

60
2 es.



UNIVERSIDAD NACIONAL AUTÓNOMA
DE MÉXICO

FACULTAD DE INGENIERIA

DISEÑO, CONSTRUCCION Y CONTROL DE
UN ROBOT MOVIL

T E S I S

QUE PARA OBTENER EL TITULO DE:
INGENIERO MECANICO ELECTRICISTA
(AREA ELECTRICA Y ELECTRONICA)

P R E S E N T A N :
DANIEL FERNANDEZ NUÑEZ
SERGIO DANIEL VERA REA



ASESOR: DR. YU TANG XU

MEXICO, D. F.

1998.

TESIS CON
FALLA DE ORIGEN

266252



Universidad Nacional
Autónoma de México

Dirección General de Bibliotecas de la UNAM

Biblioteca Central



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.

***DISEÑO, CONSTRUCCION Y
CONTROL DE UN ROBOT
MOVIL***

Agradecimientos:

- A mis madre ya que sin su apoyo no lo hubiera conseguido.
- A mi padre, quien tanto tiempo deseo esto.
- A mi ya que sin mi ayuda no lo hubiese logrado
- A Dios porque nunca me abandonó.
- A mi esposa.
- A mi familia, incluyendo mi abuelita.
- A ti por estar leyendo esto.
- Dr. Yu Tang Xu, por su apoyo y comprensión.
- DEPFI, por su espacio prestado.
- Y a todo aquel, que sí apoyo el proyecto.

☪☪☪

☪☪☪

☪☪☪



Dedicado a Sara, mi madre.
Por haberme apoyado en todo momento.

A Leobardo Anaya por tantos momentos, años y décadas de amistad; y apoyarme sin condición... gracias hermano.

A mi hermano Aaziz por enseñarme a comprender más y mejor la vida.

A Raúl, Juancho, Alvaro, Edgar, Rafael, Alfredo y Rosy, Quique, Dani, Thaís, María Elena por llenar una parcela de mi vida, por rocanrolear juntos y por los recuerdos que tengo de cada uno de ustedes.

A Pablo, Ricardo, Luis, Miguel, Héctor por ser compañeros y amigos que siempre recordaré.

A mis tíos Dora, Evangelina y Armando, Emelia y Raymundo, Sofía, Ofelia, Felipe por el asilo que me dieron durante tantas vacaciones... A mi abuela Margarita por el amor que nos da a todos... A todos mis primos y primas... en especial a Luis por compartir tu tiempo y espacio conmigo.

A Rubén, Cor, Daniel, Betito, Arturo, Luis, Lulú por darme la mano y acompañarme a través de este largo y sinuoso camino y que fue mas divertido gracias a ustedes.

La cosa más bella que podemos experimentar es lo misterioso. Es la fuente de toda verdad y ciencia. Aquel para quien esa emoción es ajena, aquel que ya no puede maravillarse y extasiarse ante el miedo, vale tanto como un muerto: sus ojos están cerrados... Saber que lo impenetrable para nosotros existe realmente, manifestándose como la prudencia máxima y la belleza más radiante que nuestras torpes capacidades pueden comprender en sus formas más primitivas... este conocimiento, este sentimiento se encuentra en el centro de la verdadera religiosidad. En ese sentido, y sólo en ese sentido, pertenezco a las filas de los hombres religiosos devotos.

Albert Einstein,
Lo que creo (1930)

Sergio Vera

RESUMEN

El presente trabajo es el resultado del desarrollo de un Robot Móvil (RM), creado en el Laboratorio de Control de la DEEFI, UNAM.

El objetivo principal de esta tesis es el diseño y construcción de un prototipo de un robot móvil (RM); así como la programación para el control de tareas básicas, como es el control de trayectoria: dirección y velocidad.

El robot móvil es denominado de esta manera ya que es capaz de realizar una trayectoria definida, y capaz de realizar la ley de control independientemente.

La trayectoria es definida por medio de una PC, en un programa en Turbo C, los datos los introduce el usuario, y posteriormente son transmitidos vía puerto serial al microcontrolador del Robot Móvil, el cual realiza la trayectoria; al finalizar la trayectoria el RM esta en condiciones de recibir un nuevo conjunto de datos, para realizar otra trayectoria.

Los tipos de trayectorias a realizar son un círculo de radio variable, o una figura regular entre 3 y 7 lados (de lado variable), o una trayectoria dada por el usuario compuesta de líneas rectas y arcos de circunferencia. A todas las trayectorias se les da una velocidad de recorrido, y sentido de giro. Las figuras regulares están compuestas de lados, mas arcos de circunferencia en los vértices.

ÍNDICE

| | |
|-----------------------------------|----|
| 1 INTRODUCCIÓN | 1 |
| 1.1 Introducción a robots móviles | 1 |
| Robots | 1 |
| Robots Móviles | 3 |
| 1.2 Motivación | 3 |
| 1.3 Objetivos | 4 |
| 1.4 Organización de la tesis | 5 |
| 1.5 Presentación | 6 |
| | |
| 2 DISEÑO | 7 |
| 2.1 Mecánico | 7 |
| Robots móviles articulados | 7 |
| Robots móviles de tracción | 7 |
| 2.2 Electrónico | 8 |
| 2.2.1 Amplificadores de potencia | 8 |
| 2.2.2 Circuitos periféricos | 9 |
| Ultrasónicos | 9 |
| Infrarrojos | 9 |
| Velocidad | 10 |
| 2.2.3 Sensores | 10 |
| 2.2.3 Triangulación | 10 |
| 2.3 Microcontrolador | 12 |
| 2.4 Trayectorias a seguir | 16 |
| | |
| 3 CONSTRUCCIÓN | 17 |
| 3.1 Eléctrico | 18 |
| 3.1.1 Alimentación y protección | 18 |
| 3.1.2 Drivers | 18 |

| | |
|--|----|
| 3.1.3 Motores | 20 |
| 3.1.4 Sensores | 20 |
| Posición | 22 |
| De golpe | 22 |
| 3.1.5 Circuitos ultrasónicos | 23 |
| Transmisor | 23 |
| Implementación física | 24 |
| circuito receptor | 25 |
| Infrarrojos | 27 |
| 3.1.6 Tarjetas | 28 |
| 3.2 Mecánico | 29 |
| 3.3 Programación | 30 |
| 3.3.1 Trayectorias | 30 |
| 3.3.2 Estructura del programa del RM | 31 |
| Puertos | 32 |
| 3.3.3 Manejo de datos | 33 |
| Paquetes de datos | 33 |
| Bit de control | 34 |
| Palabra de signo | 35 |
| 3.3.4 Bloques de recepción de datos y lectura de la memoria | 36 |
| Recepción | 36 |
| Lectura | 36 |
| 3.3.5 Mediciones | 37 |
| Distancia | 37 |
| Angulo | 37 |
| Velocidad | 38 |
| 3.3.6 Programa en lenguaje C | 38 |
| Calculos realizados | 39 |
| 3.3.7 Etapa de control | 41 |
| Diagramas de Flujo | 44 |

| | |
|--|----|
| 4 CONTROL | 51 |
| 4.1 Estructura de controladores | 51 |
| Controlador P | 51 |
| Controlador PI | 53 |
| Controlador PD | 54 |
| Controlador PID | 54 |
| 4.2 Sintonización. | 55 |
| 4.3 Digitalización de un controlador PID | 57 |
| 4.4 Control de un motor de CD | 58 |
| | |
| 5 EXPERIMENTOS y PROBLEMAS | 61 |
| 5.1 Experimentación | 61 |
| Etapa de potencia y motores | 61 |
| Sensor de velocidad | 61 |
| 5.1.2 Limitaciones en trayectorias | 62 |
| Angulo | 62 |
| Velocidad | 62 |
| Distancia | 62 |
| Lógico | 63 |
| Ultrasonido | 63 |
| Rafagas | 65 |
| 5.2 Mecánica | 65 |
| Problemas dirección | 65 |
| 5.3 Parámetros del PID, gráficas. | 67 |
| Experimentación trayectorias | 71 |
| | |
| 6 CONCLUSIONES | 72 |
| 6.1 Generales | 72 |
| 6.2 Particulares | 72 |
| 6.3 Problemas | 74 |

| | |
|------------------------------|-----------|
| Sistema de ultrasonido | 74 |
| 6.4 Ventajas y desventajas | 75 |
| 6.5 Mejoras o propuestas | 76 |
| | |
| 7 APÉNDICE | 78 |
| 7.1 Triangulación | 78 |
| 7.2 Definiciones utilizadas | 79 |
| Listado programa ensamblador | 80 |
| Listado programa en C | 91 |
| Diagrama HC11F1 utilizado | 98 |
| | |
| 8 BIBLIOGRAFIA | 99 |

1 INTRODUCCIÓN

1.1 INTRODUCCIÓN A ROBOTS MÓVILES.

Robots

Uno de los motivos para la creación de los robots es que desde hace muchos años el hombre ha querido liberarse de tareas rutinarias y pesadas, así como de reproducir a la naturaleza en diversas formas y con diferentes propósitos. Por ejemplo Leonardo da Vinci ya estudiaba la anatomía de los animales con el propósito de emular sus movimientos.

Los precursores de los robots son las máquinas industriales automatizadas que realizaban tareas de manufacturación automatizada, sin embargo, estas eran inflexibles, es decir, servían para una sola tarea determinada y, por lo tanto, para la producción de un solo producto; una vez que se decidía terminar con la tarea a la que estaban dedicadas, estas máquinas dejaban de servir.

Los robots a diferencia de las máquinas anteriores se caracterizan porque pueden realizar diferentes tareas predeterminadas. Según el *Robot Institute of America* un robot industrial se entiende como "... un manipulador multifuncional reprogramable para la realización de varias tareas". Es por esta razón que la robótica tiene un carácter multidisciplinario, comprendiendo áreas tales como mecánica, control, electrónica, informática, etc.

Existen en la actualidad diferentes tipos de robots tanto en forma, como en propósito. La siguiente es una primera clasificación de los mismos:

- Manipuladores: son sistemas mecánicos multifuncionales y cuyo control puede ser manual o bien, de secuencia variable, se altera el ciclo de trabajo. Según la definición anterior no se consideran como robots.
- Robots de aprendizaje: repiten una secuencia de movimientos con la intervención de un operador y que son reprogramables.
- Robots inteligentes: sistemas mecánicos multifuncionales controlados por computadora que son capaces de relacionarse con el entorno por medio de sensores y tomar decisiones.

Robots Móviles

Los Robots Móviles son mecanismos capaces de moverse de un lugar a otro siguiendo trayectorias determinadas y en ciertos casos pueden determinar la ruta mas adecuada en base a lo sentido del medio ambiente.

El uso de los robots móviles es variado:

- a) se le puede encontrar tanto en el traslado de productos, como puede ser dentro de una fábrica siguiendo una trayectoria definida.
- b) como de guardián de la misma, verificando la seguridad en horas de descanso, vigilando que no haya personas ajenas en las instalaciones.
- c) sensando el ambiente para prevenir incendios o fugas de gases u otras sustancias.
- d) y, tal vez la más importante, es la capacidad de explorar terrenos o áreas inaccesibles y/o peligrosas para el hombre.

La necesidad de acceder a ciertos lugares como incendios, en los cuales está presente el riesgo de explosión o en una mina con cavidades inadecuadas para la exploración o, en casos extremos como lugares contaminados por radiación; hacen de los robots móviles una posible solución. Equipados con sensores y visión

los robots móviles ya han llegado a ser utilizados como exploradores submarinos autónomos.

A pesar de la poca literatura sobre robots móviles, existen libros sobre temas relacionados como son la inteligencia artificial y visión, fundamentales para el desarrollo de estos y en los cuales se pone de manifiesto la importancia y el uso de los robots móviles. Por otra parte, no se tienen sólo proyectos de investigación, este tipo de tecnología ya ha sido puesta en práctica al haberse enviado un robot móvil a Marte para la investigación *in situ*.

Una meta en robótica es crear robots autónomos, tales robots aceptarán una descripción de tareas y las ejecutarán después sin la intervención de la mano humana, tomando sus propias decisiones. La descripción de entrada especificará que es lo que el usuario quiere hacer, más que como hacerlo.

1.2 MOTIVACIÓN

Nuestra motivación es hacer de esta investigación un primer paso para el desarrollo de un robot móvil con capacidad de seguir una ruta indicada.

Las tareas las debe realizar en forma autónoma, no debe haber intervención de una persona que guíe al robot sobre la trayectoria a seguir o las decisiones a tomar. Este tipo de tareas permitiría a un robot seguir instrucciones por medio del reconocimiento de voz, ir de un lado a otro dentro de una fábrica o como ayudante casero de alguna persona discapacitada, o simplemente una exploración o recorrido de una trayectoria en lugares inaccesibles para el hombre.

Otro punto que nos motivó a la realización de este trabajo es el poder construir y tener nuestro robot propio, y hacer una tesis aplicada.

1.3 OBJETIVOS

El objetivo principal de esta tesis es la construcción de un prototipo de un robot móvil; así como la programación para el control de tareas básicas, como es el control de trayectoria (dirección) y el de velocidad. Las tareas básicas serán la realización de ciertos tipos de trayectorias, desglosadamente:

- Construir y diseñar la parte electrónica y mecánica del prototipo en cuestión.
- Que el robot cumpla con cierta trayectoria asignada.
- Aplicar la ley de control PID (Proporcional Integral Derivativo), para relizar el control del robot móvil (control de dirección y de velocidad).
- La implementación de posibles sistemas básicos de sensores para el reconocimiento de ciertas variables (medio ambiente).
- Encontrar las ventajas y desventajas del modelo, y posibles aplicaciones.
- Proponer ciertas mejoras al modelo dados los problemas e inconvenientes del modelo actual.

1.4 ORGANIZACIÓN DE LA TESIS

Este trabajo está dividido en los siguientes capítulos, además del presente:

- **Diseño:** en este capítulo se presentan las partes relacionadas con el diseño de todos los circuitos electrónicos, parte mecánica y parte lógica asociada a la programación.
- **Construcción:** en las tres partes de este capítulo se nos da una visión como se fueron planteando los componentes y elementos de RM. Los capítulos de diseño y construcción están muy ligados, ya que fueron llevados ambos de la mano, debido a que los experimentos nos daban resultados favorables o negativos que afectaron tanto la parte de diseño como la de construcción.
- **Control:** se da una revisión breve a la teoría de control, relacionando a los pasos necesarios para la realización de un PID digital.
- **Experimentación:** este capítulo se plantea en dos apartados, la primera enfocada a la experimentación (pasos para la obtención de los parámetros de la ley de control y selección de algunos dispositivos), y la segunda a los problemas con que nos enfrentamos y las razones por lo que algunas de las partes del diseño no fueron llevadas a cabo.
- **Conclusiones:** se dan los resultados obtenidos en este trabajo, lo que se logró, si se cumplieron los objetivos y lo bueno que se puede tener y/o aprender del mismo.

- Apéndices: en él se presentan los listados de los programas utilizados.
- Bibliografía: nos da una lista de los libros consultados a lo largo de este trabajo.

1.5 PRESENTACIÓN

El presente trabajo es el resultado del desarrollo de un Robot Móvil (RM), creado en el Laboratorio de Control de la DEEFI, UNAM.

Este robot es capaz de recorrer una trayectoria pedida¹, el cual es regido por medio de un microcontrolador.

¹Con ciertas limitaciones, posteriormente mencionadas.

2 DISEÑO

2.1 MECÁNICO

Aunque últimamente se desarrollan Robots Móviles tanto aéreos como submarinos, la mayoría de los Robots Móviles son terrestres, de estos últimos existen de dos tipos: los articulados de dos hasta varias patas y los de tracción montados en 3, 4 o 6 ruedas. Cada uno con sus propias ventajas sobre los demás.

Robots Móviles Articulados.

Estos robots se caracterizan por el gran número de articulaciones que tienen, desde una por pata hasta múltiples articulaciones simulando extremidades anatómicas antropomorfas o arácnidas. La principal ventaja que tiene este tipo de robots es su capacidad de adaptación y movilidad en terrenos con desniveles muy pronunciados, como planos de transición de una altura a otra o terrenos muy accidentados. Se logran salvar estos obstáculos cuando se apoyan algunas de las articulaciones en un plano y el resto de ellas en otro. Su principal desventaja es el control de todas las articulaciones y que son relativamente lentos.

Robots Móviles de Tracción.

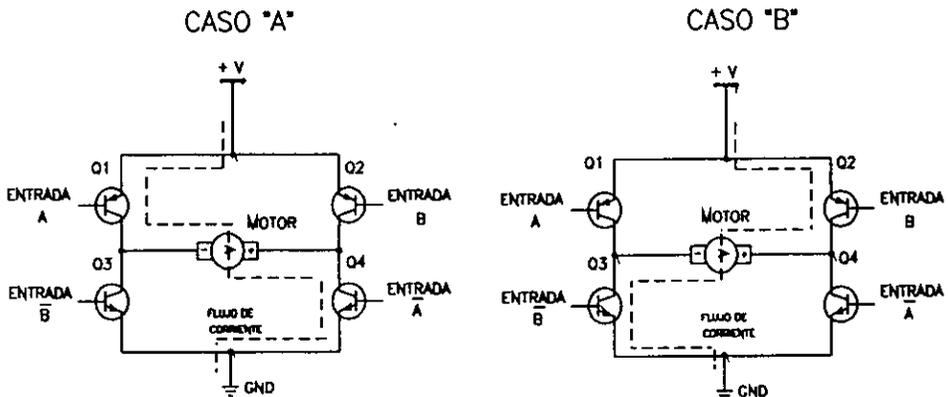
El funcionamiento más sencillo de este tipo de robots los hace un mejor candidato para el diseño de nuestro robot. No existen tantas complicaciones para su control. Pueden llevar más dispositivos periféricos y manejarlos de una manera más sencilla. Se pueden hacer modificaciones de una manera más fácil que en el caso de los robots articulados.

En nuestro caso se escogió un modelo de cuatro ruedas, un par para la tracción y otro para la dirección, de esta manera se pueden recorrer terrenos no muy accidentados a una velocidad relativamente rápida.

2.2 ELECTRÓNICO

2.2.1 Amplificadores de potencia

Dado que el microcontrolador no es capaz de entregar una corriente suficientemente grande para el funcionamiento adecuado de los motores, se hace indispensable el uso de una etapa intermedia entre el microcontrolador y los motores. Se diseñó un circuito H modificado con el cual se controla tanto la dirección de los motores como la velocidad por medio de PWM. Las bondades de este circuito residen en que, con una sola fuente, se puede polarizar un motor de corriente directa para que gire en ambos sentidos. Se utilizan transistores de potencia de 1A, que es mayor a la corriente que manejan los motores. Se muestra un circuito H en el que se advierten cuatro transistores que se activan en pares para dejar pasar la corriente en un sentido del motor o en otro. Estos transistores se activan por un circuito



lógico que determina la dirección y potencia. Para este propósito se utilizan dos pines para la dirección y un tercer pin para regular la potencia por medio de un PWM.

En la figura se muestra el circuito H, con las dos entradas para activar el sentido de giro, y el flujo de la corriente.

2.2.2 Circuitos periféricos

Ultrasónicos.

Para el manejo de los sensores se diseñan circuitos para los sensores transmisores y los receptores. Dado que los ultrasónicos funcionan exitándolos con una frecuencia de 40 KHz, se diseña un circuito oscilante utilizando un circuito integrado LM555 con un ciclo de trabajo de 50 % aproximadamente y unos transistores de salida para los transmisores ultrasónicos.

En la etapa de la recepción se utilizan amplificadores operacionales y filtrado para el acoplo de la señal recibida con el microcontrolador.

Infrarrojos

En el caso de los transmisores infrarrojos (LED's infrarrojos), el circuito que los activa es un circuito oscilante que modula la señal del LED para que tenga más alcance. El circuito receptor de infrarrojos está formado por un amplificador, un comparador y una etapa TTL para dejar la señal adecuada al microcontrolador.

Velocidad.

Se utiliza un circuito integrado que convierte una señal de frecuencia en una de voltaje, este circuito sirve para leer la velocidad de la rueda de tracción.

2.2.3 Sensores

Se utilizan sensores ultrasónicos e infrarrojos para la localización del RM, se realiza por medio de un algoritmo de triangulación, que es como se logra la ubicación del RM en el plano. La localización sirve para la implementación de un control de trayectoria.

También se utilizan estos dos tipos de sensores para librar obstáculos antes de hacer contacto, se utilizan sensores mecánicos o de golpe los cuales se activan en el momento que el RM golpea contra algún objeto.

Se dispuso de un contador binario relativo (codificador incremental) en una de las llantas, conectado a un circuito detector de infrarrojo para la medición de la velocidad y desplazamiento del RM.

2.2.4 Triangulación.

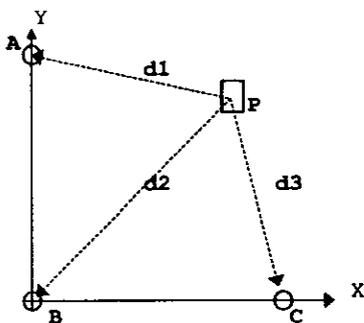
Este método sirve para la localización de un objeto en un plano coordenado por medio de tres puntos radar en el espacio. Es necesario tener la localización de estos tres puntos respecto al plano coordenado.

Para un primer caso es necesario tener la distancia entre los tres puntos y el objeto de interés (d_1 , d_2 , d_3), sólo basta añadir un poco de cálculos geométricos ² y listo.

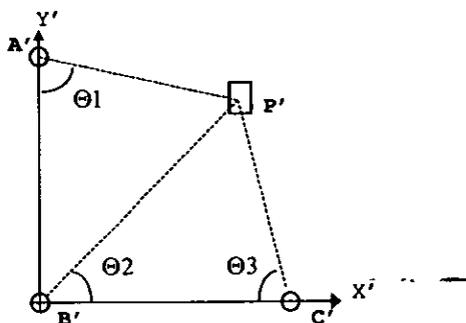
Para un segundo caso es necesario tener el ángulo respecto de la horizontal de las tres líneas que forman los puntos radar y el objeto en interés ($\theta_1, \theta_2, \theta_3$) y el resto es sólo un poco de cálculos trigonométricos.

Se utiliza en ambos casos un transmisor y tres receptores, el transmisor va montado sobre el RM (P y P') y los receptores van en tres puntos radar, en tres lugares dentro del plano coordenado (A , B , C , A' , B' , C').

Caso 1



Caso 2



Para el primer caso utilizamos transmisores y receptores ultrasónicos, y la distancia se mide contabilizando el tiempo que tarda en llegar una ráfaga de ultrasónico entre un punto (P) y el objeto en interés (A o B o C). Se considera la velocidad promedio del sonido a 340 m/s. Con este dato se programan los periodos de los temporizadores en el HC11 para que concuerden los tiempos de

² Se resuelve el sistema de tres circunferencias coincidentes en un punto.

transmisión-recepción de los ultrasónicos y el tiempo de captura en el microcontrolador.

El transmisor lanza una ráfaga cada determinado período con el tiempo suficiente para que los sensores ultrasónicos reciban la señal.³ El tiempo de una sola ráfaga debe ser mínimo con el objeto de hacer lo más rápido posible el periodo entre ráfagas, tomando en cuenta la distancia máxima a trabajar y el tiempo necesario para el encendido y apagado de los sensores.

Para el segundo caso, la triangulación está determinada por los ángulos entre los puntos radar y el RM.

El transmisor de infrarrojo va montado sobre el RM, radiando en todo el plano (360grados), los receptores captan el punto de mayor radiación (ángulo deseado) y con ello se tiene la ubicación del mismo, tomando como dato el ángulo en ese instante. Con estos ángulos se puede conocer la posición según las intersecciones de las tres líneas de los ángulos.

2.3 MICROCONTROLADOR

Se escogió el microcontrolador M68HC11F1 de Motorola por las siguientes razones:

1 Gran cantidad de líneas de entrada/salida, fijas y redireccionables.

Cada uno de estos puertos sirve para conocer el estado de los sensores (mecánicos y electrónicos), para el control de los motores (dirección y tracción) y entradas de información para tomar decisiones que sirven para la posterior realización de la tarea.

³ Las ráfagas de sonido se dispersan de una manera cónica.

2 Un gran mapa de memoria externa en RAM y EEPROM.

Se pueden hacer programas casi tan largos como se requiera, además tiene un puerto para seleccionar dispositivos externos sin entrar en conflicto con la memoria interna. La memoria RAM se utiliza para guardar las variables y también es en donde se alojan los programas para su depuración. El programa final se aloja en la memoria EEPROM, una vez que el microcontrolador opere en forma autónoma. Cuenta con 12Kbytes de memoria RAM y 8 de EEPROM.

3 Conversor Analógico/Digital de 8 canales y 8 bits integrado.

El Puerto E puede operar en dos formas: una como entradas digitales o bien puede operar como convertidor analógico digital. Para nuestros fines dos líneas de este puerto están dedicados a la conversión de entradas analógicas a datos digitales. Este convertidor A/D es de aproximaciones sucesivas, reduciéndose así el tiempo de conversión respecto a otros tipos de convertidores. Tiene una precisión de 8 bits y una exactitud de la mitad del bit menos significativo. Un multiplexor analógico se encarga de las conversiones por canal colocando los resultados en 4 registros diferentes (ADR1, ADR2, ADR3, ADR4), cada uno de estos resultados de las conversiones están controlados por el registro de control ADCTL. El tiempo en que están disponibles los resultados de la conversión es de 128 ciclos de reloj, con un ciclo de trabajo de 2 MHz, la conversión toma entonces 64 microsegundos.

4 Sistema de comunicación asíncrona.

Este sistema nos permite la comunicación serial con una computadora con la cual deseamos intercambiar información respecto a las tareas que debe seguir el robot móvil. Este tipo de información se obtiene de un programa en lenguaje C el cual manda los datos que necesita el microcontrolador para realizar

las tareas. La velocidad más alta permitida con un reloj de 8 MHz que sea compatible con la computadora es de 9600 bauds, que es la que utilizaremos.

La comunicación se realiza por medio de puerto serial asíncrono, RS-232. Con las características de: 9600 bauds, sin paridad, un bit de paro, y se transmiten 8 bits.

5 Sistema de temporizador y captura de entradas.

El Puerto A tiene una entrada dedicada a contar pulsos o eventos; 3 líneas para la captura de entradas, los cuales nos sirven para medir tiempos y 4 líneas para las salidas de comparación, los cuales se activan o desactivan en determinados tiempos programados.

La primer línea utilizada es el puerto PA7, el cual utilizamos como un contador (acumulador) de pulsos y es utilizado para medir la distancia recorrida por el RM.

Con el registro PACTL (Pulse Accumulator CONTROL) se controla el modo en que opera el acumulador de pulsos; se pueden contar eventos o pulsos de reloj cuando está activo el pin PA7/PAI/OC1. El resultado del acumulador de pulsos se almacena en el registro PACNT (Pulse Accumulator COUNTER) y con esto se compara con la distancia que debe recorrer el RM.

La segunda aplicación se utiliza para la detección de señales de ultrasonido. Para esta finalidad se programan 3 de las 4 capturas de entrada, cada una correspondiente a cada receptor de los ultrasónicos.

Se dispara la ráfaga cuando el contador principal TCNT (Time COUNTER de 16 bits) está en ceros = \$0000h. Se programa para que

el tiempo de saturación del contador sea mayor que el tiempo máximo de la señal de los ultrasónicos transmisores llega a los receptores. Este tiempo es guardado en los registros TIC1, TIC2, TIC3 (Time Interrupt Capture) con los cuales se obtiene, a partir de una triangulación, las coordenadas de la posición del RM. Para activar las interrupciones se utiliza el registro TMSK1 (Timer MaSK register 1) con la cual se enmascaran las interrupciones que deseamos utilizar. También se programa para que detecte solamente flancos de subida en los pines respectivos.

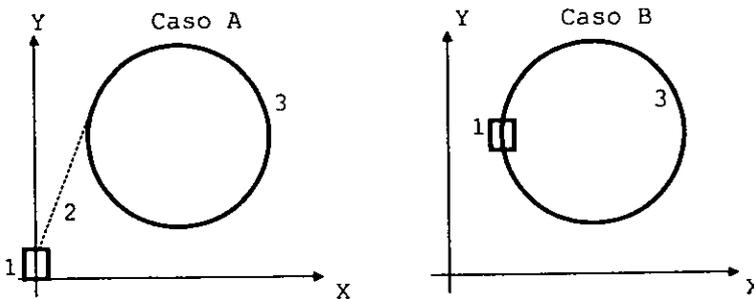
La tercera aplicación es para modular la potencia de los motores del RM utilizando una modulación por ancho de pulso, PWM(Pulse Width Modulation). Este tipo de modulación consiste en fijar un periodo fijo y variar el ciclo de trabajo, como consecuencia se tiene un nivel promedio de potencia de corriente directa. Debido a que se pueden manejar dos pines (PA6, PA5) con tres salidas de comparación (OC2, OC3, OC1 Output Compare register) se programa para que OC1 active a estos dos pines cada vez que el contador principal llegue a ceros. OC2 y OC3 se programan para que desactiven a cada uno de estos pines en el tiempo que se calculó previamente en el algoritmo de control. El tiempo se guarda en los registros TOC2 y TOC3, TOC1 siempre es cero = \$0000. En este caso no es necesario activar las interrupciones ya que la acción de estos pines es automática.

Estas acciones (activar-desactivar) se programan en los registros TCTL1 (Timer ConTroL 1) para desactivar OC2 y OC3 y en OC1M (Output Compare 1 Match) para que tome la decisión de activar o desactivar según el registro OC1D (Output Compare 1 Data), que indica el dato que se va a mandar a los pines PA6 y PA5, que en este caso es un 1 (bits OC1D6, OC1D5 = 1).

2.4 TRAYECTORIAS A SEGUIR

Se están planteando dos objetivos para dos diferentes tipos de trayectorias:

- Caso A. Partir desde cualquier punto y llegar a la trayectoria deseada por medio de una línea recta, y posteriormente recorrerla. (Ver figura)
- Caso B. Partir desde un punto interno de la trayectoria, para la realización de la misma.



Donde

- 1 Robot Móvil
- 2 Recorrido para acercarse a trayectoria a recorrer
- 3 Trayectoria a recorrer.

Programación en C

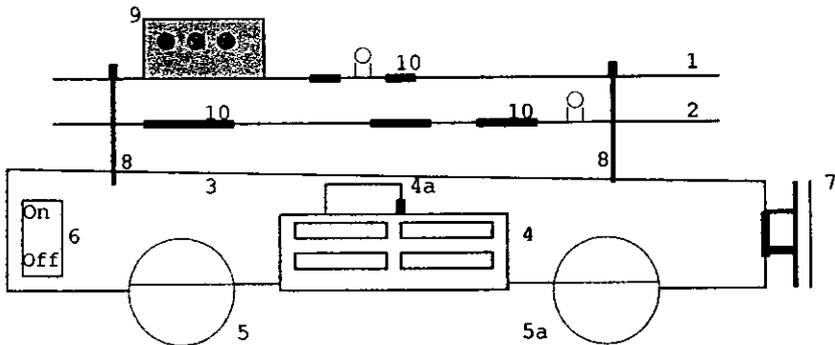
Esta parte se realizó en dicho lenguaje para la introducción de datos y poder realizar ciertas operaciones más fácilmente.

Control

Para la realización del control tanto de velocidad, dirección y posición se plantea un control PID para solucionar nuestras necesidades. Experimentalmente se realizarán la aproximación para tener los valores de los registros asociados.

3 CONSTRUCCIÓN

La construcción física del Robot Móvil quedo tal como está mostrado en la siguiente figura. Está compuesto de tres elementos centrales: la tarjeta de microcontrolador, la tarjeta de periféricos, y la base del Robot (chasis), ésta última comprende la estructura mecánica del mismo, los motores, sensores y espacio para baterías.



Donde:

- 1.- Tarjeta de periféricos.
- 2.- Tarjeta del microcontrolador (HC11F1 de Motorola).
- 3.- Chasis, o base del Robot Móvil.
- 4.- Portapilas en la parte inferior interior del RM.
- 4a.- Pila de 9 V.
- 5.- Llantas tracción.
- 5a.- Llantas dirección.⁴

⁴ Las llantas de dirección pueden realizar un giro limitado y por medio de ello es como da vuelta el modelo.

- 6.- Interruptores (switches) de las tarjetas (2).
- 7.- Barra de sensado de golpe.
- 8.- Tornillos de sujeción de las tarjetas.
- 9.- Disipador de los dispositivos asociados a los motores.
- 10.- Elementos de las tarjetas.

3.1 ELÉCTRICO

3.1.1 Alimentación y protección

Alimentación

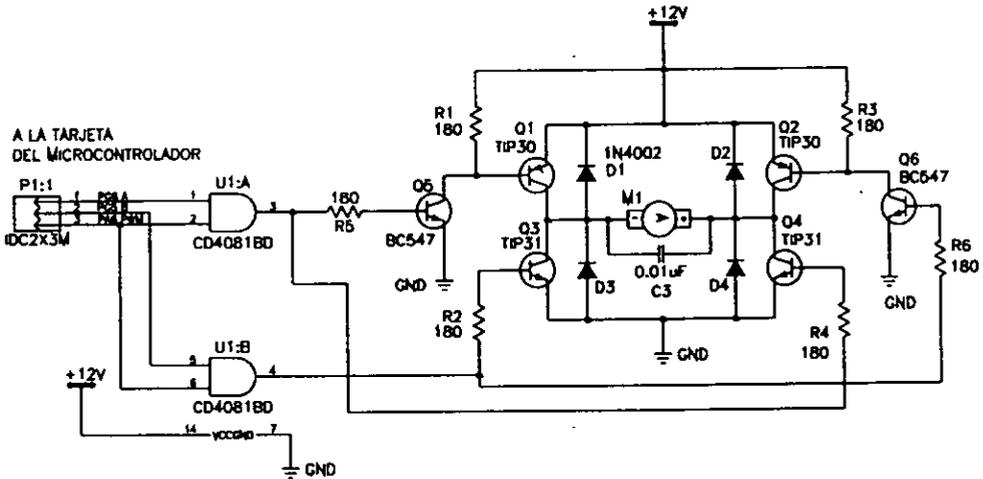
El Robot Móvil está alimentado por dos líneas de alimentación, una de 9V para ambas tarjetas, y la otra proveniente del banco de baterías dedicada a la parte de los motores y sus controladores. Se cuenta con un pila de 9V y un banco de 8 pilas, de 1.5V c/u, para tal propósito. Cada alimentación tiene su propio interruptor. La alimentación de 9V se regula a 5V por medio de un LM7805 en las dos tarjetas.

Protección

Se instaló un fusible de 1A en la parte del banco de baterías, para evitar un posible daño en las pilas, motores o los drivers de los mismos. Se toma un fusible de 1A tomando como referencia un 50% mas del consumo normal máximo de los motores.

3.1.2 Drivers

Los manejadores para los motores (drivers) son un circuito H modificado, tal como se muestra en la siguiente figura:



Driver Motor

Como se mencionó, el circuito H tiene la posibilidad de proporcionar corriente al motor en ambos sentidos con una sola fuente de alimentación. El sentido de la corriente se rige por las líneas respectivas de control. Para nuestro caso tenemos tres líneas de control: PWM, A y B. La función que realizan A y B para cada motor se explica en la siguiente tabla:

| A ⁵ | B | M. Tracción | M Dirección. |
|----------------|---|------------------|----------------|
| 0 | 0 | Alto | Alto |
| 0 | 1 | Avance | Giro izquierda |
| 1 | 0 | Retroceso | Giro derecha |
| 1 | 1 | Acción prohibida | |

En la tabla se muestran las acciones realizadas, tanto para el motor de tracción como para el de dirección.

El circuito H está compuesto en su base por 4 transistores de potencia (Q1-Q4), y dos (Q5,Q6) transistores que proporcionan la corriente necesaria a cada uno de los transistores de potencia, ya

⁵ Se refiere a un cero y uno lógicos.

que la compuerta lógica no la da, además de que algunos funcionan como inversores.

3.1.3 Motores

Se cuenta con dos motores uno para la tracción y otro para la dirección. Ambos son de corriente directa, alimentación nominal de 12V.

Ambos llevan un capacitor en paralelo para evitar el voltaje transitorio causado por la modulación de ancho de pulso.

Potencia a motores

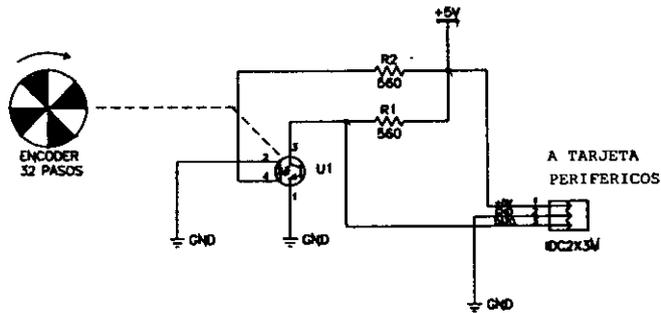
La potencia entregada a los motores es regida por una modulación por ancho de pulso (PWM). La cual tiene un equivalente en DC, por medio de una relación lineal. Así, 100% de modulación equivale al voltaje total de la fuente y 0% a 0V entregados al motor.

3.1.4 Sensores

Distancia

El sensor utilizado para medir distancia es un encoder (codificador incremental) de 32 pasos, el cual va unido a una de las ruedas de tracción.

Se construyó de 32 pasos, ya que el diámetro de las llantas es de 32 cm aproximadamente, con ello tenemos una resolución de un paso por cm. Con ello el programa sólo tendrá que detectar los pasos o transiciones del encoder que equivaldrán a un incremento físico de 1 cm en el desplazamiento del RM.



Encoder.

El encoder está acoplado a un circuito optoelectrónico de herradura, el cual está instalado en la parte inferior del modelo, y da una salida de voltaje proporcional a las transiciones del encoder. Da niveles altos y bajos no lógicos (5 y 0V), los cuales son corregidos a niveles lógicos por medio del comparador, el cual se localiza en la tarjeta de periféricos. La salida del comparador es mandada a la tarjeta del microcontrolador.

Velocidad, conversor de frecuencia a voltaje

Para sensar la velocidad del modelo se utiliza un circuito integrado conversor de frecuencia a voltaje, el LM2907. A la entrada de este integrado se presenta la misma señal del comparador antes mencionado, y la salida es mandada a la tarjeta del microcontrolador. La salida que es entregada es en un nivel de DC entre 0 y 5V, los cuales serán leídos por el microcontrolador para tener el dato de velocidad, en forma analógica.

Algunas de las propiedades del LM2907 son:

- La salida que entrega es un voltaje proporcional a la frecuencia de entrada, de forma lineal.

$$V_{out} = Frec * V_{cc} * R * C$$

donde:

Vout. - voltaje de salida.

Frec. - frecuencia de entrada.

R y *C* - elementos externos al integrado, para el cálculo de la frecuencia.

- Entrega una salida de 0V ante una entrada de frecuencia de 0 Hz.
- La salida de este circuito presenta un pequeño rizo el cual es minimizado ante la presencia de un filtro paso bajas, el cual lo disminuye. Este filtro es pasivo y de primer orden.
- Entrada en niveles lógicos.

Por medio de *R* y *C* se escoge la frecuencia máxima de entrada que equivaldrá a *Vcc* en la salida, *Vout* siempre es menor o igual que *Vcc*.

Posición

Para el sensado de posición de las ruedas delanteras se acopló un potenciómetro como detector de giro.

Este sensor manda una señal de voltaje proporcional a la posición de la ruedas delanteras, la salida del mismo va a la tarjeta del microcontrolador. Su polarización es de 0 y 5V, con dirección en reposo⁶ en 2.5V aproximadamente.

De golpe

El RM cuenta con dos sensores de golpe (microswitchs) colocados en la parte frontal, sirven para el sensado de un objeto físico que se interponga en el recorrido del RM. Estos sensores mandan su

salida a la tarjeta de microcontrolador y van polarizados en niveles lógicos. Ambos están montados sobre una barra abisagrada que permite el correcto sensado en todo el frente del RM. Hay un sensor en cada lado de la barra.

3.1.5 Circuitos Ultrasónicos

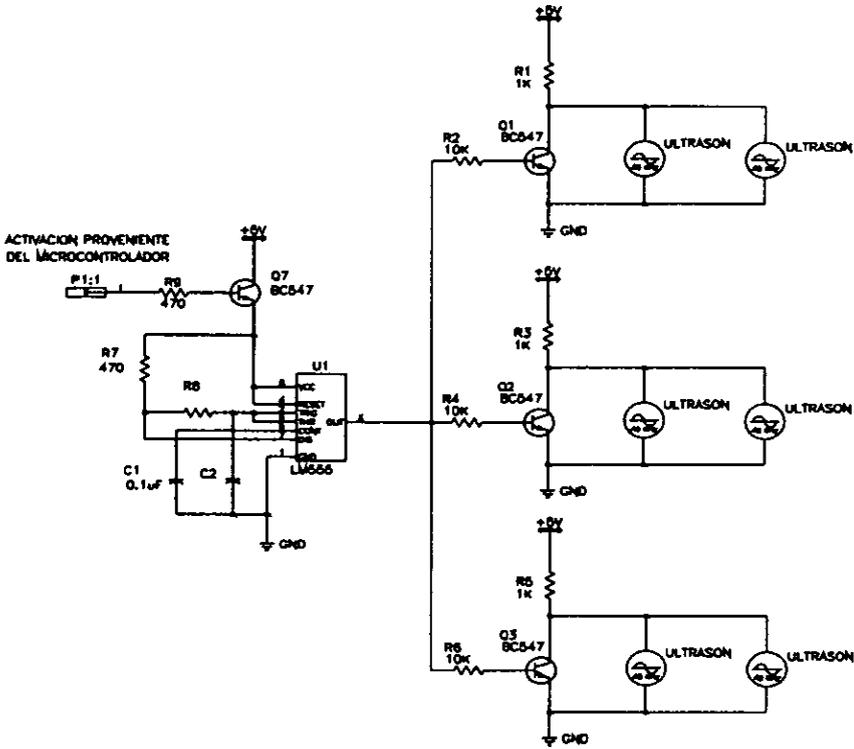
Para el sensado de la posición real del RM se planteó la siguiente solución, un transmisor (de ultrasonido) montado sobre el modelo y tres conjuntos de receptores montados en las esquinas de un triángulo.

Circuito Transmisor

El circuito transmisor está montado en la tarjeta de periféricos, es un arreglo de seis transmisores, los cuales irradian en una superficie horizontal las ráfagas.

El circuito es activado por medio del microcontrolador, el cual da la orden de ráfaga a un LM555, que activa los transistores, que trabajan en corte y saturación.

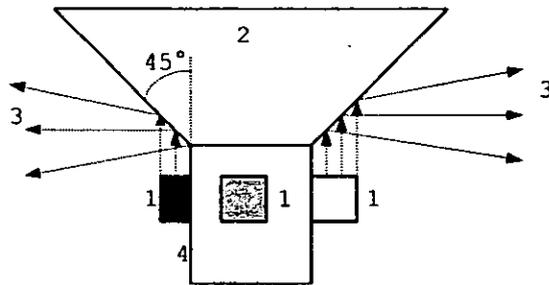
⁶ Se entiende por reposo cuando estas llantas están paralelas a las traseras.



Cto. Transmisor ultrasónico

Implementación física

La implementación para que radiara en los 360 grados fue montar los 6 transmisores sobre un cono, el cual refleja el haz de forma horizontal, tal como se muestra en la figura.



Cono vista lateral

Donde:

- 1 Transmisores ultrasónicos.
- 2 Cono reflector.
- 3 Haz (ráfaga) reflejado horizontalmente.
- 4 Base cilíndrica del cono, a la cual van unidos los transmisores.

El cono está hecho de cartulina, la cual mostró reflejar correctamente el haz del transmisor.

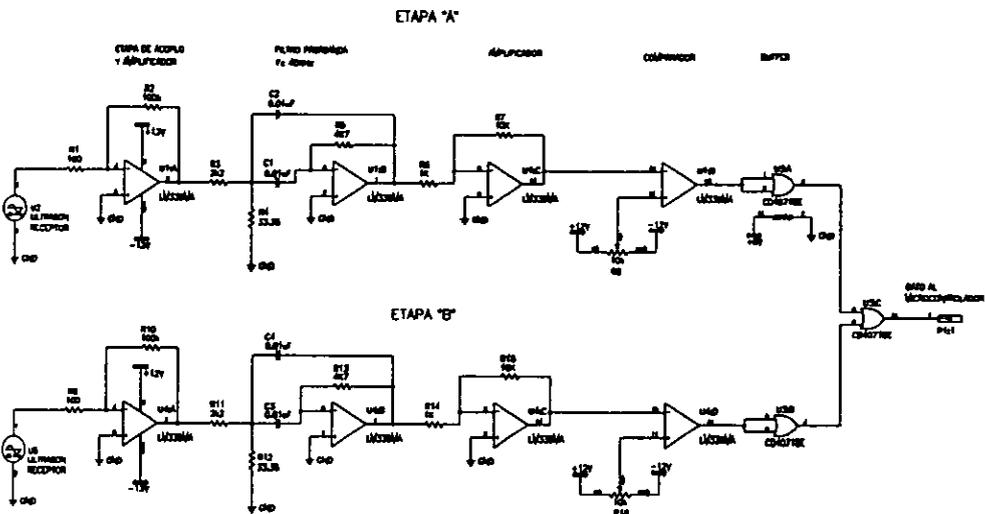
Se requirieron de 6 transmisores, a 60 grados cada uno, para poder irradiar los 360 grados a lo menos en 3m. El cono al estar a 45 grados irradia el haz en forma horizontal que en un inicio era vertical.

Circuito receptor

Se colocó un circuito de recepción ultrasónico en cada una de las esquinas de un triángulo (3m por lado), mirando hacia el centro del mismo. En cada una de las esquinas existen dos sensores receptores de ultrasonido y el circuito asociado.

El circuito receptor está formado por varias etapas, las cuales son:

- Receptor, sensor de ultrasonido.
- Primer amplificación, de 60db.
- Filtrado, filtro paso banda, de banda reducida, $f_c = 40\text{KHz}$, ancho de banda de 6KHz . (rev)
- Segunda amplificación, de 20db.
- Comparador y recortador, se compara y recorta para tener una salida en niveles lógicos.
- Suma de señales. Es necesario sumar las señales de las etapas en paralelo para el mayor rango de visión.



Cto. Receptor Ultrasonido

Como se aprecia en la figura los sensores trabajan en paralelo y posteriormente de manipular a señal, las sumamos.

La implementación física de la parte de los sensores se realizó colocando los sensores a 40 grados de separación, si cada sensor tiene un alcance aproximado de 60 grados, con ello se obtuvo la

vista de toda el área dentro del triángulo desde el vértice del mismo.

Infrarrojos

Transmisor

El requerimiento es radiar 360 grados en un plano horizontal, similar al ultrasónico.

En un primer caso se pretendió tener un transmisor que radiara los 360 grados del infrarrojo, pero requería un alto consumo de corriente y aproximadamente 12 transmisores (leds), ya que un transmisor tiene un lóbulo de 30 grados aproximadamente.

En un segundo caso se pensó en un led giratorio para radiar los 360 grados, esto no se logro por causa del ruido introducido en el RM.

Una tercer solución fue hacer girar un espejo a 45 grados, por medio de un motor, para radiar los 360 grados horizontalmente, se tiene un transmisor apuntando, dirigido, al espejo. El espejo gira de 10000 a 15000 rpm, con ello se conseguía una buena radiación.

Receptor

Ya que se tiene el transmisor el receptor será giratorio y hará un barrido al área de trabajo con el objetivo de captar el ángulo en el que existe esta máxima radiación. El receptor va acoplado a un potenciómetro el cual proporciona el ángulo.

El barrido es realizado por un motor que hace girar al receptor. Se cuenta con un filtrado en baja frecuencia para eliminar el ruido, y un derivador para tener el punto de máxima pendiente.

El circuito receptor en la parte de la lectura del led, está conformado por una etapa de amplificación, un filtro paso banda,

en la frecuencia de modulación (10kHz) y una segunda amplificación.

Para este caso, infrarrojos, el sistema no funcionó debido al ruido y la carencia de componentes para lograr el circuito deseado, tal como un mejor motor y un sistema mecánico más estable.

3.1.6 Tarjetas

Tarjeta del microcontrolador

Se elaboró una tarjeta en circuito impreso para el microprocesador de Motorola, el diseño de la misma y del impreso incluyendo pistas se elaboró en el paquete TANGO. La tarjeta está basada en el diagrama del manual del HC11, y está diseñada para sustentar nuestras necesidades tanto físicas como electrónicas en los referente a la ubicación de los componentes de la misma. El hecho de que sea en circuito impreso nos redujo muchos problemas como puede ser el de ruido y falsos, al final del apéndice se presenta el diagrama lógico.

Tarjeta de periféricos

Esta tarjeta cuenta con:

- Circuito regulador de voltaje.
- Driver del motor de tracción
- Driver del motor de dirección.
- Circuito conversor de frecuencia a voltaje.
- Circuito transmisor de ultrasonido.
- Interruptores.
- Leds indicadores de voltaje, uno por tarjeta.
- Circuitos asociados a la alimentación.

3.2 MECÁNICO

Para la parte de la construcción mecánica se partió de un carrito a control remoto, es decir, se modificó y no se elaboró la parte mecánica.

Este modelo poseía inicialmente las características de avanzar y girar en niveles discretos, es decir, a toda velocidad o paro, derecha, centro o izquierda en el giro.

Modificaciones.

La parte del modelo con la que comenzamos a trabajar contaba solamente con los dos motores, el de dirección y de tracción, y el soporte físico para las tarjetas, chasis. Se le eliminaron todos los circuitos y partes inútiles para nosotros.

Se desarmó, limpió y engrasaron las partes asociadas a los motores, para un mejor rendimiento. Además se resanaron algunos huecos relacionados a la transmisión del giro, ya que nos daban muchos espacios muertos, se hizo lo posible.

Se acoplaron las tarjetas en la parte superior del chasis (ver figura al inicio del capítulo), se colocó el encoder en la parte inferior, el sensor de posición dentro del sistema de dirección, la barra abisagrada⁷ para los sensores de golpe, el cono de los transmisores ultrasónicos, interruptores, etc.

⁷ La barra es a todo lo largo del frente del RM.

3.3 PROGRAMACIÓN

La programación se divide en dos partes, la dedicada a la programación del HC11, en ensamblador, y la parte dedicada a la programación en una PC, en lenguaje C.

Ambos programas están realizados de una manera modular para una fácil depuración, modificación y entendimiento de las rutinas.

Para la realización de una tarea (trayectoria) por el RM es necesario introducir los datos en una PC y luego transmitir los datos vía puerto serial, posteriormente el RM realizará la trayectoria de manera independiente.

3.3.1 Trayectorias

Las trayectorias son las tareas pedidas por la PC al microcontrolador, para realizar por el RM.

De aquí en adelante denominaremos trayectoria simple⁸ a la trayectoria realizada por un bloque de 4 datos (palabras), y trayectoria global al conjunto de éstas trayectorias.

Las trayectorias simples pueden ser de dos tipos: las líneas rectas y los arcos de circunferencia, dentro de las trayectorias globales tenemos las figuras regulares, los círculos y las abiertas o pedidas por el usuario. Por las limitantes del RM todos los vértices de las figuras regulares serán arcos de circunferencia de radio mínimo (radio de curvatura 45cm).

⁸Una trayectoria simple también ser una con valores de velocidad cero. Ya que lo que está haciendo es realizar la orden recibida.

Para la realización de una circunferencia se requieren de dos trayectorias simples, la de estabilización de la dirección y el recorrido de la circunferencia.

Para el caso de una figura regular de "n" lados se realizan las siguientes instrucciones n veces: estabilización de dirección, avance en línea recta, estabilización de dirección, recorrido del vértice. La estabilización de dirección siempre es para el recorrido siguiente.

Para el caso de una trayectoria global establecida por el usuario, libre, sólo basta seguir el conjunto de trayectorias simples dadas.

3.3.2 Estructura del programa del RM

La estructura del programa del RM va por módulos, quedando como se muestra en el diagrama de flujo 1^º.

- 1.Una inicialización. Incluye todos los parámetros de inicialización del control PID, comunicación con la PC, registros de control, acceso de puertos, etc.
- 2.Un "pushbutton" (inicio), con él damos por entendido que el RM está preparado para la lectura de datos provenientes de la PC.
- 3.Un bloque de recepción de datos. En este bloque el microcontrolador lee los datos provenientes de la PC y los almacena en memoria RAM. Estos datos son los que indican la trayectoria a realizar. Se siguen leyendo datos mientras no se indique el fin de recepción de datos.

^º Los diagramas se presentan numerados al final de capítulo.

4. Un "pushbutton", para indicar que puede comenzar la trayectoria global. Este es necesario ya que si no se utiliza al momento de terminar de leer los datos de la PC, comenzaría el bloque de lectura y control. Al detenerlo con el botón podemos cambiar el RM de lugar y presionar donde deseemos el inicio de trayectoria.
5. Un bloque de lectura de datos. En este bloque se leen los datos de las trayectorias a realizar, por cada vez se leen los datos de una trayectoria simple.
6. Se realiza el control PID para velocidad y dirección.
7. Se siguen leyendo trayectorias simples y realizando su control mientras no sea el fin de la trayectoria global.
8. Se va de nuevo al inicio para el comienzo de una nueva lectura de datos y por consiguiente poder realizar una nueva trayectoria.

Estructura de los Puertos

La forma en que quedaron los puertos del microcontrolador es la siguiente, en ella vemos las variables de entrada, salida y a lo que están conectados.

Puerto A

PA6 Salida, PWM Motor de tracción.

PA5 Salida, PWM Dirección.

PA2, PA1, PA0 Entrada, sensores ultrasónicos. ¹⁰

¹⁰ Los infrarrojos entrarían por el puerto analógico digital.

Puerto E

PE1 Entrada, Conversor del sensor de dirección.

PE0 Entrada, Conversor del sensor de velocidad.

Puerto G

PG6 Salida, Bit A de control de giro de motor tracción.

PG5 Salida, Bit B de control de giro de motor tracción.

PG4 Salida, Bit A de control de giro de motor dirección.

PG3 Salida, Bit B de control de giro de motor dirección.

PG2 Salida, habilitador de ráfaga de ultrasonido para el transmisor.

PG1 Entrada, sensor golpe izquierdo.

PG0 Entrada, sensor golpe derecho.

3.3.3 Manejo de los Datos**Paquetes de datos**

Para cada una de la trayectorias es necesario transmitir de la PC al microcontrolador paquetes de 4 datos, dependiendo del tipo de trayectoria y su complejidad será el número de paquetes de datos a mandar. Cada orden a realizar por el RM requiere de uno de estos paquetes.

Los paquetes de datos están conformados de la siguiente manera, los cuales son los valores deseados que realice el RM:

| | | | |
|-----------|----------------------------|----|----|
| Palabra 1 | B7 Dirección ¹¹ | B1 | B0 |
| Palabra 2 | B7 Velocidad | B1 | B0 |
| Palabra 3 | B7 CiclosH | | B0 |
| Palabra 4 | B7 CiclosL | | B0 |

Las dos primeras palabras del bloque son de 7 bits, ya que el bit menos significativo es de control (adelante explicado), para su proceso se considera a B0 como cero.

La tercer y cuarta palabra, son la parte alta y baja de una palabra de 16 bits, (H alta, L baja), sólo que se transmite en dos partes. Se manejará como una sola palabra de 16 bits, éste dato tiene dos funciones dependiendo de lo que se requiera realizar, la primera sirve para definir el tiempo de estabilización necesario para el control de dirección, esta estabilización se realiza a motor de tracción en alto. La segunda sirve para indicar la distancia a recorrer por el RM, valor dado en cm.

Para poder identificar cuando se requiere alguna de estas dos funciones nos basamos en la variable velocidad como bandera, si es diferente de cero, entonces se realizará un recorrido, en caso contrario la estabilización.

Bit de control

Este bit es el menos significativo de las dos primeras palabras transmitidas, y es utilizado para indicar el fin de transmisión.

Al momento de transmitir datos (útiles) de la PC al microcontrolador el bit de control está en cero, para poder indicar

¹¹De las llantas delanteras del RM.

el fin de la trayectoria global se transmite una palabra con este bit en alto, con ello se indica el fin de transmisión, o fin del bloque de recepción de datos.

El bit de control sirve de paro, pero el bloque que llega con éste bit también es escrito en la memoria, posteriormente es cuando sale del bloque de recepción.

Este bit también es utilizado para indicar el fin de trayectoria global al microcontrolador cuando está en el bloque de lectura de datos y funciona igual que al momento de recibir los datos provenientes de la PC, mientras sea cero seguirá leyendo datos, al momento de ser uno saldrá del bloque de lectura de datos.

Palabra (Bits) de signo

Todos los datos llegados (desde la PC) al microcontrolador están es 8 bits, y son considerados siempre como positivos, mas sin embargo al realizar el control existen datos que se pueden convertir en negativos tales como el error de posición o el de velocidad que en nuestro caso denominamos como la entrada a la etapa de control del PID ($X(n)$), y además también pueden resultar negativos los resultados de los productos de las $X(i)$ por las constantes del PID.

Para poder tener una resolución de 8 bits en la palabra se procedió a tener el bit de signo en una palabra adicional. Esta palabra de signo contiene el signo de los registros utilizados ($X(i)$) para velocidad y posición, (una para cada etapa respectivamente) todos en la misma palabra sólo que en diferente bit. Así el bit i ésimo (B_i) de la palabra de signo corresponde al registro $X(i)$ en uso. En cada muestra lo que se hace es hacer un corrimiento a este registro y con ello corremos los signos junto con los registros.

3.3.4 Bloques de recepción de datos y lectura de la memoria

Recepción

Los datos recibidos (provenientes de la PC) se van almacenando directamente en la memoria RAM, iniciando en la localidad #3000h.¹²

Para la escritura de los datos recibidos se utiliza un direccionamiento indexado, con un registro asociado, el cual se va incrementando a la orden que se recibe cada dato, al terminar el bloque de recepción de datos este registro vuelve a apuntar a la localidad inicio (#3000h) para estar listo para el bloque de lectura de datos.

Ver Diagrama 2A y 2B de recepción de datos.

Lectura

Para la lectura de la memoria se realiza de manera similar a la del bloque de recepción, por direccionamiento indexado, sin embargo éste bloque de lectura lee 4 datos, si no existe fin de lectura va a la bloque de control, a leer otros 4 datos y así sucesivamente, o sea, se va incrementando el registro de índice de 4 en 4.

Se cargan los 4 datos correspondientes en los 4 registros de trabajo y se procede a realizar el control, al fin del mismo se va de nuevo al bloque de lectura.

Ver diagrama de flujo 3A y 3B, de lectura de datos.

¹² La letra "h" al final del número implica base 16, se usará "d" en caso de pueda haber alguna confusión.

3.3.5 Mediciones

Distancia recorrida

Como ya se mencionó el RM posee una resolución de 1 bit por cada cm recorrido. Con un registro de 8 bits (longitud de palabra del micro) la máxima distancia a medir es de 2.55 m lo cual resulta poco atractivo para nuestros planes, por ello es que se usaron 16 bits dándonos una distancia mucho mayor como máximo, algo exagerada, pero útil a nuestra necesidades, de 650 m (65000 cm).

La medición de la distancia se utiliza por medio del registro acumulador de pulsos. Este registro tiene como función el incrementar un registro ante la presencia de un flanco positivo. Conectando en esta línea la señal proveniente del "encoder" de la llanta de tracción podemos medir la distancia recorrida por el RM.

El registro Ciclos (distancia deseada) se carga en el registro "Y" del microcontrolador para poder ser comparado contra la distancia medida (registro auxiliar, Raux, 16 bits), con ello sabemos si llegó a la distancia requerida.

Se emula un registro de 16 bits (Raux) copiándole lo que tiene el acumulador de pulsos a la parte baja, e incrementando la parte alta (de Raux) ante cada saturación (overflow) del acumulador de pulsos.

Medición del ángulo

La medición del ángulo se realiza por medio del convertidor analógico digital proveniente del sensor de posición de la dirección. Simplemente a nivel lógico se realiza una resta para con ello tener el error de posición y mandar este dato a la etapa de control.

Medición de velocidad

La velocidad se mide utilizando el convertidor analógico digital, la señal del circuito convertidor de frecuencia a voltaje, el dato leído es directamente proporcional a la velocidad del RM multiplicado por una constante de 1, es decir son proporcionales en la misma escala.

3.3.6 Programa en Lenguaje C

La función del programa en lenguaje C, es la de facilitar la introducción de los datos en un ambiente más amigable, y el de realizar cierto tipo de operaciones en punto flotante.

En el programa se plantean los tipos de trayectorias existentes, para ser escogida por el usuario y para que proporcione los datos pedidos.

Para las trayectorias en el programa se realizan los cálculos necesarios para transmitir los datos al microcontrolador. Los datos son mandados en formato hexadecimal.

Estructura del programa

El programa está organizado de la siguiente manera:

1. Escoger entre la trayectoria a realizar, entre una figura regular (4 a 8 lados), un círculo, o una trayectoria libre, y la opción de salir del programa.
2. En base a la selección del paso anterior se pide:

- para figura regular la velocidad, el número de lados, y derecha o izquierda del recorrido.
 - para el círculo el radio y velocidad, y derecha o izquierda.
 - para la figura libre, primero si desea una línea recta o un arco, pidiendo la velocidad, y/o radio de curvatura, según el caso, en caso de arco derecha o izquierda.
3. Se realizan los cálculos necesarios.
 4. Se mandan los datos al microcontrolador.
 5. Se regresa al paso 1.

Cálculos realizados

Al microcontrolador se mandan los datos en forma hexadecimal, por ello el programa los tiene que mandar como caracter, "char", el cual va de 0 a 255.

• Recorrido de vértices

Aquí lo que se calcula es la distancia a recorrer para cada uno de los vértices de las diferentes figuras regulares.

Así la distancia a recorrer por vértice está dado bajo la siguiente fórmula¹³ :

$$dv = \frac{2 * \pi * r}{n}$$

donde

- dv : distancia a recorrer por el vértice
- r : radio de curvatura = 45 cm
- n : número de lados de la figura regular.

• Perímetro del círculo

¹³ Los datos mandados al microcontrolador de distancia son en centímetros.

Es el mismo caso que para un vértice, pero de 360 grados.

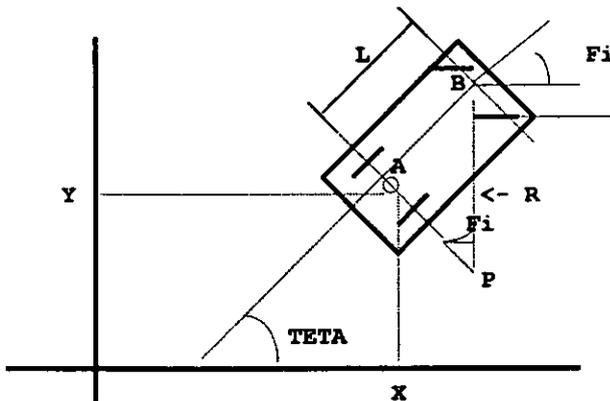
Así

$$p = 2 * \pi * r$$

donde

- p : perímetro
- r : radio del círculo

- Ángulo de giro



Donde

- L : distancia entre ejes de los dos pares de llantas (18.5cm).
- TETA : ángulo del eje principal de RM respecto al eje X.
- $Fi (\Phi)$: ángulo de giro de las llantas de dirección.
- R : radio de curvatura.

A partir de esta figura obtendremos la relación que hay entre el radio de curvatura y el ángulo de giro de las llantas delanteras.

De la figura, tomando el triángulo APB tenemos que

$$\Phi = a \operatorname{sen}\left(\frac{L}{R}\right)$$

Con ello tenemos el ángulo que debe girar las llantas de dirección para cierto radio de curvatura proporcionado.

Para nuestras necesidades tenemos:

$$\theta = a \operatorname{sen}\left(\frac{L}{R}\right) * 57.3 * \left(\frac{96}{28}\right)$$

donde

θ (teta)¹⁴ es el dato a mandar al microcontrolador.

El 57.3 se debe a la conversión de radianes a grados.

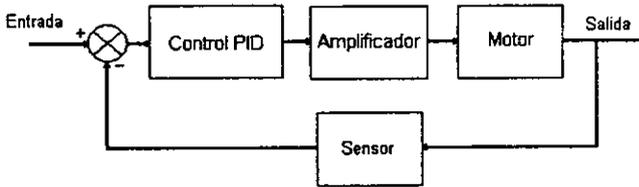
Explicando el cociente 96 / 28: 28 es la máxima torsión en grados en media carrera, 56 en carrera total; 96d = 60h corresponde al desplazamiento digital de la media carrera del potenciómetro, la carrera total va de 20h a CFh; lo que implica una regla de tres al momento de hacer la conversión del dato que queremos mandar.

3.3.7 Etapa de control

La etapa de control está cargada en la memoria del EEPROM del microcontrolador, y los registros de trabajo, así como algunos parámetros del PID van en la memoria RAM.

El diagrama de la parte de control es el siguiente:

¹⁴ Teta para ser mandado al RM debe ser sumado o restado de 127, punto en reposo, dependiendo para donde se requiera el giro, derecha o izquierda.



La ley de control para un PID digital, en ecuaciones en diferencias, es la siguiente:

$$Y(n) = Y(n-1) + A * X(n) + B * X(n-1) + C * X(n-2) \quad \dots (1)$$

donde

$$A = \left(K_p + \left(\frac{T}{2} \right) * K_i + \left(\frac{1}{T} \right) K_d \right)$$

$$B = \left(K_i * \left(\frac{T}{2} \right) - K_p - \left(\frac{2}{T} \right) * K_d \right)$$

$$C = \left(\frac{1}{T} \right) * K_d$$

Además

K_i , K_p , y K_d : constantes del controlador PID.

$X(i)$ error de la ley de control.

$Y(i)$ salida de la ley de control, en el tiempo y .

La ecuación de la ley de control que estamos trabajando es la siguiente:

$$Y(n) = Y(n-1) + \left(\frac{P0}{Q0} \right) * X(n) - \left(\frac{P1}{Q1} \right) * X(n-1) + \left(\frac{P2}{Q2} \right) * X(n-2) \quad \dots (2)$$

Básicamente esta ecuación es la misma que (1), los valores P_i y Q_i representan los valores de las constantes del controlador PID. En la ecuación (1) se tiene que el segundo signo es positivo, pero como el término B resulta ser negativo para periodos de muestreo pequeños por ello es que trabajamos el segundo término como negativo.

Las ecuaciones que asocian los términos P_i y Q_i de la ec. (2) respecto a la (1) son los siguientes:

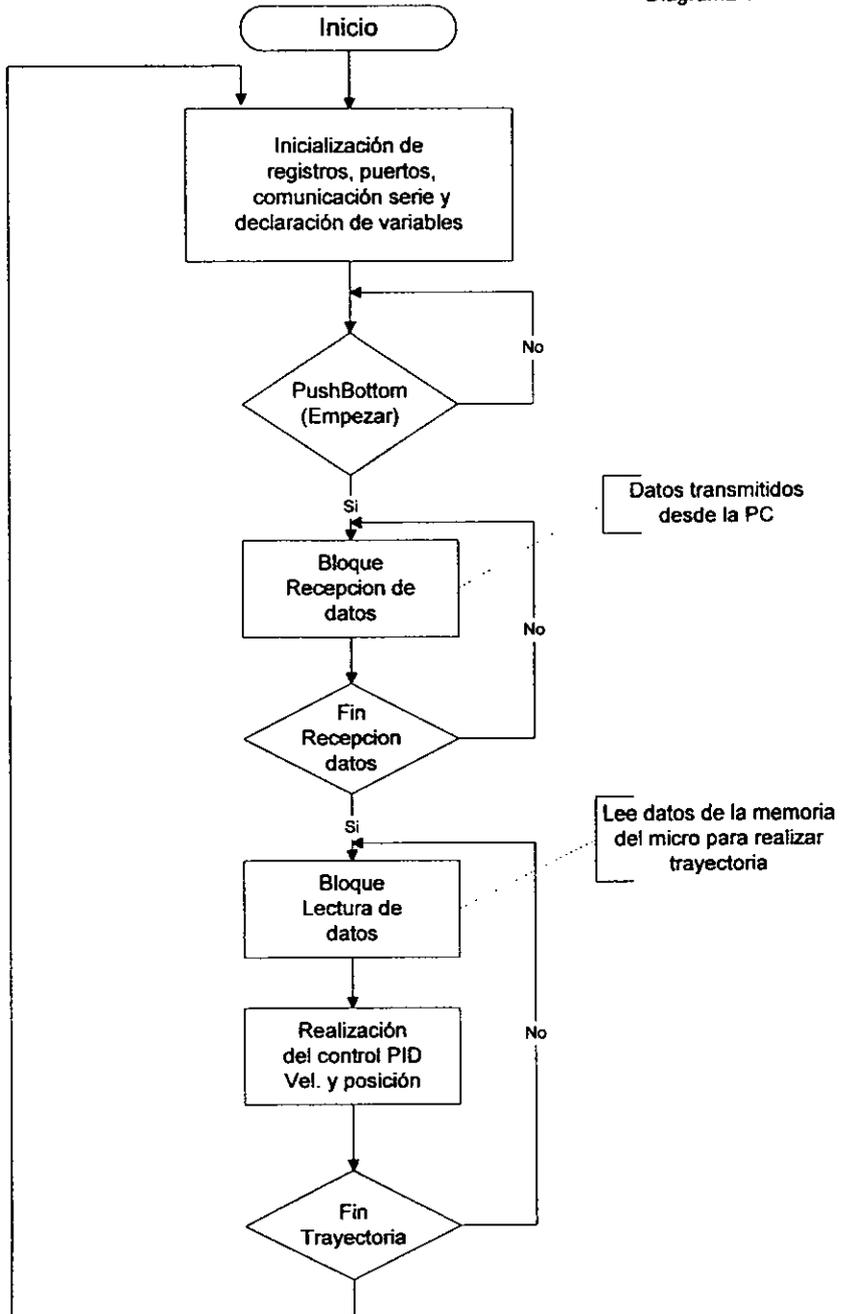
$$\begin{bmatrix} P_0/Q_0 \\ P_1/Q_1 \\ P_2/Q_2 \end{bmatrix} = \begin{bmatrix} 1 & T/2 & 1/T \\ -1 & T/2 & -2/T \\ 0 & 0 & 1/T \end{bmatrix} \cdot \begin{bmatrix} K_p \\ K_i \\ K_d \end{bmatrix}$$

Dado un poco a la complejidad de estar realizando esta operación en base a las constantes K del PID lo que se procedió a realizar fue trabajar con los valores de las P_i/Q_i .

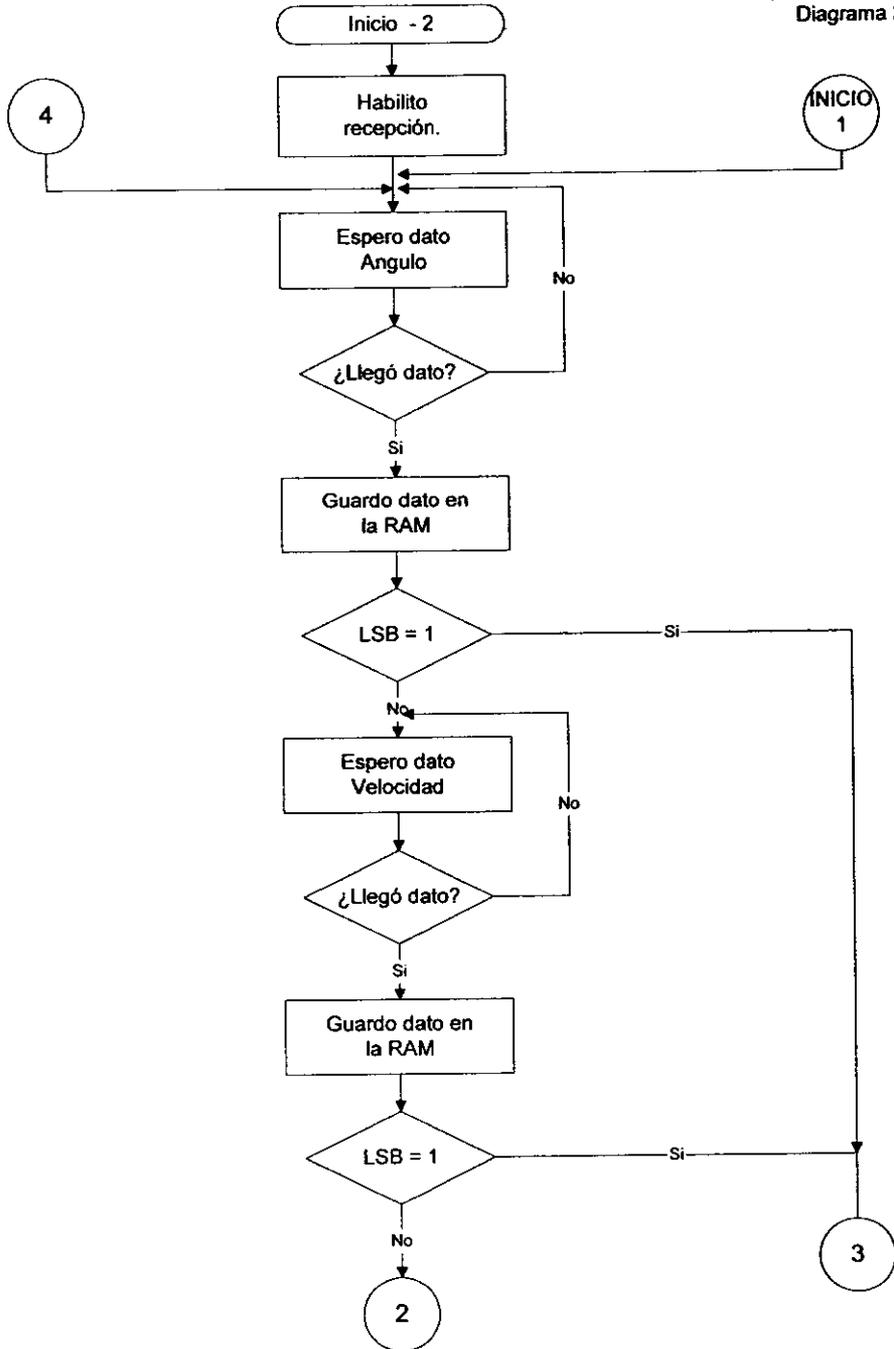
El periodo de muestreo es aproximadamente 5 ms, lo que nos da un total de 200 muestras por segundo, este periodo es aproximado, y va a variar según se varían las líneas del programa en el ensamblador.

Ver diagramas 4A y 4B en donde está el diagrama de flujo de la parte de control.

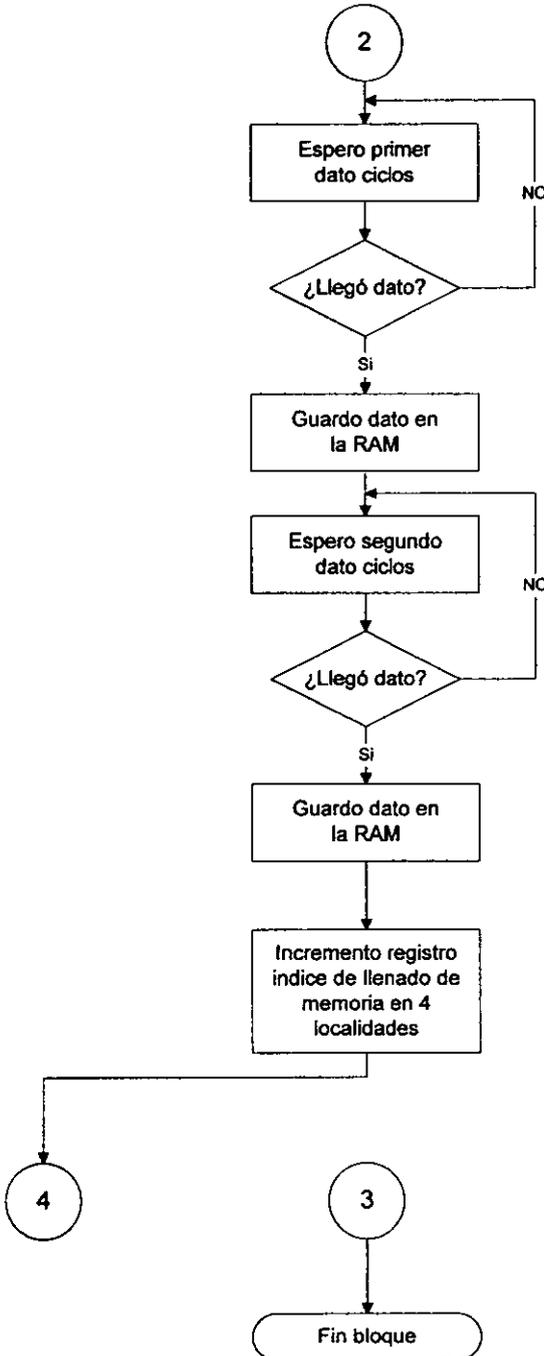
Diagrama general,
programa en ensamblador
Diagrama 1

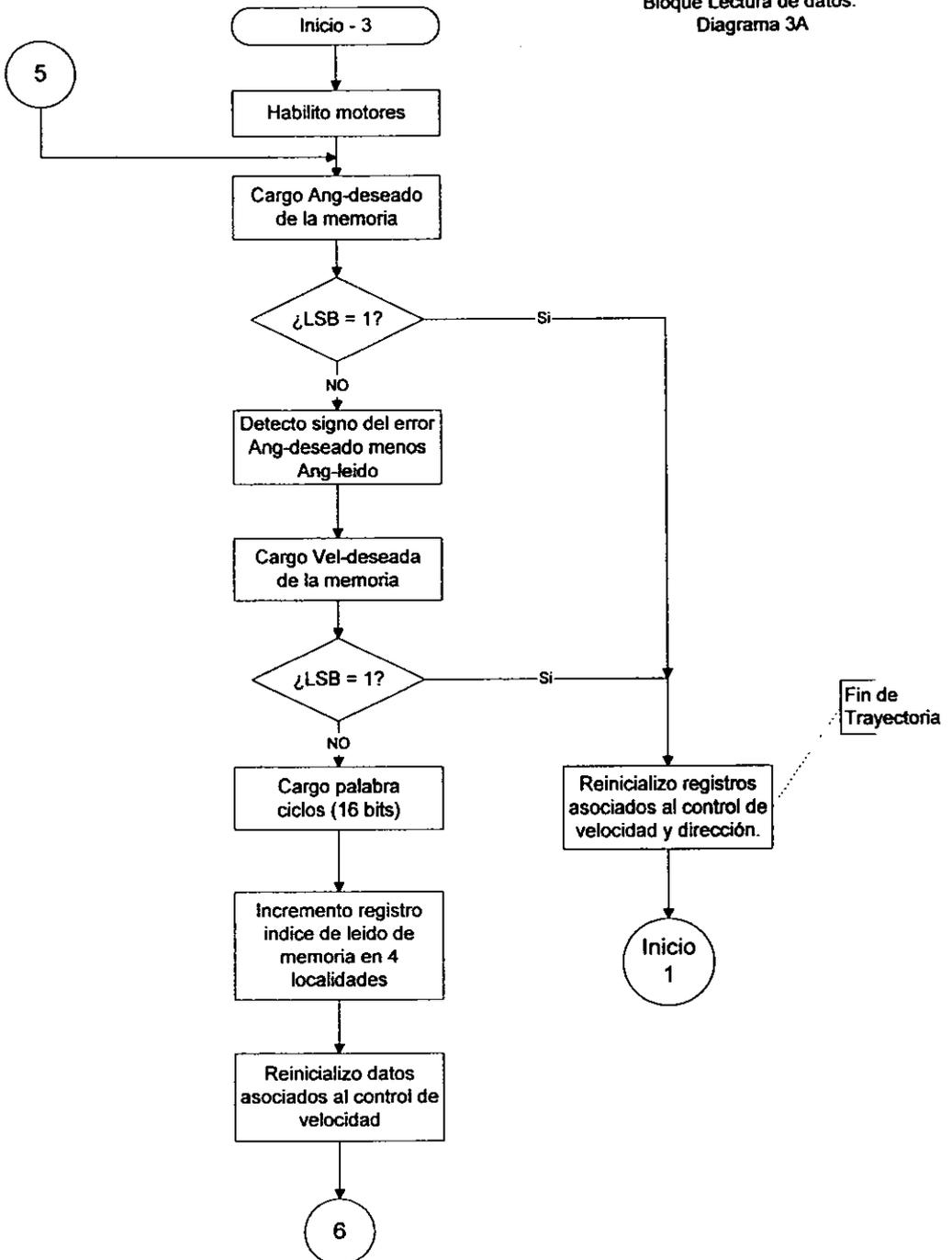


Bloque recepción de datos
Diagrama 2A

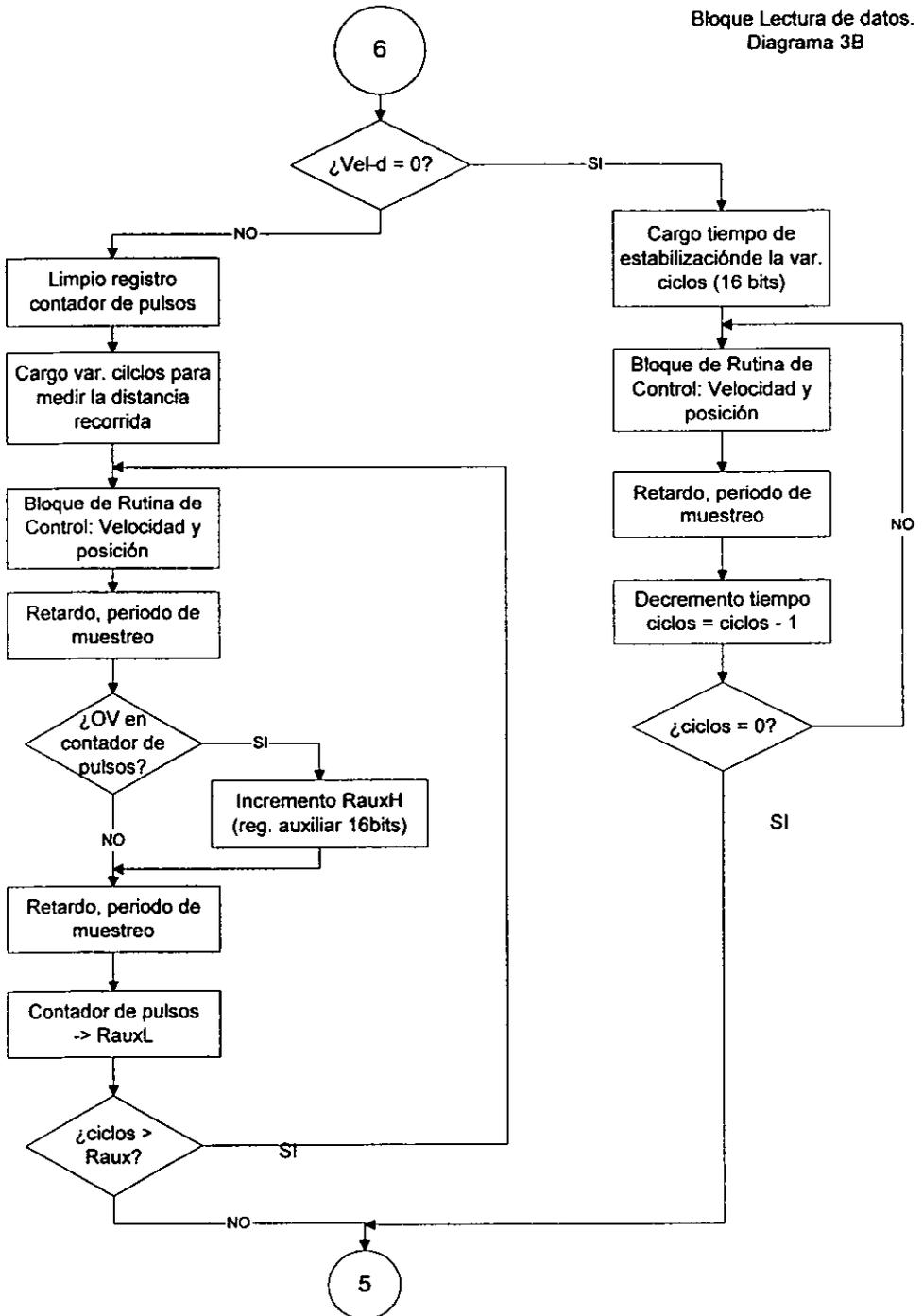


Bloque recepción de datos.
Diagrama 2B

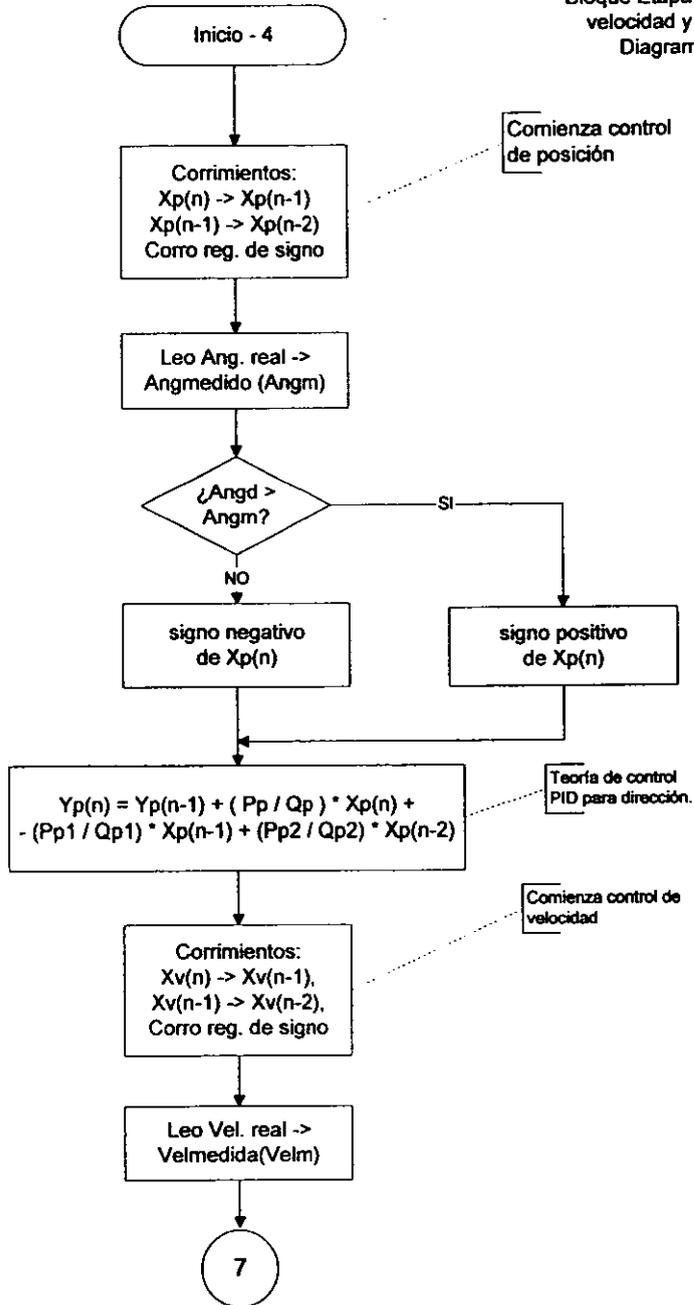


Bloque Lectura de datos.
Diagrama 3A

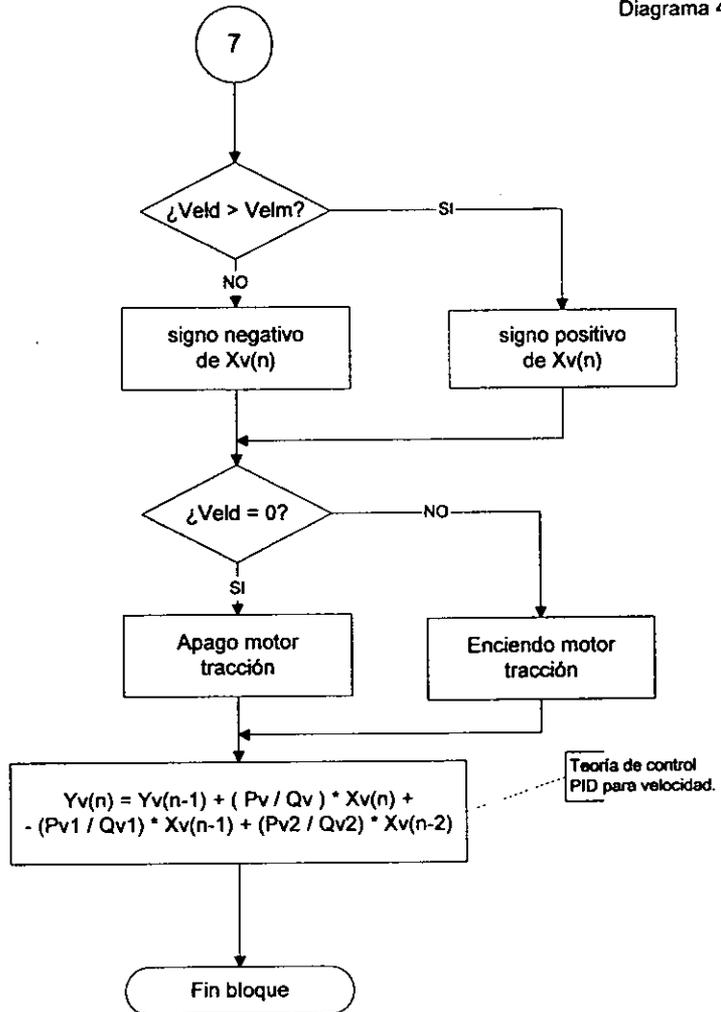
Bloque Lectura de datos.
Diagrama 3B



Bloque Etapa de control:
velocidad y posición.
Diagrama 4A

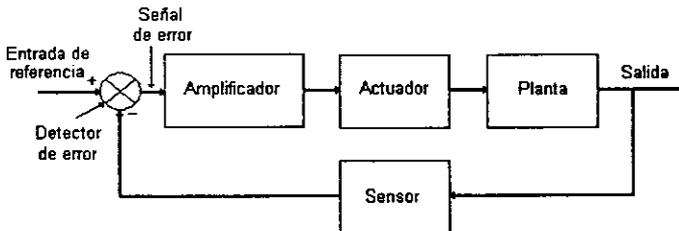


Bloque Etapa de control:
velocidad y posición.
Diagrama 4B



4 CONTROL

4.1 ESTRUCTURAS DE CONTROLADORES



En la figura apreciamos los componentes principales de una malla de control. La planta, es el objeto o variable a controlar. Un actuador que es el elemento físico que lleva al cabo el proceso de control. El sensor de retroalimentación mide la variable a controlar¹⁵.

Controlador p

La retroalimentación negativa es un concepto fundamental para el control, es el primer paso para disminuir los efectos de perturbaciones y ruidos y para alcanzar la estabilidad. Supongamos un sistema en malla abierta, con una ganancia K y una planta $AG(s)$. La ganancia K se elige de tal manera que la salida "y" sea igual a la referencia "r" cuando la perturbación "w" es 0. Este valor de K es $1/A$.

Y en estado estable sin perturbación w

$$y = Awa = A(r/A) = r.$$

Con el controlador de retroalimentación la salida del amplificador es:

¹⁵También nos da la constante de retroalimentación.

$$va = K(r - y)$$

Suponiendo una planta que presenta sólo polos, el sistema de malla cerrada se representa como:

$$[P(s) + AK]Y(s) = AKR(s) + BW(s)$$

Donde $P(s)$ es un polinomio en s . $BW(s)$ es el término debido a la distorsión. Si no existen perturbaciones y estamos en estado estable de tal manera que $dy/dt = 0$, entonces la salida es:

$$Y = [AK/(1+AK)]r$$

Se aprecia que si se escoge la ganancia K de tal manera que $AK \gg 1$ entonces, $y \cong r$.

Consideremos ahora, contrario a los casos anteriores, que existen perturbaciones.

En el caso de malla abierta, la salida y en estado estable es:

$$Y = AKr + Bw$$

Aquí con una ganancia $K = 1/A$ la salida es

$$Y = r + Bw$$

En donde Bw es el término debido a la distorsión.

En el caso de la retroalimentación la función de transferencia es:

$$Y_{ss} = [AK/(1+AK)]r + [B/(1+AK)]w.$$

Si tenemos que $AK \gg 1$ y $AK \gg B$ entonces no se tendrá un error apreciable en estado permanente.

Con ello vemos que la retroalimentación proporcional reduce los efectos de ruidos y perturbaciones. La acción general del control proporcional es $u = Ke$, o en el dominio de Laplace, $D(s) = K$.

Para sistemas de orden mayor, valores grandes de K seguramente nos llevarán a la inestabilidad. Para alcanzar un error mínimo en estado estable sin aumentar la ganancia K se introduce un controlador integral.

Control PI

El control integral actúa sobre todos los valores pasados del error "e". Así, la salida del controlador u puede tener un valor finito aunque el error alcance un valor de cero. Esto es

$$u(t) = \frac{K}{T_I} \int_0^t e d\eta$$

O en el dominio de Laplace

$$\frac{U(s)}{E(s)} = D(s) = \frac{K}{T_I s}$$

Donde T_I es el tiempo de la integral, y $1/T_I$ es una medida de la velocidad de respuesta. Considerando el modelo de un sistema

$$\tau_1 \tau_2 \ddot{y} + (\tau_1 + \tau_2) \dot{y} + y = A \left[\frac{K}{T_I} \int_0^t (r - y) d\eta \right] + Bw$$

Diferenciando para remover la integral y considerando el estado estable

$$A(K/T_I)y = A(K/T_I)r$$

O

$$Y = r$$

Así pues, si el sistema es estable, la salida en estado estable es igual a la referencia sin importar el valor de K . Ya

que estamos introduciendo un integrador en la forma $1/s$, también colocamos un polo de lazo abierto en el origen y por lo tanto la respuesta dinámica se vuelve más oscilatoria. La forma de un controlador PI es como sigue:

$$u(t) = K \left[r - y + \frac{1}{T_i} \int_0^t (r - y) d\eta \right]$$

Control PD

Este tipo de controlador se usa, en conjunto con los controladores anteriores, para incrementar el amortiguamiento y también mejorar la estabilidad del sistema. El controlador responde a la velocidad del cambio del error y por lo tanto no se utiliza solo, ya que un error constante no produciría una acción de control. La acción de control PD se adelanta a la acción del control P en T_d segundos iniciando una acción correctiva temprana. El control derivativo tiene la forma

$$u(t) = KT_d e(t)$$

O en Laplace la función de transferencia es

$$D(s) = KT_d s$$

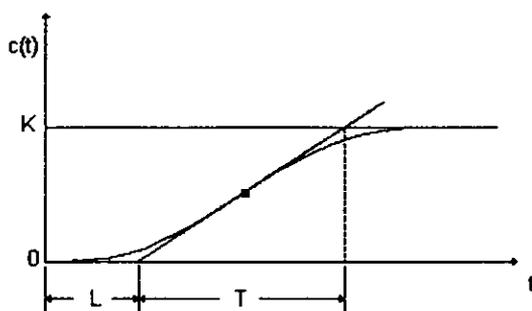
Control PID

Para el control de los errores en estado estable y trascendentes se combinan los tres tipos de control. La señal de control es una combinación lineal del error, su integral y su derivada. La forma del controlador PID está dada por

$$D(s) = K \left(1 + \frac{1}{T_i s} + T_d s \right)$$

4.2 SINTONIZACIÓN

J.G. Ziegler y N.B. Nichols encontraron que muchos sistemas tenían una curva de respuesta como la de esta figura



Esta curva, que es característica de muchos sistemas de orden mayor, puede ser aproximada por un sistema de primer orden y un retraso de tiempo t_d como

$$\frac{Y(s)}{U(s)} = \frac{Ke^{-t_d s}}{s+1}$$

Las constantes pueden ser determinadas a partir de la respuesta al escalón. Si es dibujada una tangente en el punto de inflexión y se prolonga hasta el eje del tiempo, se tienen la pendiente R y el tiempo de retraso L . Donde $R = K/t$ y $L = t_d$.

Ziegler y Nichols propusieron dos métodos de sintonización. El primero está basado en la reducción de la respuesta trascendente en un 25%, el cual decae en un periodo de oscilación. Esto corresponde a una $\xi = 0.21$ lo cual significa una buena velocidad de respuesta y un buen margen de estabilidad.

Los parámetros para este controlador se dan en la siguiente tabla.

TABLA 1

| Tipo de controlador | Ganancia optima | | |
|---------------------|-----------------|-------|------|
| | Kp | Ti | Td |
| P | 1/RL | | |
| PI | 0.9/RL | L/0.3 | |
| PID | 1.2/RL | 2L | 0.5L |

Si la curva de respuesta no presenta la forma anterior, el método no se puede aplicar.

En el segundo método el criterio para ajustar los parámetros se basa en la evaluación del sistema en el límite de estabilidad en vez de la respuesta a una entrada escalón. Se incrementa la ganancia proporcional hasta que se observan oscilaciones sostenidas, o sea, hasta que es marginalmente estable.

La ganancia crítica "Kc" (con la cual oscila) y el período de oscilación "Pc" determinan entonces la ganancia del sistema conforme a la tabla que se muestra.

TABLA 2

| Tipo de controlador | Ganancia optima | | |
|---------------------|-----------------|--------|---------|
| | Kp | Ti | Td |
| P | 0.5Kc | | |
| PI | 0.45Kc | Pc/1.2 | |
| PID | 0.6Kc | 0.5Pc | 0.125Pc |

4.3 DIGITALIZACIÓN DEL CONTROLADOR PID

Si se tiene en cuenta que el controlador analógico se puede modelar como:

$$u(t) = K \left[e(t) + \frac{1}{T_i} \int_0^t e(t) dt + T_d \frac{de(t)}{dt} \right]$$

Entonces, para obtener la función de transferencia del mismo controlador pero de manera digital se tiene que, al aproximar el término integral de manera trapezoidal y el término derivativo mediante la diferencia de dos puntos, se tiene que

$$u(kT) = K \left\{ e(kT) + \frac{T}{T_i} \sum_{h=1}^k \frac{e((h-1)T) + e(hT)}{2} + \frac{T_d}{T} [e(kT) - e((k-1)T)] \right\}$$

Lo que nos da la función de transferencia del PID digital como:

$$G_D(z) = K_p + \frac{K_I}{1-z^{-1}} + K_D(1-z^{-1})$$

Con las constantes digitales como:

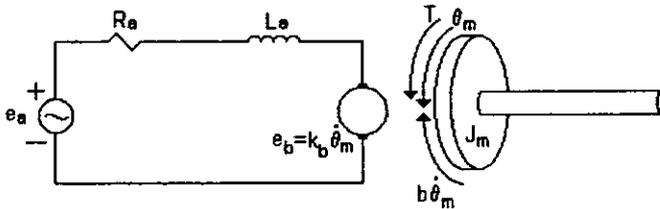
$$K_p = K - \frac{KT}{T_i} = \text{ganancia proporcional}$$

$$K_I = \frac{KT}{T_i} = \text{ganancia integral}$$

$$K_D = \frac{KTD}{T} = \text{ganancia derivativa}$$

4.4 CONTROL DE UN MOTOR DE CD

Un motor de corriente directa se puede modelar, en su parte eléctrica, como una resistencia en serie con una inductancia y una fuerza contra-electromotriz, todo esto acoplado, también de manera serial, a una fuente. Al modelar la parte mecánica se consideran la fricción y la inercia del motor y su carga que se oponen al torque generado por la corriente en la armadura. Así, a partir de la figura, se tiene que:



$$L_a \frac{di_a}{dt} + R_a i_a + e_b = e_a \qquad J \frac{d^2\theta}{dt^2} + b \frac{d\theta}{dt} = T = K i_a$$

Y tomando su transformada de Laplace:

$$(L_a s + R_a) I_a(s) + E_b(s) = E_a(s)$$

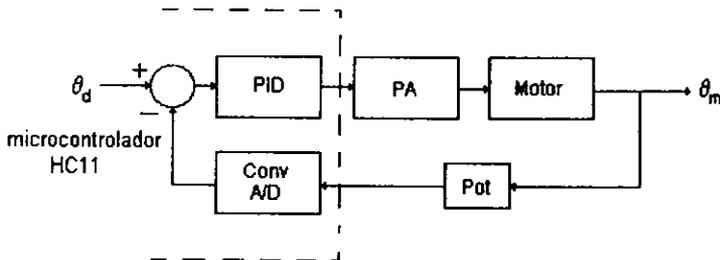
$$K_b s \Theta(s) = E_b(s)$$

Desarrollando estas ecuaciones se tiene que:

$$(J s^2 + b s) \Theta(s) = T(s) = K I_a(s)$$

$$\frac{\Theta(s)}{E_a(s)} = \frac{K}{s [L_a J s^2 + (L_a b + R_a J) s + R_a b + K K_b]}$$

Que es la ecuación de transferencia para posición, un diagrama para el sistema de control de dirección del robot móvil es el siguiente:

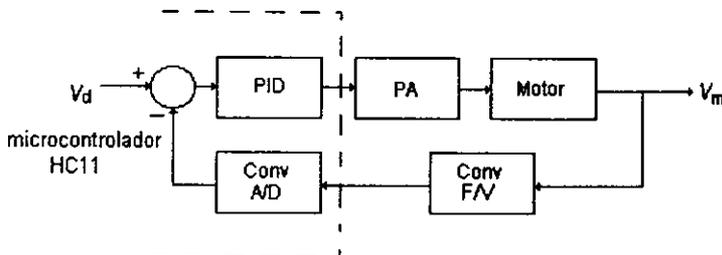


Para el caso de velocidad basta con renombrar a $d\theta/dt$ como velocidad en la siguiente ecuación y el procedimiento es el mismo.

$$J \frac{d^2\theta}{dt^2} + b \frac{d\theta}{dt} = T = K i_a$$

O en su defecto derivar la función de transferencia de velocidad, con lo que tendría una "s" en el denominador y sería velocidad respecto a voltaje.

El diagrama utilizado para nuestro caso con el microcontrolador es



Como el motor de corriente directa es muy pequeño despreciamos el valor de la inductancia, dándonos la siguiente función de transferencia para el caso de posición

$$\frac{\Theta(S)}{E_a(S)} = \frac{K/RJ}{S[S+(b+KKb)/RJ]}$$

y para el caso de velocidad:

$$\frac{V(S)}{E_a(S)} = \frac{K/RJ}{S+(b+KKb)/RJ}$$

Ya con esta ecuación y con los parámetros del motor bastaría añadir el controlador de la forma

$$\frac{K'}{[S^2 + 2\xi\omega_n S + \omega_n^2]}$$

para tener el control de velocidad requerido. Sin embargo como se carece de los parámetros del sistema no se procedió por este método, sino por el experimental.

5 EXPERIMENTOS y PROBLEMAS

5.1 EXPERIMENTACIÓN

Etapa de potencia y motores

Para la etapa de potencia de los motores se hicieron ciertas pruebas con el fin de poder limitar la corriente máxima entregada a los motores, en especial al de tracción en caso de bloqueo de la flecha. Con estas pruebas se llegaron a ciertos valores de resistencia para proporcionar la corriente requerida, en el driver de tracción. Estos valores fueron modificados a partir de los teóricos.

Se hizo la prueba de rotor bloqueado para el caso de tracción y se observó que la demanda máxima de corriente es de 650mA, lo cual nos permite tener una mayor confianza para el caso de que el RM quedara atascado, y no tener la lectura de los sensores de golpe.

Se aplicaron diferentes valores del PWM (potencia) al motor de tracción y se midió la velocidad, con ello se observó el incremento de velocidad relacionado con el incremento de potencia, y se observa que si se incrementa como es debido.

Sensor de velocidad

Se procedió al ajuste de la constante de retroalimentación de la malla de control de velocidad, esto se logró por medio de un potenciómetro en el circuito conversor de frecuencia a voltaje. Se comenzó dando un valor conocido en PWM al motor y a la salida se deseaba tener el mismo valor (un dato equivalente), esto se logró moviendo el cursor del potenciómetro hasta tener la salida requerida.

5.1.2 Limitaciones en trayectorias

Ángulo

Como ya se mencionó el RM no puede realizar radios de curvatura ideales, mecánicamente tenemos que el mínimo radio de curvatura es de 45 cm.

El RM lo más que puede torcer la dirección de las llantas delanteras respecto al reposo son 28 grados, dándonos el radio de curvatura mencionado.

Velocidad

Para las limitaciones de velocidad se tiene una velocidad mínima de 32 cm/s, esta velocidad se debe a la inercia y fricciones mecánicas internas del robot que no le permiten ir a una velocidad menor. El límite de velocidad superior está fijado por las condiciones electromecánicas del RM que es de 100cm/s.

Se observó que el motor de tracción tiene un valor mínimo de DC (o un equivalente en PWM) para que el motor arranque, siendo que a valores mas bajos el motor no logra vencer el par generado por la fricción y la inercia propia del mismo.

Por medio del control se permite tener valores menores de velocidad en ciertos instantes, pero no responde como muy establemente, sino que permanece oscilando alrededor del nivel pedido.

Distancia

Para medir la distancia por software, se puede decir que no tenemos grandes problemas, esto se debe a que el robot puede avanzar desde 1 cm hasta 65000 cm. Como la resolución de paso es de

1cm por bit y para la distancia contamos con un registro de 16 bits se podrían recorrer hasta los 16 bits en cm (65530cm aproximadamente), la limitante es en este caso por batería.

Lógico (programación)

Es necesario volver a iniciar los datos de las condiciones iniciales para el control de velocidad ante cada nueva trayectoria simple, esto es debido porque aunque la salida en velocidad sea cero el control de velocidad sigue funcionando, por lo tanto también sus registros asociados, y al entrar a la nueva trayectoria puede causar problemas. Al ser velocidad cero sólo se apagan los motores y sigue funcionando toda la parte asociada al control para conservar e mismo periodo de muestreo sin importar la tarea solicitada.

Ultrasónicos

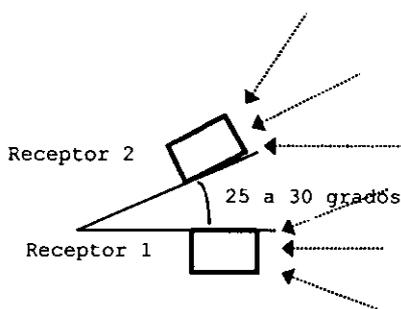
Para la parte de los ultrasónicos se realizaron gran cantidad de pruebas, ya que con esta etapa se permitiría tener el dato de posición real del RM.

Una de las pruebas está relacionada a la parte electrónica de los circuitos asociados a los sensores receptores, a los transmisores y la última relacionada con los sensores ultrasónicos. En la parte del transmisor se hicieron pruebas para ver que el elemento escogido, un cono, radiara los 360 grados, fue necesario colocar 6 transmisores de ultrasonido para lograr radiar en los 360 grados del plano horizontal. Con un numero menos de transmisores no se logra radiar en todo el plano con la potencia requerida. La ventaja del elemento escogido, un cono de papel cartón fue su facilidad para moldearlo y ser ligero.

En la parte electrónica se probaron las etapas de amplificación y el filtrado del receptor. Se comprobó que todos estos circuitos funcionaran, es decir, que lograban leer y detectar las ráfagas desde distancias de unos cuantos centímetros hasta un máximo de 4 metros de separación en línea recta. Lo que importa de la ráfaga es el primer flanco de llegada, ya que con este es con el que se realiza el cálculo.

Los sensores tanto transmisores, como receptores tienen un campo aproximado de acción de 65 grados.

Inicialmente se colocó un sólo receptor (sensor) ultrasónico en cada uno de los puestos de recepción, lamentablemente por el campo de acción de los receptores se tuvieron que colocar dos en paralelo para poder ver toda la región de trabajo correctamente, los receptores se colocaron en paralelo, ambos mirando al área de trabajo con una separación de 25 grados en su área de visión. Para cada receptor se armó un circuito electrónico independiente.



Receptores en paralelo

Se tuvo el problema de que ambos receptores en el mismo puesto de recepción no respondían al mismo tiempo, es decir su tiempo de encendido ante la llegada de una ráfaga no es el mismo, por ello es que el sistema no respondió.

Algunos de los problemas que se presentaron con los ultrasónicos es que al hacer funcionar el sistema en un cuarto pequeño y cerrado existe gran cantidad de rebotes entre las paredes y demás objetos, lo que origina un menor tiempo de muestreo y además la posibilidad de tener lecturas erróneas.

Ráfagas

La diferencia de tiempo entre ráfaga y ráfaga teóricamente esta limitada por la separación máxima entre receptor y transmisor, 4m y tomando la velocidad del sonido, tenemos un tiempo de retardo de llegada de 11.1ms, y se añade un tiempo de espera entre ráfagas de 9.9ms, dándonos 20ms. Sin embargo por todos los rebotes en el laboratorio y el tiempo de apagado de los sensores receptores se tuvo el tiempo entre ráfagas en 100ms. El tiempo de apagado es muy grande, dándonos este retardo.

5.2 MECÁNICA

Problemas

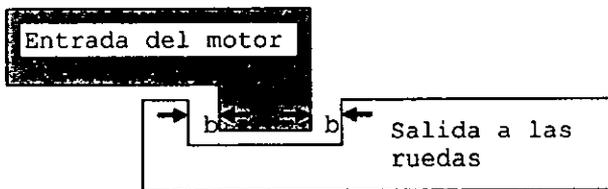
Dirección

Al momento de comenzar las pruebas con la parte de la dirección se observó que entre la flecha del motor de dirección y las llantas de la dirección existía un juego (zona muerta), es decir el eje de la flecha de motor gira (de forma considerable) y las ruedas de dirección no sufren un cambio de posición, por ello es que se procedió a resanar o rellenar ciertos espacios entre los componentes de la transmisión del giro para reducir este juego en la medida de lo posible, evitar ese retardo indeseable al momento de introducir la etapa de control. Ya que el controlador daba un valor de salida al motor para realizar el giro y por la misma zona

muerta esto se realiza un diferencia de tiempo después, que altera las condiciones el proceso.

Aún después de los resanes entre las partes más visibles queda un juego considerable, debido a que mecánicamente no es reparable el daño existente, sino que se tendría que reelaborar el sistema de dirección para evitarlo.

Otro de los problemas que se tiene es que debido a la fricción de las llantas de dirección se necesita vencer un par de arranque para poder realizar un giro, lo que implica que el motor no se puede mover tan suavemente como deseamos, sino que arranca a una velocidad mínima, algo parecido al de tracción y no permite ciertos pasos mínimos de movimiento. En otras palabras. solo hasta después de cierto voltaje de umbral es cuando la dirección gira. Esto se mejoraría con un motor de pasos en vez de uno de DC, y un mejor acoplamiento entre la flecha de motor y el eje de las llantas de dirección.



Zona muerta

La figura es una representación de un modelo físico del juego o zona muerta, entre un miembro de entrada y otro de salida. El juego representado agrupa la suma de todos los juegos entre los diferentes elemento mecánicos. "2b" es la distancia total del juego. Como se aprecia los dos miembros viajarán juntos hasta que el miembro de entrada invierta la dirección, entonces el miembro de salida permanecerá quieto hasta que el juego se mueva hasta el otro

lado. Se supone que el miembro de salida toma instantáneamente la velocidad del miembro de entrada.

5.3 PARÁMETROS DEL PID, GRÁFICAS.

Ante la carencia del modelo real, se tuvo que aproximar el modelo de ambos sistemas a uno de segundo orden, representado por un motor de DC. Para poder controlar dichos sistemas se procedió a experimentar con los valores de los parámetros de las constantes del modelo para un PID.

La aproximación a los valores de los parámetros del PID se logró por métodos experimentales.

Otro de los valores que estuvieron cambiando en todo este tiempo fue el valor del periodo de muestreo, el cual fue varando en razón a la programación y nuestros cambios a los valores de las constantes de la ley de control. El periodo influye ya que está relacionado directamente con la ley de control digital.

Se realizaron varias pruebas, la primera fue empíricamente ir recorriendo los valores de los parámetros del PID hasta tener una buena respuesta.

Lo realizado en cada etapa de experimentación era ante cada juego de valores de los parámetros del PID realizar un recorrido para ver en que casos el error se reducía y en que casos el error aumentaba o más aún el sistema se hacía inestable.

Gráficas

Se tienen dos tipos de gráficas o respuestas, la de velocidad y la de dirección.

Gráfica 1

En ella tenemos la respuesta de velocidad del motor de tracción, a velocidad de 70 cm/s (escala 0.2V/div, 1s/div), es la gráfica se aprecia un pequeño sobrepaso aproximado del 5%. Se ve mucho ruido debido al tipo de modulación y la corriente requerida por los motores.

Gráfica 2

Se tiene la velocidad de respuesta del motor a baja velocidad, 35cm/s (0.2V/div, 1s/div), por causa de la inercia y fricción, además de la baja velocidad es que se tarda en asentar el sistema, y es el sobrepaso del 27% aproximadamente.

Gráfica 3

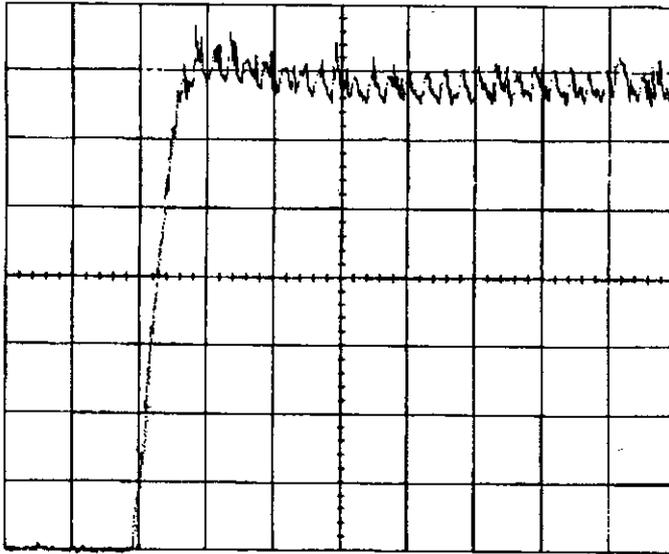
Se tiene la respuesta del sistema de dirección, se parte de la dirección totalmente a la derecha a estabilizar en el centro (posición de reposo, escala 1V/div, 2s/div), como se aprecia el sistema es lento, debido a los espacios muertos y a la no linealidad del sistema de dirección, sin embargo si se estabiliza, pero son un error de estado estable del orden del 2%, este error se debe a la alta fricción estática del sistema de dirección.

Gráfica 4

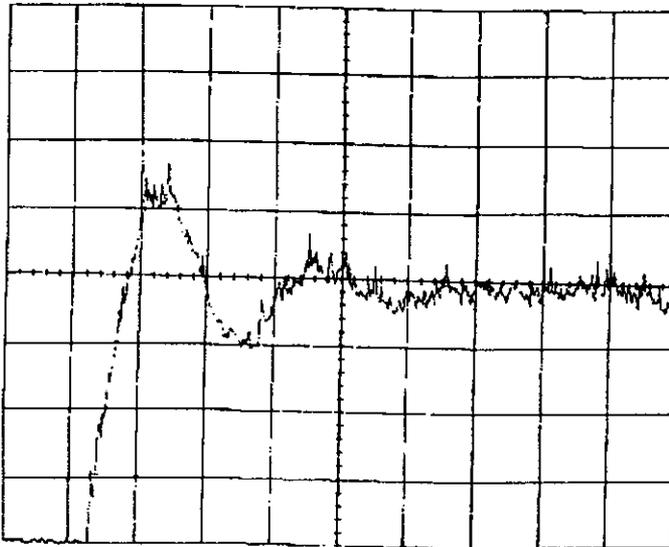
Se tiene la respuesta del sistema de dirección, se parte de la dirección totalmente a la izquierda a estabilizar en el centro (escala 1V/div, 2s/div), se puede ver que el sistema no responde de manera similar, ya que aveces es sistema es más rápido y aveces es más lento.

Gráfica 5

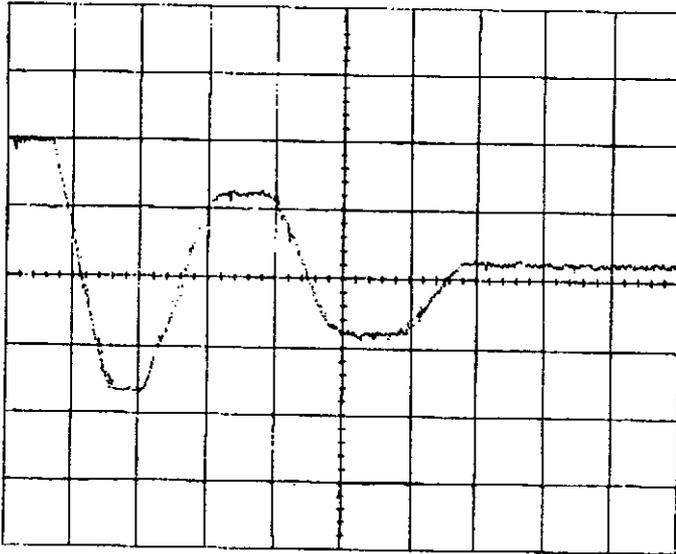
Se tiene la respuesta del sistema de dirección, se parte del centro a un giro pedido de radio de curvatura de 65cm, como se aprecia el sistema es lento pero es la mejor respuesta obtenida para el sistema no lineal (1V/div, 2s/div).



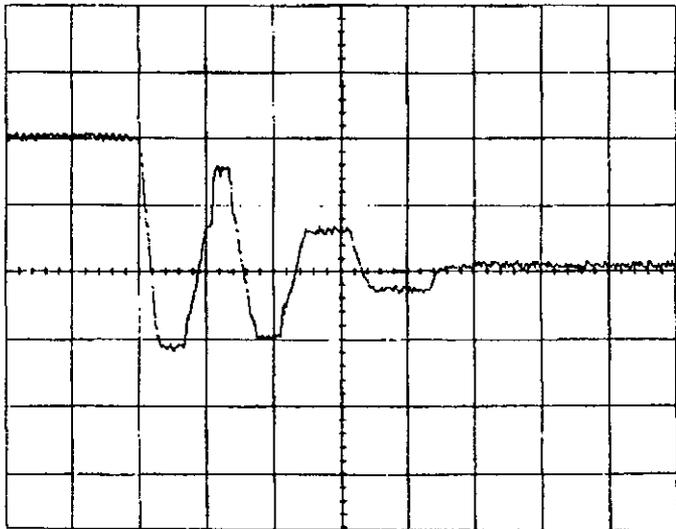
Gráfica 1



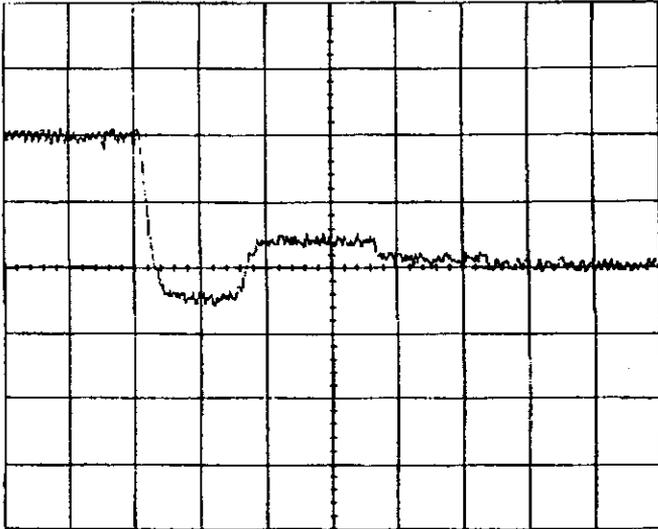
Gráfica 2



Gráfica 3



Gráfica 4



Gráfica 5

Experimentación trayectorias

Para el caso general de las trayectorias se realizaron pruebas en el laboratorio, trazando ciertos recorridos dentro del cuarto. Con ello se observaba que tanto error producía respecto a lo que queríamos lograr. Esta experimentación está asociada a la parte de la obtención de los parámetros del PID.

Como la dirección tiene problemas de estabilidad por la no linealidad existente debido al juego resulta que a veces si se puede controlar mejor la trayectoria y a veces se sale de control y no se logra estabilizar el control de la dirección.

Con el RM sobre nuestra mesa de trabajo se realizaron gran cantidad de pruebas, en ellas se observa el comportamiento del RM en velocidad y el retardo que se tiene en el motor de dirección, aquí es donde se ajustaron los parámetros, del PID, luego con carga se les dio el ajuste final.

6 CONCLUSIONES

6.1 GENERALES

Desde el punto de vista electrónico es un buen prototipo de un robot móvil, capaz de realizar tareas básicas, y que en general es un buen desarrollo, además que el costo del mismo es mínimo respecto a cualquiera que se pueda conseguir en el mercado.

Se cumplió con los objetivos propuestos, desde la construcción tanto mecánica como electrónica, hasta la etapa de control para la realización de las trayectorias, con la implementación de sensores que la experimentación permitió.

Nos atrevemos a llamarle robot móvil (RM), ya que es un mecanismo que realiza acciones por sí mismo, una vez dadas las instrucciones, en vez de pedir al usuario las decisiones.

En este trabajo se aplicaron gran cantidad de conocimientos adquiridos a lo largo de la carrera, ya sea para la investigación del tema y para el desarrollo del mismo, lo cual nos amplió visiblemente el panorama que teníamos de la carrera, además, este trabajo nos aportó entre otras cosas la satisfacción de haber construido un modelo que puede funcionar, el saber que aplicando todos los conocimientos que poseemos, desarrollamos un buen proyecto.

6.2 PARTICULARES

Se procedió a modificar un modelo mecánico, ya que crear uno nuevo partiendo desde cero requeriría mucho tiempo.

El control de velocidad resulta ser bastante efectivo y lo realiza tal como lo muestran las gráficas, sin embargo por condiciones mecánicas no puede ir a muy bajas velocidades, ya que el coeficiente de fricción estática es alto. Partiendo del reposo el control de velocidad va incrementando poco a poco el voltaje del motor hasta vencer el voltaje de umbral, para arrancar, partiendo a una velocidad alta; ya con el coeficiente de fricción dinámico es posible ir a velocidades más bajas o caer dentro de un rango de oscilaciones alrededor del valor deseado. Esto mismo sucede en el caso del control de dirección.

Para el caso del control de velocidad es recomendable un pequeño sobrepaso en la respuesta, principalmente porque la mayoría de las pruebas se hicieron en vacío, para que el tiempo de levantamiento sea menor.

Para el caso de las trayectorias de figuras regulares, es necesario hacer notar que entre más grande sea el lado de la figura, menor será la diferencia del vértice (como arco) respecto al lado.

Tomando la respuesta de dirección se aprecia que el sistema no siempre responde de manera similar debido a la no linealidad, por tanto a veces si se puede estabilizar en un tiempo de 10 a 12 segundo y posteriormente más o menos mantenerse estable y en otros el tiempo de estabilización es muy grande. Además al igual que el de velocidad se cuenta con una fricción estática muy alta y tiene que vencer primero ese par para poder girar, esto no permite hacer deslizamientos suaves de la dirección

6.3 PROBLEMAS

El sistemas de dirección, al tener los espacios muertos entre la flecha del motor y las llantas, nos da como resultado un sistema no lineal, el cual estamos aproximando a uno lineal, de segundo orden. Por ello es que se tiene tantos problemas para poder controlar la dirección, y por lo mismo se requiere un tiempo de estabilización para la misma. En general podemos decir que responde de manera lenta el control PID pero que sí logra realizar el control. La ventaja que nos dio el PID es que no necesitamos conocer los parámetros del modelo para poder controlarlo, que es nuestro caso.

El sistema mecánico, nos ocasionó grandes problemas en general, ya que en un principio hicimos muchas consideraciones que en realidad no fueron ciertas, tales como suponer que el modelo mecánico sería más lineal, y que no tendría tantas fallas, como por ejemplo, las dos ruedas delanteras no están paralelas del todo, están un poco abiertas en la parte frontal.

Sistema de ultrasonido

Originalmente se tenía planeado un sistema de triangulación por medio de ultrasonido (e infrarrojo), con ello se lograría tener la posición real del RM en un plano cartesiano, se podría comparar la posición real contra una posición deseada y realizar el control. Como no se consiguió hacer funcionar este sistema se tiene un error de posición que se va incrementando conforme se va realizando el recorrido.

El sistema de ultrasonido no funcionó, la principal razón fueron los sensores de ultrasonido que utilizamos. Estos sensores presentaron problemas al momento de trabajarlos en grupo, ya que no siempre respondían de la misma manera, es decir, el tiempo de

encendido y apagado no era el mismo para todos, unos se encendían antes que otros ante las mismas condiciones de trabajo. Tanto en transmisores como receptores, siendo más notorio en los receptores.

Sin embargo con la utilización de sensores ultrasónicos de mayor potencia, sensibilidad y más finos, o sea, mayor precisión en la respuesta de encendido, se podría hacer funcionar este sistema como estaba planteado originalmente. No se realizó así por el costo de los mismos.

6.4 VENTAJAS Y DESVENTAJAS

Ventajas:

- La parte electrónica del prototipo se puede montar sobre otro elemento mecánico con un número mínimo de modificaciones para el acoplo.
- Con el mismo hardware (electrónico) se pueden montar más sensores en las líneas disponibles de entrada salida, sólo es necesario modificar el software.
- Es tener nuestro propio robot construido a bajo costo.
- Con 8 bits la realización de una etapa de control básica.

Desventajas:

- Un control de dirección muy lento debido a las fallas mecánicas, espacios muertos.
- Una de las desventajas que tiene el modelo, en sí es la parte mecánica, ya que la misma posee una alta inercia y fricción entre sus componentes, lo que ocasiona que al parar los motores el RM se desplace un incremento no contemplado (en software), además la fricción en el sistema de dirección no permite tener desplazamientos pequeños, y afecta el control de velocidad no permitiendo bajas velocidades.

- El haber modificado un modelo (carrito) nos trajo problemas, no por el hecho de haberlo modificado, sino por el hecho de que no estaba dedicado a lo que quisimos aplicarlo. En general el modelo mecánico no resultó como nosotros esperábamos.
- Presenta problemas por el alto consumo de corriente del motor de dirección, un motor de pasos reduciría este problema.

6.5 MEJORAS O PROPUESTAS

El Robot Móvil es susceptible a una enorme cantidad de mejoras, algunas de ellas se trataron de llevar a cabo, pero no todas funcionaron, o no se lograron por la falta de tiempo. Todas ellas nos ayudarían a tener una mejor respuesta del robot, algunas de las propuestas que se tienen son:

Circuito inalámbrico

Esta es una propuesta que se trató de implementar en un inicio, pero por la falta de tiempo, componentes y alejarnos de los objetivos principales no se logró conseguir. Esto serviría para tener una comunicación entre el microcontrolador y un circuito externo o PC, además de recibir los datos provenientes de los receptores ultrasónicos, de esta manera se tendría una comunicación serial asíncrona por radio.

Modificar el sistema de dirección

En lo referente a la transmisión del giro se recomienda una flecha acoplada de manera directa al eje de las llantas o un mejor acoplo que el que existe actualmente, con ello se eliminarían en gran parte las no linealidades debido a los espacios muertos.

Con los resultados obtenidos del control de posición se recomienda el uso de un motor a pasos, partiendo de que el motor

que rige la dirección no alcanza velocidades altas, y requiere un alto par.

Sensores

Se podrían añadir más sensores tales como infrarrojos para el sentido de objetos a cierta distancia.

Con la parte del desarrollo de los sensores ultrasónicos con unos buenos sensores se podría realizar lo que en un inicio nos proponíamos, la triangulación, y con ello ubicación del modelo en el espacio.

Existen otros métodos posibles para la localización como la triangulación con infrarrojos o con radar.

7 APÉNDICE

7.1 TRIANGULACIÓN

El proceso se puede realizar por medio de dos tipos de procedimientos, el primero es tomar las distancias del objeto en cuestión, a tres puntos de referencia (tres radios) y por medio de trigonometría obtener el resultado del punto de intersección de las tres circunferencias. Este tipo de algoritmo es que utilizado con los sensores ultrasónicos, ya que sólo basta tener el tiempo de propagación entre transmisor y receptor para poder localizar el objeto en estudio.

El segundo proceso es por medio de tres ángulos, los cuales se tienen entre el objeto el estudio y tres puntos de referencia, basta con dos puntos de referencia (dos ángulos) casi para todo caso, excepto para cuando el elemento en estudio está alineado con los dos sensores, es cuando se necesita la tercer referencia. Esto se pretendía realizar utilizando sensores infrarrojos y una fuente de radiación.

Por medio de la triangulación se pretendía tener un control de trayectoria, es decir se podría uno aproximar desde un punto exterior, a la trayectoria deseada, pero por varios inconvenientes esto no se logró; el principal obstáculo fue el que los sensores ultrasónicos no funcionaron como se requería.

Además para poder hacer la aproximación desde un punto fuera de la trayectoria se requería un algoritmo para la aproximación de la misma, el cual no entra en esta tesis, pero se habían planteado algunas soluciones respecto al control y aproximación de trayectoria.

7.2 DEFINICIONES UTILIZADAS

Zona muerta o juego: Se entiende como todos los espacios entre los componentes existentes en el sistema de dirección, tales como engranes y mecanismos. El juego puede llevar a inexactitud no deseada, oscilaciones e inestabilidad en sistemas de control.¹⁶

Ráfagas: Es el conjunto de unos y ceros transmitidos a 40 KHz para poder ser leído el dato por los ultrasónicos receptores. La ráfaga va modulada por una señal cuadrada de periodo 1 a 10 ms.

RM : Robot Móvil

Robot Móvil: Es un sistema electromecánico capaz de realizar una serie de tareas tomando sus propias decisiones.

Sistema de dirección: Está formado por el eje de la flecha del motor, por 4 engranes que reducen la velocidad, y por un par de juegos de barras que transmiten el giro.

7.3 LISTADOS y DIAGRAMA

A continuación se presentan los listados de los programas utilizados y el diagrama general de la tarjeta del microcontrolador HC11 de Motorola.

¹⁶ Kuo, Benjamin. Sistemas de control automático. P. 150

 * PROGRAMA EN LENGUAJE ENSAMBLADOR PARA EL HC11, PARA LA PRESENTACION *
 * DE LA TESIS "DISEÑO, CONSTRUCCION Y CONTROL DE UN ROBOT MOVIL". *
 *

* AUTORES *

* DANIEL FERNANDEZ NUÑEZ *

* SERGIO D. VERA REA *

***** INICIALIZACIONES DE REGISTROS GENERALES

PORTG EQU \$02
 DDRG EQU \$03
 PORTE EQU \$0A
 OC1M EQU \$0C
 OC1D EQU \$0D
 TOC1 EQU \$16
 TOC2 EQU \$18
 TOC3 EQU \$1A
 TCTL1 EQU \$20
 TFLG2 EQU \$25
 PACTL EQU \$26
 PACNT EQU \$27
 BAUD EQU \$2B
 SCCR1 EQU \$2C
 SCCR2 EQU \$2D
 SCSR EQU \$2E
 SCDR EQU \$2F
 ADCTL EQU \$30
 ADR1 EQU \$31
 ADR2 EQU \$32
 OPTION EQU \$39

;REGISTROS EN LOS QUE SE GUARDAN TEMPORALMENTE
 ;LOS DATOS LEIDOS DE LA PC

ANGD EQU \$00
 VELD EQU \$01
 CICL1 EQU \$02
 CICL2 EQU \$03

;REGISTROS DE TRABAJO PARA EL CONTROL DE
 ;VELOCIDAD

ORG \$2500
 VELOC RMB 1 ;VELOCIDAD DESEADA
 YVEL RMB 2 ;SALIDA DEL CONTROL PID
 YTACO RMB 1 ;RETROALIMENTACION DEL TACOMETRO
 XVEL RMB 1 ;1er ERROR
 XVEL1 RMB 1 ;2do "
 XVEL2 RMB 1 ;3er "
 PVEL RMB 1 ;1er NUMERADOR
 PVEL1 RMB 1 ;2do "
 PVEL2 RMB 1 ;3er "
 QVEL RMB 2 ;1er DENOMINADOR
 QVEL1 RMB 2 ;2do "
 QVEL2 RMB 2 ;3er "

```
SVEL RMB 1 ;SIGNO DE LOS ERRORES DE VELOCIDAD
;REGISTROS DE TRABAJO PARA EL CONTROL DE
;POSICION
```

```
ORG $2520
```

```
DIREC RMB 1 ;ANGULO DESEADO
ANGM RMB 1 ;ANGULO MEDIDO
YPOS RMB 2 ;Salida del PID Y(n)
XPOS RMB 1 ;Error de posición
XPOS1 RMB 1 ;Primer corrimiento del error
XPOS2 RMB 1 ;Segundo corrimiento del error
PPOS RMB 1 ;1er numerador
PPOS1 RMB 1 ;2do "
PPOS2 RMB 1 ;3r "
QPOS RMB 2 ;1er denominador
QPOS1 RMB 2 ;2do "
QPOS2 RMB 2 ;3r "
SPOS RMB 1 ;Registro de signos
```

```
;REGISTROS AUXILIARES
```

```
TEMP RMB 2 ;TEMPORAL
SYPOS RMB 1 ;SIGNO DE LA SALIDA DEL PID DE POSICION
CONT1 RMB 1 ;CONTADOR 1
CONT4 RMB 2 ;CONTADOR 4
RAUX RMB 2 ;REG AUXILIAR
```

```
*****
*****
```

```
;PROGRAMA PRINCIPAL
```

```
ORG $2000
```

```
;INICIALIZACION DE REGISTROS DE CONTROL
;DEL HC11
```

```
INI LDX #$1000
LDAB #$FC
STAB DDRG,X
CLR PORTG,X
BSET PORTG,X,$20
LDAA #$90
STAA OPTION,X
LDAA #$30
STAA ADCTL,X
LDAB #$60
STAB OC1D,X
BCLR OC1M,X,$60
BSET PACTL,X,$40

LDAA #$A0
STAA TCTL1,X
LDD #$0000
STD TOC1,X
LDD #$0001
STD TOC2,X
```

```

STD     TOC3,X
JSR     DATPOS           ;SUBRUTINAS DE INICIALIZACION
JSR     DATVEL          ;DE DATOS DE POSICION Y VELOCIDAD
JSR     DATCONTR        ;INICIALIZO REGISTROS DE CONTROL

AQ1  BRCLR PORTE,X,$80,AQ1 ;ESPERO EL PUSH BUTTON

      LDY #3000
      LDAA #$30
      STAA BAUD,X
      LDAB #$04           ;HABILITAMOS
      STAB SCCR2,X       ;RECEPCION DE DATOS

*****RECEPCION DE DATOS
AQ2  BRCLR SCSR,X,$20,AQ2 ; DATO DIRECCION
      LDAA SCDR,X
      STAA ANGD,Y
      BITA #01
      BEQ  AQ3
      BRA  ONPWM

AQ3  BRCLR SCSR,X,$20,AQ3 ; DATO VELOCIDAD
      LDAA SCDR,X
      STAA VELD,Y
      BITA #01
      BEQ  AQ4
      BRA  ONPWM

AQ4  BRCLR SCSR,X,$20,AQ4 ; PARTE ALTA DE CICLOSS
      LDAA SCDR,X
      STAA CICL1,Y

AQ5  LSR  CICL1,Y           ;PARTE BAJA DE CICLOS
      BRCLR SCSR,X,$20,AQ5
      LDAA SCDR,X
      STAA CICL2,Y

AQ6  INY
      INY
      INY
      INY
      BRA  AQ2

                                           ;ENCENDEMOS Y HABILITAMOS MOTORES
ONPWM NOP
AQ7  BRCLR PORTE,X,$80,AQ7
      BSET OC1M,X,$60
      LDY #3000
      PSHY

      LDAA ADR2,X           ;OBTENCION DEL PRIMER SIGNO
      STAA ANGM
      LDAA ANGD,Y
      CMPA ANGM

```

BHI LP1
LDAB #S01
STAB SYPOS

*****LECTURA DE DATOS DE LA MEMORIA

LP1 PULY
LDAA ANGD,Y ; CARGAMOS DIRECCION
STAA DIREC
BITA #S01
BEQ LP11 ; VERIFICAMOS SI ES EL ULTIMO DATO
JMP INI

LP11 LDAA VELD,Y ; CARGAMOS VELOCIDAD
STAA VELOC
BITA #S01
BEQ LP12
JMP INI

LP12 LDAB CICL1,Y ; CARGAMOS CICLOS PARTE ALTA
STAB CONT4

LP13 LDAB CICL2,Y ; CARGAMOS CICLOS PARTE BAJA
STAB CONT4+1

LP14 INY ; INCREMENTAMOS CUATRO VECES
INY
INY
INY
PSHY
JSR DATVEL ; INICIAMOS DATOS DE VELOCIDAD

***** INICIO DE TRAMOS

LP LDAA VELOC
CMPA #S00
BNE LP2

LP3 LDAB CONT4+1
STAB CONT1

LP32 LDY #S000A

LP31 JSR RTIS
JSR DLY2
DEY
BNE LP31
DEC CONT1
BNE LP32
JMP LP1 ; ***PARTE DEL RECORRIDO

LP2 CLR PACNT,X ; LIMPIAMOS CONTADOR DE PULSOS
LDY CONT4

LP21 JSR RTIS ; CONTROL
JSR DLY2
BRCLR TFLG2,X,\$20,LP22
INC RAUX ; MEDICION DE LA DISTANCIA
LDAB #S20

```

      STAB  TFLG2,X
LP22 LDAA  PACNT,X
      STAA  RAUX+1
      CPY   RAUX
      BHS   LP21
      JMP   LP1          VA A CARGAR OTROS DATOS

```

*****RUTINA DE PERIODO DE MUESTREO

```

DLY2 PSHY
      LDY   #$0600
      NOP
L3    DEY
      BNE  L3
      PULY
      RTS

```

*****RUTINA DE INICIALIZACION DE DATOS
;DE POSICION

```

DATPOS LDAA #$00
      STAA XPOS
      STAA XPOS1
      STAA XPOS2
      STAA ANGM
      STAA YPOS
      STAA YPOS+1
      STAA SPOS
      STAA SYPOS
      RTS

```

*****RUTINA DE INICIALIZACION DE DATOS
;DE VELOCIDAD

```

DATVEL LDAB #$00
      STAB XVEL
      STAB XVEL1
      STAB XVEL2
      STAB YTACO      ;
      STAB YVEL      ; PRIMER DATO = 0
      STAB YVEL+1
      STAB SVEL      ; SIGNOS = 0
      RTS

```

*****RUTINA INICIALIZACION DE DATOS DE CONTROL
;REGISTROS DE VELOCIDAD

```

DATCONTR LDAB #$40
      STAA PVEL
      LDAB  #$20
      STAA PVEL1
      LDAB  #$08
      STAA PVEL2
      LDYB  #$0010
      STY   QVEL
      LDY   #$0010
      STY   QVEL1
      LDY   #$0010

```

```

STY QVEL2
CLR YVEL
CLR YTACO
CLR XVEL
CLR XVEL1
CLR XVEL2
CLR SVEL

```

;REGISTROS DE DIRECCION

```

LDAB  #$45
STAA  PPOS
LDAB  #$4C
STAA  PPOS1
LDAB  #$15
STAA  PPOS2
LDY   #$0A
STY   QPOS
LDY   #$10
STY   QPOS1
LDY   #$10
STY   QPOS2
CLR   YPOS
CLR   XPOS
CLR   XPOS1
CLR   XPOS2
CLR   SPOS

```

*****RUTINA DE CONTROL
; DE DIRECCION, REALIZACION DEL PID

```

RTIS  LDAB  XPOS1      ; CORRIMIENTOS
      STAB  XPOS2
      LDAB  XPOS
      STAB  XPOS1
      LSL   SPOS
      LDAA  ADR2,X
      STAA  ANGM

```

```

CNTRL LDAA  DIREC      ;   OBTENCION DEL ERROR
      CMPA  ANGM
      BMI  SNEG

```

```

SPST  LDAA  DIREC
      SUBA  ANGM
      STAA  XPOS
      LDAB  SPOS
      ANDB  #$06
      STAB  SPOS
      BRA  CTLPOS

```

```

SNEG  LDAB  ANGM
      SUBB  DIREC
      STAB  XPOS
      LDAB  SPOS
      ORAB  #$01
      STAB  SPOS

```

* ECUACION A REALIZAR

$$* YPOS(n) = YPOS(n-1) + \frac{PPOS}{QPOS} * XPOS(n) - \frac{PPOS1}{QPOS1} * XPOS(n-1) + \frac{PPOS2}{QPOS2} * XPOS(n-2)$$

CTLPOS ; INICIO DEL CONTROL DE DIRECCION

TERM1 LDAA PPOS ; PRIMER TERMINO DE LA ECUACION

LDAB XPOS

MUL

LDX QPOS

IDIV

STX TEMP

LDAA SPOS

BITA #\$01

BEQ TIPOS

TINEG LDAB SYPOS

BITB #\$01

BEQ TINP

TIES LDD YPOS

ADDD TEMP

STD YPOS

BRA TERM2

TINP LDD YPOS

CPD TEMP

BHI T1MA

TIME LDD TEMP

SUBD YPOS

STD YPOS

LDAB SYPOS

EORB #\$01

STAB SYPOS

BRA TERM2

T1MA LDD YPOS

SUBD TEMP

STD YPOS

BRA TERM2

TIPOS LDAB SYPOS

BITB #\$01

BEQ TIES

BRA TINP

;SEGUNDO TERMINO DE LA ECUACION

TERM2 LDAA PPOS1

LDAB XPOS1

MUL

LDX QPOS1

IDIV

STX TEMP

LDAA SPOS

BITA #\$02

BNE T2POS

T2NEG LDAB SYPOS
 BITB #\$01
 BEQ T2NP

T2ES LDD YPOS
 ADDD TEMP
 STD YPOS
 BRA TERM3

T2NP LDD YPOS
 CPD TEMP
 BHI T2MA

T2ME LDD TEMP
 SUBD YPOS
 STD YPOS
 LDAB SYPOS
 EORB #\$01
 STAB SYPOS
 BRA TERM3

T2MA LDD YPOS
 SUBD TEMP
 STD YPOS
 BRA TERM3

T2POS LDAB SYPOS
 BITB #\$01
 BEQ T2ES
 BRA T2NP

;TERCER TERMINO DE LA ECUACION

TERM3 LDAA PPOS2
 LDAB XPOS2
 MUL
 LDX QPOS2
 IDIV
 STX TEMP
 LDAA SPOS
 BITA #\$04
 BNE T3POS

T3NEG LDAB SYPOS
 BITB #\$01
 BEQ T3NP

T3ES LDD YPOS
 ADDD TEMP
 STD YPOS
 BRA NXT

T3NP LDD YPOS
 CPD TEMP
 BHI T3MA

T3ME LDD TEMP
 SUBD YPOS
 STD YPOS
 LDAB SYPOS
 EORB #\$01

```

      STAB SYPOS
      BRA  NXT
T3MA  LDD  YPOS
      SUBD TEMP
      STD  YPOS
      BRA  NXT

T3POS LDAB SYPOS
      BITB #$01
      BEQ  T3ES
      BRA  T3NP

NXT   LDX  #$1000
      LDD  YPOS
      STD  TOC3,X
      LDAB SYPOS
      BITB #$01
      BNE  IZQ
DER   BCLR PORTG,X,$10 ; DEFINICION DEL SENTIDO DEL GIRO
      BSET PORTG,X,$08
      BRA  NXT1
IZQ   BCLR PORTG,X,$08
      BSET PORTG,X,$10
NXT1  NOP

*****RUTINA DE CONTROL
                                           ; DE VELOCIDAD, REALIZACION DEL PID

INVEL LDAA VELOC ; APAGO MOTORES SI VELOCIDAD ES CERO
      CMPA #$00
      BNE  INVEL1

      BCLR OC1M,X,$40
      BRA  INVEL

INVEL1 BSET OC1M,X,$40
INVEL  LDAB XVEL1 ; CORRIMIENTOS DE DATOS
      STAB XVEL2
      LDAB XVEL
      STAB XVEL1
      LSL  SVEL
      LDAA ADRI,X
      STAA YTACO

      LDAA VELOC ;OBTENCION DEL ERROR
      CMPA YTACO
      BMI  SNEGV
SPOSV LDAA VELOC
      SUBA YTACO
      STAA XVEL
      LDAB SVEL
      ANDB #$06
      STAB SVEL
      BRA  CTLVEL

```

```

SNEGV LDAB YTACO
      SUBB VELOC
      STAB XVEL
      LDAB SVEL
      ORAB #S01
      STAB SVEL

```

```

*
*
*          PVEL          PVEL1          PVEL2
* YVEL(n) = YVEL(n-1) + ----- * XVEL(n) - ----- * XVEL(n-1) + ----- * XVEL(n-2)
*          QVEL          QVEL1          QVEL2

```

```

CTLVEL NOP          ;INICIO DE CONTROL DE VELOCIDAD
TERM1V LDAA PVEL
      LDAB XVEL
      MUL
      LDX QVEL
      IDIV
      STX TEMP
      LDAA SVEL
      BITA #S01
      BEQ TIPOSV

```

```

T1NEGV LDD YVEL
      SUBD TEMP          ;PRIMER TERMINO
      STD YVEL
      BRA TERM2V
TIPOSV LDD YVEL
      ADDD TEMP
      STD YVEL

```

```

TERM2V LDAA PVEL1          ;SEGUNDO TERMINO
      LDAB XVEL1
      MUL
      LDX QVEL1
      IDIV
      STX TEMP
      LDAA SVEL
      BITA #S02
      BNE T2POSV

```

```

T2NEGV LDD YVEL
      SUBD TEMP
      STD YVEL
      BRA TERM3V
T2POSV LDD YVEL
      ADDD TEMP
      STD YVEL

```

```

TERM3V LDAA PVEL2          ;TERCER TERMINO
      LDAB XVEL2
      MUL
      LDX QVEL2
      IDIV
      STX TEMP
      LDAA SVEL

```

```
    BITA  #$04
    BEQ   T3POSV
T3NEGV LDD  YVEL
        SUBD TEMP
        STD  YVEL
        BRA  NXTV
T3POSV LDD  YVEL
        ADDD TEMP
        STD  YVEL

NXTV  LDX  #$1000          ; VALOR QUE MANDA AL MOTOR
        LDD  YVEL
        STD  TOC2,X
        RTS

;FIN DE PROGRAMA.

□
```



```

        case 1: CIRCLE();
                break;
        case 2: FIG_REG();
                break;
        case 3: USUARIO();
                break;
    } /*fin switch*/
}while(op != 4);

clrscr();
}

/*REALIZACION DEL CIRCULO*/
void CIRCLE()
{
    float p, r, v, tt;
    int pl, ttl, offset = 127;
    char tt2, vl, di, end = 0xFF;
    div_t pasos;

    do{
        MARCO();
        fflush(stdin);
        gotoxy(15,8);
        printf("Radio (en cm/s) min = 45 : ");
        scanf("%f", &r);

        fflush(stdin);
        gotoxy(15,10);
        printf("Derecha \"D\" o Izquierda \"I\" ? : ");
        scanf("%c", &di);
        di = toupper(di);

        fflush(stdin);
        gotoxy(15,12);
        printf("Velocidad (en cm/s) min = 32 max = 100 ");
        scanf("%f", &v);
    }
    while (r < 45 || v < 20 || v > 100 || (di != 'D' && di != 'I'));

    tt = asin(18.7/r); /* giro */
    ttl = tt*(96/28)*57.3;
    if (di == 'D')
        ttl=-ttl;
    ttl = offset + ttl;
    tt2 = ttl;

    vl = v; /* velocidad */

    p = 2*3.1416*r;
    pl = p; /* dist, perimetro */
    pasos = div(pl,255);
}

```

```

printf("\n\nPulsa una tecla para enviar Datos");
getch(); /* ESTABILIZACION*/
bioscom(1, tt2 & 0XFE, COM1); /* manda TETA */
ESPERA();
bioscom(1, 0 & 0XFE, COM1); /* manda vel CERO */
ESPERA();
bioscom(1, 0XC8 & 0XFE, COM1); /* t 1 y 2 estabilización */
ESPERA();
bioscom(1, 0XC8 & 0XFE, COM1);
ESPERA();

/*CAMINA*/
bioscom(1, tt2 & 0XFE, COM1); /* manda teta de giro*/
ESPERA();
bioscom(1, vl & 0XFE, COM1); /* manda vel, dato */
ESPERA();

bioscom(1, (pasos.quot << 1) & 0xFE, 0); /* pasos ALTO */
ESPERA();
bioscom(1, pasos.rem & 0XFE, COM1); /* pasos BAJO */
ESPERA();

bioscom(1, end, COM1); /* manda Ffin, end */

printf("\n Terminó la transmisión \"pulsa una tecla\"");
getch();
}

void FIG_REG()
{
    int i, j;
    char *mandar, vl, dir, end = 0XFF;
    int r = 45, n, dl, vl;
    float dv, vv = 0x25;
    div_t dll;

    do{
        MARCO();

        fflush(stdin);
        gotoxy(15,8);
        printf("Lados de la figura mayor que 3, menor de 8 : ");
        scanf("%d", &n);

        fflush(stdin);
        gotoxy(15,10);
        printf("Sentido de giro ( D o I ): ");
        scanf("%c", &dir);
    }
}

```

```

dir = toupper(dir);

fflush(stdin);
printf("\n\t Distancia de los lados [cm   ");
scanf("%d", &dl);

fflush(stdin);
printf("\n\t Velocidad por lado (32<v< 100) ( en cm/s):   ");
scanf("%d", &vl);
}
while(n>8 || n<3 || vl<32 || vl>100 || (dir!='D' && dir!='I'));

if(dir == 'D')
    dir = 0x26;          /* a la derecha */
else
    dir = 0xD8;         /* a la izquierda */

r = 45;                /* radio minimo */
dv = (2*3.141592*r)/n; /* dist de c/u de los vertices */

dll = div(dl,255);

i=1;
for(j=1;j<=n;j++)
{
    /*ESPERA */
    mandar[i]=127;     /*TETA 7f */
    i++;
    mandar[i]=0;      /*velocidad cero */
    i++;
    mandar[i]= 0xC8;  /*t1 y 2 estabilización */
    i++;
    mandar[i]= 0xC8;
    i++;

    /*CAMINA*/
    mandar[i]=127;    /*teta 7f por lado, camina derecho */
    i++;
    mandar[i]=vl;     /*velocidad del lado */
    i++;
    mandar[i]=dll.quot << 1; /*d pte ALTA del lado*/
    i++;
    mandar[i]=dll.rem; /*d pte BAJA del lado*/
    i++;

    /*ESPERA */
    mandar[i]= dir;   /*teta maxima de vertice */
    i++;
    mandar[i]= 0;     /*velocidad de cero para estabilizar*/
    i++;
    mandar[i]= 0xC8;  /*timepol y 2 de estabilizacion*/
    i++;
    mandar[i]= 0xC8;
    i++;

    /*CAMINA */

```

```

    mandar[i]= dir;      /*teta maxima del vertice */
    i++;
    mandar[i]=vv;       /*velocidad del vertice*/
    i++;
    mandar[i]=0;        /*distancia parte ALTA vertice*/
    i++;
    mandar[i]=dv;       /*distancia parte BAJA vertice*/
    i++;

}      /*fin del for*/

printf("\n\nPulsa una tecla para enviar Datos");
getch();

for(i=1;i<=n*4*4;i++) /*4 grupos de 4 datos por lado*/
{
    bioscom(1, mandar[i] & 0xFE , COM1);
    ESPERA();
}

bioscom(1, end, COM1); /* Fin de transmision */
printf("\nTerminó la transmisión \"pulsa una tecla\");
getch();
}

void USUARIO()
{
    int op, vv, gr, r, offset = 127, tt1, cnt, i, pl, dl, vl;
    char *mandar, dir, tt2, vv1, end = 0xFF, v11;
    float p, tt;

    div_t dvl, dll;

    clrscr();
    MARCO();
    cnt = 0;
    do{
        gotoxy(15,5);
        printf(" ¿Que deseas? ");
        gotoxy(15,8);
        printf("1.- ARCO");
        gotoxy(15,9);
        printf("2.- RECTA");
        gotoxy(15,10);
        printf("3.- Salir y mandar datos");
        gotoxy(15,12);
        printf("Escoge tu opción : ");
        scanf("%d",&op);

        switch(op){
            case 1:do{

```

```

LIMPIA();
fflush(stdin);
gotoxy(15, 14);
printf("Sentido de giro ( D o I ): ");
scanf("%c", &dir);
dir = toupper(dir);

fflush(stdin);
gotoxy(15,15);
printf("Velocidad (32 < vv < 100) (cm/s): ");
scanf("%d", &vv);

fflush(stdin);
gotoxy(15,16);
printf("grados (10 < vv < 360): ");
scanf("%d", &gr);

fflush(stdin);
gotoxy(15,17);
printf("radio (45 < r ): ");
scanf("%d", &r);

}while(r < 45 || vv < 32 || vv > 100 || (dir != 'D'
&& dir != 'I'));

tt = asin(18.7/r);
ttl = tt*(96/28)*57.3; /* giro, direccion */
if (dir == 'D')
    ttl=-ttl;
ttl = offset + ttl;
tt2 = ttl;

vv1 = vv; /* velocidad */

p = 2*3.1416*r*gr/360;
pl = p; /* distancia recorrida */
dvl = div(pl,255);

/* ESTABILIZA*/
mandar[cnt] = tt2; /* teta */
cnt++;
mandar[cnt]= 0; /* velocidad de cero */
cnt++;
mandar[cnt]= 0xC8; /* t1, 2 estabilizacion */
cnt++;
mandar[cnt]= 0xC8;
cnt++;

/* CAMINA*/
mandar[cnt]=tt2; /* teta */
cnt++;
mandar[cnt]=vv1; /* vel arco */
cnt++;

```

```

        mandar[cnt]=dvl.quot << 1; /*dist pte ALTA arco*/
        cnt++;
        mandar[cnt]=dvl.rem;      /*dist pte BAJA arco*/
        cnt++;
        break;
    case 2:do{
        LIMPIA();
        fflush(stdin);
        gotoxy(15, 14);
        printf("Distancia:  ");
        scanf("%d", &dl);

        fflush(stdin);
        gotoxy(15,15);
        printf("Velocidad (32 < vl < 100) (cm/s):  ");
        scanf("%d", &vl);

    }while(dl = 0 || vl < 32 || vl > 100 );
    dll = div(dl,255);

        /* ESTABILIZA*/
        mandar[cnt] = 127;      /* teta */
        cnt++;
        mandar[cnt]= 0;        /* velocidad de cero */
        cnt++;
        mandar[cnt]= 0xC8;     /* t1 y 2 estabilización */
        cnt++;
        mandar[cnt]= 0xC8;
        cnt++;

        /*CAMINA*/
        mandar[cnt]=127;      /* camina derecho */
        cnt++;
        mandar[cnt]=vl;      /*velocidad */
        cnt++;
        mandar[cnt]=dll.quot << 1; /*d parte ALTA */
        cnt++;
        mandar[cnt]=dll.rem;  /*d parte BAJA del lado*/
        cnt++;
        break;
    } /*fin switch*/

}while(op != 3); /*general*/

if (cnt != 0){
    printf("\n\nPulsa una tecla para enviar Datos");
    getch();
    for(i=1;i<=cnt;i++) /* 4 gpos 4 dts por lado*/
    {
        bioscom(1, mandar[i] & 0xFE , COM1);
        ESPERA();
    }
    bioscom(1, end, COM1); /* Fin de transmision */
}

```

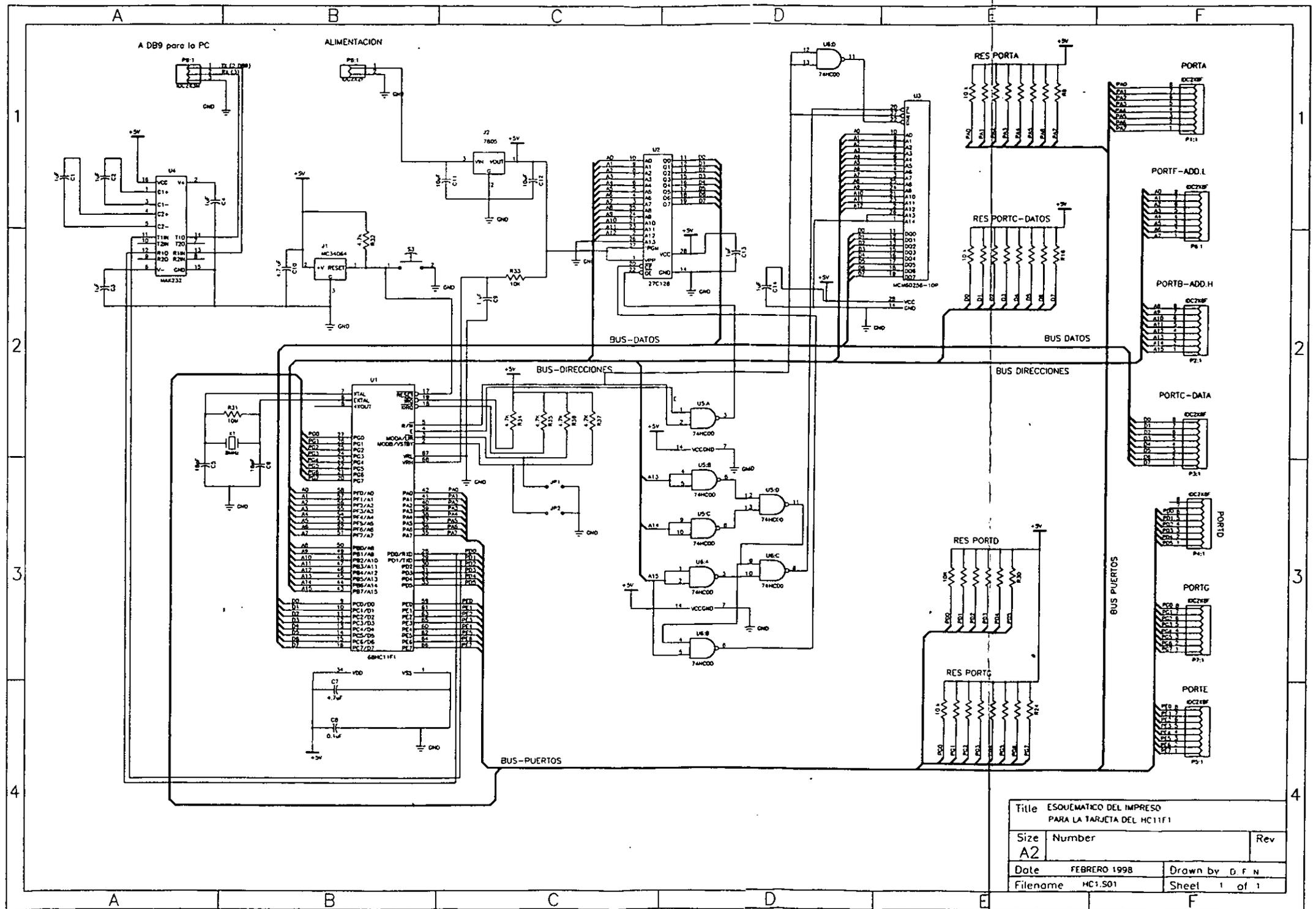
```
        printf("\nTerminó la transmisión \"pulsa una tecla\");
        getch();
    } /* del if */
}

void ESPERA()
{   int status;
    do
        status = 0x2000 & bioscom(3, 0, COM1);
        while (status != 0x2000);
}

void PORTADA()
{
    MARCO();
    gotoxy(15,5);
    printf("Diseño Construcción y control de un Robot Movil");
    gotoxy(15,7);
    printf("Programa, en Lenguaje C, alimentador de datos");
    gotoxy(15,10);
    printf("Integrantes: ");
    gotoxy(25,13);
    printf("Daniel Fernandez Nuñez");
    gotoxy(25,15);
    printf("Sergio Vera Rea");
    gotoxy(40,20);
    printf("Presiona una tecla para continuar...");
    getch();
}

void MARCO()
{   int f;
    clrscr();
    for(f=2;f<=79;f++){
        gotoxy(f,2);
        printf("±");
        gotoxy(f,24);
        printf("±");
    }
    for(f=3;f<=23;f++){
        gotoxy(2,f);
        printf("±");
        gotoxy(79,f);
        printf("±");
    }
}

void LIMPIA()
{
    int i;
    gotoxy(0,14);
    for(i=0;i<400;i++)
        printf(" ");
}
```



| | | |
|---|--------------|----------------|
| Title | | |
| ESQUEMATICO DEL IMPRESO PARA LA TARJETA DEL HC11F1 | | |
| Size | Number | Rev |
| A2 | | |
| Date | FEBRERO 1998 | Drawn by D.F.N |
| Filename | HC1.S01 | Sheet 1 of 1 |

8 BIBLIOGRAFÍA

- Joseph L. Jones, Anita M. Flynn, Mobile Robots.
Edit. Wellesley. Massachusetts. USA.
- Coughlin Robert y Driscoll Frederick. Amplificadores Operacionales y circuitos integrados lineales. Edit Prentice Hall. 1994.
- Jean Claude Latombe. Robot Motion Planning.
Edit. Kluwer Academic Publishers. 1991.
- Kuo, Benjamín. Sistemas de control Automático.
Edit. Prentice Hall. Septima edición, 1996.
- J. M. Angulo, Rafael Avilés Curso de Robótica.
Paraninfo 3ª ed. 1989
- S. Fu, R. C. González, C. S. G. Lee. Robotics: Control, Sensing, Vision and Intelligence. Ed McGraw-Hill
- John B. Peatman. Design with Microcontrollers
Edit. McGraw-Hill
- National Electronics. Linear Handbook.
- Motorola. Manual del Microcontrolador HC11.