



Prototipo de una plataforma móvil de bajo coste para simulación de vuelo de alto realismo

J.J. Ortega, M. Sigut*

Departamento de Ingeniería Informática y de Sistemas, Universidad de La Laguna, Avda. Francisco Sánchez, s/n. La Laguna, 38204. Santa Cruz de Tenerife, España.

Resumen

En este artículo se presenta un prototipo de plataforma móvil para simulación de vuelo de alto realismo. La parte central de este prototipo, que hemos denominado Albatros, es la maqueta hecha a mano. Esta maqueta es una réplica a escala de la plataforma a tamaño real que los autores pretenden construir en un futuro próximo. La maqueta está basada en la plataforma Stewart-Gough, y se ha equipado con actuadores neumáticos y potenciómetros magnéticos como sensores de posición. La plataforma móvil recibe la información de vuelo proveniente de un simulador de vuelo comercial en forma de la posición de referencia para los seis actuadores. Así, la plataforma móvil puede seguir los movimientos del avión simulado gracias a la implementación de seis controladores proporcionales-integrales. La interfaz entre el ordenador de simulación y la maqueta es una placa Arduino Mega. La simulación de vuelo de alto realismo se ha pretendido alcanzar gracias, por un lado, a un seguimiento lo más fiel posible de las consignas generadas por el software de simulación de vuelo y, por otro, a un retardo entre los movimientos del avión simulado y la maqueta tan pequeño como sea posible.

Palabras Clave:

Simulación de vuelo, Arduino, control PID, bajo coste, alto realismo, sistema multiplataforma.

1. Introducción

La meta final en la que trabajan los autores del proyecto cuyos resultados parciales se plasman en este artículo es la construcción de un simulador de vuelo móvil a tamaño real, en el que una persona pueda experimentar las sensaciones de un piloto de avión comercial. Lo que se presenta en este artículo es una maqueta a escala del prototipo a tamaño real. Las características fundamentales que deben caracterizarla, y en base a las cuales se ha diseñado y construido, son tres: alto realismo, bajo coste y ángulos de alabeo y cabeceo de, al menos, 30° en su máxima inclinación. Además, se pretende que el sistema desarrollado sea multiplataforma y esté basado en la medida de lo posible en software libre y de código abierto. La motivación para acometer este proyecto es clara. Hoy en día existe un incuestionable y creciente interés en la simulación de vuelo de aviación comercial. Este interés no sólo es compartido por compañías aéreas y fabricantes de aviones, sino también por personas sin relación con el mundo de la aviación que quieren usar este tipo de sistemas con una finalidad exclusivamente lúdica.

Así las cosas, el prototipo mencionado anteriormente tiene un gran potencial. Está claro que la aplicación más importante e inmediata es ser usado como método de entrenamiento y evaluación para pilotos de aerolíneas. Esto se consigue gracias a su movilidad, lo que lo distingue claramente de otros trabajos en los que se diseñan y construyen simuladores para entrenamiento que carecen de plataforma móvil, como en (Wu and Sun, 2013). En este sentido, (Bürki-Cohen et al., 2011) reflejan la controversia existente en un análisis sobre la literatura relativa a simulación de vuelo. Éste es un debate interesante respecto a en qué medida la movilidad de un simulador de vuelo es crucial para su uso como sistema de formación de pilotos comerciales. De hecho, estos autores hablan literalmente de la evidencia empírica de la necesidad de disponer de sistemas dotados de movimiento para una formación efectiva mediante simulación y, además, de verificar que existe una correcta correspondencia entre el simulador y el avión.

Muchas publicaciones confirman esta tesis desde diferentes puntos de vista, por ejemplo en (Bürki-Cohen, 2004), (Bussolari and Lee, 1986), (Caro, 1979), (Go et al., 2003), (Grant et al., 2006), (Hall, 1989), (Levison and Junker, 1978), (Vaden and Hall, 2005) y (Van der Pal, 1999). Es particularmente interesante la opinión de la ALPA en (ALPA, 2007) y las consideraciones

* Autor en correspondencia

Correo electrónico: marsigut@ull.es (M. Sigut)

hechas por la RAA en (RAA, 2008). Toda vez que la importancia del movimiento en los simuladores de vuelo es contrastada, el coste de dichos sistemas es otra variable a tener en cuenta. También se afirma en (Bürki-Cohen et al., 2011) que un entrenamiento apropiado es esencial para obtener y mantener pilotos con un alto nivel, y que esto requiere de recursos adecuados que puedan estar disponibles para todas las aerolíneas alrededor del mundo. Como consecuencia de ello, todos los esfuerzos para el desarrollo de dispositivos de bajo coste son bienvenidos.

Antes del desarrollo del simulador de vuelo a tamaño real y bajo coste, los autores han considerado que era muy recomendable diseñar y construir el sistema descrito en este artículo, donde la plataforma móvil es un modelo a escala del de tamaño real.

La literatura relativa a la simulación de vuelo es muy extensa. En lo que se refiere a los simuladores comerciales y militares, la mayoría están equipados con plataformas de movimiento que generalmente están basadas en variantes de la plataforma Stewart de seis cilindros. Un ejemplo de este tipo de estructuras es el Simulador de Movimiento Vertical (Ames Technology Capabilities and Facilities, 2008), que opera en el Centro de Investigación Ames de la NASA. Se trata de un diseño que permite un cambio rápido de las diferentes cabinas de los aviones. Otro ejemplo es el simulador Desdémona (AMST, 2015), fruto de la colaboración entre el Laboratorio Nacional de Física de Alemania (TNO) y la empresa austriaca AMST Systemtechnik. Por sus características, Desdémona proporciona una simulación de alta fidelidad y alto realismo para vuelo suborbital. Por su parte, el Advanced Concept Flight Simulator (National Aeronautics and Space Administration, 2014) de la NASA simula la cabina de un avión de transporte de pasajeros comercial. En su interior se encuentran los mandos de una aeronave común, que están ligados al entorno virtual de las pantallas del simulador y a los movimientos de la plataforma de seis grados de libertad que sirve como base para simular el movimiento de la nave. Esta herramienta proporciona una gran flexibilidad para simular diferentes modelos de aviones con entornos de varios aeropuertos y condiciones meteorológicas. Por citar un último ejemplo, una empresa líder a nivel mundial en el entrenamiento en aviación es Flight Safety (FlightSafety International, 2015), que ha fabricado varios simuladores para diferentes tipos de aviones, tanto civiles como militares. Para la impartición de sus cursos, esta empresa hace uso de diversos tipos de tecnología. Concretamente, en la parte de simulación de movimiento se emplea como base de las cabinas la plataforma Stewart Gough. Los ejemplos que se acaban de citar corresponden a simuladores de alta gama que, por tanto, tienen un coste elevado. En la literatura referente a simuladores de vuelo de bajo coste, como el que se presenta en este artículo, también encontramos numerosas contribuciones. Aunque la estructura de Stewart-Gough es, probablemente, la opción más extendida, como en (Álvarez et al., 2009), hay otras alternativas. Así, en (Shiga et al., 2011) los autores proponen un mecanismo de trípode en paralelo en un intento de mejorar el espacio de movimiento y los ángulos de inclinación que ofrece la plataforma Stewart-Gough convencional (Hitaka et al., 2009). En (Izaguirre et al., 2011) se presenta un esquema desacoplado de control articular para un simulador de movimiento industrial de tres grados de libertad accionado

por actuadores neumáticos. El DLR Robot Motion Simulator (Bellmann et al., 2011) es una plataforma novedosa para simulación de movimiento con siete grados de libertad. Este simulador emplea un eje lineal de 10 m para incrementar su espacio de trabajo y proporcionar una experiencia de simulación mejor que la ofrecida por diseños previos de simuladores de movimiento con cinemática serial o paralela. Otro ejemplo es la plataforma propuesta por (Pradipta et al., 2013). Está basada en una plataforma Stewart con un cilindro extra en el centro y actuadores neumáticos. Los autores proponen dos métodos de control que permiten eliminar las restricciones cinemáticas causadas por la configuración de siete cilindros. Los seis de fuera se controlan de acuerdo a trayectorias de movimiento, mientras que el cilindro central funciona como un actuador de fuerza para vencer el peso de la plataforma. En otros casos, los prototipos han sido diseñados únicamente con fines educativos, como en (Woodrow et al., 2013). En este caso, el prototipo desarrollado tiene en común con el que se presenta en este artículo el hecho de ser de bajo coste y el uso de hardware libre, concretamente la placa Arduino Mega. Sin embargo, también hay algunas diferencias importantes. Una de ellas es el hecho de que el sistema aquí descrito permite elegir el modelo de aeronave deseada, entre otras características de vuelo, sin modificar el hardware, cosa que no es posible en el trabajo de Woodrow y colaboradores.

2. Materiales y Métodos

En la figura 1 se muestra una vista esquemática de todo el sistema desarrollado. Como se puede observar, los principales elementos que componen el prototipo son los siguientes: un ordenador (en el prototipo a tamaño real habrá tres monitores para una vista panorámica de la simulación de vuelo), una placa Arduino acompañada de la electrónica diseñada para la adaptación de señales, y la maqueta de la plataforma móvil, que es el elemento central. Estos dispositivos de hardware, junto con el software instalado en el ordenador, se describen a continuación.

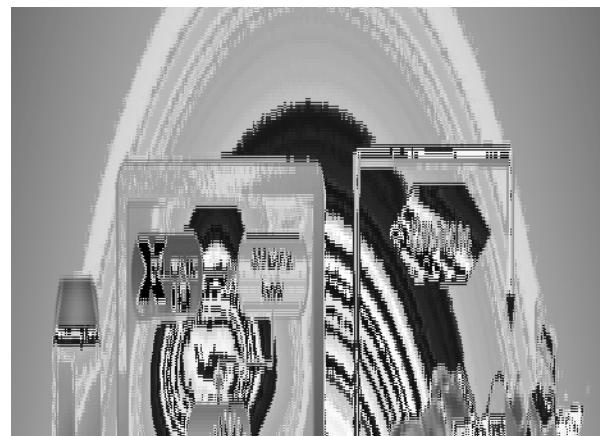


Figura 1: Elementos del sistema.

2.1. Simulación, exportado de datos de vuelo y software de control

En este proyecto se han utilizado tres paquetes de software: X-Plane 10, BFF Motion Driver y Scilab. A continuación se describe la función de cada uno de ellos dentro del sistema.

X-Plane 10: En el prototipo que los autores han diseñado y construido, los movimientos de la plataforma son los indicados por el software de simulación de vuelo. Por lo tanto, este software es responsable de la simulación de la aeronave y los sistemas que con los que está equipada, además de las condiciones externas (tráfico aéreo, meteorología, etc.) y, en general, de todo lo relativo a la navegación virtual. Además de esto, el software debe enviar a la pantalla las imágenes de navegación y exportar los datos de vuelo (alabeo, cabeceo y ángulos de giro, así como la aceleración vertical y horizontal de la aeronave, entre otros) con el fin de que puedan ser utilizados para determinar los movimientos de la plataforma móvil.

Hoy en día existe una gran variedad de paquetes de software de simulación de vuelo. Algunos de ellos son multiplataforma y, en algunos casos, también son gratuitos y de código abierto. Con el fin de seleccionar el más adecuado para este proyecto, se evaluaron las siguientes características: coste, realismo, velocidad, facilidad de conexión a dispositivos externos, documentación disponible y frecuencia con que se actualiza. Después de una selección previa, los autores evaluaron detalladamente tres paquetes de software de simulación de vuelo: Microsoft Flight Simulator X, Flight Gear y X-Plane 10, siendo este último el elegido. Las razones que justifican la elección de este simulador de vuelo civil creado por Austin Meyer se exponen a continuación. Hoy en día X-Plane 10 es muy utilizado por aficionados a la construcción de cabinas caseras de aviones. Su uso para el entrenamiento instrumental de pilotos en Estados Unidos ha sido autorizado por la FAA, lo que significa que una hora de vuelo con este simulador es equivalente a una hora de vuelo real. El propósito de X-Plane 10 es que el usuario experimente durante el vuelo simulado sensaciones lo más reales posibles. Esto se consigue gracias, entre otras cosas, a su gran precisión, a la extensa variedad de aviones simulados que ofrece, al elevado número de aeropuertos, aeródromos y helipuertos disponibles y, finalmente, a una recreación de la Tierra muy lograda. En la figura 2 se puede ver el detalle de una cabina de un Boeing 777 en X-Plane 10.

La última versión de este simulador es la 10.40. Es compatible con Windows, Mac OS X y Linux. El software está diseñado para ser conectado con cualquier tipo de hardware, por lo que el usuario puede diseñar o conectar sus propios dispositivos de hardware y hacerlos interactuar con el simulador. Es altamente configurable y puede importar y exportar datos desde o hacia cualquier tipo de dispositivo externo. Incluso puede conectar con un auténtico GPS comercial para simular la navegación a través de este dispositivo, de la misma manera que se haría en cualquier avión real. Los gráficos proporcionan un gran realismo y permite conectar varios monitores o proyectores, facilitando las futuras ampliaciones y mejoras de la plataforma. Por otra parte, existe una gran cantidad de escenarios gratuitos y aviones construidos por los usuarios de todo el mundo, así

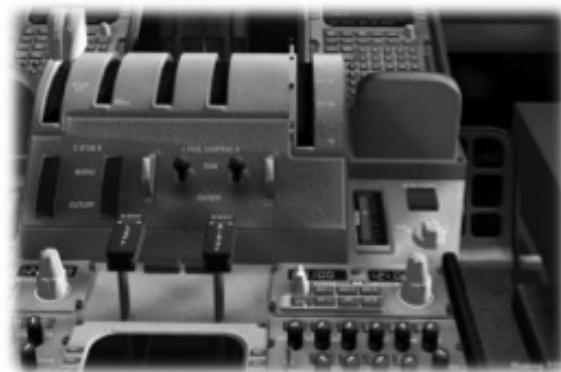


Figura 2: Parte de la cabina de un Boeing 777 en X-Plane 10.

como plugins y complementos que pueden ser creados por el usuario, proporcionando al prototipo un gran número de nuevas funcionalidades. Además, este software de simulación tiene la ventaja de que posibilita programar nuevos módulos bajo la base sólida que ofrecen sus desarrolladores.

X-Plane 10 no es de código abierto. A pesar de que no es libre, el usuario tiene la posibilidad de volar por un tiempo ilimitado de forma gratuita, siempre que no utilice un joystick por un período de tiempo largo, dado que en modo demo se desconecta pasados unos minutos (cosa que no es objeto de este proyecto). En caso de que el lector considere que el código abierto es una característica vital, existen otros simuladores de vuelo disponibles. Para ello, una buena alternativa es Flight Gear, que ha sido mencionado anteriormente. Flight Gear es de código abierto y software libre, a pesar de lo cual fue descartado por los autores de este proyecto en favor de X-Plane 10 debido principalmente a dos razones: al mayor realismo de X-Plane 10 y al hecho de que este último está autorizado por la FAA.

BFF Motion Driver: Una vez que el software de simulación fue correctamente configurado, tuvimos que enfrentarnos al problema de la extracción de información de vuelo de X-Plane 10 y su envío al controlador de la plataforma. Hay algunos paquetes de software comercial capaces de hacerlo. Uno de ellos es CHcomm, compatible con MS Flight Simulator, a pesar de que también se puede utilizar junto con X-Plane. De hecho, CHcomm puede extraer una gran cantidad de información de X-Plane y enviarla a través del puerto serie. Sin embargo, se descartó para este proyecto debido a la falta de continuidad en su desarrollo, ya que desde 2004 no ha experimentado ninguna mejora. Además de esto, el uso de CHcomm sobrecargaba la placa Arduino, que tenía que recibir los datos de la aeronave y hacer la transformación a la posición de la plataforma Stewart. Con el fin de evitar estos inconvenientes en el prototipo, se evaluó un paquete alternativo de software llamado BFF Motion Driver, que fue elegido como la mejor opción. La principal virtud de BFF Motion Driver, cuya última versión es la 3.0, lanzada al mercado el 08 de enero de 2014, es el hecho de estar diseñado específicamente para la exportación de los datos del simulador a una plataforma de Stewart. De esta manera,

controla los seis grados de libertad de la plataforma y trabaja extrayendo paquetes de datos desde el simulador y procesándolos con el fin de resolver la cinemática inversa de la plataforma. BFF Motion Driver indica la posición final de cada actuador de acuerdo con la información de vuelo procedente de X-Plane 10. Desde el punto de vista del control activo de la plataforma Stewart, esto tiene importantes implicaciones dado que las posiciones de los actuadores calculadas por BFF Motion Driver (ver Fig. 3.) actúan como valores de referencia en el cierre de los seis lazos de control SISO.

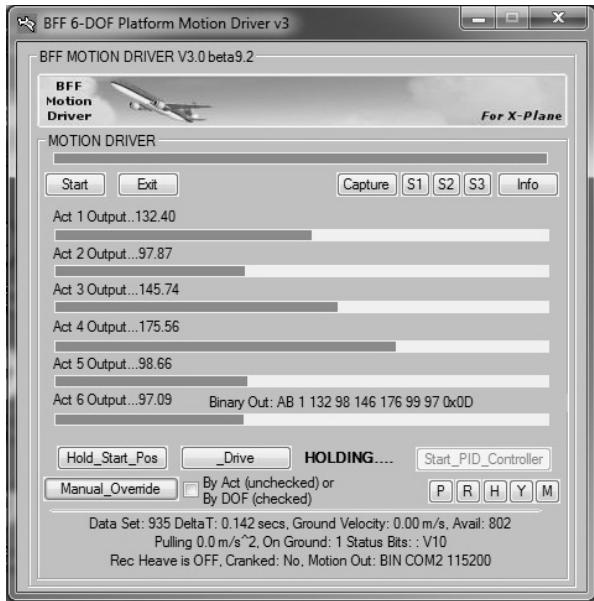
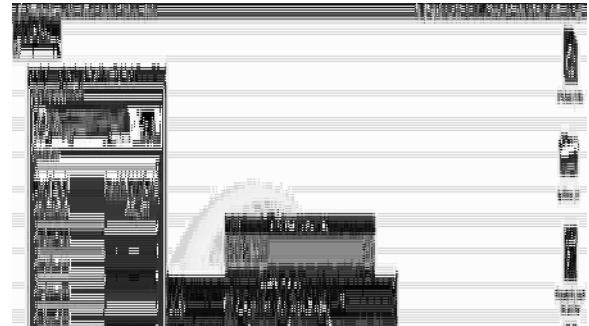


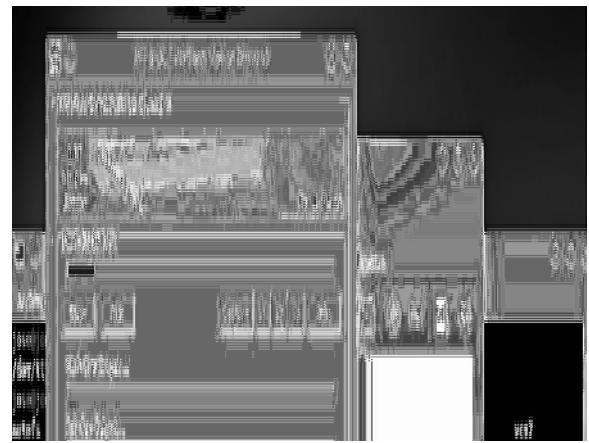
Figura 3: Captura de pantalla de BFF Motion Driver mostrando las posiciones de referencia para los diferentes actuadores.

En lo que se refiere al alto realismo, una de las características que se exige al prototipo presentado en este artículo, es bien sabido que pasa, entre otras cosas, por la percepción de las fuerzas producidas por la gravedad y las aceleraciones experimentadas por el usuario del simulador de vuelo. Para lograrlo existen varias técnicas para la generación de consignas basadas en la percepción continua de movimiento incluso en el caso de que el recorrido de la plataforma sea limitado o cuando ésta está parada. Una de estas técnicas es el algoritmo Washout, tal y como se explica en (Arai et al., 2012), (Chunping et al., 2012), (Sung-Hua and Li-Chen, 2011) y (Van Heerden et al., 2011). El BFF Motion Driver utiliza algoritmos Washout para la generación de las consignas a los actuadores. Concretamente, incorpora este tipo de filtros tanto en los movimientos de alabeo, cabeceo y guiñada, como incluso en la simulación del movimiento vertical del avión (heave).

A pesar de todas las ventajas mencionadas anteriormente, BFF Motion Driver no es ni de código abierto ni libre. De hecho, hoy en día sólo es posible utilizarlo de forma gratuita por un período de 15 minutos. Una vez que este tiempo ha terminado, es necesario cerrarlo y ejecutarlo de nuevo para usarlo otros 15 minutos. Aunque esta limitación puede resultar incómoda para el usuario, en el caso de vuelos cortos no es algo relevante.



(a) Captura de pantalla de BFF Motion Driver ejecutándose en una máquina Mac OS X.



(b) Captura de pantalla de BFF Motion Driver ejecutándose en una máquina Linux (Kubuntu 13.10).

Figura 4: Capturas de pantalla.

Además, BFF Motion Driver sólo funciona bajo Windows. Sin embargo, como se puede ver en las figuras 4a y 4b, es posible ejecutarlo en una máquina Mac OS X o Linux mediante el uso de una aplicación de software llamada Wine, que es gratuita y de código abierto. Con el fin de evitar el uso de Wine y de paso eliminar la restricción en el tiempo de uso que se señaló con anterioridad, los autores tienen en mente asumir el desarrollo de una aplicación de software libre y de código abierto multiplataforma que reemplazaría al BFF Motion Driver.

Scilab: Es un software multiplataforma, libre y de código abierto elegido para llevar a cabo el control de los movimientos de la plataforma, en un intento de reproducir los del avión simulado. Los controladores implementados en Scilab corresponden a una estrategia PI (Proporcional-Integral). La expresión de esta acción de control se muestra a continuación:

$$u(t) = K_p \left(e(t) + \frac{1}{T_i} \int e(t) dt \right) \quad (1)$$

$u(t)$ representa el comando enviado al actuador en función de los parámetros del controlador PI (K_p y T_i) y del error del sistema $e(t)$, es decir, la diferencia entre el valor de salida deseada

do y el que es medido por el sensor. Con esta estrategia de control se pretende que el sistema alcance el valor de consigna, prescindiendo de la acción derivativa por la ausencia de sobrepasamiento en el transitorio. El prototipo presentado en este artículo tiene seis actuadores y el mismo número de sensores. Así, se trata de un sistema multivariable en el que el bajo nivel de interacción permite llevar a cabo el control activo plataforma móvil mediante el cierre de seis lazos de control SISO. Como se dijo anteriormente, BFF Motion Driver convierte la posición de la aeronave simulada en las posiciones correspondientes de los seis cilindros neumáticos de la plataforma Stewart-Gough. Una vez que el BFF Motion Driver envía a Scilab estas seis posiciones, que representan las alturas de referencia para los actuadores, estas magnitudes se comparan con los valores reales medidos por los sensores con el fin de calcular los errores y, de acuerdo con la expresión en (1), se obtienen los comandos. Los controladores PI que se han implementado en la práctica para el cierre de los seis lazos de control se han ajustado experimentalmente por separado empleando el método de ensayo y error, obteniendo los valores finales que se muestra en la Tabla 1.

Tabla 1: Valores de los parámetros de los controladores PI

Actuador	K_p	$T_i(s)$
1	5	0.9
2	3.5	1.5
3	3.5	1.5
4	5	0.9
5	3.5	0.9
6	3.5	0.9

La forma en la que se ha etiquetado a los actuadores se muestra en la figura 5. Las diferencias entre los valores de los parámetros K_p y T_i que se muestran en la Tabla 1 se deben principalmente a la construcción artesanal y a los diferentes ajustes del prototipo.

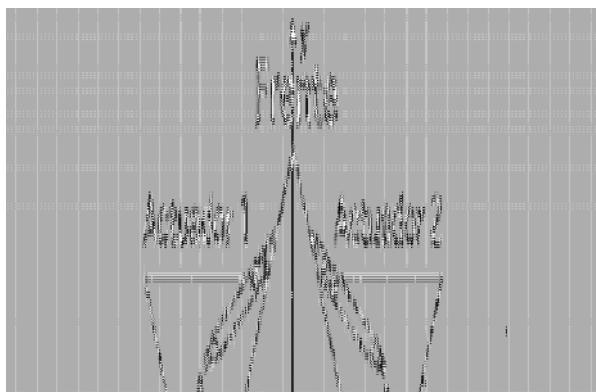


Figura 5: Vista simplificada en planta del prototipo incluyendo el etiquetado de los actuadores.

Nótese que los autores han optado por el uso de Scilab para la implementación de los seis controladores SISO, en lugar de hacerlo directamente en la placa Arduino, debido a la mayor facilidad que esta herramienta ofrece de cara a la represen-

tación gráfica de las variables involucradas en los experimentos de control (respuesta de los actuadores, señales de error y comandos), lo que resulta especialmente útil en el proceso de ajuste de los controladores. A pesar de que el Scilab es prescindible una vez que los controladores PI han sido ajustados con precisión, se ha mantenido en esta primera versión del prototipo. Los autores tienen la intención de pasar a una arquitectura distribuida, con una placa Arduino por actuador, en un futuro próximo. En esta nueva implementación cada Arduino recibirá la consigna directamente del BFF Motion Driver y calculará el comando que se debe aplicar a su actuador.

Con la configuración actual, el X-Plane 10 exporta 20 datagramas (bloques de bytes de información con los datos de vuelo) por segundo. El BFF Motion Driver envía a Scilab 720 paquetes de 10 bytes (conteniendo, entre otra información, las consignas correspondientes a los seis actuadores) por segundo. Por su parte, Scilab se comunica a esta misma velocidad con el Arduino Mega, sólo que los paquetes de datos que envía Scilab son de 6 bytes, correspondientes a los comandos a aplicar a los 6 actuadores.

2.2. El PC

X-Plane 10, BFF Motion Driver, Wine y Scilab se ejecutan en un ordenador personal con las siguientes características:

- Procesador: Intel i5 a 2,80 GHz de cuatro núcleos.
- Memoria: 4 Gb.
- Disco duro: 110 GB SSD (Solid State Drive) con un sistema operativo de 64 bits (Windows 7).
- Tarjeta gráfica: AMD Radeon HD6800 de 2GB con soporte para 3 monitores. Estos tres monitores permiten una visión panorámica de la vista simulada, y cobran especial relevancia en el prototipo a tamaño real.

Debe tenerse en cuenta que, como se comentó anteriormente, gracias a la utilización de Wine podría reemplazarse este ordenador por un equipo con Mac OS X o por una máquina Linux, dotando al prototipo de una gran versatilidad. De hecho, los experimentos de control se han realizado usando un PC con Windows, una máquina Linux y otra con Mac OS X, dando buenos resultados en todos los casos.

2.3. La maqueta de la plataforma móvil

La maqueta de la plataforma móvil es el elemento central del prototipo, como se mostró en la figura 1. Se ha diseñado y construido enteramente por los autores, siendo su aspecto final el que se muestra en la figura 6.

Se trata de una maqueta a escala 1:10, donde para el dimensionamiento del prototipo a tamaño real se ha llegado a un compromiso entre que sea lo suficientemente amplio como para que el usuario se sienta cómodo, y lo suficientemente compacto como para que no sea preciso un espacio excesivamente grande para su instalación.

Para entender algunos de los detalles con respecto a la construcción de la plataforma, es importante conocer las principales especificaciones de diseño, que son las siguientes:

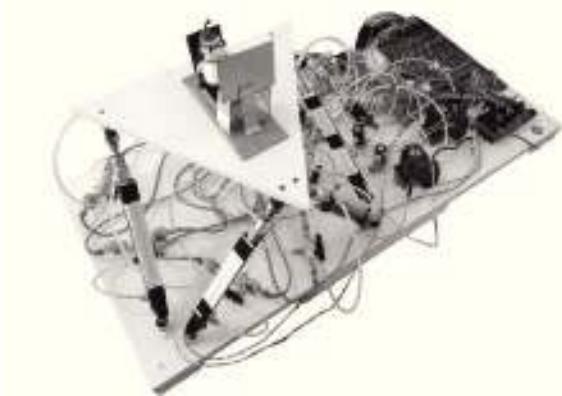


Figura 6: La maqueta final diseñada y construida para la simulación de vuelo.

- Bajo coste.
- Alto realismo, entendido como un retardo lo más pequeño posible entre el movimiento del avión en el software de simulación y los movimientos reales de la plataforma.
- Ángulos de alabeo y cabeceo de al menos 30° como máxima inclinación.

Como se puede ver en la figura 6, la maqueta es una estructura mecánica basada en la plataforma Stewart-Gough. Básicamente consiste en dos plataformas hechas de madera de aglomerado, siendo la inferior un rectángulo fijo de 80 cm de longitud por 40 cm de ancho. La plataforma superior, que es la que se mueve, consiste en un triángulo equilátero de 30 cm de lado. Al hacer los agujeros de perforación en ambas plataformas para el posicionamiento de los actuadores, su inclinación hubo de ser cuidadosamente calculada. Se colocaron roscas de 6 mm de diámetro en los agujeros de perforación de la plataforma inferior para alojar las rótulas que proporcionarían el movimiento deseado a la plataforma superior. Concretamente, si consideramos los ejes cartesianos que se muestran en la figura 5, la plataforma móvil de la maqueta tiene un recorrido en el z de 16.5 cm, un recorrido en el eje x de 26.0 cm (13.0 cm hacia cada lado), y un recorrido en el eje y de 24.5cm (12.5 cm hacia el frente y 12.0 cm hacia atrás).

En cuanto al sistema de actuación, los autores decidieron utilizar dispositivos neumáticos lineales, concretamente un modelo cuya referencia es CD85N16-150-B de SMC Pneumatics, que se muestra en la parte izquierda de la figura 7. La tecnología neumática, utilizada en dispositivos similares como el descrito en (Izaguirre et al., 2011), y concretamente este modelo, se eligió debido a su versatilidad, precio, disponibilidad y el hecho de permitir cumplir con las especificaciones de movilidad exigidas. Por lo tanto, la carrera del cilindro (150 mm) y la presión que soporta (hasta 1 MPa) permiten que la plataforma superior pueda tener la inclinación prevista. Se trata de un actuador de doble efecto activado por válvulas de 3 vías que hacen que los actuadores puedan extenderse, retraerse o detenerse en una determinada posición. La referencia de las válvulas de 3 vías es

SYJ3343-5LOU-Q de SMC. La salida de aire es común para todas sus vías y la fuente de alimentación es de 24 VCC. Este modelo tiene también un supresor de picos de tensión y soporta una presión de 0,7 MPa.

Antes de las válvulas de tres vías se colocaron válvulas proporcionales para regular el flujo de aire que llega a cada actuador. El fabricante de estas válvulas con referencia PVQ31-5G-23-01N es también SMC. Entre otras características, estos dispositivos se alimentan con 24v CC y soportan una presión de 0.35MPa. Por último, el compresor utilizado para proporcionar la presión de aire a los actuadores neumáticos tiene un volumen de 4L y una presión máxima de 8 bares, aunque este prototipo funciona a 3 bares.

Para medir la posición de los actuadores se optó por sensores MagnetoPot de SPECTRASYMBOL. El tiempo de vida del dispositivo MagnetoPot excede de un millón de desplazamientos y su resistencia interna es 10 KΩ. Además, tiene una carrera de 150 mm y la fiabilidad de las mediciones que hace depende de la potencia del imán externo y la distancia entre el imán y el sensor del potenciómetro MagnetoPot. Cuanto más potente es el imán, más precisos y menos oscilantes son sus medidas. Los imanes que vienen de fábrica dentro de los vástagos de los actuadores sugirieron el uso de este tipo de dispositivos que, gracias a la colocación de los potenciómetros sobre la superficie de los cilindros, realizan el mismo cometido que una resistencia variable. El imán del actuador atrae una esfera ferromagnética que se encuentra dentro del sensor, de tal manera que dependiendo de la posición de la primera, el valor de la resistencia es uno u otro. Gracias a ello, el sensor MagnetoPot, que se muestra en la parte derecha de la figura 7, actúa como un potenciómetro magnético.



Figura 7: A la izquierda, un actuador neumático como los instalados en la plataforma. A la derecha, el sensor MagnetoPot.

Una vez colocados los actuadores y sensores en su posición final y conectadas las placas de circuito que se describirán posteriormente, se procedió a comprobar su correcto funcionamiento. En primer lugar, los seis actuadores fueron activados de forma individual con el fin de comprobar que todos ellos eran capaces de llegar a una posición final deseada. Los autores aprovecharon este momento para ajustar las válvulas reguladoras de caudal de aire de los seis actuadores, de tal manera que todos ellos tuvieran aproximadamente los mismos flujos de entrada de aire y, por consiguiente, velocidades de elongación y retracción similares. En este punto, se envió la misma posición de referencia a los seis actuadores para comprobar si la plataforma superior alcanzaba una posición final paralela a la base, manteniendo una orientación lo más horizontal posible durante todo el movimiento. Una vez que los dispositivos de instrumentación pasaron esta prueba, se comprobó la conexión de todos

los elementos que componen la maqueta, (ver figura 1) y se probó todo el sistema permitiendo que los datos del simulador de vuelo llegaran a la plataforma, con el fin de que ésta se moviera de acuerdo con la información de vuelo.

En la figura 8 se puede observar la respuesta en lazo cerrado de uno de los seis actuadores de la maqueta, concretamente el actuador 2, junto con la consigna y el comando aplicado. Las gráficas correspondientes a los otros actuadores son muy similares y no se presentan únicamente por limitaciones de espacio. La línea de guiones y puntos representa el comando aplicado, la línea continua representa la posición de referencia obtenida a partir de un vuelo simulado y la línea discontinua la posición real (para mayor claridad, ver leyenda). Al usar este tipo de consigna se pretende mostrar el rendimiento del sistema en lazo cerrado en una situación realista. En el eje vertical, el rango de valores para las posiciones es de 0 (actuador completamente retraído) a 1024 (actuador completamente extendido). El rango de valores para el comando es -1024 a 1024, a pesar de que se utiliza el valor absoluto para regular el flujo de aire que llega al actuador, mientras que el signo se utiliza para decidir el sentido de la válvula de 3 vías y, en consecuencia, si el actuador se retrae o se extiende.

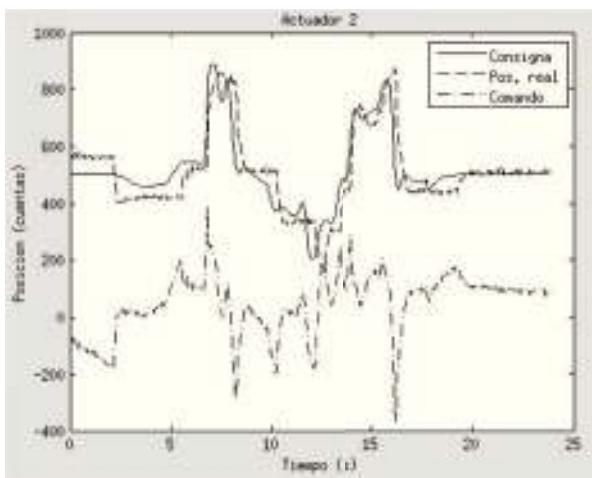


Figura 8: Respuesta en lazo cerrado del actuador 2 de la plataforma móvil ante consigna de vuelo con controladores PI.

En la figura 9 se muestra en trazo discontinuo la respuesta del mismo actuador frente a una consigna tipo escalón (en trazo continuo). El efecto más interesante que se observa en esta figura es que, a pesar de la inclusión de la acción de control integral (nótese que para la obtención de las figuras 8 y 9 los parámetros de los controladores PI son los que se muestran en la Tabla 1), el actuador no alcanza la posición de consigna. El offset que se observa se debe principalmente al hecho de que cuando la posición real del actuador está muy cerca de la deseada, el flujo de aire es tan pequeño que no es capaz de mover el actuador. Los autores no han hecho un gran esfuerzo en tratar de eliminar este error en la etapa actual del proyecto, ya este efecto no tiene una influencia significativa sobre el propósito del presente trabajo. Téngase en cuenta que, como se dijo claramente en la introducción, el objetivo principal del prototipo descrito aquí es

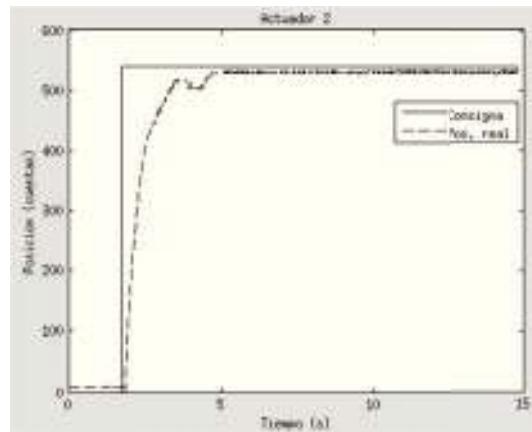


Figura 9: Respuesta en lazo cerrado del actuador 2 de la plataforma móvil ante consigna escalón con controladores PI.

ser capaz de seguir los movimientos de un avión simulado para dar al usuario las mismas sensaciones que las experimentadas por un piloto. Es evidente que tales sensaciones no dependen tanto de alcanzar exactamente la misma posición que el avión simulado, como del seguimiento fiel de los movimientos. Como se puede ver especialmente en la figura 8, este objetivo se consigue en el prototipo diseñado. En segundo lugar, los sensores MagnetoPot utilizados para medir la posición de la plataforma móvil son dispositivos analógicos, cuya salida experimenta pequeñas variaciones incluso cuando el actuador se mantiene en la misma posición. Estas leves fluctuaciones de las medidas realizadas por los sensores, que se observan en las figuras 8 y 9, se reflejan también en las señales de error y, en consecuencia, en las acciones de control, lo que también se aprecia en la figura 8. Este comportamiento oscilatorio se puede observar en el video disponible en http://youtu.be/_Q_MXp6J6sU. Estas limitaciones impuestas por los actuadores y sensores actuales se tendrán que resolver en el prototipo a tamaño real mediante el uso de otro tipo de dispositivos con el fin de que los movimientos sean más suaves.

2.4. La electrónica del prototipo

Debido a la imposibilidad de conectar el PC directamente a los actuadores y sensores de la plataforma móvil, ha sido necesaria la utilización de un microcontrolador. Se estudiaron dos posibilidades: PIC y Arduino. Aunque ambos cumplen con los requisitos de este proyecto, los autores escogieron finalmente Arduino debido a su creciente aplicación tanto en electrónica de consumo como doméstica y, sobre todo, por el hecho de ser un proyecto de hardware y software libre. De esta manera, el usuario del prototipo tiene libre acceso a toda la documentación sobre la plataforma Arduino, incluyendo el esquema de conexiones de hardware. Esto, junto con la gran cantidad de librerías y de fragmentos de código disponibles, hace que Arduino sea una solución más apropiada, especialmente si el usuario del prototipo no está familiarizado con la electrónica.

Las características más relevantes del Arduino Mega 2560 R3, basado en el chip Atmega 1280, se muestran en la Tabla 2:

Tabla 2: Características del chip Arduino Mega 2560

Arduino Mega 2560	
Memoria Flash	128KBytes
Memoria SRAM	8KBytes
Memoria EEPROM	4KBytes
Entradas/Salidas	54 (15PWM)
Número de entradas A/D	16
¿Soporta SPI?	Si
Número de EAUSART	4
¿Módulo USB?	Si

Como se puede observar en la Tabla 2, y en contraste con la mayoría de PIC, Arduino Mega tiene salidas PWM. Esto es relevante porque una de las cuestiones que los autores han pretendido evaluar con esta maqueta de cara a la construcción del prototipo a tamaño real, es la tecnología de actuación. Mientras que los actuadores de la plataforma móvil de la maqueta son cilindros neumáticos, el prototipo a tamaño real estará muy probablemente equipado con actuadores eléctricos, por lo que la disponibilidad de salidas PWM será un factor a tener en cuenta.

Tanto las válvulas de 3 vías como las proporcionales han requerido del diseño de dos tarjetas de circuito para adaptar las señales procedentes del chip del Arduino Mega. Las placas de circuito diseñadas se componen básicamente de optoacopladores, transistores y resistencias. En la figura 10 se muestra un esquema simplificado de las comunicaciones entre los principales dispositivos que componen el prototipo.

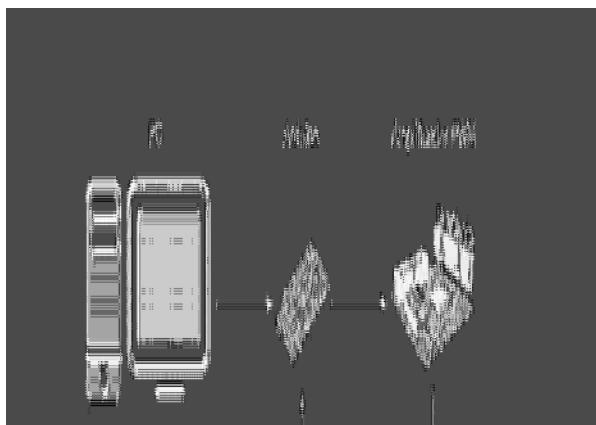


Figura 10: Comunicación entre los dispositivos que componen el prototipo.

3. Alto realismo basado en la minimización del retardo

En este trabajo los autores plantean la consecución de la especificación de realismo bajo la perspectiva de la minimización del lapso de tiempo que transcurre entre que el avión simulado realiza un movimiento y la plataforma móvil de la maqueta lo reproduce.

3.1. Medición del retardo

La característica de alto realismo es crucial para el éxito y la aplicabilidad del prototipo móvil en la simulación de vuelo, sobre todo de cara a la formación de pilotos comerciales. La necesidad de movimiento en la simulación de vuelo ha quedado debidamente documentada en la introducción. Además, con el propósito de que la persona que está utilizando el simulador experimente las mismas sensaciones que tendría si estuviera en el avión, es esencial que, además de la inclusión de algoritmos de Washout para la generación de consignas (implementados ya en el BFF Motion Driver), el retardo del sistema sea tan pequeño como sea posible. Dado que el movimiento del prototipo presentado en este artículo es ya una realidad, es necesario medir el retardo del sistema con el fin de determinar el grado de realismo que tiene. Obsérvese que los autores consideran el retardo como el tiempo transcurrido entre el momento en que el usuario observa en el monitor un movimiento determinado de la aeronave simulada y el momento en que la plataforma móvil del prototipo comienza a moverse para reproducirlo. BFF Motion Driver tiene acceso a los datos de vuelo justo en el momento en que X-Plane 10 los genera, por lo que mientras BFF Motion Driver convierte estos datos en las referencias de los actuadores, X-Plane 10 actualiza la información gráfica que se muestra en el monitor. La suposición de que el tiempo que BFF Motion Driver emplea en el cálculo de las consignas para los actuadores es más pequeño que el tiempo necesario para actualizar la salida gráfica del simulador de vuelo parece ser bastante realista. Con todo ello, podemos considerar el retardo del sistema como la suma de las contribuciones que se enumeran a continuación:

1. La generación de consignas a los actuadores (el BFF Motion Driver emplea para ello el modelo cinemático de la plataforma Steward).
2. Los cálculos efectuados por Scilab para la obtención de los comandos a aplicar a los actuadores de acuerdo a la estrategia PI.
3. La comunicación de los comandos a la placa Arduino mediante el puerto serie.
4. La generación de potencia por parte de los circuitos electrónicos que dan las consignas a las electroválvulas.
5. La respuesta de las electroválvulas, que presentan su propia dinámica.
6. La transmisión de la energía hasta los cilindros neumáticos mediante el correspondiente flujo de aire.
7. La respuesta del émbolo para llegar a la posición de referencia, exhibiendo su propia dinámica.

Con el fin de medir el retardo de sistema, se examinaron dos posibilidades. La primera consistió en medir los retrasos correspondientes a las contribuciones individuales anteriores y luego estimar el retardo total como la suma de todos de ellos. El problema de esta alternativa es que algunas de las contribuciones son difíciles de medir. La otra posibilidad consistía en tratar de

obtener el retardo del sistema por medio de una sola medición. Debido al hecho de que estamos interesados principalmente en la demora global, esta última parecía ser la mejor opción. Por otra parte, el primer método induce un mayor error al sistema, ya que sería la suma de los errores individuales debido a cada medición parcial. Por lo tanto, con el objetivo de medir el retardo global del sistema, así como el debido a la parte mecánica del prototipo, los autores han llevado a cabo los experimentos que se describen a continuación.

3.2. Experimentos

Con el propósito de medir con la mayor precisión posible el tiempo transcurrido entre el momento en que BFF Motion Driver calcula una posición de referencia y el instante en que la plataforma móvil del prototipo empieza a moverse para alcanzar dicha consigna, se realizó un primer experimento, que se describe a continuación. Este experimento consistió en colocar un haz láser apuntando hacia una de las esquinas de la plataforma móvil del prototipo. Se colocó un fototransistor justo detrás de la esquina de la plataforma en la misma dirección del haz de láser, de tal manera que al elevarse la plataforma el haz láser impactaba sobre el fototransistor. Además, se configuró un Arduino Due de tal manera que en el momento de activar de un botón (que enviaba a la plataforma la orden de elevarse), comenzaba la cuenta de un cronómetro justo al mismo tiempo que se enviaba una consigna dada a la plataforma.

Inicialmente, la plataforma está en su posición inferior, con los seis actuadores retraídos, como se muestra en la imagen izquierda de la figura 11. Cuando se pulsa un botón, BFF Motion Driver genera una consigna escalón exactamente al mismo tiempo que arranca un cronómetro en el Arduino Due. Es en ese momento cuando la plataforma móvil comienza a subir para llegar a la consigna, permitiendo que la luz láser impacte sobre el fototransistor ocasionando la desactivación del cronómetro, tal como se muestra en la imagen de la derecha de la figura 11. De este modo, el tiempo medido (en milisegundos) se muestra en pantalla gracias al software para el puerto serie de Arduino.



Figura 11: A la izquierda, el prototipo está listo para realizar la medida. En esta configuración, la esquina de la plataforma móvil evita que el haz láser impacte sobre el fototransistor. A la derecha, la plataforma móvil comienza a moverse y el fototransistor recibe la luz láser.

El experimento que se acaba de describir ha permitido medir el retardo global del prototipo y cuantificar la influencia del punto 2 del desglose mostrado en el apartado 3.1. No obstante, es bien sabido que las contribuciones más importantes a dicho retardo son las debidas a la dinámica propia de los sistemas

Tabla 3: Medidas experimentales del retardo del prototipo

	Tiempo de retardo (ms)
Medida 1	117
Medida 2	123
Medida 3	159
Medida 4	109
Medida 5	122
Medida 6	130
Medida 7	94
Medida 8	82
Medida 9	81
Medida 10	89
Tiempo medio de retardo:	110.6

mecánicos, es decir, las válvulas y los cilindros neumáticos. Para cuantificarlas se ha realizado un segundo experimento para el que se ha empleado un Arduino Mega para medir los tiempos de respuesta y un pulsador que, al ser presionado, da inicio a la cuenta. En ese mismo momento se activa un relé que activa directamente la válvula proporcional del actuador, además de la válvula todo/nada correspondiente. En cuanto el actuador comienza a subir, el cronómetro se para. Los tiempos de respuesta medidos con este segundo experimento se presentan en la Tabla 4. Comparando los resultados que en ella se muestran con los datos que se presentan en la Tabla 3, encontramos que la parte mecánica del prototipo es la causante del 63.6 % del retardo global del prototipo.

Tabla 4: Medidas experimentales del tiempo de respuesta del sistema actuador (válvula + cilindro neumático)

	Tiempo de respuesta (ms)
Medida 1	69
Medida 2	69
Medida 3	69
Medida 4	69
Medida 5	70
Medida 6	70
Medida 7	71
Medida 8	71
Medida 9	72
Medida 10	74
Tiempo medio de respuesta:	70.4

4. Resultados y discusión

Con el fin de determinar si el retardo del prototipo presentado en este artículo es aceptable, es esencial comparar los tiempos medidos en el primer experimento descrito en la sección 3.2 (ver tabla 3) con la referencia adecuada. Para encontrar una referencia válida es preciso aclarar que los autores consideran que el retardo es aceptable si el usuario del prototipo no percibe una latencia entre los movimientos del avión simulado y los de la plataforma. Por lo tanto, bajo esta perspectiva, el retardo del sistema determina la sensación de realismo experimentado.

Con el propósito de encontrar una buena estimación del retardo mínimo que es perceptible por los seres humanos, se ha hecho una revisión exhaustiva de la literatura existente. Entre todos los estudios encontrados, los autores consideran que la aportación más relevante para esta aplicación es el libro de James L. Kent y, más concretamente, su quinto capítulo dedicado al tópico relativo a los "Límites de la percepción humana", (Kent, 2010). La discusión de James L. Kent acerca de los resultados de solapamiento de imágenes resulta de gran utilidad de cara a lograr el objetivo que los autores de este artículo persiguen. Dicho objetivo podría ser descrito de la siguiente manera: imagínese un observador externo que observa simultáneamente tanto el avión simulado como los movimientos de la plataforma móvil. Aceptando que existirá un cierto retraso entre estas dos imágenes, la pregunta es: ¿a partir de qué tiempo de retardo éste es percibido por el observador externo? Así, a pesar de ser contextos muy diferentes, existe una clara conexión entre la pregunta que los autores tratan de responder y los resultados presentados en (Kent, 2010). Estos resultados tienen que ver con la idea de ver una película y ser incapaz de detectar la transición entre un fotograma y el siguiente, siendo percibido como algo continuo. Ésta es la razón por la cual la relación entre este estudio y el problema tratado aquí se hace más evidente con la inserción de la figura observador externo. Según (Kent, 2010), la percepción humana de la realidad se refresca totalmente más o menos una vez cada 77 ms. Tomando este valor como referencia, el retardo del prototipo medido en el experimento descrito en la sección 3.2 puede ser analizado correctamente.

A la luz de los resultados que se muestran en la última columna de la tabla 3 (obtenidos usando Scilab), es evidente que el retardo del prototipo está por encima de la referencia proporcionada por (Kent, 2010). También llama la atención la poca repetibilidad del experimento realizado, ya que en la muestra 3 se mide un retardo de 159 ms, mientras que en la 9 el retardo medido es de 81 ms. Esto es achacable al diseño del propio experimento de medición descrito al comienzo de la sección 3.2, en cuya mejora estamos trabajando. Sin embargo, teniendo en cuenta que se trata de la primera versión del prototipo Albatros y que aún hay muchos aspectos susceptibles de mejora, los autores consideran que los resultados obtenidos son prometedores.

5. Conclusión

Los autores se sienten bastante satisfechos con el rendimiento de la maqueta a escala que se ha descrito en este artículo. Los objetivos fijados en un principio se han alcanzado razonablemente en esta primera versión a escala del Albatros. Para entenderlo es importante evaluar el grado de cumplimiento de los requisitos impuestos. En primer lugar, después de las pruebas realizadas con el prototipo se comprobó que la inclinación máxima en todo su recorrido alcanza los 35° sin poner en riesgo en ningún momento su integridad. Esto supera las especificaciones de diseño de la sección 2.3. En segundo lugar, el requisito de alto realismo basado en un retardo lo suficientemente pequeño no se ha alcanzado satisfactoriamente y debe ser mejorado en futuras versiones de esta maqueta. Esto será crucial de cara al diseño del prototipo de tamaño real. No obstante, a pesar de

que el retardo medido está por encima de la referencia de los 77 ms, éste es lo suficientemente pequeño como para dar al usuario una alta sensación de realismo. Así se puede comprobar en el vídeo disponible en http://youtu.be/_Q_MXp6J6sU. Además, los autores habrán de plantearse cuestiones como la medición de las aceleraciones a que estaría sometida la cabeza del usuario de la plataforma. Por último, teniendo en cuenta el precio final de la maqueta, 2.295,87 Euros, los autores consideran que el requisito de bajo coste también ha sido satisfecho teniendo en cuenta las prestaciones del prototipo. Es importante notar que este precio corresponde exclusivamente a la maqueta y, en consecuencia, excluye el software de simulación de vuelo (X-Plane 10), el PC, el compresor de aire y la fuente de alimentación. En cuanto a las conclusiones que la maqueta presentada en este artículo nos ha permitido extraer de cara al diseño y la construcción del prototipo a tamaño real, así como a las principales propuestas de mejora y líneas abiertas en las que los autores ya están trabajando, cabe destacar lo siguiente. En lo que al hardware se refiere, los autores se plantean a corto plazo dotar a la maqueta de seis placas Arduino Nano junto a los actuadores, en un intento de reducir el retardo del sistema y mejorar sus prestaciones. Con esta nueva arquitectura, cada Arduino recibirá la consigna para su actuador directamente desde el BFF Motion Driver (o equivalente) y calculará el comando correspondiente, prescindiendo por tanto del Scilab. Asimismo, se estudiará el uso de actuadores eléctricos en el prototipo a tamaño real. Como es bien sabido, éstos permiten alcanzar una precisión más alta que la ofrecida por los actuadores neumáticos, presentan una mayor facilidad de instalación y una vida útil larga sin apenas precisar mantenimiento. Además, las pérdidas de aire que se producen en el sistema neumático hacen que haya que rellenar la cámara de aire cada vez que se activa, ocasionando ciertos retardos que, en la aplicación que se plantea en este trabajo, pueden llegar a ser importantes. A pesar de estas ventajas, la sustitución de la tecnología neumática por la eléctrica en el prototipo a tamaño real estará supeditada a que el tiempo de retardo del sistema permita alcanzar la sensación de alto realismo que debe caracterizar al prototipo Albatros. Como se comentó en la sección 3.2, el 63.6 % del tiempo de retardo del prototipo se debe a la parte mecánica del mismo (ver Tablas 3 y 4). Por tanto, habrá que evaluar si la pérdida de prestaciones que en este sentido se producirá al sustituir los actuadores neumáticos por dispositivos eléctricos es compensada o no por la modificación en la arquitectura de control mencionada anteriormente. En otro orden de cosas, también se estudiará el uso de un tipo diferente de sensores de posición, en un intento de evitar los problemas causados por los sensores MagnetoPot. A nivel software, los autores planean desarrollar una aplicación de software que reemplace el BFF Motion Driver y, por lo tanto, evite el uso del programa Wine. Por otra parte, se intentará mejorar la estrategia de control, usando controladores más avanzados que permitan alcanzar una mejora del rendimiento del prototipo.

English Summary

A low-cost mobile prototype for high-realism flight simulation.

Abstract

A low-cost mobile prototype for high-realism flight simulation is presented in this article. The most relevant part of this prototype that has been called Albatros is the hand-made mobile platform. The authors have the intention of constructing a real-size prototype based on the mock-up described here. This mock-up is based on a Stewart Gough platform and equipped with pneumatic actuators and magnetic potentiometers as position sensors. The mobile platform receives the flight information coming from a commercial flight simulator in the form of the reference position for the six actuators. Thus, the mobile platform can track the simulated aircraft movements thanks to the implementation of six proportional-integral controllers. An Arduino Mega circuit board is the interface between the computer and the mock-up. The high-realism flight simulation is achieved by means of the prototype motion and the short delay measured between the simulated aircraft movements and the platform ones.

Keywords:

Flight simulation, Arduino, PID control, low-cost, high-realism, multiplatform system.

Referencias

- Air Line Pilots Association (ALPA), 2007. Safety Committee Statement of Position: The Need for Motion in Flight Simulation.
- Álvarez, C., Saltaren, R., Aracil, R., García, C., 2009. Concepción, desarrollo y avances en el control de navegación de robots submarinos paralelos: El robot Remo-I. *Revista Iberoamericana de Automática e Informática Industrial* 6(3), 92-100.
- Ames Technology Capabilities and Facilities. 2008. VMS - Vertical Motion Simulator. Recuperado de <http://www.nasa.gov/centers/ames/research/technology-onepagers/vms.html>.
- AMST. (2015). Desdemona - The revolution in simulation. Recuperado de <http://www.amst.co.at/en/training-simulation-products/desdemona/>.
- Arai, S., Kondo, H., Goto, H., Tanaka, Y., 2012. Evaluation of motion with washout algorithm for flight simulator using tripod parallel mechanism. In Proc. of the 19th International Conference Mechatronics and Machine Vision in Practice, Auckland.
- Bellmann, T., Heindl, J., Hellerer, M., Kuchar, R., Sharma, K., Hirzinger, G., 2011. The DLR robot motion simulator Part I: Design and setup. In Proc. of IEEE International Conference on Robotics and Automation, Shanghai.
- Bürki-Cohen, J., Go, T.H., Chung, W.W., Schroeder, J., 2004. Simulator platform motion requirements for recurrent airline pilot training and evaluation, Final Report.
- Bürki-Cohen, J., Sparko, A.L., Bellman, M., 2011. Flight simulator motion literature pertinent to airline-pilot recurrent training and evaluation. In Proc. of AIAA Modeling and Simulation Technologies Conference, Portland.
- Bussolari, S.R., Lee, A.T., 1986. The effects of flight simulator motion on pilot performance and simulator acceptability in transport category aircraft. Massachusetts Institute of Technology/NASA Ames Research Center.
- Caro, P.W., 1979. The relationship between flight simulator motion and training requirements. *Human Factors* 4, 493-501.
- Chunping, P., Ying, L., Jianmin, L., Yongjun, G., 2012. A time varying washout approach for flight simulation hexapod motion system. In Proc. of IEEE International Conference on Computer Science and Automation Engineering, Zhangjiajie.
- FlightSafety International. (2015). FlightSafety Simulators and Training Technology. Recuperado de http://www.flightsafety.com/fs_simulation_landing.php.
- Go, T.H., Bürki-Cohen, J., Chung, W.W., Schroeder, J., Saillant, G., Jacobs, S., Longridge, T., 2003. The effects of enhanced hexapod motion on airline pilot recurrent training and evaluation. In Proc. of AIAA Modeling and Simulation Technologies Conference, Austin.
- Grant, P.R., Yam, B., Hosman, R., Schroeder, J.A., 2006. Effect of simulator motion on pilot behavior and perception. *J. Aircr.* 43(6), 1914-1924.
- Hall, J.R., 1989. The need for platform motion in modern piloted flight training simulators. In Royal Aerospace Establishment, Tech Memo FM 35, Bedford.
- Hitaka, Y., Tanaka, Y., Ichiryu, K., 2009. Motion analysis of tripod parallel mechanism. *Artif. Life and Robot.* 14(4), 494-497.
- Izaguirre, E., Hernández, L., Rubio, E., Prieto, P.J., Hernández, A., 2011. Control desacoplado de plataforma neumática de 3-GDL utilizada como simulador de movimiento. *Revista Iberoamericana de Automática e Informática Industrial* 8(4), 345-356.
- Kent, J.L., 2010. Limits on human perception, in: Psychedelic information theory. Shamanish in the age of reason. PIT Press / Supermassive, LLC, Seattle WA, pp. 37-48.
- Levison, W.H., Junker, A.M., 1978. A model for the pilot's use of motion cues in steady-state roll-axis tracking tasks. In Proc. of AIAA Flight Simulation Technologies Conference, Arlington.
- National Aeronautics and Space Administration. (2014). CVSRF Advanced Concepts Flight Simulator. Recuperado de <http://www.aviaionsystemsdivision.arc.nasa.gov/facilities/cvrsrf/acfs.shtml>.
- Pradipita, J., Klunder, M., Weickgenannt, M., Sawodny, O., 2013. Development of a pneumatically driven flight simulator stewart platform using motion and force control. In Proc. of International Conference on Advanced Intelligent Mechatronics, Wollongong, NSW.
- Regional Airline Association (RAA), 2008. Government research indicates simulator motion adds training complexity - RAA Recommends Operational Testing to Validate Effectiveness of Non-Motion Platforms, Regional Airline Industry White Paper.
- Shiga, Y., Tanaka, Y., Goto, H., Takeda, H., 2011. Design of a six degree-of-freedom tripod parallel mechanism for flight simulators. *Int. J. Automation Technol.* 5(5), 715-721.
- Sung-Hua, Ch., Li-Chen, F., 2011. An optimal washout filter design with fuzzy compensation for a motion platform. In Proc. of the 18th IFAC World Congress, Milano.
- Vaden, E.A., Hall, S., 2005. The effect of simulator platform motion on pilot training transfer: A meta-analysis. *Int. J. Aviat. Psychology* 15(4), 375-393.
- Van der Pal, J., 1999. The effect of simulator motion on parameter training for F-16 pilots, engineering psychology and cognitive ergonomics: transportation systems, medical ergonomics and training. Edited by D. Harris, Ashgate, Oxford, England, pp. 267-275.
- Van Heerden, A., Lidbetter, R., Liebenberg, L., Mathews, E.H., Meyer, J.P., 2011. Development of a motion platform for an educational flight simulator. *Int. J. of Mechanical Engineering Education* 39(4), 306-322.
- Woodrow, P.M., Tischler, M.B., Hagerott, S.G., Mendoza, G.E., 2013. Low cost flight-test platform to demonstrate flight dynamics concepts using frequency-domain system identification methods. In Proc. of AIAA Atmospheric Flight Mechanics Education Conference, Boston.
- Wu, L., Sun, Y.P., 2013. Development of a low-cost flight simulation training device for research and education. In Proc. of the 2nd International Conference on Intelligent Technologies and Engineering Systems, Kaohsiung, Taiwan.