

# Capítulo 1

## Modelo de la Aeronave

Este bloque es con diferencia el más complejo y el que más explicación necesita, por lo que se tratará de poner especial detalle en los aspectos más complejos, así como las diferentes funcionalidades adicionales que se han dejado implementadas y los aspectos inacabados que requieren de revisión.

En este bloque, mostrado en la figura 1.1, se calculan las fuerzas que actúan sobre la aeronave, se integran en el tiempo y se obtienen las variaciones de posición y actitud que experimenta la aeronave.

Este procedimiento se puede separar en varias partes claramente diferenciadas, véase, una correspondiente al cálculo de las fuerzas y momentos aerodinámicos a partir del estado de la aeronave y las derivadas de estabilidad, otra encargada del cálculo del sistema propulsivo y las fuerzas y momentos que éste genera, otro bloque encargado del cálculo de la variación del centro de gravedad y los momentos de inercia, un subsistema encargado de la integración de estas fuerzas en el dominio del tiempo y finalmente una última parte donde se estructuran los datos obtenidos y se almacenan en un bus de estados denominado “*PlantDataBus*” para facilitar su comunicación con otros bloques.

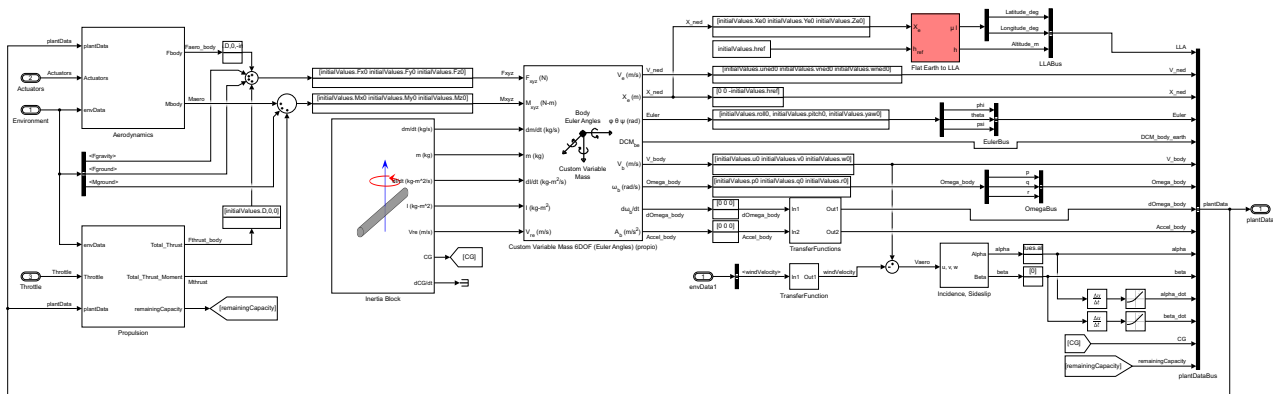


Figura 1.1: Vista general del modelo de la planta de la aeronave

## 1.1. Aerodinámica

Para el cálculo de las fuerzas y momentos aerodinámicos se ha implementado el esquema mostrado en la figura 1.2, que consiste en la obtención de los valores de los coeficientes aerodinámicos en ejes estabilidad a partir de los valores de las derivadas de estabilidad correspondientes a los valores instantáneos de  $\alpha$ ,  $\beta$ , altitud, posición del centro de gravedad y deflexión existente de las superficies aerodinámicas.

Por último, se dimensionalizan dichos coeficientes aerodinámicos obtenidos previamente con la presión dinámica y las magnitudes de referencia declaradas en “/dataImport/Stability/loadStabilityData.m” y se obtienen las fuerzas y momentos en cada eje.

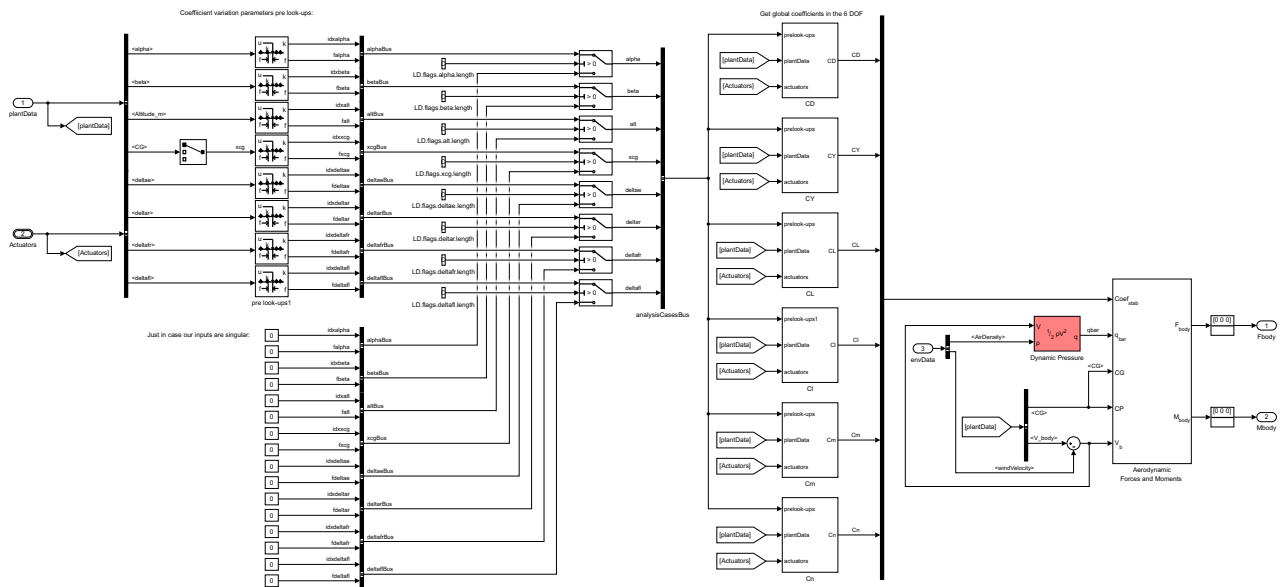


Figura 1.2: Esquema del modelo aerodinámico

### 1.1.1. Coeficientes de Estabilidad

El proceso de definición e importación de los coeficientes de estabilidad, o derivadas de estabilidad, se ha implementado de una forma, a priori poco intuitiva, pero que se ha considerado que aporta una gran versatilidad y logra solventar los problemas asociados a acoplar en el mismo proceso de cálculo la aerodinámica del vuelo como ala fija y como quadrotor, por lo que este paso se explicará en detalle.

En los simuladores de aeronaves de ala fija, es habitual seguir el procedimiento indicado anteriormente de definir las derivadas de estabilidad, calcular con ellas los coeficientes aerodinámicos y a partir de éstos las fuerzas y momentos aerodinámicos. Sin embargo, este procedimiento sólo es válido cuando las hipótesis hechas al calcular estas derivadas se siguen cumpliendo, es decir, ángulos de ataque y de resbalamiento pequeños, etc. Si bien, en el vuelo habitual de una aeronave éstas hipótesis se cumplen, esto ya no es así cuando se realiza un vuelo como quadrotor, situación en la que el ángulo de entrada de la velocidad proporciona valores de  $\alpha$  y  $\beta$  que distan mucho de ser pequeños.

A su vez, para modelizar la posibilidad de entrada en pérdida, o para poder simular las variaciones de actuaciones en función del centrado de la aeronave, es necesario definir que las derivadas de estabilidad dejen de ser constantes y que experimenten variaciones con una serie de parámetros. Pese a que en la vida real los parámetros de los que dependen son numerosos, de momento se han seleccionado los ocho parámetros siguientes: ángulo de ataque ( $\alpha$ ), ángulo de resbalamiento ( $\beta$ ), altitud de vuelo, posición longitudinal del centro de gravedad ( $x_{cg}$ ), deflexión del estabilizador horizontal ( $\delta_e$ ), deflexión del timón de dirección ( $\delta_r$ ), deflexión del flaperon derecho ( $\delta_{fr}$ ) y deflexión del flaperon izquierdo ( $\delta_{fl}$ ). Sin embargo, como se verá, resulta extraordinariamente sencillo variar éstos, o cambiar el número de parámetros de los que depende dicha variación.

Toda la definición relacionada con la aerodinámica o la estabilidad se realiza desde el script “*/dataImport/Stability/loadStabilityData.m*”. En él se deben declarar las variables de referencia, véase, superficie, cuerda, envergadura y posición longitudinal del borde de ataque. A su vez, se definirán en el campo “*LD.Stability.analysisCases*” el nombre de los parámetros de los que se ha decidido que dependan los coeficientes de estabilidad y en el campo “*LD.Stability.Coeffs*” los nombres de los diferentes coeficientes.

A continuación se declararán, en los campos “*LD.XXX*”, los valores de cada uno de los parámetros para los que se dispone de información ordenados de menor a mayor, donde “*XXX*” hace referencia al nombre definido para dicho parámetro, de forma que posteriormente Simulink interpolará linealmente los valores de cada coeficiente en función del valor instantáneo de cada parámetro entre los dos valores más cercanos de los aquí definidos. Por ejemplo, si uno de los parámetros es el ángulo de ataque y hemos definido que su nombre sea “*alpha*”, se deberá definir un vector con los valores de ángulo de ataque para los que se han calculado las derivadas de

estabilidad y almacenarlo en el campo “*LD.alpha*”. Es importante tener en cuenta que ningún parámetro puede quedarse sin definir, de forma que aunque en alguna situación no consideremos variación en ese parámetro, se debe indicar el valor para el que han sido calculadas las derivadas de estabilidad.

Por último, para introducir los valores de cada derivada de estabilidad, se debe crear una matriz del mismo número de dimensiones que el número de parámetros definidos, respetando el orden definido en “*LD.Stability.analysisCases*”, donde el tamaño de cada dimensión sea acorde a la longitud del vector de valores de dicho parámetro y almacenarla en “*LD.Stability.XXX*”, donde “*XXX*” hace referencia al nombre definido para dicho coeficiente.

Es decir, siguiendo con el ejemplo anterior, si los valores definidos en “*LD.alpha*” son  $[-\pi, -\pi/2, 0, \pi/2, \pi]$ , y “*alpha*” es el primer y único parámetro de los ocho declarados que experimenta variaciones, la matriz definida en “*LD.Stability.CD0*” deberá ser de  $[5 \times 1 \times 1 \times 1 \times 1 \times 1 \times 1 \times 1]$  donde el primer valor se corresponda con el primer valor definido de “*alpha*”, el segundo con el segundo y así sucesivamente. Si en vez de experimentar variaciones únicamente el ángulo de ataque, se definen también 10 valores de variaciones con la altitud (definida la tercera en el campo “*LD.Stability.analysisCases*”) la matriz de cada coeficiente deberá ser de  $[5 \times 1 \times 10 \times 1 \times 1 \times 1 \times 1 \times 1]$  y cada uno de los 50 valores de cada coeficiente resultantes se deben colocar en su posición adecuada de la matriz.

Como realizar este proceso de colocación de cada coeficiente de forma manual es tremendamente costoso y muy propenso a cometer errores en cuanto se definen variaciones con varios parámetros no consecutivos, se ha elaborado el script “*/dataImport/Stability/fillStabilityCoeffs.m*” que se encarga de realizarlo de forma automática.

Para ello, se le deben pasar como argumentos: la estructura “*LD*” de la que leerá los parámetros definidos, un vector de campos de texto con los nombres de los parámetros que experimentan variaciones y la matriz con los valores de los coeficientes, donde la primera dimensión de ésta representa la variación con el primer parámetro definido en el segundo argumento, la segunda dimensión la variación con el segundo parámetro y así sucesivamente.

En el caso de que haya un único parámetro que varíe, por comodidad, no es necesario introducirlo como vector columna (lo cual sería introducirlo en la primera dimensión de una matriz), sino que acepta indistintamente un vector fila o columna.

Si en algún momento se decide cambiar o aumentar el número de parámetros de los que dependen las derivadas de estabilidad, sólo es necesario modificar los campos de “*LD.Stability.analysisCases*” y modificar el nombre y número de prelookups del modelo aerodinámico de Simulink para que refleje dicha variación.

### 1.1.2. Validación de los coeficientes

Dado que el proceso de introducción de los coeficientes es complejo y pueden haber existido errores al introducirlos que impidan al modelo de Simulink su correcta lectura, como que sea diferente el número de parámetros establecidos en “*LD.Stability.analysisCases*” que el número de dimensiones de las matrices de cada coeficiente, que se hayan indicado más coeficientes en el campo “*LD.Stability.Coeffs*” que los finalmente declarados o que alguno de los parámetros se encuentre vacío, después de cargar los coeficientes de estabilidad se ejecuta desde “*/dataImport/loadData.m*” el script de validación “*/utilities/dataValidations.m*”. Este script es el encargado de comprobar estos casos y, en caso de que se hayan producido, mostrar por pantalla una advertencia para solucionarlo, junto con la pertinente información para localizar fácilmente la causa de dicho error, así como de dejar constancia de ello activando las flags pertinentes dentro de la estructura de “*checks*”.

Otra situación detectada como error y que este script debe solucionar, es el caso de que en un parámetro no se consideren variaciones, ya que el procedimiento de interpolación de los valores de los coeficientes a partir de prelookups necesita de al menos un mínimo de dos valores. Para solucionar esto, se añade al final de cada vector de parámetros un nuevo elemento que no se usará y en la dimensión pertinente de cada coeficiente se añade un elemento más. A su vez, se activa un flag de que dicho parámetro no experimenta variación, de forma que Simulink no use en ese caso la información proveniente de los prelookups, utilizando directamente la información proporcionada por el usuario.

## 1.2. Planta Propulsiva

El LIBIS obtiene la tracción necesaria para el vuelo de cinco motores eléctricos alimentados mediante dos baterías conectadas en paralelo. Cuatro de ellos se encuentran ubicados en las puntas de las alas, proporcionando tracción en el eje vertical para el vuelo como quadrotor y otro ubicado en la cola, empujando a la aeronave y proporcionando tracción para el vuelo como ala fija.

Para poder estudiar la variación de las actuaciones y actitud en función de cada uno de los motores, éstos se han definido de forma independiente, tal y como se observa en la figura 1.3, siguiendo el siguiente orden: el primer motor es el de la cola, el encargado del vuelo horizontal, el segundo motor es el correspondiente al semiala delantera derecha, y se sigue en sentido horario, siendo el tercero el de la semiala trasera derecha, el cuarto el de la trasera izquierda y el quinto motor el de la delantera izquierda.

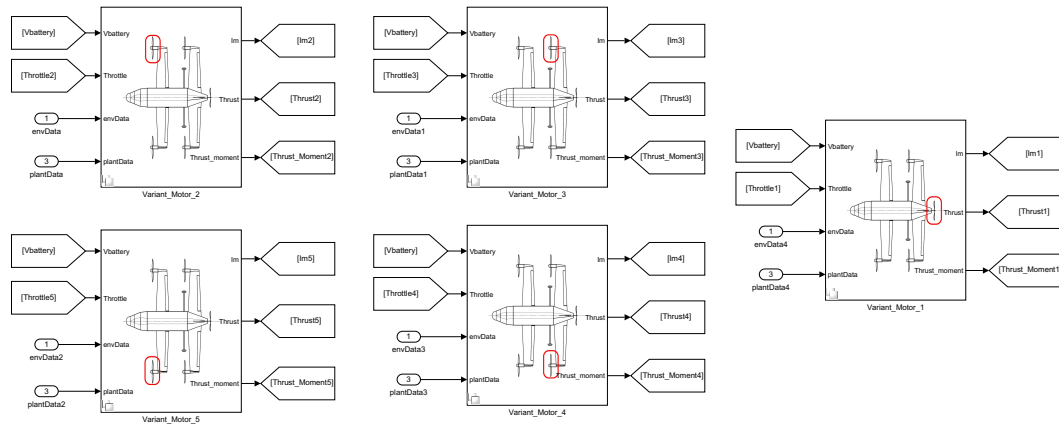


Figura 1.3: Modelos independientes para cada uno de los motores

Los datos de cada uno de los motores, de cada una de las hélices y de cada uno de los variadores se definen en el script “/dataImport/Propulsion/loadPropulsionData.m” junto con las flags pertinentes para elegir el método de cálculo del sistema propulsivo deseado, véase, que en cada instante de ejecución Simulink resuelva el acoplamiento entre la hélice y el motor para calcular la tracción, lo cual es más lento pero permite probar diferentes motores y hélices de forma instantánea, o que al cargar el proyecto se ejecute un script que calcule el acoplamiento para un determinado rango de velocidades de vuelo, alturas y voltajes de salida del variador, lo cual permite agilizar la simulación pero requiere un significativo tiempo de cálculo cada vez que se modifican los motores o las hélices.

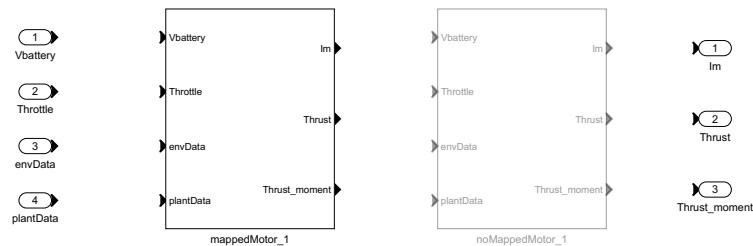


Figura 1.4: Definición de la planta propulsiva como “*Simulink Variants*”

Para posibilitar esta dualidad en el cálculo del sistema propulsivo, se ha definido cada motor como un “Simulink Variant”, tal y como se muestra en la figura 1.4, de forma que en función de los flags indicados anteriormente se encuentre activo uno u otro modelo.

### **1.2.1. Fundamento Teórico**

### **1.2.2. Obtención de datos del motor y la hélice**

### **1.2.3. Motores mapeados**

### **1.2.4. Motores sin mapear**

#### **1.2.4.1. Modelo del motor eléctrico**

#### **1.2.4.2. Modelo de la hélice**

#### **1.2.4.3. Modelo de torque**

### **1.2.5. Batería**

La batería se encuentra declarada dentro del propio bloque de Simulink por medio de constantes tal y como se muestra en la figura 1.13.

En el momento de declarar la batería no se encontró una ventaja significativa en crear expresamente un script de MATLAB para ello.

Echando la vista atrás, pudiese haber sido interesante de cara a mantener uniformidad con el resto de las declaraciones de parámetros que prácticamente la totalidad de ellos se realiza en scripts externos, sin embargo, queda pendiente para una futura ampliación el realizarlo, así como crear una biblioteca de baterías entre las que se pueda elegir para estudiar las actuaciones de la aeronave en función de ésta.

También queda pendiente para una futura fase de mejora el limitar la máxima descarga de la batería a través del parámetro de máxima descarga, garantizando de esta forma con el simulador que nunca se alcanza dicha tasa de descarga y que no peligra la batería.

Por último a partir de la capacidad de la batería y la profundidad de descarga, se puede calcular la capacidad nominal, de la cual se puede obtener en cada instante la

---

capacidad restante simplemente restandole la capacidad consumida hasta entonces.

### **1.3. Modelo de Inercia**



Name	Submodel Configuration	Variant Control	Condition
mainV03_52			
Actuators			
LIBIS Plant			
Propulsion			
Variant_Motor_1			
mappedMotor_1		mappedMotors==1	(N/A)
noMappedMotor_1		mappedMotors~=1	(N/A)
Variant_Motor_2			
mappedMotor_2		mappedMotors==1	(N/A)
noMappedMotor_2		mappedMotors~=1	(N/A)
Variant_Motor_3			
mappedMotor_3		mappedMotors==1	(N/A)
noMappedMotor_3		mappedMotors~=1	(N/A)
Variant_Motor_4			
mappedMotor_4		mappedMotors==1	(N/A)
noMappedMotor_4		mappedMotors~=1	(N/A)
Variant_Motor_5			
mappedMotor_5		mappedMotors==1	(N/A)
noMappedMotor_5		mappedMotors~=1	(N/A)
Sensors			

Name	Submodel Configuration	Variant Control	Condition
mainV03_52			
Actuators			
LIBIS Plant			
Propulsion			
Variant_Motor_1			
mappedMotor_1		mappedMotors==1	(N/A)
noMappedMotor_1		mappedMotors~=1	(N/A)
Variant_Motor_2			
mappedMotor_2		mappedMotors==1	(N/A)
noMappedMotor_2		mappedMotors~=1	(N/A)
Variant_Motor_3			
mappedMotor_3		mappedMotors==1	(N/A)
noMappedMotor_3		mappedMotors~=1	(N/A)
Variant_Motor_4			
mappedMotor_4		mappedMotors==1	(N/A)
noMappedMotor_4		mappedMotors~=1	(N/A)
Variant_Motor_5			
mappedMotor_5		mappedMotors==1	(N/A)
noMappedMotor_5		mappedMotors~=1	(N/A)
Sensors			

Name	Submodel Configuration	Variant Control	Condition
mainV03_52			
Actuators			
LIBIS Plant			
Propulsion			
Variant_Motor_1			
mappedMotor_1		mappedMotors==1	(N/A)
noMappedMotor_1		mappedMotors~=1	(N/A)
Variant_Motor_2			
mappedMotor_2		mappedMotors==1	(N/A)
noMappedMotor_2		mappedMotors~=1	(N/A)
Variant_Motor_3			
mappedMotor_3		mappedMotors==1	(N/A)
noMappedMotor_3		mappedMotors~=1	(N/A)
Variant_Motor_4			
mappedMotor_4		mappedMotors==1	(N/A)
noMappedMotor_4		mappedMotors~=1	(N/A)
Variant_Motor_5			
mappedMotor_5		mappedMotors==1	(N/A)
noMappedMotor_5		mappedMotors~=1	(N/A)
Sensors			

Figura 1.5: Valores de “*mappedMotors*” en función del modo de cálculo del sistema propulsivo deseado

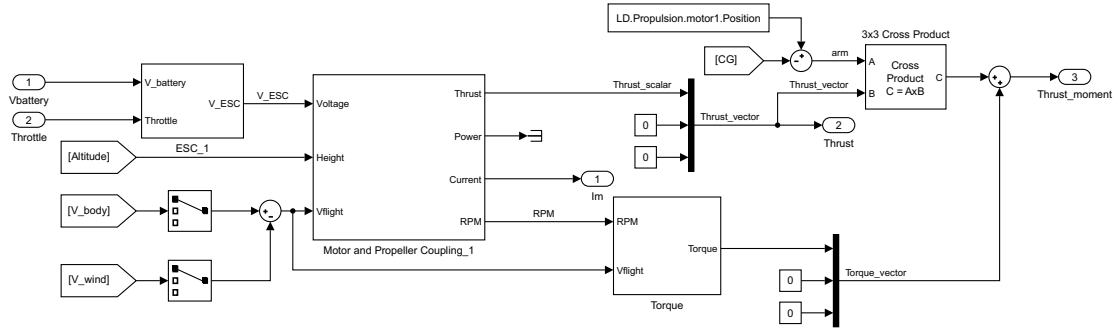


Figura 1.6: Modelo de la planta propulsiva mapeada

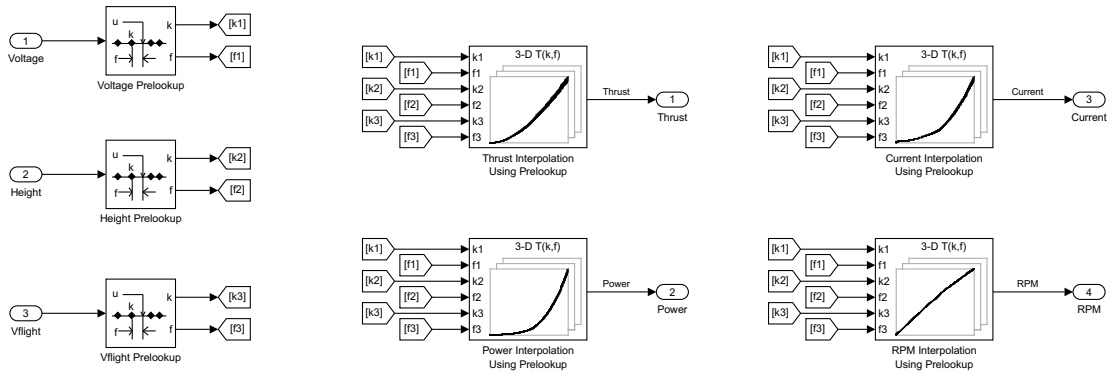


Figura 1.7: Acoplamiento del motor y la hélice

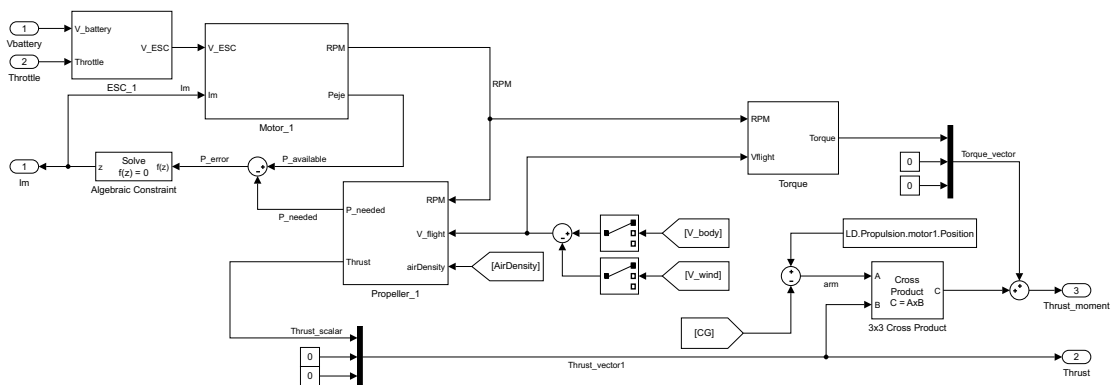


Figura 1.8: Modelo del sistema propulsivo sin necesidad de mapeado

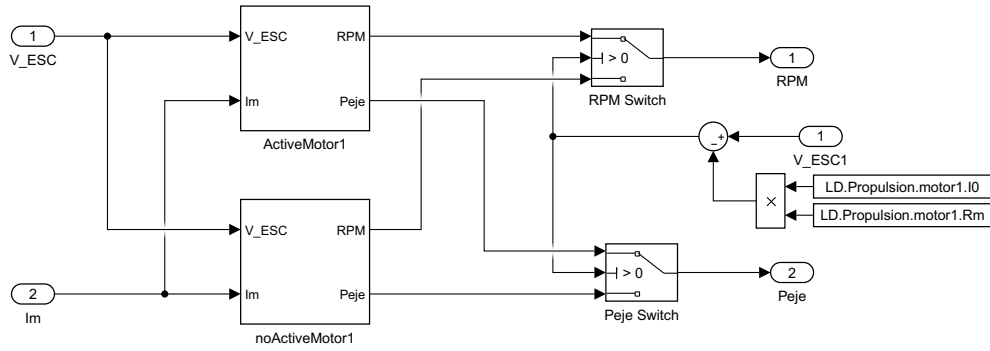


Figura 1.9: Esquema del modelo implementado para poder apagar el motor

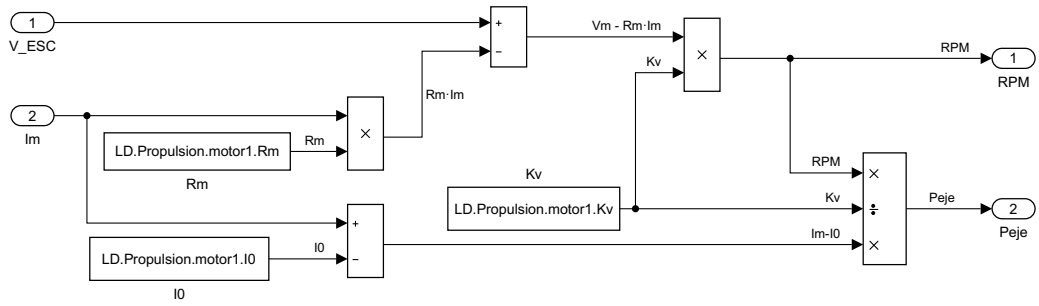


Figura 1.10: Modelo del motor eléctrico

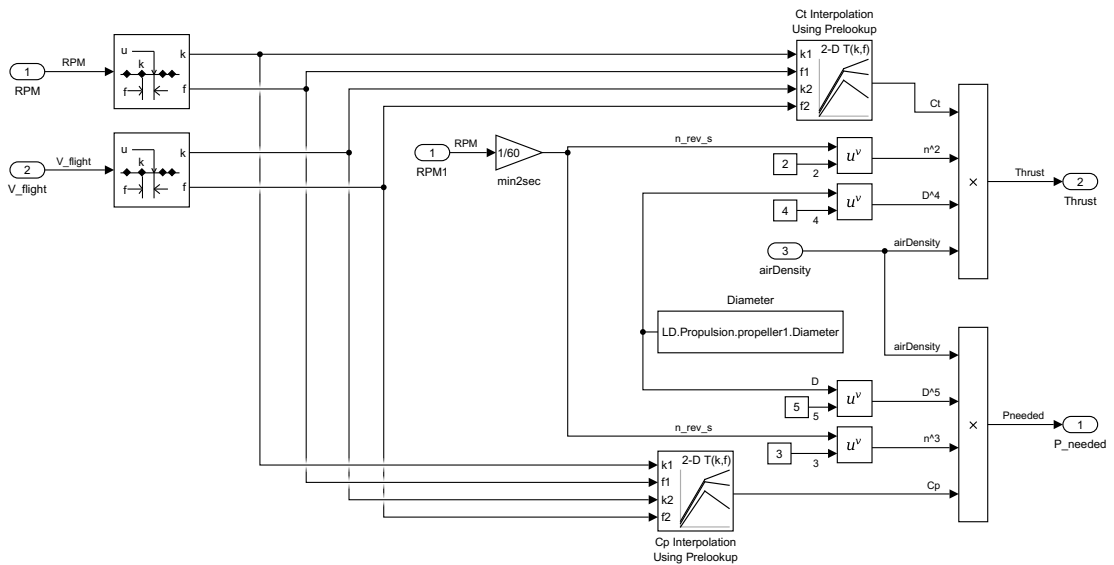


Figura 1.11: Modelo de la hélice a partir de prelookups

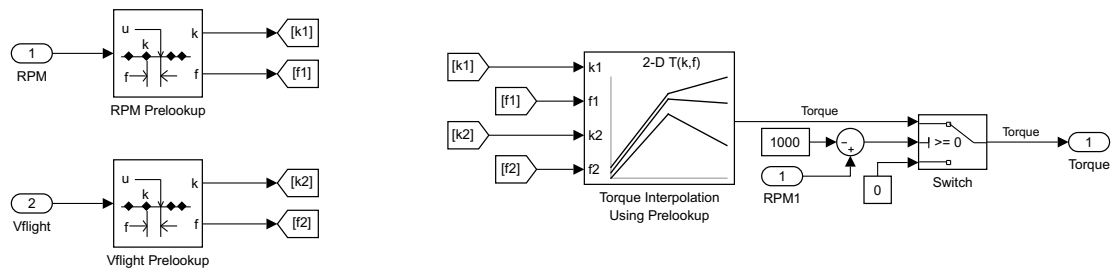
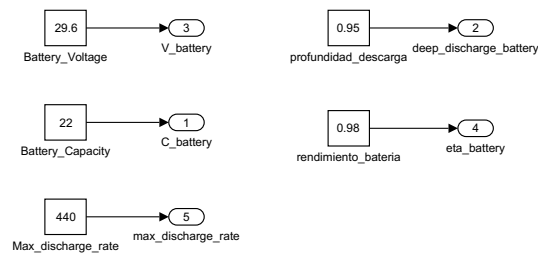


Figura 1.12: Implementación del modelo de torque

**MaxAmps LiPo 11,000 8S 29.6v Battery Pack**

- 11,000mah capacity
- 8S 29.6 volts
- True 40C rating
- Lifetime warranty
- 5C fast charge capable
- 100% waterproof
- Built today with factory fresh cells
- Built with genuine 12awg Deans Ultra wire
- We add the connector and JST-XH balancing tap for you
- 158mm x 59mm x 81mm, 1690g

<https://www.maxamps.com/li-po-11000-8s-29-6v-battery-pack>

**Hay 2 baterías en paralelo**

Figura 1.13: Datos de la batería

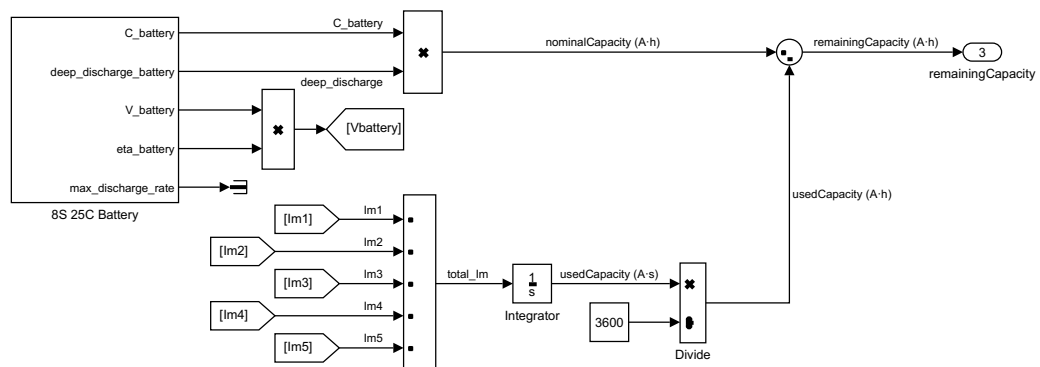


Figura 1.14: Esquema del cálculo de la autonomía restante

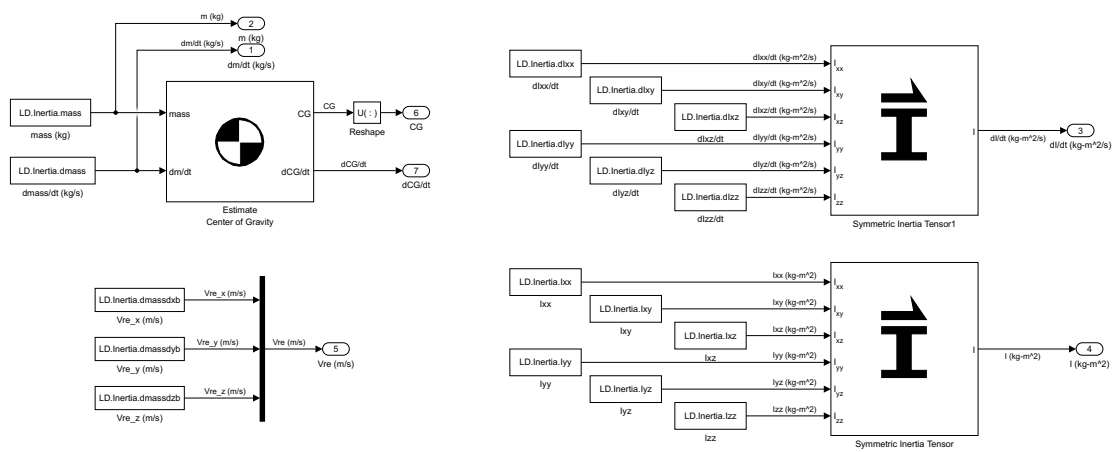


Figura 1.15: Modelo de inercia