CAPÍTULO VI – Conclusiones y recomendaciones

1. La metodología de desarrollo de software de prototipos es muy adecuada para la realización de proyectos de robótica porque…
2. La tarea de desarrollar un cuadricóptero es multidisciplinaria.
3. La aparición y vertiginoso desarrollo de te cnologías de distribución libre como Arduino han permitido acelerar el tiempo de desarrollo en proyectos de robótica. En particular, la tarjeta Arduino Nano 3.0 utilizada en el presente trabajo, y basadas en el microcontrolador ATMEGA328 son confiables? (reliable), fáciles de programar, flexibles? en cuanto a configuración/conexión, y tienen una gran relación precio/calidad/tamaño buenísima para el cuadricóptero. Librerías y cosas desarrolladas previamente!!!!
4. La implementación de un sistema de control de velocidad angular permite realizar un control más robusto, preciso y rápido? (permite que el de posición angular converja más rápidamente).
5. No sólo la distribución en capas, sino también el manejo de la frecuencia de ejecución de los algoritmos de control, permite un control más robusto y “suave”, y una menor saturación de los actuadores, al seguir la dinámica propia del proceso a controlar? Aproximada.
6. En el desarrollo de circuitos de potencia es recomendable realizar un análisis, partiendo de la respuesta de la mínima combinación de componentes bajo las condiciones de trabajo esperadas, y aumentando la complejidad de forma incremental hasta llegar al ¿circuito esperado?. Esto permite identificar las características de funcionamiento particulares de cada componente, y, del conjunto de los mismos, que pueden servir para realizar un análisis exhaustivo con el fin de diseñar el ¿mejor circuito posible?.
7. Con una estimación de estado adecuada, un algoritmo PID, o varios, en capas, pueden permitir realizar un control satisfactorio de un cuadricóptero. Se comprueba la aseveración realizada en [Nadales 2009]
8. El filtro complementario implementado permite una estimación precisa, rápida, y suficientemente robusta para la estabilización de un cuadricóptero ¿sencillo? con acelerómetro y giroscopio. Además, es ¿fácil de implementar? Libera al usuario de la necesidad de entender otras técnicas más complejas?
9. Algo sobre las interfaces de comunicación desarrolladas. Las de comandos y telemetría fueron asíncronas, y las de apagado/encendido síncronas. ¿Qué implicó eso? ¿En qué ayudó?
10. La realización de pruebas limitando el movimiento del cuadricóptero en uno, dos, o tres ejes, permite realizar un estudio exhaustivo del rendimiento de los algoritmos de estimación de estado, control y comunicación del mismo, lo que sirve para identificar posibles problemas y mejoras.
11. El software a utilizar en robótica debe estar compuesto por módulos altamente desacoplados. Muchas validaciones, no se debe confiar en la comunicación.
12. Recomendable Control de velocidad de los motores.
13. Recomendable utilizar tecnología lo más moderna posible. Motores brushless, ESC e IMU!!!
14. Recomendable que todo cuadricóptero desarrollado esté cimentado sobre una capa de control de bajo nivel. Controlarlo es una tarea altamente compleja. La interfaz de comunicación que desarrollamos podría permitir que el arduino nano (bajo nivel) reciba comandos de una unidad de procesamiento que haga SLAM (alto nivel) por el puerto serial.

Mejorar muchísimo!!!