

Desafío OBDH - CHASQUI II

En el presente trabajo primero se hara una breve definición de algunos conceptos necesarios para comprender las funciones del subsistema OBDH, luego definiremos los telecomandos usados para el desafío con su respectiva tabla de transiciones y por último mostraremos las salidas de la simulación realizada en Stateflow.

I. INTRODUCCIÓN

El objetivo de este desafío consiste en modelar y simular el comportamiento del OBDH, utilizando Simulink/Stateflow, las entradas usadas fueron brindadas por la comisión organizadora del evento.

Nuestro desafío consiste en la simulación de un plan de vuelo, para su implementación realizamos la definición de los telecomandos así como el nivel de tensión de la batería y la visibilidad del satélite frente a la estación terrestre, esto con el fin de que el subsistema OBDH opere en determinados modos.

Las arquitecturas del sistema de manejo de datos a bordo (OBDH) forman la base para administrar los datos en una nave espacial. Los datos suelen ser el motor principal y la motivación de las misiones espaciales. Debido a limitaciones como la potencia disponible y el ancho de banda de comunicación, el diseño de sistemas OBDH requiere un análisis detallado de los requisitos para los cuatro elementos de datos primarios: fuentes, sumideros, transferencias y procesamiento [1].

II. FUNDAMENTO TEÓRICO

Para comprender mejor el desafío y el subsistema de manejo de la información es necesario mencionar algunos conceptos importantes.

A. Máquina de estado u autómata finito

Es un tipo de automatización que utiliza un número finito de estados en su ejecución. Estos estados son modelos matemáticos que agrupan todas las posibles ocurrencias del sistema (Debe abarcar todas las interacciones viables entre estados).

Otro aspecto importante de las máquinas de estado son las transiciones, ya que también permiten añadir condiciones en función de las cuales saltar de un estado a otro. Éstas se representan mediante flechas o ramas entre estados.

B. OBDH

Subsistema de los satélites o naves espaciales los cuales transporta y manipula data entre los distintos dispositivos electronicos que compone la nave espacial o satélite y la estación terrena. También incluye funcionalidades autónomas de gestión de fallos para garantizar que la

nave espacial pueda recuperarse de forma autónoma de anomalías importantes y entrar en un estado seguro sin la interacción de los operadores terrestres, en caso de emergencia.

C. Estación Terrena

La transmisión por satélite consiste en una ruta de propagación en línea recta desde una estación terrestre hasta un satélite de comunicaciones (enlace ascendente) y de vuelta a una estación terrestre (enlace descendente). La estación terrestre incluye las antenas, los edificios y la electrónica necesaria para transmitir, recibir, multiplexar y demultiplexar las señales [2]. La comunicación entre la estación terrena y el satélite se basa en una transmisión de paquetes de comandos (telecomandos) que se utiliza para que el satélite entre en distintos modos de operación.

III. PROCEDIMIENTO

A. Modos de operación

Según el reglamento se ha considerado los siguiente modos de operación que pueden ser alcanzados por el nanosatélite durante la misión:

a. Off Es el estado alcanzado debido a la falta de alimentación por parte del EPS o por el telecomando Hard Reset. En Stateflow hemos definido este estado como el estado por defecto, además puede ser alcanzado desde cualquier otro modo de operación.

b. Boot Estado de inicio del software, es alcanzado una vez se ha superado cierto voltaje umbral (Mínimo voltaje requerido para el buen funcionamiento del subsistema). Se ha considerado un voltaje de 5 V debido a que es con el que trabajan la mayor cantidad de micro-controladores. Tenemos que como condición de salida al siguiente estado se debe cumplir `deploy_antennas == 1`.

c. Deployment Estado en donde se produce el despliegue de las antenas, para simular la salida de este estado después de 10 intentos fallidos se ha usado una variable que funcione como contador (trydeploy), la otra forma de salida es mediante un telecomando.

d. Safe Debe ser alcanzado luego de la salida del estado Deployment o por telecomando, además debido a que es el estado operacional del satélite cualquier falla en el modo Nominal o Transmitido debe retornar a este estado. La salida solo se puede dar por telecomando, es decir si leemos un valor distinto a -1 (Ver tabla II) se dará una transición de estados.

Tabla I. Tabla de transiciones.

Actual/Siguiente	Off	Boot	Deployment	Safe	Nominal	Transmitiendo
Off	-	$V_{baterfa} > 5 \text{ V}$	-	-	-	-
Boot	$tc == 0$	-	$contBoot == 360$ $trydeploy == 11$	-	-	-
Deployment	$tc == 0$	$tc == 1$	or $deploy_antennas == 1$	-	-	-
Safe	$tc == 0$	$tc == 1$	-	-	$tc == 2$	$tc == 4$ and $suntime == 1$
Nominal	$tc == 0$	$tc == 1$	-	$tc == 3$	-	$tc == 4$
Transmitiendo	-	$tc == 1$	-	$tc == 3$	$tc == 2$ $suntime == 0$	-

e. *Nominal* Principal estado operacional del satélite. Solo se puede pasar al estado “Safe” en caso de falla o telecomando.

f. *Transmitiendo* Transmisión de los datos de la carga útil (Usualmente es la imagen tomada por la cámara o información de los sensores). Solo es alcanzable por telecomando o cuando la estación terrena es visible, esto último implica que es necesario definir una latitud y longitud a partir de la cual puede iniciar el contacto con tierra. En caso de que no se encuentre en el intervalo de alcance de la estación terrena pasará al estado nominal.

Todo lo mencionado anteriormente puede ser sintetizado en la tabla I.

B. Telecomandos

Un telecomando es la información enviada desde la estación terrena que permite acciones de control sobre el nanosatélite, su forma más habitual es como direcciones hexadecimales que son desentramadas por el sistema de telemetría. En el desafío hemos considerado oportuno facilitar el trabajo definiendo los telecomandos como números enteros tal y como se indica en la tabla II, en una situación real se implementaría como una interrupción en la cual necesitaríamos definir prioridades.

Nosotros hemos definido los telecomandos en el Workspace de MATLAB como un objeto timseries de tal forma que una lectura continua de -1 nos indica que no se esta recibiendo información desde la estación terrena, mientras que un valor de 0 a 5 permita entrar en un modo de operación específico al CubeSat.

Tabla II. Lista de telecomandos usados.

Telecomando	Modo de Operación
-1	Sin acciones.
0	Hard Reset
1	Soft Reset
2	Nominal
3	Safe
4	Transmitiendo

IV. RESULTADOS

El modelo final puede verse en la figura 1, definimos la variable output “state” para saber en que modo de operación se encuentra en cada instante de tiempo ya que se modifica su valor cuando entra a un nuevo estado (para esto usamos “entry”). Para seguir el plan de vuelo pedido (Ver tabla III) “enviamos” los telecomandos usando el archivo main.m de tal forma que se den las transiciones en el momento adecuado de la simulación, esto lo podemos ver en la figura 2, esto fue realizado según la tabla IV.

Tabla III. Plan de vuelo.

Modo de operacion	
0	Off
1	Boot
2	Deployment
3	Safe
4	Nominal
5	Transmitiendo
4	Nominal
5	Transmitiendo
4	Nominal
3	Safe
4	Nominal

Otra de las salidas requeridas es el voltaje de la batería a lo largo de la misión (ver figura 3), acá notamos que existen pequeñas variaciones (el máximo valor es 8.5V y el mínimo es 8.0795V) y el estado de las antenas (si han sido desplegadas toman el valor de 1, en caso contrario se espera una salida de 0).

Para el correcto envío de datos es necesario que se cumplan dos condiciones, la primera es que el despliegue de las antenas se haya dado de forma correcta y el segundo es que el CubeSat sea visible desde la estación, para esto último se han definido una longitud y una latitud a partir de la cual es visible durante 10 minutos (120 pasos de simulación). Sin embargo esto trae consigo ciertos problemas ya que le quita robustez a nuestro código lo cual hara complicado el que siga otro plan de vuelo ya que los valores de latitud y longitud variarán considerable-

mente nuestros resultados, las fuentes de error se dan en el momento del envío de los telecomandos ya que lo estamos haciendo usando la variable telecomandos la cual es un arreglo de 17281 elementos por lo cual puede resultar complicado saber en cual momento enviárselo (es decir cual es el índice del array que hará que siga el plan de vuelo indicado).

Tabla IV. Modos de operación.

#	Modo de Operación
0	Off
1	Boot
2	Deployment
3	Safe
4	Nominal
5	Transmitiendo

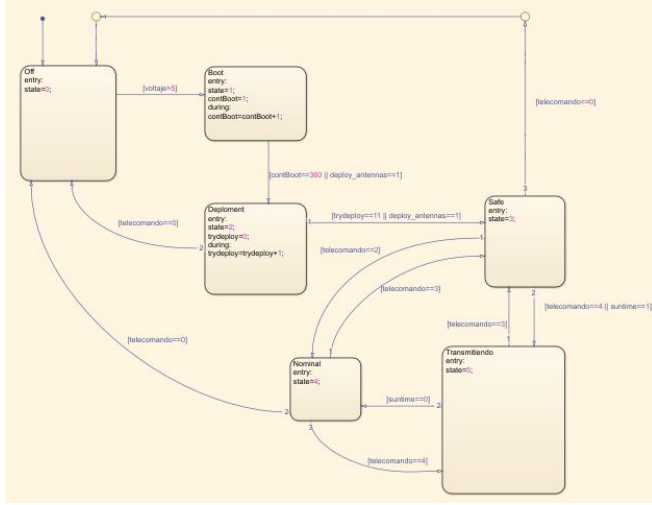


Figura 1. Modelo en Stateflow.

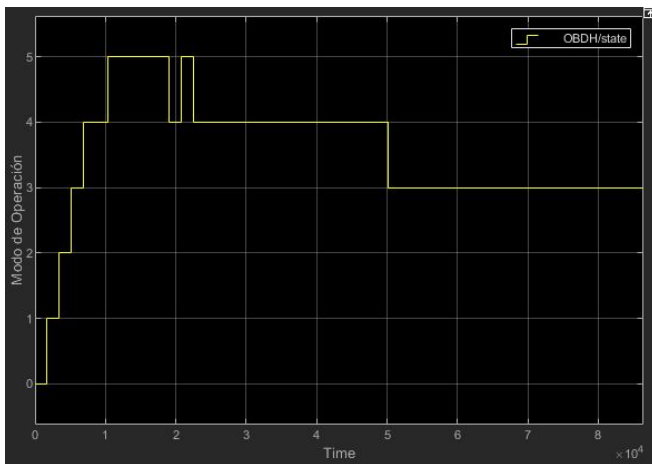


Figura 2. Modos de operación alcanzados durante la simulación según la tabla IV y considerando el plan de vuelo pedido.

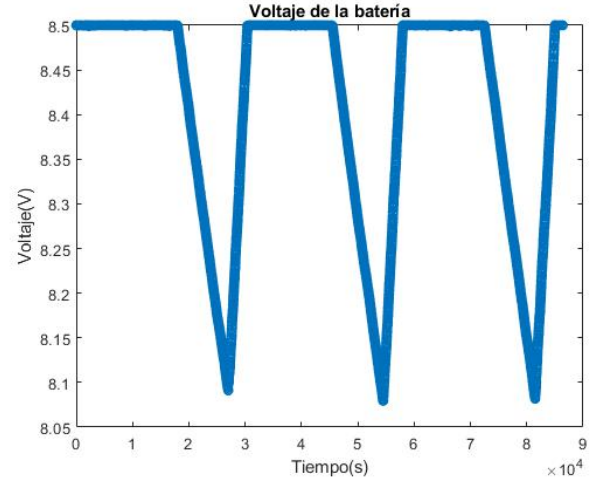


Figura 3. Voltaje de la batería a lo largo de la misión.

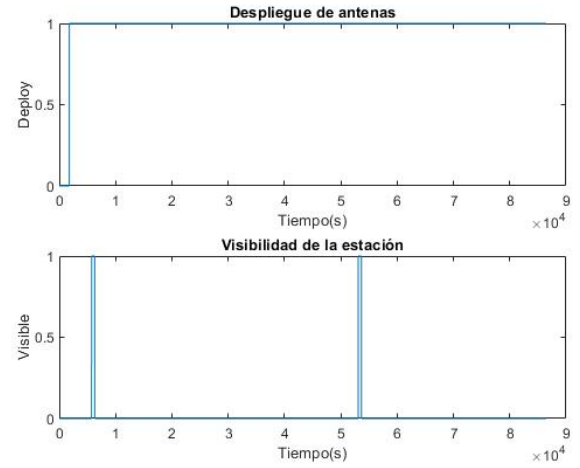


Figura 4. Visibilidad de la estación terrena.

Para calcular la potencia consumida usamos como referencia los componentes del nanosatélite ISRASAT1[3] los cuales pueden verse en la tabla V. Los voltajes y corrientes son datos provenientes de sus respectivos datasheets, resultando $P = 14,95mW$. Este valor correspondería al modo Nominal ya que estarían encendidos todos los equipos.

Tabla V. Hardware del nanosatélite ISRASAT1.

Componente	Función
ATMEGA32	Trabaja como el cerebro del microcontrolador.
LM35	Mide la temperatura interna del OBDH.
DS1307	Provee el tiempo real.
AT24C64	Guarda los datos del sensor de temperatura.

V. CONCLUSIONES

Se logró simular satisfactoriamente el plan de vuelo de la tabla III así como las gráficas del voltaje de salida y de la visibilidad de la estación terrena. Además fuimos capaces de hacer una primera aproximación de la potencia consumida por el nanosatélite en el estado Nominal .

-
- [1] NASA, *Cubesat Handbook, From Mission Desing to Operations* (2021).
 - [2] R. Meyers, *Encyclopedia of Physical Science and Technology*, 2nd ed. (2003).
 - [3] D. A. M. Osman and S. W. A. Mohamed, Hardware and software design of onboard computer of israsat1 cubesat, in *2017 International Conference on Communication, Control, Computing and Electronics Engineering (ICCC-CEE)* (2017) pp. 1–4.