PROYECTO FIN DE CARRERA

SOFTWARE DE CONTROL DE ALTO NIVEL PARA HELICÓPTERO AUTÓNOMO. ADAPTACIÓN A LA ARQUITECTURA CROMAT E IMPLEMENTACIÓN SOBRE SISTEMA OPERATIVO QNX.

Departamento de Ingeniería de Sistemas y Automática.
Escuela Superior de Ingenieros. Universidad de Sevilla.
Titulación: Ingeniería Industrial.

TUTOR: Joaquín Ferruz Melero

AUTOR: David Marcos Rodríguez

Sevilla, Septiembre de 2005

ÍNDICE

CAPÍTULO 1. INTRODUCCIÓN.

1.1. Descripción del proyecto.

El objetivo del presente proyecto de final de carrera es el portado del software de control de un *UAV* de Linux a *QNX*, mejora y ampliación del mismo, así como instalación en una máquina empotrada *PC*-104. Dichos objetivos se han ampliado con la inclusión del diseño e implementación de un generador de trayectorias.

Este trabajo se engloba dentro del proyecto de coordinación de robots aéreos y terrestres *CROMAT*, del que se hablará con más detalle más adelante en este mismo capítulo.

1.2. Justificación del proyecto.

Para la implementación del software del *UAV* se hizo necesario el uso de librerías *POSIX*. De unos años a esta parte se han venido desarrollando esta clase de herramientas en Linux, con resultados cada vez más satisfactorios. Sin embargo a día de hoy las prestaciones que este popular sistema operativo ofrece para la programación en tiempo real continúan muy por debajo de las de otros sistemas específicos para este tipo de programación. Resulta lógico por tanto el uso de alguno de estos sistemas mejor adaptados a la programación de hilos y procesos concurrentes.

Entre todos los sistemas operativos con estas propiedades que existen en el mercado se escogió *QNX* porque tanto el director del proyecto como el proyectando estaban familiarizados con su uso. Además a fecha de comienzo del proyecto ya se disponía de una instalación profesional del mismo en el laboratorio.

En lo que respecta a la mejora del software de partida, ésta se hizo necesaria debido a los fallos que el mismo contenía.

Estando este proyecto en ejecución se decidió cambiar la estructura del software *CROMAT*. Se creyó conveniente ampliar el software de partida y modificarlo de manera que se ajustara al nuevo modelo de programación.

Por último se decidió incluir también como parte del proyecto el diseño y programación de un generador de trayectorias para el *UAV*. De esta manera se rellenaba un vacío existente en la nueva estructura software entre los niveles superiores, que envían

puntos de paso y el software de control, que recibe una trayectoria y la envía al controlador de bajo nivel.

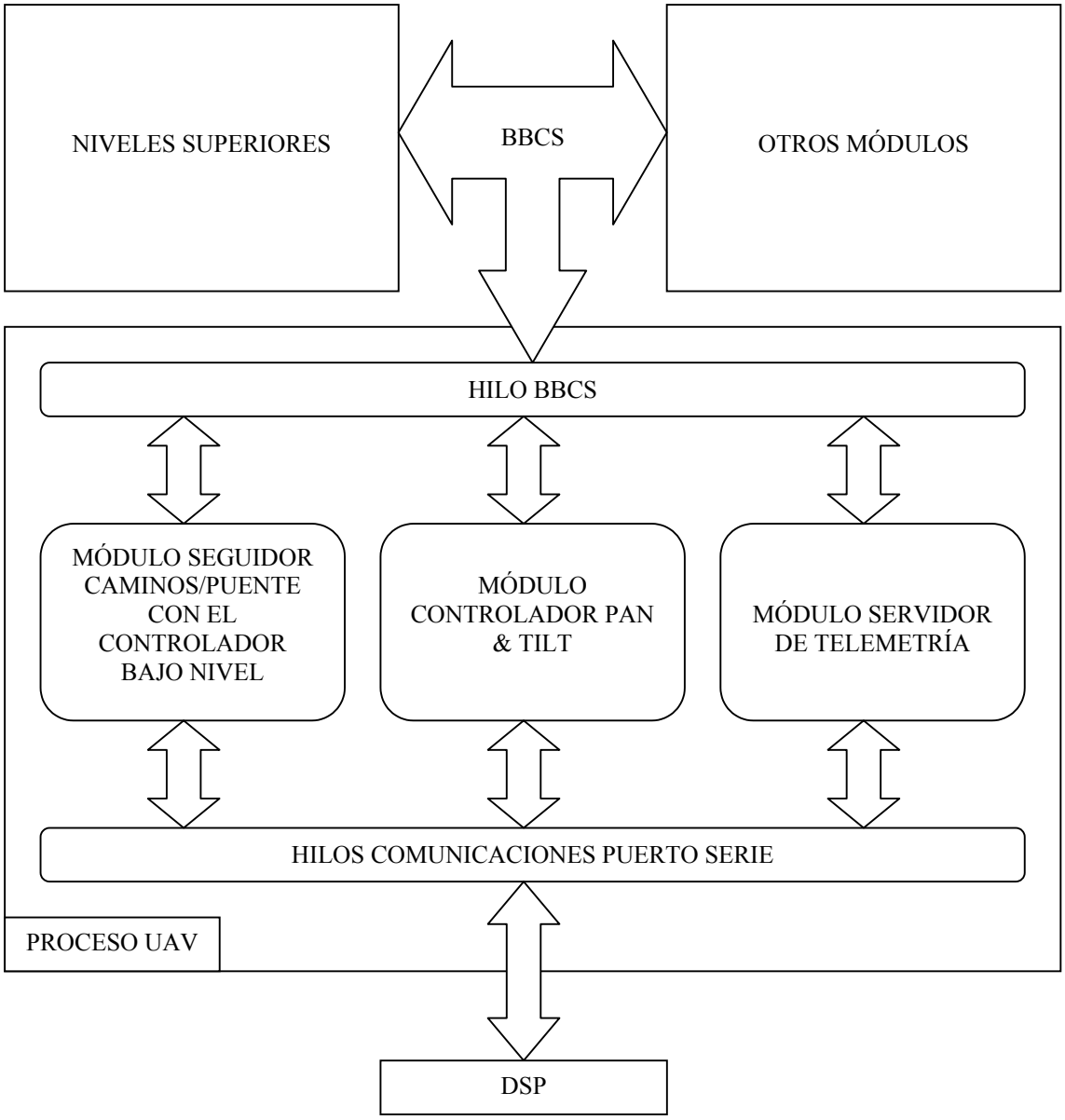

Figura 1-1. Proceso UAV para la nueva estructura software

1.3. *CROMAT.*

Acrónimo de *Coordinación de RObots Móviles Aéreos y Terrestres*, se trata de un proyecto que se realiza entre las universidades de Vigo, Málaga y Sevilla. La aportación de esta última, y en particular del departamento de Ingeniería de sistemas y Automática de la Escuela Superior de Ingenieros es un helicóptero autónomo y un vehículo terrestre de cuatro ruedas.

El objetivo de *CROMAT* es el desarrollo de técnicas de control, visión y coordinación entre robots móviles de manera que éstos realicen tareas tales como inspección de instalaciones, revisión de grandes infraestructuras, monitorización de catástrofes, vigilancia, seguridad ciudadana o detección y desactivación de minas antipersonas.

1.4. *UAV*.

Acrónimo de *Unmanned Aerial Vehicle*, que traducido al español significa *vehículo aéreo no tripulado*. En este caso en particular tal vehículo es un helicóptero de radiocontrol al que se le ha agregado el hardware y software necesarios para que vuele sin necesidad de ser manejado desde tierra. De esta manera se convierte en un robot móvil aéreo autónomo.

1.4.1. Helicópteros disponibles en *CROMAT*.

Raptor:

Vehículo pequeño con escasa capacidad de carga. Incapaz de levantar toda la electrónica, se ha venido utilizando para pequeñas pruebas con pocos elementos.

Características:

Longitud:	1150 mm
Ancho del fuselaje:	140 mm
Altura:	400 mm
Diámetro del rotor:	1254 mm
Diámetro del rotor cola:	236 mm
Relación de transmisión:	1:9.56:4.57
Peso en orden de vuelo:	3 Kg

Hirobo Eagle.

De tamaño mayor al Raptor, la capacidad de carga del *Hirobo Eagle* le permite volar con la aparamenta electrónica a bordo. Será por tanto el helicóptero en el que se ejecute el software del proyecto.

Características:

Longitud:	1430 mm
Ancho del fuselaje:	245 mm
Altura:	460 mm
Diámetro del rotor:	1560 mm
Diámetro del rotor cola:	265 mm
Relación de transmisión:	9.5:1:5.1
Peso en orden de vuelo:	4800 g

Figura 1-2. Imagen del helicóptero Hirobo Eagle.

1.4.2. Principios de funcionamiento de un helicóptero. Descripción del control.

Como cualquier sistema, un helicóptero para poder ser controlado necesita de una serie de actuadores que modifiquen su comportamiento; así como de un conjunto de sensores que describan de la manera más precisa y fiable posible el estado del aparato en cada momento.

1.4.2.1. Sensores.

GPS:

Acrónimo de *Global Position System* (sistema de posicionamiento global), el *GPS* viene utilizándose para la orientación en todo tipo de vehículos. En el caso del *UAV* se utiliza un *GPS* de tipo diferencial. Con este sistema se puede conocer la posición del helicóptero (latitud, longitud y altitud) con una precisión de centímetros en el plano, y algo menos en la altura.

IMU:

Acrónimo de *Inertial Measurement Unit*, proporciona información sobre la orientación, aceleraciones lineales y velocidades angulares en los tres ejes del espacio, a través de giróscopos, magnetómetros y acelerómetros.

Giróscopo:

Aparato que gracias al principio físico del giróscopo es capaz de calcular el giro en este caso del helicóptero sobre un eje determinado.

Sónar:

El sónar es un sensor que permite conocer la distancia respecto a un objeto. El helicóptero llevará instalado uno en la parte baja, y mirando al suelo. De esta manera cuando esté cercano a tierra puede saber de manera precisa la altura que tiene, y con ese dato realizar correctamente las complejas maniobras de despegue y aterrizaje.

Altímetro barométrico:

Da una medida de la altura más fiable que el *GPS*, ya que no depende los satélites, aunque se pierde en precisión.

Medidores de niveles:

Se tienen en el helicóptero medidores de los niveles de combustible y baterías. Aunque estos niveles no proporcionan información útil al controlador, resulta evidente la importancia de los mismos para alertar en caso de peligro por falta de carburante o descarga de las baterías.

Figura 1-3. Caja con GPS, IMU y demás electrónica.

1.4.2.2. Actuadores.

Colectivo:

La función del colectivo es incrementar o decrementar la sustentación en las palas del helicóptero. Para ello se cambia el ángulo de ataque de las mismas. Cuanto mayor sea el ángulo, mayor fuerza se ejercerá.

Cíclico:

Al igual que el colectivo, el cíclico cambia el ángulo de ataque de las palas. La diferencia reside en que el ataque de éstas varía según la posición instantánea de las mismas. De esta manera el helicóptero puede desplazarse hacia adelante, hacia atrás o hacia un lado.

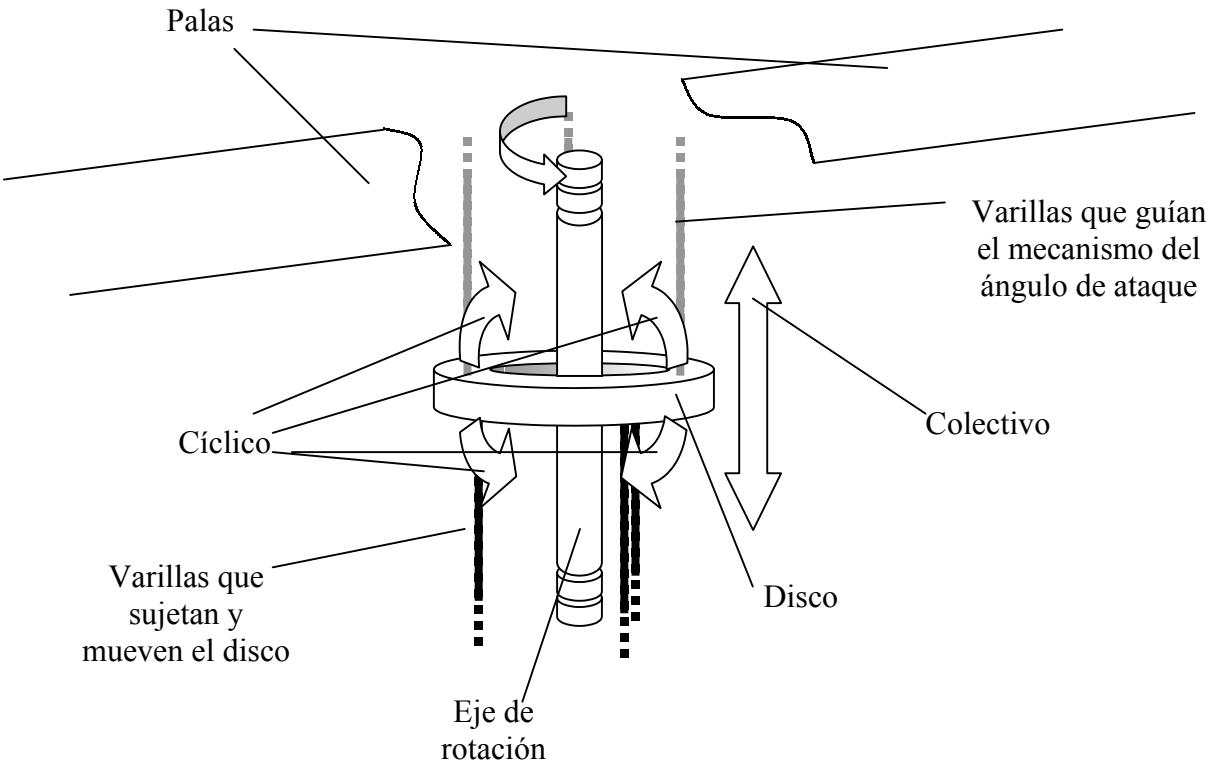

Figura 1-4. Detalle del cíclico y el colectivo.

El mecanismo para el ángulo de ataque de las palas está guiado por la posición de unas varillas que giran solidarias al eje de rotación. Éstas se deslizan sobre un disco metálico. Según la altura a la que se encuentren las bases de dichas varillas respecto a las palas, éstas se inclinarán más o menos. El colectivo mueve el disco hacia arriba o hacia abajo, y de esta manera se cambia el ángulo de ataque uniformemente para todas las posiciones de las palas. Por su parte el cíclico le da cierta inclinación al disco, por lo que las bases de las varillas irán cambiando su altura según el eje vaya dando vueltas.

Consecuentemente el ángulo de ataque de cada pala dependerá de la posición de la misma, y el helicóptero tenderá a irse hacia algún lado.

Motor:

Controlando la velocidad del motor se podría aportar o quitar sustentación. Sin embargo el control del *UAV* se va a hacer a revoluciones constantes, por lo que se obviará este actuador.

Rotor de cola:

La misión del rotor de cola es la de crear un momento que compense la rotación del rotor principal, y evite de esta manera que el cuerpo del aparato gire sobre sí mismo. Este rotor está conectado mecánicamente con el motor del helicóptero, por lo que siempre girará con la misma proporción de vueltas respecto al rotor de empuje. Sin embargo el usuario puede variar el ángulo de ataque de las palas, consiguiendo así un mayor o menos momento, según le sea necesario.

Pan & tilt:

No es propiamente un actuador del helicóptero, sino un aparato que se coloca en el frente del mismo, y donde van ubicadas las cámaras para las labores de visión. El *pan & tilt* consta de un mecanismo movido por dos servos que es capaz de orientar las cámaras, cambiando los ángulos *pan* y *tilt*.

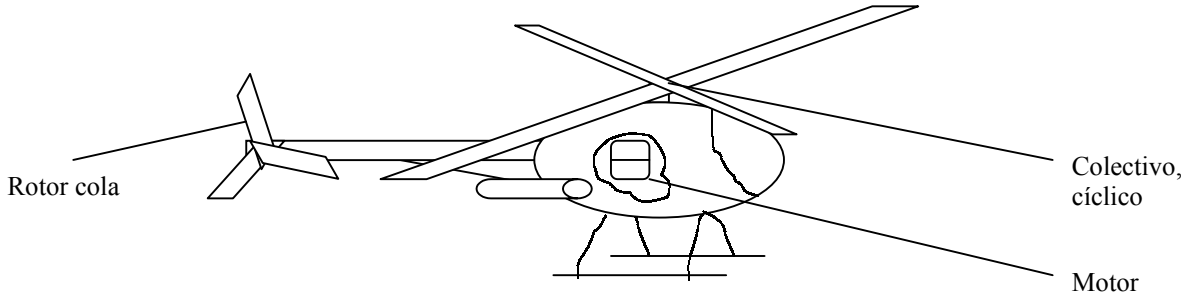

Figura 1-5. Actuadores en un helicóptero.

1.4.2.3. Problemática del control de un helicóptero.

El primer problema del control es el alto grado de acoplamiento entre todas las actuaciones. Así, si se quiere desplazar el aparato hacia un lado hay que variar el cíclico. Al hacerlo el helicóptero pierde algo de fuerza, para lo que hay que incrementar el colectivo. Por último, para evitar que el helicóptero gire sobre sí mismo se tendrá que aumentar el ángulo de ataque de las palas del rotor de cola. Al final ha habido que utilizar todos los actuadores para hacer una operación tan simple como un desplazamiento.

Otro problema es la alta inestabilidad que el sistema presenta. En efecto, cualquier variación en cualquiera de los actuadores puede provocar una inestabilización, necesitando al resto de variables para recuperar el control, mientras se procura no perderlo por otras circunstancias.

También representa un grave problema lo catastrófico de un error de control. Si a un robot terrestre le falla el control, bastará con detenerlo para evitar cualquier accidente. Y aun cuando éste sea inevitable, los daños en principio no tienen el porqué ser grandes. Sin embargo cualquier cosa que pueda sucederle al helicóptero acaba con una caída, capaz de dejar inservible el aparato.

Por último están los problemas mecánicos. El helicóptero está fabricado para volar sin carga. Aunque el motor muchas veces esté preparado para soportar un mayor peso, los aproximadamente ocho kilos de elementos electrónicos que llevará, así como el nuevo reparto de pesos complican aún más el vuelo.

1.4.3 Arquitectura hardware para el *UAV*.

Para una buena comprensión de los capítulos que siguen resulta necesario conocer la arquitectura hardware que se utilizará para el presente trabajo. Ésta se muestra en la figura 1-6.

Como se puede ver en el citado esquema, todo el control gira en torno al *DSP*. Allí se ubica el controlador de bajo nivel. Éste envía señales a los servos de los actuadores, y de esta manera controla el helicóptero. Resulta necesario, y sobre todo en fase de pruebas, la inclusión de un conmutador (*switch*) que en cualquier momento retire el control del helicóptero al *DSP* en favor de un piloto humano que lo controle con el radiomando.

Para generar la acción de control, el *DSP* necesita recoger los datos de diferentes sensores, que se detallaron en subapartados anteriores.

Además del control del movimiento, el *DSP* se encarga también del control del *pan & tilt*.

Funcionamiento normal de la arquitectura:

El usuario, que manejará el *PC* de tierra, envía una orden al *UAV*. Esta orden puede ser un *waypoint* (punto hacia donde dirigirse), un conjunto de ellos, despegue, aterrizaje, y algunas más de control, control del *pan & tilt* o de visión. Esta orden se transmitirá por *LAN*, llegando al *PC* de visión o al de control, según el caso.

Las tareas de visión quedan fuera de este proyecto, por lo que no se hablará de ellas. Suponiendo que la orden sea de control o para el *pan & tilt*, ésta llegará al *PC* de control,

donde se descompondrá en subtareas elementales; siendo éstas enviadas al *DSP*, vía puerto serie. El *PC* de control puede recibir órdenes no sólo del *PC* de tierra. También el *PC* de visión puede enviarle, por ejemplo, una serie de *waypoints* para que de esta manera el helicóptero siga un objetivo móvil, o se mueva el *pan & tilt* de manera conveniente.

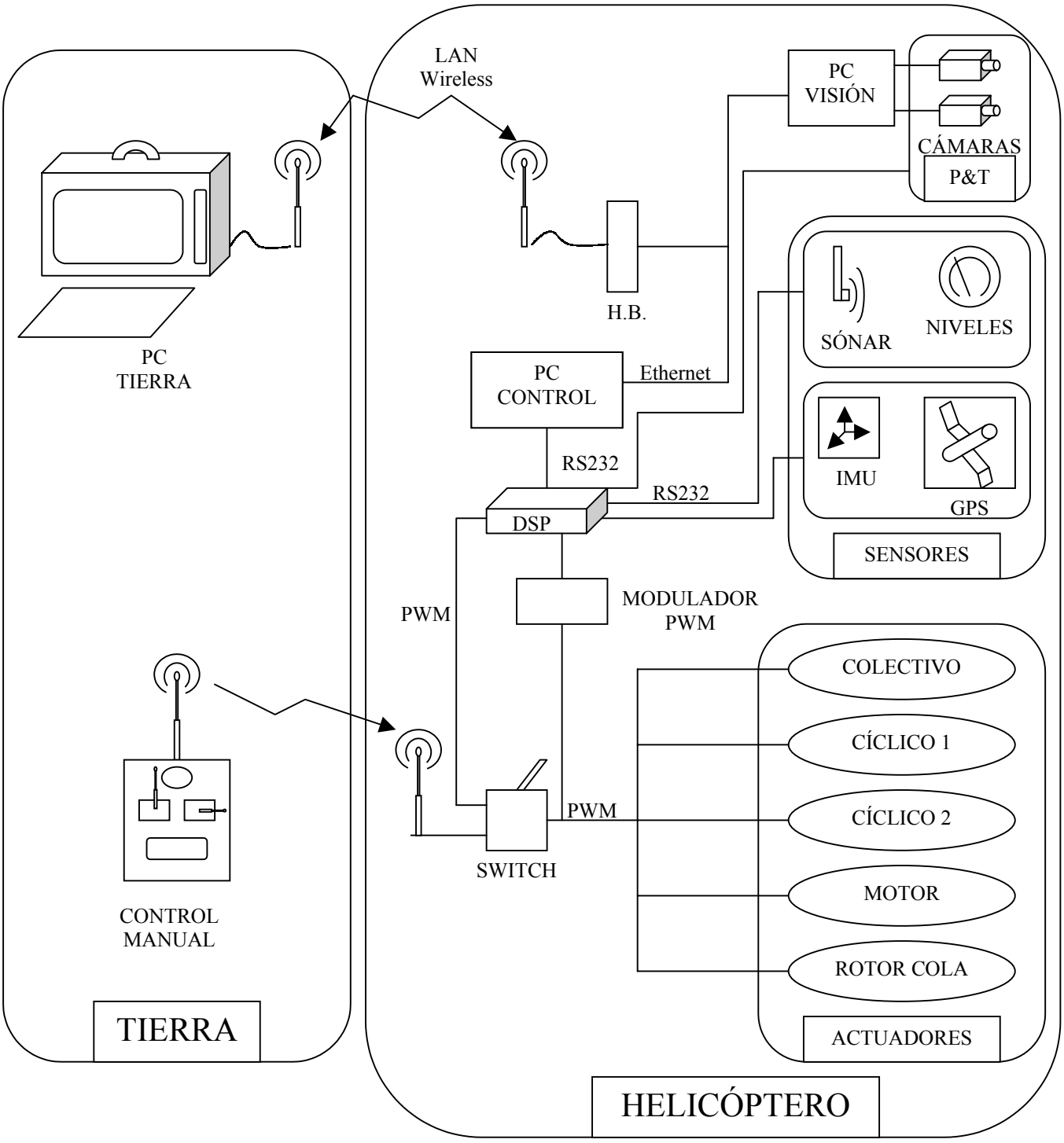

Figura 1-6. Esquema de la arquitectura para el *UAV*.

El *DSP* cumplirá las tareas asignadas, y devolverá al *PC* de control acuses de recibo, así como errores que haya podido haber y mensajes con la telemetría. El *PC* de control es el único interfaz que conecta al *DSP* con el usuario.

1.5. *QNX*.

1.5.1. Introducción.

QNX es un sistema operativo estable, fiable, basado en *UNIX*, y que está especialmente diseñado para trabajar en tiempo real. Muy recomendable para aplicaciones de robótica (tanto fija como móvil) y para cualquier tipo de sistema embebido.

1.5.2. Historia.

A principios de los años ochenta los canadienses Gordon Bell y Dan Dodge decidieron crear un sistema operativo con una interfaz similar a la de *UNIX*, basado en el paso de mensajes. Comenzaron utilizándolo para sus propios procesadores 6809 y 8088. Poco después lo adaptarían para el *IBM PC*, llamando al sistema *QUNIX* (Quick *UNIX*). Crearon la compañía *QSSL* (*Quantum Software Systems Limited*, actualmente *QNX Software Systems Limited*). Tras un aviso de la *AT&T* cambiaron de nombre su producto, adquiriendo su denominación actual, *QNX*.

A partir de entonces tanto la compañía como su producto estrella fueron evolucionando. El sistema fue adaptándose a los sucesivos procesadores *INTEL*, sacando partido de las ventajas que cada uno iba ofreciendo. En la actualidad *QNX* se comercializa en más de cien países, y ha sido utilizado por empresas e instituciones tan importantes y variadas como *INDRA*, *Honda*, *Philips*, *Texaco* o la *NASA*; aparte de innumerables universidades.

1.5.3. Principios básicos.

QNX se basa en una configuración tipo "*Modelo de proceso universal*" (*UPM*). Esta disposición utiliza un microkernel que realiza dos funciones fundamentales: el intercambio de mensajes, y la planificación. El microkernel no participa en el encolado. A él se llega sólo a través de llamadas del kernel, ya vengan de un proceso, o de una interrupción hardware.

El resto de operaciones del sistema operativo las realizan diferentes procesos, cada cual con su propio espacio de direccionamiento de memoria, protegidos entre sí. La principal ventaja de la disposición *UPM* es que el fallo de un proceso no afecta a los demás. Los procesos pueden ser detenidos, o reiniciados en caso de fallo. De esta manera se obtiene un sistema mucho más seguro y fiable que con otras configuraciones, como el ejecutivo de tiempo real o la arquitectura monolítica. De hecho *QNX* se utiliza

en algunos servicios como brazos de naves espaciales o máquinas de pulido de cristal, donde los fallos de software son inaceptables.

1.5.4. Aplicaciones.

A continuación se muestran algunos ejemplos de sistemas en los que se haya utilizado *QNX*. Como se verá las aplicaciones son muy dispares, con usos que van desde lo industrial a lo doméstico.

Sector industrial:

Empresas como Caramilk o Crunchie hacen uso de *QNX* para la monitorización y el control en la etapa de mezcla de chocolate. También se ha utilizado *QNX* en el sector cervecero para aplicaciones de inspección automatizada de botellas, con tecnologías de agarre de alta velocidad.

Sector médico:

El uso de *QNX* en este área es muy extenso. Existen aplicaciones para medir la calidad de la sangre donada, control de máquinas láser para la corrección de la vista, máquinas para electrocardiogramas, angiografías, analizadores de la densidad de los huesos; o analizadores de la composición del cuerpo para ayudar en el tratamiento contra la obesidad.

Telecomunicaciones:

Empresas como *Honda*, *Chrysler* o *Mercedes* incorporan *kits* de telefonía "manos libres" soportados por *QNX*. Además se encuentran *sets* de emisión automatizada de televisión, aparte de múltiples aplicaciones para internet, o el sistema de control de transacción de las tarjetas *VISA*.

Transportes:

QNX se utiliza para todo tipo de control de tráfico, bien sea aéreo, en aeropuertos de todo el mundo o por carretera, como el control del tráfico en puentes y vías importantes. También es utilizado en el sector ferroviario, y no sólo para el control de tráfico. Numerosas locomotoras, desde el metro hasta la alta velocidad llevan software bajo este sistema operativo.

Sector energético:

Centrales eléctricas de todos los tipos (principalmente hidroeléctricas y nucleares) hacen uso de *QNX* para sus necesidades de control.

Sector aeroespacial:

En este área *QNX* se usa tanto para el sector terrestre como el espacial. Valga como ejemplos el orientado de antenas, o algunas aplicaciones *M&C* de equipos (como convertidores de frecuencia) para la Estación Espacial Internacional.

Aplicaciones de uso general:

QNX es el sistema operativo que está detrás del teléfono de emergencias en Estados Unidos (el conocido "911"). También puede encontrarse en máquinas de autolavado de vehículos, sistemas de alarma de congestión de tráfico para conductores, máquinas automáticas de dispensado de billetes o sistemas de refrigeración de viviendas.

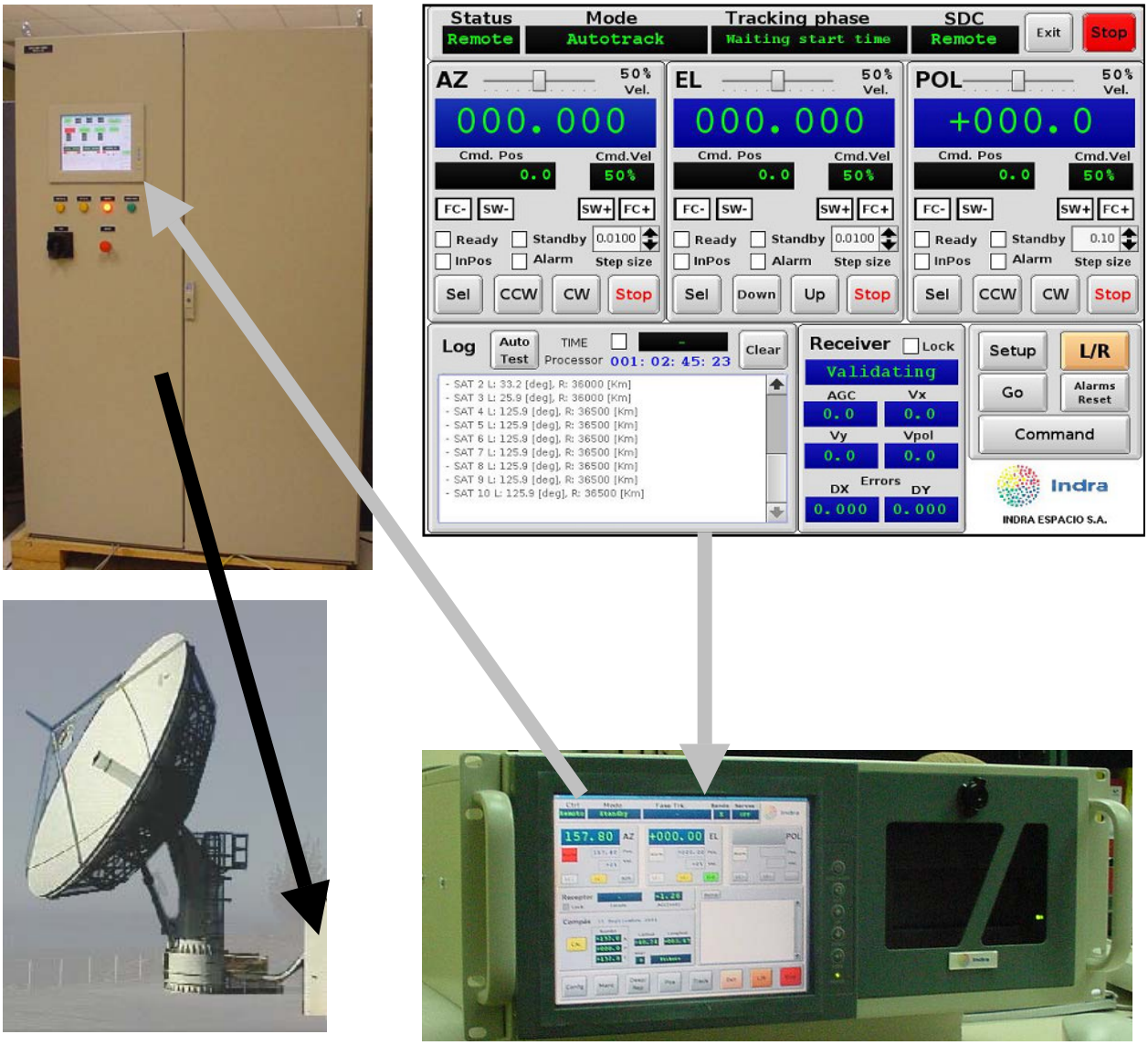

Figura 1-7. Interfaz para pantalla táctil de la unidad de control de una antena, bajo SO. QNX. INDRA espacio.