

ALMA MATER STUDIORUM-UNIVERSITA' DI BOLOGNA

SECONDA FACOLTA' DI INGEGNERIA
CON SEDE A CESENA

CORSO DI LAUREA
IN INGEGNERIA AEROSPAZIALE
Classe L-9

Sede di Forlì

ELABORATO FINALE DI LAUREA
IN FONDAMENTI D'ASTRONAUTICA

**SVILUPPO DI UN'INTERFACCIA GRAFICA PER LA STIMA
DEL POWER BUDGET DEL MICROSATELLITE
ALMASat-EO**

CANDIDATO

Federico Di Nicola

RELATORE

Prof. Paolo Tortora

CORRELATORI

Ing. Alessandro Tambini

Ing. Alberto Graziani

Anno Accademico 2011/2012
Sessione II

*Alla mia famiglia
..e a tutti quelli che hanno creduto in me,
sin dall'inizio.*

Ringraziamenti

Ringrazio in primis papà, mamma e mio fratello per avermi sostenuto in ogni momento, e avermi dato la possibilità di intraprendere questa splendida carriera universitaria che mi ha dato molte soddisfazioni; il mio relatore, il prof. Paolo Tortora, per la serietà nel suo lavoro; i correlatori, Ing. Alessandro Tambini e Ing. Alberto Graziani, per la serietà e dedizione che hanno mostrato sin dall'inizio per quest'attività, la pazienza nei miei riguardi e per avermi sostenuto sempre, anche nei momenti più difficili affrontati nella tesi e gli altri collaboratori di ALMA Space s.r.l. che mi hanno ben accolto nel laboratorio. Ringrazio i miei amici: il dott. Pietro Rauzi per le serate passate insieme in particolare "quella famosa sera", e non solo; il dott. Giovanni Padovani per i bei momenti trascorsi in cucina e poi sui libri, soprattutto nell'ultimo periodo, frenetico e interminabile di studio; Angelo, per i magnifici trascorsi in casa Ronco; la mia ragazza Sarah per essermi stata vicina, sempre, senza di lei non avrei mai sostenuto il carico di studio e di lavoro dietro la tesi, le coinquiline Vero(&Seneb), Rebe e Crì per avermi accolto con tanto calore in casa, la Nostra casa; la prof.ssa Graziani che mi ha sostenuto sin dai tempi del liceo; i miei zii Nuccio, Tina, i miei cugini Sebastiano, Andrea e Alessandro, che sono stati presenti in ogni occasione; un ringraziamento particolare a mia nonna e all'affetto che mi ha sempre dato, e a mio nonno Sebastiano, che porto nel cuore; infine i miei amici Pierfrancesco, Antonio, Nicola, Enrico, Luca, Alessandro, Simone, Daniele, Vincenzo, Sebastiano, Berna, Gabriele, Vito, Tony che mi hanno accompagnato sempre, nei momenti belli e meno belli..e tutti i compagni di corso con cui ho condiviso interessi, opinioni ma soprattutto appunti.

Sommario

1. Introduzione.....	1-13
2. Simulazione della missione ALMASat-EO.....	2-15
2.1. Caratteristiche orbitali	2-16
2.2. Orbita operativa.....	2-19
2.3. Sottosistemi del satellite.....	2-21
2.3.1. ALMA Space MultiSpectral Camera.....	2-22
2.3.2. High Rate Data Transmitter	2-22
2.3.3. Attitude and Orbit Determination and Control Systems.....	2-23
2.3.4. On Board Data Handling	2-26
2.3.5. Power Board	2-26
2.3.6. Tracking Telemetry and Control unit	2-26
3. Calcolo del power budget.....	3-29
3.1. Equazioni e modelli per il calcolo della potenza consumata	3-29
3.2. Equazioni e modelli per il calcolo della potenza prodotta	3-32
3.3. Calcolo del margine tra potenza.....	3-36
4. Sviluppo del software per il calcolo del power budget	4-39
4.1. Programma GUIDE.....	4-40
4.1.1. TabPanel Constructor	4-40
4.1.2. Elementi essenziali dell'interfaccia utente	4-41
4.1.3. Funzioni principali e loro utilizzo.....	4-43
4.2. Starting mission point	4-44
4.3. Interfaccia grafica.....	4-45
4.3.1. Mission Panel.....	4-46
4.3.2. Result panel.....	4-47
4.3.3. Interfaccia grafica dei sottosistemi	4-47
4.4. Push Buttons.....	4-51
5. Scenari di missione.....	5-53
5.1. Scenario Photo:	5-53
5.2. Scenario Download:	5-54
5.3. Scenario Photo&Download:.....	5-55
5.4. Scenario Detumbling:.....	5-57
5.5. Scenario Nominal:.....	5-58

5.6.	Scenario Custom:	5-59
5.7.	Stima dei Power Budget.....	5-59
6.	Validazione del software per il power budget.....	6-61
6.1.	Scenario Detumbling.....	6-63
6.2.	Scenario Payload.....	6-64
7.	Conclusioni e Sviluppi futuri.....	7-67
A.1	Programma per il calcolo della potenza prodotta.....	7-69
A.2	Programma per il calcolo della potenza consumata.....	7-73
A.3	Funzione esterna tabella dati e salvataggio	7-81
A.4	Funzione Default Values	7-83
A.5	Funzioni esterne per il plot delle potenze	7-85
A.5.1	Potenza Prodotta (PP).....	7-85
A.5.2	Potenza Consumata (PC).....	7-85
A.5.3	Power Budget (PB).....	7-85
A.6	Funzione Reset	7-87
A.7	Funzione Select e setting checkbox	7-89
	Bibliografia	7-93

Indice figure

Figura 1 - (a) rappresentazione CAD di ALMASat-EO, (b) ALMASat-1	2-15
Figura 2 - Parametri orbitali.....	2-16
Figura 3 – Rappresentazione tridimensionale della RAAN	2-17
Figura 4 - Rappresentazione dell'ellisse.....	2-17
Figura 5 - Rappresentazione delle orbite chiuse al variare dell'eccentricità.....	2-18
Figura 6 - Traiettorie orbitali a diverse eccentricità.....	2-19
Figura 7 - Rappresentazione Zenit – Nadir.....	2-20
Figura 8 - Rappresentazione grafica del satellite	2-20
Figura 9 - Onda quadra TTC.....	3-30
Figura 10 - Andamento potenza consumata da GPS e TTC accesi.....	3-31
Figura 11 - Andamento della potenza consumata durante l'orbita nominale	3-31
Figura 12 - Modello matematico del diodo di Shockley.....	3-32
Figura 13 - Circuito equivalente per la modellazione della cella solare	3-32
Figura 14 - Rappresentazione dell'angolo d'incidenza solare sul pannello.....	3-34
Figura 15 - Andamento Tensione-Corrente della cella variando l'angolo d'incidenza del sole ...	3-35
Figura 16 - Andamento potenza prodotta dai pannelli solari	3-36
Figura 17 - Andamento della potenza prodotta e potenza consumata.....	3-37
Figura 18 – La GUI del PB software	4-39
Figura 19 – Pannello principale del programma GUIDE.....	4-40
Figura 20 - Tab Panel Constructor, TPC.....	4-41
Figura 21 - Oggetti del programma GUIDE	4-41
Figura 22 - Property Inspector	4-42
Figura 23 - Selezione dello starting mission point.....	4-44
Figura 24 - Interfaccia Grafica.....	4-45
Figura 25 - Mission Panel.....	4-46
Figura 26 - Result Panel.....	4-47
Figura 27 - AMSC panel.....	4-48
Figura 28 - HSTX Panel	4-48
Figura 29 - OBDH Panel	4-49
Figura 30 - PB Panel.....	4-49
Figura 31 - TTC panel	4-50
Figura 32 - AODCS Panel	4-50
Figura 33 - Tasto Select.....	4-51
Figura 34 - Bottoni principali	4-51
Figura 35 - Tabella dei risultati.....	4-52
Figura 36 - Bottoni delle potenze.....	4-52
Figura 37 - Scenario di missione Photos.....	5-54
Figura 38 - Scenario di missione Download.....	5-55
Figura 39 - Scenario di missione Photos& Download	5-56

Figura 40 - Caso critico di simulazione	5-56
Figura 41 - Scenario di missione Detumbling	5-57
Figura 42 - Scenario di missione Nominal.....	5-58
Figura 43 - Schema a blocchi del banco di prova	6-61
Figura 44 - Interfaccia di controllo dei test.....	6-62
Figura 45 – Risultato dei test nel caso detumbling	6-64
Figura 46 – Risultati dei test nel caso payload.....	6-65

Indice tabelle

Tabella 1 - Tipi di orbite in funzione dell'eccentricità	2-18
Tabella 2 - Parametri orbitali operativi	2-19
Tabella 3 - Parametri operativi AMSC	2-22
Tabella 4 - Parametri operativi HSTX, trasferimento a 1 Mbit/s	2-22
Tabella 5 - Parametri operativi HSTX, trasferimento a 2 Mbit/s	2-23
Tabella 6 - Parametri operativi Earth Sensor	2-23
Tabella 7 - Parametri operativi Sun Sensor	2-24
Tabella 8 - Parametri operativi GPS	2-24
Tabella 9 - Parametri operativi Magnetometers	2-24
Tabella 10 - Parametri operativi Magnetic Coils	2-25
Tabella 11 - Parametri operativi, Ruota di reazione	2-25
Tabella 12 - Parametri operativi, Ruota di momento	2-25
Tabella 13 - Parametri operativi OBDH	2-26
Tabella 14 - Parametri operativi PB	2-26
Tabella 15 - Parametri operativi TT&C in ricezione	2-27
Tabella 16 - Parametri operativi TT&C in trasmissione	2-27
Tabella 17 - Parametri operativi TT&C in trasmissione sopra la stazione di terra	2-27
Tabella 18 - Dati caratteristici della cella solare	3-33
Tabella 19 - Valori caratteristici della cella solare	3-33
Tabella 20 - Oggetti in GUIDE e loro caratteristiche	4-42
Tabella 21 - Risultati di energia per i vari casi di scenario	5-59

1. Introduzione

Lo studio del power budget è di fondamentale importanza per la progettazione di una missione spaziale, nella quale bisogna assicurare che il margine tra potenza prodotta dai pannelli solari e potenza consumata dai sottosistemi di bordo, sia positivo. È stata dunque sviluppata un'interfaccia grafica (Graphical User Interface, GUI) in ambiente Matlab® (1) per analizzare il power budget in diversi scenari di missione. Fissando come obiettivo una missione per l'osservazione della terra, sono stati predefiniti una serie di scenari tipici per questo tipo di attività. D'altra parte è stata anche implementata una modalità di utilizzo della GUI tale da consentire la definizione di una missione personalizzata per altre missioni.

Prima di affrontare il progetto della GUI è stata fatta un'analisi di missione per studiare il funzionamento di ogni sottosistema, modellarlo e progettare le rispettive interfacce grafiche. In particolare nel Capitolo 2 sono riportati i parametri orbitali della missione e sono dettagliati i vari sottosistemi.

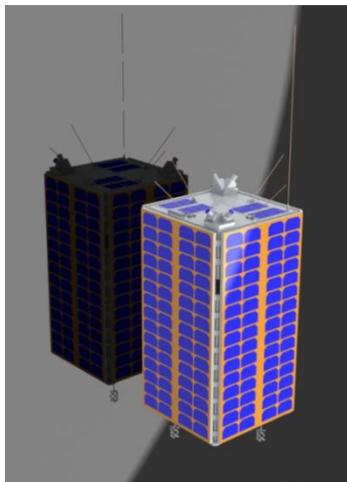
Nel Capitolo 3 sono descritte le equazioni per il calcolo della potenza prodotta dai pannelli solari, le equazioni che descrivono la potenza consumata da ogni componente acceso e il calcolo del power budget. La GUI, dettagliata nel Capitolo 4, consente la definizione della missione e riporta gli andamenti della potenza prodotta, consumata e calcola quindi il margine tra i due andamenti.

Il Capitolo 5 descrive una serie di scenari operativi, preimpostati, per l'osservazione della terra, nonché la modalità "custom" per la definizione di un profilo di missione personalizzato differente dai precedenti.

Questi dati saranno necessari per la simulazione dei test Hardware-In-the-Loop (HIL), descritti nel Capitolo 6, in cui sarà verificato il power budget in due diversi scenari operativi: il detumbling, ovvero la prima fase della missione dopo il rilascio dal lanciatore e l'attività di payload ottico, considerato operativo per acquisire immagini della Terra.

2. Simulazione della missione ALMASat-EO

ALMASat-EO (ALma MAter Satellite for Earth Observation) è il secondo microsatellite della famiglia ALMASat che nasce dalla collaborazione tra ALMASpace s.r.l. e il laboratorio di Microsatelliti e Microsistemi spaziali della Seconda Facoltà di Ingegneria dell'Università di Bologna. La Figura 1 (a) mostra il satellite ALMASat-EO che è stato sviluppato per l'osservazione della terra (Earth Observation - EO) e il suo predecessore ALMASat-1, Figura 1 (b).



(a)



(b)

Figura 1 - (a) rappresentazione CAD di ALMASat-EO, (b) ALMASat-1

L'attività di osservazione della Terra sarà garantita grazie all'utilizzo di un payload ottico capace di acquisire immagini multispettrali in una porzione di Terra di 1000 Km circa e successivamente archiviate in una memoria di bordo e inviate con un downlink alla stazione di terra, Alma Mater Ground Station (AMGS) situata presso lo stesso laboratorio (2) (3). Le immagini sono inviate a terra con un trasmettitore ad alta velocità in banda S (2.45 GHz), High Speed Transmitter (HSTX) che permette un downlink con due diverse velocità: 1 Mbit/s o 2Mbit/s. A bordo del microsatellite, oltre al payload ottico, saranno presenti anche altri sottosistemi come ad esempio le ruote di reazione e il GPS che si dovranno accendere prima di acquisire le immagini. In particolare, le ruote serviranno per acquisire l'assetto e stabilizzare il satellite durante l'acquisizione, mentre il GPS sarà utilizzato per la determinazione orbitale e la georeferenziazione delle immagini. Gli altri sistemi di bordo che saranno accesi costantemente durante la missione sono: sensore di terra, sensori di sole, ruota di

momento, magnetometri, bobine magnetiche, sistema di telemetria (Tracking, Telemetry and Control - TT&C), scheda di potenza (Power Board - PB) e il computer di bordo (On Board Data Handling - OBDH).

2.1. Caratteristiche orbitali

Gli elementi che definiscono la posizione e il moto di un oggetto in orbita attorno ad altro corpo si definiscono parametri orbitali rappresentati in Figura 2. Due di questi definiscono l'orientazione del piano orbitale e sono: l'inclinazione i e l'ascensione retta del nodo ascendente Ω .

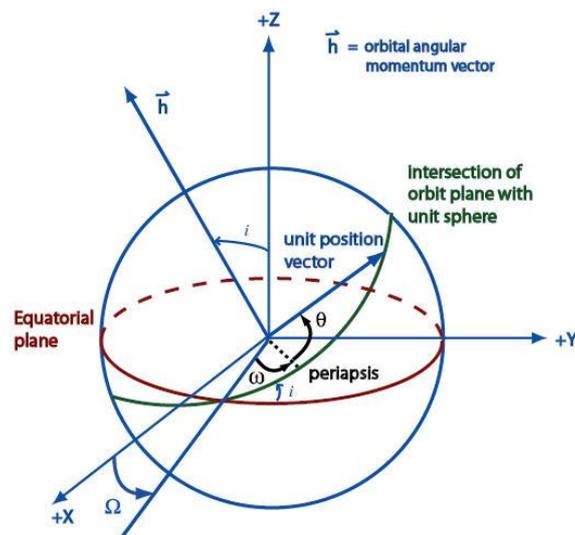


Figura 2 - Parametri orbitali

L'intersezione tra il piano orbitale con quello equatoriale descrive una retta denominata linea dei nodi.

La Figura 3 mostra l'angolo Ω tra l'asse x e la linea dei nodi, denominato appunto ascensione retta del nodo ascendente (Right Ascension of the Ascending Node, RAAN). D'altra parte l'angolo d'inclinazione del piano orbitale relativamente al piano dell'equatore terrestre identifica il parametro d'inclinazione i .

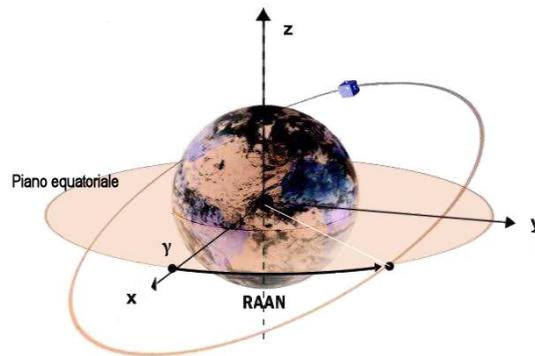


Figura 3 – Rappresentazione tridimensionale della RAAN

Un altro parametro è il semiasse maggiore a , riportata in Figura 4, definito come la metà dell'asse più lungo che contiene i due fuochi di cui, per satelliti in orbita terrestre, uno è occupato dalla Terra.

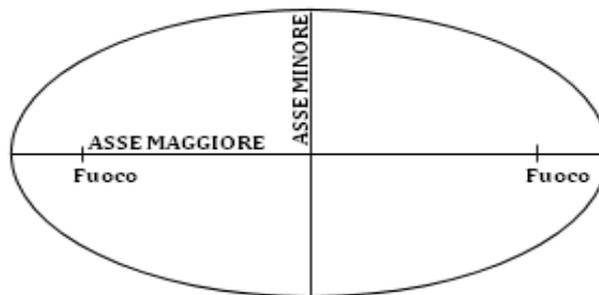


Figura 4 - Rappresentazione dell'ellisse

Vi è poi il parametro che determina la forma dell'orbita, denominato eccentricità ed è indicato con il simbolo e . La Tabella 1 elenca i tipi di orbite in funzione del valore di eccentricità.

$e=0$	Orbita circolare
$0 < e < 1$	Orbita ellittica
$e=1$	Orbita parabolica
$e > 1$	Orbita iperbolica

Tabella 1 - Tipi di orbite in funzione dell'eccentricità

Per le orbite chiuse, in particolare quelle circolari ed ellittiche, più l'eccentricità si avvicina al valore massimo, cioè l'unità, più l'orbita sarà allungata come mostra la Figura 5.

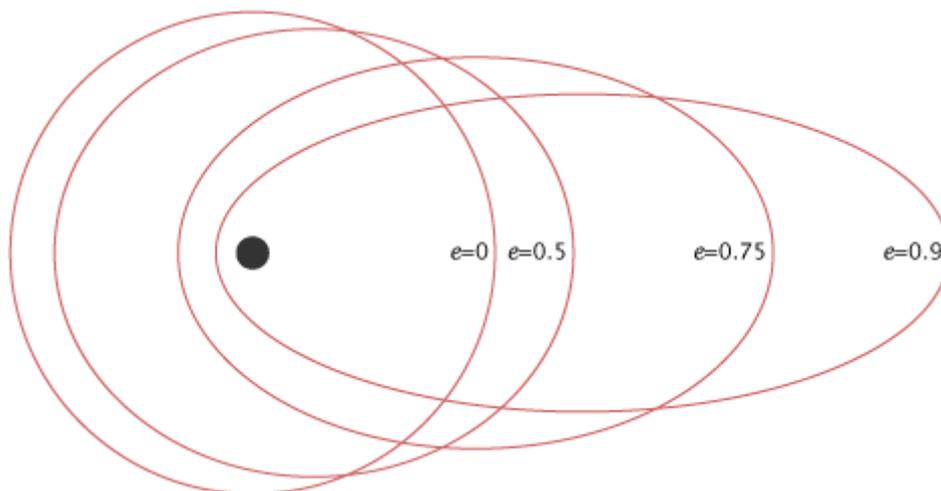


Figura 5 - Rappresentazione delle orbite chiuse al variare dell'eccentricità

Quando l'eccentricità invece raggiunge il valore 1, l'orbita diventa parabolica. Mentre continuando ad aumentare il parametro e , la traiettoria del satellite passa da parabolica a iperbolica (Figura 6).

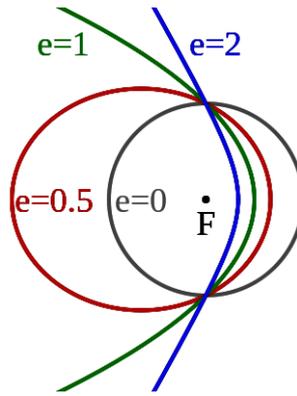


Figura 6 - Traiettorie orbitali a diverse eccentricità

Inoltre vi sono l'argomento del pericentro, ω , ossia l'angolo tra il nodo ascendente e il vettore eccentricità, misurato nella direzione del moto. Il vettore eccentricità ha come direzione la linea degli apsi (pericentro-apocentro) e come verso uscente dal pericentro. Infine, l'anomalia vera, θ , che descrive l'angolo tra il pericentro e il corpo orbitante, misurato sul piano orbitale.

2.2. Orbita operativa

ALMASat-EO è stato progettato per operare in orbita bassa (Low Earth Orbit – LEO), eliosincrona, con inclinazione di 98° circa e semiasse maggiore di 6978 Km.

In Tabella 2 sono elencati i principali parametri orbitali dell'orbita operativa.

Period	T	5908[s]
Altitude	h	686[km]
Major semi-axis	a	6378[km]
Inclination	i	98.161[deg]
RAAN	Ω	336.5[deg]
Eccentricity	e	0.075

Tabella 2 - Parametri orbitali operativi

Una caratteristica importante da tenere in considerazione è l'assetto del satellite durante l'acquisizione delle immagini è la definizione del punto di zenit, cioè il punto in cui la verticale locale dell'osservatore interseca la sfera celeste mentre il punto opposto è chiamato nadir, Figura 7.

Di conseguenza, l'assetto del satellite sarà tale da puntare il payload ottico sulla terra al suo nadir come rappresentato in Figura 8. Quest'assetto si può ottenere con la combinazione di attuatori (magnetometri e ruota di momento) previsti nell'architettura del satellite che consentiranno di mantenere un assetto cosiddetto stabilizzato su tre assi.

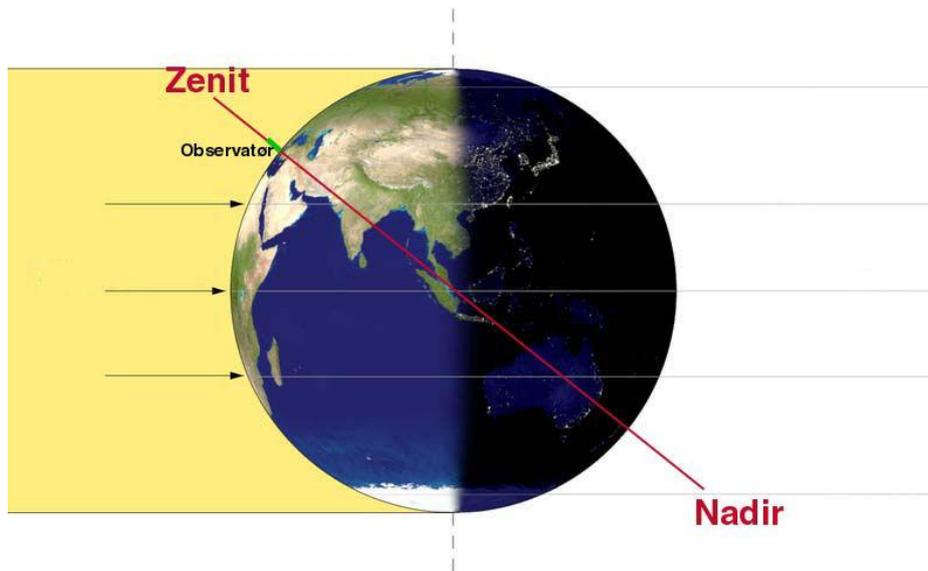


Figura 7 - Rappresentazione Zenit – Nadir

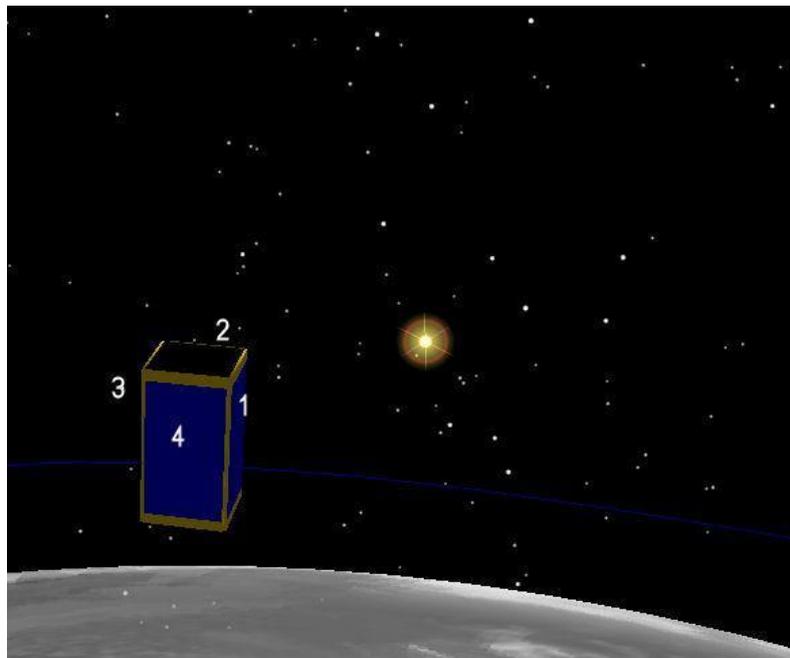


Figura 8 - Rappresentazione grafica del satellite

Inoltre, per questo tipo di missione, è necessario che l'osservazione del territorio sottostante al satellite sia effettuata sotto condizioni ottimali d'illuminazione per avere una qualità d'immagine elevata.

Si è quindi preferita un'orbita elio-sincrona la quale mantiene immutato l'angolo tra la direzione di provenienza dei raggi solari e la traccia a terra dell'orbita. Questo permette al satellite di incontrare sempre le stesse condizioni d'illuminazione del territorio sorvolato, agevolando anche l'opera di sovrapposizione e confronto tra immagini acquisite in periodi diversi. Esistono due diversi tipi di orbita eliosincrona che differiscono per il valore di Ω . In particolare, per $\Omega=90^\circ$ si ottiene l'orbita cosiddetta *mezzogiorno-mezzanotte* mentre per $\Omega=0^\circ$ si ha l'orbita cosiddetta *alba-tramonto*.

L'orbita alba-tramonto si riferisce ad una particolare orbita dove il satellite è sempre illuminato, dato che il satellite vola in un piano orbitale perpendicolare ai raggi solari. D'altra parte, nell'orbita mezzogiorno-mezzanotte, il satellite viaggia lungo la linea del terminatore ossia l'insieme dei punti sulla Terra dove i raggi che compongono la radiazione solare sono tangenti alla superficie.

Per soddisfare i requisiti di illuminazione richiesti dal payload ottico l'orbita di ALMASat-EO sarà molto vicina all'orbita mezzogiorno-mezzanotte: a circa 10.30-10.30. In questo modo si mantiene una buona illuminazione della scena, minimizzando comunque le ombre degli obiettivi a terra ed evitando possibili riflessi o saturazioni del payload a seguito della troppa illuminazione quando il Sole è nel mezzogiorno locale.

2.3. Sottosistemi del satellite

Per un corretto esame del power budget si deve tenere conto anche delle modalità operative in termini di consumi di ogni sottosistema presente a bordo durante la missione e modellarli matematicamente. Per ogni sottosistema è quindi importante individuarne il consumo continuo e gli eventuali picchi di assorbimento derivanti da particolari caratteristiche di funzionamento, oppure la definizione di un duty cycle, nel caso siano utilizzati ad intervalli regolari.

Nei successivi paragrafi sono elencati i sottosistemi del satellite e le loro specifiche d'interesse per il sistema di potenza. In particolare, sono stati definiti

una serie di parametri che sono stati presi in esame: *tempo operativo* inteso come la durata di accensione del sottosistema, *potenza consumata* istantanea e *funzionamento* cioè la caratteristica di accensione, ON se il sistema è sempre acceso durante la missione, ON-OFF se il sistema rimane acceso solo nel tempo operativo per cui è stato progettato.

2.3.1. ALMASpace MultiSpectral Camera

La ALMASpace MultiSpectral Camera (AMSC) rappresenta il payload ottico del satellite. Tale sistema, sviluppato da ALMASpace srl, sarà in grado di realizzare foto utilizzando in filtro con banda compresa tra 400 e 720 nanometri con banda passante 25 nm per ogni scatto. L'immagine coprirà un'area $80 \times 80 \text{ km}^2$ per una porzione di terra di circa 1000 km. Considerando la velocità del satellite in orbita LEO di circa 8 km/s, sono stati considerati i parametri riportati in Tabella 3.

Tempo operativo	125 s
Potenza richiesta	5 W
Funzionamento	ON-OFF

Tabella 3 - Parametri operativi AMSC

2.3.2. High Rate Data Transmitter

HSTX è il trasmettitore dati ad alta velocità. Sarà il sistema in banda S (2.45 GHz) ad alta velocità per inviare le immagini acquisite da AMSC alla stazione di terra. Una volta acquisite le immagini, queste saranno archiviate in una memoria di capacità 1Gbit e inviate a terra tramite HSTX quando il satellite sorvolerà la stazione di terra. Come già accennato le velocità di trasmissione del sistema sono di 1Mbit/s oppure 2 Mbit/s e i loro parametri operativi considerati sono riportati rispettivamente in Tabella 4 e Tabella 5.

Tempo operativo	1000 s.
Potenza richiesta	18 W
Funzionamento	ON-OFF

Tabella 4 - Parametri operativi HSTX, trasferimento a 1 Mbit/s

Tempo operativo	500 s.
Potenza richiesta	20 W
Funzionamento	ON-OFF

Tabella 5 - Parametri operativi HSTX, trasferimento a 2 Mbit/s

2.3.3. Attitude and Orbit Determination and Control Systems

Il controllo della posizione orbitale del satellite e il suo assetto sono garantiti da un unico sottosistema: l'Attitude and Orbit Determination Control System (AODCS) che integra una serie di sensori e attuatori, in particolare: earth sensor, sun sensor, Global Positioning System (GPS), magnetometers, magnetic coils, momentum e reaction wheels.

2.3.3.1. Earth Sensor

Il sensore di terra si basa su rilevatori ad infrarossi che vanno a rilevare la differenza di temperatura tra lo spazio profondo che si trova a circa 4K ed il calore dell'atmosfera terrestre. Così facendo, s'individuano due corde che tagliano la circonferenza della Terra, dai quali si calcola il centro della Terra. Per questo sottosistema sono stati considerati i parametri riportati in Tabella 6.

Tempo operativo	Intera orbita
Potenza richiesta	2 W
Funzionamento	ON

Tabella 6 - Parametri operativi Earth Sensor

2.3.3.2. Sun Sensor

Il sensore di sole è costituito da due Charge-Coupled Device (CCD) che emettono una corrente quando illuminati da radiazione solare, dal valore delle correnti emesse si può determinare l'angolo d'incidenza sul materiale. Le principali caratteristiche operative sono riportate in Tabella 7.

Tempo operativo	Intera orbita
Potenza richiesta	0.5 W
Funzionamento	ON

Tabella 7 - Parametri operativi Sun Sensor

2.3.3.3. Global Positioning System

Il GPS permette al satellite di ricostruire la propria posizione orbitale con i dati ricevuti dalla costellazione di satelliti GPS. Il dispositivo viene acceso 30 minuti prima dell'acquisizione delle immagini, e rimane acceso anche durante il tempo operativo dell'AMSC per georeferenziare le immagini acquisite.

I parametri operativi selezionati per il ricevitore GPS sono riportati in Tabella 8.

Tempo operativo	1925 s
Potenza richiesta	10 W
Funzionamento	ON-OFF

Tabella 8 - Parametri operativi GPS

2.3.3.4. Magnetometers

I magnetometri misurano il campo magnetico sfruttando l'induzione elettromagnetica in una bobina. Le caratteristiche utilizzate sono riportate in Tabella 9.

Tempo operativo	Intera orbita
Potenza richiesta	4 W
Funzionamento	ON

Tabella 9 - Parametri operativi Magnetometers

2.3.3.5. Magnetic Coils

Le bobine magnetiche sono avvolgimenti di conduttore che creano un dipolo magnetico interno che tende ad allinearsi con il campo magnetico terrestre facendo ruotare il satellite. I parametri selezionati sono riportati in Tabella 10.

Tempo operativo	Intera orbita
Potenza richiesta	Variabile nel tempo
Funzionamento	ON

Tabella 10 - Parametri operativi Magnetic Coils

2.3.3.6. Momentum and Reaction Wheels

Le ruote di reazione e di momento, sono utilizzate per smorzare i piccoli disturbi e mantenere il satellite in assetto. La ruota di momento è posta in rotazione ad alta velocità intorno al suo asse al fine di garantire la stabilità giroscopica al satellite e ridurre gli errori d'assetto dovuti alle perturbazioni. Le ruote di reazione saranno operate in momenti specifici dell'orbita, come ad esempio prima della fase di acquisizione delle immagini, per rimettere velocemente il satellite in assetto. Dal punto di vista dei consumi di potenza, la ruota di momento è sempre accesa, mentre le ruote di reazione saranno operate quando necessario.

I parametri selezionati per la ruota di reazione sono riportati in Tabella 11, mentre quelli della ruota di momento sono riportati in Tabella 12.

Tempo operativo	200 s
Potenza richiesta	10 W
Funzionamento	ON-OFF

Tabella 11 - Parametri operativi, Ruota di reazione

Tempo operativo	Intera orbita
Potenza richiesta	10 W
Funzionamento	ON / ON-OFF

Tabella 12 - Parametri operativi, Ruota di momento

2.3.4. On Board Data Handling

L'OBDH è il computer di bordo del satellite. Esso permette di acquisire e gestire tutti i dati dei sottosistemi, compresi valori di tensione, corrente, temperatura, velocità di rotazione, posizione angolare, inoltre riceve e decodifica i comandi e li trasmette ai sottosistemi. I parametri selezionati per questo sistema sono riportati in Tabella 13.

Tempo operativo	Intera orbita
Potenza richiesta	5 W
Funzionamento	ON

Tabella 13 - Parametri operativi OBDH

2.3.5. Power Board

Il sistema di potenza, PB, ha lo scopo di fornire l'alimentazione elettrica a tutti i sistemi di bordo. Tramite la Power Management Unit (PMU) monitorizza lo stato dei vari dispositivi integrati al suo interno e comunica i dati all'OBDH. I parametri considerati per la scheda sono riportati in Tabella 14.

Tempo operativo	Intera orbita
Potenza richiesta	2 W
Funzionamento	ON

Tabella 14 - Parametri operativi PB

2.3.6. Tracking Telemetry and Control unit

La TT&C unit, sarà il sistema in banda UHF (433 MHz) per trasmettere informazioni sullo stato del satellite, sulla posizione orbitale e ricevere i segnali di comando che poi saranno inviati all'OBDH per essere indirizzati a tutti i sottosistemi. Il sistema TT&C prevede una modalità in ricezione sempre operativa e due diverse modalità di trasmissione in funzione che il satellite sia o meno in vista della stazione di terra. In quest'ultimo caso, infatti, il sistema invierà a terra i dati con una frequenza maggiore in maniera da comunicare più informazioni all'operatore di stazione per permettergli di portare a termine eventuali azioni correttive e inviare nuovi comandi.

Le caratteristiche operative per il funzionamento in ricezione sono riportate in Tabella 15, mentre le due funzionalità di trasmissione sono riportate in Tabella 16 e Tabella 17, nel caso generale e in quello di sorvolo sulla stazione di terra, rispettivamente.

Tempo operativo	Intera orbita
Potenza richiesta	1 W
Funzionamento	ON

Tabella 15 - Parametri operativi TT&C in ricezione

Tempo operativo	5s acceso ogni 125s
Potenza richiesta	7 W
Funzionamento	ON-OFF

Tabella 16 - Parametri operativi TT&C in trasmissione

Tempo operativo	5s acceso ogni 15s
Potenza richiesta	7 W
Funzionamento	ON-OFF

Tabella 17 - Parametri operativi TT&C in trasmissione sopra la stazione di terra

3. Calcolo del power budget

In questo capitolo saranno presentati i modelli per il calcolo della potenza consumata e della potenza prodotta. Successivamente sarà introdotta l'equazione per il calcolo del power budget come margine delle due potenze.

3.1. Equazioni e modelli per il calcolo della potenza consumata

La potenza totale consumata (P_{cons}), è ottenuta sommando le potenze richieste (P_i) di ogni i -esimo sottosistema funzionante:

$$P_{cons} = \sum_i P_i \quad (3.1)$$

Per i sottosistemi di funzionamento ON, sono stati considerati i valori massimi di consumo (peak) di ogni sottosistema, in modo tale da rimanere conservativi. I consumi saranno costanti durante la missione poiché il tempo operativo di questi sottosistemi è l'intera orbita. La formula(3.1) diventa quindi:

$$P_{cons} = \text{peak}_{Earth} + \text{peak}_{Magn} + \text{peak}_{Sun} + \text{rms}_{OBDH} + \text{peak}_{TTC} + \text{peak}_{PB} + \text{peak}_{MW} \quad (3.2)$$

Dove,

peak_{Earth} , consumo del sensore di terra;

peak_{Magn} , consumo del Magnetometro;

peak_{Sun} , consumo del sensore di sole;

rms_{OBDH} , valore medio di consumo (rms) dell'OBDH;

peak_{TTC} , consumo del TT&C in ricezione;

peak_{PB} , consumo del PB;

peak_{MW} , consumo della ruota di momento.

Per le bobine magnetiche, che prevedono un consumo variabile durante la missione, i valori di potenza consumata utilizzati nelle simulazioni e nel calcolo della potenza consumata sono stati ottenuti da un file esterno (4).

Per gli altri sottosistemi di funzionamento ON-OFF, è stato necessario introdurre nel codice di programmazione un ciclo FOR, per generare le onde quadre dei consumi di ogni sottosistema acceso. In generale il codice si presenta con la seguente struttura:

```
>> Cons_Subsystem = zeros (1,length(mission));  
>>   for i=(t_on_Subsystem):(t_off_Subsystem)  
>>       Cons_Subsystem (:,i) = peak_Subsystem;  
>>   end
```

La funzione *ZEROS*, restituisce un vettore nullo dei consumi (*Cons_Subsystem*) di lunghezza pari alla durata della missione ed è stato introdotto per inizializzare i vettori. Il ciclo FOR associa il picco di consumo dall'istante di accensione del sottosistema (*t_on_Subsystem*) fino all'istante di spegnimento (*t_off_Subsystem*).

La Figura 9 mostra un esempio di onda quadra nel caso del TT&C acceso.

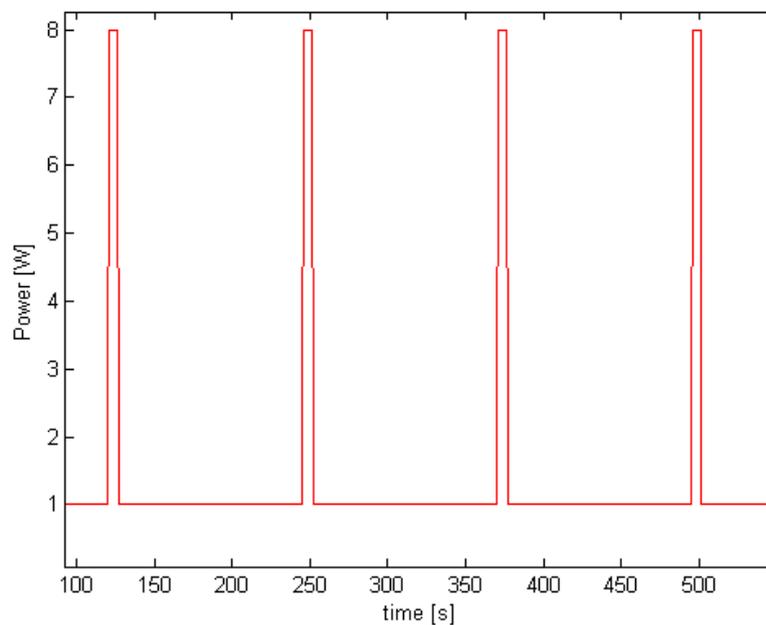


Figura 9 - Onda quadra TTC

In Figura 10 si può vedere l'andamento della potenza consumata considerando GPS e TT&C accesi. Essa è formata dalla sommatoria di:

- valore costante di consumo del TT&C in ricezione di 1W
- onda quadra del consumo GPS di 10W
- onda quadra del consumo del TT&C in trasmissione di 7W

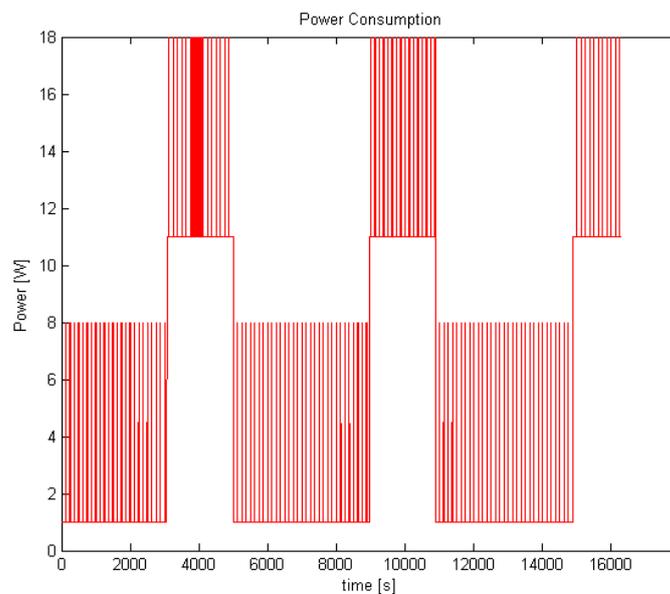


Figura 10 - Andamento potenza consumata da GPS e TTC accesi

L'ultimo esempio riportato in Figura 11, considera i sottosistemi accesi in caso di orbita nominale.

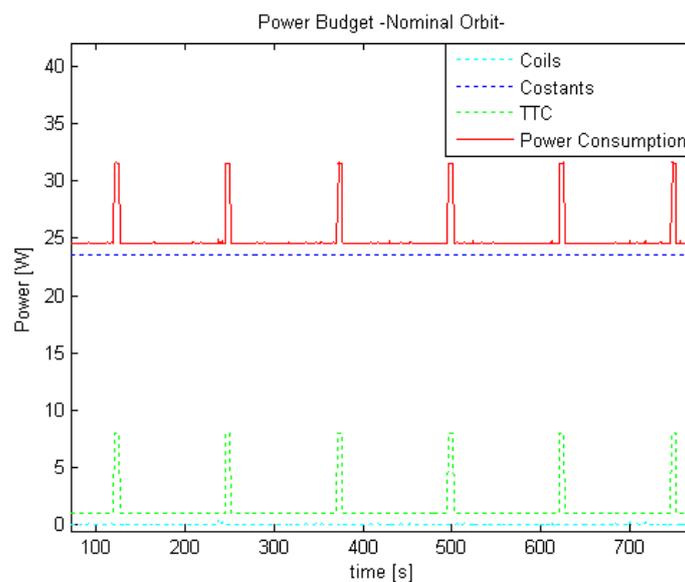


Figura 11 - Andamento della potenza consumata durante l'orbita nominale

In verde è riportato l'andamento del consumo del TT&C, in giallo i consumi costanti considerati nell'equazione (3.2) e in azzurro i consumi delle bobine magnetiche variabili nel tempo. La curva rossa rappresenta la potenza consumata, cioè la somma dei contributi dei sottosistemi accesi in accordo con l'equazione (3.1).

3.2. Equazioni e modelli per il calcolo della potenza prodotta

Per la missione ALMASat-EO, la potenza è prodotta dai quattro pannelli solari montati ai lati del satellite. Come si può vedere dalla Figura 8, data la sua forma prismatica, si avranno al massimo due pannelli adiacenti contemporaneamente illuminati e due in ombra.

Un pannello solare è un vero e proprio generatore di corrente che converte l'energia luminosa dei raggi solari in elettricità. Matematicamente per generare la caratteristica della cella solare è stato usato il modello ideale del diodo di Shockley (5) rappresentato in Figura 12 e il circuito equivalente in Figura 13.

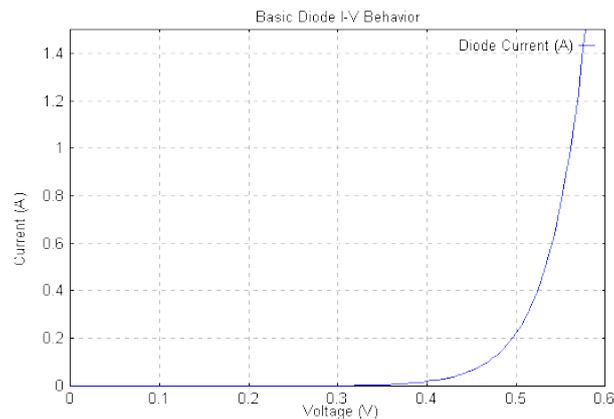


Figura 12 - Modello matematico del diodo di Shockley

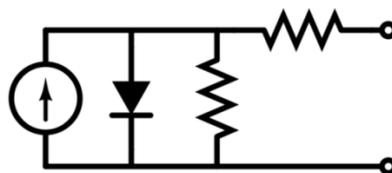


Figura 13 - Circuito equivalente per la modellazione della cella solare

La potenza generata è influenzata da due variabili di cui è necessario tener conto: l'incidenza dei raggi solari sui pannelli e la temperatura d'esercizio del pannello durante la missione. Oltre a queste variabili sono da considerare i dati caratteristici delle celle solari che costituiscono il pannello (Tabella 18), forniti dal costruttore in condizioni operative standard (Standard Operative Conditions – SOC), cioè a temperatura di 25°C e incidenza massima tra raggi solari e pannello. Queste caratteristiche sono:

I_{sc}	corrente di cortocircuito
V_{oc}	tensione di circuito aperto
I_{mp}	corrente di massima potenza
V_{mp}	tensione di massima potenza
$\frac{dI}{dT}, \frac{dV}{dT}$	derivate di corrente e tensione, funzioni della temperatura

Tabella 18 - Dati caratteristici della cella solare

La Tabella 19 riporta i valori caratteristici della cella solare, ottenuti dal datasheet del fornitore (6).

Isc	450.4	[mA]
Voc	2578	[mV]
Imp	433.1	[mA]
Vmp	2241	[mA]
$\frac{dI}{dT}$	0.23679	[mA/°C]
$\frac{dV}{dT}$	-6	[mV/°C]

Tabella 19 - Valori caratteristici della cella solare

Si procede dunque alla determinazione degli andamenti di tensione e corrente, tenendo conto delle variabili di angolo d'incidenza e temperatura, durante la missione.

In un primo momento s'introduce la variazione con l'angolo d'incidenza θ che rappresenta l'angolo tra la normale alla cella e la direzione del raggio solare (Figura 14).

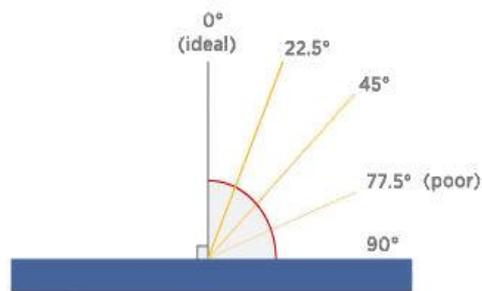


Figura 14 - Raffigurazione dell'angolo d'incidenza solare sul pannello

θ vale 0° se il raggio è coincidente alla normale

θ vale 90° se il raggio è perpendicolare alla normale

Le equazioni (3.3) e (3.4) completano lo studio in esame per gli andamenti di corrente di cortocircuito I_{sc} e corrente di massima potenza I_{mp} , al variare dell'angolo d'incidenza, Figura 15.

$$I_{sc_ang} = I_{sc} * \cos \theta \quad (3.3)$$

$$I_{mp_ang} = I_{mp} * \cos \theta \quad (3.4)$$

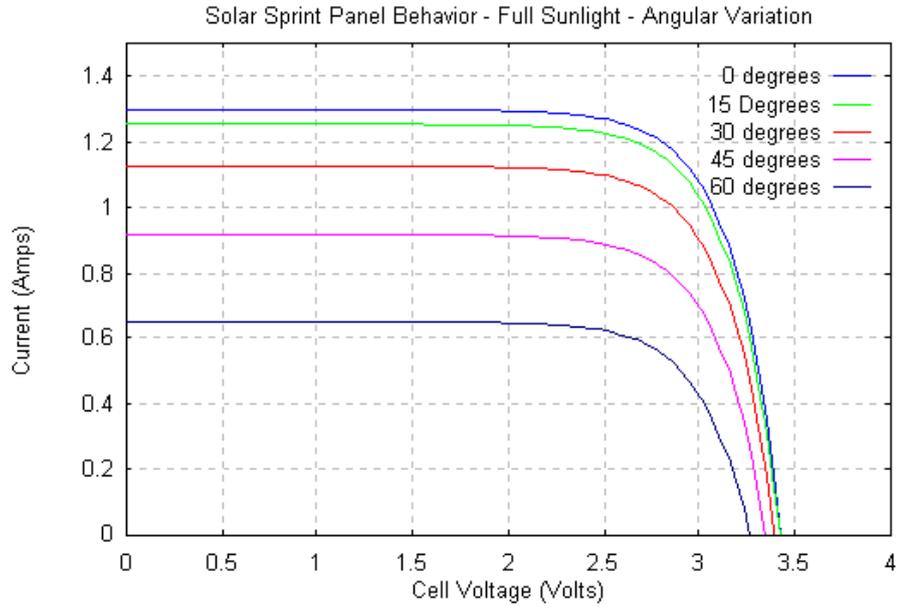


Figura 15 - Andamento Tensione-Corrente della cella variando l'angolo d'incidenza del sole

In un secondo momento s'introducono le equazioni considerando la variabile di temperatura (T):

$$I_{scT} = I_{sc_ang} + \frac{dI_{sc}}{dT} * (\Delta T) \quad (3.5)$$

$$V_{ocT} = V_{oc} + \frac{dV_{oc}}{dT} * (\Delta T) \quad (3.6)$$

$$I_{mpT} = I_{mp_ang} + \frac{dI_{sc}}{dT} * (\Delta T) \quad (3.7)$$

$$V_{mpT} = V_{mp} + \frac{dV_{oc}}{dT} * (\Delta T) \quad (3.8)$$

Dove:

$\Delta T = (T - T_{ref})$ e T_{ref} temperatura di riferimento di 298.151K (25°C).

Infine si procede con il calcolo della potenza totale prodotta, moltiplicando la corrente con il numero di stringhe ($n_{stringhe}$) in parallelo per la tensione con il numero di celle (n_{celle}) in serie:

$$P = V_T * n_{celle} * [I_{angT} * n_{stringhe}] \quad (3.9)$$

La potenza totale calcolata con l'espressione (3.9) è la somma dei contributi dei singoli pannelli. L'andamento è mostrato in Figura 16, in ascissa è riportata la durata della simulazione in secondi e in ordinata i valori di potenza generata dai pannelli espressa in Watt.

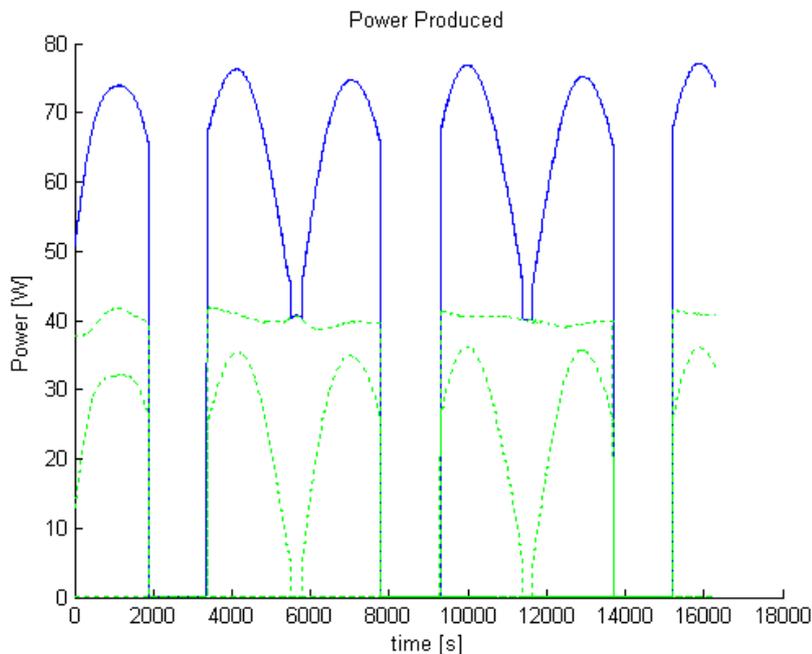


Figura 16 - Andamento potenza prodotta dai pannelli solari

3.3. Calcolo del margine tra potenza

Ricavati gli andamenti di potenza consumata, paragrafo 3.1 e potenza prodotta, 3.2, si procede con lo studio del power budget o margine di potenza, definito come la differenza tra la potenza prodotta e la potenza consumata nell'arco dell'intera orbita. In Figura 17 è riportato un esempio del power budget nel caso di un satellite della famiglia ALMASat in una serie di orbite LEO solo considerando il funzionamento nominale del satellite. Il grafico mostra in blu l'andamento della potenza prodotta e in rosso quella consumata.

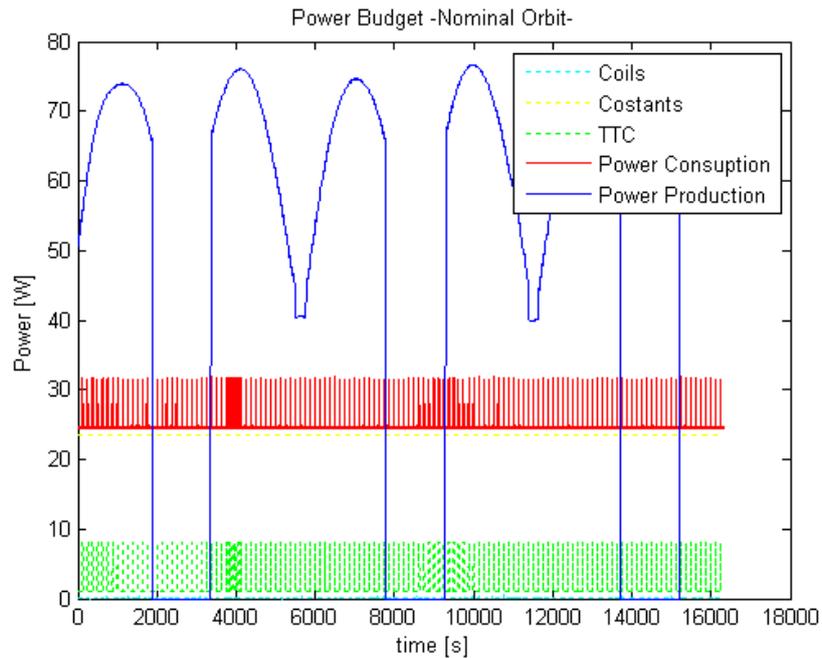


Figura 17 - Andamento della potenza prodotta e potenza consumata

L'integrale delle potenze nel tempo, equazione (3.10) che in altre parole è l'area sottesa dalla curva blu e rossa, restituisce i valori rispettivamente di energia prodotta ed energia consumata.

$$E = \int_t P dt \quad (3.10)$$

In termini di codice, la funzione *TRAPZ* calcola il risultato dell'integrale della potenza nel tempo, quindi l'energia:

```
>> Energy=trapz(Power);
```

Il margine (E_{margin}) è dato dalla differenza di energia prodotta (E_{prod}) ed energia consumata (E_{cons}):

$$E_{margin} = E_{prod} - E_{cons} \quad (3.11)$$

In accordo con l'equazione (3.11) l'energia totale consumata dai carichi di bordo non deve superare l'energia prodotta dai pannelli solari in modo da soddisfare i requisiti di progetto.

4. Sviluppo del software per il calcolo del power budget

Dopo aver modellato la potenza prodotta e quella consumata dal satellite e dopo aver definito il calcolo del power budget, si procede con la descrizione del software che è stato implementato per svolgere i suddetti calcoli.

La realizzazione dell'interfaccia grafica (Graphical User Interface, GUI), riportata in Figura 18, per la determinazione del power budget deriva dalla necessità di facilitare l'utente nel simulare gli andamenti di potenza prodotta dai pannelli solari e potenza consumata dai carichi a bordo per vari scenari di missione.

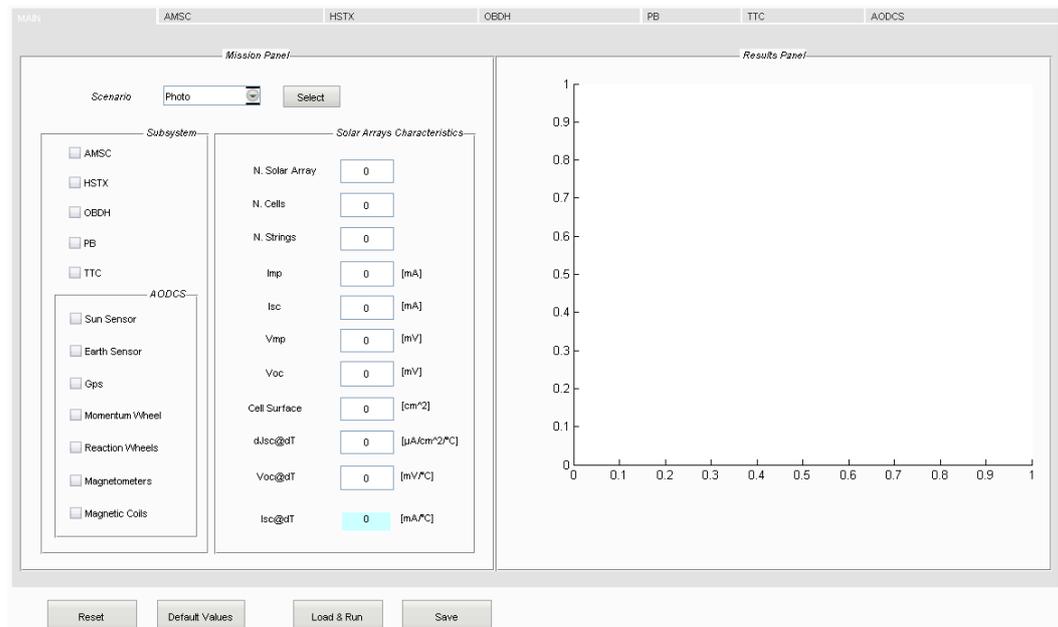


Figura 18 – La GUI del PB software

4.1. Programma GUIDE

La realizzazione dell'interfaccia grafica in Matlab® è resa possibile mediante l'approccio grafico con l'utilizzo del programma GUIDE, (Graphical User Interface Development Environment) (Figura 19) oppure direttamente a livello di codice di programmazione. Quest'ultimo rende il lavoro del programmatore più complicato e il tempo per realizzare l'interfaccia è maggiore.

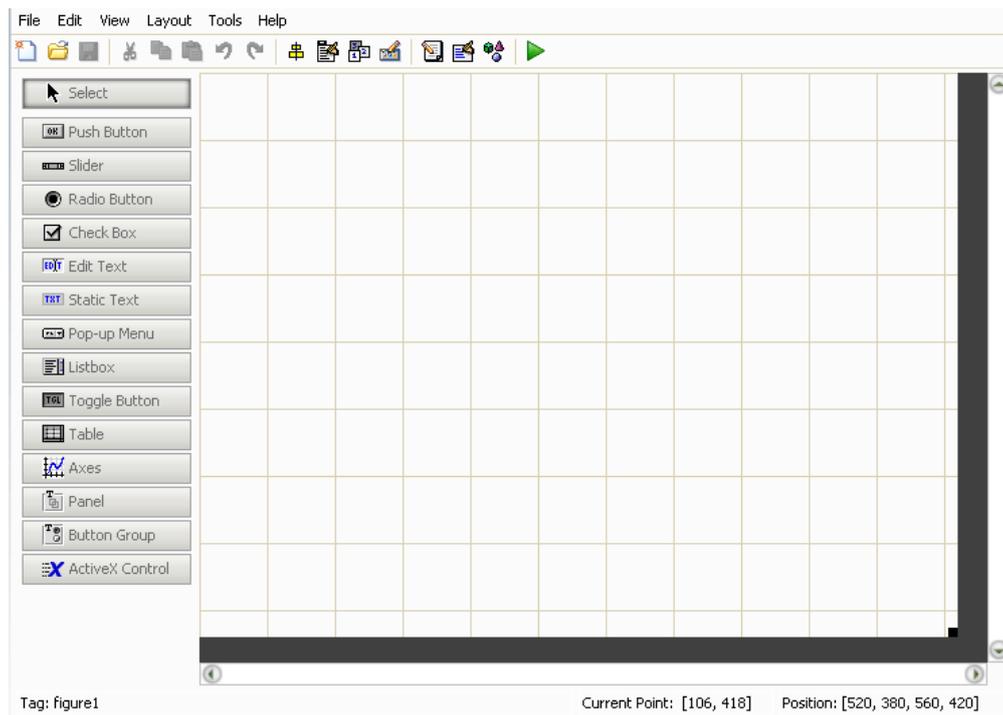


Figura 19 – Pannello principale del programma GUIDE

4.1.1. TabPanel Constructor

Tutte le GUI dei vari sottosistemi sono state unificate in un'interfaccia grafica grazie all'utilizzo del Tab Panel esterno a GUIDE, ma programmato in ambiente Matlab®, utilizzabile scrivendo nella command window la riga di codice:

```
>> tabpanel('gui_PowerBudget.fig', 'mytag');
```

Dove 'gui_PowerBudget.fig' indica il nome della GUI da sviluppare e 'mytag' la sua tag. La stringa di codice apre il Tab Panel Constructor (TPC) (Figura 20), da dove è possibile modificare e aggiungere le GUI in precedenza create.



Figura 20 - Tab Panel Constructor, TPC

Attraverso il tasto *add panel* si possono aggiungere le GUI, realizzate in GUIDE, all'interfaccia che si vuole realizzare, il tasto *edit* consente la modifica dei pannelli aggiunti che poi andranno caricati con il tasto *update*. Per modificare lo stile, il colore e i caratteri dell'interfaccia grafica si usa il tasto *settings*.

4.1.2. Elementi essenziali dell'interfaccia utente

Realizzata la parte grafica, si procede con il codice di programmazione utilizzando gli oggetti di Figura 21, già presenti in GUIDE.

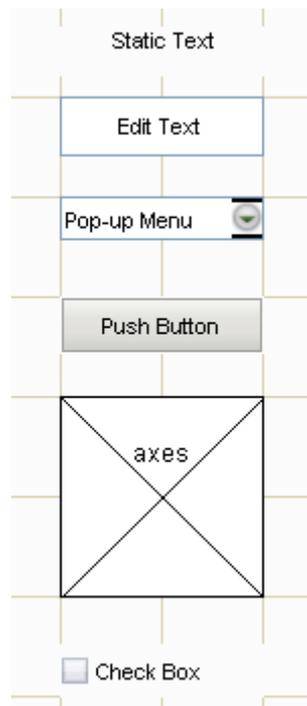


Figura 21 - Oggetti del programma GUIDE

La Tabella 20 mostra le principali caratteristiche degli oggetti.

Static text	mostrano righe di testo
Edit text	caselle di testo che l'utente può riempire per associare un valore a una variabile
Pop-Up Menu	tendina contenente una lista di parametri
Push Button	bottone in grado di eseguire una funzione
Axes	visualizza il plot dei dati
Chekbox	selezionatore/deselezionatore

Tabella 20 - Oggetti in GUIDE e loro caratteristiche

Cliccando su ogni oggetto, è possibile modificare le sue proprietà, con l'utilizzo del Property Inspector riportato in Figura 22.

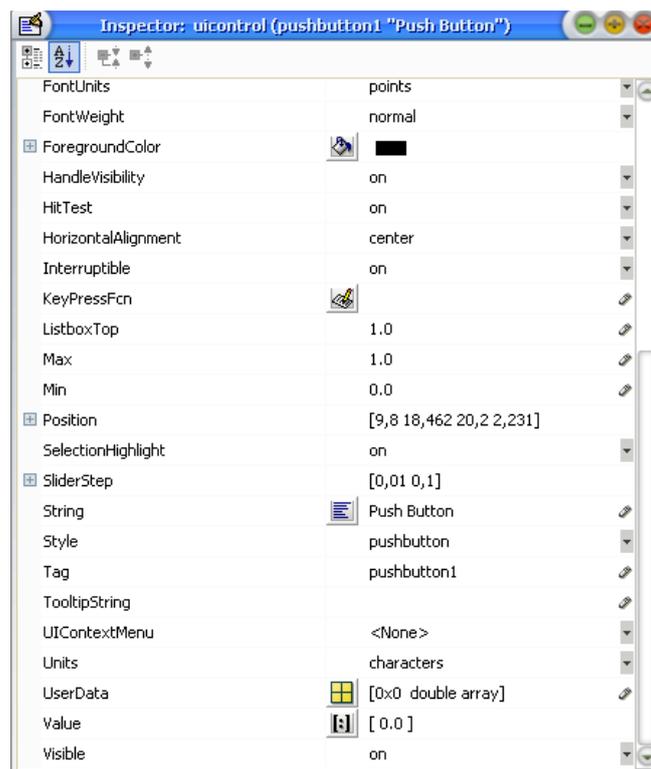


Figura 22 - Property Inspector

In particolare è possibile modificare la *String*, cioè il testo da visualizzare sopra l'oggetto e la sua *Tag*: una stringa che identifica univocamente l'oggetto.

4.1.3. Funzioni principali e loro utilizzo

Tutte le funzioni presenti nel codice relative a ogni oggetto, presentano la seguente intestazione:

```
>>function tag_oggetto_Callback (hObject,eventdata, handles)
>>hObject    puntatore dell'oggetto
>>eventdata  riservato-per future versioni di Matlab
>>handles    struttura con puntatori e dati
```

I puntatori di tutti gli oggetti che compongono la GUI sono memorizzati in modo automatico nella struttura chiamata *HANDLES* e grazie alla tag possono essere richiamati in tutte le funzioni.

Per utilizzare gli oggetti, come ad esempio gli edit text, si usa il comando *GET* che prende in ingresso le stringhe. Infine per associare un valore numerico a una variabile è necessaria la conversione da stringa a numero resa possibile dalla funzione *STR2DOUBLE*:

```
>> variable=str2double(get(handles.edit, 'string '));
```

La versatilità del codice permette di utilizzare la stessa riga di codice per gli static text e i pop-up menu. Per i checkbox è usato un approccio diverso, poiché sono oggetti di selezione o deselegione. Essi sono impiegati nel controllo dei sottosistemi accesi tramite la seguente riga di codice:

```
>> get(handles.checkbox, 'Value');
```

La stringa, con la funzione *GET*, prende il valore 'Value' assegnato all'oggetto checkbox. Se è stato selezionato allora è associato il valore 1 (acceso), viceversa se è stata deselegionata corrisponderà il valore 0 (spento).

4.2. Starting mission point

Per facilitare l'utente nel selezionare il punto da cui far partire la simulazione dei sottosistemi selezionati, si determina lo *starting mission point*. Questo punto è selezionabile dall'utente con il cursore del mouse direttamente sul grafico della potenza generata dai pannelli solari come mostra la Figura 23.

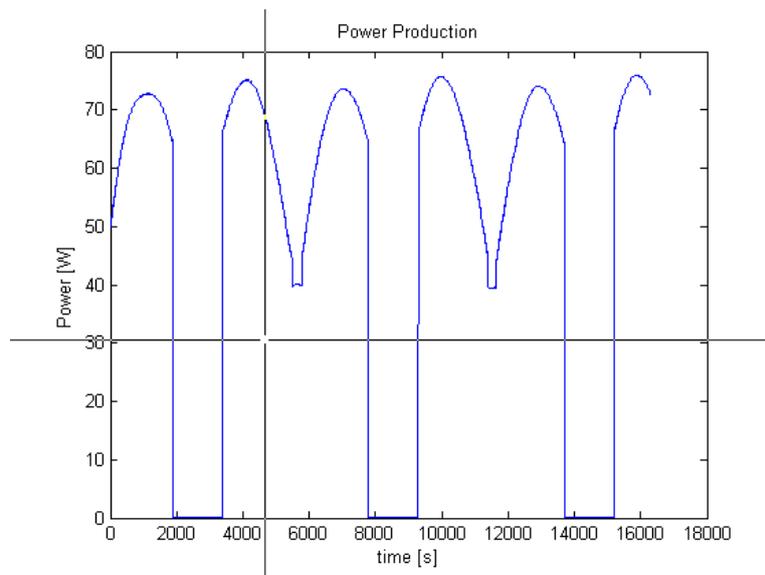


Figura 23 - Selezione dello starting mission point

In questo modo l'utilizzatore controlla l'andamento della potenza prodotta dai pannelli solari, e in base ai sottosistemi accesi, e quindi ai consumi attesi, collocherà l'istante iniziale in corrispondenza dei valori più elevati di potenza. Questa scelta è utile ad esempio per determinare l'istante d'inizio per acquisire le immagini oppure per avviare il trasferimento dati, in corrispondenza con i punti dove si troveranno le aree da acquisire o dove si troverà la stazione di terra.

Utilizzando la funzione *GINPUT*, e le relative righe di codice, si ottiene lo *starting mission point*:

```
>> [X]=ginput(1);  
>> t_on_P=round(X);
```

Dove, la funzione *GINPUT* permette di estrapolare l'ascissa sul grafico della potenza prodotta relativa al tempo [X], mentre il valore (1) stabilisce di estrarre il

valore cliccando una sola volta con il mouse. La seconda riga, assegna il valore intero, attraverso la funzione *ROUND*, alla variabile *starting mission point* definita come *t_on_P*.

4.3. Interfaccia grafica

L'interfaccia grafica è suddivisa in sette GUI:

- MAIN
- AMSC
- HSTX
- OBDH
- PB
- TTC
- AODCS

La principale, *Main Panel* riportata in Figura 24, si presenta divisa in due pannelli: *Mission Panel* e *Results Panel*, mentre le altre GUI si riferiscono ai sottosistemi presenti.

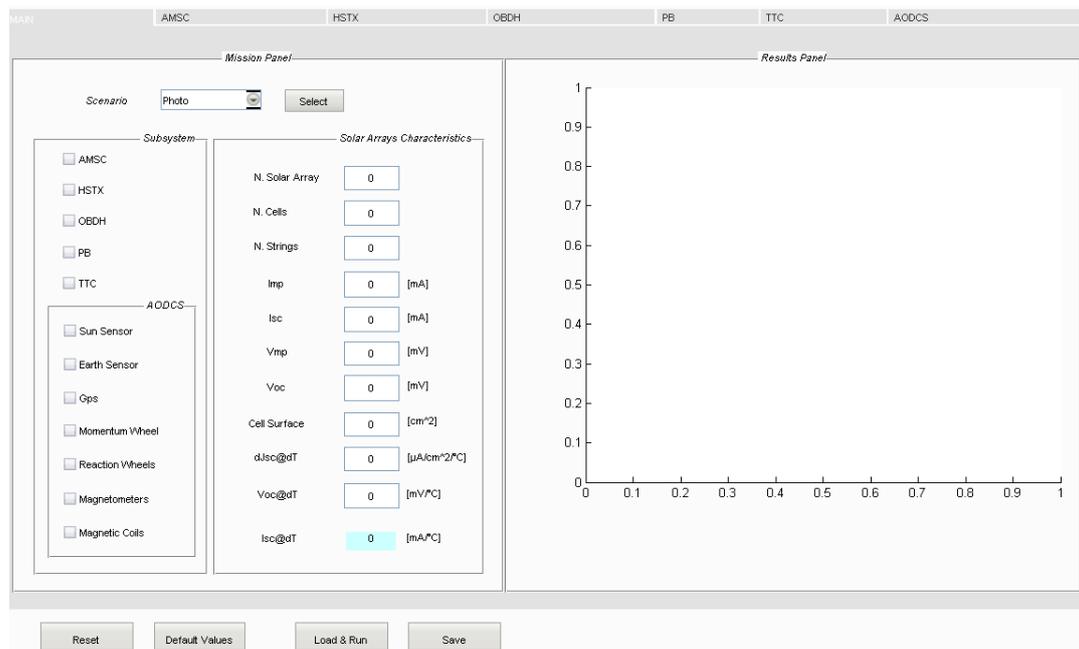


Figura 24 - Interfaccia Grafica

4.3.1. Mission Panel

Nel *Mission Panel* (Figura 25) sono presenti gli scenari di missione precaricati come Photo, Download e Photo&Download, lo scenario Custom che permette all'utente di modificare le potenze e i tempi di accensione di ciascun sottosistema nel *Subsystem Panel* (appendice A.7) ma anche i due scenari Detumbling e Nominal, che saranno dettagliati nel capitolo 5. Inoltre è compresa una parte, denominata *Solar Arrays Characteristics*, in cui l'utente può impostare le caratteristiche dei pannelli (si veda l'appendice A.1 per la parte di codice).

The screenshot shows the 'Mission Panel' interface with the following components:

- Top navigation: MAIN, AMSC, HSTX, OE
- Scenario selection: A dropdown menu set to 'Photo' and a 'Select' button.
- Subsystem selection (checkboxes):
 - AMSC
 - HSTX
 - OBDH
 - PB
 - TTC
- AODCS selection (checkboxes):
 - Sun Sensor
 - Earth Sensor
 - Gps
 - Momentum Wheel
 - Reaction Wheels
 - Magnetometers
 - Magnetic Coils
- Solar Arrays Characteristics (input fields):
 - N. Solar Array: 0
 - N. Cells: 0
 - N. Strings: 0
 - Imp: 0 [mA]
 - Isc: 0 [mA]
 - Vmp: 0 [mV]
 - Voc: 0 [mV]
 - Cell Surface: 0 [cm²]
 - dJsc@dT: 0 [μA/cm²°C]
 - Voc@dT: 0 [mV/°C]
 - Isc@dT: 0 [mA/°C]

Figura 25 - Mission Panel

4.3.2. Result panel

Definiti i vari oggetti e le loro proprietà, sono state inserite le equazioni ai fini del calcolo del power budget (capitolo 3), il cui andamento sarà visualizzato nel *Results Panel*. Come riporta la Figura 26, oltre a restituire gli andamenti di: potenza prodotta, andamento in blu, e la potenza consumata, andamento in rosso, calcola il valore di: potenza totale prodotta, la potenza totale consumata e il margine di energia durante la missione.

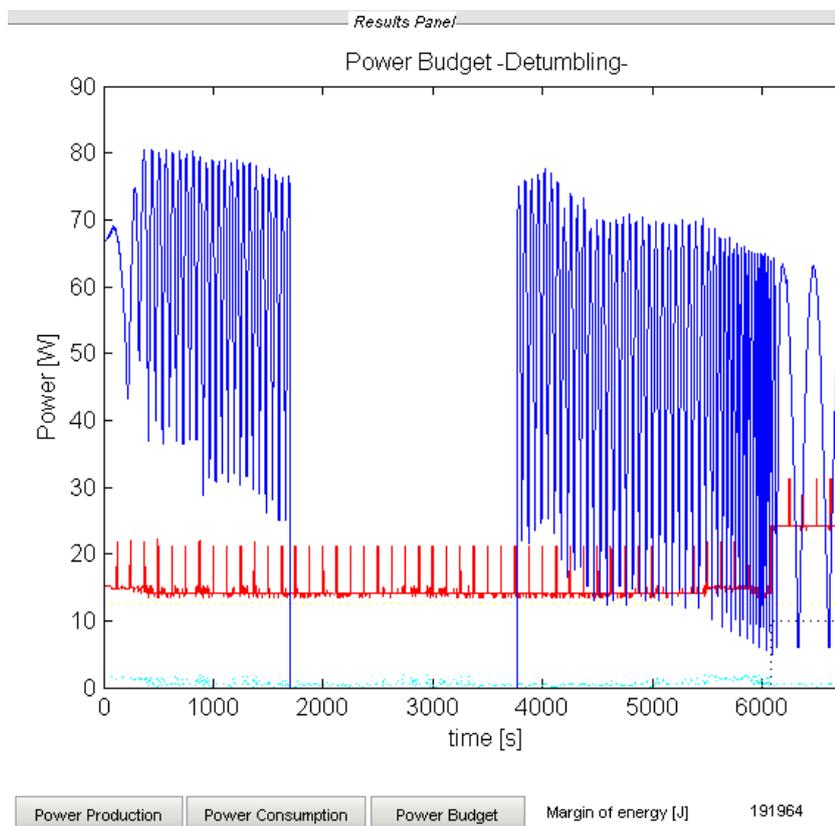


Figura 26 - Result Panel

4.3.3. Interfaccia grafica dei sottosistemi

L'interfaccia grafica dei sottosistemi, come accennato nel paragrafo 4.1.1, è stata realizzata in GUIDE (appendice A.2) e messe insieme al *MAIN panel* con il TPC.

L'interfaccia grafica del payload ottico *AMSC panel* riportata in Figura 27 richiede alcuni parametri da assegnare come il numero filtri, i secondi per ogni scatto, il campo d'acquisizione in Km (Sweep) e infine la potenza richiesta.

The screenshot shows a software interface with two tabs at the top: 'AMSC' and 'HSTX'. The 'AMSC' tab is selected. Below the tabs is a panel titled 'AMSC Panel'. Inside this panel, there are six rows of input fields, each with a label on the left and a text box on the right. The labels and their corresponding values are: 'N° Photos' (0), 'N° Bands' (0), 'Seconds per shot [s]' (0), 'Sweep [Km]' (0), 'Power Required [W]' (0), and 'Operating Time [s]' (0). The 'Operating Time [s]' field is highlighted in light blue.

Figura 27 - AMSC panel

La Figura 28 mostra l'interfaccia grafica del trasmettitore ad alta velocità, *HSTX panel*. Essa è stata sviluppata in modo da poter scegliere dal menù a tendina del Bit Rate la velocità di trasferimento: 1Mbit/s oppure 2Mbit/s.

The screenshot shows a software interface with two tabs at the top: 'AMSC' and 'HSTX'. The 'HSTX' tab is selected. Below the tabs is a panel titled 'HSTX Panel'. Inside this panel, there are three rows of input fields. The first row is 'Bit Rate', which has a dropdown menu open showing two options: '1 Mbit/s' (highlighted in blue) and '2 Mbit/s'. The second row is 'Initial Time[s]' with a text box containing '0'. The third row is 'Power Required [W]' with a text box containing '0'.

Figura 28 - HSTX Panel

La potenza richiesta e il tempo operativo, non sono modificabili dall'utente poiché sono valori predefiniti dalla velocità di trasmissione e dal tempo di sorvolo della stazione che è stato fissato a cinque minuti. Questo tempo corrisponde al passaggio di un satellite in orbita LEO ad alta elevazione su una stazione a medie latitudini.

La Figura 29 e la Figura 30 riportano le GUI riferenti all'*OBDH Panel* e al *PB Panel*, sottosistemi considerati a consumo costante durante la missione, per i quali quindi sono sufficienti definire i parametri caratteristici riportati in Tabella 13 e Tabella 14.

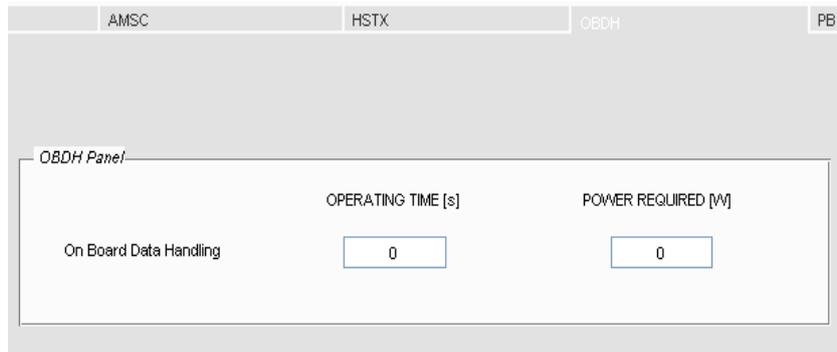


Figura 29 - OBDH Panel

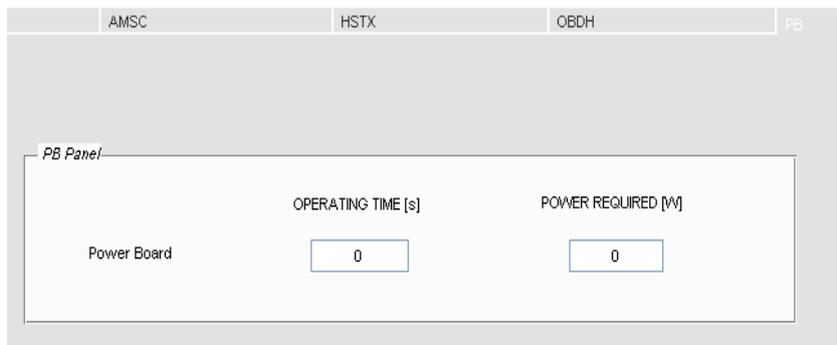


Figura 30 - PB Panel

Un diverso approccio è stato considerato per l'interfaccia grafica del trasmettitore *TTC Panel* (Figura 31) in cui l'utente può impostare la potenza consumata, il duty cycle e il tempo di accensione del trasmettitore, sia in modalità Telemetry Communication, comunicazione con la stazione di terra, che in modalità Beacon. Infine s'impostano due parametri necessari per la comunicazione con il satellite dalla stazione di terra: l'istante di tempo iniziale e il numero di orbite durante le quali si decide di intervenire.

HSTX	OBDH	PB	TTC
<i>TTC Panel</i>			
N° Orbit application	Start communication [s]	POWER REQUIRED [W]	
<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	
<i>Operating time selector</i>			
	DUTY CYCLE	OPERATING TIME [s]	
Beacon	<input type="text" value="0"/>	<input type="text" value="0"/>	
Telemetry communication	<input type="text" value="0"/>	<input type="text" value="0"/>	

Figura 31 - TTC panel

L'ultima interfaccia grafica *AODCS Panel* comprende: tutti i sensori elencati nel paragrafo 2.3.3 per i quali sono sufficienti impostare il tempo operativo e la potenza richiesta.

OBDH	PB	TTC	AODCS
<i>AODCS Panel</i>			
	OPERATING TIME [s]	POWER REQUIRED [W]	
Sun Sensor	<input type="text" value="0"/>	<input type="text" value="0"/>	
Earth Sensor	<input type="text" value="0"/>	<input type="text" value="0"/>	
Magnetometers	<input type="text" value="0"/>	<input type="text" value="0"/>	
Gps	<input type="text" value="0"/>	<input type="text" value="0"/>	
Momentum Wheel	<input type="text" value="0"/>	<input type="text" value="0"/>	
Reaction Wheels (Power value per wheel)	<input type="text" value="0"/>	<input type="text" value="0"/>	
Magnetic Coils (Load External File)			<input type="button" value="Load"/>

Figura 32 - AODCS Panel

4.4. Push Buttons

Nell'interfaccia grafica presentata nel paragrafo 4.3, sono stati inseriti diversi push buttons che permettono all'utente di interagire con il programma.

Il tasto *Select* (Figura 33), posto accanto agli scenari, visualizza i sottosistemi accesi e spenti per ogni tipo di missione e nel caso *Custom* azzerava tutte le checkbox (A.7).

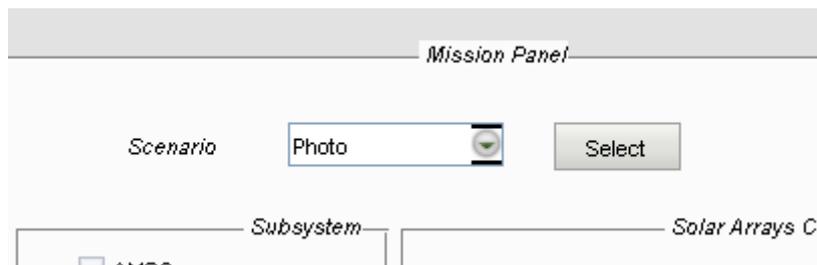


Figura 33 - Tasto Select

Nella parte inferiore della GUI sono stati aggiunti i bottoni principali come si può notare in Figura 34. Il tasto *Default Values* è stato inserito per facilitare e velocizzare l'inserimento dei parametri nel programma, e imposta i valori di default negli edit di tutti i pannelli (A.4). Il tasto *Reset* pone tutti gli oggetti e i rispettivi valori a zero e ripristina il *Results Panel* (A.6).

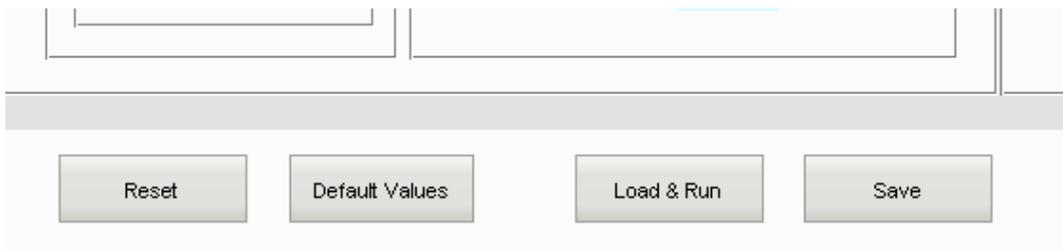


Figura 34 - Bottoni principali

Il tasto *Save* visualizza tutti i risultati di output riportati in tabella di Figura 35 cioè valori di potenza consumata e prodotta e i tre punti (I_{sc} , 0) (0, V_{oc}) (I_{mp} , V_{mp}) utili al Solar Array Simulator, per generare la curva caratteristica del pannello solare. Infine permette all'utente di esportarli in Microsoft Excel©, in formato di testo o in formato Matlab® (A.3).

	Time	Voc	Isc	Vmp	Imp	Relays(1)	Relays(2)
1	1	36.9993	2.2158	32.2813	2.1300	1	
2	2	37.0077	2.2159	32.2897	2.1301	1	
3	3	37.0161	2.2159	32.2981	2.1301	1	
4	4	37.0161	2.2161	32.2981	2.1303	1	
5	5	37.0245	2.2161	32.3065	2.1303	1	
6	6	37.0245	2.2162	32.3065	2.1303	1	
7	7	37.0329	2.2155	32.3149	2.1297	1	
8	8	37.0329	2.2156	32.3149	2.1298	1	
9	9	37.0329	2.2157	32.3149	2.1299	1	
10	10	37.0413	2.2157	32.3233	2.1299	1	
11	11	37.0413	2.2150	32.3233	2.1292	1	
12	12	37.0413	2.2151	32.3233	2.1293	1	
13	13	37.0497	2.2151	32.3317	2.1293	1	
14	14	37.0497	2.2144	32.3317	2.1286	1	
15	15	37.0497	2.2145	32.3317	2.1287	1	
16	16	37.0581	2.2145	32.3401	2.1287	1	
17	17	37.0581	2.2138	32.3401	2.1280	1	

Figura 35 - Tabella dei risultati

Il tasto *Load & Run* acquisisce tutti i valori negli edit, li associa alle variabili quindi esegue le equazioni per il calcolo delle potenze (A.1 , A.2).

Infine, come mostrato in Figura 36, ci sono i tre bottoni nella parte inferiore del *Result Panel: Power Production (A.5.1), Power Consumption (A.5.2) e Power Budget (A.5.3)*. Questi push buttons sono pulsanti di stampa degli andamenti di potenza e calcolano i relativi risultati delle equazioni:(3.1), (3.9) e (3.11).

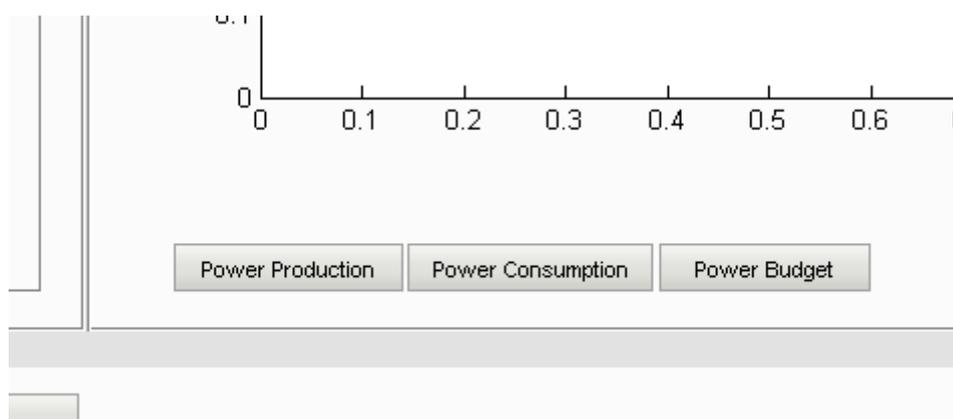


Figura 36 - Bottoni delle potenze

5. Scenari di missione

Gli scenari di missione attualmente previsti sono: Photo, Download, Photo&Download, Detumbling e Nominal. Inoltre è stato previsto un caso Custom per consentire all'utente di selezionare i sottosistemi in maniera indipendente e creare così ulteriori scenari d'interesse anche per differenti missioni. Di seguito saranno presentati nel dettaglio i vari scenari e le loro caratteristiche principali.

5.1. Scenario Photo:

In questo scenario di missione, i sottosistemi sono tutti accesi fatta eccezione per il trasmettitore ad alta velocità HSTX. Per acquisire le immagini in maniera corretta sopra l'obiettivo è importante sapere con precisione la posizione del satellite. Per questo è prevista l'accensione del GPS prima dell'inizio dell'acquisizione e poi per tutta la fase di acquisizione garantendo anche una corretta georeferenziazione delle immagini. In queste simulazioni, per cercare di avere un margine conservativo del power budget, è stato ipotizzando che il satellite uscendo dall'eclisse si trovi in un assetto completamente sbagliato, non-nadir pointing. Quindi l'unico modo per raggiungere e mantenere l'assetto desiderato velocemente è utilizzare le ruote di reazione, che saranno operate qualche minuto prima dell'inizio della fase di acquisizione.

Tutti gli altri sottosistemi sono comunque accesi per tutta la durata della missione. L'utente seleziona con la funzione *GINPUT*, descritta nel paragrafo 4.2, l'inizio della fase di acquisizione. Dalla Figura 37 si può notare il primo gradino in corrispondenza dell'accensione del GPS e il successivo picco in corrispondenza dell'accensione della ruota di reazione e l'AMSC.

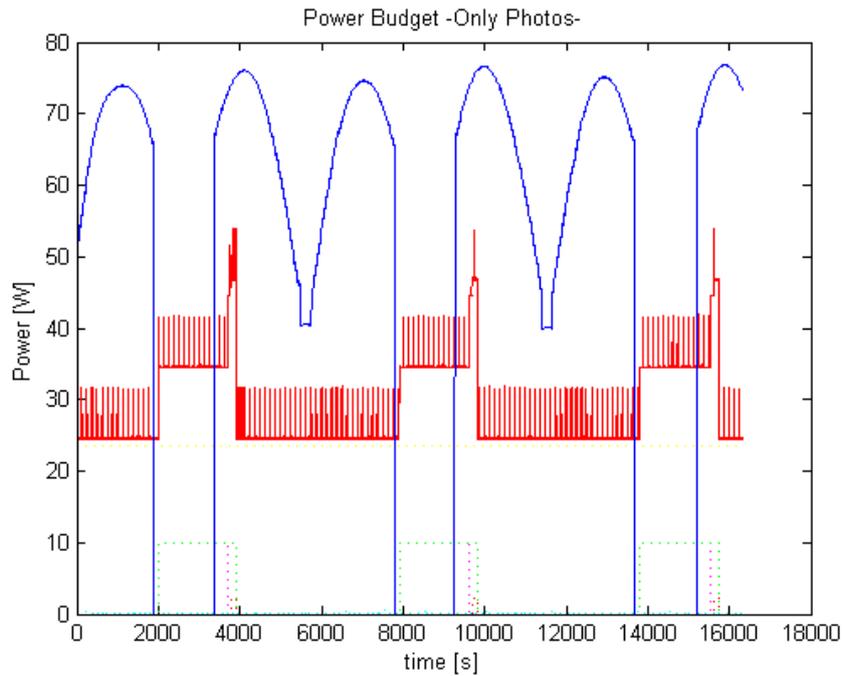


Figura 37 - Scenario di missione Photos

5.2. Scenario Download:

Per la missione che prevede solo il trasferimento dati, i sottosistemi sono tutti accesi tranne il payload ottico e i sottosistemi utilizzati per il corretto funzionamento dello stesso: le ruote di reazione e il GPS. L'utente seleziona con la funzione *GINPUT* l'inizio del trasferimento dati. Dalla Figura 38 si può notare il picco di consumi in corrispondenza dell'accensione del trasmettitore.

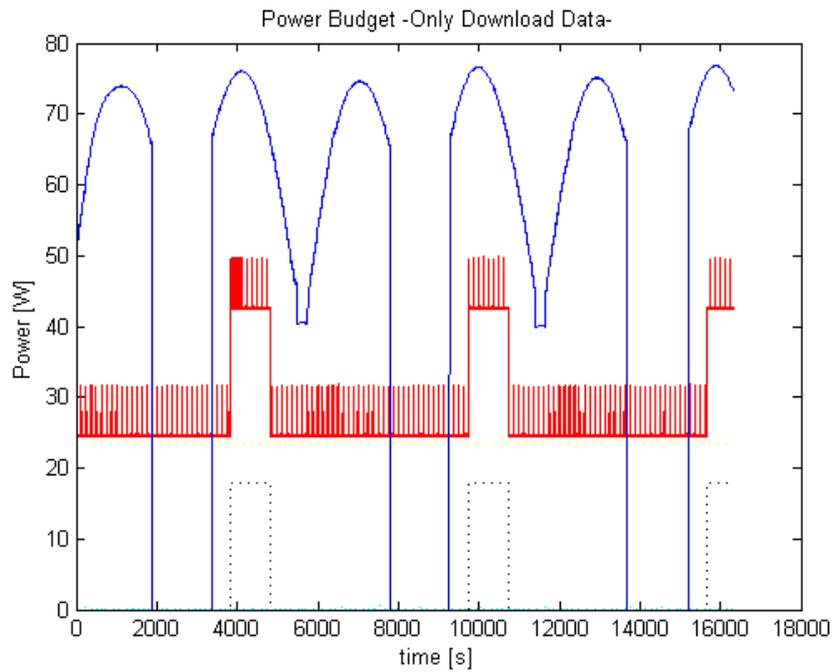


Figura 38 - Scenario di missione Download

5.3. Scenario Photo&Download:

In questo scenario, uno dei più critici, il satellite ha tutti i sottosistemi accesi dunque la potenza consumata raggiunge il valore più alto. In questo scenario è previsto che l'area d'interesse per l'acquisizione delle immagini comprenda la zona dove è installata la stazione di terra. Quindi, ci si ritroverà ad accendere il trasmettitore HSTX in corrispondenza del tempo di accensione del payload ottico. L'utente seleziona con la funzione *GINPUT* l'inizio di acquisizione delle immagini cui corrisponde il picco di consumi come si può notare in Figura 39. Questo scenario è simile a quello riportato nel paragrafo 5.1 con l'aggiunta del consumo del trasmettitore.

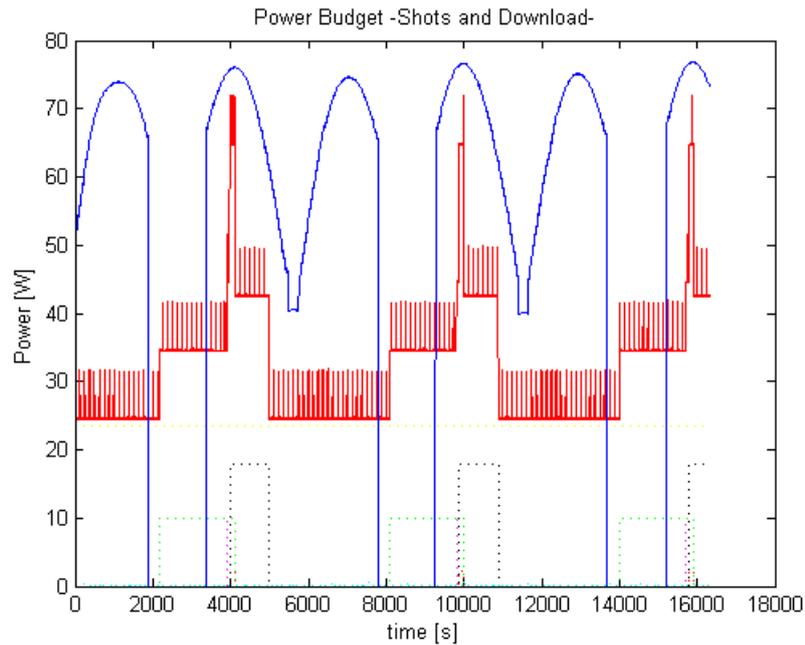


Figura 39 - Scenario di missione Photos& Download

La simulazione più critica sarà quella mostrata in Figura 40. In questo caso, l'istante iniziale per l'acquisizione delle immagini coincide con il punto di minima potenza prodotta (circa 40 W) in fase diurna dell'orbita, mentre la potenza richiesta dai carichi ha un valore di picco di 74 W.

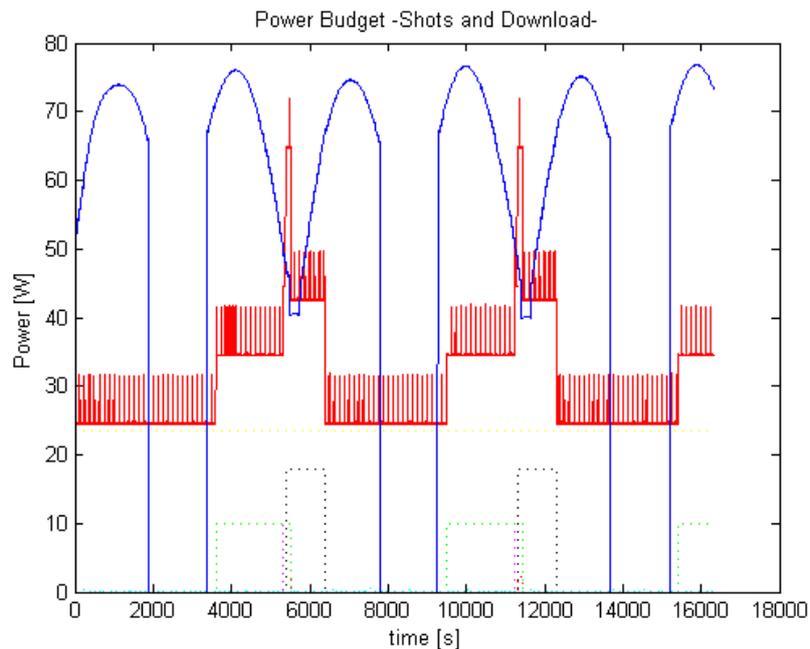


Figura 40 - Caso critico di simulazione

Questo picco di consumi può creare problemi al funzionamento del sistema in quanto anche se il satellite si trova illuminato sarà richiesta potenza dalle batterie.

5.4. Scenario Detumbling:

Il Detumbling è la prima fase della missione spaziale, cioè il momento in cui il satellite è sganciato dal lanciatore e inserito in orbita. Analizzare il power budget in questo scenario si rileva di fondamentale importanza perché i pannelli solari sono irradiati da angoli d'incidenza del sole che variano velocemente, dunque la potenza prodotta varia rapidamente nel tempo. In questo scenario alcuni sottosistemi sono accesi in sequenza poiché il satellite, una volta rilasciato, deve raggiungere il suo assetto operativo. L'utente imposta l'istante di accensione della ruota di momento che fornisce una certa rigidità giroscopica al satellite. In Figura 41 si può notare il gradino in corrispondenza dell'accensione della ruota di momento (andamento tratteggiato in nero) che rimarrà accesa per tutta la missione. Si può notare l'effetto della ruota di momento nella parte finale di simulazione in cui si ha una decisa diminuzione delle velocità angolari del satellite. Da qui in avanti si raggiungeranno le velocità di progetto, l'andamento della curva della potenza prodotta sarà quello delle orbite nominali.

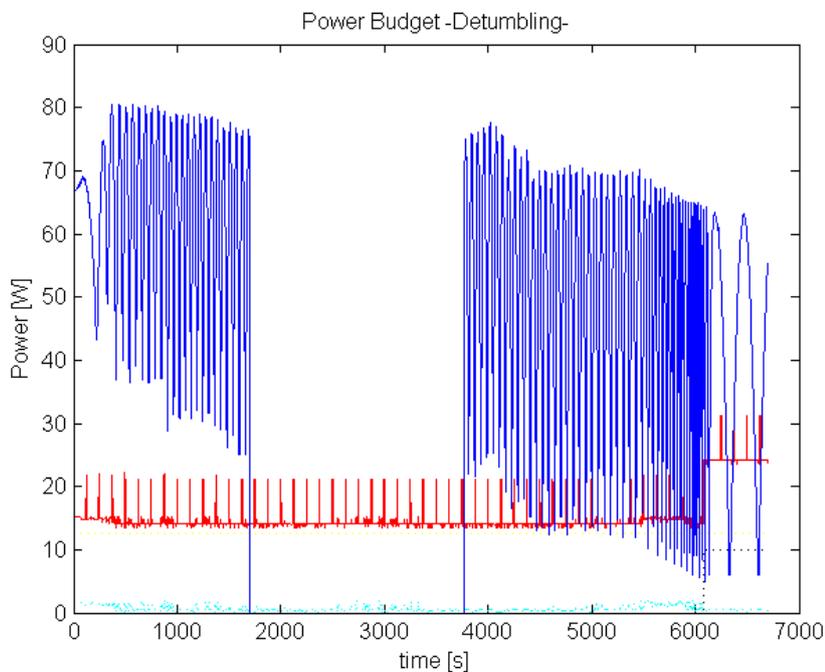


Figura 41 - Scenario di missione Detumbling

5.5. Scenario Nominal:

Questo scenario riguarda l'orbita nominale, in cui sono previsti tutti i sistemi accesi tranne il trasmettitore ad alta velocità e il payload ottico, di conseguenza il *GPS* e la ruota di reazione. Questa configurazione rappresenta il caso tipico di orbita, dove il satellite non sta sorvolando né l'area d'interesse per l'acquisizione delle immagini, né quella dove è presente la stazione di terra. In questo caso il software genera direttamente l'andamento delle potenze, riportate in Figura 42. Il payload e il trasmettitore sono spenti quindi non è definito lo *starting mission point*. Gli unici consumi sono quelli costanti (andamento tratteggiato in giallo) delle bobine magnetiche (andamento azzurro) e del TT&C in trasmissione e ricezione ma non in comunicazione con la stazione di terra.

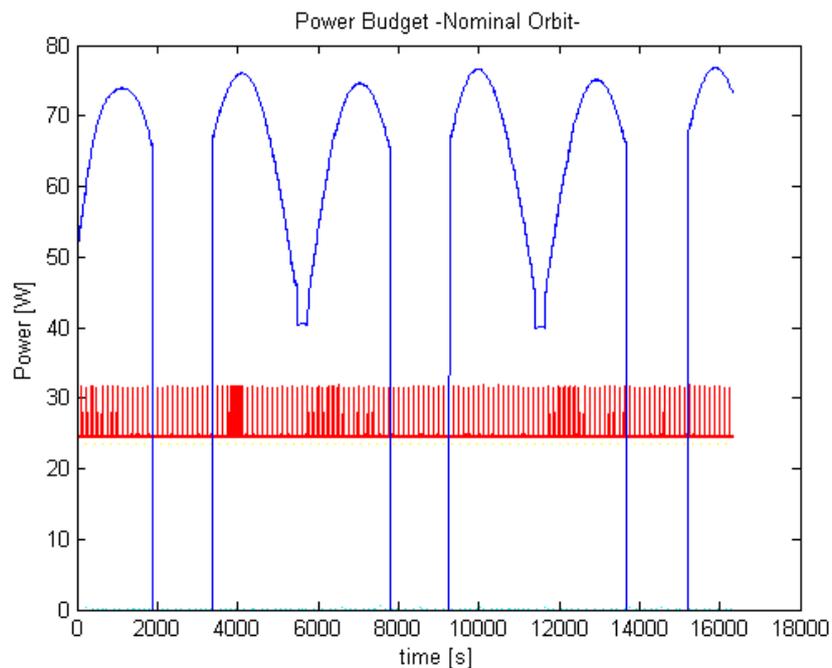


Figura 42 - Scenario di missione Nominal

5.6. Scenario Custom:

Come anticipato a inizio capitolo, questo scenario di missione non prevede nessun sistema preselezionato. Infatti, la struttura del codice è pensata per accogliere le richieste dell'analista di missione che stabilisce i parametri di funzionamento dei sottosistemi. L'utente seleziona quindi i sottosistemi che prevedrà accesi così da poter eseguire l'analisi del power budget.

5.7. Stima dei Power Budget

Per ogni tipo di scenario sono stati dunque calcolati: energia prodotta, energia consumata e margine di energia in accordo con l'equazione (3.10). La Tabella 21 elenca i risultati delle energie per ogni scenario di missione.

SCENARI OPERATIVI	ENERGIA PRODOTTA [J]	ENERGIA CONSUMATA [J]	MARGINE DI ENERGIA [J]
Photo	790834	471229	319605
Download (1Mbit\s)	790834	454644	336190
Photo&Download	790834	516688	274146
Detumbling	351727	332563	19164
Nominal	790834	406575	384259

Tabella 21 - Risultati di energia per i vari casi di scenario

Dai valori ottenuti in Tabella 21, possiamo notare che il caso più critico in termini di margine di energia è il Detumbling in cui l'energia prodotta e quella consumata hanno lo stesso ordine di grandezza. Questo scenario andrà perciò analizzato nella fase dei test soprattutto per studiare il funzionamento del sistema di potenza quando il satellite è in eclisse e quindi in assenza di potenza prodotta.

In tutti i casi, gli scenari riportano un valore positivo del power budget, si può dunque affermare che le specifiche di progetto, in queste prime fasi di simulazione, sono state rispettate.

6. Validazione del software per il power budget

Un aspetto importante del lavoro svolto riguarda la validazione dei risultati ottenuti con il software per il calcolo del power budget, illustrato nel capitolo 4. Tale validazione è stata possibile grazie all'utilizzo di un banco prova in cui sono stati testati alcuni elementi hardware del futuro sistema di potenza (7) del satellite ALMASat-EO.

Il banco prova riportato in Figura 43, esegue simulazione Hardware-In-the-Loop (HIL) in cui i profili della potenza generata sono simulati dall'Agilent® Solar Array Simulator (SAS), gestito tramite un computer. Lo stesso per mezzo della scheda controller imposta il segnale PWM (Pulse Width Modulation), e di conseguenza la corrente, sul carico elettronico della scheda controller in accordo con i dati in uscita dal SW del power budget.

The Power System Test bench block diagram

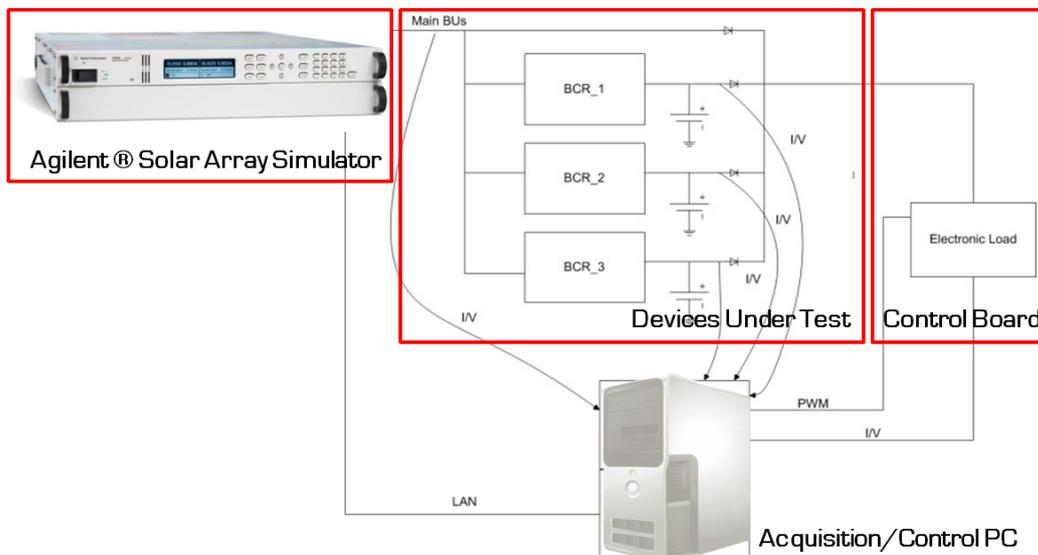


Figura 43 - Schema a blocchi del banco di prova

Il software per il controllo dei test riportato in Figura 44, è stato sviluppato in ambiente grafico LabVIEW 2010® da ALMASpace s.r.l. Esso gestisce la scheda controller e il SAS in base ai profili di potenza forniti in ingresso dal software per il calcolo del power budget. Tale interfaccia permette all'utente di verificare immediatamente il corretto funzionamento del sistema o visualizzare eventuali allarmi per mezzo di strumenti virtuali, registrare le letture dei sensori durante la

simulazione e infine memorizzare tutti i dati in un file di testo o un foglio di calcolo per una successiva analisi.

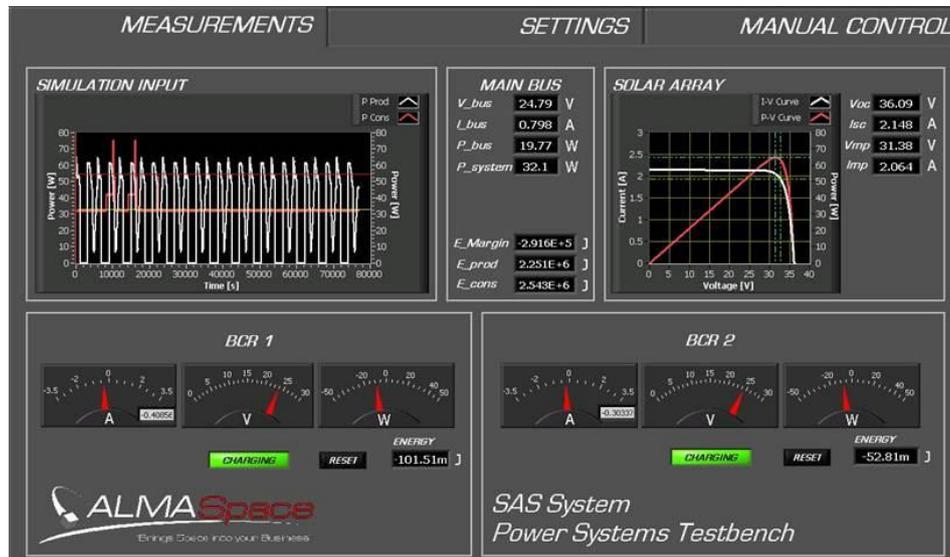


Figura 44 - Interfaccia di controllo dei test

Gli strumenti virtuali mostrano le tensioni e le correnti, le misure di alimentazione del main bus e delle batterie. Per estendere il periodo di test in tempo reale, fino ad alcune settimane, il banco esegue operazioni automatiche senza rendere necessaria la presenza dell'operatore. Come accennato i test sono realizzati tramite un file di testo ottenuto dal software del power budget, che viene usato per impostare il SAS Agilent® con i dati dei pannelli solari, attraverso i comandi standard per Strumenti programmabili (Standard Commands for Programmable Instruments, SCPI), stringhe, con le quali i consumi di potenza dal satellite, durante le operazioni in orbita, sono simulati da parte di un carico elettronico compreso nella scheda di potenza. La corrente che attraversa il carico è regolata variando il duty cycle in funzione della potenza impostata dal file d'input.

Due diversi scenari sono stati definiti e testati per valutare la capacità del sistema di potenza di ALMASat-EO. Il primo scenario considerato (caso Detumbling) è la fase di stabilizzazione del satellite, partendo dal rilascio dal lanciatore fino ad assumere l'assetto a tre assi stabilizzato, dove è stata validata la capacità del sistema di potenza per caricare le batterie, anche se il satellite non è in assetto nominale. Il secondo scenario considerato (caso payload) è una combinazione di orbite nominali e orbite dove il payload era stato messo in opera. Questo secondo scenario rappresenta un'attività tipica di missioni per EO.

Nei successivi paragrafi, dopo una descrizione dello scenario, saranno riportati i principali risultati della prova.

6.1. Scenario Detumbling

Lo scenario Detumbling è stato scelto per valutare il comportamento del sistema di potenza all'inizio della missione. Il test effettuato considera una condizione nominale di separazione del satellite: ovvero sono stati considerati, una velocità angolare di uscita molto bassa e un assetto non noto, ma sicuramente non nominale. Il test dura circa 6000 s, il tempo in cui le bobine magnetiche lavorano per fissare l'assetto del satellite nella sua configurazione nominale stabilizzata a tre assi con il payload ottico puntato verso la superficie terrestre. Questa fase della missione è cruciale per la stabilizzazione del satellite e per il rischio di perdere la comunicazione con il satellite in caso di assetto sbagliato.

Il calcolo del margine di potenza in questa configurazione è positivo: ovvero l'energia prodotta è maggiore dell'energia consumata, con un margine dell'ordine di 140 kJ.

Il test HIL dimostra inoltre la capacità del sistema di potenza del satellite nel ricaricare le batterie e di operare anche in eclisse. La Figura 45 mostra i risultati del test Detumbling per una batteria. Si possono vedere gli andamenti di: potenza prodotta (rosso), potenza consumata (giallo) e carica della batteria (blu). I valori in ordinate a sinistra riportano le potenze [W] mentre a destra vi è l'energia [Wh], in ascissa, invece abbiamo il tempo di volo in secondi.

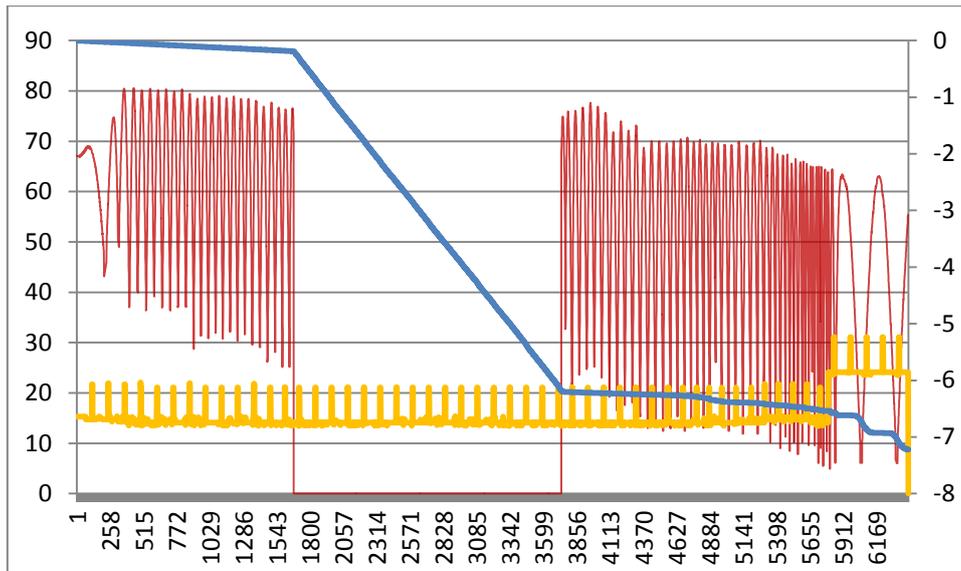


Figura 45 – Risultato dei test nel caso detumbling

Un risultato importante è rappresentato dall'andamento negativo della curva blu, che rappresenta la carica delle batterie. Ne consegue quindi che le batterie si scaricano, anche se il budget di potenza calcolata risulta positivo. Questo accade a causa dell'uso di una corrente di carica fissa, di valore non sufficiente a garantire la ricarica completa delle batterie. In particolare, poiché la Power Management Unit (PMU) non era disponibile, la corrente è stata fissata a $C / 8$, un ottavo della capacità di carica (C). Nonostante ciò, la simulazione HIL ben dimostra la performance attesa dei componenti del sistema di potenza disponibili e la capacità dei Battery Chargers Regulators (BCRs) di seguire il comportamento del sistema, anche con le limitazioni indicate.

6.2. Scenario Payload

Lo scenario payload rappresenta una tipica missione operativa EO. In questo caso si è ritenuto che il satellite sia azionato per acquisire immagini della superficie terrestre per una regione d'interesse di 1000 km in tre diverse bande e considerando la stazione di terra nella stessa area di acquisizione delle immagini. Un esempio reale di questo scenario può essere rappresentato da un obiettivo per il payload ottico rappresentato dall'intero territorio Italiano, e dalla collocazione della stazione di terra presso il laboratorio di Microsatelliti e Microsistemi spaziali dell'Università di Bologna a Forlì.

Considerando gli scenari descritti nel capitolo 5 il profilo di missione è una combinazione di scenari Nominal, Photo&Download e Download. In particolare, è stata creata una missione con cinque orbite in sequenza: un'orbita nominale, due orbite consecutive Photo&Download, un'orbita download e nuovamente un'orbita nominale. Questa combinazione è stata esaminata per studiare il comportamento del sistema di potenza quando il satellite si trova ad avere due orbite consecutive sopra l'obiettivo e in vista della stazione di terra. L'orbita successiva, Download, rappresenta un caso di sorvolo fuori dall'obiettivo ma in vista dalla stazione. L'ultima orbita nominale è stata inserita per ricaricare le batterie dopo l'intensa attività.

I risultati dei test sono riportati nella Figura 46. Anche in questo caso sono mostrati gli andamenti di potenza prodotta (rosso), consumata (giallo) e il livello di carica della batteria (blu). I valori in ordinate a sinistra sono potenze [W], a destra abbiamo le energie [Wh] mentre in ascissa l'istante di tempo [s].

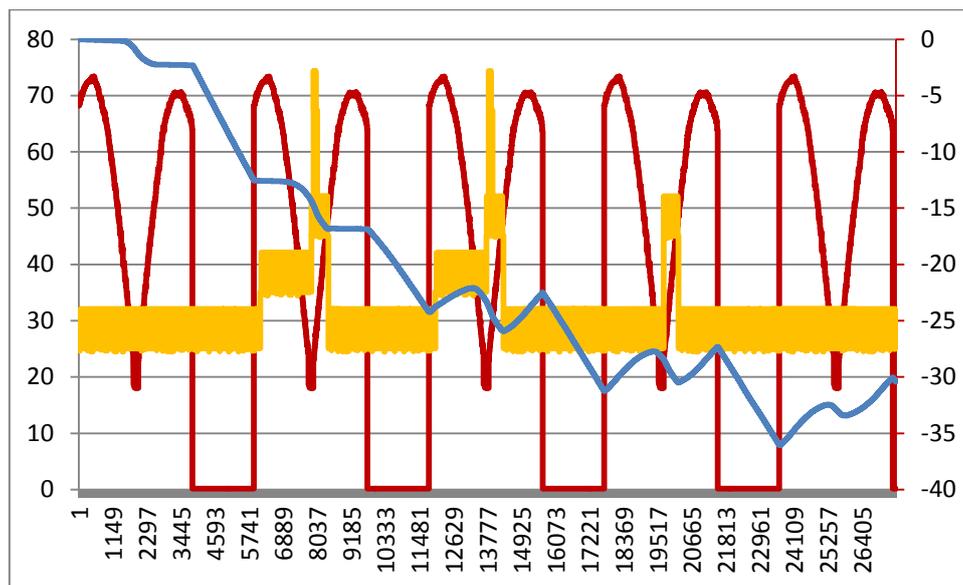


Figura 46 – Risultati dei test nel caso payload

Come per il caso Detumbling, la carica della batteria diminuisce per tutto il periodo della simulazione. In particolare, dopo le tre orbite con i payload accesi (AMSC e HSTX), le batterie si sono scaricate per 30Wh che è circa il 30% dell'energia delle batterie considerata nel test HIL. Un andamento negativo è quindi confermato anche in questa prova.

Anche se le batterie si scaricano durante le prime cinque orbite, nell'ultima orbita (payloads spenti) il livello di carica della batteria è negativo in eclisse (circa 10Wh), mentre è positivo durante la fase diurna (circa 6 Wh).

Questo test mostra nuovamente che, anche durante l'orbita nominale, la corrente di carica fissa non è sufficiente per ricaricare completamente le batterie durante la fase diurna. Per questo motivo, per ricaricare le batterie correttamente è importante impostare la corrente di carica al valore corretto tramite la PMU.

7. Conclusioni e Sviluppi futuri

Il lavoro svolto in questa tesi ha riguardato la definizione e l'implementazione di un software per il calcolo del power budget per microsattelliti della famiglia ALMASat. Il power budget è stato calcolato considerando la potenza prodotta dal satellite in base all'architettura del sistema di potenza e la potenza consumata durante la missione, considerando i sottosistemi attivati nelle varie fasi.

Uno dei principali risultati ottenuti da questo lavoro di tesi è stato la dimostrazione che il power budget sia positivo in ogni condizione operativa del satellite, compresa la modalità in orbita nominale in cui tutti i sottosistemi di bordo siano accesi e quindi i valori di potenza consumata siano massimi.

La validazione del power budget ottenuto è stata effettuata con dei test HIL dove sono stati presi in considerazione alcuni elementi del sistema di potenza del microsattellite: BCRs e batterie. Nonostante ciò, l'HIL ha evidenziato come l'utilizzo della corrente di carica fissa delle batterie non basti a garantire la corretta carica delle stesse durante i tempi d'illuminazione del satellite, sebbene il corretto funzionamento dei BCRs sia stato validato. Per ovviare a questo inconveniente bisognerà implementare l'intero sistema di potenza compresa la PMU in maniera da permettere la ricarica delle batterie al valore ottimale in funzione della reale potenza disponibile dai pannelli.

Inoltre, per un più accurato test HIL, vista la forma del satellite, sarà necessario dotare il sistema di simulazione di almeno due SAS, che genereranno la curva caratteristica contemporaneamente per 2 pannelli.

Infine, si è visto come il software per il calcolo del power budget sia stato sviluppato per essere utilizzabile nelle future missioni ALMASat. La versatilità del software consentirà anche la definizione di diversi profili di missione e scenari operativi, per verificare il power budget nelle diverse configurazioni.

A.1 Programma per il calcolo della potenza prodotta

```
function LR_PB_Callback(hObject, eventdata, handles)
% hObject      handle to LR_PB (see GCBO)
% eventdata    reserved - to be defined in a future version of
MATLAB
% handles      structure with handles and user data (see GUIDATA)

[filename, pathname] = uigetfile({'*.txt', 'Files (*.txt)'});
file=strcat(pathname,filename);

handles.n_pan=str2double(get(handles.edit1, 'string'));
handles.n_celle=str2double(get(handles.edit2, 'string'));
handles.n_stringhe=str2double(get(handles.edit3, 'string'));
handles.Imp=str2double(get(handles.edit4, 'string'));
handles.Isc=str2double(get(handles.edit5, 'string'));
handles.Vmp=str2double(get(handles.edit6, 'string'));
handles.Voc=str2double(get(handles.edit7, 'string'));
handles.cell_surf=str2double(get(handles.edit8, 'string'));
handles.dJsc_at_dT=str2double(get(handles.edit9, 'string'));
handles.dIsc_at_dT=(handles.dJsc_at_dT*handles.cell_surf)/1000; %
A/°C
handles.dVoc_at_dT=str2double(get(handles.edit11, 'string'));

%-----Power Production-----
-----[filename, pathname] = uigetfile({'*.txt', 'Files
(*.txt)'});

handles.Input=load(file);
handles.t_orbit=5908;

%%%%%%%%%%%% Solar Panel Parameters %%%%%%%%%%%%%

Tref=298.151;          % Reference conditions

handles.Isc_ang=zeros(length(handles.Input),handles.n_pan);
handles.Imp_ang=zeros(length(handles.Input),handles.n_pan);

i=1;
for w=3:6
    for g=1:length(handles.Input(:,1))
        if handles.Input(g,w)==1.57
            handles.Isc_ang(g,i)=0;          % angle [deg]
            handles.Imp_ang(g,i)=0;
        else

handles.Isc_ang(g,i)=handles.Isc*cosd(handles.Input(g,w)*(180/pi));
% angle [deg]

handles.Imp_ang(g,i)=handles.Imp*cosd(handles.Input(g,w)*(180/pi));
        end
        end
        i=i+1;
    end

mask=handles.Isc_ang>0;          % exclude negative
values
handles.Isc_ang=mask.*handles.Isc_ang; %
```

```

mask=handles.Imp_ang>0; % exclude negative
values
handles.Imp_ang=mask.*handles.Imp_ang; %

handles.Isc_ang_T=zeros(length(handles.Input(:,1)),handles.n_pan);
handles.Voc_T=zeros(length(handles.Input(:,1)),handles.n_pan);
handles.Imp_ang_T=zeros(length(handles.Input(:,1)),handles.n_pan);
handles.Vmp_T=zeros(length(handles.Input(:,1)),handles.n_pan);

jj=1;

for j=8:11
    for g=1:length(handles.Input(:,1))
        if handles.Isc_ang(g,jj)==0
            handles.Isc_ang_T(g,jj)=0;
            handles.Imp_ang_T(g,jj)=0;
        else

handles.Isc_ang_T(g,jj)=handles.Isc_ang(g,jj)+handles.dIsc_at_dT*(
handles.Input(g,j)+273.15-Tref);

handles.Imp_ang_T(g,jj)=handles.Imp_ang(g,jj)+handles.dIsc_at_dT*(
handles.Input(g,j)+273.15-Tref);
            end

handles.Voc_T(g,jj)=handles.Voc+handles.dVoc_at_dT*(handles.Input(
g,j)+273.15-Tref);

handles.Vmp_T(g,jj)=handles.Vmp+handles.dVoc_at_dT*(handles.Input(
g,j)+273.15-Tref);
            end
            jj=jj+1;
        end

mask=handles.Isc_ang_T>0; % exclude negative
values
handles.Isc_ang_T=mask.*handles.Isc_ang_T; %

mask=handles.Imp_ang_T>0; % exclude negative
values
handles.Imp_ang_T=mask.*handles.Imp_ang_T; %

handles.Vmp_temp_Lpan=handles.Vmp_T*handles.n_celle;
handles.Imp_ang_T_Lpan=handles.Imp_ang_T*handles.n_stringhe;

handles.VocT_Lpan=handles.Voc_T*handles.n_celle;
handles.Isc_ang_T_Lpan=handles.Isc_ang_T*handles.n_stringhe;

handles.Pmp_ang_temp_Lpan=(handles.Vmp_temp_Lpan.*handles.Imp_ang_
T_Lpan)./10^6;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Power Production %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
handles.PowerT=zeros(length(handles.Input),1);
handles.flag=zeros(length(handles.Input),handles.n_pan);

for k=1:length(handles.Input)
    for ii=1:(handles.n_pan)

        if handles.Pmp_ang_temp_Lpan(k,ii)<0.1
            handles.flag(k,ii)=0;
        end
    end
end

```

```

elseif handles.Pmp_ang_temp_Lpan(k,ii)>=0.1
    handles.flag(k,ii)=1;
end

end

handles.PowerT(k,1)=sum(handles.Pmp_ang_temp_Lpan(k,:));
end

Imp_box=zeros(length(handles.Input),2);
Isc_box=zeros(length(handles.Input),2);
VocT_box=zeros(length(handles.Input),2);
Vmp_box=zeros(length(handles.Input),2);

handles.Imp_new=zeros(length(handles.Input),1);
handles.Isc_new=zeros(length(handles.Input),1);
handles.VocT_new=zeros(length(handles.Input),1);
handles.Vmp_new=zeros(length(handles.Input),1);

for i=1:length(handles.Input)
    N=find(handles.flag(i,:)>0);
    for k=1:length(N)
        Imp_box(i,k)=handles.Imp_ang_T_Lpan(i,N(:,k));
        Isc_box(i,k)=handles.Isc_ang_T_Lpan(i,N(:,k));
        VocT_box(i,k)=handles.VocT_Lpan(i,N(:,k));
        Vmp_box(i,k)=handles.Vmp_temp_Lpan(i,N(:,k));
    end
    handles.Imp_new(i,1)=sum(Imp_box(i,:));
    handles.Isc_new(i,1)=sum(Isc_box(i,:));
    handles.VocT_new(i,1)=max(VocT_box(i,:));
    handles.Vmp_new(i,1)=max(Vmp_box(i,:));
end

handles.PowerT1=handles.Vmp_new.*handles.Imp_new/(10^6);

handles.E_prod=trapz(handles.PowerT);
time=1:length(handles.Input);

plot(time,handles.PowerT,'-b')
title('Power Production')
xlabel('time [s]')
ylabel('Power [W]')
hold on

handles.flag=ones(length(handles.Input),handles.n_pan);

set(handles.text4,'String','Energy Produced [J]');
set(handles.text3,'String',handles.E_prod);

```


A.2 Programma per il calcolo della potenza consumata

```
%AMSC
handles.n_Photos=str2double(get(handles.editt01, 'string'));
handles.Filters=str2double(get(handles.editt02, 'string'));
handles.s_shot=str2double(get(handles.editt03, 'string'));
handles.op_range=str2double(get(handles.editt04, 'string'));
handles.t_op_P=str2double(get(handles.editt05, 'string'));
handles.peak_Photos=str2double(get(handles.editt06, 'string'));
%DownLink
handles.t_on_DownL=str2double(get(handles.editt017, 'string'));
handles.peak_HSTX=str2double(get(handles.editt018, 'string'));
%contant consumption
handles.peak_Sun=str2double(get(handles.editt08, 'string'));
handles.peak_earth=str2double(get(handles.editt010, 'string'));
handles.peak_Magn=str2double(get(handles.editt012, 'string'));
handles.rms_OBDH=str2double(get(handles.editt020, 'string'));
handles.peak_PS=str2double(get(handles.editt022, 'string'));
handles.peak_TTC=str2double(get(handles.editt024, 'string'));
%time befor shot(125sec per photos)
handles.t_prep_Gps=str2double(get(handles.editt013, 'string'));
handles.peak_Gps=str2double(get(handles.editt014, 'string'));
%Reaction wheels
handles.t_op_RW=str2double(get(handles.editt015, 'string'));
handles.rms_RW=str2double(get(handles.editt016, 'string'));
%Momentum wheel
handles.t_on_MW=str2double(get(handles.editt0MW, 'string'));
handles.rms_MW=str2double(get(handles.editt1MW, 'string'));

Coil_power=handles.Input;

I=get(handles.popupmenu1, 'Value');
L=get(handles.popupmenu_01, 'Value');
t_orbit=5908;           %[s] orbital period

frame=handles.n_Photos*handles.Filters;

if I~=5 && I~=6
[X]=ginput(1);           %[s] ignition time
t_on_P=round(X);
t_off_P=t_on_P+handles.t_op_P;           %[s] power-off time

%%%%% Download Link @ 1Mb %%%%%
peak_TX_1Mb=18;           %[W]
% DownL_1Mb=1;           %[Mbit]

Delta_t=0;           %[s] seconds after the end of the shots
t_on_D_1Mb=t_on_P+Delta_t;           %[s] ignition time
t_op_D_1Mb=1000;           %[s] Transfer to 1 Mbit/s
t_off_D_1Mb=t_on_D_1Mb+t_op_D_1Mb;           %[s] power-off time

%%%%% Download Link @ 2Mb %%%%%
peak_TX_2Mb=20;           %[W]
% DownL_2Mb=2;           %[Mbit]

t_on_D_2Mb=t_on_P+Delta_t;           %[s] ignition time
t_op_D_2Mb=500;           %[s] Transfer to 2 Mbit/s
t_off_D_2Mb=t_on_D_2Mb+t_op_D_2Mb;           %[s] power-off time
t_on_RW=(t_on_P+handles.t_op_P)-(handles.t_op_RW);%[s]ignitiontime
t_off_RW=t_on_RW+handles.t_op_RW;           %[s] power-off time
```

```

%%%%%%%% Gps %%%%%%%%%
t_on_Gps=t_on_P-handles.t_prep_Gps;      %[s] ignition time
t_off_Gps=t_on_P+handles.t_op_P;        %[s] turn-off time
end

%%%%%%%% Momentum Wheel %%%%%%%%%
peak_MW=handles.rms_MW;    %[W]

%%%%%%%% Reaction Wheel %%%%%%%%%
peak_RW=handles.rms_MW;    %[W]

%%%%%%%% Coils %%%%%%%%%
Coil_x=Coil_power(1:length(Coil_power),12);
Coil_y=Coil_power(1:length(Coil_power),13);
Coil_z=Coil_power(1:length(Coil_power),14);

cons_Coil=(Coil_x+Coil_y+Coil_z)';
cons_Coil=[cons_Coil cons_Coil];

%%%%%%%% Costant Consumptions %%%%%%%%%
if I==5

C_cons=handles.peak_Magn+handles.peak_Sun+handles.rms_OBDH+handles
.peak_TTC+handles.peak_PS;
else

C_cons=handles.peak_earth+handles.peak_Magn+handles.peak_Sun+handl
es.rms_OBDH+handles.peak_TTC+handles.peak_PS+peak_MW;
end

n=2;

%%% Zeros Vectors %%%
C_cons=zeros(1,length(Coil_power)*n)+C_cons;
cons_Photos=zeros(1,length(Coil_power)*n);
cons_DownL=zeros(1,length(Coil_power)*n);
cons_RW=zeros(1,length(Coil_power)*n);
cons_MW=zeros(1,length(Coil_power)*n);
cons_Gps=zeros(1,length(Coil_power)*n);

cons_TTC=zeros(1,length(Coil_power)*n)+peak_Rx_TTC;

for jj=1:handles.DC_TTC:length(Coil_power)*n
    for ii=(handles.DC_TTC-handles.t_op_TTC):handles.DC_TTC
        cons_TTC(1,jj+ii)=handles.peak_TTC+peak_Rx_TTC;
    end
        for ll=1:t_orbit*handles.N_orbitTTC:length(Coil_power)*n
            for
kk=handles.start_ttc:handles.dc_ttc:(handles.start_ttc+300)
                for ww=(handles.dc_ttc-
handles.op_ttc):handles.dc_ttc

cons_TTC(1,(kk+ww)+ll)=handles.peak_TTC+peak_Rx_TTC;
                end
            end
        end
    end

end

switch I
    case 1

```

```

if t_on_P<handles.t_prep_Gps
    errordlg('Starting time mission too low', 'Error')
    return
end
for jj=1:t_orbit:length(Coil_power)*n
    for ii=t_on_P+1:t_off_P

cons_Photos(1,jj+ii)=(handles.s_shot*frame)*handles.peak_Photos;
end
    for ii=t_on_RW:t_off_RW
        cons_RW(1,jj+ii)=peak_RW;
    end
    for ii=t_on_Gps:t_off_Gps
        cons_Gps(1,jj+ii)=handles.peak_Gps;
    end
end
switch L
    case 1
    otherwise
        errordlg('Transfer is not expected', 'Error')
        set(handles.text3,'String', 'Error');
        return
    end

%% Potenza/Energia Consumata Totale %%%
a=length(cons_Photos);
b=length(cons_RW);
c=length(cons_Gps);
d=length(cons_Coil);
e=length(C_cons);

cons_Photos((length(Coil_power)+1):a)=[];
cons_RW((length(Coil_power)+1):b)=[];
cons_Gps((length(Coil_power)+1):c)=[];
cons_Coil((length(Coil_power)+1):d)=[];
C_cons((length(Coil_power)+1):e)=[];

handles.cons_tot=cons_Photos+cons_RW+cons_Gps+cons_Coil+C_cons;
%[W]
handles.E_cons=trapz(handles.cons_tot); %[J]

figure(2)
plot(time,cons_Photos','r',time,cons_RW','m',time,cons_Gps','g'
,time,cons_Coil','c',time,C_cons','y',time,handles.cons_tot','-
r',time,handles.PowerT,'-b')
legend('Photos','M.Weels','Gps','Coil','Costants','Power
Consumption','Power Production')
title('Power Budget -Only Photos-')
xlabel('time [s]')
ylabel('Power [W]')
hold off

    case 2;
        for jj=1:handles.t_orbit:length(Coil_power)*n
            switch L
                case 1
                    errordlg('Plese select the transfer rate',
'Error')
                    set(handles.text3,'String', 'Error');
                    return
                case 2
                    for ii=t_on_D_1Mb:t_off_D_1Mb

```

```

        cons_DownL(1,jj+ii)=peak_TX_1Mb;
    end
    case 3
        for ii=t_on_D_2Mb:t_off_D_2Mb
            cons_DownL(1,jj+ii)=peak_TX_2Mb;
        end
    end
end
end

%% Potenza/Energia Consumata Totale %%%
a=length(cons_DownL);
c=length(C_cons);
d=length(cons_Coil);

cons_DownL((length(Coil_power)+1):a)=[];
C_cons((length(Coil_power)+1):c)=[];
cons_Coil((length(Coil_power)+1):d)=[];

handles.cons_tot=cons_DownL+cons_Coil+C_cons;
handles.E_cons=trapz(handles.cons_tot); % [J]

figure(2)
plot(time,cons_DownL,':k',time,cons_Coil,':c',time,C_cons,':y',
time,handles.cons_tot,'-r',time,handles.PowerT,'-b')
legend('Download L.','Coils','Costants','Power Consumption','Power
Production')
title('Power Budget -Only Download Data-')
xlabel('time [s]')
ylabel('Power [W]')
hold off

    case 3
        if t_on_P<handles.t_prep_Gps
            errordlg('Starting time mission too low', 'Error')
            return
        end
        for jj=1:t_orbit:length(Coil_power)*n
            for ii=t_on_P+1:t_off_P

cons_Photos(1,jj+ii)=(handles.s_shot*frame)*handles.peak_Photos;
            end
            for ii=t_on_RW:t_off_RW
                cons_RW(1,jj+ii)=peak_RW;
            end
            for ii=t_on_Gps:t_off_Gps
                cons_Gps(1,jj+ii)=handles.peak_Gps;
            end
            switch L
                case 1
                    errordlg('Plese select the transfer rate',
'Error')
                    set(handles.text3,'String', 'Error');
                    return
                case 2
                    for ii=t_on_P+1:t_off_D_1Mb
                        cons_DownL(1,jj+ii)=peak_TX_1Mb;
                    end
                case 3
                    for ii=t_on_P:t_off_D_2Mb
                        cons_DownL(1,jj+ii)=peak_TX_2Mb;
                    end
            end
        end
end
end

```

end

```
%% Potenza/Energia Consumata Totale %%%%
a=length(cons_Photos);
b=length(cons_DownL);
c=length(cons_RW);
d=length(cons_Gps);
e=length(cons_Coil);
f=length(C_cons);

cons_Photos((length(Coil_power)+1):a)=[];
cons_DownL((length(Coil_power)+1):b)=[];
cons_RW((length(Coil_power)+1):c)=[];
cons_Gps((length(Coil_power)+1):d)=[];
cons_Coil((length(Coil_power)+1):e)=[];
C_cons((length(Coil_power)+1):f)=[];

handles.cons_tot=cons_Photos+cons_DownL+cons_RW+cons_Gps+cons_Coil
+C_cons; %[W]
handles.E_cons=trapz(handles.cons_tot); %[J]

figure(2)
plot(time,cons_Photos','r',time,cons_DownL','k',time,cons_RW',':
m',time,cons_Gps','g',time,cons_Coil','c',time,C_cons','y',time
,handles.cons_tot','-r',time,handles.PowerT,'-b')
legend('Photos','Download
L.','R.Weels','Gps','Coil','Costants','Power Consumption','Power
Production')
title('Power Budget -Shots and Download-')
xlabel('time [s]')
ylabel('Power [W]')
hold off

case 4
for jj=1:handles.t_orbit:length(Coil_power)*n
if (get(handles.checkbox1,'Value')==
get(hObject,'Max'))
for ii=t_on_P+1:t_off_P

cons_Photos(1,jj+ii)=(handles.s_shot*frame)*handles.peak_Photos;
end
else cons_Photos=zeros(1,length(Coil_power)*n);
end
switch L
case 1
if (get(handles.checkbox2,'Value')== get(hObject,'Max'))
errordlg('Plese select the transfer rate',
'Error')
set(handles.text3,'String','Error');
return
end
case 2
if (get(handles.checkbox2,'Value')==
get(hObject,'Max'))
for ii=1:handles.t_on_DownL
cons_DownL(1,jj+ii)=0;
end
for ii=handles.t_on_DownL+1:t_off_D_1Mb
cons_DownL(1,jj+ii)=peak_TX_1Mb;
end
```

```

elseif (get(handles.checkbox2,'Value') ==
get(hObject,'Min'))
    cons_DownL=zeros(1,length(Coil_power));
else
    errordlg('Transfer rate not required',
'Error')
    set(handles.text3,'String','Error');
    return
end
case 3
if (get(handles.checkbox2,'Value') ==
get(hObject,'Max'))
    for ii=1:handles.t_on_DownL
        cons_DownL(1,jj+ii)=0;
    end
    for ii=handles.t_on_DownL:t_off_D_2Mb
        cons_DownL(1,jj+ii)=peak_TX_2Mb;
    end
elseif (get(handles.checkbox2,'Value') ==
get(hObject,'Min'))
    cons_DownL=zeros(1,length(Coil_power)*n);
else
    errordlg('Transfer rate not required',
'Error')
    set(handles.text3,'String','Error');
    return
end
end
if (get(handles.checkbox9,'Value') ==
get(hObject,'Max'))
    for ii=t_on_RW:t_off_RW
        cons_RW(1,jj+ii)=peak_RW;
    end
else cons_RW=zeros(1,length(Coil_power)*n);
end
if t_on_P<handles.t_prep_Gps
    errordlg('Starting time mission too low','Error')
    return
end
if (get(handles.checkbox8,'Value') ==
get(hObject,'Max'))
    for ii=t_on_Gps:t_off_Gps
        cons_Gps(1,jj+ii)=handles.peak_Gps;
    end
else cons_Gps=zeros(1,length(Coil_power)*n);
end
end

%% Potenza/Energia Consumata Totale %%%
if (get(handles.checkbox11,'Value') == get(hObject,'Max'))
    cons_Coil=(Coil_x+Coil_y+Coil_z)';
    cons_Coil=[cons_Coil cons_Coil];
else
    cons_Coil=zeros(1,length(Coil_power)*n);
end

%% Costant Consumptions %%%
if (get(handles.checkbox09,'Value') == get(hObject,'Max'))
    p_MW=ones(1,length(Coil_power)*n)*peak_MW;
else p_MW=zeros(1,length(Coil_power)*n);
end
end

```

```

if (get(handles.checkbox7,'Value') == get(hObject,'Max'))
    p_earth=ones(1,length(Coil_power)*n)*handles.peak_earth;
else p_earth=zeros(1,length(Coil_power)*n);
end

if (get(handles.checkbox10,'Value') == get(hObject,'Max'))
    p_Magn=ones(1,length(Coil_power)*n)*handles.peak_Magn;
else p_Magn=zeros(1,length(Coil_power)*n);
end

if (get(handles.checkbox6,'Value') == get(hObject,'Max'))
    p_Sun=ones(1,length(Coil_power)*n)*handles.peak_Sun;
else p_Sun=zeros(1,length(Coil_power)*n);
end

if (get(handles.checkbox3,'Value') == get(hObject,'Max'))
    r_OBDH=ones(1,length(Coil_power)*n)*handles.rms_OBDH;
else r_OBDH=zeros(1,length(Coil_power)*n);
end

if (get(handles.checkbox5,'Value') == get(hObject,'Max'))
    p_TTC=ones(1,length(Coil_power)*n)*handles.peak_TTC;
else p_TTC=zeros(1,length(Coil_power)*n);
end

if (get(handles.checkbox4,'Value') == get(hObject,'Max'))
    p_PS=ones(1,length(Coil_power)*n)*handles.peak_PS;
else p_PS=zeros(1,length(Coil_power)*n);
end

C_cons=p_TTC+r_OBDH+p_Sun+p_Magn+p_earth+p_PS+p_MW;

a=length(cons_Photos);
b=length(cons_DownL);
c=length(cons_RW);
d=length(cons_Gps);
e=length(cons_Coil);
f=length(C_cons);

cons_Photos((length(Coil_power)+1):a)=[];
cons_DownL((length(Coil_power)+1):b)=[];
cons_RW((length(Coil_power)+1):c)=[];
cons_Gps((length(Coil_power)+1):d)=[];
cons_Coil((length(Coil_power)+1):e)=[];
C_cons((length(Coil_power)+1):f)=[];

handles.cons_tot=cons_Photos+cons_DownL+cons_RW+cons_Gps+cons_Coil
+C_cons; %[W]
handles.E_cons=trapz(handles.cons_tot); %[J]

figure(2)
plot(time,cons_Photos','r',time,cons_DownL','k',time,cons_RW','l',
time,cons_Gps','g',time,cons_Coil','c',time,C_cons','y',time
,handles.cons_tot','-r',time,handles.PowerT,'-b')
legend('Photos','Download
L.','R.Weels','Gps','Coil','Costants','Power Consuption','Power
Production')
title('Power Budget -Custom-')
xlabel('time [s]')

```

```

ylabel('Power [W]')
hold off
    case 5
        for ii=handles.t_on_MW:length(Coil_power)*n
            cons_MW(1,ii)=peak_MW;
        end

a=length(cons_MW);
b=length(C_cons);
c=length(cons_Coil);

cons_MW((length(Coil_power)+1):a)=[];
C_cons((length(Coil_power)+1):b)=[];
cons_Coil((length(Coil_power)+1):c)=[];

handles.cons_tot=cons_MW+cons_Coil+C_cons;
handles.E_cons=trapz(handles.cons_tot);    %[J]

figure(2)
plot(time,cons_MW','k',time,cons_Coil','c',time,C_cons','y',
time,handles.cons_tot','-r',time,handles.PowerT,'-b')
legend('M.Wheel','Coils','Costants','Power Consumption','Power
Production')
title('Power Budget -Detumbling-')
xlabel('time [s]')
ylabel('Power [W]')
hold off

    case 6
        b=length(C_cons);
        c=length(cons_Coil);

C_cons((length(Coil_power)+1):b)=[];
cons_Coil((length(Coil_power)+1):c)=[];

handles.cons_tot=cons_Coil+C_cons;
handles.E_cons=trapz(handles.cons_tot);    %[J]

figure(2)
plot(time,cons_Coil','c',time,C_cons','y',
time,handles.cons_tot','-r',time,handles.PowerT,'-b')
legend('Coils','Costants','Power Consumption','Power Production')
title('Power Budget -Nominal Orbit-')
xlabel('time [s]')
ylabel('Power [W]')
hold off

end

handles.matrici.Mdati=[(1:length(handles.Input))',handles.VocT_new
/1000,handles.Isc_new/1000,handles.Vmp_new/1000,handles.Imp_new/10
00,handles.flag(:,1),handles.flag(:,2),handles.flag(:,3),handles.f
lag(:,4),handles.PowerT,handles.cons_tot'];

guidata(hObject,handles)

```

A.3 Funzione esterna tabella dati e salvataggio

```
function tabella_dati(handles)

%----- Main Figure -----
form1=figure('Position',[400,300,600,400],'Name','OUTPUT DATA
',...
    'NumberTitle','off','Color',[0.941 0.941 0.941],...
    'Resize','off','WindowStyle','modal');

%----- Table -----
cnames=
{'Time','Voc','Isc','Vmp','Imp','Relays (1)','Relays (2)','Relays (3)
','Relays (4)','Power Production','Power Consumption'};
Mdati=handles.matrici.Mdati;

table= uitable('Data',Mdati,'ColumnName',cnames,...
    'Parent',form1,'Position',[10 50 580 320]);

%----- Buttons -----
Esporta_Excel= uicontrol(form1,'Style','pushbutton',...
    'Position',[20,10,70,25],'String','Export .xls',...
    'Callback',@esporta_excel_plot);

Esporta_Txt= uicontrol(form1,'Style','pushbutton',...
    'Position',[110,10,70,25],...
    'String','Export .txt','Callback',@esporta_txt_plot);

Esporta_Mat= uicontrol(form1,'Style','pushbutton',...
    'Position',[200,10,70,25],...
    'String','Export .mat','Callback',@esporta_matlab_plot);

chiudi =
uicontrol(form1,'Style','pushbutton','Position',[290,10,70,25],...
    'String','Close','Callback',@chiudi_plot);

%----- FUNZIONE PULSANTE ESPORTA EXCEL -----
function esporta_excel_plot(hObject,eventdata)
    [file,path] = uiputfile('dati.xls','Save file name');
    if file==0
        return
    end
    pathfile=joinseq(path,file);
    xlswrite(pathfile, Mdati);
end

%----- FUNZIONE PULSANTE ESPORTA TXT -----
function esporta_txt_plot(hObject,eventdata)
    [file,path] = uiputfile('dati.txt','Save file name');
    if file==0
        return
    end
    pathfile=joinseq(path,file);
    dlmwrite(pathfile, Mdati, 'delimiter',
'\t','precision',6,'newline', 'pc');
end

%----- FUNZIONE PULSANTE ESPORTA MATLAB -----
function esporta_matlab_plot(hObject,eventdata)
    [filename, pathname] =uiputfile('*.mat','Save file name');
    if filename==0
```

```

        return
    end
    file=strcat(pathname,filename);
    save(file,'Mdati');
end

%----- FUNZIONE PULSANTE ESPORTA CHIUDI -----
function chiudi_plot(hObject,eventdata)
    close
end
end

function SAVE_PB_Callback(hObject, eventdata, handles)
% hObject    handle to SAVE_PB (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)

tabella_dati(handles)

guidata(hObject,handles)

```

A.4 Funzione Default Values

```
function Def_V_Callback(hObject, eventdata, handles)
% hObject      handle to Def_V (see GCBO)
% eventdata    reserved - to be defined in a future version of
MATLAB
% handles      structure with handles and user data (see GUIDATA)

%%MAIN
set(handles.edit1,'String','4');
set(handles.edit2,'String','14');
set(handles.edit3,'String','4');
set(handles.edit4,'String','433.1');
set(handles.edit5,'String','450.4');
set(handles.edit6,'String','2241');
set(handles.edit7,'String','2578');
set(handles.edit8,'String','26.31');
set(handles.edit9,'String','9');
set(handles.edit10,'String','0.23679');
set(handles.edit11,'String','-6');

%%AMSC
set(handles.editt01,'String','15');
set(handles.editt02,'String','3');
set(handles.editt03,'String','0.01');
set(handles.editt04,'String','1000');
set(handles.editt05,'String','125');
set(handles.editt06,'String','5');

%%HSTX
set(handles.editt017,'String','1000-500');
set(handles.editt018,'String','18-20');

%%Times
val12=('whole orbit');
set(handles.editt07,'String',val12);
set(handles.editt09,'String',val12);
set(handles.editt011,'String',val12);
set(handles.editt013,'String','1800');
set(handles.editt015,'String','200');
set(handles.editt019,'String',val12);
set(handles.editt021,'String',val12);
set(handles.editt023,'String',val12);

%%Powers
set(handles.editt08,'String','0.5');
set(handles.editt010,'String','2');
set(handles.editt012,'String','4');
set(handles.editt014,'String','10');
set(handles.editt016,'String','10');
set(handles.editt020,'String','5');
set(handles.editt022,'String','2');
set(handles.editt024,'String','2');
set(handles.editt1MW,'String','10');
```


A.5 Funzioni esterne per il plot delle potenze

A.5.1 Potenza Prodotta (PP)

```
function plot_PP(handles)
PowerP=handles.PowerT;

cla;
plot(1:length(handles.Input),PowerP,'-b');
title('Power Production')
xlabel('time [s]')
ylabel('Power [W]')

set(handles.text4,'String','Energy Produced [J]');
set(handles.text3,'String',handles.E_prod);
```

A.5.2 Potenza Consumata (PC)

```
function plot_PC(handles)
PowerC=handles.cons_tot;

cla;
plot(1:length(handles.Input),PowerC,'-r');
title('Power Consumption')
xlabel('time [s]')
ylabel('Power [W]')

set(handles.text4,'String','Energy Consumed [J]');
set(handles.text3,'String',handles.E_cons);
```

A.5.3 Power Budget (PB)

```
function plot_PB(handles)
PowerP=handles.PowerT;
PowerC=handles.cons_tot;

cla;
plot(1:length(handles.Input),PowerC,'-r',1:length(handles.Input),PowerP,'-b');
title('Power Budget')
xlabel('time [s]')
ylabel('Power [W]')

set(handles.text4,'String','Margin of energy');
set(handles.text3,'String',(handles.E_prod-handles.E_cons));
```


A.6 Funzione Reset

```
function initialize_gui(handles)
num=0;

handles.n_pan = num;           %
handles.n_celle = num;        %
handles.n_stringhe = num;     %
handles.Imp = num;            %
handles.Isc = num;            %   INITIALIZATION LINES
handles.Vmp = num;            %
handles.Voc = num;            %   (Solar array)
handles.cell_surf = num;      %
handles.dIsc_at_dT = num;     %
handles.Isc_at_dT = num;     %
handles.dVoc_at_dT = num;    %
handles.start_mission = num;

handles.n_Photos=num;
handles.Filters=num;
handles.s_shot=num;
handles.op_range=num;
handles.t_op_P=num;
handles.peak_Photos=num;

handles.t_on_DownL=num;
handles.peak_HSTX=num;

handles.peak_Sun=num;
handles.peak_earth=num;
handles.peak_Magn=num;
handles.t_prep_Gps=num;
handles.peak_Gps=num;
handles.t_op_RW=num;
handles.rms_MW=num;

handles.rms_OBDH=num;
handles.peak_PS=num;
handles.peak_TTC=num;

handles.Input=[];
handles.PowerT=[];
handles.cons_tot=[];
handles.matrici.Mdati=[];

set(handles.popupmenu1, 'Value', 1);
set(handles.popupmenu_01, 'Value', 1);

set(handles.edit1, 'String', '0');
set(handles.edit2, 'String', '0');
set(handles.edit3, 'String', '0');
set(handles.edit4, 'String', '0');
set(handles.edit5, 'String', '0');
set(handles.edit6, 'String', '0');
set(handles.edit7, 'String', '0');
set(handles.edit8, 'String', '0');
set(handles.edit9, 'String', '0');
set(handles.edit10, 'String', '0');
set(handles.edit11, 'String', '0');
```

```

set(handles.editt01,'String','0');
set(handles.editt02,'String','0');
set(handles.editt03,'String','0');
set(handles.editt04,'String','0');
set(handles.editt05,'String','0');
set(handles.editt06,'String','0');

set(handles.editt07,'String','0');
set(handles.editt08,'String','0');
set(handles.editt09,'String','0');
set(handles.editt10,'String','0');
set(handles.editt11,'String','0');
set(handles.editt12,'String','0');
set(handles.editt13,'String','0');
set(handles.editt14,'String','0');
set(handles.editt15,'String','0');
set(handles.editt16,'String','0');
set(handles.editt17,'String','0');
set(handles.editt18,'String','0');
set(handles.editt19,'String','0');
set(handles.editt20,'String','0');
set(handles.editt21,'String','0');
set(handles.editt22,'String','0');
set(handles.editt23,'String','0');
set(handles.editt24,'String','0');

set(handles.text3,'String',' ');
set(handles.text4,'String',' ');

check_reset(handles)

cla;
reset(gca);
reset(legend);

guidata(handles.figure1,handles);

function RESET_PB_Callback(hObject, eventdata, handles)
% hObject    handle to RESET_PB (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)
initialize_gui(handles);

```

A.7 Funzione Select e setting checkbox

```
function pushbutton_select_Callback(hObject, eventdata, handles)
% hObject      handle to pushbutton_read (see GCBO)
% eventdata    reserved - to be defined in a future version of
MATLAB
% handles      structure with handles and user data (see GUIDATA)
check_set(handles)
```

```
function check_reset(handles)
minVal = get(handles.checkbox1, 'Min');
set(handles.checkbox1, 'Value', minVal);
minVal = get(handles.checkbox2, 'Min');
set(handles.checkbox2, 'Value', minVal);
minVal = get(handles.checkbox3, 'Min');
set(handles.checkbox3, 'Value', minVal);
minVal = get(handles.checkbox4, 'Min');
set(handles.checkbox4, 'Value', minVal);
minVal = get(handles.checkbox5, 'Min');
set(handles.checkbox5, 'Value', minVal);
minVal = get(handles.checkbox6, 'Min');
set(handles.checkbox6, 'Value', minVal);
minVal = get(handles.checkbox7, 'Min');
set(handles.checkbox7, 'Value', minVal);
minVal = get(handles.checkbox8, 'Min');
set(handles.checkbox8, 'Value', minVal);
minVal = get(handles.checkbox9, 'Min');
set(handles.checkbox9, 'Value', minVal);
minVal = get(handles.checkbox10, 'Min');
set(handles.checkbox10, 'Value', minVal);
minVal = get(handles.checkbox11, 'Min');
set(handles.checkbox11, 'Value', minVal);
```

```
function check_set(handles)
I=get(handles.popupmenu1, 'Value');
```

```
switch I
case 1
    check_reset (handles)
    maxVal = get(handles.checkbox1, 'Max');
    set(handles.checkbox1, 'Value', maxVal);
    maxVal = get(handles.checkbox3, 'Max');
    set(handles.checkbox3, 'Value', maxVal);
    maxVal = get(handles.checkbox4, 'Max');
    set(handles.checkbox4, 'Value', maxVal);
    maxVal = get(handles.checkbox5, 'Max');
    set(handles.checkbox5, 'Value', maxVal);
    maxVal = get(handles.checkbox6, 'Max');
    set(handles.checkbox6, 'Value', maxVal);
    maxVal = get(handles.checkbox7, 'Max');
    set(handles.checkbox7, 'Value', maxVal);
    maxVal = get(handles.checkbox8, 'Max');
    set(handles.checkbox8, 'Value', maxVal);
    maxVal = get(handles.checkbox09, 'Max');
    set(handles.checkbox09, 'Value', maxVal);
    maxVal = get(handles.checkbox9, 'Max');
    set(handles.checkbox9, 'Value', maxVal);
    maxVal = get(handles.checkbox10, 'Max');
    set(handles.checkbox10, 'Value', maxVal);
    maxVal = get(handles.checkbox11, 'Max');
    set(handles.checkbox11, 'Value', maxVal);
```

case 2

```
check_reset (handles)
maxVal = get(handles.checkbox2, 'Max');
set(handles.checkbox2, 'Value', maxVal);
maxVal = get(handles.checkbox3, 'Max');
set(handles.checkbox3, 'Value', maxVal);
maxVal = get(handles.checkbox4, 'Max');
set(handles.checkbox4, 'Value', maxVal);
maxVal = get(handles.checkbox5, 'Max');
set(handles.checkbox5, 'Value', maxVal);
maxVal = get(handles.checkbox6, 'Max');
set(handles.checkbox6, 'Value', maxVal);
maxVal = get(handles.checkbox7, 'Max');
set(handles.checkbox7, 'Value', maxVal);
maxVal = get(handles.checkbox09, 'Max');
set(handles.checkbox09, 'Value', maxVal);
maxVal = get(handles.checkbox10, 'Max');
set(handles.checkbox10, 'Value', maxVal);
maxVal = get(handles.checkbox11, 'Max');
set(handles.checkbox11, 'Value', maxVal);
```

case 3

```
check_reset (handles)
maxVal = get(handles.checkbox1, 'Max');
set(handles.checkbox1, 'Value', maxVal);
maxVal = get(handles.checkbox2, 'Max');
set(handles.checkbox2, 'Value', maxVal);
maxVal = get(handles.checkbox3, 'Max');
set(handles.checkbox3, 'Value', maxVal);
maxVal = get(handles.checkbox4, 'Max');
set(handles.checkbox4, 'Value', maxVal);
maxVal = get(handles.checkbox5, 'Max');
set(handles.checkbox5, 'Value', maxVal);
maxVal = get(handles.checkbox6, 'Max');
set(handles.checkbox6, 'Value', maxVal);
maxVal = get(handles.checkbox7, 'Max');
set(handles.checkbox7, 'Value', maxVal);
maxVal = get(handles.checkbox8, 'Max');
set(handles.checkbox8, 'Value', maxVal);
maxVal = get(handles.checkbox09, 'Max');
set(handles.checkbox09, 'Value', maxVal);
maxVal = get(handles.checkbox9, 'Max');
set(handles.checkbox9, 'Value', maxVal);
maxVal = get(handles.checkbox10, 'Max');
set(handles.checkbox10, 'Value', maxVal);
maxVal = get(handles.checkbox11, 'Max');
set(handles.checkbox11, 'Value', maxVal);
```

case 4

```
check_reset (handles)
```

```
case 5
    check_reset (handles)
    maxVal = get(handles.checkbox3, 'Max');
    set(handles.checkbox3, 'Value', maxVal);
    maxVal = get(handles.checkbox4, 'Max');
    set(handles.checkbox4, 'Value', maxVal);
    maxVal = get(handles.checkbox5, 'Max');
    set(handles.checkbox5, 'Value', maxVal);
    maxVal = get(handles.checkbox6, 'Max');
    set(handles.checkbox6, 'Value', maxVal);
    maxVal = get(handles.checkbox09, 'Max');
    set(handles.checkbox09, 'Value', maxVal);
    maxVal = get(handles.checkbox10, 'Max');
    set(handles.checkbox10, 'Value', maxVal);
    maxVal = get(handles.checkbox11, 'Max');
    set(handles.checkbox11, 'Value', maxVal);
```

```
case 6
    check_reset (handles)
    maxVal = get(handles.checkbox3, 'Max');
    set(handles.checkbox3, 'Value', maxVal);
    maxVal = get(handles.checkbox4, 'Max');
    set(handles.checkbox4, 'Value', maxVal);
    maxVal = get(handles.checkbox5, 'Max');
    set(handles.checkbox5, 'Value', maxVal);
    maxVal = get(handles.checkbox6, 'Max');
    set(handles.checkbox6, 'Value', maxVal);
    maxVal = get(handles.checkbox7, 'Max');
    set(handles.checkbox7, 'Value', maxVal);
    maxVal = get(handles.checkbox09, 'Max');
    set(handles.checkbox09, 'Value', maxVal);
    maxVal = get(handles.checkbox10, 'Max');
    set(handles.checkbox10, 'Value', maxVal);
    maxVal = get(handles.checkbox11, 'Max');
    set(handles.checkbox11, 'Value', maxVal);
```

```
end
```


Bibliografia

1. <http://www.mathworks.com/>. [Online]
2. **Graziani, A.** "*Progettazione, realizzazione e prove della stazione di terra satellitare Alma Mater Ground Station (AMGS)*". Forlì, A.A. 2002/03.
3. **Accurso, D.** *Progetto, "Implementazione e prove del canale di downlink in banda S per la stazione di terra satellitare AMGS. Forlì"*, A.A. 2002/03.
4. "*ALMASat-EO Attitude Determination Algorithms: Evaluation, Implementation And Numerical Simulation*", **Bosco M., V. Fabbri, P. Tortora.** 8th International ESA Conference On Guidance, Navigation and Control, Karlovy Vary, Czech Republic, 2011.
5. **Armando Bellini, Stefano Bifaretti, Vincenzo Iacovone, Cristina Cornaro.** "*Simplified Model of a Photovoltaic Module*", Applied Electronics, AE, Pilsen, 9-10 September 2009
6. **AzurSpace.** "3G - 27% CURVE PARAMETERS AND DATASHEET", 2007.
7. "*ALMASat-1 POWER SYSTEM TEST CAMPAIGN*", **Schirone, F. Antonini A. Tambini A. Graziani A. Palli P. Tortora M. Macellari A. Schiaratura L.** Proceedings of the XX Congresso Nazionale AIDAA : s.n., 29 June – 3 July 2009.
8. **Larson, Wertz J.R. and W.J.** "*SPACE MISSION ANALYSIS AND DESIGN*". Microcosm Press and Kluwer Academic Publishers, 1978.

