

01129
55



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO



FACULTAD DE INGENIERÍA

**"SISTEMA DE ADQUISICIÓN DE IMÁGENES
AÉREAS DIGITALES."**

T E S I S

QUE PARA OBTENER EL TÍTULO DE:
INGENIERO ELÉCTRICO - ELECTRÓNICO

P R E S E N T A:

RAFAEL SANDOVAL VÁZQUEZ

DIRECTOR DE TESIS:

M. en I. JORGE PRADO MOLINA

2003

A



Universidad Nacional
Autónoma de México



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

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

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

PAGINACION

DISCONTINUA

Autorizo a la Dirección General de Bibliotecas de la UNAM a difundir en formato electrónico e impreso el contenido de mi trabajo recepcional.

NOMBRE: Rafael Sandoval

Vázquez

FECHA: 16/ Octubre 12003

FIRMA: [Firma]

P.O. Yesenia Carrera Fournier

DEDICATORIA

A Dios por darme vida y salud para alcanzar y lograr todo lo que me he propuesto.

A mi papá Rafael y a mi mamá Imelda, por haberme proporcionado todo para terminar mis estudios, por sus consejos, por sus regaños, por sus desvelos a mi lado, por su amor, por su comprensión y porque los amo.

A mi hermana Claudia y a mi hermano Ricardo, que han alegrado mi vida y que sin saberlo ellos me han impulsado a conseguir y alcanzar nuevos retos.

A mis abuelos, que me vieron crecer y que fueron una fuente de aliento.

A todos mis tíos y tías, primos y primas con los que he convivido y que me han motivado.

A Yesenia por estar a mi lado estos últimos años de la carrera, por su comprensión y apoyo, por su alegría y amor que tiene para mí.

A todos mis amigos que han hecho mi estancia en la Universidad mucho más agradable: Noelas, Danielo, Andreyo, Jorge, Fabyruchis, Marce, Mago, Javo, Fer, etc.

A todas las personas que por falta de memoria no incluyo, pero que han sido importantes en mi vida.

RECONOCIMIENTOS

A la Universidad Nacional Autónoma de México que me dio la oportunidad de tener una carrera universitaria.

Al Instituto de Geografía de la UNAM que abrió las puertas a este proyecto.

Al M. I. Jorge Prado Molina que bajo su dirección termine este trabajo de tesis.

A la Unidad de Servicios de Cómputo Académico UNICA, que me brindó conocimientos y recursos valiosos para mi desarrollo, así como a todos sus integrantes.

| | |
|--|-----|
| DEDICATORIA | i |
| RECONOCIMIENTOS | ii |
| ÍNDICE | iii |
| PREFACIO | v |
| | |
| 1.- INTRODUCCIÓN. | 1 |
| 1.1.- Sistema de adquisición de imágenes. | 1 |
| 1.2.- Sistema de monitoreo. | 2 |
| 1.3.- Plataformas aéreas para adquisición de imágenes. | 3 |
| 1.3.1.- Plataformas aéreas convencionales. | 4 |
| 1.3.2.- Plataformas para cámaras de formato pequeño. | 5 |
| 1.4.- Sistema de detección de orientación y de control de estabilización. | 6 |
| 1.4.1.- Sistema de navegación inercial. | 6 |
| 1.4.2.- Brújula electrónica. | 7 |
| 1.4.3.- Sistema de control de apuntamiento. | 7 |
| 1.5.- Pruebas de funcionamiento del sistema de adquisición de imágenes. | 8 |
| 1.5.1.- Pruebas de funcionamiento en tierra. | 8 |
| 1.5.2.- Pruebas de funcionamiento en vuelo. | 8 |
| | |
| 2.- PROGRAMA DE MONITOREO PARA PLANEACIÓN Y SUPERVISIÓN DEL LEVANTAMIENTO AÉREO. | 9 |
| 2.1.- Planeación del vuelo. | 9 |
| 2.1.1.- Supervisión del levantamiento. | 12 |
| 2.2.- Pantalla de inicio. | 13 |
| 2.2.1.- Configuración del programa. | 13 |
| 2.3.- Pantalla de trazo de líneas sobre la zona de vuelo. | 14 |
| 2.3.1.- Configuración de mapas. | 14 |
| 2.4.- Pantalla de modo de vuelo y almacenamiento de datos. | 16 |
| 2.4.1.- Monitoreo de levantamiento. | 16 |
| 2.5.- Interfaz de comunicación con el sistema de orientación. | 19 |
| 2.5.1.- Disparo de cámara. | 19 |
| 2.5.2.- Almacenamiento de datos de orientación. | 20 |
| | |
| 3.- SISTEMA DE DETECCIÓN DE ORIENTACIÓN Y CONTROL DE APUNTAMIENTO. | 22 |
| 3.1.- Orientación independiente de la aeronave. | 22 |
| 3.2.- Sistema de navegación inercial en dos ejes. | 23 |
| 3.2.1.- Componentes del sistema inercial en dos ejes desarrollado. | 25 |
| 3.2.2.- Esquema de determinación de orientación asistido por gravedad. | 26 |
| 3.2.3.- Filtrado complementario. | 26 |
| 3.3.- Control de apuntamiento de la plataforma. | 27 |
| 3.3.1.- Calibración inicial de la brújula electrónica. | 27 |
| 3.3.2.- Control on/off. | 27 |

| | |
|---|-----|
| 3.4.- Corte automático del control al dar vuelta para retomar la siguiente línea de vuelo | 29 |
| 4.- PASO DE PROTOTIPOS A EQUIPOS OPERATIVOS. | 32 |
| 4.1.- Diseño de tarjetas impresas o PCB. | 32 |
| 4.1.1.- Conceptos básicos de fabricación de PCB's con PROTEL SE99. | 33 |
| 4.2.- Intervalómetro. | 34 |
| 4.3.- Circuitos de electrónica de control y manejo de sensores. | 36 |
| 4.3.1.- Circuitos de potencia. | 37 |
| 4.3.2.- Fabricación de tarjetas impresas para los circuitos de manejo de sensores y de potencia | 39 |
| 4.4.- Fuentes de energía. | 43 |
| 4.4.1.- Fuente de voltaje regulado de 12 y 5 volts DC fijos. | 43 |
| 4.4.2.- Fuente de voltaje regulado, variable de 1.2 a 17 volts DC. | 45 |
| 4.5.-Conectores y conexiones. | 47 |
| 4.5.1.- Conexión computadora-circuitos. | 49 |
| 4.5.2.- Conexión circuitos-plataforma. | 49 |
| 4.6.-Contenedor de circuitos y sensores. | 50 |
| 5.- PRUEBAS DE FUNCIONAMIENTO Y RESULTADOS. | 52 |
| 5.1.- Asignación de valores y calibración inicial. | 54 |
| 5.2.- Control de la plataforma y almacenamiento de la información. | 55 |
| 5.3.- Cambios de línea de vuelo. | 59 |
| 5.4.- Programación. | 60 |
| 5.4.2.- Características del lenguaje C. | 60 |
| 5.5.- Algoritmo de procesamiento. | 61 |
| 5.6.- Resultados. | 64 |
| 6.- CONCLUSIONES Y RECOMENDACIONES. | 67 |
| 6.1.- Conclusiones. | 67 |
| 6.2.- Recomendaciones. | 67 |
| APÉNDICE A. Encoders. | 68 |
| APÉNDICE B. Programa en lenguaje C, para el control de la plataforma. | 77 |
| BIBLIOGRAFÍA | 115 |

PREFACIO

Se presenta el desarrollo de un sistema, cuyo objetivo es el de adquirir imágenes aéreas mediante cámaras digitales de formato reducido. Este consta de una plataforma portacámaras, cuyo apuntamiento es controlado mediante un lazo de realimentación, donde un sistema de navegación inercial determina la orientación de la plataforma de manera independiente de la aeronave y un conjunto de servomotores efectúan el control del apuntamiento. El otro elemento principal del sistema de adquisición de imágenes, lo constituye un programa de monitoreo que se encarga de efectuar la planeación y la supervisión de las variables más importantes durante el levantamiento, así como de enviar la señal de disparo a la cámara, misma que se aprovecha para almacenar los datos de la posición y del vector de apuntamiento de cada toma.

Para el desarrollo de este proyecto, fue necesario llevar a cabo un procedimiento de revisión de diseños, programas y prototipos existentes y el proponer y efectuar las adaptaciones pertinentes. Durante el paso de prototipos de laboratorio a sistemas funcionales, se diseñaron y construyeron tarjetas impresas, fuentes de voltaje y arneses para el cableado. También se tomaron en consideración elementos de tipo económico y de procesos de manufactura para una posible fabricación en serie de este sistema.

Se llevaron a cabo pruebas de funcionamiento de algunos módulos, como el caso de los sistemas de navegación inercial y de control de estabilización. Dado que la plataforma portacámaras no está por el momento disponible, ya que su diseño y construcción definitiva está siendo llevada a cabo recientemente, las pruebas se efectuaron con un prototipo de laboratorio.

Se presentan los resultados de los análisis de diseño, las adecuaciones, la programación, la circuitería desarrollada y las pruebas de funcionamiento efectuadas con este sistema de adquisición de imágenes, llevadas a cabo en tierra.

CAPÍTULO 1 INTRODUCCIÓN

En el Laboratorio de Sistemas de Información Geográfica y Percepción Remota del Instituto de Geografía de la UNAM, se desarrolla un sistema de adquisición de imágenes aéreas digitales, que es utilizado para la realización de levantamientos aéreos en diversas zonas del país. Se pretende obtener un sistema alternativo de percepción remota de bajo costo de adquisición y operación. Son necesidades primordiales para la captura adecuada de las imágenes, la utilización de un sistema de monitoreo que lleve a cabo la planeación y la supervisión del levantamiento, así como una plataforma estabilizada que compense las desviaciones que sufre la aeronave durante cada toma.

El sistema de monitoreo ha sido desarrollado y extensivamente probado durante más de 150 horas de vuelo, asimismo se ha diseñado y construido, un sistema inercial que permite determinar la orientación de la plataforma portacámaras, de manera independiente de la aeronave y un sistema de control para mantener la estabilización de la misma. Con éstos últimos se han realizado pruebas de funcionamiento en el laboratorio, mientras que la plataforma de vuelo definitiva se encuentra en proceso de diseño y construcción.

Para que el sistema de adquisición de imágenes quedara completo, actualizado y en funcionamiento, fue necesario efectuar cambios importantes en la programación y llevar a cabo la transición entre los prototipos de laboratorio y aquellos equipos que serán usados en vuelo. Estos fueron los objetivos primarios cubiertos en este proyecto, sin embargo queda pendiente la tarea de efectuar pruebas de funcionamiento con la plataforma que se encuentra en construcción, para finalizar con un sistema a nivel operativo.

1.1 Sistema de Adquisición de imágenes

Las desviaciones que sufre una aeronave durante las campañas de percepción remota, son en ocasiones tan grandes, que no es posible obtener imágenes útiles durante algunas tomas; esto provoca que se repitan algunas líneas de vuelo o transectos, con los consecuentes gastos de tiempo y dinero. Aunado a esto, existe una importante cantidad de variables que deben de ser monitoreadas de manera continua por el fotógrafo aéreo. Por esta razón surgió la idea de desarrollar un Sistema de Adquisición de Imágenes que se encargue de controlar totalmente el proceso de captura de imágenes, que se realiza con

cámaras digitales de pequeño formato. Con dicho sistema se ha obtenido un aumento significativo en la cantidad de imágenes útiles.

Más aún, haciendo posible asociar a cada imagen la información del lugar y tiempo preciso de la toma, así como el vector de apuntamiento con respecto a un marco de referencia conocido, las tareas de corrección geométrica de las imágenes y la formación de mosaicos serán reducidas de manera drástica. Estas son las características principales del sistema de adquisición de imágenes que aquí se presenta; el cual debe, en su implementación final, mantener una precisión en la orientación de sus tres ejes de $\pm 1^\circ$ por un período mínimo de tres horas. En la siguiente figura se muestra un diagrama de bloques del sistema completo de adquisición de imágenes.

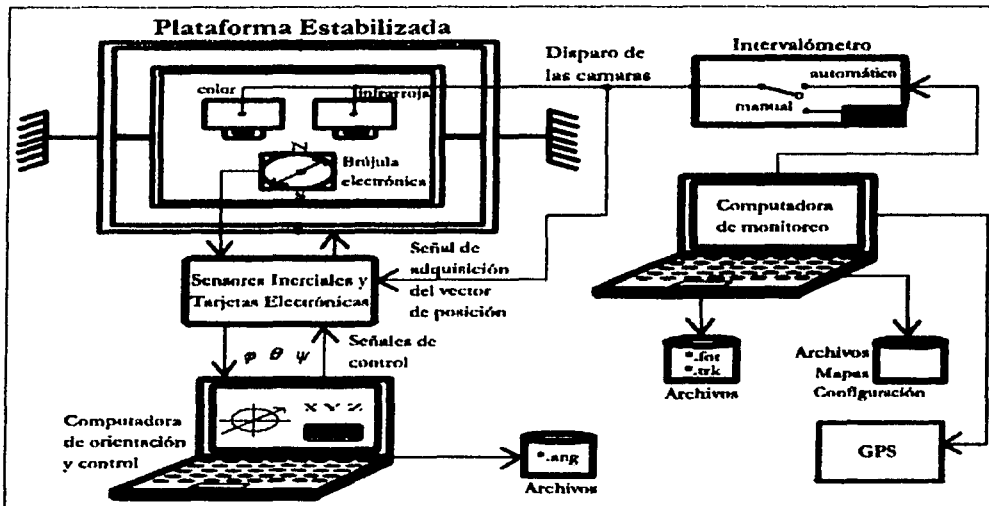


Figura 1.1 Diagrama de Bloques del sistema de adquisición de imágenes.

1.2 Sistema de Monitoreo

Este programa permite mantener bajo supervisión, los distintos parámetros que intervienen en el proceso de captura de imágenes. Uno de sus principales objetivos es el evitar errores humanos, lo cual se logra con una buena planeación de cada levantamiento, dando la posibilidad de que la misma computadora realice los cálculos tediosos para cada caso.

El contar con un sistema de monitoreo durante las campañas aerofotográficas, significa tener una herramienta indispensable para lograr un control global de todo el trabajo que se tiene que realizar. Proporciona seguridad a todo aquel que utiliza el sistema; ya que permite tener una visión de las variables más importantes durante el levantamiento como son: la velocidad de la aeronave, el rumbo, la altura sobre el terreno, etc. Nos ayuda a determinar nuestra posición geográfica exacta, en tiempo real, y es graficada sobre un mapa digital del terreno. Esto proporciona al piloto y al navegante, una visión real de cómo se va realizando

el trabajo, dando la facilidad de corregir la dirección del avión en el momento en que éste se localiza fuera de la línea de vuelo o que se considere que ha sido sobrepasado el intervalo de error que se puede tolerar. Durante la adquisición de imágenes a velocidades que algunas veces sobrepasan los 300 km/h y bajo ciertas condiciones, puede ser fácil que se pierda el rumbo, pero con la ayuda del sistema de monitoreo se disminuye la posibilidad de que se tengan que repetir líneas o partes de las mismas sin necesidad. Todo esto redundará en una mejor realización de la campaña y, por tanto, en una disminución enorme en gastos.

1.3 Plataformas aéreas para adquisición de imágenes

Las fotografías se han tomado desde el aire, casi en los comienzos del invento del proceso fotográfico. La primera foto aérea conocida se hizo en Francia en 1856, desde la cesta de un globo. En la década de 1880 se fotografiaba desde globos y a partir de 1909 desde aviones, tanto en Francia como en Estados Unidos. En 1862, en Richmond Virginia, el ejército federal estadounidense envió un fotógrafo en un globo, sujeto al suelo con una larga cuerda, hacia las alturas para que obtuviera fotografías de las posiciones enemigas.

En Gran Bretaña, J.T.C. Moore Brabazon, fotógrafo encargado del equipo de reconocimiento aéreo del ejército británico; experimentó la toma de imágenes primero con cámaras de fuelle, pero pronto se percató de que no tenía la posibilidad de mantenerlas fijas y exentas de vibraciones en la corriente de aire generada por el vuelo de un avión que avanzara a 120 Km/h. Por tanto, proyectó una cámara para ser instalada en el suelo del aeroplano: aquí aparece por primera vez la idea de crear una plataforma para una cámara fotográfica.

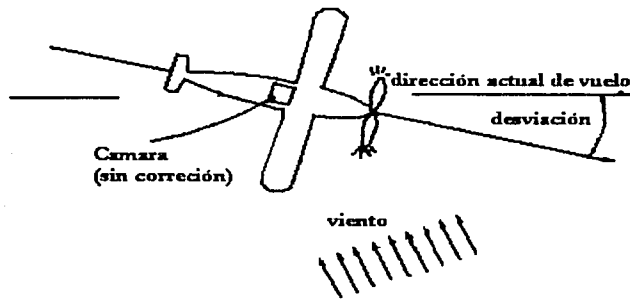


Figura 1.2. Cambio de dirección debido a vientos laterales.

Las plataformas aéreas para cámaras fotográficas son el mecanismo usado para unir la cámara fotográfica al avión. Tiene dispositivos de amortiguamiento que evitan que las vibraciones del avión sean transmitidas a la cámara fotográfica. También están diseñadas para que la cámara pueda rotar en el azimuth y así corregir la deriva (la diferencia de la orientación de la cámara en el avión con respecto a la actual dirección de vuelo, como se muestra en la figura 1.2.). Este fenómeno es causado por los vientos laterales.

La magnitud de la deriva depende de la velocidad y dirección del viento y tiene el efecto indeseable de reducir la cobertura de la imagen, como se nota en la siguiente figura.

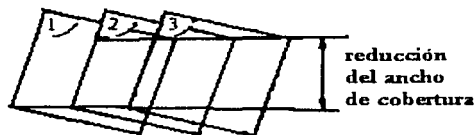


Figura 1.3. Reducción de la cobertura en el terreno, causada por la deriva.

La figura 1.4 muestra la cobertura en tierra cuando la cámara a sido rotada dentro de la plataforma montada en el avión, mejorando el paralelismo en las fotografías a pesar de la dirección actual del avión.

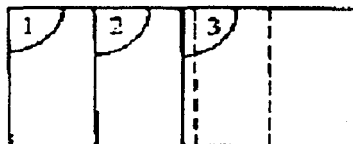


Figura 1.4. Cobertura mejorada debido a la corrección que hace la plataforma.

1.3.1 Plataformas aéreas convencionales

La fotografía aérea es común clasificarla en vertical y oblicua. La primera se toma con el eje focal de la cámara dirigido tan verticalmente como sea posible, de tal manera que el plano de la fotografía sea paralelo al de la Tierra. En la práctica es difícil mantener el eje de la cámara perfectamente vertical, debido a las inclinaciones inevitables del avión. Cuando el eje de la cámara se inclina ligeramente, la fotografía resultante es llamada inclinada. Las inclinaciones no intencionales usualmente son menores a un grado y raramente mayores a 3. Para muchas aplicaciones prácticas los procedimientos simples convenientes para analizar fotografías verdaderamente verticales se pueden utilizar para fotografías inclinadas

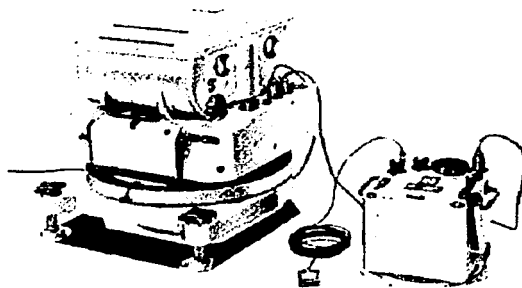


Figura 1.5. Cámara fotográfica, plataforma giro-estabilizada RMK-TOP con intervalómetro.

sin consecuencias serias. Para evitar en lo posible estas inclinaciones, que sufre la cámara fotográfica durante el recorrido del avión, se han desarrollado plataformas donde son montadas las cámaras y éstas a su vez van sujetas al avión. Un ejemplo de estas plataformas se ve en la figura 1.5, que son controladas mediante un servomecanismo que compensa los movimientos naturales de inclinación del avión.

Estos tipos de plataformas son sumamente complejas, la tecnología con la que son construidas es de punta, los resultados que se obtienen son excelentes, pero debido a esta y otras características tienen costos altísimos. Otro inconveniente es que son sumamente pesadas, generalmente arriba de 120 Kilogramos, y por tanto sus dimensiones son grandes.

1.3.2 Plataformas para cámaras de formato pequeño

En el Laboratorio de Percepción Remota Alternativa y Tecnología Avanzada del Instituto de Geografía de la UNAM, se han desarrollado algunas plataformas para adquisición de imágenes aéreas, para su utilización con cámaras de formato pequeño. Este tipo de plataformas, no son tan pesadas por lo tanto sus dimensiones son reducidas y el punto más

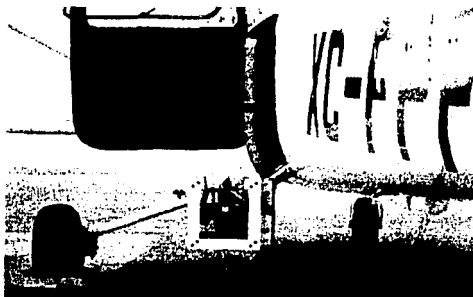


Figura 1.6. Plataforma no estabilizada para montura exterior en avionetas.

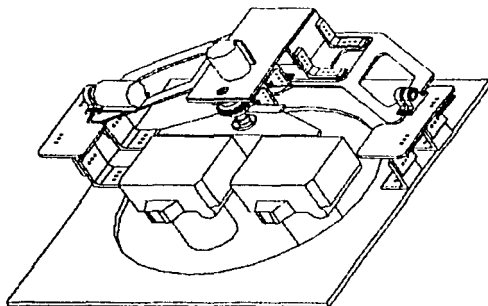


Figura 1.7. Plataforma aérea estabilizada, con capacidad de soportar dos cámaras de pequeño formato.

importante es que son de bajo costo. Algunas plataformas se pueden montar afuera de la aeronave y no cuentan con ningún sistema de estabilización (figura 1.6), mientras que se ha diseñado otra clase, que basa su método de control en el Sistema de Navegación Inercial por las ventajas que este proporciona. Estas plataformas utilizan un sistema de medición inercial compuesto de giróscopos y acelerómetros para determinar la orientación de la plataforma en todo momento. Estos sensores envían sus señales hacia un sistema de control, con lo cual es posible mantener constantemente estabilizada la plataforma. Además permiten almacenar las lecturas directas de la posición de la plataforma al momento de la toma, para un posterior procesamiento de corrección geométrica de las imágenes. Actualmente este tipo de plataformas cuentan con un sistema mecánico consistente en una suspensión estable de tres puntos y tres servo motores altamente dinámicos para compensar los movimientos angulares del avión (ver figura 1.7).

1.4 Sistema de detección de orientación y de control de estabilización

El sistema de detección de orientación y de control de estabilización utilizado en la plataforma, puede considerarse como un sistema de instrumentación el cual está basado en la realización de diferentes mediciones, en cálculos que involucran el procesamiento de datos y un sistema de despliegue de la información. Esta definición implica dos clases de operaciones principales: la primera, consiste en determinar la orientación de la plataforma con respecto a un sistema de referencia conocido (la horizontal local) y, la segunda, el control; que es la operación de modificar la orientación de la plataforma hasta que se alinee con la horizontal y el rumbo establecido.

1.4.1 Sistema de navegación inercial

La navegación inercial está basada en la mecanización de las leyes de Newton. Midiendo la aceleración de un cuerpo, es posible calcular su velocidad y su posición por medio de integraciones sucesivas. Para poder navegar en un marco de referencia es necesario saber en que dirección estamos midiendo la aceleración, para ello es necesario utilizar sensores giroscópicos que nos indiquen la dirección de los acelerómetros, entonces la *navegación inercial es el proceso en el cual, las mediciones realizadas por giróscopos y acelerómetros son utilizadas para determinar la posición y la orientación del vehículo*. A diferencia de otros tipos de sistemas de navegación, los sistemas inerciales son totalmente autónomos, en el sentido de que no necesitan ninguna señal externa para su funcionamiento, el siguiente diagrama de bloques ejemplifica un sistema de navegación inercial.

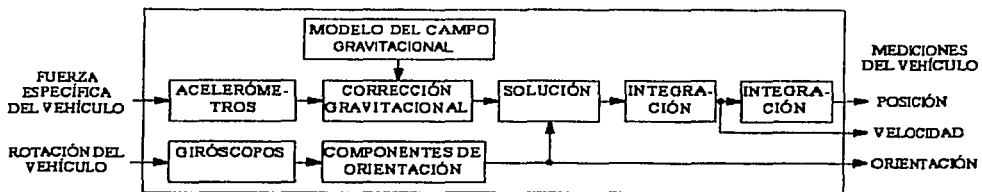


Figura 1.8. Diagrama de bloques simplificado de un sistema de navegación inercial.

Para lograr la navegación exitosa de un vehículo es necesario montar los sensores apropiados en el mismo. En general, se necesita determinar la posición del vehículo con respecto a un marco de referencia tridimensional, consecuentemente, se necesitarán tres giróscopos para proporcionar la medición de variación angular del vehículo alrededor de tres ejes distintos, al mismo tiempo que tres acelerómetros proporcionan las componentes de aceleración que el vehículo experimenta a lo largo de estos mismos ejes. Por conveniencia y precisión, los tres ejes usualmente son mutuamente perpendiculares y son coincidentes con los ejes del vehículo, usualmente llamado el juego de ejes fijos al cuerpo. Es muy importante recalcar la necesidad de multiplicar los valores de orientación obtenidos a partir de las mediciones de los sensores montados en el vehículo, por una matriz de rotación; para de esta manera obtener la orientación del vehículo con respecto a un marco de referencia inercial.

Resulta notable que el equipo de navegación inercial, debido a su inherente característica de independencia del vehículo, tiene la capacidad de medir la orientación del mismo y, de esta manera, de proveer las señales para el control de la orientación en tres ejes. En el caso particular del sistema implementado para controlar la orientación de la plataforma portacámaras, únicamente hacemos uso de las señales de orientación, la posición geográfica y la velocidad son obtenidas por medio de un receptor de GPS.

1.4.2 Brújula electrónica

Se emplea una brújula electrónica para determinar el movimiento de guiñada de la plataforma durante el vuelo. Este sensor utiliza un magnetómetro de tres ejes para determinar la orientación con respecto al vector magnético terrestre.

Se utilizó un módulo electrónico denominado EZ-COMPASS-3, cuyo diseño está basado en un microcontrolador programado por el fabricante. La comunicación con éste dispositivo se realiza en forma serie bajo el estándar RS-232. Al microcontrolador se accede mediante comandos suministrados por el operador, desde el teclado de la computadora o de forma automática programando previamente estos comandos en una rutina de comunicación serie, como en este caso. El módulo consta de un magnetómetro y dos inclinómetros.

1.4.3 Sistema de control de apuntamiento

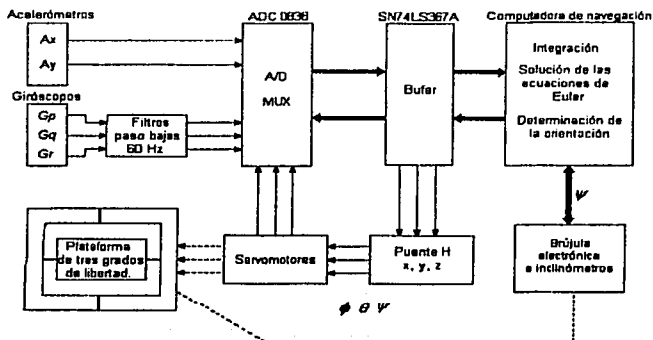


Figura 1.9 Esquema de detección de orientación y control de la plataforma.

Para llevar a cabo la detección de la orientación de la plataforma y de esta manera poder corregir su apuntamiento en tiempo real, fue necesario efectuar la implementación de un sistema como el mostrado en la figura 1.9, donde se muestra el esquema completo de orientación y control.

1.5 Pruebas de funcionamiento del sistema de adquisición de imágenes

Para determinar que el sistema sea viable, se llevaron a cabo varios tipos de pruebas. Primero se trató de determinar la precisión en las lecturas de orientación obtenidas durante un periodo determinado. Esto tuvo gran importancia ya que de ellas depende la precisión de la acción de control de la plataforma. Se espera tener al sistema en funcionamiento durante un tiempo aproximado de tres a cuatro horas en un levantamiento aéreo.

1.5.1 Pruebas de funcionamiento en tierra

Una vez que se comprobó el correcto funcionamiento por separado de las distintas etapas que conforman al sistema de orientación, estas se integraron entre sí para verificar su funcionamiento en conjunto y así poder hacer los ajustes necesarios cuando fueran requeridos, tanto en el algoritmo de control y determinación de la orientación, como en la circuitería, en los sensores y demás componentes del sistema. Para poder realizar estos ajustes, fue necesario establecer algunos parámetros como son: el máximo ángulo de corrección de orientación de la plataforma (sin que existan singularidades en las ecuaciones) y la velocidad de respuesta del sistema, en este caso, el tiempo que tarda la computadora en procesar la información proveniente de sensores hasta entregar los pulsos de control a los actuadores. Esto último involucra la comunicación con los puertos serial y paralelo y la velocidad de procesamiento, la conversión analógico-digital, así como del lenguaje de programación utilizado y las rutinas programadas.

Se establecieron diversos tipos de pruebas para cada parte; las cuales variaron desde la simple comprobación de los resultados arrojados, el tiempo que tarda la computadora en procesar la información proveniente del GPS y entregar los pulsos de control al disparador de la cámara, hasta la posibilidad de fallo del programa por factores conocidos y errores del usuario que podían dar como resultado una falla generalizada. También se realizaron pruebas de circuitería, ya que aún con las tarjetas impresas, los circuitos presentaron fallas por lo que fue necesario rediseñar y reacomodarlas para evitar ruido electromagnético inducido.

1.5.2 Pruebas de funcionamiento en vuelo

Con excepción del programa de monitoreo, los demás componentes no fueron probados en vuelo. Esto se hará cuando la plataforma portacámaras definitiva se encuentre disponible.

CAPÍTULO 2

PROGRAMA DE MONITOREO PARA PLANEACIÓN Y SUPERVISIÓN DEL LEVANTAMIENTO AÉREO

La manera de llevar a cabo el monitoreo del levantamiento y de presentar los resultados, depende en gran medida de los procedimientos establecidos para llevar a cabo las misiones aéreas, por esta razón, se determinó la necesidad de utilizar tres rutinas principales. Así en la primera, se deben realizar las operaciones referentes a la planeación del vuelo. En la segunda es necesario desplegar la zona de estudio ubicada en un mapa digital y mostrar las líneas de vuelo. En la tercera se requiere supervisar las variables críticas del sistema y almacenar los datos, tanto de recorrido de la aeronave, como de posición geográfica de cada toma fotográfica.

Con base en lo anterior, el programa de monitoreo fue dividido en tres partes, asignando a cada rutina una pantalla de presentación, las cuales se denominaron: Pantalla de configuración y estimaciones ("CONFIG"), Pantalla de Mapas ("MAPAS") y Pantalla principal ("PANT1").

Se estableció como criterio de funcionalidad que la interfaz entre la computadora y el usuario fuera lo más comprensible y sencilla posible, utilizando herramientas gráficas que facilitaran el funcionamiento del programa, y que éste pueda ser manejado por cualquier usuario involucrado en el tema de levantamientos aéreos.

Comenzaremos haciendo una descripción de todas las variables involucradas en la planeación del vuelo, por lo que en la siguiente sección mostramos este procedimiento y continuaremos con el manejo de cada una de las pantallas.

2.1. Planeación del vuelo

La toma física de la fotografía aérea, (figura 2.1.) sigue un proceso que consiste en cubrir totalmente el área de estudio con líneas o transectos de vuelo. Estas líneas tienen la finalidad de obtener secuencialmente todas las fotografías necesarias, sobreponiéndose una sobre el área de la anterior longitudinal (60%) y lateralmente (30%). Esto es muy útil cuando

se realizan estudios geográficos y topográficos para la obtención de mapas precisos de una cierta zona.

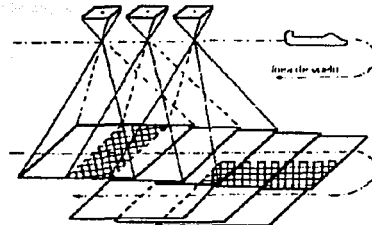


Figura 2.1. Esquema de la toma de la fotografía aérea.

Las variables a considerar durante un levantamiento aéreo son las siguientes: velocidad de la aeronave, sobreposición longitudinal y lateral, resolución de pixel, resolución de cámara (píxeles por línea), altura de vuelo, cobertura y por último el intervalo de disparo de las cámaras.

➤ **Variables proporcionadas por el usuario**

La velocidad de vuelo, las sobreposiciones longitudinal y lateral, la resolución de pixel, la resolución de la cámara y el número de fotos por tarjeta (cuando se usa este medio de

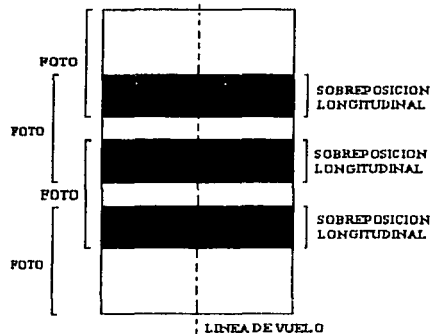


Figura 2.2. Sobreposición longitudinal en las fotografías.

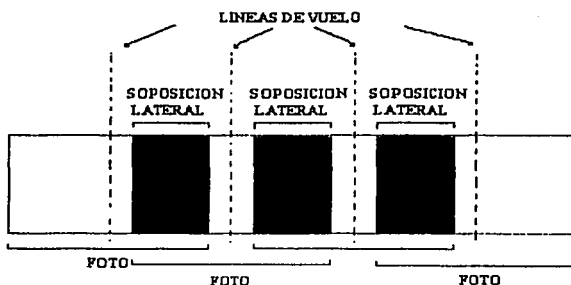
almacenamiento, ya que existe también la posibilidad de enviar las imágenes directamente al disco duro de la PC, cuando la cámara cuenta con un puerto de salida).

- v = Velocidad de vuelo: depende de las velocidades mínima y máxima de la nave, de su techo de operación, pero sobre todo, de la rapidez con la cual la cámara pueda

guardar las imágenes; ya que muchas veces ésta última es la limitante para efectuar el levantamiento de una forma más rápida.

Las sobreposiciones longitudinal (Slong), lateral (Slat) se miden en porcentaje. Con ellas se delimita el traslape entre todas y cada una de las fotografías.

- Slong = sobreposición longitudinal: Es el porcentaje de la imagen que deseamos se traslape con la imagen actual y la siguiente. 60% es el valor mas común, con ello se obtienen pares estereoscópicos, que permiten la identificación de objetos en 3D sobre el terreno.
- Slat = sobreposición lateral: Delimita el traslape entre imágenes de líneas consecutivas. Debe tener un mínimo del 30% para asegurar que no haya huecos en el mosaico.



TESIS CON
FALTA DE ORIGEN

Figura 2.3. Sobreposición lateral.

- Rt = resolución de píxel: Es el detalle necesario en la fotografía de acuerdo a las necesidades del estudio a realizar. Esto nos indica el tamaño de los objetos en el terreno que podemos identificar. En nuestro caso, los valores más utilizados se encuentran entre 0.8m a 1.5m.
- RESX = resolución de cámara: Se refiere al número de pixeles útiles de imagen que tiene cada una de las líneas del CCD.
- NF = número de fotos: Es la cantidad de imágenes posibles de almacenar en disco. Depende de la calidad de imagen que se seleccione. Tiene como objetivo dar un aviso al usuario sobre el número de imágenes y el momento oportuno para cambiar el dispositivo de almacenamiento, antes de llegar a su capacidad máxima.

Variables Calculadas

Son el resultado del cálculo de varias fórmulas, a partir de los datos proporcionados. Es parte de lo que el programa automatiza para facilitar la planeación del vuelo

- H_i = Altura de vuelo: De ella dependerá cual será la cobertura y la resolución, cuanto más alto se vuela, se cubrirá mayor área de terreno; pero a su vez se pierde detalle. La cobertura está también ligada a la distancia focal del lente utilizado. Se realizó un

experimento colocando la cámara a 1 m de distancia de una superficie plana con marcas bien definidas. Se ha tomado una fotografía, se ha medido la superficie que se ha captado y se observa que para el caso específico del lente que se utiliza, la cobertura de la cámara es de 86 cm por 56 cm; esto nos indica que para una altura cualquiera, se abarca el 86% del terreno a lo largo y el 56% a lo ancho. Basados en esta cobertura tenemos:

RESX = resolución de cámara

Rt = resolución de pixel

Hi = $(Rt * RESX) / 0.86$ = altura de vuelo

- Dlon = Distancia longitudinal. Indica la distancia que se debe tener entre cada imagen. Se calcula a partir de la sobreposición longitudinal (Slon), junto con la altura de vuelo. El factor de 56% del ancho de cobertura del lente, es utilizado para obtener esta distancia entre fotografías sobre la misma línea de vuelo.

$Dlon = (Hi * 0.56) * (1 - (slon / 100))$

- Dlat = Distancia Lateral. Nos indica cual es la separación entre líneas de vuelo. Para su cálculo intervienen Slat y las resoluciones de pixel y de la cámara.

$Dlat = (Rt * RESX) * (1 - (Slat / 100))$

- Ti = Tiempo de disparo. Este intervalo depende de la distancia entre cada fotografía (Dlon) y de la velocidad del avión, obtenida del GPS.

$Ti = (Dlon / v)$; en donde v es la velocidad del avión.

2.1.1. Supervisión del levantamiento

Una de las mayores dificultades que se presenta durante la navegación aérea, es el poder contar con datos precisos de la dirección de la nave, su velocidad, su altura etc., pero sobre todo, la localización de donde se encuentra en todo momento, siendo esta una necesidad primaria en la realización del sistema de monitoreo. Esto se logra con la utilización de dos elementos como son el sistema GPS y la computadora. El GPS envía datos precisos cada segundo y la computadora los recibe, ordena y despliega, dando además la facilidad de trabajar con la función: mapa móvil.

➤ ¿Qué es la función mapa móvil?

Se trabaja en el modo MAPA MÓVIL, cuando se observa en la pantalla de la computadora un mapa y sobre él, en tiempo real, la figura de cualquier móvil que se desplaza, indicándonos la posición actual en todo momento. Además de saber por donde se transita, se puede ver un registro de donde ya se ha pasado (track).

Mapa digital del terreno

Consiste de una imagen de satélite, un mapa o cualquier otro elemento desplegado en la pantalla principal del programa y que sirve de ayuda visual en la navegación. En él se ubican rápidamente las líneas de vuelo y la posición actual de la aeronave.

➤ **Actualización de la posición sobre el terreno con el GPS**

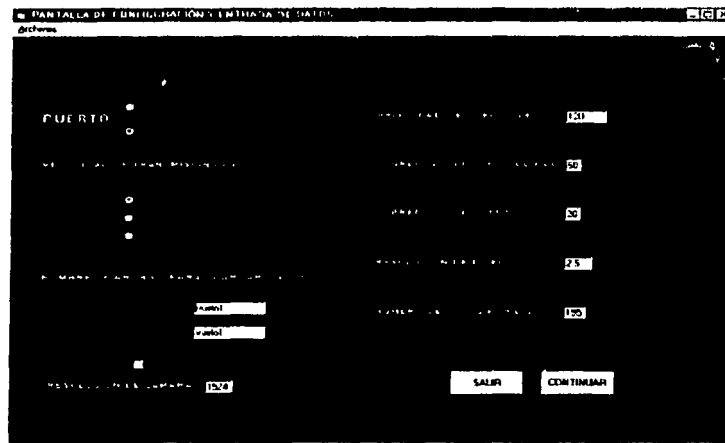
En el mapa Móvil se muestra la posición de la aeronave en tiempo real llevando a cabo una graficación georeferenciada sobre el mapa, de los diferentes movimientos seguidos por la

aeronave, a lo que llamamos: Actualización de Posición Sobre el Terreno; ésta es la parte fundamental de un Mapa Móvil. Con la ayuda del GPS se logra que la velocidad, la altura, la posición y la dirección de vuelo sean actualizados a intervalos de 1 segundo.

2.2. Pantalla de inicio

En esta pantalla se lleva a cabo una configuración del programa, por lo que se deben asignar valores iniciales, también se debe especificar la forma de comunicación con el dispositivo GPS a través de los puertos y se deben proporcionar los nombres de los archivos que serán generados durante el vuelo.

2.2.1 Configuración del Programa



TESIS CON
FALLA DE ORIGEN

Figura 2.4. Pantalla "Config" (ejemplo de datos del proyecto de vuelo de Texcoco).

En la pantalla "Config" se presentan las opciones para elegir el puerto de comunicación y la velocidad de transmisión, un campo para introducir el nombre del archivo de recorrido (*.TRK) y otro para el nombre de archivos de datos de posición de cada fotografía (*.FOT), una opción de no guardar datos de levantamiento, un campo para resolución de cámara, un campo para velocidad, otro para sobreposición longitudinal, otro para sobreposición lateral, para resolución de pixel, un campo más para anotar el número máximo de fotografías por disco, los botones "salir" y "Continuar" y por último, en la parte superior izquierda, un menú "desplegable".

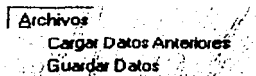


Figura 2.5 Menú disponible en la pantalla Config.

También la pantalla Config contiene un menú desplegable, y dos botones. El menú contiene las siguientes opciones: Cargar datos anteriores y Guardar datos.

Cargar datos anteriores: Carga en la pantalla "Config", archivos con extensión "udo" los cuales contienen datos de configuraciones de algún levantamiento en particular o de alguno realizado con anterioridad, son archivos con formato generado por el propio programa, los cuales son de fácil lectura en cualquier editor de texto. En dicha pantalla, aparece seleccionado el directorio C:\sadiutm\datos, donde se encuentran los posibles archivos con extensión "udo".

Botón Continuar: Al hacer click sobre este botón, se confirma que cada campo de la pantalla Config se hayan llenado correctamente, además se realizan los cálculos de la fase de planeación (dlon, dlat, etc.), para posteriormente desplegar la pantalla de Mapas. En caso de encontrar algún error se muestra un mensaje de error.

Botón Salir: Cierra el programa y regresa a Windows sin guardar ningún dato.

2.3 Pantalla de trazo de líneas sobre la zona de vuelo

En esta pantalla se ejecutan las rutinas que tienen como objetivo cargar un mapa digital desde cualquier directorio, desplegarlo en pantalla, obtener los datos necesarios para georeferenciar el mapa digital, así como también mostrar la zona de vuelo sobre éste, generando y visualizando las líneas necesarias para cubrir la zona, además de mostrar el número de líneas, número de fotografías por línea y visualizar parte de los resultados obtenidos por la rutina de inicio como son: dlon, dlat, tiempo de disparo y altura de vuelo.

2.3.1 Configuración de mapas

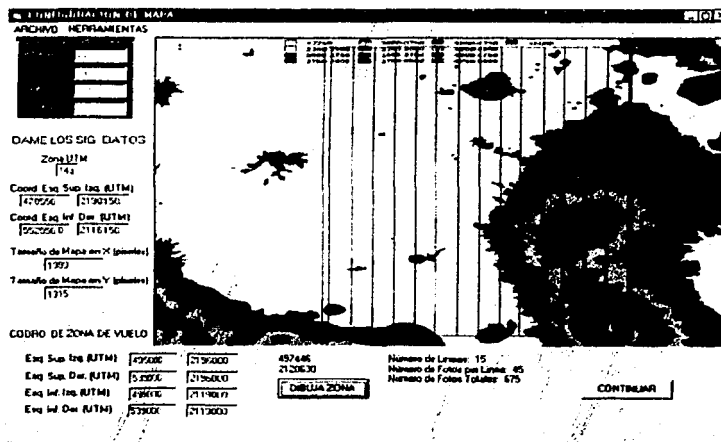


Figura 2.6. Pantalla Mapas, (ejemplo de datos de vuelo de Texcoco, Méx.).

Capítulo 2 Programa de monitoreo para planeación y supervisión del levantamiento aéreo

La rutina de despliegue y configuración de mapa de vuelo se lleva a cabo en la pantalla de "Mapas", que tiene un entorno con una serie de campos, un menú en la parte superior, dos botones, un espacio para visualizar el mapa de navegación donde se observa la zona de estudio y las líneas de vuelo. Cada campo lleva un título referido al dato que se va a teclear, como se muestra en la figura anterior.

Zona UTM: este dato indica a que zona UTM corresponde el área mostrada en el mapa digital y se obtiene directamente del mapa.

Coord. Esq. Sup. Izq: Se refiere a la Coordenada UTM (X, Y) de la Esquina superior izquierda del mapa.

Coord. Esq. Inf. Der: Se refiere a la Coordenada UTM (X, Y) de la Esquina inferior derecha del mapa.

Tamaño de mapa en X, Y (píxeles): en estos campos se introduce el tamaño en píxeles del mapa, que corresponden al ancho y largo del mapa digital. Cabe mencionar que es necesario que este tamaño sea de un máximo de 2000 por 2000 píxeles, ya que es una de las restricciones del programa para su correcto funcionamiento.

Coordenadas de zona de vuelo: Se teclean las esquinas de la zona del levantamiento aéreo. con estos datos queda especificada la zona de vuelo sobre el mapa digital.

Botón Dibujar Zona: Realiza las operaciones necesarias para determinar: la cantidad de líneas de vuelo para cubrir la zona en estudio, las coordenadas de los extremos de cada línea, el número de fotografías por línea y el número de fotografías totales, a partir de los datos de configuración de mapa y de los datos calculados en la pantalla "Config".

Botón de continuar: Botón localizado en la parte inferior derecha de la pantalla (figura 2.6), este botón al ser activado se encarga de revisar que todos los campos de configuración se hayan completado debidamente, si es así, carga la siguiente pantalla (pantalla principal) en caso contrario, manda un mensaje de error.

➤ Menú Desplegable sobre esta misma pantalla

Este menú contiene los encabezados de: Archivo y Herramientas, en Archivo se encuentran las opciones de: Abrir mapa, Guardar datos, Datos previos y Salir. Herramientas contiene la opción de: Crear el archivo de líneas (figura 2.7).



Figura 2.7 Menú de la pantalla de Mapas.

TESIS CON
FALLA DE ORIGEN

Abrir mapa: Esta opción ejecuta una pantalla auxiliar, cuyo objetivo es la elección y carga del mapa digital. Éste debe estar en formato bmp, el cual se propuso utilizar debido a que es

muy empleado por los usuarios finales, por lo tanto la pantalla auxiliar "Cargar Mapa" solo muestra archivos con extensión ".bmp".

Crear Archivo de líneas: Esta opción guarda las coordenadas extremas de cada línea de vuelo en un archivo con extensión ".lin", una vez que se capturaron dichas líneas, por medio del botón "Dibujar zona". El archivo generado es como el que se muestra a continuación:

```
sadiutm
Archivo:texcoco.lin
Coordenadas UTM de puntos extremos de líneas de vuelo
zonal4q
lineas  EASTING  NORTHING  EASTING  NORTHING
lin 1    495960.12  2188500  495960.12  2119000
lin 2    499160.52  2188500  499160.52  2119000
lin 3    502360.92  2188500  502360.92  2119000
lin 4    505561.32  2188500  505561.32  2119000
lin 5    508761.72  2188500  508761.72  2119000
lin 6    511962.12  2188500  511962.12  2119000
lin 7    515162.52  2188500  515162.52  2119000
lin 8    518362.92  2188500  518362.92  2119000
lin 9    521563.32  2188500  521563.32  2119000
lin 10   524763.72  2188500  524763.72  2119000
```

Figura 2.8. Parte del listado de líneas de vuelo.

Guardar datos: Al hacer click sobre la opción Guardar Datos, se verifica que todos los campos estén correctamente llenos y muestra la pantalla auxiliar "Salvar" (figura 2.4). En la pantalla Salvar se genera un archivo en donde se guardan los datos de Configuración de mapa con extensión ".sig".

Datos previos: Esta opción carga en la pantalla de "Mapas", un archivo con extensión ".sig" que contiene configuraciones que fueron guardadas con anterioridad, utilizando la pantalla auxiliar "Cargar Datos".

Salir: Esta opción cierra el programa y regresa a Windows

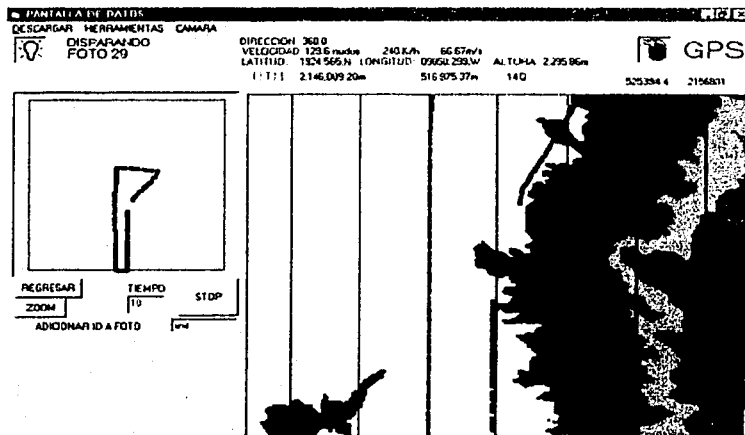
2.4 Pantalla de modo de vuelo y almacenamiento de datos

Las rutinas que ejecuta esta pantalla se encargan de realizar el despliegue y visualización tanto del mapa móvil como de las variables más críticas en el proceso de captura de imágenes, almacenado los datos de posición de cada fotografía, así como del control de la señal de disparo enviada a la cámara digital.

2.4.1 Monitoreo del levantamiento

En la pantalla "Pant1" es donde se ejecuta la rutina de vuelo, que permite monitorear el levantamiento propiamente dicho. Tiene un entorno formado por un campo donde se teclea un identificador para referencia del lugar donde se realiza una toma en particular, un recuadro para la visualización general del recorrido sobre las líneas de vuelo, la

visualización del área con las líneas de vuelo sobre el mapa digital (mapa móvil), un indicador de toma de fotografía y otro de GPS activo. También se cuenta con etiquetas con



TESIS CON
FALLA DE ORIGEN

Figura 2.9 Pantalla "Pant1", donde se muestra una simulación de vuelo sobre Texcoco, México.

los títulos de dirección, velocidad, longitud, latitud y altura, una zona destinada para las coordenadas en UTM de la posición de la aeronave, botones de zoom, regresar y disparar, un mensaje del conteo de fotografías y en la parte izquierda superior un menú desplegable. A continuación se explica cada una de estas opciones:

Campo identificador de fotografía: En este campo se teclea una clave o nombre que permita identificar la fotografía en curso, por ejemplo; Laguna verde, Pico de Orizaba, Estadio Azteca, etc. Todos ellos sirven para localizar fácilmente una fotografía.

Recuadro para visualización del recorrido: Este recuadro ocupa un espacio de alrededor del 15% de la pantalla, y en el se visualiza en forma global el levantamiento con la trayectoria ya recorrida y las líneas o transectos faltantes.

Botón Zoom: Proporciona una visión en mayor detalle de la trayectoria seguida por la aeronave, para un vuelo más fino sobre las líneas planeadas.

Botón Regresar: Después de observar en forma detallada la trayectoria, este botón nos regresa una visión completa del levantamiento (es decir, quita el zoom).

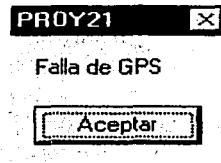
Cuando la aeronave se aleja más de dos kilómetros de la zona del levantamiento, los botones de Zoom y Regresar desaparecen de la pantalla y en su lugar aparece un mensaje de: Fuera de Zona

Recuadro de Mapa Móvil (visualización del área de vuelo): Abarca el 50% de la pantalla y nos sirve para observar, a lo largo del levantamiento, la trayectoria de la aeronave sobre las

líneas de vuelo y sobre el mapa de navegación; esto nos da una referencia del lugar sobre el cual se encuentra la aeronave, proporcionando una ayuda de navegación.

Indicador de fotografía: Está representado con la figura de un foco, localizado en la parte superior izquierda de la pantalla, el cual se observa encendido en el momento de ser enviada la señal de disparo, y apagado cuando ésta se interrumpe.

Indicador de Gps: Está representado gráficamente por un pequeño cilindro, y al momento de hacer un click sobre él, se inician las rutinas que abren el puerto de entrada de datos procedentes del Gps. Como señal de que el puerto está abierto, el cilindro se incendia. Al hacer un segundo click sobre el cilindro, el puerto de comunicación se cierra, mostrando el cilindro apagado. Cuando no se reciben datos aparece el siguiente mensaje (figura 2.10):



TESIS CON
FALLA DE ORIGEN

Figura 2.10 Indicación de falla del GPS.

Las causas de la aparición de este mensaje (figura 2.10) pueden ser: que el GPS esté desconectado, apagado, que se haya seleccionado mal el puerto de comunicación, o la mala selección del protocolo de comunicación (NMEA).

Etiquetas de dirección, velocidad, longitud, latitud y altura:

Cuando se comienza a recibir la información procedente del GPS, ésta es seleccionada, procesada y enviada a la pantalla y a otras rutinas que realizan diversos cálculos de transformación, como son: la velocidad, latitud y longitud.

La Barra de Menú superior contiene las opciones de Descargar, Herramientas y Cámara como se muestra en la siguiente figura.

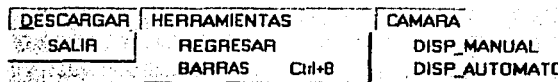


Figura 2.11 Barra de Menú superior.

Descargar: Contiene la opción de salir la cual cierra el programa y regresa a Windows

Menú Herramientas: Contiene las opciones de Regresar y Barras.

Regresar: Nos envía otra vez a la configuración de Mapa, esto da la posibilidad de cambiar a una zona de estudio distinta o simplemente cambiar sus coordenadas y ampliar o reducir la zona sin salir del programa.

Barras: Esta opción coloca barras de desplazamiento en el mapa digital tanto en X como en Y, con la finalidad de que se pueda explorar todo el mapa, cancelando el movimiento del mapa y desplazándolo conforme se necesite.

Cámara: Contiene la opción de Disparo Manual y Disparo Automático

Disparo Manual: Al elegir esta opción aparece un campo con el título de Tiempo, donde se teclea el intervalo de tiempo de disparo deseado y también aparece un botón que tiene por título "Disparo". Una vez que se ha tecleado el dato de tiempo, se da un click sobre este botón para iniciar el disparo de la cámara y cambia el nombre del botón a "Parar" para interrumpir la señal de disparo.

En realidad este es un disparo semiautomático, debido a que el disparo manual es el que se efectúa sin la ayuda de la computadora.

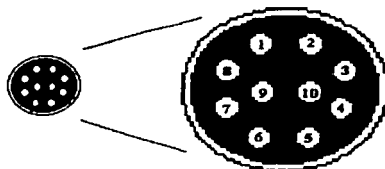
Disparo Automático: cancela la opción de disparo manual y en su lugar hace visible un mensaje donde se expresa el tiempo de disparo, calculado a partir de la velocidad en tiempo real que se toma del GPS y del dato d_{lon} (distancia longitudinal).

El primer disparo que permite el programa es en modo manual, debido a que se debe dar un valor inicial de tiempo de disparo por el usuario para después pasar al modo automático. Aun cambiando la forma de disparo, el botón debe estar con el título de "Parar", esto nos indica que se sigue enviando la señal de disparo. Una vez que se inicia el primer disparo en Manual, se puede cambiar a Automático y de Automático a manual y así sucesivamente.

2.5 Interfaz de comunicación con el sistema de orientación

La computadora controla el instante en el que debe ser tomada cada fotografía, esto se logra enviando un pulso a través del puerto paralelo, que se conecta a un dispositivo electrónico que utiliza relevadores para realizar el disparo de la cámara. Este mismo pulso es captado por la computadora que controla la plataforma y en el instante en que lo recibe son tomados los datos de: alabeo, cabeceo y guiñada como 3 variables x , y , z . Con estos datos recuperados en el preciso instante de la toma, es posible trabajar posteriormente con cada imagen, corrigiéndola geoméricamente y formando los mosaicos de la zona de estudio.

2.5.1 Disparo de cámara



ENTRADA DEL CONTROL REMOTO EN LA CAMARA.

TESIS CON
FALLA DE ORIGEN

Figura 2.12 Diagrama de la entrada del control remoto de la cámara.

La forma más adecuada de realizar los disparos controlados por la computadora, fue a través de un circuito desarrollado específicamente para esta aplicación. Éste realiza la misma tarea que el control remoto empleado en las cámaras. Después de leer los manuales del fabricante y de realizar algunas pruebas, se observó que para el disparo de la cámara solo es necesario poner en corto dos de sus entradas (ver figura 2.12), se debe hacer contacto entre los pines 1 y 10. Para lograr esto se utilizó un relevador, que proporciona la obturación y que puede ser controlado con un pulso enviado por la computadora.

Fue necesario considerar el poder realizar los disparos a intervalos de tiempo fijos y precisos, de manera manual, con la ayuda de un circuito, esto se logró con el diseño y construcción de un intervalómetro, teniéndose la posibilidad de fijar el envío de un pulso de 5 volts en un intervalo de entre 1 a 35 segundos.

2.5.2 Almacenamiento de datos de orientación

Una vez que se carga en memoria la pantalla Principal "Pant1" se ejecutan las rutinas que tiene por objeto generar los archivos de recorrido ".trk" y de fotografías ".fot", asignándoles un encabezado y los nombres elegidos en la pantalla Config. Cuando se abre el puerto de comunicación con el GPS y se tienen los primeros datos, el programa va agregando éstos al archivo de recorrido ".trk" durante el levantamiento, y una vez iniciado el primer disparo, se agregan los datos al archivo de fotografía.

El archivo de recorrido, lleva en primer lugar un encabezado que es compatible con el programa PCX5 ver 2.09, en el que se puede visualizar la trayectoria del GPS. Este archivo lleva los siguientes datos: identificador "T" (determina que se está generando un archivo de recorrido), Zona UTM, datos de coordenadas (en UTM), fecha y hora. Todo esto en intervalos de un segundo durante todo el recorrido.

```
H SOFTWARE NAME & VERSION
I PCX5 2.09

H R DATUM  IDX  DA      DF      DX
DY      DZ
M G WGS 84 121 +0.000000e+00 +0.000000e+00
+0.000000e+00 +0.000000e+00 +0.000000e+00

H COORDINATE SYSTEM
U UTM UPS

H ZONE  EASTING  NORTHING  DATE      TIME      ALT
;track
T 11R  570724   3495060  31-10-00  09:12:50
T 11R  570864   3494863  31-10-00  09:12:51
T 11R  570890   3494828  31-10-00  09:12:52
T 11R  570048   3494747  31-10-00  09:12:53
```

TESIS CON
FALLA DE ORIGEN

Figura. 2.13 Parte del listado del archivo de recorrido sobre la Sierra de San Pedro Mártir, BCN, México.

Archivo de Fotografía: Lleva un encabezado que es compatible con el programa PCX5 ver 2.09 de GARMIN, en el que se pueden visualizar los puntos de identificación (Waypoints) que en este caso corresponden a las coordenadas donde se tomó cada fotografía. Este archivo lleva los datos de identificador "W" (determina que se está generando un archivo de puntos de identificación), un identificador de fotografía, las coordenadas de cada imagen en

UTM, la fecha (mes-día-año), la hora (hh:mm:ss) y un campo de descripción donde es colocado el dato del identificador de fotografía.

| H SOFTWARE NAME & VERSION | | | | | | |
|---------------------------|---------|---------|----------|----------|---------------|---------------|
| I PCX5 2.09 | | | | | | |
| H | R | DATUM | | IDX | DA | DF |
| DX | | DY | | DZ | | |
| M | G | WGS 84 | | 121 | +0.000000e+00 | +0.000000e+00 |
| | | | | | +0.000000e+00 | +0.000000e+00 |
| H COORDINATE SYSTEM | | | | | | |
| U UTMA UPS | | | | | | |
| H | IDNT | ZONE | EASTING | NORTHING | DATE | TIME ALT |
| DESCRIPTION | | | | | | |
| | 654375 | 3436325 | 31-10-00 | 09:40:05 | | W 00001 11R |
| | W 00002 | 11R | 654378 | 3435841 | 31-10-00 | 09:40:12 |
| | lin1 | | | | | |
| | W 00003 | 11R | 654381 | 3435309 | 31-10-00 | 09:40:19 |

Figura 2.14. Parte del listado de fotografías obtenidas en la Sierra de San Pedro Mártir, BCN, México.

TESIS CON
FALLA DE ORIGEN

CAPÍTULO 3

SISTEMAS DE DETECCIÓN DE ORIENTACIÓN Y CONTROL DE APUNTAMIENTO

La navegación inercial depende de la integración de la aceleración con respecto a un marco de referencia Newtoniano. Los sistemas convencionales mantienen físicamente esta referencia por medio de anillos basculantes; esto es lo que conocemos como una plataforma estabilizada, donde los sensores que determinan su posición y orientación en todo momento (giróscopos y acelerómetros) son montados en ella, de tal manera que permanecen aislados mecánicamente de las rotaciones que sufre la nave. Muchos sistemas de este tipo han sido desarrollados y aún se siguen usando en barcos y submarinos que necesitan una precisión en la posición de 1 milla/hora durante varias horas o inclusive días. La versión analítica de esta idea ha hecho a un lado la complejidad mecánica de estas plataformas, sujetando firmemente los sensores al cuerpo de la nave y donde la transformación de la orientación a un marco de referencia inercial, es calculado en lugar de ser mecanizado.

3.1 Orientación independiente de la aeronave

Conforme pasa el tiempo cualquier navegador inercial sufre el problema de deriva, esto significa que el alineamiento inicial con respecto al vehículo se va perdiendo. Para corregir esta situación, los sistemas de navegación pueden ser asistidos desde el exterior para obtener una actualización de su posición. Para este propósito existen varios sistemas de radio navegación que proveen información sobre la ubicación de la nave. Como ejemplo, los sistemas Loran-C y Omega tienen una precisión entre 500 y 4,500 m y no proporcionan información sobre la velocidad. Los sistemas de ayuda a la navegación como el GPS o GLONASS basados en constelaciones de satélites, tienen una precisión nominal en la posición de 15 m, proporcionan la velocidad con que se mueve la nave y una base de tiempo; esto los hace muy útiles como sistemas de ayuda, porque esto permite también efectuar una corrección en el reloj de la computadora de abordo. Los sistemas de navegación inercial proporcionan información precisa de la orientación y la distancia recorrida en un tiempo corto, pero giróscopos y acelerómetros van acumulando errores con el tiempo y los errores en la

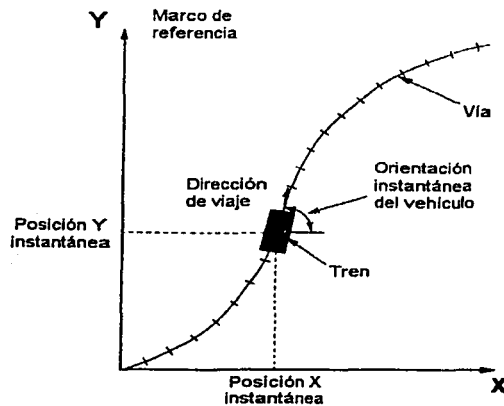
navegación crecen. El GPS no varía su precisión con el tiempo, y esta es una de sus mayores virtudes, pero no puede decirnos si estamos dando vuelta o haciendo un alabeo; de esta manera ambos sistemas se complementan.

Los métodos que dependen de la recepción de información desde una fuente externa al vehículo, ya sea desde tierra o del espacio exterior. Todos, están sujetos a errores o inoperatividad; cuando dicha información es errónea o se carece de ella, debido a causas naturales o inducidas. En este sentido, la *navegación inercial* difiere de manera fundamental de los métodos de navegación asistidos, aunque mantiene los elementos básicos de determinación de dirección, velocidad y posición. Ésta depende de mediciones realizadas enteramente en el interior del vehículo, de acuerdo con las leyes Newtonianas del movimiento y de la gravitación.

Existen varias razones prácticas para que la guía inercial haya sido desarrollada como sustituto o complemento de otros métodos. La característica de autosuficiencia, por ejemplo, resulta crucial para propósitos militares. Todas las razones para la existencia de este método único, finalmente están basadas en el hecho de que: las leyes del movimiento son simples, exactas, completas y universales.

3.2 Sistema de navegación inercial en dos ejes

Un caso simple de navegación en una dimensión involucra la determinación de la posición de un tren, el cual se mueve a lo largo de una vía recta, en un plano. Es posible calcular la velocidad instantánea del tren y la distancia que éste ha recorrido desde el punto de partida, por medio de mediciones de su aceleración a lo largo de la vía. Si un acelerómetro se encuentra fijo en el tren, este dispositivo nos proporcionará la información concerniente a la aceleración del tren. La integral en el tiempo de las mediciones de la aceleración proporciona una estimación continua de la velocidad instantánea del tren, previendo que se conoce su velocidad inicial. Una segunda integración nos da la distancia que ha viajado el tren, con respecto a un punto de inicio conocido. El acelerómetro junto con una computadora, u otro dispositivo capaz de realizar la integración, constituyen un sistema simple de navegación en una dimensión.



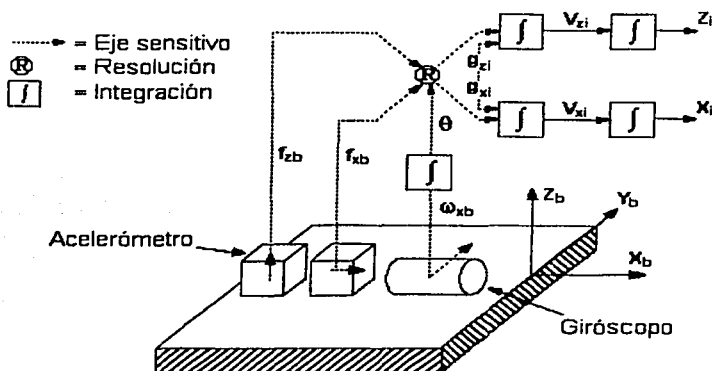
TESIS CON
FALLA DE ORIGEN

Figura 3.1 Navegación en dos dimensiones.

En general, un sistema de navegación se usa para obtener una estimación de la posición de un vehículo con respecto a un marco de referencia conocido.

El caso más general de movimiento sobre un plano, ilustrado en la figura 3.1, se presenta cuando la vía es curva, en este caso es necesario detectar continuamente el movimiento de traslación en dos direcciones y los cambios en su dirección de viaje, es decir, detectar la rotación del tren alrededor de la perpendicular al plano de movimiento conforme el tren se mueve a lo largo de la vía. Dos acelerómetros se necesitan ahora para detectar el movimiento traslacional en direcciones que van a lo largo y de manera perpendicular a la vía. Pero además necesitamos conocer la dirección del movimiento; un sensor muy adecuado para medir el movimiento rotacional es un giróscopo. Dependiendo de la forma de construcción de este sensor, puede usarse para proporcionar ya sea una medición directa de la orientación del tren con respecto al marco de referencia, o una medición de la tasa de giro del tren. En el último caso, la orientación angular del tren puede calcularse por integración de esta medición, pero sólo si se conoce el ángulo al inicio de la navegación. Con tal información, es posible relacionar las medidas de aceleración que se obtienen en un juego de ejes que está fijo en el tren, a un marco de referencia inercial. Las mediciones instantáneas de aceleración pueden por lo tanto ser resueltas en un marco de referencia e integradas con respecto al tiempo, para determinar la velocidad instantánea y la posición del vehículo con respecto a dicho marco.

Claramente entonces, es posible construir un sistema sencillo de navegación bidimensional usando un giróscopo, dos acelerómetros y una computadora. En la práctica, los sensores inerciales pueden montarse en una plataforma; la cual es estabilizada en el espacio, y aislada de las rotaciones del vehículo, o montada directamente en él para formar un sistema sujeto al vehículo o *strapdown*. Las mediciones serán procesadas en la computadora para proporcionar estimaciones continuas de la posición, velocidad y la orientación del tren. Debe enfatizarse que la navegación inercial es fundamentalmente dependiente del conocimiento exacto de los valores iniciales de posición, velocidad y orientación. Esto es debido a que el sistema utiliza para su actualización las estimaciones previas de estas variables.



TESIS CON FALLA DE ORIGEN

Figura 3.2 Diagrama de bloques de un sistema de navegación inercial en dos dimensiones del tipo sujeto al vehículo.

3.2.1 Componentes del sistema inercial en dos ejes desarrollado

En esta sección se describe el sistema de navegación inercial utilizado para determinar la orientación de la plataforma en dos ejes. El eje Z es medido directamente con la brújula electrónica, que también se usa para hacer la calibración inicial de la orientación de la plataforma. Los componentes principales del sistema son: giróscopos, acelerómetros, inclinómetros, brújula electrónica, modelos gravitacional y dinámico y computadora de abordo.

Giróscopos. Proporcionan la medición de las velocidades angulares de los tres ejes de la plataforma. Son de estado sólido, lo que ofrece ventajas significativas en cuanto a costo y duración, y su respuesta ha sido caracterizada de manera exhaustiva. Su resolución es de 0.005°/s, con una respuesta en frecuencia < 50 Hz, y su salida, es un voltaje proporcional a la velocidad angular.

Acelerómetros. Nos proporcionan la medición de la fuerza específica en dos de los ejes de la plataforma. Los acelerómetros no pueden distinguir, al efectuar sus mediciones, entre las fuerzas inerciales y las fuerzas gravitacionales, por esta razón los llamamos sensores de fuerza específica, esto quiere decir que debemos utilizar un modelo de campo gravitacional para calcular continuamente su valor conforme transcurre el vuelo y restarlo de la medición efectuada por los sensores. Su frecuencia de operación es de 0 a 250 Hz., con una sensibilidad de 1 V/g y su intervalo de operación es de ± 2 g.

Brújula electrónica e inclinómetros. Una brújula electrónica determina el movimiento de guiñada de la plataforma durante el vuelo. Este sensor está basado en un magnetómetro de tres ejes para determinar la orientación con respecto al vector magnético terrestre. Es de estado sólido y maneja 12 bits (0.08 grados) de resolución, para generar el azimut de 0 a 360 grados. El mismo módulo contiene dos inclinómetros con 12 bits de resolución que determinan la inclinación de los ejes de alabeo y cabeceo en un intervalo de ± 80 grados. Este sensor incluye compensación contra la influencia de anomalías magnéticas localizadas en la proximidad. No compensa fuentes magnéticas variables, sin embargo el módulo las descubre y las suspende o intenta corregirlas automáticamente durante la ocurrencia de dichas anomalías.

Modelo gravitacional. Por simplicidad hemos utilizado el modelo elipsoidal de revolución de la Tierra de Steiler y Winter con el cual se puede calcular el valor de g con los datos de la latitud y altitud locales.

Computadora de abordo. Este sistema está basado en la utilización de circuitos de acondicionamiento, filtrado y conversión A/D. La computadora de abordo recibe las señales de los sensores, compensa las señales de los acelerómetros, resuelve las ecuaciones de Euler y despliega la orientación en un monitor. Por lo pronto se utiliza una PC, pero se prevé la utilización de alguna microcomputadora de tipo industrial debido principalmente al ambiente de vibración en las aeronaves.

Modelo dinámico. El modelo dinámico está basado en las ecuaciones diferenciales de Euler.

$$\dot{\Theta} = Q \cos \Phi - R \sin \Phi$$

$$\dot{\Phi} = P + \tan \Theta (Q \sin \Phi + R \cos \Phi)$$

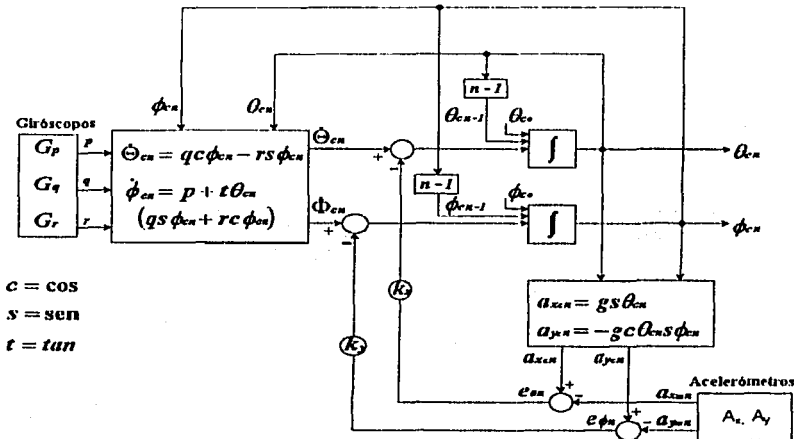
TESIS CON
FALLA DE ORIGEN

$$\dot{\psi} = Q \frac{\text{sen } \Phi}{\text{cos } \Theta} + R \frac{\text{cos } \Phi}{\text{cos } \Theta}$$

Con la integración de $\dot{\Phi}$ y $\dot{\Theta}$ se determinan los ángulos ϕ_c y θ_c los cuales ya están referidos a un espacio inercial.

3.2.2. Esquema de determinación de orientación asistido por gravedad

En esta sección se mostrarán los esquemas de determinación utilizados para el desarrollo de este proyecto. Se está hablando básicamente de un sistema asistido por gravedad con medidas redundantes de inclinación.



$c = \text{cos}$
 $s = \text{sen}$
 $t = \text{tan}$

TESIS CON FALLA DE ORIGEN

Figura 3.3. Sistema de navegación inercial desarrollado para obtener una referencia de orientación.

3.2.3 Filtrado complementario

Las fuentes de error en los sensores, y por consiguiente en los sistemas inerciales, son principalmente errores de banda ancha debidos al ruido; y errores de banda angosta, debidos a la deriva o una mala calibración. Estos dos tipos de errores, afortunadamente, son perfectamente distinguibles y esta es la clave para lograr una asistencia mutua o una fusión de las mediciones, por medio de la cual, las lecturas de por lo menos dos tipos de sensores (cada uno perteneciendo a alguna de las dos categorías mencionadas), son mezcladas con la finalidad de ayudarse mutuamente en las frecuencias donde tienen problemas. Esta ayuda mutua ha sido materializada en un método simple conocido como filtrado complementario, inicialmente implementado de manera analógica; y ahora con las computadoras digitales cada vez con mayor capacidad y rapidez, por medio del filtro de Kalman. Este concepto de filtrado complementario es el que se ha aplicado para el desarrollo de este sistema, donde los sensores primarios son los giros copos y los complementarios los acelerómetros. El

algoritmo está basado en la solución de las ecuaciones de Euler, donde se relacionan las velocidades angulares de Euler $\dot{\phi}, \dot{\theta}, \dot{\psi}$ con las velocidades angulares P, Q y R, medidas por los giróscopos montados en la plataforma, G_p, G_q y G_r , con respecto al espacio inercial. En los primeros análisis de este sistema se tenía una idea muy ambigua de los valores de K_x y K_y , pero posteriormente se encontró que si son suficientemente pequeños, entonces los términos que provocan el ruido de banda ancha de la aceleración horizontal, se mantendrán aceptablemente reducidos, en caso contrario es necesario incluir un filtro en serie con $e_{\theta n}$ y $e_{\phi n}$.

3.3 Control de apuntamiento de la plataforma

En esta sección se presentan las acciones básicas para la corrección de posición en la plataforma, empleando un sistema de control de tipo (on-off), debido a su simplicidad de implementación, ya que todo el proceso numérico para la determinación de la orientación se realiza digitalmente por medio de una computadora, y de esta forma no se requiere el uso de convertidores digitales-analógicos, sino únicamente realizar conmutaciones de voltaje por medio de puentes H, como se explica en el capítulo 4.

3.3.1 Calibración inicial de la brújula electrónica

Para poder hacer uso del magnetómetro y así poder sensar el eje de guiñada, es necesario calibrar la brújula, por lo que la calibración se vuelve una característica importante del módulo.



TESIS CON
FALLA DE ORIGEN

Figura 3.4. Calibración de la brújula para cancelar campos magnéticos locales.

Quando se instala en las cercanías de una estructura de metal, se debe hacer girar el módulo 360 grados (horario o anti-horario) en el eje de guiñada mientras la brújula graba la información magnética con los objetos a su alrededor, y de esta forma ignora la influencia que éstos puedan tener en las lecturas reales tomadas durante su operación normal.

3.3.2 Control on/off

El control en sí consiste en detectores de error, comparadores de nivel y actuadores. El elemento de medición, que en este caso se trata de un potenciómetro acoplado a

cada eje de movimiento de la plataforma, nos proporciona una señal eléctrica que se utiliza para comparar la salida con la señal de entrada de referencia. Este elemento se encuentra en el camino de realimentación del sistema de lazo cerrado y como sus constantes de tiempo son despreciablemente pequeñas en comparación con otras mismas del sistema de control, su función de transferencia simplemente se convierte en una constante, la cual debe convertir el punto de ajuste o regulación a una referencia de entrada a las mismas unidades que la señal de realimentación del elemento de medición. Lo anterior se muestra en la figura 3.5. Cabe hacer notar que tanto la diferencia entre señales de referencia y sensor, como la detección de nivel de la señal de error y la retención de la señal de control se realizan también en la computadora.

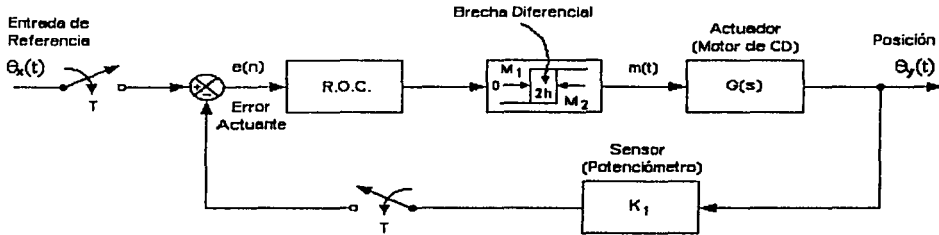


Figura 3.5 Diagrama de bloques del sistema de control.

TESIS CON FALLA DE ORIGEN

En este tipo de control el elemento actuador tiene solamente dos posiciones fijas, en este caso conectado o desconectado. Siendo la señal de control $m(t)$ y la señal de error actuante $e(n)$. La señal $m(t)$ permanecerá en un valor máximo o mínimo, según que la señal de error actuante sea positiva o negativa y sobrepase el valor de h , de modo que

$$m(t) = M_1 \quad \text{para } e(n) > h$$

$$m(t) = M_2 \quad \text{para } e(n) < -h$$

donde M_1 y M_2 son constantes y corresponden al voltaje de polarización de los puentes H que manejan a los motores. Cabe mencionar que $M_2 = -M_1$, y que es necesario incluir un bloque de retención para obtener una señal continua con pulsos de duración igual a T . El intervalo en el que se debe desplazar la señal de error actuante antes de que se produzca la conmutación se llama brecha diferencial ($2h$). La brecha diferencial hace que la salida de control $m(t)$, mantenga su valor hasta que la señal de error actuante haya pasado levemente del valor cero. Esta brecha diferencial, además de ser causada por la fricción de los elementos móviles, se le provee deliberadamente para evitar la acción excesivamente frecuente del mecanismo y es necesario determinar su valor para efectos de exactitud deseada y duración de los componentes.

Al emplear un sistema de este tipo, la señal de salida de control $m(t)$ únicamente depende de la presencia y el signo de la señal de error actuante sin importar su magnitud, lo que minimiza las ventajas de utilizar acciones de control adicionales como la proporcional, integral o derivativa. Ya que, al carecer el sistema de una correspondencia lineal entre $e(n)$ y $m(t)$ el único efecto que se logra al agregar acciones de control adicionales es reducir la brecha diferencial, lo que puede provocar inestabilidad debido a conmutaciones

sucesivas en el actuador, es por esta razón que la acción de control de tipo on-off no se utiliza conjuntamente con alguna otra.

3.4 Corte automático del control al dar vuelta para retomar la siguiente línea de vuelo

El programa que fue desarrollado con anterioridad para realizar el control de la plataforma ha sido modificado y ha crecido para ofrecer un mejor rendimiento, mayor libertad al operador, estabilidad en el tercer eje (Z) y se ha evitado la interrupción del almacenamiento de datos al momento de tomar una fotografía.

El programa original realizaba los cálculos de gravedad, ejecutaba los algoritmos para calcular la orientación a partir de los sensores inerciales y nos proporcionaba una determinada orientación referida a un sistema inercial. También mantenía la comunicación con el convertidor analógico digital, retroalimentaba el control mediante el sensado continuo de voltajes de potenciómetros y sensores inerciales, además permitía el control en tiempo real tanto del eje X, como del eje Y. Por último almacenaba la información relacionada a la orientación actual en tres ejes, en el momento de la captura de imágenes en vuelo, sin embargo este último paso presentaba errores, debido a que en ocasiones el dato no se almacenaba correctamente, es decir, se incurría en un funcionamiento que no era preciso ni continuo.

El programa ha sido cambiado y mejorado en el sentido de que por ningún motivo se interrumpe el almacenamiento de la posición al momento de la captura de imágenes, ya que es preferible almacenar este dato, que mantener la plataforma estabilizada. Esta situación se debe a que para corregir geoméricamente las imágenes, la información de la orientación es indispensable. La estabilización es importante porque no introduce distorsiones adicionales a la imagen, pero es prioritario salvar la información de la orientación.

Se ha anexado el control automático del eje Z con la ayuda de la brújula electrónica, el cual se encontraba deshabilitado por alguna razón desconocida. Esto quiere decir que la plataforma podrá orientarse de forma automática hacia la línea que se le haya programado. Durante un levantamiento aéreo, típicamente se sigue la dirección Norte-Sur y Sur-Norte o Este-Oeste y Oeste-Este. Líneas diagonales son muy raras, excepto cuando se siguen trayectorias irregulares (como el cauce de un río) o cuando la zona de estudio es una isla pequeña, por ejemplo, que no está "bien" orientada. Si se ha determinado seguir una línea Sur-Norte la plataforma mantendrá un curso hacia 0° (cero grados), debido a que así funciona para señalar el Norte.

Mediante esta modificación al programa, se ha implementado la estabilización en los tres ejes: la rotación, el cabeceo y la guiñada.

Algo nuevo en la programación fue el hacer que el operador atendiera lo menos posible el trabajo de estabilización realizado por la plataforma, en particular el programa se enfoca a que el operador al inicio de la jornada proporcione los datos y valores requeridos para que durante el vuelo no vuelva a atender a la plataforma, esto no quiere decir que se pierde el control de ella, en cualquier momento se le puede indicar al programa detenerse, para cambiar de línea o terminar. Las modificaciones importantes en el programa consistieron en establecer las opciones de:

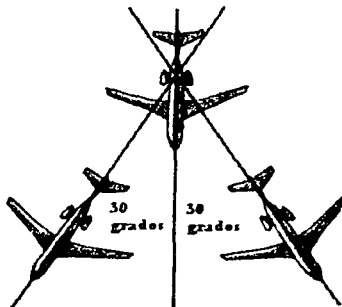
1.- Omitir las acciones de control hacia la plataforma para no efectuar la estabilización y solamente almacenar el vector de apuntamiento. Esto es útil en los momentos que existe demasiada turbulencia y la plataforma se forza demasiado, tratando de compensar desviaciones muy importantes.

2.- Solamente efectuar la estabilización de la plataforma en los ejes X,Y para el caso de que se sigan trayectorias cambiantes, como el lecho de un río, y no tratar de orientar el eje Z.

3.- Para no tener que atender al programa de control cada vez que se efectúa un cambio de línea de vuelo e independizar al usuario de esta tarea, se generaron cuatro funciones. Éstas al iniciar el programa, determinan los límites que se han de contemplar al realizar el cambio de línea. Su funcionamiento está basado en el almacenamiento del valor del voltaje de realimentación de los potenciómetros de los motores.

Estas cuatro funciones realizan esto: Mueven la plataforma a 150° y a 210° con respecto al compás, teniendo como origen los 180° y de 15° y -15° con respecto a la rotación, teniendo como origen los cero grados.

Cuando el avión está girando para tomar una nueva línea, se determinó que si la brújula indica una diferencia de 30 grados fuera de la línea programada, entonces debe decretarse un cambio en la dirección de vuelo, así por ejemplo, si la línea que se sigue es Sur – Norte la orientación está dirigida hacia los cero grados, que se toman como referencia, si se detecta un valor superior a 30 o a -30 la plataforma deja de estabilizar hasta encontrar la nueva línea que debe tener ahora una orientación Norte – Sur, esto significa que ahora va a apuntar hacia los 180 grados. Cuando la brújula detecta que el rumbo se acerca nuevamente al esperado, la plataforma se comienza nuevamente a estabilizar al acercarse a un valor para este ejemplo de 150° o 210° , o sea un valor de 30° ó -30° alrededor de la nueva línea.



ERRATA DE ORIGEN
TESIS CON

Figura. 3.6 Cantidad en grados para determinar el cambio de línea o giro del avión, en el compás.

Al mismo tiempo se considera el movimiento de rotación que efectúa la nave al dar un giro, por lo que este eje es monitoreado y cuando se detecta que ocurre una inclinación de 15 grados, entonces se determina que va a ocurrir un cambio de 180 grados en la dirección de vuelo.

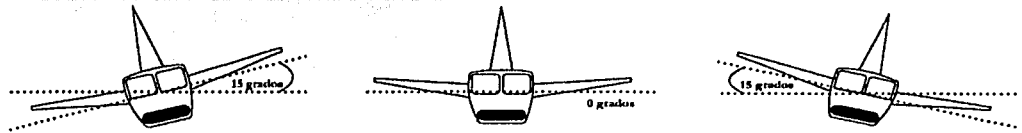


Figura. 3.7 Cantidad en grados para determinar el cambio de línea o giro del avión, en el eje de rotación.

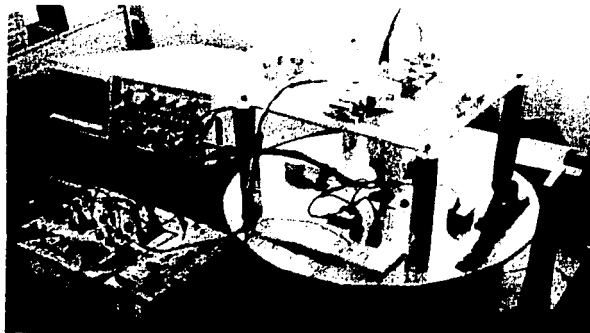
Por lo tanto se tienen dos condicionantes para determinar que el avión está girando (cambio de línea), la rotación superior a 15 grados ya sea a derecha o a izquierda y el eje Z mayor a 30 grados también a derecha ó izquierda, esto permite dejar la plataforma en un estado de latencia, esperando la nueva línea y así iniciar nuevamente la estabilización, sin que el operador haya necesitado manipular el programa para detener la plataforma e iniciar la estabilización al momento de tomar la nueva línea.

Esto permitirá mayor libertad al operador, sin tener que atender los cambios de línea y los cortes en la estabilización de la plataforma. El usuario en todo momento puede acceder el menú de opciones del programa en caso de que deba repetirse una línea, deba interrumpir momentáneamente el trabajo o lo de por terminado por alguna razón climatológica o de tráfico aéreo.

CAPÍTULO 4

PASO DE PROTOTIPOS A EQUIPOS OPERATIVOS

En este capítulo se hace una descripción detallada de aquellos equipos que han sido transformados, de prototipos de laboratorio a sistemas utilitarios, siendo una parte fundamental de este trabajo de tesis. Se hizo necesario hacer una revisión de los circuitos y de su funcionamiento, para definir e implementar posibles cambios o mejoras.



TESIS CON
FALLA DE ORIGEN

Figura 4.1. Sistema de navegación inercial y prototipo de laboratorio de la plataforma aérea.

4.1 Diseño de Tarjetas Impresas o PCB

Con el propósito de tener un mejor rendimiento en el funcionamiento general del sistema, al igual que la oportunidad de producir más fácilmente copias del conjunto o

módulos de éste, se determinó crear tarjetas impresas para cada uno de los circuitos utilizados. Esto también se hizo por la necesidad de eliminar ruido de los circuitos en "protoboard", donde se había estado trabajando en las pruebas de funcionamiento y desarrollo. Todas las tarjetas impresas fueron generadas a partir de un programa encargado del diseño de circuitos, llamado PROTEL SE99.

Protel es un sistema de diseño completo para entornos Windows y proporciona un conjunto de herramientas integradas formadas por un gestor de documentos, un capturador de esquemáticos, un simulador analógico basado en SPICE, un simulador digital basado en CUPL (lenguaje similar a ORCAD/PLD), una herramienta de diseño de placas de circuito impreso (PCBs) y soporte para trabajo en redes de computadoras, además de algunas cosas más.

En general Protel junto con Orcad constituyen las dos empresas de software de CAD electrónico más importante en entornos PC.

4.1.1 Conceptos básicos de fabricación de PCB's con PROTEL SE99

Los paquetes de diseño de PCB's están compuestos por tres herramientas, el editor de componentes, el editor de placa ó sustrato (layout) y el postprocesador, que se encarga de la impresión.

Cuando comenzamos el diseño de una nueva placa de circuito impreso, las medidas vienen impuestas por restricciones de diseño, normalmente se trabaja con medidas inglesas, se emplea la milésima de pulgada.

El trazado de las pistas de cobre se realizaba antiguamente empleando una hoja cuadrículada transparente de guía, sobre la que se colocaban una cinta opaca con la que se trazaban las pistas. Se trabajaba al doble del tamaño natural y luego se reducía por medios fotográficos, de esta forma cualquier error cometido en el trazado quedaba también reducido.

Se hizo práctica común usar múltiples hojas transparentes para las diferentes capas del trabajo, la máscara de soldadura donde aparecen las referencias de los componentes, normalmente en blanco, un patrón de taladrado para la posterior fabricación, el rutado de las pistas de cobre de una o más capas, normalmente en pares.

Este método de trabajo hacía difíciles posteriores correcciones al diseño. Hoy día debido a la profusión de computadoras personales potentes, existen herramientas para facilitarnos el trabajo.

La librería de componentes es la parte más importante del programa de diseño. Un programa de captura de esquemáticos con herramientas de diseño de PCB tiene enlaces que definen; como un símbolo del esquemático se relaciona con un símbolo de PCB. El procesado del esquemático genera un archivo de "netlist" para el programa de ruteado. Hay que prestar atención a como se relacionan la numeración de los pines del símbolo en el esquemático y en el encapsulado del componente. E, B y C de un transistor no siempre coinciden con 1, 2 y 3 del componente físico.

Sin embargo, siempre podemos diseñar encapsulados especiales. La librería de componentes puede ser tan sencilla o complicada como necesitemos. Toda la información de lista de materiales se puede almacenar con cada componente si creamos un PCB a partir de un esquemático.

La mayoría de los diseños tienen un tamaño de PCB predefinido. Comenzamos definiendo las dimensiones exteriores de la placa y los agujeros de montaje necesarios. Resulta útil imprimir un dibujo de la placa para recortarla y colocarla en su emplazamiento para comprobar que se ha tomado la medida correctamente, así como la posición de los agujeros de montaje.

Luego, leemos el "netlist" creado por el programa de captura de esquemáticos. Este proceso lee componentes de las librerías y los coloca uno encima de otro en el origen del dibujo. El segundo paso importante es la correcta colocación de los componentes. Cuando se mueve un componente a una zona libre de la placa, observamos unas líneas que lo unen a otros componentes, estas nos sirven para encontrar la disposición más adecuada con respecto a los demás componentes.

Una vez que se hayan colocado los componentes se puede imprimir una prueba para comprobar la colocación de los conectores y zonas donde puedan existir problemas.

La tercera fase consiste en colocar las pistas de cobre que unen los componentes, esta fase puede ser automática o manual.

Por último la cuarta fase es la impresión de alta resolución del diseño final.

4.2 Intervalómetro

El intervalómetro es un equipo que permite el disparo de la cámara de manera independiente de la computadora. Cuenta con circuitos temporizadores basados en astables y biestables que permiten arrancar y parar el ciclo de disparo en cualquier momento. Un led indica la parte alta o baja del ciclo, el potenciómetro con contavoltas permite ajustar el intervalo de operación y un interruptor conmuta la señal de la computadora y la del circuito interno para que alguno de los dos efectúe el disparo de la cámara.

La ubicación de algunos botones fue cambiada para su mejor operación durante los levantamientos aéreos y otros fueron sustituidos por componentes de mejor calidad. Antes de generar la tarjeta impresa del intervalómetro, fue necesario actualizar algunos circuitos y conseguir equivalentes debido a que algunos de ellos ya no se encontraban disponibles en el mercado.

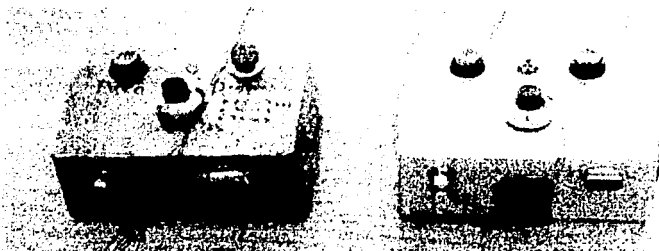


Figura 4.2. Comparación entre el intervalómetro anterior y el desarrollado para esta tesis.

TESIS CON
FALLA DE ORIGEN

Uno de los elementos más importantes de este circuito es el relevador, ya que es físicamente el que cierra el circuito de disparo de la cámara. En un inicio se manejó el relevador EDR201A0500, el cual fue sustituido por el RLYF71A05, que presenta las mismas características de operación y misma asignación de pines.

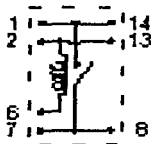


Figura 4.3 Relevador RLYF71A05.

En las siguientes figuras se muestran: el diagrama eléctrico del circuito completo del intervalómetro (figura 4.4), el diseño en PROTEL y su correspondiente tarjeta impresa (figura 4.5):

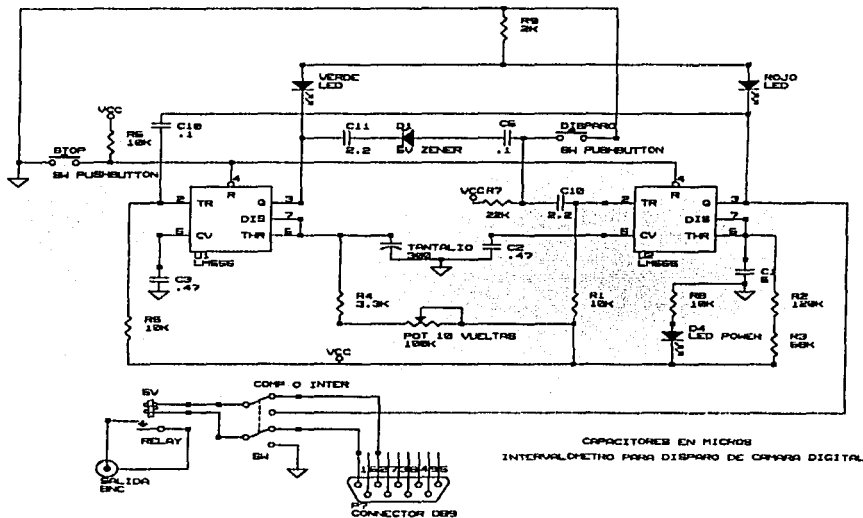


Figura 4.4 Diagrama esquemático del intervalómetro.

4.3 Circuitos de electrónica de control y manejo de sensores

Para este circuito se hicieron algunas evaluaciones y pruebas de funcionamiento con la finalidad de establecer su rendimiento. Se exploró la posibilidad de cambiar los potenciómetros de realimentación en los motores que controlan los tres ejes de la

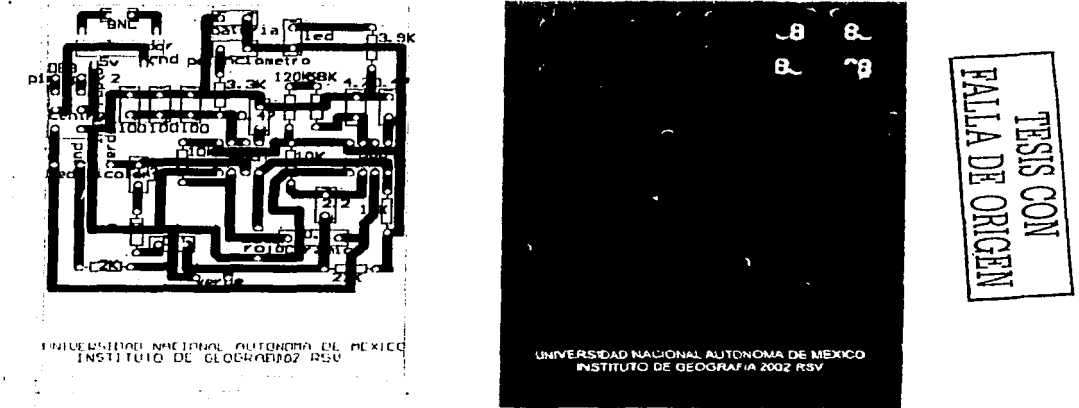


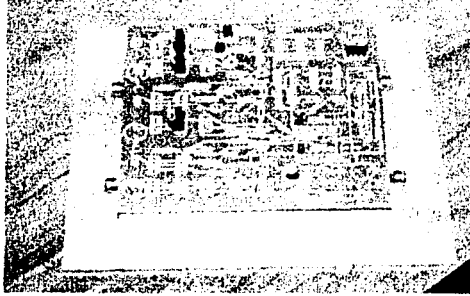
Figura 4.5 Diseño en PROTEL del circuito del intervalómetro y su tarjeta impresa.

plataforma, por "encoders"; sin embargo, después de un estudio sobre el funcionamiento de éstos últimos se llegó a la conclusión de que eran necesarias bastantes modificaciones en el sistema, incluyendo un microcontrolador que proporcionara un valor directo en grados, por lo que se determinó que no era viable su implementación dado que el tiempo no iba a ser suficiente. Por esta razón se decidió no utilizar estos dispositivos de realimentación y si mantener los potenciómetros, aunque se buscaría utilizar los de mejor calidad. En el apéndice A se presenta una descripción del funcionamiento de los "encoders", los tipos principales que existen y los pasos que hay que seguir para su utilización como dispositivos de realimentación en sistemas de control de lazo cerrado.



Figura 4.6. Circuitos de potencia y de manejo de señales de sensores, en tarjetas prototipo.

En un principio se desarrolló una sola tarjeta impresa conteniendo los circuitos electrónicos de potencia para el control de los motores y de la electrónica encargada de manejar las señales provenientes de los sensores inerciales.



TESIS CON
FALLA DE ORIGEN

Figura 4.7 Tarjeta impresa con los circuitos de potencia y de manejo de señales de sensores.

Este primer intento de pasar a una tarjeta impresa fue infructuosos debido a algunos errores en el diseño, pero principalmente a que se encontró con mucha interferencia en las señales de los sensores y el funcionamiento del sistema era bastante errático. Debido a este problema se decidió llevar a cabo un segundo diseño dividiendo el circuito en dos partes. Una tarjeta contendría todos los circuitos que harían uso de la información digital, por comodidad la llamamos "tarjeta digital", la cual se encarga de polarizar y obtener las señales de los sensores utilizados en este proyecto como son, los giróscopos y acelerómetros, al mismo tiempo también se encargará del manejo de las señales de los potenciómetros y será la tarjeta que sirva de interfaz con la computadora.

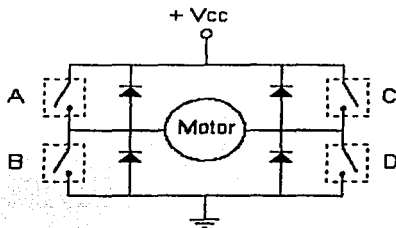
La segunda tarjeta, la cual llamamos "tarjeta de potencia", contiene la etapa de salida de corriente hacia los motores para su control. Con tres puentes H, con sus respectivos convertidores de nivel y sus controles de on/off, esta tarjeta es manipulada por la primera, para efectuar el movimiento de los motores.

Con este nuevo esquema se logró separar los voltajes, esto es, la tarjeta digital utiliza 5 y 12 Volts DC fijos regulados, mientras que la tarjeta de potencia utiliza un voltaje variable, que depende principalmente de la demanda de los motores utilizados, así como de las características de los fets de potencia y transistores que se ocupan. Así, de esta manera se logró aislar de ruido a la tarjeta que maneja los sensores.

4.3.1 Circuitos de Potencia

Un circuito digital no tiene la capacidad de corriente necesaria para hacer que un motor eléctrico gire. La manera más sencilla de manejar un elemento electromecánico con un circuito digital es utilizando un TRANSISTOR como interruptor.

En este caso la etapa última de control la lleva a cabo un circuito que controla la posición de los motores, por medio de pulsos obtenidos desde el puerto paralelo de la computadora y determinados por el algoritmo de control. Por la forma en que se obtiene el proceso de control de la posición de la plataforma, explicado en el capítulo 3, se necesita un circuito que permita realizar la conmutación del flujo de corriente que se envía al motor de CD para que éste gire en ambas direcciones, pero además se necesita suprimir este flujo para detener al motor. Uno de los circuitos que permiten controlar motores eléctricos de CD en dos direcciones desde un circuito digital (TTL, CMOS, el puerto de una computadora, un microcontrolador, etc.) es conocido como "Puente H", porque su forma recuerda a esa letra. Un puente H es básicamente un arreglo de cuatro interruptores acomodados como se muestra en la figura 4.8.



TESIS CON
FALLA DE ORIGEN

Figura 4.8 Arreglo básico de un puente H.

Estos interruptores (A, B, C y D) pueden ser transistores bipolares, mosfets, jfets, relevadores o cualquier combinación de elementos. El puente H sirve para que un motor eléctrico de corriente directa funcione en ambos sentidos, sin tener que manejar voltajes negativos. Si se cierran solamente los contactos A y D, la corriente circulará en un sentido a través del motor, y si se cierran solamente los contactos B y C, la corriente circulará en sentido contrario. Resulta evidente que los contactos A y B ó C y D no deben cerrarse al mismo tiempo para evitar un corto circuito.

La figura 4.9 muestra el puente H, los transistores Q1, Q2, Q3 y Q4 son transistores NPN pequeños (2N3904) que se usan como convertidores de nivel. Estos transistores están conectados como inversores y en este caso convierten los niveles lógicos TTL a +12 y 0.0 V para controlar los FETs.

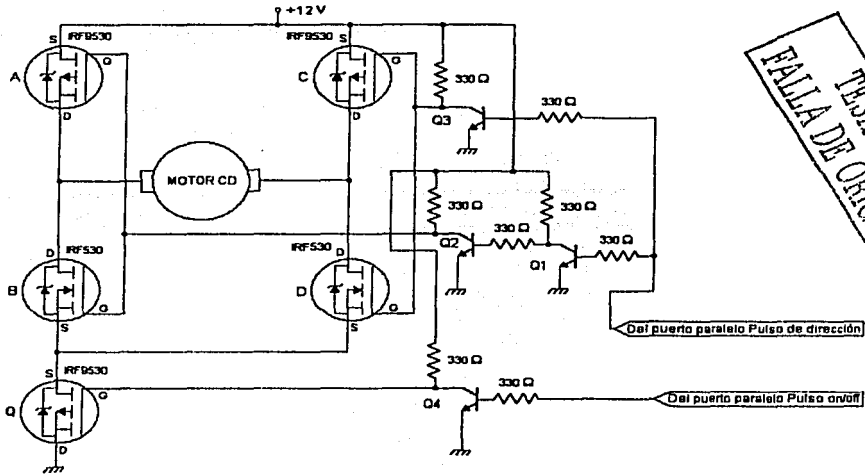


Figura 4.9 Puente H con MOSFETs.

4.3.2 Fabricación de tarjetas impresas para los circuitos de manejo de sensores y de potencia

Dadas las observaciones posteriores al diseño y construcción de la primera tarjeta, se inició el nuevo diseño tomando en consideración un mayor tamaño de las pistas en el circuito de potencia, para que fueran más gruesas y fueran capaces de manejar motores suficientemente grandes (con mayor corriente). De datos del fabricante de las tarjetas, se menciona que pistas con un ancho de 2 mm son capaces de manejar fácilmente 2 Amp., dada la cantidad de material empleado para la pista. La separación entre elementos se contempló también para que los fets utilizados disiparan rápidamente el calor. Para el circuito de manejo de señales existen menos restricciones, dado que se manejan corrientes y voltajes pequeños. El diseño del circuito se basa en el siguiente diagrama esquemático que conjunta ambos circuitos, este diagrama se realizó también con el programa PROTEL.

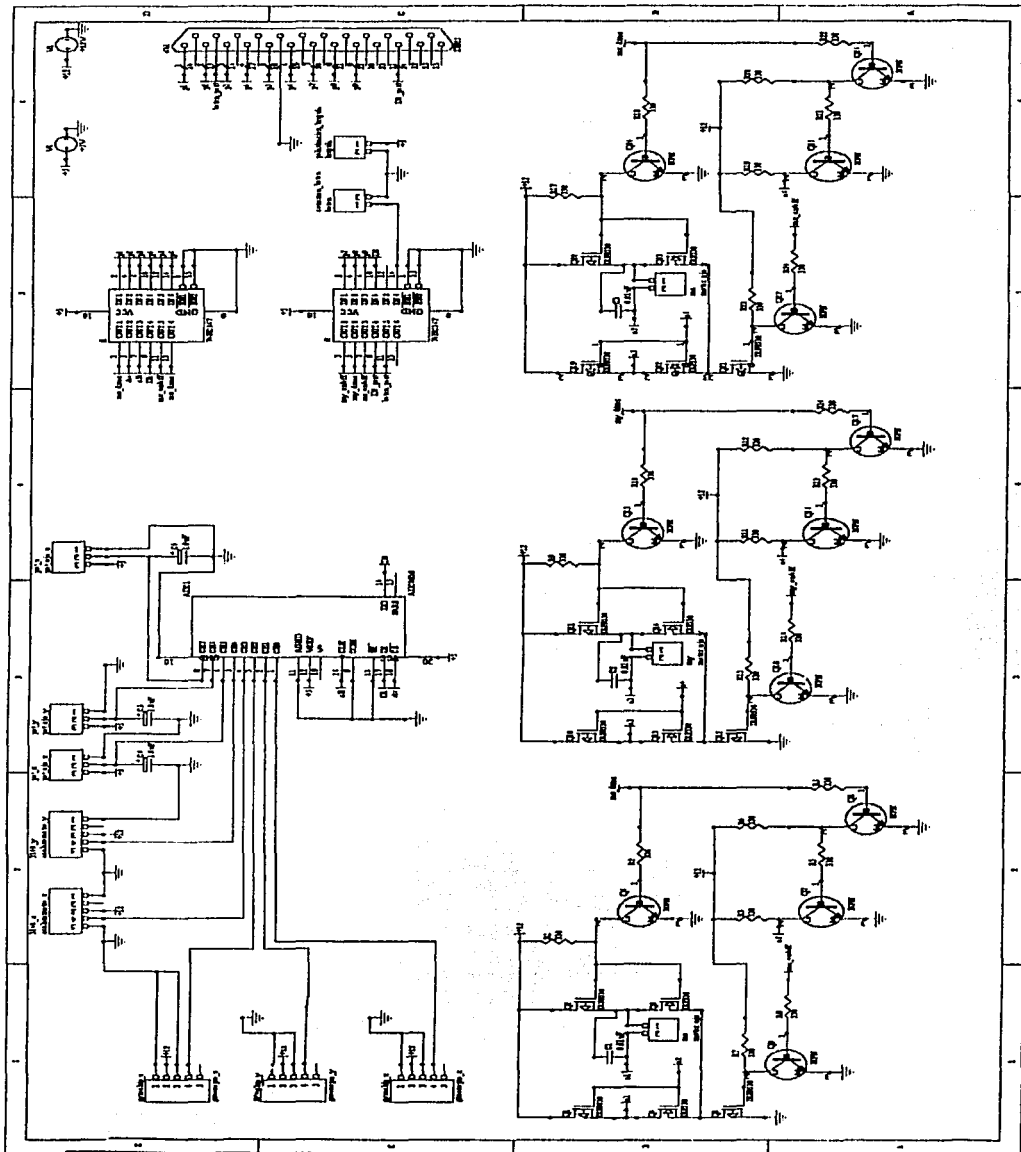
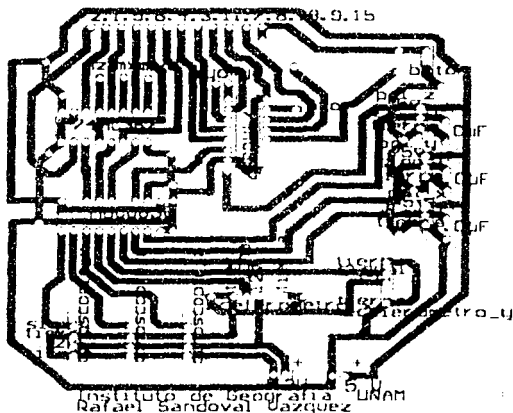


Figura 4.10. Diagrama esquemático de los circuitos de control de motores y manejo de señales de sensores.

TESIS CON
FALLA DE ORIGEN

La tarjeta digital es la siguiente en su diseño final.



TESIS CON
 FALTA DE ORIGEN

Figura 4.11. Tarjeta impresa del circuito digital.

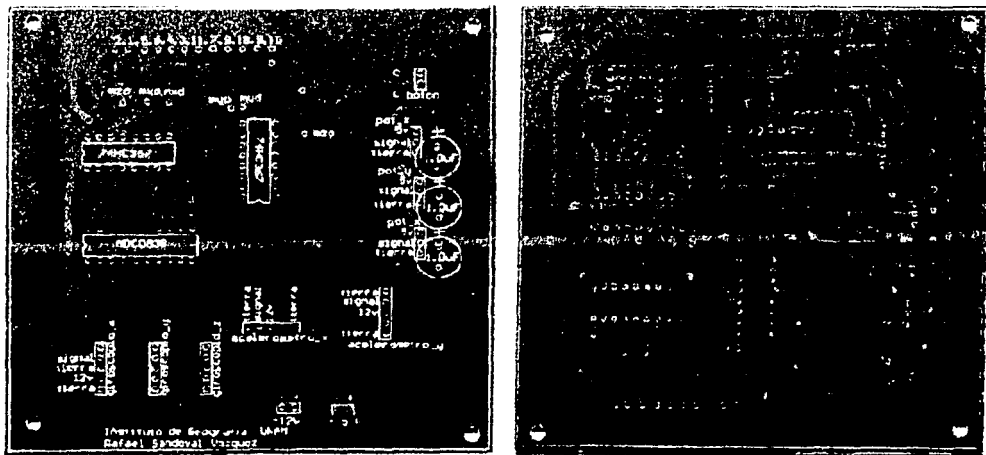
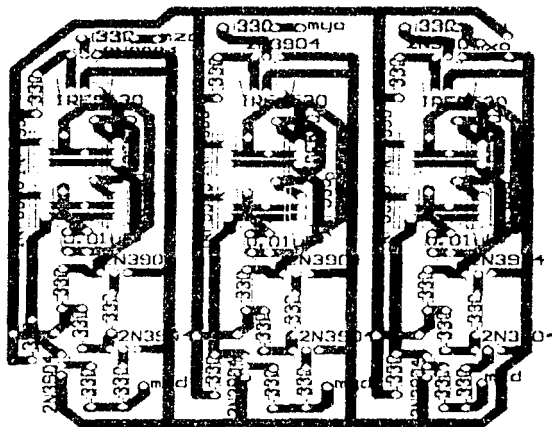


Figura 4.12. Caras superior e inferior de la tarjeta digital.

La tarjeta de potencia es la siguiente:



TESIS CON
FALLA DE ORIGEN

Figura 4.13 Tarjeta impresa del circuito de potencia.

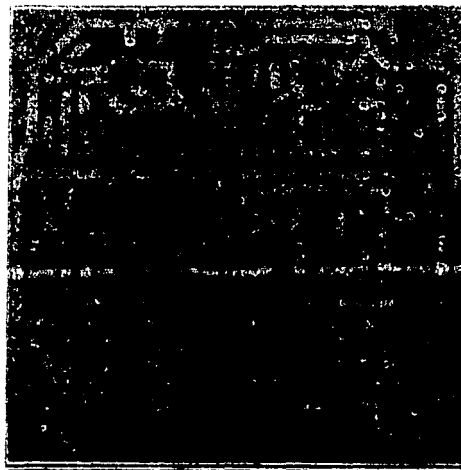
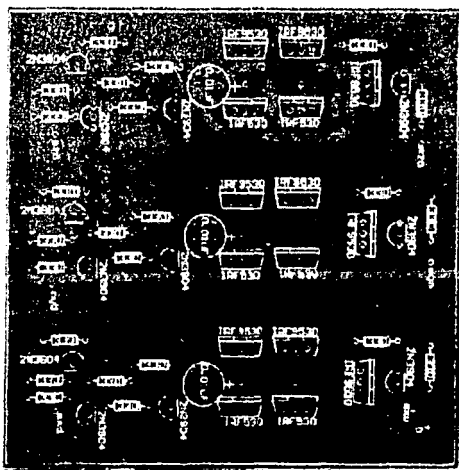


Figura 4.14 Caras superior e inferior de la tarjeta de potencia.

4.4 Fuentes de Energía

Para este proyecto se diseñaron y se construyeron fuentes de voltaje para el funcionamiento de la plataforma de manera autónoma. Proporcionan 12 y 5 Volts DC fijos para la etapa digital y un voltaje variable de 3 a 17 Volts DC para la etapa de potencia.

Estas juegan un papel importante debido a que su uso será en vuelo, durante los trayectos a los que se destine la plataforma, durante los levantamientos aéreos.

4.4.1 Fuente de Voltaje Regulado de 12 y 5 Volts DC fijos

Este equipo está destinado a alimentar la tarjeta digital y entrega 12 y 5 Volts DC fijos, los cuales se obtienen del L7812CV y del L7805CV respectivamente. Estos son circuitos reguladores de voltaje con salida fija capaces de entregar 1 Amp cada uno. Cuentan con una protección interna que apaga el integrado protegiéndolo de un exceso de corriente o sobrecalentamiento. No requieren de componentes externos para funcionar. Los reguladores trabajan en el siguiente intervalo:

| Ci | V _{IN} mínimo | V _{IN} máximo |
|--------|------------------------|------------------------|
| L7805C | 7.5 | 20 |
| L7812C | 14.5 | 27 |

Con estos intervalos de operación se determinaron los voltajes que deberían tener las baterías, considerando la necesidad de mantener una adecuada regulación. El voltaje con que se alimentan los reguladores es de 16 Volts de DC, lo que significa que se conectaron en serie dos baterías: una de 4 Volts DC @ 9 Amp-hr. y otra de 12 Volts DC @ 7 Amp-hr. Las baterías tienen las siguientes características:

Especificaciones Generales para la batería de 4 Volts DC

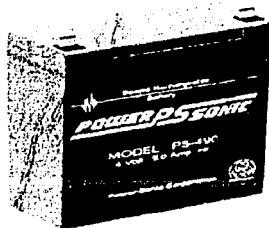
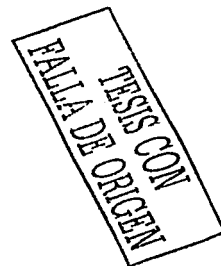


Figura 4.15 Batería de 4 Volts DC 9.0 AH.



Dimensiones:

| | |
|-----------------|-----------------|
| Longitud | 101 milímetros |
| Ancho | 44 milímetros |
| Altura | 95 milímetros |
| Peso Aproximado | 1.28 Kilogramos |
| Envase | Plástico ABS |

Especificaciones Eléctricas:

| | |
|---|------------------------------------|
| <i>Voltaje nominal</i> | <i>4 Volts (2 celdas en serie)</i> |
| <i>Capacidad nominal</i> | <i>9.0 AH</i> |
| <i>Resistencia Interna (carga máxima)</i> | <i>10 mΩ (aprox.)</i> |

Especificaciones Generales para la batería de 12 Volts DC

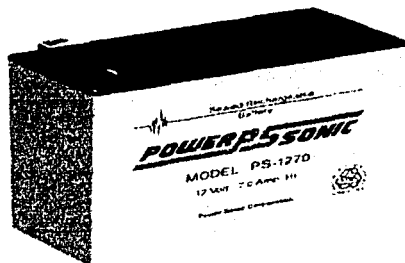
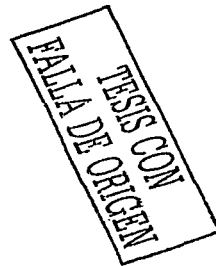


Figura 4.16 Batería de 12 Volts DC 7.0 AH.



Dimensiones:

| | |
|------------------------|------------------------|
| <i>Longitud</i> | <i>151 milímetros</i> |
| <i>Ancho</i> | <i>65 milímetros</i> |
| <i>Altura</i> | <i>94 milímetros</i> |
| <i>Peso Aproximado</i> | <i>2.60 Kilogramos</i> |
| <i>Envase</i> | <i>Plástico ABS</i> |

Especificaciones Eléctricas:

| | |
|---|-------------------------------------|
| <i>Voltaje nominal</i> | <i>12 Volts (6 celdas en serie)</i> |
| <i>Capacidad nominal</i> | <i>7.0 AH</i> |
| <i>Resistencia Interna (carga máxima)</i> | <i>22 mΩ (aprox.)</i> |

Para la construcción física de esta fuente se diseñó y construyó un gabinete metálico, el cual alberga los circuitos utilizados, disipadores, cables, conectores, carátulas e interruptores permitiendo un manejo adecuado y fácil interconexión. La caja se proveyó de dos bornes que permiten recargar las baterías sin necesidad de sacarlas del contenedor, esta recarga será muy simple ya que solo con mover el mismo interruptor que se usa para el encendido, hacia el lugar indicado (apagado), se podrá efectuar la recarga. El contenedor cuenta con dos carátulas para voltaje de DC, ambas pueden funcionar en el intervalo de 0 a 15 Volts DC, las cuales permitirán monitorear en todo momento los voltajes de salida (5 Volts DC y 12 Volts DC). Tiene un LED indicador de encendido, al igual que un fusible de protección.

El circuito utilizado es el siguiente:

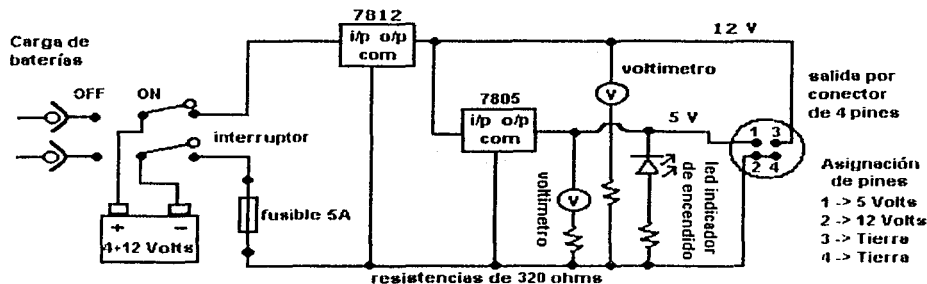
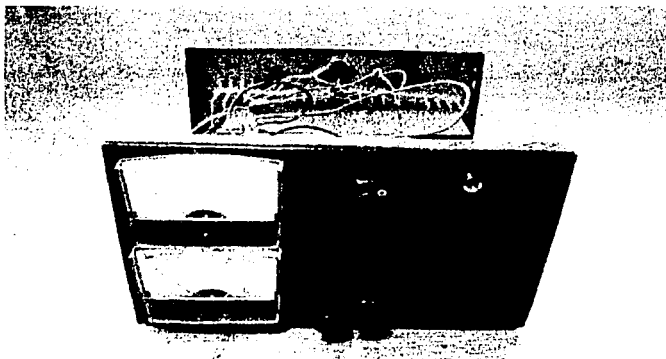


Figura 4.17. Circuito utilizado en la fuente regulada con voltajes fijos de 12 y 5 volts.



TESIS CON
 FALLA DE ORIGEN

Figura 4.18. Fuente de 12 y 5 volts diseñada y construida específicamente para esta aplicación.

4.4.2 Fuente de Voltaje Regulado, variable de 1.2 a 17 Volts DC

Esta fuente está destinada a alimentar los motores utilizados como actuadores en el control de la orientación de la plataforma. Utiliza el circuito integrado LM350K (encapsulado metálico TO-3) el cual permite variar el voltaje de salida entre 1.2 y 33 Volts con una corriente hasta de 3 Amperes. En nuestro caso la variación de voltaje se limitará entre 1.2 y 17 Volts, debido al voltaje de entrada de 18 volts que se suministra por medio de baterías recargables. La cantidad de corriente que drene la fuente dependerá de los motores utilizados, y de las especificaciones de los circuitos que trabajan en la etapa de potencia. Cada circuito integrado LM350K puede manejar un máximo de 3 Amperes, en el circuito que se presenta para la construcción de esta fuente se colocaron dos LM350K en paralelo para poder drenar un máximo de 6

Amperes, en caso de ser necesario. Los fets de potencia que se están manejando, drenan un máximo de 17 Amperes, por lo que no constituyen ninguna limitante.

La principal precaución que se debe tomar, es montar los LM350K en un buen disipador térmico debidamente aislado, y de proporcionar al circuito protección contra algún corto que se pueda presentar.

El circuito utilizado en esta fuente de voltaje variable es el siguiente:

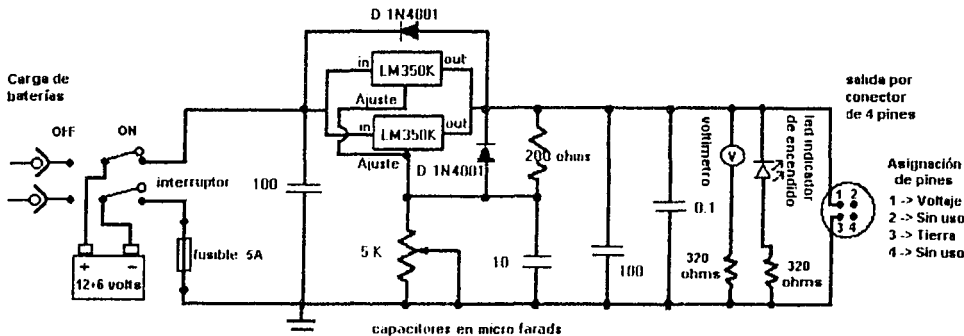


Figura 4.19. Circuito utilizado para la fuente regulada con voltaje variable.

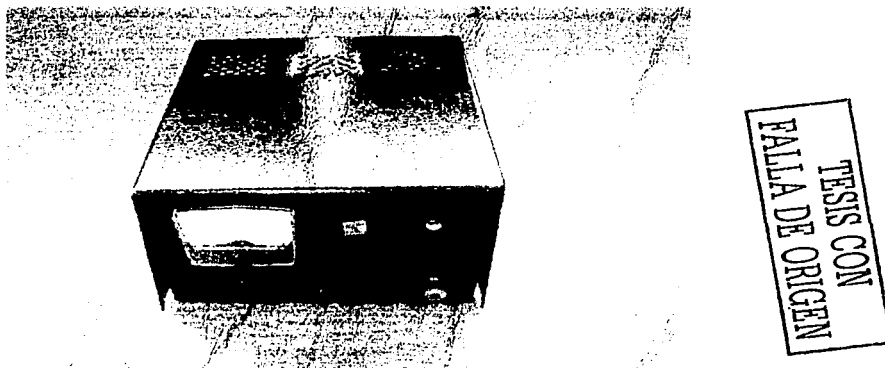


Figura 4.20. Fuente de voltaje variable.

Las baterías que se utilizaron para esta fuente son de 12 Volts DC @ 7 AH y 6 Volts DC @ 8 AH, que conectadas nos proporcionan 18 Volts DC, de ahí que nuestro valor máximo de voltaje para la fuente sea de 17 Volts. También cuenta en su chasis con dos bornes para la carga de las baterías, la carátula que indica el voltaje de salida, una perilla de ajuste de voltaje, el LED indicador de encendido, el interruptor para

encenderla o para cargar baterías y el conector del cual tomaremos el voltaje hacia los motores (a través del puente H), así como un fusible de protección.

Las características de la batería de 12 Volts han sido descritas anteriormente, para la de 6 Volts @ 8 Amp-hr son las siguientes:



TESIS CON
FALLA DE ORIGEN

Figura 4.21 Batería de 6 Volts DC @ 8.0 A-H.

Dimensiones:

| | |
|------------------------|------------------------|
| <i>Longitud</i> | <i>98 milímetros</i> |
| <i>Ancho</i> | <i>56 milímetros</i> |
| <i>Altura</i> | <i>118 milímetros</i> |
| <i>Peso Aproximado</i> | <i>1.50 Kilogramos</i> |
| <i>Envase</i> | <i>Plástico ABS</i> |

Especificaciones Eléctricas:

| | |
|---|------------------------------------|
| <i>Voltaje nominal</i> | <i>6 Volts (3 celdas en serie)</i> |
| <i>Capacidad nominal</i> | <i>8.0 AH</i> |
| <i>Resistencia Interna (carga máxima)</i> | <i>15 mΩ (aprox.)</i> |

4.5 Conectores y Conexiones

Para lograr la comunicación entre la plataforma aérea y el módulo que contiene los sensores inerciales y las tarjetas de la electrónica de potencia y manejo de sensores, ha sido necesario efectuar una serie de cableados en los que se utilizaron conectores y cables con determinadas especificaciones (figura 4.22). De la misma manera se hicieron algunos conectores para transferir la energía de las fuentes de voltaje hacia sus respectivas cargas.

Para comunicar la plataforma con el contenedor de los circuitos se ocuparon conectores de tipo militar dada la aplicación aeronáutica con que nos enfrentamos. Se han utilizado conectores hembra para cable MS3116F 14-19S y conectores macho para chasis MS3112E 14-19P marca Amphenol, de 19 pines fabricados de metal y protegidos contra la humedad y la corrosión.

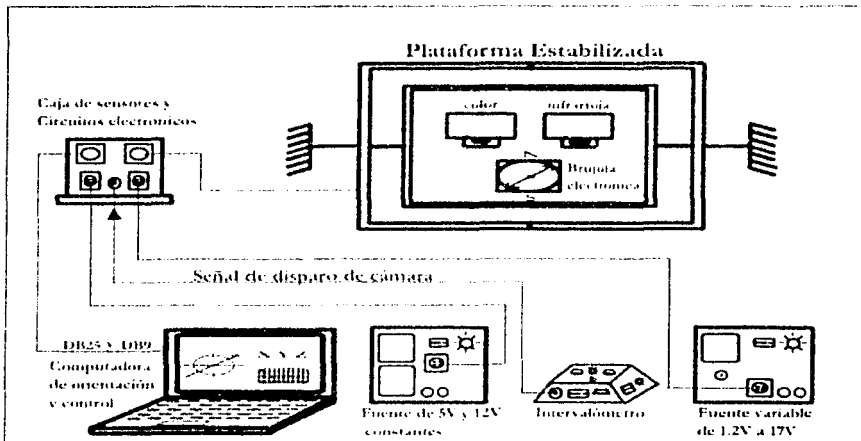
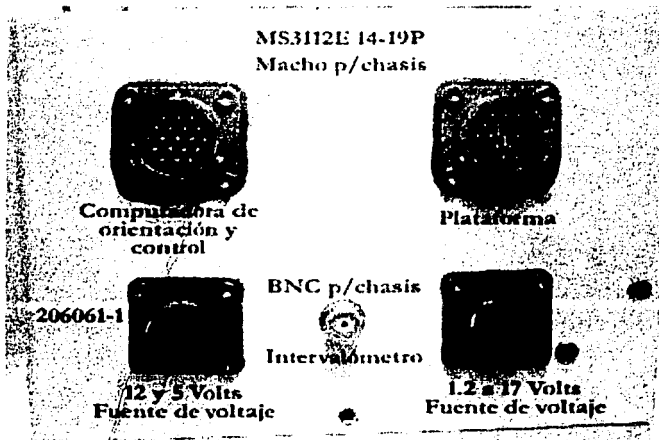


Figura 4.22 Diagrama de bloques para la conexión del sistema.

En los cables que conectan las fuentes de voltaje para alimentar los circuitos, se utilizaron conectores para cable y para chasis 206061-1 marca AMP, de 4 pines, éstos pueden convertirse en macho o hembra según se necesite.



TESIS CON
 FALLA DE ORIGEN

Figura 4.23. Conectores empotrados en la caja de sensores.

Para la conexión entre la caja de sensores con la computadora de orientación y control, y la plataforma, se utilizó cable blindado de 25 vías, cada vía tiene calibre 24, y para los conectores 206061-1 se ocupó un cable blindado de 10 vías, también de calibre 24.

4.5.1 Conexión Computadora-Circuitos

El cable empleado para comunicar el programa de control (residente en la computadora) con el contenedor de circuitos, utiliza los conectores: MS3116F 14-19S, (hembra p/cable), DB25 macho para cable y un DB9 hembra para cable.

Las conexiones fueron las siguientes:

Tabla 4.1

| Pines del MS3116F 14-19S (hembra p/cable) | Pines del DB25 (macho p/cable) | Función |
|---|--------------------------------|---------------------------------|
| A | 1 | Dirección del motor eje Z |
| B | 2 | Chip Select del ADC0838 |
| C | 3 | Señal de reloj del ADC0838 |
| D | 4 | Entrada Digital del ADC0838 |
| E | 5 | On/off del motor eje X |
| F | 6 | Dirección del motor eje X |
| G | 7 | On/off del motor eje Y |
| H | 8 | Dirección del motor eje Y |
| J | 9 | On/off del motor eje Z |
| K | 11 | Salida digital del ADC0838 |
| L | 15 | Comunicación con Intervalómetro |
| M | 18 | GND |

Tabla 4.2

| Pines del MS3116F 14-19S (hembra p/cable) | Pines del DB9 (hembra p/cable) | Función |
|---|--------------------------------|---------------------------------------|
| P | 2 | Recepción en la Brújula Electrónica |
| V | 3 | Transmisión en la Brújula Electrónica |
| T | 5 | GND |

4.5.2 Conexión Circuitos-Plataforma

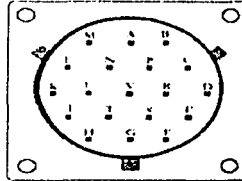
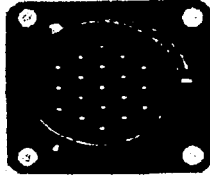
El cable empleado para comunicar el contenedor de los circuitos con la plataforma utiliza: dos conectores MS3116F 14-19S (hembra p/cable), uno de los cuales se conecta al mismo contenedor y el otro en la plataforma. En la figura 4.24 se muestra la asignación de pines de los conectores MS3116F 14-19S (hembra p/cable) y el MS3112E 14-19P (macho p/chasis).

La conexión queda de la siguiente forma:

Tabla 4.3

| Pines del MS3116F 14-19S (hembra p/cable) | Pines del MS3112E 14-19P (macho p/chasis) | Función |
|---|---|---------------------------|
| A | A | 5 V DC |
| B | B | GND |
| C | C | Señal del Potenciómetro X |
| D | D | Señal del Potenciómetro Y |
| E | E | Señal del Potenciómetro Z |

| | | |
|---|---|---------------------|
| F | F | Control Motor X uno |
| G | G | Control Motor X dos |
| H | H | Control Motor Y uno |
| J | J | Control Motor Y dos |
| K | K | Control Motor Z uno |
| L | L | Control Motor Z dos |
| P | P | Rx |
| V | V | Tx |
| T | T | GND |



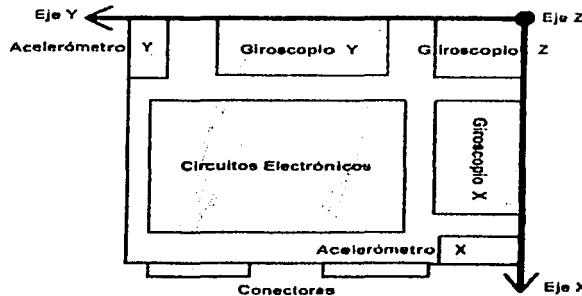
TESIS CON FALLA DE ORIGEN

Figura 4.24 Conector MS3112E 14-19P.

4.6 Contenedor de Circuitos y Sensores

El material del que está formado el contenedor de circuitos y sensores es Aluminio 6061-T6, la composición de este Aluminio es: 0.40/0.80% de silicio, 0.7% máximo de hierro, 0.15/0.40% de cobre, 0.8/1.2 de magnesio, 0.04/0.35 de cromo, 0.25 máximo de zinc y 0.015 máximo de titanio. Adicionalmente a sus características naturales como una excelente conductividad, ligereza, nula toxicidad y que no produce chispa presenta: Resistencia superior a las aleaciones 6063 (más suaves), tratado térmicamente, envejecido artificialmente, óptima conformación en frío, excelente resistencia a la corrosión, gran resistencia a la tensión y excelente maquinabilidad.

La distribución de los sensores dentro de este cubo de Aluminio fue la siguiente:



TESIS CON FALLA DE ORIGEN

Figura 4.25. Distribución de sensores y tarjetas electrónicas dentro del contenedor.

Las dimensiones del cubo de aluminio son las siguientes:

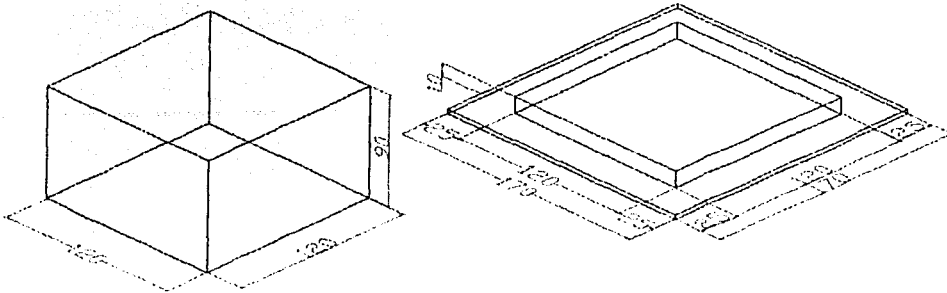


Figura 4.26. Dimensiones de contenedor de circuitos dadas en mm.

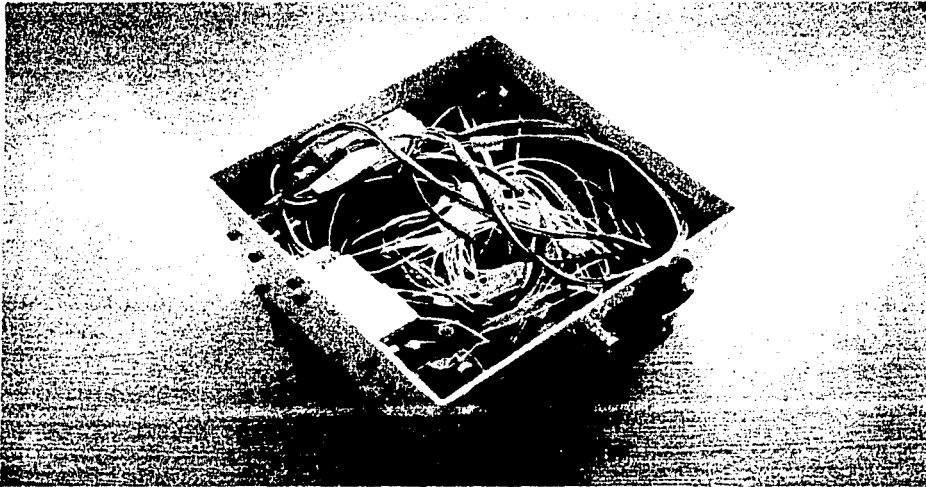


Figura 4.27. Interior de la caja de sensores, donde se aprecian los sensores inerciales. Las tarjetas digital y de potencia se encuentran en la parte baja, cubiertas por los cables.

TESIS CON
FALLA DE ORIGEN

CAPÍTULO 5

PRUEBAS DE FUNCIONAMIENTO Y RESULTADOS

El objetivo de este capítulo es dar a conocer los resultados obtenidos en este trabajo de tesis, así como hacer una descripción de las pruebas de funcionamiento realizadas tanto al programa, como a la plataforma de prueba. Debido a que hasta este momento no se cuenta con la plataforma definitiva, las pruebas de funcionamiento se han efectuado solamente en tierra con un prototipo de laboratorio de la plataforma. Se llevó a cabo la comprobación del correcto funcionamiento por separado y en conjunto de los distintos programas y equipos que conforman el sistema de adquisición de imágenes, para poder de esta forma hacer los ajustes necesarios.

La interfaz entre la computadora y el usuario se trató de que fuera lo más comprensible y sencilla posible, para evitar al máximo la interacción del operador con los programas de monitoreo y control, y así prestar mayor atención a otras actividades.

El programa de monitoreo sigue los procedimientos establecidos para llevar a cabo los levantamientos. Utiliza tres rutinas principales: configuración inicial, configuración de mapa (delimitación de la zona de vuelo) y modo de vuelo. Así, en la primera parte, se captura la información concerniente a la planeación del vuelo, en la segunda se captura un mapa y se trazan las líneas sobre la zona de estudio y la tercera, se encarga de efectuar el monitoreo de las variables más importantes del levantamiento. Este programa ha sido extensivamente probado en vuelo por lo que se considera que su funcionamiento es adecuado. Únicamente fue necesario hacer una pequeña modificación en cuanto a la distancia focal del lente utilizado en un levantamiento en particular. Por esta razón, en este capítulo únicamente discutiremos los cambios y las pruebas de funcionamiento efectuados al programa de control de estabilización de la plataforma portacámaras.

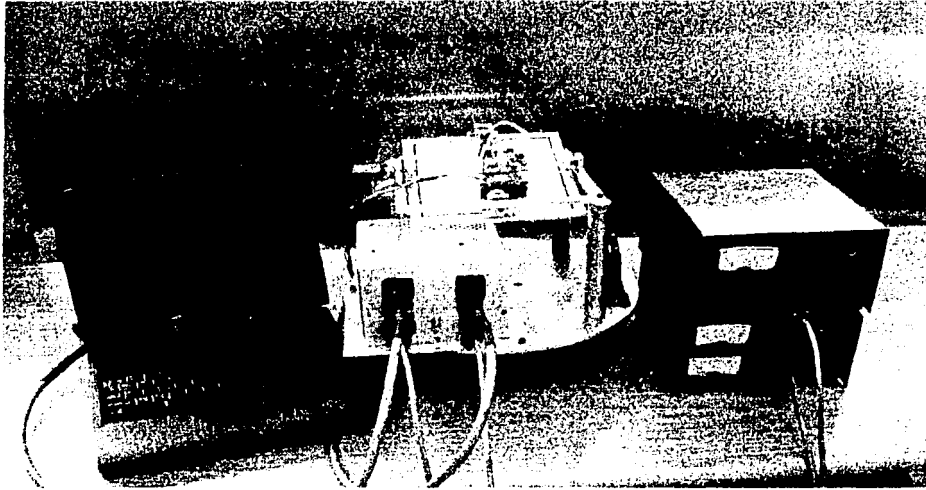


Figura 5.1 Sistema de detección de orientación y de control de estabilización para una plataforma aérea.

El programa de detección de orientación y de control de estabilización de la plataforma, efectúa los siguientes pasos: captura la configuración inicial, lleva a cabo la determinación de la horizontal local y el Norte (calibración inicial). Captura la dirección de la primera línea que se habrá de seguir durante el levantamiento, se encarga de llevar a cabo el control automático de la plataforma en tiempo real y almacena la información de los tres ejes de orientación al momento de capturar una imagen. También es capaz de determinar cuando la nave termina una línea y se prepara para tomar la siguiente, para ello compara los ángulos de rotación y guiñada contra umbrales preestablecidos y así inicia, de forma totalmente automática, el control de la nueva línea.

```

Introduzca el nombre del archivo para almacenamiento:
                (Enter para default)
                posicion.txt

Introduzca la latitud:
                Grados: 19           Minutos: 43

Introduzca la altitud en metros: _
  
```

Figura 5.2. Pantalla de configuración inicial del programa de control de estabilización de la plataforma.

5.1 Asignación de Valores y Calibración Inicial

Al momento de iniciar la ejecución del programa, internamente se está llevando una revisión de los posibles puertos de entrada y salida de datos, tanto serie como paralelo, que se tienen en la computadora. También se determina si éstos se encuentran ocupados por un dispositivo ajeno al proyecto, si es el caso; el siguiente mensaje es mostrado:

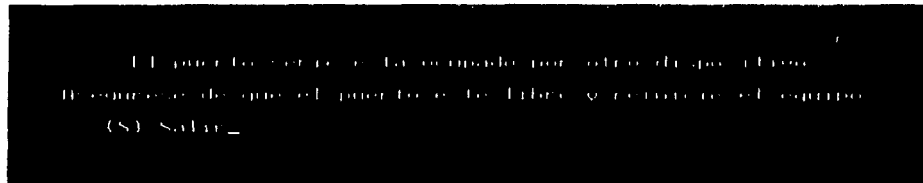


Figura 5.3. Mensaje de que el puerto serie no se encuentra disponible.

En caso de que no se detecte la comunicación específica que se debe tener con la brújula electrónica, a causa de que ésta no se haya conectado en el puerto serie o no se encuentre polarizada, el mensaje será este:

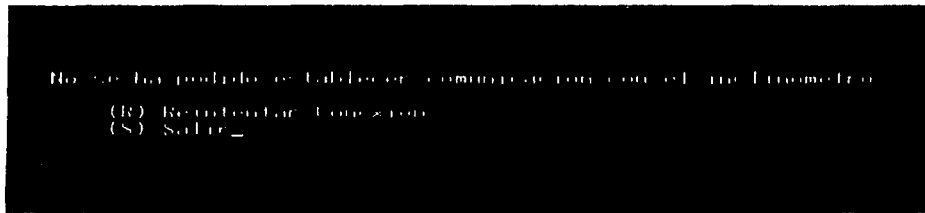


Figura 5.4. Mensaje de error en la conexión de la brújula.

Si ya ha sido conectada la brújula electrónica en el puerto serie y la caja de sensores inerciales al puerto paralelo y han sido polarizados en forma adecuada; la pantalla que se nos muestra es la siguiente:

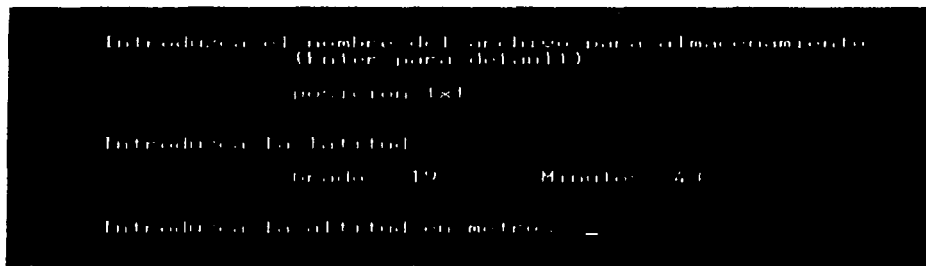


Figura 5.5. Nombre del archivo de datos de orientación e información sobre la localización geográfica de la zona de vuelo.

La pantalla solicita el nombre que debe asignarse al archivo que almacenará los datos de los tres ejes de orientación, asociados a cada imagen, al momento de efectuar la toma. También pide información para calcular la gravedad local, la que posteriormente será utilizada en los cálculos de orientación por el sistema inercial de determinación de la vertical local.

Inmediatamente después de proporcionada la información anterior, la plataforma realiza una calibración inicial en busca de la horizontal local. En esta parte también se estableció que la plataforma se orientara a 180° . Esta calibración permite conocer al programa los voltajes de referencia de los potenciómetros en el momento en que la plataforma se encuentra completamente en la horizontal y orientada a 180° , para ocupar estos valores posteriormente en el control de estabilización en tiempo real.

Ahora el siguiente paso es que el programa lea los valores de voltaje correspondientes a los límites que determinarán el cambio de línea. Como hemos mencionado, se tomará en cuenta que cuando el avión se desvía tanto en su eje de rotación como de guiñada, consideraremos que se está efectuando una vuelta para tomar la siguiente línea de vuelo. Para obtener esta información el programa mueve a la plataforma en el eje de rotación a -15° y a $+15^{\circ}$ y lee los valores correspondientes, donde se garantiza que el avión está girando y por lo tanto está cambiando de línea. Lo mismo hace con el eje Z, el cual es movido hacia -30° y a $+30^{\circ}$. Con base en los valores de voltaje obtenidos al hacer girar los ejes mencionados, se establece y conoce el umbral que deberán sobrepasar cualquiera de ellos durante el levantamiento para determinar, que existe un cambio de línea.

Después de tener estos valores de voltaje, nuevamente se calibra la plataforma con respecto a la horizontal local.

Con esto termina la primera fase del programa, en la cual se comprobó que hubiera comunicación con los sensores involucrados en el control de la plataforma, se capturaron los valores que permiten determinar la gravedad local con la que trabajará el algoritmo del sistema inercial, se estableció el nombre del archivo de datos de orientación y por último, se determinaron los valores de voltaje cuando la plataforma está completamente horizontal y aquellos que servirán de referencia al momento de hacer un cambio de línea.

5.2 Control de la plataforma y almacenamiento de la información

En esta segunda fase el programa preguntará al operador la dirección de vuelo que el avión seguirá al tomar la primera línea. Con esta información, se determinará cuando se está efectuando un cambio de línea y cual será la dirección de la siguiente, sin necesidad de que intervenga el operario.

En la pantalla es desplegado el siguiente menú:

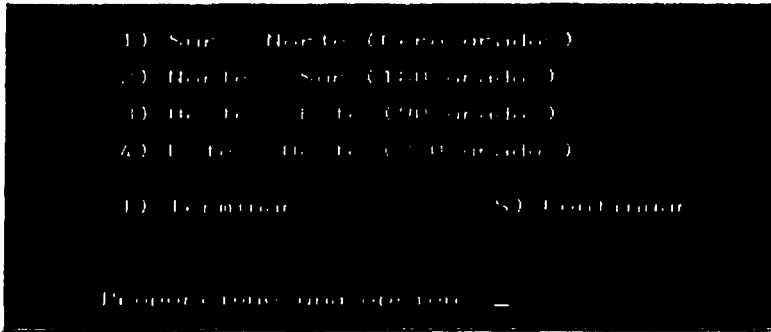


Figura 5.6. Determinación de la dirección de la primera línea de vuelo.

En esta parte del menú se puede indicar que se tome la dirección por designación (default), que en un principio sería Norte - Sur (180 grados), se puede terminar el programa o se le puede indicar cual es la nueva línea de vuelo que se seguirá.

TESIS CON
 FALLA DE ORIGEN

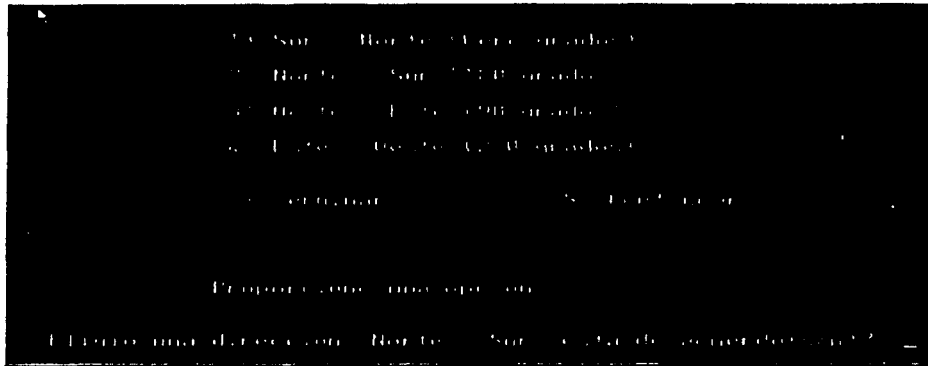


Figura 5.7. Opción de confirmar o de cambiar la línea seleccionada.

Se presenta la opción de corregir en caso de algún error en la selección de la dirección de vuelo. También muestra entre paréntesis una cantidad en grados por cada línea, esto es para comodidad del piloto y el navegante. La orientación que tiene cada punto cardinal es la siguiente:

- Norte → Cero grados
- Este → 90 grados
- Sur → 180 grados
- Oeste → 270 grados

La convención que se maneja en este proyecto en la orientación, y de acuerdo con la pantalla anterior, es la siguiente:

Si se selecciona 1) Sur – Norte, la plataforma se orientará al Norte, si se selecciona 2) Norte –Sur la plataforma se orientará hacia el Sur, y lo mismo pasa para los siguientes dos casos.

Una vez que nos encontramos cerca de la zona de estudio, la plataforma puede comenzar a corregir su orientación conforme a la primera línea seleccionada, durante este periodo se muestra en pantalla la corrección que se está realizando, en donde también se da un mensaje que permite cancelar esta operación:



Figura 5.8. Despliegue de la orientación de la plataforma, antes de tomar la primera línea.

TESIS CON
 FALLA DE ORIGEN

Este mensaje lo muestra el programa hasta alcanzar la dirección adecuada. Hasta este momento se ha llegado al punto en que la plataforma empezará a trabajar de forma autónoma, es decir, ya no requerirá ningún dato para poder llevar a cabo el control, a pesar de hacer cambios de línea; siempre y cuando éstos sean consecutivamente de 180 grados.

El programa; con la información proporcionada por los sensores inerciales, los valores de voltaje de los potenciómetros de cada motor y las señales que entrega la brújula electrónica, determinará de forma automática cuando la plataforma necesite ser corregida. El circuito de lazo cerrado que se forma con todos estos componentes permitirá retroalimentar al programa, con lo cual se generan valores que determinan el posible error de la posición actual de la plataforma, de esta manera se conoce cuanto y hacia donde debe girar ésta, para corregir su orientación.

Estas correcciones son en tiempo real, ya que el sistema de componentes inerciales detectarán el posible movimiento que tenga el avión en cualquiera de sus tres ejes (rotación, cabeceo y guiñada), esta información será enviada al programa que ejecuta el algoritmo de orientación en dos ejes y efectuará la corrección en caso de que ocurra alguna desviación en la plataforma. El eje Z es manejado de manera independiente y directa, a través de la lectura de la brújula electrónica.

En el capítulo 3 se ha explicado como es que trabaja un sistema de navegación inercial, pero ahora se harán algunas precisiones acerca de la forma real en que el sistema inercial para la determinación de la vertical, se aplica en este proyecto y de cómo se lleva a cabo la orientación de la plataforma y de la función que operativamente juega este sistema inercial en el control de la misma.

Es un hecho que si la plataforma no contara con el sistema inercial, sería imposible detectar los movimientos del avión, esto debido a que la plataforma inicialmente seguirá al avión en su traslado, no importando en que eje se presente el movimiento. La inercia de la plataforma tendrá la tendencia de seguir los movimientos de la nave y posiblemente se detecte algún ligero cambio de voltaje en cualquiera de los potenciómetros que tiene cada motor colocado en su eje, pero esta no es información suficiente para que la plataforma se corrija con respecto al plano horizontal de la tierra,

ya que no existe una referencia absoluta ni información que permita determinar cual es esa horizontal, los voltajes de los potenciómetros por si solos, no la dan.

Así es que, los giróscopos y acelerómetros son los elementos que suplen esta deficiencia de información; importante en cuanto a los movimientos del avión, permitiendo tener una referencia inercialmente fija con respecto a la horizontal local, que ha sido tomada al momento de la calibración inicial. Esto permite corregir de forma eficaz y en tiempo real la rotación y el cabeceo.

Los sensores inerciales detectan el mínimo movimiento del avión en cualquiera de sus ejes (en el intervalo de un grado o menos), por lo cual es ya posible determinar el defasamiento que se ha producido por dicho movimiento con respecto al plano horizontal de la tierra. Y es así que la manipulación de los motores no depende directamente del valor de voltaje de los potenciómetros, sino que intervienen de forma esencial los valores de orientación determinados a partir de los sensores inerciales.

Por otro lado, la brújula electrónica también proporciona una medida absoluta de la dirección de vuelo, por lo que su utilización es directa. La corrección es diferente a lo que se realiza con los ejes X,Y, ya que la información de la orientación, para este caso, viene del magnetómetro de la brújula electrónica. Éste está programado para entregar información a intervalos de 0.68 segundos entre cada dato, lo cual podría en primera instancia hacer pensar que limita enormemente la velocidad de respuesta en el proceso de corrección de la plataforma, pero no es así. El programa se diseñó de tal forma, que la corrección en el eje Z no fuera inmediata, es decir, que no corrigiera completamente su orientación en una ejecución completa del algoritmo de control, si no que fuera corrigiendo este eje en intervalos aproximadamente de 1.5° por cada ejecución del algoritmo.

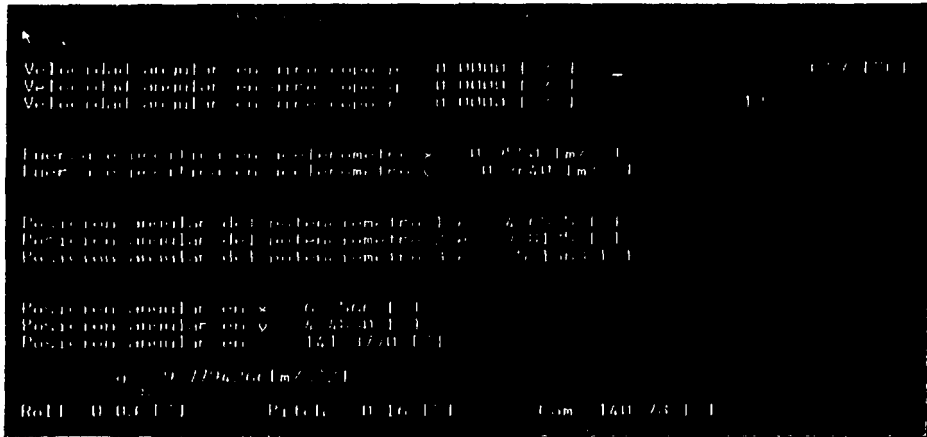
El haber efectuado este procedimiento permitió conservar una muy buena velocidad en la respuesta de corrección de los tres ejes de la plataforma, como veremos más adelante. Esto se pudo hacer, atendiendo a la experiencia en vuelo de que el eje Z no varía de manera importante.

En esta fase el programa de control también almacena información crítica para el levantamiento fotográfico que se está realizando. Uno de los objetivos de este proyecto es contar con la información de la orientación de la plataforma donde van colocadas las cámaras fotográficas, para poder efectuar su estabilización cuando existan desviaciones en sus ejes; con respecto al marco de referencia inercial. Esto es debido, por una parte, a que es deseable obtener las imágenes cuando las cámaras apuntan directamente hacia el nadir, pero también se necesita conocer los datos de orientación de cada fotografía en el instante en que fueron tomadas. Esto con la finalidad de llevar a cabo la corrección geométrica de cada imagen y poder formar los mosaicos de la zona de estudio, de una manera más sencilla.

Para tal propósito el programa de control cuenta con una función de almacenamiento de los datos de orientación de cada toma fotográfica, la cual se activa cada vez que se recibe un pulso externo por el pin número 15 del puerto paralelo, proveniente del intervalómetro; que es manipulado a su vez por el sistema de monitoreo, el cual determina el instante y la frecuencia con que deben ser hechas las tomas durante un levantamiento aéreo.

Los datos de orientación de cada toma fotográfica, eran difíciles de almacenar en un principio en el instante preciso de su ocurrencia, debido a que todos los procesos involucrados en el programa, hacían perder esta información por la gran cantidad de

funciones y ciclos que se están ejecutando al momento de corregir la plataforma. Este problema fue reparado, haciendo que ahora el programa capture con precisión los valores de la orientación al momento de una toma fotográfica, de manera sincronizada con la señal de disparo. La comprobación de que esto ocurre así, la podemos ver más claramente cuando se hace la comparación entre los intervalos de tiempo programados en el intervalómetro y los valores devueltos por el programa de control durante la adquisición de los ángulos.



TESIS CON
FALLA DE ORIGEN

Figura 5.9. Pantalla de despliegue de datos en tiempo real.

Durante la etapa de control de estabilización de la plataforma, la pantalla de despliegue del sistema muestra en tiempo real los valores que presenta cada sensor, la orientación determinada por el algoritmo encargado de ello, y los valores de: la gravedad local, la temperatura y los ángulos de rotación, cabeceo y guiñada, así como el número de tomas fotográficas realizadas. La pantalla se muestra en la figura 5.9.

5.3 Cambios de línea de vuelo

A lo largo de todo este capítulo se han mencionado los cambios de línea que debe realizar el avión para efectuar la campaña aérea. El programa cuenta con un módulo que determina de forma automática los posibles cambios de línea que ocurren para evitar de esta forma que el operador tenga que introducir información sobre la nueva línea de vuelo. Los cambios de línea son importantes especialmente por la corrección en la dirección de vuelo que se hace en el eje Z. El avión no se puede dirigir al Norte mientras la plataforma se está orientando hacia el Sur, por lo que el detectar adecuadamente los cambios de dirección al pasar de una línea a otra, es primordial.

Como ya se mencionó, los cambios de línea automáticos se realizarán siempre y cuando estos correspondan a una variación de 180 grados con respecto de la línea de vuelo anterior.

Los cambios de línea automáticos posibles programados en este proyecto son:

Tabla 5.1.

| Línea de vuelo actual | Orientación en Grados | Nueva línea de vuelo | Orientación en Grados |
|-----------------------|-----------------------|----------------------|-----------------------|
| Sur – Norte | 0 | Norte – Sur | 180 |
| Norte – Sur | 180 | Sur – Norte | 0 |
| Oeste – Este | 90 | Este – Oeste | 270 |
| Este - Oeste | 270 | Oeste – Este | 90 |

Como se ve en todos los casos el cambio a una nueva línea de vuelo corresponderá a un cambio en 180 grados de la orientación que se tenía. ¿Y qué pasa cuando no se realiza un cambio de línea de 180 grados?

El programa detectará el cambio de línea y esperará a retomar la nueva línea defasada de la anterior 180 grados, la plataforma no se moverá durante este tiempo ni tratará de corregir sus ejes, ya que no le es posible encontrar la línea que le corresponde, por lo que en todo momento se protege a los circuitos y motores para que no se vean forzados a realizar movimientos o acciones que estén fuera de sus intervalos de operación. En este momento se presenta en la pantalla la orientación actual de la brújula y le permite al operador ir al menú principal, para corregir el nuevo rumbo de la línea o terminar la ejecución del programa en caso de que ya se haya terminado con el levantamiento.

TESIS CON
 FALLA DE ORIGEN

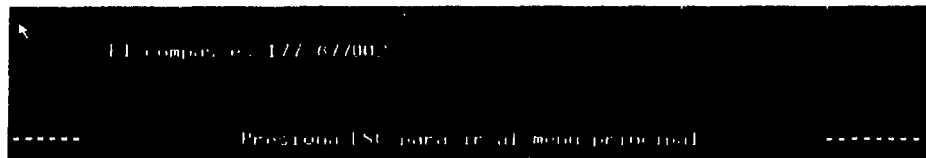


Figura 5.10. Ante un cambio de línea hacia una dirección no esperada, el programa indica la orientación y espera alguna instrucción, omitiendo toda compensación.

5.4 Programación

En esta sección se hace un esbozo del proceso de programación del sistema aquí presentado, así como los requisitos que éste impone para la selección del lenguaje de programación. La velocidad de procesamiento y la precisión en los resultados dependen en gran medida del lenguaje de programación que se utilice para llevar a cabo la adquisición de datos, el proceso de orientación y control del sistema. Una buena elección del lenguaje de programación depende de las características de los algoritmos del proceso del sistema, C es una alternativa excelente para este propósito.

5.4.1 Características del lenguaje C.

El desarrollo del algoritmo de control exigió el hacer prototipos de programación de fácil estructura los cuales pudieran ser modificados en caso de alguna corrección. Para ello se requirió trabajar con una herramienta flexible y amigable con la capacidad

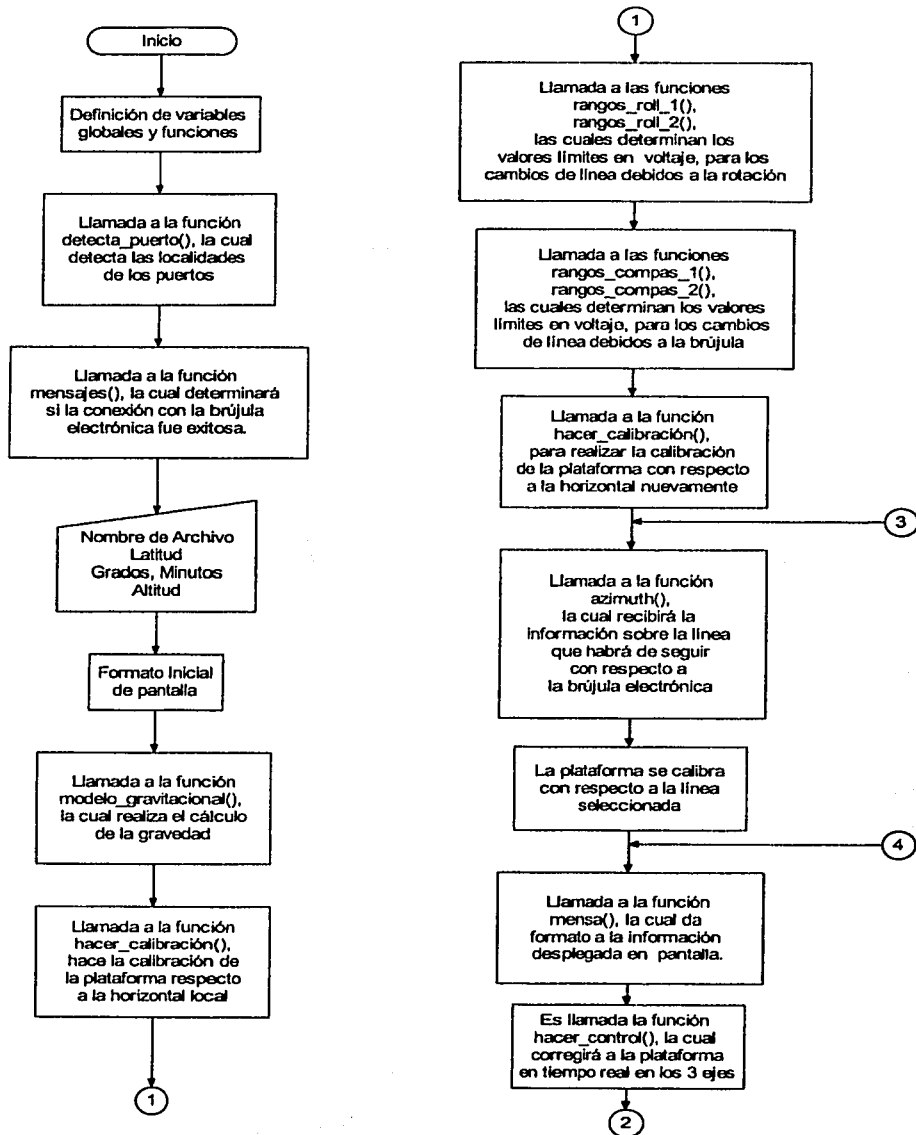
de cambiar fácilmente los detalles de funcionamiento. El tipo de lenguaje que se seleccionó para llevar a cabo el proceso de orientación, y en general todo el proceso de adquisición de datos de sensores y transmisión de las señales de control, es el lenguaje C; que siendo un lenguaje de nivel intermedio, permite tanto el uso de funciones pre-construidas, como la manipulación directa de bits y bytes. Esto permite una gran facilidad en el manejo de operaciones y comunicación a periféricos.

C es un lenguaje modular poderoso y flexible, usado en proyectos tan diversos como sistemas operativos, procesadores de palabras, gráficos, hojas de cálculo y hasta compiladores para otros lenguajes. Es un lenguaje transportable, es decir, que un programa C escrito para un sistema de computadora puede ser compilado y ejecutado en otro sistema con pocas o ninguna modificación. El código de C puede ser escrito en rutinas llamadas funciones. Estas funciones pueden ser reutilizadas en otras aplicaciones o programas. Pasando información a las funciones, se puede crear código útil y reutilizable.

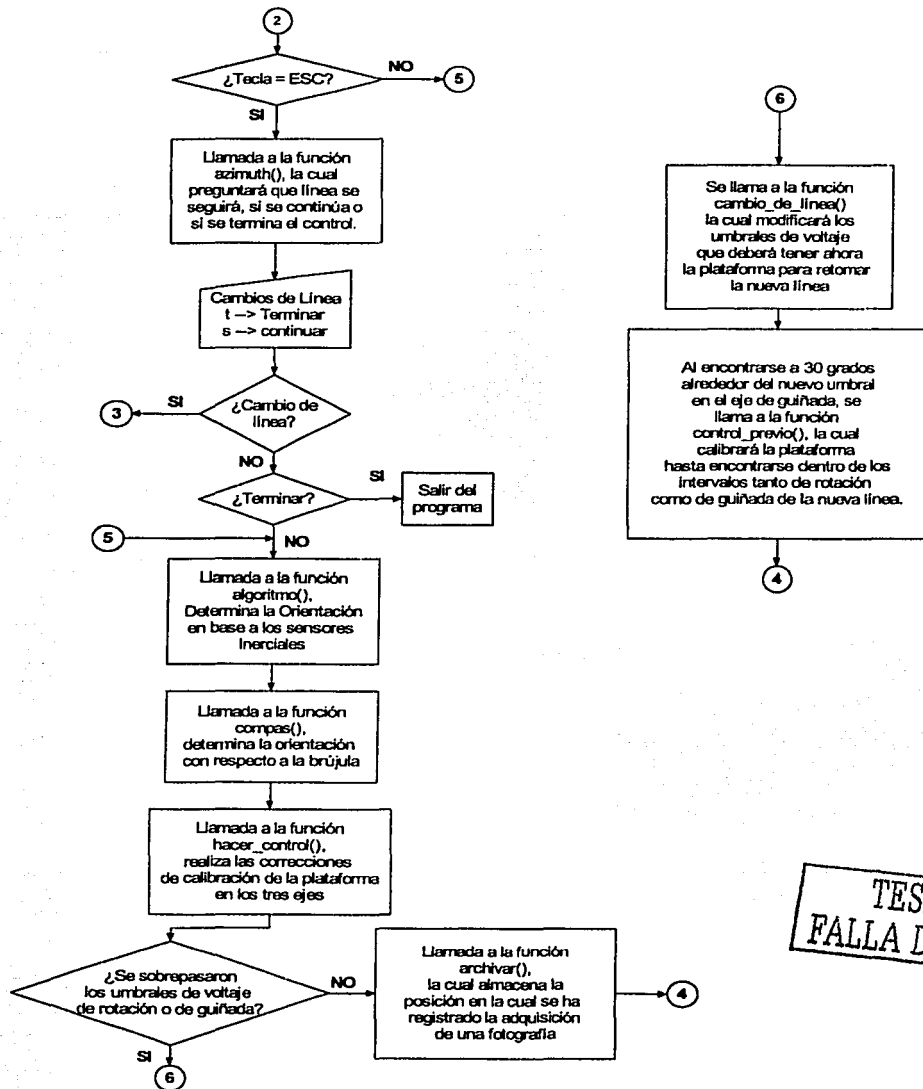
5.5 Algoritmo de procesamiento.

A continuación se presenta un diagrama de flujo del algoritmo que se elaboró para los procesos de orientación, adquisición de datos de sensores, transmisión de las señales de control, almacenamiento de los datos de orientación y cambios de línea de vuelo.

En el apéndice B se encuentra el listado completo del programa.



TESIS CON FALLA DE ORIGEN



TESIS CON FALLA DE ORIGEN

Figura 5.11. Algoritmo programado para el control de la plataforma aérea.

5.6 Resultados

El sistema de control automático para la plataforma realiza sus ciclos de control con una frecuencia de 8.5 ciclos cada segundo.

Las curvas presentadas en la siguiente figura prueban que la plataforma se estabiliza por medio del programa de control automático dentro del intervalo de ± 1.0 grados, en los ejes de rotación, de cabeceo y guiñada.

En las siguientes gráficas se muestra el comportamiento de la plataforma ante una perturbación introducida de manera intencional aproximadamente de 10 grados en los ejes X,Y,Z.

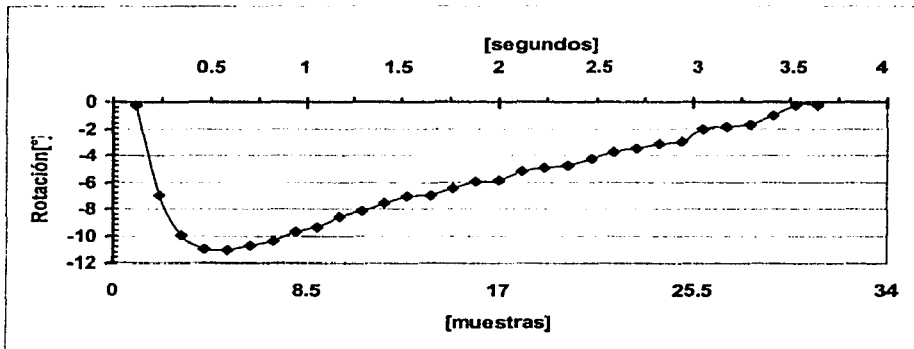


Figura 5.12. Respuesta de la plataforma en el eje X (rotación), ante una perturbación.

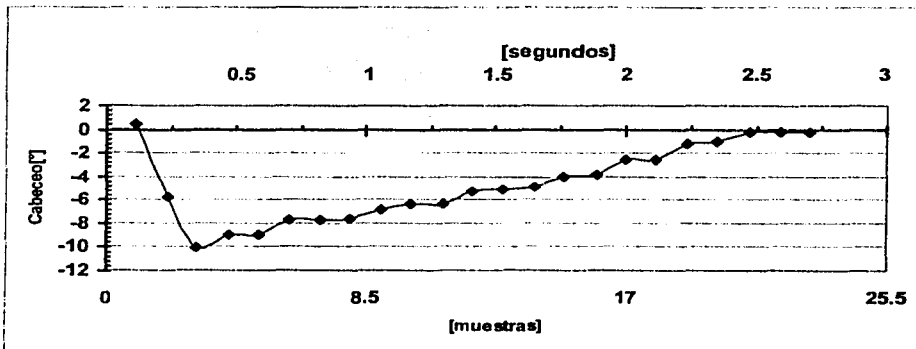


Figura 5.13. Respuesta de la plataforma en el eje Y (cabeceo), a una perturbación intencional.

TESIS CON
 FALLA DE ORIGEN

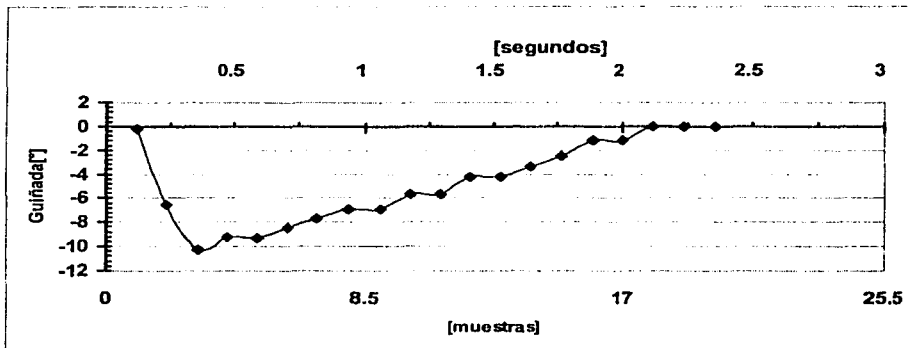


Figura 5.14. Respuesta en el eje Z (guñada), ante una desviación.

El tiempo en que la plataforma alcanzó la estabilización con respecto a la horizontal local fue en 3.25 segundos en el eje X, en el eje Y fue de 2.2 segundos y de 2.1 segundos para el eje Z.

El comportamiento general de los tres ejes puede ser observado de mejor forma en la siguiente gráfica donde se muestra el tiempo, la cantidad de muestras y los grados de desviación de los ejes X, Y, y Z al mismo tiempo:

TESIS CON
FALLA DE ORIGEN

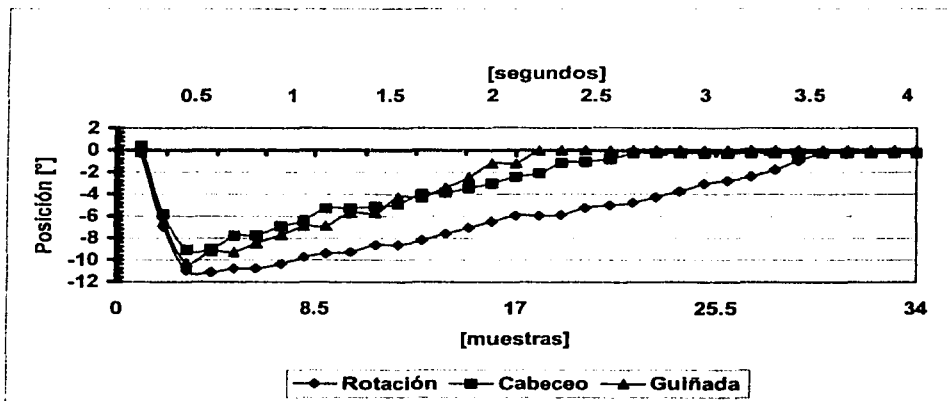


Figura 5.15. Respuesta en el eje X, Y y Z, ante una desviación.

Para comprobar el correcto almacenamiento de los datos de orientación, en cada toma, en el archivo se guardaron los siguientes datos, mismos que permitieron crear las siguientes gráficas que también sirvieron para evaluar el comportamiento de la brújula.

En la siguiente gráfica se ve el comportamiento que tiene cada dato que es registrado al momento de la captura de una imagen fotográfica, la cual anteriormente no seguía con precisión los intervalos de tiempo programados en el sistema de monitoreo o en el intervalómetro. Se ve claramente que ningún dato se extravía o se deja de almacenar.

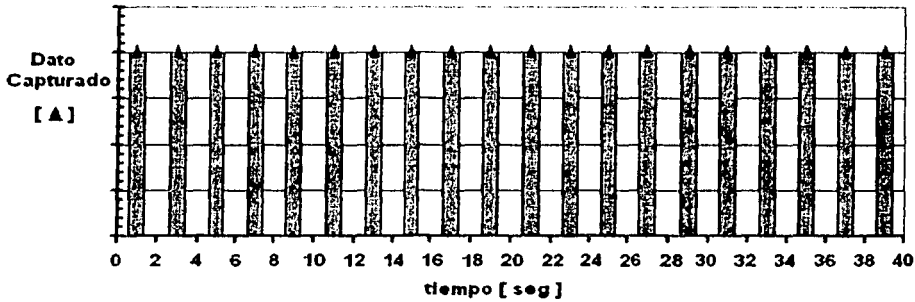


Figura 5.16. Sincronización entre el disparo de la cámara y el almacenamiento de datos.

TESIS CON
FALLA DE ORIGEN

CAPÍTULO

6

CONCLUSIONES Y RECOMENDACIONES

Del desarrollo del presente trabajo de tesis podemos desprender las siguientes conclusiones:

- ❖ Se cumplió el objetivo principal del trabajo de tesis, el cual es adquirir imágenes aéreas mediante cámaras digitales de formato reducido.
- ❖ Se han logrado avances significativos en el desarrollo de un sistema de adquisición de imágenes aéreas digitales de pequeño formato.
- ❖ Se llevó a cabo con éxito, la transición entre prototipos y equipos operativos, con excepción de la plataforma porta-cámaras que se encuentra en proceso de construcción.
- ❖ Fue modificado de manera importante el programa que controla la orientación de la plataforma, poniendo en funcionamiento el control del eje Z y añadiendo el procedimiento de control automático de estabilización al efectuarse el cambio en las líneas de vuelo.
- ❖ Se han efectuado pruebas de funcionamiento en tierra con este equipo, con resultados satisfactorios.

Para dar continuidad al presente trabajo, se recomienda:

- ❖ Hacer más amigable la interfaz con el usuario en el programa de control de la plataforma.
- ❖ Efectuar pruebas de funcionamiento con la plataforma definitiva, tanto en tierra como en vuelo para llevar a cabo los ajustes finales del equipo.

APÉNDICE A ENCODERS

Introducción

El encoder o codificador, es un transductor rotativo que transforma un movimiento angular en una serie de impulsos digitales. Éstos pueden ser utilizados para controlar los desplazamientos de tipo angular o de tipo lineal si se asocian por ejemplo, a cremalleras. Las señales eléctricas de rotación pueden ser procesadas mediante controles numéricos (CNC), contadores lógicos programables (PLC), sistemas de control, etc. Las aplicaciones principales de estos transductores están en las máquinas herramientas, en los robots, en los sistemas de motores y en los aparatos de medición y de control.

En los encoders (figura A.1), la detección del movimiento angular se efectúa con base en una exploración fotoeléctrica. El funcionamiento del sistema de lectura se basa en la rotación de un disco graduado con un reticulado radial formado por líneas opacas, alternadas con espacios transparentes. Este conjunto está iluminado de modo perpendicular por una fuente de rayos infrarrojos. El disco proyecta de este modo su imagen sobre la superficie de varios receptores también enmascarados por otro reticulado, que tiene el mismo paso del anterior, llamado colimador. Los receptores tienen la tarea de detectar las variaciones de luz que se producen con el desplazamiento del disco, convirtiéndolas en las correspondientes variaciones eléctricas. La señal eléctrica detectada, para generar impulsos correctamente escuadrados y sin interferencias, debe ser procesada electrónicamente.

Para incrementar la calidad y estabilidad de las señales, la lectura se efectúa generalmente de manera diferencial, comparando dos señales casi idénticas, pero desfasados 180 grados. Se hace la diferencia de las dos señales, eliminando las interferencias "de modo común" porque están superpuestas de igual manera en toda forma de onda.

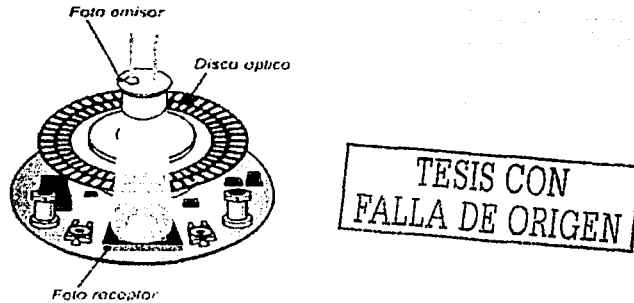
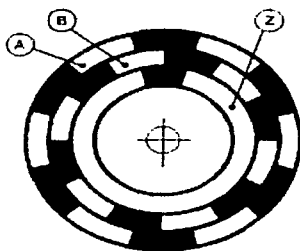


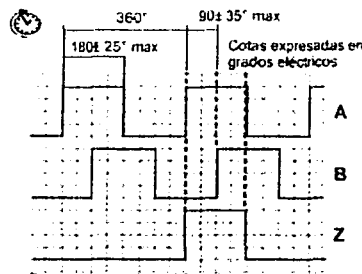
Figura A.1. Componentes Básicos de un encoder.

Encoders Incrementales

El encoder incremental proporciona normalmente dos formas de ondas cuadradas desfasadas entre si 90 grados, los cuales por lo general forman el "canal A" y "canal B". Con la lectura de un solo canal se dispone de la velocidad de rotación, mientras que si se capta la señal "B" es posible discriminar el sentido de rotación en base a la secuencia de datos que producen ambas señales. Está disponible otra señal llamada "canal Z" o cero del cero, del eje del encoder. Esta señal se presenta bajo la forma de impulso cuadrado con fase y amplitud centrada en el canal A. (figura A.2a), que proporciona la posición absoluta del cero, del eje del encoder. Esta señal se presenta bajo la forma de impulso cuadrado con fase y amplitud centrada en el canal A. (figura A.2b).



(a)



(b)

Figura A.2. Representación de las señales incrementales A,B,Z en: a) un disco óptico y b) gráficamente.

La precisión de un encoder incremental depende de factores mecánicos y eléctricos entre los que tenemos: el error en la división de la retícula, la excentricidad del disco, la de los rodamientos, el error introducido por la electrónica de la lectura, e imprecisiones de tipo óptico. La unidad de medida para definir la precisión de un encoder es el grado eléctrico, éste determina la división de un impulso generado por el encoder. Veamos, los 360 grados eléctricos corresponden a la rotación mecánica del eje, necesaria para hacer que se realice un ciclo o impulso completo de la señal de salida. Para saber a cuántos grados mecánicos corresponden 360 grados eléctricos es suficiente aplicar la fórmula siguiente:

$$360^\circ \text{ _eléctricos} = \frac{360^\circ \text{ _mecánicos}}{n^\circ \text{ _impulsos / giro}}$$

El error de división en un encoder, está dado por el máximo desplazamiento expresado en grados eléctricos, de dos frentes de onda consecutivos, Este error existe en cualquier encoder y se debe a los factores antes citados.

Encoders Absolutos

El principio de funcionamiento de un encoder absoluto es muy similar al de uno incremental en el cual un disco rotatorio, con zonas transparentes y opacas, interrumpe un haz luminoso adquirido por unos fotorreceptores, por lo tanto, ellos transforman los impulsos luminosos en impulsos eléctricos que se tratan y transmiten por la parte electrónica de salida.

La codificación absoluta

Con respecto a los incrementales, los encoders absolutos muestran importantes diferencias desde el punto de vista funcional. Mientras que en los incrementales la posición está determinada por el cómputo del número de impulsos con respecto a la marca de cero, en los encoders absolutos la posición queda determinada mediante la lectura del código de salida, el cual es único para cada una de las posiciones dentro de la vuelta. Por consiguiente los encoders absolutos no pierden la posición real cuando se corta la alimentación (incluso en el caso de desplazamientos), hasta un nuevo encendido (gracias a una codificación directa en el disco), la posición está actualizada y disponible sin tener que efectuar, como en el caso de los encoders incrementales la búsqueda del punto de cero.

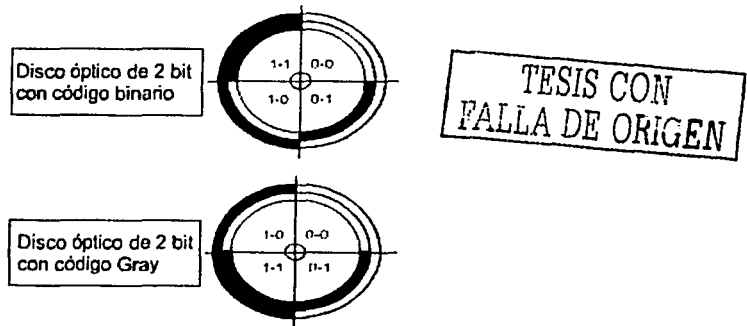


Figura A.3. Representación del código binario y Gray en los discos de un encoder.

Analicemos ahora el código de salida que se deberá utilizar para definir la posición absoluta. La elección más obvia es la del código binario, porque fácilmente puede ser manipulado por los dispositivos de control externos para la lectura de la posición, sin tener que efectuar operaciones de conversión. En vista de que el código se toma directamente desde el disco (que se encuentra en rotación) la sincronización y la captación de la posición en el momento de la variación entre un código y otro se vuelve muy problemática (figura A.3). Por ejemplo si

tomamos dos códigos binarios consecutivos como 7(0111) 8(1000), se nota que todos los bits del código sufren un cambio de estado, una lectura efectuada en el momento de la transición podría resultar completamente errónea porque es imposible pensar que las variaciones sean instantáneas y que se produzcan todas en el mismo momento. Debido a este problema se utiliza una variante del código binario: el código Gray (tabla A.1), el cual tiene la particularidad que al pasar entre dos códigos consecutivos (o desde el último código al primero), solo un bit cambia su estado.

Tabla A.1 Código Gray.

| DECIMAL | BINARIO | GRAY |
|---------|---------|------|
| 0 | 0000 | 0000 |
| 1 | 0001 | 0001 |
| 2 | 0010 | 0011 |
| 3 | 0011 | 0010 |
| 4 | 0100 | 0110 |
| 5 | 0101 | 0111 |
| 6 | 0110 | 0101 |
| 7 | 0111 | 0100 |
| 8 | 1000 | 1100 |
| 9 | 1001 | 1101 |
| 10 | 1010 | 1111 |
| 11 | 1011 | 1110 |
| 12 | 1100 | 1010 |
| 13 | 1101 | 1011 |
| 14 | 1110 | 1001 |
| 15 | 1111 | 1000 |

TESIS CON
 FALLA DE ORIGEN

Código Gray quebrado o Código Gray con exceso N

Cuando el número de posiciones no es una potencia de 2, la propiedad de cambiar un solo bit se pierde al pasar del último valor al primero y viceversa. Por ejemplo, deseamos realizar un encoder absoluto con 12 posiciones/giro (Tabla A.2). Se nota que al pasar de la posición 11 a 0, el cambio de estado simultáneo de 3 bits puede producir errores de

Tabla A.2. Ejemplo de codificación.

| POSITION | GRAY |
|----------|------|
| 0 | 0000 |
| 1 | 0001 |
| 2 | 0011 |
| 3 | 0010 |
| 4 | 0110 |
| 5 | 0111 |
| 6 | 0101 |
| 7 | 0100 |
| 8 | 1100 |
| 9 | 1101 |
| 10 | 1111 |
| 11 | 1110 |
| 0 | 0000 |

Error

Tabla A.3. Resultado de la codificación.

| POSITION | GRAY |
|----------|------|
| 0 | 0011 |
| 1 | 0010 |
| 2 | 0110 |
| 3 | 0111 |
| 4 | 0101 |
| 5 | 0100 |
| 6 | 1100 |
| 7 | 1101 |
| 8 | 1111 |
| 9 | 1110 |
| 10 | 1010 |
| 11 | 1011 |

TESIS CON
 FALLA DE ORIGEN

lectura y esto, como se ha visto anteriormente no es aceptable. Para no perder la característica de la conmutación de un solo bit, se utiliza el código Gray quebrado o bien con exceso N, haciendo corresponder la posición 0 al valor N del código Gray correspondiente. Donde N es el número que, substrayéndolo al código Gray convertido a número binario, da el valor exacto de la posición. El cálculo del número N se efectúa según la fórmula:

$$N = \frac{2^n - IMP}{2}$$

Donde

IMP es el número de impulsos/ giro (solo impulsos pares)

2^n es el número de impulsos múltiplo potencia de 2, inmediatamente superior al valor de posiciones giro que necesitamos.

En este caso la codificación será:

$$N = \frac{2^4 - IMP}{2} = \frac{16 - 12}{2} = 2$$

por lo tanto ahora la posición 0 de esta nueva codificación (Tabla A.3), será la posición 2 del código Gray de la tabla A.1.

Encoder Incremental HEDS-5640

En esta sección vamos a tratar un caso particular de un encoder, el HEDS-5640 de Hewlett-Packard del cual se tienen algunas piezas en el Laboratorio. El fabricante enfatiza la confiabilidad, alta resolución y el fácil ensamblado. El encoder contiene una fuente de luz (LED), un lente que permite la dispersión de la misma, un circuito integrado con su respectiva circuitería de detectores y salidas, una rueda codificada, la cual gira entre el emisor y el circuito integrado, que hace las funciones de detector.

Este encoder tiene tres salidas digitales, dos de las cuales son ondas cuadradas que se encuentran en cuadratura, la tercer salida digital conocida como índice (index) permite detectar un giro completo de la rueda codificada, que tiene un resolución entre 96 y 1024 cuentas por revolución.

Teoría de Operación del HEDS-5640

Como se ve en la figura A.4, el encoder tiene un LED como fuente luminosa. La luz es enfocada de forma paralela por un lente de policarbonato hacia el circuito integrado que actúa como detector. Este consiste de varios conjuntos de fotodetectores y circuitos de procesamiento, necesarios para producir las señales digitales de salida.

La rueda codificada gira a través del emisor y del detector, lo que causa que la luz sea interrumpida por las barras marcadas en la barra codificada. Los fotodiodos que detectan estas interrupciones, están colocados de tal forma, que corresponden a una sección de la rueda codificada. Estos detectores tienen un espaciamiento entre ellos, lo que provoca que en un periodo corto, un par de detectores corresponda a un periodo de sombra, y en un par adyacente, un periodo de luz. La señal del canal "A" se encontrará en cuadratura con la señal del canal "B" (90 grados fuera de fase). La señal de salida del canal "I" es enviada al circuito de procesamiento, generando un pulso cada que se produce una vuelta completa en la rueda codificada. El pulso de esta señal coincide con los estados bajos de los canales "A" y "B".

Diagrama de Bloques

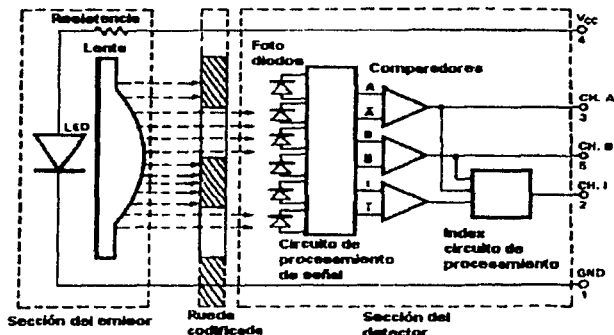


Figura A.4. Diagrama de bloques del encoder HEDS-5640

Potenciómetros e Encoders

Como ya se ha mencionado, para el desarrollo de este proyecto fue necesario llevar a cabo una revisión de los diseños y prototipos existentes y proponer o descartar el efectuar los cambios y/o adaptaciones pertinentes en todos los módulos del sistema.

Entre las adaptaciones que se propusieron fue la de reemplazar los potenciómetros utilizados para referenciar la posición de los tres ejes de la plataforma portacámaras, por encoders. Esta tarea requirió de un estudio, así como el realizar las pruebas necesarias para definir si eran convenientes para el desarrollo del proyecto.

Antes de iniciar con las ventajas y desventajas de los encoders para este trabajo, hay que recordar que uno de los objetivos era el crear un sistema de bajo costo así como de excelente funcionalidad y mínimo espacio, por lo que al momento de realizar el análisis debían de tenerse presentes estos objetivos.

Los potenciómetros son utilizados para realimentar la posición de los motores, en el ciclo de control de tipo on/off que se maneja, los cuales detectan el error en la posición actual, estos se encuentran acoplados a cada eje de movimiento de la plataforma, nos proporcionan una señal eléctrica que se utiliza para comparar la señal de salida con la señal de referencia inicial, alcanzada esta última al momento de la primer estabilización, cada potenciómetro se encuentra en el camino de realimentación del sistema de lazo cerrado.

Es elemento importante en la compensación final que se hace en la plataforma, por la comparación que se realiza entre los valores que entrega la brújula electrónica (roll y pitch), y la posición angular que se obtiene a partir de los valores obtenidos de los potenciómetros.

A partir de esto podemos empezar a comparar y separar los beneficios de los encoders y potenciómetros.

En el prototipo original se maneja un convertidor analógico/digital ADC0838 de 8 bits, de aproximaciones sucesivas con una salida serie, el cual incluye un multiplexor de 8 canales para las entradas, el cual se controla mediante una entrada serie, la cual recibe una palabra

de control de 5 bits proveniente de la computadora para hacer la medición del canal que se requiera. Este CAD recibe los voltajes analógicos de los giróscopos, acelerómetros y potenciómetros, la sensibilidad de este convertidor analógico/digital garantiza, utilizando como voltaje de referencia en el convertidor 5 V, 256 niveles en el rango de 0.0 V a 5.0 V, una sensibilidad considerable siendo que:

$$\text{Sensibilidad} = \frac{V_{ref}}{256}$$

Con respecto a los encoders y en particular el HEDS-5640 (encoder incremental), ya que es con el encoder que se cuenta podemos observar lo siguiente:

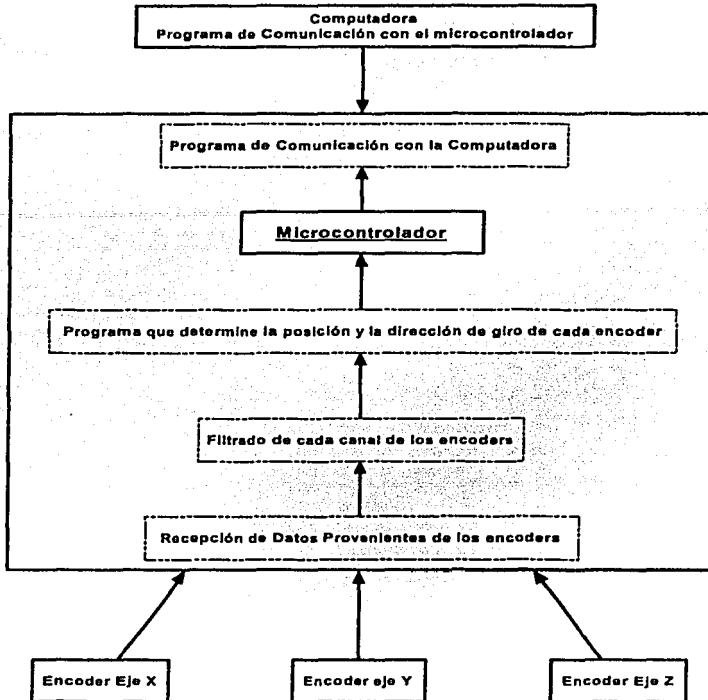
El encoder devuelve salidas TTL, que a su vez acoplados al circuito HCTL-2016 (detector de cuadratura/contador), da gran inmunidad al ruido, puede operar en 8, 12 o 16 bits, la salida paralela la presenta en 8 bits, la cual debe ser acoplada a un microcontrolador para hacer la rutina de control de dicho circuito.

En nuestro caso utilizaríamos un encoder (figura 4.8) por cada eje de movimiento esto es decir tres encoders, los cuales se pueden manipular tanto en su adquisición, filtrado y compensación de señales con un microcontrolador.

Los requisitos que requiere el desarrollo de un sistema de sensado y filtrado digital se ajustan a los recursos que ofrece el MC68HC11F1, las funciones que permite llevar a cabo un sistema implementado con un microcontrolador son muy variadas y flexibles, esto hace muy conveniente su utilización. Permite la posibilidad de manejar datos en puertos de entrada y salida, la ampliación de memoria en los casos que se requiera, velocidades de operación del orden de Mhz y variedad de operaciones aritméticas que ya incluye en su conjunto.

El microcontrolador obtendrá la información de los 3 canales que proporciona cada uno de los encoders (canal "A", canal "B", canal "I"), a partir de aquí mediante el código Gray con exceso a N realizará el conteo que permitirá obtener la posición de la plataforma en cada uno de sus ejes, a su vez determinará el sentido de giro y determinará la cantidad de veces que se alcance una vuelta completa en el disco codificado del encoder, para finalmente enviar esta información a la computadora, y así seguir con el proceso de estabilización.

De igual forma en la computadora es necesario desarrollar el programa adecuado que se comunicará con el microcontrolador, para recibir la información de este.



TESIS CON
 FALTA DE ORIGEN

Figura 4.8 Diagrama de bloques, computadora – microcontrolador – encoders.

Como se nota el uso de los encoders aumenta la complejidad en el proyecto, no por eso lo hace necesariamente mejor, como se pudo ver principalmente en la sensibilidad tanto de los potenciómetros como de los encoders manejados en este análisis.

Observamos también que se requerirá el uso de más circuitos entre ellos el microcontrolador (figura 4.8), que regularmente para trabajar requiere de una tarjeta electrónica que permita su uso, lo cual implica que la computadora debe tener al menos un puerto serie o un puerto paralelo más, para permitir la comunicación entre estos dos.

El programa de control hecho en lenguaje C con que se trabaja actualmente, se basa en la comparación de los valores digitales de los potenciómetros recibidos por el convertidor analógico/digital, como referencia de estabilización, lo que quiere decir que el proyecto realizado con los encoders implica desarrollar una nueva programación para acoplar los valores enviados por el microcontrolador.

La circuitería electrónica ya desarrollada no se vería muy afectada ya que los cambios radicarían en la eliminación del sensado de los potenciómetros, pero si sería fundamental el desarrollo de la tarjeta adecuada que tendría el manejo del microcontrolador y las interfaces necesarias para comunicarse con el contenedor de los circuitos así como las interfaces que se comunicaran con la computadora.

El diseño mecánico se ve afectado también, ya que se quiere utilizar un solo contenedor para todas las tarjetas impresas y circuitos electrónicos, y el añadir el microcontrolador, al igual que sus interfaces de comunicación con la plataforma y computadora variará el diseño y dimensiones de dicho contenedor.

APÉNDICE B

PROGRAMA EN LENGUAJE C, PARA EL CONTROL DE LA PLATAFORMA

```

/*****
** Control de Plataforma A,rea
**
** Realizar mediciones de voltajes en modo simple en cualquiera de los
** 8-canales del C.A.D. ADC0838 de National Semiconductors y obtiene
** los resultados directamente en la PC. Se asume que V_ref = 5.0 VDC.
**
**
**
**
** Terminal del Puerto P. Terminal del Circuito I.
** BUSY ----- 11 DO ----- 14
** Data_0 ----- 2 /CS ----- 18
** Data_1 ----- 3 CLK ----- 16
** Data_2 ----- 4 DI ----- 17
** Data_3 ----- 5 ----- on/off motor 1 (x)
**
** Data_4 ----- 6 ----- direccin motor 1 (x)
** Data_5 ----- 7 ----- on/off motor 2 (y)
** Data_6 ----- 8 ----- direccin motor 2 (y)
** Data_7 ----- 9 ----- on/off motor 3 (z)
** C0 ----- 1 ----- direccin motor 3 (z)
** C1 ----- 14 ----- pulso de encendido
** S3 ----- 15 ----- pulso de almacenamiento
** C2 ----- 16 ----- frecuencia de proceso
** GND ----- 18 ----- GND
**
**
*****/

```

```

#include <stdio.h>
#include <dos.h>
#include <conio.h>
#include <math.h>
#include <ctype.h>
#include <time.h>
#include <string.h>

```

```
#define ESTADO DATO+1
```

TESIS CON
FALLA DE ORIGEN

```
#define CONTROL DATO+2
#define grad 57.29577951308232087
#define rad 0.01745329251994329

int DATO;
int dato = 0x00;
int datol = 0x08;
int dato2 = 0x18;
unsigned int PUERTO;
FILE *fp;
char nombre[18];
int valor[8];
int toma = 0;
int m, c, ch;
float pit;
float rol;
float co;
float tem;
float pitch;
float pitch_2;
float roll;
float roll_2;
float com;
float temp_2;
float com_inicial;
char s[10], *p;
float L, h, g;
float segundos;
float t = 0;
float x = 0;
float y = 0;
float z = 0;
float temperatura;
float voltaje[8];
float comodin;
float rango[4] = {0};
float linea[4] = {0};
float velocidad_angular[3];
float posicion_angular[3];
float posicion_en_potenciometro[3];
float voltaje_inicial[8] = {0, 0, 0, 0, 0, 0, 0, 0};
float aceleracion[2] = {0, 0};
time_t tiempo_local;
char hora[26];
int pulso = 0;
int estado = 1;
int accion;
int otra;
int cual;
int caso=27;

int azimuth(int caso);
void mensa(void);
unsigned int detecta_puertos(void);
int hacer_medicion(int canal);
int tomar_valor(void);
int hacer_palabra_de_control(int canal);
```

```

void enviar_palabra_de_control(int palabra);
void mensajes(void);
void reloj();
void CS_alto(void), CS_bajo(void);
void hacer_calibracion();
void modelo_gravitacional();
void algoritmo();
void compas();
void hacer_control();
void rangos_roll_1();
void rangos_roll_2();
void rangos_compas_1();
void rangos_compas_2();
void potenciometros();
void archivar();
void control();
void cambio_de_linea();
void calib_mensaje();
void control_previo();

void main(void)
{
    int n;

    textattr(0x0F);
    for(n=0; n<49; n++) cprintf("\n");
    DATO = detecta_puertos();
    mensajes();
    modelo_gravitacional();

    /* Hace calibración de la plataforma con respecto a la horizontal local */

    hacer_calibracion();
    rangos_compas_1();
    rangos_compas_2();
    calib_mensaje();
    hacer_calibracion();
    rangos_roll_1();
    rangos_roll_2();
    calib_mensaje();
    hacer_calibracion();
    n=0;
    clrscr();
    control();
}

void control(void)
{
do
{
    if (kbhit()) caso = getch();
    if (es+ndo != 0)
    {
        if (caso == 27)
        {
            clrscr();

```

```

        caso=azimuth(caso);
        if (caso != 113)
        {
            clrscr();
            gotoxy(67,4);
            printf("%.2f [øC]",tem);
            gotoxy(8,24);
            printf("%.2f [ø]",rol);
            gotoxy(30,24);
            printf("%.2f [ø]",pit);
            gotoxy(51,24);
            printf("%.2f [ø]",co);
            gotoxy(10,22);
            cprintf("g = %.7f[m/s\^2]",g);
            mensa();
        }
        else
        {
            clrscr();
            fclose(fp);
            outportb(CONTROL,0x09);
            exit(0);
        }
    }
}
else
{
    clrscr();
    mensa();
    gotoxy(67,4);
    cprintf("No disponible");
    gotoxy(26,20);
    cprintf("No disponible");
    gotoxy(8,24);
    cprintf("No disponible");
    gotoxy(30,24);
    cprintf("No disponible");
    gotoxy(51,24);
    cprintf("No disponible");
    mensa();
}

algoritmo();
hacer_control();
if(estado !=0) compas();

/* Revisa el estado del pin 15 para almacenar datos */
if((inportb(ESTADO) | 0xF7) == 0xFF) pulso = 1;
if(((inportb(ESTADO) | 0xF7) == 0xF7) && pulso == 1) archivar();

    if (caso == 113)
    {
        clrscr();
    }
, while (caso != 113);
fclose(fp);

```

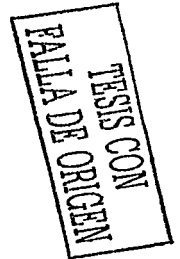


```

    outportb(CONTROL, 0x09);
}

int azimuth(int caso)
{
    caso = 0;
    gotoxy(10,23);
    cprintf("
");
    gotoxy(22,6);
    cprintf("1\ Sur - Norte (Cero grados)\n");
    gotoxy(22,8);
    cprintf("2\ Norte - Sur (180 grados)\n");
    gotoxy(22,10);
    cprintf("3\ Oeste - Este (90 grados)\n");
    gotoxy(22,12);
    cprintf("4\ Este - Oeste (270 grados)\n");
    gotoxy(22,15);
    cprintf("T\ Teminar");
    gotoxy(45,15);
    cprintf("S\ Continuar");
    gotoxy(20,20);
    cprintf(" Proporcione una opcion: ");
    accion = getch();
    switch (accion)
    {
        case 115:  caso = 0;          /* s -> 115 Continuar */
                  break;
        case 116:  caso = 113;       /* t -> 116, q -> 113. Terminar */
                  break;
        case 49:   /* 1) Sur - Norte apuntar a cero grados */
                  do
                  {
                      gotoxy(10,23);
                      cprintf("Eligio una direccion \"Sur - Norte\" esta de
acuerdo(s/n)? ");
                      cual = getch();
                      if ( cual == 115 )
                      {
                          otra = 1;
                      }
                      else if (cual == 110)
                      {
                          otra = 1;
                      }
                      else
                      {
                          otra = 0;
                      }
                  }while( otra != 1 );
                  if(cual == 115)
                  {
                      do
                      {
                          /* do grande de verificacion de posicion */
                          outportb(PUERTO, 'c');

```



```

outportb(PUERTO, '?');
outportb(PUERTO, '13');
do{
    c = inportb(PUERTO + 5);
    if (c & 1)
    {
        ch = inportb(PUERTO);
    }
    else
    {
        outportb(PUERTO, 'c');
        outportb(PUERTO, '?');
        outportb(PUERTO, '13');
    }
}while(ch != 'C');
p = s;
do{
    c = inportb(PUERTO + 5);
    if (c & 1)
    {
        ch = inportb(PUERTO);
        if (isdigit(ch) || ch == '-' || ch ==
        *p++ = ch;
    }
    else
    {
        outportb(PUERTO, 'c');
        outportb(PUERTO, '?');
        outportb(PUERTO, '13');
    }

    clrscr();
}while(ch != 13);
co = atof(s);
clrscr();
gotoxy(5,1);
cprintf("Usted decidio continuar con Sur - Norte
\n");

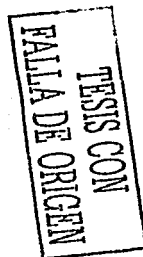
gotoxy(5,3);
cprintf("Espere un momento mientras se orienta la
plataforma ...");

if (co > 200)
{
    co = co - 360;
}

/* para proteger el potenciómetro */
if (( voltaje[7] > 1) && (voltaje[7] < 4))
{
    posicion_angular[2] = 0.004;
    if((co <= 0.7) && (co >= -0.3))

    else
    {
        if (co > 0)

```



```

        {
            outportb(CONTROL, 0x0B);
            outportb(DATO, dato1 << 4);
            delay(85);
            outportb(CONTROL, 0x0B);
            outportb(DATO, 0x00);
        }
        if (co < 0)
        {
            outportb(CONTROL, 0x0B);
            outportb(DATO, dato2 << 4);
            outportb(CONTROL, 0x0A);
            delay(85);
            outportb(CONTROL, 0x0B);
            outportb(DATO, 0x00);
        }
    }

    /* fin de proteccion de potenciómetros */

    } while (posicion_angular[2] >= 0.004); /* fin del do
while que inicio pidiendo valor a la brújula */

    linea[0] = 1;
    linea[1] = -1;
    linea[2] = 1;
    linea[3] = 0;

    } /* fin del if del cual == 115 */

else
{
    gotoxy(18,3);
    cprintf("

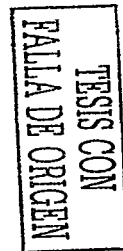
");

    gotoxy(25,3);
    cprintf(" Eliga la opcion Correcta");
    azimuth(caso);
}
control_previo();
control_previo();
break;

case 50: /* 2) Norte - Sur apuntar a 180 grados */
do
{
    gotoxy(10,23);
    cprintf("Eligio una direccion \"Norte - Sur\" esta de
acuerdo(s/n)?: ");

    cual = getch();
    if (cual == 115)
    {
        otra = 1;
    }
    else if (cual == 110)
    {

```



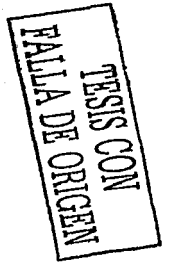
```

        otra = 1;
    }
    else
    {
        otra = 0;
    }
}while( otra != 1 );

if(cual == 115) /* Acepto continuar con esta direccion */
{
    do
    { /* do grande de verificacion de posicion */

        outportb(PUERTO, 'c');
        outportb(PUERTO, '?');
        outportb(PUERTO, '13');
        do{
            c = inportb(PUERTO + 5);
            if (c & 1)
            {
                ch = inportb(PUERTO);
            }
            else
            {
                outportb(PUERTO, 'c');
                outportb(PUERTO, '?');
                outportb(PUERTO, '13');
            }
        }while(ch != 'C');
        p = s;
        do{
            c = inportb(PUERTO + 5);
            if (c & 1)
            {
                ch = inportb(PUERTO);
                if (isdigit(ch) || ch == '-' || ch ==
                    *p++ = ch;
            }
            else
            {
                outportb(PUERTO, 'c');
                outportb(PUERTO, '?');
                outportb(PUERTO, '13');
            }
        }
        clrscr();
    }while(ch != 13);
    co = atof(s);
    clrscr();
    gotoxy(5,1);
    cprintf("Usted decidio continuar con Norte - Sur
\n");
    gotoxy(5,3);
    cprintf("Espere un momento mientras la plataforma
se orienta ...");
    gotoxy(5,5);

```



```

co);

printf("El valor actual del compas es: %.2f",

/* para proteger el potenciómetro */
if (( voltaje[7] > 1) & (voltaje[7] < 4))
{
    posicion_angular[2] = 0.004;
    if((co <= 180.7) && (co >= 179.3))
    else
    {
        if (co > 180)
        {
            outportb(CONTROL, 0x0B);
            outportb(DATO, dato1 << 4);
            delay(85);
            outportb(CONTROL, 0x0B);
            outportb(DATO, 0x00);
        }
        if (co < 180)
        {
            outportb(CONTROL, 0x0B);
            outportb(DATO, dato2 << 4);
            outportb(CONTROL, 0x0A);
            delay(85);
            outportb(CONTROL, 0x0B);
            outportb(DATO, 0x00);
        }
    }
}

) /* fin de proteccion de potenciómetros */

} while (posicion_angular[2] >= 0.004); /* fin del do
while que inicio pidiendo valor a la brujula */

linea[0] = 2;
linea[1] = 179;
linea[2] = 181;
linea[3] = 180;

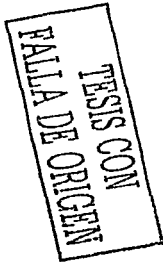
} /* fin del if del cual == 115 */
else
{
    gotoxy(18,3);
    cprintf("

");

    gotoxy(25,3);
    cprintf(" Elija la opcion Correcta");
    azimuth(caso);
}
control_previo();
control_previo();
break;

case 51: /* 3) Oeste - Este apuntar a 90 grados */

```



```

do
{
gotoxy(10,23);
cprintf("Eligio una direccion \"Oeste - Este\" esta de
acuerdo(s/n)?: ");
cual = getch();
if ( cual == 115 )
{
otra = 1;
}
else if (cual == 110)
{
otra = 1;
}
else
{
otra = 0;
}
}while( otra != 1 );
if(cual == 115)
{
do
{
/* do grande de verificacion de posicion */

outportb(PUERTO, 'c');
outportb(PUERTO, '?');
outportb(PUERTO, '13');
do{
c = inportb(PUERTO + 5);
if (c & 1)
{
ch = inportb(PUERTO);
}
else
{
outportb(PUERTO, 'c');
outportb(PUERTO, '?');
outportb(PUERTO, '13');
}
}while(ch != 'C');
p = s;
do{
c = inportb(PUERTO + 5);
if (c & 1)
{
ch = inportb(PUERTO);
if (isdigit(ch) || ch == '-' || ch ==
'..')
*p++ = ch;
}
else
{
outportb(PUERTO, 'c');
outportb(PUERTO, '?');
outportb(PUERTO, '13');
}
}
}
}

```

**TESIS CON
FALLA DE ORIGEN**

```

        clrscr();
    }while(ch != 13);
    co = atof(s);
    clrscr();
    gotoxy(5,1);
    cprintf("Usted decidio continuar con Oeste - Este
\n");

    gotoxy(5,3);
    cprintf("Espere un momento mientras la plataforma
se orienta ...");

    gotoxy(5,5);
    cprintf("El valor actual del compas es: %.2f",
co);

    /* para proteger el potenciometro */
    if (( voltaje[7] > 1) && (voltaje[7] < 4))
    {
        posicion_angular[2] = 0.004;
        if((co <= 90.7) && (co >= 89.3))

        posicion_angular[2] = 0.003;

        else
        {
            if (co > 90)
            {
                outportb(CONTROL, 0x0B);
                outportb(DATO, dato1 << 4);
                delay(85);
                outportb(CONTROL, 0x0B);
                outportb(DATO, 0x00);
            }
            if (co < 90)
            {
                outportb(CONTROL, 0x0B);
                outportb(DATO, dato2 << 4);
                outportb(CONTROL, 0x0A);
                delay(85);
                outportb(CONTROL, 0x0B);
                outportb(DATO, 0x00);
            }
        }
    }

    } /* fin de proteccion de potenciometros */

    } while (posicion_angular[2] >= 0.004); /* fin del do
while que inicio pidiendo valor a la brujula */

    linea[0] = 3;
    linea[1] = 89;
    linea[2] = 91;
    linea[3] = 90;

    } /* fin del if del cual == 115 */
else

```

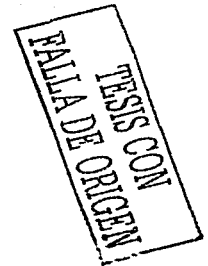
**TESIS CON
FALLA DE ORIGEN**

```

    {
        gotoxy(18,3);
        cprintf("
");
        gotoxy(25,3);
        cprintf(" Eliga la opcion Correcta");
        azimuth(caso);
    }
    control_previo();
    control_previo();
    break;

case 52:          /* 4) Oeste - Este apuntar a 270 grados */
do
{
    gotoxy(10,23);
    cprintf("Eligio una direccion \"Este - Oeste\" esta de
acuerdo(s/n)? ");
    cual = getch();
    if ( cual == 115 )
    {
        otra = 1;
    }
    else if (cual == 110)
    {
        otra = 1;
    }
    else
    {
        otra = 0;
    }
}while( otra != 1 );
if(cual == 115)
{
do
{
    /* do grande de verificacion de posicion */
    outportb(PUERTO, 'c');
    outportb(PUERTO, '?');
    outportb(PUERTO, '13');
    do{
        c = inportb(PUERTO + 5);
        if (c & 1)
        {
            ch = inportb(PUERTO);
        }
        else
        {
            outportb(PUERTO, 's');
            outportb(PUERTO, '?');
            outportb(PUERTO, '13');
        }
    }while(ch != 'C');
    p = s;
}do{

```




```

c = inportb(PUERTO + 5);
if (c & 1)
{
    ch = inportb(PUERTO);
    if (isdigit(ch) || ch == '-' || ch ==
'.')
        *p++ = ch;
    }
else
{
    outportb(PUERTO, 'c');
    outportb(PUERTO, '2');
    outportb(PUERTO, '13');
}
clrscr();
}while(ch != 13);
co = atof(s);
clrscr();
gotoxy(5,1);
cprintf("Usted decidio continuar con Este - Oeste
\n");
gotoxy(5,3);
cprintf("Espere un momento mientras la plataforma
se orienta ...");
gotoxy(5,5);
cprintf("El valor actual del compas es: %.2f",
co);

/* para proteger el potenciometro */
if (( voltaje[7] > 1) && (voltaje[7] < 4))
{
    posicion_angular[2] = 0.004;
    if((co <= 270.7) && (co >= 269.3))
        posicion_angular[2] = 0.003;
    else
    {
        if (co > 270)
        {
            outportb(CONTROL, 0x0B);
            outportb(DATO, dato1 << 4);
            delay(85);
            outportb(CONTROL, 0x0B);
            outportb(DATO, 0x00);
        }
        if (co < 270)
        {
            outportb(CONTROL, 0x0B);
            outportb(DATO, dato2 << 4);
            outportb(CONTROL, 0x0A);
            delay(85);
            outportb(CONTROL, 0x0B);
            outportb(DATO, 0x00);
        }
    }
}

} /* fin de proteccion de potenciometros */

```

TESIS CON
 FALLA DE ORIGEN

```

        ) while (posicion_angular[2] >= 0.004); /* fin del do
while que inicio pidiendo valor a la brújula */

        linea[0] = 4;
        linea[1] = 269;
        linea[2] = 271;
        linea[3] = 270;

        } /* fin del if del cual == 115 */

    else
    {
        gotoxy(18,3);
        cprintf("

");
        gotoxy(25,3);
        cprintf(" Elija la opcion Correcta");
        azimuth(caso);
    }
    control_previo();
    control_previo();
    break;

default: gotoxy(10,3);
        cprintf("

");
        gotoxy(22,3);
        cprintf("Proporcione una opcion correcta");
        azimuth(caso);
        break;

    }

return(caso);
}

unsigned int detecta_puertos(void)
{
    unsigned int far *apuntador_a_direccion;
    unsigned int far *apuntador; /* COM1 0x03F8 */
    unsigned int direcciones[3];
    int i, LPT;
    int contador = 0;
    char seleccion;

    clrscr();
    textattr(0x0F); /* negro blanco */

    /* Sección Puerto Serie */
    apuntador = (unsigned int far *)0x00000400;
    PUERTO = *apuntador;
    /* Fija par metros */
    outportb(PUERTO + 1, 0); /* Apaga las interrupciones */
    outportb(PUERTO + 3, 0x80); /* SET DLAB ON */
    /* Fija Baud Rate - retardo del Byte bajo de división */

```

TESIS CON
 FALLA DE ORIGEN

```

outportb(PUERTO + 0, 0x06); /* Default 0x02 = 57,600 BPS */
                          /*      0x03 = 38,400 BPS */
                          /*      0x06 = 19,200 BPS */
                          /*      0x0C = 9,600 BPS */
/* Fija Baud Rate - retardo del Byte alto de divisiçn */
outportb(PUERTO + 1, 0x00);
outportb(PUERTO + 3, 0x03); /* 8 Bits, Sin paridad, 1 Bit de paro */
outportb(PUERTO + 2, 0xC7); /* FIFO Registro de control) */
outportb(PUERTO + 4, 0x0B); /* Enciende DTR, RTS, y OUT2 */
outportb(PUERTO, 13);

/*                               Sección Puerto Paralelo                               */
apuntador_a_direccion = (unsigned int far *)0x408;
for(i=0; i<3; i++)
{
  gotoxy(20,10+i);
  if(*apuntador_a_direccion == 0)
    printf("%d. No se detecta puerto en LPT%d \n\r", i+1, i+1);
  else
  {
    LPT = i;
    printf("%d. Puerto detectado en LPT%d\n\r",
           i+1, i+1, *apuntador_a_direccion);
    direcciones[i] = *apuntador_a_direccion;
    contador = contador + 1;
  }
  apuntador_a_direccion++;
}
if(contador == 1) DATO = direcciones[LPT];
else
{
  gotoxy(10,15);
  printf("Seleccione el nmero de puerto que desee utilizar: ");
  do
  {
    gotoxy(61,15);
    seleccion = getch();
    switch (seleccion)
    {
      case '1':
        DATO = direcciones[0];
        break;
      case '2':
        DATO = direcciones[1];
        break;
      case '3':
        DATO = direcciones[2];
        break;
    }
    gotoxy(20,18);
    if(DATO == 0x0) printf("Opciçn no v lida");
  }while (DATO == 0x0);
}
return DATO;
}

void mensajes(void)

```

TESIS CON
 FALLA DE ORIGEN

```

{
  int n;
  int i=0;
  int tecla;
  int extension = 0;
  char grados[10];
  char minutos[10];
  char altitud[10];

  clrscr();
  outportb(PUERTO, 'p');
  outportb(PUERTO, 'e');
  outportb(PUERTO, 'r');
  outportb(PUERTO, 'i');
  outportb(PUERTO, 'o');
  outportb(PUERTO, 'd');
  outportb(PUERTO, '=');
  outportb(PUERTO, 2);
  outportb(PUERTO, 13);

  do { c = inportb(PUERTO + 5);
      if (c & 1) ch = inportb(PUERTO);
      else
      {
        i++;
        if (i > 30000)
        {
          clrscr();
          gotoxy(10,10);
          cprintf("No se ha podido establecer comunicaci#n con el inclin#metro:");
          cprintf("\n\n\r          (R) Reintentar Conexi#n");
          cprintf("\n\n\r          (S) Salir");
          tecla = getch();
          if (tecla == 'R' || tecla == 'r') mensajes();
          if (tecla == 'S' || tecla == 's') exit(1);
        }
      }
      if (isascii(ch) == 0)
      {
        estado = 0;
        gotoxy(16,10);
        cprintf("El puerto serie esta ocupado por otro dispositivo");
        gotoxy(11,12);
        cprintf("Aseg#rese de que el puerto est, libre y reinicie el equipo");
        cprintf("\n\n\r          (S) Salir");
        tecla = getch();
        if (tecla == 'S' || tecla == 's') exit(1);
        getch();
        break;
      }
    } while(ch != ':');

  clrscr();
  do
  {
    gotoxy(12,4);
    cprintf("Introduzca el nombre del archivo para almacenamiento:");
  }

```

TESIS CON
 FALLA DE ORIGEN

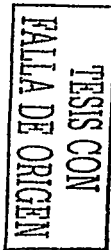
```

    cprintf("\n\r                                (Enter para default)");
    gotoxy(25,7);
    cprintf("                                ");
    gotoxy(25,7);
    gets(nombre);
    for (n=0; n<18; n++) if (nombre[n] == '.') extension = 1;
    if (extension == 0)
    {
        strcat(nombre, ".txt");
        gotoxy(25,7);
        cprintf("%s", nombre);
    }
    gotoxy(20,20);
    if (strlen(nombre) > 13) cprintf("Debe ser menor de 10 caracteres");
    else cprintf("                                ");
}while(strlen(nombre) > 10);

if((fp = fopen(nombre, "w")) == NULL)
{
    gotoxy(25,7);
    cprintf("posicion.txt");
    strcpy(nombre, "posicion.txt");
}
if((fp = fopen(nombre, "w")) == NULL)
{
    gotoxy(12,22);
    fprintf(stderr, "Error de apertura de %s.", nombre);
}
tiempo_local = time(NULL);
strcpy(hora, ctime(&tiempo_local));
fprintf(fp, "\t\tDate: ");
for(n=0; n<4; n++) fprintf(fp, "%c", hora[n]);
for(n=8; n<11; n++) fprintf(fp, "%c", hora[n]);
for(n=4; n<8; n++) fprintf(fp, "%c", hora[n]);
for(n=20; n<24; n++) fprintf(fp, "%c", hora[n]);
fprintf(fp, "\n\n Toma           X           Y           Z           Hora\n");

do
{
    gotoxy(12,10);
    cprintf("Introduzca la latitud:");
    gotoxy(25,12);
    cprintf("Grados:           Minutos:           ");
    gotoxy(33,12);
    gets(grados);
    if (strcmp(grados, "\0") == 0)
    {
        strcpy(grados, "19");
        gotoxy(33,12);
        cprintf("%s", grados);
    }
    gotoxy(51,12);
    gets(minutos);
    if (strcmp(minutos, "\0") == 0)
    {
        strcpy(minutos, "43");
        gotoxy(51,12);
    }
}

```



```

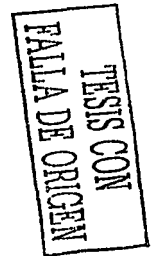
    cprintf("%s", minutos);
}
L = atof(grados) + (atof(minutos) / 60);
gotoxy(20,20);
if(L<0 | L>90) cprintf("Latitud incorrecta ");
else cprintf(" ");
} while(L<0 | L>90);
do
{
gotoxy(12,15);
cprintf("Introduzca la altitud en metros: ");
gotoxy(45,15);
gets(altitud);
if (strcmp(altitud, "\0") == 0)
{
strcpy(altitud, "2200");
gotoxy(45,15);
cprintf("%s", altitud);
}
h = atof(altitud);
if(h<0 | h>6000)
{
gotoxy(20,20);
cprintf("Altura incorrecta ");
}
} while(h<0 | h>6000);

if (estado != 0)
{
calib_mensaje();
}
} /* Fin de la funcion */

void mensa(void)
{
int n;

textattr(0x1B);
gotoxy(20,1);
cprintf("ESTADO ACTUAL DEL SISTEMA");
gotoxy(40,25);
if(estado != 0)
{
cprintf("PRESIONE LA TECLA \"ESC\" PARA DETENER");
}
else
{
cprintf("Presione LA TECLA \"q\" PARA SALIR");
}
gotoxy(10,3);
cprintf(" GIRÀSCOPOS:");
gotoxy(10,8);
cprintf(" ACELERÀMETROS:");
gotoxy(10,12);

```



```

cprintf(" POSICION ANGULAR DE LOS SERVOS:");
gotoxy(10,17);
cprintf(" RESULTADO DEL ALGORITMO DE ORIENTACIãN:");
gotoxy(10,23);
cprintf(" INCLINãMETRO:");
gotoxy(54,4);
cprintf("TEMPERATURA: ");
gotoxy(54,6);
cprintf("TOMA NO.");
gotoxy(10,21);
cprintf("GRAVEDAD LOCAL");
textattr(0x0F);

gotoxy(2,24);
cprintf("Roll: ");
gotoxy(23,24);
cprintf("Pitch: ");
gotoxy(46,24);
cprintf("Com: ");
gotoxy(1,4);
for (n=112; n<115; n++)
  cprintf(" Velocidad angular en girçscopo %c :\n\r", n);
cprintf("\n\n");
for (n=120; n<122; n++)
  cprintf(" Fuerza especifica en acelerçmetro %c :\n\r", n);
cprintf("\n\n");
for (n=1; n<4; n++)
  cprintf(" Posiciçn angular del potenciçmetro %d es:\n\r", n);
cprintf("\n\n");
for (n=120; n<123; n++)
  cprintf(" Posiciçn angular en %c :\n\r", n);
}

void modelo_gravitacional(void)
{
  L = L * 0.0174532925199432957692390768;
  g = (9.780318*((1 + 0.0053024*pow(sin(L),2)) - (0.0000059*pow(sin(2*L),2))))
    / (pow((1 + ((h*(1 - 0.0069943*pow(sin(L),2)) / 6351767.123)),2)));
  gotoxy(10,22);
  cprintf("g = %.7f [m/sý]", g);
}

void hacer_calibracion(void)
{
  int n;
  if(estado == 0)
  {
    gotoxy(67,4);
    cprintf("No disponible");
    gotoxy(26,20);
    cprintf("No disponible");
    gotoxy(8,24);
    cprintf("No disponible");
    gotoxy(30,24);
    cprintf("No disponible");
  }
}

```

TESIS CON
 FALLA DE ORIGEN

```

else
{
    for (n=2; n>=0; n--)
    {
        do {
            outportb(PUERTO, 's');
            outportb(PUERTO, '?');
            outportb(PUERTO, 13);
            do { c = inportb(PUERTO + 5);
                if (c & 1) ch = inportb(PUERTO);
                else
                {
                    outportb(PUERTO, 's');
                    outportb(PUERTO, '?');
                    outportb(PUERTO, 13);
                }
            } while(ch != 'R');
            for(m=80; m<89; m+=4)
            {
                p = s;
                do
                {
                    c = inportb(PUERTO + 5);
                    if (c & 1)
                    {
                        ch = inportb(PUERTO);
                        if(isdigit(ch) || ch=='-' || ch=='.' )
                            *p++ = ch;
                    }
                    else
                    {
                        outportb(PUERTO, 's');
                        outportb(PUERTO, '?');
                        outportb(PUERTO, 13);
                    }
                } while(ch != m);
                switch (m)
                {
                    case 80: roll = atof(s);
                    break;
                    case 84: pitch = atof(s);
                    break;
                    case 88: temperatura = atof(s);
                    break;
                }
            }
            do { c = inportb(PUERTO + 5);
                if (c & 1) ch = inportb(PUERTO);
                else
                {
                    outportb(PUERTO, 's');
                    outportb(PUERTO, '?');
                    outportb(PUERTO, '13');
                }
            } while(ch != 'C');

```

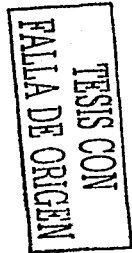
TESIS CON
 FALLA DE ORIGEN


```

p = s;
do { c = inportb(PUERTO + 5);
    if (c & 1)
    {
        ch = inportb(PUERTO);
        if(isdigit(ch) || ch == '-' || ch == '.')
            *p++ = ch;
    }
    else
    {
        outportb(PUERTO, 's');
        outportb(PUERTO, '?');
        outportb(PUERTO, 13);
    }
} while (ch != '*');
com = atof(s);
com_inicial = com;
/* do { c = inportb(PUERTO + 5);
    if (c & 1) ch = inportb(PUERTO);
    else
    {
        outportb(PUERTO, 's');
        outportb(PUERTO, '?');
        outportb(PUERTO, '13');
    }
} while (ch != 13); */
roll = (0.8439734273 * roll) + 0.5086346521;
pitch = (0.9270909213 * pitch) + 0.2787453369;
gotoxy(67,4);
printf("%.2f [°C] ", temperatura);
tem=temperatura;
gotoxy(8,24);
cprintf("%.2f [°] ", roll);
rol=roll;
gotoxy(30,24);
cprintf("%.2f [°] ", pitch);
pit=pitch;
gotoxy(51,24);
cprintf("%.2f [°] ", com);
co=com;

roll = rad * roll;
pitch = rad * pitch;
posicion_angular[0] = -pitch;
posicion_angular[1] = -roll;
posicion_angular[2] = co;
valor[n+5] = hacer_medicion(n+5);
voltaje[n+5] = 5.0 * valor[n+5] / 256.0;
if (n != 2)
{
    if (posicion_angular[n] > 0)
    {
        outportb(CONTROL, 0x0B);
        outportb(DATO, dato1 << n*2);
        delay(130);
        outportb(CONTROL, 0x0B);
    }
}

```



```

        outportb(DATO, 0x00);
    }
    if (posicion_angular[n] < 0)
    {
        outportb(CONTROL, 0x0B);
        outportb(DATO, dato2 << n*2);
        if(n == 2) outportb(CONTROL, 0x0A);
        delay(190);
        outportb(CONTROL, 0x0B);
        outportb(DATO, 0x00);
    }
}
else
{
    if ((posicion_angular[n] >= 179.5) && (posicion_angular[n] <=
180.5)) posicion_angular[n] = 0.003;
    else
    {
        if (posicion_angular[n] > 180)
        {
            outport(CONTROL, 0x0B);
            outportb(DATO, dato1 << n*2);
            delay(90);
            outportb(CONTROL, 0x0B);
            outportb(DATO, 0x00);
        }
        if (posicion_angular[n] < 180)
        {
            outportb(CONTROL, 0x0B);
            outportb(DATO, dato2 << n*2);
            if(n == 2) outportb(CONTROL, 0x0A);
            delay(90);
            outportb(CONTROL, 0x0B);
            outportb(DATO, 0x00);
        }
    }
} /* fin del else n != 2 */
/* proteccion de los potses */
} while(fabs(posicion_angular[n]) >= 0.004);
}
/* Realiza una medicion inicial en cada uno de los canales del CAD */
delay(2000);
for(n=0; n<8; n++)
{
    valor[n] = hacer_medicion(n);
    voltaje_inicial[n] = 5.0 * valor[n] / 256.0;
}
}

int hacer_medicion(int canal)
/* Realiza la medicion en un canal especifico */
{
    int palabra_de_control, valor;

    CS_alto();
    CS_bajo(); /* Reinicia el convertidor A/D */
}

```

TESIS CON
 FALLA DE ORIGEN

```

palabra_de_control = hacer_palabra_de_control(canal);
enviar_palabra_de_control(palabra_de_control);
valor = tomar_valor();
CS_alto();          /* Apaga el convertidor A/D */
return(valor);
}

int tomar_valor(void)
{
    int bit, valor=0x00, n;

    /* Ahora la salida digital (DO) est  en alta impedancia */
    reloj();
    for(n=7; n>=0; n--)
    {
        bit = ((inportb(ESTADO)^0x80)>>7)&0x01;
        valor = valor | (bit << n);
        reloj();
    }
    return(valor);
}

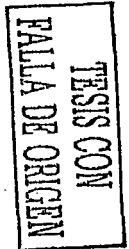
int hacer_palabra_de_control(int canal)
{
    int palabra, bitmenos;

    palabra = 0x18;          /* 1 1000 */
    bitmenos = canal & 0x01;
    palabra = palabra | (bitmenos <<2); /* 1 1 bit - sig. 0 0 */
    palabra = palabra | (canal >>1);
    /* 1 1 bit menos significativo, bit mas significativo, medio */
    return(palabra);
}

void enviar_palabra_de_control (int palabra)
/* Envía la palabra de control, un bit ala vez en Data_2S */
/* comenzando con el mas significativo */
{
    int n, bit;
    for (n=4; n>=0; n--)
    {
        bit = (palabra >>n) & 0x01;          /* Aisla al bit */
        dato = dato & (~0x04);          /* Posición cero 2 */
        dato = dato | (bit <<2);          /* Inserta el bit */
        outportb(DATO, dato);
        reloj();
    }
}

void reloj(void)
/* Pone a Data_1 en alto y despu,s bajo */
{
    dato = dato | 0x02;
    outportb(DATO, dato);
    delay(1);
    dato = dato & (~0x02);
    outportb(DATO, dato);
}

```



```

    delay(1);
}

void CS_alto(void)
{
    dato = dato | 0x01;
    outportb(DATO, dato);
}

void CS_bajo(void)
{
    dato = dato & (~0x01);
    outportb(DATO, dato);
}

void algoritmo(void)
{
    int n, k;
    time_t t1, t2;
    float xc[2];
    float yc[2];
    float kx = 0.12653637;
    float ky = -0.12653637;
    float axm, aym;
    float p, q, r;

    for(k=0; k<2; k++)
    {
        for(n=0; n<5; n++)
        {
            if (n == 1)
            {
                if (k == 0)
                {
                    t1 = clock();
                    delay(3000);
                }
                else
                {
                    t2 = clock();
                    t = (t2 - t1) / CLK_TCK;
                }
            }
            valor[n] = hacer_medicion(n);
            voltaje[n] = 5.0 * valor[n] / 256.0;
            if (n < 3)
            {
                if(fabs(voltaje[n]-voltaje_inicial[n]) < 0.02)
                {
                    voltaje[n] = voltaje_inicial[n];
                    velocidad_angular[n] = 1.14960891783 * (voltaje[n] - voltaje_inicial[n]);
                    gotoxy(37,n+4);
                    cprintf("%.4f [ø/s]    ", grad * velocidad_angular[n]);
                }
                else
                {
                    if(fabs(voltaje[n]-voltaje_inicial[n]) < 0.03)
                    {
                        voltaje[n] = voltaje_inicial[n];
                    }
                }
            }
        }
    }
}

```

```

    aceleracion[n-3] = 9.7794266 * (voltaje[n] - voltaje_inicial[n]);
    gotoxy(40,n+6);
    cprintf("%.4f [m/sý] ", aceleracion[n-3]);
}
}
p = velocidad_angular[0];
q = velocidad_angular[1];
r = velocidad_angular[2];
xc[k] = q * cos(y) - r * sin(y);
yc[k] = p + tan(x) * ((q * sin(y)) + (r * cos(y)));
axm = aceleracion[0];
aym = aceleracion[1];
axm = g * sin(x) - axm;
aym = -g * cos(x) * sin(y) - aym;
axm = kx * axm;
aym = ky * aym;
}
x = ((xc[0] + xc[1]) / 2) * t + x - axm;
y = ((yc[0] + yc[1]) / 2) * t + y - aym;
posicion_angular[0] = x;
posicion_angular[1] = y;
/* Despliega el resultado del algoritmo */
gotoxy(26,18);
cprintf("%.4f [º] ", grad * x);
gotoxy(26,19);
cprintf("%.4f [º] ", grad * y);
gotoxy(26,20);
cprintf("%.4f [º] ", co);
potenciometros();

/* Revisa el estado del pin 15 para almacenar datos */
if((inportb(ESTADO) | 0xF7) == 0xFF) pulso = 1;
if(((inportb(ESTADO) | 0xF7) == 0xF7) && pulso == 1) archivar();
}

void compas(void)
{
    if((inportb(ESTADO) | 0xF7) == 0xFF) pulso = 1;
    if(((inportb(ESTADO) | 0xF7) == 0xF7) && pulso == 1) archivar();
    p=s;

    outportb(PUERTO, 's');
    outportb(PUERTO, '?');
    outportb(PUERTO, 13);
    do
    {
        c = inportb(PUERTO + 5);
        if (c & 1) ch = inportb(PUERTO);
        else
        {
            if((inportb(ESTADO) | 0xF7) == 0xFF) pulso = 1;
            if(((inportb(ESTADO) | 0xF7) == 0xF7) && pulso == 1)
                archivar();

            outportb(PUERTO, 's');
            outportb(PUERTO, '?');
        }
    }
}

```

```

        outportb(PUERTO, 13);
    }
} while(ch != 'R');
for(m=80; m<89; m+=4)
{
    p = s;
    do
    {
        c = inportb(PUERTO + 5);
        if (c & 1)
        {
            ch = inportb(PUERTO);
            if(isdigit(ch) || ch=='-' || ch=='.')
                *p++ = ch;
        }
        else
        {
            if((inportb(ESTADO) | 0xF7) == 0xFF) pulso = 1;
            if(((inportb(ESTADO) | 0xF7) == 0xF7) && pulso == 1)
                archivar();

            outportb(PUERTO, 's');
            outportb(PUERTO, '?');
            outportb(PUERTO, 13);
        }
    }while(ch != m);
    switch (m)
    {
        case 80: roll_2 = atof(s);
                break;
        case 84: pitch_2 = atof(s);
                break;
        case 88: temp_2 = atof(s);
                break;
    }
}

if((inportb(ESTADO) | 0xF7) == 0xFF) pulso = 1;
if(((inportb(ESTADO) | 0xF7) == 0xF7) && pulso == 1) archivar();

do {
    c = inportb(PUERTO + 5);
    if (c & 1) ch = inportb(PUERTO);
    else
    {
        if((inportb(ESTADO) | 0xF7) == 0xFF) pulso = 1;
        if(((inportb(ESTADO) | 0xF7) == 0xF7) && pulso == 1)
            archivar();

        outportb(PUERTO, 's');
        outportb(PUERTO, '?');
        outportb(PUERTO, 13);
    }
} while(ch != 'C');
p = s;
do
{
    c = inportb(PUERTO + 5);
    if (c & 1)

```

```

    {
        if(((inportb(ESTADO) | 0xF7) == 0xFF) pulso = 1;
        if(((inportb(ESTADO) | 0xF7) == 0xF7) && pulso == 1)
archivar();

        ch = inportb(PUERTO);
        if(isdigit(ch) || ch == '-' || ch == '.')
            *p++ = ch;
    }
    else
    {
        if(((inportb(ESTADO) | 0xF7) == 0xFF) pulso = 1;
        if(((inportb(ESTADO) | 0xF7) == 0xF7) && pulso == 1)
archivar();

        outportb(PUERTO, 's');
        outportb(PUERTO, '?');
        outportb(PUERTO, 13);
    }
}while (ch != '*');
co = atof(s);

```

```

gotoxy(67, 4);
cprintf("%.2f [SC]", temp_2);
gotoxy(8, 24);
cprintf("%.2f [Ø] ", roll_2);
gotoxy(30, 24);
cprintf("%.2f [Ø] ", pitch_2);
gotoxy(51, 24);
cprintf("%.2f [Ø] ", co);
}

```

```

void potenciometros(void)

```

```

{
    int n;

    for(n=0; n<3; n++)
    {
        valor[n+5] = hacer_medicion(n+5);
        voltaje[n+5] = 5.0 * valor[n+5] / 256.0;
        posicion_en_potenciometro[n] = -0.6981317008 * (voltaje[n+5]
            - voltaje_inicial[n+5]);
        gotoxy(43, n+13);
        cprintf("%.4f [Ø] ", grad * posicion_en_potenciometro[n]);
    }
}

```

```

void hacer_control(void)

```

```

{
    int n;
    float error;
    mensa();
    for (n=0; n<(estado+2); n++)
    {
        if ((voltaje[n+5] > 1) && (voltaje[n+5] < 4))
        {
            if ( n < 2)
            {

```

```

        if((voltaje[6] > rango[0]) && (voltaje[6] < rango[1]))
        {
            error = posicion_en_potenciometro[n] -
posicion_angular[n];
            if ( (( (roll_2 >= 1) || (roll_2 <= -1) ) && ( n ==
1) ) || ((pitch_2 >= 1) || (pitch_2 <= -1)) && (n == 0))
            {
                error = posicion_en_potenciometro[n] -
posicion_angular[n];
                compas();
                if (kbhit()) break;
                /* if (error < 0) */
                if (n == 0)
                {
                    if (pitch_2 < 0)
                    {
                        outportb(CONTROL, 0x0B);
                        outportb(DATO, dato1 << n*2);
                        delay(140);
                        outportb(CONTROL, 0x0B);
                        outportb(DATO, 0x00);
                    }
                }
                /* if (error > 0) */
                if (pitch_2 > 0)
                {
                    outportb(CONTROL, 0x0B);
                    outportb(DATO, dato2 << n*2);
                    if (n == 2) outportb(CONTROL,
0x0A);

                    delay(200);
                    outportb(CONTROL, 0x0B);
                    outportb(DATO, 0x00);
                }
            }
            if (n == 1)
            {
                if (roll_2 < 0 )
                {
                    outportb(CONTROL, 0x0B);
                    outportb(DATO, dato1 << n*2);
                    delay(140);
                    outportb(CONTROL, 0x0B);
                    outportb(DATO, 0x00);
                }
            }
            /* if (error > 0) */
            if (roll_2 > 0 )
            {
                outportb(CONTROL, 0x0B);
                outportb(DATO, dato2 << n*2);
                if (n == 2) outportb(CONTROL,
0x0A);

                delay(200);
                outportb(CONTROL, 0x0B);
                outportb(DATO, 0x00);
            }
        }
    }
}

```



```

    }
}

    algoritmo();
    valor[n+5] = hacer_medicion(n+5);
    voltaje[n+5] = 5.0 * valor[n+5] / 256.0;
    posicion_en_potenciometro[n] = -0.6981317008
* (voltaje[n+5] - voltaje_inicial[n+5]);
    error = posicion_en_potenciometro[n] -
posicion_angular[n];
    }
} /* fin de swicheo del roll */
else
{
    cambio_de_linea();
}
} /* Fin del if n < 2 */
else
{
    if((voltaje[7] > rango[2]) && (voltaje[7] < comodin))
    {
        if ((linea[0] == 1) && (co > 200))
        {
            co = co - 360;
        }
        if ((co >= linea[1]-0.5) && (co <= linea[2]+0.5))
posicion_angular[2] = 0.003;
        else
        {
            if (co > linea[3])
            {
                outport(CONTROL, 0x0B);
                outportb(DATO, dato1 << n*2);
                delay(90);
                outportb(CONTROL, 0x0B);
                outportb(DATO, 0x00);
            }
            if (co < linea[3])
            {
                outportb(CONTROL, 0x0B);
                outportb(DATO, dato2 << n*2);
                if(n == 2)outportb(CONTROL, 0x0A);
                delay(90);
                outportb(CONTROL, 0x0B);
                outportb(DATO, 0x00);
            }
        }
    }
}
else
{
    cambio_de_linea();
}
} /* fin del else del if n < 2 */

```

```

        } /* de la proteccion del los potes */
    } /* fin del for */
} /* fin de la funcion */

void archivar(void)
{
    int c;
    float posicion[3];

    tiempo_local = time(NULL);
    strcpy(hora, ctime(&tiempo_local));
    if (toma < 10000) toma++;
    fprintf(fp, "\n %04d  ", toma);
    gotoxy(63,6);
    cprintf("%i  ", toma);
    fprintf(fp, " %-13.5f", roll_2);
    fprintf(fp, " %-13.5f", pitch_2);
    fprintf(fp, " %-13.5f\t", co);
    for (c=11; c<19; c++) fprintf(fp, "%c", hora[c]);
    pulso = 0;
}

void rangos_roll_1(void)
{
    clrscr();
    do
    {
        outportb(PUERTO, 'r');
        outportb(PUERTO, '?');
        outportb(PUERTO, 13);
        do
        {
            c = inportb(PUERTO + 5);
            if (c & 1) ch = inportb(PUERTO);
            else
            {
                outportb(PUERTO, 'r');
                outportb(PUERTO, '?');
                outportb(PUERTO, 13);
            }
        }while(ch != 'R');
        p = s;
        do
        {
            c = inportb(PUERTO + 5);
            if(c & 1)
            {
                ch = inportb(PUERTO);
                if(isdigit(ch) || ch == '-' || ch == '.') *p++ = ch;
            }
            else
            {
                outportb(PUERTO, 'r');
                outportb(PUERTO, '?');
                outportb(PUERTO, 13);
            }
        }
    }
}

```

```

    }while(ch != 13);
    rol = -1 * atof(s);
    /*
     * esta girando a la derecha */
    outportb(CONTROL, 0x0B);
    outportb(DATO, dato1 << 2);
    delay(250);
    outportb(CONTROL, 0x0B);
    outportb(DATO, 0x00);
    valor[6] = hacer_medicion(6);
    rango[0] = 5.0 * valor[6] / 256.0;
    gotoxy(10, 10);
    cprintf("Buscando límites en ROL. rol = %.2f, con rango =
%.2f", rol, rango[0]);
    }while(rol > -16);
} /* fin de funcion */

void rangos_roll_2(void)
{
    clrscr();
    do
    {
        outportb(PUERTO, 'r');
        outportb(PUERTO, '?');
        outportb(PUERTO, 13);
        do
        {
            c = inportb(PUERTO + 5);
            if (c & 1) ch = inportb(PUERTO);
            else
            {
                outportb(PUERTO, 'r');
                outportb(PUERTO, '?');
                outportb(PUERTO, 13);
            }
        }while(ch != 'R');
        p = s;
        do
        {
            c = inportb(PUERTO + 5);
            if(c & 1)
            {
                ch = inportb(PUERTO);
                if(isdigit(ch) || ch == '-' || ch == '.') *p++ = ch;
            }
            else
            {
                outportb(PUERTO, 'r');
                outportb(PUERTO, '?');
                outportb(PUERTO, 13);
            }
        }
    }while(ch != 13);
    rol = -1 * atof(s);
    /*
     * esta girando a la izquierda */

```

```

        outportb(CONTROL, 0x0B);
        outportb(DATO, dato2 << 2);
        delay(390);
        outportb(CONTROL, 0x0B);
        outportb(DATO, 0x00);

        valor[6] = hacer_medicion(6);
        rango[1] = 5.0 * valor[6] / 256.0;
        gotoxy(10,10);
        cprintf("Buscando límites en ROL. rol = %.2f, con rango =
%.2f",rol,rango[1]);
    }while(rol < 16);

} /* fin de funcion */

void rangos_compas_1(void)
{
    clrscr();
    do
    {
        outportb(PUERTO, 'c');
        outportb(PUERTO, '?');
        outportb(PUERTO, 13);
        do
        {
            c = inportb(PUERTO + 5);
            if (c & 1) ch = inportb(PUERTO);
            else
            {
                outportb(PUERTO, 'c');
                outportb(PUERTO, '?');
                outportb(PUERTO, 13);
            }
        }while(ch != 'C');
        p = s;
        do
        {
            c = inportb(PUERTO + 5);
            if(c & 1)
            {
                ch = inportb(PUERTO);
                if(isdigit(ch) || ch == '-' || ch == '.') *p++ = ch;
            }
            else
            {
                outportb(PUERTO, 'c');
                outportb(PUERTO, '?');
                outportb(PUERTO, 13);
            }
        }
        }while(ch != 13);
        co = atof(s);
        outportb(CONTROL, 0x0B);
        outportb(DATO, datol << 4);
        delay(90);

```

```

        outportb(CONTROL, 0x0B);
        outportb(DATO, 0x00);
        valor[7] = hacer_medicion(7);
        rango[2] = 5.0 * valor[7] / 256.0;
        gotoxy(10, 10);
        cprintf("Buscando limites en el COMPAS. compas = %.2f, con rango =
%.2F", co, rango[2]);
    }while(co > 145);
} /* fin de funcion */

void rangos_compas_2(void)
{
    clrscr();
    do
    {
        outportb(PUERTO, 'c');
        outportb(PUERTO, '?');
        outportb(PUERTO, 13);
        do
        {
            c = inportb(PUERTO + 5);
            if (c & 1) ch = inportb(PUERTO);
            else
            {
                outportb(PUERTO, 'c');
                outportb(PUERTO, '?');
                outportb(PUERTO, 13);
            }
        }while(ch != 'C');
        p = s;
        do
        {
            c = inportb(PUERTO + 5);
            if(c & 1)
            {
                ch = inportb(PUERTO);
                if(isdigit(ch) || ch == '-' || ch == '.') *p++ = ch;
            }
            else
            {
                outportb(PUERTO, 'c');
                outportb(PUERTO, '?');
                outportb(PUERTO, 13);
            }
        }while(ch != 13);
        co = atof(s);
        outportb(CONTROL, 0x0B);
        outportb(DATO, dato2 << 4);
        outportb(CONTROL, 0x0A);
        delay(100);
        outportb(CONTROL, 0x0B);
        outportb(DATO, 0x00);
        valor[7] = hacer_medicion(7);
        rango[3] = 5.0 * valor[7] / 256.0;

```

```

        comodin = rango[3];
        gotoxy(10,10);
        cprintf("Buscando límites en el COMPAS. compas = %.2f, con rango =
%.2f",co,rango[3]);
        }while(co < 210);
} /* fin de funcion */

void cambio_de_linea()
{
int n;
float error;
clrscr();
switch( linea[0] )
{
    case 1:
        linea[0] = 2;
        linea[1] = 179;
        linea[2] = 181;
        linea[3] = 180;
        break;

    case 2:
        linea[0] = 1;
        linea[1] = -1;
        linea[2] = 1;
        linea[3] = 0;
        break;

    case 3:
        linea[0] = 4;
        linea[1] = 269;
        linea[2] = 271;
        linea[3] = 270;
        break;

    case 4:
        linea[0] = 3;
        linea[1] = 89;
        linea[2] = 91;
        linea[3] = 90;
        break;

    default:
        hacer_control();
        break;
}
}

do
{
if ( kbhit() ) caso = getch();
if (caso == 27 )
{
    clrscr();
    control();
}
}

```

```

else
{
    outportb(PUERTO, 'c');
    outportb(PUERTO, '?');
    outportb(PUERTO, 13);
    do
    {
        c = inportb(PUERTO + 5);
        if(c & 1) ch = inportb(PUERTO);
        else
        {
            outportb(PUERTO, 'c');
            outportb(PUERTO, '?');
            outportb(PUERTO, 13);
        }
    }while(ch != 'C');
    p = s;
    do
    {
        c = inportb(PUERTO + 5);
        if(c & 1)
        {
            ch = inportb(PUERTO);
            if(isdigit(ch) || ch == '-' || ch == '.') *p++ = ch;
        }
        else
        {
            outportb(PUERTO, 'c');
            outportb(PUERTO, '?');
            outportb(PUERTO, 13);
        }
    }while(ch != 13);
    co = atof(s);
    if (( linea[0] == 1 ) && (co > 200))
    {
        co = co - 360;
    }
    clrscr();
    {
        for(n=0; n<3; n++)
        {
            valor[n+5] = hacer_medicion(n+5);
            voltaje[n+5] = 5.0 * valor[n+5] / 256.0;
            posicion_en_potenciometro[n] = -0.6981317008 *
(voltaje[n+5] - voltaje_inicial[n+5]);
        }
    }
    printf("\n\n\tEl comp s es %f\n\n",co);
    printf("\n\n\n*****          Presiona ESC para ir al menf
principal          ***** \n");
}

}while (!(( co >= linea[3]-30) && (co <= linea[3]+30)));
clrscr();
control_previo();
control_previo();

```

```

}

void calib_mensaje(void)
{
    clrscr();
    gotoxy(20,7);
    cprintf("SISTEMA DE NAVEGACION INERCIAL");
    gotoxy(8,13);
    cprintf("SE ESTA LLEVANDO A CABO LA CALIBRACION INICIAL ESPERE .... ");
    gotoxy(54,4);
    cprintf("TEMPERATURA:");
    gotoxy(10,21);
    cprintf("GRAVEDAD LOCAL");
    gotoxy(2,24);
    cprintf("Roll: ");
    gotoxy(23,24);
    cprintf("Pitch: ");
    gotoxy(46,24);
    cprintf("Com: ");
}

void control_previo(void)
{
    int n,i;
    float error;
    mensa();
    algoritmo();
    potenciometros();
    for (n=2; n>=0; n--)
        {
            if ( n < 2)
                {
                    error = posicion_en_potenciometro[n] -
posicion_angular[n];
                    while (fabs(error) > 0.05)
                        {
                            if (kbhit()) break;
                            if ((error < 0) &&
(posicion_en_potenciometro[n] < 1.4))
                                {
                                    outportb(CONTROL, 0x0B);
                                    outportb(DATO, dato1 << n*2);
                                    delay(150);
                                }
                            /* giro a la izquierda */
                            if ((error > 0) &&
(posicion_en_potenciometro[n] > -1.4))
                                {
                                    outportb(CONTROL, 0x0B);
                                    outportb(DATO, dato2 << n*2);
                                    if (n == 2) outportb(CONTROL, 0x0A);
                                    delay(150);
                                }
                        }
                    algoritmo();
                    valor[n+5] = hacer_medicion(n+5);
                    voltaje[n+5] = 5.0 * valor[n+5] / 256.0;
                }
        }
}

```



```

                                posicion_en_potenciometro[n] = -0.6981317008
* (voltaje[n+5] - voltaje_inicial[n+5]);
                                error = posicion_en_potenciometro[n] -
posicion_angular[n];
                                } /* fin del while (fabs(error) > 0.05) */
                                } /* Fin del if n < 2 */
                                else
                                {
do
                                { /* do grande de verificacion de posicion */
                                outportb(PUERTO, 'c');
                                outportb(PUERTO, '?');
                                outportb(PUERTO, '13');
                                do{
                                    c = inportb(PUERTO + 5);
                                    if (c & 1)
                                    {
                                        ch = inportb(PUERTO);
                                    }
                                    else
                                    {
                                        outportb(PUERTO, 'c');
                                        outportb(PUERTO, '?');
                                        outportb(PUERTO, '13');
                                    }
                                }while(ch != 'C');
                                p = s;
                                do{
                                    c = inportb(PUERTO + 5);
                                    if (c & 1)
                                    {
                                        ch = inportb(PUERTO);
                                        if (isdigit(ch) || ch == '-' || ch ==
',.')
                                        *p++ = ch;
                                    }
                                    else
                                    {
                                        outportb(PUERTO, 'c');
                                        outportb(PUERTO, '?');
                                        outportb(PUERTO, 13);
                                    }
                                }while(ch != 13);
                                co = atof(s);
                                if ((linea[0] == 1) && (co > 200))
                                {
                                    co = co - 360;
                                }
                                if (co > linea[3])
                                {
                                    outportb(CONTROL, 0x0B);
                                    outportb(DATO, dato1 << 4);
                                    delay(90);
                                    outportb(CONTROL, 0x0B);
                                }

```

```
        outportb(DATO, 0x00);
    }
    if (co < linea[3])
    {
        outportb(CONTROL, 0x0B);
        outportb(DATO, dato2 << 4);
        outportb(CONTROL, 0x0A);
        delay(90);
        outportb(CONTROL, 0x0B);
        outportb(DATO, 0x00);
    }
    } while (!(co >= linea[3]-2) && (co <= linea[3]+2));
    } /* fin del else del if n < 2 */
} /* fin del for */
}
```

BIBLIOGRAFÍA

- 1.- ADVANCED ORIENTATION SYSTEMS, Inc., EZ-COMPASS-3 Application Manual. Linden, NJ 1999.
- 2.- Bar Itzhack I.Y. and Ziv I. "Frequency and time domain designs of a strapdown vertical determination system". Proceedings AIAA Guidance, Navigation and Control Conference. Williamsburg, August 1986. Paper 86-2149.
- 3.- Becerril Báez J. Manuel, Rodríguez Díaz Mario. "Sistema de Navegación Inercial basado en giroscopios y acelerómetros". Tesis de Licenciatura. Facultad de Ingeniería, UNAM 2000
- 4.- Benjamin C. Kuo. Sistemas de Control Digital, 1ª Edición en español. Ed. CECSA, México, 1997.
- 5.- Buechler D. and Foss M. "Integration of GPS and strapdown inertial subsystems into a single unit". Navigation, Journal of the Institute of Navigation. 34,2, pp 140-159. 1987.
- 6.- Considine Douglas M. "Encyclopedia of instrumentation and control". Huntington, New York 1981. Mc Graw-Hill Inc. 1971.
- 7.- Farrell J.A, Barth M. The Global Positioning System and Inertial Navigation. Mc Graw-Hill 1999.
- 8.- Franco M. "Sistema de control autónomo de una plataforma aérea, para adquisición de imágenes". Tesis de Licenciatura, Facultad de Estudios Superiores Cuautitlán, UNAM 2000.
- 9.- Garg S.C, Morrow L.D. and Mamen R. "Strapdown Navigation Technology: A Literature Survey". Journal of Guidance Control and Dynamics. Vol. 1, No. 3. May-June 1978.
- 10.- G.R. Pitman (editor), "Inertial Guidance". Ed. Jhon Wiley & Sons, Inc. U.S.A., 1962.
- 11.- Herrera Bernard. "Elementos de Fotogrametría. Uso de materiales aerofotográficos". Universidad Autónoma de Chapingo. Limusa, 1987.
- 12.- ICSENSORS. Hoja de especificaciones técnicas. Acelerómetro Modelo 3140. U.S.A.,1999.
- 13.- J. H. Blakelock. "Automatic Control of Aircraft and Missiles". Wiley and Sons, New York, 1991.
- 14.- Katsuhiko Ogata. "Ingeniería de Control Moderna". Ed. Prentice Hall. México, 1984.
- 15.- Lawrence A. Modern Inertial Technology. Springer. Berlin, Germany.1998.
- 16.- Martínez E.U., Bermúdez G.R. "Despliegue y almacenamiento de datos para un sistema de imágenes aéreas". Tesis de Licenciatura. Facultad de Ingeniería, UNAM 2000.

- 17.- Merhav S. *Aerospace Sensor Systems and Applications*. Springer, New York, 1993.
- 18.- Ortega J.A. y Domínguez H. "Análisis matemático de las vibraciones no lineales a casos concretos de la ingeniería rotodinámica". Dirección de estudios de posgrado e investigación, IPN 1997.
- 19.- Prado J., Sadovnichii S., Bisiacchi G., Franco M.J. "Control de orientación de una plataforma para adquisición de imágenes aéreas". XIII Cong. Nal. Inst. Ensenada BCN, Oct.1998. pp418-422.
- 20.- Scott D. Orlosky and Harold D. Morris "A QUARTZ ROTATIONAL RATE SENSOR". Systron Donner Co. pps. 1-2. U.S.A., Febrero 1995.
- 21.- SISTRON DONNER, Inc., GyroChip II. Solid State Rotation Sensor. U.S.A., 1999.
- 22.- Titterton D. H. and Weston J.L. *Strapdown Inertial Navigation Technology* .Peter Peregrinus Ltd. London, UK. 1997.

Ligas a sitios Web

- 23.- Cámaras aéreas.
<http://www.ziimaging.com>
- 24.- Circuitos electrónicos.
http://www.sunybroome.edu/~grace_t/micro/lab/ml-tr-.html
- 25.- Navegación inercial.
<http://www.imar-navigation.de/englishside/imar.htm>
- 26.- Fotografía aérea.
<http://www.1001fotos.com/B2C/fotoaerea.asp>
27. - Potenciómetros.
<http://www.presicioneselectronics.com>
<http://www.etisystems.com>
http://www.nteinc.com/pot_web/
<http://www.presicionsales.com/stsg.html>
- 28.- Fuentes de energía.
<http://www.teambigbro.co.uk/TechSupport.html>
- 29.- Transistores.
<http://es.geocities.com/allcircuits4/tiristores.htm>
- 30.- Encoders.
<http://www.hohner.es/index.html>
<http://www.hohner.es/de/increm/increm2.htm>
http://www.eltra.it/spa/enc_inc2.htm