

CAPÍTULO 4

CONCLUSIONES

4.1 INTRODUCCIÓN

Los objetivos fijados en este proyecto afrontados con el método estándar de una simulación en tiempo real, nos habrían llevado a elegir soluciones difíciles de implementar y a la vez inadecuadas para el objetivo que estamos buscando (soluciones desproporcionadas a las exigencias).

Las soluciones adoptadas en este proyecto, intentan utilizar la solución más fácil del problema. Así pues, la elección de utilizar el RT Blockset y el FSController, ambos disponibles en ambiente Matlab, van justo en esta dirección.

Destacaremos la importancia de utilizar Matlab, ya que se trata de un ambiente relativamente poco costoso en comparación con otras herramientas usadas en el ámbito aeroespacial. Se trata de un instrumento muy usado y extendido en el campo de la investigación (sobretudo universitaria) y de la simulación en general.

Para la realización de este proyecto hemos partido del modelo de mini-helicóptero proporcionado por la Universidad EAFIT de Medellín (Colombia). Dicho modelo nos ha sido muy útil, pero a la hora de abordar nuestro cometido se han encontrado una serie de problemas, los cuales son debidos al hecho de haber intentado aplicar soluciones complejas a un problema simple como el que se nos presenta.

4.2 PROBLEMÁTICA Y SOLUCIONES

A continuación se mencionan los problemas que se han ido obteniendo.

En primer lugar, el problema de la simulación en tiempo real se resuelve mediante la elección del RT Blockset, del cual se ha hablado ya en capítulos precedentes.

Después, hubo que “modificar” el bloque de interfaz con el joypad, para así poder utilizar el bloque proporcionado por Aerosim para la conexión con un joystick.

Luego, se tuvo que modificar a su vez la interfaz de conexión con el MSFS: Simulink realiza un envío de datos de tipo asíncrono mediante una conexión TCP/IP; dichos datos van llegando y encolándose en una cola de datos gracias al FSController,

el cual se encargará de transmitir dichos datos al MSFS mediante una comunicación de tipo IPC. De esta forma se consigue enviar datos al MSFS sin hacer que el Simulink tenga que esperar a que el simulador vaya “recogiendo” dichos datos (ya que dicha espera por parte de Simulink no haría posible una simulación en tiempo real).

Por otra parte estaba el problema de la estabilidad del helicóptero. Tal y como estaban configurados los parámetros del modelo (controladores), nuestro modelo era inestable. Con el joystick estamos cambiando la velocidad, y el modelo, tal y como estaba configurado al principio, funciona sólo hasta un cierto valor de velocidad. Si se sobrepasaba dicho valor límite de velocidad, nuestro helicóptero presenta un comportamiento extraño. Los PID que había en nuestro modelo no funcionaban bien, así que han sido configurados con los valores proporcionados por uno de los artículos de la Universidad de Medellín (Colombia). Dicho modelo, funcionaba bien para pequeñas variaciones de *actitud* pero para una velocidad superior a un cierto valor, deja de funcionar. Con el PID (control proporcional), cuanto más inclinamos el joy, más se inclina el helicóptero. Pero si se inclina demasiado, comienza a acelerarse, de manera que superamos el valor de velocidad a partir del cual nuestro sistema pasa a ser inestable. Para solucionar este problema de estabilidad, se coloca antes un controlador Fuzzy, el cual no permitirá sobrepasar un cierto valor de velocidad. El Fuzzy tiene como entrada la variación de posición (gradientes de posición) y luego gestiona él mismo la velocidad sin dejar que éste supere el valor a partir del cual nuestro modelo comenzaría a comportarse en un modo extraño. Los problemas del PID se observan dando pequeñas variaciones a la entrada.

Otro problema relacionado con la estabilidad sería la situación inicial de la cual partía nuestro modelo inicial. Éste partía desde una situación de no equilibrio (por lo que se han tenido que modificar las condiciones iniciales).

Además, se observó que se partía de una cota distinta respecto del valor de *throttle*, por lo que el helicóptero tardaba en estabilizarse (así que se le ha hecho partir con una cota correcta y con un valor estable de *throttle*).

Todos los problemas anteriores han sido resueltos y, mediante la solución adoptada, se han obtenido unas prestaciones y resultados más que adecuados respecto a las exigencias de una simulación en tiempo real.

BIBLIOGRAFÍA

[1] ADI, “Speeding Aircraft Development from Preliminary Design through Certification”, white paper published in the ADI technical library, <http://www.adi.com/>, 2004

[2] James S., “Simulation-centric processes for aerospace”, published in <http://www.embedded.com>, 2005

[3] Ledin J., “Simulation Takes Off with Hardware”, published in <http://www.embedded.com>, 2002

[4] Sanchez, E.N. Becerra, H.M. Velez, C.M. , “Combining fuzzy and PID control for an unmanned helicopter”, *Information Sciences: an International Journal*, Volume 177 , Issue 10 May 2007, ISSN:0020-0255, Elsevier Science Inc., New York, NY, USA

[5] V. Gavrillets, B. Mettler and E. Feron, "Dynamic model for a miniature aerobatic helicopter", MIT-LIDS report, no. LIDS-P-2580, 2003.

[6] Wikipedia Community, “Ring (Computer Security)”, [http://en.wikipedia.org/wiki/Ring_\(computer_security\)](http://en.wikipedia.org/wiki/Ring_(computer_security))

[7] Wikipedia Community, “Microsoft Windows”, http://en.wikipedia.org/wiki/Microsoft_Windows

[8] Wikipedia Community, “Operating Systems”, http://en.wikipedia.org/wiki/Operating_system

[9] Wikipedia Community, “Computer Multitasking”, http://en.wikipedia.org/wiki/Computer_multitasking

[10] The Mathworks, *Real Time Workshop User's Guide*, The MathWorks, Inc., 2007

[11] The Mathworks, *Real Time Windows Target User's Guide*, The MathWorks, Inc., 2007

[12] The Mathworks, *xPC User's Guide*, The MathWorks, Inc., 2007

[13] Leonardo Daga, “RT Blockset”, <http://digilander.libero.it/LeoDaga/Simulink/RTBlockset.htm>

[14] Unmanned Dynamics, *Aerosim Blockset User's Guide* version 1.2, <http://www.u-dynamics.com>

[15] Leonardo Daga, “FS Controller”,
<http://digilander.libero.it/LeoDaga/Simulink/FSController.htm>

[16] Leonardo Daga, “Struttura generica di un sistema di controllo aeronautico”,
<http://digilander.libero.it/LeoDaga/SistemaAeroStruct.htm>

AGRADECIMIENTOS

Por último, querría dedicar unas palabras de agradecimiento a todos aquellos que me han apoyado durante estos años de carrera y, más recientemente, durante la realización de este proyecto.

En primer lugar, quiero dar las gracias a mis padres y hermanos, los cuales me han ayudado siempre a mantener viva la motivación, incluso en los momentos más difíciles.

Gracias a Sergio Esteban Roncero, por su paciencia infinita y por su orientación durante la realización de este proyecto como inmejorable profesor de la Escuela de Ingenieros. Gracias por la ilusión que has puesto siempre en este proyecto, Sergio.

Y por último, aunque no por ello menos importante, querría mencionar y agradecer su colaboración a Leonardo Daga, Doctor Ingeniero Aeronáutico (Roma). Sin duda, sin él no habría sido posible la realización de este proyecto. Ha sido la luz que me ha guiado durante todo el camino, así como un verdadero apoyo. Por eso, Leonardo, no tengo palabras para agradecerte toda la ayuda incondicional y continua que me has ofrecido durante todo este tiempo. Gracias de todo corazón por confiar en mí y por ayudarme a cumplir una de las metas de mi vida.