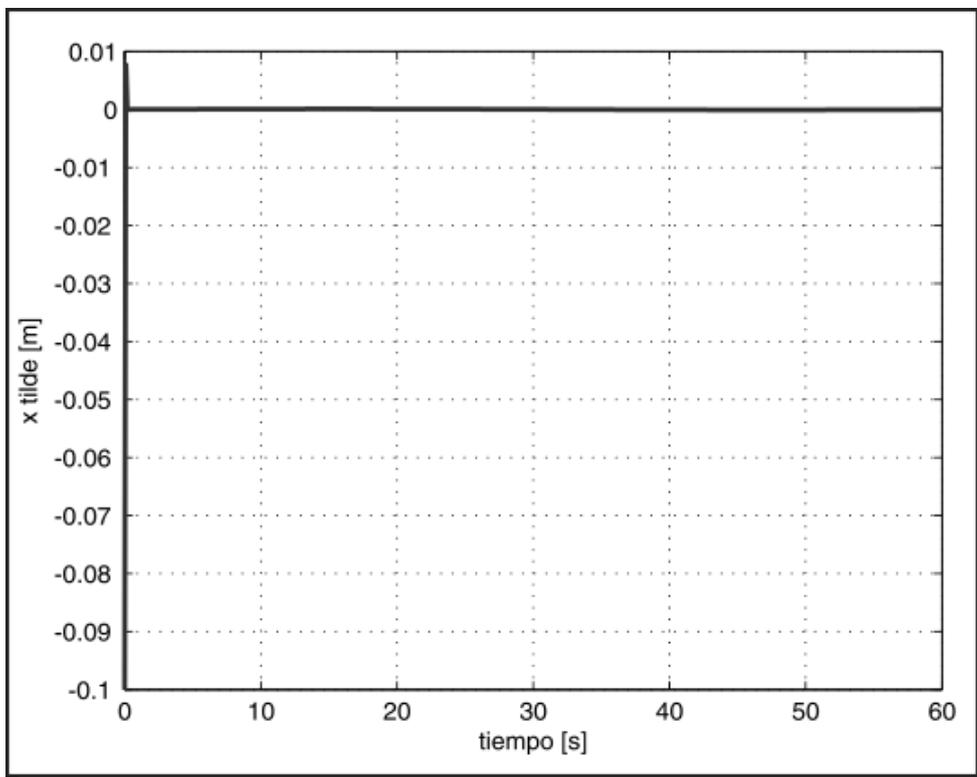


Fig. 4.8 Error en x (\tilde{x})

Variable \tilde{y} (figura 4.9):

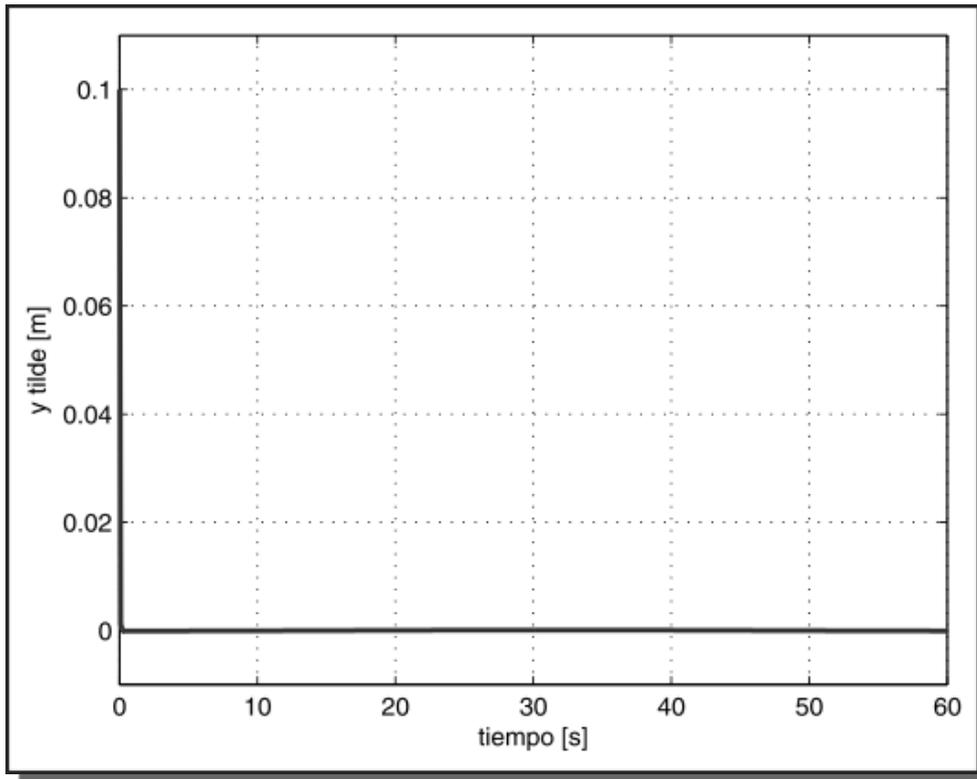


Fig. 4.9 Error en y (\tilde{y})

Variable $\tilde{\theta}$ (figura 4.10):

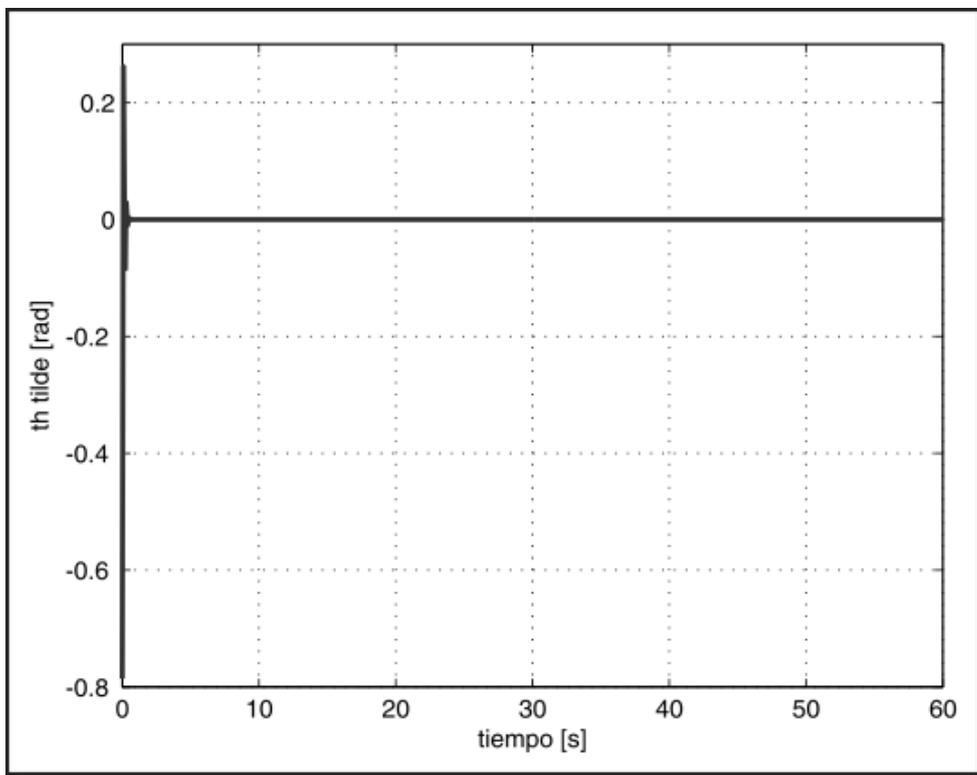


Fig. 4.10 Error en θ ($\tilde{\theta}$)

Velocidad lineal aplicada al robot (figura 4.11):

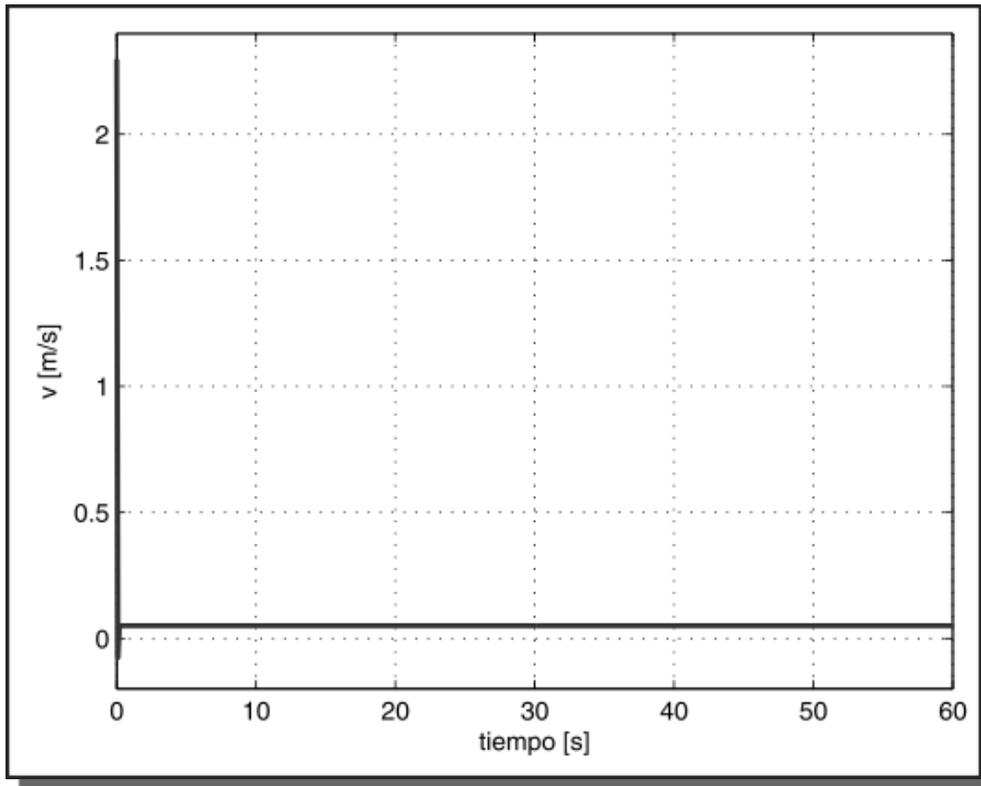


Fig. 4.11 velocidad lineal del robot, v

Velocidad angular aplicada al robot (figura 4.12):

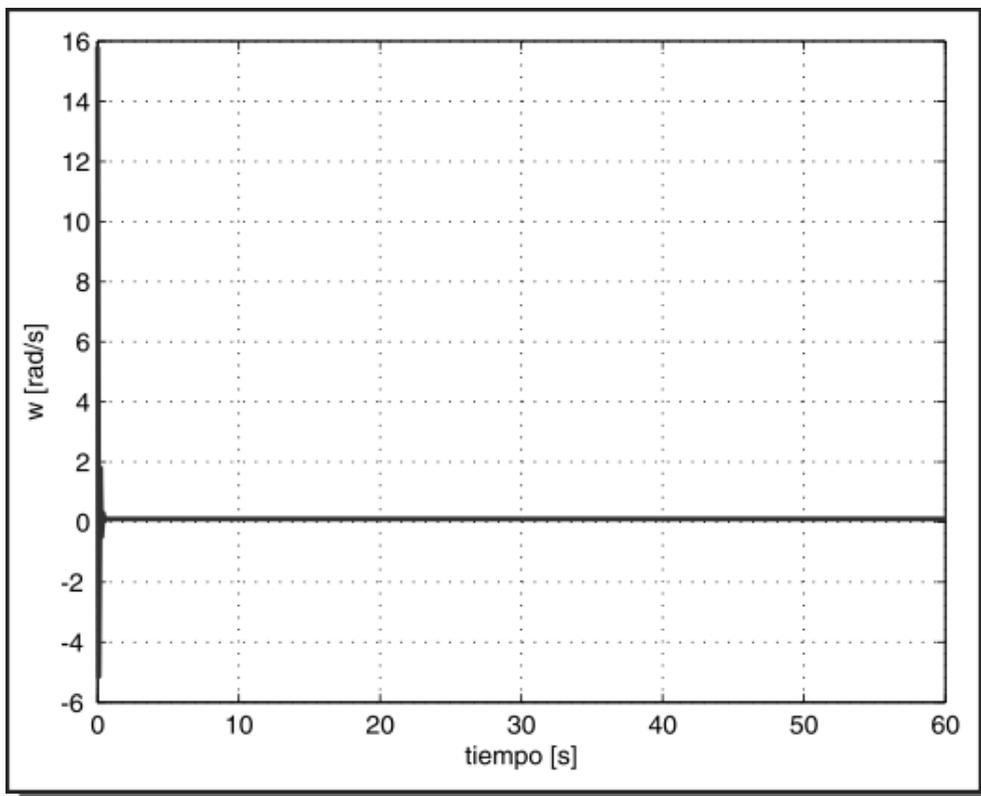


Fig. 4.12 velocidad angular del robot, ω

4.3 Simulación usando control lineal

La figura 4.13 muestra el programa a bloques en Simulink:

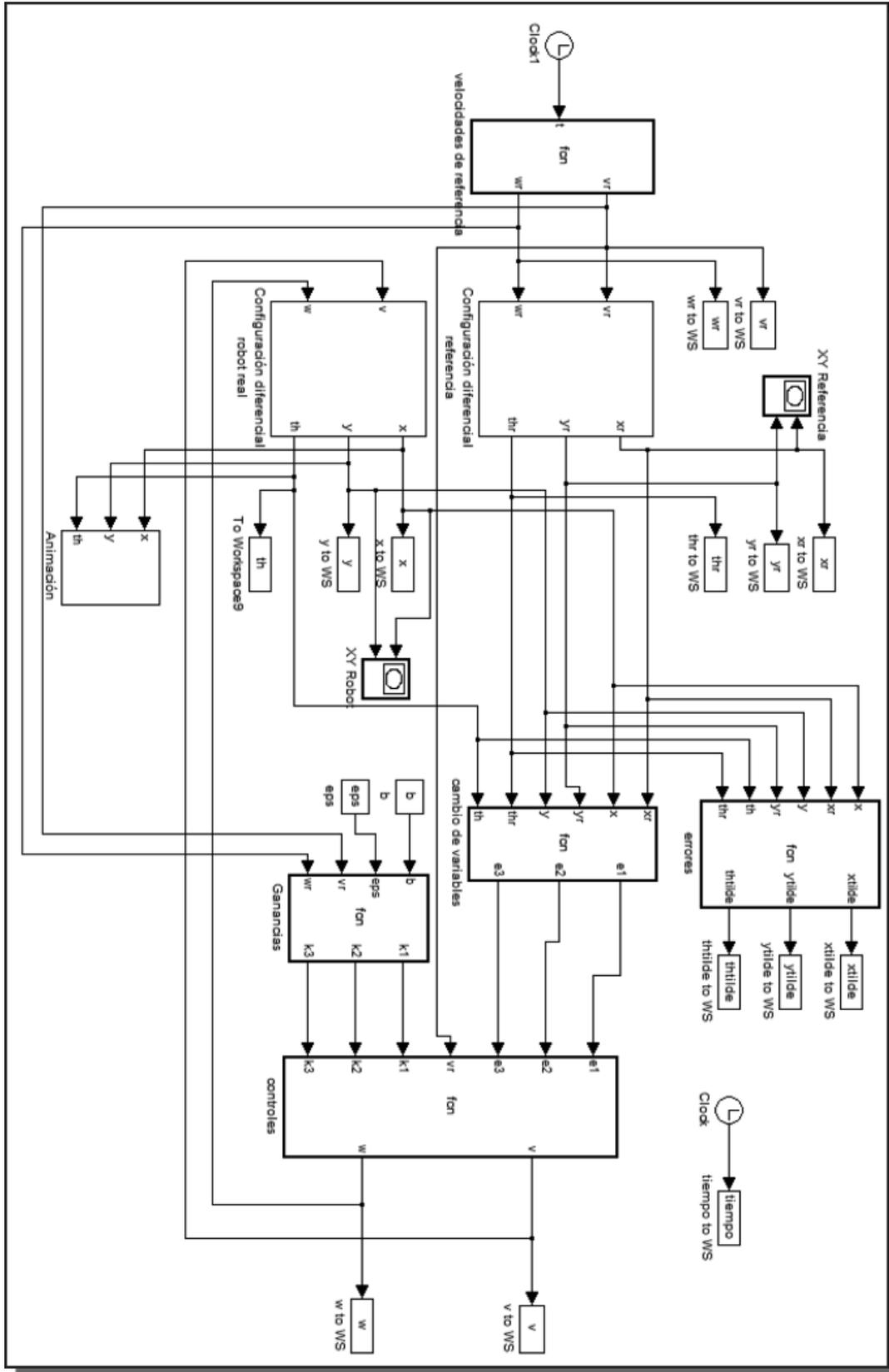


Fig. 4.13 Programa en Simulink del sistema en lazo cerrado

El bloque “Ganancias” de la misma figura implementa el “escalado de velocidad”, descrito en la sección 3.3 y dado por (3.25):

```
function [k1,k2,k3] = fcn(b,eps,vr,wr)

k1 = 2*eps*sqrt(wr^2+b*vr^2);

k2 = b*abs(vr);

k3 = k1;
```

Las velocidades para el modelo de referencia (figura 4.13) están dadas por:

$$v_r = \begin{cases} 0.15 \text{ m/s}, & 0 \leq t < 20 \text{ s} \\ 0.1 \text{ m/s}, & t \geq 20 \text{ s} \end{cases}$$

$$\omega_r = \begin{cases} 0.2 \text{ rad/s}, & 0 \leq t < 10 \text{ s} \\ 0.1 \text{ rad/s}, & t \geq 10 \text{ s} \end{cases}$$

El robot inicia en (0.2, -0.2) con un ángulo de $\pi/2$.

Los parámetros de control son:

$$b = 30$$

$$\xi = 0.9$$

El tiempo de la simulación es de 60 segundos.

Los resultados de la simulación se pueden observar en las figuras 4.14 – 4.19.

Se puede notar que el robot va ligeramente detrás de la trayectoria de referencia, teniéndose de este modo errores pequeños en las variables controladas.

Trayectoria del robot en el plano XY (figura 4.14):

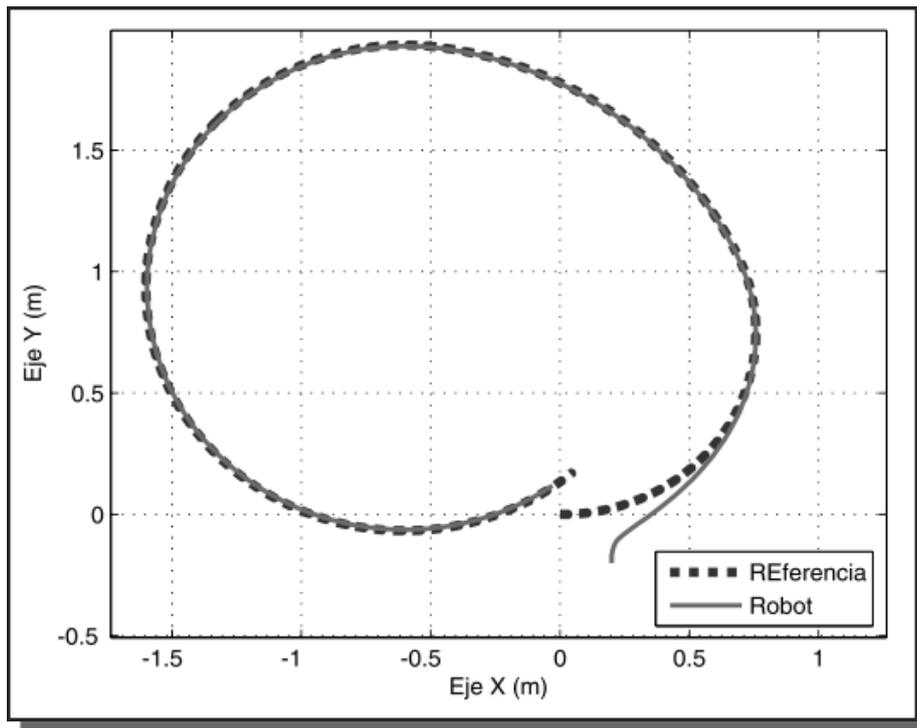


Fig. 4.14 Trayectoria en el plano XY

Variable \tilde{x} (figura 4.15):

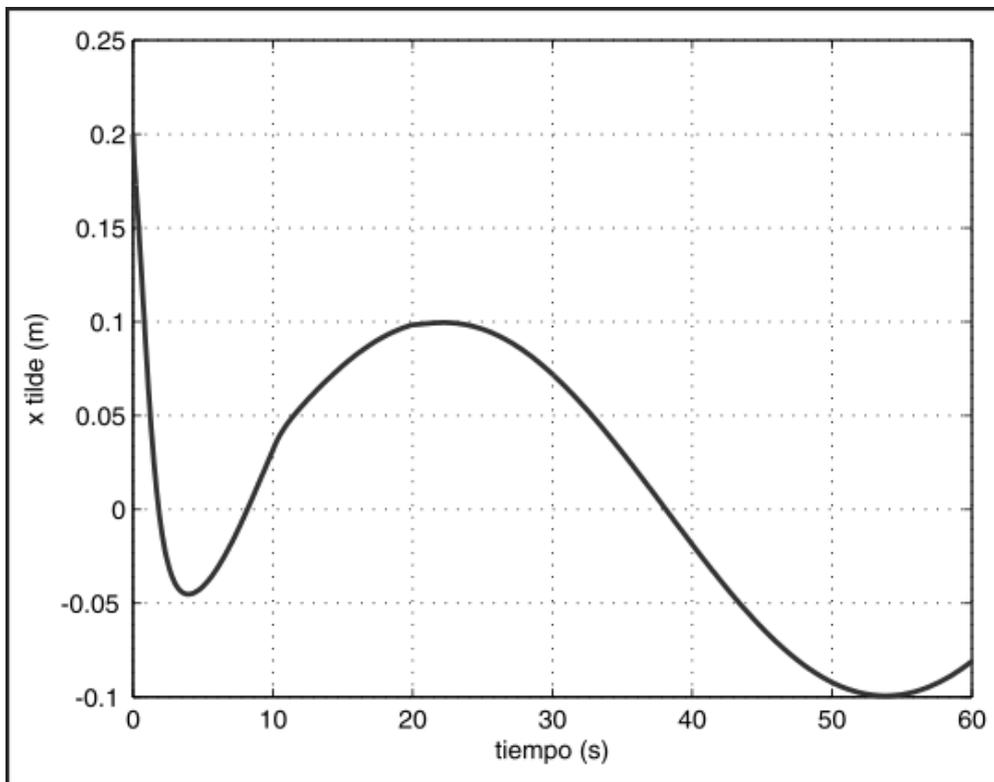


Fig. 4.15 Error en x (\tilde{x})

Variable \tilde{y} (figura 4.16):

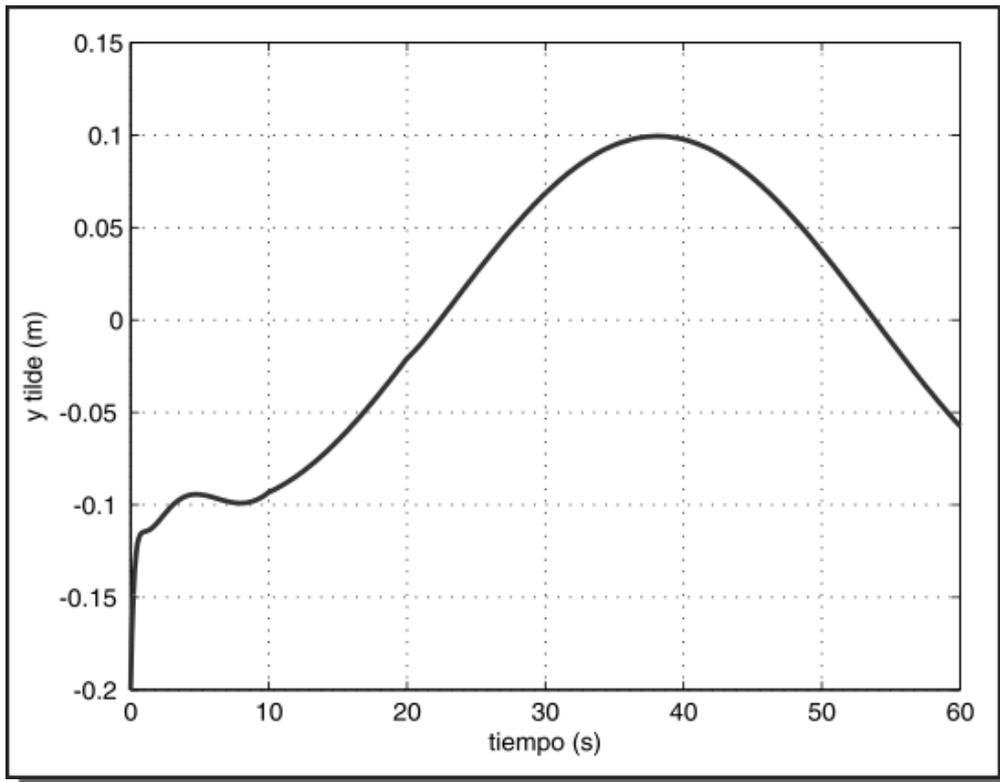


Fig. 4.16 Error en y (\tilde{y})

Variable $\tilde{\theta}$ (figura 4.17):

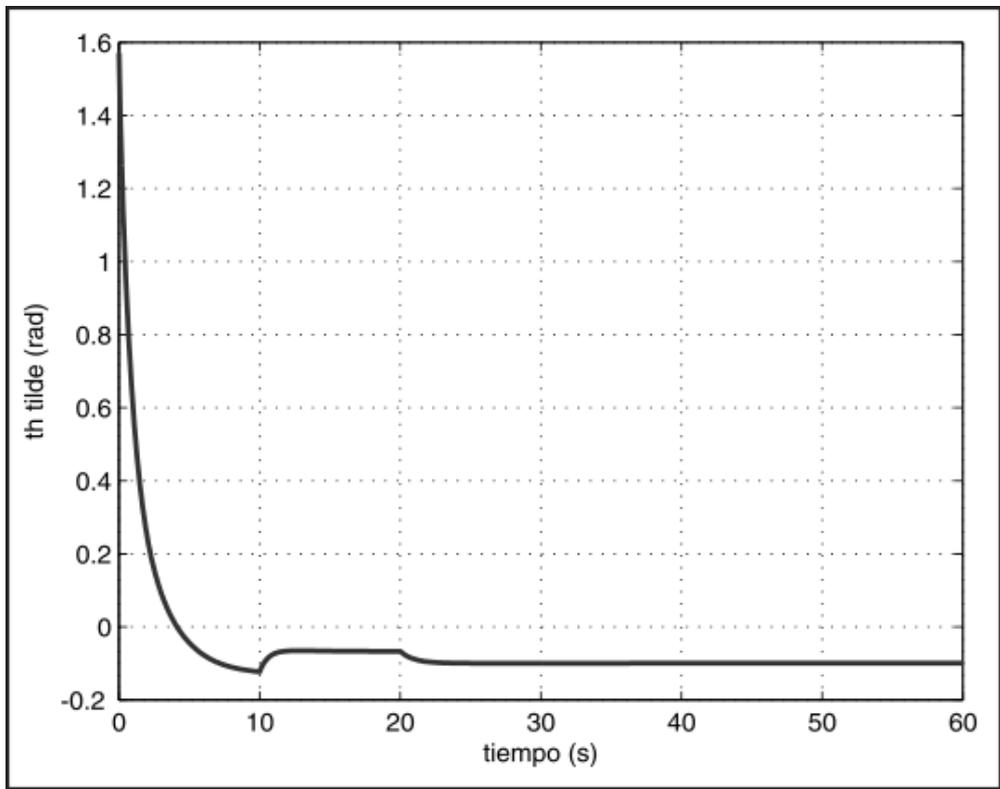


Fig. 4.17 Error en θ ($\tilde{\theta}$)

Velocidad lineal aplicada al robot (figura 4.18):

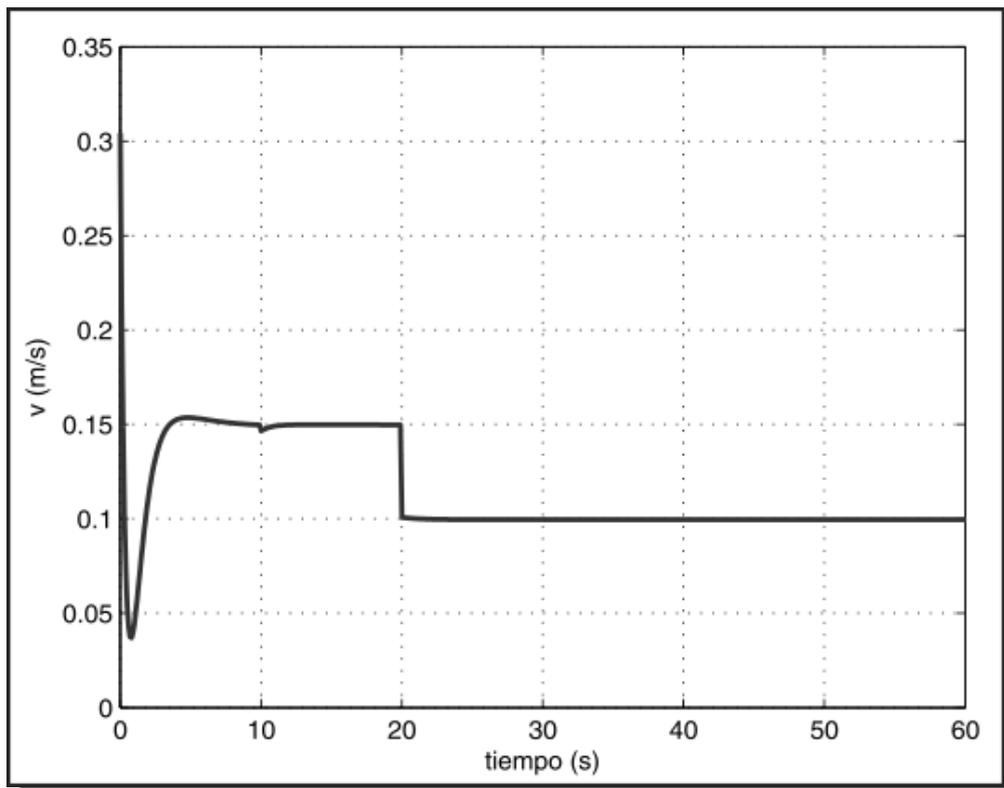


Fig. 4.18 velocidad lineal del robot, v

Velocidad angular aplicada al robot (figura 4.19):

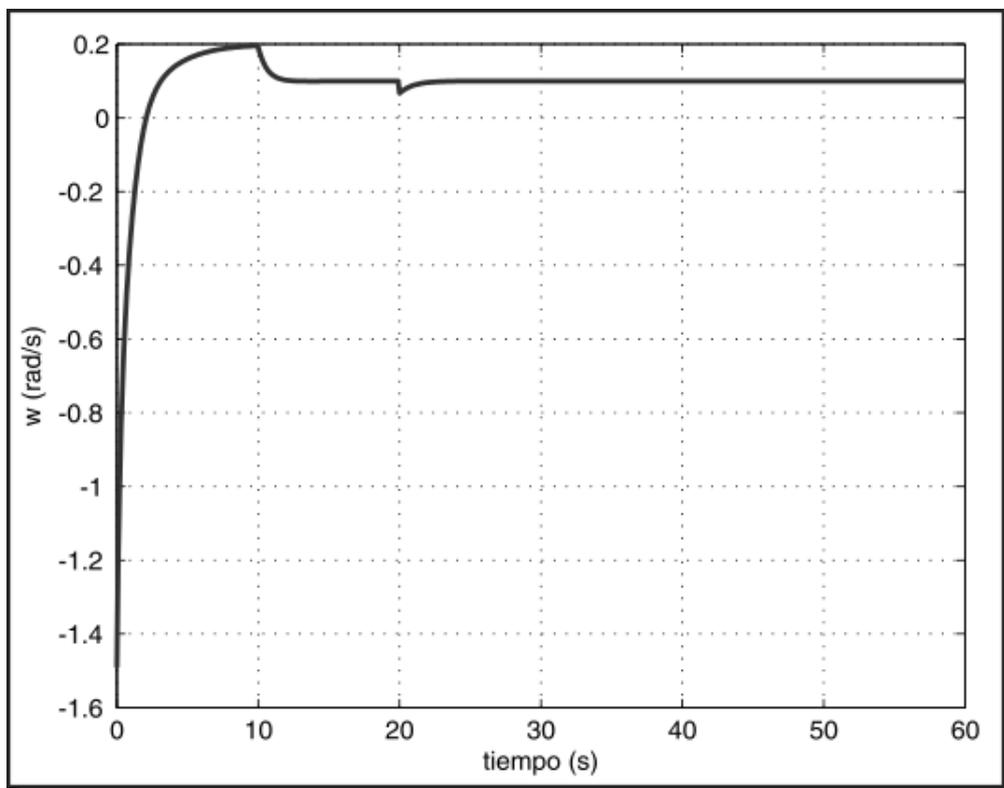


Fig. 4.19 velocidad angular del robot, w

4.4 Simulación usando control por modos deslizantes

Este caso corresponde al de la sección 3.4, y el programa se muestra en la figura 4.20:

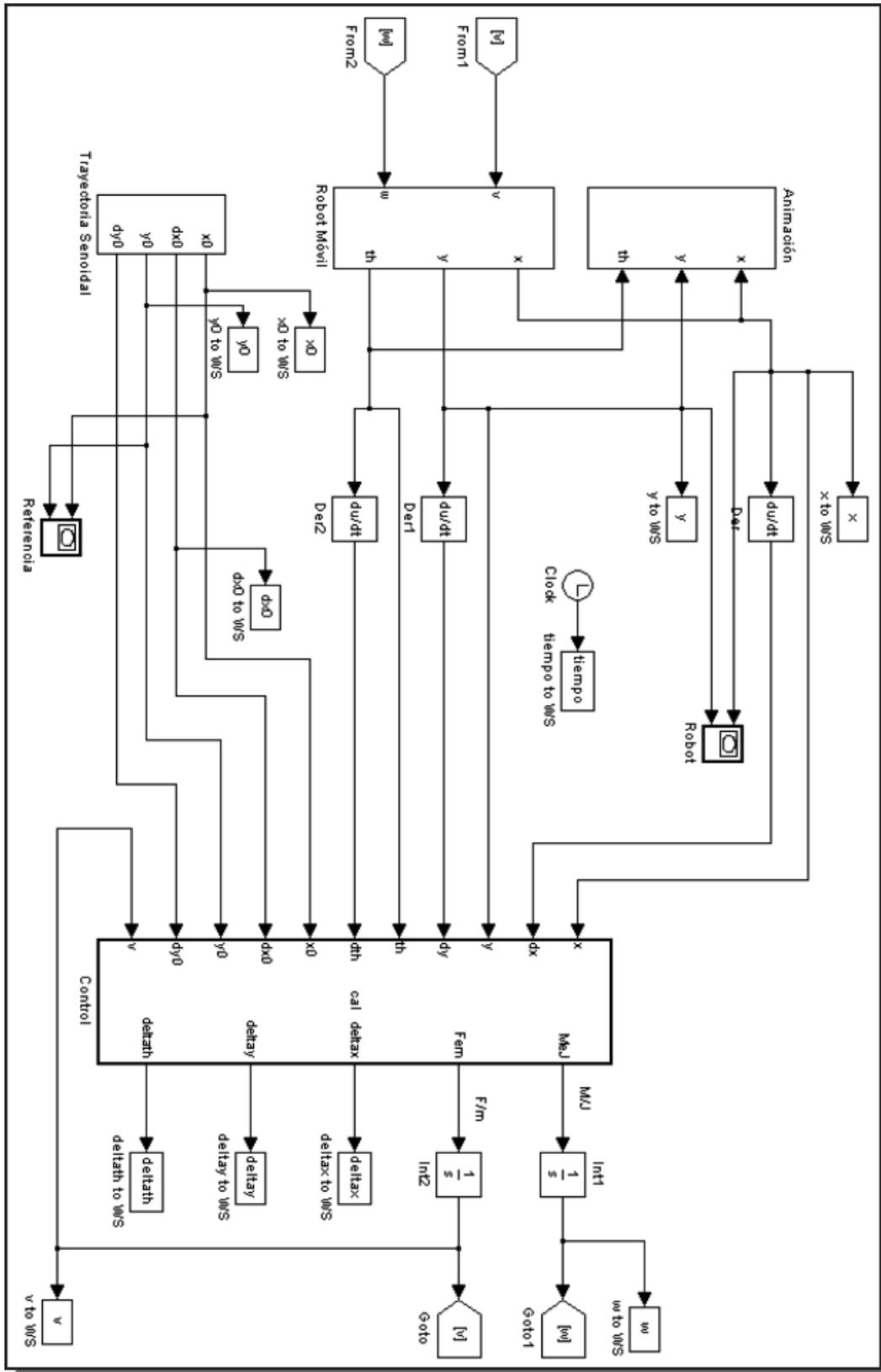


Fig. 4.20 Programa en Simulink del sistema en lazo cerrado

La trayectoria corresponde a una senoidal, y está incluida en el bloque rotulado como “Trayectoria Senoidal”. El valor x objetivo, x_o , se generó para que variará de forma suave usando un polinomio de tercer orden, tal como se describe en [13] y como se muestra enseguida:

Las condiciones inicial y final para x son:

$$x(t_0) = -1$$

$$x(t_f) = 1$$

Donde:

t_0 – tiempo inicial, en este caso $t_0 = 0$.

t_f – tiempo final.

Por lo tanto $\dot{x}(t_0) = \dot{x}(t_f) = 0$

La variable x varía de acuerdo al polinomio:

$$x(t) = a_0 + a_1t + a_2t^2 + a_3t^3 \quad (4.1)$$

Y su derivada es:

$$\dot{x}(t) = a_1 + 2a_2t + 3a_3t^2 \quad (4.2)$$

Sustituyendo las condiciones iniciales en (4.1) y (4.2):

$$x(0) = -1 = a_0$$

$$\dot{x}(0) = 0 = a_1$$

Entonces $a_0 = -1$, $a_1 = 0$. Considerando estos valores y las condiciones finales para (4.1) y (4.2):

$$x(t_f) = 1 = -1 + a_2t_f^2 + a_3t_f^3 \quad (4.3)$$

$$\dot{x}(t_f) = 0 = 2a_2t_f + 3a_3t_f^2 \quad (4.4)$$

Resolviendo (4.3) y (4.4) para encontrar a_2 y a_3 , resulta:

$$a_2 = \frac{6}{t_f^2}$$

$$a_3 = -\frac{4}{t_f^3}$$

Por lo tanto:

$$x(t) = -1 + \frac{6}{t_f^2}t^2 - \frac{4}{t_f^3}t^3 \quad (4.5)$$

$$\dot{x}(t) = \frac{12}{t_f^2}t - \frac{12}{t_f^3}t^2 \quad (4.6)$$

El comportamiento de estos polinomios para $t_f = 60$ s se muestra en las figuras 4.21 y 4.22.

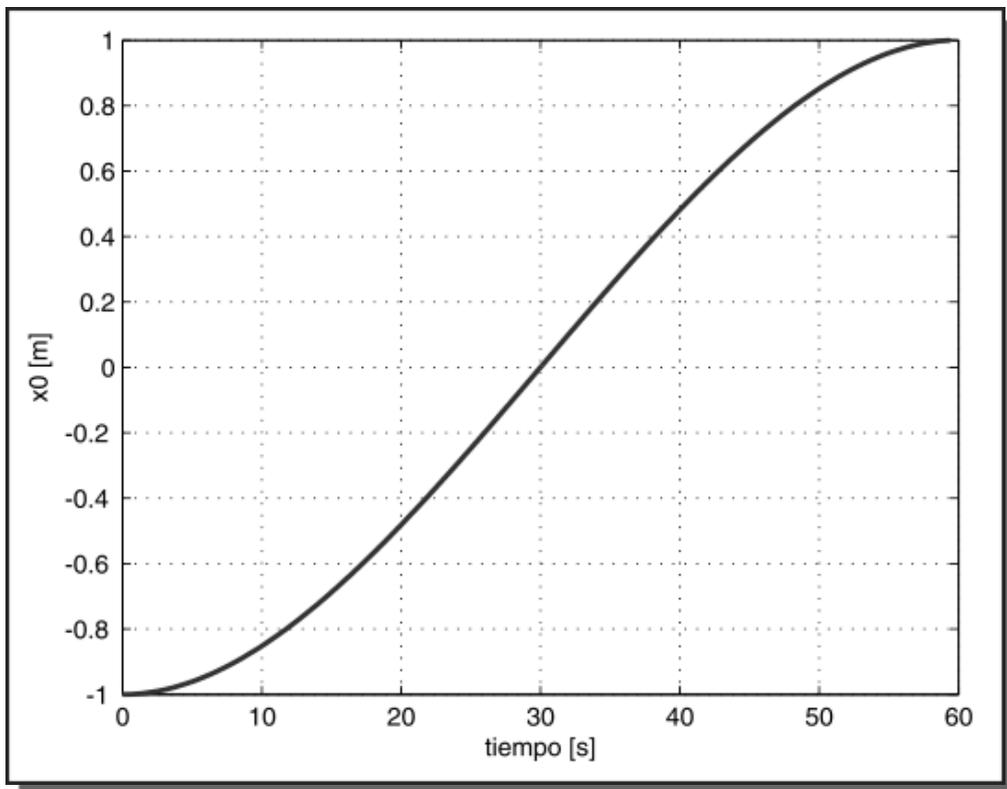


Fig. 4.21 x objetivo

Mientras que y objetivo y su derivada se definen de la siguiente forma:

$$y_0 = -0.3 \cdot \sin(x_0 \cdot \pi);$$

$$dy_0 = -0.3 \cdot \pi \cdot \cos(x_0 \cdot \pi);$$

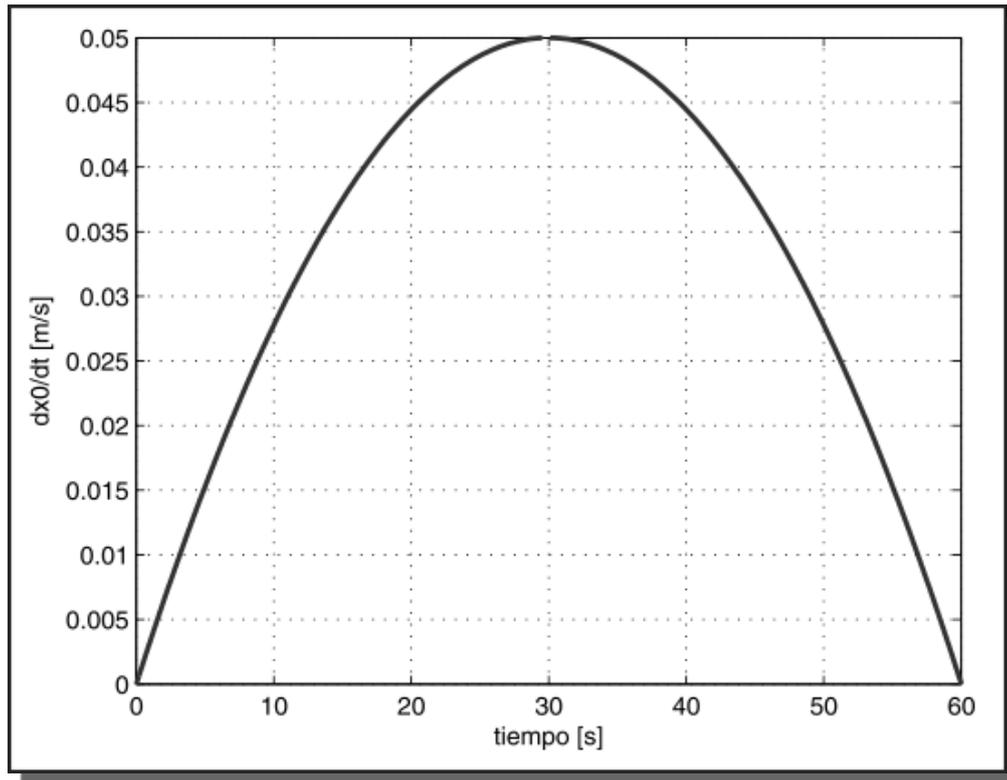


Fig. 4.22 derivada con respecto al tiempo de x objetivo

La implementación del controlador, que está en el bloque “Control” (figura 4.20), consiste del siguiente código:

```
function [MeJ, Fem, deltax, deltay, deltath] =
cal(x, dx, y, dy, th, dth, x0, dx0, y0, dy0, v)

c1 = 1;

c2 = 1;

k1 = .3;

k2 = .6;

epsilon = .001;

deltax = x - x0;

deltay = y - y0;

S1 = dx + k1 * deltax;

S2 = dy + k1 * deltay;

uu = -( S1 * cos(th) + S2 * sin (th) );
```

```

if ( v > 0 ) || ( uu >= 0 )
    Fem = c1 * uu;

elseif ( v == 0 ) && ( uu < 0 )
    Fem = 0;

else
    Fem = 0;

end

if deltax^2 + deltay^2 > epsilon
    th0 = atan2( -deltay, -deltax );

elseif ( deltax^2 + deltay^2 <= epsilon ) && ( dx0^2 + dy0^2 ~= 0 )
    th0 = atan2( -dy0, -dx0 );

elseif ( deltax^2 + deltay^2 <= epsilon ) && ( dx0^2 + dy0^2 == 0 )
    th0 = 0.0;
else
    th0 = 0.0;

end

deltath = th - th0;

S3 = dth + k2 * deltath;

MeJ = -c2 * S3;

```

Donde podemos notar que las ganancias de control están dadas por:

$$c_1 = 1$$

$$c_2 = 1$$

$$k_1 = 0.3$$

$$k_2 = 0.6$$

El vehículo inicia en (-1.1, 0.1) con una orientación de $\pi/4$.

El tiempo de simulación es de 60 segundos.

Los resultados se muestran en las figuras 4.23 – 4.28.

Trayectoria del robot en el plano XY (figura 4.23):

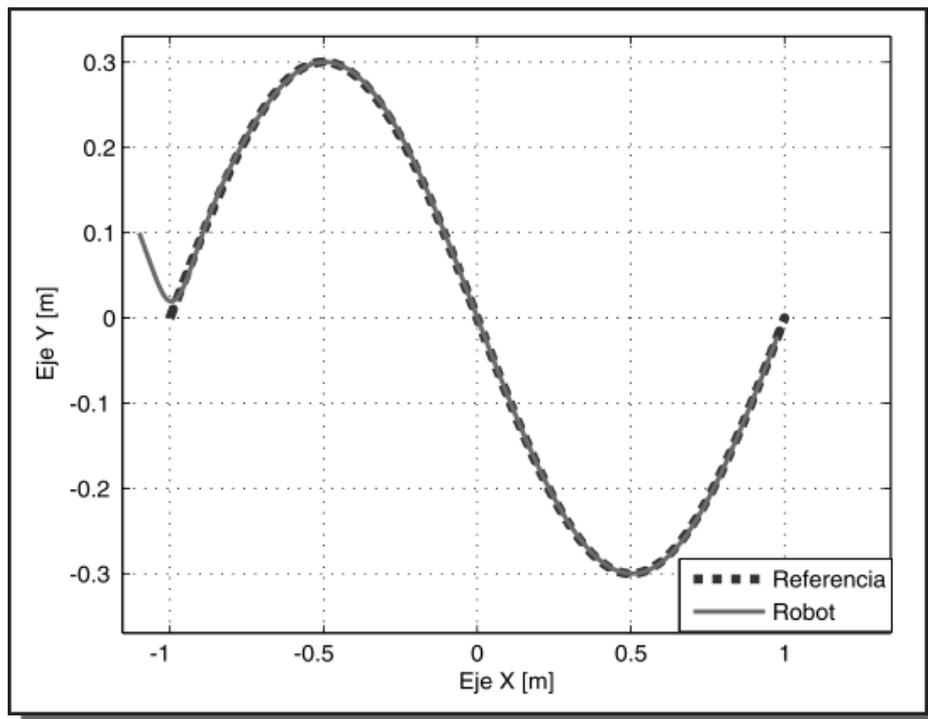


Fig. 4.23 Trayectoria en el plano XY

Variable Δx (figura 4.24):

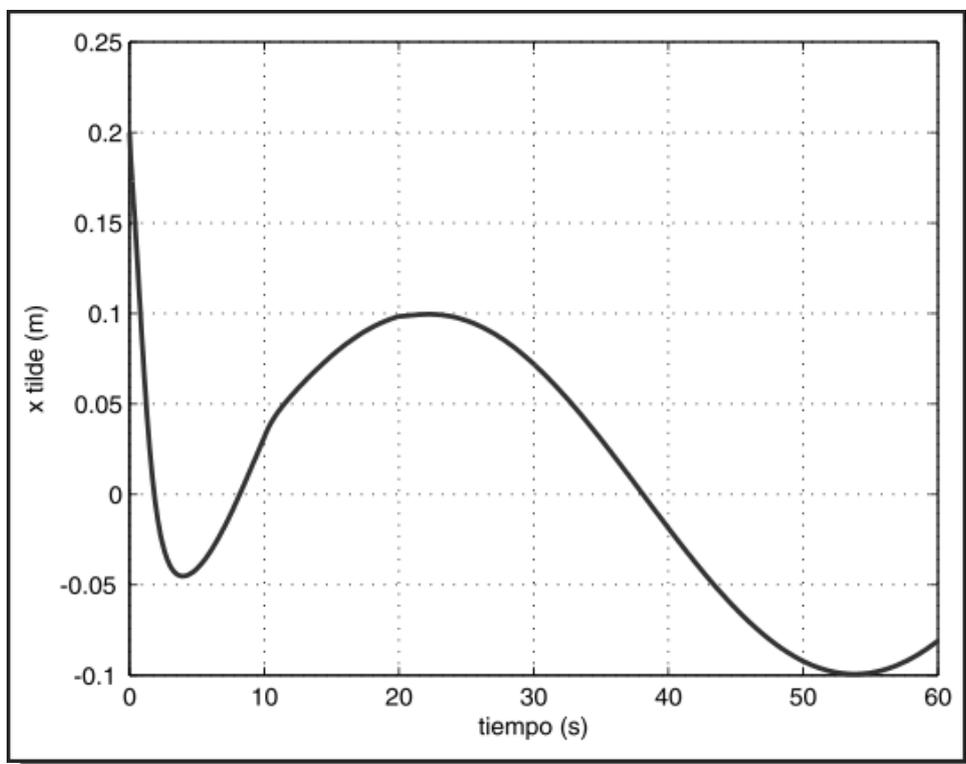


Fig. 4.24 Error en x (Δx)

Variable Δy (figura 4.25):

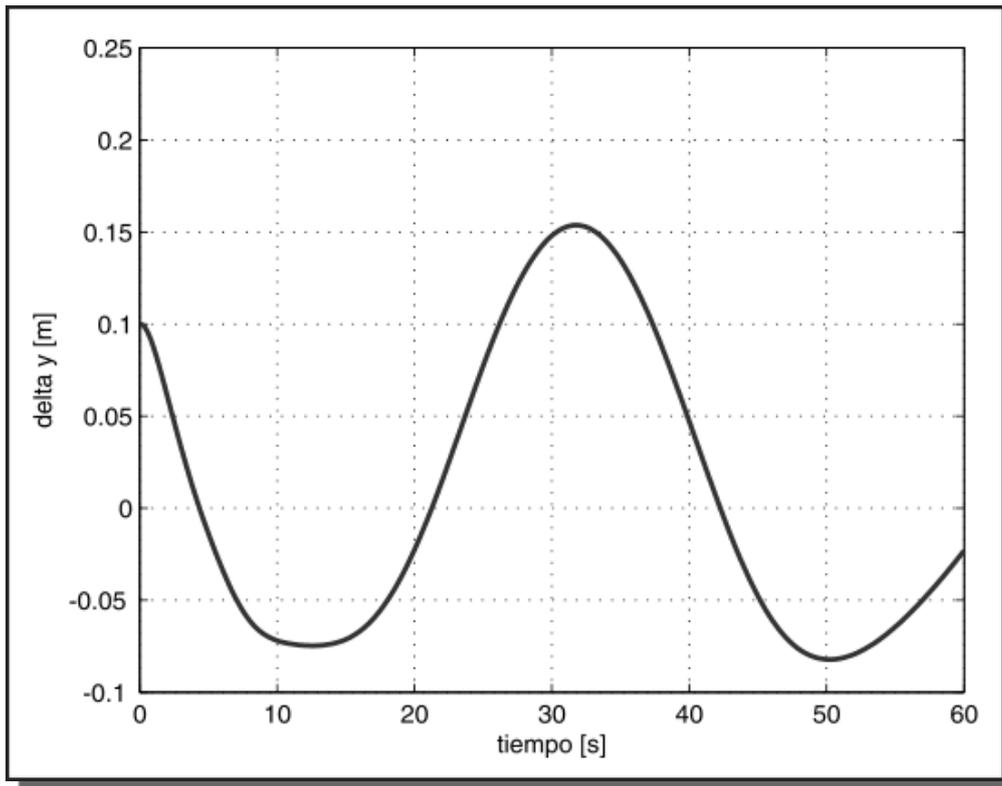


Fig. 4.25 Error en y (Δy)

Variable $\Delta \theta$ (figura 4.26):

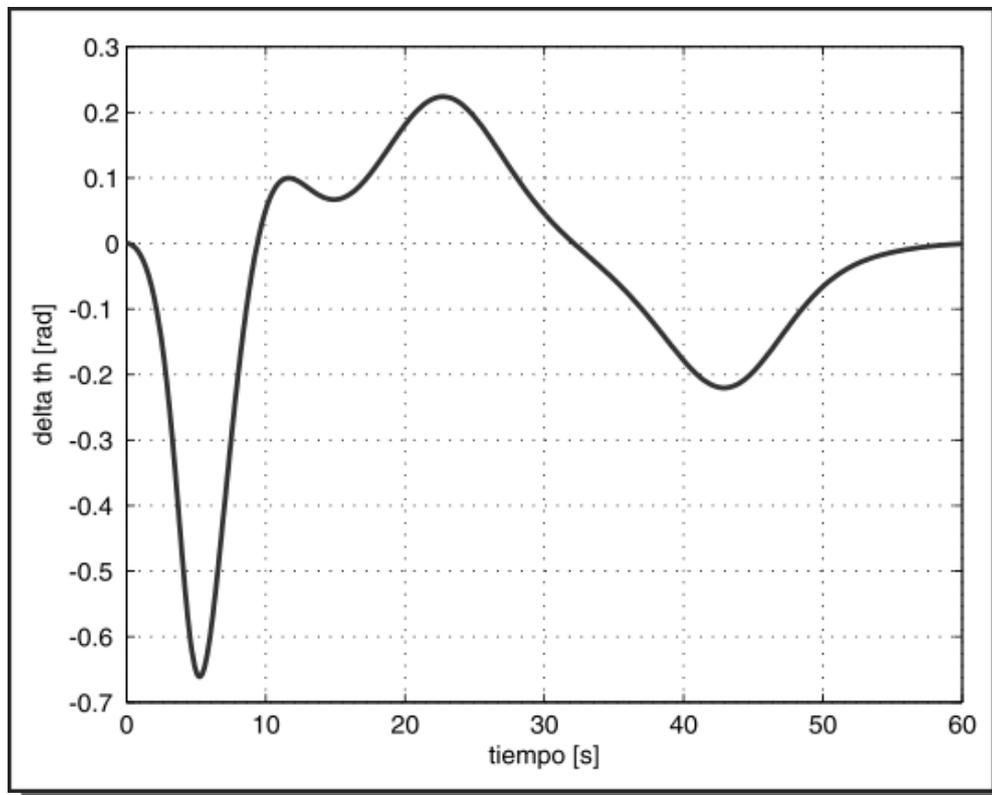


Fig. 4.26 Error en θ ($\Delta \theta$)

Velocidad lineal aplicada al robot (figura 4.27):

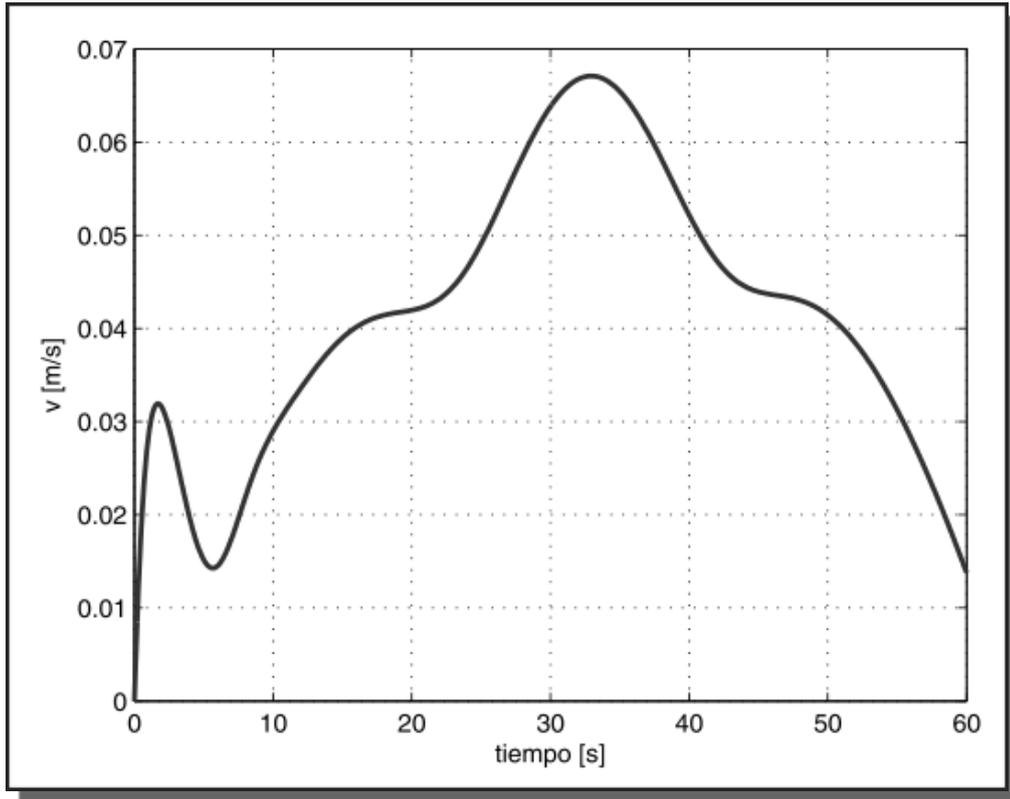


Fig. 4.27 velocidad lineal del robot, v

Velocidad angular aplicada al robot (figura 4.28):

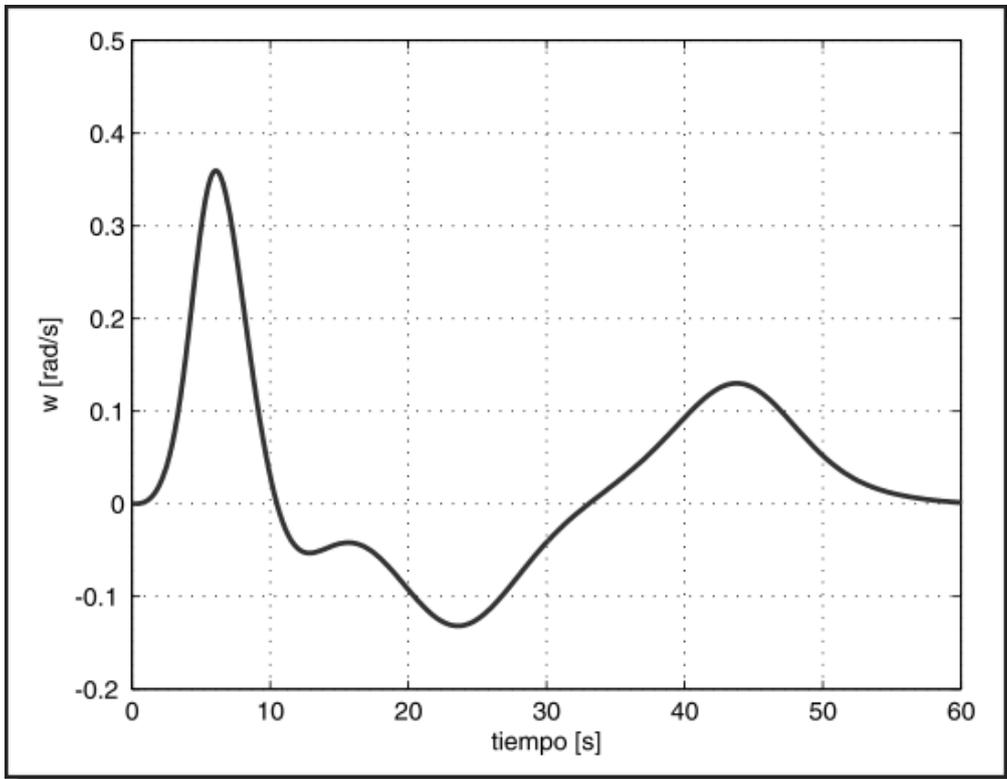


Fig. 4.28 velocidad angular del robot, ω

4.5 Animación del robot móvil en el módulo de realidad virtual de Simulink.

La animación del robot móvil muestra su desplazamiento en un entorno virtual, lo que sirve para ver mejor el movimiento del robot durante el tiempo de simulación. En particular, permite apreciar cómo va cambiando la orientación del vehículo, cosa que no se puede ver en gráficas de movimiento en el plano XY.

Para crear modelos interactivos de realidad virtual existe un lenguaje estándar conocido como VRML (Virtual Reality Modeling Language – Lenguaje de modelado de realidad virtual). MATLAB permite incorporar modelos creados en VRML para generar una animación en tercera dimensión.

Para realizar la animación, se creó un modelo en el programa V-Realm Builder de un vehículo básico (figuras 4.29 y 4.30), se incorporó también un pequeño disco de referencia en el origen. La animación permite ver al robot móvil desde distintos ángulos y a diferentes distancias.

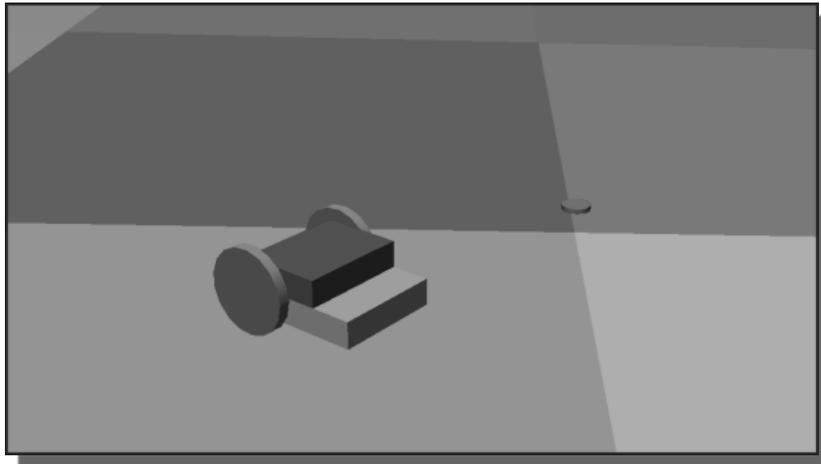


Fig. 4.29 Robot móvil creado en V-Realm Builder

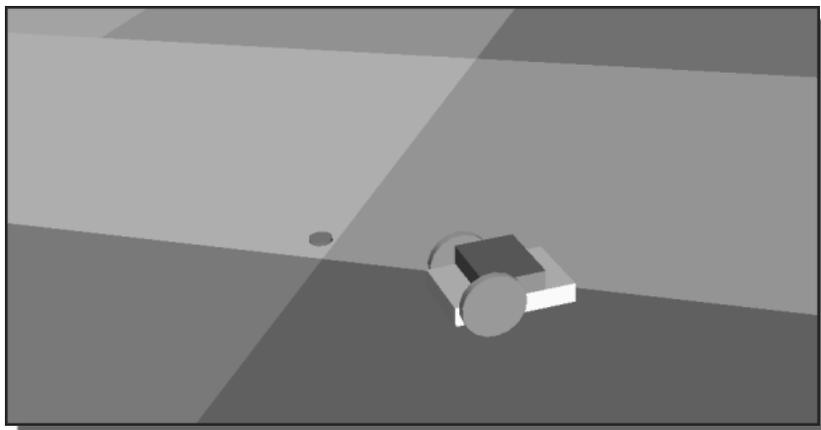


Fig. 4.30 Robot móvil creado en V-Realm Builder

La definición de los ejes de coordenadas en VRML y en MATLAB no es la misma, tal como se ilustra en la figura 4.31:

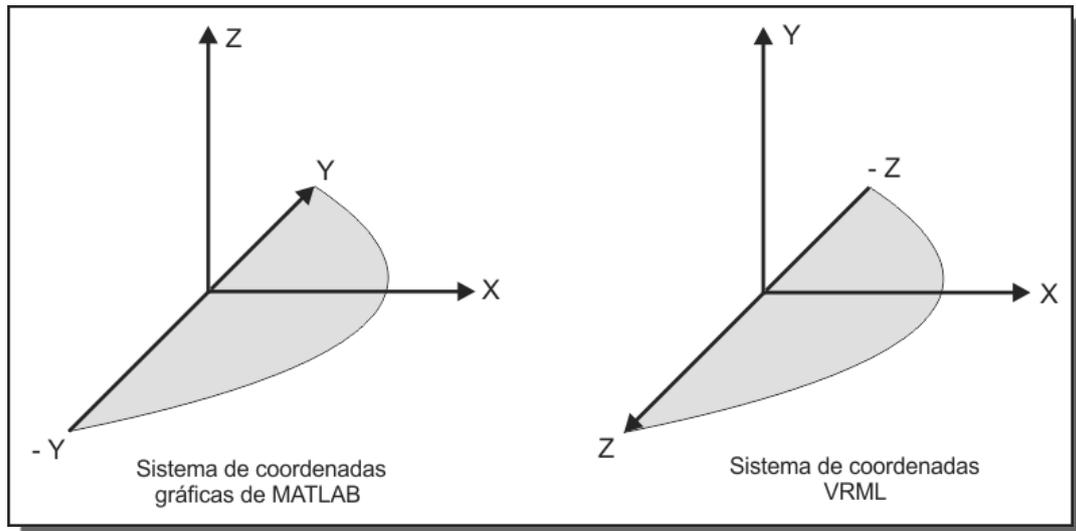


Fig. 4.31 Ejes de coordenadas en MATLAB y VRML

Por lo tanto es necesario hacer una adecuación. Los bloques que permiten generar el movimiento en tercera dimensión del robot móvil se muestran en la figura 4.32, y se incluyeron en cada una de las simulaciones mostradas en las secciones anteriores.

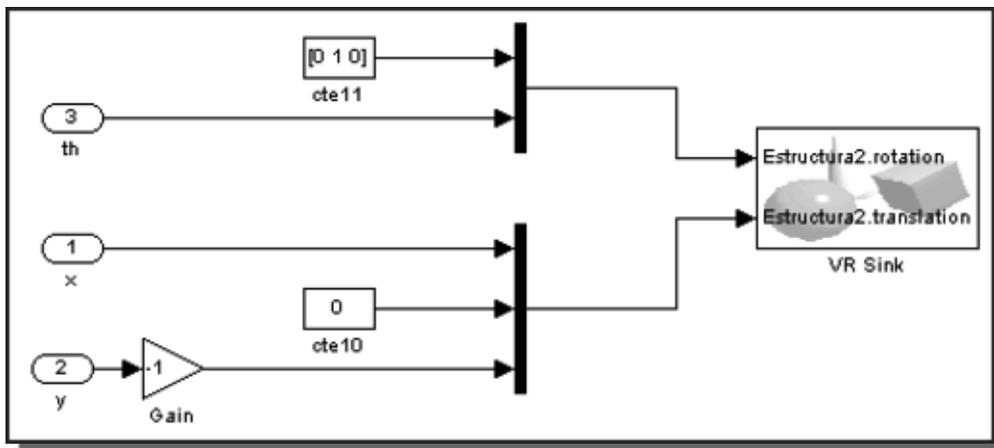


Fig. 4.32 Bloques para la animación en Simulink

CAPÍTULO 5

Diseño de los circuitos electrónicos y programación

5.1 Descripción del robot móvil utilizado en el trabajo

La implementación del sistema de control se realizó en el robot García de la empresa Acroname, el cual se muestra en la figura 5.1, y cuyos componentes se describen a continuación.

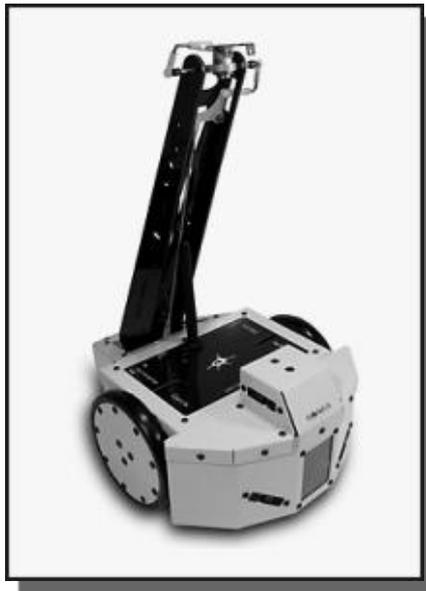


Fig. 5.1 Robot García de Acroname

Estructura mecánica

El chasis del robot está hecho de aluminio duro. Todos los tornillos están asegurados con tuercas tratadas con nylon para prevenir que estas se aflojen. El chasis sostiene paneles de plástico que pueden ser modificados para incorporar nuevos elementos mecánicos o electrónicos al robot.

El espacio desde la placa inferior a un suelo plano es de 12 mm (cerca de 0.5 in). La figura 5.2 muestra las dimensiones del robot expresadas en pulgadas.

Dos datos importantes que deben considerarse a la hora de implementar el sistema de control son: la distancia b entre las dos ruedas, y el radio r de las ruedas (las dos ruedas son iguales). Estas distancias son:

$$b = 7.175 \text{ in} = 0.182245 \text{ m}$$

$$r = 2 \text{ in} = 0.0508 \text{ m}$$

El robot García está diseñado para manejar una carga útil moderada. Se recomienda mantener la masa limitada a un máximo de 2 Kg.

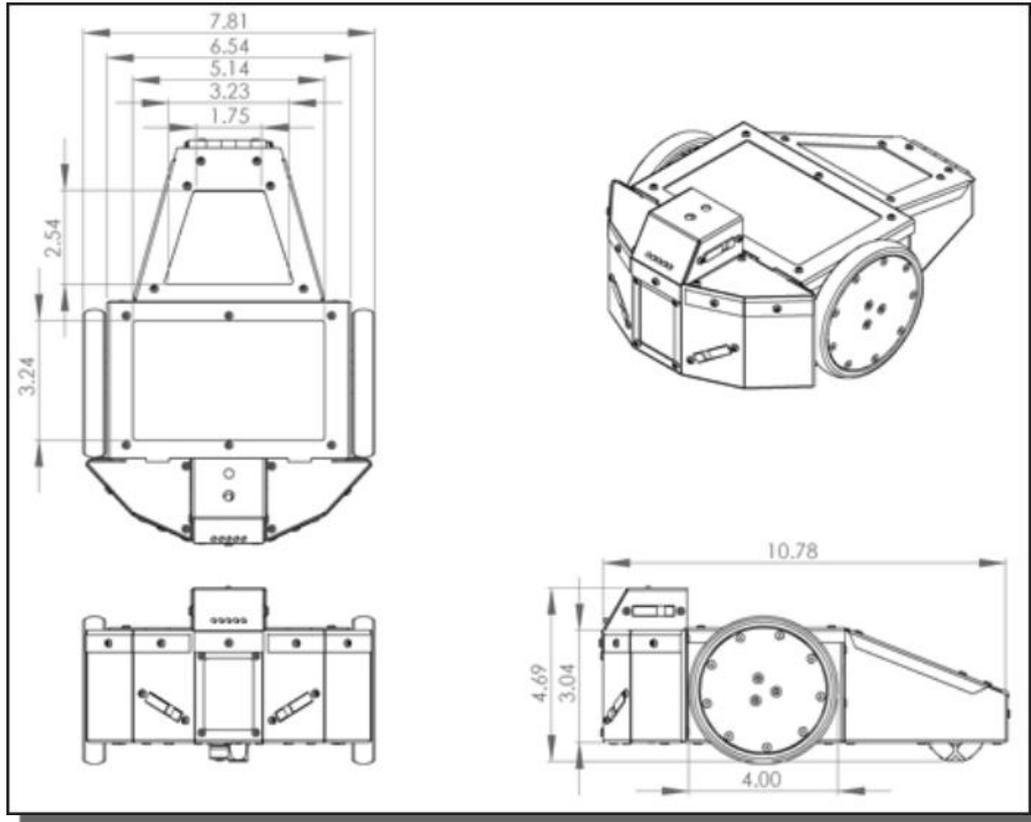


Fig. 5.2 Dimensiones del robot García (en pulgadas)

Batería

El robot se alimenta con una batería estándar de Ni-MH (Níquel e Hidruro Metálico), constituida de 6 pilas que proporcionan un total de 7.2 V y una capacidad de 4200 mA·H (figura 5.3), lo cual permite varias horas de operación continua. El tiempo de ejecución depende de la configuración del robot, la cantidad de uso de los motores y la velocidad de estos.

Se incluye con el robot un pequeño cargador programable con detector de picos (figura 5.3). Este puede recargar un empaque de baterías completamente descargado en un par de horas.



Fig. 5.3 Batería y cargador

Motores y encoders (codificadores) ópticos de cuadratura

Para su locomoción, el robot García cuenta con dos motores, de la marca Maxon, con encoders de cuadratura incorporados de 16 cpr (cuentas por revolución) y caja de engranes planetarios de 19:1 (figura 5.4). Una etapa de reducción de engranes extra 3:1, entre el motor y la rueda, soporta el peso del robot (figura 5.5).



Fig. 5.4 Motor con engranaje y encoder

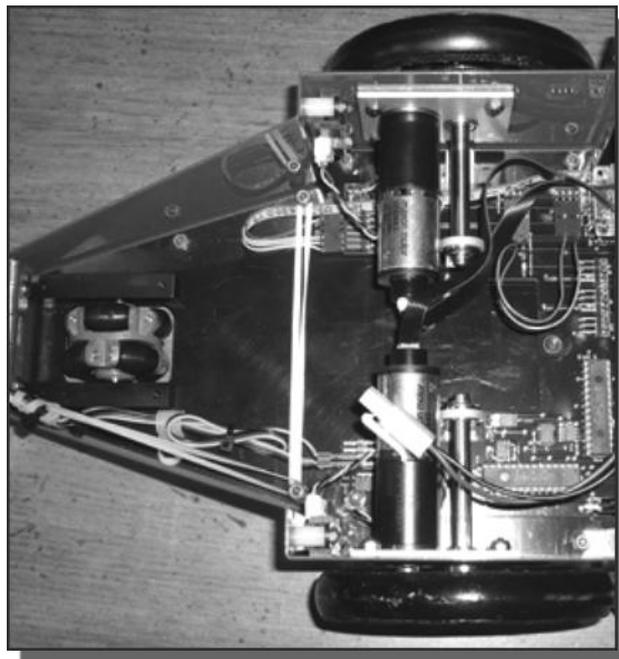


Fig. 5.5 Motores y ruedas del robot

Un encoder o codificador de cuadratura óptico es un dispositivo que genera un par de señales pulsantes desfasadas 90°. Estas señales se pueden utilizar para determinar el desplazamiento angular del motor acoplado al eje del encoder, así como su sentido de giro. Su principio de funcionamiento se puede apreciar en la figura 5.6, en donde se observa un disco ranurado que permite o no el paso de luz del LED emisor a los receptores ópticos durante su avance angular, esto genera dos señales (canal A y canal B), como se aprecia en la figura 5.7 (nota: la figura no corresponde al encoder y rueda del robot, sólo se utiliza como ejemplo ilustrativo). Adicionalmente, un encoder de este tipo puede incluir una tercera señal llamada índice, que envía un pulso cada vez que se completa una revolución del disco.

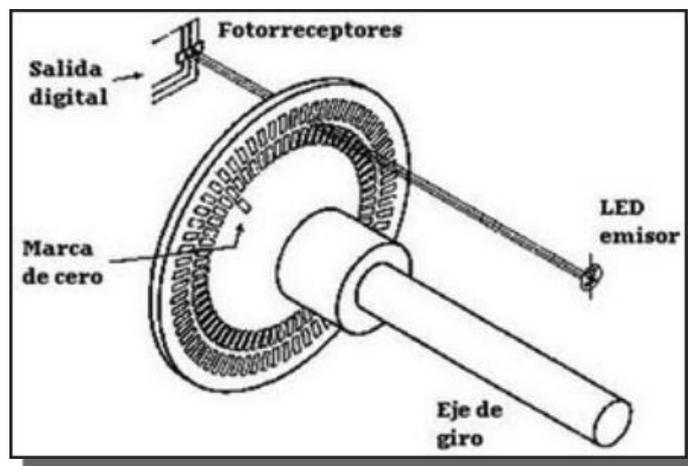


Fig. 5.6 Principio de funcionamiento de un encoder de cuadratura

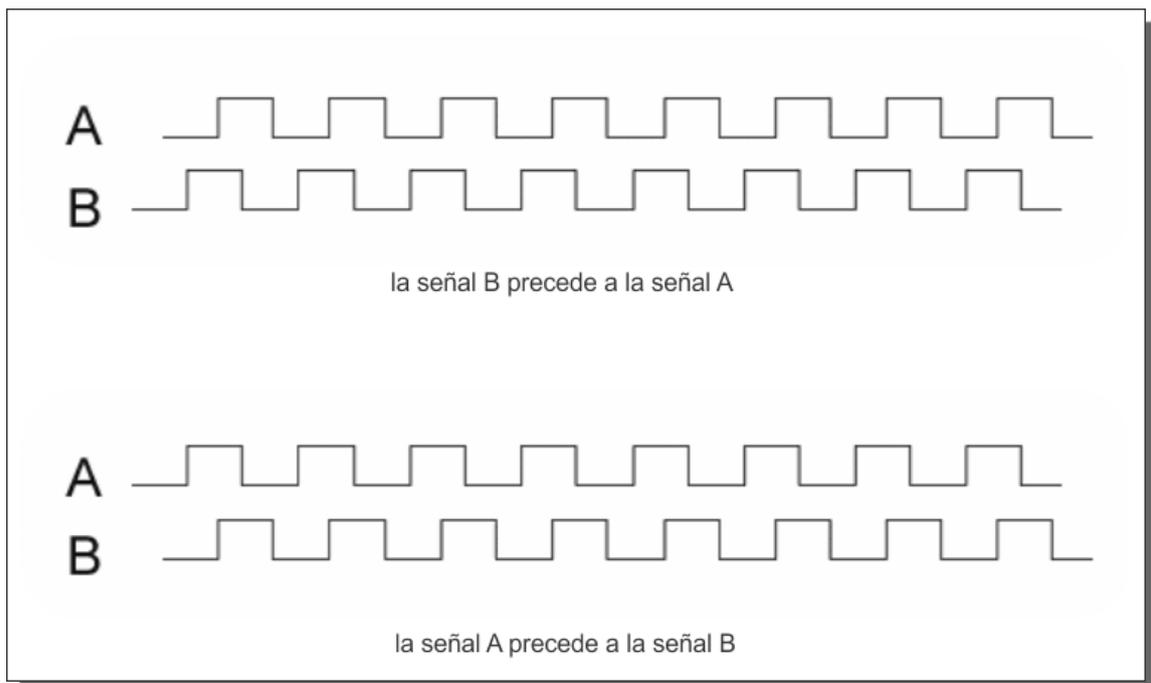


Fig. 5.7 Señales generadas por el encoder de cuadratura

La figura 5.8 muestra un encoder óptico acoplado a una rueda (nota: la figura no corresponde al encoder y rueda del robot, sólo se utiliza como ejemplo ilustrativo):

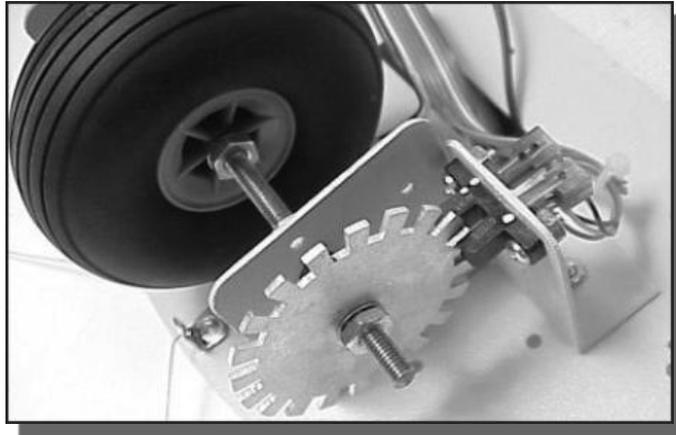


Fig. 5.8 Encoder óptico y rueda

Rueda omnidireccional

Conocida como rueda Caster ó rueda “loca”, consiste de una rueda no actuada tipo sueca de Poliuretano. Esta rueda proporciona estabilidad y permite que el robot gire con libertad en cualquier dirección (figura 5.9).

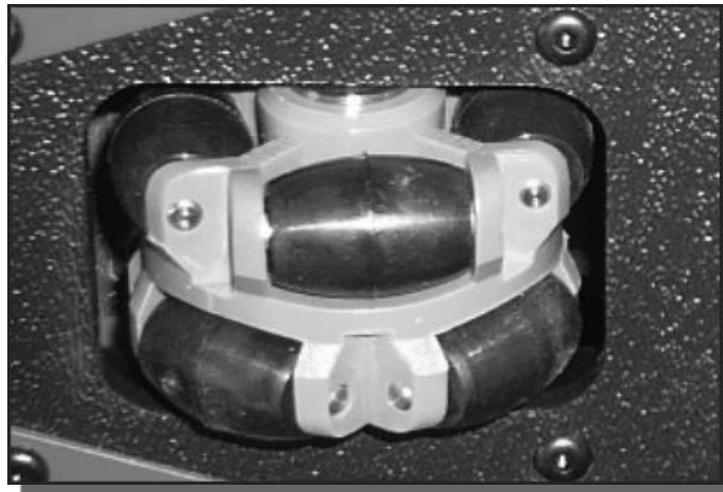


Fig. 5.9 Rueda omnidireccional

Sensores infrarrojos (IR)

El robot cuenta con sensores de distancia infrarrojos que proveen mediciones de distancia válidas en un rango de 4 a 18 pulgadas. Estos se utilizan cuando se desea que el robot siga paredes o detecte obstáculos mientras maniobra (esto puede ser útil en tareas de laberintos).

Bajo la parte frontal existe un par de detectores de proximidad (figura 5.10). Estos sensores le pueden indicar al robot si está cercano a rodar sobre una saliente para que este apague los motores.

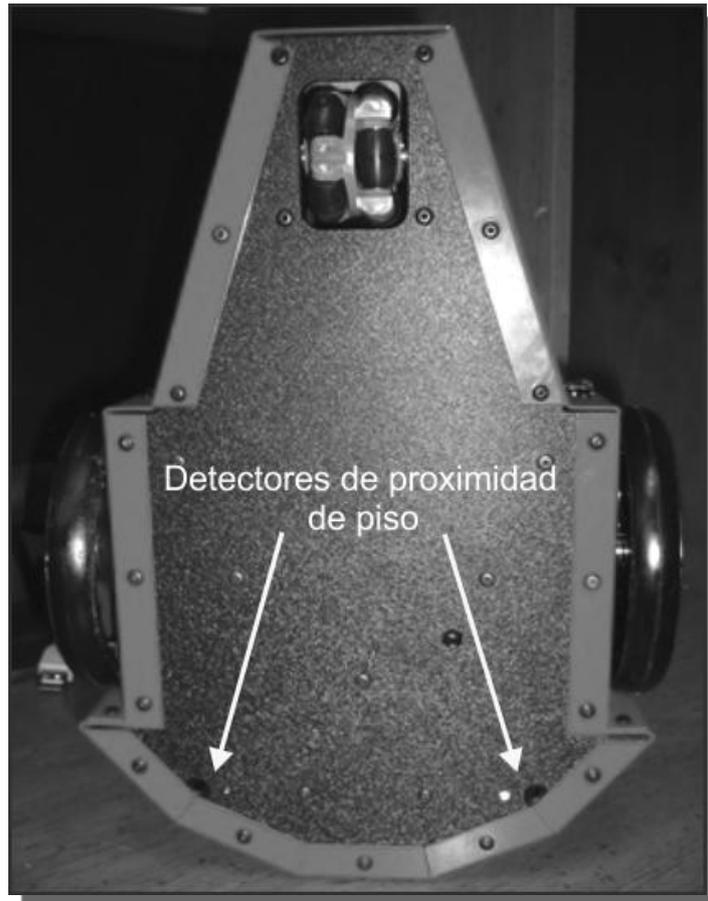


Fig. 5.10 Detectores IR de piso

LEDs indicadores

Hay cuatro LEDs indicadores en el frente del robot (figura 5.11). Dos son para indicar el funcionamiento de los microcontroladores internos, uno es para indicar que existe alimentación eléctrica, y otro para señalización por parte del usuario. Existe un quinto LED central transparente; es un emisor IR para enviar información a otros robots García o electrónicos de consumo que tienen capacidades remotas IR.

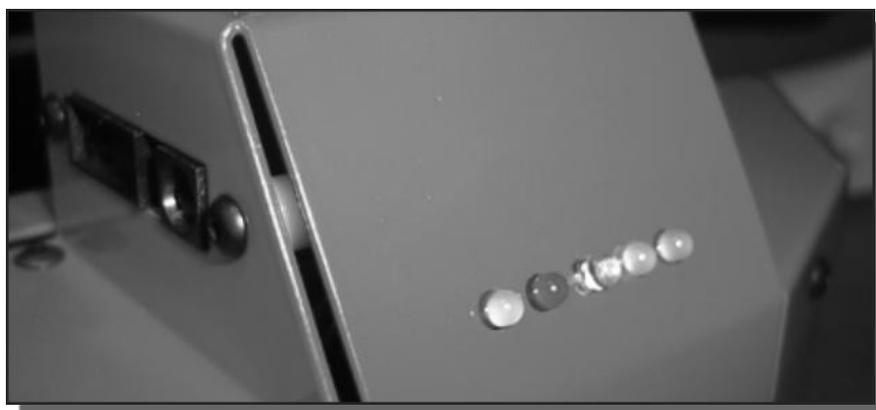


Fig. 5.11 LEDs indicadores y sensor de distancia IR

Puerto Serial

La comunicación entre un equipo de cómputo y el robot García se puede realizar a través de un puerto serie con niveles TTL (0 V – 5 V). Para establecer la comunicación es necesario adecuar los niveles de voltaje entre el microcontrolador y la computadora, esto se hace con base en el circuito integrado MAX232.

Otros componentes

El robot García cuenta con un par de microcontroladores que permiten controlar sus componentes mediante programación en C++ o en Java. En este caso no se utilizaron ya que se optó por un esquema de control que incluye una tarjeta de adquisición de datos, que permite una mayor flexibilidad y un control más directo de los actuadores.

Este robot se puede adquirir con una tarjeta Ethernet inalámbrica, una torre para el montaje de una cámara, o un sensor escáner LASER para la elaboración de mapas de su entorno de trabajo (figura 5.12).

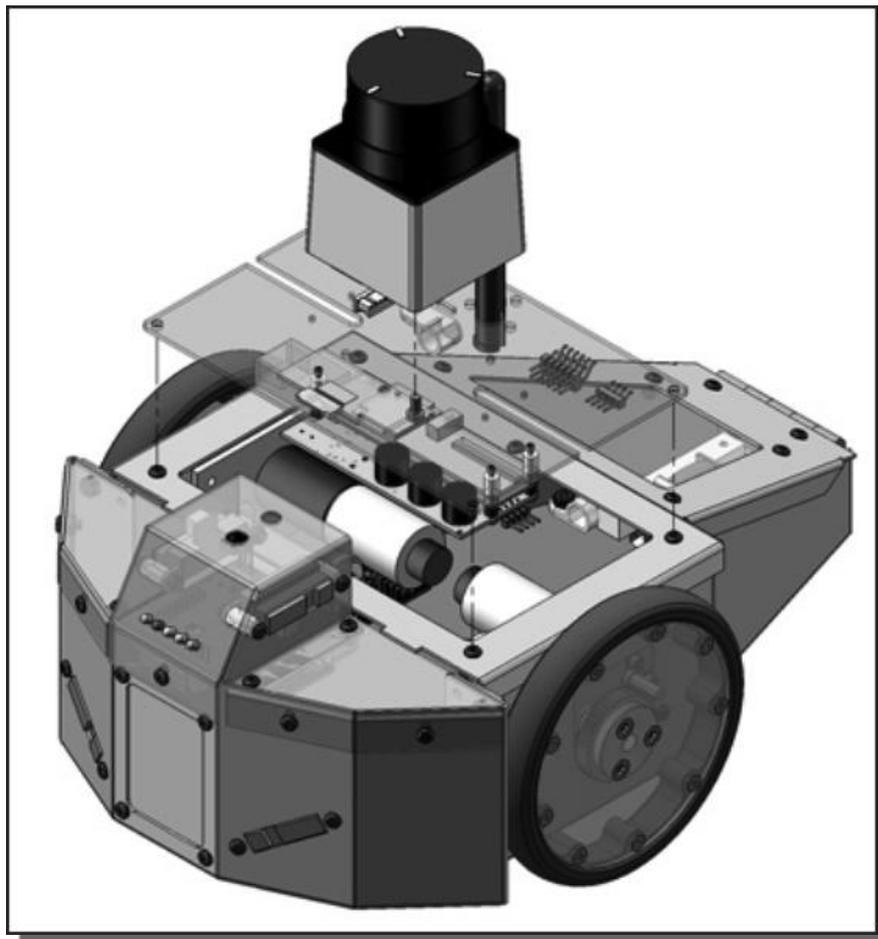


Fig. 5.12 Montaje de escáner LASER en robot García

5.2 Circuitos de puente H del robot

Un circuito de puente H es un circuito de potencia que sirve para aplicar un voltaje variable a un motor, usando modulación por ancho de pulsos – PWM (ver sección 5.4), y que permite cambiar el sentido de giro del motor a partir de una señal de entrada.

Un puente H puede ser construido utilizando transistores de potencia funcionando como interruptores, o se puede optar por un circuito integrado que desempeña esta función (L293D, L298N, LM18200T, etc.). El tipo de puente H a utilizar depende de factores como los niveles de corriente, voltaje, velocidad de respuesta, etcétera, que necesiten manejarse. La figura 5.13 muestra un circuito de puente H, de la forma del circuito proviene su nombre.

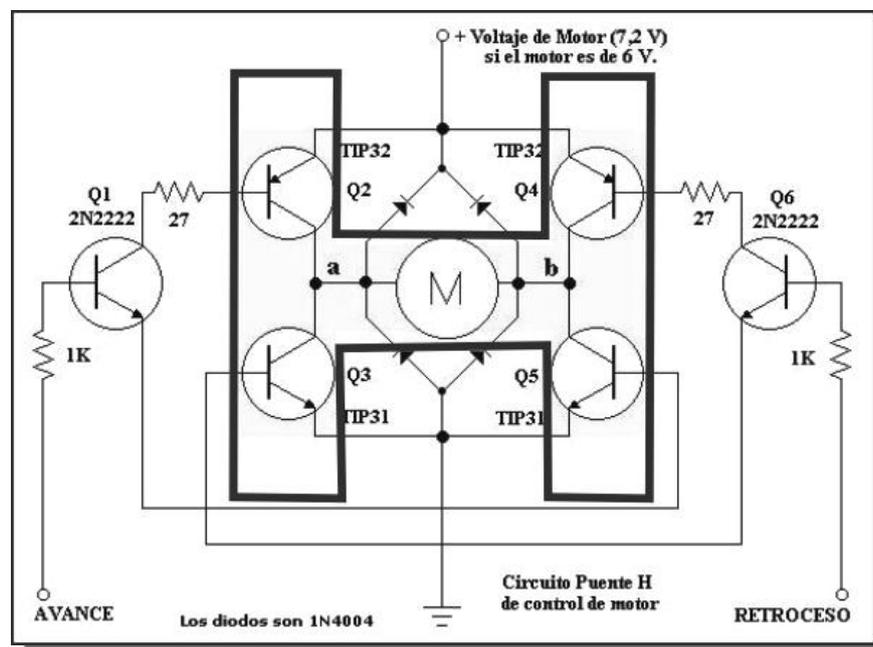


Fig. 5.13 Circuito de puente H

Para la implementación del control se utilizaron los puentes H incluidos con el robot García. En la figura 5.14 se muestra este dispositivo, su diagrama a bloques se puede apreciar en la figura 5.15.

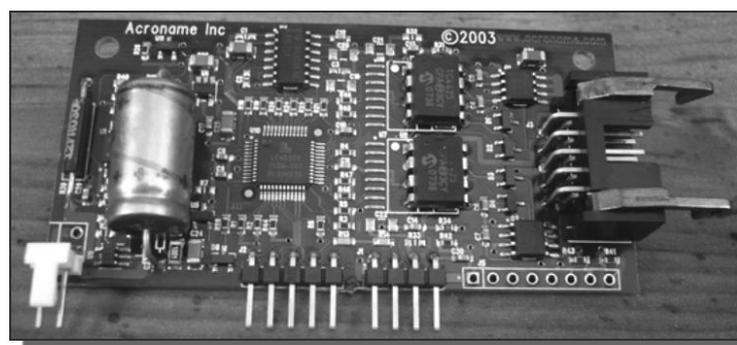


Fig. 5.14 Puente H del robot García

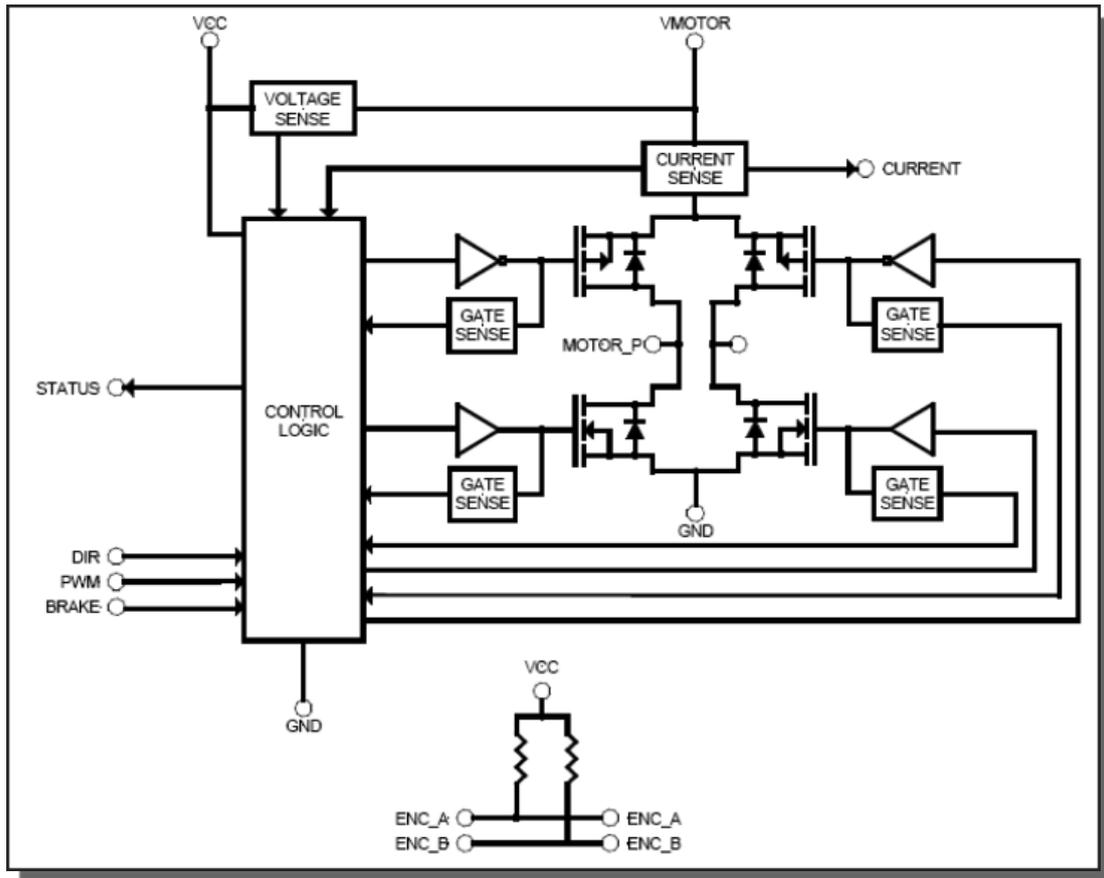


Fig. 5.15 Diagrama a bloques del puente H

Algunas de las características de este circuito son:

- Voltaje de operación con baterías de 3.6 V a 12 V.
- Entrega una salida de hasta 3.5 A de corriente continua.
- Resistencia de “encendido” de menos de 250 m Ω .
- Compatible con entradas TTL y CMOS.
- Salida para monitorear corriente en forma analógica.
- Salida digital del estado.
- Salida digital de freno.
- Protección contra sobre corriente ó sub voltaje.
- Conexiones separadas para la alimentación de los circuitos lógicos y para el motor.
- Paso de encoders de cuadratura a través de resistencias pull-up.
- Conector integrado que combina motor y encoder.
- Operación de PWM hasta 50 kHz.

La figura 5.16 muestra la disposición de los pines de la tarjeta:

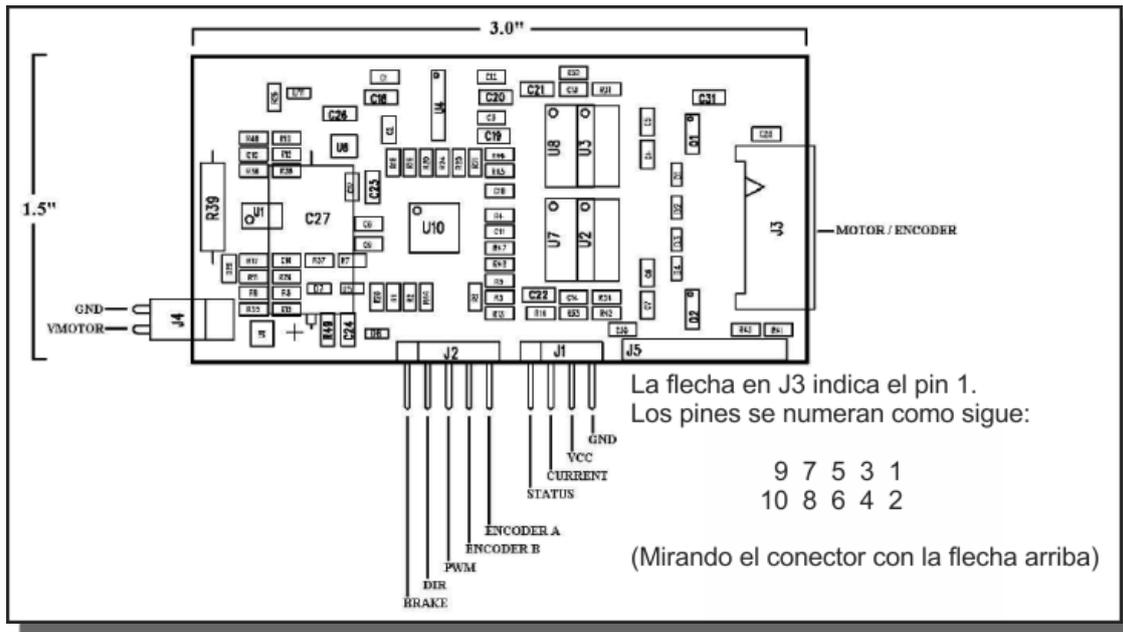


Fig. 5.16 Descripción de pines del puente H

La tabla 5.1 corresponde a la lógica de control, mientras que la descripción de los pines se muestra en la tabla 5.2

Entradas			Salidas		Función
BRAKE	DIR	PWM	MOTOR_P	MOTOR_N	
0	0	0	VMOTOR	VMOTOR	Freno
0	0	1	GND	VMOTOR	Reversa
0	1	0	VMOTOR	VMOTOR	Freno
0	1	1	VMOTOR	GND	Adelante
1	0	0	FLOAT	FLOAT	Giro libre
1	0	1	GND	GND	Freno
1	1	0	FLOAT	FLOAT	Giro libre
1	1	1	VMOTOR	VMOTOR	Freno
X	X	X	FLOAT	FLOAT	STATUS = 0

Tabla 5.1 Lógica de control del puente H

Pin	Nombre de la señal	Tipo	Descripción
J1-1	STATUS	Salida digital	STATUS está activo (uno lógico) cuando la tarjeta está funcionando adecuadamente. Se vuelve inactivo (cero lógico) debido a una sobre corriente ó a una condición de voltaje bajo. Cuando esto último sucede la tarjeta se pone en un estado flotante (giro libre).
J1-2	CURRENT	Salida analógica	CURRENT es una señal de voltaje analógico que corresponde a la cantidad de corriente que fluye a través del puente. Las unidades son 1 V / A.
J1-3	VCC	Entrada de alimentación	VCC es nominalmente 5 V.
J1-4	GND	Tierra	Tierra lógica de referencia.
J2-1	BRAKE	Entrada digital	Cuando BRAKE está en cero lógico y PWM está en cero lógico el motor se frena. Si PWM está en uno lógico el motor gira hacia un sentido u otro dependiendo de la entrada DIR.
J2-2	DIR	Entrada digital	Esta entrada controla la dirección de giro del motor.
J2-3	PWM	Entrada digital	La entrada PWM controla cuando se aplica un voltaje al motor.
J2-4	ENC_B	Salida digital	ENC_B es un paso del canal B del encoder de cuadratura desde la conexión del motor. Hay una resistencia de pull-up de 4.75 K Ω entre esta señal y VCC.
J2-5	ENC_A	Salida digital	ENC_A es un paso del canal A del encoder de cuadratura desde la conexión del motor. Hay una resistencia de pull-up de 4.75 K Ω entre esta señal y VCC.
J3-1	MOTOR_P	Salida analógica	MOTOR_P debe conectarse al motor de tal forma que cuando se aplique un voltaje positivo con respecto a MOTOR_N, el motor gire hacia adelante.
J3-2	VCC	Salida de alimentación	VCC se usa para alimentar al encoder.
J3-3	ENC_A	Entrada digital	ENC_A es el canal A del encoder de cuadratura.
J3-4	ENC_B	Entrada digital	ENC_B es el canal B del encoder de cuadratura.
J3-5	GND	Tierra	Tierra lógica para ser usada por el encoder.
J3-6	MOTOR_N	Salida analógica	MOTOR_N debe conectarse al motor de tal forma que cuando se aplique un voltaje positivo con respecto a MOTOR_P, el motor gire hacia atrás.
J4-1	VMOTOR	Entrada de alimentación del motor	VMOTOR es el voltaje del motor.
J4-2	GND	Tierra del motor	GND es la tierra del voltaje del motor, se conecta a la tierra lógica en la placa.

Tabla 5.2 Descripción de los pines del puente H

5.3 Tarjeta de conexión de los motores y encoders

Los motores / encoders incluyen un cable plano de 6 hilos, que termina en un conector tipo IDC de 10 terminales (figura 5.17). Cuando se quiere hacer la conexión a la tarjeta construida, que incluye los puentes H y que se describe en la siguiente sección, estos cables resultan muy cortos. Su tamaño, junto con su manipulación, pueden provocar que estos se dañen y reemplazarlos sería complicado.

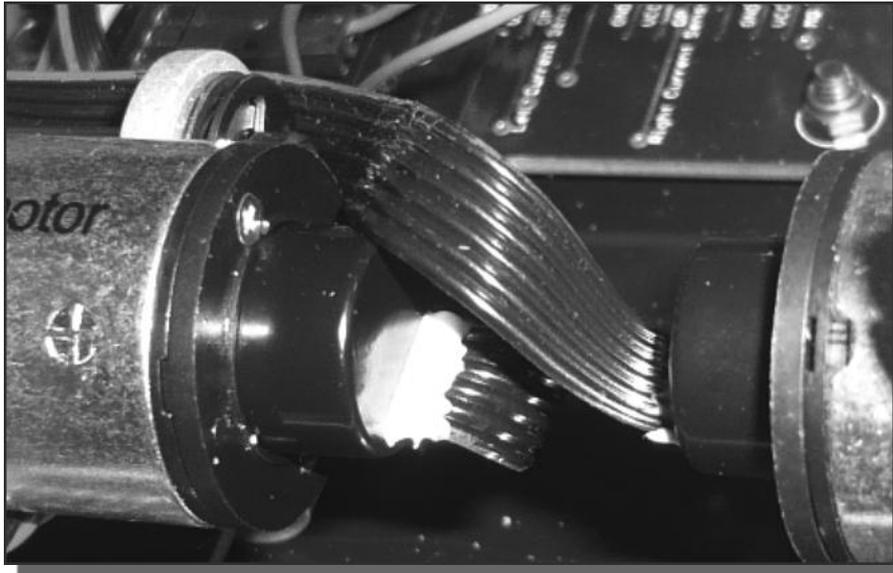


Fig. 5.17 Cable de conexión de motor / encoder

Debido a esto, se construyó una placa de circuito impreso que evita que estos cables se estén moviendo, y facilita la conexión a la placa de los puentes H (figura 5.18).

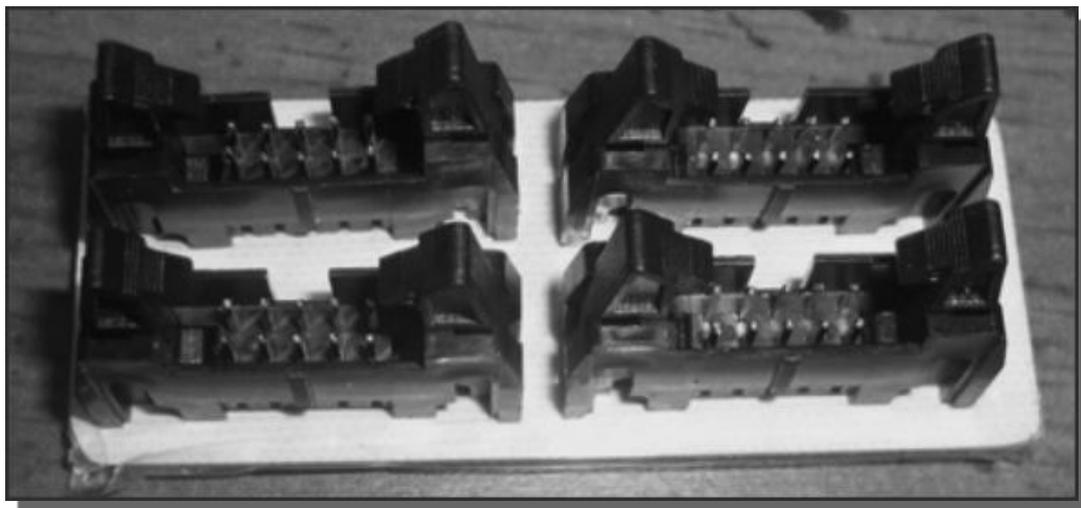


Fig. 5.18 Placa de conexión de los motores / encoders

La figura 5.19 corresponde al diseño de la placa impresa, en el software E.A.G.L.E.

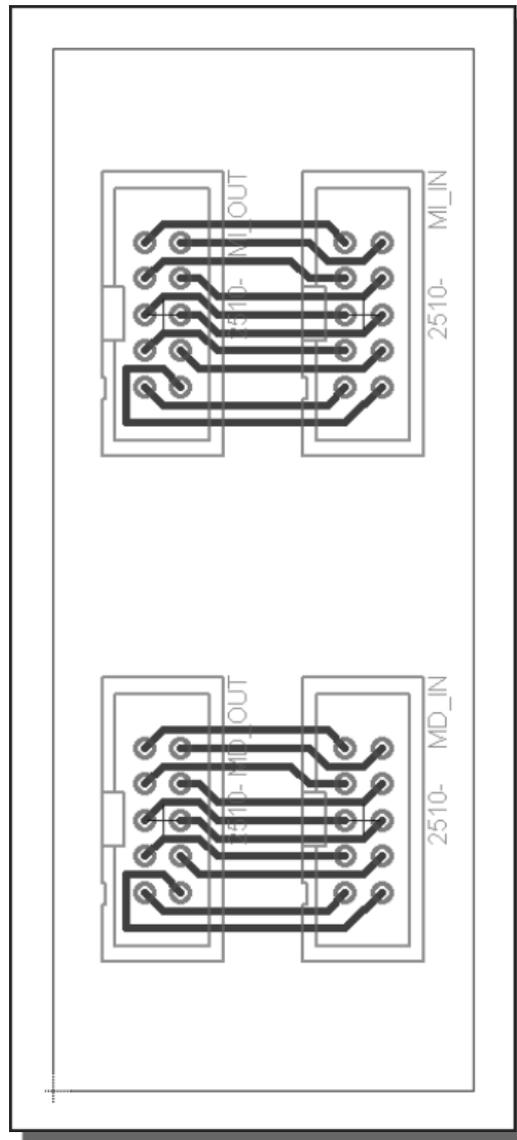


Fig. 5.19 Diseño de la placa para la conexión de los motores / encoders

5.4 Modulación por ancho de pulsos (PWM), tarjeta con microcontrolador PIC16F887 y puentes H

La modulación por ancho de pulsos (PWM – Pulse Width Modulation) permite variar la potencia suministrada a un circuito eléctrico, como se muestra en la figura 5.20. Si un cero lógico (0) representa al interruptor apagado y un uno lógico (1) representa el interruptor encendido, la energía que la carga consume será directamente proporcional a la duración del pulso. El tiempo que la señal permanece en uno lógico se le conoce como el ciclo útil.

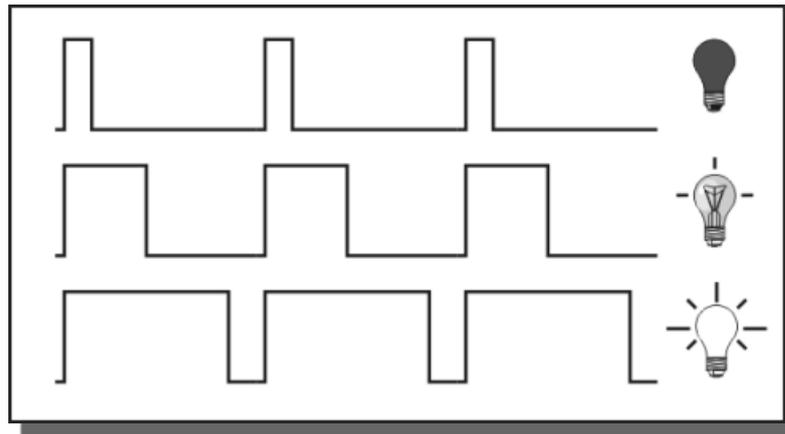


Fig. 5.20 Variación de potencia mediante PWM

En nuestro caso, la señal PWM se utiliza para variar la velocidad de los motores. Los puentes H requieren esta señal PWM para operar. Puede notarse una equivalencia entre el ciclo útil de la señal y el voltaje aplicado a los motores, lo cual produce una velocidad que es proporcional al ciclo útil aplicado (figura 5.21).

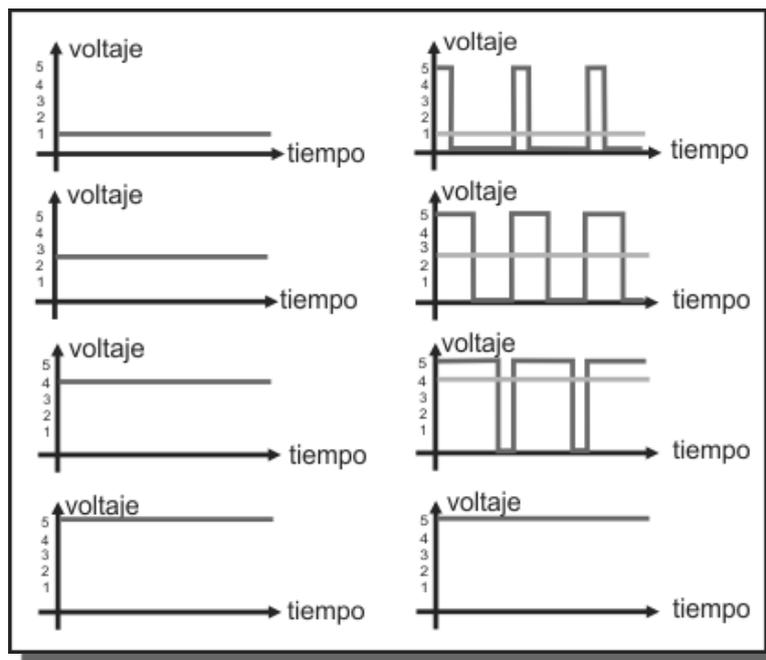


Fig. 5.21 Equivalencia entre el voltaje y el ciclo útil de la señal PWM

Para generar los voltajes PWM se utilizó un microcontrolador de la empresa Microchip el PIC16F887 (figura 5.22). Algunas características de este dispositivo son las siguientes:

- 35 instrucciones para programación en ensamblador.
- Velocidad de operación de DC a 20 MHz.
- Capacidad de manejo de interrupciones.
- Modos de direccionamiento directo e indirecto.

- Oscilador interno calibrado a $\pm 1\%$.
- Modo de ahorro de energía (Sleep).
- Voltaje de operación de 2 V – 5.5 V.
- Memoria EEPROM de 256 bytes que permite un millón de ciclos de escritura y retiene los datos cerca de 40 años.
- Memoria de programa de 8192 localidades de 14 bits.
- 35 pines de entrada / salida digitales.
- Módulo con dos comparadores analógicos.
- 14 módulos convertidores A / D con 10 bits de resolución.
- 2 temporizadores / contadores de 8 bits y uno de 16 bits.
- 2 Módulos PWM con resolución de hasta 10 bits.
- Módulos de comunicación serie RS-485, RS-232 y LIN 2.0.

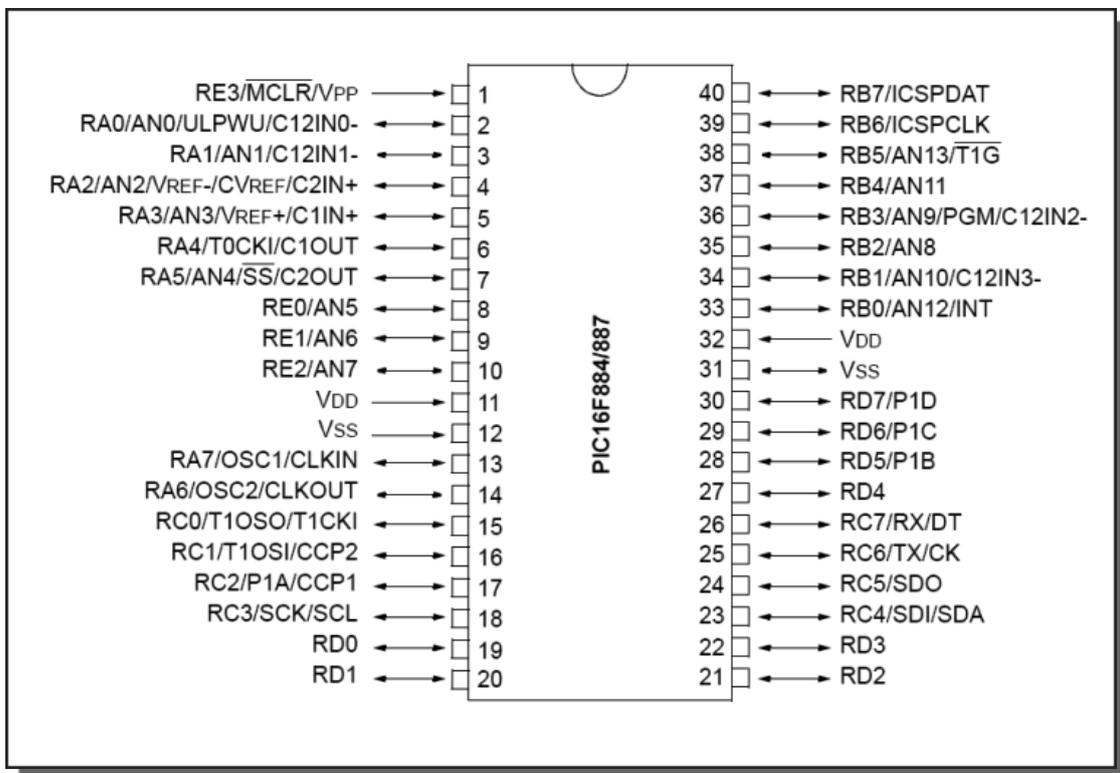


Fig. 5.22 Disposición de los pines del microcontrolador PIC16F887

La tarjeta diseñada para alojar al microcontrolador (figura 5.23) tiene las siguientes características:

- Cuenta con una terminal para la conexión de la batería del robot.
- Incluye un circuito integrado LM7806CV que entrega los 6 V nominales a los motores del robot.
- Se incluye el microcontrolador PIC16F887 junto con un cristal de 4 MHz como reloj de este circuito.
- Un conector DB15 envía / recibe los siguientes datos hacia / desde la tarjeta de adquisición de datos:

- Envío:
 - Señal CURRENT de los dos puentes H.
 - Canales A y B de los dos encoders.
- Recepción:
 - Señal de +5V para alimentar el microcontrolador.
 - Dos señales digitales para el sentido de giro de los motores.
 - Dos señales analógicas (0-5V), que son las consignas de control para los motores.
- LED que indica la presencia de los 5 V para la alimentación del microcontrolador y la lógica de los puentes H.
- LED que indica la presencia de los 6 V para alimentación de los motores.
- Se considera el apagado de los motores cuando se tiene una sobre corriente o un voltaje insuficiente para los motores.
- Se incluyen dos LEDS que indican cuando se han apagado los motores debido a la condición de sobre corriente ó voltaje insuficiente, estos LEDs se pueden programar para mostrar otras señales.

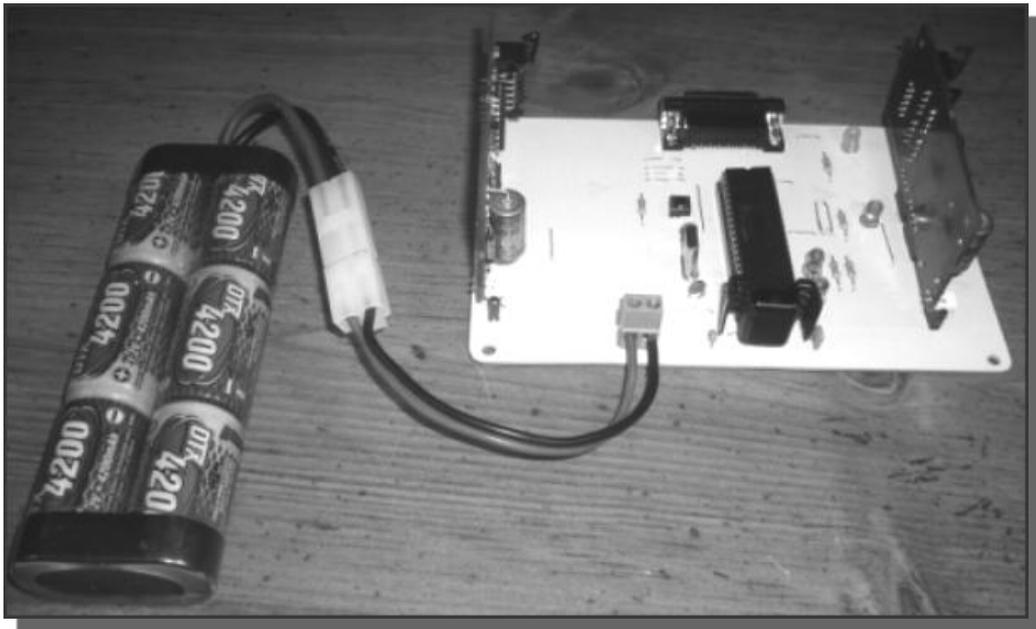


Fig. 5.23 Tarjeta con PIC16F887, puentes H y regulador de 6 V

La figura 5.24 corresponde al esquema electrónico de esta placa.

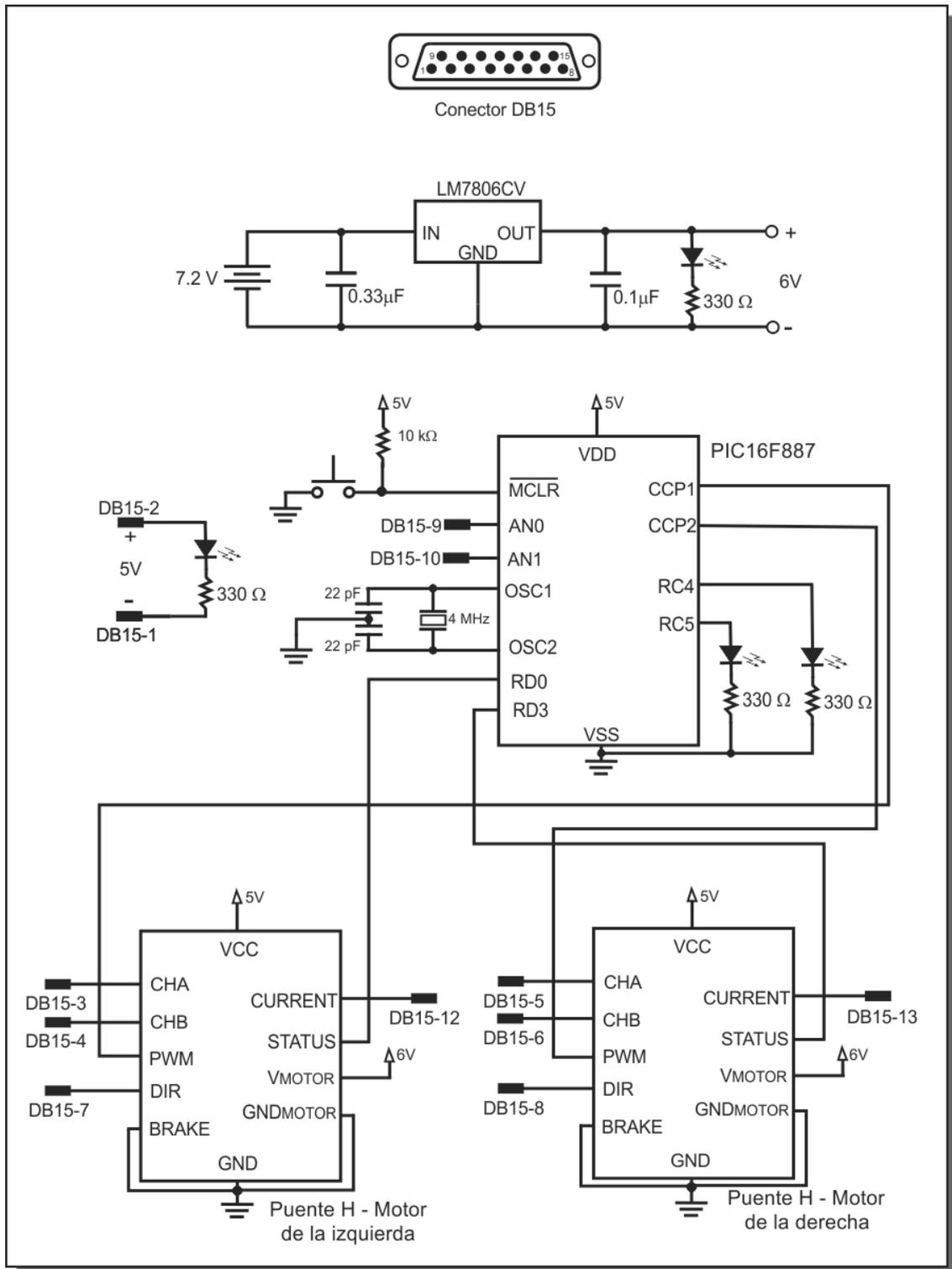


Fig. 5.24 Esquema electrónico de la placa con PIC16F887, puentes H y regulador

El diseño en E.A.G.L.E. se muestra en la figura 5.25, una foto del robot con la tarjeta montada se puede ver en la figura 5.26.

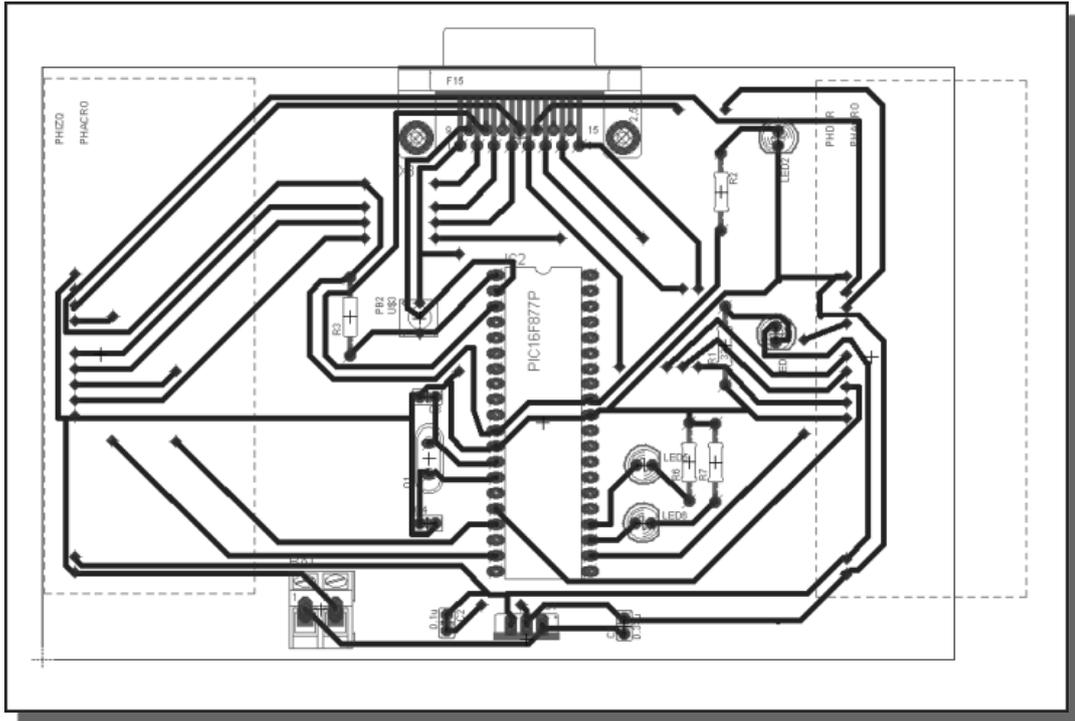


Fig. 5.25 Diseño de la placa con PIC16F887, puentes H y regulador de voltaje

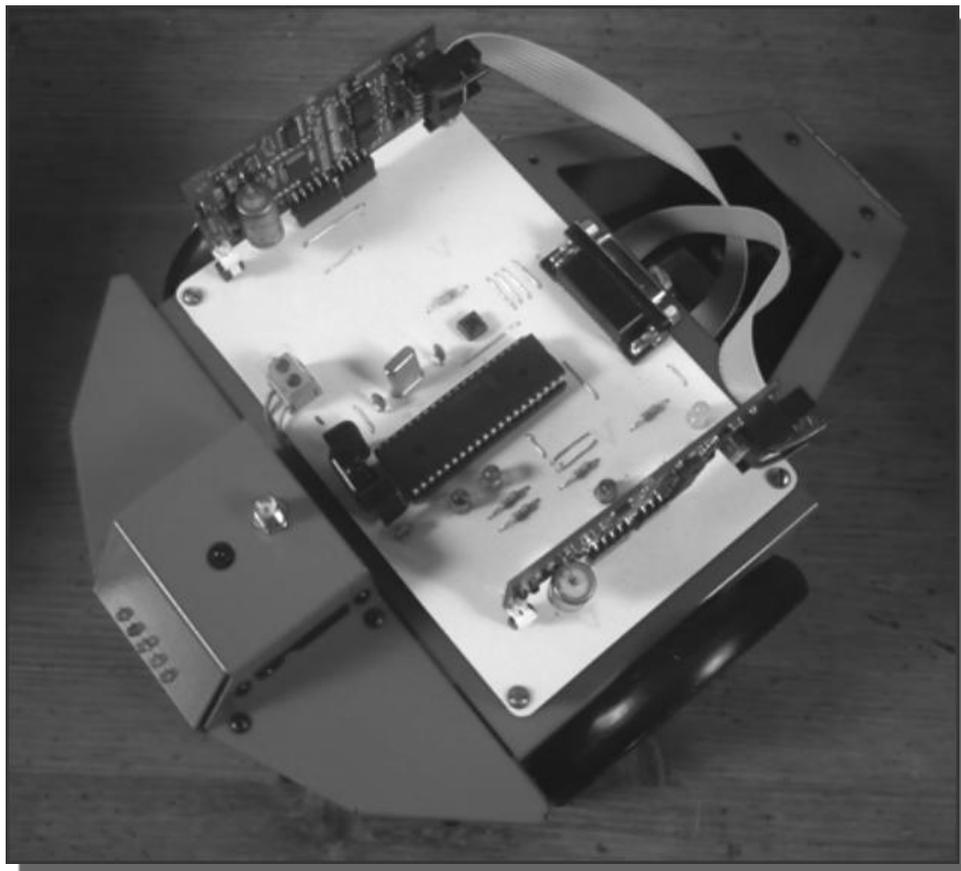


Fig. 5.26 Robot García con la placa montada en la parte superior

5.5 Programa en el compilador mikroC para el PIC16F887

La programación de los microcontroladores se puede realizar en lenguaje ensamblador, o se puede aprovechar la ventaja de compiladores de alto nivel como Basic, Pascal ó C. En este caso se utilizó el compilador mikroC de la empresa mikroElektronika. El programa se muestra a continuación:

```
// Programa para generar señales PWM para los motores y proteger los mismos de
// condiciones de sobre corriente o voltaje insuficiente.
char estadoI = 1;
char estadoD = 1;
unsigned int temp_resI;
unsigned int temp_resD;
unsigned short ciclo_utilI = 0;
unsigned short ciclo_utilD = 0;

void main()
{
    ANSEL = 0x03;           // AN0 y AN1 son analógicas
    TRISA = 0xFF;          // El puerto A es de entrada
    ANSELH = 0;            // los otros pines AN se configuran como
                           // I/O digitales
    TRISC = 0;             // El puerto C es de salida
    TRISD = 0xFF;          // El puerto D es de entrada

    PWM1_Init(300);        // se inicializa el módulo PWM1
                           // a una frecuencia de 300 Hz
    PWM2_Init(300);        // se inicializa el módulo PWM2
                           // a una frecuencia de 300 Hz

    PWM1_Start();
    PWM2_Start();

    while ( estadoI == 1 && estadoD == 1 )
    {
        if ( PORTD.F0 == 0 ) // Si el pin STATUS del puente H de la izquierda
        {                    // es cero - condición de alta corriente
                               // o bajo voltaje
            estadoI = 0;
        }

        if( PORTD.F3 == 0 ) // Si el pin STATUS del puente H de la derecha
        {                    // es cero - condición de alta corriente
                               // o bajo voltaje
            estadoD = 0;
        }

        temp_resI = ADC_Read(0);
        temp_resD = ADC_Read(1);

        ciclo_utilI = temp_resI / 4;
        ciclo_utilD = temp_resD / 4;

        PWM1_Set_Duty( ciclo_utilI );
        PWM2_Set_Duty( ciclo_utilD );

    }
    PWM1_Set_Duty( 0 );
    PWM2_Set_Duty( 0 );
}

```

5.6 Tarjeta de adquisición de datos Sensoray 626

Como parte del sistema de control se utilizó una tarjeta de adquisición de datos de la empresa Sensoray, el modelo es 626 y se instala en una computadora de escritorio a través de un puerto PCI (figura 5.27). La tarjeta Sensoray 626 tiene características que resultan de gran ayuda para implementar un sistema de control, además de la ventaja de que puede ser usada desde el software Simulink. Algunas de sus características son:

- 48 Canales digitales I / O.
 - 20 de estos canales digitales tienen detección de flancos y capacidad de interrupción.
 - 7 de estas salidas digitales se pueden usar como salidas de desborde de contador.
 - Los conectores digitales I / O son pines de salida estándar en la industria.
- 6 contadores de 24 bits arreglados en 3 pares que pueden usarse para adquirir las señales de hasta 6 encoders ópticos.
- 16 entradas analógicas con 14 bits de resolución.
- 4 salidas analógicas (13 bits de resolución).
- Protección de los puertos de la tarjeta.



Fig. 5.27 Tarjeta de adquisición de datos Sensoray modelo 626

En la figura 5.28 se muestra un diagrama a bloques de la tarjeta, y en la 5.29 la disposición de los componentes, ambas figuras obtenidas del manual del fabricante.

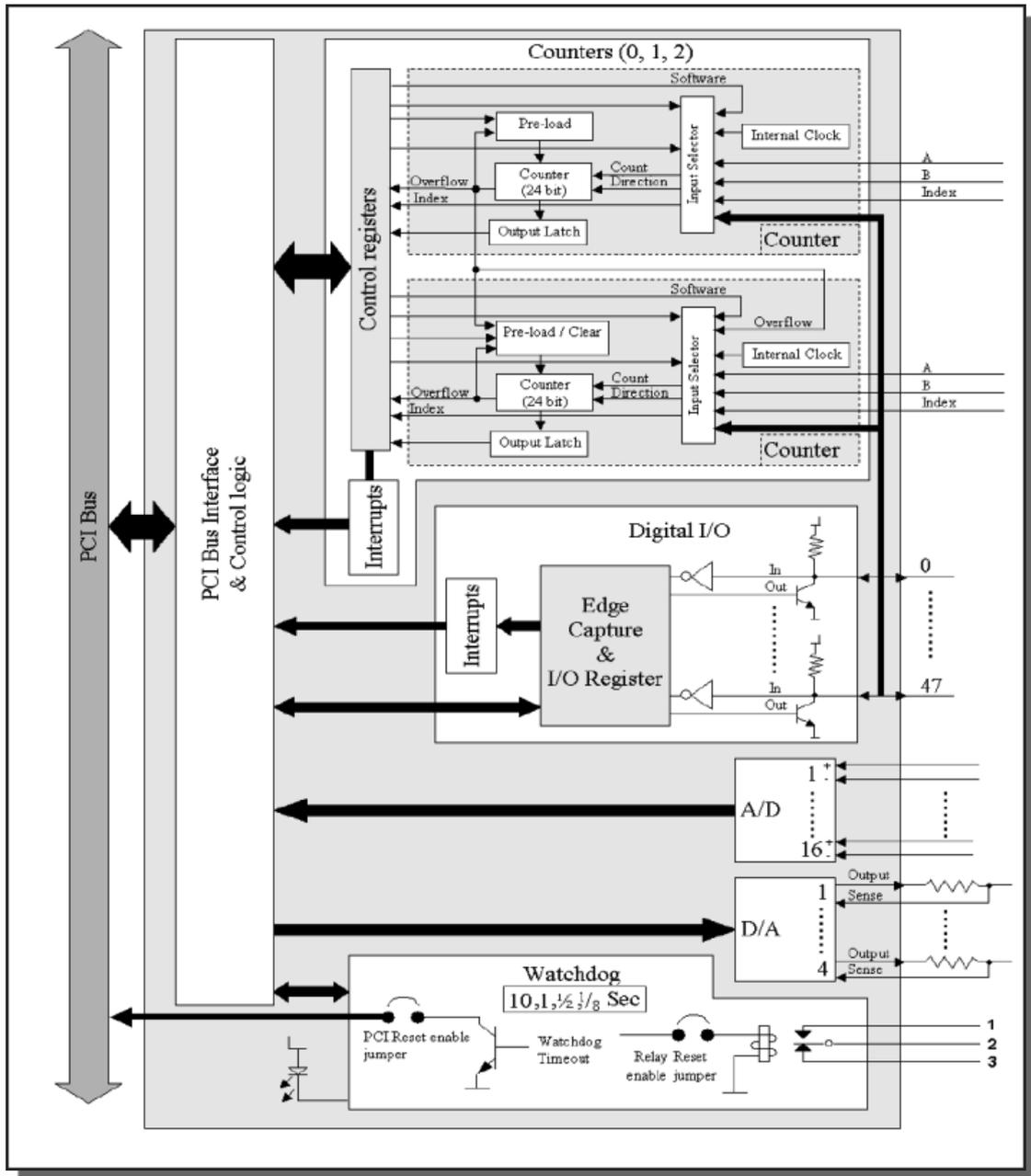


Fig. 5.28 Diagrama a bloques de la tarjeta de adquisición de datos

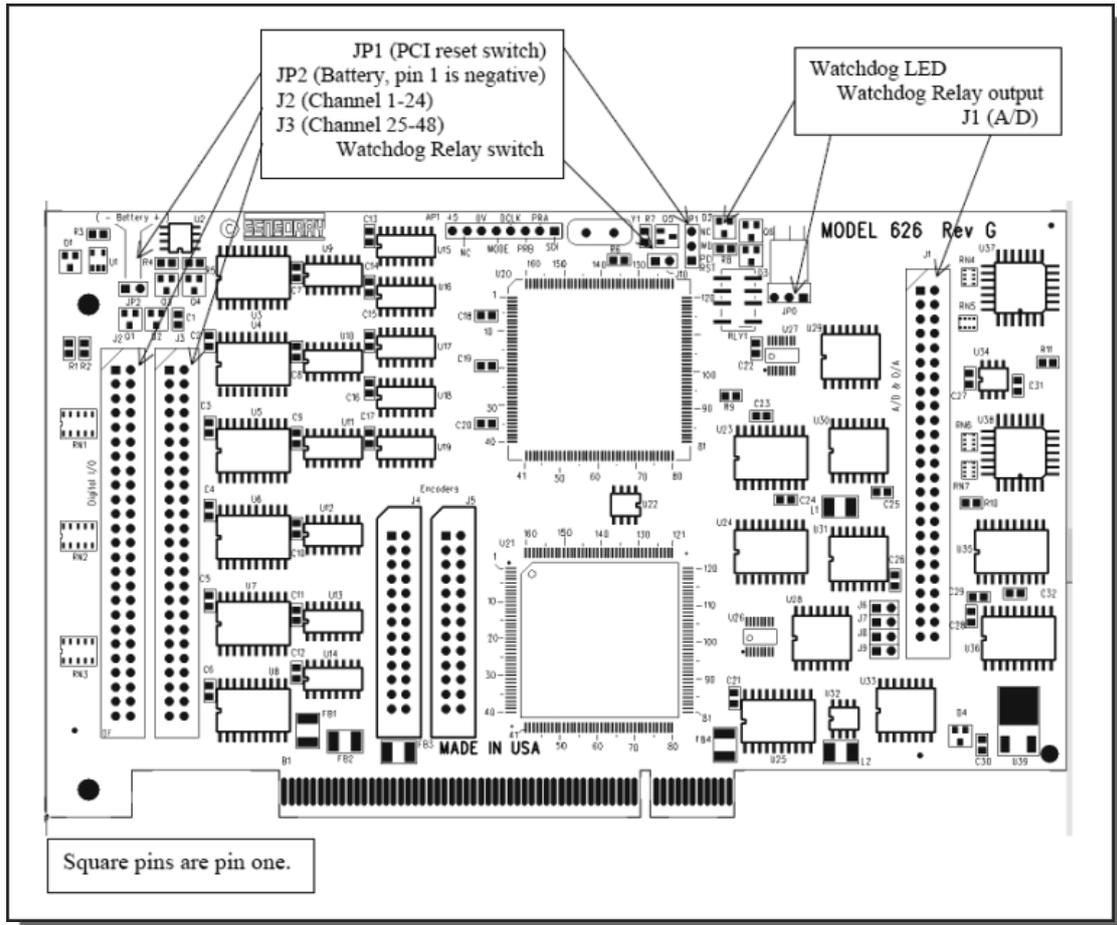


Fig. 5.29 Disposición de componentes de la tarjeta de adquisición de datos

5.7 Conexión entre la tarjeta de adquisición de datos y el robot

La conexión entre la tarjeta de adquisición de datos y la placa a bordo del robot se hace a través de un cable plano de 14 hilos, elegido de este tipo debido a su flexibilidad. Para conectar el cable a las tarjetas se usan conectores tipo DB15.

Se creó una tarjeta para concentrar en un conector DB15 las señales que se están utilizando, de la tarjeta de adquisición de datos: conector analógico (IDC50), conector digital (IDC50) y conector de encoders (IDC26), y así ser enviadas por un solo cable a la tarjeta del robot.

La figura 5.30 corresponde al esquemático de la tarjeta. La tarjeta de adquisición de datos requiere una entrada negada para las señales de los encoders, por esto se incluyó el circuito integrado 4069 con 6 inversores lógicos.

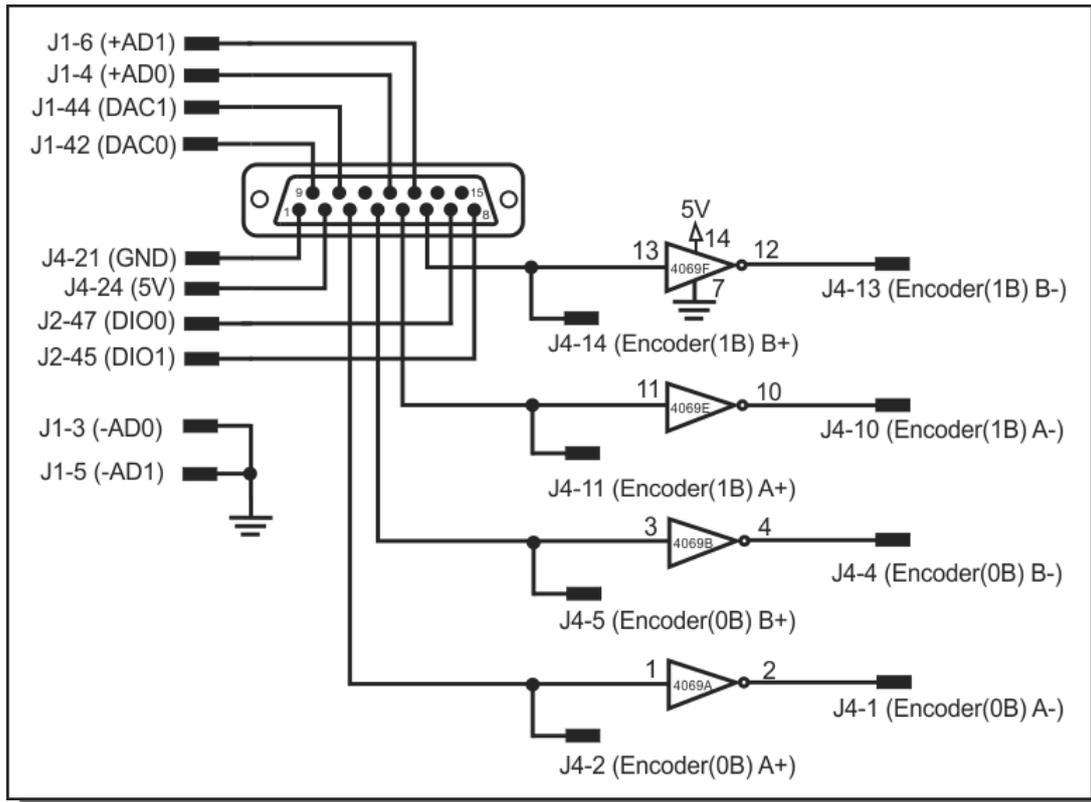


Fig. 5.30 Esquema electrónico de la tarjeta de conexión entre la Sensoray 626 y el robot

La figura 5.31 corresponde a una foto de esta tarjeta, el diseño en E.A.G.L.E. se muestra en la figura 5.32

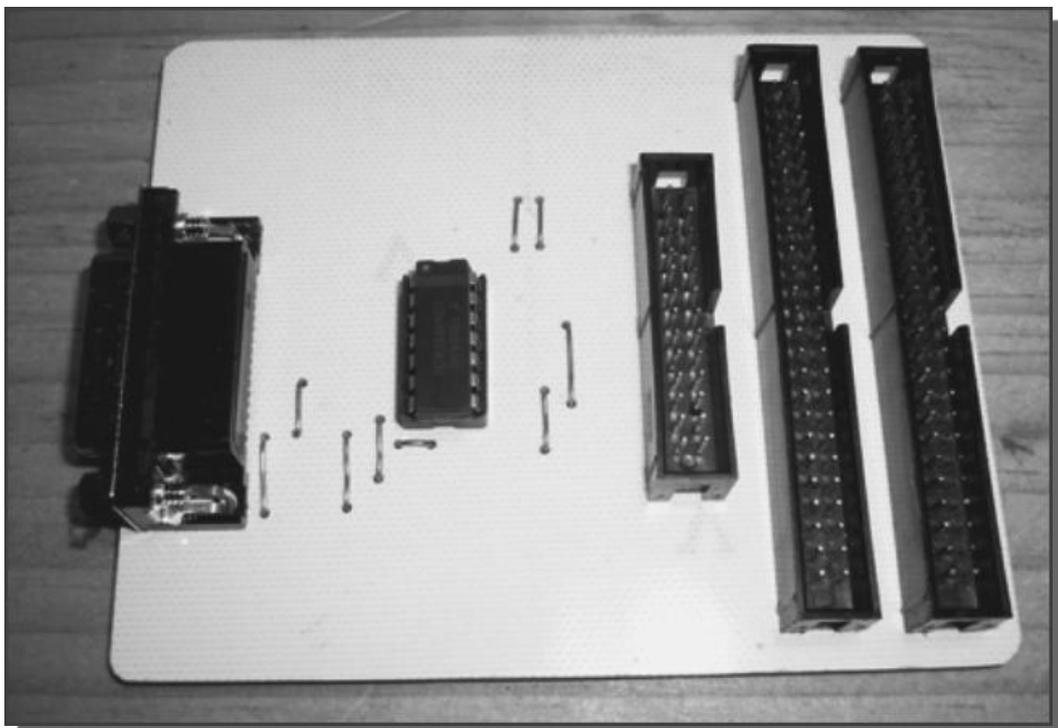


Fig. 5.31 Aspecto físico de la tarjeta de conexión

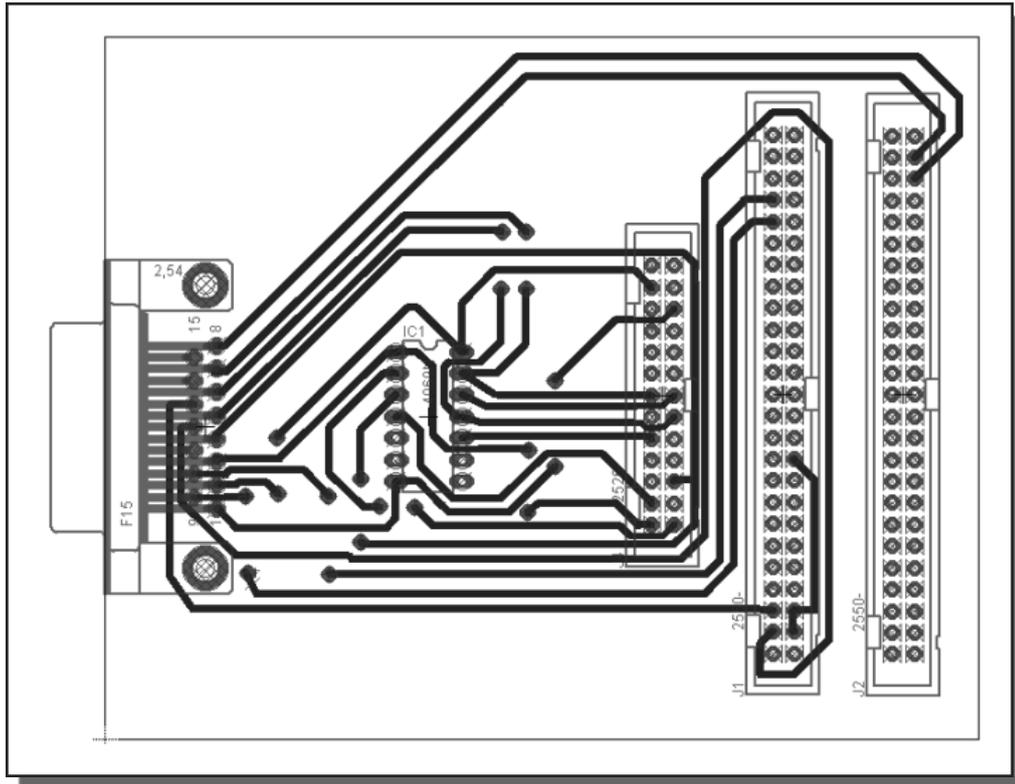


Fig. 5.32 placa de circuito impreso diseñada en E.A.G.L.E.

CAPÍTULO 6

Implementación y pruebas

6.1 Implementación de sistema de visión para la localización del robot

Como se vio en el capítulo 2, la posición y orientación del robot se puede conseguir a través del uso de los encoders ópticos. Esta forma de localización del robot presenta problemas debido a que si el piso no es totalmente plano se generan errores en el conteo de los encoders que poco a poco, se van acumulando y se van generando errores cada vez mayores.

Se hicieron pruebas utilizando los encoders, pero en el laboratorio el piso está formado por mosaicos lo que produce problemas en las uniones, además, la resolución de los encoders no es muy grande por lo que los errores fueron apreciables.

Los esquemas de localización absoluta proporcionan la posición y orientación del robot en cada instante de tiempo, y sin importar donde inició ó donde se localizaba unos instantes antes. Una forma de implementar un esquema de localización absoluta se basa en el uso de sensores ultrasónicos, en el que se monta un emisor en el robot y se colocan 3 ó 4 receptores alrededor del robot delimitando el área de trabajo. Con base en los tiempos que tarda la señal en llegar del emisor a los receptores se puede inferir la posición del robot.

Otra opción es el uso de una cámara de video que capture cada determinado tiempo una imagen del área de trabajo en donde se encuentra el robot; cada una de estas imágenes se procesa mediante un algoritmo para obtener la posición y además la orientación. En este trabajo se optó por este último enfoque, para lo cual se utilizó una cámara web y la ayuda de las herramientas de adquisición y procesamiento de imágenes de MATLAB / Simulink.

La cámara utilizada fue de la marca Logitech diseñada para laptops (figura 6.1), por lo que su tamaño es reducido; su resolución máxima es de 640 x 480 pixeles en formato RGB. La conexión de la cámara a la computadora es a través de un cable USB.



Fig. 6.1 Cámara web para la localización del robot en el plano

Al montar la cámara se obtuvo un área de trabajo de aproximadamente 2.44 m x 1.79 m (figura 6.2). Al principio se utilizó una resolución de 640 x 480 píxeles, pero para tener más datos por unidad de tiempo se usó la resolución de 320 x 240 píxeles. La misma figura muestra como se asignaron los ejes X-Y con base en estos datos:

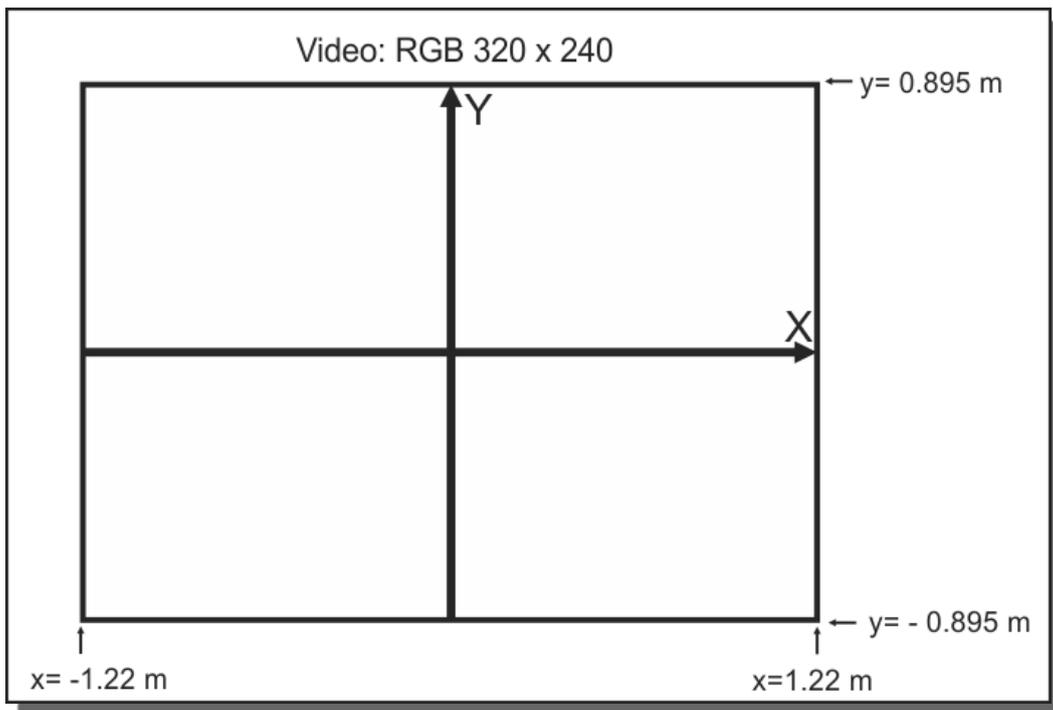


Fig. 6.2 Área de trabajo del robot con base en el área cubierta por la cámara

En este caso se detectó el color negro sobre fondo blanco, por lo que al robot se le colocó encima una cubierta blanca, y sobre ella se colocaron dos círculos que se utilizaron para la detección y cálculo de la posición y la orientación (figura 6.3).

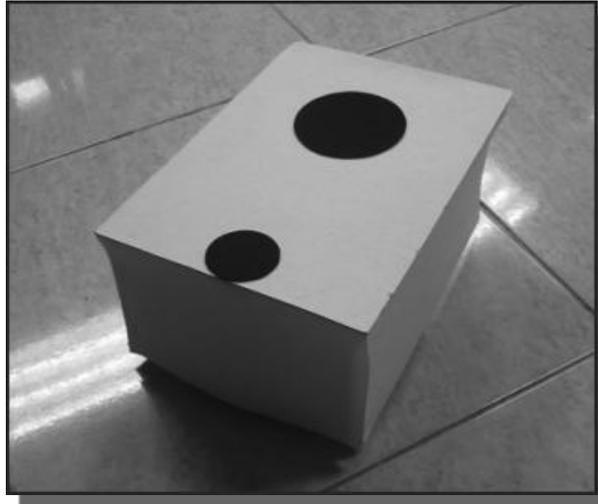


Fig. 6.3 Robot cubierto para la correcta detección de la cámara

Al querer implementar en tiempo real la aplicación de la cámara se presentó el problema de que las herramientas de adquisición y procesamiento de imágenes de MATLAB / Simulink no están definidas en los procesos de tiempo real. La solución fue adquirir los datos de la cámara en una computadora y enviarlos vía puerto serie a la computadora con la tarjeta de adquisición de datos.

El programa en Simulink se muestra en la figura 6.6. Este programa adquiere una imagen de la cámara Logitech (figura 6.4), posteriormente la imagen adquirida se pasa a través de un bloque que detecta los objetos en color negro en el área de la cámara (en este caso dos círculos) y produce una imagen “binarizada”, que es una imagen en donde se resaltan los objetos encontrados (objetos en color blanco sobre un fondo negro), esto se hace a través de un umbral de detección del color (figura 6.5).

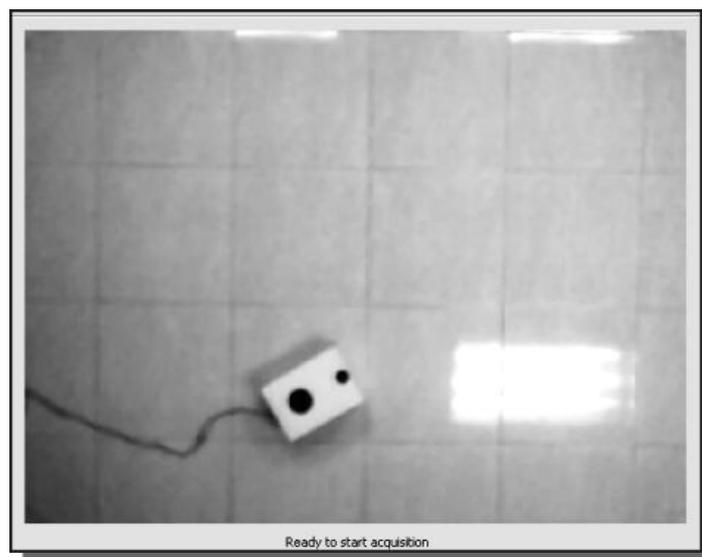


Fig. 6.4 Imagen adquirida

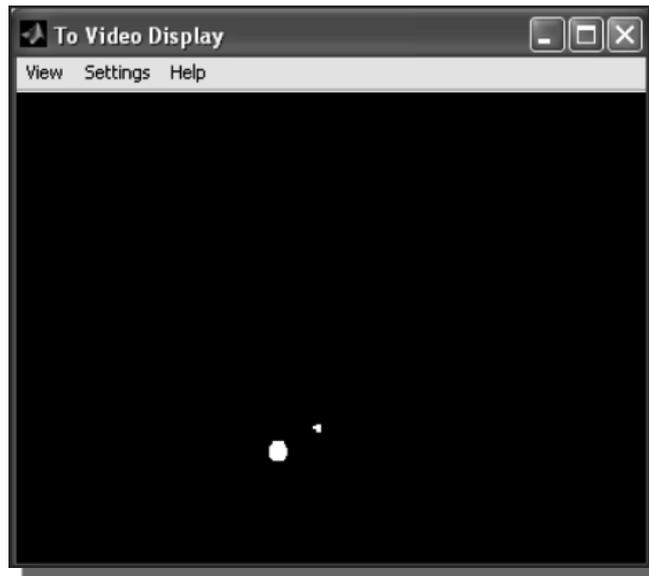


Fig. 6.5 Imagen binarizada

El bloque marcado como “Obtención de posición y orientación” (figura 6.6) llama al siguiente programa:

```
function postura = localizacion (u)

global xant;
global yant;
global thant;

planeG = bwlabeln(u,26);

a = regionprops(planeG,'Area');
c = regionprops(planeG,'Centroid');

if a(1).Area < a(2).Area
    xp1 = c(2).Centroid(1);
    yp1 = c(2).Centroid(2);

    xp2 = c(1).Centroid(1);
    yp2 = c(1).Centroid(2);
else
    xp1 = c(1).Centroid(1);
    yp1 = c(1).Centroid(2);

    xp2 = c(2).Centroid(1);
    yp2 = c(2).Centroid(2);
end

xM = ( xp1 - 160 ) * 2.44 / 320;
yM = ( 240 - yp1 - 120 ) * 1.79 / 240;

xm = ( xp2 - 160 ) * 2.44 / 320;
ym = ( 240 - yp2 - 120 ) * 1.79 / 240;

x = (xm + xM)/2;

y = (ym + yM)/2;
```

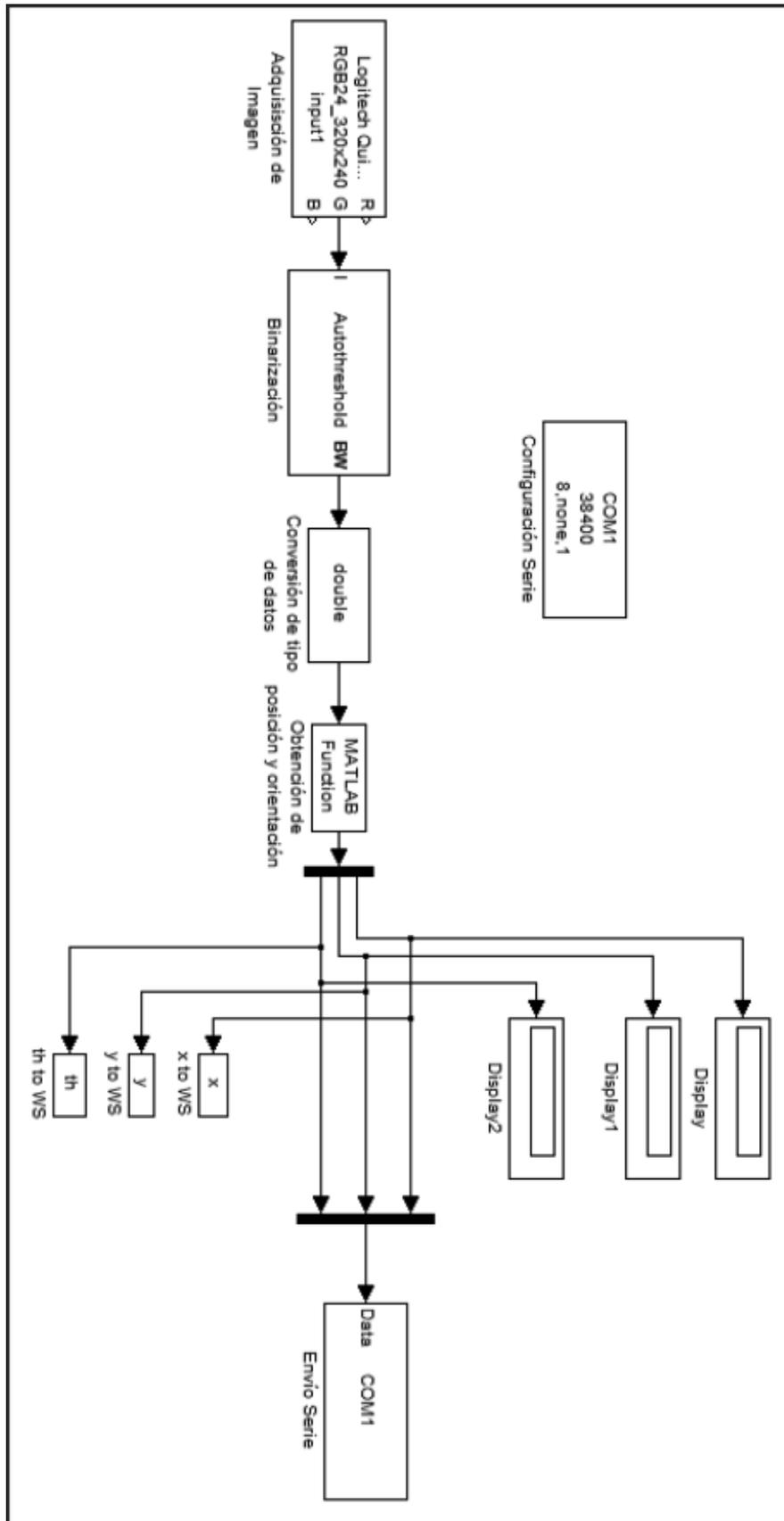


Fig. 6.6 Programa para captura y envío de la posición y orientación del robot

```

th = atan2( ym - yM, xm - xM );

if th < 0
    th = th + 2*pi;
end

if a(1).Area < 8 || a(2).Area < 8
    x = xant;
    y = yant;
    th = thant;
end

xant = x;
yant = y;
thant = th;

postura = [x y th];

```

Para la obtención de la posición y la orientación se utilizan funciones para determinar el área y el centroide de los objetos localizados; se identifica el área menor, que corresponde al frente del robot, y el área mayor. Después, se realiza la conversión de la localización del centroide de píxeles a milímetros. Una vez hecho esto se obtiene un promedio de los dos centroides que define el punto medio de las dos ruedas del robot (coordenadas x e y).

Para la orientación (ángulo θ) se usan los centroides de las dos áreas y la función arco tangente. Dicha función entrega el ángulo de $-\pi$ a π ; si la trayectoria establecida requiere el ángulo de 0 a 2π se debe hacer una conversión. Además se convierte el valor de radianes a grados.

Se considera también el caso en que se detectan objetos diferentes a los círculos (debidos generalmente a sombras y con un área menor a los círculos), estos se excluyen porque generan datos erróneos.

6.2 Pruebas con los motores del robot

De forma ideal, la respuesta de un motor (voltaje – velocidad) es lineal. Al realizar pruebas con los motores del robot se encontró que se tenía una respuesta que no era exactamente una línea recta, por lo que se procedió a hacer una caracterización de las velocidades de los mismos aplicando la señal PWM generada con el microcontrolador.

Se aplicaron también factores de conversión para tener la velocidad angular de las ruedas en rad/s, que es como se maneja en los esquemas de control de las simulaciones.

Para las pruebas se conectó un canal del encoder a un canal de entrada del osciloscopio y se aplicó la señal PWM con ciclos útiles de 0 al 100 % (figuras 6.7 y 6.8).

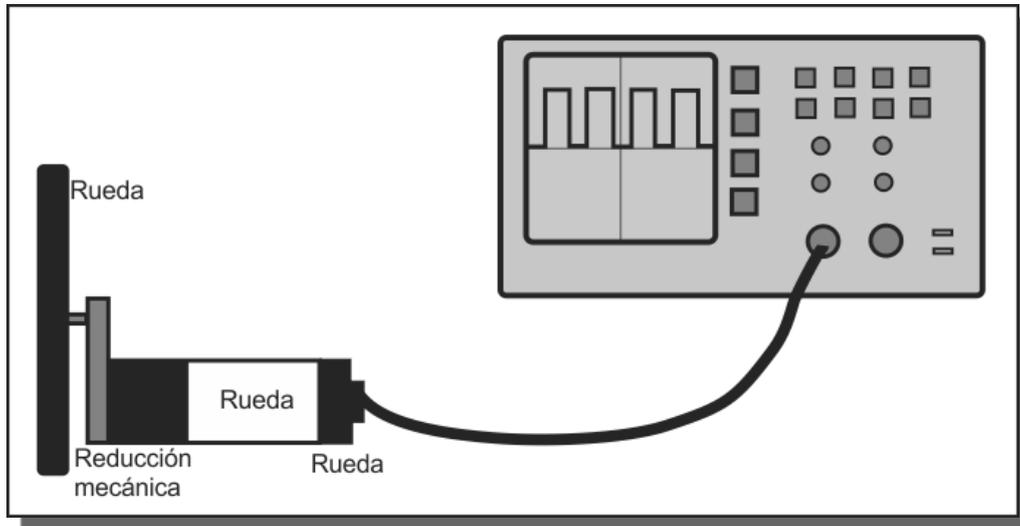


Fig.6.7 Experimento para caracterizar las velocidades de las ruedas

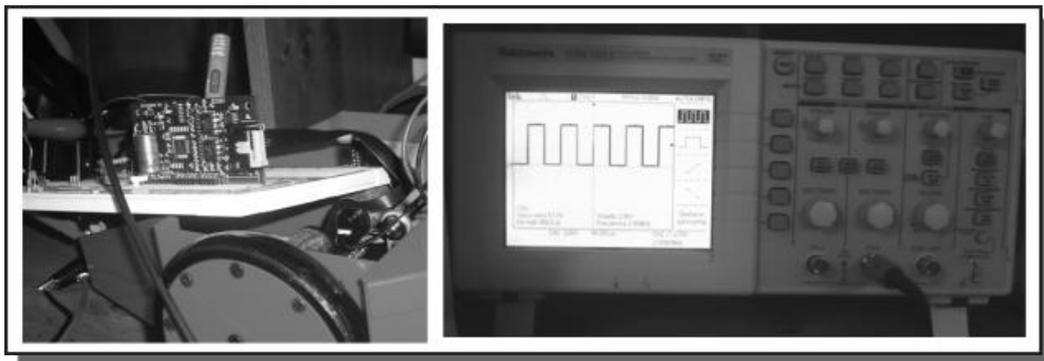


Fig.6.8 Fotos del experimento

Ejemplo de cálculo:

Para una frecuencia medida de la señal del encoder, $f = 2.414 \text{ kHz}$.

Periodo:

$$T = \frac{1}{f} = \frac{1}{2414 \text{ Hz}} = 414.25 \mu\text{s}$$

Considerando que los encoders son de 16 cuentas por revolución, el tiempo de una revolución es:

$$t_{rev} = 16T = 16 \times 414.25 \times 10^{-6} \text{ s} = 6.628 \text{ s} \left(\frac{1 \text{ min}}{60 \text{ s}} \right) = 110.47 \times 10^{-6} \text{ min}$$

Velocidad angular del motor en rpm:

$$\omega_{motor} = \frac{1 \text{ rev}}{110.47 \times 10^{-6} \text{ min}} = 9052.5 \frac{\text{rev}}{\text{min}}$$

Para obtener la velocidad en la rueda se debe de considerar la reducción del tren de engranajes del motor (19:1) y la reducción adicional que va a la rueda (3:1), lo que da una reducción total de 57:1. La velocidad angular de la rueda en rpm es:

$$\omega_{rueda} = \left(9052.5 \frac{\text{rev}}{\text{min}}\right) \left(\frac{1}{57}\right) = 158.82 \frac{\text{rev}}{\text{min}}$$

Finalmente, la velocidad de la rueda en rad/s está dada por:

$$\omega_{rueda} = 158.82 \frac{\text{rev}}{\text{min}} \left(\frac{1 \text{ min}}{60 \text{ s}}\right) \left(\frac{2\pi \text{ rad}}{1 \text{ rev}}\right) = 16.6312 \frac{\text{rad}}{\text{s}}$$

Las tablas 6.1 y 6.2 muestran los resultados obtenidos para las velocidades de las dos ruedas, mientras que las figuras 6.9 y 6.10 corresponden a las curvas (ciclo útil – velocidad angular).

Ciclo útil	Frecuencia [Hz]	Velocidad angular de la rueda izquierda [rad/s]
0	0	0.0000
15	107	0.7372
31	237	1.6328
47	357	2.4595
63	475	3.2725
79	591	4.0717
95	716	4.9329
111	845	5.8216
127	979	6.7448
143	1118	7.7024
159	1276	8.7909
175	1437	9.9002
191	1612	11.1058
207	1795	12.3666
223	1996	13.7514
239	2205	15.1913
255	2414	16.6312

Tabla 6.1 Mediciones para la rueda izquierda

Ciclo útil	Frecuencia [Hz]	Velocidad angular de la rueda derecha [rad/s]
0	0	0.0000
15	107	0.7372
31	233	1.6052
47	349	2.4044
63	463	3.1898
79	578	3.9821
95	704	4.8502
111	830	5.7182
127	967	6.6621
143	1097	7.5577
159	1249	8.6049
175	1406	9.6866
191	1579	10.8785
207	1767	12.1737
223	1966	13.5447
239	2175	14.9846
255	2375	16.3625

Tabla 6.1 Mediciones para la rueda derecha

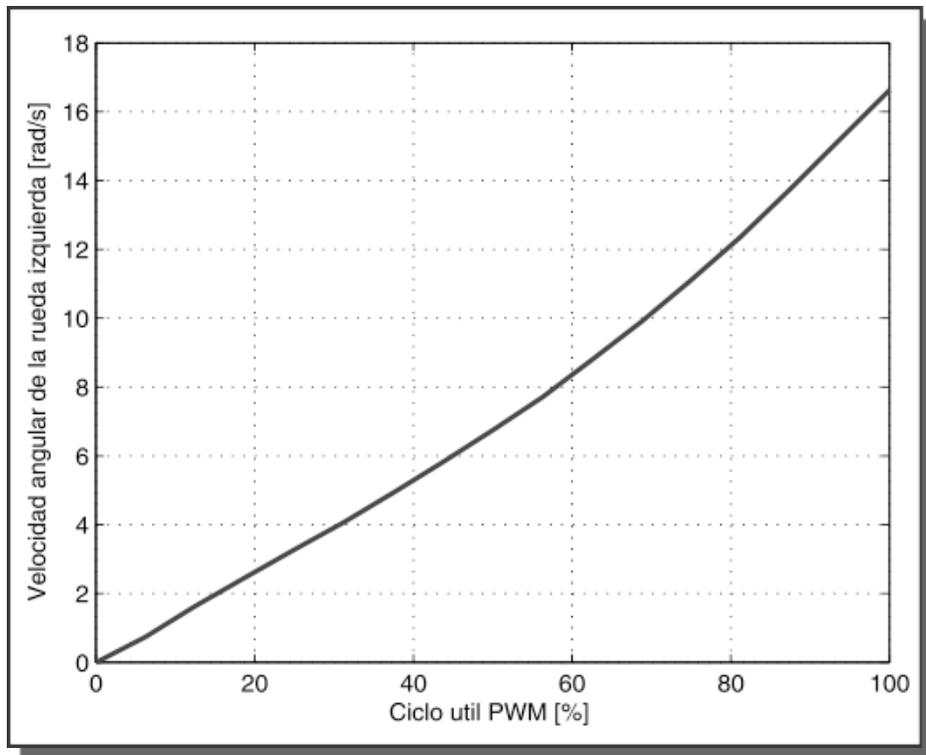


Fig.6.9 Velocidad angular de la rueda izquierda

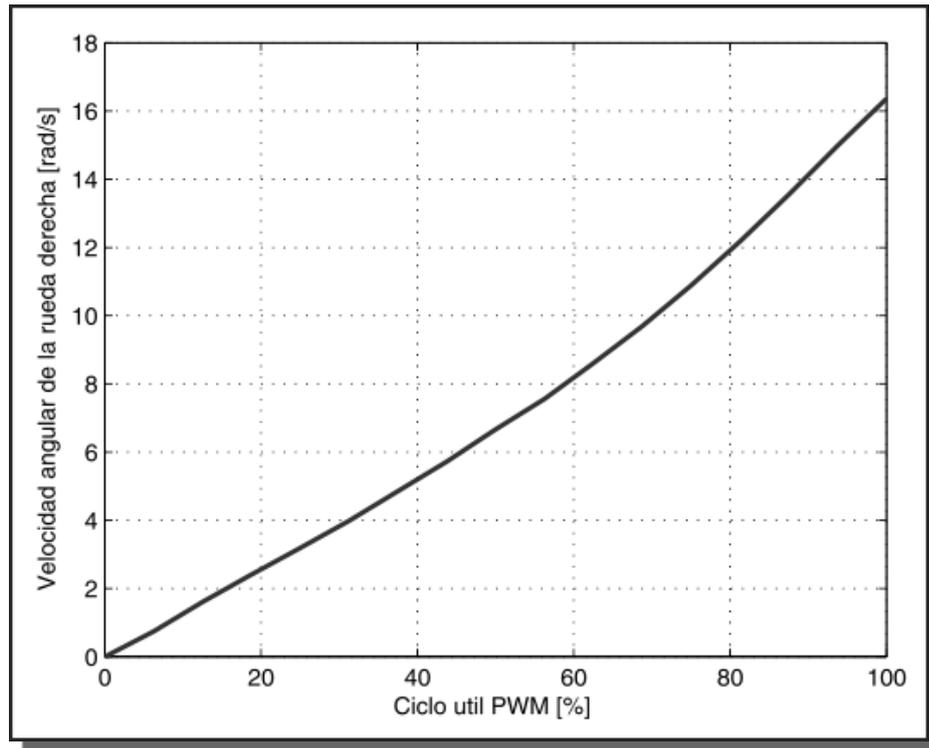


Fig.6.10 Velocidad angular de la rueda derecha

6.3 Descripción del sistema de control implementado

El sistema de control que se llevó a la práctica corresponde al de la sección 3.4, es decir, está basado en la técnica de modos deslizantes, esto por tratarse de un esquema que resulta más robusto ante incertidumbres en el sistema. Debido a esto se ahonda más en este tipo de control en esta sección.

El control por modos deslizantes es un esquema de estructura variable, en el que la ley de control se cambia de forma deliberada, de acuerdo a reglas que dependen del estado del sistema, originándose una ley de control discontinua.

Para entender cómo funciona este esquema de control se mostrarán dos ejemplos ilustrativos. Considerando el doble integrador dado por:

$$\ddot{y} = u(t) \quad (6.1)$$

Se considera inicialmente el efecto del uso de una ley de control realimentado como la siguiente:

$$u(t) = -ky(t) \quad (6.2)$$

Donde k es un escalar estrictamente positivo. Una forma de analizar el movimiento del sistema en lazo cerrado es por medio de un plano de fase, que esencialmente es una gráfica de velocidad contra posición. Sustituyendo la

acción de control (6.2) en la ecuación (6.1), y multiplicando la ecuación resultante por \dot{y} , se tiene:

$$\dot{y}\ddot{y} = -k\dot{y}y \tag{6.3}$$

Integrando esta expresión se obtiene la siguiente relación entre la velocidad y la posición:

$$\dot{y}^2 + ky^2 = c \tag{6.4}$$

Donde c representa una constante de integración resultado de las condiciones iniciales y es estrictamente positiva. En el caso en que $k = 1$, la ecuación (6.4) representa un círculo con centro en el origen y radio \sqrt{c} . De forma general, la gráfica de \dot{y} contra y es una elipse que depende de las condiciones iniciales como se muestra en la figura 6.11. Como se puede ver en dicha figura, la ley de control (6.2) no es apropiada ya que \dot{y} e y no se mueven hacia el origen (no se tiene estabilidad asintótica).

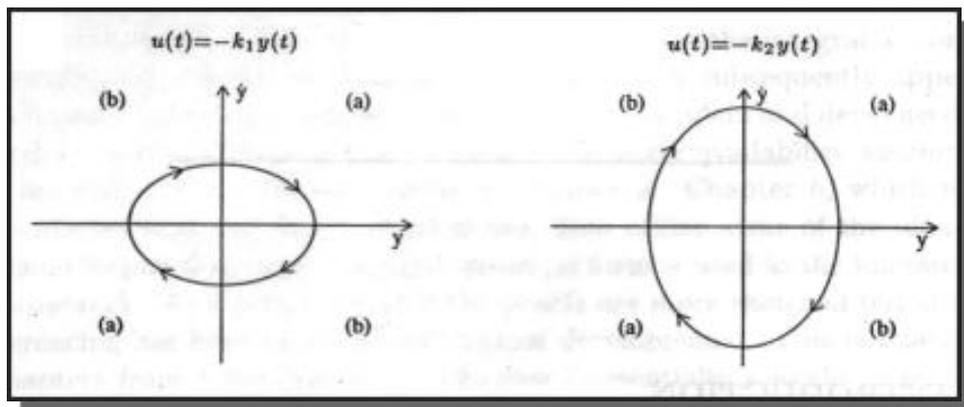


Fig.6.11 Planos de fase para dos valores de k

Considerando ahora la ley de control:

$$u(t) = \begin{cases} -k_1y(t) & \text{si } y\dot{y} < 0 \\ -k_2y(t) & \text{en otro caso} \end{cases} \tag{6.5}$$

Donde $0 < k_1 < 1 < k_2$. El plano de fase (y, \dot{y}) se divide por la regla de conmutación en cuatro cuadrantes separados como se muestra en la figura 6.11. La ley de control $u = -k_2y(t)$ actuará en los cuadrantes del plano de fase marcados como (a). En esta región, la distancia desde el origen a los puntos en el plano de fase decrece a lo largo de la trayectoria del sistema. De la misma forma, en la región (b), cuando la ley de control $u = -k_1y(t)$ está en operación, la distancia desde el origen a los puntos en el plano de fase decrece

a lo largo de la trayectoria del sistema. El plano de fase resultante se muestra en la figura 6.12.

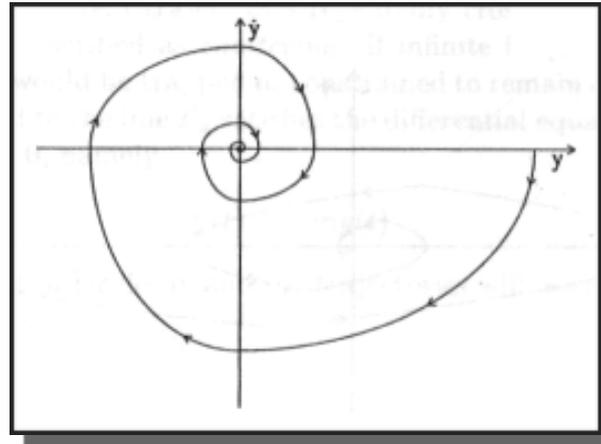


Fig.6.12 Plano de fase para el sistema de control de estructura variable

Otro ejemplo más significativo resulta del uso de la ley dada por:

$$u(t) = \begin{cases} -1 & \text{si } s(y, \dot{y}) > 0 \\ 1 & \text{si } s(y, \dot{y}) < 0 \end{cases} \quad (6.6)$$

Donde la función de conmutación se define como:

$$s(y, \dot{y}) = my + \dot{y} \quad (6.7)$$

Donde m es un escalar positivo de diseño. La razón para el uso del término 'función de conmutación' es clara, ya que la función dada en (6.7) se usa para decidir que estructura de control está en uso en cualquier punto (y, \dot{y}) en el plano de fase. La expresión en la ecuación (6.6) se escribe usualmente de forma más concisa:

$$u(t) = -sgn(s(t)) \quad (6.8)$$

Donde $sgn(\cdot)$ es la función signo. Esta función tiene la propiedad siguiente:

$$s \, sgn(s(t)) = |s| \quad (6.9)$$

La expresión dada en (6.6) se usa para controlar el doble integrador. Para valores grandes de \dot{y} del plano de fase, obtenido uniendo las componentes parabólicas de las leyes constituyentes, se muestra en la figura 6.13. La línea punteada en la figura representa el conjunto de puntos para los cuales $s(y, \dot{y}) = 0$; en este caso es una línea a través del origen de pendiente $-m$. Sin embargo, para valores de \dot{y} que satisfacen la desigualdad $m|\dot{y}| < 1$ se tiene:

$$s\dot{s} = s(m\dot{y} + \ddot{y}) = s(m\dot{y} - \text{sgn}(s(t))) < |s|(m|\dot{y}| - 1) < 0$$

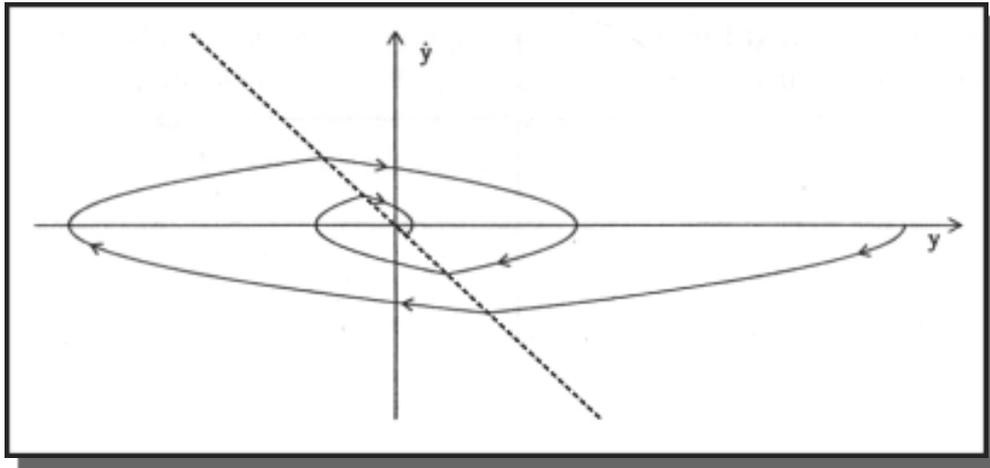


Fig.6.13 Plano de fase del sistema con \dot{y} grande

O de forma equivalente:

$$\lim_{s \rightarrow 0^+} \dot{s} < 0 \text{ y } \lim_{s \rightarrow 0^-} \dot{s} > 0 \quad (6.10)$$

Consecuentemente, cuando $m|\dot{y}| < 1$ las trayectorias en cualquier lado de la línea

$$\mathcal{L}_s = \{(y, \dot{y}) : s(y, \dot{y}) = 0\} \quad (6.11)$$

apuntan hacia la línea. Esto se demuestra en la figura 6.14, que muestra diferentes planos de fase que se interceptan en la línea \mathcal{L}_s desde diferentes condiciones iniciales.

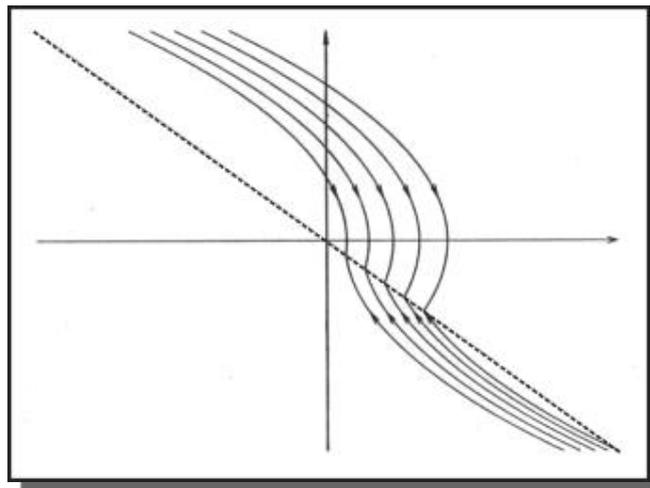


Fig.6.14 Plano de fase del sistema con estructura variable cerca del origen

De forma intuitiva, se origina una conmutación de alta frecuencia entre las dos estructuras de control conforme las trayectorias del sistema cruzan la línea \mathcal{L}_s .

Este movimiento de alta frecuencia se conoce como castaño (chattering en Inglés). Si es posible la conmutación a alta frecuencia, el movimiento queda restringido a evolucionar en la línea \mathcal{L}_s . El movimiento confinado a la línea \mathcal{L}_s satisface la ecuación diferencial obtenida de hacer $s(y, \dot{y}) = 0$:

$$\dot{y}(t) = -my(t) \tag{6.12}$$

Esta ecuación representa un decaimiento de primer orden y las trayectorias del sistema se ‘deslizan’ a lo largo de la línea \mathcal{L}_s hacia el origen (figura 6.15).

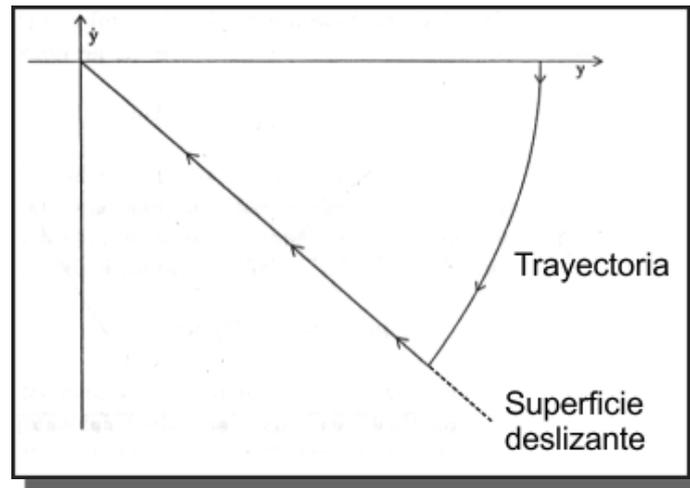


Fig.6.15 Plano de fase de un movimiento deslizante

Tal comportamiento dinámico se describe como un modo deslizante ideal o movimiento deslizante ideal a la línea \mathcal{L}_s se le llama superficie de deslizamiento. Para más información acerca del control por modos deslizantes se recomienda revisar la bibliografía.

Enseguida se procede a describir de forma general el sistema de control implementado, la figura 6.16 muestra el flujo de señales de los componentes principales del sistema.

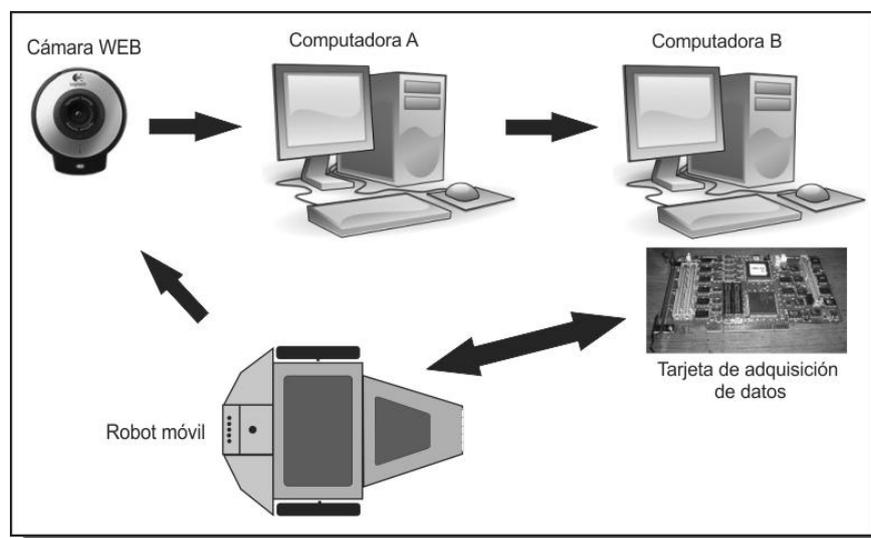


Fig.6.16 Flujo de señales del sistema de control

A grandes rasgos, el sistema de control se puede describir como sigue: se tiene una cámara web que adquiere imágenes del área de trabajo y las envía vía USB a la computadora A, esta computadora se encarga de procesar estas imágenes para obtener la posición y la orientación del robot móvil. Posteriormente la computadora A envía los datos de posición y orientación vía puerto serie a la computadora B.

El esquema de control está implementado en tiempo real en la computadora B, por lo que las consignas de control (velocidades angulares de las ruedas del robot) se generan en esta computadora y se envían por cable a la tarjeta montada en el robot, que es la etapa de potencia de los motores del robot.

Al irse moviendo el robot en el plano de trabajo se produce, en cada instante de tiempo, una nueva localización que se retroalimenta para compararse con el valor deseado. Con base en esto se van generando los controles para lograr el objetivo del seguimiento de la trayectoria predeterminada.

El programa en Simulink que implementa el control en modos deslizantes se muestra en la figura 6.17, mientras que la figura 6.18 muestra el contenido del bloque marcado como “Posición y Orientación CAM”. El código del algoritmo de control (dentro del bloque “Control”) se muestra al final de esta sección.

Las variables (x, y, θ) son recibidas en la computadora B, previo a la obtención de las derivadas, las variables se pasan por un filtro que suaviza los datos, teniéndose valores de las derivadas sin cambios bruscos.

El bloque “Trayectoria” define las variables de referencia para generar la trayectoria deseada. El algoritmo de control está implementado en el bloque “Control” (figura 6.17), donde se generan las aceleraciones lineales y angulares que pasan por integradores para generar las velocidades v y ω . A partir de ellas se generan las velocidades angulares para las dos ruedas ω_d y ω_i .

Enseguida, se hace una correspondencia entre las velocidades angulares y la señal a ser aplicada al microcontrolador, para generar la señal PWM que va a los puentes H y los motores del robot, esto último se hace mediante un ajuste a un polinomio de grado 8. También se incluye una saturación de las velocidades generadas para prevenir que se alcancen los valores máximos. Como resultado, se generan las señales v_1 y v_2 que se envían por salidas analógicas de la tarjeta Sensoray.

Dependiendo del signo de las velocidades ω_d y ω_i es el sentido de giro de los motores. Para esto se envía un nivel alto ó un nivel bajo por medio de salidas digitales de la Sensoray, estas señales se envían a los puentes H que se encargan del trabajo de hacer girar los motores a la izquierda ó a la derecha, según corresponda.

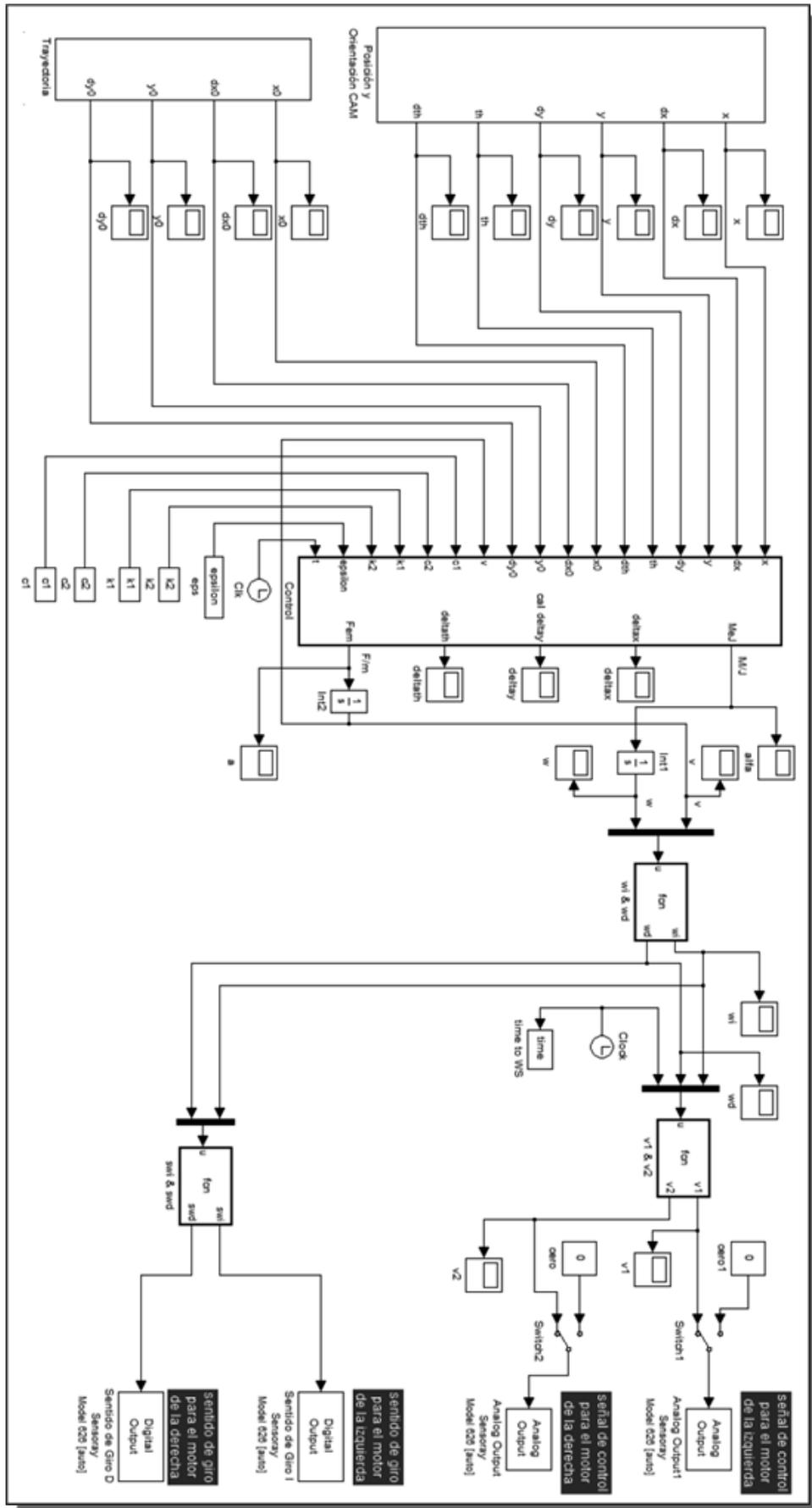


Fig.6.17 Programa de control por modos deslizantes

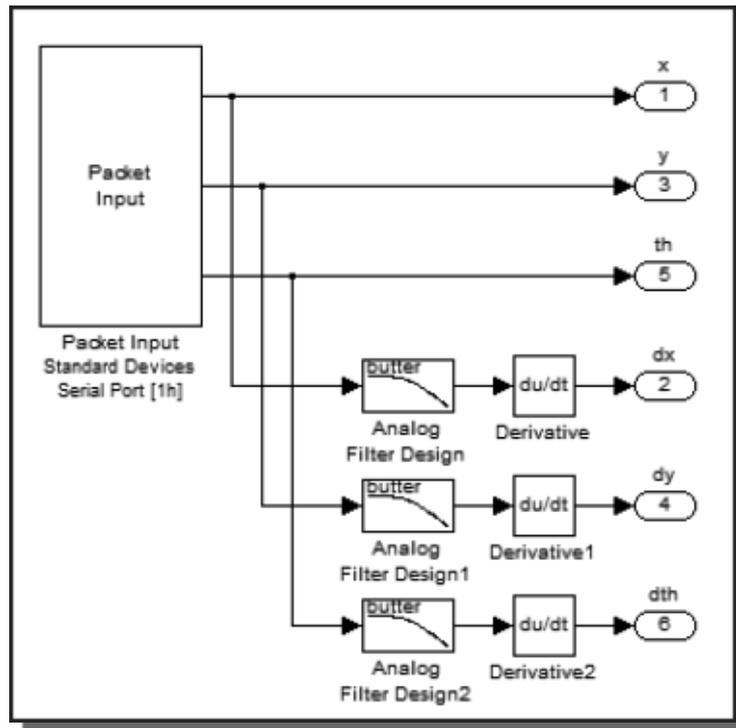


Fig.6.18 Recepción de las variables de localización del robot

Código del algoritmo de control (figura 6.17):

```
function [MeJ, deltax, deltay, deltath, Fem] =
cal(x, dx, y, dy, th, dth, x0, dx0, y0, dy0, v, c1, c2, k1, k2, epsilon, t)

deltax = x - x0;

deltay = y - y0;

S1 = dx + k1 * deltax;
S2 = dy + k1 * deltay;

uu = -( S1 * cos(th) + S2 * sin (th) );

if t > 0.1

    if ( v > 0 ) || ( uu >= 0 )
        Fem = c1 * uu;

    elseif ( v == 0 ) && ( uu < 0 )
        Fem = 0;
    else
        Fem = 0;
    end

else
    Fem = 0;
end

if deltax^2 + deltay^2 > epsilon
```

```

    th0 = atan2( -deltay, -deltax );
elseif ( deltax^2 + deltay^2 <= epsilon ) && ( dx0^2 + dy0^2 ~= 0 )
    th0 = atan2( -dy0, -dx0 );
elseif ( deltax^2 + deltay^2 <= epsilon ) && ( dx0^2 + dy0^2 == 0 )
    th0 = 0.0;
else
    th0 = 0.0;
end

deltath = th - th0;
S3 = dth + k2 * deltath;

if t > 0.1
    MeJ = -c2 * S3;
else
    MeJ = 0;
end

```

6.4 Pruebas en tiempo real

Las pruebas del sistema de control consistieron en una trayectoria recta y una senoidal. Aquí se presentan los resultados para estos experimentos.

La trayectoria **recta** está definida como:

$$x_0 = -0.5 + (3.6 / t_f^2) * t^2 - (2.4 / t_f^3) * t^3;$$

$$y_0 = -0.5 + (3.6 / t_f^2) * t^2 - (2.4 / t_f^3) * t^3;$$

Que consiste de una recta del punto (-0.5, -0.5) al punto (0.7, 0.7).

Las ganancias de control utilizadas fueron:

$$c_1 = 1$$

$$c_2 = 1$$

$$k_1 = 0.2$$

$$k_2 = 0.4$$

El tiempo de simulación fue de 30 segundos.

La trayectoria **senoidal** está definida como:

$$x_0 = -1 + (6 / t_f^2) * t^2 - (4 / t_f^3) * t^3;$$

$$y_0 = -0.8 * \sin(x_0 * \pi);$$

La cual genera una senoidal que va de -1 a 1 en el eje x, y que tiene una amplitud de 0.8 m.

El tiempo de simulación fue de 60 segundos.

Las ganancias son las mismas que para la trayectoria recta.

Cabe señalar que antes de aplicar el control a los motores del robot se deja correr un intervalo de 0.1 segundos, para permitir que se generen y transmitan la posición y orientación desde la computadora que tiene la cámara web a la computadora que genera el control.

Las gráficas siguientes muestran los resultados de las pruebas en tiempo real para la trayectoria recta y para la senoidal.

Trayectoria recta

Trayectoria del robot en el plano XY (figura 6.19):

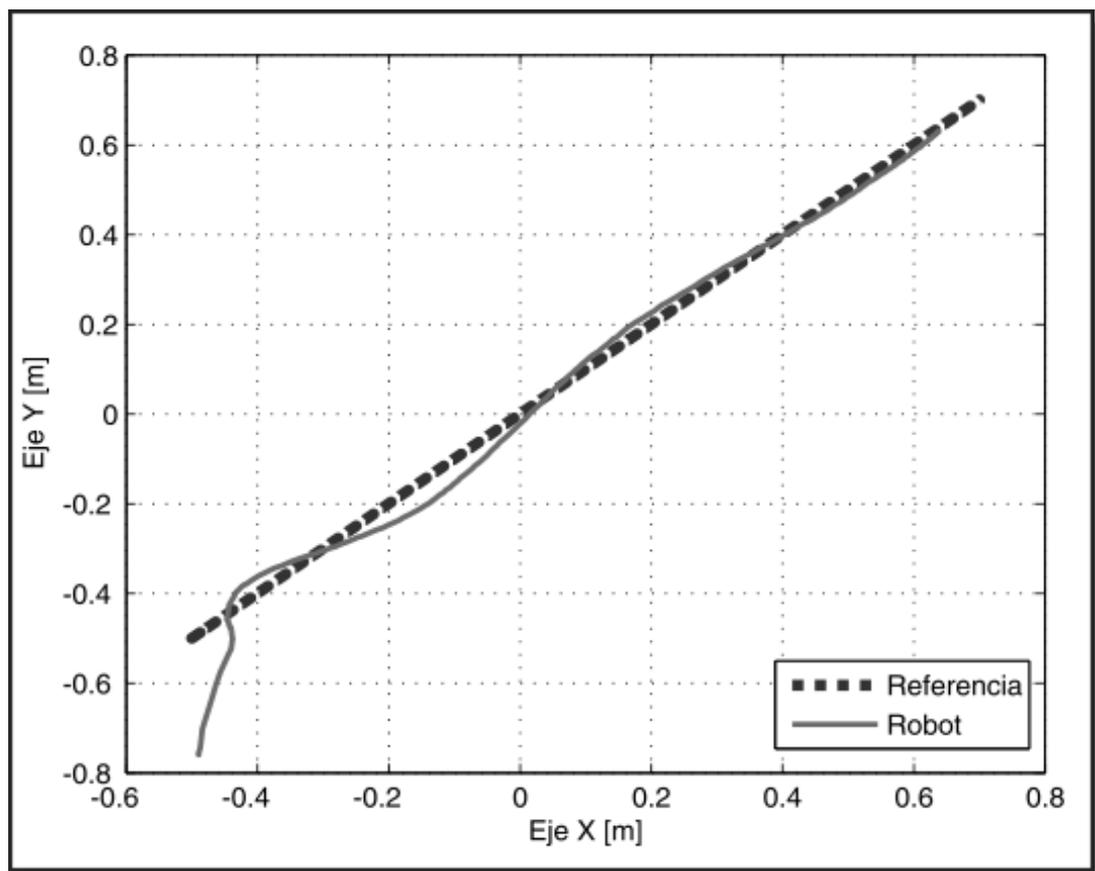


Fig. 6.19 Trayectoria en el plano XY

Variable Δx (figura 6.20):

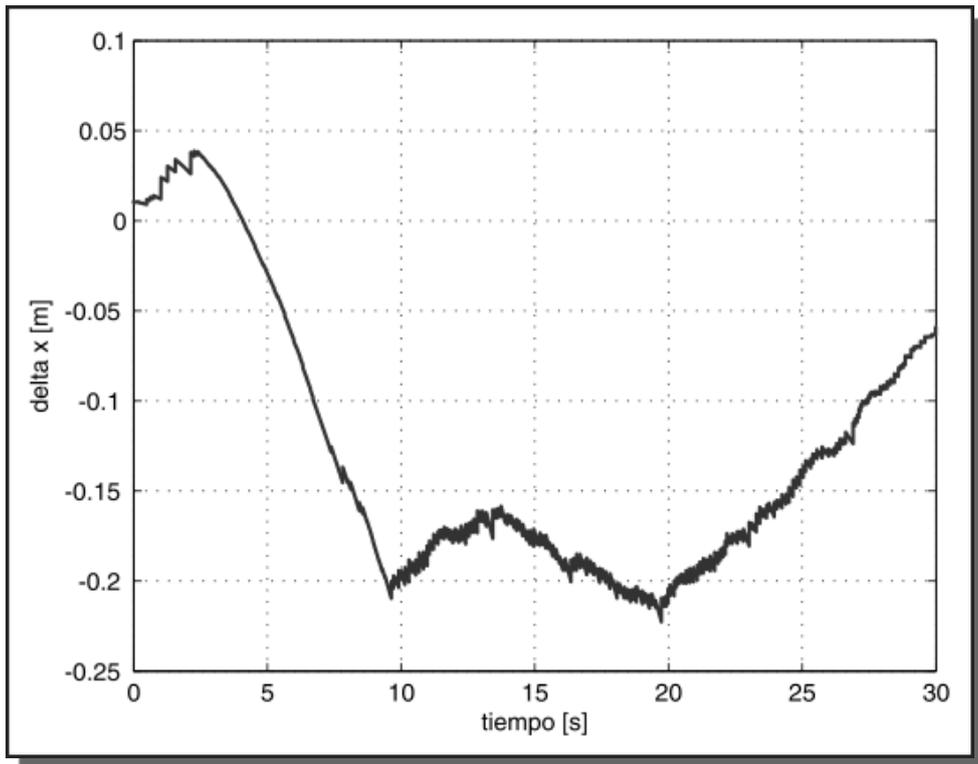


Fig. 6.20 Error en x (Δx)

Variable Δy (figura 6.21):

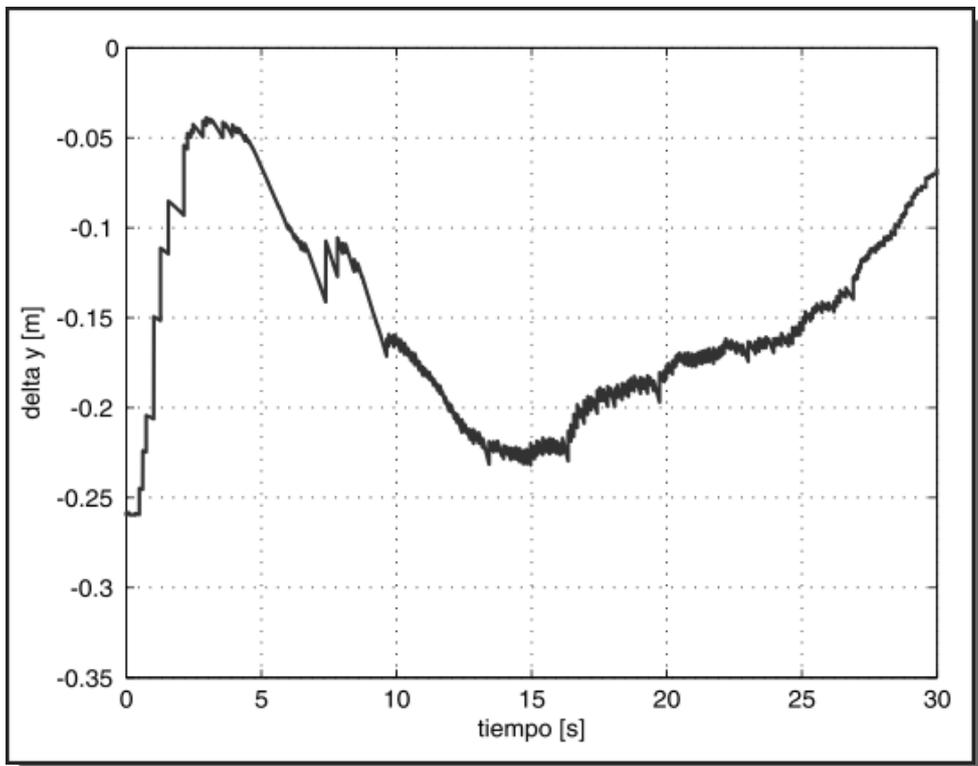


Fig. 6.21 Error en y (Δy)

Variable $\Delta\theta$ (figura 6.22):

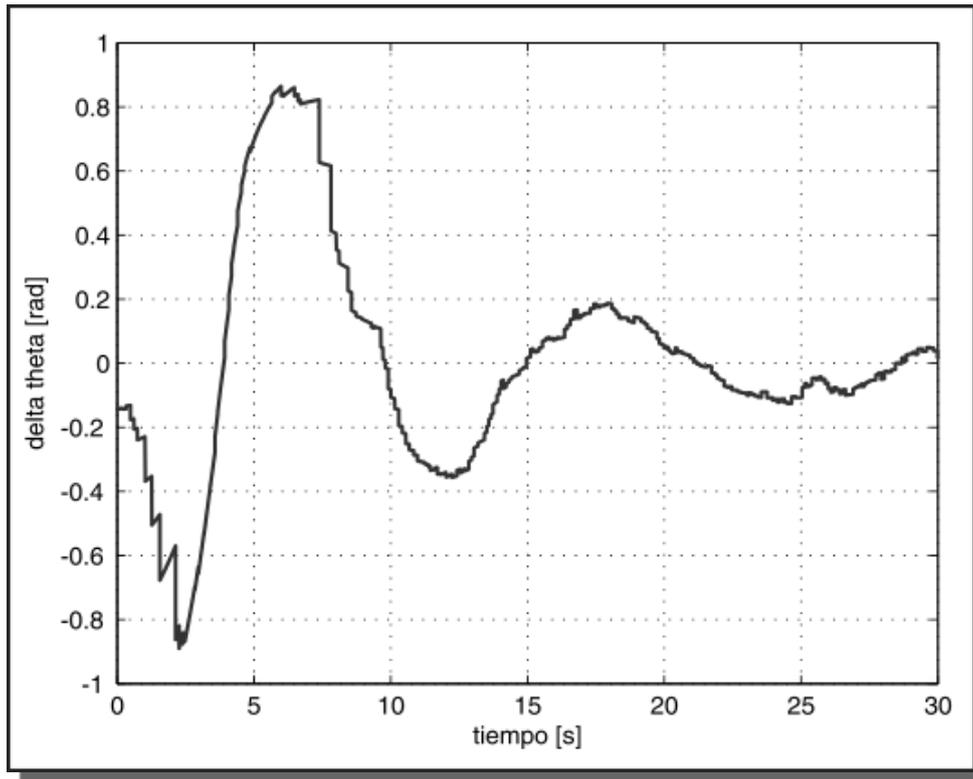


Fig. 6.22 Error en θ ($\Delta\theta$)

Aceleración lineal, $a = F/m$ (figura 6.23):

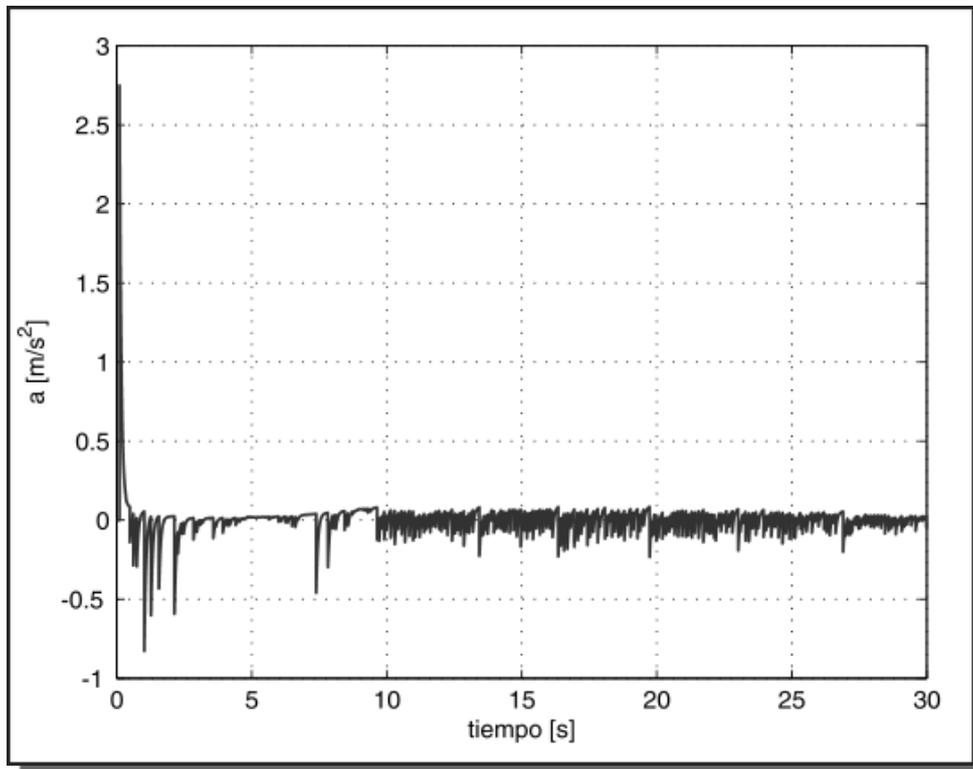


Fig. 6.23 Aceleración lineal, $a = F/m$

Aceleración angular, $\alpha = M/J$ (figura 6.24):

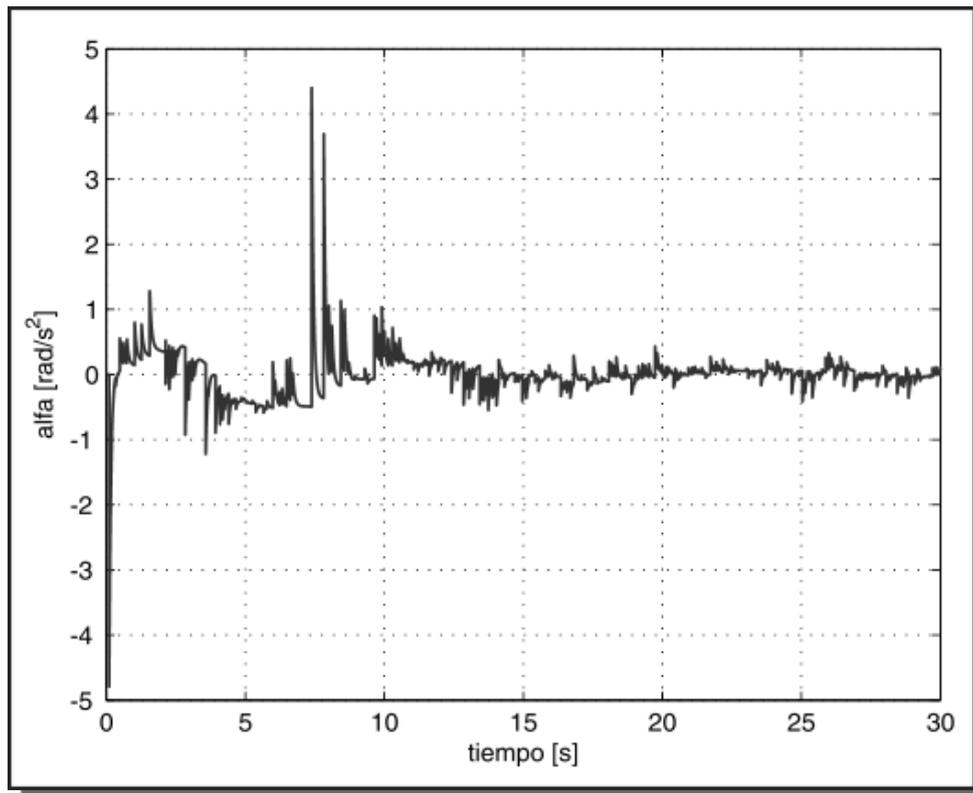


Fig. 6.24 Aceleración angular, $\alpha = M/J$

Velocidad lineal aplicada al robot (figura 6.25):

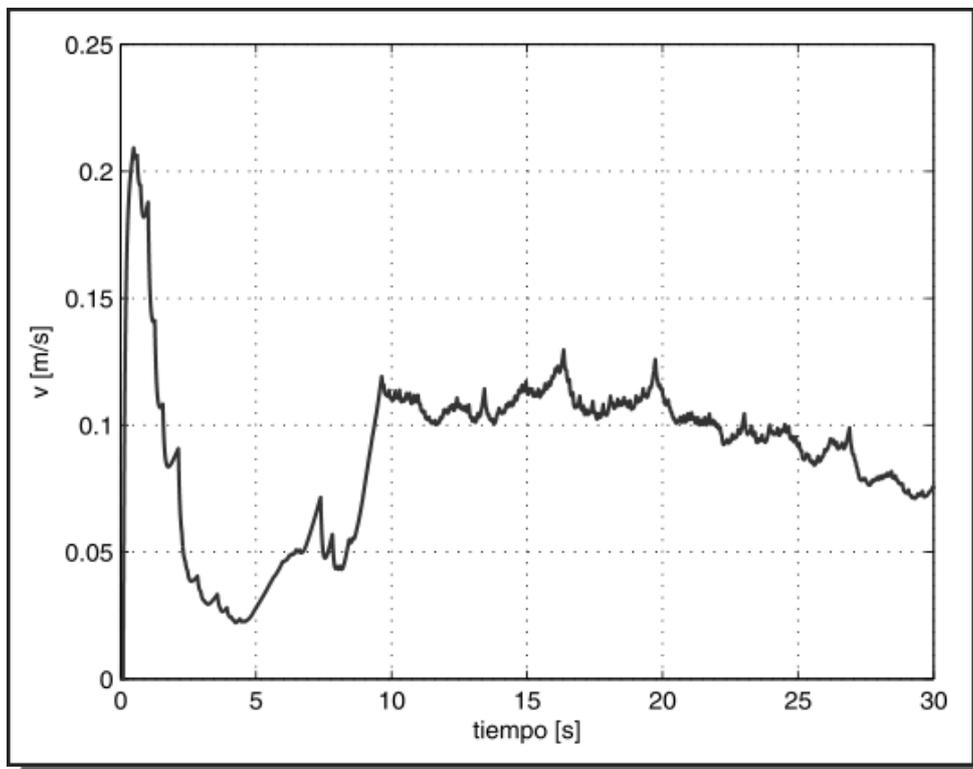


Fig. 6.25 velocidad lineal aplicada al robot, v

Velocidad angular aplicada al robot (figura 6.26):

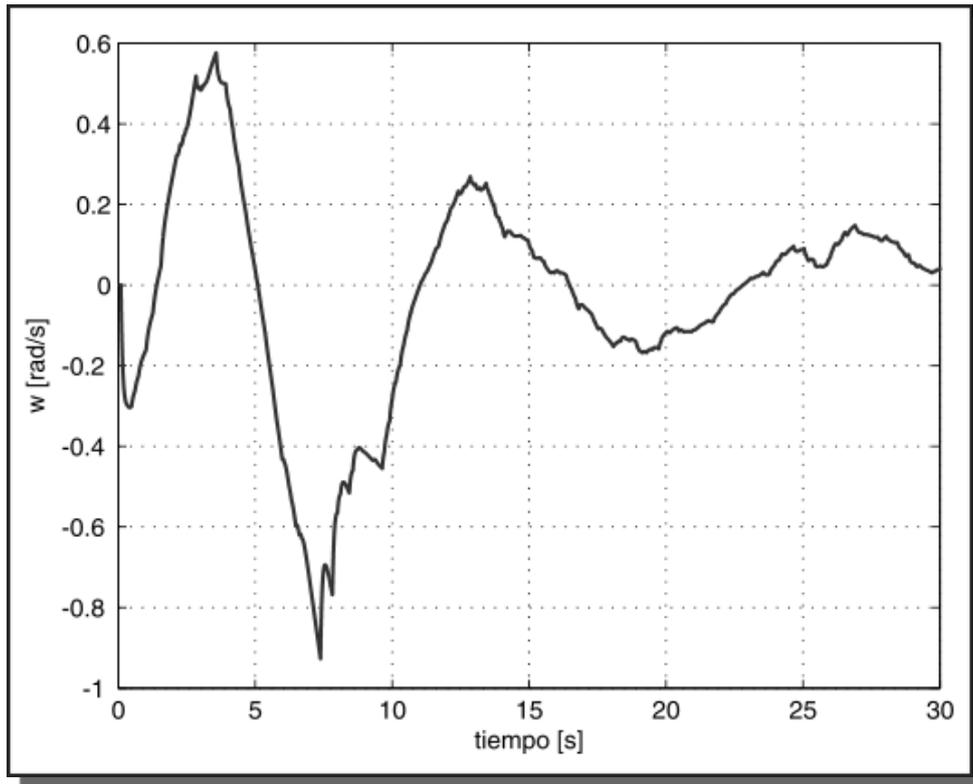


Fig. 6.26 velocidad angular aplicada al robot, ω

Velocidad angular de la rueda izquierda (figura 6.27):

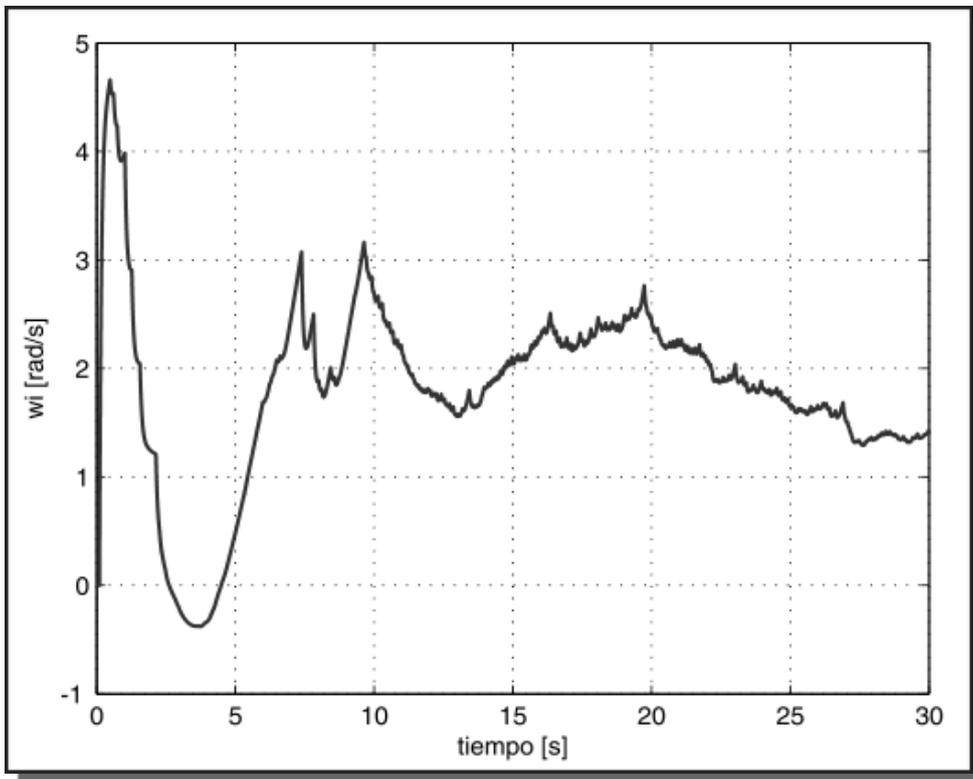


Fig. 6.27 velocidad angular de la rueda izquierda

Velocidad angular de la rueda derecha (figura 6.28):

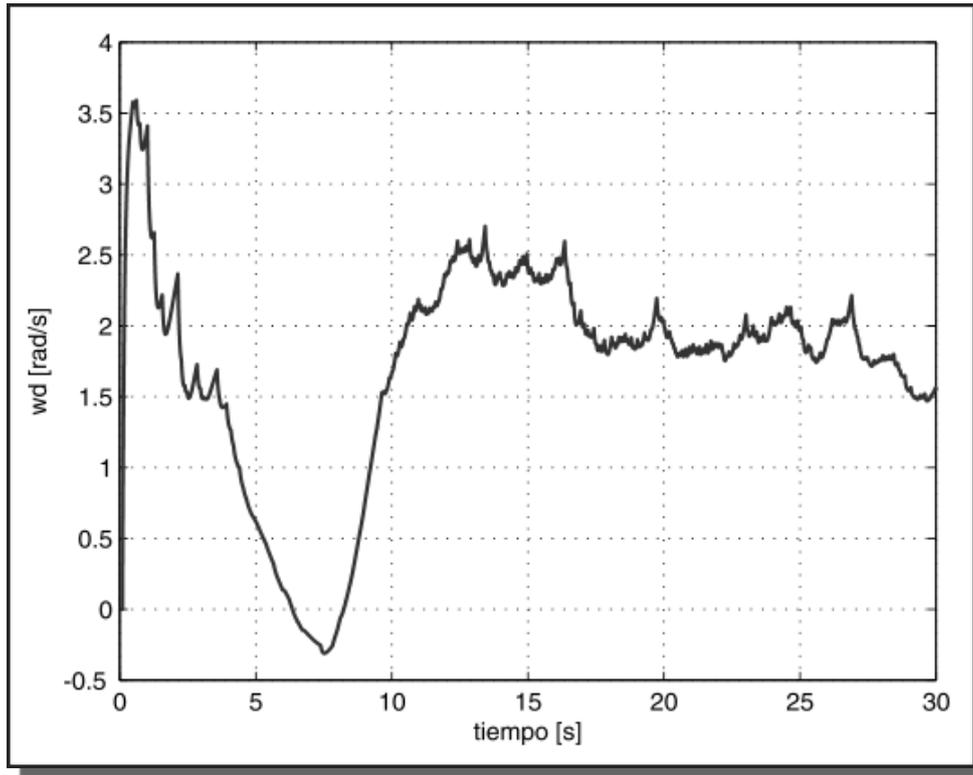


Fig. 6.28 velocidad angular de la rueda derecha

Trayectoria senoidal

Trayectoria del robot en el plano XY (figura 6.29):

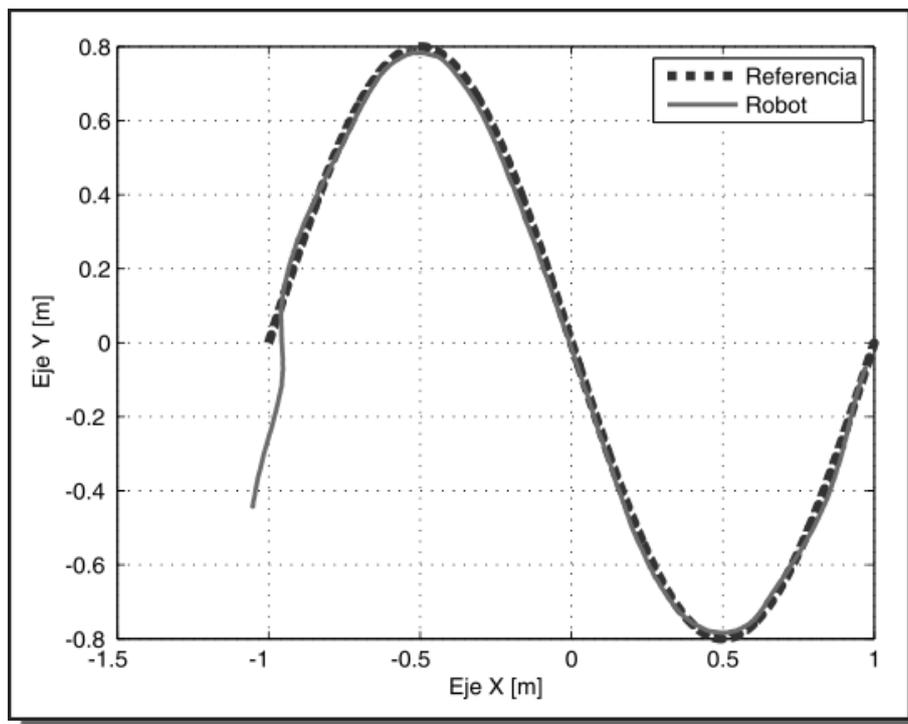


Fig. 6.29 Trayectoria en el plano XY

Variable Δx (figura 6.30):

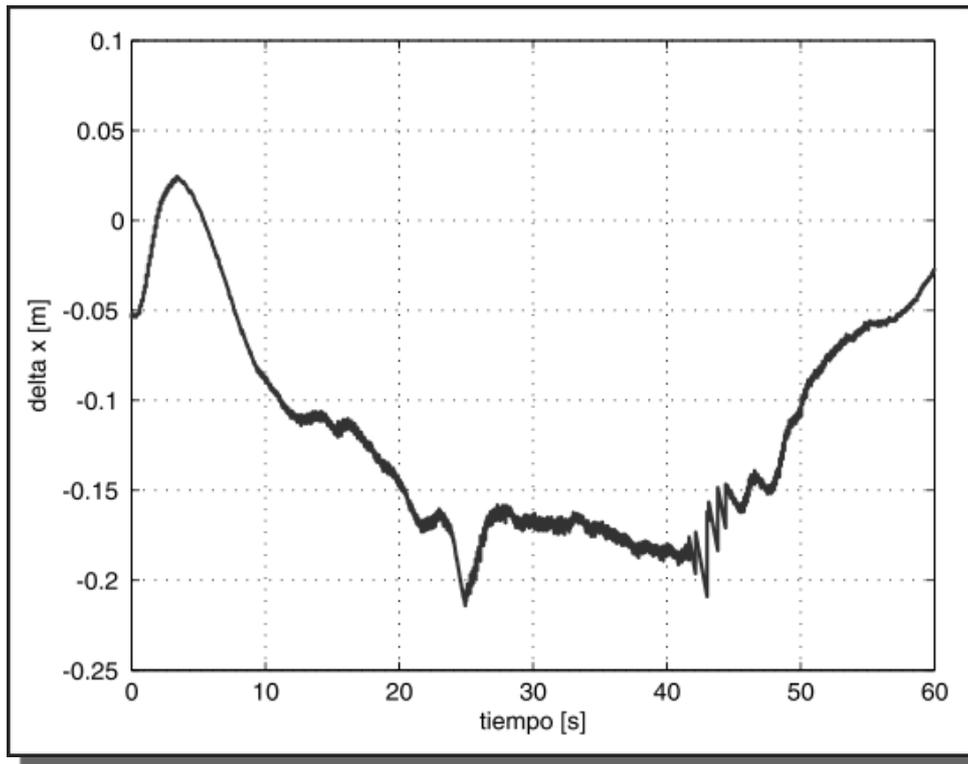


Fig. 6.30 Error en x (Δx)

Variable Δy (figura 6.31):

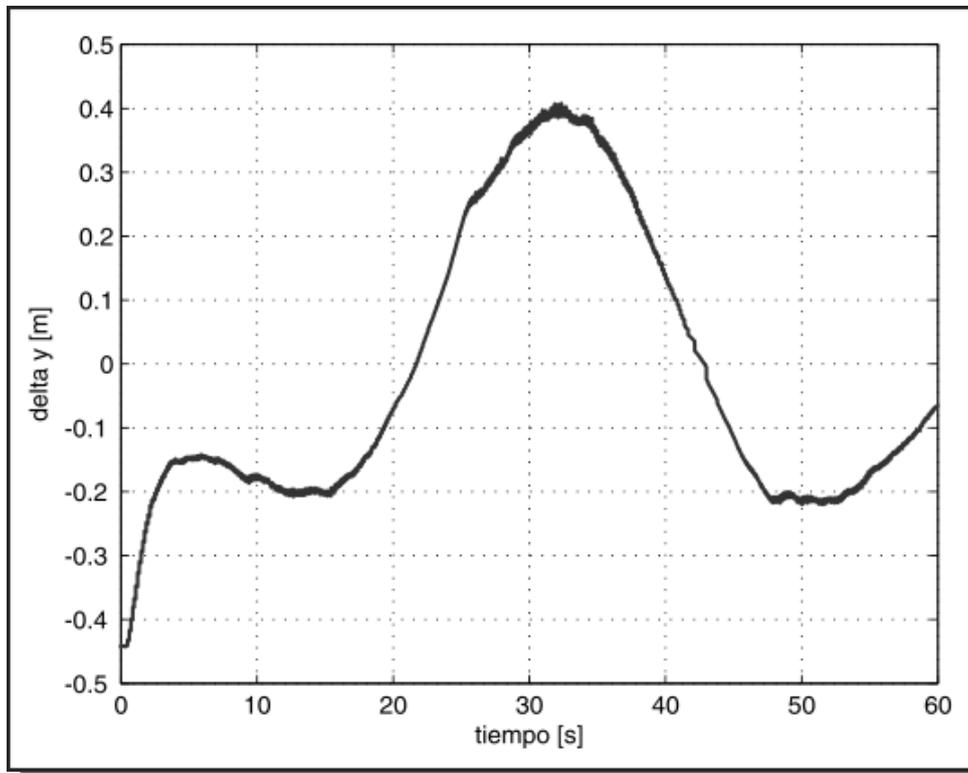


Fig. 6.31 Error en y (Δy)

Variable $\Delta\theta$ (figura 6.32):

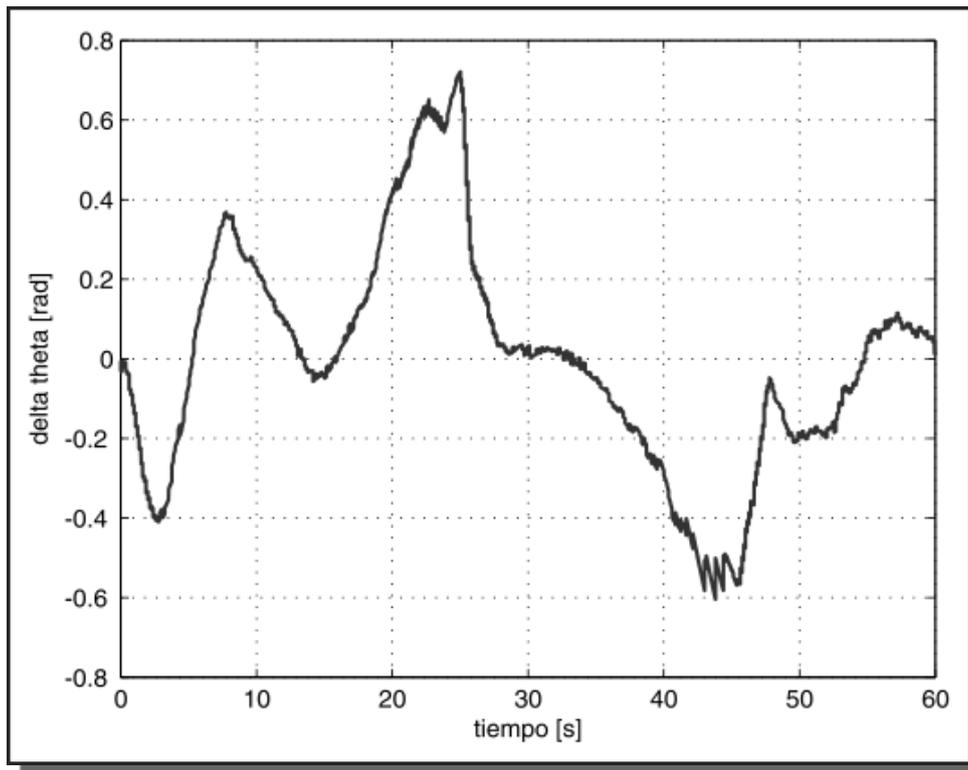


Fig. 6.32 Error en θ ($\Delta\theta$)

Aceleración lineal, $a = F/m$ (figura 6.33):

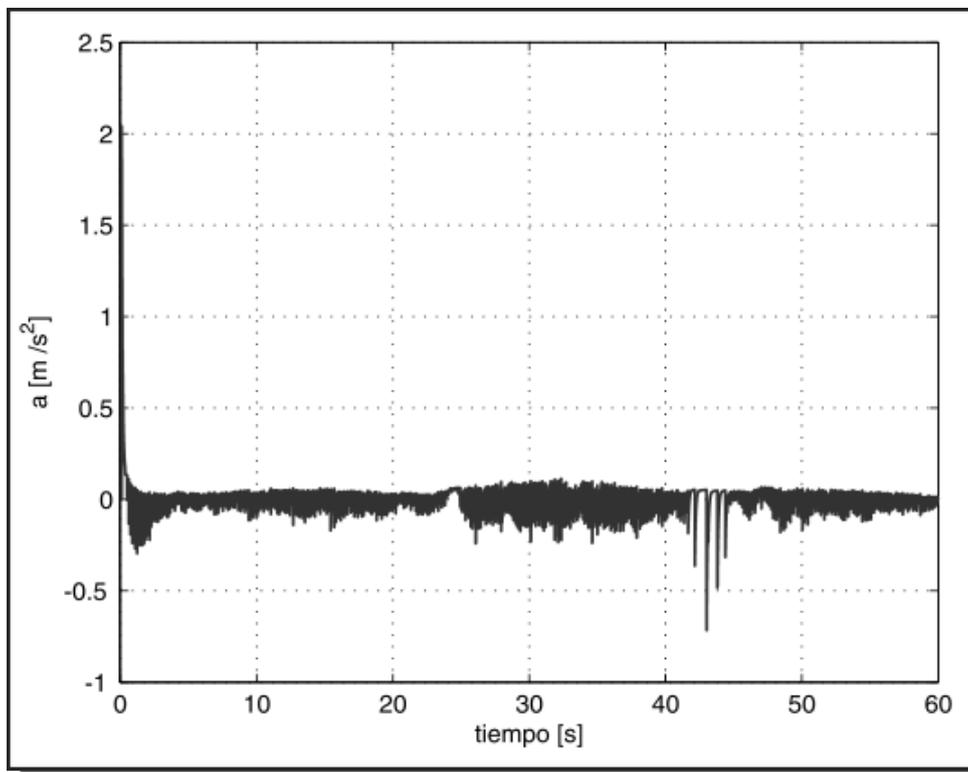


Fig. 6.33 Aceleración lineal, $a = F/m$

Aceleración angular, $\alpha = M/J$ (figura 6.34):

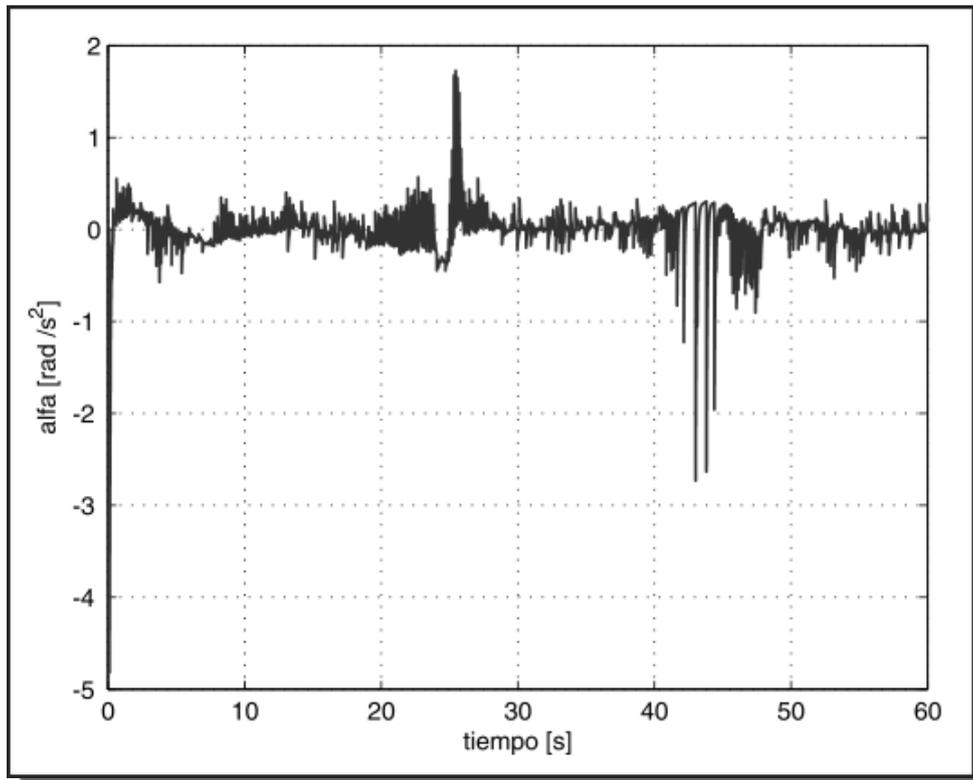


Fig. 6.34 Aceleración angular, $\alpha = M/J$

Velocidad lineal aplicada al robot (figura 6.35):

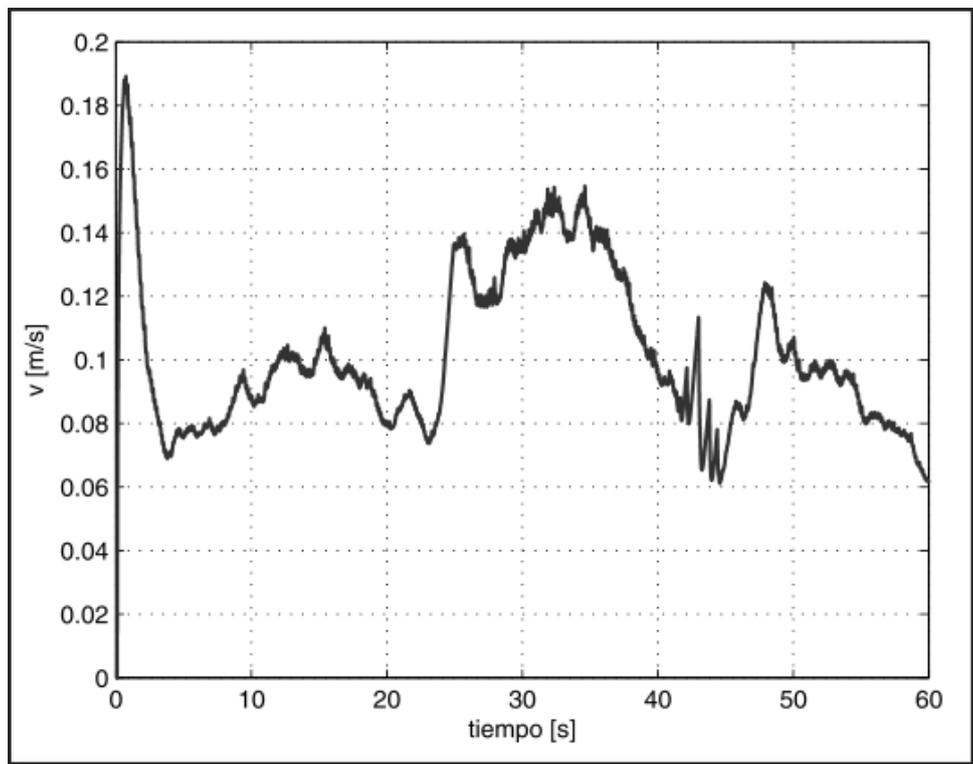


Fig. 6.35 velocidad lineal aplicada al robot, v

Velocidad angular aplicada al robot (figura 6.36):

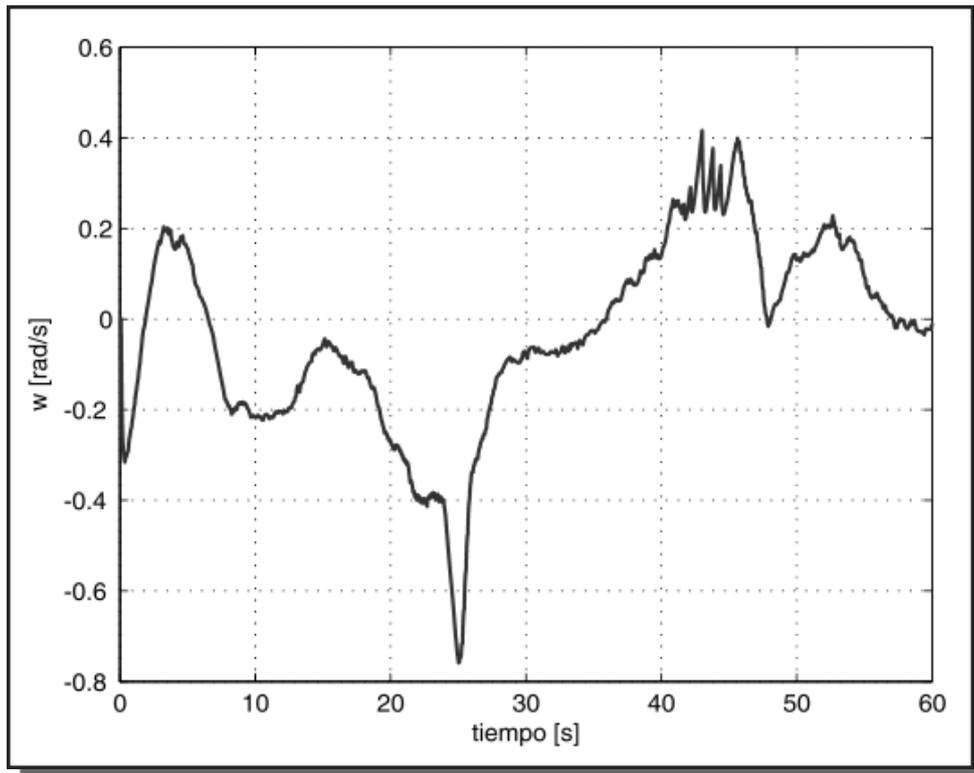


Fig. 6.36 velocidad angular del robot, ω

Velocidad angular de la rueda izquierda (figura 6.37):

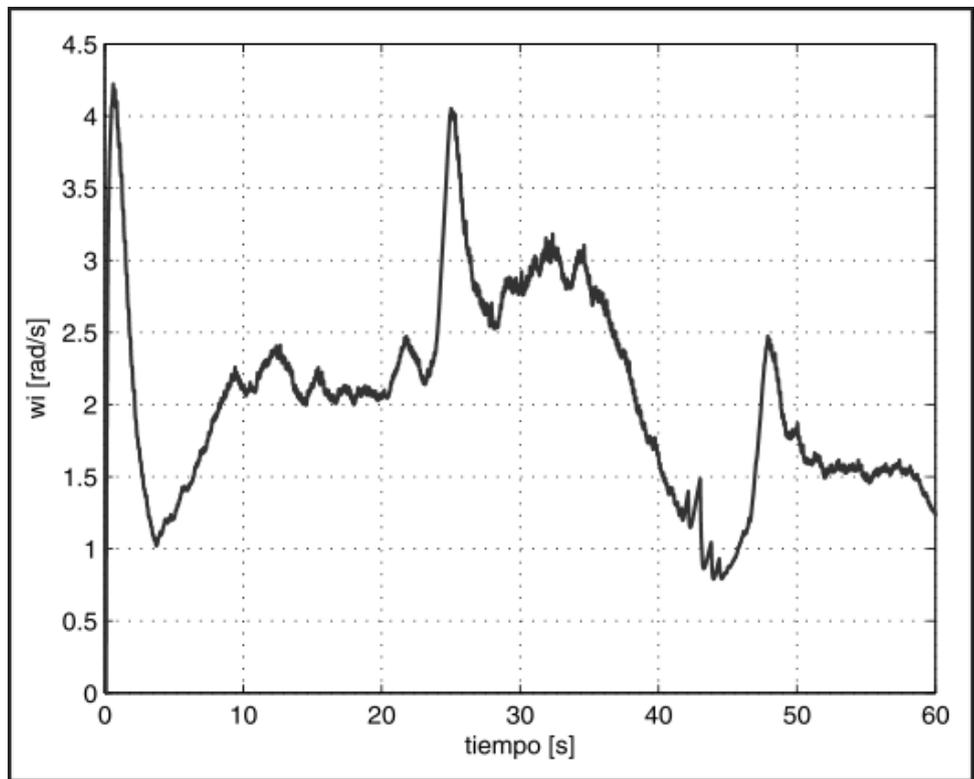


Fig. 6.37 velocidad angular de la rueda izquierda

Velocidad angular de la rueda derecha (figura 6.38):

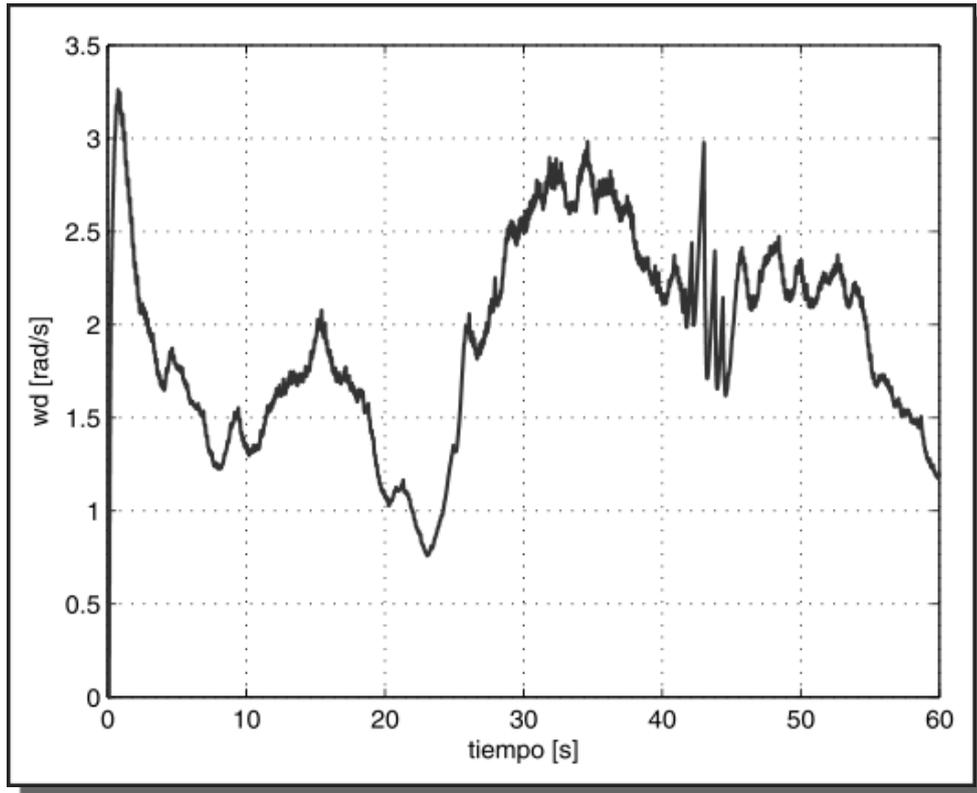


Fig. 6.38 velocidad angular de la rueda derecha

Conclusiones

La realización de este trabajo aparte de ser muy interesante para un servidor, permitió aplicar los conocimientos adquiridos durante la carrera y, al mismo tiempo, aprender cuestiones prácticas útiles en la implementación de un sistema de control.

Se mostró la importancia del campo de la robótica móvil y la necesidad de incrementar su estudio en nuestro país, tal como se hace en otras naciones.

Los modelos matemáticos encontrados se utilizaron tanto para las simulaciones como para la implementación, ya que los modelos establecen como se deben aplicar las velocidades a las ruedas del robot, partiendo de las consignas de control. El modelo odométrico no fue muy útil en la práctica, esto debido a que los encoders del robot no tienen mucha resolución, además de que el desplazamiento del robot, y como consecuencia la medición de los encoders, se ve afectado por el piso irregular del espacio de trabajo.

Respecto a las simulaciones realizadas para el seguimiento de trayectorias, los mejores resultados se obtuvieron con el controlador no lineal, pero el problema es que se generan velocidades muy grandes, que requieren motores que deben girar a velocidades mayores a las máximas del robot García.

La implementación usando modos deslizantes resultó ser una buena opción, ya que a pesar de las irregularidades del piso en el área de trabajo, se logró un buen seguimiento de trayectorias. Se observó que el robot va un poco atrás de la trayectoria de referencia, y que esto podría mejorarse incrementándose las ganancias, pero esto a costa de una mayor oscilación o de un error grande en la orientación al final de la trayectoria.

Trabajo Futuro

Hay algunas cosas que se pueden hacer para tener un mejor desempeño del robot, una de ellas es trabajar sobre un piso parejo, esto para evitar que el robot se vaya deteniendo ó desviando ligeramente de su trayectoria de referencia.

Otra mejora es la utilización de una cámara especializada, junto con una tarjeta de captura, para lograr mejorar la velocidad de muestreo del sistema de visión. Se pueden usar además algoritmos de visión basados en rasgos descriptores y redes neuronales. Estos algoritmos utilizan algunas imágenes del objeto en estudio (en este caso el robot) para obtener diversas características (forma, bordes, color, tamaño, etc.), y luego junto con las redes neuronales, reconocer de forma única al

objeto para todos los casos de ubicación en su área de trabajo. Esto permite el uso de un área de trabajo en donde la iluminación sea menos importante, reconocimiento de objetos ajenos (que se pueden evadir), reconocimiento de múltiples robots, entre otras ventajas.

Para eliminar el cable de comunicación con el robot se puede usar transmisión por radiofrecuencia o programar controles sencillos en el microcontrolador a bordo, para tener un robot autónomo.

Una opción que permitiría tener mejores resultados a velocidades mayores sería el uso de un modelo dinámico del robot y la inclusión de la dinámica de sus motores, junto con un sistema de localización más rápido.

Bibliografía

- [1] K. Åström, R. Murray, *Feedback Systems*, E.U.A., Princeton University Press, 2008.
- [2] G. Branwyn, *Absolute Beginner's Guide to Building Robots*, E.U.A., Que Publishing, 2003.
- [3] C. Canudas, B. Siciliano, G. Bastin, *Theory of Robot Control*, Gran Bretaña, Springer, 1997.
- [4] W. Dixon, D. Dawson, *Nonlinear Control of Wheeled Mobile Robots*, Gran Bretaña, Springer, 2001.
- [5] C. Edwards, S. Spurgeon, *Sliding Mode Control: Theory and Applications*, Gran Bretaña, Taylor & Francis, 1998.
- [6] H. Khalil, *Nonlinear Systems*, E.U.A., Prentice-Hall, 1996.
- [7] J. Morales, J. Martínez, M. Martínez, A. Mandow, "Pure-pursuit reactive path tracking for nonholonomic mobile robots with a 2D laser scanner," *EURASIP Journal on Advances in Signal Processing*, art. 3, vol. 2009.
- [8] K. Ogata, *Ingeniería de Control Moderna*, 4ª. Edición, España, Pearson Educación, 2003.
- [9] A. Ollero, *Robótica, Manipuladores y Robots Móviles*, Marcombo, España, 2007.
- [10] L. Scharf, W. Harthill, P. Moose, "A comparison of expected flight times for intercept and pure-pursuit missiles," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 5, no. 4, pp. 672–673, 1969.
- [11] R. Siegwart, *Introduction to Autonomous Mobile Robots*, E.U.A., MIT Press, 2004.
- [12] J. Slotine, W. Li, *Applied Nonlinear Control*, E. U. A., Prentice-Hall, 1991.
- [13] M. Spong, S. Hutchinson, M. Vidyasagar, *Robot Modeling and Control*, E.U.A., John Wiley & Sons, 2006.
- [14] V. Utkin, J. Guldner, J. Shi, *Sliding Mode Control in Electromechanical Systems*, E.U.A., CRC Press, 1999.
- [15] Y. Zolotukhin, K. Kotov, A. Nesterov, "Decentralized Control of Mobile Robots in Formation," *Optoelectronics, Instrumentation and Data Processing*, vol. 43, no. 3, pp. 218–225, 2007.

Páginas de Internet:

Acroname Robotics, robot García, [en línea]. Dirección URL: <<http://www.acroname.com>>. [Consulta: agosto 2009].

CadSoft, software E.A.G.L.E., [en línea]. Dirección URL: <<http://www.cadsoftusa.com>>. [Consulta: noviembre 2009].

Mathworks, MATLAB y Simulink, [en línea]. Dirección URL: <<http://www.mathworks.com>>. [Consulta: septiembre 2009].

Microchip, microcontroladores PIC, [en línea]. Dirección URL: <<http://www.microchip.com>>. [Consulta: noviembre 2009].

MikroElektronika, compilador mikroC para PICs, [en línea]. Dirección URL: <<http://www.mikroe.com>>. [Consulta: enero 2010].

Motores Maxon, [en línea]. Dirección URL: <<http://www.maxonmotor.com>>. [Consulta: agosto 2009].

Sensoray, productos de adquisición de datos, [en línea]. Dirección URL: <<http://www.sensoray.com>>. [Consulta: septiembre 2009].

Wikipedia, la enciclopedia de contenido libre, [en línea]. Dirección URL: <http://en.wikipedia.org/wiki/Mobile_robot>. [Consulta: marzo 2010].