

Capítulo 5

Conclusiones

5.1 Resultados

El principal resultado del proyecto ha sido que toda la estructura software del HERO, según se define en el artículo “*Arquitectura para múltiples robots heterogéneos*”, se ha completado. Esto, que en un principio sólo supondría la creación de la capa RIL, se ha querido complementar con simuladores del helicóptero: ahora se podrán realizar todo tipo de pruebas en simulación sobre todo el proyecto HERO.

Las pruebas con el UAV Simulator fueron de una grandísima importancia, ya que fueron las primeras con la estructura completa. El simulador podía no comportarse como un helicóptero de verdad (de hecho no lo hacía) pero podíamos comprobar que los *waypoints* le llegaban al DSP y que el helicóptero obedecía; que el DSP asentía a los comandos y que los errores (generados voluntariamente) llegaban al PC. En definitiva, que el sistema de comunicaciones funcionaba sin problemas. Además podíamos empezar a hacer análisis posteriores a las pruebas, gracias a la potente herramienta que suponen los logs.

Pero con este simulador no se podía hacer ningún tipo de pruebas sobre el control que resultaran realistas. Para ello se necesitaba un modelo físico del helicóptero, que respondiera de una forma más parecida a la del verdadero HERO. Este modelo se encuentra incluido dentro del UAV Real Simulator, que implementa además un control LQR. Aunque éste no sea el tipo de control que actualmente implementa el DSP real,

las pruebas realizadas en simulación resultan muy prometedoras. De todas formas, nuestras experiencias de campo con el HERO real nos han demostrado que el control del helicóptero es lo suficientemente complejo y delicado como para no tomárselo a la ligera.

Durante las numerosas pruebas que se realizaron, ya fueran con el UAV Simulator o con el UAV Real Simulator, se fueron corrigiendo los fallos que surgían, con lo que finalmente se llegó al punto en el que el completo funcionamiento se encontraba libre de errores. El tener una estructura firme y sin fallos (ya sean de diseño o de código) será de vital importancia para las posibles mejoras futuras que marquen las sucesivas líneas de investigación. Algunos de estos futuros proyectos, que no se encontrarán con dificultades añadidas por fallos en la estructura, se proponen en la siguiente sección.

5.2 Posibles desarrollos futuros

Presentamos aquí algunas líneas a seguir que se creen interesantes para el futuro:

- Revisión de las estructuras de datos en general. Concretamente, una revisión de los tipos empleados en determinados campos sería de gran interés, ya que ahorraría la confusión que genera la demasiado frecuente conversión de datos mediante *casts*. Además muchas estructuras se encuentran obsoletas, ya que en la evolución del proyecto HERO se ha ido desestimando la utilización de muchos de sus campos. En el presente proyecto no se ha realizado tal revisión porque carece de sentido mientras que el DSP siga programado como en la actualidad. Recordemos que nuestro software no deberá ser capaz de distinguir entre el DSP Simulator y el DSP real.
- Revisión del protocolo de comunicaciones DSP↔PC. Además de por los motivos expuestos en el anterior punto, por un fallo concreto que se ha encontrado: cuando el tamaño del estado sobrepasa un límite determinado, el comportamiento en la transmisión del mismo es totalmente inesperado. El primer punto donde habría que corregir es el DSP real. Este fallo se ha comprobado, por ejemplo, al incluir las actuaciones PWM en la máscara de estados. Para máscaras que comprenden menos datos, como la utilizada actualmente, el funcionamiento es sin embargo perfectamente correcto.
- Sistema de transmisión de velocidades. Ya vimos el problema de sincronismo que teníamos asociado a la transmisión de velocidades, que resultaba en que era imposible determinar qué velocidad corresponde a qué *waypoint*. Esto sería fácilmente corregible mediante, por ejemplo, el encolado también de las velocidades en el DSP.
- Obtención de los parámetros del HERO para el modelo actual. Para lo cual se proponen una serie de experimentos en la sección 4.4.2. Con ello se dispondría ya de un modelo más fiel aún al HERO real, con lo que podríamos considerar más fiables las simulaciones de cara a estudiar el control.
- Simulación del control. Si se quiere seguir utilizando el control PID en el helicóptero real, sería interesante disponer de una implementación para la simulación de dicho control. Para ello sólo habría que crear una nueva clase de simulador muy parecido al

UAV Real Simulator que, con la misma interfaz y usando el mismo modelo de helicóptero, controlara en función del error con respecto a la consigna. Esto permitiría hacer pruebas previas a las de campo. Si por el contrario se optara por implementar un LQR en el DSP, se recomendaría realizar un mejor ajuste del controlador. El ajuste actual sacrifica velocidad de respuesta frente a estabilidad, pero es mejorable. En lo que respecta a la observación de los estados, no hay ningún problema ya que la mayoría de ellos son salidas directas de nuestros sensores.

- Simulación de los sensores. Actualmente no se realiza ninguna simulación de los sensores. Para obtener un modelo del robot más realista, sería necesario incluir algún modelo de los sensores, de forma que las salidas del modelo del helicóptero pasaran previamente por los sensores antes de llegar al controlador o al PC.