Capítulo 10
Pruebas y Resultados

En este capítulo:

Pruebas y Resultados

En este capítulo se presentan las pruebas más importantes realizadas para comprobar el correcto funcionamiento del sistema implementado junto con los resultados obtenidos.

Medida estática de estabilidad del AHRS

El objetivo de esta prueba es medir la desviación estándar en el pitch, roll y yaw. Recordando lo que significa la desviación estándar, también llamada desviación típica, es una medida de dispersión usada en estadística que nos dice cuánto tienden a alejarse los valores concretos del promedio en una distribución. En resumen, este parámetro nos viene a indicar que el 68% de las muestras se encuentran en el rango [+1σ, -1σ] para una variable aleatoria que sigue una distribución normal como se muestra en la Figura 1.

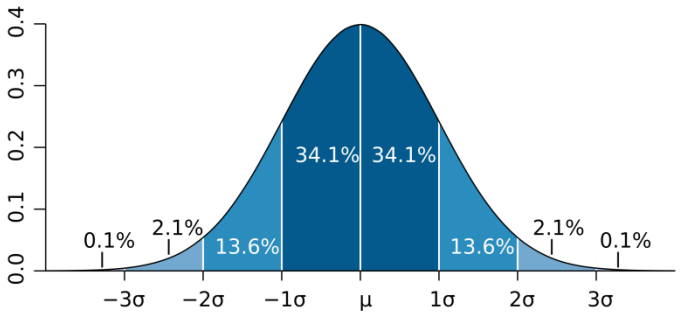

Figura 1 – El área marcada en azul oscuro indica una desviación estándar respecto a la media en ambos lados. Para una distribución normal, representa el 68.27% del conjunto.

El teorema del límite central dice que la distribución de un promedio de muchas variables aleatorias independientes, idénticamente distribuidas tiende hacia una distribución normal con la famosa forma de campana de gaus. Por ello se tomaron muestras durante 24h, cada 20ms variando la temperatura exterior. Estas muestras se introdujeron en Matlab donde se

hizo la estimación de la desviación estándar utilizando la función std. Estas pruebas se hicieron solamente en estático. Para hacer medidas en dinámico habría que montar una plataforma estable, y disponer de un AHRS que presente una mejor estimación o un sistema de calibración óptico de alta velocidad para poder medir la desviación del AHRS implementado. Las medidas en estático son muy sencillas de hacer. Se fijó la PCB a una superficie estable, que no sufra movimientos ni vibraciones. Para esta prueba se utilizó el suelo y se buscó una zona donde no pasaran los cables de electricidad para distorsionar lo menos posible el campo magnético. Se tomaron medidas durante 24 horas y la temperatura de la habitación se fue variando con el aire acondicionado. En la Figura 2 se muestra cómo se fijó la placa.

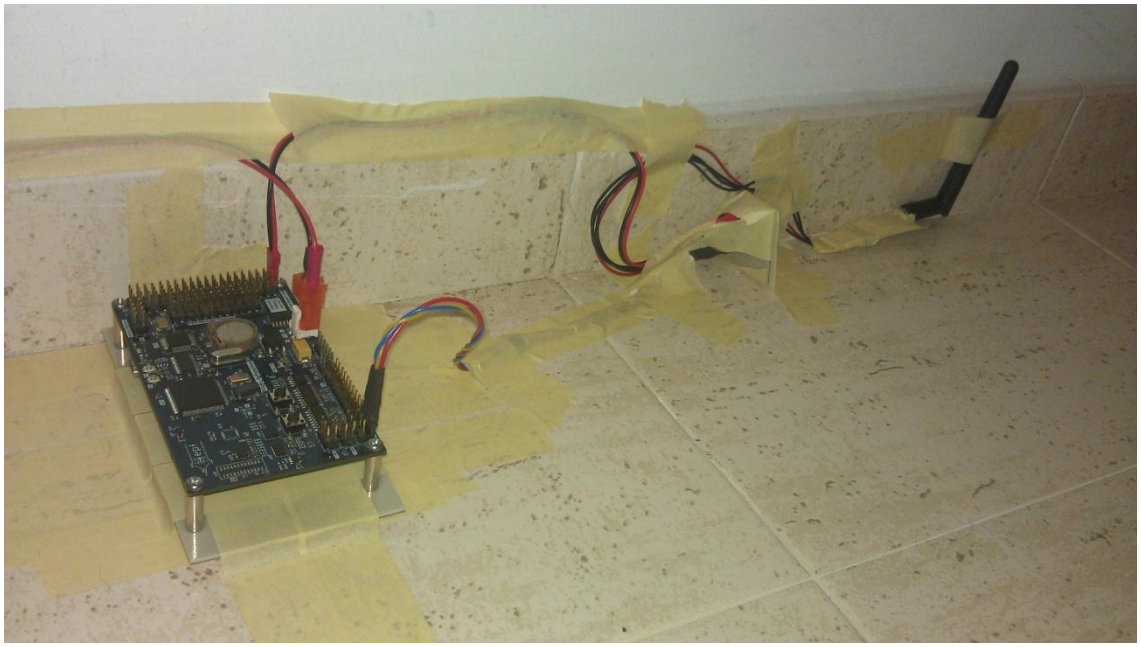

Figura 2 – Prueba en estático del AHRS.

Después de capturar las 4 millones de muestras aproximadamente, se introducen en Matlab y se calcula la desviación estándar. En la Tabla 1 se resumen los resultados obtenidos.

Tabla 1 – Resultados obtenidos en la desviación estándar de la estimación del yaw, pitch y roll.

Parámetro	Desviación estándar (1σ)
Roll	<0.05º
Pitch	<0.05º
Yaw	<0.10º

En la Tabla 2 se muestra una comparativa entre los resultados obtenidos y varios productos existentes en el mercado.

- VECTOR, es el mejor producto de la empresa española UAV Navigation. Se trata de un piloto automático de altas prestaciones para UAVs.
- Sebastian Madgwick para su tesis doctoral en la universidad XXX realizó un AHRS y en su algoritmo consiguió los resultados bastante buenos.
- La empresa XSense posee una amplia gama de IMU en su catálogo de productos. El más importante es el MTi-G-700 GPS/INS donde implementa un filtro Kalman extendido para la estimación de la *attitude*.
- EPF AHRS es el algoritmo propio implementado para este proyecto y como se puede ver supera en estabilidad a los otros productos existentes.

Tabla 2 – Comparativa con otros productos del mercado.

Parámetro	EPF AHRS (1σ)	UAV Navigation VECTOR	PhD Sebastian Madgwick	XSense MTi-G-700
Roll	**0.05º**	0.5º	0.594º	0.2º
Pitch	**0.05º**	0.5º	0.497º	0.2º
Yaw	**0.1º**	1.0º	1.0º	1.0º

Desde la Figura 3 a la Figura 8 se presentan las muestras capturadas en función de la temperatura y la distribución que presentan.

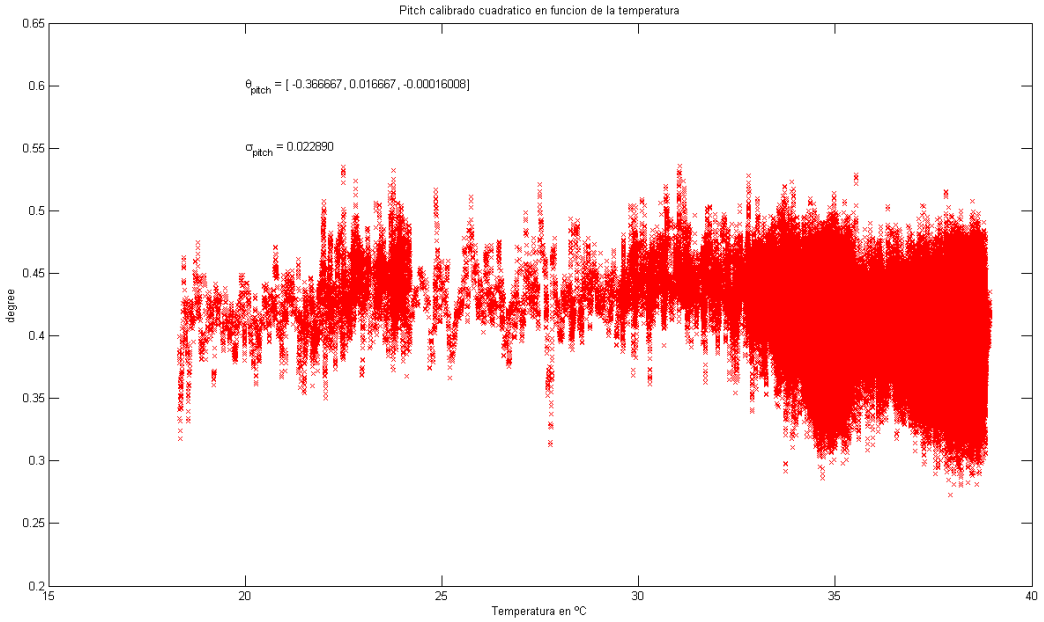

Figura 3 – Muestras capturadas del Pitch en función de la temperatura.

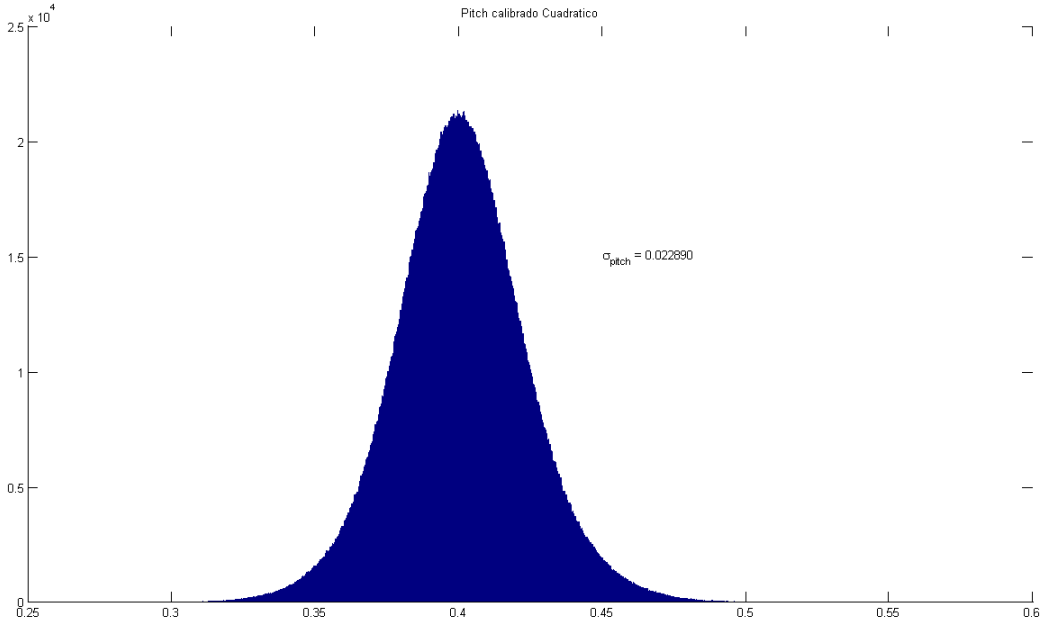

Figura 4 – Distribución de las muestras capturadas del Pitch.

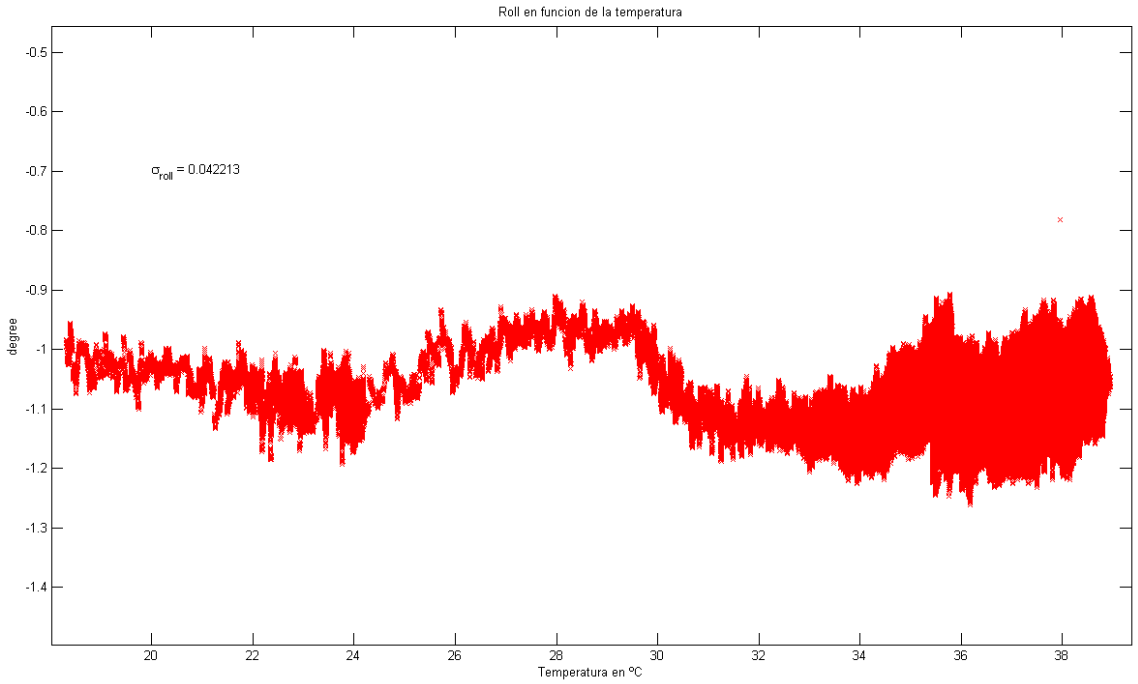

Figura 5 – Muestras capturadas del Roll en función de la temperatura.

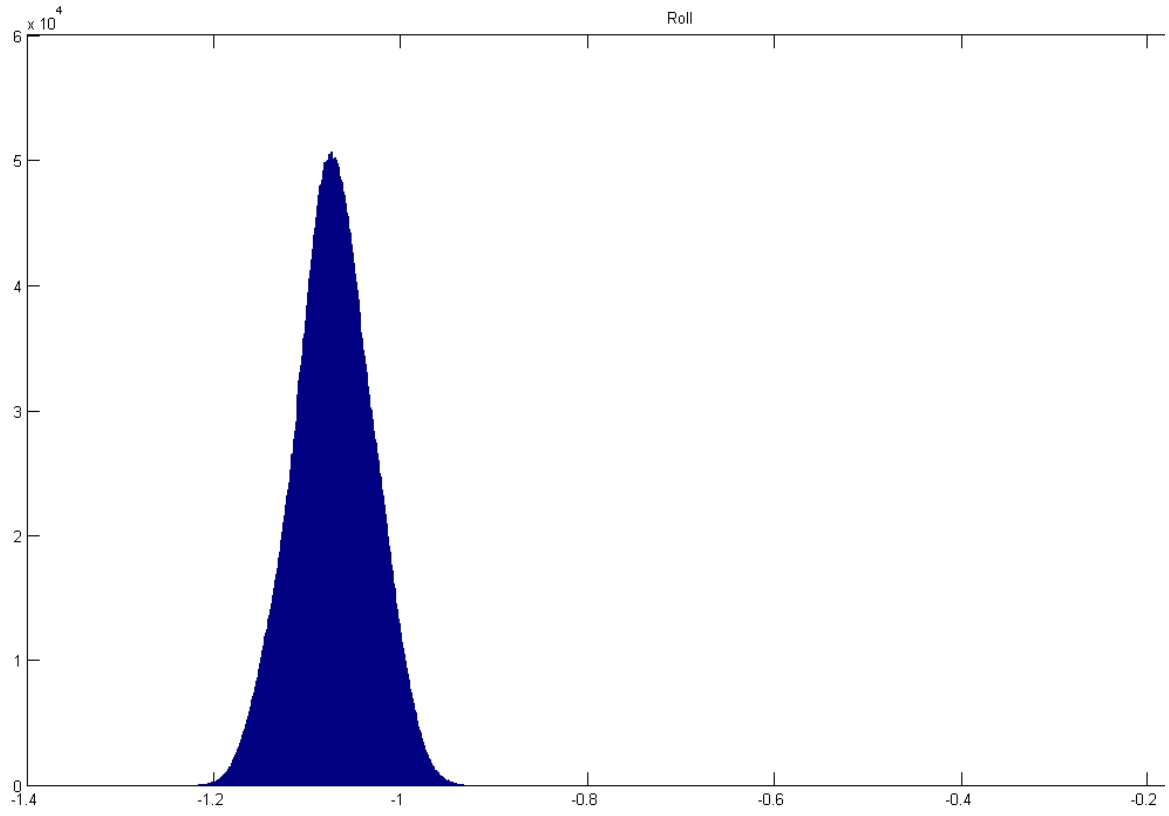

Figura 6 – Distribución de las muestras capturadas del Roll.

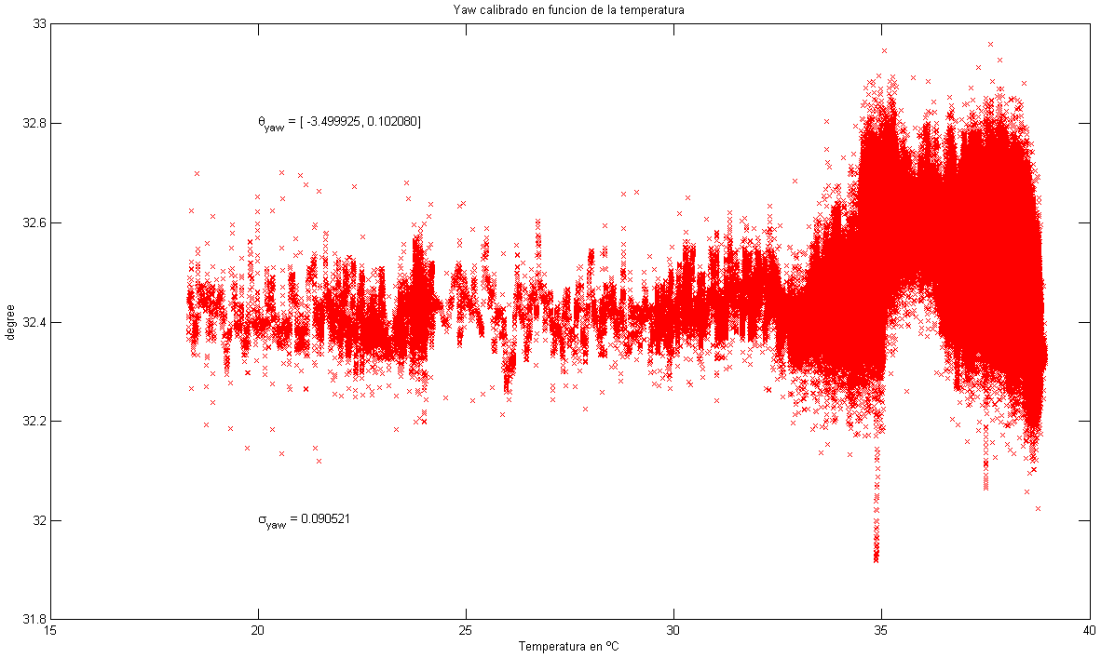

Figura 7 – Muestras capturadas del Yaw en función de la temperatura.

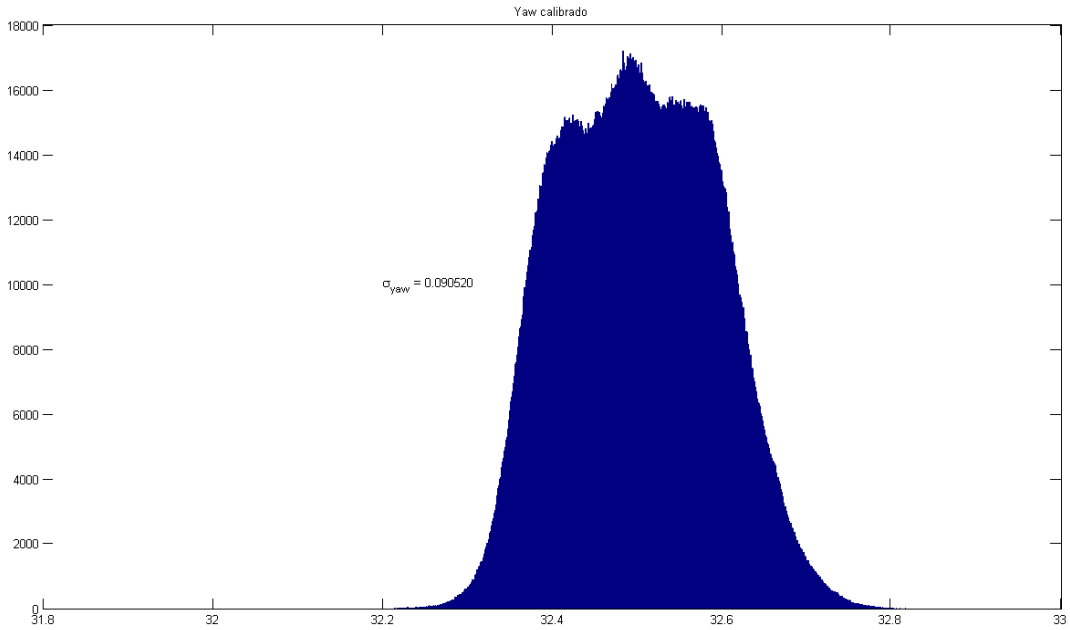

Figura 8 – Distribución de las muestras capturadas del YAW.

Hardware in the Loop – Algoritmos del piloto automático y de navegación

Antes de realizar un vuelo real y con el fin de no estropear la aeronave, es importante comprobar que los algoritmos implementados funcionan correctamente. Para ello se realizan simulaciones con HIL (*hardware in the loop*). Esta simulación consiste en tener un modelo matemático que represente lo más fielmente posible la dinámica de vuelo real. La función de este modelo lo realiza un simulador de vuelo, que permite introducirle estímulos y leer las reacciones. En otras palabras, el simulador permite controlar las superficies de vuelo como se

haría en una aeronave real, y permite obtener la misma información que se leería con los sensores que van colocados en la aeronave.

Cuando se modifica una superficie de control e vuelo en el simulador, la aeronave reacciona según sus modelos matemáticos y esto hace que los sensores internos muestren información de este cambio. Cuanto más parecidos a la realidad sean estos modelos matemáticos, más precisa será la información reflejada en los sensores del simulador y por tanto, más válido serán los resultados obtenidos en la simulación HIL.

En la Figura 9 se muestra de forma simbólica el esquema del HIL. Para testear los algoritmos del piloto automático y navegación, se desconecta la información proveniente de los sensores reales, y se les pasa la información de los sensores que provienen del simulador. Así mismo, las señales de control que se envían a los servos para controlar las superficies de control de vuelo, ahora se envían a las superficies de control de vuelo del simulador y de ésta forma se cierra el bucle. El hardware del piloto automático no sabe que está volando en un simulador, el realiza los mismos cálculos y algoritmos que utiliza normalmente. De esta forma, se puede validar que el algoritmo implementado realiza su función de forma correcta sin el riesgo de perder la aeronave real.

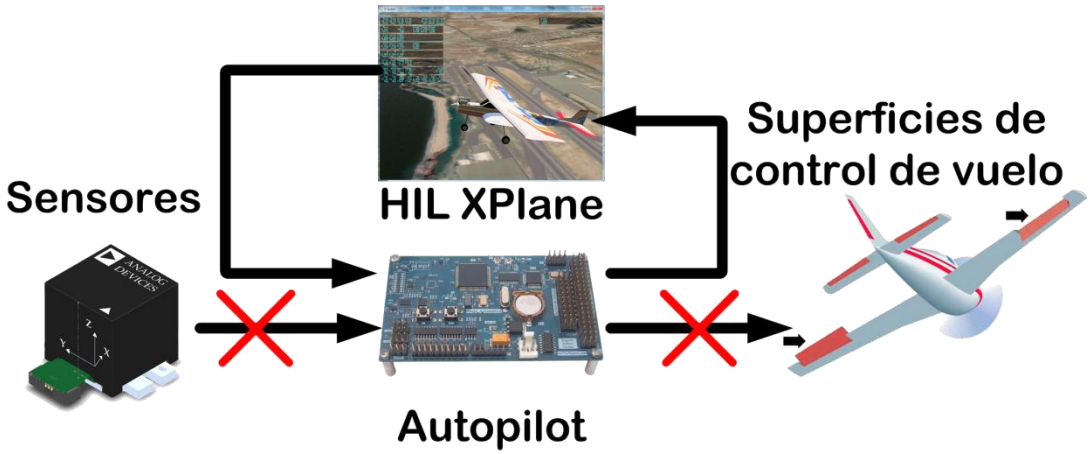

Figura 9 – Esquema simbólico de la simulación HIL.

Para los algoritmos de navegación se realizaron dos test en simulación. El primero es un recorrido sobre el área del aeropuerto de Gando de unos 5 minutos de duración. Y el segundo es un recorrido de unos 50 minutos de duración donde se hace una ruta por la isla.

Ruta Gando

En la Tabla 3 se muestra el plan de vuelo utilizado para realizar el test. El plan de vuelo está formado por las coordenadas del *waypoint*, la altura y velocidad a la que hay que alcanzar el *waypoint* y por último de la distancia a partir de la cual el sistema salta al siguiente *waypoint*.

Tabla 3 – Plan de vuelo de la ruta Gando.

Waypoint	Longitud	Latitud	Altura (ft)	Velocidad (knots)	Threshold (m)
0	-15.390685	27.922577	100.00	40.00	70.00
1	-15.394481	27.924416	300.00	40.00	90.00
2	-15.386260	27.941339	1000.00	60.00	150.00
3	-15.365483	27.933229	2500.00	50.00	200.00
4	-15.387690	27.917963	200.00	50.00	100.00

En la Figura 10 se muestra en rojo el plan de vuelo fijado y en amarillo la trayectoria recorrida por la aeronave.

Figura 10 – Recorrido capturado para la ruta de Gando. En rojo se marca el plan de vuelo y en amarillo se marca el trayecto recorrido.

Ruta Isla

Para la ruta de 50 minutos, el plan de vuelo se presenta en la Tabla 4 y en la Figura 11 y Figura 12 se muestran el recorrido realizado por la aeronave. Como se puede ver, el piloto automático alcanza todos los *waypoints*. El sistema va continuamente estimando el rumbo óptimo para alcanzar el *waypoint* como se explicó en el Capítulo 5. De esta forma, el sistema siempre busca el camino más corto y no hay oscilaciones en el recorrido.

Tabla 4 – Plan de vuelo para la ruta sobre la isla.

Waypoint	Longitud	Latitud	Altura (ft)	Velocidad (knots)	Threshold (m)
0	-15.395923	27.928810	500.00	60.00	200.00
1	-15.612427	27.970514	7000.00	60.00	200.00
2	-15.455198	28.032061	6000.00	60.00	200.00
3	-15.453924	28.071165	1300.00	60.00	200.00
4	-15.419434	28.174870	1000.00	60.00	200.00
5	-15.384541	27.937145	1000.00	60.00	200.00

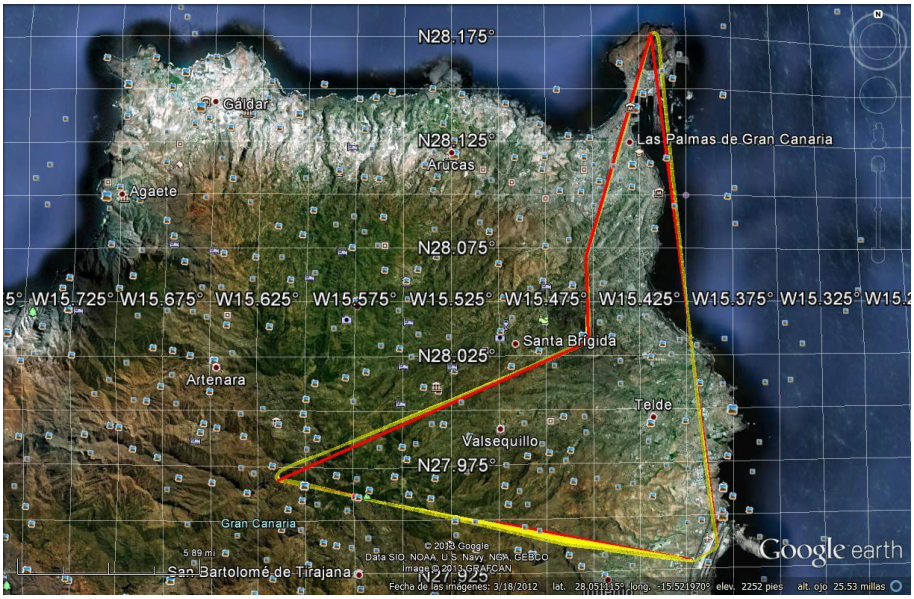

Figura 11 – Ruta isla.

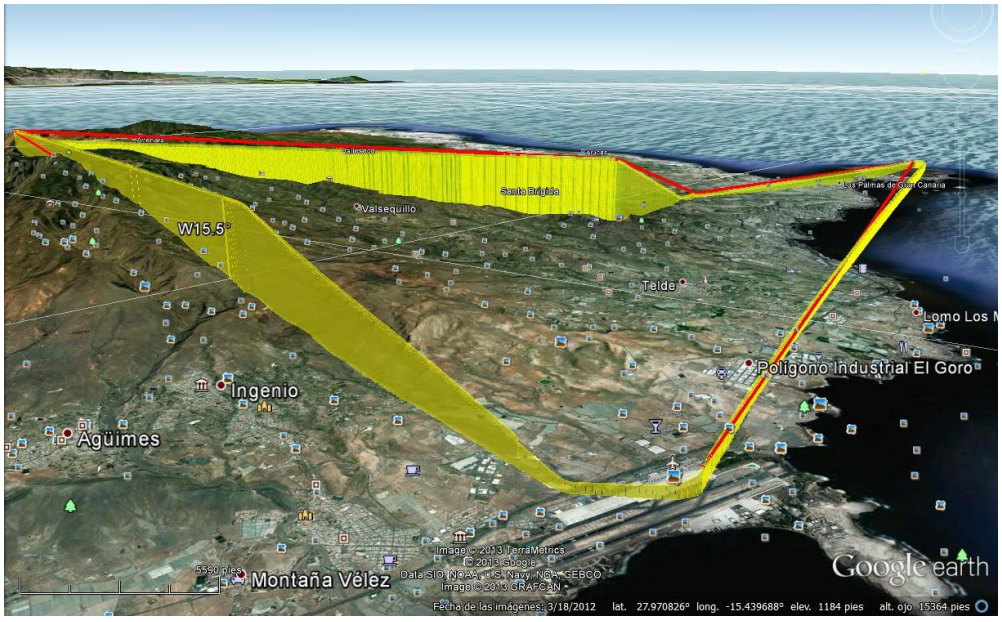

Figura 12 – Ruta Isla. Vista en perspectiva.

Uso de la CPU de la arquitectura software implementada

En la Tabla 5 se resume el uso de la CPU con los distintos módulos de la arquitectura software implementada en este proyecto y el número de líneas de código involucradas sin contar el kernel del RTOS. Si se tiene en cuenta los dos micros, la suma total de las líneas de código escrita para la placa de abordo es de aproximadamente 36.300 líneas de código.

Tabla 5 – Uso de la CPU.

Test	Uso de CPU	Condiciones
SensorDSC		
SupersonicOS ≈18600 líneas de código.	1%	Kernel+CSP+BSP+Test. Todas las tareas del BSP ejecutándose. Leyendo del GPS y extrayendo información. Leyendo de todos los sensores inerciales por I2C y aplicando calibración. Leyendo de los sensores analógicos, aplicando filtrado digital y calibración. Comunicación con el ControlDSC cada 10ms. Comunicación continua por los 3xUART al mismo tiempo a 115200bps. Lectura del 1xbotón. Parpadeo de 1xLED.
SupersonicOS con AHRS y algoritmos del piloto automático. >28500 líneas de código.	9%	Kernel+CSP+BSP+AHRS+AP. Las mismas condiciones que el apartado superior pero realizando los algoritmos del AHRS, del *fly-by-wire*, del piloto automático y de navegación.
SupersonicOS+AHRS+SDCARD >30400 líneas de código	40%	Kernel+CSP+BSP+AHRS+AP+FlightRecorder. Las mismas condiciones que el apartado superior pero realizando operaciones de ficheros sobre la tarjeta uSDCard por SPI.
ControlDSC ≈5900 líneas de código.	2%	Kernel+CSP+BSP. Comunicación con el SensorDSC por SPI cada 10ms. Lectura de 12 señales PWM. Generación de 20 señales PWM para el control de servos. Control del OSD. Comunicación por 1 UART a 115200bps. Control de 6xGPIO.

Bibliografía

[1] Sebastian O.H. Madgwick. “An efficient orientation Filter for inertial and inertial/magnetic sensor arrays”. http://www.x-io.co.uk/res/doc/madgwick_internal_report.pdf

[2] UAV Navigation – VECTOR: High Performance autonomous UAV Flight Control unit. http://www.uavnavigation.com/avionics/vector

[3] XSense – Mti-G-700 GPS/INS. http://www.xsens.com/en/general/mti-g-100-series