

Control de actitud (ADCS)

Gastón Valdez

Carrera de Especialización en Sistemas Embebidos

Director: Amilcar Rincon Charris (UPM)

Jurados:

Jurado 1 (pertenencia)

Jurado 2 (pertenencia)

Jurado 3 (pertenencia)

Ciudad de La plata, Octubre de 2025

Resumen

El resumen debe escribirse en uno o dos párrafos. Debe ser breve y conciso, sin ningún elemento de formato en el texto como itálicas o negritas. Tampoco se deben usar siglas ni acrónimos que no resulten obvios para un lector promedio de la memoria, ni referencias bibliográficas o notas al pie de página. No debe faltar qué es lo que se hizo/logró, qué importancia/valor tiene el proyecto/resultado, qué va a encontrar el lector en la memoria y qué contenidos de la especialización/maestría se aplicaron en el proyecto.

Agradecimientos

Esta sección es para agradecimientos personales y es totalmente **OPCIONAL**.

Índice general

Resumen	I
1. Introducción general	1
1.1. Control de actitud	1
1.2. Estado del Arte	2
1.3. Desarrollo del trabajo	2
1.4. Objetivos	3
2. Introducción específica	5
2.1. Estilo y convenciones	5
2.1.1. Rotaciones activas vs Pasivas	5
3. Diseño e implementación	7
3.1. Análisis del software	7
4. Ensayos y resultados	9
4.1. Pruebas funcionales del hardware	9
5. Conclusiones	11
5.1. Conclusiones generales	11
5.2. Próximos pasos	11

Índice de figuras

1.1. ADCS comercial - modelo ADCS-MTQ.	1
1.2. Diagrama en bloques del trabajo desarrollado.	3

Índice de tablas

Dedicado a... [OPCIONAL]

Capítulo 1

Introducción general

Un sistema de control de actitud¹ es un conjunto de componentes de hardware y software que permite a un satélite o nave espacial mantener o modificar su orientación en el espacio. Para realizar estos ajustes, el sistema utiliza **actuadores**². Su funcionamiento integra tres funciones clave:

1. **Adquisición de datos:** Lectura de sensores (magnéticos, inerciales, etc.).
2. **Procesamiento:** Estimación de la orientación mediante algoritmos.
3. **Corrección:** Aplicación de fuerzas/torques mediante actuadores.

Estas correcciones minimizan la diferencia entre la orientación actual y la deseada, garantizando precisión en aplicaciones como vehículos espaciales, drones o sistemas de estabilización.

1.1. Control de actitud

En la actualidad existen pequeños satélites que orbitan alrededor de la tierra denominados CubeSats. Estos usan componentes electrónicos modulares y uno de ellos se llama ADCS. En la siguiente imagen puede verse un módulo comercial

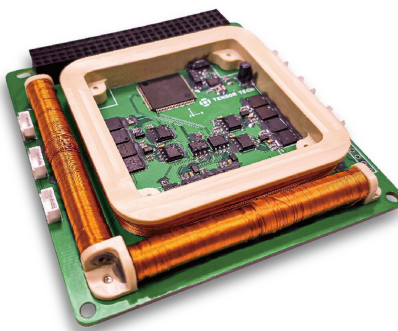


FIGURA 1.1. ADCS comercial - modelo ADCS-MTQ.

¹ ADCS, por sus siglas en inglés *Attitude Determination and Control System*

² Dispositivo físico que genera movimiento o aplica fuerza para modificar la posición/orientación de un objeto.

El sistema ADCS se encarga de mantener y corregir la orientación del satélite a una orientación mediante un algoritmo de control.

Un sistema ADCS consta de una unidad inercial de medición (IMU por sus siglas en inglés). Este dispositivo mide campo magnético, aceleración y velocidad angular en tres ejes. Con estos datos determina la orientación dentro del espacio usando algoritmos como TRIAD, Davenport's, entre otros. Una vez calculada la orientación, el sistema genera una actuación (por ejemplo, mediante magnetorqueres o ruedas de reacción) para aplicar un torque (o par) en alguno de los ejes principales. Esto permite mantener la orientación deseada y compensar perturbaciones externas, como la resistencia atmosférica o la influencia gravitatoria.

1.2. Estado del Arte

El control de actitud en vehículos espaciales es un área ampliamente estudiada, con múltiples enfoques que combinan sensores, algoritmos de estimación y actuadores para mantener o modificar la orientación del vehículo en el espacio. Existen diversas plataformas de simulación y validación de sistemas ADCS, muchas de ellas desarrolladas en entornos como MATLAB/Simulink, FreeFlyer, o STK. Sin embargo, estas soluciones presentan ciertas limitaciones, especialmente en contextos educativos o de bajo presupuesto.

Durante la etapa inicial de esta tesis se realizó una revisión exploratoria de herramientas disponibles, lo cual permitió identificar algunas dificultades frecuentes. Por un lado, muchas plataformas requieren licencias costosas, lo que puede restringir su uso en entornos académicos. Por otro, su complejidad técnica y curva de aprendizaje representan una barrera para estudiantes o investigadores que se inician en esta área.

Además, se observó que pocas de estas herramientas permiten integración directa con microcontroladores u otros sistemas embebidos, lo cual complica la validación práctica de los algoritmos desarrollados. Del mismo modo, en varios casos la simulación de sensores se realiza bajo condiciones ideales, sin considerar la presencia de ruido, que limita la validez del modelo en entornos reales.

Frente a estas limitaciones, el desarrollo propuesto en esta tesis busca ofrecer una alternativa accesible, de bajo costo y fácil de utilizar, que permita tanto la simulación como la conexión directa con hardware físico. Entre sus características, se destaca la capacidad de simular sensores con ruido, modificar algoritmos y configuraciones de sensores de manera modular, y validar los resultados directamente sobre un microcontrolador. Esto proporciona un entorno de desarrollo más realista y adaptable, especialmente útil en entornos educativos o de investigación aplicada.

1.3. Desarrollo del trabajo

El trabajo desarrolla la simulación de un control de actitud utilizando el algoritmo TRIAD para estimar la orientación del satélite. Este método compara dos referencias conocidas (por ejemplo campo magnético, aceleración de la gravedad, dirección del sol, etc) con las mediciones de los sensores a bordo. A partir de esta comparación, el TRIAD determina la orientación relativa del Cubesat respecto

a un sistema de coordenadas de referencia. El diagrama en bloques del trabajo desarrollado se muestra a continuación:

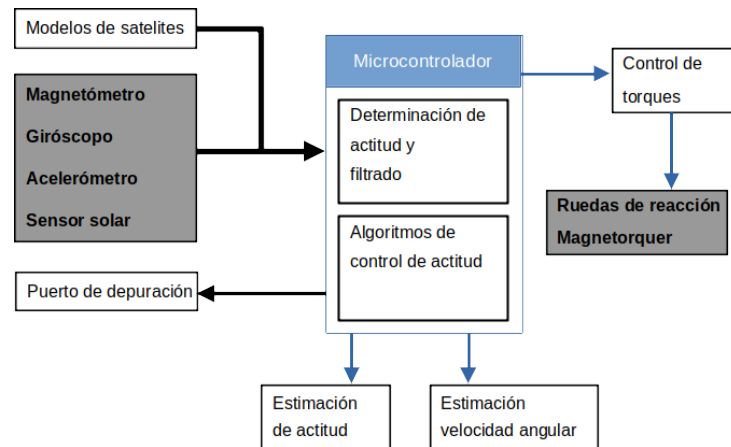


FIGURA 1.2. Diagrama en bloques del trabajo desarrollado.

El desarrollo de los algoritmos se realizó en dos lenguajes de programación: C para la implementación embebida y Python para prototipado rápido y validación de modelos. El código en C se integrará en un sistema embebido utilizando la metodología HIL (Hardware-in-the-Loop), que permite validar el comportamiento del sistema de control en un entorno simulado antes de su implementación física.

1.4. Objetivos

El objetivo inicial del proyecto consistía en desarrollar una placa de circuito impreso (PCB) con sistemas de actuación para control de actitud, ya sea mediante ruedas de reacción o magnetorquers. Sin embargo, durante la fase inicial de desarrollo se identificaron varias limitaciones significativas:

- **Alto costo de soluciones comerciales:** Los modelos existentes para simulación presentaban costos prohibitivos para instituciones académicas.
- **Limitaciones en software libre:** Las herramientas analizadas (GMAT, Octave, entre otros) carecían de capacidad nativa para integración HIL (Hardware-in-the-Loop) con microcontroladores, siendo Matlab la única alternativa viable pero con sus propias dificultades técnicas.
- **Barreras económicas:** Los costos asociados a las soluciones comerciales resultaban inviables para su implementación en el contexto universitario.

Estos hallazgos motivaron una ampliación del alcance del trabajo, incorporando los siguientes objetivos técnicos:

- Desarrollo de un modelo dinámico de simulación que incluya:
 - Modelado matemático del satélite y su entorno orbital
 - Simulación de sensores (generación de datos sintéticos)
- Implementación de una plataforma HIL que permita:
 - Conexión con hardware real (microcontrolador/procesador)

- Inyección de datos de sensores simulados al sistema embebido
- Validación de los algoritmos de control en condiciones emuladas

Este enfoque permitirá evaluar el rendimiento del sistema de control antes de proceder al diseño final del circuito impreso, reduciendo riesgos técnicos y costos asociados a iteraciones de hardware. Cabe aclarar que la simulación parte del supuesto de que el sistema ya se encuentra en la órbita deseada, sin modelar el proceso de inserción orbital ni el trayecto previo.

Capítulo 2

Introducción específica

Se presentan las bases matemáticas del control de actitud. El control de actitud se basa en seleccionar distintos sistemas de referencia orbitales.

2.1. Estilo y convenciones

2.1.1. Rotaciones activas vs Pasivas

Se recomienda no utilizar una sección de glosario sino colocar la descripción de las abreviaturas como parte del mismo cuerpo del texto. Por ejemplo, RTOS (*Real Time Operating System*, Sistema Operativo de Tiempo Real) o en caso de considerarlo apropiado mediante notas a pie de página.

$$ds^2 = c^2 dt^2 \left(\frac{d\sigma^2}{1 - k\sigma^2} + \sigma^2 [d\theta^2 + \sin^2 \theta d\phi^2] \right) \quad (2.1)$$

Es importante tener presente que si bien las ecuaciones pueden ser referidas por su número, también es correcto utilizar los dos puntos, como por ejemplo “la expresión matemática que describe este comportamiento es la siguiente:”

$$\frac{\hbar^2}{2m} \nabla^2 \Psi + V(\mathbf{r}) \Psi = -i\hbar \frac{\partial \Psi}{\partial t} \quad (2.2)$$

Para generar la ecuación 2.1 se utilizó el siguiente código:

```
\begin{equation}
\label{eq:metric}
ds^2 = c^2 dt^2 \left( \frac{d\sigma^2}{1-k\sigma^2} + \right.
\left. \sigma^2 \left[ d\theta^2 + \sin^2 \theta d\phi^2 \right] \right)
\end{equation}
```

Y para la ecuación 2.2:

```
\begin{equation}
\label{eq:schrodinger}
\frac{\hbar^2}{2m} \nabla^2 \Psi + V(\mathbf{r}) \Psi =
-i\hbar \frac{\partial \Psi}{\partial t}
\end{equation}
```


Capítulo 3

Diseño e implementación

Todos los capítulos deben comenzar con un breve párrafo introductorio que indique cuál es el contenido que se encontrará al leerlo. La redacción sobre el contenido de la memoria debe hacerse en presente y todo lo referido al proyecto en pasado, siempre de modo impersonal.

3.1. Análisis del software

La idea de esta sección es resaltar los problemas encontrados, los criterios utilizados y la justificación de las decisiones que se hayan tomado.

Se puede agregar código o pseudocódigo dentro de un entorno `lstlisting` con el siguiente código:

```
\begin{lstlisting}[caption= "un epígrafe descriptivo"]
las líneas de código irían aquí...
\end{lstlisting}
```

A modo de ejemplo:

```
1 #define MAX_SENSOR_NUMBER 3
2 #define MAX_ALARM_NUMBER 6
3 #define MAX_ACTUATOR_NUMBER 6
4
5 uint32_t sensorValue[MAX_SENSOR_NUMBER];
6 FunctionalState alarmControl[MAX_ALARM_NUMBER]; //ENABLE or DISABLE
7 state_t alarmState[MAX_ALARM_NUMBER]; //ON or OFF
8 state_t actuatorState[MAX_ACTUATOR_NUMBER]; //ON or OFF
9
10 void vControl() {
11
12     initGlobalVariables();
13
14     period = 500 ms;
15
16     while(1) {
17
18         ticks = xTaskGetTickCount();
19
20         updateSensors();
21
22         updateAlarms();
23
24         controlActuators();
25
26         vTaskDelayUntil(&ticks, period);
27     }
```

28 }

CÓDIGO 3.1. Pseudocódigo del lazo principal de control.

Capítulo 4

Ensayos y resultados

Todos los capítulos deben comenzar con un breve párrafo introductorio que indique cuál es el contenido que se encontrará al leerlo. La redacción sobre el contenido de la memoria debe hacerse en presente y todo lo referido al proyecto en pasado, siempre de modo impersonal.

4.1. Pruebas funcionales del hardware

La idea de esta sección es explicar cómo se hicieron los ensayos, qué resultados se obtuvieron y analizarlos.

Capítulo 5

Conclusiones

Todos los capítulos deben comenzar con un breve párrafo introductorio que indique cuál es el contenido que se encontrará al leerlo. La redacción sobre el contenido de la memoria debe hacerse en presente y todo lo referido al proyecto en pasado, siempre de modo impersonal.

5.1. Conclusiones generales

La idea de esta sección es resaltar cuáles son los principales aportes del trabajo realizado y cómo se podría continuar. Debe ser especialmente breve y concisa. Es buena idea usar un listado para enumerar los logros obtenidos.

En esta sección no se deben incluir ni tablas ni gráficos.

Algunas preguntas que pueden servir para completar este capítulo:

- ¿Cuál es el grado de cumplimiento de los requerimientos?
- ¿Cuán fielmente se pudo seguir la planificación original (cronograma incluido)?
- ¿Se manifestó algunos de los riesgos identificados en la planificación? ¿Fue efectivo el plan de mitigación? ¿Se debió aplicar alguna otra acción no contemplada previamente?
- Si se debieron hacer modificaciones a lo planificado ¿Cuáles fueron las causas y los efectos?
- ¿Qué técnicas resultaron útiles para el desarrollo del proyecto y cuáles no tanto?

5.2. Próximos pasos

Acá se indica cómo se podría continuar el trabajo más adelante.