ÍNDICE:

ÍNDICE DE ILUSTRACIONES:

1 Introducción. Control de helicópteros

1.1 Objetivos

El objetivo de este proyecto es el vuelo autónomo de un helicóptero de radiocontrol.

Para ello se adaptara un helicóptero comercial de radiocontrol, al cual se le acoplará un pequeño ordenador embebido (placa Hercules EBX), y los sensores necesarios.

La placa Hercules tomara los datos de los distintos sensores, procesará la información y le mandara las señales de control a los actuadores, buscando la estabilización (control) del helicóptero.

En esta fase del proyecto el trabajo consiste en crear el software y la puesta a punto de la placa Hercules para la adquisición de datos arrojados por los sensores del helicóptero en vuelo.

Tras la adquisición de datos estos serán tratados, grabados y enviados en vuelo al control de tierra.

Debido a lo complejo de controlar un helicóptero en vuelo (6 grados de libertad de su posición en el espacio mas 6: sus derivadas) nos enfrentamos a un problema de control que requiere el estudio a través de distintas técnicas de control, como veremos a continuación.

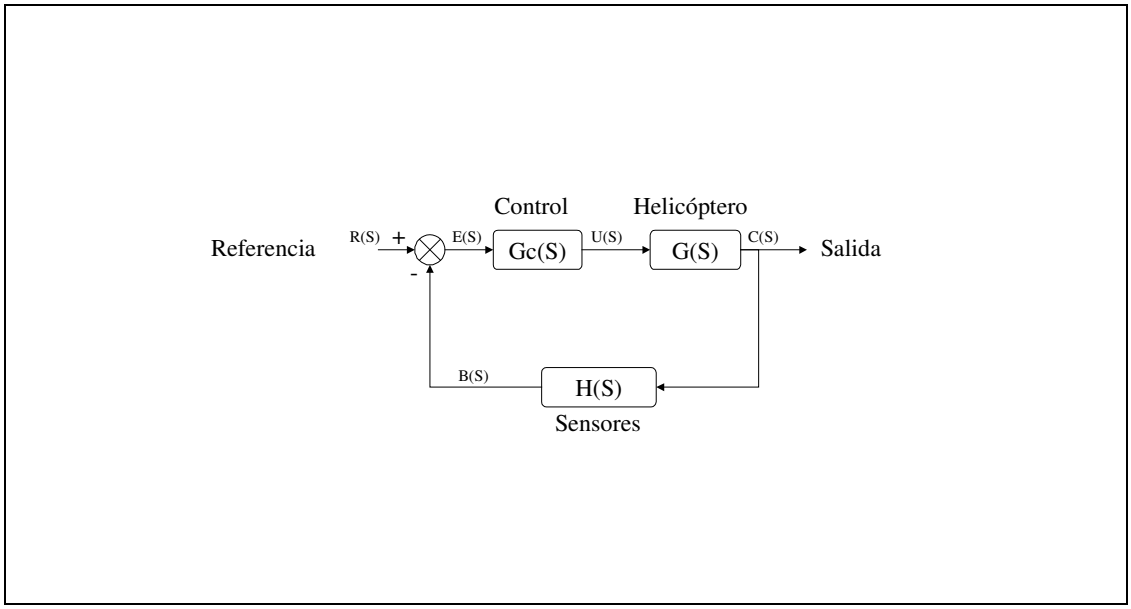

Ilustración 1: Sistema de control.

1.2 Técnicas de control

En un sistema de control realimentado nos encontramos los siguientes elementos;
La referencia a seguir por las variables a controlar (el objetivo): R(s)
Los sensores, que nos dan la medida de las variables a controlar: H(s)
El error entre la referencia y la medida tomada por los sensores: E(s)
El controlador: Gc(s)
La consigna dada a los actuadores: U(s)
La planta, en este caso el Helicóptero: G(s)
Para el estudio del sistema de control tendremos que realizar las siguientes cuestiones;
Modelado físico del sistema. G(s)
Identificación de los parámetros de las ecuaciones del modelo físico.
Diseño de Gc(s): elección del tipo de control a utilizar, aquí trataremos el control en el espacio de estados y el control pid.
Implementación del controlador en el ordenador.

1.2.1 Modelado del sistema

El modelo dinámico teórico de un helicóptero en vuelo vertical proviene de la combinación de las teorías del momento y de la teoría de las palas.
Tanto el estudio de estas teorías como las ecuaciones resultantes se aleja del propósito de este proyecto por lo que se obviarán y trabajara sin ellas.
Las ecuaciones del modelo dinámico teórico contienen una serie de parámetros constantes no definidos que tendremos que identificar basándonos en los resultados experimentales que se realizaran con el helicóptero montado en la plataforma de pruebas.

1.2.2 Identificación

Para la identificación de los parámetros constantes y desconocidos que tenemos en las ecuaciones del modelo dinámico haremos pruebas con el helicóptero montado en su plataforma de pruebas. Así iremos aislando los parámetros e identificándolos.

1.2.3 Implementación del controlador

Veremos dos técnicas de control diferentes; el controlador PID y el control en el espacio de estados

1.2.3.1 Control PID

Pid es un control proporcional, integral y derivativo con respecto al error.
En el dominio temporal continuo tiene la forma;

$$u(t) = K \times e(t) + K \times \int e(t) \times \delta t + K \times \frac{\delta e}{\delta t}$$

Escrito en el dominio de Laplace la función de transferencia continua de un pid nos queda;

$$Gc(s) = \frac{U(s)}{E(s)} = k_c\left(1 + \frac{1}{T_i s} + T_d s\right)$$

Para pasar a discreto utilizaremos la aproximación por operador derivada, en la que

Integral = sumatorio

Derivada = diferencia

$$u(t) = K \times e(t) + K \times \int e(t) \times \delta t + K \times \frac{\delta e}{\delta t}$$

$$\Downarrow$$

$$u_k = K\left(e_k + \frac{T}{T_i}\sum e_k + \frac{T_d}{T} \times (e_k - e_{k-1})\right)$$

con lo que la actuación queda:

$$u_k = u_{k-1} + (q_0 \times e_k + q_1 \times e_{k-1} + q_2 \times e_{k-2})$$

Por lo tanto para conocer la actuación en un instante tenemos que conocer la actuación anterior, el error (referencia – posición medida) en el instante actual y los dos instantes anteriores, además de los parámetros del controlador PID: q0, q1 y q2. Siendo estos:

$$q_0 = K\left(1 + \frac{T}{T_i} + \frac{T_d}{T}\right) \qquad q_1 = K\left(-1 - 2\frac{T_d}{T}\right) \qquad q_2 = K\left(\frac{T_d}{T}\right)$$

y cambiando al dominio en frecuencia la función de transferencia en discreto nos queda:

$$Gc(z) = \frac{U(z)}{E(z)} = \left(\frac{q_0 + z^{-1} q_1 + z^{-2} q_2}{1 - z^{-1}} \right)$$

1.2.3.1.1 Implementación del PID

Para implementar un controlador PID en nuestro helicóptero escribiremos el siguiente código en el programa control.c

```
void control (void)
{
        for(i=0;i<4;i++) //bucle para los 4 actuadores
        {
//las medidas de referencias han sido leidas por la imu
//calculo el error actual
ek[i]=ref[i]-med[i];
//calculo la actuación que será utilizada por la función pwm que saca la actuación (entre 0 y 100) para los servomecanismos
uk[i]=uk1[i]+q0*ek[i]+q1*ek1[i]+q2*ek2[i];
//actualizo las variables
uk1[i]=uk[i];
ek1[i]=ek[i];
ek2[i]=ek1[i];
}
}
```

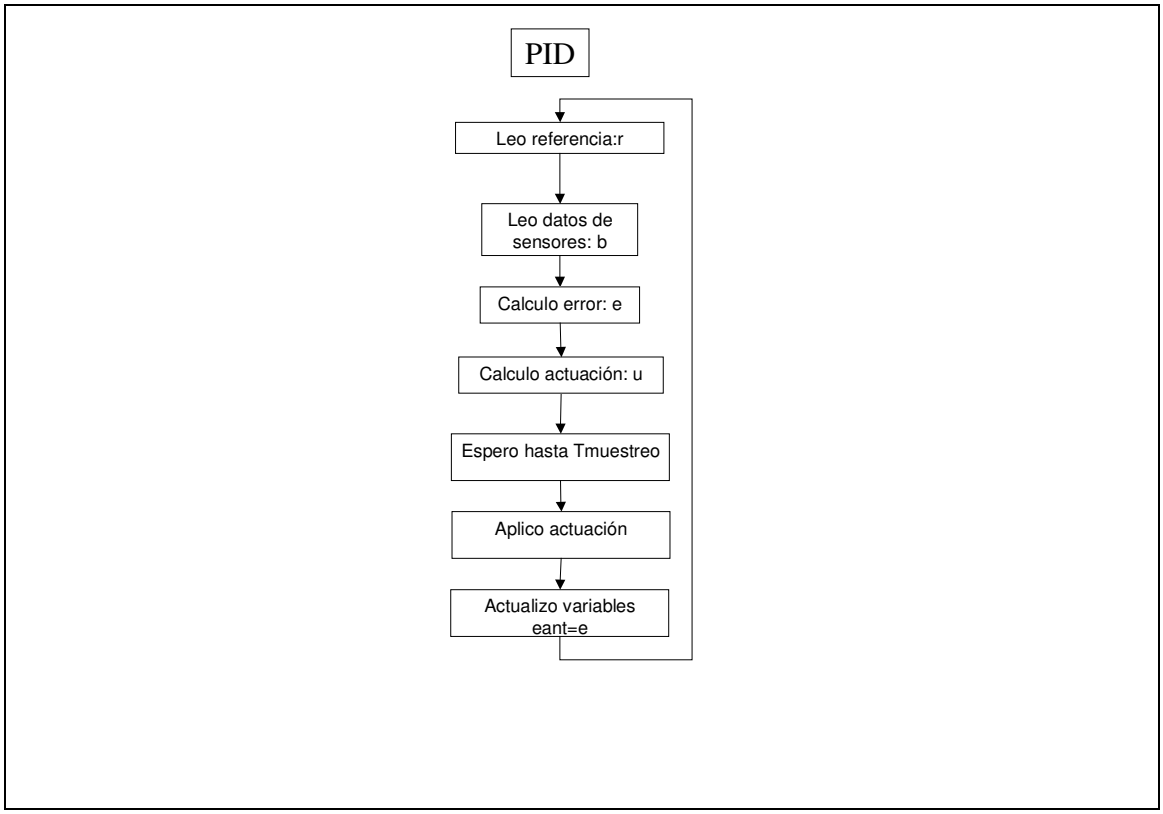

Ilustración 2: Control PID.

1.2.3.2 Control LQR

La técnica de control lineal multivariable LQR (regulador lineal cuadrático) ofrece una técnica de control lineal simple (u=-kx) y robusta.

Tenemos un sistema multivariable (MIMO: múltiples entradas y salidas) y no lineal, por lo que tenemos que linealizar el sistema previamente.

La técnica LQR calcula la matriz [K(t)] tal que realizando un control con realimentación de estados expresada mediante la ecuación

[u]=-[K(t)]·[x]

se minimiza la función de costes J

$$\frac{\delta}{\delta T}[X]=[A][X]+[B][U]$$

$$[Y]=[C][X]+[D][U]$$

$$J=\int_0^T ([X]^T\cdot[Q][X]+[U]^T\cdot[R][U])\delta t+[X(T)]^T\cdot[M][X(T)]$$

el objetivo del control LQR consiste en llevar los estados [x] los mas cerca de sus estados de referencia.

El diseñador del control LQR diseña las matrices [Q] [R] y [M] de forma que sus elementos deben ser positivos o cero.

En un helicóptero podemos definir el vector del espacio de estados de la siguiente manera;

$$X=\begin{pmatrix} u \\ w \\ q \\ \theta \\ v \\ r \\ p \\ \phi \end{pmatrix}=\begin{pmatrix} \chi_1 \\ \chi_2 \\ \chi_3 \\ \chi_4 \\ \chi_5 \\ \chi_6 \\ \chi_7 \\ \chi_8 \end{pmatrix}$$

y el vector del espacio de control;

$$U=\begin{pmatrix} \delta_e \\ \delta_c \\ \delta_a \\ \delta_r \end{pmatrix}=\begin{pmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{pmatrix}$$

La matriz A tiene la forma;

$$A = \begin{matrix} X_u & X_w & X_q & -g & X_v & X_r & X_p & 0 \\ Z_u & Z_w & Z_q & -g\theta_0 & Z_v & Z_r & Z_p & -g\Phi_0 \\ M_u & M_w & M_q & 0 & M_v & M_r & M_p & 0 \\ 0 & 0 & 1 & 0 & 0 & -\Phi_0 & 0 & 0 \\ Y_u & Y_w & Y_q & 0 & Y_v & Y_r & Y_p & g \\ N_u & N_w & N_q & 0 & N_v & N_r & N_p & 0 \\ L_u & L_w & L_q & 0 & L_v & L_r & L_p & 0 \\ 0 & 0 & 0 & 0 & 0 & \theta_0 & 1 & 0 \end{matrix}$$

G es una matriz de 8x4 con la cuarta y octava filas iguales a cero

Dando valores de posición y ángulos (alrededor del punto de equilibrio) nos queda;

$$\dot{x} = u$$

$$\dot{h} = -w$$

$$\dot{\psi} = r + \phi_0 q$$

$$\dot{y} = v$$

Para el control del vector de velocidad en los ejes y de velocidad en eje z (alrededor del punto de equilibrio) utilizamos la función de costes J:

$$J = \int_0^\infty (u^2 + w^2 + v^2 + r^2 + \delta_e{}^2 + \delta_c{}^2 + \delta_a{}^2 + \delta_r{}^2)dt$$

Nomenclatura	
U	Velocidad en el eje x
W	Velocidad en el eje z
Q	Velocidad angular alrededor del eje y
P	Velocidad angular alrededor del eje x
V	Velocidad en el eje y
R	Velocidad angular alrededor del eje z
θ	Pitch
ϕ	Roll
ψ	Yaw
δ_e	Ángulo longitudinal del cíclico
δ_c	Ángulo de pala de rotor principal
δ_a	Ángulo transversal del cíclico
δ_r	Ángulo de palas de cola

1.2.3.2.1 Implementación del control LQR

Para implementar un controlador PID en nuestro helicóptero escribiremos el siguiente codigo en el programa control.c

```
void control (void)
{
	//las medidas de referencias han sido leidas por los sensores
	//calculo el vector u=-Kx:multiplico la matriz [K]·[x]
	//calculo la matriz K previamente
	for(j=0;j<4;j++)//cuatro filas de la matriz K
	{
		for(i=0;i<8;i++)//ocho columnas de la matriz K
		{
			u[j]=-K[i][j]*x[j];
		}
	}
}
```

1.3 Tecnologías

Haremos una revisión de las tecnologías involucradas a partir del siguiente grafico.

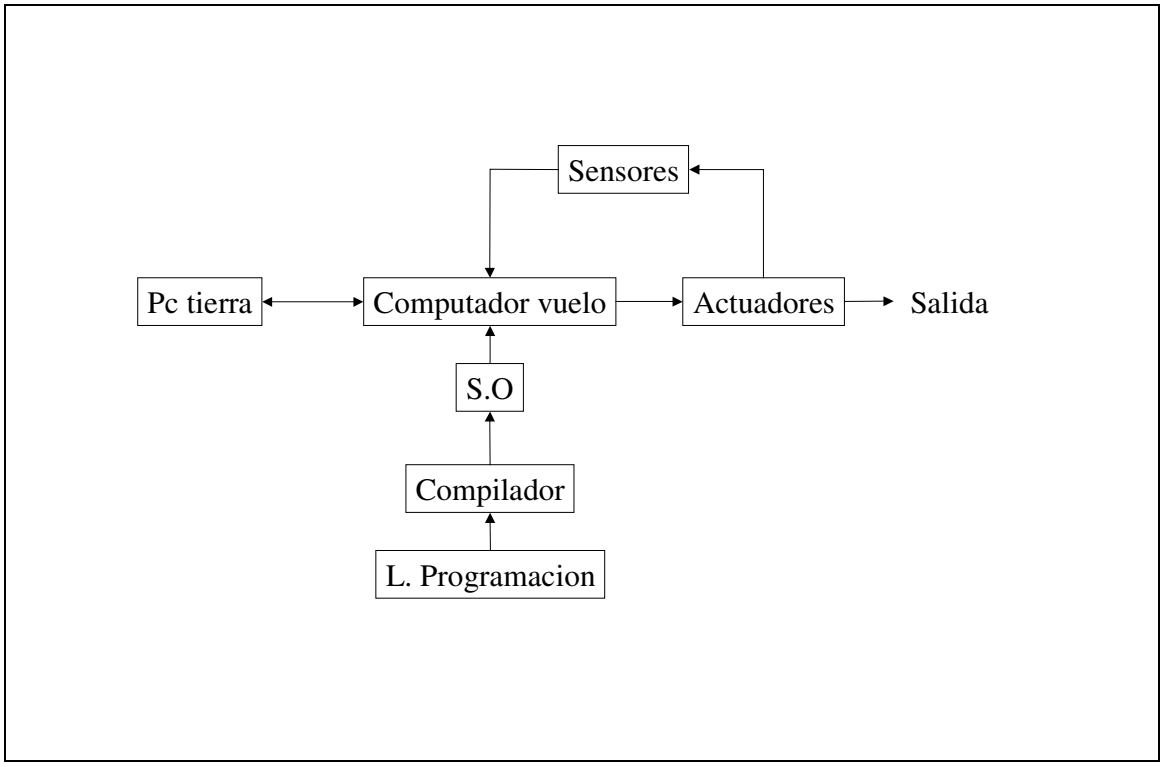

Ilustración 3: Tecnologías involucradas.

1.3.1 Computador de vuelo

El computador de vuelo es la placa Hercules EBX. Consiste en una placa con un microprocesador funcionando a 400MHz, una flash disk de 128Mb donde ira alojado el Sistema Operativo y una memoria RAM de 128Mb. La placa cuenta con entradas y salidas A y D, 4 puertos RS-232, usb y ethernet, pudiéndose conectar los periféricos necesarios, como discos duros o lector de CD que en vuelo no llevará conectados debido al problema de vibraciones.

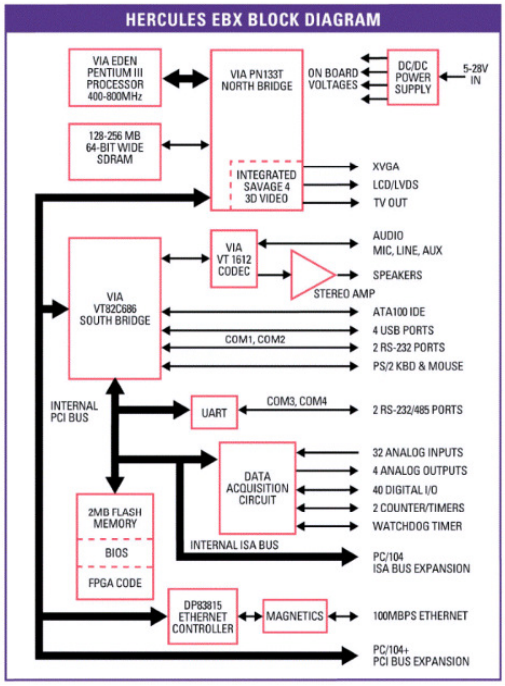

Ilustración 4: Diagrama de bloques de la placa Hercules EBX.

1.3.2 Sistema Operativo

Un sistema operativo (SO) es un conjunto de programas o software destinado a permitir la comunicación del usuario con un ordenador y gestionar sus recursos de manera eficiente. Comienza a trabajar cuando se enciende el ordenador, y gestiona el hardware de la máquina desde los niveles más básicos, abstrayéndolo.
El SO utilizado es Linux en su versión debian Woody (r3.0)
El Kernel de Linux es uno de los más robustos y versátiles de todos los sistemas operativos conocidos e incluso ha sido mejorado en posteriores versiones, toda vez que el sistema original recibe constantemente importantes aportes de miembros de las comunidades de Linux.
Las características de los sistemas operativos de Linux son; distribución gratuita, alta configurabilidad, gran numero de usuarios lo que genera una gran información sobre su uso.

La principal característica de debian (la versión que usamos) es que ocupa poco espacio en disco, lo que será de gran ayuda para poder instalarlo en la flash disk (128Mb).

1.3.3 Lenguaje de programación

Un lenguaje de programación es una técnica estándar de comunicación que permite expresar las instrucciones que han de ser ejecutadas en una computadora. Consiste en un conjunto de reglas sintácticas y semánticas que definen un programa informático.

El lenguaje de programación utilizado es C. El lenguaje C es un lenguaje de nivel medio desarrollado a principios de los años 70 en Bell laboratiories por Brian Kernighan y Dennis Ritchie.

Esta caracterizado por ser un lenguaje de uso general con una síntesis sumamente compacta y de alta portabilidad.

Una de sus principales características es el uso de librerías externas que contiene la definición de funciones.

Ademas de las funciones de C y de diversas librerias utilizamos las del driver de la placa: DSCud V5.3.

1.3.4 Compilador

Un compilador acepta programas escritos en un lenguaje de alto nivel y los traduce a otro lenguaje, generando un programa equivalente independiente, que puede ejecutarse tantas veces como se quiera. Este proceso de traducción se conoce como compilación.

Se ha utilizado el compilador GCC de linux, GCC es un compilador rápido, muy flexible, y riguroso con el estándar de C ANSI.

GCC es un compilador integrado del proyecto GNU para C, C++, Objective C y Fortran; es capaz de recibir un programa fuente en cualquiera de estos lenguajes y generar un programa ejecutable binario en el lenguaje de la máquina donde ha de correr.

La sigla GCC significa "GNU Compiler Collection". Originalmente significaba "GNU C Compiler"; todavía se usa GCC para designar una compilación en C.

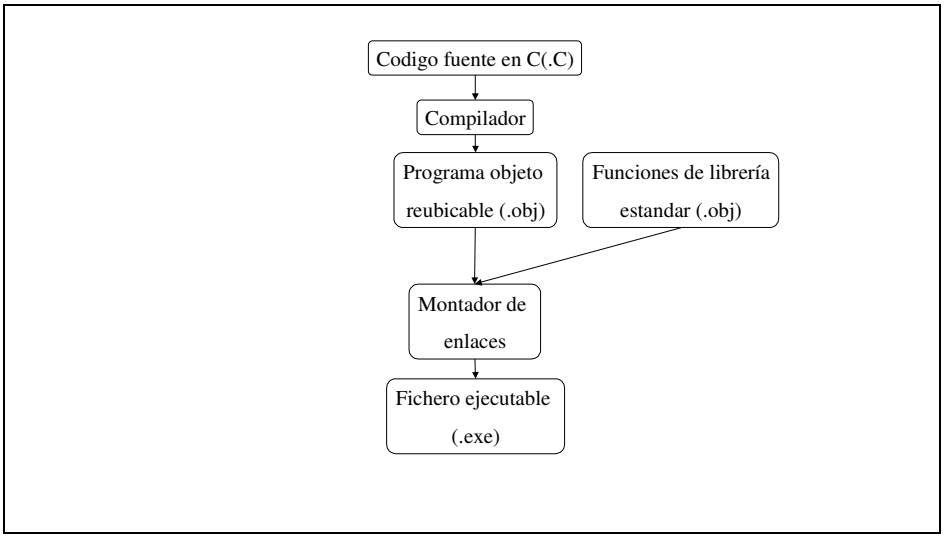

Ilustración 5: Esquema de compilacíon.

1.3.5 Sensores

1.3.5.1 Umi;

Unidad de medida inercial. Pequeño sensor (75 gramos) de la casa Microstrain,Inc. Conectada a través del puerto RS-232 con la placa Hercules le proporciona los siguientes datos;

Timer ticks (tiempo del muestreo)

Temperatura (en el interior del sensor)

Ángulos de Euler; pitch, roll y yaw.

Velocidades angulares.

Matriz de quartenions y matriz de orientación.

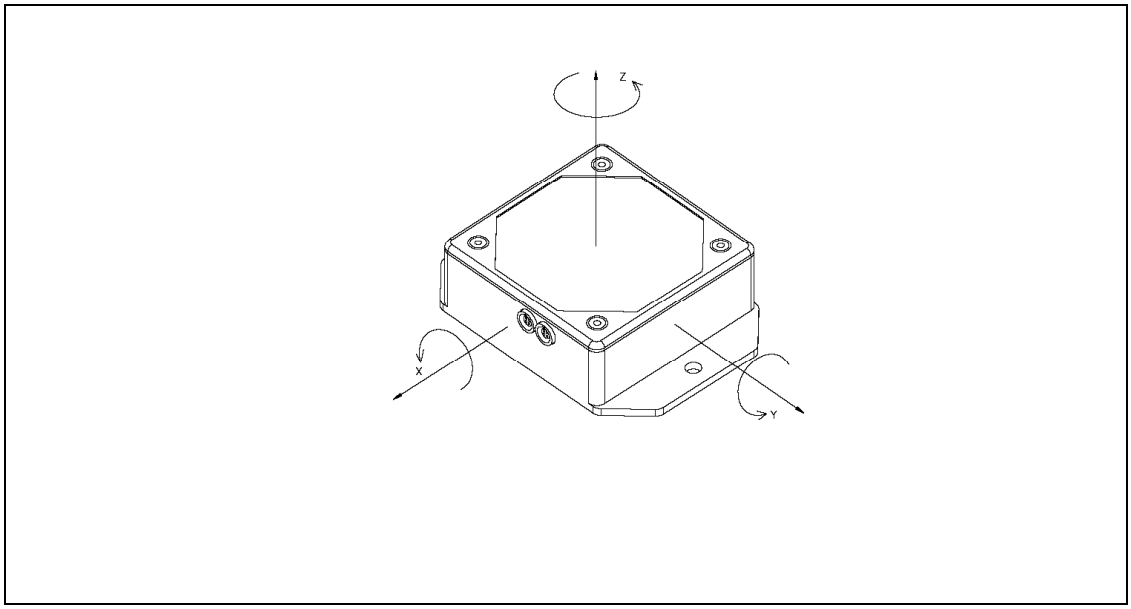

Ilustración 6: Sensor IMU.

1.3.5.2 Altímetro;

Durante las pruebas en la plataforma utilizaremos un altímetro basado en un potenciómetro que me dará una lectura analógica en función de la altura del helicóptero.

1.3.6 Actuadores

Son los encargados de mover los controles del helicóptero, compuestos por 5 servomotores alimentados a 5v.

Un servomotor es un motor que funciona posicionándose de manera precisa en un rango angular.

En nuestro helicóptero trabajan a 1000 hz. y el valor que se le pasa es el ancho de pulso (PWM) variando este desde 0 a 100, con lo que variamos su posición.

1.3.7 Comunicación del helicóptero con el ordenador de tierra

Para comunicar el Helicóptero con tierra utilizamos el conector de red de la placa Hercules al que conectamos un puente inalámbrico Wi-Fi.

Wi-Fi, acrónimo de *Wireless Fidelity*, es un conjunto de estándares para redes inalámbricas basado en las especificaciones IEEE 802.11.

Debido a que el helicóptero no tendrá instalado ningún elemento hardware de control (ratón, teclado o pantalla) realizaremos la teleoperación desde el ordenador de tierra, por lo tanto habrá una comunicación bidireccional en la que desde tierra se le irán mandando ordenes al helicóptero y este ira mandando datos de los sensores en tiempo real.