

49

UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO

FACULTAD DE INGENIERIA



SISTEMA AUTOMATICO PARA LA DETERMINACION DE LA POSICION DE UN ROBOT MOVIL USANDO TRANSMISORES ULTRASONICOS

T E S I S

QUE PARA OBTENER EL TITULO DE INGENIERO MECANICO ELECTRICISTA EN EL AREA ELECTRICA-ELECTRONICA

P R E S E N T A

PABLO KRAUSE MANTILLA

DIRECTOR DE TESIS: YUKIHIRO MINAMI KOYAMA

MEXICO, D. F.

2000

279798



Universidad Nacional
Autónoma de México



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

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

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

A Papá y Mamá

A mis hermanos

A mis amigos

Deseo agradecer especialmente
a Yukihiro por su infinita
paciencia y ayuda.

Índice

1. Introducción	1
2 Análisis del problema.....	3
2.1 Clasificación de los sistemas de posición.....	3
2.1.1 Sistemas de posición relativa.....	3
2.1.1.1 Odométricos.....	3
2.1.1.2 Inerciales.....	3
2.1.2 Sistemas de posición absoluta.....	3
2.1.2.1 Independientes de la infraestructura.....	4
2.1.2.2 Dependientes de la infraestructura.....	4
2.1.2.2.1 Sistemas pasivos	4
2.1.2.2.2 Sistemas activos	4
2.2 Selección del sistema de posición.....	5
2.3 Componentes del sistema.....	6
2.3.1 Bocinas	7
2.3.2 Amplificador de Potencia.....	7
2.3.2.1 Clase A	7
2.3.2.2 Clase B.....	8
2.3.2.3 Clase AB.....	9
2.3.2.4 Clase C.....	10
2.3.2.5 Clase D	10
2.3.3 Micrófonos	10
2.3.3.1 Tipos de micrófono.....	10
2.3.3.1.1 Micrófonos dinámicos.....	11
2.3.3.1.2 Micrófonos de condensador	11
2.3.3.2 Patrones de recepción.....	11
2.3.4 Sistema de Procesamiento digital de señales.....	12
2.3.4.1 Filtro de entrada	13
2.3.4.2 Convertidor A/D.....	16
2.3.4.2.1 Convertidores de aproximación.....	16
2.3.4.2.2 Convertidores de tipo paralelo.....	17
2.3.4.2.3 Convertidores de integración.....	18
2.3.4.3 Filtros digitales.....	18
2.3.4.3.1 Filtros IIR.....	19
2.3.4.3.2 Filtro digital por medio de secciones bicuadráticas	23
3 Diseño y construcción.....	26

3.1 Cálculo de la posición por medio de las distancias a los faros.....	26
3.1.1 Aspectos Prácticos	28
3.1.2 Realización.....	28
3.1.2.1 Tarjeta de evaluación DSP56002	29
3.1.2.1.1 DSP56002	30
3.2 Cálculo de la posición por medio de la diferencia de las distancias a los faros.....	33
3.2.1 Aspectos prácticos	36
3.2.2 Realización	37
3.2.2.1 Circuito de control	39
4. Resultados	42
4.1 Configuración 1	44
4.2 Configuración 2	47
4.3 Configuración 3	49
5. Conclusiones	52
6. Referencias	53
7 Apéndices.....	55
7.1 Listado de programas.....	55
7.2 Hoja de Especificaciones del LM1875.....	72

1. Introducción

El aspecto medular de la robótica móvil es la navegación. Un robot móvil¹ debe ser capaz de trasladarse de un lugar a otro de manera eficaz y eficiente. El problema de la navegación se puede desglosar en tres partes.

- 1 determinar la posición real del robot;
2. determinar la posición a donde debe ir;
- 3 determinar la forma de llegar allí.

Las áreas que mayor atención reciben en la robótica son tal vez las relacionadas con inteligencia artificial, redes neuronales y lógica difusa. De nada sirve, sin embargo, el uso de estas fascinantes disciplinas si no se resuelve primero el problema de la posición del robot. pues ¿cómo saber a dónde ir si no se sabe dónde se está? ¿Cómo saber cuál es la mejor ruta a seguir si no se sabe el punto de partida? ¿Cómo saber si ya se llegó a donde se quiere si no se sabe dónde se está? Este trabajo se enfoca precisamente a explorar una posible solución a dicho problema.

La idea del sistema que se pretende desarrollar surgió a partir de la necesidad de controlar un robot móvil que fue adquirido por el Departamento de Computación de la Facultad de Ingeniería de la UNAM y con base en la publicación de Savage, Lepe-Casillas, Minami, et al (ref. [1]). El robot está construido alrededor de una microcomputadora personal con un microprocesador Pentium y sistema operativo Linux. Cuenta con arreglos de sensores ultrasónicos e infrarrojos alrededor de su estructura que le permiten detectar objetos cercanos. En la parte inferior se encuentran sensores de contacto que se activan en caso de colisiones inesperadas. El robot en el que se pretende implantar el sistema cuenta con un mecanismo de posición relativa de tipo odométrico (ver siguiente capítulo).

El objetivo de este trabajo es desarrollar un sistema de determinación de posición que sustituya o complemente al que posee actualmente el robot, con el fin de subsanar las deficiencias y limitaciones que presenta.

En el capítulo 2 se presenta la clasificación de los diferentes tipos de sistemas de localización, con una breve explicación de cada uno y algunas de sus ventajas y

¹ En adelante llamaremos simplemente robot para referirnos a robots móviles

desventajas. Posteriormente se presentan los conceptos y elementos que se requerirán para llevar a cabo el sistema propuesto.

En el capítulo 3 se presentan dos soluciones al problema planteado así como aspectos prácticos y de diseño. La primera correspondiente al cálculo de la posición por medio de las distancias a los faros. La segunda, el cálculo de la posición por medio de la diferencia de distancias a los faros, es un acercamiento diferente que simplifica algunos de los problemas prácticos que impone la realización del primer método.

En el capítulo 4 se presentan los resultados de los experimentos que resultan de interés para evaluar el desempeño del sistema desarrollado. Entre estos se encuentran experimentos con los que se evaluó el desempeño de los diferentes componentes así como del sistema en su conjunto.

2. Análisis del problema

2.1 Clasificación de los sistemas de posición²

2.1.1 Sistemas de posición relativa

Los sistemas de posición relativa se basan en la medición continua de la distancia y dirección del movimiento del robot a partir de un punto inicial conocido. Tiene las ventajas de ser autocontenido y de no requerir de preparación del entorno. El problema principal que presentan es que el error en la medición es acumulativo, lo que significa que crece indefinidamente con el paso del tiempo. Gran cantidad de robots móviles en la industria utilizan, al menos en parte, alguno de estos métodos para obtener su posición. Los dos principales sistemas de posición relativa son los odométricos y los inerciales.

2.1.1.1 Odmétricos

Estos sistemas se basan en la medición de la rotación y orientación de las ruedas del robot durante su movimiento. Estas mediciones son convertidas en vectores de desplazamiento que se van sumando para obtener la posición. Dado a que la tracción de las llantas del robot no es perfecta existen ligeros deslizamientos, principalmente en trayectorias curvas, que generan un error de medición que se va acumulando durante el recorrido del robot.

2.1.1.2 Inerciales

Los sistemas inerciales utilizan giróstatos y acelerómetros para detectar giro y aceleración del robot. Integrando las mediciones es posible obtener la posición del robot. Además del problema de la acumulación del error, este tipo de sistemas presenta la desventaja adicional del alto costo de los componentes.

2.1.2 Sistemas de posición absoluta

A diferencia de los sistemas de posición relativa, que son autocontenidos, los sistemas de posición absoluta dependen de elementos externos presentes en el entorno. La ventaja

² Adaptado de ref. [8] y ref. [7].

de estos mecanismos es que el error es acotado y predecible. Existen dos tipos fundamentales de sistemas de posición absoluta, los independientes de la infraestructura y los que son dependientes de ella.

2.1.2.1 Independientes de la infraestructura

Estos sistemas se basan en el mismo principio en el que los seres humanos y muchos otros organismos utilizan para desplazarse. La posición se determina en base a la detección y reconocimiento de características distintivas del entorno. La información obtenida se compara con mapas conocidos de antemano o generados durante un primer recorrido de reconocimiento. Este tipo de sistemas presenta grandes ventajas de flexibilidad, capacidad de adaptación y no requiere de preparación del entorno. El costo que significa la realización de un sistema de este tipo es sin embargo muy elevado, no sólo en términos económicos, sino también en complejidad.

2.1.2.2 Dependientes de la infraestructura

Este tipo de sistemas requiere de preparación previa del entorno para su funcionamiento. Dentro de esta categoría podemos distinguir dos tipos principales de sistemas. pasivos y activos.

2.1.2.2.1 Sistemas pasivos

En lugar de detectar y reconocer características propias del entorno, estos sistemas se basan en la detección de marcas artificiales distintivas colocadas de antemano en posiciones conocidas del entorno, por ejemplo pastillas magnéticas en el piso que pueden ser detectadas por el robot cuando pasa sobre ellas, cuerpos con superficies especiales que producen ecos característicos cuando son alcanzadas por ondas ultrasónicas o marcas ópticas detectables por sistemas de visión estereoscópica

2.1.2.2.2 Sistemas activos

La idea general detrás de los sistemas activos dependientes de la infraestructura consiste en colocar en el entorno emisores de algún tipo de señal para ser utilizados por el robot de manera similar en la que un barco hace uso de los faros para orientarse en medio de la obscuridad. Estos emisores, que en adelante llamaremos faros, se colocan en distintos lugares dentro de un área determinada de tal forma que un robot, equipado con un sensor adecuado, sea capaz de recibir las señales en cualquier lugar que se encuentre dentro de dicha área. La obtención de la posición se logra mediante la medición del

tiempo que tarda cada señal en viajar del emisor al sensor montado en el robot, o viceversa. También es posible determinar la posición midiendo el desfase de las señales, sobre todo cuando resulta difícil o imposible medir el tiempo de transmisión como en el caso de ondas electromagnéticas que viajan a gran velocidad.

El sistema puede ser independiente del robot o puede haber retroalimentación hacia uno u otro lado. También es posible que los emisores estén montados en el robot de tal manera que la detección se lleve a cabo en un sistema externo a éste, el cual después de calcular su posición la transmite al robot.

2.2 Selección del sistema de posición

El robot en el que se planea implantar el sistema cuenta con un sistema de posición relativa de tipo odométrico. Como se mencionó en la sección anterior, el error que presentan este tipo de sistemas crece indefinidamente con el paso del tiempo. Se pretende desarrollar un sistema que elimine este problema. Esto nos coloca en el terreno de los sistemas de posición absoluta. Dentro de éstos, los independientes de la infraestructura son tal vez los de más interés teórico con miras al desarrollo de un verdadero autómata, pero a la vez son los más complejos y los menos prácticos. Finalmente, los sistemas dependientes de la infraestructura cuentan con ventajas que los convierten en los más convenientes para resolver el problema planteado. Entre ellas la más importante es la precisión. El error que se presenta en estos sistemas tiene la ventaja de ser acotado y conocido. Además, al conocer el error existe la posibilidad de introducir técnicas para tratar de compensarlo o minimizarlo. Otra ventaja de gran importancia desde el punto de vista práctico es su simplicidad. El principio de funcionamiento del sistema es sencillo y relativamente fácil de desarrollar con componentes de bajo costo. Una posible ventaja adicional es que el sistema puede ser utilizado simultáneamente por varios robots utilizando la misma infraestructura. Dentro de los sistemas dependientes de la infraestructura, los activos son los más flexibles y requieren menor preparación del entorno. Es por esto que se eligió diseñar un sistema activo dependiente de la infraestructura.

En la figura 2.1 se muestra una posible configuración de un sistema de localización activo. Se colocan tres (o posiblemente más) faros en un área determinada. Cada faro emite una señal distintiva, la cual es recibida por un sensor adecuado montado en el robot. Conocida la velocidad de transmisión de la señal, y midiendo el tiempo que tarda cada una en alcanzar al robot es posible determinar la posición de éste. Para esto es

necesario poder distinguir cada una de las señales y posteriormente realizar cálculos matemáticos que permitan determinar la posición del robot

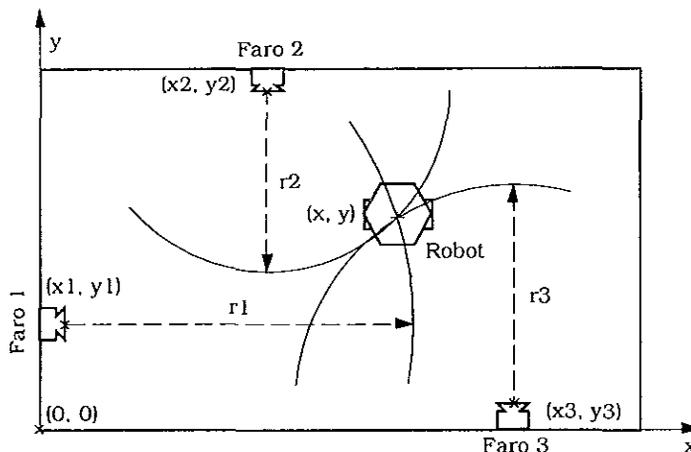


Figura 2.1. Sistema de localización activo, dependiente de la infraestructura³.

A continuación se presentan los elementos que se requieren para llevar a cabo el sistema previsto

2.3 Componentes del sistema

Para llevar a cabo un sistema de este tipo se identifican los siguientes componentes:

- Bocinas, como faros del sistema.
- Amplificadores de potencia para las bocinas.
- Un circuito de control encargado de activar las bocinas según convenga⁴.
- Un micrófono montado en el robot para detectar las señales emitidas por las bocinas
- Sistema de procesamiento digital de señales.

³ Figura adaptada de ref. [1].

⁴ El circuito de control depende del esquema que se adopte, por lo que éste se describe en el capítulo 3 de diseño y construcción.

- Un filtro para limitar el ancho de banda de la señal para poder llevar a cabo la conversión A/D.
- Un convertidor analógico/digital (A/D) para traducir la señal analógica proveniente del micrófono a una señal digital que pueda ser utilizada por dispositivo mencionado anteriormente
- Un dispositivo electrónico que permita analizar las señales captadas por el micrófono y realizar los cálculos necesarios para obtener la posición del robot.

2.3.1 Bocinas

Las bocinas son dispositivos electromecánicos que convierten señales eléctricas en señales acústicas. Las bocinas más comunes están compuestas de un imán cilíndrico permanente fijo al armazón, alrededor del cual se encuentra una bobina, la cual a su vez está adherida a la parte más angosta del cono. El cono se encuentra unido al armazón del dispositivo en su parte más ancha mediante un material flexible que permite el movimiento del cono. Al fluir corriente a través de la bobina se produce un campo magnético, que dependiendo de la dirección de la corriente, es atraído o repelido por el campo magnético del imán permanente, lo que produce la vibración del cono y por consiguiente del aire. Las características a tener en cuenta para la selección de las bocinas es su respuesta en frecuencia, su ángulo de dispersión, el cual deseamos que sea lo más amplio posible, y su potencia.

2.3.2 Amplificador de Potencia

El objetivo de un amplificador de potencia es entregar, con la menor distorsión posible, el máximo nivel de voltaje a un dispositivo de baja impedancia, en este caso una bocina. Las características principales de un amplificador de potencia son su eficiencia, la potencia máxima de operación y la impedancia de salida. Los amplificadores de potencia se clasifican de acuerdo al porcentaje de tiempo en el que fluye corriente (la corriente es diferente de 0) por el colector del transistor que lo constituye para un ciclo completo de la señal de entrada (ref. [2]). Enunciado de otra forma, los amplificadores de potencia se clasifican de acuerdo a la variación de la señal de salida para un ciclo completo de la señal de entrada (ref.[3]).

2.3.2.1 Clase A

La corriente en el colector es diferente de 0 en todo momento, o lo que es lo mismo, la señal de salida varía para la totalidad del ciclo (360°) de entrada. El punto de operación del transistor (Q) debe ser tal que permita la amplificación tanto de la parte positiva del ciclo

de entrada como el de la parte negativa, sin que el transistor entre en corte o en saturación (figura 2.2). Esta clase de amplificadores son ineficientes debido a que fluye corriente por el colector en todo momento, aun cuando la señal de entrada es cero. La principal ventaja que presenta este tipo de amplificadores es su baja distorsión.

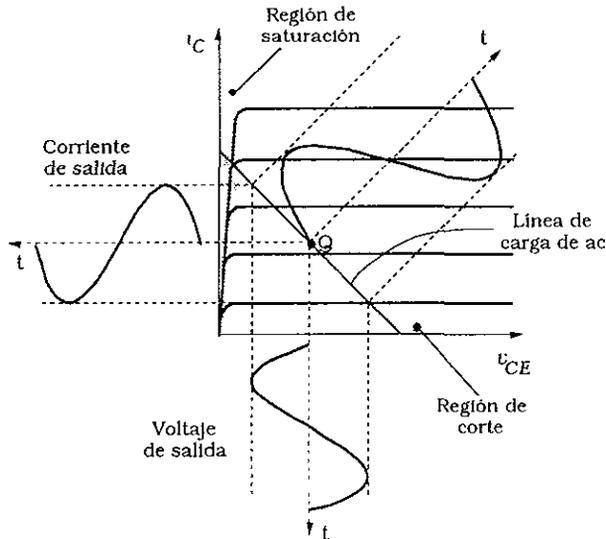


Figura 2.2. Operación de un amplificador clase A.

2.3.2.2 Clase B

Este tipo de amplificadores generan señal de salida para la mitad (180°) del ciclo de la señal de entrada. El punto de operación del transistor es de $0V$ (figura 2.3-a). Por supuesto que la distorsión de un amplificador de este tipo es muy alta, pero si se utiliza uno para amplificar la parte positiva del ciclo y otro para la parte negativa se obtiene una configuración conocida con el nombre *push-pull* o de simetría complementaria (figura 2.4), capaz de amplificar la totalidad del círculo de la señal de entrada. Aún así existe distorsión cerca de cero ya que esta región de operación (región de corte) es no lineal (figura 2.3-b). La ventaja de una configuración de este tipo es que cuando la señal de entrada es cero, la disipación de energía es casi nula ya que no fluye corriente por el colector de los transistores.

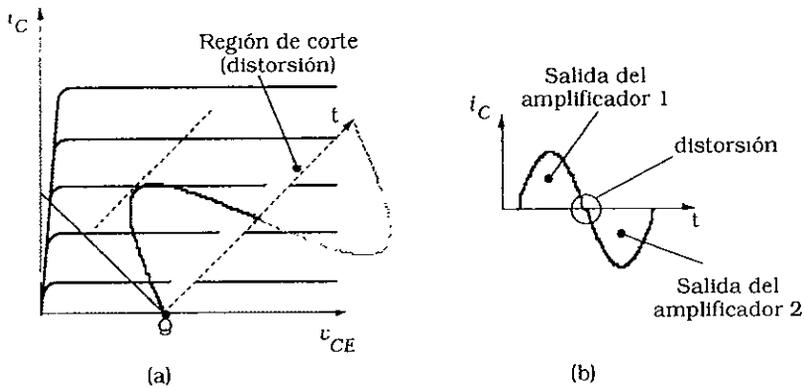


Figura 2.3. Operación de un amplificador clase B.

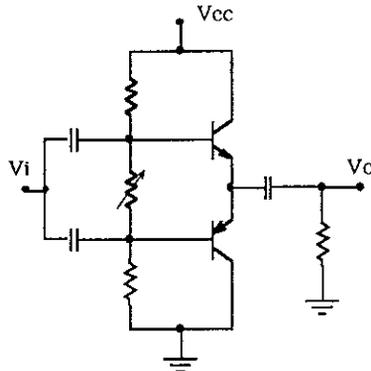


Figura 2.4. Amplificadores clase B en configuración *push-pull*.

2.3.2.3 Clase AB

Si hacemos que el punto de operación del transistor sea mayor a cero, lograremos que éste opere todo el tiempo en la región lineal, lo que disminuye su distorsión. Dependiendo del punto de operación que se seleccione este tipo de amplificadores operan de entre la mitad (180°) y el total (360°) del ciclo de la señal de entrada. Para amplificar un ciclo completo de la señal de entrada es necesario utilizar también una configuración *push-pull* como la descrita anteriormente. Este tipo de amplificadores combina la baja distorsión de los amplificadores clase A y la eficiencia de los clase B.

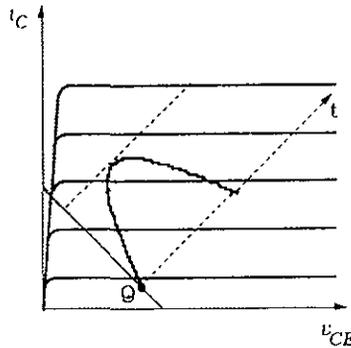


Figura 2.5. Operación de un amplificador clase AB.

2.3.2.4 Clase C

Los amplificadores clase C son aquellos que operan durante menos de la mitad de un ciclo ($< 180^\circ$) de la señal de entrada, es decir, la corriente en el colector del transistor es diferente de cero menos del 50% del tiempo. Este tipo de amplificadores se utilizan en combinación con un circuito resonante el cual hace que la salida se aproxime a una señal senoidal cuya frecuencia fundamental es la misma que la de la señal de entrada. Son utilizados en aplicaciones muy específicas donde se desea amplificar a una señal senoidal pura, como en radio y comunicaciones.

2.3.2.5 Clase D

Este es un tipo especial de amplificadores que operan mediante pulsos de entrada, o señales digitales que son positivas durante lapsos muy cortos y cero durante lapsos relativamente largos (ref.[3]). Mediante el uso de técnicas digitales es posible generar una señal de salida para un ciclo completo. Esta clase de amplificadores pueden llegar a tener una eficiencia muy alta de alrededor del 90%.

2.3.3 Micrófonos

Los micrófonos son transductores: son dispositivos que convierten energía acústica en energía eléctrica

2.3.3.1 Tipos de micrófono

Existe gran variedad de micrófonos en el mercado, principalmente por el gran uso en la industria de la música, sin embargo existen dos tipos básicos de micrófonos:

2.3.3.1.1 *Micrófonos dinámicos*

Los micrófonos dinámicos están formados por un diafragma, una bobina de material conductor, adherida al diafragma, e inmersa en el campo magnético de un imán permanente. Las ondas acústicas provocan la vibración del diafragma, y por consiguiente, la de la bobina. El movimiento vibratorio de la bobina inmersa en el campo magnético produce minúsculas corrientes en ésta, permitiendo la transformación de sonido en señales eléctricas.

2.3.3.1.2 *Micrófonos de condensador*

El micrófono de condensador básico está formado por un diafragma conductor montado a una muy pequeña distancia de una placa metálica, formando un condensador, o más precisamente, un capacitor. La vibración del diafragma produce cambios minúsculos en la capacitancia del dispositivo, cosa que es aprovechada para generar una señal eléctrica con ayuda de una pila conectada a dicho capacitor.

Una variante de este tipo de micrófonos son los llamados micrófonos electret, en los cuales, el diafragma, o en ocasiones la placa trasera, está fabricado de un material, generalmente plástico, con carga eléctrica propia.

2.3.3.2 *Patrones de recepción*

Atendiendo a su capacidad de recepción desde distintos ángulos, existen los siguientes tipos de micrófonos.

- Unidireccional
- Bidireccional
- Cardioide
- Omnidireccional

Las bondades o defectos de los micrófonos en boca de ingenieros de sonido y audiófilos son generalmente muy subjetivos, pues en el ámbito de la música el micrófono que mejor reproduzca la realidad no es necesariamente, ni generalmente, el preferido. En el caso de nuestro sistema se buscan dos características fundamentales: que su recepción sea omnidireccional, y que tenga una respuesta en frecuencia adecuada para la señal que se desee utilizar.

2.3.4 Sistema de Procesamiento digital de señales

El procesamiento digital de señales es la transformación y/o extracción de información (procesamiento) de una función de una o mas variables independientes (señales) por medio de un dispositivo electrónico digital (digital)⁵ En la (figura 2.6) se muestra un diagrama de bloques de un sistema general de procesamiento digital de señales. El proceso comienza filtrando la señal analógica para limitar su ancho de banda con el fin de impedir el paso de componentes de alta frecuencia al convertidor A/D. Eso se hace para evitar que se produzcan señales "fantasma" de baja frecuencia debido al muestreo de componentes de alta frecuencia, un fenómeno conocido como *traslape*⁶

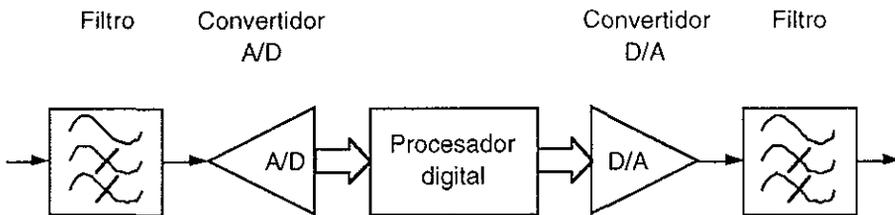


Figura 2.6. Sistema general de procesamiento digital de señales.

La frecuencia de muestreo impone un límite en la frecuencia máxima que puede contener la señal que deseamos muestrear. El teorema de Nyquist indica que una señal sólo puede ser reconstruida a partir de sus muestras si la frecuencia de muestreo empleada es mayor a dos veces la máxima frecuencia presente en la señal original. Es importante tomar en cuenta este criterio al seleccionar la frecuencia de las señales con las que se desea trabajar. Si se alimentan señales con frecuencias mayores a la mitad de la frecuencia de muestreo se produce el fenómeno de *traslape* que se mencionó anteriormente. Es precisamente por esta razón por la que es importante contar con un filtro de entrada que limite el ancho de banda de la señal.

Una vez convertida la señal al mundo digital ésta es procesada mediante un microprocesador. Existen microprocesadores específicamente diseñados para este fin que reciben el nombre de procesadores digitales de señales o DSP, por sus siglas en inglés. Un DSP es un microprocesador diseñado y optimado para realizar operaciones de filtrado, convolución, correlación, rectificación, amplificación y transformación de señales. Lo que distingue a un DSP de un microprocesador tradicional es su arquitectura. En general, los

⁵ ref. [12]

⁶ Concido en inglés como *aliasing*.

DSP utilizan arquitectura Harvard, que se distingue por contar con *buses* independientes de datos, de direcciones y de programa, así como varias unidades de ejecución independientes. Esto permite efectuar operaciones de tipo multiplicación y acumulación (MAC) en pocos ciclos de reloj.

Una vez procesada la señal es posible, si es necesario, producir una señal de salida analógica para lo cual se requiere de un convertidor D/A y un filtro paso bajas. El filtro se encarga de "suavizar" la señal que sale del convertidor D/A, la cual presenta bordes debido al fenómeno de cuantización, producto de una señal digital. En el caso de nuestro sistema no se requiere de una salida analógica, pues el proceso concluye una vez procesada la señal y obtenida la posición del robot.

Se pretende utilizar el procesamiento digital de señales para la realización del sistema debido a las ventajas que ofrece. El procesamiento digital de señales transforma el problema del diseño de un sistema electrónico complejo en un problema de programación, lo que reduce la complejidad, y por ende, el costo y tiempo de desarrollo del sistema. Además permite gran flexibilidad, pues cualquier cambio o ajuste del sistema se puede llevar a cabo fácilmente modificando el programa. Además, estos sistemas son más confiables ya que no presentan problemas relacionados con la variación de los parámetros eléctricos de los componentes debido a cambios de temperatura o envejecimiento de los dispositivos.

2.3.4.1 Filtro de entrada

La señal de entrada puede ser amplificada y filtrada por un filtro activo paso banda de segundo orden del tipo de realimentación múltiple como el que se muestra en la figura 2.7.

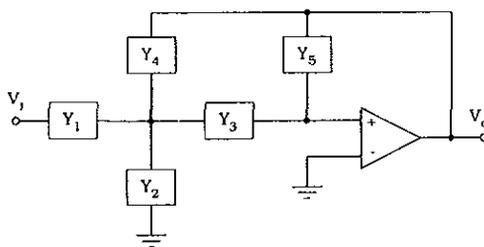


Figura 2.7 Estructura general de realimentación múltiple.

La función de transferencia de este circuito está dado por la siguiente expresión, en términos de la admitancia de los componentes

$$\frac{V_o}{V} = - \frac{Y_1 Y_3}{Y_3(Y_1 + Y_2 + Y_3 + Y_4) + Y_1 Y_4} \quad (2.1)$$

Si queremos diseñar un filtro paso banda necesitamos seleccionar los componentes de tal manera que la función de transferencia tenga la siguiente forma

$$H(j\omega) = \frac{H_o}{1 + jQ\left(\frac{\omega}{\omega_o} - \frac{\omega_o}{\omega}\right)} \quad (2.2)$$

En esta expresión, H_o es la ganancia del filtro, ω_o es la frecuencia central de la banda de paso, y Q es el factor de calidad. Podemos expresar la función de transferencia en función de la variable compleja s de la siguiente manera

$$H(s) = \frac{H_o 2\alpha s}{s^2 + 2\alpha s + \omega_o^2} \quad (2.3)$$

El factor de calidad se define como $Q = \frac{\omega_o}{2\alpha}$, donde α es la constante de amortiguamiento

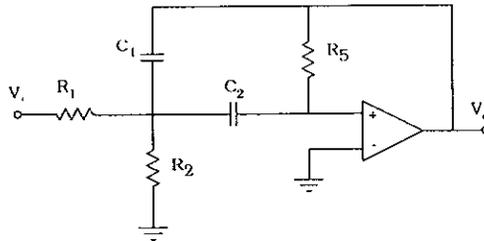


Figura 2.8 Filtro paso banda de segundo orden con realimentación múltiple.

Para llevar a cabo un filtro de estas características se requiere de una estructura como la que se muestra en la figura 2.8, cuya función de transferencia, según la expresión (2.1), es

$$H(s) = - \frac{s \frac{1}{R_1 C_1}}{s^2 + s \frac{C_1 + C_2}{R_1 C_1 C_2} + \frac{R_1 + R_2}{R_1 R_1 R_1 C_1 C_2}} \quad (2.4)$$

para simplificar el diseño hacemos

$$C = C_1 = C_2 \quad (2.5)$$

y la función de transferencia se convierte en

$$H(s) = \frac{s \frac{1}{R_1 C}}{s^2 + s \frac{2}{R_1 C} + \frac{R_1 + R_2}{R_1 R_1 R_1 C^2}} \quad (2.6)$$

Si comparamos nuestra función de transferencia con la de la forma general de un filtro paso bajas, dada por la expresión (2.3), podemos obtener las siguientes relaciones

$$H_o 2\alpha = \frac{1}{R_1 C} \quad (2.7)$$

$$2\alpha = \frac{2}{R_1 C} \quad (2.8)$$

$$\omega_o^2 = \frac{R_1 + R_2}{R_1 R_1 R_1 C^2} \quad (2.9)$$

A partir de estas tres expresiones podemos calcular los valores de los elementos del filtro en función de ω_o , H_o y α

$$R_1 = \frac{1}{H_o 2\alpha C} \quad (2.10)$$

$$R_1 = \frac{1}{\alpha C} \quad (2.11)$$

y finalmente, sustituyendo (2.10) y (2.11) en (2.9) obtenemos

$$R_2 = \frac{\alpha}{\omega_o^2 C - 2\alpha^2 H_o C^2} \quad (2.12)$$

Recordemos que

$$\omega_c = 2\pi f_c$$

$$Q = \frac{\omega_c}{2\alpha}$$

$$Q \approx \frac{f_c}{BW}$$

$$BW = f_h - f_l$$

$$f_c = \sqrt{f_l f_h}$$

donde BW es el ancho de banda del filtro, f_l y f_h son las frecuencias de corte de -3 dB, inferior y superior respectivamente.

2.3.4.2 Convertidor A/D

Como se mencionó anteriormente, el convertidor A/D nos permite convertir una señal analógica a una digital. Existen dos parámetros importantes a considerar en la elección de un convertidor A/D: su resolución y su frecuencia de muestreo.

Debido a que una señal digital cuenta con una cantidad finita, y generalmente pequeña, de bits para representar cada muestra (resolución), la señal digital resultante presenta un error llamado error de cuantización. Entre mayor sea el número de bits que se utilicen para codificar cada muestra menor será el error de cuantización

Existe una gran variedad de convertidores A/D, sin embargo existen tres tipos fundamentales ref.[4]: de aproximación, paralelos, de integración.

2.3.4.2.1 Convertidores de aproximación

Este tipo de convertidores A/D utilizan un convertidor digital a analógico (D/A) invirtiendo su función⁷. La forma básica cuenta con un comparador analógico, un contador digital y, por supuesto, un convertidor D/A (figura 2.9). Las salidas del contador están conectadas a las entradas del convertidor D/A. El contador inicia un conteo secuencial a partir de 0, con lo que se produce una señal en escalera creciente a la salida del convertidor D/A. Esta señal es comparada continuamente con la señal que se desea convertir V_i . En cuanto la señal en escalera alcanza el valor de V_i , el comparador se activa con lo que se detiene el contador y se activa una señal de fin de conversión (EOC). En ese instante, la salida del contador es una representación digital de la señal de entrada.

⁷ Existen a su vez una gran variedad de convertidores D/A, consultar ref. [4].

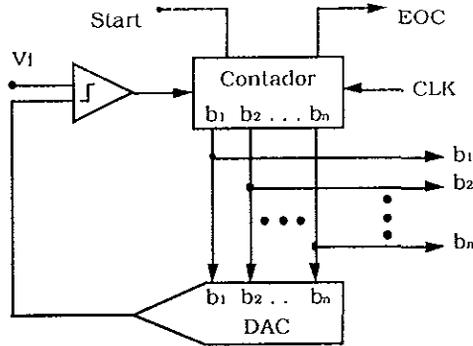


Figura 2.9 Estructura general de realimentación múltiple.

Existen algunas variantes de este sistema que lo hacen más eficiente, como iniciar el conteo desde algún valor cercano a la última muestra en vez de empezar de cero cada vez. El esquema más popular, conocido como el de aproximaciones sucesivas, consiste en empezar por el bit más significativo (MSB). Si al activar el MSB la salida del convertidor D/A es mayor que la señal analógica, el bit es puesto en cero y se repite el proceso con el siguiente bit más significativo. Este proceso converge mucho más rápido que empezar el conteo a partir de 0 cada vez.

2.3.4.2.2 Convertidores de tipo paralelo

Este tipo de convertidores A/D, también llamados de tipo *flash*, llevan a cabo la conversión A/D en un solo paso. Cuentan con una serie de 2^n resistencias que dividen el voltaje V_i en $(2^n - 1)$ niveles que sirven de referencia a un conjunto de comparadores analógicos (figura 2.10). El valor de la señal analógica V_i es comparada con los diferentes voltajes de referencia. Esto produce que se active un número determinado de comparadores. Finalmente se utiliza un codificador para convertir el estado de los comparadores a una señal binaria.

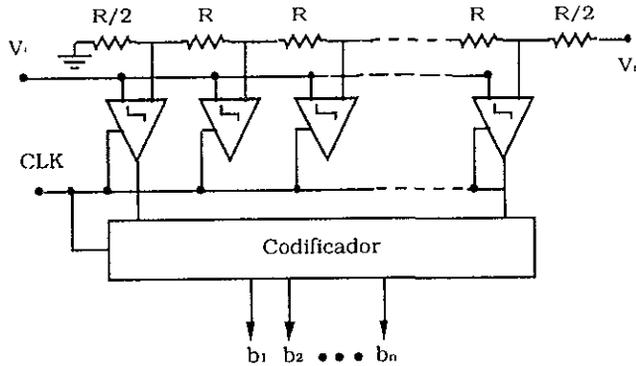


Figura 2.10 Estructura general de realimentación múltiple.

2.3.4.2.3 Convertidores de integración

Este tipo de convertidores A/D llevan a cabo la conversión de manera indirecta convirtiendo la señal analógica de entrada a una señal lineal y de ahí a una forma digital. Por ejemplo los de balanceo de carga convierten la señal de entrada a frecuencia la cual es medida por un contador y convertida a un código digital proporcional a la señal de entrada. Otro convertidor de este tipo es el de pendiente o rampa doble. La señal de entrada V_i es integrada por un periodo fijo. Posteriormente se integra una señal interna de referencia V_r , de polaridad inversa, hasta que la salida del integrador regrese a cero. El tiempo que tarda la señal del integrador en volver a cero es proporcional al promedio de V_i durante el periodo inicial de integración. Este tiempo es convertido a un código digital que representa la señal deseada.

2.3.4.3 Filtros digitales

Una de las operaciones más comunes en el procesamiento digital de señales es el filtrado. Filtrar es el proceso mediante el cual se modifica la amplitud y/o fase de las componentes en frecuencia de una señal con el fin de mejorar la calidad de las señales o bien, para extraer información de ellas.

Un filtro digital es aquél que lleva a cabo el filtrado (según la definición anterior) de una señal mediante un algoritmo matemático dentro de un sistema de cómputo. Existen dos tipos generales de filtros digitales. IIR (*Infinite impulse response*) y FIR (*Finite impulse*

response). Si la respuesta de un sistema ante una excitación de tipo pulso unitario tiene una duración finita se dice que el sistema es de respuesta finita al impulso (FIR), de lo contrario se dice que el sistema es de respuesta infinita al impulso (IIR). Los filtros IIR son filtros con realimentación, mientras que los filtros FIR en general no cuentan con realimentación.

El tipo de filtro que se debe utilizar depende de cada de la aplicación. Los filtros FIR se utilizan en aplicaciones que requieren una respuesta en frecuencia lineal en la banda de paso. Además son estables y poco sensibles a cambios en los parámetros. Por otro lado, los filtros IIR no tienen respuesta en frecuencia lineal, son más sensibles a cambios en los parámetros y en ocasiones presentan problemas de estabilidad, sin embargo logran un mejor desempeño con menor número de parámetros, requieren menos memoria y menos operaciones que los filtros FIR. Estas características los hacen ideales para aplicaciones en tiempo real siempre y cuando no se requiera de una respuesta en frecuencia lineal. Como se verá más adelante, para nuestro sistema requerimos de filtros que sean rápidos y eficientes tanto en memoria como en procesamiento y no requerimos de una respuesta lineal por lo que nuestro interés se centra en los filtros IIR los cuáles se analizan a continuación.

2.3.4.3.1 Filtros IIR⁹

La función de transferencia de un filtro digital está dada por

$$H(z) = \frac{\sum_{k=0}^M b_k z^{-k}}{1 + \sum_{k=1}^N a_k z^{-k}} \quad (2.13)$$

de donde podemos obtener

$$Y(z) = X(z) \sum_{k=0}^M b_k z^{-k} - Y(z) \sum_{k=1}^N a_k z^{-k} \quad (2.14)$$

o bien

$$y(n) = \sum_{k=0}^M b_k x(n-k) - \sum_{k=1}^N a_k y(n-k) \quad (2.15)$$

⁹ La información de esta sección fue adaptada de ref. [12]

Esta expresión representa la forma directa I, cuyo diagrama de bloques se muestra en la figura 2.11

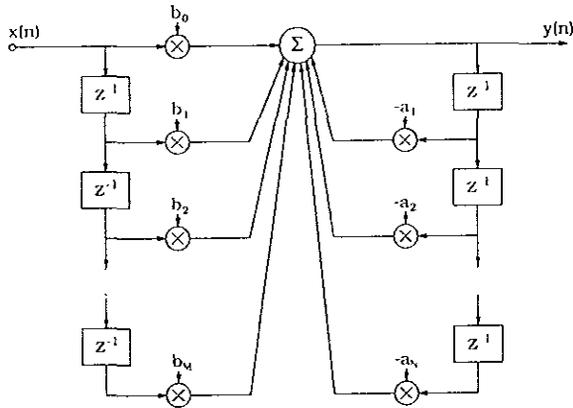


Figura 2.11 Forma directa I de un filtro digital.

Existe sin embargo una forma más eficiente de expresar la función de transferencia de un filtro digital utilizando menos retrasos. Para obtener dicha expresión primero descomponemos la función de transferencia original en dos

$$H(z) = H_1(z)H_2(z) \quad (2.16)$$

donde

$$H_1(z) = \frac{1}{1 + \sum_{t=1}^N a_t z^{-t}} \quad (2.17)$$

y

$$H_2(z) = \sum_{t=0}^M b_t z^{-t} \quad (2.18)$$

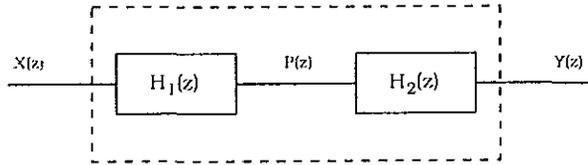


Figura 2.12 Descomposición de la función de transferencia para la realización de la forma directa II.

De la figura podemos ver que

$$H_1(z) = \frac{P(z)}{X(z)} \quad (2.19)$$

$$P(z) = X(z)H_1(z) \quad (2.20)$$

es decir

$$P(z) = X(z) \frac{1}{1 + \sum_{t=0}^N a_t z^{-t}} \quad (2.21)$$

de donde

$$P(z) = X(z) - P(z) \sum_{t=1}^N a_t z^{-t} \quad (2.22)$$

cuya transformada inversa es

$$p(n) = x(n) - \sum_{t=1}^N a_t p(n-t) \quad (2.23)$$

De la misma manera que con H_1 ,

$$H_2(z) = \frac{Y(z)}{P(z)} \quad (2.24)$$

$$Y(z) = P(z)H_2(z) \quad (2.25)$$

es decir

$$Y(z) = P(z) \sum_{k=0}^M (b_k z^{-k}) \quad (2.26)$$

cuya transformada inversa es

$$y(n) = \sum_{k=0}^M b_k p(n-k) \quad (2.27)$$

Sustituimos (2.23) en (2.27) para obtener finalmente

$$y(n) = b_0 x(n) - b_0 \sum_{k=1}^N a_k p(n-k) + \sum_{k=1}^M b_k p(n-k) \quad (2.28)$$

o también

$$y(n) = b_0 x(n) + \sum_{k=1}^N (b_k - b_0 a_k) p(n-k) \quad (2.29)$$

donde $b_k = 0$ para $M < k \leq N$.

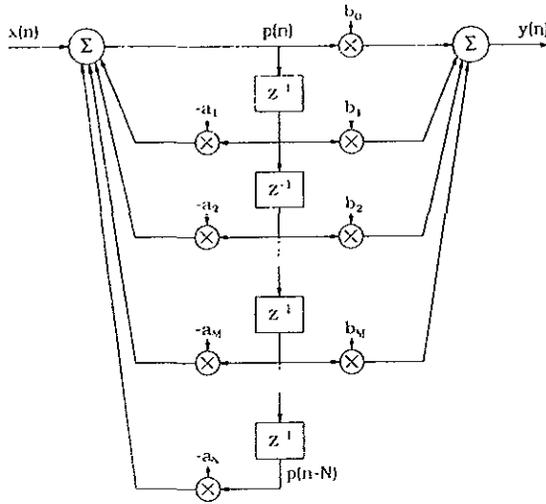


Figura 2.13 Forma directa II de un filtro digital.

2.3.4.3.2 Filtro digital por medio de secciones bicuadráticas

Una forma más práctica para llevar a cabo un filtro digital es mediante el encadenamiento de secciones bicuadráticas. La forma general de una de estas secciones es

$$H(z) = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2}}{1 + a_1 z^{-1} + a_2 z^{-2}} \quad (2.30)$$

o bien

$$y(n) = b_0 x(n) + b_1 x(n-1) + b_2 x(n-2) - a_1 y(n-1) + a_2 y(n-2) \quad (2.31)$$

Ésta es la forma directa I de cada sección bicuadrática. En la figura 2.14 se muestra la forma directa II, dada por las ecuaciones (2.32) y (2.33).

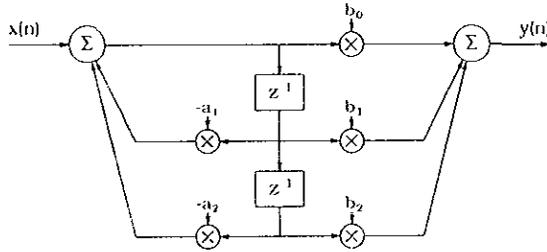


Figura 2.14 Forma directa II de una sección bicuadrática de un filtro digital.

$$p(n) = x(n) - a_1 p(n-1) - a_2 p(n-2) \quad (2.32)$$

$$y(n) = b_0 p(n) + b_1 p(n-1) + b_2 p(n-2) \quad (2.33)$$

La función de transferencia () también se puede expresar de la siguiente manera:

$$H(z) = \frac{b_0 (1 + b'_1 z^{-1} + b'_2 z^{-2})}{1 + a_1 z^{-1} + a_2 z^{-2}} \quad (2.34)$$

donde

$$b'_1 = \frac{b_1}{b_0} \quad (2.35)$$

y

$$b'_2 = \frac{b_2}{b_0} \quad (2.36)$$

con lo que se llega a una expresión alternativa de la forma directa II dada por las expresiones

$$p(n) = b_0 x(n) - a_1 p(n-1) - a_2 p(n-2) \quad (2.37)$$

$$y(n) = p(n) + b_1 p(n-1) + b_2 p(n-2) \quad (2.38)$$

y cuyo diagrama de bloques se muestra en la figura 2.15.

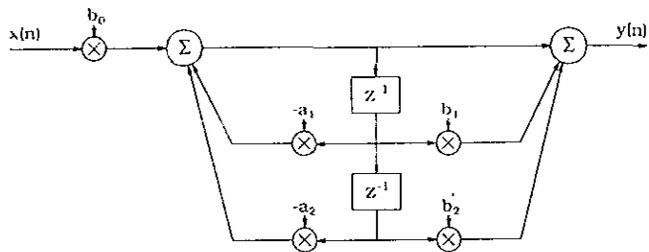


Figura 2.15 Forma directa II alternativa de una sección bicuadrática de un filtro digital.

3. Diseño y construcción

En este capítulo se describen dos soluciones al problema planteado. La primera es la más natural y corresponde al cálculo de la posición por medio de las distancias a los faros. La segunda, el cálculo de la posición por medio de la diferencia de distancias a los faros, es un acercamiento diferente que simplifica algunos de los problemas prácticos que impone la realización del primer método.

3.1 Cálculo de la posición por medio de las distancias a los faros

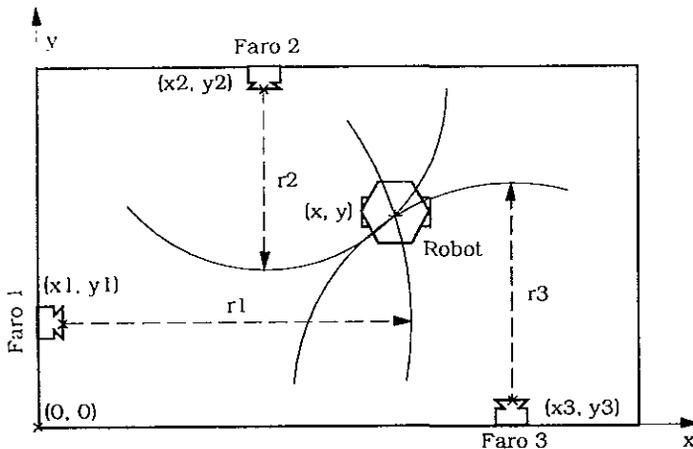


Figura 3.1 Robot móvil y faros dentro de un área determinada.⁹

Consideremos tres faros colocados en puntos conocidos en un área determinada dentro de la cual se mueve un robot cuya posición deseamos conocer (figura 3.1). La distancia entre el robot y cada uno de los faros representa el radio de una circunferencia cuyo centro coincide con el faro, y en cuyo perímetro se encuentra el robot. La posición del robot coincide con la intersección de dichas circunferencias

Podemos expresar a las circunferencias por medio de las ecuaciones

⁹ Figura adaptada de ref [1]

$$(x - x_1)^2 + (y - y_1)^2 = r_1^2 \quad (3.1)$$

$$(x - x_2)^2 + (y - y_2)^2 = r_2^2 \quad (3.2)$$

$$(x - x_3)^2 + (y - y_3)^2 = r_3^2 \quad (3.3)$$

donde x_i, y_i son las coordenadas del el faro i ; r_i es la distancia entre el robot y el faro i ; x y y son las coordenadas del robot que deseamos conocer. Desarrollando estas ecuaciones obtenemos las siguientes expresiones

$$x^2 - 2xx_1 + x_1^2 + y^2 - 2yy_1 + y_1^2 = r_1^2 \quad (3.4)$$

$$x^2 - 2xx_2 + x_2^2 + y^2 - 2yy_2 + y_2^2 = r_2^2 \quad (3.5)$$

$$x^2 - 2xx_3 + x_3^2 + y^2 - 2yy_3 + y_3^2 = r_3^2 \quad (3.6)$$

restamos la ecuación (3.5) a las ecuaciones (3.4) a (3.6) para obtener

$$2(x_2 - x_1)x + 2(y_2 - y_1)y + x_1^2 - x_2^2 + y_1^2 - y_2^2 = r_1^2 - r_2^2 \quad (3.7)$$

$$2(\lambda_2 - x_3)x + 2(y_2 - y_3)y + x_3^2 - x_2^2 + y_3^2 - y_2^2 = r_3^2 - r_2^2 \quad (3.8)$$

despejamos x en ambas e igualamos

$$\frac{y_2 - y_1}{x_2 - x_1} y + \frac{-x_1^2 + x_2^2 - y_1^2 + y_2^2 + r_1^2 - r_2^2}{2(x_2 - x_1)} = \frac{y_2 - y_3}{x_2 - x_3} y + \frac{-x_3^2 + x_2^2 - y_3^2 + y_2^2 + r_3^2 - r_2^2}{2(x_2 - x_3)} \quad (3.9)$$

de donde despejamos y

$$y = \frac{1}{2} \frac{(x_2 - \lambda_1)(-x_1^2 + x_2^2 - y_1^2 + y_2^2 + r_1^2 - r_2^2) - (x_2 - x_3)(-x_3^2 + x_2^2 - y_3^2 + y_2^2 + r_3^2 - r_2^2)}{(y_1 - y_2)(x_2 - x_3) - (y_2 - y_3)(x_2 - \lambda_1)} \quad (3.10)$$

de igual forma despejamos y de (3.7) y (3.8) e igualamos

$$\frac{x_2 - x_1}{y_2 - y_1} x + \frac{-x_1^2 + x_2^2 - y_1^2 + y_2^2 + r_1^2 - r_2^2}{2(y_2 - y_1)} = \frac{x_2 - x_3}{y_2 - y_3} x + \frac{-x_3^2 + x_2^2 - y_3^2 + y_2^2 + r_3^2 - r_2^2}{2(y_2 - y_3)} \quad (3.11)$$

para despejar x

$$x = \frac{1}{2} \frac{(y_2 - y_1)(-x_1^2 + x_2^2 - y_1^2 + y_2^2 + r_1^2 - r_2^2) - (y_2 - y_3)(-x_3^2 + x_2^2 - y_3^2 + y_2^2 + r_3^2 - r_2^2)}{(\lambda_1 - x_2)(y_2 - y_1) - (x_2 - x_3)(y_2 - y_1)} \quad (3.12)$$

Las expresiones (3.10) y (3.12) nos permiten conocer la posición del robot en función de las coordenadas de los faros y de las distancias entre el robot y éstos.

3.1.1 Aspectos Prácticos

El cálculo de la posición del robot a partir de las distancias a los faros presenta algunos retos y dificultades prácticas. Para determinar la distancia a cada faro necesitamos medir el tiempo que tarda la señal en viajar hasta el robot. Una vez medido el tiempo t y conocida la velocidad v de transmisión de las señales ($331 \frac{m}{s}$ @ $0^\circ C$ para el sonido en el aire), es fácil determinar la distancia d mediante la relación física elemental $d = vt$. Esto no es tan sencillo como parece, pues para medir el tiempo que tardan las señales en llegar hasta el robot es necesario saber con precisión en que instante fueron transmitidas. Esto implica que existe algún mecanismo de sincronización entre el robot y el circuito de control de los faros. El robot puede, por ejemplo, transmitir una señal de radio que le indique al circuito de control de los faros en qué momento debe de iniciar la transmisión de las señales.

Otro problema es el de discernir las señales provenientes de los faros. Al ser transmitidas simultáneamente las señales pueden llegar al robot en cualquier orden, inclusive simultáneamente, dependiendo de la posición de éste. Para poder distinguir sin ambigüedad cada una de las señales es necesario dotar a cada una de ellas con alguna característica única que la distinga de las demás, por ejemplo la frecuencia. Los filtros digitales juegan un papel determinante para poder identificar cada señal

3.1.2 Realización

En la figura 3.2 se muestra el diagrama de bloques de un sistema para determinar la posición del robot mediante las distancias a los faros. El circuito de control activa simultáneamente a los faros. La señal recogida por el micrófono en el robot es alimentada a una tarjeta en donde es muestreada. Las muestras de la señal se envían al DSP a través de la interfaz serie síncrona (SSI), las cuales son procesadas por tres filtros digitales IIR con el fin de diferenciar cada una de las señales provenientes de los faros. Se utilizaron filtros IIR realizados a partir de secciones bicuadráticas en cascada como las descritas en el capítulo anterior. En el apéndice se muestra un programa de MatLab para el cálculo de los coeficientes de los filtros dadas las frecuencias de la banda de paso y la frecuencia de muestreo. Una vez filtrada la señal de entrada nos encontramos con el problema de

determinar el arribo de las señales. La forma más sencilla de determinar la presencia de cada señal consiste en comparar la amplitud de las señales provenientes de los filtros con un umbral predeterminado. Se puede fijar un umbral, el cual al ser rebasado indica la presencia de dicha señal. También es posible fijar este umbral de manera dinámica en relación al valor máximo de un conjunto de muestras que contiene la señal

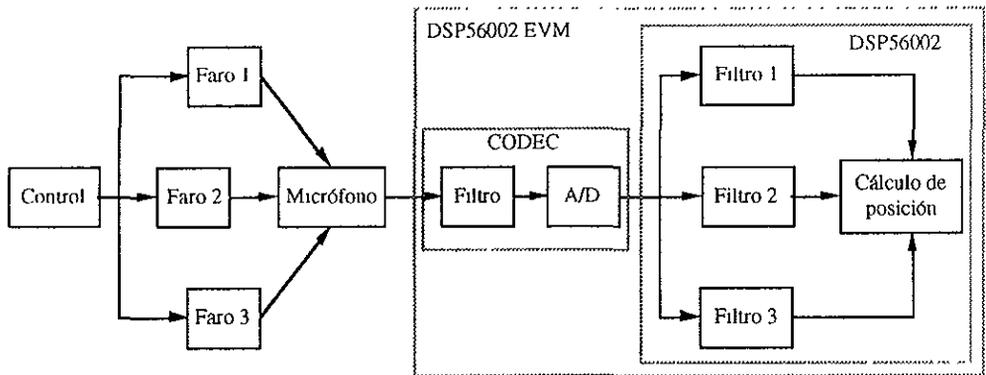


Figura 3.2 Diagrama de bloques de un sistema para calcular la posición por medio de las distancias a los faros.

Si se conoce la frecuencia de muestreo se conoce el tiempo entre cada muestra. Si se lleva un conteo del número de muestras procesadas desde que se envía la señal que activa los faros hasta el instante en que se detecta la presencia de cada señal, es posible obtener el tiempo que transcurre, al multiplicar el número de muestras por el período de muestreo (tiempo entre muestras). La distancia se obtiene al multiplicar la velocidad del sonido por el tiempo obtenido. Finalmente estas distancias se utilizan para determinar la posición del robot por medio de las expresiones (3.10) y (3.12).

A continuación se describen brevemente los elementos que se utilizaron en las pruebas iniciales de este sistema.

3.1.2.1 Tarjeta de evaluación DSP56002

Debido a su disponibilidad, se utilizó una tarjeta de evaluación del DSP56002 de Motorola la cual cuenta con 32 k de memoria de 24 bits que puede ser utilizada como un solo bloque de memoria, o puede ser dividido en dos áreas de 16 k para hacer uso de la

arquitectura Harvard doble del DSP. La tarjeta también cuenta con un *codec*¹⁰ que permite la entrada y salida de dos canales de audio de 16 bits, con una frecuencia de muestreo máxima de 48 kHz. Por supuesto, la tarjeta cuenta con un DSP56002 a 40 MHz. La tarjeta funciona con un cristal de cuarzo de 4 MHz, pero es posible seleccionar la frecuencia de operación del DSP mediante la programación del PLL interno que permite multiplicar esta frecuencia.

3.1.2.1.1 DSP56002

El DSP56002 es un miembro de la familia de procesadores digitales de señales de 24 bits de punto fijo HCMOS de baja potencia de Motorola. Está constituido por tres unidades de ejecución que operan de manera paralela: la unidad aritmética y lógica (ALU), la unidad generadora de direcciones y la unidad de control de programa (PCU). Su arquitectura es de tipo Harvard doble, ya que cuenta con dos buses independientes de datos y uno de instrucciones que permiten el acceso simultáneo a dos espacios de datos y uno de programa. Cuenta además con dos unidades generadoras de direcciones y dos acumuladores en la unidad aritmética y lógica. Los principales componentes del DSP56002 son (ref [5]):

- buses de datos
- buses de direcciones
- unidad aritmética y lógica (ALU)
- unidad generadora de direcciones (AGU)
- unidad de control de programa (PCU)
- espacio de memoria X
- espacio de memoria Y
- espacio de memoria de programa
- dispositivos de entrada y salida
- expansión de memoria (Puerto A)
 - puertos de entrada y salida de propósito general (Puertos B y C)
 - interfaz paralela (*Host Interface*, HI)
 - interfaz serie de comunicaciones (SCI)

¹⁰ Un *codec* (codificador decodificador) es mucho más que un convertidor A/D. Entre otras cosas tiene la capacidad de codificar la señal digitalizada en formatos no lineales como μ -Law y A-law

- interfaz serie síncrona (SSI)

En la figura 3.3 se muestra el diagrama de bloques del DSP56002.

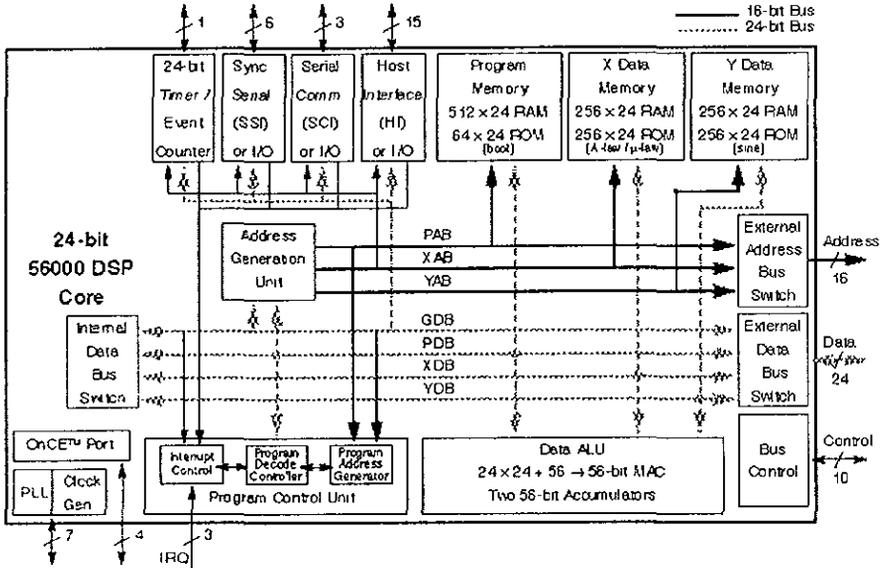


Figura 3.3 Diagrama de bloques del DSP56002 (ref. [5]).

En el modelo de programación del DSP56002 de la figura 3 4 se muestran los registros con los que cuenta este procesador digital de señales.

Esta realización fue desechada por la complejidad que significaba la sincronización del circuito de control con el robot, así como la programación del algoritmo y finalmente la conectividad con el robot. En la siguiente sección se describe un método alternativo que surgió a partir de la necesidad de simplificar el sistema, en el cual no se requiere de sincronización alguna y que además se puede llevar a cabo directamente en la computadora del robot

3.2 Cálculo de la posición por medio de la diferencia de las distancias a los faros

Otra forma para determinar el origen de cada señal consiste en enviar las señales de manera secuencial, una detrás de otra, con un retraso tal que permita siempre el arribo ordenado de las señales hasta el robot. Este último método de discernimiento de las señales nos sugiere otra forma de calcular la posición del robot, que además nos permite prescindir del mecanismo de sincronización del primer método. Este acercamiento es similar al anterior, pero se basa en la medición de las diferencias de distancias entre el robot y los faros en lugar de la medición de las distancias mismas¹¹. Lo que se hace es esperar a que llegue la primera señal hasta el robot; una vez detectada ésta se mide el tiempo que transcurre hasta detectar la segunda señal. De la misma manera se mide el tiempo que transcurre entre la llegada de la segunda señal y el arribo de la última. Es posible convertir estos tiempos a distancias, tomando en cuenta la velocidad a la que viajan las señales. Estas diferencias se pueden expresar como

$$\Delta r_{12} = r_2 - r_1 \tag{3.13}$$

$$\Delta r_{23} = r_3 - r_2 \tag{3.14}$$

Si escribimos las ecuaciones (3.1), (3.2) y (3.3) en función de r_2 , Δr_{12} y Δr_{23} obtenemos

$$(x - x_1)^2 + (y - y_1)^2 = (r_2 - \Delta r_{12})^2 \tag{3.15}$$

$$(x - x_2)^2 + (y - y_2)^2 = r_2^2 \tag{3.16}$$

$$(x - x_3)^2 + (y - y_3)^2 = (r_2 + \Delta r_{23})^2 \tag{3.17}$$

que podemos desarrollar para obtener

¹¹ Esta idea surgió a partir de un trabajo en la que se propone un método para determinar el origen de una vibración sísmica, ref. [6].

$$x^2 - 2x_1x + \lambda_1^2 + y^2 - 2y_1y + y_1^2 = r_1^2 - 2\Delta r_{12}r_2 + \Delta r_{12}^2 \quad (3.18)$$

$$x^2 - 2x_2x + \lambda_2^2 + y^2 - 2y_2y + y_2^2 = r_2^2 \quad (3.19)$$

$$x^2 - 2x_3x + \lambda_3^2 + y^2 - 2y_3y + y_3^2 = r_3^2 + 2\Delta r_{23}r_2 + \Delta r_{23}^2 \quad (3.20)$$

si restamos (3.19) a (3.18) y (3.20) tenemos

$$2(x_2 - \lambda_1)x + 2(y_2 - y_1)y + x_1^2 - x_2^2 + y_1^2 - y_2^2 = -2\Delta r_{12}r_2 + \Delta r_{12}^2 \quad (3.21)$$

$$2(x_2 - x_3)x + 2(y_2 - y_3)y + x_3^2 - x_2^2 + y_3^2 - y_2^2 = 2\Delta r_{23}r_2 + \Delta r_{23}^2 \quad (3.22)$$

despejamos x de ambas ecuaciones e igualamos

$$-\frac{y_2 - y_1}{\lambda_2 - \lambda_1}y - \frac{x_1^2 - \lambda_2^2 + y_1^2 - y_2^2 - \Delta r_{12}^2}{2(x_2 - \lambda_1)} - \frac{\Delta r_{12}r_2}{x_2 - x_1} = -\frac{y_2 - y_3}{x_2 - \lambda_1}y - \frac{\lambda_3^2 - \lambda_2^2 + y_3^2 - y_2^2 - \Delta r_{23}^2}{2(x_2 - x_3)} + \frac{\Delta r_{23}r_2}{x_2 - x_3} \quad (3.23)$$

agrupamos términos

$$\left(\frac{\lambda_2 - \lambda_1}{\lambda_2 - \lambda_1} - \frac{y_2 - y_1}{\lambda_2 - \lambda_1} \right) y = \frac{x_1^2 - \lambda_2^2 + y_1^2 - y_2^2 - \Delta r_{12}^2}{2(x_2 - \lambda_1)} - \frac{\lambda_3^2 - \lambda_2^2 + y_3^2 - y_2^2 - \Delta r_{23}^2}{2(x_2 - x_3)} + \left(\frac{\Delta r_{12}}{x_2 - \lambda_1} + \frac{\Delta r_{23}}{x_2 - x_3} \right) r_2 \quad (3.24)$$

definimos

$$a = \frac{x_1^2 - \lambda_2^2 + y_1^2 - y_2^2 - \Delta r_{12}^2}{2} \quad (3.25)$$

y

$$b = \frac{x_3^2 - \lambda_2^2 + y_3^2 - y_2^2 - \Delta r_{23}^2}{2} \quad (3.26)$$

factorizamos

$$\frac{(y_2 - y_1)(x_2 - \lambda_1) - (y_2 - y_1)(\lambda_2 - \lambda_1)}{(\lambda_2 - \lambda_1)(x_2 - \lambda_1)} y = \frac{a(x_2 - \lambda_1) - b(x_2 - \lambda_1)}{(\lambda_2 - \lambda_1)(x_2 - \lambda_1)} + \frac{\Delta r_{12}(\lambda_2 - \lambda_1) + \Delta r_{23}(x_2 - \lambda_1)}{(\lambda_2 - \lambda_1)(x_2 - \lambda_1)} r_2 \quad (3.27)$$

ahora definimos

$$A = \frac{\Delta r_{12}(x_2 - x_3) + \Delta r_{23}(x_1 - x_3)}{(y_2 - y_1)(x_1 - x_3) - (y_1 - y_3)(x_2 - x_3)} \quad (3.28)$$

y

$$B = \frac{a(x_2 - x_3) - b(x_1 - x_3)}{(y_2 - y_1)(x_2 - x_3) - (y_1 - y_3)(x_1 - x_3)} \quad (3.29)$$

y despejamos y obtenemos

$$y = Ar_1 + B \quad (3.30)$$

De la misma forma que se hizo para x , ahora despejamos y de (3.21) y (3.22) e igualamos:

$$-\frac{x_1 - x_3}{y_1 - y_3} x - \frac{x_1^2 - x_2^2 + y_1^2 - y_2^2 - \Delta r_{12}^2}{2(y_2 - y_1)} - \frac{\Delta r_{12}r_2}{y_2 - y_1} = -\frac{x_2 - x_3}{y_2 - y_3} x - \frac{x_2^2 - x_3^2 + y_2^2 - y_3^2 - \Delta r_{23}^2}{2(y_2 - y_3)} + \frac{\Delta r_{23}r_2}{y_2 - y_3} \quad (3.31)$$

$$\left(\frac{x_1 - x_3}{y_1 - y_3} - \frac{x_2 - x_3}{y_2 - y_3} \right) x = \frac{x_1^2 - x_2^2 + y_1^2 - y_2^2 - \Delta r_{12}^2}{2(y_2 - y_1)} - \frac{x_2^2 - x_3^2 + y_2^2 - y_3^2 - \Delta r_{23}^2}{2(y_2 - y_3)} + \frac{\Delta r_{12}r_2}{y_2 - y_1} - \frac{\Delta r_{23}r_2}{y_2 - y_3} \quad (3.32)$$

si factorizamos obtenemos

$$\frac{(x_1 - x_3)(y_2 - y_1) - (x_2 - x_3)(y_1 - y_3)}{(y_1 - y_3)(y_2 - y_1)} x = \frac{a(y_2 - y_3) - b(y_1 - y_3)}{(y_2 - y_1)(y_2 - y_3)} + \frac{\Delta r_{12}(y_2 - y_3) + \Delta r_{23}(y_1 - y_3)}{(y_2 - y_1)(y_2 - y_3)} r_2 \quad (3.33)$$

ahora definimos

$$C = \frac{\Delta r_{12}(y_2 - y_3) - \Delta r_{23}(y_1 - y_3)}{(x_1 - x_3)(y_2 - y_1) - (x_2 - x_3)(y_1 - y_3)} \quad (3.34)$$

y

$$D = \frac{a(y_2 - y_3) - b(y_1 - y_3)}{(x_1 - x_3)(y_2 - y_1) - (x_2 - x_3)(y_1 - y_3)} \quad (3.35)$$

y despejamos x obtenemos

$$x = Cr_2 + D \quad (3.36)$$

Si sustituimos las expresiones encontradas para x y para y en la ecuación (3.2) y desarrollamos tenemos

$$(Cr_2 + D - \lambda_1)^2 + (Ar_2 + B - y_1)^2 = r_1^2 \quad (3.37)$$

$$C^2 r_2^2 + 2C(D - \lambda_1)r_2 + (D - \lambda_1)^2 + A^2 r_2^2 + 2A(B - y_1)r_2 + (B - y_1)^2 - r_2^2 = 0 \quad (3.38)$$

$$(C^2 + A^2 - 1)r_2^2 + 2(C(D - \lambda_1) + A(B - y_1))r_2 + (D - \lambda_1)^2 + (B - y_1)^2 = 0 \quad (3.39)$$

Sustituimos la solución válida de r_2 (positiva) de esta ecuación cuadrática en (3.30) y (3.36) para obtener finalmente la posición del robot.

3.2.1 Aspectos prácticos

El cálculo de la posición por medio de la diferencia de las distancias a los faros requiere de ciertas consideraciones prácticas. La selección del tiempo de retraso entre las señales se debe efectuar tomando en cuenta las dimensiones máximas del ambiente de trabajo para garantizar que las señales siempre lleguen en orden, independientemente de la ubicación del robot. Para lograr esto es necesario que el tiempo de retraso sea mayor al máximo tiempo que le toma a las señales en recorrer la distancia entre el faro y el robot para el peor caso, i.e. para el punto en el cual la distancia entre el robot y alguno de los faros es máxima. Si consideramos un cuarto rectangular (figura 3.5), la máxima distancia que puede existir entre alguno de los faros y el robot en cualquier instante esta dado por

$$d_{\max} = \sqrt{a^2 + b^2} \quad (3.40)$$

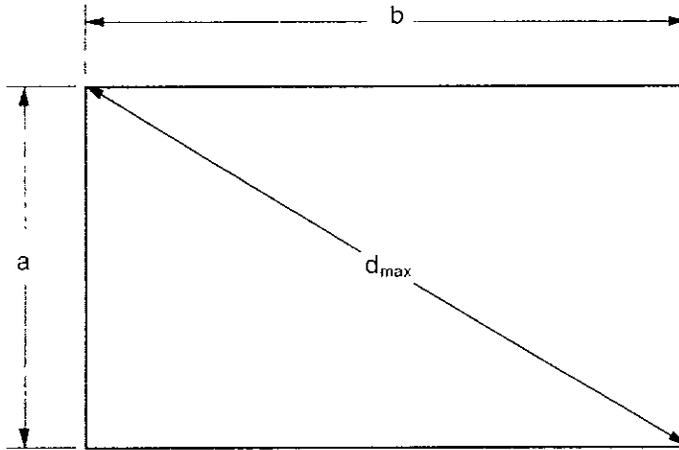


Figura 3.5 Distancia máxima entre un faro y el robot en un área rectangular.

por lo tanto el retraso t , debe ser tal que

$$t > \frac{d_{\max}}{V_s} \quad (3.41)$$

donde V_s es la velocidad del sonido.

Este método permite utilizar la misma frecuencia para todas las señales, por lo que el uso de los filtros digitales no es crítico.

Para poder llevar a cabo los cálculos de posición descritos anteriormente necesitamos calcular el tiempo entre cada una de las señales. La primera dificultad a la que nos enfrentamos es la detección de cada juego y a su vez de cada señal. Dado que la naturaleza del sistema garantiza el arribo ordenado de las señales no es indispensable dotar a cada señal de una frecuencia única

3.2.2 Realización

Dado a que sabemos que las señales llegarán de manera ordenada y nunca de manera simultánea podemos utilizar la misma frecuencia para todas las señales. Por la misma razón podemos prescindir de los filtros digitales lo que reduce las exigencias de procesamiento para llevar a cabo la obtención de la posición del robot. Por esta razón se decidió llevar a cabo el sistema directamente en una computadora, de características similares con la que cuenta el robot, con el fin de integrarlo directamente en la

computadora de éste. Esto simplifica la integración del sistema al robot ya que no es necesario agregar ningún elemento adicional ni es necesario preocuparse por integrar ambos sistemas. La desventaja de este acercamiento es que el sistema está ligado intimamente al robot, lo que lo hace imposible de implantar en un robot de características diferentes.

El sistema está compuesto básicamente de los mismos elementos descritos anteriormente. La única diferencia es que ahora el procesamiento se lleva a cabo en una computadora (figura 3.6). La conversión A/D se lleva a cabo mediante una tarjeta de sonido, la cual incorpora su propio preamplificador y filtro.

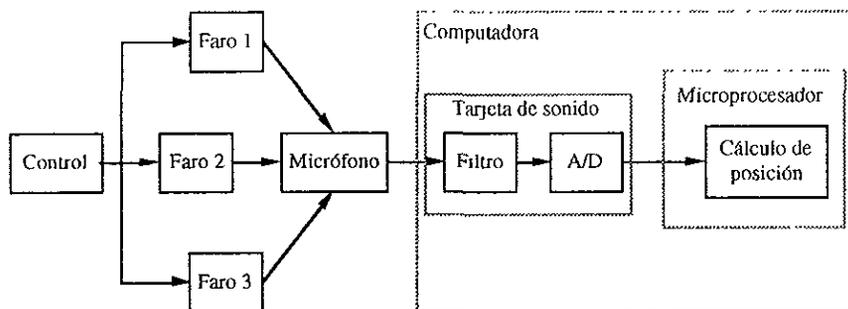


Figura 3.6 Realización mediante una computadora.

La señal captada por el micrófono se alimenta a la computadora a través de la tarjeta de sonido, la cual se encarga de la conversión A/D. Para analizar la señal, obtenemos un número suficiente de muestras de tal forma que garantice la existencia de un juego completo de señales. Obtenemos el valor absoluto de estas muestras, e identificamos cada señal de la siguiente manera: barre todas las muestras y determinamos los puntos máximos que presenten una separación mínima que garantice que no se trata de la misma señal. El número de muestras entre los máximos obtenidos nos indican tanto el orden de las señales como el tiempo entre el arribo de cada una, lo que utilizamos para finalmente calcular la posición del robot.

El proceso mencionado se lleva a cabo por la rutina `getpos`, que se encuentra listada al final del documento, dentro del archivo `lps.h`. A parte de esta rutina se cuenta con otras dos que permiten establecer y recuperar de manera programática la posición de los faros, ya que esta información es fundamental para el cálculo de la posición. Estas rutinas son `getBeaconPosition` y `setBeaconPosition`, que se encuentran dentro del mismo archivo. Cualquier programa que desee hacer uso del sistema debe incluir el archivo de cabecera `lps.h`, que contiene los prototipos de las rutinas mencionadas. El archivo `main.c` contiene el listado de un programa de pruebas que hace uso del sistema, y dibuja la posición del robot en la pantalla. El archivo `lpsdevel.h` contiene los prototipos de las demás rutinas de

uso interno del sistema y que no son accesibles para los programas que deseen hacer uso del mismo.

3.2.2.1 Circuito de control

El circuito de control se encarga de generar las señales intermitentes que llegan hasta los faros (figura 3.8). La precisión del sistema depende en gran medida de este circuito. Por esta razón se decidió utilizar un oscilador de cuarzo integrado. Mediante un par de contadores se divide la frecuencia de la señal proveniente del oscilador hasta una frecuencia útil según el criterio mencionado anteriormente. Esta señal dispara a un circuito monoestable el cual se encarga de generar pulsos de una duración predeterminada t_i . Los pulsos provenientes de dicho circuito son alimentados a un contador a través de un inversor. El contador controla a su vez a un demultiplexor que se encarga de distribuir los pulsos provenientes del circuito monoestable a tres líneas. Cada una de estas líneas controla a su vez un interruptor analógico de estado sólido que permite o inhibe el paso de la señal de un oscilador a los faros.

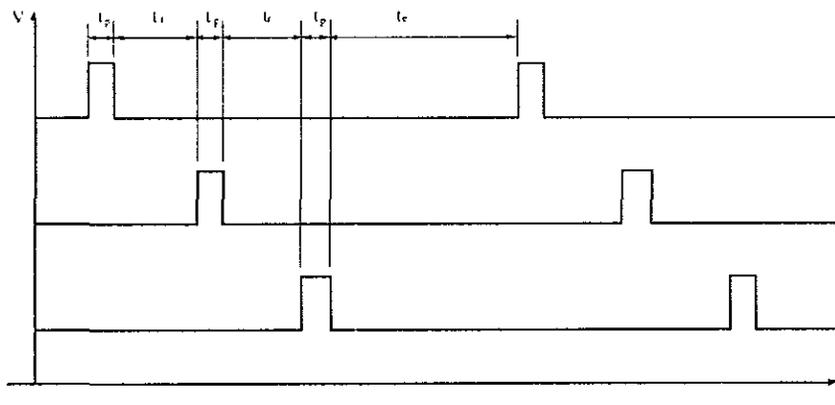
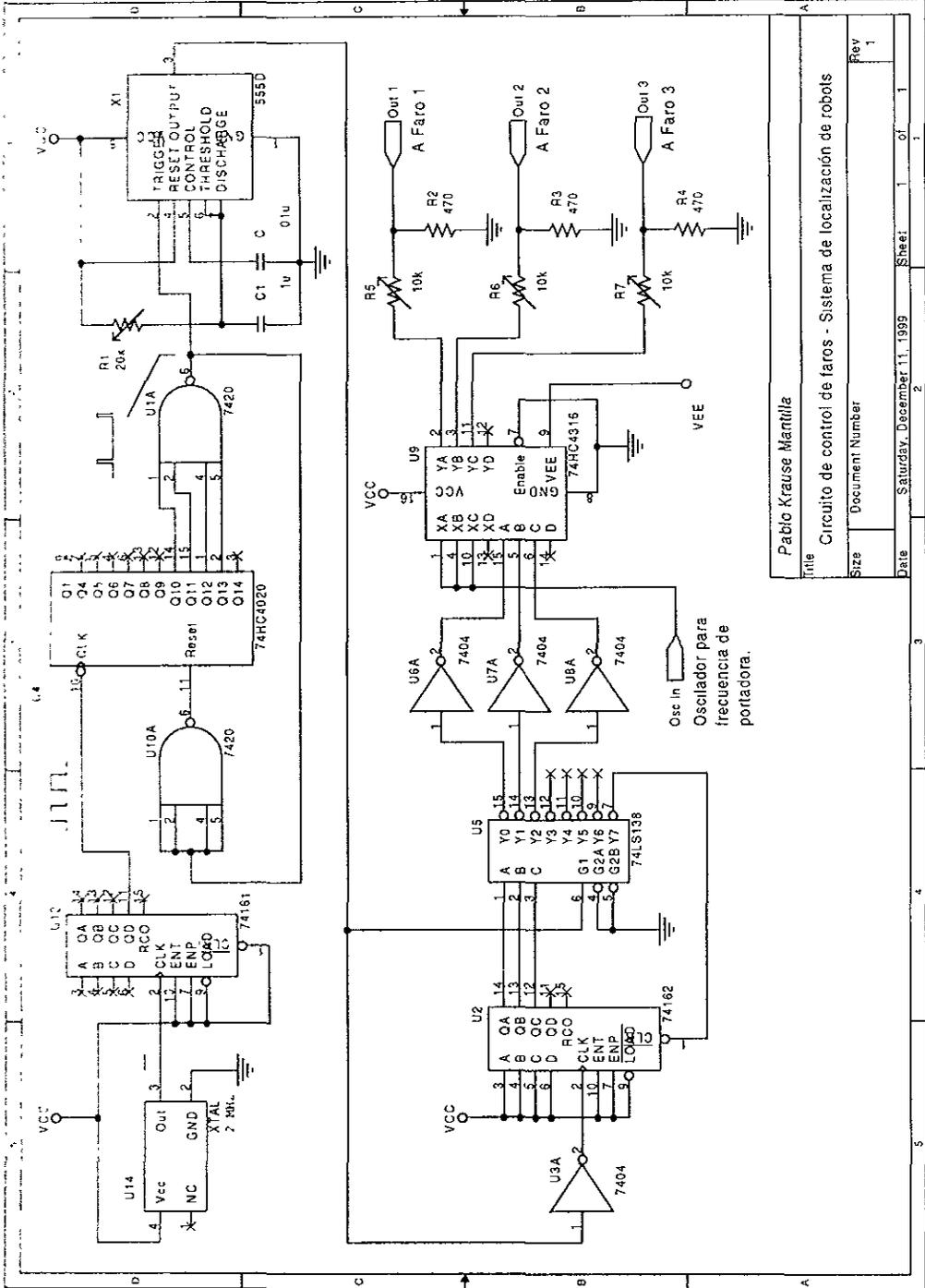


Figura 3.7 Señales de control.

En la figura 3.7 se muestra el diagrama de las señales que controlan a los interruptores analógicos de estado sólido. t_p es el tiempo que se activa cada faro para la emisión de las señales. t_i es el tiempo de retraso entre la activación de cada faro. t_e es el tiempo entre cada juego de señales. t_e debe ser al menos el doble de t_i con el fin de poder diferenciar cada juego de señales.

Cada una de las señales provenientes del demultiplexor es alimentada a un amplificador de potencia para poder ser emitidas por las bocinas. Afortunadamente,

existen en el mercado gran variedad de amplificadores de potencia en circuitos integrados que facilitan enormemente la elaboración de un circuito de amplificación. Se seleccionó el LM1875, un amplificador de potencia integrado con capacidad de entregar 20W de potencia con muy buenas características de distorsión, estabilidad y desempeño. En el apéndice se encuentra la hoja de especificaciones de este amplificador, donde podemos ver un diagrama de su estructura interna. Este amplificador está construido por un circuito bastante complejo, en donde podemos apreciar en la etapa de salida una configuración push-pull como la descrita en el capítulo anterior. En esas mismas hojas se encontrará el diagrama del circuito impreso que se utilizó para la construcción de los amplificadores de las bocinas.



Pablo Krause Mantilla	
Circuito de control de faros - Sistema de localización de robots	
Size	Document Number
Date	Revision
Saturday, December 11, 1999	1 of 1
Sheet	1

Figura 3.8 Circuito de control.

4. Resultados

En este capítulo se presentan los resultados de algunos experimentos que resultan de interés para evaluar el desempeño del sistema desarrollado

Uno de las características más importantes que se requieren del micrófono es su capacidad para captar señales independientemente de la dirección de la que provengan. Para evaluar el desempeño de dicho micrófono se realizó un experimento en el que se colocó una bocina a una distancia fija de 1 m del micrófono y se midió la amplitud promedio de una señal senoidal de 10 kHz para diferentes ángulos de incidencia. El resultado fue bastante satisfactorio como se muestra en la figura 4.1.

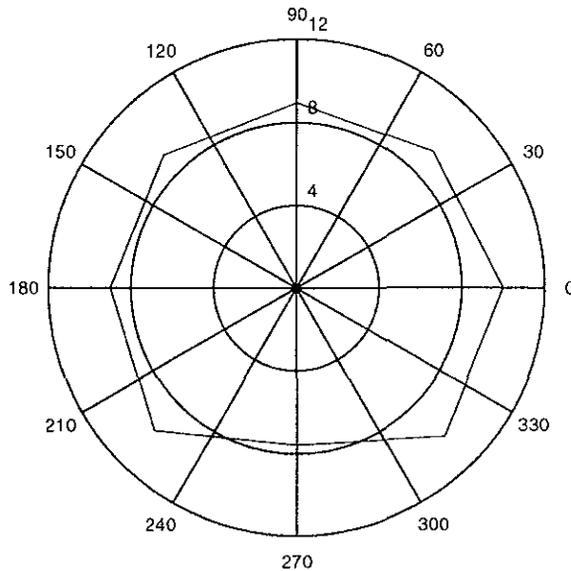


Figura 4.1 Respuesta direccional del micrófono.

En la figura 4.2 se presenta la respuesta de las bocinas utilizadas como faros para diferentes frecuencias. Podemos observar que la respuesta más uniforme se logra para una frecuencia de alrededor de 5 kHz. Podemos considerar que la detección es aceptable hasta unos 50° con respecto al centro de simetría, donde la magnitud de la señal se reduce a la

mitad de la máxima. A pesar de que la detección es posible aún a más del ángulo mencionado, si consideramos de manera conservadora un ángulo de 45° tenemos una cobertura efectiva de 90° , con lo que es posible cubrir un área rectangular o cuadrada como la que se muestra en la figura 4.3. Las líneas segmentadas indican el ángulo máximo de influencia de cada faro, y el área sombreada representa el área dentro de la cual es posible determinar la posición del robot. Esta área es la intersección del área cubierta por cada faro

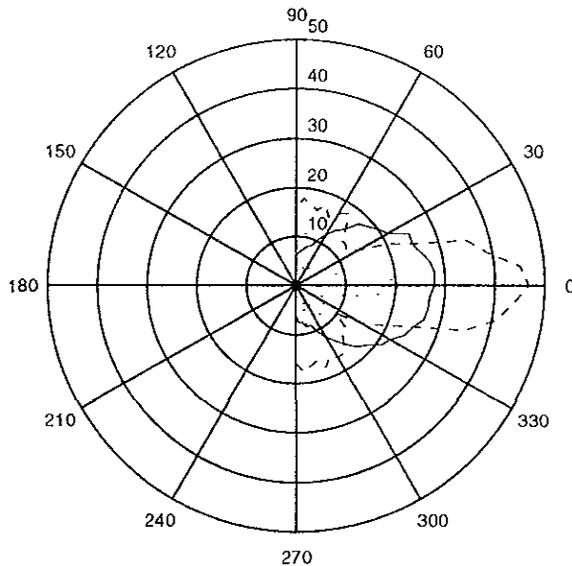


Figura 4.2 Respuesta direccional de las bocinas a diferentes frecuencias.
 — 1kHz - - - 5kHz - · - · 10kHz

La idea original era utilizar señales ultrasónicas, sin embargo la mala transmisión de señales acústicas en el aire a esas frecuencias, aunado a la dificultad para conseguir tanto sensores omnidireccionales como emisores con buenas características de dispersión angular hicieron inconveniente su uso. Otro inconveniente que presenta el uso de las señales ultrasónicas es la necesidad de contar con un convertidor A/D con capacidad de muestrear a una mayor frecuencia, para satisfacer el criterio de Nyquist.

Los resultados que se presentan a continuación se obtuvieron recolectando 50 muestras válidas en cada punto de prueba. Nos referimos a muestras válidas a aquellas

que corresponden a una coordenada dentro del área de prueba. Se descartan todas las muestras que no cumplan con este criterio, o que generen un error (un número complejo) al calcular la posición. La frecuencia con que se presentan muestras válidas se presenta en una tabla. Cada casilla de esta tabla corresponde a un punto de prueba según aparecen en las gráficas

Para este experimento se utilizaron tres configuraciones del ambiente con el fin de determinar la importancia de la posición de los faros. En la configuración 1 los faros se encuentran colocados en tres de los vértices de un cuadrado. La configuración 2 consiste en colocar dos de los faros en los vértices de un lado y el tercer faro a la mitad del lado opuesto, formando un triángulo isósceles. En la configuración 3, dos de los faros se colocan uno junto al otro a la mitad de uno de los lados del cuadrado y el tercero a la mitad del lado opuesto.

Los puntos de prueba están marcados con una x en las figuras correspondientes. Las muestras obtenidas están marcadas con puntos. El promedio de cada conjunto de muestras está indicado con un +.

Se presentan los siguientes indicadores

Error medio cuadrático total:

$$rme = \sqrt{\frac{\sum_{i=1}^N d^2}{N-1}}$$

donde d es el error (distancia) entre cada posición calculada y el punto real.

Error promedio

$$me = \frac{\sum_{i=1}^N d}{N}$$

Error máximo

$$em = \max(d)$$

Desviación estándar

$$s = \sqrt{\frac{\sum_{i=1}^N (d - \bar{d})^2}{N-1}}$$

Para tratar de reducir el error, se calcula el promedio de las desviaciones en x y en y de las muestras con respecto al punto real. Posteriormente restamos estas desviaciones a cada una de las muestras y se calculan nuevamente los errores

4.1 Configuración 1

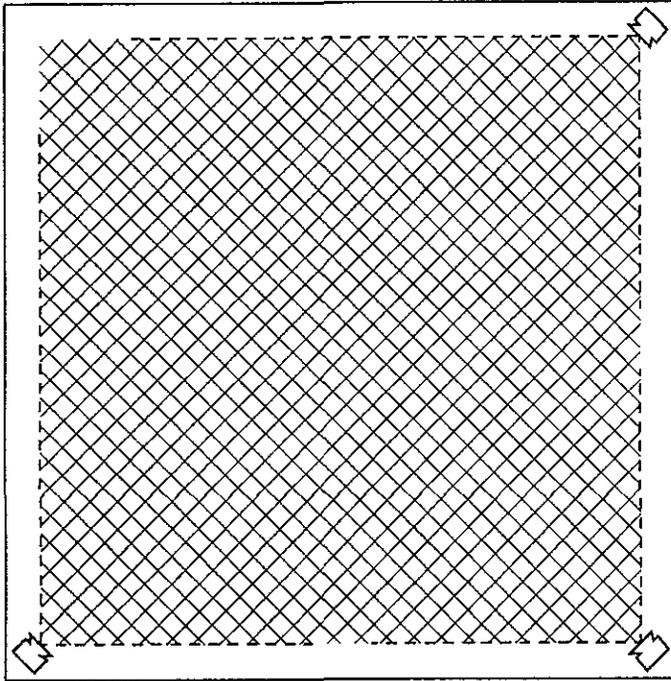


Figura 4.3 Disposición de los faros para la configuración 1.

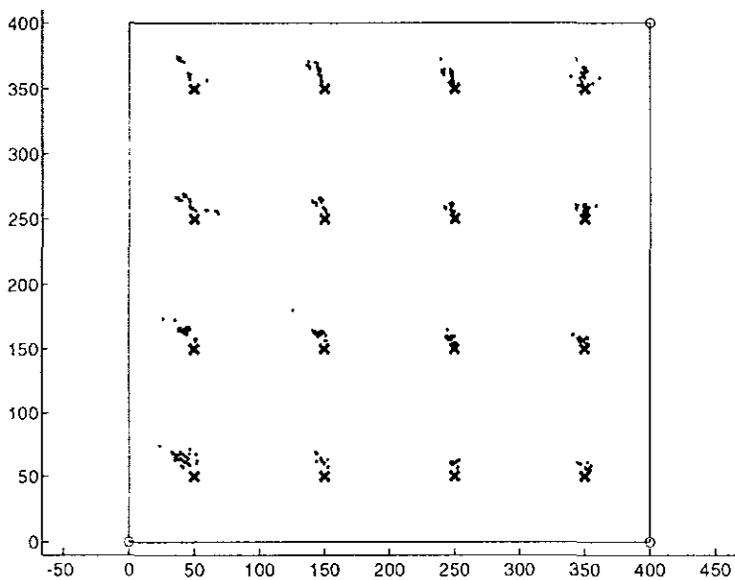


Figura 4.4 Posición obtenida para diferentes puntos para la configuración 1.

Error cuadrático medio total = 12.626 [cm]

Error promedio = 11.436 [cm]

Error máximo = 32.398 [cm]

Desviación estándar de la posición = 5.335 [cm]

Error sistemático en $x = -3.0$ [cm]

Error sistemático en $y = 10.419$ [cm]

Ajustamos las muestras con estos valores y obtenemos:

Error cuadrático medio total = 6.459 [cm]

Error promedio = 5.370 [cm]

Error máximo = 21.894 [cm]

Desviación estándar de la posición = 3.583 [cm]

Podemos ver que corrigiendo las muestras el error cuadrático medio total, y el error promedio se reducen casi en un 50%, y el error máximo en aproximadamente 30%.

A continuación se muestra una tabla con el número de muestras que resultaron en posiciones inválidas. Cada casilla corresponde a un punto de prueba según aparecen en las gráficas

73.53%	98.04%	98.04%	60.98%
96.15%	100.00%	100.00%	100.00%
100.00%	100.00%	98.04%	100.00%
81.97%	100.00%	100.00%	37.59%

4.2 Configuración 2

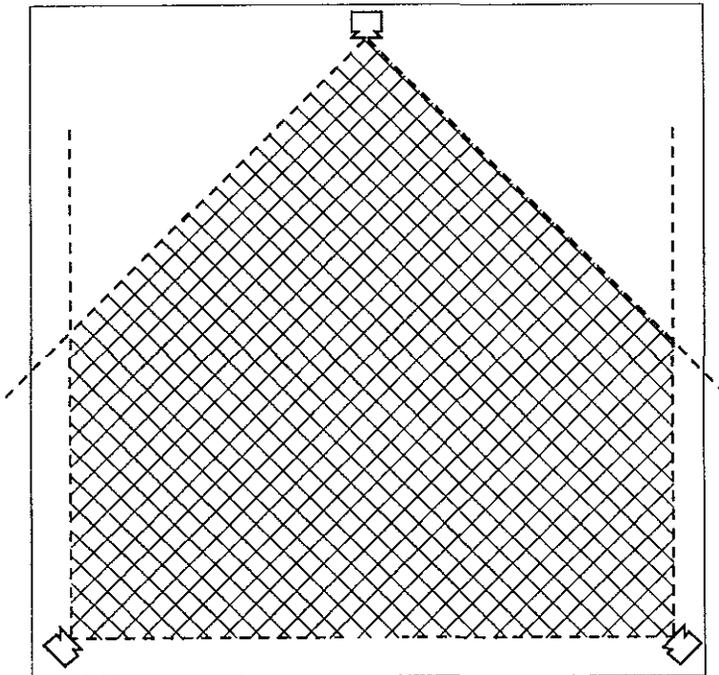


Figura 4.5 Disposición de los faros para la configuración 2.

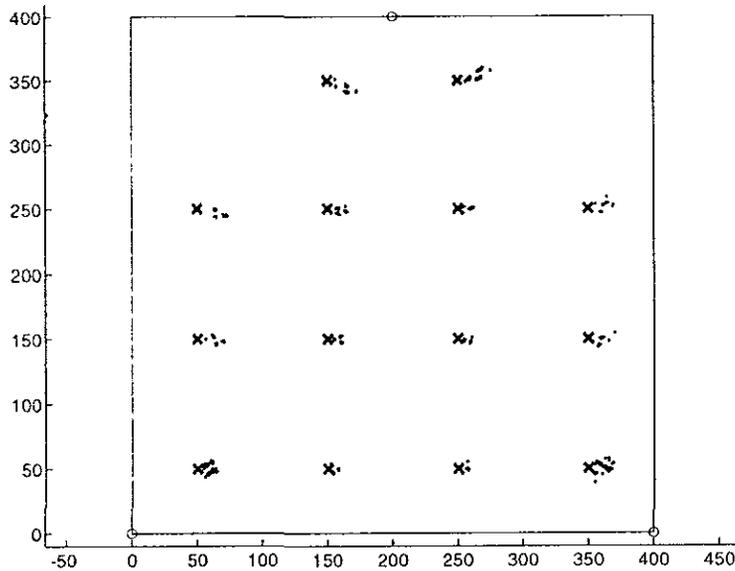


Figura 4.6 Posición obtenida para diferentes puntos para la configuración 2.

Error cuadrático medio total = 20.869 [cm]

Error promedio = 13.809 [cm]

Error máximo = 252.237 [cm]¹²

Desviación estándar de la posición = 15.638 [cm]

Error sistemático en x = -2.956 [cm]

Error sistemático en y = 12.314 [cm]

Ajustamos las muestras con estos valores y obtenemos que.

Error cuadrático medio total = 16.580 [cm]

Error promedio = 5.205 [cm]

Error máximo = 249.678 [cm]

¹² Este error aparentemente grande sólo se manifestó en una muestra. Desechando ésta, el error máximo sería de 29 [cm]

Desviación estándar de la posición = 15.741 [cm]

100.00%	100.00%	87.72%	94.34%
100.00%	100.00%	100.00%	100.00%
98.04%	100.00%	98.04%	98.04%
78.13%	98.04%	100.00%	44.64%

4.3 Configuración 3

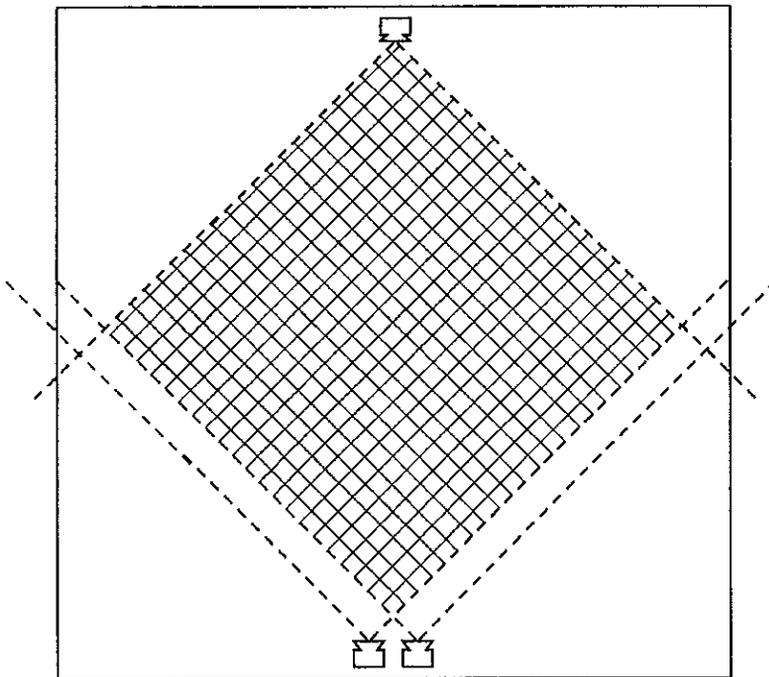


Figura 4.7 Disposición de los faros para la configuración 3.

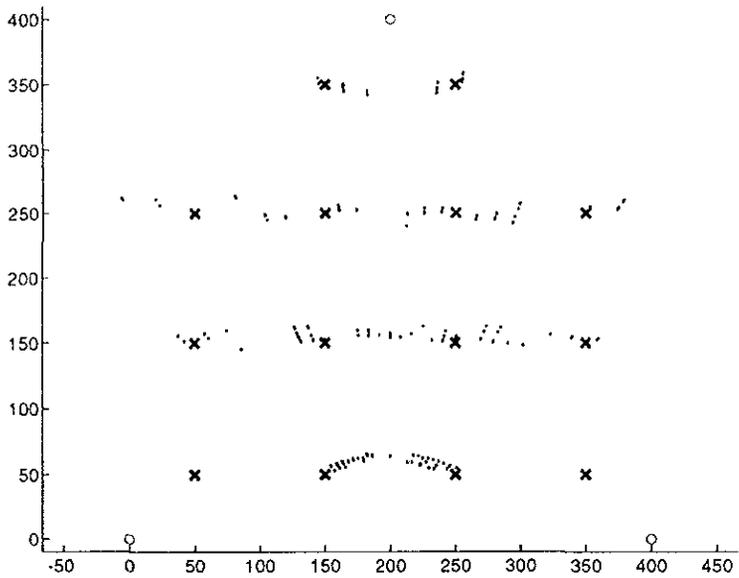


Figura 4.8 Posición obtenida para diferentes puntos para la configuración 3.

Error cuadrático medio total = 45.002 [cm]

Error promedio = 34.227 [cm]

Error máximo = 186.626 [cm]

Desviación estándar de la posición = 27.380 [cm]

Error sistemático en x = 0.3835 [cm]

Error sistemático en y = 4.934 [cm]

Ajustamos las muestras con estos valores y obtenemos que:

Error cuadrático medio total = 44.729 [cm]

Error promedio = 33.822 [cm]

Error máximo = 184.257 [cm]

Desviación estándar de la posición = 24.845 [cm]

0%	78.3%	76.94%	0%
57.85%	88.92%	87.33%	61.21%
67.32%	87.32%	85.76%	65.65%
0%	81.20%	80.56%	0%

5. Conclusiones

La limitación más evidente del sistema es que cualquier obstáculo entre el faro y el robot evita que la señal acústica llegue hasta el micrófono, impidiendo así el funcionamiento del sistema. Sin embargo es importante resaltar que no es necesario que el sistema esté disponible en todo momento, ya que el robot cuenta con sus propios medios para determinar su posición de manera relativa. Cada vez que el robot se encuentre en un área donde el sistema funcione puede actualizar su posición real y volver a utilizar sus propios mecanismos de posición.

Si se desea abarcar un área muy amplia, o un área irregular o disconexa es posible utilizar varios juegos de faros con diferentes frecuencias para cubrir áreas mayores, para lo cual sólo se requieren modificaciones menores al programa

A pesar de haber utilizado inicialmente señales ultrasónicas por ser estas inaudibles por el ser humano, la generación de ruido fue inevitable debido al encendido y apagado constante de los faros. Es así que se decidió utilizar señales de baja frecuencia, ya que estas presentan mejor transmisión en el aire y permiten utilizar sistemas de muestreo menos exigentes y no incrementan de manera considerable el ruido.

Por la naturaleza acústica del sistema, éste no es apropiado para ambientes con presencia seres humanos ya que la exposición al ruido que el sistema produce es muy molesto

La posición de los faros en el ambiente de trabajo es importante, ya que esto determina tanto el área útil del sistema así como la precisión de éste

6. Referencias

- 1 Savage, J. Lepe-Casillas, F., Minami, Y., Rivera, C., Psenicka, B., *Robot Localization Using Ultrasonic Beacons*, Proceedings of the 8th International Conference on Signal Processing, Applications and Technology, 1997.
- 2 Savant, C J., Roden, M.S , Carpenter, G.L , *Electronic Design, Circuits and Systems*, Addison Wesley, 1991.
- 3 Boylestad, R L , Nashelsky, L., *Electronic Devices and Circuit Theory*, Prentice Hall, 1996.
- 4 Franco, Sergio. *Operational amplifiers and analog integrated circuits*, McGraw-Hill International Editions, 1988.
5. *DSP56000/DSP56001 Digital Signal Processor User's Manual*, Motorola, 1990.
- 6 . Vlastimir D., Pavlovic Zoran S. Velickovic, *Localization of Unknown Source of Seismic Vibrations*, Proceedings of the 8th International Conference on Signal Processing, Applications and Technology, 1997.
- 7 Borenstein, J , Everett, H.R., Feng, L, *Navigating Mobile Robots*, A K Peters, Ltd., 1996.
- 8 Berger, M.O. Kubitz, O., *Application of Wireless Communication in Robotics*, Informatik und Technik, Forschungsbeiträge des Aachener Graduiertenkollegs 1996, pp 25-66, 1996. (<http://www-14.informatik.rwth-aachen.de/>)
9. Desoer, Kuh, *Basic Circuit Theory*, McGraw-Hill., 1969.
10. *High-Speed CMOS Data*, Motorola, 1988.
11. *Implementing IIR/FIR Filters with Motorola's DSP56000/DSP56001*, Motorola, 1993.
12. Ludeman, Lonnie C., *Fundamentals of digital signal processing*, Harper & Row, Publishers, Inc., 1986.
- 13 Matthew, Neil, Stones, Richard, *Beginning Linux Programming*, Wrox Press Ltd, 1997.

14 *TTL Logic DataBook*, Texas Instruments, 1988.

7. Apéndices

7.1 Listado de programas

Archivo `lps.h`

Este archivo contiene los prototipos de las funciones que expone el sistema de localización por diferencias de distancias, así como una estructura de datos básica que representa un punto en un plano coordenado. Este archivo debe de ser incluido por el programa que desee hacer uso de este sistema.

```
#ifndef LPS_H
#define LPS_H

typedef struct {
    float x;
    float y;
} tpoint;

void initlps(void);
int getlps(float *x, float *y);
struct getBeaconPosition(int beacon);
struct setBeaconPosition(int beacon, float x, float y);

#endif
```

Archivo `lps.c`

Este archivo contiene las rutinas del sistema de localización por medio de la diferencia de distancias a los faros

```
#include <math.h>
#include <sys/scandcard.h>
#include <stdio.h>
#include <unistd.h>
#include <sys/stat.h>
#include <fcntl.h>
#include <math.h>
#include <vga.h>
#include "lpsdevel.h"

#define BUFSIZE 35000
#define WINSIZE 3
```

```

static double k1, k2, k3, k4, A, B, C, D;
static float dx23, dx21, dy23, dy21;
static float fs=22000; /* fs = Frecuencia de muestreo */
static float vs=34000; /* vs = velocidad del sonido (cm/s) */
static int in_fd;
static unsigned char buf[BUFSIZE];
static int bufsize, n = 0, n_1 = 0, n_2 = 0;
static coord bcns[3].

void Update() {
    /* Inicializamos constantes para hacer mas eficientes los calculos */
    k1 = (bcns[0].x*bcns[0].x - bcns[1].x*bcns[1].x + bcns[0].y*bcns[0].y -
bcns[1].y*bcns[1].y)/2;
    k2 = (bcns[2].x*bcns[2].x - bcns[1].x*bcns[1].x + bcns[2].y*bcns[2].y -
bcns[1].y*bcns[1].y)/2;
    k3 = (bcns[1].y-bcns[2].y)*(bcns[1].x-bcns[0].x) - (bcns[1].y-bcns[0].y)*(bcns[1].x-
bcns[2].x);
    k4 = (bcns[1].x-bcns[2].x)*(bcns[1].y-bcns[0].y) - (bcns[1].x-bcns[0].x)*(bcns[1].y-
bcns[2].y);
    dx23 = bcns[1].x - bcns[2].x;
    dx21 = bcns[1].x - bcns[0].x;
    dy23 = bcns[1].y - bcns[2].y;
    dy21 = bcns[1].y - bcns[0].y;
}

/*
 * Dadas las diferencias entre las distancias del robot al faro,
 * esta función regresa la posición del robot.
 *
 * Regresa -1 si no se encuentra una solución válida
 */
int pbs(float *x, float *y) {
    float root1, root2, dr12, dr23;
    double a, b, c, temp;

    dr12 = *x;
    dr23 = *y;

    A = (dr12*(dx23) + dr23*(dx21)) / k3;
    B = ((k1 - dr12*dr12/2)*(dx23) - (x2 - dr23*dr23/2)*(dx21)) / k3;

    C = (dr12*(dy23) + dr23*(dy21)) / k4;
    D = ((k1 - dr12*dr12/2)*(dy23) - (x2 - dr23*dr23/2)*(dy21)) / k4;

    a = C*C + A*A - 1;
    b = 2*(C*(D-bcns[1].x) + A*(B-bcns[1].y));
    c = (D-bcns[1].x)*(D-bcns[1].x) + (B-bcns[1].y)*(B-bcns[1].y);

    temp = (b*b - 4*a*c);
    if temp < 0)
        return -1;
    temp = sqrt(temp)/(2*a);
    root1=-b/(2*a)+temp;
    root2=-b/(2*a)-temp;
}

```

```

if(root1>0) {
    if(root2>0)
        return -1;
    *y = C*root1 + D;
    *y = A*root1 + B;
} else {
    if(root2<0)
        return -1;
    *x = C*root2 + D;
    *y = A*root2 + B;
}

**
* Rutina principal, la cual devuelve en x y y las coordenadas del robot
* En caso de no poder determinar la posición, la rutina devuelve -1.
*
int getpos(float *x, float *y) {
    int n, k;
    int first; /* Índice del primero de un conjunto de tres picos contiguos */

    float umbral=20; /* [0-128] */
    float dr12, dr23;
    FILE *out;

    unsigned char maxval[10]={0, 0, 0, 0, 0, 0, 0, 0, 0, 0};
    int maxindx[10]={0, 0, 0, 0, 0, 0, 0, 0, 0, 0};
    int deltaindx[10]={0, 0, 0, 0, 0, 0, 0, 0, 0, 0};

    _initSound();
    bufsize = read(in_fd, buf, BUFSIZE);
    close in_fd;

    out = fopen("in.raw", "wb");
    fwrite(buf, 1, bufsize, out);
    fclose(out);

    x=0; y=0;
    while(k < bufsize) {

        if((buf[k]=abs(buf[k]-128)) > umbral) {
            if(buf[k] > maxval[m]) {
                maxval[m]=buf[k];
                maxindx[m]=k;
            } else if(k > maxindx[m]+800) {
                if(++m>5) break;
            }
        }
        k++;
    }

    for(x=0; x<m; k++) {
        deltaindx[x] = maxindx[k+1]-maxindx[k];
    }
}

```

```

if((deltaidx[0]<2000 && deltaidx[1] < 2000) {
    first=0;
} else if((deltaidx[1]<2000 && deltaidx[2] < 2000) {
    first=1;
} else if((deltaidx[2]<2000 && deltaidx[3] < 2000) {
    first=2;
} else
    return -2;

* printf{'d12=%6d d23=%6d\n', deltaidx[first], deltaidx[first+1]}. */
* 1360 y 1350 son constantes empíricas de tal forma que cuando el micrófono
  se encuentra equidistante de los faros dr12 = dr23 */
dr12 = (deltaidx[first]-1360)/fs*vs;
dr23 = (deltaidx[first+1]-1350)/fs*vs;

if (posix(&dr12 &&dr23) == -1)
    return -1;
*x = dr12;
*y = dr23;
return 0;
}

void initSound() {
int soundFormat = AFMT_U8, stereo = 0;
int fstep;

* Let's call aumix to set the mic input volume */
system('aumix -m 100');
if (in_fd = open('/dev/dsp', O_RDONLY) == -1) {
    perror('/dev/dsp');
    exit(-1);
}
if (ioctl(in_fd, SNDCTL_DSP_SETPMT, &soundFormat) == -1) {
    perror('SNDCTL_DSP_SETPMT');
    exit(-1);
}
if (soundFormat != AFMT_U8) {
    printf('Error: Formato de audio requerido no disponible\n');
    exit(-1);
}
if (ioctl(in_fd, SNDCTL_DSP_STEREO, &stereo) == -1) {
    perror('SNDCTL_DSP_STEREO');
    exit(-1);
}
if (stereo == 0) {
    printf('Error: Audio mono no disponible\n');
    exit(-1);
}
fstep = (int)fs;
if (ioctl(in_fd, SNDCTL_DSP_SPEED, &fstep) == -1) {
    perror('SNDCTL_DSP_SPEED');
}
}

```

```

        exit(-1);
    }
    /* printf('fs=%d\n', fstemp); */
    if fstemp != (int)fs) {
        printf,'Frecuencia de muestreo no disponible'\n';
        exit(-1);
    }
}

ccord getBeaconPosition(int beacon) {
    return bcns[beacon];
}

ccord setBeaconPosition(int beacon, float x, float y) {
    bcns[beacon].x=x;
    bcns[beacon].y=y;
    update();
}

```

Archivo main.c

Este es un pequeño programa de pruebas que hace uso del sistema desarrollado. El programa dibuja en la pantalla una representación esquemática del área de trabajo así como de la posición de los faros. Sobre el dibujo se grafica la posición que reporta el sistema

```

#include <stdio.h>
#include <unistd.h>
#include <sys/stat.h>
#include <fcntl.h>
#include <math.h>
#include <vga.h>
#include 'lps.h'

void Plot(float x, float y, int color, char mark);

float ppax, ppuy;
float xmin=-80, xmax=480, ymin=-10, ymax=410;
float left=0, right=400, top=400, bottom=0;

int main(int argc, char *argv[]) {
    float x, y, ox=1, oy=1;
    FILE *file, *file2;
    int quit=0;
    int rc /* Return code */
    int n,n1;
    int err, err2;
    char basura[10];
    int tp=0;
}

```

```

if(argc==2.
    file = fopen(argv[1], 'w+');
else
    file = fopen('out.dat', 'w+');

file2 = fopen('errs.dat', 'w+');

initlpsf.;
setBeaconPosition(0, 210, 0);
setBeaconPosition(1, 200, 400);
setBeaconPosition(2, 190, 0);

vga_setmode(G640x480x16);

ppuv=(vga_getxdim()-2)/(xmax-xmin);
ppuy=(vga_getydim()-2)/(ymax-ymin);
Plot(getBeaconPosition(0).x, getBeaconPosition(0).y, 4, 'o');
Plot(getBeaconPosition(1).x, getBeaconPosition(1).y, 4, 'o');
Plot(getBeaconPosition(2).x, getBeaconPosition(2).y, 4, 'o');

Plot(200, 150, 4, 'o');

for(n=0; n<4; n++) {
    for(m=0; m<4; m++) {
        Plot(n*100+50, m*100+50, 2, 'x ');
    }
}

do {
    err=0;
    err2=0;
    do {
        if((rc=getpos(&x, &y)) == 0) {
            if(x<left || x>right || y<bottom || y>top) {
                /* La posicion cae fuera del cuarto.
                Ignoramos el resultado */
                printf('Oops' Error Outside boundaries! \n');
                err2++;
            } else {
                Plot(ox, oy, 7, 'x');
                Plot(x, y, 4, 'x');
                ox = x;
                oy = y;
                fprintf(xfile, "%4.6f ", x);*/
                fwrite(&x, sizeof(float), 1, file);
                fwrite(&y, sizeof(float), 1, file);
                quit++;
            }
        } else {
            printf('Cops' Error %d\n', rc);
            err++;
        }
    } while(quit<50);
    fwrite(&err, sizeof(int), 1, file2);
    fwrite(&err2, sizeof(int), 1, file2);
}

```

```

printf("Errores tipo 1=%d\n", err),
printf("Errores tipo 2=%d\n\n", err2),
Plot(5 5, 9, 'x');
scanf("%s" basura);
Plot(5 5, 7, 'x');
quit=0;
while(++tp<16 && strcmp(basura, "quit")):
printf("Errors= %d\n", err);
fclose(file);
fclose(file2);
exit(0);
}

void Plot(float x, float y, int color, char mark) {
vga_setcolor(color);
switch(mark) {
case 'x':
vga_drawpixel(ppux*x-ppux*xmin-1, vga_getydim()-1-ppuy*y+ppuy*ymin-1);
vga_drawpixel(ppux*x-ppux*xmin+1, vga_getydim()-1-ppuy*y+ppuy*ymin-1);
vga_drawpixel(ppux*x-ppux*xmin, vga_getydim()-1-ppuy*y+ppuy*ymin);
vga_drawpixel(ppux*x-ppux*xmin-1, vga_getydim()-1-ppuy*y+ppuy*ymin+1);
vga_drawpixel(ppux*x-ppux*xmin+1, vga_getydim()-1-ppuy*y+ppuy*ymin+1);
break
case 'o'
vga_drawpixel(ppux*x-ppux*xmin, vga_getydim()-1-ppuy*y+ppuy*ymin-2);
vga_drawpixel(ppux*x-ppux*xmin+1, vga_getydim()-1-ppuy*y+ppuy*ymin-1);
vga_drawpixel(ppux*x-ppux*xmin-1, vga_getydim()-1-ppuy*y+ppuy*ymin-1);
vga_drawpixel(ppux*x-ppux*xmin+2, vga_getydim()-1-ppuy*y+ppuy*ymin);
vga_drawpixel(ppux*x-ppux*xmin-2, vga_getydim()-1-ppuy*y+ppuy*ymin);
vga_drawpixel(ppux*x-ppux*xmin+1, vga_getydim()-1-ppuy*y+ppuy*ymin+1);
vga_drawpixel(ppux*x-ppux*xmin-1, vga_getydim()-1-ppuy*y+ppuy*ymin+1);
vga_drawpixel(ppux*x-ppux*xmin, vga_getydim()-1-ppuy*y+ppuy*ymin+2);
break;
default:
vga_drawpixel(ppux*x-ppux*xmin, vga_getydim()-1-ppuy*y+ppuy*ymin);
break;
}
}
}

```

Archivo lpsdevel.h

Este archivo se utiliza para compilar el código del sistema de localización. Contiene los prototipos de las funciones disponibles. No es necesario incluir este archivo para hacer uso del sistema

```

#ifndef POSITION_H
#define POSITION_H

#include'lps.h'

```

```

void initSound(void);
int pos(float *x, float *y);
void filtra(unsigned char *buf, int bufsize);
void update(void). /* Recalcula las constantes si se modifican las posiciones de los
farcos */

#endif

```

main.asm

Este es un programa en ensamblador para el DSP56002 con el cuál se hicieron pruebas de concepto de los filtros digitales y de la detección de las señales. Este programa no incluye los cálculos de posición.

```

; Sistema de localización para robots móviles.

include 'codec.asm'

TONE_OUTPUT EQU HEADPHONE_EN+LINEOUT_EN+(4*LEFT_ATTEN)+(4*RIGHT_ATTEN)
TONE_INPUT EQU MIC_IN_SELECT+(15*MONITOR_ATTEN)

PBDDR equ $FFE2
PBD equ $FFE4
DELAY equ 400

nsec equ 4 ; Numero de segmentos de 2o orden
winsize equ 32
nbral equ 0 05

.....
; Mapa de memoria X
.....
org X.
states ds 2*nsec ; Reserva un bloque para almacenar los estados
timer dc 0
control dc 0
time1 dc 0
time2 dc 0
TABLE BSM winsize

.....
; Mapa de memoria Y
.....
org Y:
coefs1 equ *
include 'coef_18.asm'
coefs2 equ *
include 'coef_20.asm'
coefs3 equ *
include 'coef20.asm'

```

```

señales      ds 3
AVG          dc 0      . promedio de la seál en la ventana
counter1dc 0  ; numero de veces que la se\ñal rebasa el umbral

.....
: Programa
.....
      org      p
:air
      .....
      Inicializamos registros, variables, tablas, etc
      .....
movep #S1,x PBDDR      ; Set pin 0 for output (PORT b)

move  #TCNE_OUTPUT,y0      ; headphones, line out, mute spkr, no attn.
move  y0,x:TX_BUFF_BASE+2
move  #TCNE_INPUT,y0      ; no input gain, monitor mute
move  y0,x:TX_BUFF_BASE+3

move  #TABLE,r3
move  #{winsize-1},m3

move  #>ccefsf1,x0
move  x0,y:handles
move  #>coefs2,x0
move  x0,y:handles+1
move  #>coefs3,x0
move  x0,y:handles+2
move  #handles,r7
move  #1 m7      ; Modulo 3 *ahora modulo 2 para probar

MULBsp
;set  #2,x:SSISR,*      . Wait for frame sync to pass.
;clr  #2,x:SSISR,*      ; Wait for frame sync.

move  x:RX_BUFF_BASE,y1      ; Cargamos y guardamos se\ñal original

.....
      Filtro
      .....
      E7 apunta al filtro activo. En la primera localidad está
      almacenada la ganancia En las localidades siguientes
      estan almacenados los coeficientes.
move  y:(r7),r4
move  r4,a0
inc  a
move  y:(r4).y0      ; Cargamos la ganancia del filtro en y0
move  a0,r4      . Carga los coeficientes adecuados en r4

move  #states,r0
ori  #S08,mr      ; Sets scale up mode
mpy  y0,y1,b      x (r0)+,x0      y:(r4)+,y0

```

```

do #nsec endDoF
mac  x0,y0,b x (r0)+,y1  y:(r4)+,y0
mac  x1,y0,b x1,y (r0)+  y:(r4)+,y0
mac  x0,y0,b  b,x (r0)+  y:(r4)+,y0
macr x1 y0,b  x:(r0)+,x0  y:(r4)+,y0
endDoF
abs  r
move b,y0
andi  #0x7fff  r , Termina modo de escalamiento

move y0,x:TX_BUFF_BASE ; La se;al filtrada al canal izquierdo
move y1,x:TX_BUFF_BASE+1 ; La se;al original al canal derecho
jmp  MLoop

; Calculamos el promedio
move  #1.0/winsize,x0
mpy  x0,y0,a  y:AVG,b
move  x (r3),x0 ; Saca el valor más antiguo
move  a,x:(r3)+ ; Mete el nuevo valor
sub  x0,b
add  a,b
move  b y:AVG
move  b x0

move y0,x:TX_BUFF_BASE ; La se;al filtrada al canal izquierdo
move x0,x:TX_BUFF_BASE ; El promedio original al canal derecho
jmp  MLoop

move  #umbral,a
cmp  x0,a . a>x0 ?

jgt  no
--
move  #0,x0
move  x0,x:TX_BUFF_BASE+1 ; La se;al filtrada al canal izquierdo

clr  b
move  y:counter1,b0
inc  b
clr  a
move  #>{winsize/2},a0
; cmp  a,b
jgt  enough
. move  b0,y:counter1
jmp  MLoop

enough
move  #0.9,x0
move  x0,x:TX_BUFF_BASE+1 ; La se;al filtrada al canal izquierdo

clr  b
. move  b,y:counter1
movep #1,x:PBD ; Prendemos led para indicar que detectamos se;nal
move y:(r7)+,x0 ; Incrementamos r7 para cambiar de filtro

```

```

; Borro promedio, vector de estados y tabla
move #states,r2
move #0,x0
move x0,y:AVG
rep #2*nsec
move x0,x:{r2}+
rep #winsize
move x0,x:{r3}+

clr a
move x:timer,x0
move x0,x timer1 , Guardamos el tiempo
move a0,x:timer , Reiniciamos el timer
move #DELAY,x0

Una vez detectada la se\nal esperamos cierto tiempo
para evitar multiples detecciones de la misma se\nal
aquí move x:timer,a1
cmp x0,a
jlt aqui

jmp MLoop

no
move #0,x0
move x0,x TX_BUFF_BASE+1 , La se\al filtrada al canal izquierdo

movep #0 x:PBD: Apagamos LED
jmp MLoop

end

```

codec.asm

Este programa es necesario para hacer uso del codec de la tarjeta EVM56002. Se encarga de programar ciertos aspectos de operación como frecuencia de muestreo, amplitud del preamplificador, tipo de codificación, etc.

```

page 132,60
.....
CODEC ASM
Initialization program for EVM56002 to communicate with CS4215
input connected to Mic inputs

Copyright (c) MOTOROLA 1994
Semiconductor Products Sector
Digital Signal Processing Division
.....
ports usage

```

```

    bit8: SSI TX (from DSP to Codec)
    bit7:
    bit6:
    bit5:
    bit4: cdec reset (from DSP to Codec)
    bit3:
        bit2: data/control bar
            0=control
            1=data

```

PROGRAM OUTLINE.

```

1 program fsync and sclk == output
2 write pc0 = 0 (control mode)
3 send 64 bit frame x times, with dcb bit = 0, keep doing until read back as 0
4 send 64 bit frame x times, with dcb bit = 1, keep doing until read back as 1
5 re-program fsync and sclk == input
6 write pc0 = 1 (data mode)
7 receive/send data (echo slots 1,2,3,4, slots 5,6,7,8 == DIP switched)

```

.....

PAGE 255

```

NO_PREAMP      equ      $100000
LC_OVT_DRV     equ      $080000
HI_PASS_FILT   equ      $008000
SAMP_RATE_9    equ      $003800
SAMP_RATE_48   equ      $003000
SAMP_RATE_32   equ      $001800
SAMP_RATE_27   equ      $001000
SAMP_RATE_16   equ      $000800
SAMP_RATE_8    equ      $000000
STEPEO        equ      $000400
DATA_BLIN     equ      $200300
DATA_8A       equ      $200200
DATA_8V       equ      $200100
DATA_16       equ      $200000
IMMED_3STATE  equ      $800000
XTAL2_SELECT  equ      $200000
BITS_64       equ      $000000
BITS_28       equ      $040000
BITS_256      equ      $080000
CODEC_MASTER   equ      $020000
CODEC_TX_OFF  equ      $010000

CTRL_WD_12     equ      NO_PREAMP+HI_PASS_FILT+SAMP_RATE_48+STEREO+DATA_16    ;CLB=0
CTRL_WD_34     equ      IMMED_3STATE+XTAL2_SELECT+BITS_64+CODEC_MASTER
CTRL_WD_56     equ      $000000
CTRL_WD_78     equ      $000000

HEADPHONE_EN  equ      $800000
LINEOUT_EN    equ      $400000
LEFT_ATTN     equ      $010000 ;63*LEFT_ATTN = -94.5 dB, 1.5 dB steps

```

```

SPEAKER_EN      equ      $004000
RIGHT_ATTEN     equ      $000100 .63*RIGHT_ATTEN = -94.5 dB, 1 5 dB steps
MIC_IN_SELECT   equ      $100000
LEFT_GAIN       equ      $010000 ;15*LEFT_GAIN   = 22.5 dB, 1.5 dB steps
MONITOR_ATTEN   equ      $001000 ;15*MONITOR_ATTEN = mute,    6 dB steps
RIGHT_GAIN       equ      $000100 ;15*RIGHT_GAIN   = 22.5 dB, 1 5 dB steps
OUTPUT_SET      equ      HEADPHONE_EN+LINEOUT_EN+(LEFT_ATTEN*4)
INPUT_SET       equ      MIC_IN_SELECT+(15*MONITOR_ATTEN)+(RIGHT_ATTEN*4)

```

--ESP56502 on-chip peripheral addresses

```

PFI             equ      $FPE5
FPEDP          equ      $FPE3
PFI            equ      $FPE1
PBC            equ      $FPE0
PBA            equ      $FPEC
PRB            equ      $FPEF
SSISR          equ      $FPEF
IPR            equ      $FFFF
BCP            equ      $FFFE
SSISP          equ      $FPEE
PLL            equ      $FFFD
START          equ      $40

```

```

      org      x*0
RX_BUFF_BASE   equ      *
RX_data_1_2    ds      1      ;data time slot 1/2 for RX ISR
RX_data_3_4    ds      1      ;data time slot 3/4 for RX ISR
RX_data_5_6    ds      1      ;data time slot 5/6 for RX ISR
RX_data_7_8    ds      1      ;data time slot 7/8 for RX ISR

TX_BUFF_BASE   equ      *
TX_data_1_2    ds      1      ;data time slot 1/2 for TX ISR
TX_data_3_4    ds      1      ;data time slot 3/4 for TX ISR
TX_data_5_6    ds      1      ;data time slot 5/6 for TX ISR
TX_data_7_8    ds      1      ;data time slot 7/8 for TX ISR

RX_PTR         ds      1      ; Pointer for rx buffer
TX_PTR         ds      1      ; Pointer for tx buffer

```

STACK ds 6 ; Stack para las rutinas de interrupcion

```

org      p.0
jmp      START

```

```

dup 11
nop
end*

```

```

jsr      ssi_rx_isr      ;SSI RX
jsr      ssi_rx_isr      ;SSI RX w/Exception
jsr      ssi_tx_isr      ;SSI TX
jsr      ssi_tx_isr      ;SSI TX w/Exception

```

```

dw; 54
nop
endr

```

Main coded initialization program

```

org     p START

movew  #S261009,x:PLL      ;set PLL for MPY of 10x
movew  #S0000,x:BCR       , number of wait states
ori    #3,mr              ;disable interrupts
movew  #0,sp
movew  #0,omr             ; single chip mode

movew  #S40,r6            ; initialize stack pointer
movew  #STACK,r6         ; initialize stack pointer
movew  #-1,m6             ; linear addressing
movew  #RX_BUFF_BASE,x0
movew  x0,x:RX_PTR        ; Initialize the rx pointer
movew  #TX_BUFF_BASE,x0
movew  x0,x:TX_PTR        ; Initialize the tx pointer
jst    codec_init

```

The following line of code jumps to 'main' which is the location for the main code that will run after the codec is initialized

```
jmp main
```

```

.....
*****      initialize the CS4215 codec          *****
;.....
; readphones and line out, and set up for no gain or attenuation, and no
; monitor feedback
;.....
;.....

```

initialize ssi -- fsync and sclk ==> outputs

```

codec_init
movew  #S0000,x:PC0       ; turn off ssi port
movew  #S4303,x:CRA       ; 40MHz/16 = 2.5MHz SCLK, WL=16 bits, 4W/F
movew  #SFB30,x:CRB       ; RIE,TIE,RE,TE, NTWK, SYN, PSR/RSR->bit
movew  #S14,x:PCDDR       ; setup pc2 and pc4 as outputs
movew  #S0,x:PCD          ; D/C- and RESET- = 0 ==> control mode
;----reset delay for codec ----
do     #500,_delay_loop
rep    #2000              ; 100 us delay
nop

_delay_loop
hset   #4,x:PCD           , RESET- = 1
movew  #S3000,x:IPR       , set interrupt priority level

```

```

andi    #SFC,rr      , enable interrupts

.....

; The following data sets up the CS4215 control mode data
; CTS = Control Time Slot, U/LN = upper/lower Nibble)
;
;----- CTS1-UN:  0      0      1      MLB      0 0 0 0
;|----- CTS1-LN:  OLB    CLB    X      X      0 0 0 0
;|----- CTS2-UN:  HPP    X      DFR2   DFR1   0 0 1 0
;||----- CTS2-LN:  DFR0   ST     DF1    DF0    1 1 0 0
x0 = $002Cxx

;----- CTS3-UN:  ITS     MCK2   MCK1   MCK0   1 0 0 0
;|----- CTS3-LN:  BSEL1  BSEL0  XCLK   XEN    1 0 0 0
;||----- CTS4-UN:  TEST   TEST   TEST   TEST   (TEST MUST BE 0)
;|||----- CTS4-LN:  TEST   TEST   ENL    DAD    0 0 0 0
; x0 = $8800xx

.....

--- set up buffer with control mode data
move    #CTRL_WD_12,x0
move    x0,x:TX_BUFF_BASE
move    #CTRL_WD_34,x0
move    x0,x:TX_BUFF_BASE+1
move    #CTRL_WD_56,x0
move    x0,x:TX_BUFF_BASE+2
move    #CTRL_WD_78,x0
move    x0,x:TX_BUFF_BASE+3

movep   #S01E8,x.PCC      , Turn on ssi port

CLB == 0

;clr    #3,x:SSISR,*      ; wait until rx frame bit==1
;set    #3,x:SSISR,*      ; wait until rx frame bit==0
;clr    #3,x:SSISR,*      ; wait until rx frame bit==1
;set    #18,x:RX_BUFF_BASE,* ; loop until CLB set

;
; CLB == 1

;set    #18,x TX_BUFF_BASE ;set CLB
;dc     #4,_init_loopB
;clr    #2,x SSISR,*      ; wait until tx frame bit==1
;set    #2,x SSISR,*      ; wait until tx frame bit==0
_init_loopB
;movep  #0,x PCC          ;disable, reset SSI

.....

; now CLB should be 1 -- re-program fsync and sclk direction (i/p) -- also,
; circular buffer pointers for echoing data r0=current, r1=old data to send
; i frame later

```

ESTA TESIS NO DEBE SALIR DE LA BIBLIOTECA

```

movewp #54303,x:CRA      - 16bits,4 word/frame, /2/4/2=2.5 MHz
movewp #5FB00,x:CRB     , rcv.xmt & int ena,netwk,syn,sclk=1/p.msb lst
movewp #514,x:PCD       , D/C- pin = 1 ==> data mode
movewp #501E8,x:PCC     ; turn on ssi port
rts

;..... SSI TRANSMIT ISR .....
ssi_tx_isr
movewp r0,x:{r6}+      , Save r0 to the stack
movewp r0,x:{r6}+      ; Save m0 to the stack.
movewp #3,m0           ; Modulus 4 buffer
movewp x:TX_PTR,r0     , Load the pointer to the tx buffer
jclr #2,x:SSISR,next_tx ; If not frame sync, jump to transmit data
movewp #TX_BUFF_BASE+1,r0 ; If frame sync, reset pointer
nop
next_tx
movewp x:{r0}+ x:SSIDR , SSI transfer data register
movewp r0,x:TX_PTR    , Update tx buffer pointer.
movewp {r6}-
movewp x:{r6}-,m0     ; Restore m0.
movewp x:{r6},r0     ; Restore r0.
rts

;..... SSI receive ISR.....
ssi_rx_isr
movewp r0,x:{r6}+      ; Save r0 to the stack.
movewp r0,x:{r6}+      ; Save m0 to the stack.
movewp a0,x:{r6}+      ; Save m0 to the stack
movewp a1,x:{r6}+      ; Save m0 to the stack

movewp #3,m0           ; Modulo 4 buffer
movewp x:RX_PTR,r0     ; Load the pointer to the rx buffer.
jclr #3,x:SSISR,next_rx ; If not fr syc, jump to receive data.
movewp #RX_BUFF_BASE,r0 ; If frame sync, reset base pointer.
nop
; .....
; Timer para registrar tiempo entre señal y señal
; .....
movewp x:timer,a0
inc a
movewp a0,x:timer
next_rx
movewp x:SSIDR x {r0}+ ; Read out received data to buffer
movewp r0,x:RX_PTR    , Update rx buffer pointer.

movewp {r6}-
nop
movewp x:{r6}-,a1     ; Restore a1
movewp x:{r6}-,a0     ; Restore a0
movewp x:{r6}-,m0     ; Restore m0.
movewp x:{r6},r0     ; Restore r0.
rts

```

Programa de MatLab para el cálculo de los coeficientes del filtro digital IIR

```
% Funcion que devuelve la ganancia y los coeficientes
% de los polinomios normalizados de segundo orden de
% las secciones bicuadraticas de un filtro IIR paso bandas.
% Ademas genera un archivo de coeficientes para usarse
% con el ensamblador del DSP56002

function [a1,b1,a2,b2,a3,b3,a4,b4]=bqirf(fl, fh, fs, filename)
clear all
%format compact
%format long
w=[fl/(fs/2), fh/(fs/2)];
[Z,P,gain]=butter(4,W);
Z=cplxpair(Z); % Ordenamos por pares conjugados
P=cplxpair(P); % Ordenamos por pares conjugados

% Con los polos y ceros podemos calcular los coeficientes
% de cada seccion bicuadratica.

gain
%disp('---- Primera seccion: ----');
a12=P(1)*P(2);
a11=P(1)+P(2);
b12=Z(1)*Z(2);
b11=Z(1)+Z(2);

%disp('---- Segunda seccion: ----'),
a22=P(3)*P(4);
a21=P(3)+P(4);
r22=Z(3)*Z(4);
r21=Z(3)+Z(4);

%disp('---- Tercera seccion: ----'),
a32=P(5)*P(6);
a31=P(5)+P(6);
r32=Z(5)*Z(6);
r31=Z(5)+Z(6);

%disp('---- Cuarta seccion: ----')
a42=P(7)*P(8);
a41=P(7)+P(8);
b42=Z(7)*Z(8);
b41=Z(7)+Z(8);

fid=fopen(filename,'wt');
fprintf(fid, '\t:Coeficientes bpf (%0.2f, %0.2f) para fs=%0.2f:\n\n',fl, fh, fs);
fprintf(fid, '\t:dc\t%.7f\t, Ganancia\n',gain);
fprintf(fid, '\t:gain\tequ\t%.7f\t: Ganancia\n\n',gain);

fprintf(fid, '\t:Seccion bicuadratica 1:\n');
fprintf(fid, '\t:dc\t%.7f/2\t:a11\n',a11);
```

```

fprintf(fid, '\tde\t% 7f/2\t;a12\n', a12);
fprintf(fid, '\tde\t%.7f/2\t;b11\n', b11);
fprintf(fid, '\tde\t%.7f/2\t;b12\n\n', b12);

fprintf(fid, '\t;Seccion bicuadratica 2 \n');
fprintf(fid, '\tde\t%.7f/2\t;a21\n', a21);
fprintf(fid, '\tde\t% 7f/2\t;a22\n', a22);
fprintf(fid, '\tde\t% 7f/2\t;p21\n', p21);
fprintf(fid, '\tde\t% 7f/2\t;b22\n\n', b22);

fprintf(fid, '\t;Seccion bicuadratica 3:\n');
fprintf(fid, '\tde\t% 7f/2\t;a31\n', a31);
fprintf(fid, '\tde\t% 7f/2\t;a32\n', a32);
fprintf(fid, '\tde\t%.7f/2\t;b31\n', b31);
fprintf(fid, '\tde\t%.7f/2\t;b32\n\n', b32);

fprintf(fid, '\t;Seccion bicuadratica 4 \n');
fprintf(fid, '\tde\t%.7f/2\t;a41\n', a41);
fprintf(fid, '\tde\t%.7f/2\t;a42\n', a42);
fprintf(fid, '\tde\t% 7f/2\t;b41\n', b41);
fprintf(fid, '\tde\t%.7f/2\t;b42\n\n', b42);

fclose(fid);

a1=[a11 a12];
n1=[b11 b12];

a2=[a21 a22];
n2=[p21 b22];

a3=[a31 a32];
n3=[p31 p32];

a4=[a41 a42];
n4=[b41 b42];

```

7.2 Hoja de Especificaciones del LM1875

LM1875 20W Audio Power Amplifier

General Description

The LM1875 is a monolithic power amplifier offering very low distortion and high quality performance for consumer audio applications.

The LM1875 delivers 20 watts into a 4Ω or 8Ω load on ±25V supplies. Using an 8Ω load and ±30V supplies, over 30 watts of power may be delivered. The amplifier is designed to operate with a minimum of external components. Device overload protection consists of both internal current limit and thermal shutdown.

The LM1875 design takes advantage of advanced circuit techniques and processing to achieve extremely low distortion levels even at high output power levels. Other outstanding features include high gain, fast slew rate and a wide power bandwidth, large output voltage swing, high current capability, and a very wide supply range. The amplifier is internally compensated and stable for gains of 10 or greater.

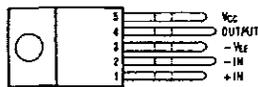
Features

- Up to 30 watts output power
- A_{VO} typically 90 dB
- Low distortion 0.015%, 1 kHz, 20 W
- Wide power bandwidth: 70 kHz
- Protection for AC and DC short circuits to ground
- Thermal protection with parole circuit
- High current capability 4A
- Wide supply range 16V-60V
- Internal output protection diodes
- 94 dB ripple rejection
- Plastic power package TO-220

Applications

- High performance audio systems
- Bridge amplifiers
- Stereo phonographs
- Servo amplifiers
- Instrument systems

Connection Diagram

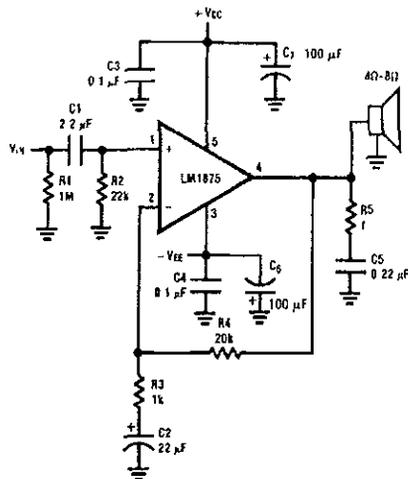


Front View

TL/H/5030-1

Order Number LM1875T
See NS Package Number T05B

Typical Applications



TL/H/5030-2

Absolute Maximum Ratings

If Military/Aerospace specified devices are required, please contact the National Semiconductor Sales Office/Distributors for availability and specifications.

Supply Voltage 60V
Input Voltage $-V_{EE}$ to V_{CC}

Storage Temperature -65°C to $+150^{\circ}\text{C}$
Junction Temperature 150°C
Lead Temperature (Soldering, 10 seconds) 260°C
 θ_{JC} 3°C
 θ_{JA} 73°C

Electrical Characteristics

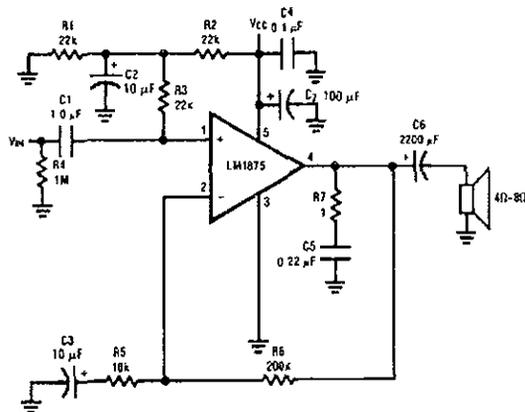
$V_{CC} = +25\text{V}$, $-V_{EE} = -25\text{V}$, $T_{\text{AMBIENT}} = 25^{\circ}\text{C}$, $R_L = 8\Omega$, $A_V = 20$ (26 dB), $f_o = 1\text{ kHz}$, unless otherwise specified

Parameter	Conditions	Typical	Tested Limits	Units
Supply Current	$P_{\text{OUT}} = 0\text{W}$	70	100	mA
Output Power (Note 1)	THD = 1%	25		W
THD (Note 1)	$P_{\text{OUT}} = 20\text{W}$, $f_o = 1\text{ kHz}$ $P_{\text{OUT}} = 20\text{W}$, $f_o = 20\text{ kHz}$ $P_{\text{OUT}} = 20\text{W}$, $R_L = 4\Omega$, $f_o = 1\text{ kHz}$ $P_{\text{OUT}} = 20\text{W}$, $R_L = 4\Omega$, $f_o = 20\text{ kHz}$	0.015 0.05 0.022 0.07	0.4 0.6	% % % %
Offset Voltage		± 1	± 15	mV
Input Bias Current		± 0.2	± 2	μA
Input Offset Current		0	± 0.5	μA
Gain-Bandwidth Product	$f_o = 20\text{ kHz}$	5.5		MHz
Open Loop Gain	DC	90		dB
PSRR	V_{CC} : 1 kHz, 1 Vrms V_{EE} : 1 kHz, 1 Vrms	95 83	52 52	dB dB
Max Slew Rate	20W, 8 Ω , 70 kHz BW	8		V/ μs
Current Limit	$V_{\text{OUT}} = V_{\text{SUPPLY}} - 10\text{V}$	4	3	A
Equivalent Input Noise Voltage	$R_S = 600\Omega$, CCIR	3		μVrms

Note 1: Assumes the use of a heat sink having a thermal resistance of 1°C/W and no insulator with an ambient temperature of 25°C . Because the output limiting circuitry has a negative temperature coefficient, the maximum output power delivered to a 4Ω load may be slightly reduced when the tab temperature exceeds 55°C .

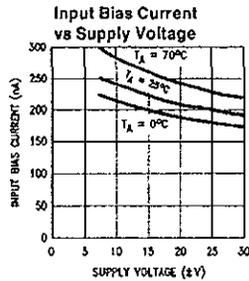
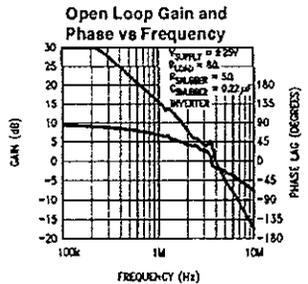
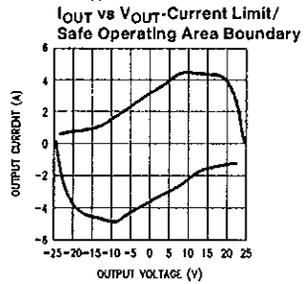
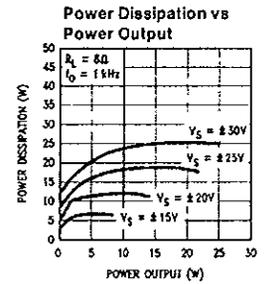
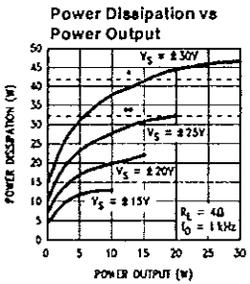
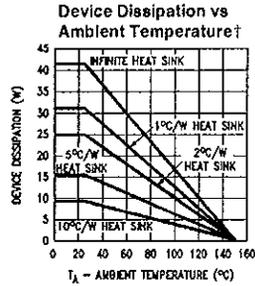
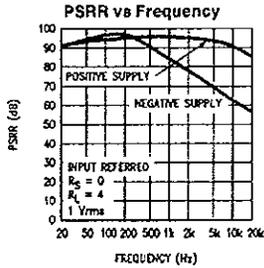
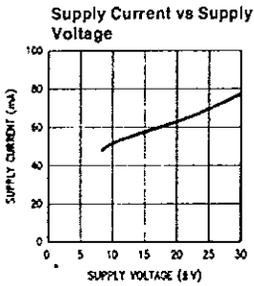
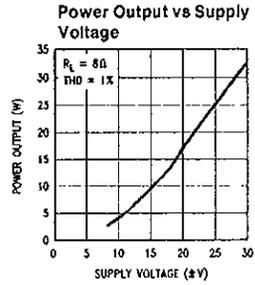
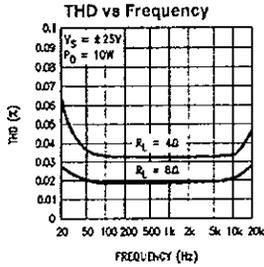
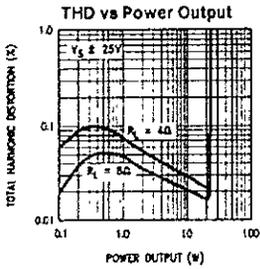
Typical Applications (Continued)

Typical Single Supply Operation



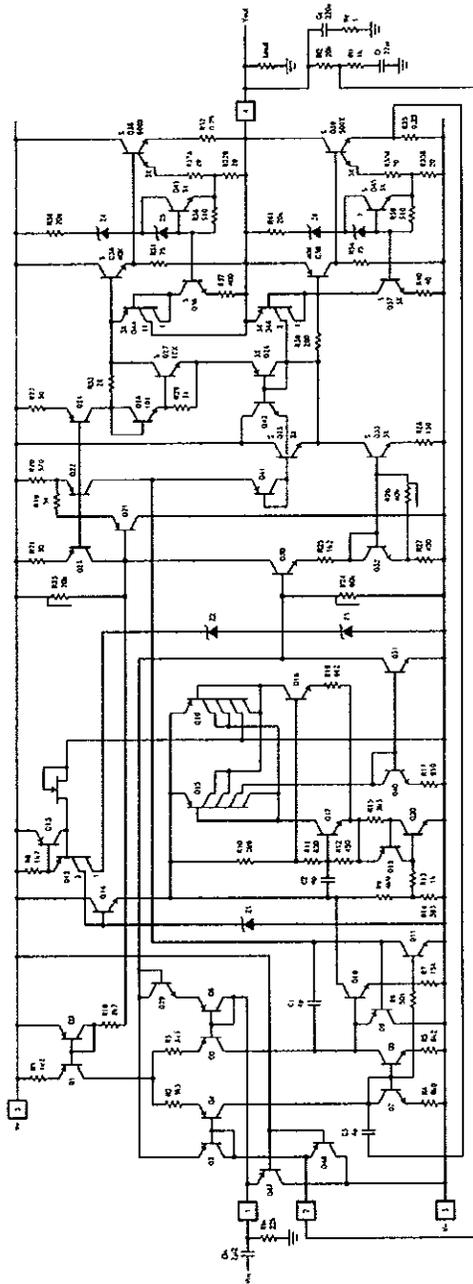
TL/H/6030-3

Typical Performance Characteristics



*Thermal shutdown with infinite heat sink
 **Thermal shutdown with 1°C/W heat sink

Schematic Diagram



TL/H/830-5

Application Hints

STABILITY

The LM1875 is designed to be stable when operated at a closed-loop gain of 10 or greater, but, as with any other high current amplifier, the LM1875 can be made to oscillate under certain conditions. These usually involve printed circuit board layout or output/input coupling.

Proper layout of the printed circuit board is very important. While the LM1875 will be stable when installed in a board similar to the ones shown in this data sheet, it is sometimes necessary to modify the layout somewhat to suit the physical requirements of a particular application. When designing a different layout, it is important to return the load ground, the output compensation ground, and the low level (feedback and input) grounds to the circuit board ground point through separate paths. Otherwise, large currents flowing along a ground conductor will generate voltages on the conductor which can effectively act as signals at the input, resulting in high frequency oscillation or excessive distortion. It is advisable to keep the output compensation components and the 0.1 μF supply decoupling capacitors as close as possible to the LM1875 to reduce the effects of PCB trace resistance and inductance. For the same reason, the ground return paths for these components should be as short as possible.

Occasionally, current in the output leads (which function as antennas) can be coupled through the air to the amplifier input, resulting in high-frequency oscillation. This normally happens when the source impedance is high or the input leads are long. The problem can be eliminated by placing a small capacitor (on the order of 50 pF to 500 pF) across the circuit input.

Most power amplifiers do not drive highly capacitive loads well, and the LM1875 is no exception. If the output of the LM1875 is connected directly to a capacitor with no series resistance, the square wave response will exhibit ringing if the capacitance is greater than about 0.1 μF . The amplifier can typically drive load capacitances up to 2 μF or so without oscillating, but this is not recommended. If highly capacitive loads are expected, a resistor (at least 1 Ω) should be placed in series with the output of the LM1875. A method commonly employed to protect amplifiers from low impedances at high frequencies is to couple to the load through a 10 Ω resistor in parallel with a 5 μH inductor.

DISTORTION

The preceding suggestions regarding circuit board grounding techniques will also help to prevent excessive distortion levels in audio applications. For low THD, it is also necessary to keep the power supply traces and wires separated from the traces and wires connected to the inputs of the LM1875. This prevents the power supply currents, which are large and nonlinear, from inductively coupling to the LM1875 inputs. Power supply wires should be twisted together and separated from the circuit board. Where these wires are soldered to the board, they should be perpendicular to the plane of the board at least to a distance of a couple of inches. With a proper physical layout, THD levels at 20 kHz with 10W output to an 8 Ω load should be less than 0.05%, and less than 0.02% at 1 kHz.

CURRENT LIMIT AND SAFE OPERATING AREA (SOA) PROTECTION

A power amplifier's output transistors can be damaged by excessive applied voltage, current flow, or power dissipation. The voltage applied to the amplifier is limited by the design of the external power supply, while the maximum current passed by the output devices is usually limited by internal circuitry to some fixed value. Short-term power dissipation is usually not limited in monolithic audio power amplifiers, and this can be a problem when driving reactive loads, which may draw large currents while high voltages appear on the output transistors. The LM1875 not only limits current to around 4A, but also reduces the value of the limit current when an output transistor has a high voltage across it.

When driving nonlinear reactive loads such as motors or loudspeakers with built-in protection relays, there is a possibility that an amplifier output will be connected to a load whose terminal voltage may attempt to swing beyond the power supply voltages applied to the amplifier. This can cause degradation of the output transistors or catastrophic failure of the whole circuit. The standard protection for this type of failure mechanism is a pair of diodes connected between the output of the amplifier and the supply rails. These are part of the internal circuitry of the LM1875, and needn't be added externally when standard reactive loads are driven.

THERMAL PROTECTION

The LM1875 has a sophisticated thermal protection scheme to prevent long-term thermal stress to the device. When the temperature on the die reaches 170°C, the LM1875 shuts down. It starts operating again when the die temperature drops to about 145°C, but if the temperature again begins to rise, shutdown will occur at only 150°C. Therefore, the device is allowed to heat up to a relatively high temperature if the fault condition is temporary, but a sustained fault will limit the maximum die temperature to a lower value. This greatly reduces the stresses imposed on the IC by thermal cycling, which in turn improves its reliability under sustained fault conditions.

Since the die temperature is directly dependent upon the heat sink, the heat sink should be chosen for thermal resistance low enough that thermal shutdown will not be reached during normal operation. Using the best heat sink possible within the cost and space constraints of the system will improve the long-term reliability of any power semiconductor device.

POWER DISSIPATION AND HEAT SINKING

The LM1875 must always be operated with a heat sink, even when it is not required to drive a load. The maximum idling current of the device is 100 mA, so that on a 60V power supply an unloaded LM1875 must dissipate 6W of power. The 54°C/W junction-to-ambient thermal resistance of a TO-220 package would cause the die temperature to rise 324°C above ambient, so the thermal protection circuitry will shut the amplifier down if operation without a heat sink is attempted.

Application Hints (Continued)

In order to determine the appropriate heat sink for a given application, the power dissipation of the LM1875 in that application must be known. When the load is resistive, the maximum average power that the IC will be required to dissipate is approximately

$$P_D(\text{MAX}) \approx \frac{V_S^2}{2\pi^2 R_L} + P_Q$$

where V_S is the total power supply voltage across the LM1875, R_L is the load resistance, and P_Q is the quiescent power dissipation of the amplifier. The above equation is only an approximation which assumes an "ideal" class B output stage and constant power dissipation in all other parts of the circuit. The curves of "Power Dissipation vs Power Output" give a better representation of the behavior of the LM1875 with various power supply voltages and resistive loads. As an example, if the LM1875 is operated on a 50V power supply with a resistive load of 8Ω, it can develop up to 19W of internal power dissipation. If the die temperature is to remain below 150°C for ambient temperatures up to 70°C, the total junction-to-ambient thermal resistance must be less than

$$\frac{150^\circ\text{C} - 70^\circ\text{C}}{19\text{W}} = 4.2^\circ\text{C/W}$$

Using $\theta_{JC} = 2^\circ\text{C/W}$, the sum of the case-to-heat-sink interface thermal resistance and the heat-sink-to-ambient thermal resistance must be less than 2.2°C/W. The case-to-heat-sink thermal resistance of the TO-220 package varies with the mounting method used. A metal-to-metal interface will be about 1°C/W if lubricated, and about 1.2°C/W if dry.

If a mica insulator is used, the thermal resistance will be about 1.6°C/W lubricated and 3.4°C/W dry. For this example, we assume a lubricated mica insulator between the LM1875 and the heat sink. The heat sink thermal resistance must then be less than

$$4.2^\circ\text{C/W} - 2^\circ\text{C/W} - 1.6^\circ\text{C/W} = 0.6^\circ\text{C/W}$$

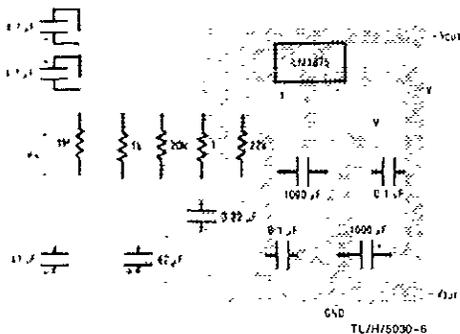
This is a rather large heat sink and may not be practical in some applications. If a smaller heat sink is required for reasons of size or cost, there are two alternatives. The maximum ambient operating temperature can be reduced to 50°C (122°F), resulting in a 1.6°C/W heat sink, or the heat sink can be isolated from the chassis so the mica washer is not needed. This will change the required heat sink to a 1.2°C/W unit if the case-to-heat-sink interface is lubricated.

Note When using a single supply, maximum transfer of heat away from the LM1875 can be achieved by mounting the device directly to the heat sink (tab is at ground potential), this avoids the use of a mica or other type insulator.

The thermal requirements can become more difficult when an amplifier is driving a reactive load. For a given magnitude of load impedance, a higher degree of reactance will cause a higher level of power dissipation within the amplifier. As a general rule, the power dissipation of an amplifier driving a 60° reactive load (usually considered to be a worst-case loudspeaker load) will be roughly that of the same amplifier driving the resistive part of that load. For example, a loudspeaker may at some frequency have an impedance with a magnitude of 8Ω and a phase angle of 60°. The real part of this load will then be 4Ω, and the amplifier power dissipation will roughly follow the curve of power dissipation with a 4Ω load.

Component Layouts

Split Supply



Single Supply

