

MUSE

Inst. Univ. de Microgravedad Ignacio da Riva
Plaza Cardenal Cisneros, 3
28040, Madrid
España

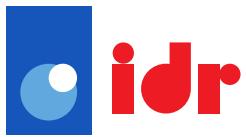
Caso de Estudio I

Módulo del Sistema de Control de Actitud para la Sala de Diseño Concurrente

Preparado por	Criado Pernía, David; Lubián Arenillas, Daniel
Dirigido por	Cubas Cano, Javier; Roibás Millán, Elena
Referencia	MUSE-1718-CE1-CDF-ADCS
Edición	1
Revisión	1
Fecha de edición	2018-06-19

Índice general

Índice de figuras	4
Índice de tablas	5
Nomenclatura	5
1 Introducción	7
1.1 Resumen	7
1.2 Objetivos	7
2 Resultados	9
2.1 Hojas de cálculo	9
2.2 Interfaz del catálogo	10
3 Conclusiones	13
3.1 Trabajo futuro	13
Referencias	15
A Guía del usuario	17
A.1 Descripción general	17
A.1.1 Flujo de trabajo	17
A.1.2 Nociones generales de funcionamiento	18
A.2 <i>Inputs</i>	18
A.3 <i>Outputs</i>	19
A.4 Hoja Overview	19
A.5 Hoja <i>Disturbance torques</i>	20
A.5.1 Teoría	20
A.5.2 Hoja de cálculo	23
A.6 Hoja <i>Sizing</i>	24
A.6.1 Teoría	25
A.6.2 Hoja de cálculo	28
A.7 Hojas <i>Actuators</i> y <i>Sensors</i>	28
A.7.1 Hoja <i>Actuators</i>	28
A.7.2 Hoja <i>Sensors</i>	29
A.8 Interfaz del catálogo	30
B Guía del desarrollador	33
B.1 Requisitos de instalación	33
B.2 Hoja de cálculo	33
B.3 Interfaz del catálogo	34
B.3.1 Código fuente de la interfaz	34

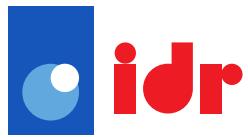


Índice de figuras

2.1 Hojas del archivo Excel clasificadas según su función	9
A.1 Flujo de trabajo del archivo Excel	17
A.2 Captura de pantalla de la vista resumen	21
A.3 Captura de pantalla de la hoja de perturbaciones orbitales.	24
A.4 Captura de pantalla de la hoja de dimensionamiento de actuadores.	28
A.5 Captura de pantalla de la hoja de actuadores	29
A.6 Captura de pantalla de la hoja de sensores	29
A.7 Captura de pantalla de la pestaña de la base de datos	31
A.8 Captura de pantalla de la pestaña del modelo	31
A.9 Captura de pantalla del cuadro de diálogo de especificaciones	32
A.10 Captura de pantalla de la pestaña de creación de elementos	32

Índice de tablas

A.1	<i>Inputs</i> del módulo.	18
A.2	<i>Outputs</i> del módulo.	19
A.3	Valores seleccionables en la sección <i>Characteristics</i> en la hoja <i>Overview</i> .	19
B.1	Códigos RGB de los colores empleados en la hoja de cálculo, con su significado.	34



Nomenclatura

ADCS *Attitude Determination and Control System* (Sistema de Determinación y Control de Actitud)

CDF *Concurrent Design Facility* (Sala de Diseño Concurrente)

ESA *European Space Agency* (Agencia Espacial Europea)

LEO *Low Earth Orbit* (Órbita Baja Terrestre)

1. Introducción

1.1. Resumen

Este Caso de Estudio consiste en el desarrollo de un módulo para la CDF del Instituto de Microgravedad “Ignacio da Riva” para simplificar los cálculos a realizar en el diseño preliminar de un sistema de determinación y control de actitud. Para ello, se ha tomado como base el módulo desarrollado por la ESA [1, 2] y se ha vuelto a implementar siguiendo las recomendaciones establecidas en anteriores trabajos en el IDR [3]. En este proceso, se ha mejorado el módulo introduciendo un catálogo de unos sesenta componentes de los principales suministradores del sector, al que se puede acceder de forma fácil y clara con la aplicación desarrollada.

1.2. Objetivos

Para el desarrollo de este Caso de Estudio se plantean los siguientes objetivos:

- Revisar los módulos disponibles de otras CDF y de desarrollos previos para estudiar si es necesario replantear el módulo o se puede partir de una base previa.
- Desarrollar un módulo de acuerdo a los criterios establecidos en trabajos previos para la CDF del IDR.
- Introducir un catálogo de componentes completo, fácilmente accesible e integrado con la hoja de cálculo en Excel.
- Escribir una guía de usuario para facilitar el manejo del módulo desarrollado.
- Escribir una guía del desarrollador para facilitar la creación de futuros módulos usando este módulo como base, o la mejora de este.

2. Resultados

2.1. Hojas de cálculo

El archivo Excel está comprendido por una serie de hojas cuya misión es atender a los diferentes aspectos del sistema control de actitud de un satélite. En la hoja *Design Process* se recogen los pasos a seguir por el usuario para operar el módulo, a modo de guía general. En la hoja *Overview* se recogen las características y configuraciones más importantes del sistema, como son el presupuesto de masa y potencia, el rango operacional de temperaturas y los distintos componentes del sistema. La hoja *Disturbance torques* se pone a disposición del usuario para estimar los momentos de perturbación típicos de una misión espacial. En la hoja *Sizing* se establece una relación entre dichas perturbaciones y los requisitos mínimos de los actuadores. En las hojas *Actuators* y *Sensors* se muestran los componentes del sistema seleccionados para la misión.

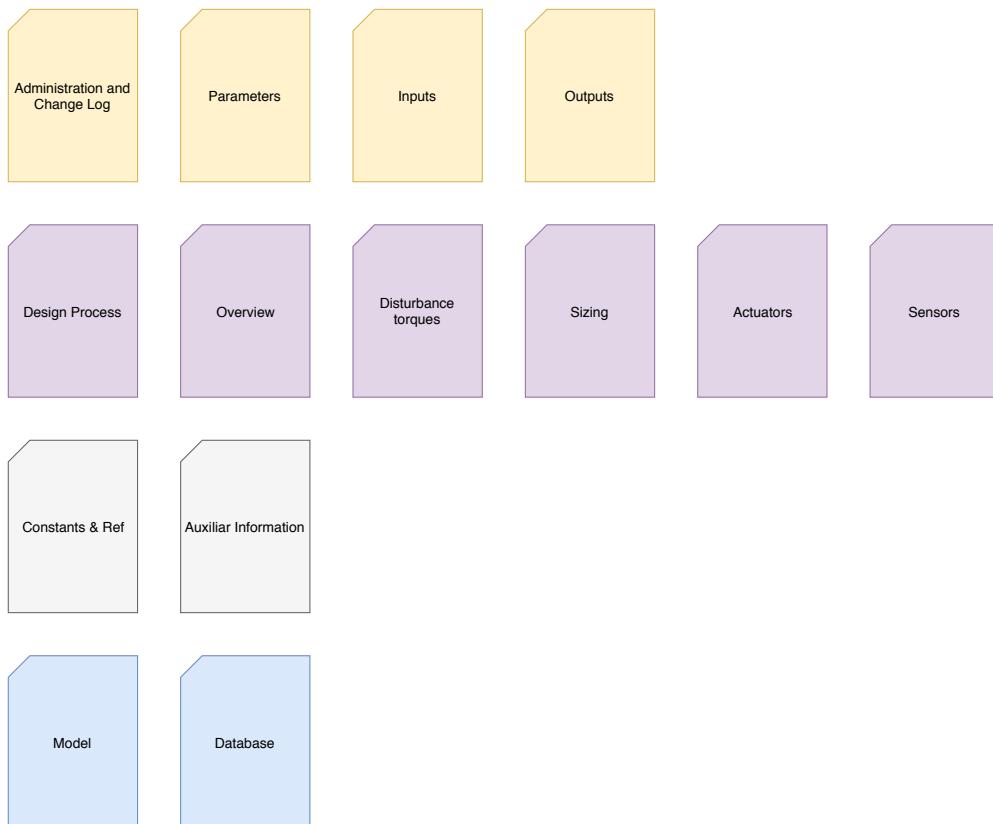
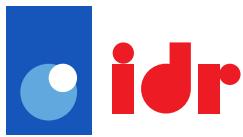


Figura 2.1: Hojas del archivo Excel clasificadas según su función

Además de estas hojas, destinadas a las operaciones de los usuarios, el libro de Excel consta de las hojas involucradas en el proceso de intercambio de datos con el resto de módulos:



- *Administration and Change Log*
- *Parameters*
- *Inputs*
- *Outputs*

Por otra parte, se han dispuesto otras hojas con carácter auxiliar para alojar las constantes empleadas en los cálculos, las referencias utilizadas y las opciones de los desplegables:

- *Constants & Ref*
- *Auxiliar Information*

Por último, en las hojas de *Model* y *Database* se guardan los datos del modelo y el catálogo utilizados por la interfaz de usuario y por las hojas de manipulación del archivo Excel.

2.2. Interfaz del catálogo

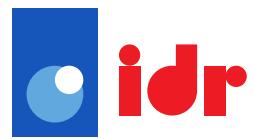
La principal segmentación de las tareas en la interfaz del catálogo se realiza dividiendo su espacio en las siguientes pestañas:

- *Model*
- *Database*
- *Create item*
- *Settings*

La pestaña *Model* tiene como función presentar una vista resumen del modelo seleccionado por el usuario en la interfaz, a través de una cuadrícula donde cada recuadro pertenece a cada una de las categorías de sensores o actuadores definida. Cabe comentar que dichas clasificación en categorías surge como resultado de la búsqueda de elementos para la construcción de la base de datos. El modelo seleccionado en la interfaz no es llevado a Excel hasta que el usuario lo decide, pudiendo realizar las modificaciones que estime oportunas antes de guardar el resultado en la hoja *Model* del archivo.

La pestaña *Database* sirve para extraer las entradas en la base de datos y agregar el elemento deseado al modelo. La búsqueda se filtra por categoría principalmente, aunque también es posible filtrar por masa, potencia o rango de temperaturas operacional. Además, los elementos del resultado de la búsqueda se pueden ordenar por nombre, fabricante, masa o potencia, siempre en orden ascendente. Para agregar un elemento, se ha de seleccionar en la tabla, elegir el número de unidades deseado tras lo cual se pulsa el botón de añadir.

La pestaña *Create item* consta de un formulario con todas las características de interés para incluir un nuevo elemento en la base de datos. Tras pulsar el botón de guardado, el elemento se añade a la hoja *Database* del archivo Excel. El identificador se genera de manera aleatoria, convirtiendo un número racional aleatorio entre 0 y 1 a formato hexadecimal.



Finalmente, se enumeran los proveedores cuyos componentes se han incluido en el catálogo:

- GomSpace
- Lens R&D
- CubeSpace
- MAI
- SolarMEMS
- NewSpace Systems
- SolarMEMS
- SolarMEMS
- NewSpace Systems
- NST-1
- Clyde Space
- Makesat
- ISIS
- Blue Canyon Technologies
- SODERN
- CubeSpaceZA

3. Conclusiones

En este Caso de Estudio se ha llevado a cabo el desarrollo de un módulo para la CDF del Instituto de Microgravedad “Ignacio da Riva” que permitirá simplificar los cálculos y las operaciones en la fase preliminar de diseño de un sistema de control de actitud. Tomando como base el trabajo previo ya citado, el módulo se ha basado en un libro de cálculo Excel apoyado en una interfaz para la selección y creación de equipos en un catálogo.

El archivo Excel desarrollado posee conectividad con el entorno Concorde de la CDF, varias hojas de edición y presentación de datos y resultados, hojas auxiliares y hojas de datos fuente, en donde se alojan el modelo seleccionado desde la interfaz y la base de datos con el catálogo de elementos disponibles.

La interfaz de usuario gráfica para el catálogo es capaz de leer los elementos de la base de datos, filtrar las búsquedas, añadir los que se deseen al modelo, crear nuevos elementos y exportar el modelo al archivo Excel.

3.1. Trabajo futuro

- Añadir modelos de dimensionado de sensores.
- Implementar la interfaz del catálogo en la nueva versión Matlab R2018a, para aprovechar las mejoras en la interfaz y en la manipulación de tablas.
- Compartimentar las operaciones auxiliares y las de la interfaz de usuario en sendas clases que alberguen las respectivas funciones específicas.
- Extender las preferencias de configuración de la interfaz de usuario.
- Añadir la funcionalidad de edición de entradas de la base de datos directamente desde la GUI.
- Añadir más criterios de ordenación de elementos en la tabla mostrada en la pestaña *Database* de la GUI.
- Integración de `satsearch.co` en la interfaz del catálogo. A 18 de junio de 2018 de la escritura de este informe, las integraciones con Matlab, Excel o Python está en desarrollo.

Referencias

- [1] Zornoza, Eduardo, Gómez-Calero, Carlos y López de Echazarreta, Carlos. *Student Concurrent Design Tool (SCDT): Attitude & Orbit Control Subsystem (AOCS)*. Ref.: ESA-LEX-COV-LEX-COV-TN-00030. Jul. de 2011.
- [2] Gómez-Calero, Carlos et al. *Student Concurrent Design Tool (SCDT): User Manual*. ESA. Jul. de 2011.
- [3] Álvarez Romero, José Miguel et al. *Manual de la sala CDF*. 1.^a ed. IDR/UPM. Feb. de 2018.
- [4] Larson, W.J. y Wertz, J.R. *Space Mission Analysis and Design. Second edition*. Microcosm, Inc., 1992.
- [5] Ley, Wilfried, Wittmann, Klaus y Hallmann, Willi. *Handbook of space technology*. Vol. 22. John Wiley & Sons, 2009.
- [6] Wertz, James R. *Spacecraft attitude determination and control*. Vol. 73. Kluwer Academic Publishers, 1978.
- [7] Singer, S Fred. *Torques and attitude sensing in earth satellites*. Academic Press, 1964.

A. Guía del usuario

A.1. Descripción general

El módulo de ADCS consta de un archivo Excel comprendido por una serie de hojas cuya misión es atender a los diferentes aspectos del sistema control de actitud de un satélite. En la hoja *Overview* se recogen las características y configuraciones más importantes del sistema, como son el presupuesto de masa y potencia, el rango operacional de temperaturas y los distintos componentes del sistema. La hoja *Disturbance torques* se pone a disposición del usuario para estimar los momentos de perturbación típicos de una misión espacial. En la hoja *Sizing* se establece una relación entre dichas perturbaciones y los requisitos mínimos de los actuadores. En las hojas *Actuators* y *Sensors* se muestran los componentes del sistema seleccionados para la misión.

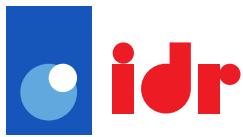
A.1.1. Flujo de trabajo

Los pasos que marcan el flujo de trabajo que se ha de seguir para manipular el módulo correctamente se detallan a continuación:

1. En la Hoja *Overview*, seleccionar la orientación y el tipo de estabilización.
2. En la Hoja *Disturbance torques*, introducir los parámetros necesarios para estimar el momento total de perturbación.
3. En la Hoja *Sizing*, y según las maniobras y requisitos de la misión, introducir valores adecuados a los parámetros de los componentes de la columna de la izquierda. Para la columna de la derecha, se recomienda trabajar con los encargados de Misión y de Propulsión.
4. En las Hojas *Actuators* y *Sensors*, elegir componentes del catálogo (ver A.8 Interfaz del catálogo) de acuerdo al dimensionado de actuadores realizado en la Hoja *Sizing*. Para los sensores, escoger aquellos que permitan cumplir con los requisitos de la misión, como podría ser la precisión de apuntamiento.



Figura A.1: Flujo de trabajo del archivo Excel



A.1.2. Nociones generales de funcionamiento

Las celdas en azul claro () indican celdas en las que el usuario debe introducir un valor o seleccionar una opción. Generalmente, incluyen un valor predeterminado que se establece cuando no se introduce nada en la celda o se suprime el que había. Además, suelen incluir un pequeño mensaje de información que aparece junto a la celda cuando se selecciona dicha celda.

Por último, algunas celdas llevan una comprobación de validez para los datos introducidos. Por ejemplo: los campos que se espera que sean números enteros no permiten introducir otro tipo de dato, o en aquellos que la variable se trate de un porcentaje únicamente se puede poner un valor entre 0 y 1.

A.2. Inputs

Tabla A.1: *Inputs* del módulo.

Parámetro	Unidades	Origen
Foco atractor	-	ORB
Radio en el periapsis	m	ORB
Velocidad en el periapsis	m s^{-1}	ORB
Densidad en el periapsis	kg m^{-3}	ORB
Periodo orbital	s	ORB
Irradiancia solar máxima	W m^{-2}	PWR
Coeficiente de resistencia	-	STR
Momento de inercia en x	kg m^2	STR
Momento de inercia en y	kg m^2	STR
Momento de inercia en z	kg m^2	STR
Precisión de apuntamiento	$^\circ$	SYS
CubeSat	-	SYS
Área proyectada en la dirección de vuelo	m^2	STR
Área proyectada en la dirección del Sol	m^2	STR
Dimensión en x	m	STR
Dimensión en y	m	STR
Dimensión en z	m	STR
Reflectividad	-	THE

A.3. Outputs

Tabla A.2: Outputs del módulo.

Parámetro	Unidades
Masa total	kg
Potencia consumida	W
Tipo de estabilización	-
Orientación	-
Eje de rotación	-
Velocidad de rotación	rad s ⁻¹
Temperatura mínima	°C
Temperatura máxima	°C
<i>Thrusters</i>	
Empuje	N
Tiempo de quemado	s
Posición	m
Cantidad	m

A.4. Hoja Overview

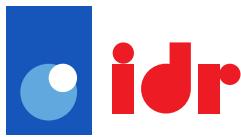
Esta hoja pretende dar una visión general del sistema de forma organizada y resumida, así como establecer las características más importantes de este. Se divide en cuatro bloques diferenciados:

Characteristics Esta es la única parte con la que puede interactuar el usuario. Aquí se escoge el tipo de orientación (ver Tabla A.3), el eje principal para esa actitud (ejes X, Y o Z) y la velocidad de rotación (dar valor 0 en caso de no haber). Además, se elige el tipo de estabilización (ver Tabla A.3) y se muestra la precisión de apuntamiento.

Tabla A.3: Valores seleccionables en la sección *Characteristics* en la hoja *Overview*.

Orientation	Stabilisation
Nadir pointing	No
Sun pointing	1-axis
Spinner	3-axis
Precessing Spinner	
Fixed	

Budget & Requirements Se muestran los valores totales de masa y potencia para el sistema completo, exceptuando los *thrusters*. Se incluyen también el rango de temperaturas necesario para



el funcionamiento del sistema.

Actuators Tipos: *reaction wheels, magnetorquers, momentum wheels, thrusters*. En cada uno se muestra:

- Nombre del componente.
- Cantidad.
- Masa.
- Potencia.

Nota: la información de los *thrusters* se debe obtener del módulo del sistema de propulsión, al ser en este donde se eligen a partir del dimensionamiento realizado en la Hoja *Sizing*.

Sensors Tipos: *IMU/gyroscopes, star sensors, magnetometers, Earth sensors, GNSS, Sun sensors*. En cada uno se muestra:

- Nombre del componente.
- Cantidad.
- Masa.
- Potencia.

A.5. Hoja *Disturbance torques*

En esta hoja se hace una estimación de los momentos de perturbación más importantes, principalmente para un misión terrestre. Los parámetros se clasifican por colores para una rápida identificación de la influencia de este. Se han introducido las siguientes perturbaciones, que se describen en la sección A.5.1:

- Momento de gradiente gravitatorio.
- Momento de presión solar.
- Momento magnético.
- Momento aerodinámico.

Nota: todos los momentos están en unidades del Sistema Internacional, esto es: N m.

A.5.1. Teoría

El entorno espacial genera una serie de perturbaciones cuya magnitud depende principalmente de la altitud de la órbita y a distintas propiedades físicas del satélite. En este módulo se han incluido las perturbaciones principales. Se recomienda acudir a referencias de este apartado para más información, como son las publicaciones de Zornoza *et al.* [1], Larson y Wertz [4], Ley *et al.* [5], Wertz [6] y Singer [7].

CHARACTERISTICS			BUDGET & REQUIREMENTS		
Parameter	Value	Units	Parameter	Value	Units
Orientation Axis	Sun pointing Z	-	Mass	1.126	kg
Spin rotation		rad/s	Power	12.6	W
Stabilisation	3-axis	-	Maximum temperature	85	degC
Pointing precision	3	deg	Minimum temperature	-40	degC

ACTUATORS					
Reaction Wheels			Magnetorquers		
Name	CubeWheel (from: CubeSpaceZA)		Name	NanoTorque GST-600 (from: GomSpace)	
Amount	5	-	Amount	1	-
Mass	0.75	kg	Mass	0.156	kg
Power	11.5	W	Power	0	W
Momentum Wheels			Thrusters		
Name	-		Name	PRO	
Amount	-	-	Amount	-	-
Mass	-	kg	Mass	PRO	kg
Power	-	W	Power	PRO	W

SENSORS					
IMU/Gyrosopes			Star Sensors		
Name	-		Name	Auriga (from: SODERN)	
Amount	-	-	Amount	1	-
Mass	-	kg	Mass	0.21	kg
Power	-	W	Power	1	W
Magnetometres			Earth Sensors		
Name	-		Name	-	
Amount	-	-	Amount	-	-
Mass	-	kg	Mass	-	kg
Power	-	W	Power	-	W
GNSS			Sun Sensors		
Name	-		Name	NCSS-SA055 (from: NewSpace Systems)	
Amount	-	-	Amount	2	-
Mass	-	kg	Mass	0.01	kg
Power	-	W	Power	0.1	W

Figura A.2: Captura de pantalla de la vista resumen

A.5.1.1. Momento de gradiente gravitatorio (T_g)

Esta perturbación tiende a hacer que los satélites se alineen con su eje longitudinal hacia nadir. Se va a considerar que ese eje longitudinal es aquel con menor momento de inercia. La ecuación que modela este fenómeno es:

$$\mathbf{T}_g = 3 \frac{\mu}{\mathbf{r}^3} [\mathbf{r} \times (\mathbf{I}\mathbf{r})] \Rightarrow T_g = \frac{3\mu}{r_p^3} |I_{\text{máx}} - I_{\text{mín}}| \sin(2\theta) \quad (\text{A.1})$$

- $\mu = GM$: parámetro estándar gravitacional [$\text{m}^3 \text{s}^{-2}$]. Para la Tierra, $\mu = 3,986 \cdot 10^{14} \text{ m}^3 \text{s}^{-2}$.
- r_p : radio en el perigeo [m].
- $I_{\text{mín}} = \min\{I_x, I_y, I_z\}$: momento de inercia mínimo [kg m^2], en principio el del eje longitudinal del satélite.
- $I_{\text{máx}} = \max\{I_x, I_y, I_z\}$ momento de inercia máximo [kg m^2].
- θ : máxima desviación [$^\circ$] del eje longitudinal frente al eje nadir-cenit.

A.5.1.2. Momento de presión solar (T_{sp})

En satélites asimétricos, la radiación electromagnética incidente causa una presión y un esfuerzo de cortadura sobre la superficie interceptada. Esta radiación tiene su origen fundamentalmente en el Sol, por lo que se utiliza la irradiancia solar para estimar este efecto perturbador. En el caso de la Tierra, esta tiene un valor medio anual de 1367 W m^{-2} .

$$\mathbf{T}_{sp} = (\mathbf{c}_{sp} - \mathbf{c}_g) \times \left(\frac{\Phi}{c} (1+q) A (-\hat{\mathbf{u}}_{\text{Sol}}) \right) \Rightarrow T_{sp} = \frac{\Phi}{c} A_{sp} (1+q) \cos(i) \|\mathbf{c}_{sp} - \mathbf{c}_g\| \quad (\text{A.2})$$

- Φ : irradiancia solar [W m^{-2}]. Órbita terrestre: $\sim 1367 \text{ W m}^{-2}$.
- $c = 3 \cdot 10^8 \text{ m s}^{-1}$: velocidad de la luz.
- q : reflectados.
- i : ángulo [$^\circ$] de incidencia solar.
- A_{sp} : área proyectada en la dirección del Sol [m^2].
- \mathbf{c}_g : posición del centro de gravedad del satélite [m].
- \mathbf{c}_{sp} : posición del centro de presión solar [m].

A.5.1.3. Momento magnético (T_m)

Es el resultado de la interacción entre el campo magnético terrestre y el dipolo magnético residual de la nave, que se intenta minimizar en las fases finales de diseño a través del rutado de los mazos eléctricos.

$$\mathbf{T}_m = \mathbf{D}_m \times \mathbf{B} \Rightarrow T_m \approx D_m \frac{2M_m}{r_p^3} \quad (\text{A.3})$$

- M_m : momento magnético del cuerpo central. Para la Tierra vale $7,96 \cdot 10^{15} \text{ Tm}^3$. Este valor se puede calcular para otros cuerpos a partir de un desarrollo en armónicos esféricos del campo magnético de ese cuerpo. En el apéndice H de Wertz [6] se describe este desarrollo para la Tierra y se alcanza un valor parecido al mostrado en la ecuación (H-18).

- r_p : radio en el periapsis [m].
- D_m : dipolo magnético residual del satélite [$\text{m}^2 \text{ A}$].

A.5.1.4. Momento aerodinámico (T_a)

Para satélites en LEO, la interacción de su superficie con la atmósfera residual no puede ignorarse. Es la perturbación más importante en órbitas por debajo de los 400 km de altitud.

$$\mathbf{T}_a = (\mathbf{c}_a - \mathbf{c}_g) \times \left(-\frac{1}{2} \rho_{\text{atm}}(\mathbf{r}, t) C_D A \frac{\dot{\mathbf{r}}^3}{\dot{r}} \right) \Rightarrow T_a = \frac{1}{2} \rho_{\text{atm}, p} C_D A_a V_p^2 \|\mathbf{c}_a - \mathbf{c}_g\| \quad (\text{A.4})$$

- $\rho_{\text{atm}, p}$: densidad atmosférica en el periapsis [kg m^{-3}]
- C_D : coeficiente de resistencia, normalmente entre 2 y 2,5
- A_a : área proyectada en la dirección de vuelo [m^2]
- \mathbf{c}_a : posición del centro de presiones [m]
- \mathbf{c}_g : posición del centro de gravedad del satélite [m]
- V_p : velocidad del satélite en el periapsis [m s^{-1}]

A.5.1.5. Momento total de perturbaciones (T_D)

El momento a utilizar para dimensionar será aquel que combine los peores casos. Por ello, se coge la suma de los anteriores:

$$T_D = T_g + T_{sp} + T_m + T_a \quad (\text{A.5})$$

Por último, [5] da las siguientes relaciones entre pares de perturbación para una órbita de 700 km en la Tierra:

$$T_g : T_m : T_{sp} : T_a = 2000 : 500 : 4 : 1 \quad (\text{A.6})$$

Otros momentos de perturbación: Los momentos de perturbación dominantes ya han sido descritos, pero otros que se deben considerar en fases más avanzadas de diseño están relacionados con el uso de propulsantes (escapes, movimientos dentro de los tanques) o partes móviles de mecanismos del satélite (desplegado de paneles), así como impactos de micrometeoroides. Dada la dificultad de su modelado y que no son dominantes [5], se ha considerado que su inclusión en el módulo no es necesaria.

A.5.2. Hoja de cálculo

En la parte superior de la Figura A.3 se muestran los valores de cada uno de los momentos de perturbación, así como su suma. Mediante un código de colores se destaca la influencia del parámetro de la misma fila en cada uno de los pares de perturbación (ver introducción del Apartado A.5).

Los valores que debe introducir el usuario son:

TORQUES						
Parameter		Value	Units	%	Relative	
T_g	Gravity gradient	1.23E-02	N·m	99.59	1.00E+00	
T_{sp}	Solar pressure	1.74E-07	N·m	0.00	1.41E-05	
T_m	Magnetic	5.11E-05	N·m	0.41	4.14E-03	
T_a	Aerodynamic	9.70E-10	N·m	0.00	7.86E-08	
T_D	Total	1.24E-02	N·m	100.00		

SATELLITE						
Parameter	Origin	Value	Units	Affected torque		Remarks
θ	Max deviation z with z_h	20.00	deg			Maximum deviation of z axis with the local vertical
q	Reflectivity factor	THE	-			Reflectivity factor, in [0,1]
i	Sun incident angle	5.00	deg			Sun incident angle
mdp	Magnetic dipole satellite	1.00	A·m^2			Residual magnetic dipole of the satellite
C_D	Drag coefficient	CNF	2.3	-		Drag coefficient
I_x	Moment of inertia X	STR	7.500E+03	kg·m^2		Moment of inertia in x axis
I_y	Moment of inertia Y	STR	5.000E+03	kg·m^2		Moment of inertia in y axis
I_z	Moment of inertia Z	STR	1.00E+04	kg·m^2		Moment of inertia in z axis
I_{min}	Moment of inertia min	-	5.00E+03	kg·m^2		Minimum moment of inertia
I_{max}	Moment of inertia max	-	1.00E+04	kg·m^2		Maximum moment of inertia
A_a	Projected area in flight dir	CNF	0.5	m^2		Area projected in flight direction
A_{sp}	Solar dir proj area	CNF	0.8	m^2		Area projected in solar direction
CoSP - CoG	Solar pressure arm		0.03	m		Moment arm for solar radiation torques
CoAP - CoG	Aero pressure arm		0.03	m		Moment arm for aerodynamic forces

ENVIRONMENT						
Parameter	Origin	Value	Units	Affected torque		Remarks
CB	Central body	ORB	Earth	-		Central body of the orbit
μ	Gravitational parametre	-	3.99E+14	m^3/s^2		Standard gravitational parameter of the central body
r_p	Periapsis radius	ORB	6.78E+06	m		Minimum distance to the centre of the central body
V_p	Periapsis velocity	ORB	7500	m/s		Maximum velocity, usually at periapsis
ρ_{atm}	Periapsis atm density	ORB	1.00E-15	kg/m^3		Atmospheric density at periapsis
SF_{max}	Maximum solar flux	PWR	1367	W/m^2		Maximum solar flux. Default is average at the Earth
MM_{CB}	Magnetic moment of central body		7.96E+15	T·m^3		Magnetic moment of the central body (Earth)

Figura A.3: Captura de pantalla de la hoja de perturbaciones orbitales.

- θ : máxima desviación [°] del eje longitudinal frente al eje nadir-cenit.
- i : ángulo [°] de incidencia solar.
- D_m : dipolo magnético residual del satélite [$m^2 A$].
- $\|\mathbf{c}_a - \mathbf{c}_g\|$: brazo de aplicación de momento aerodinámico [m].
- $\|\mathbf{c}_{sp} - \mathbf{c}_g\|$: brazo de aplicación de momento de presión solar [m].
- M_m : momento magnético del cuerpo central. Si el cuerpo no es la Tierra, toma valor nulo, lo cual podría no ser correcto. Para su cálculo, remitirse al apartado correspondiente en A.5.1.

A.6. Hoja Sizing

En esta hoja se hace uso de relaciones para encontrar valores que permitan estimar las características necesarias de los actuadores del subsistema ADCS a partir de las perturbaciones ambientales y de maniobras típicas. Se contemplan aquí los siguientes actuadores:

- Ruedas de reacción (RW, Reaction Wheels).
- Magnetic Torquers, o magnetorquers (MT).
- Momentum Wheels (MW).
- Thrusters.

A.6.1. Teoría

A.6.1.1. Ruedas de reacción

Las ruedas de reacción son capaces de generar un momento a través de la aceleración o deceleración de un volante de inercia. Este giro viene dado por un motor eléctrico que es capaz de variar así el par generado en el eje de rotación del volante.

Las características a dimensionar son el momento que puede generar la rueda (T_{RW}), y el momento almacenado por la rueda (H_{RW}).

Momento que puede generar la RW (T_{RW}) Será el máximo entre el momento necesario para compensar perturbaciones y el necesario para realizar una maniobra de rotación (*slewing manoeuvre*).

$$T_{RW} [\text{N m}] = \max\{T_{RW,d}, T_{RW,s}\} \quad (\text{A.7})$$

$$T_{RW,d} = T_D(1 + \text{SM}) \quad (\text{A.8})$$

$$T_{RW,s} = 4\theta \frac{\pi}{180} I / (60 \cdot t_s)^2 \quad (\text{A.9})$$

- SM: margen de seguridad. Típicamente 20 %.
- θ : máximo ángulo de rotación [°].
- I : momento de inercia [kg m^2] del eje de rotación la *slewing manoeuvre*. $I_{\text{máx}}$ en caso conservativo.
- t_s : duración mínima de la maniobra considerada [min].
- T_D : momento de perturbaciones ambientales [N m]. Ver A.5.1.5.

Momento almacenado por la RW (H_{RW})

$$H_{RW} [\text{N m s}] = T_D P \quad (\text{A.10})$$

- P : periodo orbital [s].
- T_D : momento de perturbaciones ambientales [N m]. Ver A.5.1.5.

A.6.1.2. Magnetic Torquers

Los *magnetotorquers* generalmente son unas bobinas planas. Cuando se aplica una tensión, se crea una corriente que genera un dipolo magnético que interactúa con el campo magnético terrestre, de tal forma que el bobinado intenta alinear su propio campo magnético con él.

El valor característico de este componente es el dipolo magnético que es capaz de generar ($D_{m,MT}$).

$$T_{MT} = D_{m,MT} B \approx D_{m,MT} \frac{2M_m}{r_p^3} \quad (\text{A.11})$$

$$D_{m,MT} = T_D \frac{r_p^3}{2M_m} \quad (\text{A.12})$$

- $D_{m,MT}$: dipolo magnético creado por el sistema [$\text{m}^2 \text{ A}$].
- M_m : momento magnético del cuerpo central [Tm^3]. Ver A.5.1.3.
- r_p : radio en el periapsis [m].
- T_D : momento de perturbaciones ambientales [N m]. Ver A.5.1.5.

A.6.1.3. Momentum Wheels

Son ruedas a reacción que se diseñan para funcionar a alta velocidad de rotación, alcanzando así una estabilización por efecto giroscópico: un momento transversal al eje de rotación causa una precesión que compensa ese momento.

Tanto si se eligen una o varias ruedas de momento, el par utilizado para el dimensionamiento será siempre el mismo, esto es, el par de perturbaciones calculado en la hoja *Disturbances* (ver A.5). El diseño típico de las ruedas de momento sirve al propósito de que el momento angular sea lo suficientemente alto como para mantener el giro provocado por la perturbación sin la necesidad de usar ningún otro actuador durante una órbita completa. Así, el par de perturbaciones medio para un cuarto o media órbita determina el momento angular requerido.

$$T_D \frac{P}{4} = H_{MW} \cdot \delta \frac{\pi}{180} \quad (\text{A.13})$$

$$H_{MW} = \omega_{MW} \frac{2\pi}{60} \cdot I_{MW} \quad (\text{A.14})$$

$$m_{MW} = \frac{2H_{MW}}{\omega_{MW} \frac{2\pi}{60} \cdot R_{MW}^2} \quad (\text{A.15})$$

- P : periodo orbital [s].
- T_D : momento de perturbaciones ambientales [N m]. Ver A.5.1.5.
- δ : máximo desplazamiento/desviación angular permitido [°].
- ω_{MW} : velocidad angular [rpm].
- $I_{MW} = m_{MW} R_{MW}^2$: momento de inercia de la rueda [kg m^2].
- m_{MW} : masa de la rueda [kg].
- R_{MW} : radio de la rueda [m].

A.6.1.4. Thrusters

Las propiedades que hacen falta para dimensionar los *thrusters* son el empuje (F) y el tiempo de quemado (t_b). El resto de los parámetros (I_{sp} , masa de propelente, etc) se estiman en el módulo del subsistema de Propulsión. También es importante la distancia entre el centro de gravedad del satélite y la posición de los *thrusters*, que sería el brazo de los *thrusters* (L).

El valor tomado para el empuje es el máximo del requerido para las distintas maniobras, y el tiempo de quemado es el acumulado por todas ellas.

Compensación de perturbaciones Los *thrusters* también pueden usarse para generar un momento que compense las perturbaciones (T_D) vistas en el Apartado A.5.1. Si se considera una pareja de *thrusters*:

$$F_D = \frac{T_D}{2L} \quad (\text{A.16})$$

Maniobra de rotación (en un zero momentum system) En un sistema con RWs, el par que deben generar los *thrusters* para realizar un maniobra de rotación es la siguiente:

$$T = 2F_{RW}L = I\ddot{\theta} = I\frac{\dot{\theta}}{t_{ac}} = I\frac{\theta}{t_b t_s} \quad (\text{A.17})$$

$$F_{RW} = \frac{I}{2L} \frac{\theta \pi/180}{t_b \cdot 60 \cdot t_s} \quad (\text{A.18})$$

- I : momento de inercia del eje de rotación de la *slewing manoeuvre* [kg m^2].
- θ : ángulo de rotación [°].
- t_b : tiempo de quemado del motor [s].
- t_s : duración de la maniobra [min].

Maniobra de rotación (en un bias momentum system) En un sistema con MWs, el par que deben generar los *thrusters* para realizar un maniobra de rotación es la siguiente:

$$T = F_{MW} \cdot 2L \frac{t_b}{t_s} = H_{MW}\dot{\theta} \quad (\text{A.19})$$

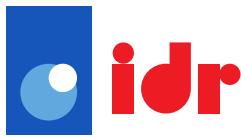
$$F_{MW} = \frac{H_{MW}\dot{\theta}}{2L} \frac{t_b}{60 \cdot t_s} \quad (\text{A.20})$$

- H_{MW} : momento angular del sistema [N m s].
- $\dot{\theta}$: ángulo de rotación [rad s^{-1}].
- t_b : tiempo de quemado del motor [s].
- t_s : duración de la maniobra [min].

Desaturación de momento angular Cuando hay tanto RWs como MWs, las perturbaciones seculares deben ser contrarrestadas con la aplicación de pares externos (magnetopares o *thrusters*). En el caso de las ruedas de reacción, estas perturbaciones seculares causan que alcancen su velocidad de rotación máxima y por tanto no puedan almacenar más momento angular.

$$F_S = \frac{H_{RW}}{2Lt_b} \quad (\text{A.21})$$

- H_{MW} : momento angular del sistema [N m s].
- t_b : tiempo de quemado del motor [s].
- El 2 viene de asumir parejas de motores.



A.6.2. Hoja de cálculo

REACTION WHEELS (RW)					THRUSTERS				
Parameter		Origin	Value	Units	Parameter		Origin	Value	Units
T_{RW}	Torque	-	0.71	N·m	F_T	Thrust	-	530516.477	N
H_{RW}	Momentum	-	68.87	N·m·s	t_b	Total burning time	-	16	s
Torque for disturbance rejection			0.0248	N·m	L	Arm		0.27	m
T_D	Disturbances torque	-	0.0124	N·m	N	Number of thrusters		2	
SM	Security margin for disturbances	1	100%	-	Thrust for disturbance rejection			0.023	N
Torque for slewing manoeuvre			0.7074	N·m	T_D	Disturbances torque	-	0.0124	N·m
θ	Maximum slew angle	1	1.00	deg	n	Number of times		1	-
I	Rotating axis = [X,Y,Z] -> Moment of inertia	max	1.00E+04	kg·m^2	Thrust for zero momentum system (RW)			530516.477	N
t_s	Minimum slew manoeuvre duration	30	30	min	θ	Maximum slew angle	90.00	deg	

MAGNETORQUERS				
Parameter		Origin	Value	Units
Magnetic dipole		-	242.412	A·m^2

MOMENTUM WHEELS (MW)				
Parameter		Origin	Value	Units
H_{MW}	Angular momentum	-	2.16307E-05	N·m·s
δ	Maximum angular deviation		5	deg
ω_{MW}	Angular velocity		1000	rpm
R_{MW}	Wheel radius		0.00E+00	m
m_{MW}	Wheel mass		0	kg

Figura A.4: Captura de pantalla de la hoja de dimensionamiento de actuadores.

En la parte izquierda de la hoja se muestran los valores mínimos de momento y par para las ruedas de reacción, además del dipolo magnético necesario para los magnetopares. En la parte derecha de la hoja se muestran los requisitos de empuje, tiempo de combustión, brazo, y número de *thrusters*, calculados a partir de los datos que aparecen más abajo (ver introducción de A.6).

A.7. Hojas Actuators y Sensors

En las hojas *Actuators* y *Sensors* se muestran los componentes del sistema seleccionados para la misión. Ambas hojas extraen los datos de la hoja operacional *Model*, donde se encuentran las entradas con los equipos elegidos desde la interfaz del catálogo en *Matlab*.

A.7.1. Hoja *Actuators*

En la hoja *Actuators* se muestran los componentes elegidos para tal fin en la interfaz del catálogo, junto a sus características generales y específicas y a un cómputo global de masa y potencia. En la parte inferior, el usuario dispone de un botón para acceder directamente a la interfaz (ver Figura A.5). Los siguientes componentes¹ están incluidos en el catálogo:

- Reaction Wheels
- Magnetorquers
- Momentum Wheels

¹Breves descripciones disponibles en el Apartado A.6.1.

Type	Name	Supplier	Mass	Power	Amount	Total mass	Total power	Temp max	Temp min	Width	Length	Height				
-	-	-	kg	W	-	kg	W	degC	degC	mm	mm	mm				
Reaction Wheel	CubeWheel (from: CubeSpaceZA)												ω max	Torque max	Momentum max	
	CubeWheel	CubeSpaceZA	0.15	2.3	5	0.75	11.5	70	-10	46	46	31.5	6000	1	10.82	
Momentum Wheel	-	-	-	-	-	-	-	-	-	-	-	-	ω max	Torque max	Momentum max	
Magnetotorquer	NanoTorque GST-600 (from: GomSpace)												Nr of axes	resistance	Temp coef	Dipole moment
	NanoTorque GST-600	GomSpace	0.156	0	1	0.156	0	85	-40	90.5	96.9	17.2	3	30	0	355
Other actuator	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

Run GUI

Figura A.5: Captura de pantalla de la hoja de actuadores

A.7.2. Hoja Sensors

En la hoja *Sensors* se muestran los componentes elegidos para tal fin en la interfaz del catálogo, junto a sus características generales y específicas y a un cómputo global de masa y potencia. Al igual que la hoja *Actuators*, en la parte inferior, el usuario dispone de un botón para acceder directamente a la interfaz (ver Figura A.6).

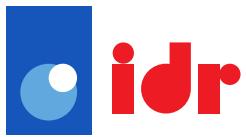
Type	Name	Supplier	Mass	Power	Amount	Total mass	Total power	Temp max	Temp min	Width	Length	Height	Precision	Accuracy	FoV x	FoV y
-	-	-	kg	W	-	kg	W	degC	degC	mm	mm	mm				
Star sensor	Auriga (from: SODERN)												Precision	Accuracy	FoV x	FoV y
	Auriga	SODERN	0.21	1	1	0.21	1	40	-30	56	66	94	deg	deg	deg	deg
Sun sensor	NCSS-SA055 (from: NewSpace Systems)												Precision	Accuracy	FoV x	FoV y
	NCSS-SA055	NewSpace Systems	0.005	0.05	2	0.01	0.1	50	-25	33	11	6	deg	deg	deg	deg
Earth sensor	-	-	-	-	-	-	-	-	-	-	-	-	Precision	Accuracy	FoV x	FoV y
	-	-	-	-	-	-	-	-	-	-	-	-	deg	deg	deg	deg
Magnetometer	-	-	-	-	-	-	-	-	-	-	-	-	Noise	Range	-	-
IMU	-	-	-	-	-	-	-	-	-	-	-	-	-	-	Drift	-
GNSS	-	-	-	-	-	-	-	-	-	-	-	-	Position Acc	Velocity Acc	GNSS	-
Dock	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Other sensor	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

Run GUI

Figura A.6: Captura de pantalla de la hoja de sensores

Los siguientes sensores están disponibles:

- **Sensores de estrellas:** detectan patrones de estrellas y por tanto son capaces de dar la



orientación en tres ejes.

- **Sensor solar:** dan la dirección en la que está el Sol, por lo que dan información en dos ejes.
- **Sensores de Tierra:** detectan el horizonte terrestre.
- **Magnetómetros:** miden el campo magnético terrestre.
- **Giróscopos:** miden rotaciones en un sistema de referencia inercial, por lo que son independientes del exterior.
- **GNSS:** son capaces de medir la actitud comparando las pequeñas diferencias de tiempo de contacto entre las antenas de la nave y el/los satélites de la red GNSS.

Asimismo, se incluye el *Dock* en la hoja de sensores, elemento que representa el controlador del sistema ADCS, incluido en el modelo por su relevancia en cuanto a masa y consumo.

A.8. Interfaz del catálogo

En las Figuras A.7, A.8, A.9 y A.10 se recogen las distintas pestañas de la interfaz del catálogo, resaltando la utilidad de cada una. La pestaña *Database* sirve para que el usuario busque los elementos ya creados y añadirlos al modelo, ayudado por filtros que acoten u ordenen los resultados. La pestaña *Model* contiene una vista resumen del modelo seleccionado; el usuario puede seleccionar una u otra categoría y ver las especificaciones del elemento añadido, crear uno nuevo, buscarlo o quitarlo del modelo. La pestaña *Create item* sirve para añadir elementos nuevos a la base de datos, elegibles una vez guardados.

CDF - ADCS module

Model Database Create item Settings

I'm searching for a sensor In particular a sun sensor Sort by Name (AZ)

Select	Name	Supplier	Mass	Power	Vnom	Width	Length	Height
<input type="checkbox"/>	BiSon64	Lens R&D	0.0217	0	NaN	NaN	NaN	NaN
<input checked="" type="checkbox"/>	BiSon64-B	Lens R&D	0.0217	0	NaN	NaN	NaN	NaN
<input type="checkbox"/>	CubeSense	CubeSpace	0.0800	0.3000	3.3000	98	96	
<input type="checkbox"/>	CubeSense	CubeSpace	0.0800	0.3600	3.3000	90	96	
<input type="checkbox"/>	NCSS-SA...	NewSpace...	0.0050	0.0500	5	33	11	
<input type="checkbox"/>	NFSS-411	NewSpace...	0.0350	0.0500	5	34	32	
<input type="checkbox"/>	Nano-SSO...	SolarMEMS	0.0037	0.0066	3.3000	27.4000	14	5.9
<input type="checkbox"/>	Nano-SSO...	SolarMEMS	0.0250	0.0360	5	30	30	
<input type="checkbox"/>	Nano-SSO...	SolarMEMS	0.0065	0.0759	3.3000	43	14	5.9
<input type="checkbox"/>	Nano-SSO...	SolarMEMS	0.0065	0.1150	5	60	30	
<input type="checkbox"/>	NanoSens...	GomSpace	0.0022	0	NaN	22	11	

Search filters

- Max. mass 0.00 kg
- Max. power 0.00 W
- Max. temperature 0 °C
- Min. temperature 0 °C

Checkout

Item selected Amount

BiSon64-B 2

Add item to model

Total power 0.00 W Total mass 0.00 kg

Figura A.7: Captura de pantalla de la pestaña de la base de datos

CDF - ADCS module

Model Database Create item Settings

View specs Create item Search item Remove item Export model to Excel

Sun sensors	Star sensors	Earth sensors	Magnetometers
BiSon64-B	CubeStar	No items yet	No items yet
Select	Select	Select	Select
Docks	GNSS receivers	Inertial measurement units	Other sensors
No items yet	No items yet	3-axis FOG IMU	No items yet
Select	Select	Unselect	Select
Reaction wheels	Magnetorques	Momentum wheels	Other actuators
CubeWheel	MagneTorquer board (iMTQ)	No items yet	No items yet
Select	Select	Select	Select

Total power 11.23 W Total mass 3.07 kg

Figura A.8: Captura de pantalla de la pestaña del modelo

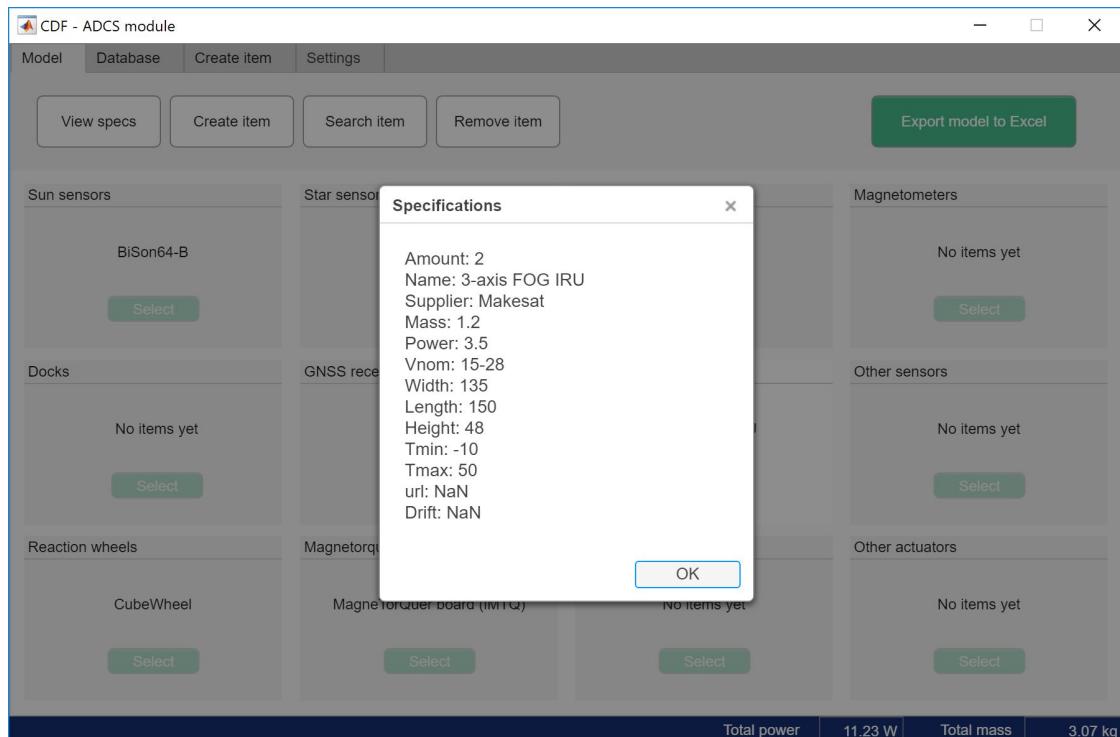
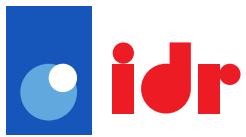


Figura A.9: Captura de pantalla del cuadro de diálogo de especificaciones

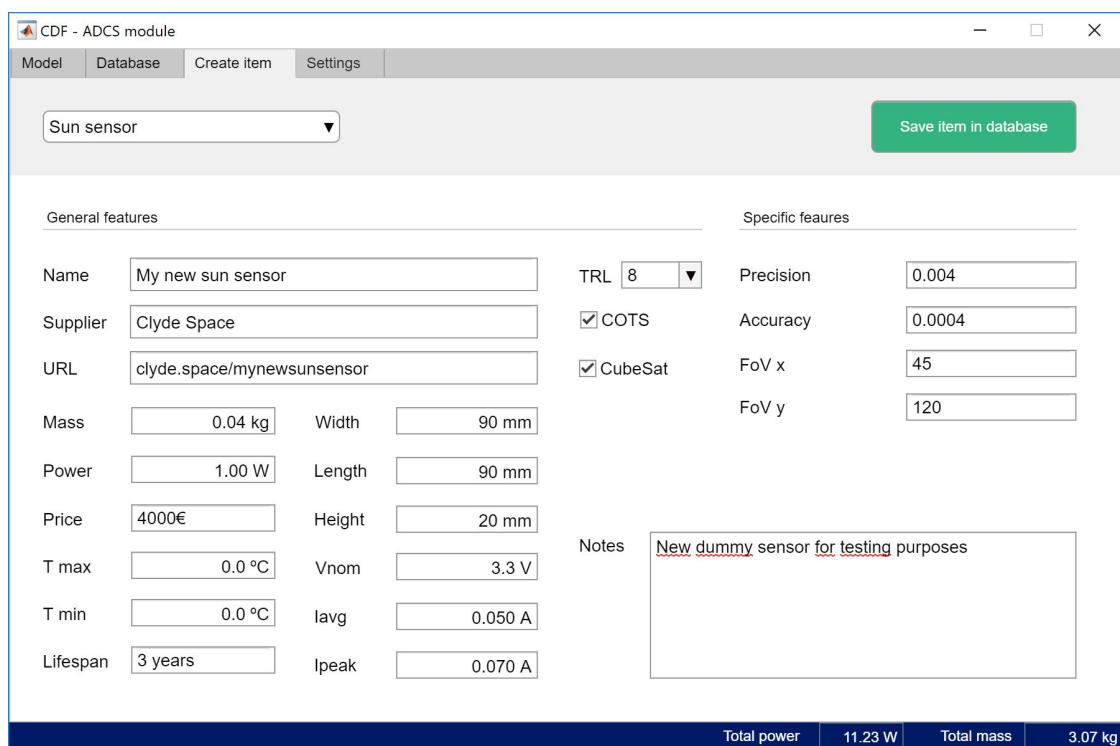


Figura A.10: Captura de pantalla de la pestaña de creación de elementos

B. Guía del desarrollador

B.1. Requisitos de instalación

Los programas necesarios y suficientes para el funcionamiento del módulo ADCS son, atendiendo a la versión utilizada:

1. Microsoft Excel 2016
2. Matlab R2017b
3. Matlab Spreadsheet Link

Es necesario activar la extensión Matlab Spreadsheet Link en Excel para completar el enlace de ambos programas y así hacer uso de las subrutinas encargadas de abrir la interfaz del catálogo desde las hojas *Actuators* y *Sensors*¹.

Una vez añadida la extensión a Excel, habrá que activar la referencia de la librería de Matlab Spreadsheet Link en el código VBA². Realizados estos dos pasos, la instalación habrá concluido y el módulo funcionará correctamente.

B.2. Hoja de cálculo

En la elaboración de la hoja de cálculo Excel se ha hecho uso de una serie de características de la herramienta que han facilitado el desarrollo. A continuación se comenta el uso que se ha dado a cada una de ellas:

- **Administrador de nombres**, pestaña “Fórmulas” de la cinta – Esta herramienta asigna nombres de variables como en la programación tradicional a celdas, haciendo la creación de fórmulas mucho más sencilla, legible y verificable. No todas las variables han sido nombradas, pero las más relevantes sí.

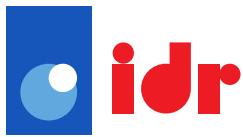
Unos nombres especialmente importantes son los que se definen en las hojas *Inputs* y *Outputs*, por lo que modificar estas hojas debe hacerse con extremo cuidado.

- **Validación de datos**, pestaña “Datos” de la cinta – Con esto es posible delimitar los valores que se pueden introducir en las celdas (sólo números positivos, enteros, seleccionar de una lista, fechas, etc), mostrar pequeños mensajes de ayuda o información adicional al seleccionar la celda, y lanzar ventanas de aviso o error en el caso de que el valor metido no sea válido.

Todas las celdas que con las que debe interactuar el usuario incorporan un pequeño mensaje de ayuda con el objetivo de orientarle. Por ejemplo, cuando una variable incorpora un valor

¹Seguir las instrucciones recogidas en <https://goo.gl/1CJV6S> (ver *Configure Microsoft Excel*)

²Seguir las instrucciones recogidas en <https://goo.gl/1CJV6S> (ver *Work with the Microsoft Visual Basic Editor*)



predeterminado, en el mensaje se indica que la celda se debe dejar vacía para que el valor predeterminado sea el escogido.

Nota: Los valores de las listas desplegables que aparecen en algunas celdas de entrada del usuario se encuentran en la hoja *Auxiliar Information*.

Nota: Los valores predeterminados de algunas variables se encuentran definidos en la hoja *Auxiliar Information*. Ahí se asigna el correspondiente nombre, por lo que simplemente cambiar el valor es suficiente.

El código RGB de los colores utilizados en la hoja de cálculo se recoge en la Tabla B.1. Se recomienda su uso en futuros desarrollos para mantener un patrón de diseño coherente.

Tabla B.1: Códigos RGB de los colores empleados en la hoja de cálculo, con su significado.

Color	R	G	B	Significado
	221	235	247	Celda con la que interactúa el usuario
	0	32	96	Título de sección
	68	114	196	Cabecero de tabla
	142	168	219	Nombre de variable
	180	198	231	Valor de la variable
	112	173	71	Momento de perturbación del gradiente gravitatorio
	255	192	0	Momento de perturbación del presión solar
	237	125	49	Momento de perturbación magnético
	91	155	213	Momento de perturbación aerodinámico

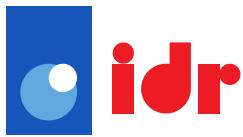
B.3. Interfaz del catálogo

B.3.1. Código fuente de la interfaz

Código B.1: Database.m

```
1 classdef Database
2     %SEARCHENGINE Summary of this class goes here
3     % Detailed explanation goes here
4 % ===== %
5 properties (Access = public)
6     filename
7     sheetname
8     table
9     sortedtable
10 end
11 % ===== %
12 properties (Access = private)
13     class_array = {'actuator','sensor','dock'};
14     subclass_array = {{'reaction wheel','momentum ...
15         wheel','magnetorquer','other actuator'},...
```





```

100    specific = table(:,ismember(table.Properties.VariableNames,... ...
101        % Select subclass specific data
102        obj.getSpecificFieldsBySubclass(subclass)));
103
104
105    table = [col common specific];                                % ...
106    % Group all and name first column
107    table.Properties.VariableNames{1} = 'select';
108    raw = obj.table2raw(table);                                     % ...
109    % Split table into header and data
110    thead = obj.headerValues2Text(raw(1,:));
111    tbody = raw(2:end,:);
112
113    function fields = getSpecificFieldsBySubclass(~,subclass)      % ...
114        Gets specific columns to show in db table
115        switch subclass
116            case {'reaction wheel','momentum wheel'}
117                fields = {'w_max','torque_max','momentum_max'};
118            case 'magnetorquer'
119                fields = {'nr_axis','resistance','temp_coef','dipole_moment'};
120            case {'sun sensor','star sensor','earth sensor'}
121                fields = {'prec_deg','acc_deg','fov_x','fov_y'};
122            case 'magnetometer'
123                fields = {'noise','range'};
124            case 'gnss'
125                fields = {'acc_pos','acc_vel','gnss'};
126            case 'imu'
127                fields = {'drift'};
128            otherwise
129                fields = {};
130        end
131    end
132
133    function createItem(obj,app)                                     % CREATEITEM ...
134        Add new item to database
135
136        % UPDATE IN TABLE
137        % ===
138        subclass = app.CreateItemDropdown.Value;                      % ...
139        % Get subclass from dropdown
140        % Load values from form data
141        general = {app.NameEditField, app.SupplierEditField, ...
142                    app.URLEditField,     app.MassEditField,   app.PowerEditField, ...
143                    app.PriceEditField, ...
144                    app.TmaxEditField,  app.TminEditField, ...
145                    app.LifespanEditField, app.WidthEditField, ...
146                    app.LengthEditField, app.HeightEditField, ...
147                    app.VnomEditField,  app.IavgEditField, ...
148                    app.IpeakEditField, appTRLDropDown, ...
149                    app.COTSCheckBox,  app.CubeSatCheckBox, ...
150                    app.NotesTextArea};
151        specific = {app.Spec1EditField, app.Spec2EditField, ...
152                    app.Spec3EditField, app.Spec4EditField, app.Spec5EditField, ...
153                    app.Spec6EditField};

```

```

142     formdata = [general specific];
143     general_fields = {'name','supplier','url','mass','power',...
144                         % ...
145                         'Prepare field names
146                         'price','temp_max','temp_min','lifespan',...
147                         'dim_x','dim_y','dim_z','v_nom','i_avg',...
148                         'i_peak','trl','cots','cubesat','notes'};
149     specific_fields = app.db.getSpecificFieldsBySubclass(subclass);
150     fields = [general_fields specific_fields];
151     % Get size of current database table
152     rows = size(app.db.table,1);
153     % Set subclass and create new row in db table
154     app.db.table(rows+1,'subclass') = {subclass};
155     % Get class from subclass
156     actuators = {'reaction wheel','momentum ...
157                 'wheel','magnetorquer','other actuator'};
158     sensors = {'sun sensor','star sensor','earth ...
159                 'sensor','magnetometer','imu','gnss','other sensor'};
160     % Add timestamp
161     app.db.table(rows+1,'t_insert') = {datestr(now,'dd/mm/yyyy')};
162     if ismember(subclass,actuators)
163         class = 'actuator';
164     elseif ismember(subclass,sensors)
165         class = 'sensor';
166     else
167         class = 'dock';
168     end
169     app.db.table(rows+1,'class') = {class};

170     % Load rest of the values in db table
171     for i=1:length(fields)
172         header = app.db.table.Properties.VariableNames{fields(i)};
173         value = formdata{i}.Value;
174         column = app.db.table{:,header};
175         % Prefilters
176         if strcmp(header,'trl') && strcmp(value,'None')
177             value = '';
178         end
179         if any(strcmp(header,{ 'cots','cubesat'}))
180             if value == 0
181                 value = 'no';
182             else
183                 value = 'yes';
184             end
185         end
186         if any(strcmp(header,{ 'v_nom','i_peak','i_avg',...
187                         'dim_x','dim_y','dim_z'}))
188             if value == 0
189                 value = [];
190             end
191         end
192         if ~isempty(value)
193             if ischar(value)
194                 if iscell(column)
195                     app.db.table{rows+1,header} = {value};
196                 elseif ischar(column)
197                     app.db.table{rows+1,header} = value;
198                 end
199             end
200         end
201     end
202 
```

```

195         end
196     elseif isnumeric(value)
197         if isnumeric(column)
198             app.db.table{rows+1,header} = value;
199         elseif iscell(column)
200             app.db.table{rows+1,header} = num2cell(value);
201         end
202     end
203 end
204 % Set item ID
205 id = num2hex(rand(1));
206 app.db.table(rows+1,1) = cell2table({id});
207
208 % WRITE IN EXCEL
209 % ====
210 [~, ~, raw] = xlsread(obj.filename,obj.sheetname); % ...
211 % Read excel catalog
212 range = obj.getExcelNextEmptyRange(raw); % ...
213 % Get range of the next empty line
214
215 % Get array to post in database
216 data = table2cell(app.db.table(rows+1,:));
217
218 % Write data in Excel
219 xlswrite(obj.filename, data, obj.sheetname, range);
220
221 % Clear form
222 for i=1:length(formdata)
223     value = formdata{i}.Value;
224     if isnumeric(value)
225         formdata{i}.Value = 0;
226     elseif iscell(value)
227         formdata{i}.Value = {};
228     elseif ischar(value)
229         if strcmp(value,'None')
230             formdata{i}.Value = value;
231         else
232             formdata{i}.Value = '';
233         end
234     end
235
236 % Save operation in log
237 obj.log(sprintf('Item %s (%s) created and added to database',id, ...
238 subclass));
239
240 function array_text = headerValues2Text(~,array_values)
241 fieldvalues = {'select','id','name',...
242                 'supplier','mass','power','v_nom',...
243                 'dim_x','dim_y','dim_z','temp_max','temp_min',...
244                 'w_max','torque_max','momentum_max',...
245                 'nr_axis','resistance','temp_coef','dipole_moment',...
246                 'prec_deg','acc_deg','fov_x','fov_y',...
247                 'noise','range','acc_pos','acc_vel','gnss','drift'};

```

```

248
249     fieldlabels = ...
250         {'Select','ID','Name','Supplier','Mass','Power','Vnom',...
251             'Width','Length','Height','Tmax','Tmin',...
252             'Wmax','Max Torque','Max Momentum','Nr of Axes',...
253             'Resistance','Temp coef','Dipole Moment',...
254             'Precision','Accuracy','FoV x','FoV y',...
255             'Noise','Range','Position Acc. ','Velocity Acc.',...
256             'GNSS','Drift'};;
257
258     array_text = fieldlabels(ismember(fieldvalues,array_values));
259 end
260
261 function range = getExcelNextEmptyRange(obj,raw) % ...
262     GETEXCELNEXTEMPTYRANGE Gets range of the next empty line
263     nr_rows = size(raw,1); % ...
264         Get number of rows and cols in database
265     nr_cols = size(raw,2); % ...
266     clear raw;
267     str_col = obj.getExcelCol(nr_cols); % ...
268         Convert column number to column name A-Z AA-AZ ...
269     str_row = num2str(nr_rows + 1); % ...
270         Get target row
271     range = strcat('A',str_row,':',str_col,str_row); % ...
272         Make range string
273 end
274
275 function str = getExcelCol(~,nr) % ...
276     GETEXCELCOL Converts column number to string A-Z AA-AZ ...
277     letters = ... % ...
278         {'A','B','C','D','E','F','G','H','I','J','K','L','M','N','O',...
279             'P','Q','R','S','T','U','V','W','X','Y','Z'};
280     N = length(letters);
281     if nr <= N
282         str = letters{nr}; % ...
283             Single-letter column name
284     else
285         first_letter = letters{floor(nr/N)};
286         second_letter = letters{nr - N * floor(nr/N)};
287         str = strcat(first_letter,second_letter); % ...
288             Double-letter column name
289     end
290 end
291
292 function log(~,str)
293     if ischar(str) % ...
294         LOG Saves user actions in text file
295         fileID = fopen('registry.log','a'); % ...
296             Open log file or create it
297         fprintf(fileID,'%25s\t',datestr(now,'HH:MM:SS dd/mm/yyyy'));% ...
298             Timestamp
299         fprintf(fileID,'%s\r\n',str); % ...
300             Write message
301         fclose(fileID);
302     else
303         error('Input variable is not a string type');
304 end

```



```

290         end
291     end
292
293 end
294 % ===== %
295 methods (Access = private)
296
297
298
299 function [table,thead,tbody] = raw2table(~,raw)
300     % ... RAW2TABLE Aux function that gets table from raw cell array
301     if size(raw,1) < 1
302         error('raw2table: Raw array does not have any data.');
303     end
304     thead = raw(1,:);
305     % ... Split headers from data
306     tbody = raw(2:end,:);
307     table = cell2table(tbody);
308     % ... Convert data to a table
309     table.Properties.VariableNames = thead;
310 end
311
312
313
314 function [raw] = table2raw(~,table)
315     % ... TABLE2RAW Aux function that gets raw cell array from table
316     C = table2cell(table);
317     % ... Get table body
318     raw = [table.Properties.VariableNames;C];
319     % ... Concatenate table headers
320
321 end
322
323
324 end
325 end

```

Código B.2: Model.m

```

1 classdef Model
2     %MODEL Summary of this class goes here
3     % Detailed explanation goes here
4
5     properties (Access = public)
6         table
7         amountvalues
8         sname
9     end
10
11    methods (Access = public)
12        function obj = Model(db,varargin)
13            obj.table      = obj.loadTableFromHeaders(db);
14            obj.amountvalues = [];
15            obj.sname       = 'Model';
16        end
17
18        function obj = addItem(obj,id,amount,dbtable,varargin)
19            ADDITEM Adds selected item to model from db checkout
20            row      = find(strcmp(dbtable.id,id),1);
21        end
22    end
23
```

```

    finds target row of db table with the unique ID
20   if size(obj.table,1) > 0
21       subclass = dbtable(row,:).subclass;                                % ...
22           get subclass of item
23       modelRow = find(strcmp(obj.table.subclass,subclass),1);          % ...
24           check if subclass is already in model
25       if modelRow ~= false
26           old_id = obj.table(modelRow,:).id;                            % ...
27               get previous ID to log removal
28           old_id = old_id{1,1};                                         % ...
29               convert cell to string
30           if(~strcmp(old_id,id))
31               check that previous and current do not match
32               str = sprintf('Item removed from model.\tID: %30s',old_id);
33               obj.log(str);                                            % ...
34                   print in log file
35               obj.table(modelRow,:) = dbtable(row,:);                  % ...
36                   substitute data in model
37               obj.amountvalues(modelRow) = amount;                      % ...
38                   substitute amount
39               str = sprintf('Item added to model.\t\tID: ...
39                   %30s\tAmount: %2s',id,int2str(amount));
40               obj.log(str);                                            % ...
41                   print in log file
42           else                                                       % if ...
43               they match, just check change in amount
44               old_amount = obj.amountvalues(modelRow);                 % ...
45                   get previous amount
46               if(amount ~= old_amount)                                    % if ...
47                   amounts do not match, change amount
48                   str = sprintf('Amount altered in model.\tID: ...
48                   %30s\tAmount: %2s',id,int2str(amount));
49                   obj.log(str);                                            % ...
50                       print in log file
51               end
52           end
53       else
54           obj.table = [obj.table; dbtable(row,:)];                     % ...
55               append data
56           obj.amountvalues = [obj.amountvalues; amount];                % ...
57               append amount
58       end
59   else
60       obj.table = dbtable(row,:);                                     % ...
61           init table
62       obj.amountvalues = amount;                                     % ...
63           init amount
64       str = sprintf('Item added to model.\t\tID: %30s\tAmount: ...
64           %2s',id,int2str(amount));
65       obj.log(str);                                                % ...
66           print in log file
67   end
68 end

```

```

53     function obj = removeItem(obj,table,id,varargin) % ...
54         REMOVEITEM Removes selected item from the model
55         row = find(strcmp(table.id,id),1); % ...
56             finds target row of model table with the unique ID
57         obj.table(row,:) = [];
58         obj.amountvalues(row) = [];
59         str = sprintf('Item removed from model.\tID: %30s',id); % ...
60             print in log file
61         obj.log(str);
62     end
63
64     function exportModel(obj,app) % ...
65         EXPORTMODEL Saves model table to Excel file
66         xlswrite(app.db.filename, cell(80,80), obj.sname,'A2:CB81'); % ...
67             Clear previous model in file
68         amount = num2cell(app.model.amountvalues); % ...
69             Get amount values
70         data = table2cell(app.model.table); % ...
71             Get array to post in database
72         data = [data(:,1:4) amount data(:,5:end)]; % ...
73         xlswrite(app.db.filename, data, obj.sname,'A2'); % ...
74             Write data in Excel
75         obj.log('Model exported to Excel');
76     end
77
78     end
79
80     methods (Access = private)
81
82         function log(~,str)
83             if ischar(str) % ...
84                 LOG Saves user actions in text file
85                 fileID = fopen('registry.log','a'); % ...
86                     Open log file or create it
87                 fprintf(fileID, '%25s\t',datestr(now,'HH:MM:SS dd/mm/yyyy'));% ...
88                     Timestamp
89                 fprintf(fileID, '%s\r\n',str); % ...
90                     Write message
91                 fclose(fileID);
92             else
93                 error('Input variable is not a string type');
94             end
95         end
96
97         function table = loadTableFromHeaders(~,db) % ...
98             Loads model table fields from the db table
99             table = cell2table(db.table.Properties.VariableNames); % ...
100                Converts fields cell array to a single-row table
101                table.Properties.VariableNames = table{1,:}; % ...
102                    Assigns first row to fields of the table
103                    table(1,:) = []; % ...
104                        Clears first row with temp values
105
106        end
107    end
108 end

```

Código B.3: app.mapp

```

1 classdef appl < matlab.apps.AppBase
2
3 % Properties that correspond to app components
4 properties (Access = public)
5     UI
6         TabGroup
7             ModelTab
8                 RWPanel
9                     AdditemRW
10                    LabelRW
11                    SSPanel
12                        AdditemSS
13                        LabelSS
14                        STPanel
15                            AdditemST
16                            LabelST
17                            ESPanel
18                                AdditemES
19                                LabelES
20                                MTPanel
21                                    AdditemMT
22                                    LabelMT
23                                    MGPanel
24                                        AdditemMG
25                                        LabelMG
26                                        IMPanel
27                                            AdditemIM
28                                            LabelIM
29                                            GNPanel
30                                                AdditemGN
31                                                LabelGN
32                                                OAPanel
33                                                    AdditemOA
34                                                    LabelOA
35                                                    ModelToolbox
36                                                    Viewspecs
37                                                    Createitem
38                                                    Searchitem
39                                                    Removeitem
40                                                    ExportModel
41                                                    MWPanel
42                                                    AdditemMW
43                                                    LabelMW
44                                                    OSPanel
45                                                    AdditemOS
46                                                    LabelOS
47                                                    DOPanel
48                                                    AdditemDO
49                                                    LabelDO
50                                                    DatabaseTab
51                                                    UITable
52                                                    SortbyDropDownLabel
53                                                    Sortby
54                                                    ImsearchingforDropDownLabel
55                                                    Class

```

UI	matlab.ui.Figure
TabGroup	matlab.ui.container.TabGroup
ModelTab	matlab.ui.container.Tab
RWPanel	matlab.ui.container.Panel
AdditemRW	matlab.ui.control.Button
LabelRW	matlab.ui.control.Label
SSPanel	matlab.ui.container.Panel
AdditemSS	matlab.ui.control.Button
LabelSS	matlab.ui.control.Label
STPanel	matlab.ui.container.Panel
AdditemST	matlab.ui.control.Button
LabelST	matlab.ui.control.Label
ESPanel	matlab.ui.container.Panel
AdditemES	matlab.ui.control.Button
LabelES	matlab.ui.control.Label
MTPanel	matlab.ui.container.Panel
AdditemMT	matlab.ui.control.Button
LabelMT	matlab.ui.control.Label
MGPanel	matlab.ui.container.Panel
AdditemMG	matlab.ui.control.Button
LabelMG	matlab.ui.control.Label
IMPanel	matlab.ui.container.Panel
AdditemIM	matlab.ui.control.Button
LabelIM	matlab.ui.control.Label
GNPanel	matlab.ui.container.Panel
AdditemGN	matlab.ui.control.Button
LabelGN	matlab.ui.control.Label
OAPanel	matlab.ui.container.Panel
AdditemOA	matlab.ui.control.Button
LabelOA	matlab.ui.control.Label
ModelToolbox	matlab.ui.container.Panel
Viewspecs	matlab.ui.control.Button
Createitem	matlab.ui.control.Button
Searchitem	matlab.ui.control.Button
Removeitem	matlab.ui.control.Button
ExportModel	matlab.ui.control.Button
MWPanel	matlab.ui.container.Panel
AdditemMW	matlab.ui.control.Button
LabelMW	matlab.ui.control.Label
OSPanel	matlab.ui.container.Panel
AdditemOS	matlab.ui.control.Button
LabelOS	matlab.ui.control.Label
DOPanel	matlab.ui.container.Panel
AdditemDO	matlab.ui.control.Button
LabelDO	matlab.ui.control.Label
DatabaseTab	matlab.ui.container.Tab
UITable	matlab.ui.control.Table
SortbyDropDownLabel	matlab.ui.control.Label
Sortby	matlab.ui.control.DropDown
ImsearchingforDropDownLabel	matlab.ui.control.Label
Class	matlab.ui.control.DropDown

56	CheckoutPanel	matlab.ui.container.Panel
57	AmountSpinnerLabel	matlab.ui.control.Label
58	Amount	matlab.ui.control.Spinner
59	Additem	matlab.ui.control.Button
60	ItemselectedEditFieldLabel	matlab.ui.control.Label
61	Itemselected	matlab.ui.control.EditField
62	SearchfiltersPanel	matlab.ui.container.Panel
63	Maxmass	matlab.ui.control.NumericEditField
64	Maxpower	matlab.ui.control.NumericEditField
65	Maxtemp	matlab.ui.control.NumericEditField
66	Mintemp	matlab.ui.control.NumericEditField
67	CheckBoxMaxmass	matlab.ui.control.CheckBox
68	CheckBoxMaxpower	matlab.ui.control.CheckBox
69	CheckBoxMaxtemp	matlab.ui.control.CheckBox
70	CheckBoxMintemp	matlab.ui.control.CheckBox
71	MaxmassLabel	matlab.ui.control.Label
72	MaxpowerLabel	matlab.ui.control.Label
73	MaxtempLabel	matlab.ui.control.Label
74	MintempLabel	matlab.ui.control.Label
75	InparticularDropDownLabel	matlab.ui.control.Label
76	Subclass	matlab.ui.control.DropDown
77	CreateitemTab	matlab.ui.container.Tab
78	ModelToolbox_2	matlab.ui.container.Panel
79	SaveItemInDatabase	matlab.ui.control.Button
80	CreateItemDropdown	matlab.ui.control.DropDown
81	GeneralfeaturesPanel	matlab.ui.container.Panel
82	MassEditFieldLabel	matlab.ui.control.Label
83	MassEditField	matlab.ui.control.NumericEditField
84	PowerEditFieldLabel	matlab.ui.control.Label
85	PowerEditField	matlab.ui.control.NumericEditField
86	PriceEditFieldLabel	matlab.ui.control.Label
87	PriceEditField	matlab.ui.control.EditField
88	TmaxEditFieldLabel	matlab.ui.control.Label
89	TmaxEditField	matlab.ui.control.NumericEditField
90	TminEditFieldLabel	matlab.ui.control.Label
91	TminEditField	matlab.ui.control.NumericEditField
92	NameEditFieldLabel	matlab.ui.control.Label
93	NameEditField	matlab.ui.control.EditField
94	SupplierEditFieldLabel	matlab.ui.control.Label
95	SupplierEditField	matlab.ui.control.EditField
96	URLEditFieldLabel	matlab.ui.control.Label
97	URLEditField	matlab.ui.control.EditField
98	WidthEditFieldLabel	matlab.ui.control.Label
99	WidthEditField	matlab.ui.control.NumericEditField
100	LengthEditFieldLabel	matlab.ui.control.Label
101	LengthEditField	matlab.ui.control.NumericEditField
102	HeightEditFieldLabel	matlab.ui.control.Label
103	HeightEditField	matlab.ui.control.NumericEditField
104	LifespanEditFieldLabel	matlab.ui.control.Label
105	LifespanEditField	matlab.ui.control.EditField
106	COTSCheckBox	matlab.ui.control.CheckBox
107	CubeSatCheckBox	matlab.ui.control.CheckBox
108	TRLDropDownLabel	matlab.ui.control.Label
109	TRLDropDown	matlab.ui.control.DropDown
110	VnomEditFieldLabel	matlab.ui.control.Label
111	VnomEditField	matlab.ui.control.NumericEditField

```

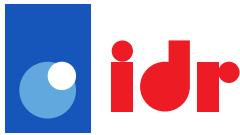
112      IavgEditFieldLabel    matlab.ui.control.Label
113      IavgEditField       matlab.ui.control.NumericEditField
114      IpeakEditFieldLabel matlab.ui.control.Label
115      IpeakEditField      matlab.ui.control.NumericEditField
116      SpecificfeauresPanel matlab.ui.container.Panel
117      Spec1EditField      matlab.ui.control.EditField
118      Spec2EditField      matlab.ui.control.EditField
119      Spec3EditField      matlab.ui.control.EditField
120      Spec4EditField      matlab.ui.control.EditField
121      Spec5EditField      matlab.ui.control.EditField
122      Spec6EditField      matlab.ui.control.EditField
123      Spec1EditFieldLabel matlab.ui.control.Label
124      Spec2EditFieldLabel matlab.ui.control.Label
125      Spec3EditFieldLabel matlab.ui.control.Label
126      Spec4EditFieldLabel matlab.ui.control.Label
127      Spec5EditFieldLabel matlab.ui.control.Label
128      Spec6EditFieldLabel matlab.ui.control.Label
129      NotesTextAreaLabel matlab.ui.control.Label
130      NotesTextArea       matlab.ui.control.TextArea
131      SettingsTab        matlab.ui.container.Tab
132      FieldstodisplayonsearchresultsPanel matlab.ui.container.Panel
133      Panel                matlab.ui.container.Panel
134      NameCheckBox         matlab.ui.control.CheckBox
135      SupplierCheckBox     matlab.ui.control.CheckBox
136      MassCheckBox         matlab.ui.control.CheckBox
137      PowerCheckBox        matlab.ui.control.CheckBox
138      CubeSatCheckBox_3    matlab.ui.control.CheckBox
139      COTSCheckBox_3       matlab.ui.control.CheckBox
140      DimensionsCheckBox matlab.ui.control.CheckBox
141      ThermalfeaturesCheckBox matlab.ui.control.CheckBox
142      ElectricalfeaturesCheckBox matlab.ui.control.CheckBox
143      SpecificfeaturesCheckBox matlab.ui.control.CheckBox
144      LifespanCheckBox     matlab.ui.control.CheckBox
145      PriceCheckBox        matlab.ui.control.CheckBox
146      ExcelsettingsPanel   matlab.ui.container.Panel
147      DatabasesheetnameEditFieldLabel matlab.ui.control.Label
148      DatabasesheetnameEditField  matlab.ui.control.EditField
149      ModelsheetnameEditFieldLabel matlab.ui.control.Label
150      ModelsheetnameEditField  matlab.ui.control.EditField
151      FilenameEditFieldLabel matlab.ui.control.Label
152      FilenameEditField     matlab.ui.control.EditField
153      ModelToolbox_3        matlab.ui.container.Panel
154      SaveItemInDatabase_2  matlab.ui.control.Button
155      SaveItemInDatabase_3  matlab.ui.control.Button
156      TotalmassEditFieldLabel matlab.ui.control.Label
157      Totalmass             matlab.ui.control.NumericEditField
158      TotalpowerEditFieldLabel matlab.ui.control.Label
159      Totalpower            matlab.ui.control.NumericEditField
160  end
161
162
163 properties (Access = public)
164     db                      % Holds the Database properties and methods ...
165     Catalog with items
166     model                  % Holds the Model properties and methods ...
167     Selected items

```

```

166     SearchSelection           % Stores the previous selection in Database tab
167     ItemselectedID          % Holds the unique ID of the item added to the ...
168         checkout area
169     ModelSelectionID        % Holds the unique ID of the current selection in ...
170         Model tab
171     ModelBoxSubclass         % Stores the current selection subclass in Model tab
172     ArrayModelLabels
173     ArrayModelPanels
174     ArrayModelButtons
175     ArraySubclasses
176 end
177
178 methods (Access = private)
179     % ...
180     =====
181     %
182     % Enables or disables database filter editfields with checkboxes
183     function checkbox2editfield(app,target,value)
184         obj_array = {app.Maxmass, app.Maxpower, app.Maxtemp, app.Mintemp};
185         label_array = {app.MaxmassLabel, app.MaxpowerLabel, ...
186                         app.MaxtempLabel, app.MintempLabel};
187         if(value == 1)
188             state = 'on';
189         else
190             state = 'off';
191         end
192         obj_array{target}.Enable = state;
193         label_array{target}.Enable = state;
194         obj_array{target}.Value = 0;
195     end
196     %
197     =====
198     %
199     % Refresh subclass dropdown (category within actuators or sensors)
200     function refreshSubclass(app,varargin)
201         switch(app.Class.Value)
202             case 'actuator'
203                 names_array = {'a reaction wheel','a momentum wheel','a ...
204                                 magnetorquer','other actuator'};
205                 values_array = {'reaction wheel','momentum ...
206                                 wheel','magnetorquer','other actuator'};
207                 enable_box = 'on';
208             case 'sensor'
209                 names_array = {'a sun sensor','a star sensor','an earth ...
210                                 sensor','a magnetometer','an imu','a gnss ...
211                                 receiver','other sensor'};
212                 values_array = {'sun sensor','star sensor','earth ...
213                                 sensor','magnetometer','imu','gnss','other sensor'};
214                 enable_box = 'on';
215             case 'dock'
216                 names_array = {'not available'};
217                 values_array = {''};
218                 enable_box = 'off';
219             end
220             app.Subclass.Items      = names_array;
221             app.Subclass.ItemsData = values_array;

```



```

210     app.Subclass.Enable      = enable_box;
211 end
212 %
213 % Refresh search table in the GUI
214 function refreshTable(app)
215     % Load, sort, filter and prompt table from Excel
216     table_filtered      = ...
217     app.db.filterData(app.db.table,app.Class.Value,app.Subclass.Value);
218     app.db.sortedtable = app.db.sortData(table_filtered,app.Sortby.Value);
219     [app.UITable.ColumnName,app.UITable.Data] = ...
220         app.db.promptData(app.db.sortedtable,app.Subclass.Value);
221     % Reset search
222     app.SearchSelection = -1;
223 end
224 %
225 %
226 % Displays values on the Model tab
227 function printLabel(app,name,subclass)
228     % Find subclass index
229     row = find(strcmp(app.ArraySubclasses,subclass),1);
230     % Assign name to label
231     app.ArrayModelLabels{row}.Text = name;
232
233 end
234 %
235 %
236 % Refresh model selection and toolbox
237 function refreshModelSelection(app,trigger)
238     % OPTION #1: FUNCTION TRIGGERED FROM A 'SELECT' BUTTON
239     if ismember(trigger,app.ArraySubclasses)
240         % Get index of trigger in array
241         index = find(strcmp(app.ArraySubclasses,trigger),1);
242         % Toggle button
243         str = app.ArrayModelButtons{index}.Text;
244         if strcmpi(str,'select')
245             % Fade everything
246             for i=1:length(app.ArrayModelPanels)
247                 app.ArrayModelPanels{i}.BackgroundColor = [0.94,0.94,0.94];
248                 app.ArrayModelButtons{i}.Enable = 'off';
249             end
250             % Toggle button
251             app.ArrayModelButtons{index}.Enable = 'on';
252             app.ArrayModelButtons{index}.Text = 'Unselect';
253             app.ArrayModelButtons{index}.BackgroundColor = [0,0.5,0.9];
254             % Store selected subclass
255             app.ModelBoxSubclass = app.ArraySubclasses{index};
256             % Check if there is an item selected
257             str = app.ArrayModelLabels{index}.Text;
258             if strcmpi(str,'no items yet')
259                 % Toggle toolbox buttons
260                 enable_values = {'off','on','on','off'};
261             else

```

```

258         % Store selected ID too
259         id = ...
260         app.model.table(ismember(app.model.table.subclass,trigger),:).id;
261         app.ModelSelectionID = id{1,1};
262         % Toggle toolbox buttons
263         enable_values = {'on','on','on','on'};
264     end
265     % Highlight panel
266     app.ArrayModelPanels{index}.BackgroundColor = [1.0,1.0,1.0];
267 else
268     app.ArrayModelButtons{index}.Text = 'Select';
269     app.ArrayModelButtons{index}.BackgroundColor = [0.2,0.7,0.5];
270     % Reset highlight
271     for i=1:length(app.ArrayModelPanels)
272         app.ArrayModelPanels{i}.BackgroundColor = [1.0,1.0,1.0];
273         app.ArrayModelButtons{i}.Enable = 'on';
274     end
275     % Reset selected subclass, ID and toolbox
276     app.ModelBoxSubclass = '';
277     app.ModelSelectionID = '';
278     enable_values = {'off','off','off','off'};
279 end
280
281     % OPTION #2: RESET MODEL UI AFTER REMOVAL
282 else
283     % Reset highlight
284     for i=1:length(app.ArrayModelPanels)
285         app.ArrayModelPanels{i}.BackgroundColor = [1.0,1.0,1.0];
286         app.ArrayModelButtons{i}.Enable = 'on';
287         app.ArrayModelButtons{i}.Text = 'Select';
288         app.ArrayModelButtons{i}.BackgroundColor = [0.2,0.7,0.5];
289     end
290     app.ModelBoxSubclass = '';
291     enable_values = {'off','off','off','off'};
292 end
293     % Enable/disable toolbox buttons
294     app.Viewspecs.Enable = enable_values{1};
295     app.Createitem.Enable = enable_values{2};
296     app.Searchitem.Enable = enable_values{3};
297     app.Removeitem.Enable = enable_values{4};
298 end
299 %
300 % Refresh Model tab, Mass, Power in the GUI
301 function refreshModelTab(app)
302     % Get Model table
303     table = app.model.table;
304     amount = app.model.amountvalues;
305     % Zero mass and power
306     mass = 0;
307     power = 0;
308     % Assign values to labels
309     if size(table,1) > 0
310         for i = 1:size(table,1)
311             subclass = table(i,:).subclass;

```

```

311         name      = table(i,:).name;
312         printLabel(app,name,subclass);
313         mass     = mass + table(i,:).mass * amount(i);
314         power   = power + table(i,:).power * amount(i);
315     end
316 end
317 % Refresh mass and power
318 app.Totalmass.Value = mass;
319 app.Totalpower.Value = power;
320 % Refresh labels in model boxes with no items
321 for i=1:length(app.ArrayModelLabels)
322     if ~ismember(app.ArraySubclasses{i},table.subclass)
323         app.ArrayModelLabels{i}.Text = 'No items yet';
324     end
325 end
326 end
327 %
328 =====
329 % Get model item specifications by
330 function specs_str = getSpecsByID(app,id,table,amountvalues)
331     % Get position of selected item
332     row    = find(strcmp(table.id,id),1);
333     amount = amountvalues(row);
334     item   = table(row,:);
335     % Get subclass and filter fields
336     subclass = item.subclass{1,1};
337     specific_fields = app.db.getSpecificFieldsBySubclass(subclass);
338     common_fields = ...
339         {'name','supplier','mass','power','v_nom','dim_x','dim_y','dim_z','temp_max','temp_...
340     % Print amount
341     specs_str = sprintf('Amount: %s\n',num2str(amount));
342     % Generate string rows with item info
343     for i=1:size(item,2)
344         % Get field and value pair
345         field = table.Properties.VariableNames{i};
346         % Filter fields
347         if ismember(field,common_fields) || ismember(field,specific_fields)
348             if ~isempty(app.db.headerValues2Text(field))
349                 field = app.db.headerValues2Text(field);
350             end
351             if iscell(field)
352                 field = field{1,1};
353             end
354             value = item{1,i};
355             % Convert 1x1 cell to string
356             if iscell(value)
357                 value = value{1,1};
358             else
359                 if isnan(value)
360                     value = 'NaN';
361                 else
362                     value = num2str(value);
363                 end
364             end
365             specs_str = sprintf('%s %s',specs_str,sprintf('%s: ...'

```

```

364           %s\n',field,value));
365       end
366   end
367   % ...
368   =====
369   % Manages checkboxes in the GUI table
370   function selectItem(app)
371       % Get table data
372       table = app.UITable.Data;
373
374       % Count rows of Table
375       rows = size(table,1);
376       values = zeros(rows,1);
377
378       % Store values (0 or 1)
379       for i=1:rows
380           values(i) = table{i,1};
381       end
382
383       % Check if every box has been unselected
384       if isequal(values,zeros(rows,1))
385           app.Itemselected.Value = "";
386           % Clear ...
387           checkout field
388           app.SearchSelection = -1;
389           % Reset ...
390           selection history
391       else
392           % One box and previous checked simultaneously: remove previous ...
393           check
394           % and refresh values vector
395           if(app.SearchSelection ~= -1)
396               values = abs(values-app.SearchSelection); % Get the ...
397               difference
398               for i=1:rows % Refresh ...
399                   table in the GUI
400                   app.UITable.Data{i,1} = logical(values(i));
401               end
402           end
403           row = find(values,1); % Find the ...
404           checked box
405           item_name = table{row,2}; % Get the ...
406           name of the checked item
407           app.Itemselected.Value = item_name; % Paste it ...
408           in the checkout field
409           app.ItemselectedID = app.db.sortedtable{row,1}; % Save the ...
410           ID of the checked item
411           app.ItemselectedID = app.ItemselectedID{1}; % Convert ...
412           1x1 cell to pure string
413           app.SearchSelection = values; % Save ...
414           selection
415       end
416   end
417   % ...
418   =====
419   %

```

```

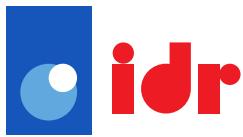
404     % Boots app
405     function init(app)
406         clc;
407
408         % Set directory
409         p = mfilename('fullpath');
410         [folder,~,~] = fileparts(p);
411         cd(folder);
412
413         % Set dropdowns
414         app.Class.Value = 'actuator';
415         app.Sortby.Value = 'name';
416         refreshSubclass(app);
417
418         % Create Database and Model instance
419         app.db      = Database;
420         app.model   = Model(app.db);
421
422         % Store other values
423         app.ArrayModelPanels = {app.DOPanel, app.RWPanel, app.MWPanel, ...
424                                 app.MTPanel, app.OAPanel, app.SSPanel, app.STPanel, ...
425                                 app.ESPanel, app.MGPanel, app.IMPanel, ...
426                                 app.GNPanel, app.OSPanel};
427         app.ArrayModelButtons = ...
428             {app.AddItemDO, app.AddItemRW, app.AddItemMW, app.AddItemMT, app.AddItemOA, app.AddItemES, app.AddItemMG, app.AddItemIM, app.AddItemGN, app.AddItemST, app.LabelDO, app.LabelRW, app.LabelMW, app.LabelMT, app.LabelOA, app.LabelSS, app.LabelST, app.LabelES, app.LabelMG, app.LabelIM, app.LabelGN, app.LabelOS};
429         app.ArrayModelLabels = {app.LabelDO, app.LabelRW, app.LabelMW, ...
430                                 app.LabelMT, app.LabelOA, app.LabelSS, app.LabelST, ...
431                                 app.LabelES, app.LabelMG, app.LabelIM, ...
432                                 app.LabelGN, app.LabelOS};
433         app.ArraySubclasses = {'dock', 'reaction wheel', 'momentum ...
434                               wheel', 'magnetorquer', 'other actuator', 'sun sensor', 'star ...
435                               sensor', ...
436                               'earth ...
437                               sensor', 'magnetometer', 'imu', 'gnss', 'other ...
438                               sensor'};
439
440     end
441     %
442
443 end
444
445 methods (Access = private)
446
447     % Code that executes after component creation
448     function startupFcn(app)
449         init(app);           % Boot app
450         refreshTable(app); % Refresh table with items
451         CreateItemDropdownValueChanged(app); % Refresh dropdown in 'create ...
452                                         item' tab
453     end
454
455     % Value changed function: Sortby
456     function SortbyValueChanged(app, event)
457         refreshTable(app); % Refresh table with items

```

```

448     end
449
450     % Value changed function: Class
451     function ClassValueChanged(app, event)
452         refreshSubclass(app); % Refresh subclass dropdown
453         refreshTable(app);   % Refresh table with items
454     end
455
456     % Value changed function: Subclass
457     function SubclassValueChanged(app, event)
458         refreshTable(app);   % Refresh table with items
459     end
460
461     % Cell edit callback: UITableView
462     function UITableViewCellEdit(app, event)
463         selectItem(app);      % Interact with tickboxes
464     end
465
466     % Value changed function: CheckBoxMaxmass
467     function CheckBoxMaxmassValueChanged(app, event)
468         value = app.CheckBoxMaxmass.Value;
469         checkbox2editfield(app,1,value)
470     end
471
472     % Value changed function: CheckBoxMaxpower
473     function CheckBoxMaxpowerValueChanged(app, event)
474         value = app.CheckBoxMaxpower.Value;
475         checkbox2editfield(app,2,value)
476     end
477
478     % Value changed function: CheckBoxMaxtemp
479     function CheckBoxMaxtempValueChanged(app, event)
480         value = app.CheckBoxMaxtemp.Value;
481         checkbox2editfield(app,3,value)
482     end
483
484     % Value changed function: CheckBoxMintemp
485     function CheckBoxMintempValueChanged(app, event)
486         value = app.CheckBoxMintemp.Value;
487         checkbox2editfield(app,4,value)
488     end
489
490     % Button pushed function: Additem
491     function AdditemButtonPushed(app, event)
492         app.model = ...
493             app.model.addItem(app.ItemselectedID,app.Amount.Value,app.db.table); ...
494                 % Add an item to the Model
495         refreshModelTab(app);
496         refreshModelSelection(app,'');
497         app.TabGroup.SelectedTab = app.ModelTab;
498     end
499
500     % Button pushed function: AdditemSS
501     function AdditemSSButtonPushed(app, event)
502         refreshModelSelection(app,'sun sensor');
503     end

```



```
502 % Button pushed function: AdditemDO
503 function AdditemDOButtonPushed(app, event)
504     refreshModelSelection(app, 'dock');
505 end
506
507 % Button pushed function: AdditemOS
508 function AdditemOSButtonPushed(app, event)
509     refreshModelSelection(app, 'other sensor');
510 end
511
512 % Button pushed function: AdditemRW
513 function AdditemRWButtonPushed(app, event)
514     refreshModelSelection(app, 'reaction wheel');
515 end
516
517 % Button pushed function: AdditemST
518 function AdditemSTButtonPushed(app, event)
519     refreshModelSelection(app, 'star sensor');
520 end
521
522 % Button pushed function: AdditemES
523 function AdditemESButtonPushed(app, event)
524     refreshModelSelection(app, 'earth sensor');
525 end
526
527 % Button pushed function: AdditemMT
528 function AdditemMTButtonPushed(app, event)
529     refreshModelSelection(app, 'magnetorquer');
530 end
531
532 % Button pushed function: AdditemMG
533 function AdditemMGButtonPushed(app, event)
534     refreshModelSelection(app, 'magnetometer');
535 end
536
537 % Button pushed function: AdditemIM
538 function AdditemIMButtonPushed(app, event)
539     refreshModelSelection(app, 'imu');
540 end
541
542 % Button pushed function: AdditemGN
543 function AdditemGNButtonPushed(app, event)
544     refreshModelSelection(app, 'gnss');
545 end
546
547 % Button pushed function: AdditemOA
548 function AdditemOAButtonPushed(app, event)
549     refreshModelSelection(app, 'other actuator');
550 end
551
552 % Button pushed function: Removeitem
553 function RemoveitemButtonPushed(app, event)
554     if ~strcmp(app.ModelSelectionID, '')
555         app.model = ...
556             app.model.removeItem(app.model.table, app.ModelSelectionID); ...
```

```

557         % Remove an item from the Model
558         refreshModelSelection(app,'') % Reset Model tab UI
559         refreshModelTab(app);
560     end
561 end

562 % Button pushed function: Searchitem
563 function SearchitemButtonPushed(app, event)
564     % Get selected box origin
565     % Transform to class and subclass
566     % Modify Database tab with the search query
567     % Go to Tab
568     app.TabGroup.SelectedTab = app.DatabaseTab;
569 end

570 % Button pushed function: Viewspecs
571 function ViewspecsButtonPushed(app, event)
572     specs_str = ...
573     getSpecsByID(app,app.ModelSelectionID,app.model.table,app.model.amountvalues)
574     uialert(app.UI,specs_str,'Specifications','Icon','','Modal',true);
575 end

576 % Button pushed function: Createitem
577 function CreateitemButtonPushed(app, event)
578     % Get subclass and filter fields for label titles
579     subclass = app.ModelBoxSubclass;
580     % Change subclass dropdown
581     app.CreateItemDropdown.Value = subclass;
582     % Go to Tab
583     app.TabGroup.SelectedTab = app.CreateitemTab;
584 end

585 % Value changed function: CreateItemDropdown
586 function CreateItemDropdownValueChanged(app, event)
587     subclass = app.CreateItemDropdown.Value;
588     % Get specific fields
589     specific_fields = app.db.getSpecificFieldsBySubclass(subclass);
590     specific_fields = app.db.headerValues2Text(specific_fields);
591     specific_textboxes = ...
592         {app.Spec1EditField,app.Spec2EditField,app.Spec3EditField,app.Spec4EditField,
593     specific_labels = ...
594         {app.Spec1EditFieldLabel,app.Spec2EditFieldLabel,app.Spec3EditFieldLabel,app.
595     % Assign titles to textboxes
596     for i=1:length(specific_textboxes)
597         if i<length(specific_fields)
598             specific_labels{i}.Text = specific_fields{i};
599             specific_labels{i}.Visible = 'on';
600             specific_textboxes{i}.Visible = 'on';
601         else
602             specific_labels{i}.Visible = 'off';
603             specific_textboxes{i}.Visible = 'off';
604         end
605     end
606 end
607 end
608

```

```

609     % Button pushed function: SaveItemInDatabase
610     function SaveItemInDatabaseButtonPushed(app, event)
611         % Check that required fields are filled
612         str_req_fields = {'Name', 'Supplier'};
613         required_fields = {app.NameEditField, app.SupplierEditField};
614         filled = true;
615         for i=1:length(required_fields)
616             if isempty(required_fields{i}.Value)
617                 uialert(app.UI,sprintf('%s is not ...  
filled',str_req_fields{i}),'Form not ...  
filled','Icon','error','Modal',true);
618                 filled = false;
619             end
620         end
621         if filled == true
622             % Check if file exists
623             if ~xlsfinfo(app.db.filename)
624                 uialert(app.UI,'Excel file does not exist. Check the name ...  
of the file.','Item not ...  
created.','Icon','error','Modal',true);
625             else
626                 % Check if file is open by other program
627                 [fid, ~] = fopen(app.db.filename,'a');
628                 if fid== -1
629                     uialert(app.UI,'Please, close the Excel file.','Excel ...  
file opened. Item not ...  
created.','Icon','error','Modal',true);
630                 else
631                     fclose(fid);
632                     % Create item in db table and write in Excel file
633                     app.db.createItem(app);
634                     uialert(app.UI,'Item added to Excel database ...  
successfully.','Item ...  
created','Icon','success','Modal',true);
635                 end
636             end
637         end
638     end
639
640     % Close request function: UI
641     function UICloseRequest(app, event)
642         selection = uiconfirm(app.UI,'Do you really want to close the ...  
GUI?','Please ...  
confirm','Options',{ 'Yes', 'No' },'DefaultOption',1,'CancelOption',2);
643         if strcmpi(selection,'yes')
644             app.db.log('GUI closed')
645             delete(app)
646         end
647     end
648
649     % Button pushed function: AdditemMW
650     function AdditemMWButtonPushed(app, event)
651         refreshModelSelection(app, 'momentum wheel');
652     end
653
654     % Button pushed function: ExportModel

```

```

655     function ExportModelButtonPushed(app, event)
656         if size(app.model.table,1) == 0
657             selection = uiconfirm(app.UI,'There are no items selected. ....
658                                         Exporting an empty model will erase the previous data in ...
659                                         the spreadsheet. Do you want to continue?','Please ...
660                                         confirm','Options',{'Yes','No'},'DefaultOption',1,'CancelOption',2);
661         if strcmpi(selection,'yes')
662             do_it = true;
663         else
664             do_it = false;
665         end
666         % Proceed with exporting
667         if do_it
668             % Check if file exists
669             if ~xlsfinfo(app.db.filename)
670                 uialert(app.UI,'Excel file does not exist. Check the name ...
671                                         of the file.','Model not ...
672                                         exported.','Icon','error','Modal',true);
673             else
674                 % Check if file is open by other program
675                 [fid, ~] = fopen(app.db.filename,'a');
676                 if fid== -1
677                     uialert(app.UI,'Please, close the Excel file.','Excel ...
678                                         file opened. Model not ...
679                                         exported.','Icon','error','Modal',true);
680                 else
681                     fclose(fid);
682                     % Export items
683                     app.model.exportModel(app);
684                     uialert(app.UI,'Model exported to Excel file ...
685                                         successfully.','Exporting ...
686                                         ended','Icon','success','Modal',true);
687                 end
688             end
689         end
690
691         % App initialization and construction
692         methods (Access = private)
693
694             % Create UI
695             app.UI = uifigure;
696             app.UI.Color = [0 0.102 0.4];
697             app.UI.Position = [100 100 896 563];
698             app.UI.Name = 'CDF - ADCS module';
699             app.UI.Resize = 'off';
700             app.UI.CloseRequestFcn = createCallbackFcn(app, @UICloseRequest, true);
701
702             % Create TabGroup

```

```
702     app.TabGroup = uitabgroup(app.UI);
703     app.TabGroup.Position = [1 23 896 541];
704
705     % Create ModelTab
706     app.ModelTab = uitab(app.TabGroup);
707     app.ModelTab.Title = 'Model';
708     app.ModelTab.BackgroundColor = [0.9804 0.9804 0.9804];
709
710     % Create RWPanel
711     app.RWPanel = uipanel(app.ModelTab);
712     app.RWPanel.BorderType = 'none';
713     app.RWPanel.Title = 'Reaction wheels';
714     app.RWPanel.BackgroundColor = [1 1 1];
715     app.RWPanel.Position = [12 13 207 132];
716
717     % Create AdditemRW
718     app.AdditemRW = uibutton(app.RWPanel, 'push');
719     app.AdditemRW.ButtonPushedFcn = createCallbackFcn(app, ...
720         @AdditemRWButtonPushed, true);
721     app.AdditemRW.BackgroundColor = [0.2 0.702 0.502];
722     app.AdditemRW.FontColor = [1 1 1];
723     app.AdditemRW.Position = [68 22 74 21];
724     app.AdditemRW.Text = 'Select';
725
726     % Create LabelRW
727     app.LabelRW = uilabel(app.RWPanel);
728     app.LabelRW.HorizontalAlignment = 'center';
729     app.LabelRW.Position = [17 71 175 15];
730     app.LabelRW.Text = 'No items yet';
731
732     % Create SSPanel
733     app.SSPanel = uipanel(app.ModelTab);
734     app.SSPanel.BorderType = 'none';
735     app.SSPanel.Title = 'Sun sensors';
736     app.SSPanel.BackgroundColor = [1 1 1];
737     app.SSPanel.Position = [12 296 207 132];
738
739     % Create AdditemSS
740     app.AdditemSS = uibutton(app.SSPanel, 'push');
741     app.AdditemSS.ButtonPushedFcn = createCallbackFcn(app, ...
742         @AdditemSSButtonPushed, true);
743     app.AdditemSS.BackgroundColor = [0.2 0.702 0.502];
744     app.AdditemSS.FontColor = [1 1 1];
745     app.AdditemSS.Position = [68 22 74 21];
746     app.AdditemSS.Text = 'Select';
747
748     % Create LabelSS
749     app.LabelSS = uilabel(app.SSPanel);
750     app.LabelSS.HorizontalAlignment = 'center';
751     app.LabelSS.Position = [17 71 175 15];
752     app.LabelSS.Text = 'No items yet';
753
754     % Create STPanel
755     app.STPanel = uipanel(app.ModelTab);
756     app.STPanel.BorderType = 'none';
757     app.STPanel.Title = 'Star sensors';
```

```

756     app.STPanel.BackgroundColor = [1 1 1];
757     app.STPanel.Position = [234 296 207 132];
758
759     % Create AdditemST
760     app.AdditemST = uibutton(app.STPanel, 'push');
761     app.AdditemST.ButtonPushedFcn = createCallbackFcn(app, ...
762         @AdditemSTButtonPushed, true);
763     app.AdditemST.BackgroundColor = [0.2 0.702 0.502];
764     app.AdditemST.FontColor = [1 1 1];
765     app.AdditemST.Position = [71 22 74 21];
766     app.AdditemST.Text = 'Select';
767
768     % Create LabelST
769     app.LabelST = uilabel(app.STPanel);
770     app.LabelST.HorizontalAlignment = 'center';
771     app.LabelST.Position = [20 71 175 15];
772     app.LabelST.Text = 'No items yet';
773
774     % Create ESPanel
775     app.ESPanel = uipanel(app.ModelTab);
776     app.ESPanel.BorderType = 'none';
777     app.ESPanel.Title = 'Earth sensors';
778     app.ESPanel.BackgroundColor = [1 1 1];
779     app.ESPanel.Position = [455 296 207 132];
780
781     % Create AdditemES
782     app.AdditemES = uibutton(app.ESPanel, 'push');
783     app.AdditemES.ButtonPushedFcn = createCallbackFcn(app, ...
784         @AdditemESButtonPushed, true);
785     app.AdditemES.BackgroundColor = [0.2 0.702 0.502];
786     app.AdditemES.FontColor = [1 1 1];
787     app.AdditemES.Position = [68 22 74 21];
788     app.AdditemES.Text = 'Select';
789
790     % Create LabelES
791     app.LabelES = uilabel(app.ESPanel);
792     app.LabelES.HorizontalAlignment = 'center';
793     app.LabelES.Position = [17 71 175 15];
794     app.LabelES.Text = 'No items yet';
795
796     % Create MTPanel
797     app.MTPanel = uipanel(app.ModelTab);
798     app.MTPanel.BorderType = 'none';
799     app.MTPanel.Title = 'Magnetorques';
800     app.MTPanel.BackgroundColor = [1 1 1];
801     app.MTPanel.Position = [234 13 207 132];
802
803     % Create AdditemMT
804     app.AdditemMT = uibutton(app.MTPanel, 'push');
805     app.AdditemMT.ButtonPushedFcn = createCallbackFcn(app, ...
806         @AdditemMTButtonPushed, true);
807     app.AdditemMT.BackgroundColor = [0.2 0.702 0.502];
808     app.AdditemMT.FontColor = [1 1 1];
809     app.AdditemMT.Position = [68 22 74 21];
810     app.AdditemMT.Text = 'Select';

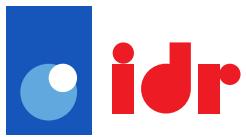
```

```
809 % Create LabelMT
810 app.LabelMT = uilabel(app.MTPanel);
811 app.LabelMT.HorizontalAlignment = 'center';
812 app.LabelMT.Position = [17 71 175 15];
813 app.LabelMT.Text = 'No items yet';
814
815 % Create MGPanel
816 app.MGPanel = uipanel(app.ModelTab);
817 app.MGPanel.BorderType = 'none';
818 app.MGPanel.Title = 'Magnetometers';
819 app.MGPanel.BackgroundColor = [1 1 1];
820 app.MGPanel.Position = [676 296 207 132];
821
822 % Create AdditemMG
823 app.AdditemMG = uibutton(app.MGPanel, 'push');
824 app.AdditemMG.ButtonPushedFcn = createCallbackFcn(app, ...
825     @AdditemMButtonPushed, true);
826 app.AdditemMG.BackgroundColor = [0.2 0.702 0.502];
827 app.AdditemMG.FontColor = [1 1 1];
828 app.AdditemMG.Position = [68 22 74 21];
829 app.AdditemMG.Text = 'Select';
830
831 % Create LabelMG
832 app.LabelMG = uilabel(app.MGPanel);
833 app.LabelMG.HorizontalAlignment = 'center';
834 app.LabelMG.Position = [17 71 175 15];
835 app.LabelMG.Text = 'No items yet';
836
837 % Create IMPanel
838 app.IMPanel = uipanel(app.ModelTab);
839 app.IMPanel.BorderType = 'none';
840 app.IMPanel.Title = 'Inertial measurement units';
841 app.IMPanel.BackgroundColor = [1 1 1];
842 app.IMPanel.Position = [455 154 207 132];
843
844 % Create AdditemIM
845 app.AdditemIM = uibutton(app.IMPanel, 'push');
846 app.AdditemIM.ButtonPushedFcn = createCallbackFcn(app, ...
847     @AdditemIMButtonPushed, true);
848 app.AdditemIM.BackgroundColor = [0.2 0.702 0.502];
849 app.AdditemIM.FontColor = [1 1 1];
850 app.AdditemIM.Position = [68 22 74 21];
851 app.AdditemIM.Text = 'Select';
852
853 % Create LabelIM
854 app.LabelIM = uilabel(app.IMPanel);
855 app.LabelIM.HorizontalAlignment = 'center';
856 app.LabelIM.Position = [17 71 175 15];
857 app.LabelIM.Text = 'No items yet';
858
859 % Create GNPanel
860 app.GNPanel = uipanel(app.ModelTab);
861 app.GNPanel.BorderType = 'none';
862 app.GNPanel.Title = 'GNSS receivers';
863 app.GNPanel.BackgroundColor = [1 1 1];
864 app.GNPanel.Position = [234 154 207 132];
```

```

863
864     % Create AdditemGN
865     app.AdditemGN = uibutton(app.GNPanel, 'push');
866     app.AdditemGN.ButtonPushedFcn = createCallbackFcn(app, ...
867         @AdditemGNButtonPushed, true);
868     app.AdditemGN.BackgroundColor = [0.2 0.702 0.502];
869     app.AdditemGN.FontColor = [1 1 1];
870     app.AdditemGN.Position = [68 22 74 21];
871     app.AdditemGN.Text = 'Select';
872
873     % Create LabelGN
874     app.LabelGN = uilabel(app.GNPanel);
875     app.LabelGN.HorizontalAlignment = 'center';
876     app.LabelGN.Position = [17 71 175 15];
877     app.LabelGN.Text = 'No items yet';
878
879     % Create OAPanel
880     app.OAPanel = uipanel(app.ModelTab);
881     app.OAPanel.BorderType = 'none';
882     app.OAPanel.Title = 'Other actuators';
883     app.OAPanel.BackgroundColor = [1 1 1];
884     app.OAPanel.Position = [676 13 207 132];
885
886     % Create AdditemOA
887     app.AdditemOA = uibutton(app.OAPanel, 'push');
888     app.AdditemOA.ButtonPushedFcn = createCallbackFcn(app, ...
889         @AdditemOAButtonPushed, true);
890     app.AdditemOA.BackgroundColor = [0.2 0.702 0.502];
891     app.AdditemOA.FontColor = [1 1 1];
892     app.AdditemOA.Position = [68 22 74 21];
893     app.AdditemOA.Text = 'Select';
894
895     % Create LabelOA
896     app.LabelOA = uilabel(app.OAPanel);
897     app.LabelOA.HorizontalAlignment = 'center';
898     app.LabelOA.Position = [17 71 175 15];
899     app.LabelOA.Text = 'No items yet';
900
901     % Create ModelToolbox
902     app.ModelToolbox = uipanel(app.ModelTab);
903     app.ModelToolbox.BorderType = 'none';
904     app.ModelToolbox.BackgroundColor = [0.9412 0.9412 0.9412];
905     app.ModelToolbox.Position = [1 439 895 78];
906
907     % Create Viewspecs
908     app.Viewspecs = uibutton(app.ModelToolbox, 'push');
909     app.Viewspecs.ButtonPushedFcn = createCallbackFcn(app, ...
910         @ViewspecsButtonPushed, true);
911     app.Viewspecs.BackgroundColor = [1 1 1];
912     app.Viewspecs.Enable = 'off';
913     app.Viewspecs.Position = [22 19 100 42];
914     app.Viewspecs.Text = 'View specs';
915
916     % Create Createitem
917     app.Createitem = uibutton(app.ModelToolbox, 'push');
918     app.Createitem.ButtonPushedFcn = createCallbackFcn(app, ...

```



```
916     @CreateitemButtonPushed, true);
917     app.Createitem.BackgroundColor = [1 1 1];
918     app.Createitem.Enable = 'off';
919     app.Createitem.Position = [129 19 100 42];
920     app.Createitem.Text = 'Create item';
921
922     % Create Searchitem
923     app.Searchitem = uibutton(app.ModelToolbox, 'push');
924     app.Searchitem.ButtonPushedFcn = createCallbackFcn(app, ...
925         @SearchitemButtonPushed, true);
926     app.Searchitem.BackgroundColor = [1 1 1];
927     app.Searchitem.Enable = 'off';
928     app.Searchitem.Position = [236 19 100 42];
929     app.Searchitem.Text = 'Search item';
930
931     % Create Removeitem
932     app.Removeitem = uibutton(app.ModelToolbox, 'push');
933     app.Removeitem.ButtonPushedFcn = createCallbackFcn(app, ...
934         @RemoveitemButtonPushed, true);
935     app.Removeitem.BackgroundColor = [1 1 1];
936     app.Removeitem.Enable = 'off';
937     app.Removeitem.Position = [343 19 100 42];
938     app.Removeitem.Text = 'Remove item';
939
940     % Create ExportModel
941     app.ExportModel = uibutton(app.ModelToolbox, 'push');
942     app.ExportModel.ButtonPushedFcn = createCallbackFcn(app, ...
943         @ExportModelButtonPushed, true);
944     app.ExportModel.BackgroundColor = [0.2 0.702 0.502];
945     app.ExportModel.FontColor = [1 1 1];
946     app.ExportModel.Position = [693 19 165 42];
947     app.ExportModel.Text = 'Export model to Excel';
948
949     % Create MWPanel
950     app.MWPanel = uipanel(app.ModelTab);
951     app.MWPanel.BorderType = 'none';
952     app.MWPanel.Title = 'Momentum wheels';
953     app.MWPanel.BackgroundColor = [1 1 1];
954     app.MWPanel.Position = [455 13 207 132];
955
956     % Create AdditemMW
957     app.AdditemMW = uibutton(app.MWPanel, 'push');
958     app.AdditemMW.ButtonPushedFcn = createCallbackFcn(app, ...
959         @AdditemMWButtonPushed, true);
960     app.AdditemMW.BackgroundColor = [0.2 0.702 0.502];
961     app.AdditemMW.FontColor = [1 1 1];
962     app.AdditemMW.Position = [68 22 74 21];
963     app.AdditemMW.Text = 'Select';
964
965     % Create LabelMW
966     app.LabelMW = uilabel(app.MWPanel);
967     app.LabelMW.HorizontalAlignment = 'center';
968     app.LabelMW.Position = [17 71 175 15];
969     app.LabelMW.Text = 'No items yet';
970
971     % Create OSPanel
```

```

967     app.OSPanel = uipanel(app.ModelTab);
968     app.OSPanel.BorderType = 'none';
969     app.OSPanel.Title = 'Other sensors';
970     app.OSPanel.BackgroundColor = [1 1 1];
971     app.OSPanel.Position = [676 154 207 132];
972
973     % Create AdditemOS
974     app.AdditemOS = uibutton(app.OSPanel, 'push');
975     app.AdditemOS.ButtonPushedFcn = createCallbackFcn(app, ...
976         @AdditemOSButtonPushed, true);
977     app.AdditemOS.BackgroundColor = [0.2 0.702 0.502];
978     app.AdditemOS.FontColor = [1 1 1];
979     app.AdditemOS.Position = [68 22 74 21];
980     app.AdditemOS.Text = 'Select';
981
982     % Create LabelOS
983     app.LabelOS = uilabel(app.OSPanel);
984     app.LabelOS.HorizontalAlignment = 'center';
985     app.LabelOS.Position = [17 71 175 15];
986     app.LabelOS.Text = 'No items yet';
987
988     % Create DOPanel
989     app.DOPanel = uipanel(app.ModelTab);
990     app.DOPanel.BorderType = 'none';
991     app.DOPanel.Title = 'Docks';
992     app.DOPanel.BackgroundColor = [1 1 1];
993     app.DOPanel.Position = [12 154 207 132];
994
995     % Create AdditemDO
996     app.AdditemDO = uibutton(app.DOPanel, 'push');
997     app.AdditemDO.ButtonPushedFcn = createCallbackFcn(app, ...
998         @AdditemDOButtonPushed, true);
999     app.AdditemDO.BackgroundColor = [0.2 0.702 0.502];
999     app.AdditemDO.FontColor = [1 1 1];
999     app.AdditemDO.Position = [71 22 74 21];
999     app.AdditemDO.Text = 'Select';
1000
1001    % Create LabelDO
1002    app.LabelDO = uilabel(app.DOPanel);
1003    app.LabelDO.HorizontalAlignment = 'center';
1004    app.LabelDO.Position = [20 71 175 15];
1005    app.LabelDO.Text = 'No items yet';
1006
1007    % Create DatabaseTab
1008    app.DatabaseTab = uitab(app.TabGroup);
1009    app.DatabaseTab.Title = 'Database';
1010    app.DatabaseTab.BackgroundColor = [1 1 1];
1011
1012    % Create UITable
1013    app.UITable = uitable(app.DatabaseTab);
1014    app.UITable.ColumnName = {'Col1'};
1015    app.UITable.RowName = {};
1016    app.UITable.ColumnEditable = true;
1017    app.UITable.CellEditCallback = createCallbackFcn(app, ...
1018        @UITableCellEdit, true);
1019    app.UITable.Position = [240 -1 656 432];

```

```
1020
1021     % Create SortbyDropDownLabel
1022     app.SortbyDropDownLabel = uilabel(app.DatabaseTab);
1023     app.SortbyDropDownLabel.HorizontalAlignment = 'right';
1024     app.SortbyDropDownLabel.Position = [591 466 44 15];
1025     app.SortbyDropDownLabel.Text = 'Sort by';
1026
1027     % Create Sortby
1028     app.Sortby = uidropdown(app.DatabaseTab);
1029     app.Sortby.Items = {'Name (AZ)', 'Supplier (AZ)', 'Power ...
1030         consumption [W]', 'Mass [kg]'};
1031     app.Sortby.ItemsData = {'name', 'supplier', 'power', 'mass'};
1032     app.Sortby.ValueChangedFcn = createCallbackFcn(app, ...
1033         @SortbyValueChanged, true);
1034     app.Sortby.Position = [650 462 220 22];
1035     app.Sortby.Value = 'name';
1036
1037     % Create ImsearchingforDropDownLabel
1038     app.ImsearchingforDropDownLabel = uilabel(app.DatabaseTab);
1039     app.ImsearchingforDropDownLabel.Position = [24 466 94 15];
1040     app.ImsearchingforDropDownLabel.Text = 'I''m searching for';
1041
1042     % Create Class
1043     app.Class = uidropdown(app.DatabaseTab);
1044     app.Class.Items = {'an actuator', 'a sensor', 'an onboard computer'};
1045     app.Class.ItemsData = {'actuator', 'sensor', 'dock'};
1046     app.Class.ValueChangedFcn = createCallbackFcn(app, ...
1047         @ClassValueChanged, true);
1048     app.Class.Position = [117 462 115 22];
1049     app.Class.Value = 'actuator';
1050
1051     % Create CheckoutPanel
1052     app.CheckoutPanel = uipanel(app.DatabaseTab);
1053     app.CheckoutPanel.BorderType = 'none';
1054     app.CheckoutPanel.TitlePosition = 'centertop';
1055     app.CheckoutPanel.Title = 'Checkout';
1056     app.CheckoutPanel.BackgroundColor = [1 1 1];
1057     app.CheckoutPanel.FontSize = 10;
1058     app.CheckoutPanel.Position = [1 -1 238 150];
1059
1060     % Create AmountSpinnerLabel
1061     app.AmountSpinnerLabel = uilabel(app.CheckoutPanel);
1062     app.AmountSpinnerLabel.Position = [170 104 47 15];
1063     app.AmountSpinnerLabel.Text = 'Amount';
1064
1065     % Create Amount
1066     app.Amount = uispinner(app.CheckoutPanel);
1067     app.Amount.Limits = [1 100];
1068     app.Amount.RoundFractionalValues = 'on';
1069     app.Amount.ValueDisplayFormat = '%.0f';
1070     app.Amount.Position = [170 79 56 22];
1071     app.Amount.Value = 1;
1072
1073     % Create Additem
1074     app.Additem = uibutton(app.CheckoutPanel, 'push');
1075     app.Additem.ButtonPushedFcn = createCallbackFcn(app, ...
```

```

1073     @AdditemButtonPushed, true);
1074 app.Additem.BackgroundColor = [0.2 0.702 0.502];
1075 app.Additem.FontColor = [1 1 1];
1076 app.Additem.Position = [20 23 207 42];
1077 app.Additem.Text = 'Add item to model';

1078 % Create ItemselectedEditFieldLabel
1079 app.ItemselectedEditFieldLabel = uilabel(app.CheckoutPanel);
1080 app.ItemselectedEditFieldLabel.Position = [19 104 77 15];
1081 app.ItemselectedEditFieldLabel.Text = 'Item selected';

1082 % Create Itemselected
1083 app.Itemselected = uieditfield(app.CheckoutPanel, 'text');
1084 app.Itemselected.Position = [19 79 137 22];

1085 % Create SearchfiltersPanel
1086 app.SearchfiltersPanel = uipanel(app.DatabaseTab);
1087 app.SearchfiltersPanel.BorderType = 'none';
1088 app.SearchfiltersPanel.TitlePosition = 'centertop';
1089 app.SearchfiltersPanel.Title = 'Search filters';
1090 app.SearchfiltersPanel.BackgroundColor = [1 1 1];
1091 app.SearchfiltersPanel.FontSize = 10;
1092 app.SearchfiltersPanel.Position = [1 148 239 283];

1093 % Create Maxmass
1094 app.Maxmass = uieditfield(app.SearchfiltersPanel, 'numeric');
1095 app.Maxmass.ValueDisplayFormat = '%5.2f kg';
1096 app.Maxmass.Enable = 'off';
1097 app.Maxmass.Position = [147 229 77 22];

1098 % Create Maxpower
1099 app.Maxpower = uieditfield(app.SearchfiltersPanel, 'numeric');
1100 app.Maxpower.ValueDisplayFormat = '%5.2f W';
1101 app.Maxpower.Enable = 'off';
1102 app.Maxpower.Position = [147 188 77 22];

1103 % Create Maxtemp
1104 app.Maxtemp = uieditfield(app.SearchfiltersPanel, 'numeric');
1105 app.Maxtemp.ValueDisplayFormat = '%5.0f C';
1106 app.Maxtemp.Enable = 'off';
1107 app.Maxtemp.Position = [147 148 77 22];

1108 % Create Mintemp
1109 app.Mintemp = uieditfield(app.SearchfiltersPanel, 'numeric');
1110 app.Mintemp.ValueDisplayFormat = '%5.0f C';
1111 app.Mintemp.Enable = 'off';
1112 app.Mintemp.Position = [147 108 77 22];

1113 % Create CheckBoxMaxmass
1114 app.CheckBoxMaxmass = uicheckbox(app.SearchfiltersPanel);
1115 app.CheckBoxMaxmass.ValueChangedFcn = createCallbackFcn(app, ...
1116     @CheckBoxMaxmassValueChanged, true);
1117 app.CheckBoxMaxmass.Text = '';
1118 app.CheckBoxMaxmass.Position = [20 233 15 15];

1119 % Create CheckBoxMaxpower

```

```
1127 app.CheckBoxMaxpower = uicheckbox(app.SearchfiltersPanel);
1128 app.CheckBoxMaxpower.ValueChangedFcn = createCallbackFcn(app, ...
1129     @CheckBoxMaxpowerValueChanged, true);
1130 app.CheckBoxMaxpower.Text = '';
1131 app.CheckBoxMaxpower.Position = [20 192 15 15];
1132
1133 % Create CheckBoxMaxtemp
1134 app.CheckBoxMaxtemp = uicheckbox(app.SearchfiltersPanel);
1135 app.CheckBoxMaxtemp.ValueChangedFcn = createCallbackFcn(app, ...
1136     @CheckBoxMaxtempValueChanged, true);
1137 app.CheckBoxMaxtemp.Text = '';
1138 app.CheckBoxMaxtemp.Position = [20 152 15 15];
1139
1140 % Create CheckBoxMintemp
1141 app.CheckBoxMintemp = uicheckbox(app.SearchfiltersPanel);
1142 app.CheckBoxMintemp.ValueChangedFcn = createCallbackFcn(app, ...
1143     @CheckBoxMintempValueChanged, true);
1144 app.CheckBoxMintemp.Text = '';
1145 app.CheckBoxMintemp.Position = [20 112 15 15];
1146
1147 % Create MaxmassLabel
1148 app.MaxmassLabel = uilabel(app.SearchfiltersPanel);
1149 app.MaxmassLabel.Enable = 'off';
1150 app.MaxmassLabel.Position = [45 233 64 15];
1151 app.MaxmassLabel.Text = 'Max. mass';
1152
1153 % Create MaxpowerLabel
1154 app.MaxpowerLabel = uilabel(app.SearchfiltersPanel);
1155 app.MaxpowerLabel.Enable = 'off';
1156 app.MaxpowerLabel.Position = [45 192 68 15];
1157 app.MaxpowerLabel.Text = 'Max. power';
1158
1159 % Create MaxtempLabel
1160 app.MaxtempLabel = uilabel(app.SearchfiltersPanel);
1161 app.MaxtempLabel.Enable = 'off';
1162 app.MaxtempLabel.Position = [45 152 100 15];
1163 app.MaxtempLabel.Text = 'Max. temperature';
1164
1165 % Create MintempLabel
1166 app.MintempLabel = uilabel(app.SearchfiltersPanel);
1167 app.MintempLabel.Enable = 'off';
1168 app.MintempLabel.Position = [45 112 96 15];
1169 app.MintempLabel.Text = 'Min. temperature';
1170
1171 % Create InparticularDropDownLabel
1172 app.InparticularDropDownLabel = uilabel(app.DatabaseTab);
1173 app.InparticularDropDownLabel.Position = [257 466 68 15];
1174 app.InparticularDropDownLabel.Text = 'In particular';
1175
1176 % Create Subclass
1177 app.Subclass = uidropdown(app.DatabaseTab);
1178 app.Subclass.Items = {'a reaction wheel', 'other category'};
1179 app.Subclass.ValueChangedFcn = createCallbackFcn(app, ...
1180     @SubclassValueChanged, true);
1181 app.Subclass.Position = [324 462 140 22];
1182 app.Subclass.Value = 'a reaction wheel';
```

```

1179
1180    % Create CreateitemTab
1181    app.CreateitemTab = uitab(app.TabGroup);
1182    app.CreateitemTab.Title = 'Create item';
1183    app.CreateitemTab.BackgroundColor = [1 1 1];
1184
1185    % Create ModelToolbox_2
1186    app.ModelToolbox_2 = uipanel(app.CreateitemTab);
1187    app.ModelToolbox_2.BorderType = 'none';
1188    app.ModelToolbox_2.BackgroundColor = [0.9412 0.9412 0.9412];
1189    app.ModelToolbox_2.Position = [1 439 895 78];
1190
1191    % Create SaveItemInDatabase
1192    app.SaveItemInDatabase = uibutton(app.ModelToolbox_2, 'push');
1193    app.SaveItemInDatabase.ButtonPushedFcn = createCallbackFcn(app, ...
1194        @SaveItemInDatabaseButtonPushed, true);
1195    app.SaveItemInDatabase.BackgroundColor = [0.2 0.702 0.502];
1196    app.SaveItemInDatabase.FontColor = [1 1 1];
1197    app.SaveItemInDatabase.Position = [693 19 165 42];
1198    app.SaveItemInDatabase.Text = 'Save item in database';
1199
1200    % Create CreateItemDropdown
1201    app.CreateItemDropdown = uidropdown(app.ModelToolbox_2);
1202    app.CreateItemDropdown.Items = {'Sun sensor', 'Star sensor', 'Earth ... ...
1203        sensor', 'Magnetometer', 'IMU', 'GNSS receiver', 'Other ... ...
1204        sensor', 'Reaction wheel', 'Momentum wheel', 'Magnetorquer', ...
1205        'Other actuator', 'Dock'};
1206    app.CreateItemDropdown.ItemsData = {'sun sensor', 'star sensor', ...
1207        'earth sensor', 'magnetometer', 'imu', 'gnss', 'other sensor', ...
1208        'reaction wheel', 'momentum wheel', 'magnetorquer', 'other ...
1209        actuator', 'dock'};
1210    app.CreateItemDropdown.ValueChangedFcn = createCallbackFcn(app, ...
1211        @CreateItemDropdownValueChanged, true);
1212    app.CreateItemDropdown.FontSize = 14;
1213    app.CreateItemDropdown.BackgroundColor = [1 1 1];
1214    app.CreateItemDropdown.Position = [27 27 239 26];
1215    app.CreateItemDropdown.Value = 'sun sensor';
1216
1217    % Create GeneralfeaturesPanel
1218    app.GeneralfeaturesPanel = uipanel(app.CreateitemTab);
1219    app.GeneralfeaturesPanel.BorderType = 'none';
1220    app.GeneralfeaturesPanel.Title = 'General features';
1221    app.GeneralfeaturesPanel.BackgroundColor = [1 1 1];
1222    app.GeneralfeaturesPanel.Position = [27 30 530 384];
1223
1224    % Create MassEditFieldLabel
1225    app.MassEditFieldLabel = uilabel(app.GeneralfeaturesPanel);
1226    app.MassEditFieldLabel.FontSize = 14;
1227    app.MassEditFieldLabel.Position = [1 203 39 18];
1228    app.MassEditFieldLabel.Text = 'Mass';
1229
1230    % Create MassEditField
1231    app.MassEditField = uieditfield(app.GeneralfeaturesPanel, 'numeric');
1232    app.MassEditField.Limits = [0 1000];
1233    app.MassEditField.ValueDisplayFormat = '%5.2f kg';
1234    app.MassEditField.FontSize = 14;

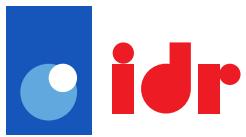
```

```
1227 app.MassEditField.Position = [72 202 116 22];  
1228  
1229 % Create PowerEditFieldLabel  
1230 app.PowerEditFieldLabel = uilabel(app.GeneralfeaturesPanel);  
1231 app.PowerEditFieldLabel.FontSize = 14;  
1232 app.PowerEditFieldLabel.Position = [1 164 45 18];  
1233 app.PowerEditFieldLabel.Text = 'Power';  
1234  
1235 % Create PowerEditField  
1236 app.PowerEditField = uieditfield(app.GeneralfeaturesPanel, 'numeric');  
1237 app.PowerEditField.Limits = [0 1000000];  
1238 app.PowerEditField.ValueDisplayFormat = '%5.2f W';  
1239 app.PowerEditField.FontSize = 14;  
1240 app.PowerEditField.Position = [72 163 116 22];  
1241  
1242 % Create PriceEditFieldLabel  
1243 app.PriceEditFieldLabel = uilabel(app.GeneralfeaturesPanel);  
1244 app.PriceEditFieldLabel.FontSize = 14;  
1245 app.PriceEditFieldLabel.Position = [1 125 37 18];  
1246 app.PriceEditFieldLabel.Text = 'Price';  
1247  
1248 % Create PriceEditField  
1249 app.PriceEditField = uieditfield(app.GeneralfeaturesPanel, 'text');  
1250 app.PriceEditField.FontSize = 14;  
1251 app.PriceEditField.Position = [72 124 116 22];  
1252  
1253 % Create TmaxEditFieldLabel  
1254 app.TmaxEditFieldLabel = uilabel(app.GeneralfeaturesPanel);  
1255 app.TmaxEditFieldLabel.FontSize = 14;  
1256 app.TmaxEditFieldLabel.Position = [1 87 44 18];  
1257 app.TmaxEditFieldLabel.Text = 'T max';  
1258  
1259 % Create TmaxEditField  
1260 app.TmaxEditField = uieditfield(app.GeneralfeaturesPanel, 'numeric');  
1261 app.TmaxEditField.Limits = [-270 1000];  
1262 app.TmaxEditField.ValueDisplayFormat = '%5.1f C';  
1263 app.TmaxEditField.FontSize = 14;  
1264 app.TmaxEditField.Position = [72 86 116 22];  
1265  
1266 % Create TminEditFieldLabel  
1267 app.TminEditFieldLabel = uilabel(app.GeneralfeaturesPanel);  
1268 app.TminEditFieldLabel.FontSize = 14;  
1269 app.TminEditFieldLabel.Position = [1 49 40 18];  
1270 app.TminEditFieldLabel.Text = 'T min';  
1271  
1272 % Create TminEditField  
1273 app.TminEditField = uieditfield(app.GeneralfeaturesPanel, 'numeric');  
1274 app.TminEditField.Limits = [-270 1000];  
1275 app.TminEditField.ValueDisplayFormat = '%5.1f C';  
1276 app.TminEditField.FontSize = 14;  
1277 app.TminEditField.Position = [72 48 116 22];  
1278  
1279 % Create NameEditFieldLabel  
1280 app.NameEditFieldLabel = uilabel(app.GeneralfeaturesPanel);  
1281 app.NameEditFieldLabel.FontSize = 14;  
1282 app.NameEditFieldLabel.Position = [1 321 62 18];
```

```

1283 app.NameEditFieldLabel.Text = 'Name';
1284
1285 % Create NameEditField
1286 app.NameEditField = uieditfield(app.GeneralfeaturesPanel, 'text');
1287 app.NameEditField.FontSize = 14;
1288 app.NameEditField.Position = [71 317 328 27];
1289
1290 % Create SupplierEditFieldLabel
1291 app.SupplierEditFieldLabel = uilabel(app.GeneralfeaturesPanel);
1292 app.SupplierEditFieldLabel.FontSize = 14;
1293 app.SupplierEditFieldLabel.Position = [1 283 62 18];
1294 app.SupplierEditFieldLabel.Text = 'Supplier';
1295
1296 % Create SupplierEditField
1297 app.SupplierEditField = uieditfield(app.GeneralfeaturesPanel, 'text');
1298 app.SupplierEditField.FontSize = 14;
1299 app.SupplierEditField.Position = [71 279 328 27];
1300
1301 % Create URLEditFieldLabel
1302 app.URLEditFieldLabel = uilabel(app.GeneralfeaturesPanel);
1303 app.URLEditFieldLabel.FontSize = 14;
1304 app.URLEditFieldLabel.Position = [1 246 62 18];
1305 app.URLEditFieldLabel.Text = 'URL';
1306
1307 % Create URLEditField
1308 app.URLEditField = uieditfield(app.GeneralfeaturesPanel, 'text');
1309 app.URLEditField.FontSize = 14;
1310 app.URLEditField.Position = [71 242 328 27];
1311
1312 % Create WidthEditFieldLabel
1313 app.WidthEditFieldLabel = uilabel(app.GeneralfeaturesPanel);
1314 app.WidthEditFieldLabel.FontSize = 14;
1315 app.WidthEditFieldLabel.Position = [220 203 41 18];
1316 app.WidthEditFieldLabel.Text = 'Width';
1317
1318 % Create WidthEditField
1319 app.WidthEditField = uieditfield(app.GeneralfeaturesPanel, 'numeric');
1320 app.WidthEditField.Limits = [0 Inf];
1321 app.WidthEditField.ValueDisplayFormat = '%5.0f mm';
1322 app.WidthEditField.FontSize = 14;
1323 app.WidthEditField.Position = [285 202 114 22];
1324
1325 % Create LengthEditFieldLabel
1326 app.LengthEditFieldLabel = uilabel(app.GeneralfeaturesPanel);
1327 app.LengthEditFieldLabel.FontSize = 14;
1328 app.LengthEditFieldLabel.Position = [219 164 48 18];
1329 app.LengthEditFieldLabel.Text = 'Length';
1330
1331 % Create LengthEditField
1332 app.LengthEditField = uieditfield(app.GeneralfeaturesPanel, 'numeric');
1333 app.LengthEditField.Limits = [0 Inf];
1334 app.LengthEditField.ValueDisplayFormat = '%5.0f mm';
1335 app.LengthEditField.FontSize = 14;
1336 app.LengthEditField.Position = [285 162 114 22];
1337
1338 % Create HeightEditFieldLabel

```



```
1339     app.HeightEditFieldLabel = uilabel(app.GeneralfeaturesPanel);
1340     app.HeightEditFieldLabel.FontSize = 14;
1341     app.HeightEditFieldLabel.Position = [219 125 46 18];
1342     app.HeightEditFieldLabel.Text = 'Height';
1343
1344     % Create HeightEditField
1345     app.HeightEditField = uiditfield(app.GeneralfeaturesPanel, 'numeric');
1346     app.HeightEditField.Limits = [0 Inf];
1347     app.HeightEditField.ValueDisplayFormat = '%5.0f mm';
1348     app.HeightEditField.FontSize = 14;
1349     app.HeightEditField.Position = [285 123 114 22];
1350
1351     % Create LifespanEditFieldLabel
1352     app.LifespanEditFieldLabel = uilabel(app.GeneralfeaturesPanel);
1353     app.LifespanEditFieldLabel.FontSize = 14;
1354     app.LifespanEditFieldLabel.Position = [1 11 58 18];
1355     app.LifespanEditFieldLabel.Text = 'Lifespan';
1356
1357     % Create LifespanEditField
1358     app.LifespanEditField = uiditfield(app.GeneralfeaturesPanel, 'text');
1359     app.LifespanEditField.FontSize = 14;
1360     app.LifespanEditField.Position = [72 10 116 22];
1361
1362     % Create COTSCheckBox
1363     app.COTSCheckBox = uiccheckbox(app.GeneralfeaturesPanel);
1364     app.COTSCheckBox.Text = 'COTS';
1365     app.COTSCheckBox.FontSize = 14;
1366     app.COTSCheckBox.Position = [433 285 90 18];
1367
1368     % Create CubeSatCheckBox
1369     app.CubeSatCheckBox = uiccheckbox(app.GeneralfeaturesPanel);
1370     app.CubeSatCheckBox.Text = 'CubeSat';
1371     app.CubeSatCheckBox.FontSize = 14;
1372     app.CubeSatCheckBox.Position = [432 246 90 18];
1373
1374     % Create TRLDropDownLabel
1375     app.TRLDropDownLabel = uilabel(app.GeneralfeaturesPanel);
1376     app.TRLDropDownLabel.FontSize = 14;
1377     app.TRLDropDownLabel.Position = [432 320 32 18];
1378     app.TRLDropDownLabel.Text = 'TRL';
1379
1380     % Create TRLDropDown
1381     app.TRLDropDown = uidropdown(app.GeneralfeaturesPanel);
1382     app.TRLDropDown.Items = {'None', '1', '2', '3', '4', '5', '6', '7', ...
1383     '8', '9'};
1384     app.TRLDropDown.Editable = 'on';
1385     app.TRLDropDown.FontSize = 14;
1386     app.TRLDropDown.BackgroundColor = [1 1 1];
1387     app.TRLDropDown.Position = [466 319 65 22];
1388     app.TRLDropDown.Value = 'None';
1389
1390     % Create VnomEditFieldLabel
1391     app.VnomEditFieldLabel = uilabel(app.GeneralfeaturesPanel);
1392     app.VnomEditFieldLabel.FontSize = 14;
1393     app.VnomEditFieldLabel.Position = [220 86 42 18];
1394     app.VnomEditFieldLabel.Text = 'Vnom';
```

```

1394
1395 % Create VnomEditField
1396 app.VnomEditField = uieditfield(app.GeneralfeaturesPanel, 'numeric');
1397 app.VnomEditField.Limits = [0 Inf];
1398 app.VnomEditField.ValueDisplayFormat = '%5.1f V';
1399 app.VnomEditField.FontSize = 14;
1400 app.VnomEditField.Position = [285 85 114 22];
1401
1402 % Create IavgEditFieldLabel
1403 app.IavgEditFieldLabel = uilabel(app.GeneralfeaturesPanel);
1404 app.IavgEditFieldLabel.FontSize = 14;
1405 app.IavgEditFieldLabel.Position = [219 47 32 18];
1406 app.IavgEditFieldLabel.Text = 'Iavg';
1407
1408 % Create IavgEditField
1409 app.IavgEditField = uieditfield(app.GeneralfeaturesPanel, 'numeric');
1410 app.IavgEditField.Limits = [0 Inf];
1411 app.IavgEditField.ValueDisplayFormat = '%5.3f A';
1412 app.IavgEditField.FontSize = 14;
1413 app.IavgEditField.Position = [285 45 114 22];
1414
1415 % Create IpeakEditFieldLabel
1416 app.IpeakEditFieldLabel = uilabel(app.GeneralfeaturesPanel);
1417 app.IpeakEditFieldLabel.FontSize = 14;
1418 app.IpeakEditFieldLabel.Position = [219 8 40 18];
1419 app.IpeakEditFieldLabel.Text = 'Ipeak';
1420
1421 % Create IpeakEditField
1422 app.IpeakEditField = uieditfield(app.GeneralfeaturesPanel, 'numeric');
1423 app.IpeakEditField.Limits = [0 Inf];
1424 app.IpeakEditField.ValueDisplayFormat = '%5.3f A';
1425 app.IpeakEditField.FontSize = 14;
1426 app.IpeakEditField.Position = [285 6 114 22];
1427
1428 % Create SpecificfeauresPanel
1429 app.SpecificfeauresPanel = uipanel(app.CreateitemTab);
1430 app.SpecificfeauresPanel.BorderType = 'none';
1431 app.SpecificfeauresPanel.Title = 'Specific feaures';
1432 app.SpecificfeauresPanel.BackgroundColor = [1 1 1];
1433 app.SpecificfeauresPanel.Position = [587 155 271 259];
1434
1435 % Create Spec1EditField
1436 app.Spec1EditField = uieditfield(app.SpecificfeauresPanel, 'text');
1437 app.Spec1EditField.FontSize = 14;
1438 app.Spec1EditField.Position = [135 194 137 22];
1439
1440 % Create Spec2EditField
1441 app.Spec2EditField = uieditfield(app.SpecificfeauresPanel, 'text');
1442 app.Spec2EditField.FontSize = 14;
1443 app.Spec2EditField.Position = [135 158 137 22];
1444
1445 % Create Spec3EditField
1446 app.Spec3EditField = uieditfield(app.SpecificfeauresPanel, 'text');
1447 app.Spec3EditField.FontSize = 14;
1448 app.Spec3EditField.Position = [135 123 137 22];
1449

```

```
1450 % Create Spec4EditField
1451 app.Spec4EditField = uieditfield(app.SpecificfeauresPanel, 'text');
1452 app.Spec4EditField.FontSize = 14;
1453 app.Spec4EditField.Position = [135 88 137 22];
1454
1455 % Create Spec5EditField
1456 app.Spec5EditField = uieditfield(app.SpecificfeauresPanel, 'text');
1457 app.Spec5EditField.FontSize = 14;
1458 app.Spec5EditField.Position = [135 53 137 22];
1459
1460 % Create Spec6EditField
1461 app.Spec6EditField = uieditfield(app.SpecificfeauresPanel, 'text');
1462 app.Spec6EditField.FontSize = 14;
1463 app.Spec6EditField.Position = [135 18 137 22];
1464
1465 % Create Spec1EditFieldLabel
1466 app.Spec1EditFieldLabel = uilabel(app.SpecificfeauresPanel);
1467 app.Spec1EditFieldLabel.FontSize = 14;
1468 app.Spec1EditFieldLabel.Position = [1 196 135 18];
1469 app.Spec1EditFieldLabel.Text = 'Spec 1';
1470
1471 % Create Spec2EditFieldLabel
1472 app.Spec2EditFieldLabel = uilabel(app.SpecificfeauresPanel);
1473 app.Spec2EditFieldLabel.FontSize = 14;
1474 app.Spec2EditFieldLabel.Position = [1 160 135 18];
1475 app.Spec2EditFieldLabel.Text = 'Spec 2';
1476
1477 % Create Spec3EditFieldLabel
1478 app.Spec3EditFieldLabel = uilabel(app.SpecificfeauresPanel);
1479 app.Spec3EditFieldLabel.FontSize = 14;
1480 app.Spec3EditFieldLabel.Position = [1 124 135 18];
1481 app.Spec3EditFieldLabel.Text = 'Spec 3';
1482
1483 % Create Spec4EditFieldLabel
1484 app.Spec4EditFieldLabel = uilabel(app.SpecificfeauresPanel);
1485 app.Spec4EditFieldLabel.FontSize = 14;
1486 app.Spec4EditFieldLabel.Position = [1 90 135 18];
1487 app.Spec4EditFieldLabel.Text = 'Spec 4';
1488
1489 % Create Spec5EditFieldLabel
1490 app.Spec5EditFieldLabel = uilabel(app.SpecificfeauresPanel);
1491 app.Spec5EditFieldLabel.FontSize = 14;
1492 app.Spec5EditFieldLabel.Position = [1 55 135 18];
1493 app.Spec5EditFieldLabel.Text = 'Spec 5';
1494
1495 % Create Spec6EditFieldLabel
1496 app.Spec6EditFieldLabel = uilabel(app.SpecificfeauresPanel);
1497 app.Spec6EditFieldLabel.FontSize = 14;
1498 app.Spec6EditFieldLabel.Position = [1 20 135 18];
1499 app.Spec6EditFieldLabel.Text = 'Spec 6';
1500
1501 % Create NotesTextAreaLabel
1502 app.NotesTextAreaLabel = uilabel(app.CreateitemTab);
1503 app.NotesTextAreaLabel.FontSize = 14;
1504 app.NotesTextAreaLabel.Position = [458 133 42 18];
1505 app.NotesTextAreaLabel.Text = 'Notes';
```

```

1506
1507 % Create NotesTextArea
1508 app.NotesTextArea = uitextarea(app.CreateitemTab);
1509 app.NotesTextArea.FontSize = 14;
1510 app.NotesTextArea.Position = [515 35 343 118];
1511
1512 % Create SettingsTab
1513 app.SettingsTab = uitab(app.TabGroup);
1514 app.SettingsTab.Title = 'Settings';
1515 app.SettingsTab.BackgroundColor = [1 1 1];
1516 app.SettingsTab.ForegroundColor = [0.149 0.149 0.149];
1517
1518 % Create FieldstodisplayonsearchresultsPanel
1519 app.FieldstodisplayonsearchresultsPanel = uipanel(app.SettingsTab);
1520 app.FieldstodisplayonsearchresultsPanel.BorderType = 'none';
1521 app.FieldstodisplayonsearchresultsPanel.Title = 'Fields to display ...  
on search results';
1522 app.FieldstodisplayonsearchresultsPanel.BackgroundColor = [1 1 1];
1523 app.FieldstodisplayonsearchresultsPanel.Position = [21 244 247 178];
1524
1525 % Create Panel
1526 app.Panel = uipanel(app.FieldstodisplayonsearchresultsPanel);
1527 app.Panel.BorderType = 'none';
1528 app.Panel.BackgroundColor = [1 1 1];
1529 app.Panel.Position = [1 1 247 153];
1530
1531 % Create NameCheckBox
1532 app.NameCheckBox = uicheckbox(app.Panel);
1533 app.NameCheckBox.Enable = 'off';
1534 app.NameCheckBox.Text = 'Name';
1535 app.NameCheckBox.Position = [8 134 80 15];
1536 app.NameCheckBox.Value = true;
1537
1538 % Create SupplierCheckBox
1539 app.SupplierCheckBox = uicheckbox(app.Panel);
1540 app.SupplierCheckBox.Enable = 'off';
1541 app.SupplierCheckBox.Text = 'Supplier';
1542 app.SupplierCheckBox.Position = [8 111 80 15];
1543 app.SupplierCheckBox.Value = true;
1544
1545 % Create MassCheckBox
1546 app.MassCheckBox = uicheckbox(app.Panel);
1547 app.MassCheckBox.Text = 'Mass';
1548 app.MassCheckBox.Position = [8 88 80 15];
1549 app.MassCheckBox.Value = true;
1550
1551 % Create PowerCheckBox
1552 app.PowerCheckBox = uicheckbox(app.Panel);
1553 app.PowerCheckBox.Text = 'Power';
1554 app.PowerCheckBox.Position = [8 65 80 15];
1555 app.PowerCheckBox.Value = true;
1556
1557 % Create CubeSatCheckBox_3
1558 app.CubeSatCheckBox_3 = uicheckbox(app.Panel);
1559 app.CubeSatCheckBox_3.Text = 'CubeSat';
1560 app.CubeSatCheckBox_3.Position = [8 42 80 15];

```

```

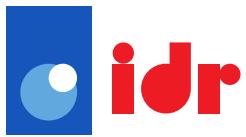
1561
1562     % Create COTSCheckBox_3
1563     app.COTSCheckBox_3 = uicheckbox(app.Panel);
1564     app.COTSCheckBox_3.Text = 'COTS';
1565     app.COTSCheckBox_3.Position = [8 19 80 15];
1566
1567     % Create DimensionsCheckBox
1568     app.DimensionsCheckBox = uicheckbox(app.Panel);
1569     app.DimensionsCheckBox.Text = 'Dimensions';
1570     app.DimensionsCheckBox.Position = [122 134 85 15];
1571     app.DimensionsCheckBox.Value = true;
1572
1573     % Create ThermalfeaturesCheckBox
1574     app.ThermalfeaturesCheckBox = uicheckbox(app.Panel);
1575     app.ThermalfeaturesCheckBox.Text = 'Thermal features';
1576     app.ThermalfeaturesCheckBox.Position = [122 111 113 15];
1577     app.ThermalfeaturesCheckBox.Value = true;
1578
1579     % Create ElectricalfeaturesCheckBox
1580     app.ElectricalfeaturesCheckBox = uicheckbox(app.Panel);
1581     app.ElectricalfeaturesCheckBox.Text = 'Electrical features';
1582     app.ElectricalfeaturesCheckBox.Position = [122 88 117 15];
1583
1584     % Create SpecificfeaturesCheckBox
1585     app.SpecificfeaturesCheckBox = uicheckbox(app.Panel);
1586     app.SpecificfeaturesCheckBox.Text = 'Specific features';
1587     app.SpecificfeaturesCheckBox.Position = [122 65 111 15];
1588     app.SpecificfeaturesCheckBox.Value = true;
1589
1590     % Create LifespanCheckBox
1591     app.LifespanCheckBox = uicheckbox(app.Panel);
1592     app.LifespanCheckBox.Text = 'Lifespan';
1593     app.LifespanCheckBox.Position = [122 42 80 15];
1594
1595     % Create PriceCheckBox
1596     app.PriceCheckBox = uicheckbox(app.Panel);
1597     app.PriceCheckBox.Text = 'Price';
1598     app.PriceCheckBox.Position = [122 19 80 15];
1599
1600     % Create ExcelsettingsPanel
1601     app.ExcelsettingsPanel = uipanel(app.SettingsTab);
1602     app.ExcelsettingsPanel.BorderType = 'none';
1603     app.ExcelsettingsPanel.Title = 'Excel settings';
1604     app.ExcelsettingsPanel.BackgroundColor = [1 1 1];
1605     app.ExcelsettingsPanel.Position = [294 244 349 178];
1606
1607     % Create DatabasesheetnameEditFieldLabel
1608     app.DatabasesheetnameEditFieldLabel = uilabel(app.ExcelsettingsPanel);
1609     app.DatabasesheetnameEditFieldLabel.VerticalAlignment = 'center';
1610     app.DatabasesheetnameEditFieldLabel.Position = [7 92 120 15];
1611     app.DatabasesheetnameEditFieldLabel.Text = 'Database sheetname';
1612
1613     % Create DatabasesheetnameEditField
1614     app.DatabasesheetnameEditField = ...
           uieditfield(app.ExcelsettingsPanel, 'text');
1615     app.DatabasesheetnameEditField.Position = [142 88 208 22];

```

```

1616
1617 % Create ModelsheetnameEditFieldLabel
1618 app.ModelsheetnameEditFieldLabel = uilabel(app.ExcelsettingsPanel);
1619 app.ModelsheetnameEditFieldLabel.VerticalAlignment = 'center';
1620 app.ModelsheetnameEditFieldLabel.Position = [7 55 101 15];
1621 app.ModelsheetnameEditFieldLabel.Text = 'Model sheetname';
1622
1623 % Create ModelsheetnameEditField
1624 app.ModelsheetnameEditField = uieditfield(app.ExcelsettingsPanel, ...
1625     'text');
1626 app.ModelsheetnameEditField.Position = [142 51 208 22];
1627
1628 % Create FilenameEditFieldLabel
1629 app.FilenameEditFieldLabel = uilabel(app.ExcelsettingsPanel);
1630 app.FilenameEditFieldLabel.VerticalAlignment = 'center';
1631 app.FilenameEditFieldLabel.Position = [7 129 55 15];
1632 app.FilenameEditFieldLabel.Text = 'Filename';
1633
1634 % Create FilenameEditField
1635 app.FilenameEditField = uieditfield(app.ExcelsettingsPanel, 'text');
1636 app.FilenameEditField.Position = [142 125 208 22];
1637
1638 % Create ModelToolbox_3
1639 app.ModelToolbox_3 = uipanel(app.SettingsTab);
1640 app.ModelToolbox_3.BorderType = 'none';
1641 app.ModelToolbox_3.BackgroundColor = [0.9412 0.9412 0.9412];
1642 app.ModelToolbox_3.Position = [1 439 895 78];
1643
1644 % Create SaveItemInDatabase_2
1645 app.SaveItemInDatabase_2 = uibutton(app.ModelToolbox_3, 'push');
1646 app.SaveItemInDatabase_2.BackgroundColor = [0.2 0.702 0.502];
1647 app.SaveItemInDatabase_2.FontColor = [1 1 1];
1648 app.SaveItemInDatabase_2.Position = [693 19 165 42];
1649 app.SaveItemInDatabase_2.Text = 'Save changes';
1650
1651 % Create SaveItemInDatabase_3
1652 app.SaveItemInDatabase_3 = uibutton(app.ModelToolbox_3, 'push');
1653 app.SaveItemInDatabase_3.BackgroundColor = [1 1 1];
1654 app.SaveItemInDatabase_3.Position = [21 19 119 42];
1655 app.SaveItemInDatabase_3.Text = 'Set default';
1656
1657 % Create TotalmassEditFieldLabel
1658 app.TotalmassEditFieldLabel = uilabel(app.UI);
1659 app.TotalmassEditFieldLabel.BackgroundColor = [0 0.102 0.4];
1660 app.TotalmassEditFieldLabel.HorizontalAlignment = 'right';
1661 app.TotalmassEditFieldLabel.FontColor = [1 1 1];
1662 app.TotalmassEditFieldLabel.Position = [744 5 63 15];
1663 app.TotalmassEditFieldLabel.Text = 'Total mass';
1664
1665 % Create Totalmass
1666 app.Totalmass = uieditfield(app.UI, 'numeric');
1667 app.Totalmass.Limits = [0 Inf];
1668 app.Totalmass.ValueDisplayFormat = '%5.2f kg';
1669 app.Totalmass.Editable = 'off';
1670 app.Totalmass.FontColor = [1 1 1];

```



```
1671     app.Totalmass.Position = [817 1 80 22];  
1672  
1673     % Create TotalpowerEditFieldLabel  
1674     app.TotalpowerEditFieldLabel = uilabel(app.UI);  
1675     app.TotalpowerEditFieldLabel.BackgroundColor = [0 0.102 0.4];  
1676     app.TotalpowerEditFieldLabel.VerticalAlignment = 'center';  
1677     app.TotalpowerEditFieldLabel.FontColor = [1 1 1];  
1678     app.TotalpowerEditFieldLabel.Position = [575 5 78 15];  
1679     app.TotalpowerEditFieldLabel.Text = 'Total power';  
1680  
1681     % Create Totalpower  
1682     app.Totalpower = uieditfield(app.UI, 'numeric');  
1683     app.Totalpower.ValueDisplayFormat = '%5.2f W';  
1684     app.Totalpower.Editable = 'off';  
1685     app.Totalpower.FontColor = [1 1 1];  
1686     app.Totalpower.BackgroundColor = [0 0.102 0.4];  
1687     app.Totalpower.Position = [652 1 68 22];  
1688 end  
1689 end  
1690  
1691 methods (Access = public)  
1692  
1693     % Construct app  
1694     function app = appl  
1695  
1696         % Create and configure components  
1697         createComponents(app)  
1698  
1699         % Register the app with App Designer  
1700         registerApp(app, app.UI)  
1701  
1702         % Execute the startup function  
1703         runStartupFcn(app, @startupFcn)  
1704  
1705         if nargout == 0  
1706             clear app  
1707         end  
1708     end  
1709  
1710         % Code that executes before app deletion  
1711         function delete(app)  
1712  
1713             % Delete UIFigure when app is deleted  
1714             delete(app.UI)  
1715         end  
1716     end  
1717 end
```