29

UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

"SISTEMA DE MONITOREO DE UN SIMULADOR PARA CONTROL DE ORIENTACIÓN DE SATÉLITES."

T E S I S

QUE PARA OBTENER EL TÍTULO DE:

INGENIERA ELÉCTRICA - ELECTRÓNICA

P R E S E N T A:

MARCELA MESINAS ORTIZ

DIRECTOR DE TESIS:

M. en I. JORGE PRADO MOLINA

MÉXICO, D.F., SEPTIEMBRE DE 2002

PAGINACIÓN DISCONTINUA

DEDICATORIA

A mis padres:

Por haber contribuido en mi preparación académica y personal, haciendo posible la culminación de mi carrera. Y por darme su apoyo durante la realización de esta tesis.

A Sandra, que siempre me ha brindado su apoyo en lo que emprendo, por ser una persona incondicional que me a animado a seguir adelante y por ser una gran amiga.

A Ruth, de quien he aprendido a tomar las situaciones de la mejor manera posible.

A todos mis amigos: Elia, Yazmín, Alicia, Hugo, Alex, Chucho, Tonatiuh, Arturo, Marco Antonio, José Luis, Adrián, Juan Carlos, Uriel, Rafa, Fabis y Dani, que a lo largo de mi carrera me brindaron un apoyo incondicional, y gracias a ellos logré llegar al fin, con mucha satisfacción.

Y a Daniel por su gran ayuda, recomendaciones y consejos. Por la motivación que me dio a lo largo de este proyecto y por estar a mi lado en estos dos últimos años, compartiendo sus ideales y sueños, y por hacerme muy feliz.

Un soñador no es un hombre
si no se esfuerza
en realizar lo que sueña.
José Luis Jiménez.

RECONOCIMIENTOS

Agradezco a la Universidad Nacional Autónoma de México, Facultad de Ingeniería, por haberme brindado una formación profesional.

Al Instituto de Geografía de la UNAM, por promover la realización de este tipo de proyectos.

Al M. en I. Jorge Prado Molina por su excelente disposición durante la dirección de mi tesis, por sus recomendaciones y apoyo a lo largo de este trabajo, y por la meticulosa revisión del proyecto.

Al Ing. Salvador Zamora Alarcón por sus valiosos consejos, su gran ayuda durante el desarrollo de la programación del microcontrolador y por la revisión final del programa de control.

A Tonatiuh Cruz López por su colaboración en la realización del programa de graficación, por su esfuerzo y tiempo dedicado en el mismo.

Al Ing. Antonio Salvá Calleja por su contribución y apoyo en el manejo de la tarjeta del microcontrolador, FÁCIL_11.

ÍNDICE

RESUMEN

SISTEMA DE MONITOREO DE UN SIMULADOR PARA CONTROL DE ORIENTACIÓN DE SATÉLITES.

Se presentan el diseño y la implementación de un sistema para monitorear la orientación de una plataforma de simulación, utilizada para llevar a cabo pruebas de control de orientación de satélites pequeños. Durante el periodo de ensayo, es necesario contar con un monitoreo continuo de la orientación de la plataforma; para ello, fue necesario diseñar y construir un sistema que nos permitiera alcanzar los siguientes objetivos:

1.- La visualización directa de la orientación para tener una referencia rápida de lo que está ocurriendo con el experimento, esto nos proporciona una evaluación rápida y objetiva de los algoritmos de orientación y control que están siendo probados.

2.- El balanceo, de manera manual y automática del simulador, ya que un pequeño desequilibrio provoca una caída súbita de la plataforma.

3.- El almacenamiento de la información de la orientación, en tres ejes, en archivos que permitan analizar posteriormente, el comportamiento de la plataforma durante diferentes pruebas.

La plataforma se encuentra suspendida sobre un balero de aire esférico donde se genera un medio con fricción despreciable. Un parámetro crítico para el resultado óptimo de las pruebas que se realizan en este medio sin fricción, es el balanceo, por tal razón, no es permisible enviar la información de la orientación por medio de cables; ya que se producirá un desbalanceo inadmisible en la plataforma. Debido a esto, la comunicación entre la plataforma y la computadora, donde se despliega y almacena la información de la orientación, se llevó a cabo a través de una interfaz RS232 inalámbrica.

Una brújula electrónica, equipada con dos inclinómetros, proporciona los ángulos de orientación en tres ejes (alabeo, cabeceo y guiñada), con una frecuencia de doce ciclos por segundo; misma frecuencia con que se actualizan los datos en la pantalla de la PC y en el archivo de datos de orientación.

Se desarrolló la programación, la circuitería y se llevaron a cabo diferentes pruebas de funcionamiento con este sistema, por lo que podemos resumir que los alcances logrados, y que se enlistan a continuación, fueron más que satisfactorios:

- La implementación de un sistema de monitoreo, para un simulador físico de un medio sin fricción, para visualizar la orientación en tres ejes, tanto en el procedimiento de balanceo, como en las pruebas de control de orientación de satélites.

- El desarrollo un circuito de comunicación inalámbrica, unidireccional, para una interfaz RS232.

- El establecer un procedimiento para lograr el balanceo estático de la plataforma situándonos a ±9 grados respecto a la horizontal.

- El balancear de manera automática el simulador, equilibrando la plataforma en el intervalo de ±1 grado, lo que significa un par residual de 1.2066×10^{-3} [N m].

- La graficación y el almacenamiento de los datos de orientación de la plataforma a una tasa de 12 lecturas por segundo.

Capítulo

1

INTRODUCCIÓN

Los satélites tienen dos clasificaciones principales, que son: de observación y de comunicaciones. Los de observación básicamente vigilan el medio ambiente de la Tierra, éstos pueden reconocer y explorar áreas que son prácticamente inaccesibles por otros medios; los de comunicaciones se usan para la transmisión, distribución y diseminación de la información desde diversas ubicaciones de la Tierra.

Existe una clase de satélites pequeños que están siendo desarrollados en universidades y centros de investigación en varios países: los microsatélites, que son de bajo peso, bajo costo de construcción y lanzamiento, y son generalmente de tipo experimental. Éstos constituyen una oportunidad de desarrollo de equipo espacial a bajo costo, y además presentan una serie de ventajas muy interesantes, ya que es posible llevar a cabo en ellos una gran diversidad de experimentos. Aunque dicha diversidad y complejidad se ve reducida al no contar con un sistema que permita mantener la orientación de la nave de manera continua. Un sistema de orientación y control, permite ampliar de manera significativa la cantidad de experimentos que es posible llevar a cabo en órbita terrestre a bordo de estas naves.

El desarrollo de sistemas de detección y control de orientación para satélites pequeños, es llevado a cabo en el Laboratorio de Percepción Remota Alternativa y Tecnología Avanzada del Instituto de Geografía de la UNAM, donde se cuenta, entre otra infraestructura, con un simulador físico de un medio sin fricción, el cual permite emular una de las condiciones orbitales más importantes en el desarrollo de estos sistemas: la falta de fricción.

En este proyecto de tesis se diseñará e implementará un sistema de control de lazo cerrado, para efectuar el balanceo estático, de manera automática, a esta plataforma de simulación para control de orientación de satélites pequeños. Dicha plataforma debe ser balanceada de manera adecuada, de otra manera puede sufrir una caída súbita y obstaculizar los experimentos. La calidad de éstos está directamente relacionada con la precisión del balanceo, ya que éste permitirá que se lleven a cabo evaluaciones de una manera realista y objetiva, del desempeño de sensores, actuadores y algoritmos que serán incorporados posteriormente en satélites artificiales.

A diferencia de la metodología utilizada con anterioridad, donde se llevaba a cabo un balanceo dinámico, haciendo uso de varios sensores y resolviendo un sistema de ecuaciones de Euler; se ha establecido que el sistema queda balanceado estáticamente, cuando éste se encuentra perfectamente horizontal, ya que esto significa que no existe ningún par actuando fuera del centro de masa del conjunto. Para lograr esto, se pretende desarrollar un sistema de control que permita equilibrar de manera automática la plataforma de simulación en los ejes de alabeo (X) y cabeceo (Y), dentro de un intervalo de $\pm$ 1° con respecto a la propia horizontal. Como actuadores del sistema de balanceo se emplean dos

masas deslizantes, que pueden efectuar un recorrido a lo largo de cada uno de los ejes mencionados.

Un sistema de monitoreo es necesario para supervisar los movimientos de la plataforma durante la fase de balanceo y durante las pruebas de control de orientación, por lo que se ha desarrollado la programación necesaria para desplegar en pantalla (y así tener una referencia rápida de lo que está ocurriendo con la orientación de la plataforma) y también para almacenar en un archivo de datos la orientación.

El balanceo es un parámetro muy crítico en un medio sin fricción, por esta razón, no es permisible enviar la información de la orientación por medio de cables; ya que se produciría un desbalanceo inadmisible en la plataforma. Debido a esto, la comunicación entre ésta y la computadora, donde se despliega y almacena la información de la orientación, se lleva a cabo a través de una interfaz RS232 inalámbrica. Una brújula electrónica, equipada con dos inclinómetros, es el sensor de orientación que proporciona los ángulos de desviación en los tres ejes, tanto en la fase de balanceo, como cuando se realicen las pruebas de control de orientación.

1.1. PLATAFORMA DE SIMULACIÓN.

Para poder llevar a cabo el desarrollo y las pruebas de funcionamiento de los sistemas de control de orientación de satélites, es necesario contar con un simulador que nos permita tener grandes momentos de inercia, movimiento angular en los tres ejes y un equilibrio neutral bajo cualquier ángulo de deflexión.

El diseño básico de este tipo de simuladores, consiste de una plataforma móvil donde se colocarán los componentes de los sistemas de control, suspendida sobre un soporte que permita el movimiento en los tres ejes de rotación con fricción despreciable. Esto nos lleva a la utilización de un balero de aire esférico como una muy buena solución para soportar a la plataforma.

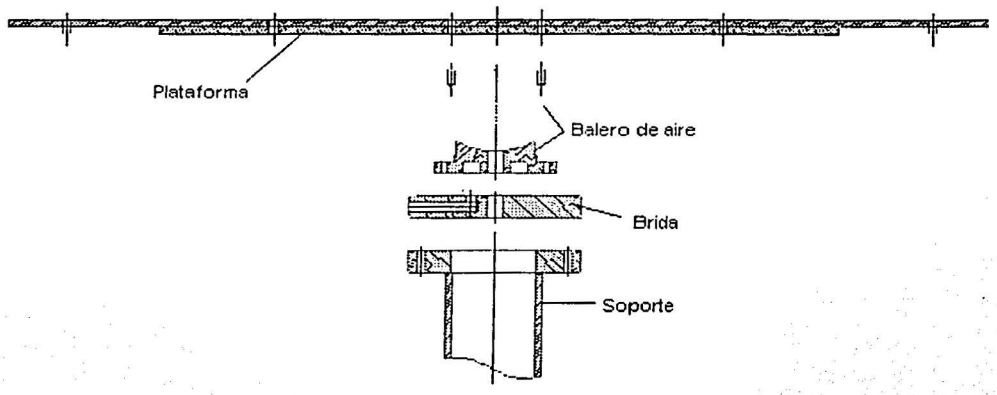


Figura 1.1 Diagrama general de la plataforma de simulación donde se muestran los componentes principales.

El simulador debe ser totalmente autónomo, es decir, la energía para funcionar debe venir de él mismo, y la transmisión de información sólo es permisible a través de señales electromagnéticas. En la figura 1.1 se muestra el esquema general de la plataforma de simulación completa; sus principales componentes son: mesa o plataforma, balero de aire esférico, brida para suministro de aire y soporte.

El simulador utilizado tiene una plataforma de aluminio, un balero de aire esférico multiflujo, con diámetro de 10 [cm], maquinado en bronce SAE 62 con un acabado muy fino y con tolerancias muy cerradas, con una capacidad de carga de 80 [kg] y que genera un medio con fricción despreciable.

Para facilitar la operación de este tipo de equipos, el satélite completo no debe colocarse en la plataforma, solamente el sistema de control de orientación con el tensor de inercia del satélite reproducido o escalado.

1.1.1. Medio sin fricción.

Los instrumentos y sistemas de control diseñados para mantener estabilizadas las naves espaciales, deben ser probados exhaustivamente en Tierra, en un medio ambiente simulado. Como el medio ambiente espacial no puede ser completamente simulado, y debido a que lo más importante es tener un medio sin fricción, fue necesario encontrar una manera de generarlo. Un balero de aire es un dispositivo adecuado para crear esta condición, ya que nos permite probar instrumentos y sistemas de control de orientación de vehículos espaciales, debido a que los efectos por fricción, pueden ser considerados como nulos.

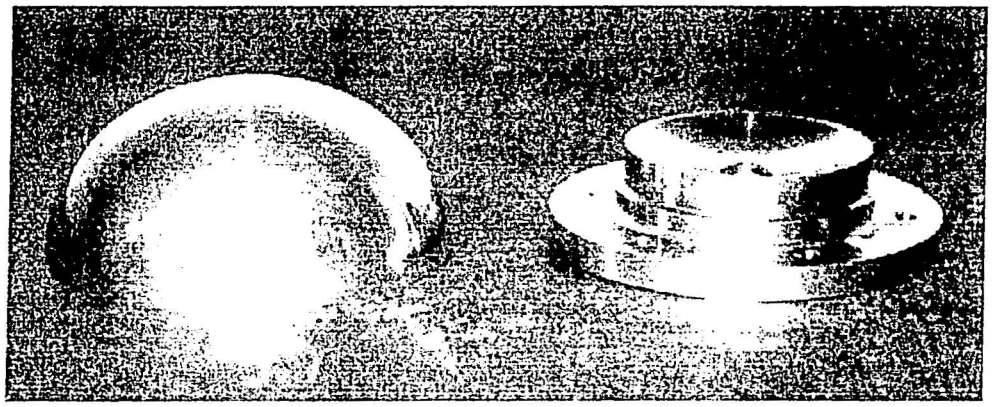


Figura 1.2 Semiesfera y copa, partes fundamentales de un balero de aire esférico.

Un balero de aire esférico, consiste básicamente de una semiesfera y una copa; es la unión de estos dos componentes lo que produce el colchón de aire que provee del medio sin fricción (figura 1.2). El aire es introducido por la parte baja de la copa y sale por la periferia, es decir, por su parte superior. La semiesfera se sujeta firmemente a la plataforma y se completa el sistema que simula un medio sin fricción.

1.1.1.1. Baleros de aire esféricos multiflujo.

Un balero esférico multiflujo, es utilizado en el simulador; es llamado de esa manera debido a que el aire fluye hacia el interior de la copa a través de seis orificios o tubos capilares y sale por la periferia de la esfera y por la perforación ubicada en el centro de la copa. Ver figura 1.3

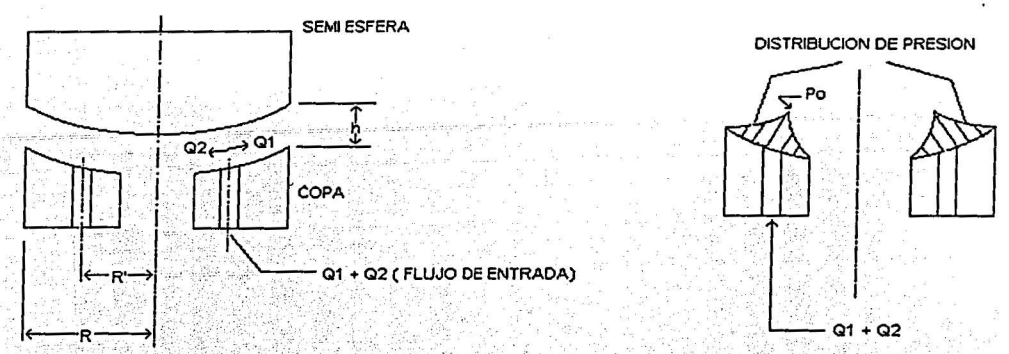


Figura 1.3 Esquema de un balero de aire multiflujo y perfil de distribución de presión.

Una de las principales ventajas de utilizar baleros esféricos multiflujo, a diferencia de los monoflujo (con una sola perforación capilar en la copa y salida de aire por la periferia) es que provee mayor estabilidad al conjunto.

Una característica importante de los baleros de aire, es el hecho de necesitar pequeñas presiones y gastos de aire para soportar una carga dada, aunque es necesario colocar filtros para agua y aceite, con el fin de mantener en condiciones adecuadas el medio sin fricción.

1.1.2. Balanceo de la plataforma.

Uno de los aspectos más importantes a considerar para la utilización práctica de este simulador, en pruebas de detección de orientación y control de estabilización, es que el sistema completo (incluyendo los componentes de los subsistemas de detección de orientación y control de estabilización, baterías y demás sistemas bajo prueba) debe ser sometido a un procedimiento que permita tenerlo perfectamente balanceado, primero en forma manual y luego en modo automático; particularmente éste último, ya que sin él, las pruebas de control de orientación no pueden realizarse de manera adecuada; debido a que el simulador se precipita súbitamente hacia abajo, debiendo pivotearlo en dos puntos, lo que restringe los movimientos a un solo eje. Esto introduce por supuesto fricción, lo que no permite cuantificar correctamente los resultados, estando éstos muy poco cercanos a la realidad.

1.2. BRÚJULA ELECTRÓNICA.

Como sensor de orientación, se emplea una brújula electrónica, ésta nos sirve para determinar la orientación en los tres ejes ortogonales de la plataforma. Los ejes X y Y se localizan sobre el plano de la misma, mientras que el eje Z es perpendicular a dicho plano, con dirección positiva hacia arriba. La brújula proporciona de manera continua la inclinación de los ejes de alabeo (X) y cabeceo (Y), y la desviación del eje de guiñada (Z). Los dos primeros son usados durante el proceso de balanceo, mientras que para la realización de las pruebas de control, se utilizan los tres ejes.

La brújula consiste de dos inclinómetros (para los ejes X y Y) y de un magnetómetro (para el eje Z) y está contenida en un módulo electrónico denominado EZ-COMPASS-3, cuyo diseño está basado en un microcontrolador programado por el fabricante. La comunicación con este dispositivo se realiza en forma serie, bajo el estándar RS232. Al microcontrolador de este módulo se accesa mediante comandos suministrados desde el teclado de la computadora, donde ha sido programada una rutina de comunicación serie, para este caso.

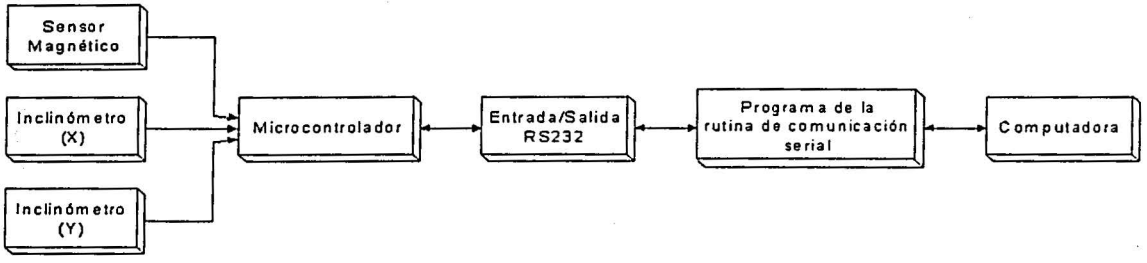


Figura 1.4 Diagrama de bloques de la brújula electrónica y su interfaz con la computadora.

1.2.1 Magnetómetro e inclinómetros.

El módulo EZ-COMPASS-3 utiliza un magnetómetro de estado sólido de tres ejes, con 12 bits (0.08 grados) de resolución, para determinar la orientación con respecto a las 3 componentes vectoriales del campo magnético terrestre y generar el azimut de 0 a 360 grados.

Contiene dos inclinómetros con 12 bits de resolución, que determinan la desviación de los ejes de alabeo y cabeceo en un intervalo de ±70 grados.

1.2.1.1 Campo magnético terrestre.

La utilización del campo magnético sufrió un cambio considerable cuando Karl Gauss inventó un magnetómetro para medir su valor absoluto, esto lo llevó a cabo en varios lugares alrededor de la Tierra. Así desarrolló la instrumentación para medir y la lectura fue reducida a un sólo número para representar la intensidad del imán terrestre, conocido también como "momento magnético". Dicha medición se hizo por primera vez en 1835, a partir de ésta y hasta nuestros días se ha continuado haciendo, observándose que su intensidad ha ido disminuyendo.

Se sabe que un imán de barra tiene un polo norte y un polo sur, se entiende por que una aguja imantada se utiliza como brújula, ya que uno de sus extremos buscará apuntar hacia el polo norte geográfico de la Tierra.

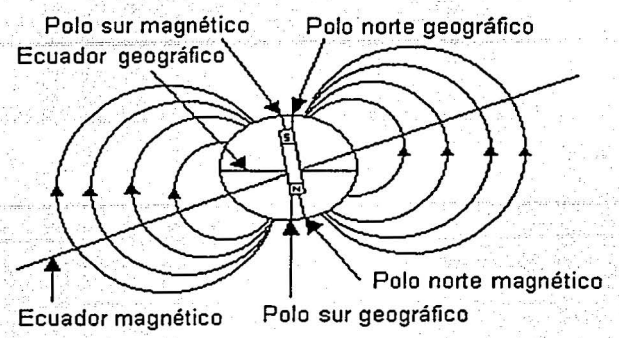


Figura 1.5 Una aguja imantada se alinea con el campo magnético terrestre.

De lo anterior se concluye que el polo norte magnético está localizado cerca del polo sur geográfico, y que el polo sur magnético se encuentra cerca del polo norte geográfico, es decir su configuración sería como la mostrada en la figura 1.5.

Aun cuando el patrón del campo magnético de la Tierra es similar al que tendría una gigantesca barra de imán colocada en su centro, es fácil entender que la fuente del campo magnético no es una gran masa de material magnetizado permanentemente. La Tierra tiene grandes depósitos de hierro en las profundidades de su interior, pero las altas temperaturas en su núcleo hacen suponer que el hierro no retiene ninguna magnetización permanente. Si se considera este hecho con más detenimiento, se verá que la fuente verdadera, la constituyen las corrientes convectivas de carga que existen en su núcleo.

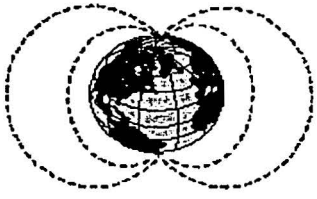


Figura 1.6 Líneas del campo magnético terrestre.

La circulación de iones o electrones en el líquido interior pudieran producir un campo magnético, tal como una corriente en una espira de alambre produce un campo magnético.

La principal fuente de perturbación del campo magnético es el sol, que emite constantemente un plasma neutro llamado viento solar. Éste distorsiona al campo geomagnético, particularmente a grandes alturas (a partir de 8 o 10 radios terrestres). En

este punto, el plasma se rompe y algunas de las partículas cargadas son atrapadas por el campo magnético y empiezan a circular entrando y saliendo por los polos.

La radiación solar incluye todas las ondas electromagnéticas emitidas por el sol, desde los rayos X hasta ondas de radio; ambos, radiación solar y viento solar, producen fuerzas perturbadoras que afectan la orientación, la órbita y las comunicaciones de las naves espaciales y satélites.

Existen en la naturaleza algunas rocas como la magnetita pura (imán de piedra) que distorsionan el campo, haciendo erróneas las lecturas de las brújulas. Los objetos ferrosos y los campos magnéticos que se encuentran en la cercanía de nuestra plataforma, como el cableado eléctrico, la compresora que suministra el aire y la computadora, también afectan al campo magnético. Es por eso que la brújula electrónica debe calibrarse antes de su utilización para compensar la influencia de anomalías magnéticas localizadas en la proximidad del módulo. No es posible compensar contra interferencias magnéticas variables, sin embargo el módulo las descubre y las suspende o intenta corregirlas automáticamente durante su ocurrencia. Esto se explica con mayor detalle en el capítulo 2.

1.3. BALANCEO AUTOMÁTICO.

El proceso de balanceo de la plataforma resulta particularmente difícil de realizar debido a la complejidad de los componentes que se utilizan en las diferentes pruebas. El problema radica en que dichos componentes, la mayoría de las veces, no tienen una forma geométrica regular o que su centro de masa no se puede determinar fácilmente. Esto causa que a medida que aumenta la cantidad de componentes el proceso se vuelva cada vez más tedioso y tardado. El balanceo manual es indispensable para localizar la plataforma a ± 9 grados de la horizontal, para que entonces entre en funcionamiento el balanceo automático y se lleve a la plataforma a la horizontal con precisión de ± 1 grado. El balanceo automático de la plataforma, es de gran importancia, ya que elimina una tarea aburrida y tardada, además que en un futuro puede ser utilizado para hacer un balanceo dinámico, en caso de ser necesario. Una vez equilibrada la plataforma se podrán realizar una gran variedad de pruebas de orientación, es decir, utilizar en ella diferentes métodos de control de orientación como pueden ser: par magnético, toberas de reacción, ruedas inerciales o sistemas combinados.

1.3.1. Determinación del centro de masa.

Las partes en rotación pueden, y generalmente deben, ser diseñadas como inherentemente equilibradas por su configuración geométrica. Sin embargo, las variaciones debidas a las tolerancias en su manufactura hacen que haya algún pequeño desequilibrio (o desbalance) en cada una de ellas. Así que en cada parte debe ser aplicado un procedimiento de equilibramiento después de su fabricación. La magnitud y localización de cualquier desbalanceo pueden ser determinadas con bastante exactitud, y compensadas al agregar o quitar material en las ubicaciones correctas.

Las piezas maquinadas cuidadosamente tienen mayor probabilidad de estar mejor balanceadas que las piezas fundidas. En muchos casos es más económico permitir cierto desbalanceo durante la manufactura y balancear la pieza agregando o quitando material.

Existen disponibles comercialmente, máquinas para balancear, que permiten procesar piezas a tasas de producción masiva.

Con frecuencia es conveniente que un objeto de forma complicada se modele como varios objetos interconectados de formas más sencillas, y cuyas configuraciones individuales permitan calcular fácilmente sus masas y las ubicaciones de sus centros de gravedad (CG) o centros de masa propios. El CG del conjunto se puede localizar después, a partir de los primeros momentos de las partes simples, igualados a cero.

1.3.2. Reubicación del centro de masa.

La reubicación del centro de masa (CM), en la plataforma de simulación, se realizará por medio de un sistema de masas deslizantes. Dichas masas llevan a cabo el ajuste fino de localización del CM. Se cuenta con dos actuadores, cada uno de ellos consiste en una platina de aluminio que tiene una masa de 615 gramos que se mueve por medio de un motor de pasos, a lo largo de una varilla roscada, milimétricamente. El motor de pasos al recibir una señal de 4 pulsos, desplaza la platina una distancia de 0.005 [mm] completando un paso; el desplazamiento máximo de cada unidad es de 5 [cm], por esta razón es indispensable llevar a cabo un balanceo estático manual, previo al balanceo automático, para situarnos dentro de este intervalo.

1.3.3. Motores de pasos.

Los motores a pasos son capaces de lograr un posicionamiento de muy alta precisión sin la necesidad de usar complicados y costosos dispositivos de realimentación, aunque éstos pueden ser incorporados si se desea la comparación de posición. Debido a su bajo costo y tamaño reducido, en comparación con otros tipos de motores, pueden ser usados en sistemas neumáticos, hidráulicos y sistemas de servomotores.

Características de los motores de pasos:

- Libres de escobillas y con imán permanente.
- Convierten pulsos eléctricos en movimientos rotacionales discretos
- Pueden operar a incrementos de pasos completos o de medio paso.
- Ofrecen una precisión de ±3% y ±5% no acumulativa.
- No son muy rápidos en términos de revoluciones por minuto, en comparación con otros tipos de motores
- Pueden dar torques de 60 hasta 5330 [oz-in] (42.4 a 3764 [N-cm]).
- Siempre necesitan de un circuito especial externo para controlarlos, debido a que no se le puede conectar directamente a una fuente de alimentación
- Fácilmente adaptables a diferentes tipos de control, incluyendo sistemas basados en microcontroladores (este punto resultó de gran importancia para su selección).
- Pueden operar a temperaturas de -40 a +40 [°C].
- Tienen rodamientos libres de mantenimiento (lubricados de por vida).

1.3.4. Sistema de control realimentado.

El sistema de control realimentado para llevar a cabo el balanceo de la plataforma, se hace por medio de un microcontrolador de la familia MC68HC11 de Motorola, se utiliza el modelo MC68HC11F1, el cual se encarga de recibir en forma serial la información de la brújula electrónica y compararla contra el umbral de ±1° en los ejes de alabeo y cabeceo; del resultado obtenido de dicha comparación, el microcontrolador manda una secuencia de pulsos, los cuales hacen que los motores de pasos giren en sentido horario o anti-horario, lo cual induce el movimiento a las masas deslizantes en una dirección u otra. Cada eje se maneja de manera independiente; es decir, primero se ajusta el eje X y posteriormente el Y, hasta lograr que la plataforma esté totalmente horizontal.

Cada 4 pulsos (una secuencia), que manda el microcontrolador corresponde a cuatro pasos en el motor, lo que significa un movimiento de 0.02 [mm] en las masas deslizantes.

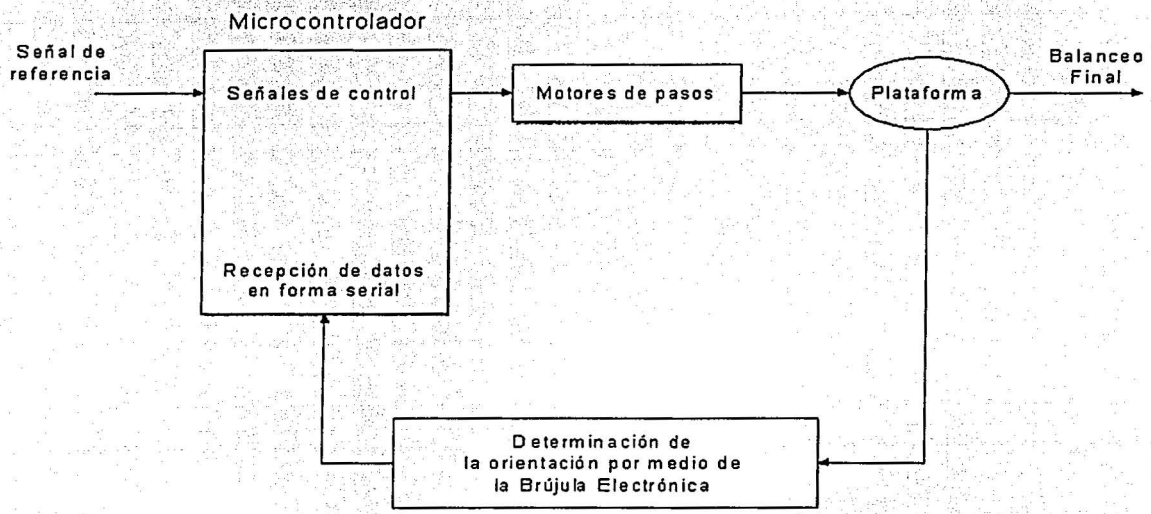


Figura 1.7 Diagrama de bloques del sistema de control realimentado para balanceo de la plataforma.

1.4 INTERFAZ INALÁMBRICA.

Como se ha mencionado, es necesario incluir un sistema de comunicaciones inalámbrico para la transmisión de información entre la plataforma y el exterior, debido a que el cableado induce un desbalanceo. La transmisión es unidireccional, es decir, desde la brújula (ubicada en el origen de los ejes ortogonales de la plataforma) hacia la computadora de escritorio, donde se despliega y almacena la información de orientación. La secuencia de datos enviados por medio de la interfaz RS232 de la brújula, es modulada por un circuito llamado HP Serie II, que ofrece un método eficaz de transmisión inalámbrica por medio de señales de

radio frecuencia. Permite transmitir datos analógicos y digitales en un intervalo de frecuencias que van de 902 a 928 [MHz] utilizando modulación FSK. En condiciones normales puede transmitir la información hasta una distancia aproximada de 100 metros.

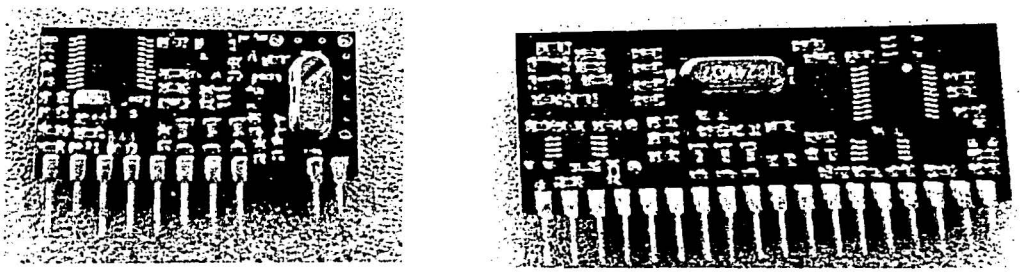


Figura 1.8 Circuitos de transmisión / recepción.

1.5 SISTEMA DE MONITOREO

La brújula electrónica proporciona información sobre la orientación de la plataforma en los tres ejes. Se ha establecido que es muy conveniente estar visualizando directamente la orientación de la plataforma, por lo que es necesario realizar un despliegue en pantalla. Para ello se utiliza un programa de graficación en la PC como herramienta de despliegue de datos y simulación de orientación.

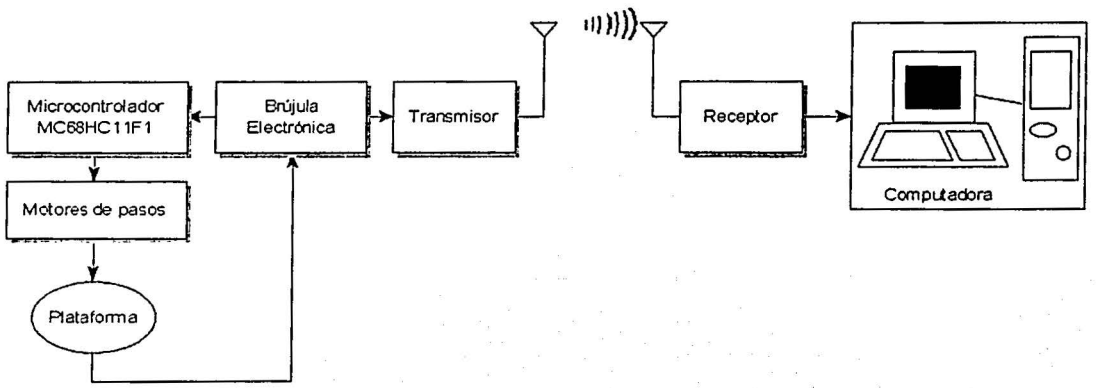


Figura 1.9 Diagrama de bloques del sistema de monitoreo.

El sistema de monitoreo y el programa de graficación, presentado en este trabajo, es una herramienta utilizada como medio auxiliar para analizar el comportamiento del sistema, tanto en la fase de balanceo, como durante las pruebas de control de orientación. Para comprender su funcionamiento, en el capítulo 5 se hace una descripción detallada de su funcionamiento.

Capítulo

2

BRÚJULA ELECTRÓNICA.

En este capítulo se hace una descripción de las características de la brújula electrónica, que es la herramienta de detección de la orientación de la plataforma y que nos sirve para determinar el movimiento de guiñada (Z) y la inclinación en los ejes de alabeo (X) y cabeceo (Y).

2.1. CARACTERÍSTICAS DE CONSTRUCCIÓN Y DESEMPEÑO.

El funcionamiento del eje de guiñada de la brújula, está basado en la medición del campo magnético terrestre. Éste es producido por un dipolo magnético extremadamente poderoso y de dimensiones muy reducidas, ubicado en las proximidades del centro de la Tierra.

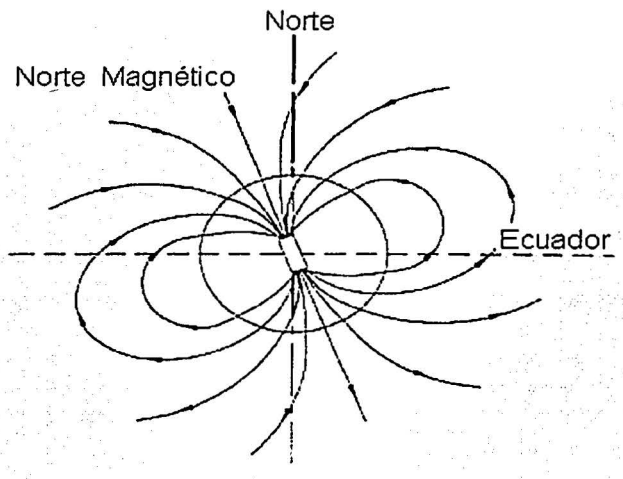


Figura 2.1 El campo magnético terrestre

Existe una red magnética terrestre de líneas isógonas e isóclinas que son las que determinan la orientación de la brújula. Las isóclinas son una red de líneas con la misma inclinación, que corresponderían a los paralelos geográficos; y las isógonas se forman al unir todos los puntos con el mismo ángulo de declinación entre las líneas de fuerza y los meridianos; sobre la superficie terrestre. Estas líneas nos permiten definir dos puntos ficticios llamados polos magnéticos.

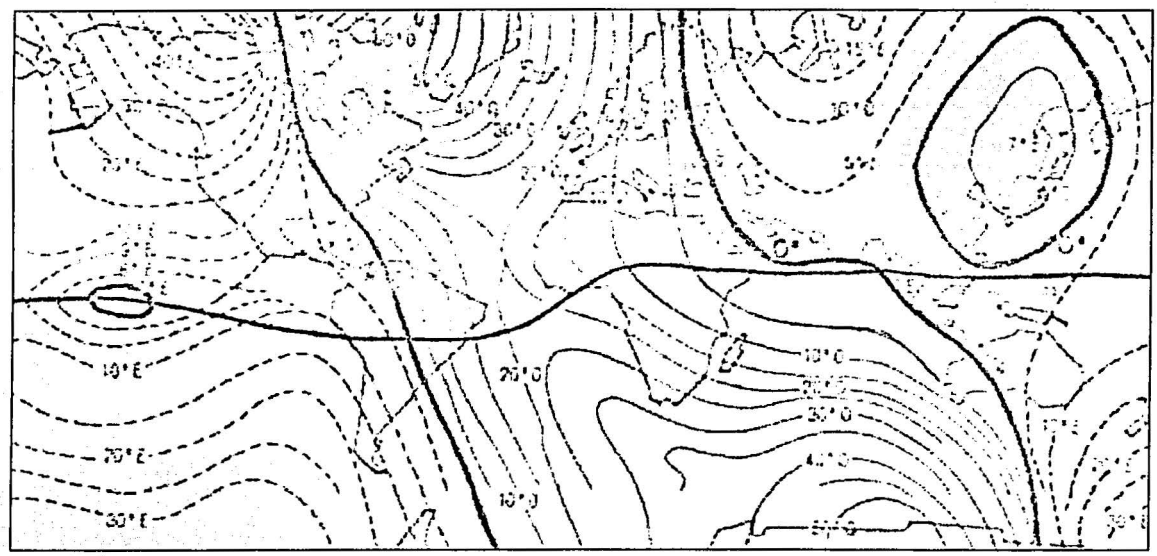


Figura 2.2 Líneas isóclinas, correspondientes a los paralelos geográficos, e isógonas, correspondientes a los meridianos.

Se estableció en un principio, que las características del campo magnético terrestre son una muy buena alternativa para determinar la desviación en el eje Z, por esta razón se decidió emplear una brújula electrónica. Se localizó en el mercado, un módulo constituido por un magnetómetro que proporciona los vectores que proveen la dirección y la magnitud del campo magnético, y dos inclinómetros extremadamente sensibles que sirven para medir la inclinación en dos ejes perpendiculares entre sí. En un mismo módulo se podía contar con todos los sensores necesarios para la determinación de la orientación de la plataforma en tres ejes.

Las principales características por las que utilizamos la brújula electrónica EZ- COMPASS-3 es la precisión, el bajo costo, el tamaño reducido y el hecho de que utiliza el protocolo de comunicación RS232, que nos permite el intercambio de información tanto con la computadora donde se realiza el monitoreo, como con el microcontrolador del sistema de balanceo con masas deslizantes, de una manera muy sencilla.

En la siguiente figura se muestra el módulo EZ-COMPASS-3, donde es posible apreciar dos paralelepípedos colocados perpendicularmente, correspondientes a los inclinómetros. El magnetómetro se encuentra sobre la misma cara de la tarjeta en un circuito integrado.

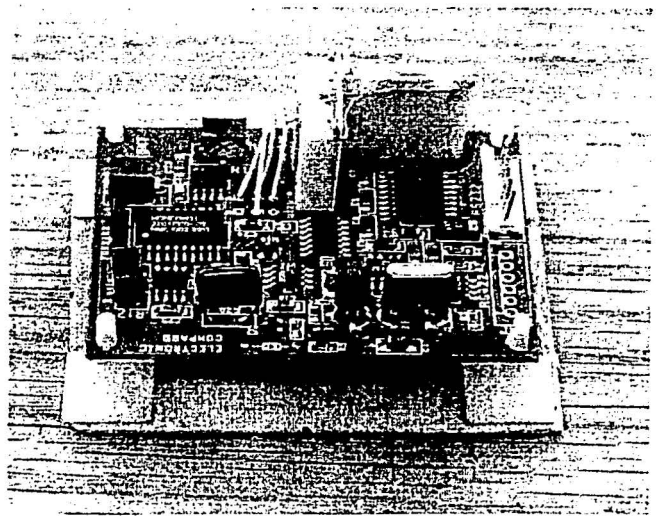


Figura 2.3 Módulo EZ-COMPASS-3 que consta de un magnetómetro y de dos inclinómetros.

La siguiente tabla contiene las características técnicas del módulo EZ-COMPASS-3:

Azimut	0 – 360 grados
Cabeceo	± 70 grados
Alabeo	± 70 grados
Resolución de azimut	12 bits
Resolución de inclinación	12 bits, escala completa, ambos ejes
Precisión de azimut	< 0.5 grados
Repetibilidad de inclinación	< 0.02 grados
Campo magnético	± 2 Gauss, máximo
Resolución magnética	< 1 mGauss
Polarización	5 VCD
Temperatura	-30 a +85 °C
Peso	< 45 gramos
Calibración	EEPROM no volátil
Comunicación	300 – 38400 bauds
Formato de salida	Serial RS232
Actualización máxima	5 veces/seg

Tabla 2.1 Características principales de la brújula electrónica.

Una ventaja importante del módulo, es que nos permite hacer una calibración en la respuesta de la brújula para la medición del eje Z. El módulo EZ-COMPASS-3 incluye compensación contra la influencia de anomalías magnéticas localizadas en su proximidad,

como por ejemplo, cuando éste se instala en las cercanías de una estructura de metal. En el balanceo de la plataforma sólo utilizamos los ejes de alabeo y cabeceo, pero para la graficación de los datos durante pruebas de orientación, utilizamos también el eje de guiñada, por lo que es necesario calibrar su respuesta.

Durante el procedimiento de compensación, se debe hacer girar el módulo 360 grados (horario o anti-horario) en el eje de guiñada a intervalos de 5 o 10 grados, mientras la brújula graba la información magnética estando presentes los objetos y las condiciones que influyen de alguna manera en las mediciones de campos magnéticos a su alrededor, y de esta forma ignora la influencia que éstos puedan tener en las lecturas reales, tomadas durante el funcionamiento normal de adquisición de la orientación en dicho eje.

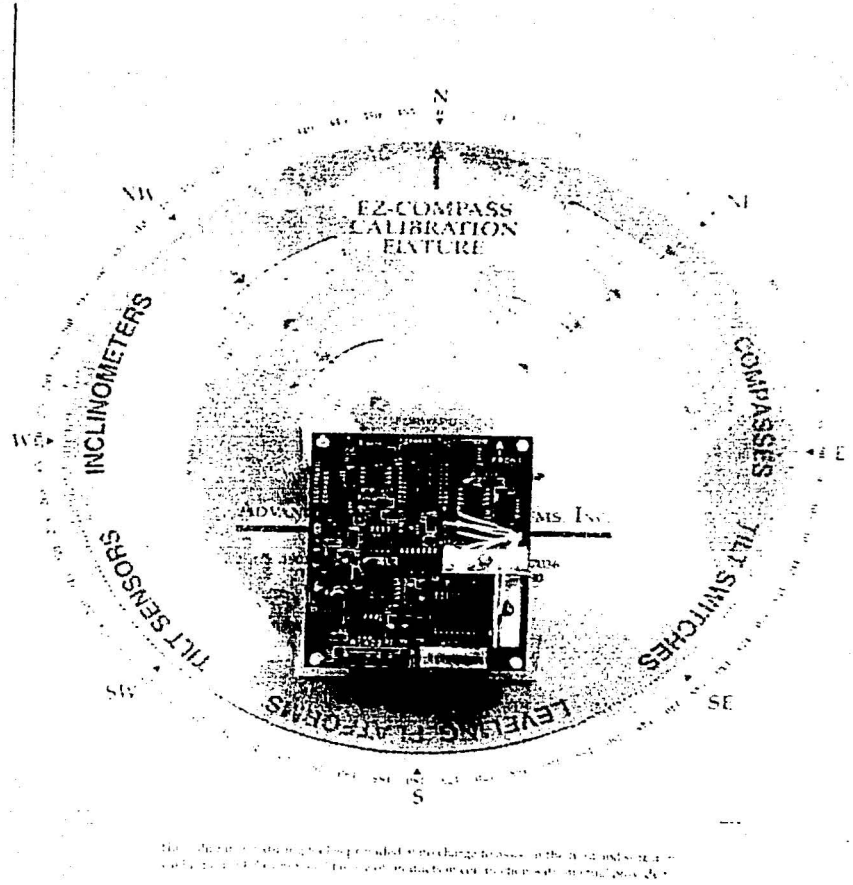


Figura 2.4 Diagrama de calibración contra anomalías, de la brújula electrónica.

El módulo utilizado para la orientación de la plataforma, envía datos de manera continua durante todo el proceso de balanceo y operación normal durante los experimentos de orientación. La siguiente figura muestra el diagrama de flujo correspondiente a la rutina de comunicación con este dispositivo.

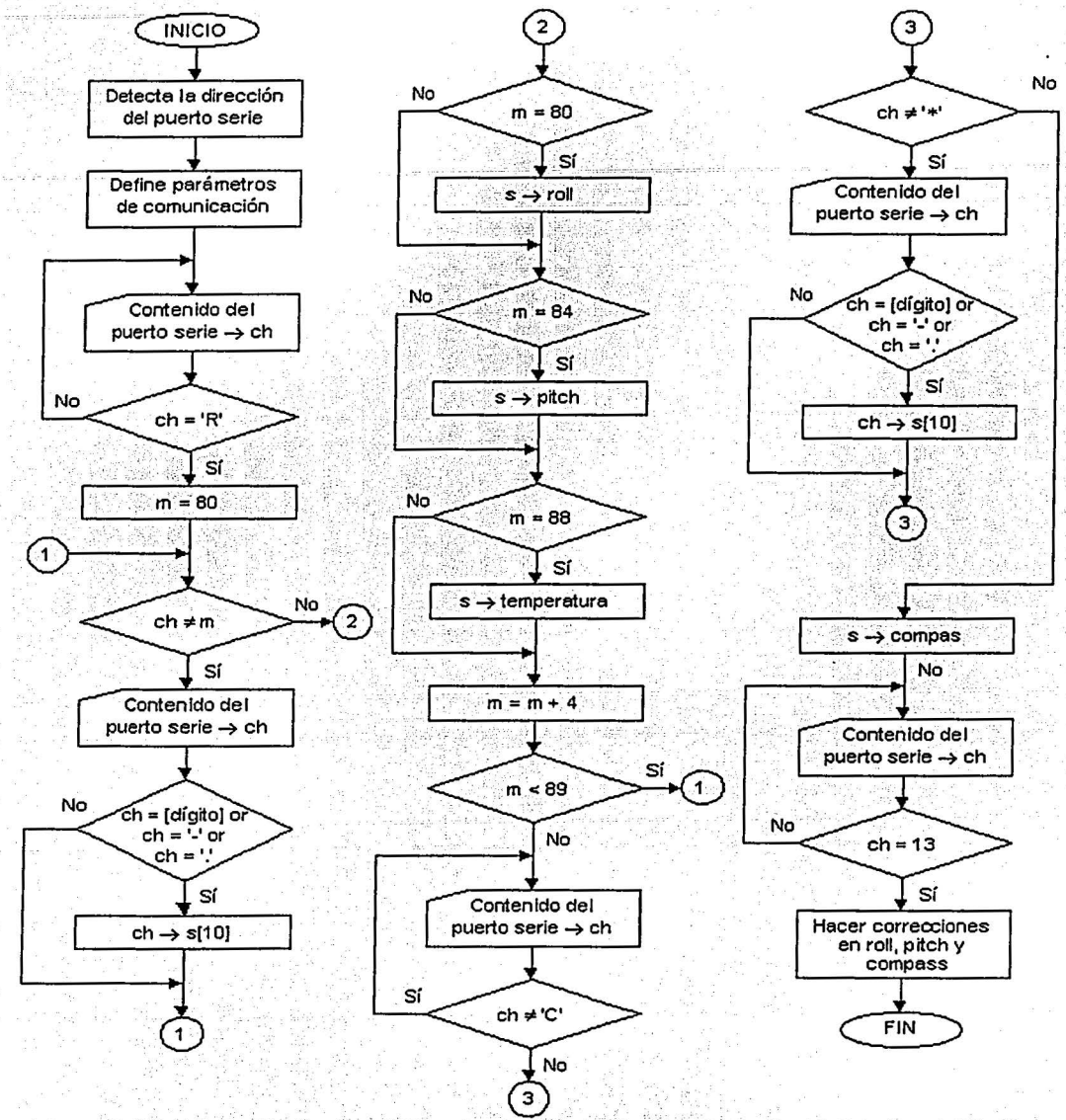


Figura 2.5 Diagrama de flujo para la comunicación con el EZ-COMPASS-3.

El módulo transmite cadenas de caracteres ASCII a una velocidad de 9600 [bauds], y es capaz de enviar datos de manera continua y autónoma mediante los siguientes comandos:

COMANDOS:	
go	Despliega la línea de datos de manera continua
h	Detiene el comando "go"
calib	Comienza a calibrar contra la influencia de anomalías magnéticas
save	Guarda los cambios después de modificar los comandos
FIJAR COMANDOS:	
b=[1-7]	Selección del baudaje [1=300, 2=1200, 3=2400, 4=4800, 5=9600, 6=19200, 7=38400]
cont=[d-e]	Habilita o deshabilita el modo continuo [d=deshabilita, e=habilita]
sdo=[modo]	Modo [a=aosi, t=ttcm, n=nmea, l=LCD]; modo usado: ttcm
e=[datos]=[d-e]	Habilita o deshabilita datos del comando "go" [d=deshabilita, e=habilita]
DATOS:	
r	Roll (alabeo)
p	Pitch (cabeceo)
c	Compass (guiñada)
t	Temperatura
x	Componente magnética X
y	Componente magnética Y
z	Componente magnética Z
CONSULTA DE COMANDOS:	
b?	Baudaje
cont?	Verifica el modo de continuidad
c?	Azimut
p?	Pitch
r?	Roll
s?	Muestra la misma línea de datos que "go", pero sólo una vez
t?	Temperatura
LÍNEA DE DATOS:	$R-2.61P1.08T21.5X206.7Y95.4Z367.5C43.9*4d
$	Caracter de inicio
*	Caracter de fin
4d	Suma de comprobación
R,P,C	Roll, pitch y azimut
T	Temperatura
X,Y,Z	Componentes magnéticas

Tabla 2.2 Comandos principales de la brújula electrónica.

2.2. MAGNETÓMETROS.

Los magnetómetros son usados como sensores de orientación en muchas aplicaciones. En las aeronaves o en las naves espaciales son particularmente útiles por muchas razones: proporcionan los vectores que proveen la dirección y la magnitud del campo magnético, son

confiables y tienen poco consumo de energía, operan sobre un intervalo bastante amplio de temperatura y no poseen partes móviles.

También son utilizados en otras aplicaciones como en la minería y la exploración petrolera.

Un magnetómetro consta de 2 partes: un transductor magnético y una unidad electrónica que transforma las medidas del transductor en un formato adecuado.

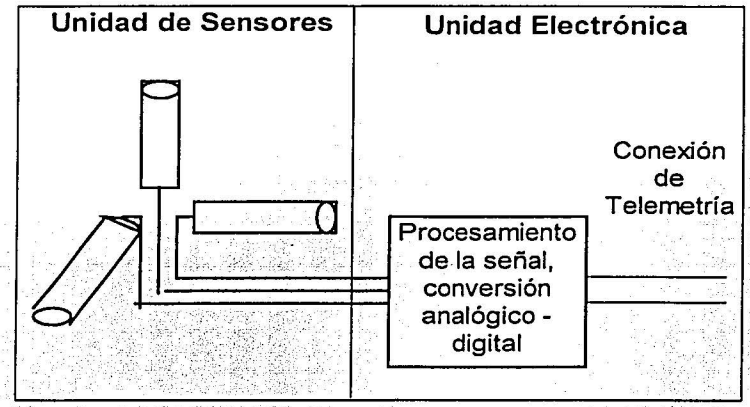


Figura 2.6 Diagrama de bloques de un magnetómetro para aplicaciones aeroespaciales.

2.3. INCLINÓMETROS.

Los inclinómetros son sensores extremadamente sensibles que sirven para medir la inclinación. Miden el ángulo de un objeto con respecto a un eje horizontal o vertical. Están formados por un electrolito (líquido conductor) situado en un recipiente en el cual hay introducidos dos electrodos de platino, ambos con una parte fuera del electrolito. Cuando el sensor se inclina, uno de los electrodos entra más en contacto con el electrolito y el otro menos. Si se miden las corrientes de salida de los electrodos, es posible determinar el ángulo de inclinación. Algunas aplicaciones típicas son: mediciones de pendientes de carreteras, alineamiento de túneles, montaje de armamento, control de inclinación en trenes, etc.

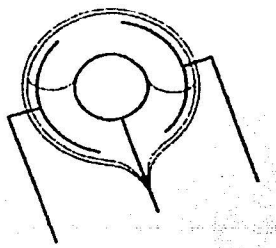


Figura 2.7 Principio de funcionamiento del inclinómetro.

Los inclinómetros entregan generalmente sus lecturas en grados directamente por medio de un programa de comunicación con una interfaz para micro-computadora.

2.4 INTERFAZ DE COMUNICACIÓN.

La brújula electrónica se comunica con la computadora mediante el protocolo de comunicación serie RS232.

El envío de datos a través de un puerto serial, se usa normalmente para efectuar comunicaciones asíncronas, es decir, cuando el transmisor y el receptor no necesitan coordinarse para la transmisión de la información. Esta forma de comunicación es muy útil en aplicaciones donde se transmiten datos ocasionalmente. El puerto serie de la PC es un dispositivo asíncrono y es compatible con el estándar RS-232. El formato de transmisión de datos asíncrono en forma serial del módulo, consiste de cuatro partes: el bit de inicio, los ocho bits de datos, sin bit de paridad y un bit de paro. El envío de datos se hace sobre caracteres de 8 bits =1 byte (equivalente a un caracter en código ASCII), enviando un caracter a la vez.

Hay dos tipos de paridad que se usan: el primero, *marca,* donde el bit de paridad que se intercala siempre es un uno; y el segundo, *espacio*, donde el bit que se intercala siempre es un cero.

El bit de paridad es seguido de un bit de paro, que indica cuando no se envía más información. Bajo esta norma se manda *marca* (1 lógico) y el inicio de la información es un *espacio*; si hay un espacio largo, indicación de un evento especial, se tiene un rompimiento (break). El bit de más peso generalmente es de paridad, pudiendo ser par, impar, marca, espacio o sin paridad, como es en este caso.

Dado que la condición ociosa envía *marca*, para reconocer un dato se envía primero un bit de inicio en *espacio* y para finalizar bits de paro en *marca*, que pueden ser 1, 1.5 o 2. La marca permite la detección de error, mediante el bit de paridad, pero no de corrección de errores.

Cada bit permanece durante cierto tiempo en el que debe ser reconocido, el tiempo depende de la tasa de transmisión que ambos dispositivos acuerden, las tasas normalizadas de ambos pueden ser 300, 1200, 2400,4800, 9600, 19200 y 38,400 [bauds] (1 baud = 1bit/seg); nosotros estamos transmitiendo los datos de la brújula a 9600 [bauds]; estas tasas de transmisión incluyen sólo los bits de información y no los de control.

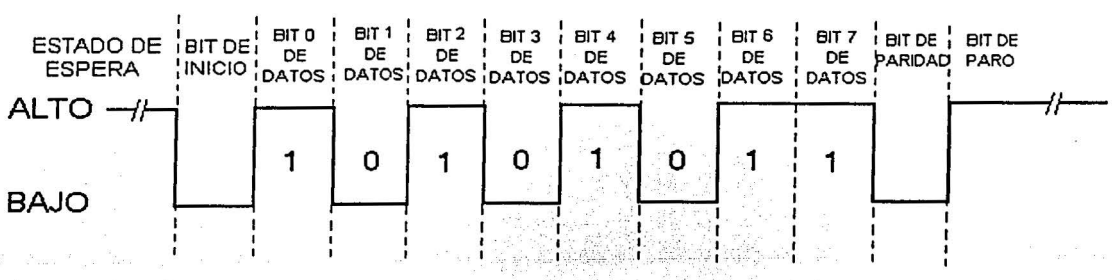


Figura 2.8 Formato de transmisión de datos asíncrono.

Para la comunicación física, la interfaz serial RS232, emplea el conector llamado DB-9 (Fig. 2.9), un cableado de 9 líneas que no deberá exceder de 15 [m], como se establece en la norma; si se requieren cables mayores hay que usar otros medios de interconexión.

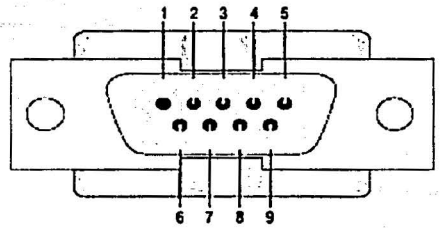


Figura 2.9 Conector DB-9

En la siguiente tabla se da la asignación de pines del conector DB-9.

PIN	SEÑAL	EXPLICACIÓN
1	DCD	Detección de portadora.
2	RX	Recepción de datos.
3	TX	Transmisión de datos.
4	DTR	Terminal lista para enviar datos.
5	GND	Tierra.
6	DSR	Terminal lista para recibir datos.
7	RTS	Indica cuando está listo para enviar datos.
8	CTS	Limpia el bus para enviar datos.
9	RI	Timbre.

Tabla 2.3 Descripción de los pines del conector DB-9.

Para la transmisión y recepción de datos del módulo EZ-COMPASS-3, tan sólo es necesario conectar los pines de transmisión, recepción y tierra, como se muestra en la tabla 2.4.

PC	PIN	EZ-COMPASS-3	PIN
RXD	2	TXD	1
TXD	3	RXD	2
GND	5	GND y Tierra de la fuente	4
		VCC, 5[V]	6

Tabla 2.4 Conexión entre la PC y el módulo EZ-COMPASS-3

Capítulo

3

BALANCEO AUTOMÁTICO.

Se presenta el modelo matemático que define el comportamiento estático de la plataforma y el procedimiento que debe seguirse para balancear un sistema compuesto por diferentes masas. En la plataforma de simulación se ha realizado la integración de sensores, actuadores, baterías, microcontrolador y los aditamentos para poder efectuar el balanceo estático automático con masas deslizantes. Este procedimiento de balanceo ayuda también a cancelar en cierta medida, los efectos de la gravedad.

3.1 ACTUADORES.

Los actuadores son dispositivos que transforman la energía (eléctrica, neumática, hidráulica, térmica, etc.) en un movimiento controlable. Como ejemplo de algunas de las funciones que son llevadas a cabo por los actuadores, para aplicaciones espaciales, se hace un listado a continuación; aunque éste solamente tiene un carácter ilustrativo, ya que nuestro sistema funcionará únicamente en Tierra y toda la complejidad de diseñar y construir sistemas que deban operar en el difícil ambiente espacial alrededor de la órbita terrestre es innecesaria.

Los actuadores producen movimientos para realizar acciones tales como:

- Acople, desenganche o separación de componentes de la nave.
- Apertura o cierre de cubiertas o blindajes.
- Despliegue, posicionamiento, retracción de sensores, paneles solares, antenas e instrumentos.
- Conectar, desconectar y controlar sistemas electromecánicos, neumáticos e hidráulicos.

En la plataforma se utilizan como actuadores dos masas deslizantes para lograr el balanceo automático del simulador. Su funcionamiento está basado en el movimiento de una masa sobre un tornillo de rosca fina, conectado a la flecha de un motor de pasos. Con la ayuda de dos inclinómetros colocados en los ejes de rotación y cabeceo, un microcontrolador toma la decisión de mover hacia delante o hacia atrás cada una de las masas hasta lograr la horizontal de la plataforma y de esta manera conseguir un equilibrio.

3.1.1 Masas deslizantes

Los sistemas de masas deslizantes son los componentes más importantes en el balanceo de la plataforma, ya que de ellos dependerá que se logre un bajo par residual. Esto último está a su vez estrechamente ligado a la resolución que se logre en el movimiento de las masas

por el conjunto motor-tornillo. Dicho movimiento debe ser lo más preciso y lo más fino posible.
Los sistemas de masas deslizantes cumplen con las siguientes características:

- Desplazan una masa siguiendo un movimiento rectilíneo.
- Para su construcción se trató de emplear en la medida de lo posible materiales no magnéticos.
- La longitud de su desplazamiento es de cinco centímetros.
- Tienen la capacidad de mover de 700 a 1000 gramos.
- Su centro de masa está localizado prácticamente en el centro geométrico.

Para poder cumplir con la resolución que debe tener el movimiento lineal, utiliza una platina acoplada a un tornillo de cuerda milimétrica, el cual a su vez está sujeto al motor de pasos; que cuando recibe una serie de pulsos eléctricos hace que la masa se deslice en un movimiento fino. El tornillo tiene una cuerda milimétrica M7X1, con baleros en los extremos.

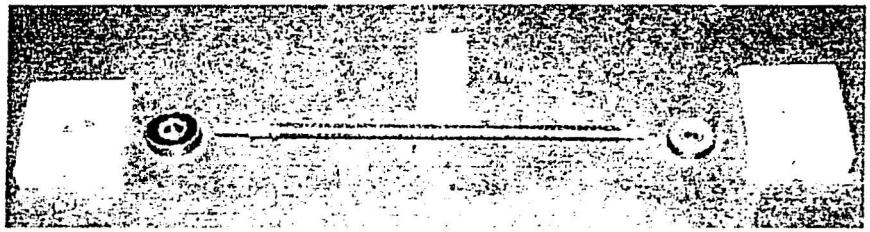


Figura 3.1 Tornillo milimétrico y partes de montaje.

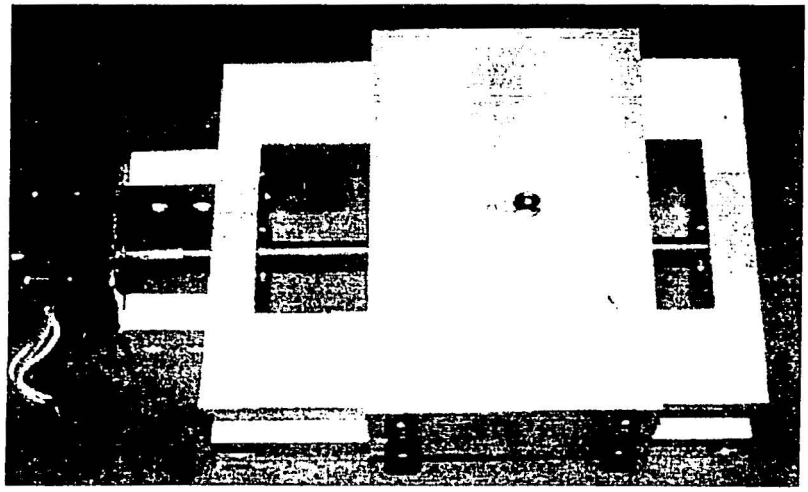


Figura 3.2 Mesa deslizante completa.

3.1.2 Motores de pasos.

La selección del tipo de motor a emplearse en la plataforma, se llevó a cabo considerando los siguientes parámetros: al tipo de acción (y a la rapidez con la que puede cambiar de sentido de giro), a los pares que puede proporcionar, al bajo costo; pero por encima de todo lo anterior, a la precisión en el posicionamiento. Esta última característica fue definitiva para tomar la decisión de optar por el uso de un motor de pasos como parte integral del actuador.

Los motores de pasos son de conmutación electrónica. Son apropiados para mover su eje una cantidad de giro exacta, no tienen escobillas ni conmutador mecánico. La acción de conmutación es lograda por transistores externos. El rotor no tiene devanado de armadura, simplemente es una colección de imanes permanentes salientes

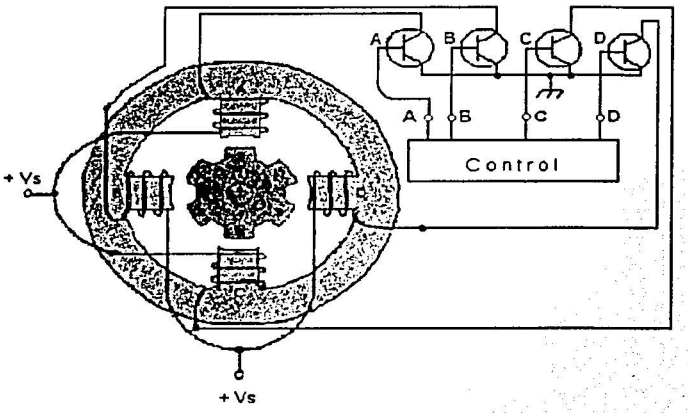


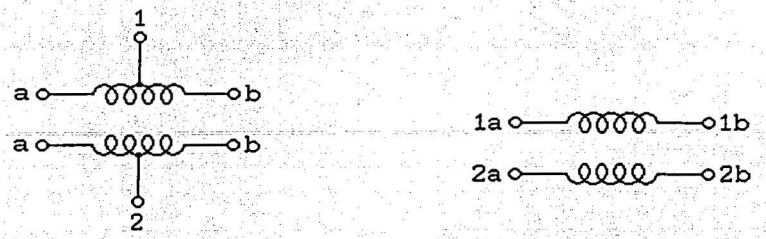
Figura 3.3 Construcción interna de un motor de pasos.

Un motor de pasos es un dispositivo que convierte pulsos eléctricos en movimientos discretos de rotación mecánica (por pasos). Al aplicar a sus bobinas una serie adecuada de impulsos eléctricos, éstas giran sobre su eje un ángulo fijo. Este ángulo recorrido, que depende de las características del motor, se le llama paso, del que podemos controlar la cantidad, su velocidad y el sentido de giro.

Cuando el circuito de control enciende un transistor (A, B, C o D en particular), hay un flujo de corriente de alimentación de CD ($+V_s$) que pasa por ese devanado a través del transistor a tierra. Cuando un solo devanado es energizado, está enrollado de tal manera que su polo se vuelve norte magnético. Su flujo emerge de la cara del polo, pasa a través del rotor, entonces completa su trayectoria entrando en la cara del polo directamente opuesta a él.

Hay dos tipos básicos de motores de pasos, los bipolares y los unipolares. En los primeros, que se componen de dos bobinas, la corriente que circula por éstas cambia de sentido en función de la tensión que se aplica, por lo que un mismo bobinado puede tener en uno de sus extremos distinta polaridad. Los unipolares, que se componen de dos bobinas, se llaman así, porque la corriente por los diferentes embobinados siempre circula en el mismo sentido.

Una manera de diferenciar los dos tipos de motores es considerando que los bipolares solo tienen cuatro conexiones, dos para cada bobina, mientras que los unipolares normalmente presentan seis cables, dos para cada bobina y dos para la alimentación de cada par de éstas. El tipo de motor que utilizamos en este proyecto es unipolar.



a) Unipolar *b) Bipolar*

Figura 3.4 Esquema de los tipos de motores de pasos

Hay diversas formas de mover los motores paso a paso, una manera muy eficiente de hacerlo es por medio de un microcontrolador.

3.1.2.1 Control de los motores de pasos.

Existen varios tipos de secuencias que pueden usarse para manejar los motores de pasos. En todos los casos, los pasos se repiten al terminar las secuencias. Siguiendo los pasos en orden ascendente el giro del motor se efectúa en una dirección, en orden descendente gira en la dirección opuesta.

La secuencia utilizada para mover los motores, como se muestra en la siguiente tabla, es proporcionada por medio de un microcontrolador; donde cuatro pasos, de cuatro bits, son una secuencia, la cual se repite hasta completar un giro completo que corresponde a 200 pasos.

Pasos	Bobina 1	Bobina 2	Bobina 2	Bobina 4
1	0	0	1	1
2	0	1	1	0
3	1	1	0	0
4	1	0	0	1

Tabla 3.1 Secuencia para mover los motores de pasos

Para activar los motores se utiliza un circuito de potencia externo, debido a que no se les puede conectar directamente al puerto de salida del microcontrolador. Es necesario un circuito que entregue suficiente potencia de alimentación para moverlos.

Para el control de los motores unipolares, se deben tomar las derivaciones o taps centrales y conectarlos a la fuente de alimentación positiva. El circuito controlador (figuras 3.5 y 3.6) se encargará de poner cada bobina a tierra para energizarla de manera secuencial (cuando hay un uno lógico en la base del transistor). El número de fases (A, B, C y D) es el doble del número de bobinas, ya que cada bobina está dividida en dos, por medio del tap central.

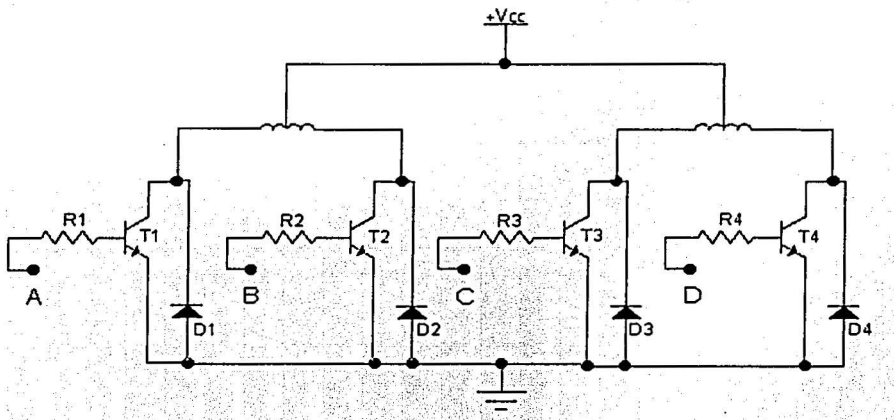


Figura 3.5 Diagrama del circuito de potencia entre el microcontrolador y los motores de pasos.

Donde:

D1, D2, D3, D4:	1N4001
R1, R2, R3, R4:	1[kΩ]
T1, T2, T3, T4:	TIP120

Los diodos D1, D2, D3 y D4 protegen las líneas del circuito frente a los picos de tensión inversa producidos por la conmutación de los devanados, y evitan que la fuerza contraelectromotriz de las bobinas dañe a los transistores; mientas que las resistencias R1, R2, R3 y R4 limitan la corriente de las líneas de datos que llegan desde la salida del puerto B del microcontrolador. Los transistores T1, T2, T3 y T4 amplifican la corriente de salida del puerto para poder controlar al motor.

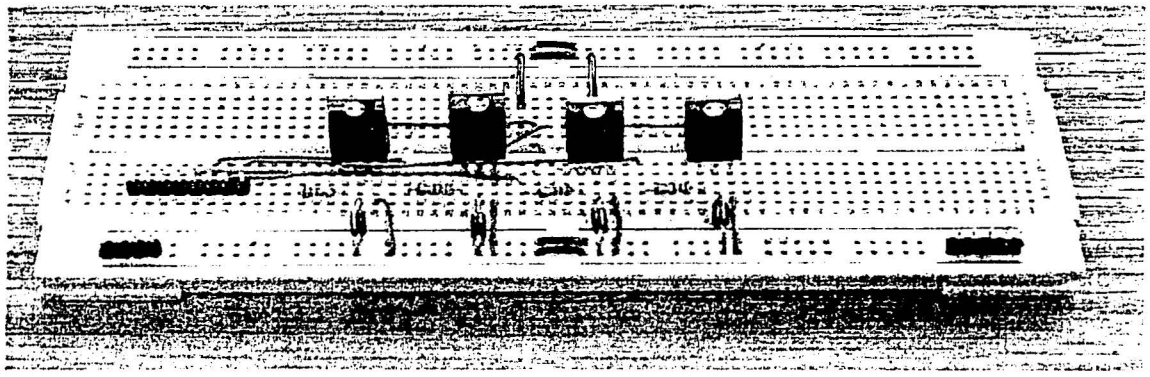


Figura 3.6 a. Circuito de potencia para un solo motor, armado en una tableta de prototipos.

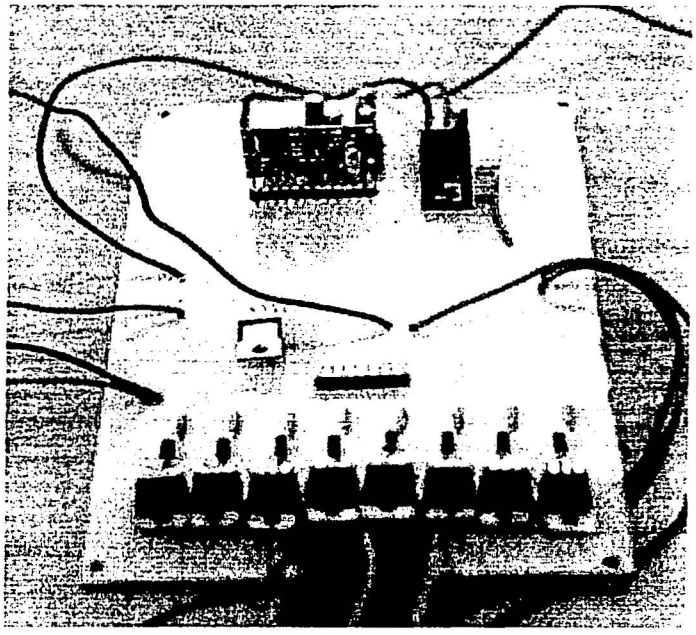


Figura 3.6b Circuito de potencia para los dos motores en su tarjeta impresa.

3.2 NECESIDAD DE BALANCEO

Es de gran importancia resaltar la utilidad del balanceo de la plataforma (manual y automático) particularmente éste último, ya que sin él, las pruebas de control de orientación no pueden realizarse de manera adecuada. En esta sección se hace una descripción de

cómo se llevan a cabo ambos procedimientos de balanceo y el criterio para la aplicación de esta metodología.

3.2.1. Balanceo estático.

A pesar de la expresión, la noción de equilibrio estático se aplica también a objetos en movimiento. Las fuerzas de interés en desequilibrio, se deben a aceleraciones de masas en el sistema. Los requisitos para el equilibrio estático son simplemente que la suma de todas las fuerzas en el sistema móvil (incluidas las fuerzas de inercia de d' Alambert) sean iguales a cero.

$$\Sigma F - ma = 0$$

Otro nombre para el balance estático es: *equilibrio en un plano*, lo cual significa que todas las masas que generan las fuerzas inerciales están prácticamente en el mismo plano. Esencialmente se tiene un problema en dos dimensiones, algunos ejemplos de dispositivos comunes que cumplen con este criterio, y pueden ser equilibrados estáticamente con éxito son: un engrane o una polea montados en un eje, una rueda de bicicleta o motocicleta, un volante delgado, una hélice de aeroplano, los álabes de turbina. El común denominador de todos estos aparatos es que son de corta extensión en la dirección axial, comparada con su magnitud en la dirección radial, y por tanto, pueden ser considerados como cuerpos en un solo plano.

Cualquier elemento que se encuentre en rotación pura puede, teóricamente, estar perfectamente equilibrado para eliminar todas las fuerzas y momentos de vibración. Es aceptado equilibrar todos los elementos o piezas de rotación de una máquina en un plano, a menos que la vibración o sacudimiento sean necesarios.

Este es el argumento principal, para poder balancear nuestra plataforma manteniendo todos los elementos en un plano, el horizontal.

Un elemento rotatorio puede estar equilibrado (o balanceado) tanto estática como dinámicamente. El equilibrio estático es una variante del equilibrio dinámico. Para lograr un equilibrio completo se requiere establecer el equilibrio (o balance) dinámico.

$$\Sigma F = 0 \text{ ; suma de fuerzas igual a cero.}$$

$$\Sigma M = 0 \text{ ; suma de momentos igual a cero.}$$

En nuestro caso el equilibrio estático es un sustituto aceptable para el equilibrio dinámico y es más fácil de alcanzar.

El grado al que un elemento rotatorio se debe balancear dinámicamente depende de la velocidad a la que va a operar. A pequeñas velocidades es tolerable un pequeño desbalanceo de masa debido a que la fuerza de inercia que lo representa puede ser pequeña; pero debido a que la fuerza desbalanceadora aumenta conforme al cuadrado de la velocidad, el desbalanceo transmitido a los cojinetes puede ser grande a altas velocidades.

Nuestra plataforma se mantendrá prácticamente sin movimientos o con una rotación muy lenta durante su funcionamiento por lo que el balanceo dinámico no es importante. Se estima que la velocidad máxima será de una revolución cada 90 minutos.

3.2.1.1. Procedimiento para balancear la plataforma.

Dado el hecho de que la plataforma de simulación se encuentra inicialmente balanceada cuando no tiene ningún otro componente, debido a que la masa se encuentra distribuida de manera homogénea en toda su superficie, el procedimiento que seguiremos consiste básicamente en mantener balanceada la plataforma; añadiendo un par de componentes cada vez, contrarrestando cada uno el efecto de desbalanceo causado por el otro. La importancia de esta estrategia radica en que una vez alcanzado el equilibrio, es posible añadir más componentes, siguiendo el mismo procedimiento y manteniendo sin alteraciones el balance.

Una vez que ya se cuenta con todos los elementos que van a ser colocados en la plataforma de simulación (en este caso los componentes del sistema de balanceo, las baterías y el circuito de control) se procede a la identificación y clasificación de cada uno de ellos, como se observa en la tabla 3.2. Todos los pasos subsecuentes de fijación de elementos deben de mantener el centro de masa sobre el centro del balero, determinando cual debe ser la localización de cada uno de ellos, siguiendo el método analítico. Con este procedimiento se logra el balanceo manual y finalmente se realiza la reubicación del centro de masa en el centro del balero, mediante un corrimiento de las masas deslizantes con el método automático; el cual, como hemos dicho, solamente toma en cuenta la desviación de la horizontal para lograr este efecto.

Elemento	Masa [kg]	Peso f=mg[N] g=9.81[m/s^2]	Forma	Ubicación del Centroide	Notas
Mesa deslizante completa	1.955	19.178	Prismática	Está localizado sobre el eje F-F'.	Ver figura 3.7 a
Platina deslizante	0.615	6.033	Placa delgada	Está localizado sobre el eje E-E'.	Ver figura 3.7 b
Circuito de control	0.658	6.454	Prismática	Centro geométrico	
Baterías	1.800	17.658	Prismática	Centro geométrico	
Masa equilibrante	0.466	4.571	Cilindro	Centro geométrico	

Tabla 3.2 Componentes principales del sistema y sus características.

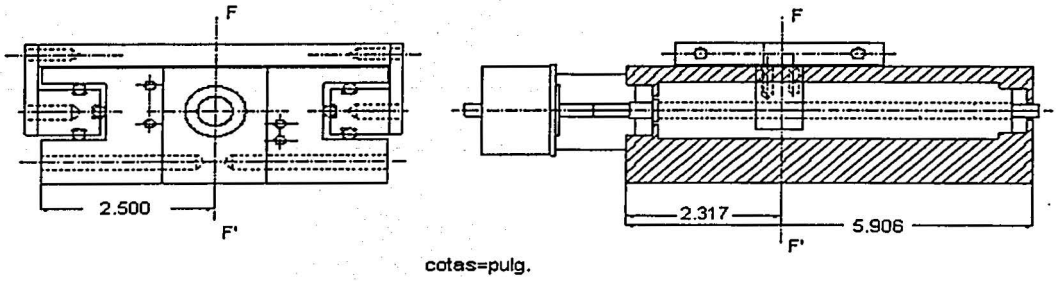


Figura 3.7a Localización del eje F-F' sobre el cual se encuentra el centro de masa de la mesa deslizante.

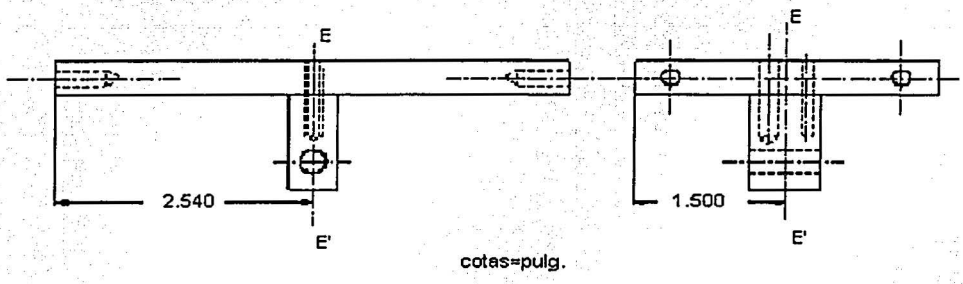


Figura 3.7b Localización del eje E-E' sobre el cual también se encuentra el centro de masa de la platina deslizante.

Una vez que se identificó cada uno de los elementos a integrar a la plataforma, se consideró que la localización de los componentes se hiciera de tal forma que la distribución se optimizara, para que posteriormente en el espacio restante se puedan colocar otros elementos para realizar pruebas en la plataforma. La manera de colocar los componentes es muy sencilla, primero se seleccionan aquellos que tienen necesidades específicas de localización; como los sensores y los actuadores y se fijan en un determinado lugar, a continuación, y siguiendo un eje imaginario que une el centro de masa del dispositivo en cuestión y que pasa por el centro de la plataforma, se coloca otro componente o un lastre, de tamaño conveniente, a una determinada distancia del centro de la plataforma para que el sistema quede balanceado (de preferencia en la parte baja de la plataforma para que el centro de masa se mantenga en el plano de la plataforma). La manera de encontrar esta distancia, dada la masa, es muy sencilla como se verá un poco mas adelante.

El arreglo de los componentes quedó como se muestra en la siguiente figura.

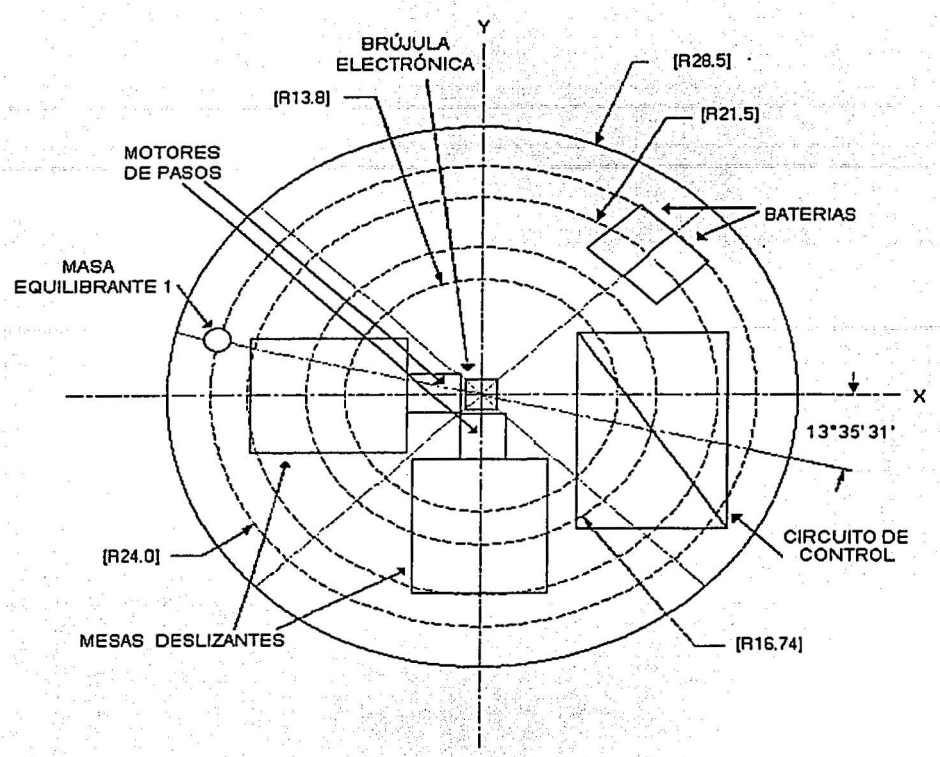


Figura 3.8 Distribución de las masas en la plataforma.

En este caso se ubicó a la brújula en el centro de la plataforma para hacer coincidir los ejes X, Y y Z de ésta, con los del simulador y lograr que las mediciones se realicen con respecto a los mismos ejes de referencia. No fue necesario colocar una masa para contrarrestarla, debido a su bajo peso y localización. En segundo lugar se fijaron las mesas deslizantes sobre los ejes X y Y respectivamente, debido a que son los ejes que van a ser controlados. Como contraparte a ambas, se colocaron las baterías. Posteriormente se fijó el circuito de control y para contrarrestarlo se colocó una masa metálica de forma cilíndrica.

Para obtener las distancias, medidas desde el origen del sistema de coordenadas, a las que deben fijarse las masas de balanceo del sistema mostrado en la figura anterior, se procedió de la siguiente manera:

Se consideró la siguiente nomenclatura:

f : Es el peso de los elementos que se encuentran sobre la plataforma en [N]
R: Es la distancia del centro de la plataforma al centro de masa de cada elemento montado sobre la plataforma en [m]; llamado radio de localización.
M: Es el momento en [N m]

El momento M se obtiene:

$$\mathbf{M} = \text{fuerza} \times \text{distancia} = f\,\mathbf{R} \qquad [\text{N m}]$$

Tomando en cuenta que debemos cumplir con la condición de que la suma de fuerzas y de momentos deben ser cero:

$$\Sigma\,\mathbf{F} = 0$$
$$\Sigma\,\mathbf{M} = 0$$

1. Calculamos la fuerza resultante de las masas deslizantes:

$$f_{musas\_deslizantes} = \sqrt{(f_{masa\_deslizante\_1})^2 + (f_{masa\_deslizante\_2})^2} = \sqrt{(19.178)^2 + (19.178)^2} = 27.12\,[\text{N}]$$

Calculamos el ángulo de ubicación de la resultante:

$$\theta = \tan^{-1}\frac{f_{masa\_deslizante\_2}}{f_{masa\_deslizante\_1}} = \frac{19.178}{19.178} = 45^{\circ}$$

Como la resultante se encuentra en el tercer cuadrante:

$$\theta' = 45^{\circ} + 180^{\circ} = 225^{\circ}$$

Calculamos el radio de localización de las masas deslizantes, que están ubicadas a 0.0976[m] sobre los ejes X y Y respectivamente:

$$R_{masas\_deslizantes} = \sqrt{R_{masa\_deslizante\_1} + R_{masa\_deslizante\_2}} = \sqrt{(0.0976)^2 + (0.0976)^2} = 0.138 \quad [\text{m}]$$

De los cálculos anteriores, obtenemos el momento de las masas deslizantes:

$$M_{masas\_deslizantes} = f_{masas\_deslizantes} R_{masas\_deslizantes} = (27.12)(0.138) = 3.743 \quad [\text{N m}] \qquad ...(1)$$

2. El circuito de control fue colocado arbitrariamente de acuerdo a los requerimientos de espacio sobre la plataforma; su posición se ubica a 0.1674 [m] de radio y con un ángulo de 346.4°. Con estos datos calculamos el momento del circuito de control:

$$M_{control} = f_{control} R_{control} = (6.454)(0.1674) = 1.0805 \quad [\text{N m}] \qquad ...(2)$$

3. Para contrarrestar estos momentos, ubicamos las masas de compensación como se muestra en la figura 3.8, es decir, sobre el mismo eje, a una determinada distancia opuesta al origen.

   Como ya conocemos los momentos de las masas deslizantes y del circuito de control; la masa de las baterías y la masa equilibrante, sólo resta calcular los radios de localización.

   Como ya se mencionó, las baterías compensan a las masas deslizantes y la masa equilibrante al circuito de control. Calculando la suma de momentos:

$$f_{masas\_deslizantes} R_{masa\_deslizantes} = f_{baterías} R_{baterías} \quad \text{[N m]} \qquad ...(3)$$

$$f_{control} R_{control} = f_{masa\_equilibrante} R_{masa\_equilibrante} \quad \text{[N m]} \qquad ...(4)$$

De la ecuación (3), obtenemos el radio de localización de las baterías:

$$R_{baterías} = \frac{f_{masas\_deslizantes} R_{masas\_deslizantes}}{f_{baterías}} \quad \text{[m]}$$

$$R_{baterías} = \frac{3.743}{17.658} = 0.215 \quad \text{[m]}$$

De la ecuación (4), obtenemos el radio de localización de la masa equilibrante:

$$R_{masa\_equilibrante} = \frac{f_{control} R_{control}}{f_{masa\_equilibrante}} \quad \text{[m]}$$

$$R_{masa\_equilibrante} = \frac{1.0805}{4.571} = 0.24 \quad \text{[m]}$$

Con los datos obtenidos a partir de los cálculos anteriores, en la siguiente tabla se muestra la ubicación final de cada elemento. Esto hace que la plataforma se encuentre balanceada estáticamente:

Elemento	Masa [kg]	Radio de localización R[m]	Ángulo de localización [grados]	Localización en el eje X [m]	Localización en el eje Y [m]
Resultante de las masas deslizantes	2.764	0.1380	225	-0.0975	-0.0975
Circuito de control	0.658	0.1674	346.65	0.1629	-0.0386
Baterías	1.800	0.2150	45	0.1586	0.1586
Masa equilibrante	0.466	0.2400	166.65	-0.2425	0.0575

Tabla 3.3 Sistema completo de masas del simulador.

3.2.1.2. Reubicación del centro de masa.

Dado que se trata de compensar el desequilibrio de la plataforma de simulación con las mesas deslizantes, éstas al igual que todos los elementos, se tienen que fijar; por lo que se procede de la siguiente manera.

Ya que se conoce la ubicación de todos los componentes, se tienen que hacer las perforaciones necesarias para sujetarlos a la plataforma de simulación. Es en este tipo de operaciones donde resalta la importancia de las mediciones ya que de ello depende en gran medida que el sistema se pueda balancear, con relativa facilidad.

El balanceo que se ha obtenido siguiendo este procedimiento, ya es bueno, pero aún no logramos ubicarnos dentro del intervalo permisible de balanceo en los ejes de cabeceo y rotación que ha sido establecido en ±1° con respecto a la horizontal, por lo que es necesario realizar el balanceo automático, el cual se describe en la siguiente sección.

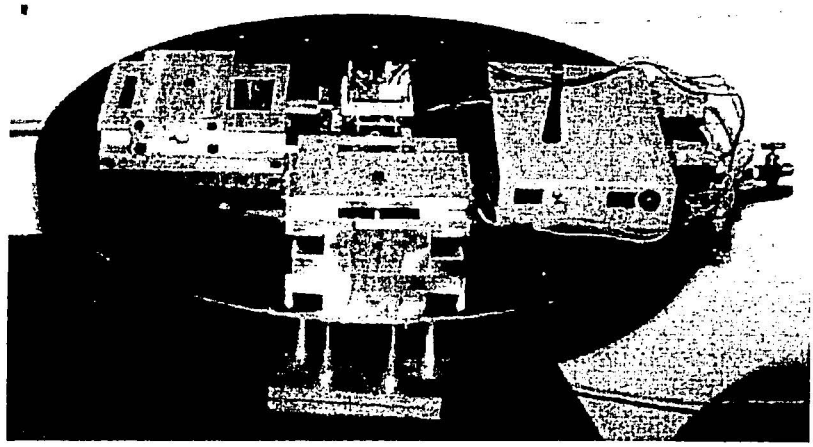


Figura 3.9 Plataforma de simulación con masas deslizantes y componentes de control montados en posiciones fijas.

3.2.2. Balanceo automático.

El sistema de control realimentado (figura 3.10) que lleva a cabo el balanceo automático, tiene como elemento principal un microcontrolador, lo que permite una importante disminución de espacio y energía en la plataforma. Los requisitos que impone el sistema de control deben ser cubiertos de manera satisfactoria por los recursos ofrecidos por este dispositivo. Por ello, su elección depende de la calidad y cantidad de funciones y también de su disponibilidad. En el mercado existen muchos de ellos, con características especiales para cada aplicación; entre los más comerciales se encuentran el 68HC11 de Motorola, el 8096 de Intel, la serie AT89xxxx de Atmel, etc.

El diagrama del circuito electrónico que se encarga de realizar el control junto con el módulo de transmisión-recepción, se muestra en el Apéndice A.

3.2.2.1. Sistema de control.

Un sistema de control de lazo cerrado es aquel en el que la señal de salida tiene efecto directo sobre la acción de control, por lo que los sistemas de lazo cerrado son también llamados sistemas de control realimentados. La señal de error actuante, que es la diferencia entre la señal de entrada y la de realimentación, entra al detector, o control, con la finalidad de reducir el error y llevar la salida del sistema al valor deseado.

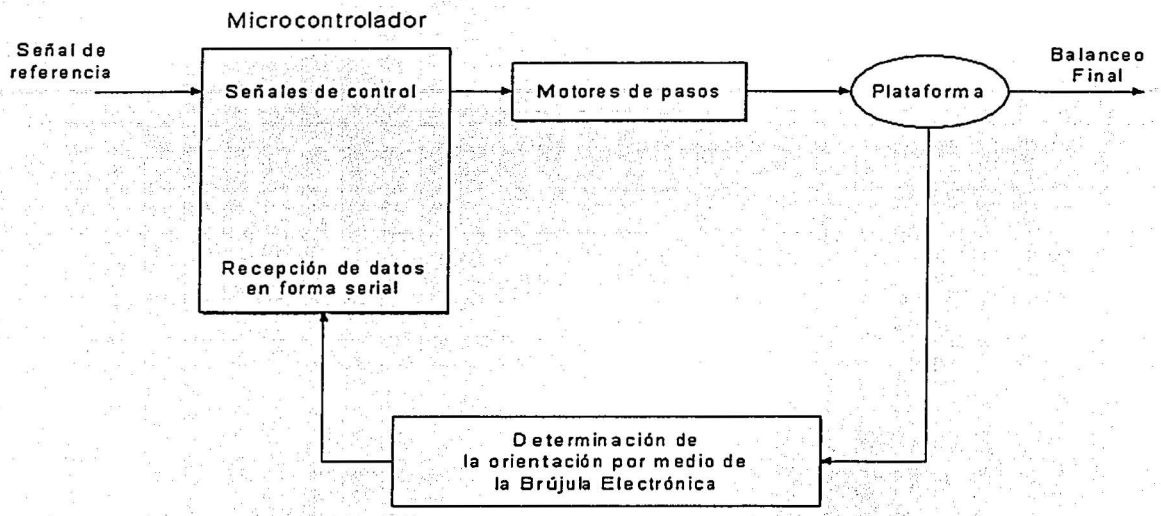


Figura 3.10 Diagrama de bloques del balanceo automático

De acuerdo al diagrama de bloques del sistema, la entrada la constituyen los datos de orientación de la brújula electrónica, éstos son recibidos por el microcontrolador, que a su vez mueve a los actuadores que llevan al sistema a un nuevo estado, la nueva posición leída de la plataforma será la salida y a la vez será el lazo de entrada por retroalimentación, el proceso se continúa hasta que la plataforma quede estabilizada.

Uno de los requisitos más importantes que deben cumplirse cuando se diseña un sistema de control es que sea estable. Un sistema de control lineal invariante en el tiempo es estable si la salida retorna a su estado de equilibrio cuando es sometido a una perturbación; pero cuando el sistema es inestable, la salida presenta una oscilación continua.

Cuando un sistema físico de control involucra almacenamiento de energía, la salida del sistema; relacionada con una entrada, no puede seguir a ésta inmediatamente sino que presenta una respuesta transitoria antes de poder alcanzar un estado estacionario. La respuesta transitoria frecuentemente presenta oscilaciones amortiguadas antes de alcanzar un estado de equilibrio, por lo cual resulta necesario examinar también el comportamiento de esta respuesta transitoria.

Existen varios métodos para determinar la estabilidad de un sistema de control, uno de ellos consiste en proporcionar una señal impulso a la entrada del sistema y ver la respuesta. Si ésta se aproxima a cero para un tiempo $t \rightarrow \infty$, entonces el sistema es estable, en caso contrario, el sistema es inestable.

También es posible determinar la estabilidad de un sistema por medio de métodos analíticos usando la transformada de Laplace para sistemas continuos y la transformada Z para sistemas discretos; partiendo de la ecuación característica del sistema y obteniendo sus raíces. A este método se le llama *criterio de estabilidad por medio de la localización de las raíces* y es aplicable tanto para sistemas continuos como para discretos. Una vez obtenidas las raíces (las cuales pueden localizarse tanto en el plano real como en el imaginario), se puede determinar la estabilidad del sistema. En el caso de sistemas discretos, las raíces deben aparecer dentro del círculo unitario para que el sistema sea estable.

3.3 MICROCONTROLADOR

Un microcontrolador es un circuito integrado que incorpora una unidad central de proceso (CPU) y una serie de recursos internos. La CPU permite que el microcontrolador pueda ejecutar instrucciones almacenadas. Los recursos internos son: memorias RAM, ROM, y EEPROM; puerto serie, puertos de entrada/salida, temporizadores y comparadores.

Se puede decir que es una evolución del microprocesador, al añadirle a este último las funciones que era necesario situar externamente con otros circuitos.

Existen diversos microcontroladores en el mercado, pero el 68HC11 destaca por sus recursos, simplicidad y facilidad de manejo.

3.3.1. Criterios de selección.

Se seleccionó un microcontrolador de la familia 68HC11 de Motorola y en particular el modelo MC68HC11F1, por ser un circuito integrado independiente, que no necesita memoria ni puertos externos pues los lleva en su interior, que facilita la tarea de diseño y reduce el espacio. Cumple los requerimientos de recepción de datos en forma serial, rutinas de control, ejecución en forma autónoma, puertos de salida disponibles, bajo peso, optimización de espacio, bajo consumo de energía y bajo costo; además de que se puede polarizar con baterías, las cuales se pueden fijar a la plataforma. También se considera que un microcontrolador es capaz de realizar una gran variedad de funciones, como la de manejar datos en puertos paralelos de entrada y salida, la posibilidad de ampliación de memoria para casos de programas elaborados, comunicación serial con otros equipos, velocidades de operación en el orden de 10^6 [Hz] y algunas operaciones aritméticas incluidas, todo en un solo circuito integrado.

Para este proyecto se eligió usar este microcontrolador debido a su bajo costo, a la facilidad para su obtención en el mercado nacional, a la cantidad y variedad de recursos que ofrece y a que ya se contaba con experiencias previas sobre el manejo de la familia 68HC11, además de la gran cantidad de aplicaciones existentes y programas de soporte como simuladores y tarjetas de desarrollo.

3.3.2. Características principales del microcontrolador MC68HC11F1.

La facilidad de manejo del MC68HC11F1 y los recursos que ofrece lo hacen adecuado para el sistema de control. Contiene un procesador de 8 bits con puertos que pueden dar el resultado de multiplicaciones, por ejemplo, hasta en 16 bits. Dos acumuladores de 8 bits que operan independientes o unidos como uno sólo a 16 bits. Cuenta con dos registros de índice para referenciar rápidamente el direccionamiento a localidades de memoria. La frecuencia de trabajo máxima en el bus local es de 2 [MHz]. Cuenta con tres puertos paralelos de 8 bits entrada/salida programables, dos puertos paralelos de 8 bits de sólo salida, un puerto de entrada con 8 canales con convertidor analógico-digital de 8 bits, un puerto de comunicaciones seriales, síncronas y asíncronas; una salida de reloj para el control de otro microcontrolador, capacidad de direccionamiento de memoria total de 64 kbytes, una RAM interna de 1 kbyte y una EEPROM de 512 bytes.

Está calificado como de alto desempeño, bajo consumo de energía y con alta inmunidad al ruido, debido a que está fabricado con tecnología HCMOS (metal-óxido semiconductor complementario de alta densidad), además, este es el único microcontrolador de la familia MC68HC11 que posee un bus de direccionamiento y datos no multiplexado, es decir, que no requiere de un multiplexor externo para el control del bus, ya sea para el direccionamiento de una localidad de memoria externa o para el manejo de los datos que se escriben o se leen desde la misma. Además posee un selector inteligente que realiza una conexión sencilla a una memoria externa de programa, sin la necesidad de utilizar ningún componente lógico externo para ello. Este microcontrolador es posible conseguirlo en presentación de tipo montaje de superficie PLCC de 68 y QFP de 80 pines.

3.3.3. Programación del microcontrolador.

En el microcontrolador se construyen las instrucciones que serán utilizadas para realizar cálculos, y manejar los recursos internos y periféricos del sistema. Antes de trabajar con el lenguaje de máquina en forma hexadecimal, es conveniente utilizar una nomenclatura más conveniente que represente las instrucciones de forma comprensible al usuario. Al lenguaje de programación que se utiliza para programar el microcontrolador se le denomina lenguaje ensamblador.

El lenguaje de programación ensamblador del MC68HC11F1 posee una gran variedad de instrucciones dentro de las cuales podemos encontrar: instrucciones para la carga, almacenamiento y transferencia de datos, operaciones aritméticas como son suma, resta, comparaciones, multiplicación y división, operaciones lógicas, prueba de datos, manipulación de bits, corrimientos y rotaciones, instrucciones de control de programa como son brincos, llamadas a subrutinas, interrupciones, etc., además de poseer instrucciones para la condición de los códigos de registro y muchas más, en total 145 instrucciones de programación, seis diferentes tipos de direccionamiento para accesar a la memoria. El MC68HC11F1 posee un bloque de 96 bytes de registros y bits de control los cuales manejan, desde los modos de operación, hasta la configuración y funcionamiento de los diferentes puertos del microcontrolador.

3.3.4. Modos de operación.

Todos los microcontroladores de la familia MC68HC11 poseen cuatro modos de operación, éstos son escogidos con los pines de selección de modo, denotados por MODA y MODB. Los valores que estas entradas presenten durante el proceso de arranque o de reinicialización del microcontrolador, determinarán su modo de operación. En la siguiente tabla se muestran las diferentes opciones.

MODA	MODB	MODO DE OPERACIÓN
1	0	Single chip
1	1	Expandido
0	0	Bootstrap
0	1	Prueba

Tabla 3.4 Modos de operación de los microcontroladores de la familia 68HC11

El microcontrolador normalmente opera en: single chip o expandido; los modos de operación especiales bootstrap y de prueba, se usan para operaciones más exclusivas. A continuación se dará una breve explicación de las características de cada uno de estos diferentes modos de operación.

- *Single chip.*

Sólo se usan los recursos internos del microcontrolador, debido a que no tiene conectados los buses externos de direccionamiento y de datos. Todos los puertos se encuentran completamente disponibles, ya sea para comunicación o entrada y salida de datos.

- *Expandido.*

El microcontrolador puede accesar a memorias externas o a dispositivos periféricos direccionándolos vía un bus de datos y uno de direcciones. Se tiene un direccionamiento externo de 64 kbytes, incluyendo la memoria interna usada en el modo single chip.

- *Bootstrap.*

Es una variación del modo single chip, en el que los vectores de interrupción se encuentran en una ROM llamada de arranque; al iniciar en este modo, automáticamente comienza a ejecutarse el programa BOOTSTRAP el cual permite cargar programas a través de la interfaz SCI (Serial Communications Interface) del propio microcontrolador, a la RAM interna para su posterior ejecución

- *Prueba.*

Puede accesar a fuentes internas del microcontrolador. Es una variación del modo expandido, que es usado principalmente durante el proceso de prueba interna en la producción, y muy pocas veces por el usuario.

Para nuestra aplicación utilizamos el modo bootstrap, para grabar en la memoria EEPROM del microcontrolador, las instrucciones realizadas en lenguaje ensamblador para el control de la plataforma, y el modo single chip, para que el programa funcione en forma autónoma.

3.3.5. Mapa de memoria

Cada modo de operación tiene un mapa de memoria propio. El microcontrolador direcciona 64 kbytes de memoria. En la siguiente figura mostramos el mapa de memoria de los dos modos utilizados.

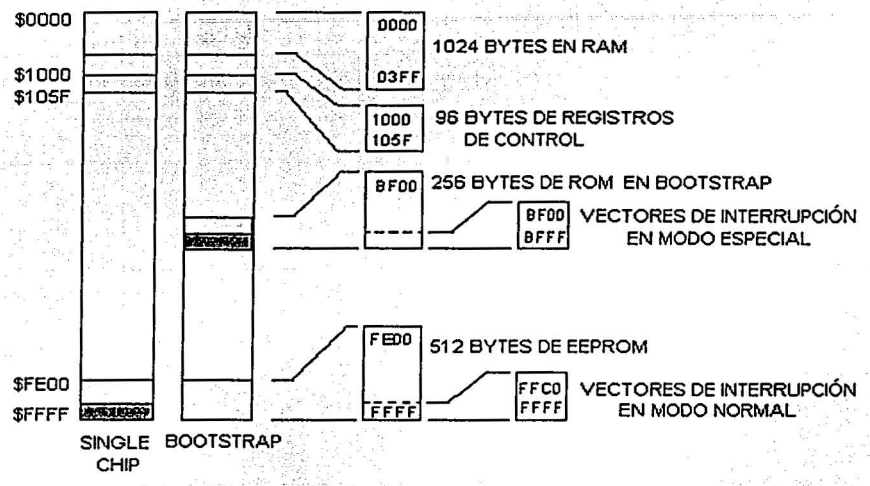


Figura 3.11 Mapa de memoria de los modos single chip y bootstrap.

3.3.6. Puertos de entrada-salida.

El microcontrolador cuenta con 7 puertos de 8 bits disponibles, puerto A, B, C, D, E, F y G. Además de comportarse como puertos normales, sus pines están compartidos con algunos recursos internos.

Nosotros utilizamos el puerto B, en donde sus 8 bits son de salida, para el control de los motores de pasos, y el puerto D de sólo 6 bits, el cual está compartido con el SCI (Serial Communications Interface) que permite realizar comunicaciones asíncronas a distintas velocidades y con paquetes de 8 y 9 bits. Para configurar los parámetros de la comunicación para enviar y recibir datos y comprobar el estado de transmisión, el SCI dispone de 5 registros mapeados de memoria.

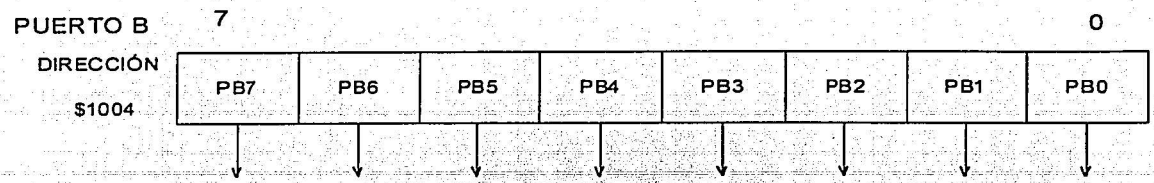


Figura 3.12 Puerto B

Figura 3.13 Bits 0 y 1 del puerto D, compartidos con el puerto serie

3.3.6.1. Unidad de transmisión y unidad de recepción.

El SCI está formado por una unidad de transmisión y una de recepción que son totalmente independientes, lo que permite que las comunicaciones sean bidireccionales, es decir, se puede transmitir y recibir a la vez.

Para la programación del microcontrolador, configuramos los registros del SCI sólo para recepción de datos provenientes de la brújula electrónica, a una velocidad de 9600 [bauds] y 8 bits de datos.

REGISTRO	DIRECCIÓN	DESCRIPCIÓN
BAUD	$102B	Registro de velocidad
SCCR1	$102C	Registro de control 1
SCCR2	$102D	Registro de control 2
SCSR	$102E	Registro de estado
SCDR	$102F	Registro de datos

Tabla 3.5 Registros del SCI

3.3.7. Tarjeta del microcontrolador.

Se adquirió una tarjeta desarrolladora llamada FÁCIL_11, con el fin de no demorar el proyecto desarrollando una tecnología ya existente. Podemos configurar la tarjeta para que opere en alguno de los modos single chip, bootstrap o expandido. Con la ayuda de la tarjeta FÁCIL_11 y los circuitos que la componen es posible llevar a cabo el control automático de la plataforma.

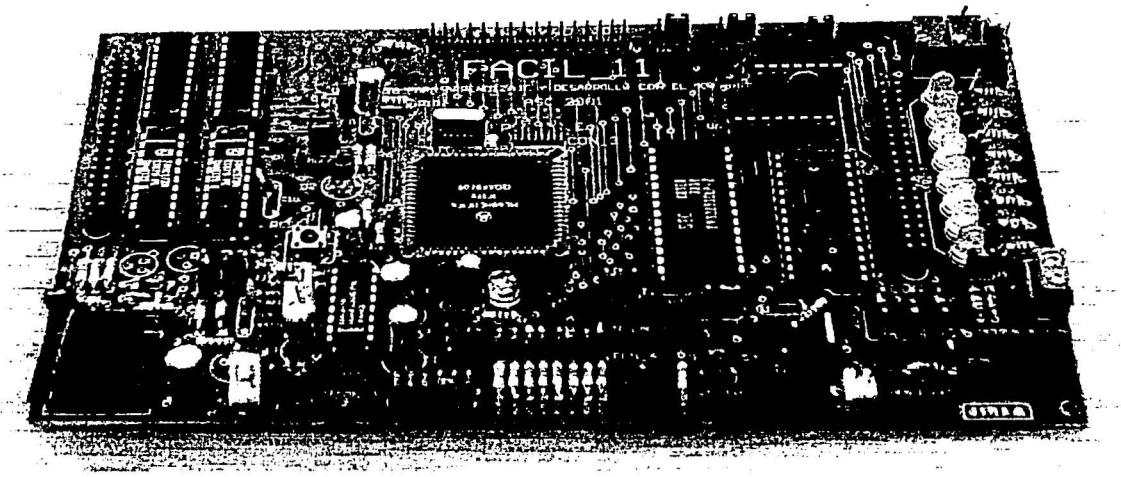


Figura 3.14 Tarjeta desarrolladora utilizada (FACIL_11).

3.4. PROGRAMA DE CONTROL.

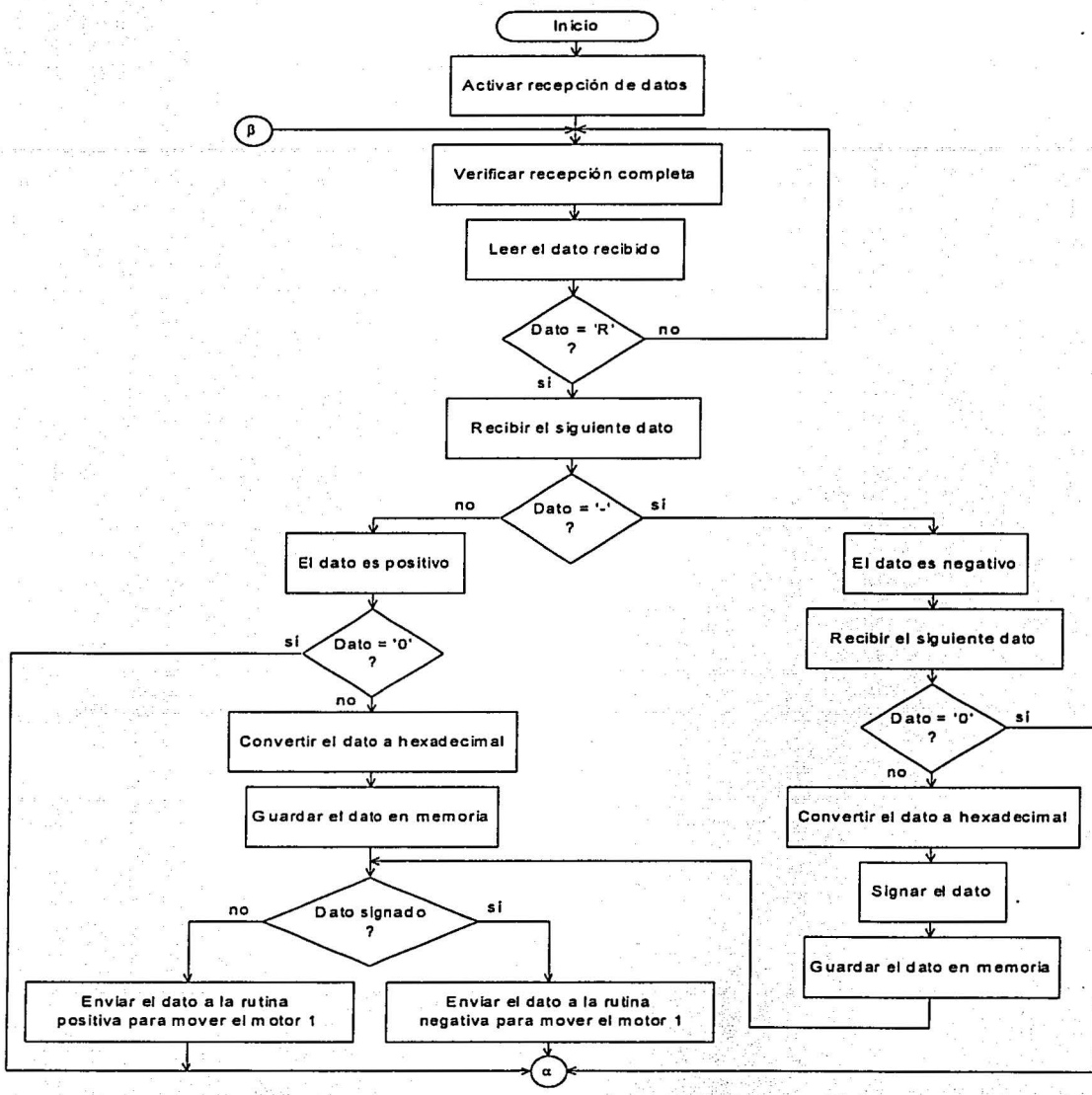
El programa de aplicación del sistema de control se encuentra almacenado en la memoria interna EEPROM del microcontrolador, este programa se diseñó para recibir los datos de orientación (alabeo y cabeceo, en grados) de la brújula electrónica, por el puerto serie y compararlos con el umbral de ±1°. Si no cumple con este valor, activa a los motores de pasos los cuales mueven a las mesas deslizantes en sentido horario o anti-horario, y así sucesivamente hasta estabilizar la plataforma dentro del intervalo deseado.

Durante el proceso de desarrollo del programa, primero se hizo un diagrama de flujo; en el cual se mostraban tanto los cálculos a realizar, como las diferentes decisiones y acciones. Una vez establecido lo anterior se procedió a la codificación del programa, primero en mnemónicos y posteriormente en lenguaje ensamblador. Después de determinar y seleccionar las operaciones a ejecutar se procedió a la construcción de la rutina de control. Esto se llevó a cabo con el auxilio de un programa de edición especializado, el cual, no sólo permitía la escritura del programa en mnemónicos, sino también la conversión del mismo a un lenguaje ensamblador, para su posterior almacenaje y ejecución (el programa editor usado es el IASM11). Una vez que se logró la correcta compilación del programa, se procedió a su prueba por medio de un simulador virtual (el AVSIM11). Una vez que el programa pudo simularse sin ningún error, se procedió a cargarlo en la memoria EEPROM y a ejecutarlo en el microcontrolador.

El procedimiento para cargar el programa de control, en la memoria del microcontrolador, se llevó a cabo con el auxilio del programa PCBUG11; configurando el microcontrolador en el modo bootstrap y a través del puerto serie fue posible la comunicación del microcontrolador con la PC, descargando así de manera directa el programa de aplicación a la memoria EEPROM. Una vez cargado el programa, la ejecución del mismo puede ser realizada en modo single chip, lo cual permite que en el momento de encender el microcontrolador o cada

vez que se realice un reset al sistema, el programa empieza a ejecutarse de manera automática, cuantas veces sea necesario ya que se encuentra almacenado de manera permanente dentro del microcontrolador.

A continuación se muestra el diagrama de flujo del programa de control, el listado completo del programa se muestra en el Apéndice B.



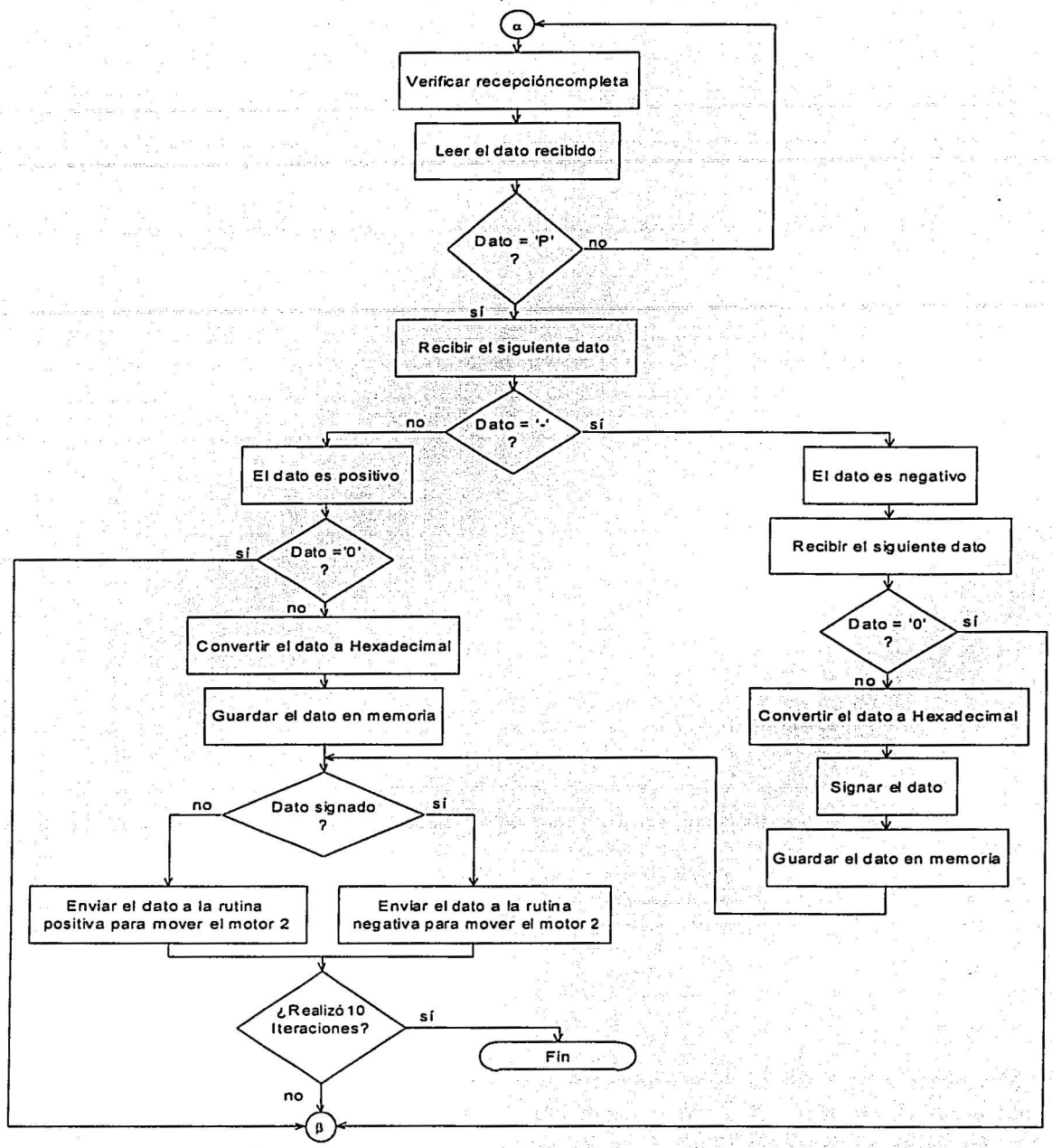


Figura 3.15 Diagrama de flujo del programa para realizar el balanceo automático de un simulador físico de un medio sin fricción.

Capítulo

4

INTERFAZ RS232 INALÁMBRICA

En este capítulo se explica como se lleva a cabo la comunicación inalámbrica, entre la plataforma y una PC, que se utiliza para desplegar y almacenar la información de la orientación de la plataforma, generada por la brújula electrónica. Además, se explica el protocolo de comunicación entre ésta y el transmisor (RS232).

Como se ha explicado con anterioridad, la comunicación entre la plataforma y la computadora de despliegue, debe hacerse de manera inalámbrica; ya que enviar la información de la orientación por medio de cables, produciría un desbalanceo inadmisible en el simulador.

4.1 PROTOCOLO RS232.

Los equipos de comunicaciones serie se pueden dividir en: simplex, half-duplex y full-duplex. Una comunicación simplex envía información en una sola dirección, half-duplex significa que los datos pueden ser enviados en ambas direcciones entre dos sistemas, pero en una sola dirección al mismo tiempo; en una transmisión full-duplex cada sistema puede enviar y recibir datos al mismo tiempo.

Un protocolo es un conjunto de reglas establecidas entre dos dispositivos para permitir la comunicación entre ambos. Los protocolos pueden ser síncronos (gobernados por un reloj maestro) o asíncronos (sin reloj de control en los retrasos relativos entre datos). Los primeros extraen la señal de reloj de los datos incidentes, mediante circuitería. En una transmisión síncrona los datos son enviados en bloques, el transmisor y el receptor son sincronizados por uno o más caracteres especiales. Los protocolos asíncronos tratan cada caracter como un mensaje particular, pero cada caracter tiene bits con intervalos a reloj (sincronía en un caracter, asincronía entre caracteres). El puerto serie de la PC es un dispositivo asíncrono y es compatible con el estándar RS232.

El puerto serie RS232 tiene sus inicios en los años 60´s por la EIA de los EE.UU., y fue creado para ofrecer una interfaz entre equipos, con necesidades de comunicación o envío de datos. Hoy, el más popular estándar de comunicación serial en uso es el EIA/TIA-232-E, el cual ha sido desarrollado por la EIA (Electronic Industry Association) y la TIA (Telecommunication Industry Association), es más conocido popularmente como "RS232", donde RS es "Recomendation Standard". Como todos los asíncronos es más simple, pero al mismo tiempo más lento y está fuera de norma la corrección de errores (como prueba de paridad), lo que generalmente se resuelve por programación. La gran ventaja de tener un puerto serie estándar utilizado internacionalmente, permite que distintos fabricantes produzcan aparatos con esa norma y se interconecten entre sí aumentando las posibles aplicaciones y la conectividad entre equipos.

El protocolo RS232 fue normalizado por la ANSI (American National Standars Institute), tiempo antes de pensar en computadoras. El propósito original fue comunicar un equipo terminal de datos (DTE, Data Terminal Equipment) a un equipo de comunicación de datos (DCE, Data Communication Equipment), sobre una línea telefónica (semi-duplex). Cuando se conectan equipos bajo esta norma, tiene que establecerse quién es la terminal y quién el módem, independientemente de lo que sean físicamente, además no necesariamente se transmite sobre línea telefónica (aunque esté especificado por la norma). La interfaz serial RS232, emplea conectores de 25 o 9 pines o tipo RJ45 y proporciona tanto especificaciones eléctricas, como la mecánicas.

CONSIDERACIONES MECÁNICAS:
Se utilizará un conector normalizado, hacia el lado del equipo terminal (DTE) será conector macho y del lado del módem (DCE) conector hembra, en cada elemento el cable no excederá de 15.22 m (50 pies), si se necesitan cables mayores hay que usar otros medios de interconexión.

CONSIDERACIONES ELÉCTRICAS:
Los niveles de voltaje son simétricos respecto a tierra, de al menos 3 [V] y –3 [V], para marca y espacio (cero y uno con lógica negativa). En la actualidad los valores más usados son ±12 [V] y ±15 [V], tal que la diferencia entre los niveles lógicos podrán estar arriba de 20 [V].

El formato de transmisión de datos asíncrono en forma serial, consiste de cuatro partes: el bit de inicio, los ocho bits de datos, un bit de paridad y por lo menos un bit de paro. El cable RS232, se construye con conectores de 25 o 9 pines, cuya descripción se explica en la siguiente tabla.

DB 9	DB 25	SEÑAL	EXPLICACIÓN
1	8	DCD	Detección de portadora.
2	3	RX	Recepción de datos.
3	2	TX	Transmisión de datos.
4	20	DTR	Terminal lista para enviar datos.
5	7	GND	Tierra.
6	6	DSR	Terminal lista para recibir datos.
7	4	RTS	Indica cuando está listo para enviar datos.
8	5	CTS	Limpia el bus para enviar datos.
9	22	RI	Timbre.
	1	PG	Tierra de seguridad
	23	DSRD	Indicador de Velocidad

Tabla 4.1 Asignación de los pines de los conectores DB9 y DB25.

4.2. COMUNICACIÓN INALÁMBRICA.

En los sistemas de comunicación que utilizan la atmósfera como canal de transmisión, las consideraciones de interferencia y propagación dependen en gran medida de la frecuencia de transmisión. En teoría, se podría usar cualquier tipo de modulación (en amplitud, en frecuencia, banda lateral única, transmisión por corrimiento de fase o de frecuencia, etc.) a cualquier frecuencia de transmisión. Las características de propagación son el resultado de

cambios en la velocidad de las ondas de radio en función de las condiciones circundantes. La velocidad de las ondas depende de la temperatura, la densidad y los niveles de ionización del aire. Cuando los transmisores y receptores son móviles, como el radio de dos vías o el teléfono móvil, las comunicaciones vía infrarrojo o vía cable son imposibles. Por lo tanto, el espacio libre en la atmósfera de la Tierra se usa frecuentemente como medio de transmisión.

Una antena emisora convierte la energía eléctrica entregada por el transmisor, en ondas electromagnéticas que pueden viajar por el espacio, llevando la información hacia uno o varios receptores. La frecuencia debe ser muy alta para producir ondas de intensidad aprovechable que, una vez formadas, viajan por el espacio a la velocidad de la luz. Cuando una de esas ondas encuentra una antena, parte de su energía pasa a los electrones libres de la antena y los pone en movimiento, formando una corriente alterna cuya frecuencia es la misma que la de la onda. Este es el principio de la comunicación por radio.

4.2.1. Radiofrecuencia (RF).

Las radiaciones electromagnéticas con frecuencias entre 10 [kHz] y 100 [GHz] son conocidas como radiofrecuencias (RF). Las RF están divididas en grupos con características similares llamadas "bandas". Las bandas a su vez están divididas en pequeños rangos de frecuencia llamados "canales".

Para que una señal de radio pueda ser transmitida, es necesario que sea enviada al espacio libre, para después ser recuperada y restaurada a su forma original. Dos dispositivos se utilizan para lograr esta tarea: el transmisor y el receptor.

El transmisor se encarga de preparar la señal que va a ser enviada, a través de una modulación. El transmisor tiene tres componentes primarios: una fuente de la frecuencia (oscilador), una etapa de amplificación y un acoplador con el medio (antena). El oscilador genera la frecuencia en la cual funcionará el transmisor. Esta frecuencia se llama "fundamental", para que ésta sea transmitida con eficacia es necesario amplificar la señal. Una vez que se haya amplificado la frecuencia del oscilador, debe transmitirse a través de una línea de transmisión. La antena transmisora permite que la energía sea irradiada eficientemente al espacio libre, y es el puente entre el transmisor y el espacio libre.

El receptor debe recibir la señal modulada, y recuperar la señal original. Este proceso se llama demodulación. Primero la antena de recepción intercepta las ondas electromagnéticas irradiadas, al llegar inducen una pequeña corriente que contiene la misma frecuencia que la señal original en la antena de transmisión; el mezclador toma esta señal y la combina con la frecuencia de un oscilador local, esto convierte la señal a una nueva frecuencia más baja, llamada frecuencia intermedia (IF), el detector filtra la frecuencia intermedia y de esta manera se recupera la información original.

4.2.2. Necesidades de modulación.

La modulación cumple con el objetivo de permitirnos separar los datos válidos, del ruido; el resultado es que la señal final es idéntica a la señal inicial. En esencia, hay tres técnicas de modulación digital que se suelen utilizar en un sistema de radio digital: modulación por corrimiento de frecuencia (FSK), por desplazamiento de fase (PSK), y de amplitud en cuadratura (QAM). Se hace énfasis en la primera de ellas, debido a que es la que se utiliza

en este trabajo de tesis. La modulación FSK ofrece ventajas significativas por encima de otros métodos de modulación, incrementa la inmunidad al ruido y la habilidad de recibir múltiples señales. Esto se aprecia particularmente en bandas saturadas como en la RF. Aunque no es la manera más eficiente de modular datos digitales, es una opción excelente, fiable y económica.

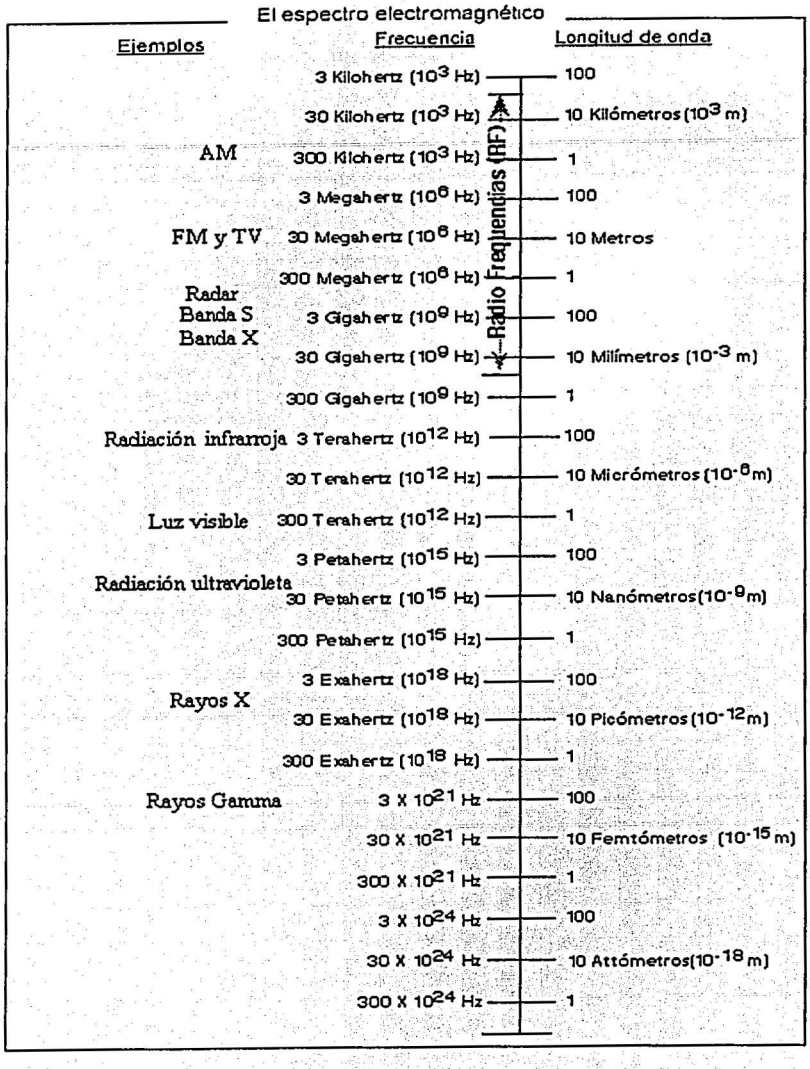


Figura 4.1 Espectro electromagnético.

4.2.2.1. Modulación FSK

La modulación por corrimiento de frecuencia (FSK, por sus siglas en inglés: Frecuency Shift Keying) consiste en desplazar la frecuencia de una portadora senoidal, desde una frecuencia de marca, hasta una frecuencia de espacio; de acuerdo con la señal de bandabase digital.

En un transmisor binario, una frecuencia central o portadora, es variada según los datos de entrada, es decir, la FSK se desplaza entre dos frecuencias: una de marca y una de espacio. Con la FSK binaria, hay un cambio de frecuencia en la salida, por cada condición lógica de la señal de entrada. Así, la razón de cambio de salida, es igual a la razón de cambio de entrada.

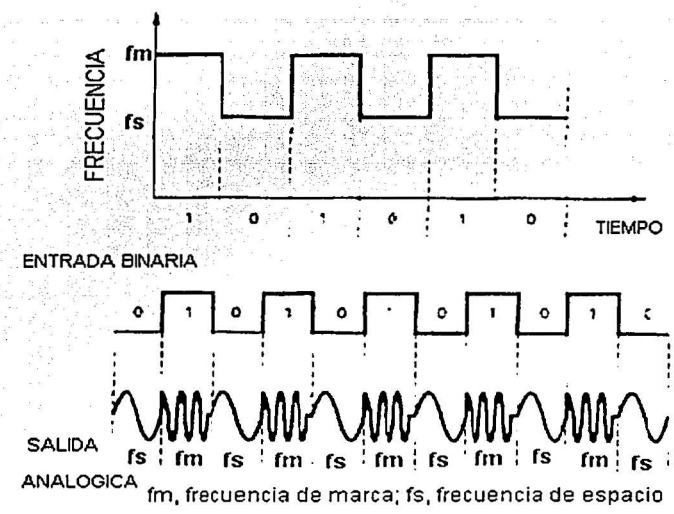


Figura 4.2 Señales de una transmisión FSK binaria.

En la modulación digital, la razón de cambio (rapidez) en la entrada del modulador, se llama razón de bit y tiene unidades de bits por segundo [bps]. La razón de cambio de la salida del modulador se llama baud y es igual al recíproco del tiempo de salida. En la FSK binaria, las razones de cambio de entrada y salida son iguales; en consecuencia, la razón de bit y la razón de baud son iguales. Las señales en una transmisión FSK binaria se muestran en la figura 4.2.

4.2.2.2. Demodulación FSK.

El medio más utilizado para demodular las señales FSK, es el circuito de amarre de fase PLL (Phase Locked Loop). Conforme cambia la entrada del PLL entre las frecuencias de marca y espacio, el voltaje de CD a la salida del comparador de fase, sigue el desplazamiento de frecuencia. Debido a que sólo hay dos frecuencias de entrada (marca y espacio), también hay sólo dos niveles de voltaje a la salida. Uno representa un 1 lógico y el otro un 0 lógico. En consecuencia, la salida es una representación de los niveles (binarios) de la entrada de FSK. Por lo regular, la frecuencia natural del PLL se hace igual a la frecuencia central del

modulador de FSK. Como resultado, los cambios en el voltaje de CD, sigue a los cambios en la frecuencia de entrada y son simétricos alrededor de 0 [V]. La FSK binaria tiene un error mas alto que PSK ó QAM y en consecuencia, rara vez se utiliza para sistemas de radio digital de alto rendimiento.

4.3. MÓDULOS DE TRANSMISIÓN Y RECEPCIÓN INALÁMBRICA.

Para resolver el problema de la comunicación inalámbrica entre la brújula electrónica y la PC, se había utilizado con anterioridad un circuito FSK empleando el XR2206 (como modulador) y el XR2211 (como demodulador), pero se tuvieron algunos problemas con la potencia de salida, ya que no era suficiente, por lo que esta posibilidad se descartó. Otra solución propuesta fue la transmisión infrarroja, pero también se deshecho debido a que se encontró que no era muy confiable, por ser susceptible a interferencias. Dada la necesidad de un circuito de bajo peso, bajo costo y poco volumen, buscamos alguno que se adaptara a esas características. De los circuitos buscados, el más conveniente fue el HP-II (de Linx Technologies), la razón más importante por la cual escogimos este módulo, es que puede manejar el protocolo de comunicación RS232.

El módulo HP-II es de alto rendimiento para datos analógicos o digitales, opera dentro de la banda de RF, es decir, va de 902 a 928 [MHz], es capaz de recibir la información a una distancia aproximada de 100 metros (bajo condiciones óptimas), emplea una avanzada arquitectura de un microcontrolador y no requiere de componentes externos, excepto una antena.

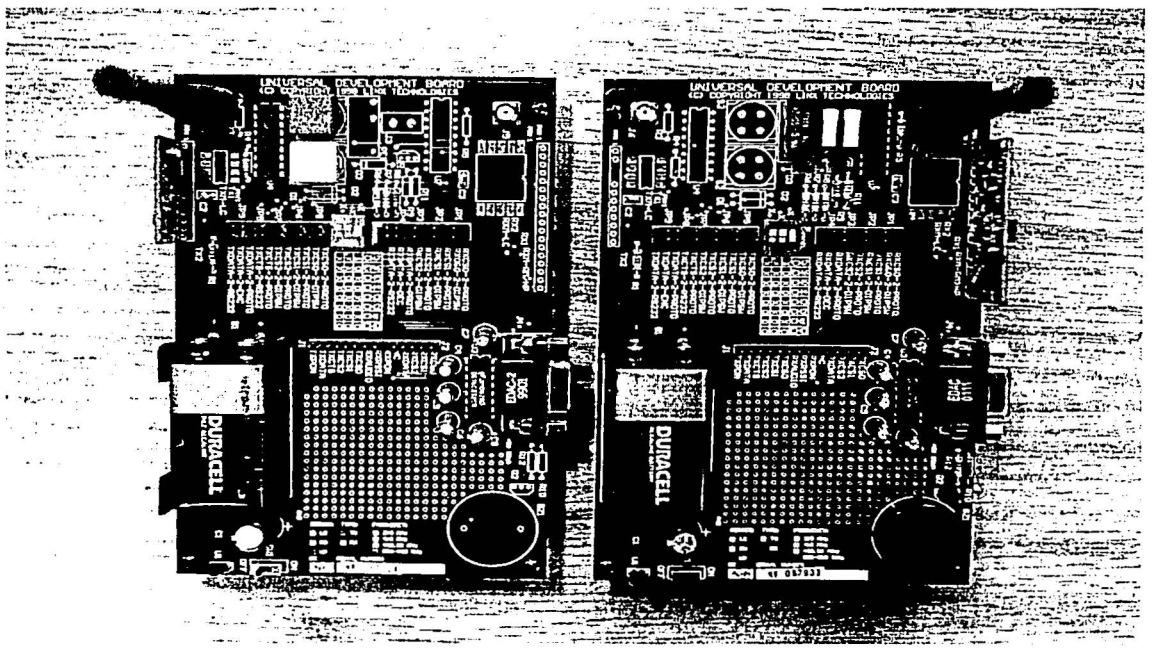


Figura 4.3 Foto de los módulos Tx y Rx HP-II y de las tarjetas de desarrollo.

El módulo de comunicación, fabricado por Linx Tech, cuenta con dos tarjetas que ofrecen un método simple, eficiente y económico para hacer circuitos inalámbricos; nos permiten diseñar módulos de evaluación rápida y cuentan con un área para desarrollar prototipos. Estas tarjetas permiten muchas otras aplicaciones tales como:

- Automatización industrial o casera
- Redes inalámbricas
- Control y acceso remoto
- Monitoreo y telemetría inalámbrica
- Radiofrecuencia de rango largo
- Transferencia de señales analógicas.

Las tarjetas de desarrollo, permiten configurar al circuito de comunicación inalámbrica en ocho diferentes canales, con una frecuencia asociada, que se pueden seleccionar con un "dip switch" (tabla 4.2), también permiten definir el tipo de datos a transmitir, ya sean analógicos o digitales.

CS2	CS1	CS0	CANAL	FRECUENCIA
0	0	0	0	903.37
0	0	1	1	906.37
0	1	0	2	907.87
0	1	1	3	909.37
1	0	0	4	912.37
1	0	1	5	915.37
1	1	0	6	919.37
1	1	1	7	921.37

Tabla 4.2 Selección de canales en las tarjetas de desarrollo.

Para comprobar el funcionamiento de los circuitos de transmisión inalámbrica, se realiza la siguiente prueba: una de las tarjetas del módulo se acondiciona como transmisor y la segunda como receptor. Por medio de dos botones es posible encender un relevador y una bocina, ubicados en la tarjeta receptora, comprobando así el funcionamiento de los circuitos. Para que exista transmisión de datos es importante verificar que ambas tarjetas estén configuradas para trabajar en el mismo canal; además, para que la transmisión no tenga ruido, es necesario colocar los circuitos a una distancia de por lo menos 1.5 [m].

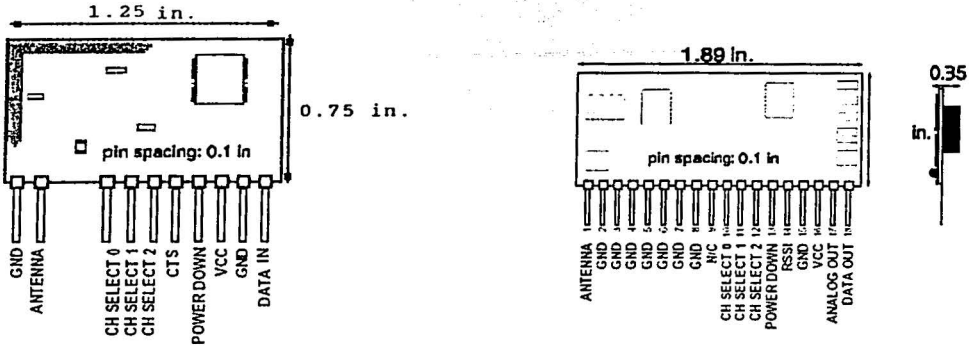
4.3.1. Circuitos de transmisión y recepción.

Los circuitos de transmisión y recepción están diseñados para la transferencia eficaz de datos analógicos y digitales. Para asegurar el buen funcionamiento, los módulos emplean la modulación-demodulación FSK y una arquitectura sintetizada de un microcontrolador.

Las características de los circuitos de transmisión y recepción son:

- Selección de 8 Frecuencias binarias.
- Sensibilidad: -95 [dBm]
- Velocidad máxima: 50 [kbps]

- Interfaz serie directa.
- No requiere ningún componente de RF externo excepto la antena.
- Suministro de energía: 2.7 [V] a 16 [V]
- Temperatura de operación: 0 [°C] a 70 [°C]
- Impedancia de entrada: 50 [Ω]
- Ancho de banda de ruido: 280 [kHz[
- Ancho de banda de los datos: 300 a 50,000 [bps]



(a) Transmisor *(b) Receptor*

Figura 4.4 Circuitos de transmisión-recepción HP-II y su asignación de pines.

Para utilizar de manera adecuada los circuitos de transmisión y recepción, de modo que permitan una comunicación RS232 inalámbrica, es necesario conectar adicionalmente algunos componentes como: capacitores y el conocido circuito MAX-232 para convertir los niveles de voltaje TTL de los módulos, a niveles de voltaje adecuados para el funcionamiento de la interfaz.

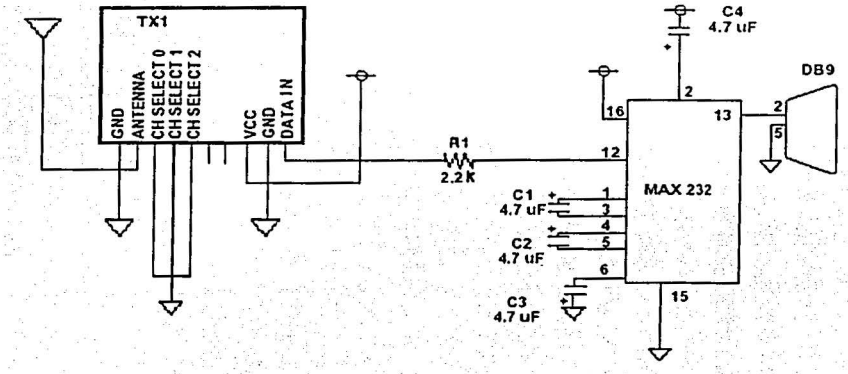


Figura 4.5 Configuración para la transmisión RS232.

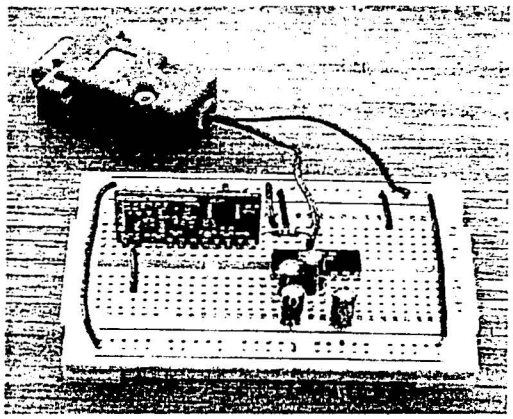


Figura 4.6 Prototipo para la transmisión RS232.

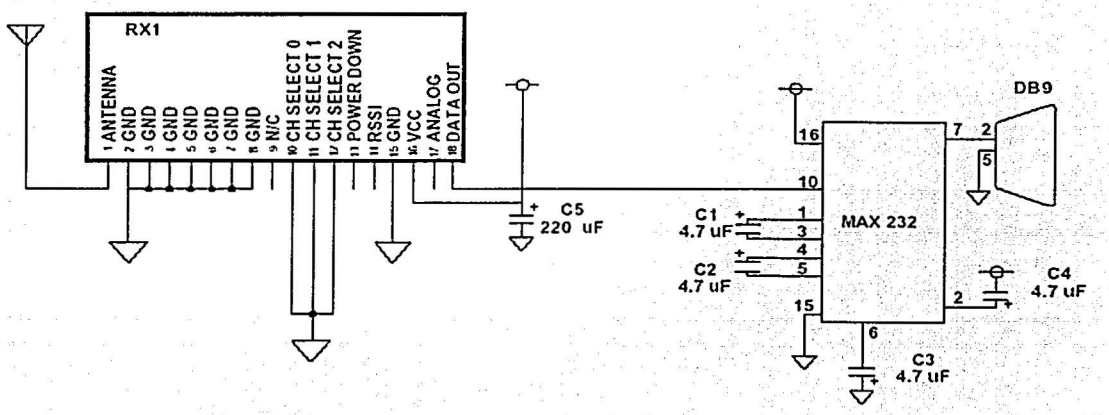


Figura 4.7 Configuración para la recepción RS232.

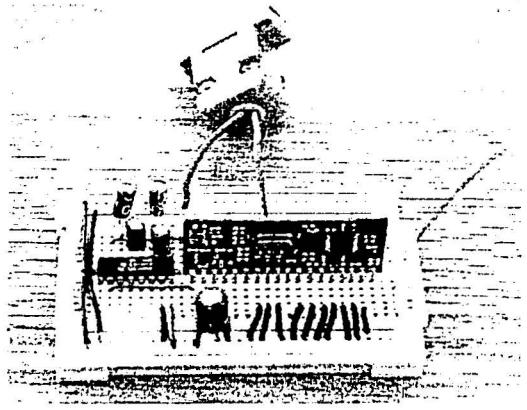
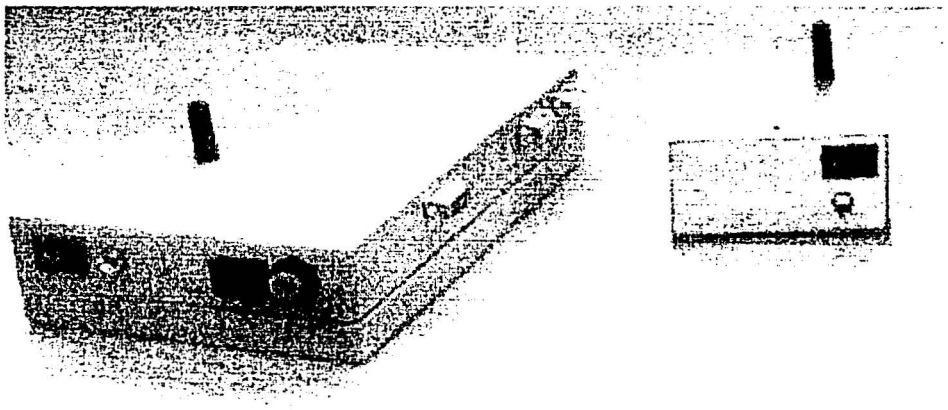


Figura 4.8 Prototipo para la recepción RS232.



(a) Transmisor (b)Receptor

Figura 4.9 Módulos de transmisión y recepción en su forma final.

Durante las pruebas realizadas de la transmisión-recepción de la orientación de la brújula electrónica (figura 4.10), pudimos observar algunos problemas. Uno de ellos fue que los datos recibidos no eran los reales, sino ruido, esto se debía a que los circuitos estaban muy próximos, a menos de 1.5 [m]. Otro problema que tuvimos fue que no recibíamos datos en los pines del DB9, la solución fue realizar la conexión módem nulo en la cual cruzamos los pines rx-tx, esto para que el transmisor de la brújula quedara conectado al receptor de la computadora. También ocurrió la llegada de un caracter erróneo en los datos recibidos;

debido a que éste era consistente, para corregirlo se hizo una modificación en el programa que recibe los datos. El procedimiento de corrección de errores no fue tratado por salir del alcance de esta tesis.

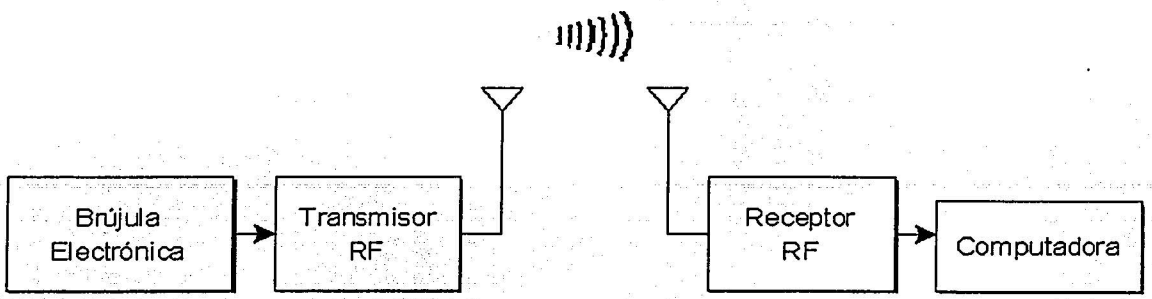


Figura 4.10 Esquema de la transmisión-recepción entre la brújula electrónica y la computadora.

Capítulo

5

SISTEMA DE MONITOREO.

Al inicio del proyecto se analizó la necesidad tanto de visualizar, como de almacenar en un archivo, la orientación de la plataforma. Se decidió que la mejor forma de presentar la información de la orientación era auxiliándose de un programa gráfico para poder observar el movimiento del simulador en sus tres grados de orientación (alabeo, cabeceo y guiñada). El almacenamiento de la misma información, se llevaría a cabo en un archivo susceptible de ser aprovechado por programas de manejo de datos como Lotus o Excel, por ejemplo. También se estableció que la interfaz entre la computadora y el usuario fuera lo más comprensible y sencilla posible, utilizando herramientas gráficas que faciliten el funcionamiento del programa, y que éste pueda ser manejado por cualquier persona involucrada en pruebas posteriores de orientación, que requieran el almacenamiento de datos.

5.1. VARIABLES A MONITOREAR.

La orientación de un satélite está dada por tres ejes de rotación denominados: alabeo (X), cabeceo (Y) y guiñada (Z), estas son las variables que el programa de graficación se encarga de monitorear, durante las pruebas de orientación.

La denominación de estos ejes es análoga a los utilizados en aviación y se describen de la siguiente manera: el eje de alabeo o rotación, se encuentra localizado de manera axial al vector velocidad, es decir, se encuentra sobre la trayectoria orbital, esto se ilustra en la figura 5.1.

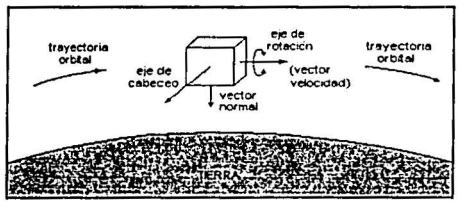


Figura 5.1 Eje de rotación.

El eje de guiñada se define como el vector que une el centro de masa del satélite, con el centro de masa de la tierra (ver figura 5.2).

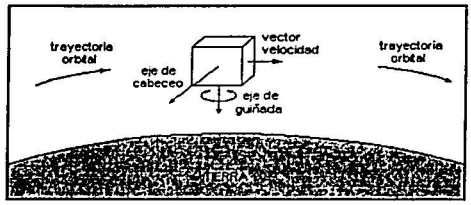


Figura 5.2 Eje de guiñada.

El eje de cabeceo es perpendicular a los ejes de rotación y de guiñada, de tal manera que se forma un sistema de ejes ortogonales (ver siguiente figura).

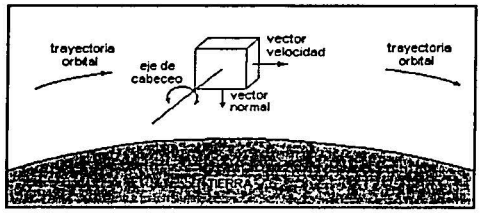


Figura 5.3 Eje de cabeceo.

5.2. GRAFICACIÓN.

La graficación de la orientación, es llevada a cabo por un programa que despliega la información enviada por la brújula, a través del sistema de comunicación unidireccional inalámbrico. Este programa tiene la capacidad de mostrar dos perspectivas o vistas diferentes de la orientación, es decir, como si tuviéramos dos cámaras que muestran de manera gráfica, la posición real de la plataforma.

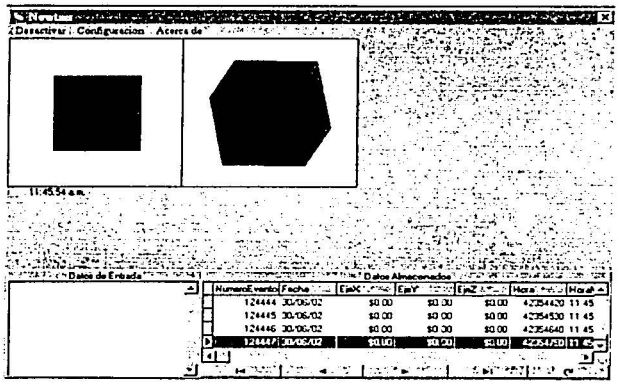


Figura 5.4 Pantalla principal del programa de graficación.

Además del despliegue visual, también se almacenan los datos de orientación en un archivo para su análisis posterior y para poder graficar los resultados. La rutina de almacenamiento se describe mas adelante.

Se hizo una evaluación de los lenguajes de programación mas adecuados para esta aplicación, lo cual se explica en la siguiente sección.

5.2.1. Lenguajes de programación.

Se consideró qué lenguaje sería el más apropiado para nuestra aplicación y se evaluaron sus ventajas y desventajas. Fue necesario revisar aquellos lenguajes que cumplían con los requisitos que impone la aplicación, para obtener los mejores resultados en un tiempo más corto, además de tomar en cuenta los tiempos de práctica y aprendizaje y considerar la disponibilidad de compiladores y editores.

En la actualidad existen numerosas opciones de programación, como son: Pascal, C, C++, Visual C, Visual C++, Visual Basic, Delphi, Java, entre otros.

El lenguaje que más se adaptó a nuestro proyecto fue Delphi 3.0, ya que es de mediano nivel y nos facilita la creación de una gran variedad de aplicaciones de Windows, en comparación con otros lenguajes de programación. Delphi nos facilita el manejo de imágenes ya sean dinámicas o estáticas, el diseño y enlace de pantallas, así como la creación de líneas, círculos, menús, cajas de texto, barras de desplazamientos para texto o imágenes, con la capacidad de hacer visible o invisible cada control, lo que permite contar con la ventaja de aprovechar una sola zona de la pantalla para tener distintos datos, y algo muy importante; nos permite hacer una liga directa para almacenar en una base de datos las lecturas tomadas.

5.2.2. Lenguaje Delphi

Delphi tiene la facilidad de la programación visual de las herramientas tipo RAD (Rapid Application Development) de que carece C++. Se puede ejecutar con cualquier tipo de Windows disponible. Cuenta no sólo de un compilador muy rápido, sino también de potentes herramientas para la creación visual y el manejo de bases de datos. Es una herramienta de desarrollo de programas que permite la creación de aplicaciones para Windows 3.x, Windows95 y Windows NT. Las aplicaciones pueden colocarse de forma muy sencilla en la pantalla según el principio de módulos. Para ello se dispone de una paleta dotada de una gran variedad de componentes, algo así como los bloques de construcción de cada programa.

Este lenguaje dispone del Object Pascal, un lenguaje de programación que está a la altura del C++ y que incluso lo supera en algunos aspectos. Este lenguaje surge a partir del desarrollo del Borland Pascal 7.0, muy importante en la programación de computadoras personales. El Object Pascal es totalmente compatible con el Borland Pascal 7.0. lo que permite que programas desarrollados con este último puedan ser convertidos a Delphi. Delphi nos proporcionó las herramientas necesarias para poder realizar la graficación y el almacenamiento de los datos de orientación de la plataforma de manera exitosa.

5.3 ALMACENAMIENTO DE DATOS.

Como ya se mencionó, Delphi tiene la facilidad de permitirnos trabajar con bases de datos. Para efectos del almacenamiento de los datos de orientación, utilizamos la herramienta "Data Base Desktop", que nos genera una base de datos con los parámetros que intervienen en las pruebas: alabeo, cabeceo, guiñada, número de evento, tiempo y fecha.

Una gran ventaja es que los archivos que se generan pueden ser manipulados por programas comunes como es Lotus o Excel, que nos permiten la realización de gráficas para analizar el comportamiento del sistema.

En el Apéndice C, se muestra el código del programa de graficación y almacenamiento de datos.

Database Desktop - [Table : C:\cubo\Datos\Datos.DB]

File Edit View Table Record Tools Window Help

Datos	Fecha	EjeX	EjeY	EjeZ	HoraNormal
1	16/06/02	0.00	0.00	0.00	01:02:35 p.m.
2	16/06/02	0.00	0.00	0.00	01:02:36 p.m.
3	16/06/02	0.60	-1.35	35.00	01:02:36 p.m.
4	16/06/02	0.60	-1.35	35.00	01:02:36 p.m.
5	16/06/02	0.60	-1.35	35.00	01:02:36 p.m.
6	16/06/02	0.60	-1.35	35.00	01:02:36 p.m.
7	16/06/02	0.60	-1.35	35.00	01:02:36 p.m.
8	16/06/02	7.60	-1.35	35.00	01:02:36 p.m.
9	16/06/02	7.60	-1.35	35.00	01:02:36 p.m.
10	16/06/02	7.60	-1.35	35.00	01:02:36 p.m.
11	16/06/02	7.60	-1.35	35.00	01:02:36 p.m.
12	16/06/02	7.60	-1.35	35.00	01:02:36 p.m.
13	16/06/02	7.60	-1.35	35.00	01:02:37 p.m.
14	16/06/02	7.60	-1.35	35.00	01:02:37 p.m.
15	16/06/02	7.60	-1.35	35.00	01:02:37 p.m.
16	16/06/02	7.60	-1.35	35.00	01:02:37 p.m.
17	16/06/02	7.60	-1.35	35.00	01:02:37 p.m.
18	16/06/02	7.60	-1.35	35.00	01:02:37 p.m.
19	16/06/02	7.60	-1.35	34.90	01:02:37 p.m.
20	16/06/02	7.60	-1.35	34.90	01:02:37 p.m.
21	16/06/02	7.60	-1.35	34.90	01:02:37 p.m.
22	16/06/02	7.60	-1.35	34.90	01:02:37 p.m.
23	16/06/02	7.60	-1.35	34.90	01:02:38 p.m.
24	16/06/02	7.60	-1.35	34.90	01:02:38 p.m.
25	16/06/02	7.60	-1.35	34.90	01:02:38 p.m.
26	16/06/02	7.60	-1.35	34.90	01:02:38 p.m.

Figura 5.5 Almacenamiento de los datos de orientación.

Capítulo

6

RESULTADOS Y CONCLUSIONES.

Una vez que se comprobó el correcto funcionamiento, por separado, de los sistemas de balaceo automático y de graficación, éstos se integraron entre sí para verificar su funcionamiento en conjunto y así poder realizar las pruebas del sistema completo.

6.1 PRUEBAS FINALES DE FUNCIONAMIENTO.

El sistema de control automático (figura 6.1), que utiliza como elemento principal un microcontrolador, se encarga de recibir la información de los inclinómetros, contenidos en el módulo de la brújula electrónica, con una frecuencia de doce ciclos por segundo. Una vez recibidos los datos, éstos son comparados contra el umbral de referencia de ±1°, y si no cumple la condición de localizarse en esa posición, entonces manda 10 secuencias de pulsos por cada grado fuera del intervalo de referencia, lo que significa un desplazamiento de 0.05 [mm], en las masas deslizantes.

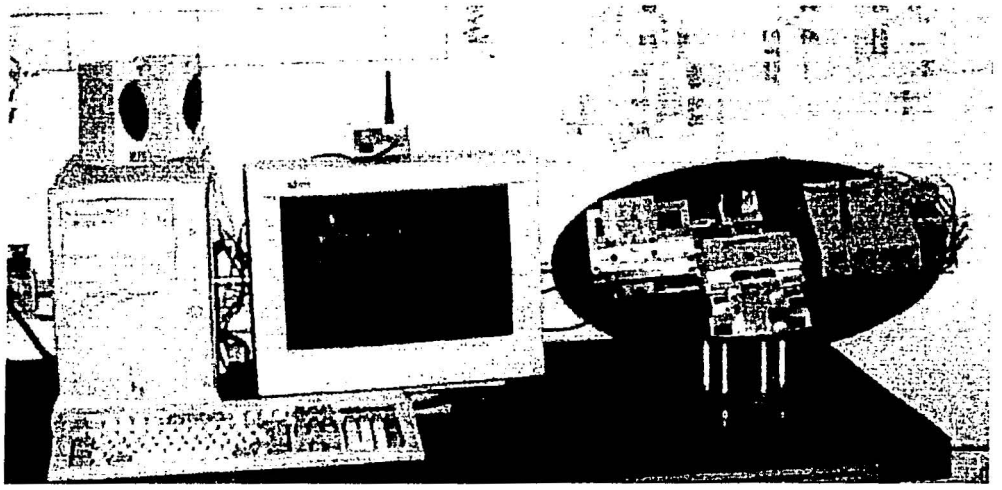


Figura 6.1 Sistema de balanceo automático y monitoreo inalámbrico de la orientación de la plataforma de simulación de un medio sin fricción.

Las curvas presentadas en la figura 6.2 prueban que la plataforma se logra balancear por medio del control automático dentro del intervalo de ±1°, tanto en el eje de alabeo como en el

de cabeceo. El tiempo en el que la plataforma alcanzó el balanceo adecuado, fue en los primeros 4 minutos (dentro de un total de 15 minutos que duró la prueba). En la base de datos se almacenan 12 muestras por segundo, estos datos nos sirven para conocer el comportamiento, de manera gráfica, de la plataforma. En las siguientes figuras, únicamente se utilizaron 2400 muestras, correspondientes al tiempo de estabilización del sistema.

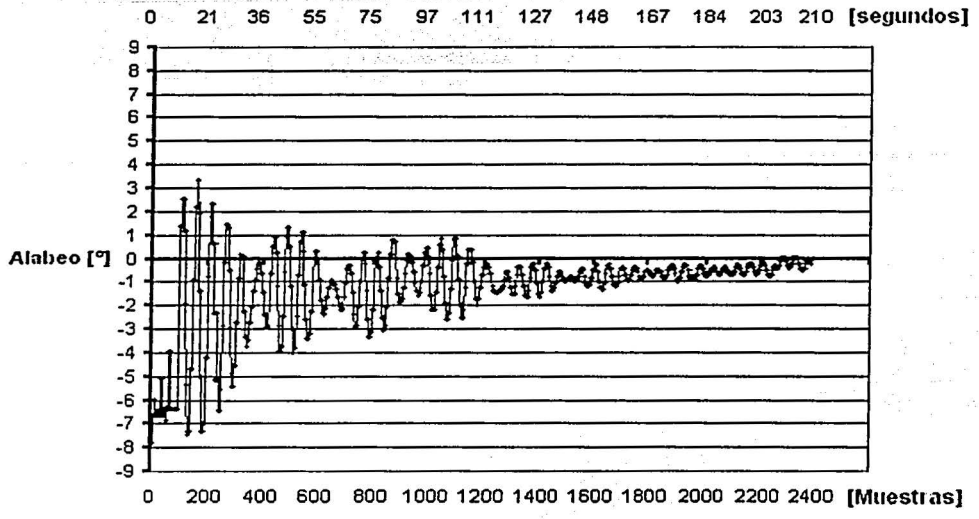


Figura 6.2(a) Balanceo en el eje de alabeo.

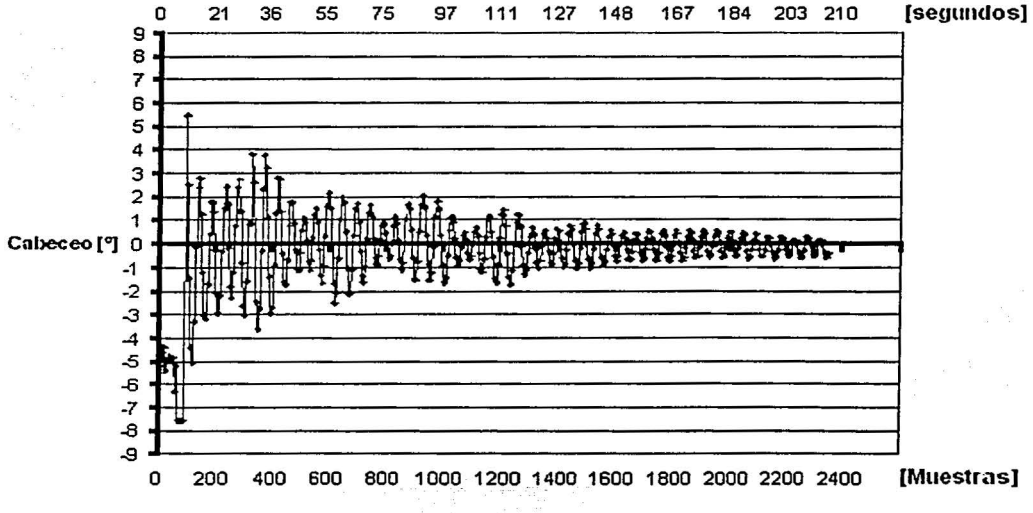


Figura 6.2(b) Balanceo en el eje de cabeceo

Aunque el eje de guiñada no es controlado en la operación de balanceo automático, se ha graficado para comprobar el funcionamiento de dicho eje, que será utilizado en las pruebas de control de estabilización de satélites.

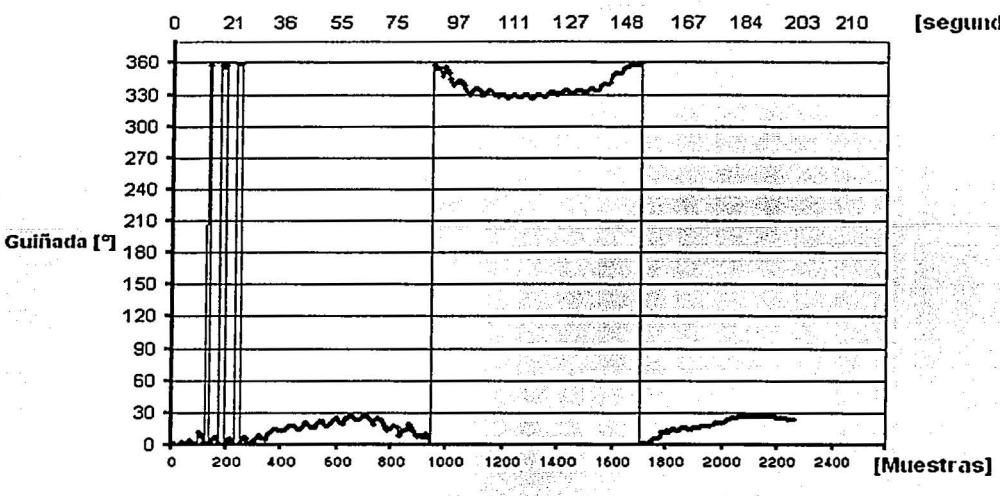


Figura 6.2(c) Comportamiento en el eje de guiñada.

Como se puede apreciar fácilmente en las gráficas, el comportamiento del sistema es bastante satisfactorio, y se han cumplido cabalmente todas las expectativas generadas al inicio del proyecto.

A continuación se muestran las gráficas del comportamiento de la plataforma ya balanceada automáticamente, cuando se le dió un impulso inicial para evaluar el tiempo que tardan en dejar de oscilar. Esto se hizo para los ejes balanceados.

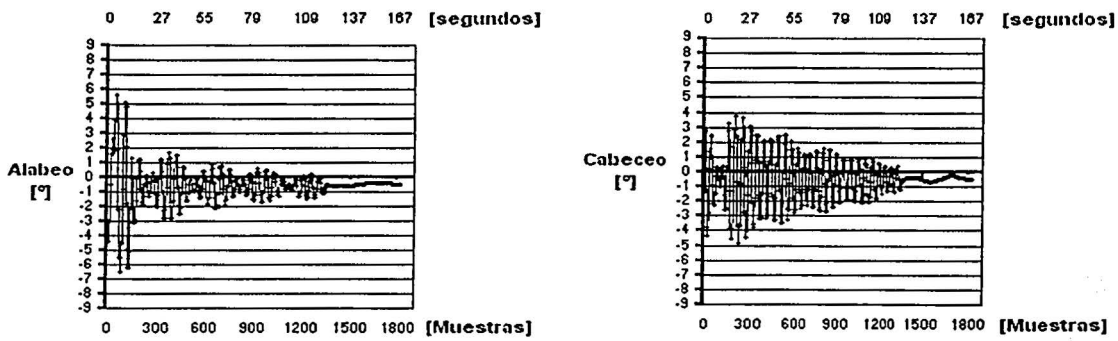


Figura 6.3 Oscilación en los ejes de alabeo y cabeceo, una vez balanceada la plataforma.

6.2 CONCLUSIONES Y RECOMENDACIONES.

A partir del desarrollo del presente trabajo se pueden desprender las siguientes conclusiones:

- Se ha diseñado e implementado un sistema de control automático, para desplazar la posición de las masas deslizantes dentro de la plataforma de simulación, lo que significa que es posible balancearla de manera automática.
- Se ha diseñado, construido y probado un sistema de comunicación inalámbrico, unidireccional, entre la plataforma y una PC, para la visualización y el almacenamiento de los datos de orientación del simulador.
- Se ha contribuido al desarrollo de tecnología e investigación en México, en el área de ingeniería espacial.
- Se han establecido las bases para futuros ensayos de sistemas de orientación de satélites, los cuales deben ser sometidos a extensivas comprobaciones de funcionamiento antes de ser lanzados al espacio; cabe destacar que las pruebas físicas de orientación no pueden realizarse si no se cuenta con una plataforma sin fricción y correctamente balanceada, simulando la condición orbital más importante desde el punto de vista de sistemas dinámicos.
- Se llevaron a cabo pruebas de funcionamiento del sistema de balanceo automático, cumpliendo cabalmente con los objetivos de lograr una capacidad de estabilización de la plataforma menor a $\pm$ 1 grado, lo que se traduce en un par residual de 1.2066×10^{-3} [N m].
- Se concluye, basándonos simplemente en las pruebas de balanceo, que el sistema de control es estable e invariante en el tiempo.

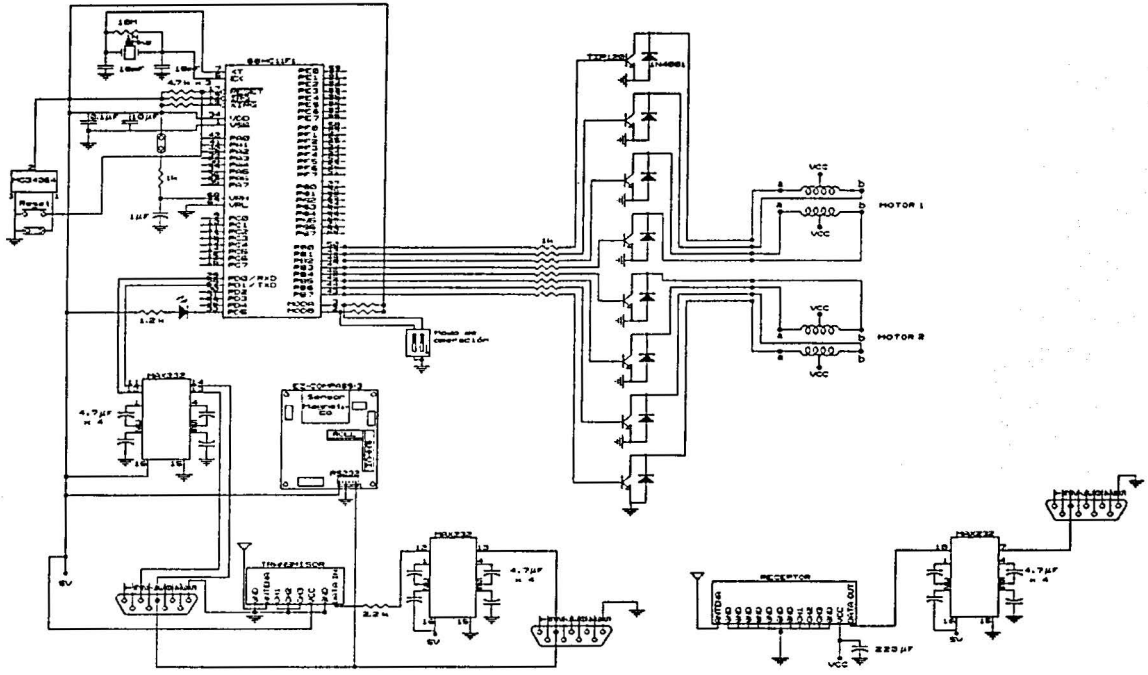
Para dar continuidad a este trabajo es pertinente hacer las siguientes recomendaciones:

- Utilizar aire doblemente filtrado durante la operación del simulador físico de un medio sin fricción, ya que las partículas de aceite y agua que comúnmente se encuentran en las líneas de aire a presión, llegan a causar problemas en el desempeño del balero, como la disminución de la carga máxima de simulación y el aumento en la fricción del dispositivo.
- Separar el módulo de transmisión de otros circuitos, porque le causa ruido el estar en la misma tarjeta que la electrónica de potencia.
- Manufacturar de manera profesional las tarjetas de los circuitos impresos, para que tengan mejor calidad de operación.
- Lubricar las masas deslizantes frecuentemente para evitar la fricción.

Apéndice

A

DIAGRAMA ELECTRÓNICO DEL SISTEMA DE CONTROL



Apéndice

B

PROGRAMA DE CONTROL

```
********Programa de balanceo automático de un simulador físico en un medio sin fricción,
        utilizado para efectuar pruebas de control de orientación de satélites********

*Registros del SCI
BAUD    EQU   $2B
SCCR1   EQU   $2C
SCCR2   EQU   $2D
SCSR    EQU   $2E
SCDR    EQU   $2F

*Constantes
PORTB   EQU   $1004
ITERA   EQU   $0A          ;localidad en donde se guarda el número de
iteraciones
DATREC  EQU   $0B          ;localidad donde se guardan los datos recibidos
CTAPASOS EQU  $0C          ;localidad donde se guardan los pasos que girarán los
                           ;motores
        ORG   $FE00
        LDX   #$1000       ;para acceder a los reg. del SCI
        LDS   #$03FF       ;inicializamos el stack para las subrutinas

*Configuración del SCI
        LDAA  #$30
        STAA  BAUD,X       ;velocidad de transmisión, 9600 bauds
        LDAA  #$00
        STAA  SCCR1,X      ;8 bits de datos
        LDAA  #$04
        STAA  SCCR2,X      ;inhibir interrupciones y activar receptor

*Contador de iteraciones
        LDAA  #$14         ;realiza la compensación 20 veces
        STAA  ITERA

*Compensación en Roll
INI     BRCLR SCSR,X,$20,INI ;espera hasta que llegue un caracter
        LDAA  SCDR,X       ;lee el dato recibido
        CMPA  #'R'         ;¿se ha recibido un roll?
        BEQ   ROLL         ;va a la rutina roll
        BRA   INI
ROLL    BRCLR SCSR,X,$20,ROLL  ;espera el primer dato de roll
        LDAA  SCDR,X       ;lee el dato recibido
        CMPA  #'-'         ;¿el dato es negativo?
        BEQ   NEGATIVO     ;si lo es, va a una rutina para signarlo
```

```
                CMPA    # '0'                ;¿está compensada la plataforma?
                BEQ     RECIBE              ;si lo está, va a compensar en pitch
                ANDA    #$0F                ;si no, convierte el dato a hexadecimal
SIGUE           STAA    DATREC              ;guarda el dato en memoria
MOTOR1          BRSET   DATREC,$80,RUTINA1  ;lleva a la rutina para mover la mesa 1, en
                                            ;sentido antihorario

*Rutina para mover la mesa 1, en sentido horario
RUTINA2         LDY     #TABLA2             ;apuntamos a la tabla2: $90,$A0,$60,$50
                LDAA    DATREC
                BSR     PASOS
                BRA     RECIBE

*Rutina para mover la mesa 1, en sentido antihorario
RUTINA1         LDY     #TABLA1             ;apuntamos a la tabla1: $50,$60,$A0,$90
                LDAA    DATREC
                ANDA    #$0F
                BSR     PASOS

*Compensación en pitch
RECIBE          BRCLR   SCSR,X,$20,RECIBE   ;espera hasta que llegue un caracter
                LDAA    SCDR,X              ;lee el dato recibido
                CMPA    # 'P'               ;¿se ha recibido un pitch?
                BEQ     PITCH               ;si es así, entonces va a la rutina de pitch
                JMP     RECIBE
PITCH           BRCLR   SCSR,X,$20,PITCH    ;espera el primer dato de pitch
                LDAA    SCDR,X              ;lee el dato recibido
                CMPA    # ' - '             ;¿el dato es negativo?
                BEQ     NEGATIVO1           ;si lo es, va a una rutina para signarlo
                CMPA    # '0'               ;¿está compensada la plataforma?
                BEQ     CICLO               ;si lo está, realiza otra iteración
                ANDA    #$0F                ;si no, convierte el dato a hexadecimal
SIGUE1          STAA    DATREC              ;guarda el dato en memoria
                BRA     MOTOR2
MOTOR2          BRSET   DATREC,$80,RUTINA3  ;lleva a rutina para mover la mesa 2, en
                                            ;sentido antihorario

*Rutina para mover la mesa 2, en sentido horario
RUTINA4         LDY     #TABLA4             ;apuntamos a la tabla4: $09,$0A,$06,$05
                LDAA    DATREC
                BSR     PASOS
                BRA     CICLO

*Rutina para mover la mesa 2, en sentido antihorario
RUTINA3         LDY     #TABLA3             ;apuntamos a la tabla3: $05,$06,$0A,$09
                LDAA    DATREC
                ANDA    #$0F
                BSR     PASOS
```

```
*Ajuste de la plataforma
CICLO       DEC     ITERA
            LDAA    ITERA
            CMPA    #$00
            BNE     INI
FIN         BRA     FIN

*Asignación de un "1" en el bit 7 para considerar que el dato es negativo
NEGATIVO    BRCLR   SCSR,X,$20,NEGATIVO  ;espera el siguiente caracter
            LDAA    SCDR,X              ;lee el dato recibido
            CMPA    # '0'               ;¿está compensada la plataforma?
            BEQ     RECIBE              ;si lo está, va a compensar en pitch
            ANDA    #$0F                ;si no, convierte el dato a hexadecimal
            ADDA    #$80                ;signa el dato
            JMP     SIGUE               ;manda el dato a mover la mesa 1

NEGATIVO1   BRCLR   SCSR,X,$20,NEGATIVO1 ;espera el siguiente caracter
            LDAA    SCDR,X              ;lee el dato recibido
            CMPA    # '0'               ;¿está compensada la plataforma?
            BEQ     CICLO               ;si lo está, inicia la compensación
            ANDA    #$0F                ;si no, convierte el dato a hexadecimal
            ADDA    #$80                ;signa el dato
            JMP     SIGUE1              ;manda el dato a mover la mesa 2

*Rutina que realiza la compensación, y mueve los motores en pasos completos
PASOS       PSHY
            LDAB    #$10                ;constante de compensación
            MUL                         ;algoritmo para realizar la compensación
UNO         LDAA    #$04
            STAA    CTAPASOS
            PULY
            PSHY
DOS         LDAA    0,Y                 ;secuencias 0101,0110,1010,1001 por paso
            JSR     PAUSA               ;va a una rutina de retardo entre cada paso
            INY
            DEC     CTAPASOS
            BNE     UNO
            DECB
            BNE     DOS
            PULY
TIEMPO      PSHX
            PSHY
            LDX     #$2C                ;retraso para esperar que se estabilice la plataforma
T2          LDY     #$HF
T1          DEY
            BNE     T1
            DEX
            BNE     T2
            PULY
            PULX
            RTS
```

```
*Retardo de 10ms entre cada paso de los motores
PAUSA   PSHX
        STAA    PORTB           ;puerto de salida de los datos
        LDX     #$0B29          ;retardo de 10ms entre paso
TRES    DEX
        BNE     TRES
        PULX
        RTS

*Secuencias de datos para mover los motores
TABLA1  DB $50,$60,$A0,$90
TABLA2  DB $90,$A0,$60,$50
TABLA3  DB $05,$06,$0A,$09
TABLA4  DB $09,$0A,$06,$05

        ORG     $FFFE
        DW      $FE00
```

Apéndice

C

PROGRAMA DE ALMACENAMIENTO DE DATOS Y GRAFICACIÓN

```
//CÓDIGO DE LA PANTALLA PRINCIPAL

unit main;
interface
uses
Windows, Messages, SysUtils, Classes, Graphics,
Controls, Forms, Dialogs, StdCtrls, ExtCtrls, Db,
DBTables, DBCtrls, Grids, DBGrids, Buttons,
ComPort, Menus;

type
TFrmNewton = class(TForm)
Timer1: TTimer;
Panel1: TPanel;
Valores: TButton;
Label7: TLabel;
Label8: TLabel;
Label9: TLabel;
Edit1: TEdit;
Edit2: TEdit;
Edit3: TEdit;
Panel2: TPanel;
ResetCube: TButton;
XRot: TScrollBar;
YRot: TScrollBar;
ZRot: TScrollBar;
Xpositivo: TCheckBox;
yPositivo: TCheckBox;
ZPositivo: TCheckBox;
Activar: TButton;
DatosEjes: TTable;
DataSource1: TDataSource;
DatosEjesNumeroEvento: TAutoIncField;
DatosEjesFecha: TDateField;
h: TFloatField;
DatosEjesEjeY: TFloatField;
DatosEjesEjeZ: TFloatField;
Tiempo: TLabel;
Repeticion: TTimer;
DatosEjesHora: TFloatField;
ComPort1: TComPort;
MainMenu1: TMainMenu;
MenuActivacion: TMenuItem;
Configuracion1: TMenuItem;
Comunicacines1: TMenuItem;
Acercade1: TMenuItem;
Angulodevisualizacion1: TMenuItem;
Panel4: TPanel;
Panel3: TPanel;
DBGrid1: TDBGrid;
DBNavigator1: TDBNavigator;
Label3: TLabel;
Label2: TLabel;
Panel5: TPanel;
Label1: TLabel;
Memo1: TMemo;
TablaCamaras: TTable;
FteTablaCamaras: TDataSource;
puertos: TTable;
DatosEjesHoraNormal: TTimeField;
procedure FormCreate(Sender: TObject);
procedure FormClose(Sender: TObject;
var Action: TCloseAction);
procedure Timer1Timer(Sender: TObject);
procedure ResetCubeClick(Sender: TObject);
procedure ActivarClick(Sender: TObject);
procedure ValoresClick(Sender: TObject);
Procedure RotacionPrimerCubo;
Procedure RotacionSegundoCubo;
procedure SpeedButton1Click(Sender: TObject);
procedure SegunderoTimer(Sender: TObject);
procedure FormActivate(Sender: TObject);
procedure DBGrid1CellClick(Column: TColumn);
procedure Button1Click(Sender: TObject);
procedure ComPort1RxChar(Sender: TObject);
Procedure InterpretaCadena(Cadena : String);
procedure MenuActivacionClick(Sender: TObject);
procedure Angulodevisualizacion1Click (Sender:
TObject);
procedure Comunicacines1Click(Sender: TObject);
private
{ Private declarations }
public
{ Public declarations }
end;

var
FrmNewton: TFrmNewton;

implementation
uses
frmCamaras, Comunica;
{$R *.DFM}
Type

//Se usa para crear una matriz de rotación
Matrix = Array [0..3,0..3] of real;
//Estructura para almacenar un punto de 3d
TDPoint = Record
X : Real;
Y : Real;
```

```
Z : Real;
end;
Var
//Lo usamos para dibujar aquí y luego hacer el copyrect
DoubleBuffer : TBitMap;
//Mapa de bits para almacenar antecedentes
BlankBuffer  : TBitMap;
//Puntos Rotados
PntsOut      b : Array [1..8] of TDPoint;
PntsOut02    : Array [1..8] of TDPoint;
//Representación de puntos de 2 dimensiones
TPPnts       : Array [1..8] of TPoint;
TPPnts02     : Array [1..8] of TPoint;
//Imagen Original
Pnts         : Array [1..8] of TDPoint;
Pnts02       : Array [1..8] of TDPoint;
XAng         : Real;
YAng         : Real;
ZAng         : Real;
//Lo usamos para dibujar aquí y luego hacer el copyrect
DoubleBuffer02 : TBitMap;
//Mapa de bits para almacenar antecedentes
BlankBuffer02  : TBitMap;
BanderaGiroAutomatico : Boolean;
CadenaLectura        : String;
ValorCamara2X : Integer;
ValorCamara2Y : Integer;
ValorCamara2Z : Integer;

//Crea un arreglo o matriz que establece la rotación con base en ángulos que pasan en términos de radianes
Procedure MatrixRotate (Var m: Matrix; x,y,z : Real);
var
SinX, CosX,
SinY, CosY,
//Guardar en este punto para que sólo tengamos que calcular los ciclos una vez
SinZ, CosZ : Real ;
c1,  C2   : Integer;
begin
SinX := sin(x);
CosX := Cos(x);
SinY := sin(y);
CosY := Cos(y);
SinZ := sin(Z);
CosZ := Cos(Z);
//Asigna la matriz a la identidad
For C1 := 0 to 3 do
      For C2 := 0 to 3 do
        if C1= C2 then
            m[C1,C2] := 0
        else
            m[C1,C2] := 1;
M[0,0] := (CosZ * CosY);
M[0,1] := (CosZ * -SinY * -SinX + sinZ * CosX);
M[0,2] := (CosZ * -SinY * CosX + sinZ * SinX);
M[1,0] := (-SinZ * cosY);
M[1,1] := (-SinZ * -SinZ * -sinX + cosZ * CosX);
M[1,2] := (-SinZ * -Siny * CosX + cosZ * SinX);
M[2,0] := (SinY);
M[2,1] := (cosY * -SinX);
M[2,2] := (CosY * CosX);
end;

//Aplica la matriz de rotación a un punto de 3d y retorna un nuevo punto de 3d
Procedure ApplyMatToPoint (PointIn : TdPoint; Var PointOut : TdPoint ; Mat : Matrix);
var
x,y,z: Real;
begin
X := (PointIn.X * mat[0,0]) + (PointIn.y * mat[0,1]) + (PointIn.z * mat[0,2]) + mat[0,3];
Y := (PointIn.X * mat[1,0]) + (PointIn.y * mat[1,1]) + (PointIn.z * mat[1,2]) + mat[1,3];
Z := (PointIn.X * mat[2,0]) + (PointIn.y * mat[2,1]) + (PointIn.z * mat[2,2]) + mat[2,3];
pointOut.x := X;
pointOut.y := y;
pointOut.Z := Z;
end;

//Se establecen las coordenadas de cada punto. La mitad del cubo esta en (0,0,0)
Procedure InitCube;
begin
    Pnts[1].X  := -50;
    Pnts[1].y  := -50;
    Pnts[1].Z  := -50;
    Pnts[2].X  := 50;
    Pnts[2].Y  := -50;
    Pnts[2].Z  := -50;
    Pnts[3].X  := 50;
    Pnts[3].y  := 50;
    Pnts[3].Z  := -50;
    Pnts[4].X  := -50;
    Pnts[4].y  := 50;
    Pnts[4].Z  := -50;
    Pnts[5].X  := -50;
    Pnts[5].Y  := -50;
    Pnts[5].Z  := 50;
    Pnts[6].X  := 50;
    Pnts[6].Y  := -50;
    Pnts[6].Z  := 50;
    Pnts[7].X  := 50;
    Pnts[7].Y  := 50;
    Pnts[7].Z  := 50;
    Pnts[8].X  := -50;
    Pnts[8].Y  := 50;
    Pnts[8].Z  := 50;
end;

Procedure InitCube02;
begin
   Pnts02[1].X  := -50;
   Pnts02[1].y  := -50;
   Pnts02[1].Z  := -50;
   Pnts02[2].X  :=  50;
   Pnts02[2].Y  := -50;
   Pnts02[2].Z  := -50;
   Pnts02[3].X  :=  50;
```

```
    Pnts02[3].y  :=  50;
    Pnts02[3].Z  := -50;
    Pnts02[4].X  := -50;
    Pnts02[4].y  :=  50;
    Pnts02[4].Z  := -50;
    Pnts02[5].X  := -50;
    Pnts02[5].Y  := -50;
    Pnts02[5].Z  :=  50;
    Pnts02[6].X  :=  50;
    Pnts02[6].Y  := -50;
    Pnts02[6].Z  :=  50;
    Pnts02[7].X  :=  50;
    Pnts02[7].Y  :=  50;
    Pnts02[7].Z  :=  50;
    Pnts02[8].X  := -50;
    Pnts02[8].Y  :=  50;
    Pnts02[8].Z  :=  50;
end;

//La siguiente función retorna verdadero si la suma
de los parámetros es mayor que cero; y falso si es
menor de 0. Usaremos esta función para determinar
que lados del cubo se deben mostrar y cuales hay
que ocultar
function ShowSide (V1,v2,v3,v4 : Real): Boolean;
begin
   If (V1+v2+v3+v4) > 0 then
     ShowSide := True
   else
      ShowSide := false;
end;

//Estamos usando un buffer doble. Esta función
determina si un lado es visible; si lo es, dibuja la
representación de dos dimensiones de nuestro
buffer de mapa de bits con el relleno configurado al
color pasado
procedure AddSide (P1,P2,P3,P4 : Integer; sidecolor :
Tcolor);
begin
   if
ShowSide(pntsOut[P1].Z,pntsOut[P2].Z,pntsOut[P3].Z,p
ntsOut[P4].Z) then
   begin
      DoubleBuffer.Canvas.brush.Color := sidecolor;

DoubleBuffer.Canvas.Polygon([TPPnts[P1],TPPnts[P2],
TPPnts[P3],TPPnts[P4],TPPnts[P1]]);
   end;
end;

procedure AddSide02 (P1,P2,P3,P4 : Integer; sidecolor :
Tcolor);
begin
   if
ShowSide(pntsOut02[P1].Z,pntsOut02[P2].Z,pntsOut02[
P3].Z,pntsOut02[P4].Z) then
   begin
      DoubleBuffer02.Canvas.brush.Color := sidecolor;

DoubleBuffer02.Canvas.Polygon([TPPnts02[P1],TPPnts
02[P2],TPPnts02[P3],TPPnts02[P4],TPPnts02[P1]]);
   end;
end;

Procedure HabilitarControl (Bandera : boolean);
begin
   FrmNewton.xrot.Enabled := not Bandera;
   FrmNewton.Yrot.Enabled := not Bandera;
   FrmNewton.Zrot.Enabled := not Bandera;
   FrmNewton.xpositivo.Enabled := not Bandera;
   FrmNewton.ypositivo.Enabled := not Bandera;
   FrmNewton.Zpositivo.Enabled := not Bandera;

   FrmNewton.Edit1.Enabled := Bandera;
   FrmNewton.Edit2.Enabled := Bandera;
   FrmNewton.Edit3.Enabled := Bandera;
   FrmNewton.Valores.Enabled := Bandera;
end;

//Al cargar la forma, crea e inicializa nuestro mapa
de bits antecedentes e inicializa nuestro mapa de
bits del doble buffer

procedure TFrmNewton.FormCreate(Sender: TObject);
Var
  Velocidad : TBaudRate;
  Puerto    : String;
  Datos     : TDataBits;
  Parada    : TStopBits;
  Paridad   : TParity;

begin
   DoubleBuffer := TBitmap.Create;
   DoubleBuffer.Height := 200;
   DoubleBuffer.Width  := 200;
   BlankBuffer :=  TBitmap.Create;
   BlankBuffer.Height := 200;
   BlankBuffer.Width  := 200;
   BlankBuffer.Canvas.brush.Color := ClWhite;
   BlankBuffer.Canvas.Rectangle(0,0,200,200);

   DoubleBuffer02 := TBitmap.Create;
   DoubleBuffer02.Height := 200;
   DoubleBuffer02.Width  := 200;
   BlankBuffer02 :=  TBitmap.Create;
   BlankBuffer02.Height := 200;
   BlankBuffer02.Width  := 200;
   BlankBuffer02.Canvas.brush.Color := ClWhite;
   BlankBuffer02.Canvas.Rectangle(0,0,200,200);

   InitCube();
   InitCube02();

   XAng := 0;
   YAng := 0;
   ZAng := 0;
   HabilitarControl(True);
   DatosEjes.open;
   TablaCamaras.Open;
   puertos.Open;

//Para  obtener el valor de las cámaras
   TablaCamaras.First;
```

```
    if TablaCamaras.FindKey(['2']) then
    begin
      ValorCamara2X                    :=
TablaCamaras.fieldbyname('EjeX').AsInteger;
      ValorCamara2Y                    :=
TablaCamaras.fieldbyname('EjeY').AsInteger;
      ValorCamara2Z                    :=
TablaCamaras.fieldbyname('EjeZ').AsInteger;
    end
    else
    begin
      ValorCamara2X := 0;
      ValorCamara2Y := 0;
      ValorCamara2Z := 0;
    end;

//Para obtener la configuración del puerto serie
  IF PUERTOS.FindKey(['Newton']) THEN
  begin
     if  PUERTOS.FieldByName('Velocidad').AsString  =
'300' then Velocidad := br300;
     if  PUERTOS.FieldByName('Velocidad').AsString  =
'1200' then Velocidad := br1200;
     if  PUERTOS.FieldByName('Velocidad').AsString  =
'2400' then Velocidad := br2400;
     if  PUERTOS.FieldByName('Velocidad').AsString  =
'4800' then Velocidad := br4800;
     if  PUERTOS.FieldByName('Velocidad').AsString  =
'9600' then Velocidad := br9600;
     if  PUERTOS.FieldByName('Velocidad').AsString  =
'19200' then Velocidad := br19200;
     if  PUERTOS.FieldByName('Velocidad').AsString  =
'38400' then Velocidad := br38400;

     Puerto                                :=
PUERTOS.FieldByName('Puerto').AsString;
     if  PUERTOS.FieldByName('Datos').AsString  =  '7'
then Datos := db7;
     if  PUERTOS.FieldByName('Datos').AsString  =  '8'
then Datos := db8;

     if  PUERTOS.FieldByName('Parada').AsString  =  '1'
then Parada := sb1;
     if  PUERTOS.FieldByName('Parada').AsString  =  '2'
then Parada := sb2;

     if   PUERTOS.FieldByName('Paridad').AsString   =
'NONE' then Paridad := paNone;
     if   PUERTOS.FieldByName('Paridad').AsString   =
'MARK' then Paridad := paMark;
     if   PUERTOS.FieldByName('Paridad').AsString   =
'SPACE' then Paridad := paSpace;
     ComPort1.BaudRate   := Velocidad;
     ComPort1.DeviceName :=  Puerto;
     ComPort1.DataBits   := Datos;
     ComPort1.StopBits   := Parada;
     ComPort1.Parity     := Paridad;
  end;
//Al terminar se necesitan liberar los recursos de los
mapas de bits
procedure   TFrmNewton.FormClose(Sender:   TObject;
var Action: TCloseAction);
begin
    BlankBuffer.free;
    DoubleBuffer.free;

    BlankBuffer02.free;
    DoubleBuffer02.free;
    DatosEjes.close;
    TablaCamaras.Close;
    puertos.close;
    ComPort1.Close;
end;

procedure TFrmNewton.RotacionPrimerCubo;
var
//La matriz utilizada para rotar el cubo
  M : Matrix;
//Se usan para efectuar los ciclos a través de los
puntos
  count, Count2 : Integer;
  Valorx, ValorY,ValorZ : integer;
  Present  : TDateTime;
  Hour, Min, Sec, MSec: Word;

begin
   if BanderaGiroAutomatico then
    begin
       Valorx := XRot.Position;
       ValorY := YRot.Position;
       ValorZ := ZRot.Position;
   end
   else
   begin
       Valorx := 0;
       ValorY := 0;
       ValorZ := 0;
   end;

   if not Xpositivo.Checked then
     Valorx := Valorx * -1;
   if not ypositivo.Checked then
     ValorY := ValorY * -1;
   if not Zpositivo.Checked then
     ValorZ := ValorZ * -1;

   XAng := XAng + ValorX;
   YAng := YAng + ValorY;
   ZAng := ZAng + ValorZ;

   present := Now;
   DecodeTime(Present, Hour, Min, Sec, MSec);


//INGRESO LOS VALORES LEIDOS A LA BASE DE DATOS

   DatosEjes.Insert;
   DatosEjes.FieldByName('Fecha').AsString          :=
DateToStr(Date);
  DatosEjes.FieldByName('Hora').AsFloat := Msec + (sec
* 1000) + (Min  * 1000 * 60) + (Hour * 1000 * 60 * 60);

   DatosEjes.FieldByName('EjeX').Asfloat  := XAng;
```

```
    DatosEjes.FieldByName('EjeY').Asfloat   := YAng;
    DatosEjes.FieldByName('EjeZ').Asfloat   := ZAng;
    DatosEjes.FieldByName('HoraNormal').AsString    :=
TimeToStr(Time);
    DatosEjes.Post;

    Tiempo.Caption := timeToStr(Time);
//Ajusta los grados y construye la matriz de rotación
    MatrixRotate(M,   (pi*XAng)/180,(pi*YAng)   /   180,
(pi*ZAng) / 180);

//Hace un ciclo a través de todos los puntos y gira
para obtener la representación en dos dimensiones

    for Count2 := 1 to 8 do
    begin

ApplyMatToPoint(PNTS[Count2],pntsOut[Count2],M);
      TPPnts[Count2]                              :=
Point(Trunc(PntsOut[Count2].x+100),Trunc(PntsOut[Cou
nt2].Y+100));
    end;

//Borra el buffer doble copiando el antecedente con
copy rect

DoubleBuffer.Canvas.CopyRect(rect(0,0,200,200),Blank
Buffer.Canvas,Rect(0,0,200,200));
//Construye el cubo llamado addside para cada uno
de los seis lados
    AddSide(1,2,3,4,clblue);
    AddSide(5,6,7,8,clred);
    AddSide(1,2,6,5,clYellow);
    AddSide(2,3,7,6,clGreen);
    AddSide(3,4,8,7,clPurple);
    AddSide(4,1,5,8,clSilver);
//Copia el buffer doble a la forma
//Aquí es para mover la posición del cubo
FrmNewton.Canvas.CopyRect(rect(0,0,200,200),Double
Buffer.canvas,rect(0,0,200,200));

XRot.Position := ValorxTemp ;
YRot.Position := ValorYTemp ;
ZRot.Position := ValorZTemp ;

XAng := XAngTemp;
YAng := YAngTemp;
ZAng := ZAngTemp;
end;

procedure TFrmNewton.Timer1Timer(Sender: TObject);
begin
RotacionPrimerCubo;
RotacionSegundoCubo;
end;

procedure TFrmNewton.ResetCubeClick(Sender:
TObject);
begin
XAng := 0;
YAng := 0;
ZAng := 0;
Timer1Timer(Timer1);
end;

procedure TFrmNewton.ActivarClick(Sender: TObject);
begin
BanderaGiroAutomatico := True;
if Timer1.Enabled then
begin
Timer1.Enabled := False;
Activar.Caption := 'Activar';
ComPort1.Close;
HabilitarControl(True);

end
else
begin
TablaCamaras.First;
if TablaCamaras.FindKey(['2']) then
begin
ValorCamara2X :=
TablaCamaras.fieldbyname('EjeX').AsInteger;
ValorCamara2Y :=
TablaCamaras.fieldbyname('EjeY').AsInteger;
ValorCamara2Z :=
TablaCamaras.fieldbyname('EjeZ').AsInteger;
end
else
begin
ValorCamara2X := 0;
ValorCamara2Y := 0;
ValorCamara2Z := 0;
end;
ComPort1.Open;
Timer1.Enabled := True;
Activar.Caption := 'Desactivar';
HabilitarControl(False);
end;
end;

procedure TFrmNewton.ValoresClick(Sender: TObject);
Var
xTemp,YTemp,ZTemp : real;
Valor,Code : Integer;
DatosCorrectos      : Boolean;
begin

procedure TFrmNewton.RotacionSegundoCubo;
var

//La matriz utilizada para rotar el cubo
  M : Matrix;
//Se usan para efectuar los ciclos a través de los
puntos
  count, Count2 : Integer;
  Valorx, ValorY,ValorZ : integer;
  ValorxTemp, ValorYTemp,ValorZTemp : integer;
  XAngTemp,
  YAngTemp,
  ZAngTemp : Real;

begin
    Valorx := xRot.Position ;
```

```
    Valory := yRot.Position;
    ValorZ := zRot.Position;

    ValorxTemp := XRot.Position;
    ValorYTemp := YRot.Position;
    ValorZTemp := ZRot.Position;
    XAng := xAng + Valorx;
    YAng := yAng + Valory;
    ZAng := ZAng + ValorZ;

    XAngTemp := XAng ;
    YAngTemp := YAng ;
    ZAngTemp := ZAng ;

    if not Xpositivo.Checked then
      Valorx := Valorx * -1;
    if not ypositivo.Checked then
      ValorY := ValorY * -1;
    if not Zpositivo.Checked then
      ValorZ := ValorZ * -1;

//Ajusta los grados y construye la matriz de rotación
MatrixRotate(M,        (pi*(XAng+        ValorCamara2X)
)/180,(pi*(YAng+ValorCamara2y))/  180,  (pi*  (Zang  +
ValorCamara2Z)) / 180);
//Hace un ciclo a través de todos los puntos y gira
para obtener la representacion en dos dimensiones
    for Count2 := 1 to 8 do
    begin

ApplyMatToPoint(PNTS02[Count2],pntsOut02[Count2],
M);
      TPPnts02[Count2]                          :=
Point(Trunc(PntsOut02[Count2].x+100),Trunc(PntsOut0
2[Count2].Y+100));
    end;

//Borra el buffer doble copiando el antecedente con
copy rect

DoubleBuffer02.Canvas.CopyRect(rect(00,0,200,200),Bl
ankBuffer02.Canvas,Rect(0,0,200,200));

//Voy a validar primero edits correctos
    DatosCorrectos := True;
    Val(Edit1.Text,XTemp,Code);
    if Code<> 0 then
    Begin

procedure      TFrmNewton.SpeedButton1Click(Sender:
TObject);
Var
  Present   : TDateTime;
  Hour, Min, Sec, MSec: Word;
begin
//Voy a leer de los datos y después restauro la
animación. Obtengo el primer valor de donde me
encuentro posicionado actualmente

DecodeTime(DatosEjes.FieldByName('Hora').AsDateTi
me, Hour, Min, Sec, MSec);
end;

procedure      TFrmNewton.SegunderoTimer(Sender:
TObject);
begin
//Voy a dibujar el segundero conforme se cumpla el
ciclo;

end;

procedure TFrmNewton.FormActivate(Sender: TObject);
begin

//ComPort1.Open;
end;

procedure      TFrmNewton.DBGrid1CellClick(Column:
TColumn);
begin
//Voy a convertir la hora guardada en milésimas de
segundo a un formato reconocible

end;

procedure TFrmNewton.Button1Click(Sender: TObject);
Var
  Hour, Min, Sec, MSec : Word;
  ValorSeg          : Longint;
begin
    ValorSeg                                        :=
DatosEjes.FieldByName('Hora').AsInteger;
    Hour    := (ValorSeg div 1000);
    Hour    := (Hour div 60);
    Hour    := (Hour div 60);

    ValorSeg                                        :=
DatosEjes.FieldByName('Hora').AsInteger ;
    Min     := (ValorSeg div 1000);
    Min     := (Min div 60);

end;


Procedure   TFrmNewton.InterpretaCadena(Cadena   :
String);
Var
  LecturaX,
  LecturaY,
  LecturaZ          : String;
  BanderaInicio     : Boolean;
  BanderaFin        : Boolean;
  Contador          : Integer;
  Longitud          : integer;
  CadenaCorrecta    : Boolean;
  TresPuntosOk      : Boolean;
  PrimerPuntoOk     : Boolean;
  SegundoPuntoOk    : Boolean;
  TercerPuntoOk     : Boolean;
  PosicionBanderaInicio : integer;
  PosicionBanderaFin  : Integer;
  TamanoLectura     : Integer;
  PosicionPunto     : Integer;
```

```
  PosicionInicio      : Integer;
  PosicionFinal       : Integer;
  EncontrePunto       : Boolean;
begin
   LecturaX := '';
   LecturaY := '';
   LecturaZ := '';
//Voy a obtener el valor String de X; está entre la R y
la P por lo cual las voy a buscar primero
   BanderaInicio  := False;
   BanderaFin     := False;
   CadenaCorrecta := true;
   Longitud       := length(Cadena);
   PosicionBanderaInicio := 0;
   PosicionBanderaFin    := 0;
   tamanolectura         := 0;

//Primero voy a validar que en la cadena existan tres
puntos
   Contador      := 1;
   TresPuntosOk  := False;
   PrimerPuntoOk  := False;
   SegundoPuntoOk := False;
   TercerPuntoOk  := False;

   repeat
      if Cadena[Contador] = '.' then
      begin
         if PrimerPuntoOk = false then
           PrimerPuntoOk := True
         else
           if SegundoPuntoOk = False then
             SegundoPuntoOk := true
           else
             TercerPuntoOk := True;
      end;
      if PrimerPuntoOk and SegundoPuntoOk and
TercerPuntoOk then TresPuntosOk := True;
      Contador := Contador + 1;
   until (Contador= Longitud) or TresPuntosOk;

   if TresPuntosOk then
   begin
//Función que busca la "R"
//Me posiciono al principio de la cadena
      contador := 1;
//Salto el caracter correspondiente a la "R"
      PosicionInicio := 2;
      PosicionFinal  := 0;
//Voy a buscar el primer punto
      EncontrePunto := False;
      repeat
         PosicionFinal := PosicionFinal + 1;
         if Cadena[PosicionFinal] = '.' then
           EncontrePunto := true;
      until   EncontrePunto   or   (PosicionFinal   >=
Longitud);

      if EncontrePunto then
        PosicionFinal := PosicionFinal + 2;
      TamanoLectura := PosicionFinal - 1;
      LecturaX                                        :=
copy(Cadena,PosicionInicio,TamanoLectura);

//Función que busca la 'P'
//voy a posicionarme en la posición inicio
      PosicionInicio := PosicionFinal + 2;
//Voy a buscar el segundo punto
      PosicionFinal  := PosicionInicio;
      EncontrePunto := False;
      repeat
         PosicionFinal := PosicionFinal + 1;
         if Cadena[PosicionFinal] = '.' then
           EncontrePunto := true;
      until   EncontrePunto   or   (PosicionFinal   >=
Longitud);

      if EncontrePunto then
        PosicionFinal := PosicionFinal + 2;
      TamanoLectura := (PosicionFinal -posicionInicio)
+ 1 ;
      LecturaY                                        :=
copy(Cadena,PosicionInicio,TamanoLectura);

//Función que busca la 'C'
//voy a posicionarme en la posición inicio
      PosicionInicio := PosicionFinal + 2;
//Voy a buscar el segundo punto
      PosicionFinal  := PosicionInicio;
      EncontrePunto := False;
      repeat
         PosicionFinal := PosicionFinal + 1;
         if Cadena[PosicionFinal] = '.' then
           EncontrePunto := true;
      until   EncontrePunto   or   (PosicionFinal   >=
Longitud);

      if EncontrePunto then
        PosicionFinal := PosicionFinal + 1;
      TamanoLectura := (PosicionFinal -posicionInicio)
+ 1 ;
      LecturaZ                                        :=
copy(Cadena,PosicionInicio,TamanoLectura);
   end;

   Edit1.Text := LecturaX;
   Edit2.Text := LecturaY;
   Edit3.Text := LecturaZ;
   ValoresClick(Valores);
end;


procedure       TFrmNewton.ComPort1RxChar(Sender:
TObject);
var
  text : string;
  Contador : integer;
begin
 Text := ComPort1.ReadString;
 For contador := 1 to Length(Text) do
 begin
    Memo1.SelText := text[Contador];
```

```
      if (text[Contador] <> '$') or
        (text[Contador] <> '*') then
      begin
         CadenaLectura      :=      CadenaLectura      +
text[Contador];
      end;

      if text[Contador] = '$' then
      begin
         CadenaLectura := '';

      end;
      if text[Contador] = '*' then
      begin
//Termine de armar la cadena Lectura y procedo a
vaciar datos a editx
         InterpretaCadena(CadenaLectura);
      end;
  end;

end;

procedure     TFrmNewton.MenuActivacionClick(Sender:
TObject);
begin
    if MenuActivacion.Caption = 'Activar' then
      MenuActivacion.Caption := 'Desactivar'
    else
       MenuActivacion.Caption := 'Activar';
    ActivarClick(Activar);
end;

procedure
TFrmNewton.Angulodevisualizacion1Click(Sender:
TObject);
begin
    Camaras.ShowModal;
end;

procedure     TFrmNewton.Comunicacines1Click(Sender:
TObject);
begin
Comunicaciones.ShowModal;
end;

end.
```

//INTERFAZ DE COMUNICACIÓN.

```
unit comunica;

interface

uses
  Windows, Messages, SysUtils, Classes, Graphics,
Controls, Forms, Dialogs,
  Db, ExtCtrls, DBCtrls, Grids, DBGrids, DBTables;

type
  TComunicaciones = class(TForm)
    PUERTOS: TTable;
    DBGrid1: TDBGrid;
    DBNavigator1: TDBNavigator;
    DataSource1: TDataSource;
    procedure FormActivate(Sender: TObject);
    procedure  FormClose(Sender:  TObject;  var  Action:
TCloseAction);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Comunicaciones: TComunicaciones;

implementation

{$R *.DFM}


procedure        TComunicaciones.FormActivate(Sender:
TObject);
begin
    PUERTOS.open;
end;

procedure          TComunicaciones.FormClose(Sender:
TObject; var Action: TCloseAction);
begin
PUERTOS.close;
end;

end.
```

//CÁMARAS.

```
unit frmCamaras;

interface

uses
  Windows, Messages, SysUtils, Classes, Graphics,
Controls, Forms, Dialogs,
  ExtCtrls, DBCtrls, Grids, DBGrids, Db, DBTables;

type
  TCamaras = class(TForm)
    Camaras: TTable;
    FteCamaras: TDataSource;
    DBGrid1: TDBGrid;
    DBNavigator1: TDBNavigator;
    procedure FormActivate(Sender: TObject);
    procedure  FormClose(Sender:  TObject;  var  Action:
TCloseAction);
  private
    { Private declarations }
  public
    { Public declarations }
  end;
```

```
var
  Camaras: TCamaras;

implementation

{$R *.DFM}

procedure TCamaras.FormActivate(Sender: TObject);
begin
    Camaras.Open;
end;

procedure  TCamaras.FormClose(Sender:  TObject;  var
Action: TCloseAction);
begin
    Camaras.Close;
end;

end.
```

Apéndice

D

CÁLCULO DEL PAR RESIDUAL.

El momento, o par, proporciona una medida de la tendencia de la fuerza a causar la rotación de un cuerpo alrededor de un punto o eje.

Para nuestro trabajo se considera al par residual como la tendencia de la fuerza de la platina deslizante, que provoca el movimiento de la plataforma en un grado.

Para calcular el par residual, se utiliza la ecuación:

$$M = Fd \ [Nm]$$

Donde:

M: Momento o par [N m]
F: Fuerza [N]
d: Distancia [m]

Masa de la platina deslizante: $m = 0.615$ [kg]
Aceleración de la gravedad: $g = 9.81$ [m/s^2]
Fuerza de la platina deslizante $F = mg = (0.615)(9.81) = 6.033$ [N]

De acuerdo a las pruebas de balanceo automático realizadas sabemos que:

- **40 pasos del motor inclinan a la plataforma un grado**
- **Un paso del motor desplaza la platina 0.005 [mm]**

La distancia que se desplaza la platina para mover la plataforma un grado es:

d = (0.005 [mm])(40 pasos) = 0.2 [mm]

Por lo tanto, el par residual es:

M = F d [N m]
M = (6.033 [N]) (0.2 [mm]) = 1.2066×10^{-3} [N m]

Par residual = 1.2066×10^{-3} [N m]

REFERENCIAS

1. Haussermann W. and Kennel H. "A Satellite Motion Simulator". Astronautics Vol 5 No. 12 December 1960.

2. Rizos I, Arbes J and Raoult J.C. "A Spherical Air-Bearing-Supported Test Facility for Performance Testing of Satellite Attitude Control Systems". ESRO-CR66, also 4th. IFAC Symposium. Dubrovnic, Yugoslavia. Sept. 6-10 1971.

3. J. Prado, A. Peralta-Higuera, G. Bisiacchi. "Simulador Físico para Prueba de Sistemas de Detección de Orientación de Satélites, en un Medio sin Fricción". SOMI XII Congreso Nacional de Instrumentación. San Luis Potosí. México. 1997. Memorias del Congreso pp738-742.

4. Advanced Orientation Systems Inc., "EZ-COMPASS-3 Application Manual". Linden, NJ 1999.

5. Katsuhiko Ogata. "Ingeniería de Control Moderna". Ed. Prentice Hall. México, 1984.

6. M68HC11, "Reference Manual". Motorola. U.S.A., 1991.

7. MC68HC11F1, "Technical Data". Motorola. U.S.A., 1993.

8. Wayne Tomasi. "Sistemas de Comunicaciones Electrónicas". Prentice-Hall Hispanoamericana, S.A. 2ª Edición, 1996.

9. Revista Electro-Electrónica de la Pontificia Universidad Católica del Perú. "Motores de Pasos",. 2do. semestre 1998, N° 10

10. Maloney, Timothy J. "Electrónica industrial moderna". México : Prentice-Hall, 2a ed, 1997

11. Martin James. "Comunication Satellite Systems". Prentice-Hall. U.S.A, 1986.

12. Nicolás Baran. Revista PC/TIPS Byte "Redes Inalámbricas". pag 94-98, Artículo, Abril 1992.

13. Linx Technologies Inc. "Manual de Referencia, Módulo HP-II". Ashley Place, 2001

14. Juárez Durán Alejandro. "Balanceo Automático de un Simulador Para Control De Orientación de Satélites". Tesis de Licenciatura, Facultad de Ingeniería, UNAM 2001

15. Dan Osier, Steve Grobman, Steve Baston. "Aprendiendo Delphi 3", Prentice-Hall, S.A., México, 1998.

16. Juárez Durán Gustavo. "Utilización de bobinas magnéticas para control de orientación de satélites pequeños." Tesis de Licenciatura, Facultad de Ingeniería, UNAM 1999.

17. Prado J., Bisiacchi G., Mesinas M., Ruiz D. 2002. "Sistema de Monitoreo Inalámbrico de un Simulador Para Control de Orientación de Satélites". SOMI XVII Congreso Nacional de Instrumentación. Mérida, Yucatán, México, Octubre 14-18.