

POLITECNICO DI TORINO

I FACOLTÀ DI INGEGNERIA

Corso di Laurea in Ingegneria Aerospaziale

Tesi di Laurea di primo livello

**Dimensionamento e sviluppo dei sistemi
di variazione e controllo di assetto per
satellite modulare AraMiS**



Relatori:

Prof. Ing. Leonardo Reyneri

Prof.ssa Ing. Manuela Battipede

Candidato:

Danilo Muru

Dicembre 2011

Per aspera sic itur ad astra

Alla mia famiglia

Per aver sempre creduto in me;

Ai miei amici,

Per avermi reso quello che sono;

Ai miei compagni,

Senza i quali non ce l'avrei fatta;

INDICE

1.	La navigazione spaziale	- 10 -
1.1.	La fase di lancio	- 10 -
1.2.	Principali manovre per il controllo di assetto	- 11 -
1.3.	Sistemi di riferimento	- 13 -
1.3.1.	Orbit frame.....	- 13 -
1.3.2.	<i>Body frame</i> (Assi corpo)	- 14 -
1.1.1.	Earth-centered inertial (ECI)	- 15 -
1.1.2.	Earth-centered earth fixed (ECEF)	- 16 -
1.1.3.	Angoli di Eulero	- 17 -
1.1.4.	Quaternioni	- 19 -
1.1.5.	Parametri di Eulero	- 19 -
1.2.	Trasformazioni tra sistemi di riferimento	- 21 -
1.2.1.	Da ECEF a ECI	- 21 -
1.2.2.	Da ECI a Orbit Frame	- 21 -
1.2.3.	Da Orbit Frame a Body Frame	- 23 -
2.	Sistemi per il controllo d'assetto	- 24 -
2.1.	Tecniche di controllo passive	- 24 -
2.1.1.	Controllo a gradiente di gravità.....	- 24 -
2.1.2.	Controllo tramite magnete permanente	- 26 -
2.1.3.	Controllo mediante <i>spin stabilization</i>	- 27 -
2.2.	Tecniche di controllo attive.....	- 28 -
2.2.1.	Sistemi a espulsione di massa	- 28 -
2.2.2.	Sistemi di tipo rotorico	- 29 -
2.2.3.	Sistemi di tipo "Magnetic Torquer"	- 31 -
2.3.	I diversi tipi di sensori.....	- 33 -
2.3.1.	Sensori di sole	- 33 -
2.3.2.	Sensori di stelle	- 33 -
2.3.3.	Sensori di orizzonte	- 34 -
2.3.4.	Magnetometri.....	- 34 -
2.3.5.	Ricevitori GPS	- 34 -
2.3.6.	Giroscopi	- 34 -
3.	il Satellite	- 35 -
3.1.	PiCPoT.....	- 35 -
3.2.	AraMiS	- 38 -

4.	Scelte progettuali.....	- 43 -
4.1.	Ambiente Operativo	- 43 -
4.2.	Effetti di disturbo ambientali	- 46 -
4.2.1.	Il gradiente di gravità	- 46 -
4.2.2.	Il campo magnetico.....	- 47 -
4.2.3.	La radiazione solare	- 47 -
4.2.4.	L'aerodinamica	- 48 -
4.2.5.	Calcolo delle coppie di disturbo	- 49 -
4.3.	Schede Tecniche	- 49 -
4.3.1.	Satellite	- 49 -
4.3.2.	Parametri di missione	- 51 -
4.3.3.	Equipaggiamento del satellite: sensori	- 52 -
4.3.4.	Equipaggiamento del satellite: attuatori	- 53 -
5.	Dimensionamento dei sistemi di controllo attivi	- 54 -
5.1.	Magnetic Torquer	- 54 -
5.1.1.	Fisica del solenoide	- 54 -
5.1.2.	Pilotaggio del solenoide	- 55 -
5.1.3.	Reazione a coppie esterne	- 58 -
5.1.4.	Dimensionamento del solenoide.....	- 59 -
5.2.	Ruote d'inerzia	- 63 -
5.2.1.	Fisica delle ruote d'inerzia	- 63 -
5.2.2.	Pilotaggio delle ruote d'inerzia	- 64 -
5.2.3.	Dimensionamento del sistema.....	- 65 -
6.	Programmi di controllo d'assetto	- 69 -
6.1.	Considerazioni Preliminari	- 69 -
6.2.	Attitude control and determination system (ACDS).....	- 70 -
6.3.	Spin e De-spin	- 70 -
6.3.1.	Algoritmo di spin	- 72 -
6.4.	Rotazione del satellite	- 76 -
6.4.1.	Individuazione del problema	- 77 -
6.4.2.	Rotazione in tempo minimo	- 79 -
6.4.3.	Algoritmo di rotazione	- 82 -
6.5.	Puntamento inerziale	- 83 -
6.5.1.	Algoritmo di puntamento inerziale	- 83 -

6.6.	Ground tracking	- 85 -
6.6.1.	Geometria del problema pseudo-bidimensionale	- 85 -
6.6.2.	Geometria del problema tridimensionale	- 89 -
7.	Simulazione dei programmi.....	- 93 -
7.1.	Simulazione di spin	- 93 -
7.1.1.	Imposizione spin da condizioni iniziali nulle	- 93 -
7.1.2.	Arresto di un satellite a partire da spin non nullo	- 96 -
7.2.	Simulazione di rotazione	- 98 -
7.2.1.	Rotazione con $Esns = E0$	- 99 -
7.2.2.	Rotazione con $Eat = E0$	- 101 -
7.2.3.	Rotazione generica.....	- 103 -
7.3.	Simulazione di puntamento inerziale	- 106 -
7.4.	Simulazione di ground tracking	- 110 -
7.4.1.	Politecnico di Torino	- 110 -
7.4.2.	Boston, MIT	- 115 -
7.4.3.	Tokio.....	- 118 -
7.4.4.	Reykjavik	- 121 -
7.4.5.	Cape Town	- 124 -
8.	Appendici	- 127 -
	Appendice A: File di matlab	- 127 -
A.1.	Attitude control system (Att_CtrlSystem.m)	- 127 -
A.2.	Spin sub system (Att_Sub_Spi.m)	- 128 -
A.3.	Rotation sub system (Att_Sub_Rot.m).....	- 130 -
A.4.	Inertial pointing sub system (Att_Sub_IFix.m)	- 131 -
A.5.	Ground tracking sub system (Att_Sub_Tra.m).....	- 132 -
	Appendice B: Soubroutine di caricamento dati:	- 139 -
B.1.	Load Data	- 139 -
B.2.	Load Sensor	- 140 -
	Appendice C: Functions di matlab	- 143 -
C.1.	Rotazione ECI-ORBIT	- 143 -
C.2.	Rotazione ECEF-ECI.....	- 144 -
C.3.	Rotazione ORBIT-BODY	- 144 -
C.4.	Derivazione angoli di Eulero	- 145 -
C.5.	Rotazione angoli di Eulero	- 146 -
C.6.	Angoli di Eulero da terna di assi orientata	- 148 -
C.7.	Grafici dell'orbita (Print_Orbit)	- 151 -

Appendice D: File .dat.....	- 153 -
D.1. DataAttitude.dat.....	- 153 -
D.2. DataMechanics.dat.....	- 153 -
D.3. DataOrbit.dat	- 154 -
D.4. DataSensor.dat.....	- 154 -
Appendice E: Altro	- 155 -
E.1. Maxton EC 32 Flat motor Datasheet.....	- 155 -
9. Bibliografia	- 156 -

1. LA NAVIGAZIONE SPAZIALE

1.1. LA FASE DI LANCIO [14]

La prima fase della missione di un satellite è costituita dal lancio e dalla successiva immissione in orbita. Vengono pertanto impiegati dei razzi vettore che hanno il compito di fornire la spinta necessaria per abbandonare la superficie terrestre per raggiungere la quota, la velocità e la direzione richiesta per l'immissione in orbita. Il satellite a questo punto viene espulso e, seguendo una traiettoria balistica opportunamente calcolata, si posiziona approssimativamente lungo la traiettoria prevista.

A questo punto occorre impiegare sistemi di traslazione che permettano di correggere gli errori balistici e portare con maggiore precisione il satellite lungo l'orbita prestabilita. Per far sì che ciò avvenga è necessario che il satellite sappia con precisione sia dove si trova che dove dovrebbe trovarsi. Si rende quindi necessaria la comunicazione con le stazioni di terra che forniscono un *tracking* del satellite, verificando posizione e velocità tramite tecniche radio e radar. In alternativa il satellite deve essere equipaggiato con i parametri di missione in modo da permettere che questi raggiunga l'orbita prestabilita in modo autonomo.

In queste situazioni, e in tutte quelle dove si preferisce che il satellite abbia un controllo dei parametri indipendenti dalle trasmissioni provenienti da terra, il satellite è dotato di alcuni sensori che permettono di determinarne la posizione spaziale e le componenti di velocità e accelerazione lungo i propri assi.

1.2. PRINCIPALI MANOVRE PER IL CONTROLLO DI ASSETTO [5]

Un satellite in orbita può generalmente compiere due tipi di operazioni per variare le sue condizioni operative.

La prima è una manovra detta di *shifting*, in altre parole una traslazione che consente al corpo di spostarsi variare la sua orbita, sia in termini di ampiezza (altezza dal suolo) sia in termini di inclinazione, ma anche di conferire al satellite una maggiore o minore velocità. Tale operazione si svolge usualmente con l'impiego di *thrusters* multipli a getto di gas o di ioni. Tale tipo di attuatore si basa sulla conservazione della quantità di moto. Infatti, un'espulsione di massa (gas) con una certa velocità genera una variazione di velocità sul satellite, in funzione della massa, in direzione uguale e contraria.

$$\sum Q = m_g v_g + m_s v_s = 0$$

$$v_s = -v_g \frac{m_g}{m_s}$$

La seconda manovra che un satellite può compiere prende il nome di *slewing*. Si tratta di una rotazione dello stesso attorno al suo centro di massa. Le rotazioni sono necessarie per permettere al satellite di puntare una delle sue facce in una direzione desiderata. I motivi sono i più disparati, ad esempio dal puntamento dell'antenna per telecomunicazioni all'esposizione di un determinato pannello solare verso il sole o proteggere la strumentazione sensibile dai raggi cosmici. Solitamente tali manovre non sono rapide e i sistemi impiegati per tali rotazioni sono ruote d'inerzia (*momentum wheels, MW*), ruote di reazione (*reaction wheels, RW*) attuatori di momento magnetici (*Magnetic Torquers, MT*). Per

manovre in cui è richiesta una rotazione più rapida, vengono solitamente impiegati i thruster, che forniscono però una precisione minore sull'angolo di rotazione. La scelta del sistema da impiegare varia dunque in funzione delle velocità di rotazione necessarie a ottenere l'assetto desiderato.

I sistemi impiegati per lo *slewing* sono utilizzati anche per smorzare attivamente eventuali momenti rotatori di disturbo al fine di mantenere l'assetto corretto.

La correzione di assetto può essere eseguita sia in modo passivo sia in modo attivo:

Nel primo caso si parla infatti di compensazione di assetto e sono impiegati sistemi quali: gradiente di gravità, stabilizzazione di spin e magneti permanente.

Nel secondo caso invece si ha una variazione comandata dell'assetto e vengono impiegati gli stessi sistemi sopraindicati per la manovra di slewing (RW, MW, MT, Thrusters).

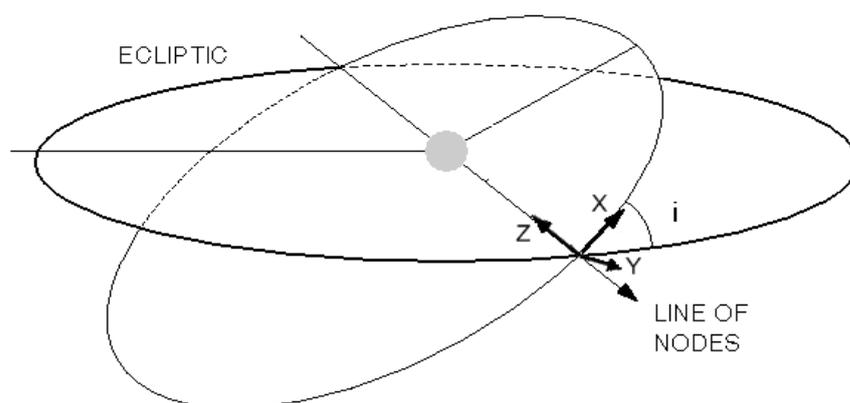
1.3. SISTEMI DI RIFERIMENTO [6] [10] [8]

Lo studio del moto di un satellite in orbita attorno alla terra è piuttosto complesso, in quanto si tratta di un moto con sei gradi di libertà: tre coordinate per individuare il centro del velivolo nello spazio altre tre per determinare l'assetto attorno al centro. Si rende quindi necessario definire dei sistemi di riferimento per definire la posizione del satellite ad ogni istante e poter quindi impartire comandi e ricevere dati posizionali univoci.

1.3.1. Orbit frame

Con *Orbit frame* si intende un sistema di assi avente origine nel centro di massa del satellite. Gli assi sono solidali all'orbita percorsa dal satellite, con l'asse Z_O rivolto verso il centro dell'orbita, l'asse X_O tangente all'orbita e rivolto in direzione del moto e l'asse Y_O ortogonale ai due precedenti tale da rendere la terna destrorsa.

Lo scostamento tra Orbit frame e Body frame si misura tramite tre rotazioni attorno agli assi. Questi angoli di scostamento rappresentano l'assetto del satellite, più precisamente lo scostamento dalla condizione zero rappresentata dall'Orbit frame.



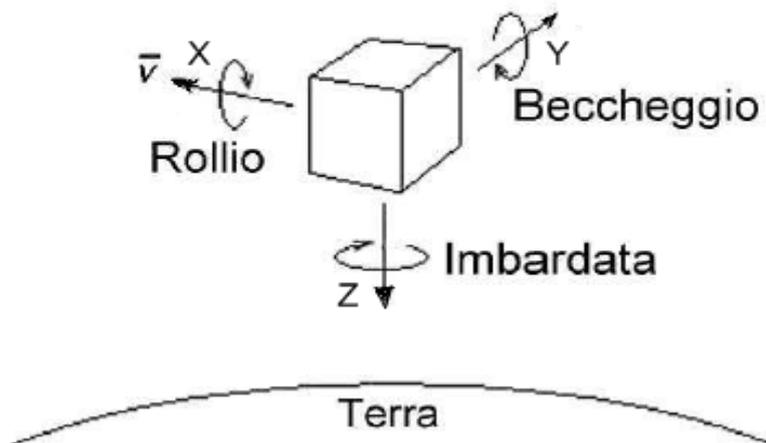
1.1 Riferimento orbitale

1.3.2. *Body frame* (Assi corpo)

Iniziamo a definire un sistema di assi corpo (*Body Frame*). Con tale dicitura si intende una terna destrorsa di assi cartesiani centrata nel centro di massa del satellite, con l'asse X_B nella direzione del moto, l'asse Z_B rivolto verso terra e l'asse Y_B ortogonale al piano $X_B Z_B$ tale che la terna sia destrorsa.

Si definiscono inoltre i moti di rollio (*roll*) come rotazione antioraria attorno all'asse X_B , beccheggio (*pitch*) come rotazione antioraria rispetto all'asse Y_B e imbardata (*yaw*) come rotazione antioraria rispetto all'asse Z_B .

I tre assi X_B , Y_B , Z_B assumono quindi rispettivamente la denominazione di asse di rollio, di beccheggio e di imbardata.

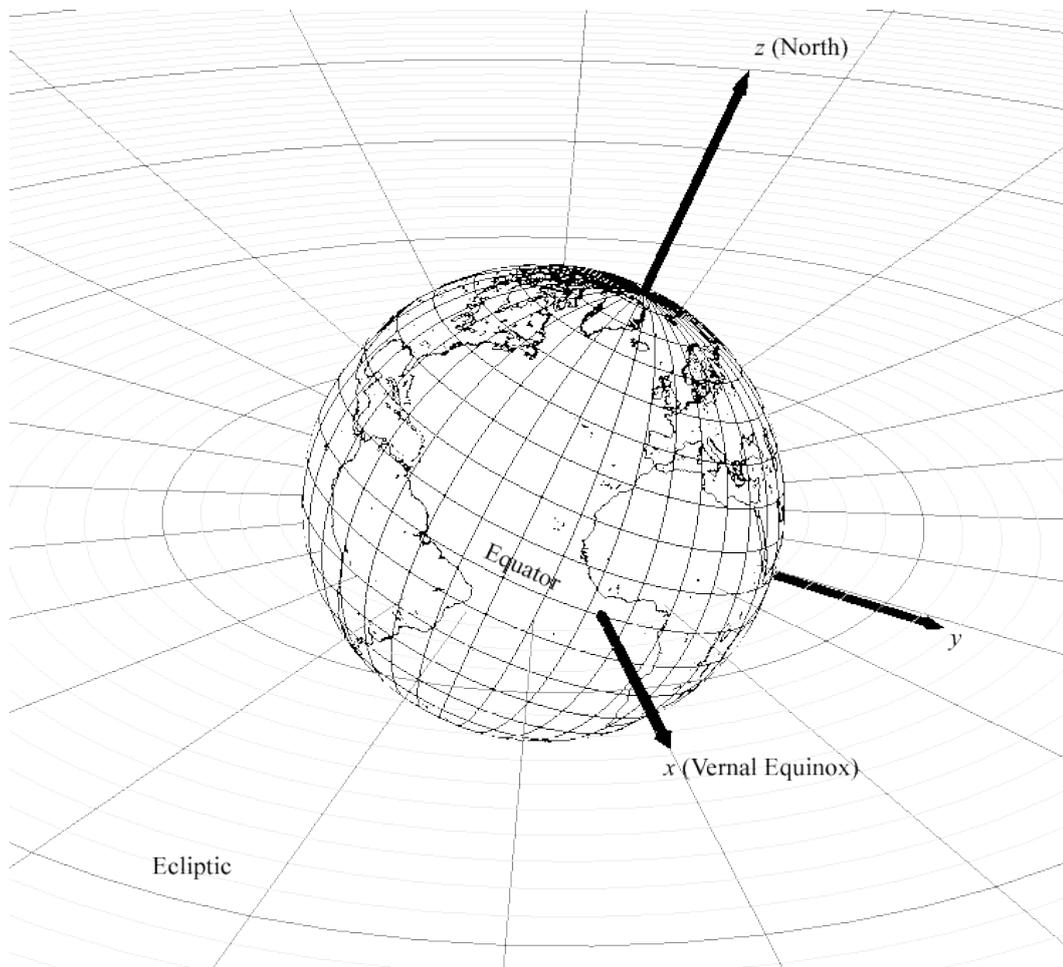


1.2 Assi corpo di un satellite e rispettive rotazioni

1.1.1. Earth-centered inertial (ECI)

L'ECI frame, anche chiamato *Inertial Geocentric Reference Frame* è un sistema di riferimento inerziale centrato sulla terra, con asse Z_I diretto verso il polo nord. Il riferimento è di tipo inerziale, significa che è orientato rispetto alle stelle fisse. L'asse X_I è orientato verso Aries (Equinozio di primavera), e l'asse Y_I ortogonale a entrambi tale che la terna sia destrorsa.

Questo sistema di assi è solitamente impiegato per i dati di navigazione orbitale e per lo studio del moto di un corpo orbitante attorno alla terra.



1.3 Assi ECI

1.1.2. Earth-centered earth fixed (ECEF)

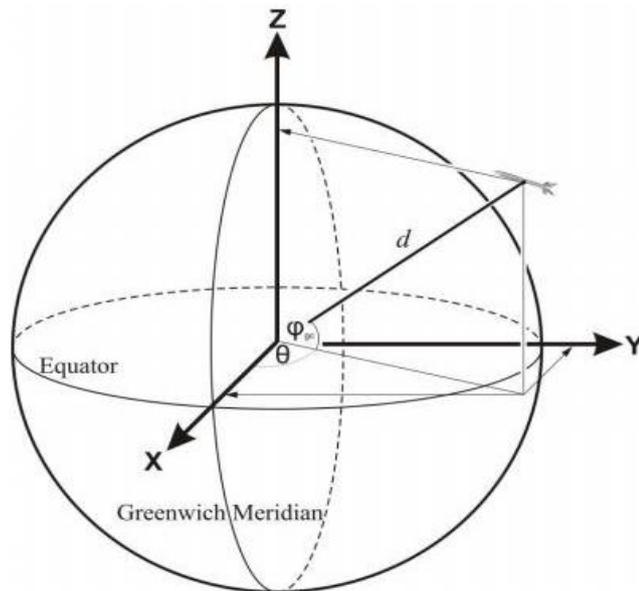
Questo riferimento terrestre è molto utilizzato per l'analisi e la rielaborazione di dati provenienti dai satelliti. Definisce la posizione di un generico punto nello spazio attraverso tre coordinate riferite al centro di massa dell'oggetto considerato.

L'ECEF ha origine nel centro della terra e gli assi sono fissi e solidali alla stessa. L'asse Z_E ha direzione del polo nord, l'asse X_E punta in direzione del meridiano di Greenwich e giace sul piano equatoriale, così come l'asse Y_E che è ortogonale ai precedenti e orientato in modo da rendere la terna destrorsa.

Si introducono inoltre due angoli caratteristici: λ , longitudine e φ latitudine, come mostrato nella figura seguente.

Il sistema ECEF si trova in rotazione rispetto al sistema ECI con velocità angolare $\Omega_E = 7.272 \cdot 10^{-5}$ rad/s rispetto all'asse Z_E , che è coincidente con l'asse Z_I . Ω_E è la velocità di rotazione della terra rispetto a un sistema inerziale.

La terra possiede anche una velocità di rivoluzione attorno al sole, $\Omega_R = 2 \cdot 10^{-7}$ rad/s, che verrà però trascurata nel resto della trattazione.



1.4 Coordinate di un punto nel sistema ECEF

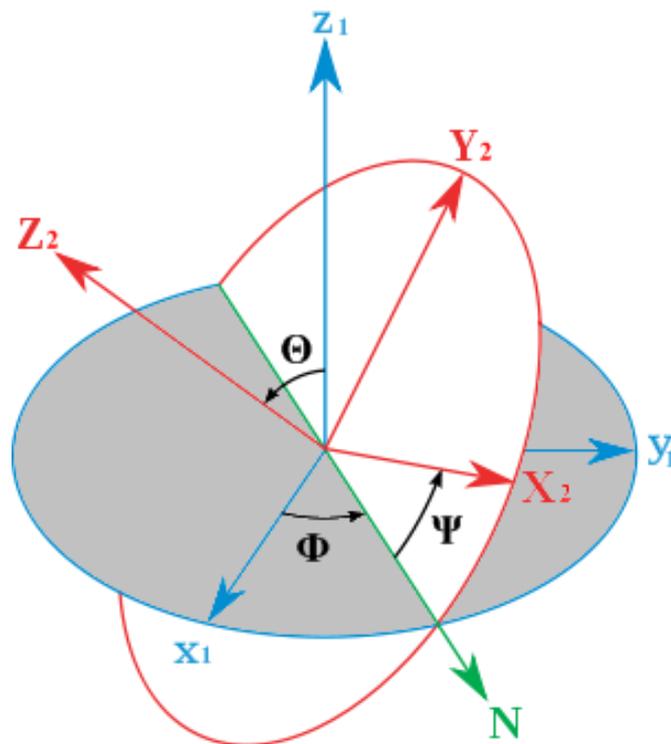
1.1.3. Angoli di Eulero

Gli angoli di Eulero (Ψ , Θ , Φ) sono tre quantità indipendenti utili a definire la posizione di un generico sistema di riferimento rispetto a un altro, non necessariamente inerziale. Tutte le terne di assi sono considerate cartesiane e destrorse.

Questi angoli definiscono una sequenza di rotazioni (Ψ , Θ , Φ) non invertibile, e che quindi non costituisce una base lineare. Tale trasformazione permette tuttavia di esprimere le componenti di un vettore nell'uno o nell'altro sistema di riferimento considerato.

I tre angoli sono determinati dati due sistemi Σ_1 e Σ_2 nel seguente modo (vedi anche fig. 1.4):

- Θ , ($0 < \Theta < \pi$) è l'angolo compreso fra l'asse Z_1 e l'asse Z_2 .
- Φ , ($0 < \Phi < 2\pi$) è l'angolo compreso tra l'asse X_1 e la linea dei nodi, che indicheremo con N . la linea dei nodi è l'intersezione tra i piani X_1Y_1 e X_2Y_2 .



1.5 Determinazione degli angoli di Eulero

- Ψ , ($0 < \Psi < 2\pi$) è l'angolo tra la linea dei nodi N e l'asse X_2 .

E' possibile sovrapporre il sistema Σ_2 con il sistema Σ_1 mediante tre rotazioni Ψ , Θ , Φ ben definite:

- I. Rotazione antioraria di angolo Ψ , eseguita attorno all'asse Z_2 . Indichiamo con Σ'_2 il sistema così ottenuto;
- II. Rotazione antioraria di angolo Θ , eseguita attorno alla linea dei nodi (che dopo la prima rotazione coincide con X_2): questa rotazione sovrappone Z_2 con Z_1 . Indichiamo il nuovo sistema come Σ''_2
- III. Rotazione antioraria di angolo Φ intorno all'asse Z ($Z_2 = Z_1$). Tale rotazione porta a far coincidere gli assi X_2 con X_1 .

Poiché gli assi X e Z coincidono, allora anche gli assi Y coincidono in quanto i due sistemi sono entrambi ortogonali destrorsi.

Alle tre rotazioni è possibile assegnare tre matrici di rotazione, che indicheremo con R_Ψ , R_Θ , R_Φ :

$$R_\Psi = \begin{bmatrix} \cos \Psi & \sin \Psi & 0 \\ -\sin \Psi & \cos \Psi & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$R_\Theta = \begin{bmatrix} \cos \Theta & 0 & \sin \Theta \\ 0 & 1 & 0 \\ -\sin \Theta & 0 & \cos \Theta \end{bmatrix}$$

$$R_\Phi = \begin{bmatrix} \cos \Phi & \sin \Phi & 0 \\ -\sin \Phi & \cos \Phi & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Si dimostra quindi che:

$$\Sigma_1 = R * \Sigma_2$$

Dove R è la matrice di rotazione globale data dal prodotto delle tre matrici di rotazione R_Ψ , R_Θ , R_Φ nell'ordine:

$$R = R_\Psi * R_\Theta * R_\Phi$$

E' ovviamente possibile anche il passaggio inverso tra i due sistemi di riferimento utilizzando la trasposta della matrice di rotazione globale R^T :

$$\Sigma_2 = R^T * \Sigma_1$$

Con:

$$R^T = (R_\psi * R_\theta * R_\phi)^T = R_\phi^T * R_\theta^T * R_\psi^T$$

1.1.4. Quaternioni

In matematica i quaternioni sono entità introdotte da William Rowan Hamilton nel 1843, come estensione quadridimensionale dei numeri complessi.

I quaternioni costituiscono un corpo non commutativo, vale a dire che hanno tutte le proprietà usuali di un campo eccetto la commutatività del prodotto. Essi costituiscono un campo quadridimensionale complesso, scrivibile come:

$$\eta + e_1 i + e_2 j - e_3 k$$

Con η, e_1, e_2, e_3 numeri reali e i, j, k , simboli letterali.

Somma e prodotto tra quaternioni sono definiti tenendo conto della relazione:

$$i^2 = j^2 = k^2 = ijk = -1$$

1.1.5. Parametri di Eulero

I parametri di Eulero sono una parametrizzazione dell'assetto riferita ai quaternioni. Si tratta di quattro parametri non singolari, i cui vincoli sono relativamente facili da stabilire.

Una rotazione α attorno a un asse $\bar{a} = a_1 \bar{l} + a_2 \bar{m} + a_3 \bar{n}$

È ottenibile con i quaternioni come:

$$\bar{q} = \begin{bmatrix} \eta \\ e_1 \\ e_2 \\ e_3 \end{bmatrix} = \begin{bmatrix} \cos \alpha/2 \\ a_1 \cdot \sin \alpha/2 \\ a_2 \cdot \sin \alpha/2 \\ a_3 \cdot \sin \alpha/2 \end{bmatrix}$$

I quaternioni non sono tra loro indipendenti e vale la relazione:

$$\|q\| = \eta^2 + e_1^2 + e_2^2 + e_3^2 = 1$$

Presi u, v :

$$u = \begin{bmatrix} 0 \\ u \\ \end{bmatrix} = \begin{bmatrix} 0 \\ u_1 \\ u_2 \\ u_3 \end{bmatrix} \quad v = \begin{bmatrix} 0 \\ v \\ \end{bmatrix} = \begin{bmatrix} 0 \\ v_1 \\ v_2 \\ v_3 \end{bmatrix}$$

La rotazione da u a v è descritto da:

$$\begin{bmatrix} 0 \\ u_1 \\ u_2 \\ u_3 \end{bmatrix} = q_1 \otimes \begin{bmatrix} 0 \\ v_1 \\ v_2 \\ v_3 \end{bmatrix} \otimes \bar{q}_1$$

con q_1 e \bar{q}_1 i seguenti quaternioni

$$q_1 = \begin{bmatrix} \eta_1 \\ e_1 \end{bmatrix} = \begin{bmatrix} \eta_1 \\ e_{11} \\ e_{12} \\ e_{13} \end{bmatrix} \quad \bar{q}_1 = \begin{bmatrix} \eta_1 \\ -e_1 \end{bmatrix} = \begin{bmatrix} \eta_1 \\ -e_{11} \\ -e_{12} \\ -e_{13} \end{bmatrix}$$

E definendo il prodotto tra quaternioni \otimes come:

$$q_1 \otimes q_2 = \begin{bmatrix} \eta_1 \\ e_1 \end{bmatrix} \otimes \begin{bmatrix} \eta_2 \\ e_2 \end{bmatrix} = \begin{bmatrix} \eta_1 e_2 + \eta_2 e_1 + S(e_1) e_2 \\ \eta_1 \eta_2 - e_1^T e_2 \end{bmatrix}$$

E alla quale si fa riferimento alla matrice S , definita come:

$$S(e) = \begin{bmatrix} 0 & -e_3 & e_2 \\ e_3 & 0 & -e_1 \\ -e_2 & e_1 & 0 \end{bmatrix}$$

1.2. TRASFORMAZIONI TRA SISTEMI DI

RIFERIMENTO [15] [16]

Vediamo ora mediante quali rotazioni sia possibile passare da un riferimento a un altro, tra quelli sopra descritti.

1.2.1. Da ECEF a ECI

La rotazione tra questi due sistemi di riferimento è pari a quella terrestre. Un riferimento infatti è rotante rispetto all'altro attorno all'asse Z. Possiamo quindi portarci dal riferimento fisso a quello inerziale mediante una rotazione

$$\alpha = \omega_e \cdot t$$

Dove t è il tempo in secondi trascorso da quando i due sistemi erano allineati e ω_e è la velocità angolare di rotazione della terra, pari a

$$\omega_e = 7,27 \cdot 10^{-5} \text{ rad/sec}$$

Il passaggio tramite matrice di rotazione R_E^I è dunque:

$$\Sigma_F = R_E^I \cdot \Sigma_I$$

Con:

$$R_E^I = \begin{bmatrix} \cos \alpha & \sin \alpha & 0 \\ -\sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

E' possibile anche il passaggio inverso impiegando la matrice trasposta:

$$\Sigma_I = R_E^{I^T} \cdot \Sigma_E = R_I^E \cdot \Sigma_E$$

1.2.2. Da ECI a Orbit Frame

È una rotazione che ci consente di passare dal riferimento inerziale della terra a quello rotante del satellite. Dipende da alcuni parametri quali la velocità angolare del satellite ω_0 , l'inclinazione dell'orbita i_{or} e β , che rappresenta l'angolo che sussiste tra il sistema inerziale e l'orbit frame. Più in generale dall'orbita di quest'ultimo.

Abbiamo quindi una prima rotazione rispetto all'asse dei nodi tra l'equatore e il piano dell'orbita del satellite. L'angolo è i_{or} che equivale all'inclinazione dell'orbita.

Si porta poi una rotazione $\beta = \omega_o \cdot t - \beta_0$ attorno all'asse Z che porta l'asse X della terna Eci a essere sempre parallelo alla tangente all'orbita nel punto in cui si trova il satellite. Il termine β_0 rappresenta l'angolo sul piano equatoriale tra X_{ECI} e X_O quando il corpo si trova allineato con l'asse X_{ECI} .

Un'ulteriore rotazione attorno a X di $90^\circ(\pi/2)$ porta poi gli assi Z, e di conseguenza tutta la terna, a coincidere.

Si tratta in sintesi di una rotazione di angoli di Eulero del tipo YZX in cui gli angoli di eulero valgono:

$$\begin{cases} \Psi = i_{or} \\ \Theta = \beta \\ \Phi = \pi/2 \end{cases}$$

Le singole trasformazioni sono le seguenti:

$$R_{Y_{E.O}} = \begin{bmatrix} \cos i_{or} & 0 & \sin i_{or} \\ 0 & 1 & 0 \\ -\sin i_{or} & 0 & \cos i_{or} \end{bmatrix}$$

$$R_{Z_{E.O}} = \begin{bmatrix} \cos \beta & \sin \beta & 0 \\ -\sin \beta & \cos \beta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$R_{X_{E.O}} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \pi/2 & \sin \pi/2 \\ 0 & -\sin \pi/2 & \cos \pi/2 \end{bmatrix}$$

$$R_{EtoO} = R_{Y_{E.O}} \cdot R_{Z_{E.O}} \cdot R_{X_{E.O}}$$

La matrice di rotazione completa assume la seguente forma:

$$R_{EtoO} =$$

$$\begin{bmatrix} \cos \vartheta \cos \varphi & \sin \vartheta & -\cos \vartheta \sin \varphi \\ -\cos \psi \sin \vartheta \cos \varphi + \sin \psi \sin \varphi & \cos \psi \cos \vartheta & \cos \psi \sin \vartheta \sin \varphi + \sin \psi \cos \varphi \\ \sin \psi \sin \vartheta \cos \varphi + \cos \psi \sin \varphi & -\sin \psi \cos \vartheta & -\sin \psi \sin \vartheta \sin \varphi + \cos \psi \cos \varphi \end{bmatrix}$$

1.2.3. Da Orbit Frame a Boby Frame

Questa è la rotazione più importante in quanto ci consente di esaminare l'assetto vero e proprio del satellite nei confronti dell'assetto ideale dato dall'orbit frame, impiegando gli angoli di Eulero.

$$R_{OtoB} =$$

$$: \begin{bmatrix} \cos\Theta \cos\psi & \cos\Theta \sin\psi & -\sin\Theta \\ \sin\Phi \sin\Theta \cos\psi - \cos\Phi \sin\psi & \sin\Phi \sin\Theta \sin\psi + \cos\Phi \cos\psi & \sin\Phi \cos\Theta \\ \cos\psi \sin\Theta \cos\Phi + \sin\psi \sin\Phi & \cos\Phi \sin\Theta \sin\psi - \sin\Phi \cos\psi & \cos\Phi \cos\Theta \end{bmatrix}$$

La matrice $R_{O.to.B}$ Rappresenta la rotazione a partire dal sistema di riferimento dell'orbita a quello del satellite. Per la rotazione inversa si usa la matrice trasposta

$$R_{BtoO} = (R_{OtoB})'$$

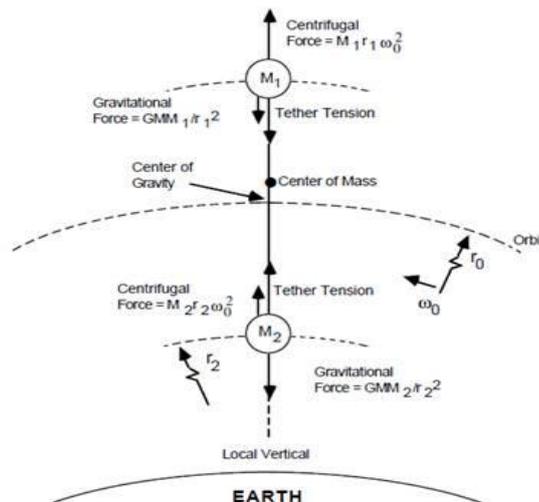
2. SISTEMI PER IL CONTROLLO D'ASSETTO

2.1. TECNICHE DI CONTROLLO PASSIVE [3] [8] [14]

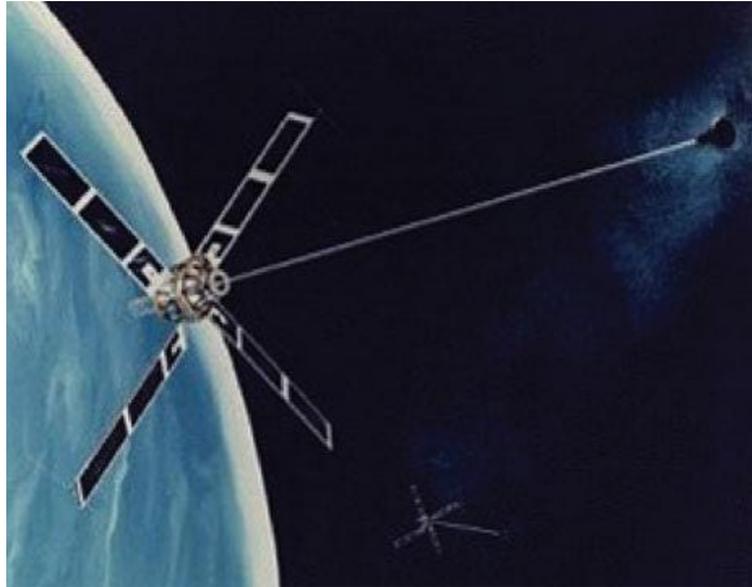
2.1.1. Controllo a gradiente di gravità.

Con l'espressione *Gravity Gradient* si indica un sistema inerziale che sfrutta come meccanismo per il controllo dell'assetto la tendenza di un corpo allungato a disporsi con l'asse di minima inerzia nella direzione della verticale locale.

Per comprendere il sistema a gradiente di gravità occorre immaginare due satelliti legati come in figura 2.1. Il sistema ruota alla velocità del centro di massa e in questo punto l'accelerazione di gravità g è completamente bilanciata dall'accelerazione centrifuga $a_0 = \omega_0^2 r_0$, dove r_0 è la distanza del centro di massa dal centro della terra. Il satellite più vicino è vincolato a orbitare con una velocità inferiore a quella che avrebbe se fosse libero e pertanto viene meno l'equilibrio tra a_0 e g , e risulta $a_0 < g$.



2.1 Schematizzazione di un satellite "gravity gradient".



2.2 Esempio di satellite controllato con gradiente di gravità

La prevalenza dell'attrazione gravitazionale fa sì che il satellite inferiore sia "tirato" verso la terra portandolo ad allinearsi con la direzione nadir. Allo stesso modo il satellite superiore poiché più distante sente una prevalenza dell'accelerazione centrifuga $a_0 > g$ e tende a portarsi verso l'esterno.

La posizione che si ottiene è di equilibrio e tende a contrastare le perturbazioni che provocherebbero rotazioni lungo gli assi appartenenti al piano tangente all'orbita. Tuttavia restano libere le rotazioni attorno all'asse di imbardata, si rende pertanto necessario un controllo d'assetto supplementare sull'asse verticale.

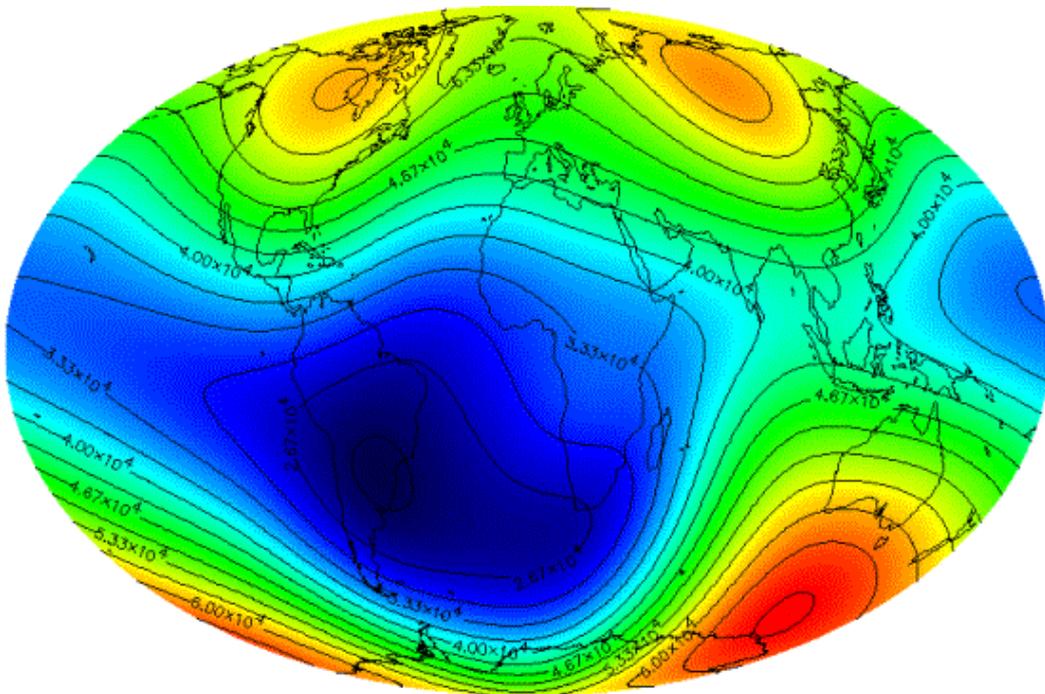
Condizione necessaria perché il metodo di stabilizzazione sia efficace è che il momento d'inerzia maggiore sia quello perpendicolare al piano tangente dell'orbita. Inoltre poiché il campo gravitazionale decresce col cubo della distanza, le coppie di torsione che si generano sono efficaci solo in orbite basse.

2.1.2. Controllo tramite magneti permanente

Si tratta di una tecnica piuttosto semplice da introdurre che prevede l'equipaggiamento del satellite con dei magneti fissi. L'interazione tra il campo magnetico terrestre e l'effetto dei magneti interni al satellite generano delle coppie torcenti sul satellite lo portano ad allinearsi con le linee di campo. Non vengono tuttavia generate coppie attorno all'asse posto in direzione del campo magnetico ed è quindi necessario introdurre un altro tipo di compensazione oltre a sistemi di smorzamento delle oscillazioni che si generano a causa dell'incostanza del campo magnetico.

Tale sistema è dunque più efficace in orbite equatoriali poiché le interazioni col campo sono più costanti. Inoltre il suo impiego è limitato esclusivamente in orbite medie o basse a causa dell'indebolimento del campo magnetico terrestre con la distanza.

L'accuratezza del puntamento è paragonabile al sistema *gravity gradient*, approssimativamente $\pm 5^\circ$.



2.3 Intensità del campo magnetico terrestre sulla superficie

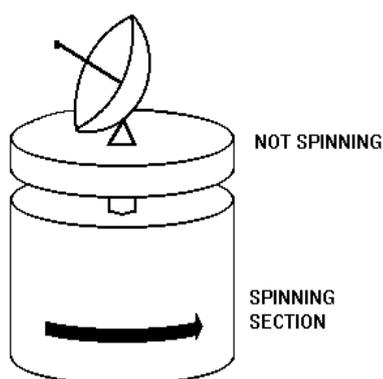
2.1.3. Controllo mediante *spin stabilization*

La tecnica di stabilizzazione di *spin* è di tipo passivo e prevede che l'intero satellite ruoti su se stesso in modo che il suo vettore di momento angolare rimanga pressochè fissato nello spazio inerziale.

Gli *spinner*, come vengono definiti tali oggetti, utilizzano la stabilità giroscopica per resistere passivamente alle torsioni di disturbo lungo due dei tre assi. Il movimento di rotazione è stabile se il satellite gira attorno all'asse che ha momento d'inerzia minimo.

Rispetto alle due precedenti è più complessa dal punto di vista della manovrabilità, richiede un'accurata disposizione delle masse interne ed un consumo di propellente per eventuali manovre di ripuntamento (a causa della rigidità giroscopica). Tutto ciò genera un puntamento molto accurato ma intermittente.

Per ovviare all'intermittenza del puntamento si impiega una tecnica a doppia stabilizzazione detta di *dual spin*. Si tratta di una tecnica molto più complessa dal punto di vista costruttivo e prevede una parte rotante ad alta velocità, detta rotore, che fornisce le proprietà giroscopiche al satellite esattamente come nei satelliti a singolo spin. Vi è poi una seconda sezione, fissa o rotante a velocità dell'ordine di 1 o 2 rotazioni per orbita,



la quale ha il compito di ospitare generalmente il payload. Il satellite che ne risulta è estremamente versatile, stabile, preciso e a puntamento continui, a discapito di un costo e di una complessità meccanica superiori, e una vita operativa più breve a causa della connessione elettrica tra le due sezioni rotanti.

2.4 Rappresentazione di un satellite dual spin

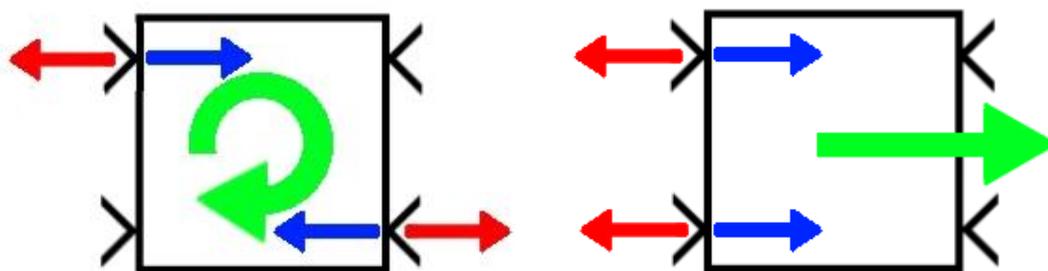
2.2. TECNICHE DI CONTROLLO ATTIVE

2.2.1. Sistemi a espulsione di massa

Sono i sistemi attivi maggiormente impiegati e si tratta solitamente di ugelli collegati a serbatoi che espellono gas (tipicamente idrazina) o ioni in piccoli impulsi calibrati in modo da ottenere forze o momenti in base al movimento desiderato. Il loro impiego spazia dai grandi movimenti, quali cambio di orbita e grandi spostamenti angolari, smorzamenti e desaturazioni delle ruote, d'inerzia o di momento, fino al controllo dello spin.

I problemi progettuali maggiori sono la determinazione del numero e della collocazione di ogni ugello e dell'impulso minimo necessario.

Come in figura 2.5 si nota che per avviare una traslazione si attivano entrambi gli ugelli della faccia opposta alla direzione del moto voluto, mentre per una rotazione vengono attivati ugelli di due facce opposte all'asse di rotazione in modo che la risultante delle forze resti nulla e si generi solamente il momento torcente desiderato.



2.5 Schema delle forze agenti sul satellite nel caso di slewing (a sinistra) e di shifting (a destra).

In rosso è rappresentato il getto del gas, in blu la forza di reazione corrispondente e in verde la risultante delle forze agenti sul satellite

Il sistema a getto di gas ha però alcuni svantaggi: In primo luogo la massa di propellente imbarcata riduce il payload massimo e una volta espulso richiede che vi sia una redistribuzione nella massa per mantenere corretto il centraggio. Inoltre anche se il propellente viene espulso in piccole quantità, una volta esaurito non si ha più la possibilità di controllo del satellite. I gas impiegati dai thrusters possono ricadere sulla strumentazione ottica danneggiandola o accorciandone notevolmente la durata operativa.

Per quanto riguarda la propulsione a ioni, la quantità di propellente necessaria, a parità di utilizzo col getto di gas, è nettamente minore, ma il peso degli apparati necessari al funzionamento è maggiore, giungendo di fatto a un'equivalenza tra i due sistemi. Dal punto di vista energetico il sistema a ioni necessita di un'elevata potenza elettrica, e la spinta generata viene convertita con bassa efficienza. Inoltre il getto di ioni genera radiazioni ad ampio spettro che può disturbare i sensori del payload, le telecomunicazioni e la strumentazione di bordo.

2.2.2. Sistemi di tipo rotorico

Si tratta dei sistemi già citati a "*Reaction Wheels*" (RW) e a "*Momentum Wheels*" (MW).

I rotori a reazione (RW) operano lungo uno o più assi del satellite con momento nullo. Essi hanno il compito di assorbire momenti ciclici e fungere da serbatoio momentaneo di momenti angolari, come in fase di ri-orientamento. Vengono messe in movimento quando necessario, ottenendo una rotazione opposta del satellite. Fermando la ruota si fermerà anche il satellite.

I rotori a momento (MW) possiedono un momento angolare non nullo. Il loro compito è fornire stabilità inerziale e, se pilotate, piccole possibilità di rotazione attorno al proprio asse.



2.6 Esempio di ruota di reazione

Le ruote di reazione consentono un controllo estremamente accurato, pagato però in termini di peso totale del sistema (finanche a 5Kg). La regolazione è indipendente dalla quota e permette un controllo costante delle coppie di disturbo. Inoltre è possibile operare manovre di ripuntamento in breve tempo. Le coppie di disturbo assorbite portano la ruota a saturazione, vale a dire che la ruota gira alla velocità di sincronismo del motore elettrico:

$$\omega_{rw} = \omega_{sinc}$$

Ciò comporta l'impossibilità di compiere ulteriori correzioni di assetto. Per desaturare la ruota si impiegano solitamente dei getti di gas che generano una coppia esterna in grado di azzerare il momento angolare immagazzinato nella ruota.

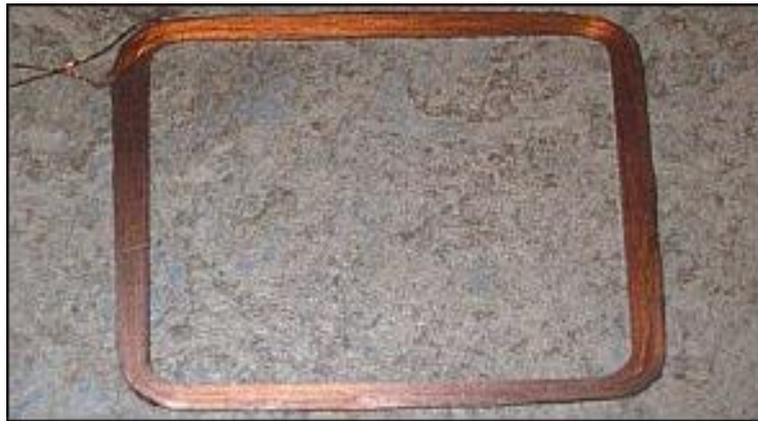
Le ruote di momento si comportano come le ruote di reazione, generano maggiore stabilità anche se consentono un minor assorbimento delle coppie di disturbo e una grande facilità di saturazione. È necessario che siano presenti sistemi di desaturazione o "*Momentum Dumping*".

2.2.3. Sistemi di tipo “Magnetic Torquer”

Il sistema ad attuatori magnetici o *Magnetic Torquer* (MT) è fondamentalmente un’evoluzione del concetto di controllo a magnete permanente. Si tratta infatti di equipaggiare il satellite con dei solenoidi (2.7), che si comportano come magneti di intensità variabile in funzione della corrente che vi circola all’interno.

E’ un sistema piuttosto semplice che non possiede parti meccaniche in movimento soggette a usura e richiede energia elettrica e un magnetometro per poter misurare il campo magnetico terrestre locale. L’impiego è limitato a orbite medio basse a causa dell’indebolimento del campo con il cubo della distanza.

Al contrario del sistema a magnete permanente è possibile invertire il campo magnetico generato dai solenoidi per ottenere rotazioni contrarie semplicemente facendo circolare la corrente nel verso opposto. Inoltre è possibile sfruttare i solenoidi per annullare eventuali magnetismi residui del satellite, compensare disturbi e compiere lente rotazioni. Poiché la coppia magnetica generata è esterna al satellite, è possibile impiegare gli attuatori magnetici per desaturare le ruote d’inerzia del satellite senza spreco di prezioso propellente.



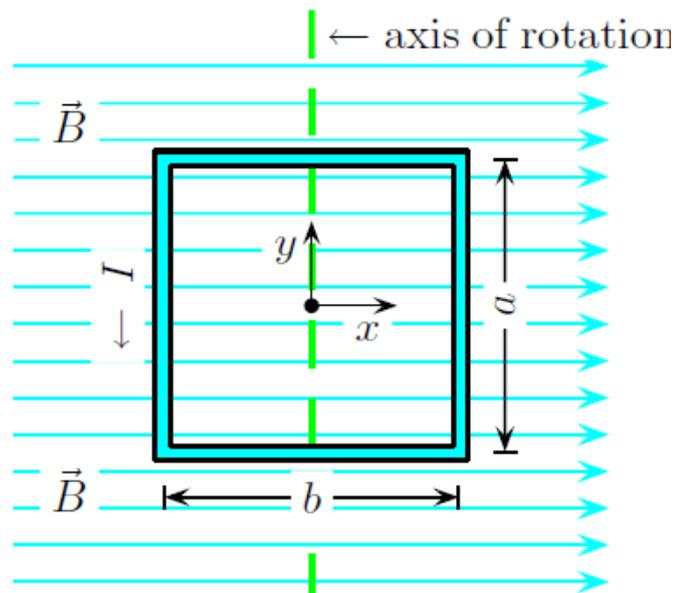
2.7 Esempio di solenoide per attuatori magnetici

Il principio di funzionamento è relativamente semplice, simile a quello di un motore elettrico. Sappiamo infatti dalla fisica che una spira percorsa da corrente genera un momento di dipolo magnetico che interagendo col campo magnetico terrestre genera un momento torcente proporzionale all'intensità del campo in quel punto e del momento di dipolo generato. Sia \vec{D} il momento magnetico del solenoide, con N numero di spire, I corrente circolante, A area racchiusa dalle spire e \vec{n} il versore dell'asse della bobina:

$$\vec{D} = N \cdot I \cdot A \cdot \vec{n}$$

Allora, dato \vec{B} il campo magnetico terrestre, la Coppia di torsione \vec{C} è:

$$\vec{C} = \vec{D} \wedge \vec{B}$$



2.8 Schema di una spira immersa in un campo magnetico. il versore n è, in questo caso, uscente dal piano.

Le coppie generate da questo sistema sono comunque molto piccole, dell'ordine dei 10^{-6} Nm, e non sono adatte per ripuntamenti rapidi. Non è possibile, come nel caso a magnete permanente, generare coppie in direzione del campo magnetico.

Occorre porre attenzione alla strumentazione di bordo in quanto dalle bobine vengono prodotti campi elettromagnetici che potrebbero interferire col corretto funzionamento degli apparati interni.

2.3. I DIVERSI TIPI DI SENSORI [3] [8]

Affrontiamo ora il discorso sui sensori più diffusi per l'applicazione satellitare. Dotare un satellite di sensori lo svincola dal dialogo continuo con la stazione di terra e fornisce una certa autonomia di comando. In taluni casi i sensori svolgono anche la funzione di sonde nel payload, ad esempio un magnetometro per campionare il campo magnetico terrestre.

2.3.1. Sensori di sole

I sensori solari sono dei rilevatori di luce visibile che misurano la differenza tra due angoli prodotti dalla luce attraverso delle fessure nella loro montatura. Per funzionare al meglio necessitano di angoli di vista molto aperti, e sono molto accurati e attendibili. Perciò sono molto diffusi.

In orbita bassa occorre considerare però il rapido alternarsi tra la semiorbita illuminata e quella eclissata e prevedere altri sistemi per determinare la posizione in assenza del sole. Inoltre in orbite LEO (*Low Earth Orbit*) La terra occupa circa il 40% del cielo e talvolta può causare interferenze nei sensori a causa della luce riflessa.

2.3.2. Sensori di stelle

I sensori di stelle sono estremamente precisi e sono stati oggetto di rapida evoluzione negli ultimi anni. Si distinguono in due tipologie, *Scanner* e *Tracker*.

I sensori di tipo *scanner (analizzatori)* sono degli analizzatori e vengono montati su satelliti rotanti: le stelle passano nel campo di vista dello scanner attraverso ad una serie di fenditure e, dopo alcuni passaggi, è possibile risalire all'assetto del veicolo.

I sensori di tipo *Tracker (puntatori)* vengono usati per stabilizzare il satellite sui 3 assi: puntando una o più stelle si ricavano le informazioni necessarie al calcolo dell'assetto del veicolo.

2.3.3. Sensori di orizzonte

I sensori di orizzonte sono dei rilevatori a infrarossi che percepiscono il contrasto tra il freddo dello spazio e il calore dell'atmosfera. Tramite algoritmo è quindi possibile definire le due tangenti alla terra per determinarne il centro e, quindi, poter puntare il satellite verso la direzione nadir. La precisione del sistema è notevole (0.1° - 0.25°)

2.3.4. Magnetometri

I magnetometri sono dei semplici e affidabili rilevatori di direzione e intensità del campo magnetico. Comparando la loro misurazione con la mappa del campo terrestre noto si può stabilire l'assetto del satellite. La precisione non è molto elevata a causa di disturbi magnetici e la variabilità del campo magnetico terrestre stesso. Pertanto è consigliabile accoppiare la rilevazione con quella di altri sensori

2.3.5. Ricevitori GPS

I *Global Positioning System* (GPS) sono dispositivi di navigazione estremamente precisi. I GPS sono usati principalmente per identificare la posizione nello spazio del satellite, e in certi casi per la determinazione dell'assetto utilizzando i diversi segnali provenienti da antenne separate a bordo dello stesso satellite. Questi sensori vengono spesso montati anche a bordo di piccoli satelliti su orbite LEO a causa del loro basso costo e del peso limitato.

2.3.6. Giroscopi

I giroscopi vengono impiegati come sensori inerziali che misurano la velocità lo spostamento angolare del satellite. Non sono in grado di fornire un riferimento assoluto e pertanto necessitano di essere accoppiati con altri sensori in grado di fornire riferimenti esterni, ottenendo così indicazioni d'assetto estremamente precise

3. IL SATELLITE

Nell'ultimo decennio si è assistito alla realizzazione di molti satelliti di dimensioni contenute, grazie allo sviluppo tecnologico a basso costo. Con l'introduzione, nel 2001, dello standard CUBESAT¹, nato per facilitare la realizzazione e il lancio di satelliti provenienti dal mondo universitario. Molti atenei tra i quali il Politecnico di Torino hanno progettato e realizzato il loro satellite.

Il satellite oggetto di questa tesi è AraMiS, diretto discendente del PiCPoT che verrà qui descritto.

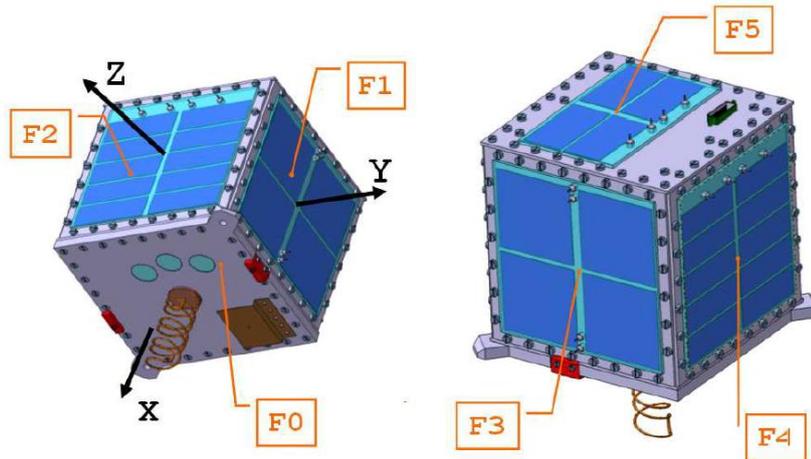
3.1. PICPoT [7] [13]

Nel 2003 il Politecnico di Torino iniziò lo sviluppo del proprio satellite, nominato PiCPoT, acronimo di: *Piccolo Cubo del Politecnico di Torino*.

Gli obiettivi della realizzazione di questo satellite erano:

- Far interagire attraverso un'attività interdisciplinare, i diversi dipartimenti, docenti e dottorandi del Politecnico.
- Testare il funzionamento e affidabilità della componentistica commerciale in orbita
- Effettuare acquisizione di immagini e misurazioni in ambiente spaziale per poi inviare i dati registrati alla stazione di terra

¹ Questo standard è stato sviluppato nel 2001 dal Professor Robert Twiggs, docente alla Stanford University, USA, in collaborazione con lo Space Systems Development Laboratory (SSDL) della Stanford University e la California Polytechnic State University. Il CUBESAT `e uno standard per picosatelliti di forma cubica di 10 cm di lato e con una massa di 1 Kg la cui struttura è definita in funzione dell'adattamento al lanciatore POD (Picosatellie Orbital Deployer).



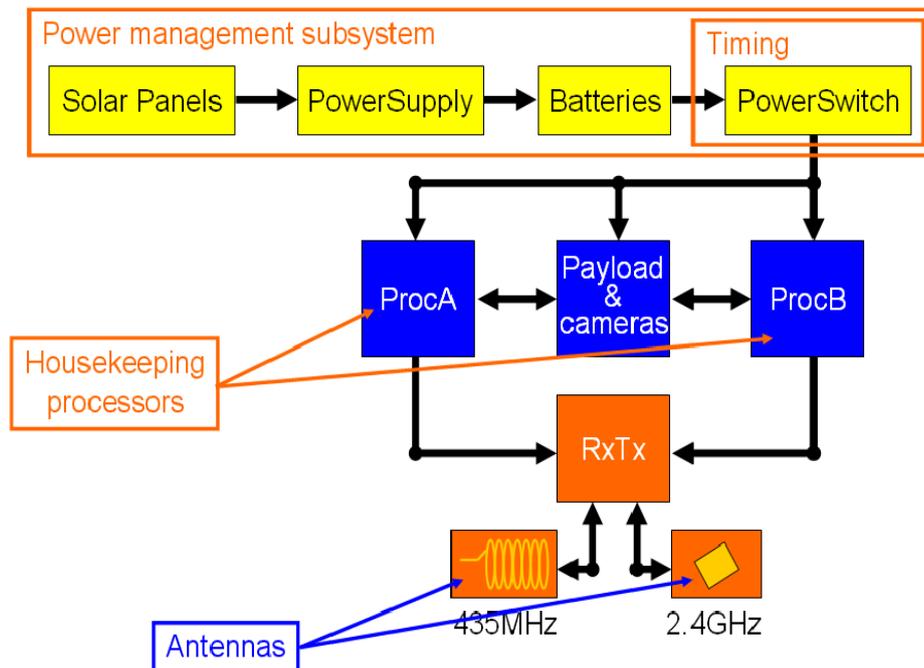
3.1 *Struttura esterna del satellite PiCPoT*

Il PiCPoT si presenta come un satellite di forma cubica di lato 13cm e massa di 2.5Kg, con le facce esterne realizzate in lega di alluminio di tipo 5000 AlMn. Come si può vedere dalla figura 3.1, all'esterno su 5 delle 6 facce sono presenti pannelli solari, che hanno il compito di fornire l'alimentazione al satellite.

Sull'ultima faccia F0 invece sono presenti tra telecamere e due antenne di frequenza 437 MHz e 2.4 GHz impiegate per comunicare con la stazione di terra. Sulla stessa faccia sono presenti due *Kill Switch*, vale a dire due interruttori che hanno il compito di tagliare l'alimentazione al satellite fintanto che questi non viene messo in orbita dal lanciatore.

L'equipaggiamento elettronico interno si compone dei seguenti elementi:

- Solar Panel: Pannelli solari per la conversione di energia solare in energia elettrica a 5v.
- Power Supply: Convertitore di tensione dai 5v in entrata dai pannelli solari ai 7.2v necessari a ricaricare le batterie, gestendo le fasi di ricarica.
- Batteries: 2 accumulatori di energia elettrica necessari ad alimentare il satellite anche in condizioni di buio. Sono di tipologia differente (NiCd e NiMh) per sopperire ad un eventuale malfunzionamento di uno dei due pacchi.

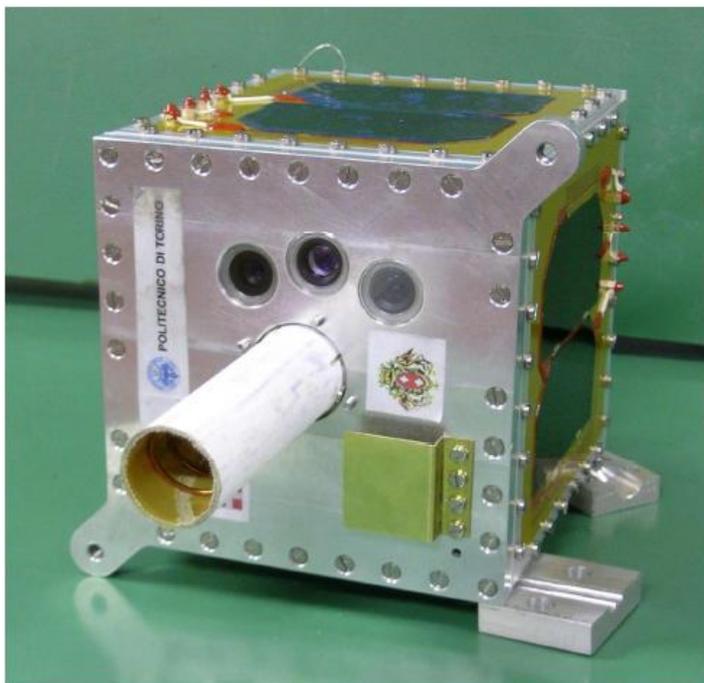


3.2 Equipaggiamento elettronico PiCPoT

- Power Switch: Gestore dell'alimentazione dell'elettronica. Converte la tensione in ingresso in funzione delle esigenze di ogni utilizzatore. Per motivi di risparmio energetico l'elettronica non in funzione viene scollegata. Questo componente genera anche una base di temporizzazione.
- ProcA e ProcB: Moduli con le stesse funzionalità, cioè acquisiscono dati telemetrici e comunicano con payload e Rx/Tx, ma realizzati con tecnologie tra loro differenti. Questo si rende necessario per motivi di affidabilità e garantire il corretto funzionamento di almeno uno dei due moduli. Non sono comunque mai attivi contemporaneamente per limitare il consumo energetico.
- Rx/Tx: Modulo di ricetrasmisione tra il satellite e la stazione di terra. È provvisto di due antenne di comunicazione di frequenze differenti, 437 Mhz e 2.4 GHz, utilizzabili indistintamente, sempre per questioni di affidabilità.

- Payload: in questo caso si tratta di 3 fotocamere che acquisiscono immagini della superficie terrestre. Vengono poi successivamente salvate e compresse in Jpeg e poi trasmesse ai moduli procA o procB per la trasmissione a terra.

Il lancio del satellite è avvenuto il 26 Luglio 2006 dal lanciatore Dnepr dalla stazione di Baikonur in Kazakistan. Per motivi idraulici del lanciatore non è stato purtroppo possibile effettuare test in orbita.



3.3 *Il satellite PiCPoT*



3.4 *Il lanciatore Dnepr*

3.2. ARAMIS [4] [7] [13]

AraMiS, acronimo di Architettura Modulare per Satelliti, è il nome del progetto che costituisce l'evoluzione di PiCPoT sul quale, dall'autunno 2006, è impegnato il Politecnico di Torino.

Gli obiettivi dei due satelliti restano pressoché gli stessi, ma si è desiderato rivoluzionare l'approccio base alla progettazione favorendo la

modularità del progetto. L'architettura del satellite è suddivisa in blocchi funzionali adibiti allo svolgimento di un ben preciso compito e quindi indipendenti tra loro, permettendo un riutilizzo e una diversa combinazione tra i blocchi in funzione delle specifiche esigenze della missione. Ogni sottosistema è pertanto realizzato su un singolo circuito stampato separato dagli altri.

Questa peculiare modularità di AraMiS permette di:

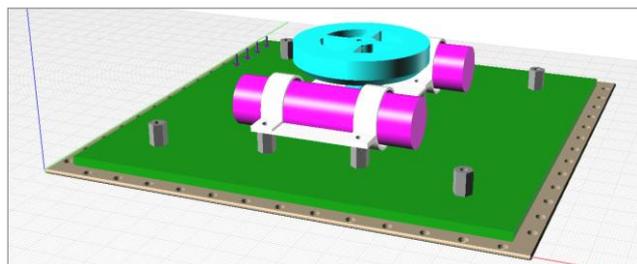
- Ridurre i costi di realizzazione, in quanto una volta progettato il modulo è riutilizzabile per missioni di tipo diverso.
- Ridurre i tempi di progettazione grazie alla suddivisione del lavoro in più gruppi con l'unico anello di connessione il linguaggio con cui dialogano i sottosistemi.
- Ottenere diverse configurazioni del satellite in base agli obiettivi delle missioni.

I diversi moduli di AraMiS sono raggruppati in due fondamentali categorie che accomunano caratteristiche concettuali, funzioni e codici comuni. Prendono il nome di *Tile* (mattonelle), hanno dimensione 165mmx165mm e sono poste nelle facce esterne del cubo.

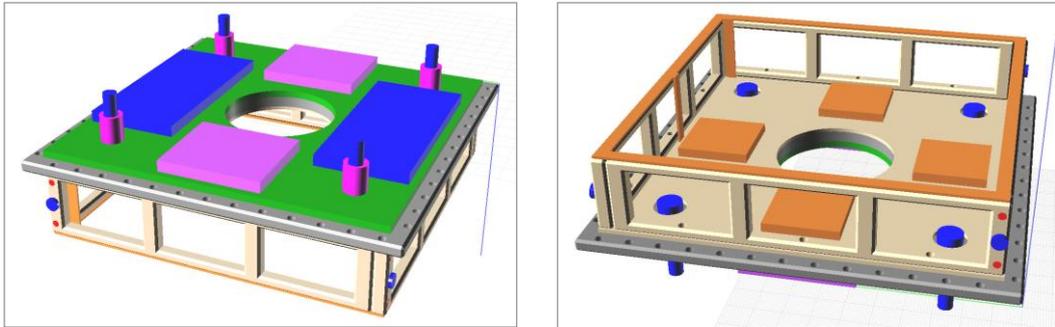
Distinguiamo quindi in:

- *Power Management Tile*, gestisce le alimentazioni dei componenti di bordo, provvede alla ricarica delle batterie e alla conversione delle tensioni. Questamattonella ingloba i sistemi precedentemente descritti del PiCPoT quali: Power Supply, Power Switch, Solar Panel e aggiunge in più un controllo di assetto attivo;

3.5
Power management tile.
dettagli:
batterie, viola
ruota di reazione, azzurro



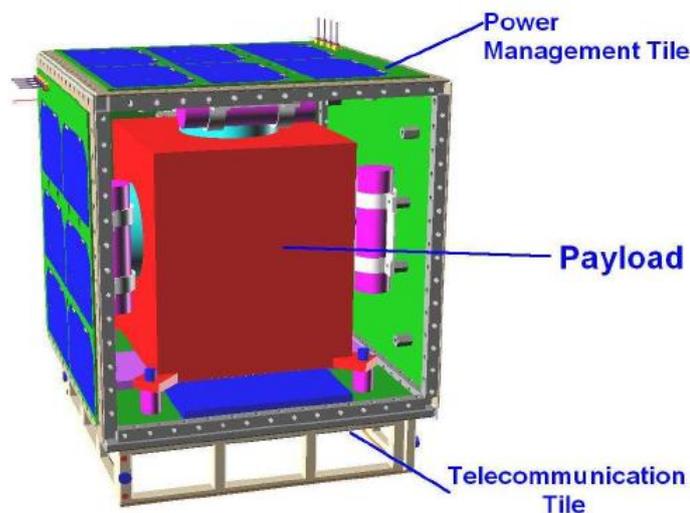
- *Telecommunication Tile*, Riceve e trasmette dati e comandi con la stazione di terra sulle stesse due frequenze utilizzate da PiCPoT (437 MHz e 2.4 GHz). La ridondanza dei canali è relativa all'affidabilità del sistema, a compensazione di eventuali guasti.



3.6 *Telecommunication tile, interno (sinistra) e esterno (destra)*

All'interno del satellite sono invece presenti i due moduli non standard:

- On Board Processor, gestisce il controllo e il funzionamento di tutto il sistema;
- Payload, Può variare ampiamente in funzione della missione, ad esempio potrebbe avere funzioni analoghe a quelle del PiCPoT (fotografare la superficie terrestre).

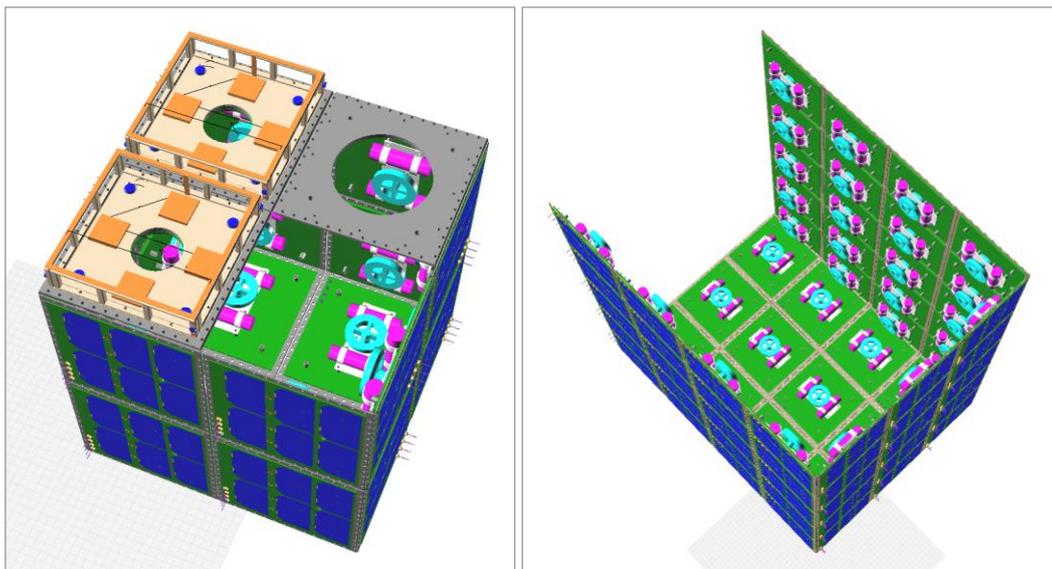


3.7 *visione interna di AraMiS.*

Sulla Telecommunication tile è inoltre presente un foro di circa 50mm di diametro per consentire un eventuale accesso all'esterno del payload. Se per esigenze non fosse sufficiente si può agevolmente rimuovere una tile laterale del satellite e ospitarne una ad hoc per il payload.

La forma cubica appena presentata è la più piccola configurazione che è possibile creare sfruttando la modularità dell'architettura. Si compone di PMW (power management tile) e una TCT (Telecommunication Tile). Le dimensioni finali sono dunque 165mmx165mmx165mm e prevede uno spazio massimo per il payload all'interno di 1 dm³.

E' possibile generare molte altre forme in funzione del payload e degli obiettivi di missione. Ad esempio per payload maggiori si possono creare strutture a parallelepipedo con 2x2x2 tile o 3x4x5tile (fig 3.8).



3.8 *Disposizione 2x2x2 (sinistra) e 3x4x5 (destra)*

Queste configurazioni sono solo due esempi ma mostrano come l'architettura del satellite sia fortemente adattabile alle esigenze del cliente potendo ospitare ipoteticamente ogni tipo di payload, con l'unico compito di progettare, in base alla configurazione, la struttura di supporto del satellite costituiti di correntini, in acciaio o altre leghe.

Per una configurazione di questo genere risulta agevole il calcolo del volume interno disponibile per il payload in base alla seguente formula:

$$V_{Payload} = (A - 7 \text{ cm}) * (B - 7 \text{ cm}) * (C - 7 \text{ cm})$$

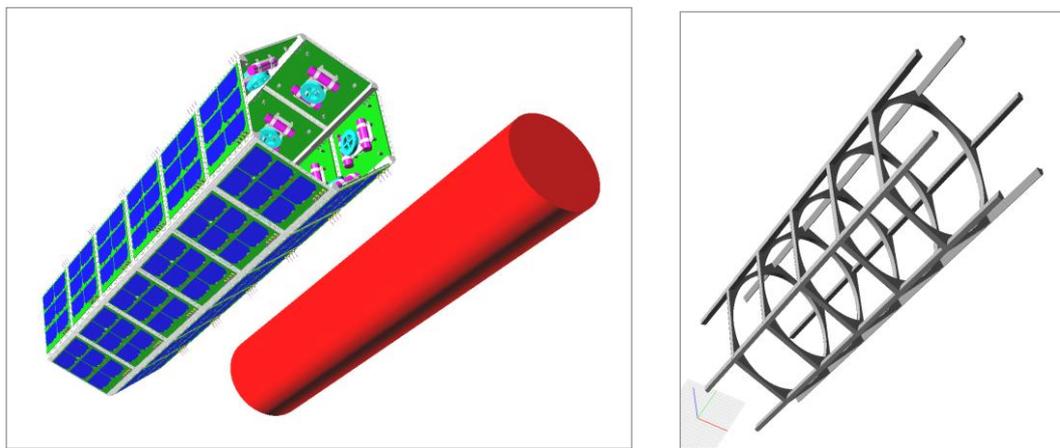
Dove:

$$A = (\#tiles \text{ asse } x) * 16.5 \text{ cm}$$

$$B = (\#tiles \text{ asse } y) * 16.5 \text{ cm}$$

$$C = (\#tiles \text{ asse } z) * 16.5 \text{ cm}$$

Un'altra configurazione interessante su cui è opportuno focalizzare l'analisi riguarda una configurazione prismatica di base esagonale o ottagonale (fig 3.9)



3.9 *Struttura prismatica esagonale (a sinistra), payload cilindrico (in rosso) e struttura portante della configurazione (a destra)*

Peculiarità di questa configurazione è la capacità di ospitare payload con una dimensione molto maggiore rispetto alle altre due. Si ha un limite alla dimensione del payload, di circa 20cm di diametro per struttura esagonale e di circa 32cm per quella ottagonale. In termini di lunghezza non ci sono particolari vincoli (oltre a quelli strutturali) in quanto è virtualmente possibile aggiungere tile.

Limitazioni progettuali sono la progettazione di un apposito telaio e una nuova progettazione per le tile di base poiché caratterizzate da forma e necessità differenti, come ad esempio l'installazione di una ruota di reazione maggiorata per far fronte alle necessità del satellite, oppure installare antenne particolari.

4. SCELTE PROGETTUALI

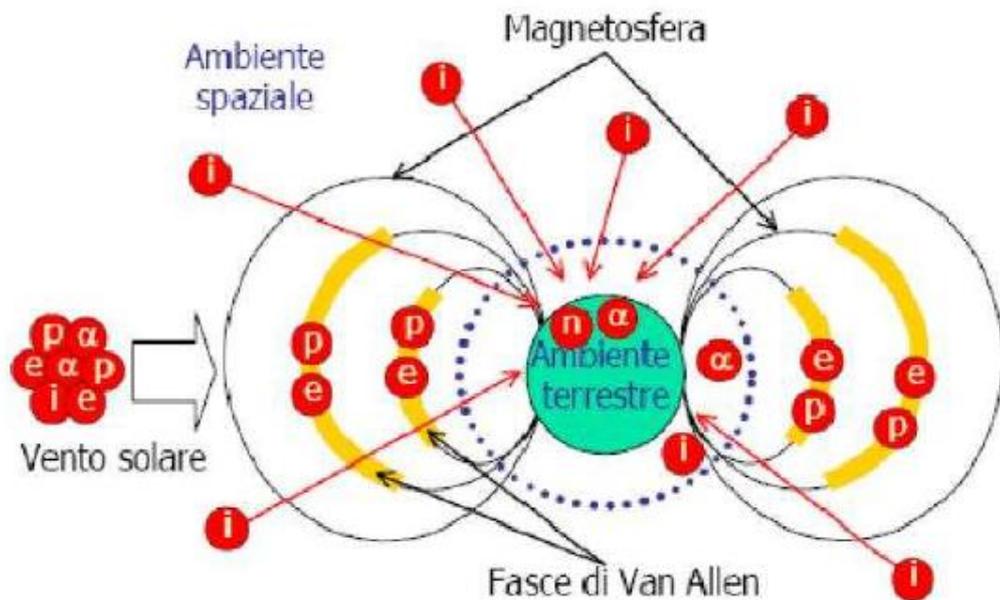
4.1. AMBIENTE OPERATIVO [7] [13] [14]

Per realizzare un sistema di controllo di assetto efficace occorre innanzitutto tenere in considerazione l'ambiente operativo della missione.

Il satellite AraMiS è progettato per operare ad una quota compresa tra i 600 km e gli 800 km, in orbite di tipo LEO (*Low Earth Orbit*, orbita terrestre bassa). Si tratta di un'orbita circolare compresa tra l'atmosfera e le fasce di Van Allen dove si svolgono la maggior parte delle missioni spaziali. Un corpo che orbita su una LEO ha un periodo di rivoluzione approssimativo di 90 minuti, e viaggia a circa 27'400 km/h.

L'orbita LEO offre alcuni vantaggi:

- La presenza delle fasce di Van Allen, alla quota di 2000 km, protegge il satellite da un'eccessiva radiazione solare ad alta energia che potrebbe pregiudicare il corretto funzionamento della strumentazione;
- Si è sufficientemente in quota da evitare il contatto con l'atmosfera, il cui limite è a circa 200 km. Non si hanno quindi effetti di tipo aerodinamico sul satellite, che oltre a dissipare energia porterebbero ad un'eccessiva sollecitazione del corpo riducendone notevolmente la vita utile operativa;
- La vicinanza con la terra porta alcuni vantaggi al livello delle telecomunicazioni. Il ritardo di propagazione è limitato a circa 20-25 ms, che è paragonabile a sistemi di comunicazione terrestri poco potenti;
- Ulteriore vantaggio è la possibilità di sfruttare sistemi a gravità o magnetici in quanto entrambi i campi sono ancora sufficientemente potenti da poter essere impiegati.



4.1 Radiazioni e fonti di disturbo in ambiente spaziale

Un altro aspetto essenziale in orbita LEO sono le condizioni di pressione e temperatura: Le condizioni infatti sono pressoché di vuoto, che porta a temperature medie molto basse, con quasi totale assenza di efficaci scambi termici convettivi. Se poi consideriamo gli effetti della radiazione solare non schermata da nubi o atmosfera, si possono raggiungere temperature molto elevate sulle facce esposte al sole, che tenderanno a disperdere calore per irraggiamento una volta portate nella zona buia. All'interno del satellite, grazie all'ottima conducibilità termica della sua struttura, dovremmo trovarci in condizioni operative comprese tra i -30°C e i 30°C .

Come sappiamo è di fondamentale importanza conoscere l'andamento del campo magnetico: Avere indicazioni su intensità e direzione permette di poter ottimizzare il funzionamento del controllo magnetico (permanente o magneto torque indifferentemente).

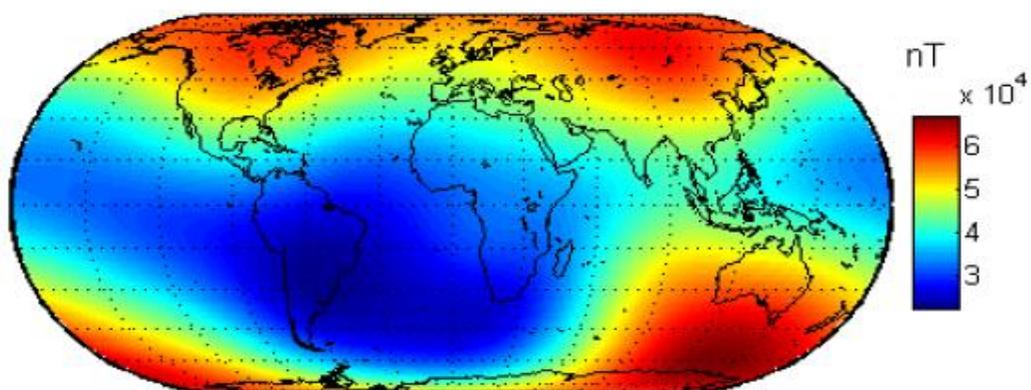
Ogni qual volta si effettua una misura di campo occorre tenere presente che il valore ottenuto è dato dalla sovrapposizione di diversi

fattori. Questi contributi sono di origine diversa e possono essere considerati separatamente:

- *Campo principale*, generato nel nucleo fluido
- *Campo crostale*, generato dalle rocce magnetizzate della crosta terrestre;
- *Campo esterno*, generato da correnti elettriche circolanti nella ionosfera e nella magnetosfera.
- *Campo di induzione elettromagnetica*, generato da correnti indotte nella crosta e nel mantello dal campo esterno variabile nel tempo.

Il campo principale rappresenta il 99% del campo misurato, e si comporta con buona approssimazione come un magnete posto al centro della terra e con asse inclinato rispetto all'asse di rotazione della terra di $11^{\circ}30'$

Il campo magnetico è misurato in termini del vettore di induzione magnetica \vec{B} . La sua unità di misura nel Sistema Internazionale è il Tesla (T), ma per applicazione pratiche terrestri viene solitamente preferito il Gauss (G) che è un multiplo del Tesla , ($1G = 10^{-4}T$). Il modulo del campo magnetico è molto variabile, da 0.2G all'equatore, a 0.7G ai poli, come mostrato in figura 4.2.



4.2 Intensità del campo magnetico terrestre misurato nel 2009

4.2. EFFETTI DI DISTURBO AMBIENTALI

In questa sezione effettuiamo una breve analisi delle principali cause di disturbo all'assetto di un satellite e della loro intensità. Per un corpo orbitante attorno alla terra queste sono essenzialmente quattro:

- Effetti del gradiente di gravità;
- Torsioni sul velivolo dovute al campo magnetico;
- Impatto con la radiazione solare;
- Torsioni aerodinamiche;

Riporto brevemente le equazioni per stimare l'intensità dei vari disturbi.

4.2.1. Il gradiente di gravità

E' già stato esaminato in precedenza ed è utilizzabile per stabilizzare un satellite lungo la verticale locale. Tuttavia se questa non è la configurazione d'assetto richiesta il gradiente di gravità si comporta da coppia di disturbo ed è necessario compensarla.

La massima torsione gravitazionale è data da:

$$T_g = \frac{3\mu}{2R^3} |J_{yaw} - J_{min}| \sin(2\theta)$$

Dove R è la distanza del satellite dal centro della terra, θ è l'angolo di massima deviazione dell'asse z dalla verticale locale. J_{yaw} rappresenta il momento d'inerzia attorno all'asse di imbardata, mentre J_{min} rappresenta il minore tra i momenti d'inerzia di beccheggio e di rollio. μ rappresenta la costante di gravitazione geocentrica e vale:

$$\mu = Gm_{earth} = 3.986 * 10^{14} [m^3/s^2]$$

La torsione gravitazionale è fortemente dipendente dall'inerzia del satellite e dall'altezza dell'orbita.

4.2.2. Il campo magnetico

Gli effetti del campo magnetico, come quello gravitazionale, possono essere favorevoli o sfavorevoli. Infatti tramite magneti o magnetotorquer è possibile orientare il satellite secondo le linee di campo. Tuttavia talvolta il satellite presenta un dipolo magnetico residuo M che interagisce autonomamente col campo magnetico terrestre generando una coppia magnetica torcente:

$$T_m = M B$$

Dove B è il modulo del campo magnetico terrestre locale, variabile dal polo all'equatore e approssimabile per una stima di calcolo a:

$$B_{polare} = \frac{2M_{earth}}{R^3}$$
$$B_{equatoriale} = \frac{M_{earth}}{R^3}$$

Con M_{earth} momento magnetico terrestre pari a: $M_{earth} = 7.96 * 10^{-15} [Tm^3]$ e R la distanza tra il centro del dipolo terrestre e il satellite.

I parametri da cui dipende maggiormente questo effetto di disturbo sono l'altezza dell'orbita, il dipolo magnetico residuo e l'inclinazione dell'orbita.

4.2.3. La radiazione solare

Le torsioni di radiazione solare sono causate dalla pressione derivante dal trasferimento di momento tra la radiazione solare e le superfici del satellite, che possono rifletterla o assorbirla. Tale pressione è dunque fortemente dipendente dal tipo di superficie illuminata.

Gli effetti della radiazione solare possono manifestarsi ciclicamente per satelliti orientati verso la terra, e con torsioni costanti per satelliti puntati inerzialmente.

La massima torsione dovuta alla pressione solare è data da:

$$T_{sp} = \frac{F_s}{c} A_s (1 + q) (c_{ps} - c_g) \cos \varphi$$

Dove c è la velocità della luce, A_s è la superficie esposta all'azione della radiazione solare, c_{ps} è il centro della pressione solare, c_g il centro di massa del satellite, φ è l'incidenza della radiazione solare e F_s è la costante solare ($F_s = 1367 [W/m^2]$)

Il termine q prende il nome di fattore di riflettività ed è compreso tra 0 e 1 e varia in funzione del tipo di superficie colpita. Vale 0 per superfici totalmente assorbenti e 1 per superfici totalmente riflettenti. In un satellite generalmente si considerano superfici assorbenti i pannelli solari e superfici riflettenti le altre parti del velivolo.

Questo genere di disturbo dipende principalmente dalla geometria, di forma e di massa, e dal tipo di superfici del satellite.

4.2.4. L'aerodinamica

Gli effetti dell'aerodinamica per orbite molto basse si verificano come effetti di resistenza aerodinamica, che possono generare torsioni. Si avranno torsioni cicliche per satelliti orientati inerzialmente, e costanti per satelliti puntati verso la terra.

La massima torsione aerodinamica è data da:

$$T_a = D(c_{pa} - c_g)$$

Dove D è la forza di attrito, o resistenza all'avanzamento (Drag)

$$D = \frac{1}{2} \rho C_d A V^2$$

Con C_d coefficiente di resistenza, adimensionale, tipicamente compreso tra 2 e 2.5, ρ è la densità atmosferica, A la sezione esposta alle azioni aerodinamiche, V è la velocità rispetto all'atmosfera e $(c_{pa} - c_g)$ è la distanza tra centro di pressione aerodinamico e il centro di massa.

I disturbi di tipo aerodinamico dipendono da altezza dell'orbita, forma e posizione del baricentro.

4.2.5. Calcolo delle coppie di disturbo

Per avere un'idea dell'intensità dei disturbi riporto i valori delle torsioni per un satellite piuttosto grosso ($A=12m^2$, $L=1m$, $J_z=1000 \text{ kg/m}^2$, $J_y=5000 \text{ kg/m}^2$) orbitante a una quota di 800 km.

$$T_g = 2.5 \cdot 10^{-4} \text{ Nm}$$

$$T_{sp} = 8.9 \cdot 10^{-5} \text{ Nm}$$

$$T_m = 9.4 \cdot 10^{-5} \text{ Nm}$$

$$T_a = 5.6 \cdot 10^{-5} \text{ Nm}$$

Si nota quindi in conclusione che gli effetti di disturbo esterno più rilevanti in queste condizioni sono quelli relativi al gradiente di gravità.

4.3. SCHEDE TECNICHE [7] [13]

4.3.1. Satellite

Effettuiamo un confronto tra il satellite PicPoT e le configurazioni del satellite AraMiS.

		PiCPoT	AraMiS cubo minimo	AraMis Esagono
Massa	$M \text{ [kg]}$	2.5	10	50
Dimensioni:	$L_x \text{ [m]}$	0.13	0.165	0.330
	$L_y \text{ [m]}$	0.13	0.165	0.286
	$L_z \text{ [m]}$	0.13	0.165	0.825
Momento d'inerzia:	$J_{xx} \text{ [kg * m}^2\text{]}$	0.022	0.045	2.836
	$J_{yy} \text{ [kg * m}^2\text{]}$	0.022	0.045	2.836
	$J_{zz} \text{ [kg * m}^2\text{]}$	0.022	0.045	0.575

In particolare per quest'ultimo si analizzano le configurazioni di cubo minimo e di configurazione esagonale, con una composizione in lunghezza di 5 tiles, e assi orientati come segue:

L'asse z è l'asse orientato in lunghezza del satellite, secondo l'asse del payload. L'asse x e l'asse y data la geometria del satellite sono assi di simmetria, abbiamo quindi considerato l'asse x come congiungente due vertici opposti (lato del satellite 0.330m) e l'asse y ortogonale congiungente il punto medio dei due lati opposti paralleli all'asse x (lato 0.286m).

Per la determinazione del momento d'inerzia si è considerate una distribuzione uniforme di massa su tutto il volume. Nel caso del cubo la formula esatta è:

$$J = \frac{1}{2}M(2l^2) \qquad J = J_{xx} = J_{yy} = J_{zz}$$

Per il caso della configurazione esagonale prismatica il calcolo è più complesso, e si è fatto uso di un'approssimazione. L'esagono è facilmente approssimabile ad un cerchio.

$$J_{zz\text{esagono}} = \frac{1}{2}MR^2$$

Con la formula precedente è possibile calcolare l'inerzia attorno all'asse Z, analizzando le possibili approssimazioni. L'inerzia del cilindro costituito dalla circonferenza circoscritta vale $J_{circ} = 0.681$, l'inerzia del cilindro creato con la circonferenza inscritta vale $J_{ins} = 0.510$, che costituiscono gli estremi che includono il vero momento di inerzia. Poiché la differenza è piuttosto elevata si è considerato il cilindro la cui circonferenza di base avesse un raggio pari alla media dei raggi della circonferenza inscritta e circoscritta, da cui si ricava $J_{mean} = 0.590$, che costituisce ancora un'approssimazione per eccesso. Se consideriamo un cilindro la cui area di base sia pari all'area dell'esagono si ottiene $J_{area} = 0.563$, che approssima in difetto il momento d'inerzia.

Possiamo quindi con buona approssimazione considerare il momento di inerzia $J_{zz} = 0.575$.

Per i momenti di inerzia lungo x e lungo y si considera, senza approssimazione alcuna, che si comporti come un corpo affusolato, e il calcolo viene affrontato con la seguente:

$$J_{xx_{esagono}} = J_{yy_{esagono}} = \frac{1}{2}ML_z^2$$

4.3.2. Parametri di missione

Si assume per semplicità di calcolo un'orbita ideale circolare per lo studio d'assetto, in quanto l'orbita reale sarà nota solo al momento del lancio del satellite. Inoltre tale assunzione non comporta approssimazioni molto sensibile in quanto, in orbita LEO, si è molto prossimi alla superficie terrestre.

Quota	$h [km]$	600
-------	----------	-----

Parametri orbitali

Inclinazione orbitale	$ln[^\circ]$	86
Longitudine del nodo ascendente	$\Omega[^\circ]$	90
Argomento di perielio	$P[^\circ]$	90
Eccentricità	e	0
Periodo di rivoluzione	$n[^\circ]$	90
Anomalia media	M	/

Parametri utili

Velocità di rotazione della terra	$\omega_{er} [rad/s]$	7.272 E-5
Velocità angolare orbitale del satellite	$\omega_{or} [rad/s]$	1.164 E-3

4.3.3. Equipaggiamento del satellite: sensori

Il satellite oggetto di questa tesi è equipaggiato dei seguenti dispositivi per l'analisi dell'assetto:

- *Magnetometri*, avremo due magnetometri ortogonali tra loro posti su almeno tre facce del satellite, ortogonali tra loro. In tal modo si avrà una misura del campo magnetico lungo i tre assi del satellite. Si hanno in totale sei magnetometri, che forniranno la stessa rilevazione a coppie di due. Questo per ottenere una ridondanza e poter quindi sopperire ad un eventuale malfunzionamento di un componente.

Abbinando tali sensori ad una mappa del campo magnetico terrestre (la cui precisione non è elevata) è possibile determinare assetto e posizione del satellite con una precisione non elevata, ma sufficiente.

- *Sensori di stelle, di sole e di orizzonte*, avranno il compito di determinare l'assetto del satellite rispetto al riferimento orbitante.
- *Sensori giroscopici*, Utilizzando i dati di assetto ottenuti tramite altri sensori è possibile determinare spostamenti angolari e velocità angolari lungo i tre assi del satellite. Ciò avviene montando i sensori come i magnetometri, in modo da avere ridondanza e affidabilità.
- *Ricevitori GPS*. Dotazione non ancora confermata del satellite, se equipaggiati avranno lo scopo di determinare la posizione spaziale rispetto alla terra (quota, latitudine e longitudine).

La precisione ottenibile dalle combinazioni dei vari sensori è elevata e permette di conseguenza di ottenere un'elevata precisione nella successiva fase del controllo di assetto.

4.3.4. Equipaggiamento del satellite: attuatori

Gli attuatori di cui dispone il nostro satellite sono i seguenti:

- *Ruote d'inerzia o di reazione (RW)*, sono tre, disposte con l'asse di rotazione lungo i tre differenti assi del satellite per permettere un controllo tridimensionale dell'assetto.
- *Attuatori magnetici (MT)*, anche in questo caso sono tre, disposti con l'asse del solenoide parallelo a ognuno dei 3 assi del satellite. Sfruttando le rilevazioni dei magnetometri è possibile scegliere un assetto relativamente all'andamento del campo magnetico.

Nel prossimo capitolo andremo ad analizzare e a dimensionare i due sistemi attuatori di cui dispone il nostro satellite.

5. DIMENSIONAMENTO DEI SISTEMI DI CONTROLLO ATTIVI

5.1. MAGNETIC TORQUER [1][3][7][13]

5.1.1. Fisica del solenoide

Come abbiamo già visto un sistema di tipo magnetic torquer altro non è che un solenoide. Si tratta di un avvolgimento di spire metalliche che, se percorse da corrente sono in grado di generare un momento magnetico. Se il solenoide è immerso in un campo magnetico, essi interagiscono sviluppando una coppia meccanica \vec{C} nel verso che tende ad allineare tra loro i due campi.

Per semplicità i calcoli che seguono sono riferiti solo ad un solo solenoide, e quindi ad un solo asse del satellite.

Possiamo ricavare il momento magnetico \vec{D} del solenoide nel modo seguente:

$$\vec{D} = N \cdot I \cdot A \cdot \vec{n}$$

Dove:

- N rappresenta il numero di spire
- A rappresenta l'area racchiusa dalla spira
- I rappresenta la corrente che scorre nella spira
- \vec{n} rappresenta il versore ortogonale alla superficie del solenoide, che si considera uscente e allineato con l'asse corpo del satellite.

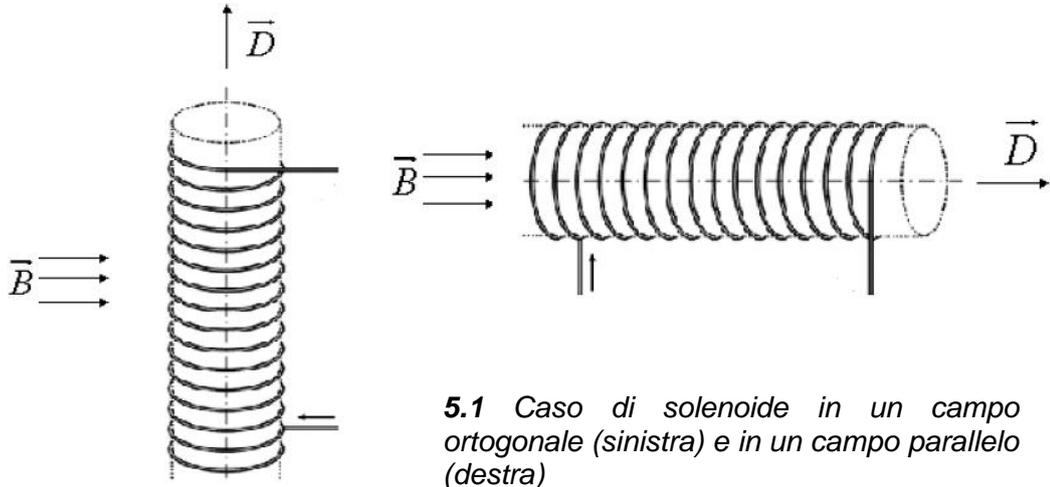
La coppia \vec{C} che deriva dall'interazione tra \vec{D} e \vec{B} è calcolata nel modo seguente:

$$\vec{C} = \vec{D} \wedge \vec{B}$$

Tale coppia è ortogonale sia al momento magnetico generato dalle spire, sia al campo magnetico esterno. L'intensità della coppia sarà:

$$|\vec{C}| = |\vec{D}| \cdot |\vec{B}| \cdot \sin \alpha$$

Dove α rappresenta l'angolo tra la direzione dei due vettori. Si nota dunque che la coppia sarà massima quando i vettori saranno tra loro ortogonali ($\sin \alpha = 1$) mentre sarà nulla quando saranno allineati ($\sin \alpha = 0$).



L'intensità del campo magnetico è nota in ogni punto con bassa precisione, ma è comunque misurata istante per istante dai magnetometri posti sul satellite.

5.1.2. Pilotaggio del solenoide

Nota quindi la coppia generata possiamo analizzare gli effetti di reazione ottenuti sul satellite.

$$\vec{C} = J_{SAT} \cdot \frac{d\omega_{SAT}}{dt}$$

Dove J_{SAT} è il momento d'inerzia del satellite sull'asse considerato, e ω_{SAT} è la velocità angolare del satellite. Possiamo calcolare la velocità angolare ottenuta dall'applicazione di una coppia per un tempo T che sarà data da:

$$\omega_{SAT} = \int_0^T \frac{\vec{C}}{J_{SAT}} dt$$

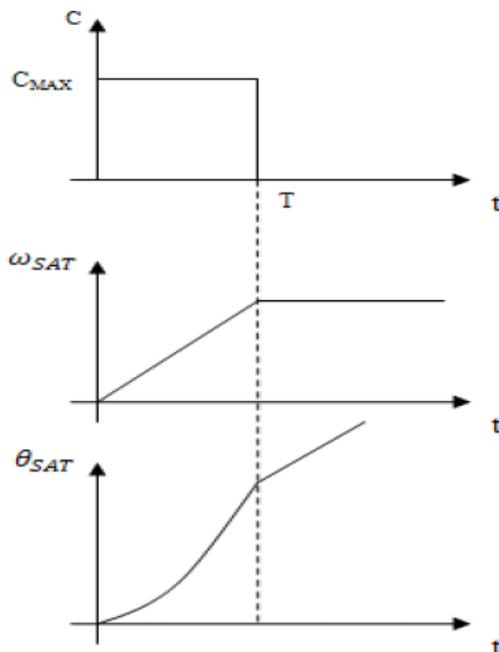
Analogamente per integrazione successive otteniamo l'angolo θ_{SAT} percorso nello stesso tempo:

$$\theta_{SAT} = \int_0^T \omega_{SAT} dt$$

Analizzando le formule sopra riportate si nota che all'applicazione di una coppia costante, al satellite viene impresso un moto rotatorio la cui velocità angolare cresce linearmente col tempo fintanto che \bar{C} è applicata. Una volta cessata il satellite resta in rotazione con velocità angolare costante.

La posizione angolare del satellite invece ha andamento parabolico crescente fintanto che vi è applicata la coppia. Una volta che questa cessa però il satellite continua a ruotare, la sua posizione continua a variare linearmente col tempo.

Si può osservare più chiaramente questo discorso grazie ai seguenti grafici:

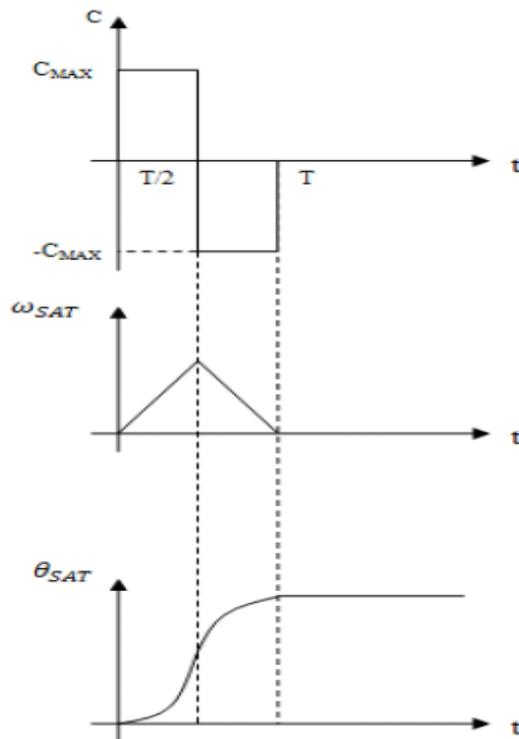


5.2 velocità angolare e posizione angolare in funzione del tempo, con pilotaggio del solenoide a gradino.

Per poter fermare il satellite dopo che questi abbia compiuto la rotazione desiderata, occorre applicare una coppia frenante che riporti a 0 la velocità angolare. Tale coppia deve necessariamente essere uguale e

opposta a quella applicata precedentemente, e deve agire per un tempo identico a T.

Pilotando a gradino il solenoide con una coppia C_{MAX} per un tempo T/2 e con una coppia $-C_{MAX}$ per altri T/2 la situazione che si ottiene è quella riportata in figura 5.3.



5.3 Pilotaggio del solenoide per compiere un angolo predefinito

Si nota che per poter ottenere e mantenere un determinato angolo occorre che la coppia frenante $-C_{MAX}$ sia tale da azzerare la velocità ω_{SAT} .

Le espressioni analitiche per questo tipo di pilotaggio sono le seguenti:

$$\omega_{SAT} = \int_0^T \frac{C}{J_{SAT}} dt = \int_0^{T/2} \frac{C_{MAX}}{J_{SAT}} dt + \int_{T/2}^T \frac{-C_{MAX}}{J_{SAT}} dt$$

Da cui:

$$\theta_{SAT} = \int_0^T \omega_{SAT} dt = -\frac{C_{MAX} \cdot T^2}{4 \cdot J_{SAT}}$$

Dalla scrittura precedente si possono esprimere la coppia massima necessaria per ruotare il satellite dell'angolo θ_{SAT} nel tempo T , o il tempo necessario a compiere la rotazione θ_{SAT} una volta fissata la coppia C_{MAX} :

$$C_{MAX} = -\frac{4 \cdot \theta_{SAT} \cdot J_{SAT}}{T^2}$$

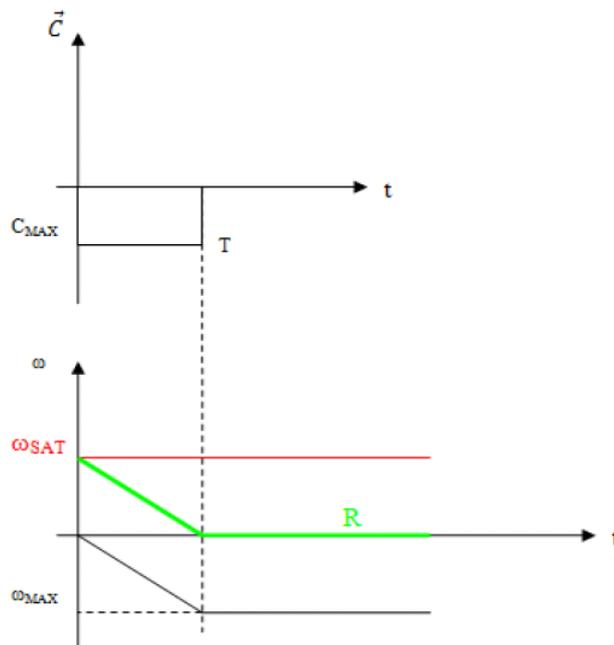
$$T = \sqrt{-\frac{4 \cdot \theta_{SAT} \cdot J_{SAT}}{C_{MAX}}}$$

5.1.3. Reazione a coppie esterne

Se invece il satellite è in moto con una certa velocità angolare ω_{SAT} , dovuta ad esempio all'effetto di coppie disturbo, il solenoide per fermarne la rotazione dovrà applicare una coppia frenante che imprima una velocità uguale e opposta al satellite.

$$\omega_{MAX} = \int_0^T \frac{C}{J_{SAT}} dt = \frac{C_{MAX} \cdot T}{J_{SAT}}$$

Dove porremo $\omega_{SAT} = \omega_{MAX}$ come condizione per fermare il satellite



5.4 Pilotaggio del solenoide per arresto, con velocità iniziale non nulla

Dall'equazione precedente, possiamo ricavare la coppia che il solenoide deve fornire per fermare il satellite con una certa velocità iniziale ω_{SAT} nel tempo T

$$C_{MAX} = \frac{\omega_{SAT} \cdot J_{SAT}}{T}$$

Il corrispondente tempo, fissata la coppia massima applicabile, è:

$$T = \frac{\omega_{SAT} \cdot J_{SAT}}{C_{MAX}}$$

5.1.4. Dimensionamento del solenoide

Per poter generare le coppie esaminate in precedenza è necessario che il solenoide possa essere pilotato in entrambi i sensi, per generare un certo momento di dipolo, e un momento uguale e contrario. A questo scopo si utilizza un tipo di circuito elettronico chiamato ponte ad H.

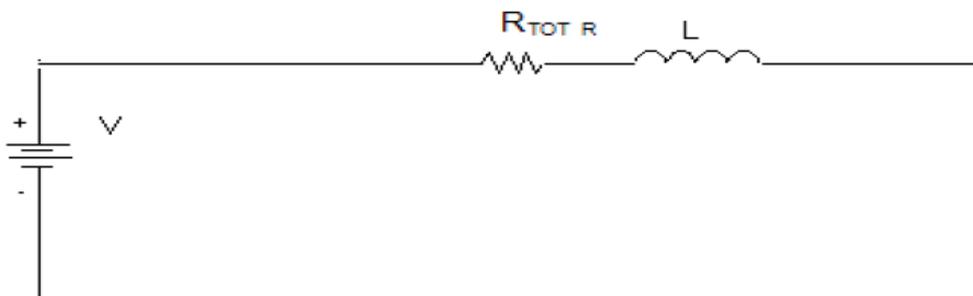
La peculiarità del circuito di pilotaggio è consentire l'alimentazione del solenoide nei due versi e scaricare la corrente immagazzinata nel solenoide tramite diodi di ricircolo.

Verrà tralasciato lo schema del ponte ad H in quanto esula dallo scopo della nostra tesi. Occorre notare che può essere considerato come una resistenza in serie al solenoide nei calcoli successivi, del valore pari a due volte la resistenza di un diodo in conduzione (R_D).

$$R_{TOT} = 2 \cdot R_D + R_{SOL}$$

Dove R_{SOL} è la resistenza del solenoide calcolata come segue:

$$R_{SOL} = \rho \cdot \frac{l}{S}$$



5.5 Circuito schematico del solenoide

Dove:

- l è la lunghezza totale del filo
- S è la sezione del filo
- ρ è la resistività, che dipende dal materiale con cui è realizzato l'avvolgimento. Per il rame $\rho = 1,7 * 10^{-8} \Omega m$

Verrà tralasciato lo studio del transitorio. Tuttavia con opportune piccole approssimazioni si arriva al risultato per cui la corrente media che circola nel circuito è pari alla corrente :

$$I_{MAX} = \frac{V}{R_{TOT}}$$

Possiamo ora effettuare il dimensionamento del solenoide, in quanto siamo in possesso di tutte le equazioni necessarie.

Per trovare la configurazione ideale si sono realizzati quattro layer su circuito stampato. Ogni layer è composto da 60 spire quadrate, la cui spira esterna ha lato 143x143 mm. Le piste sono larghe 0.15mm, spesse 17 μm . Il passo tra ogni pista è di 0.15mm.

Con le formule sopracitate si ricava la resistenza totale di un singolo layer, da cui poi è facile ricavare le resistenze per le quattro configurazioni che andremo a esaminare:

- Due layer in serie
- Quattro Layer in serie
- Due layer in parallelo
- Quattro layer in parallelo

La configurazione con la resistenza maggiore è quella con 4 layer in serie mentre quella con resistenza minore è quella con 4 layer in parallelo. Conoscendo la tensione di alimentazione ($V = 7.2v$) ci ricaviamo la corrente che scorre all'interno dei solenoidi nelle 4 configurazioni. Successivamente avendo le correnti possiamo ricavarci i dipoli magnetici generati e le potenze dissipate in ogni configurazione.

Tabella 5.1 Corrente, dipolo e potenza dissipata dalle quattro configurazioni

Configurazione	Corrente [A]	Dipolo magnetico [Am^2]	Potenza [W]
Due Layer Serie	0.0165	0.0367	0.1191
Quattro layer serie	0.0083	0.0367	0.0595
Due Layer Parallelo	0.0662	0.1468	0.4765
Quattro layer Parall.	0.1324	0.2937	0.9529

Per poter effettuare confronti significativi simuliamo ora una rotazione di 180° nelle quattro configurazioni per testare quale sia effettivamente la più efficiente, quale coppia viene sviluppata e quanto tempo è necessario per eseguire la rotazione.

Per tale test supporremo che il campo magnetico resti costante al valore $\bar{B} = 0.266 \text{ G}$, e il momento d'inerzia $J_{sat} = 0.022 \text{ Kgm}^2$.

Si riportano nella tabella 5.2 i risultati dei calcoli:

Tabella 5.2 Prestazioni nelle quattro configurazioni

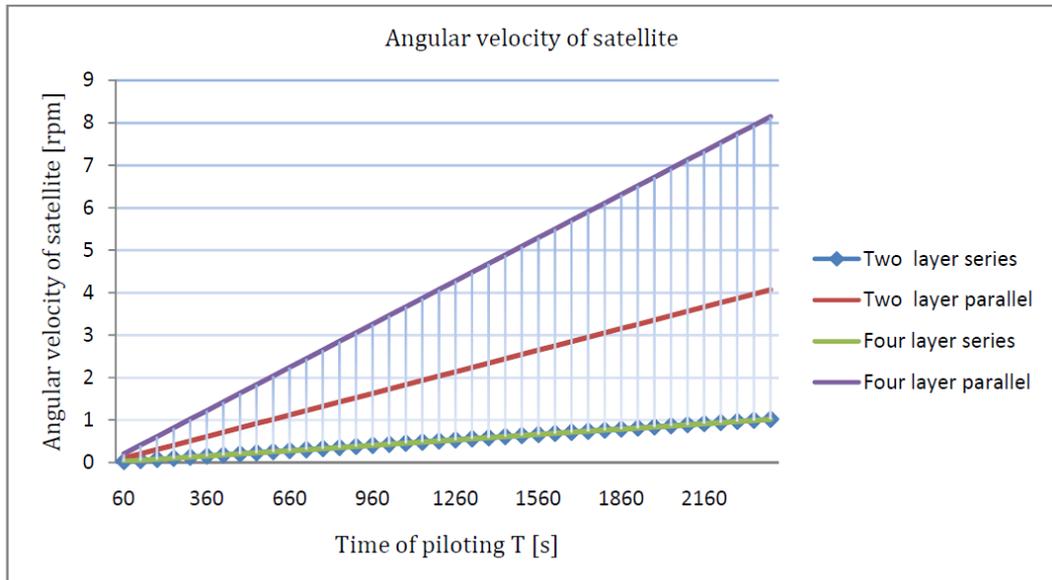
Configurazione	Coppia [Nm]	Tempo [s]	Energia [J]
Due Layer Serie	1*E-06	532	63.3
Quattro layer serie	1*E-06	532	31.7
Due Layer Parallelo	4*E-06	266	126.7
Quattro layer Parallelo	8*E-06	188	179.7

Osservando i dati precedentemente raccolti si nota che la configurazione con 4 layer in parallelo è quella che offre le coppie maggiori e il tempo di rotazione minore tra le altre, anche se con dispendi energetici più elevati. Le altre configurazioni tuttavia seppur con risparmi energetici notevoli, conducono a coppie torcenti piccole e a tempi di rotazione elevati (quasi 9 minuti).

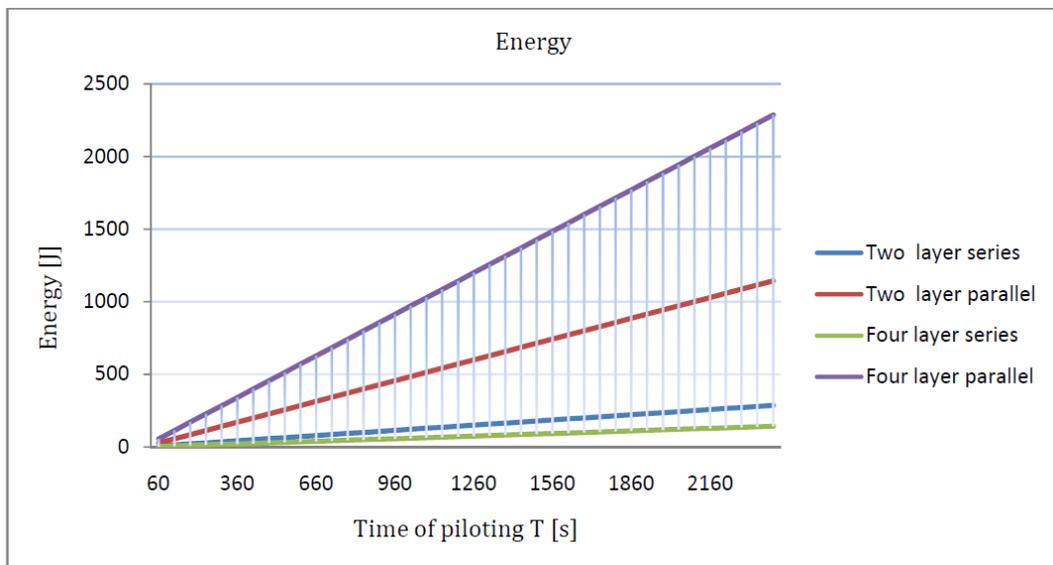
La configurazione con 4 layer in serie si presterebbe bene come configurazione per un riassetto di emergenza, in caso di limitata energia disponibile. Tuttavia la soluzione migliore a livello di prestazioni è quella con 4 layer in parallelo.

Vediamo ora un'ulteriore prova.

Se effettuiamo un pilotaggio a gradino possiamo meglio distinguere il comportamento delle quattro configurazioni in funzione del tempo di pilotaggio. In figura 5.6 è riportata la velocità angolare, in figura 5.7 l'energia necessaria.



5.7 Velocità angolari imposte al satellite nelle quattro configurazioni



5.6 Energia dissipata dalle quattro configurazioni

Risulta interessante esaminare la quantità di energia necessaria per imporre una velocità angolare di un giro al minuto (1 RPM)

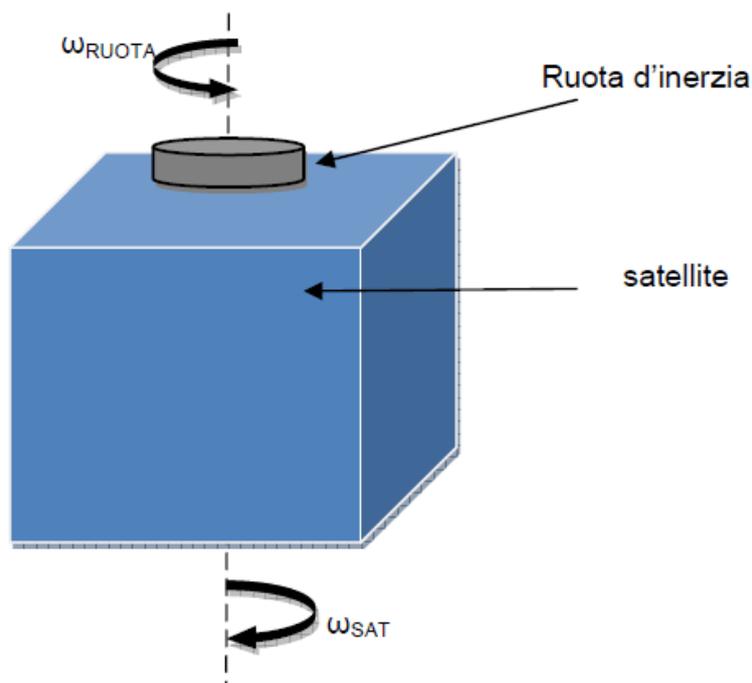
Per la configurazione con due layer in serie e per quella con due in parallelo si ottiene un valore di circa 280 J/RPM, mentre per le configurazioni a quattro layer, sia parallelo che serie, tale valore scende a 140 J/RPM.

Come già anticipato si nota che queste due ultime configurazioni sono ottimali, ma si preferisce la configurazione in parallelo grazie ai minori tempi necessari per eseguire le rotazioni.

5.2. RUOTE D'INERZIA [3] [7] [13]

5.2.1. Fisica delle ruote d'inerzia

Abbiamo già visto in precedenza i principi di funzionamento di una ruota d'inerzia. Si tratta infatti di un disco con una certa massa che viene sfruttata per governare il moto rotatorio di un corpo. Come mostrato in figura 5.8 se la ruota viene azionata con una certa velocità angolare, il satellite si metterà in rotazione nella direzione opposta.



5.8 Ruota d'inerzia a bordo del satellite

Questo comportamento si deve alla legge di conservazione del momento angolare, che viene espressa nel modo che segue:

“Il momento angolare L di un sistema è costante nel tempo se è nulla la coppia totale delle forze esterne che agiscono su di esso”

In assenza di coppie esterne quindi il momento angolare totale si conserva.

$$\sum C_{INTERNE} = \frac{dL}{dt} = 0$$

Quando la ruota d'inerzia viene messa in rotazione, si ha l'effetto di una coppia interna sviluppata dal motore elettrico. Poiché non vi sono coppie esterne, l'effetto di tale coppia sarebbe incrementare il momento angolare. Si rende necessario lo sviluppo di una coppia di reazione uguale e contraria tale da annullare la prima coppia e mantenere costante il momento angolare, soddisfacendo così il principio appena enunciato.

Il momento angolare è una grandezza fisica importante nella descrizione del moto di corpi rotanti attorno ad un asse. Esso tiene conto della distribuzione di massa e da indicazioni sulla resistenza offerta dal corpo a subire variazioni del proprio moto.

La sua espressione matematica è:

$$L = J \cdot \omega$$

Dove J rappresenta il momento d'inerzia e ω la velocità angolare di rotazione.

5.2.2. Pilotaggio delle ruote d'inerzia

Mentre il momento angolare globale del satellite resta costante, nel momento in cui viene azionata la ruota d'inerzia per controllare l'assetto del satellite il satellite viene messo in moto. Ciò comporta una rotazione di entrambi gli organi. Tenendo presente la legge di conservazione espressa come segue:

$$L_{INIZIALE} = L_{FINALE}$$

Esplicitiamo i termini:

$$L_{INIZIALE} = (J_{RUOTA} + J_{SAT}) \cdot \omega_{TOT}$$
$$L_{FINALE} = J_{RUOTA} \cdot \omega_{RUOTA} + J_{SAT} \cdot \omega_{SAT}$$

Da queste formule si vede come occorre intervenire per modificare l'assetto del nostro satellite in funzione del parametro arbitrario, ovvero la velocità angolare impartita alla ruota.

$$\omega_{RUOTA} = \frac{(J_{RUOTA} + J_{SAT}) \cdot \omega_{TOT}}{J_{RUOTA}} - \frac{J_{SAT} \cdot \omega_{SAT}}{J_{RUOTA}}$$

Dove J_{RUOTA} e J_{SAT} sono rispettivamente i momenti d'inerzia della sola ruota e del satellite esclusa la ruota d'inerzia.

Se ad esempio vogliamo fermare il nostro satellite che inizialmente ruota con una velocità $\omega_{TOT} \neq 0$, otteniamo:

$$\omega_{RUOTA} = \omega_{TOT} \frac{(J_{RUOTA} + J_{SAT})}{J_{RUOTA}}$$

Se invece vogliamo mettere in moto il nostro satellite che inizialmente è fermo, le due velocità sono legate nel modo che segue:

$$\omega_{RUOTA} = - \frac{J_{SAT}}{J_{RUOTA}} \cdot \omega_{SAT}$$

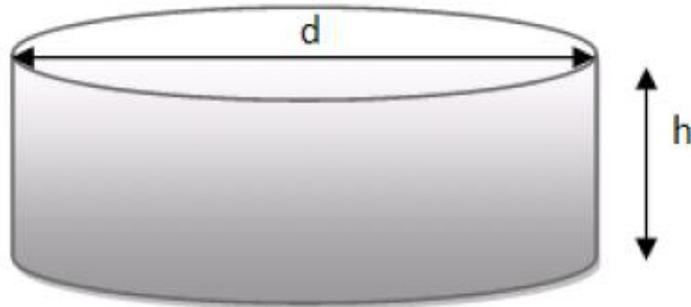
5.2.3. Dimensionamento del sistema

Per dimensionare la ruota d'inerzia occorre porre un limite di circa 1 RPM alla velocità di rotazione del satellite, ovvero alla grandezza ω_{SAT}

Scegliamo poi un cilindro di materiale ad alta densità, come ad esempio il piombo ($\rho_{PB} = 10340 \text{ Kg/m}^3$).

La forma più semplice che possiamo scegliere è dunque un cilindro come quello riportato in figura 5.9:

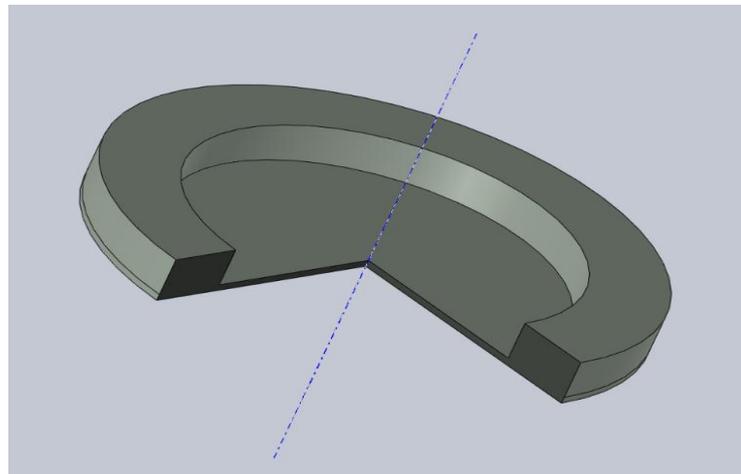
Conoscendo il diametro $d = 30 \text{ mm}$, e l'altezza $h = 3 \text{ mm}$, possiamo calcolarne il momento d'inerzia. $J_{cilindro} = 27 \text{ g} \cdot \text{cm}^2$. Il peso della ruota è invece di 24g.



5.9 *Ruota d'inerzia semplice*

Solitamente si usano ruote con uno spessore minore nella parte centrale e maggiore sulla corona esterna. In questo modo si ottengono ruote con inerzia maggiore e peso minore, quindi più efficienti.

Ad esempio, prendiamo un disco di piombo dello spessore $h_d = 1 \text{ mm}$ e diametro $d_e = 35 \text{ mm}$, al quale sovrappriamo una corona circolare con diametro esterno $d_e = 35 \text{ mm}$, diametro interno $d_i = 25 \text{ mm}$, e uno spessore di $h_c = 6 \text{ mm}$, sempre in piombo. (Fig 5.10)



5.10 *Ruota d'inerzia migliorata*

Ne risulta una ruota del peso di 14g, contro i 24g di quella precedente, e un momento d'inerzia analogo di $J_{cilindro} = 26.5 \text{ g} \cdot \text{cm}^2$.

Nel nostro satellite accoppieremo questa ruota d'inerzia ad un motore elettrico di tipo brushless (maxon EC 32 flat) che presenta un momento d'inerzia del rotore $J_{ROTOR} = 13.9 \text{ g} \cdot \text{cm}^2$, che sommato a quella del cilindro ci fornisce un momento d'inerzia complessivo della ruota

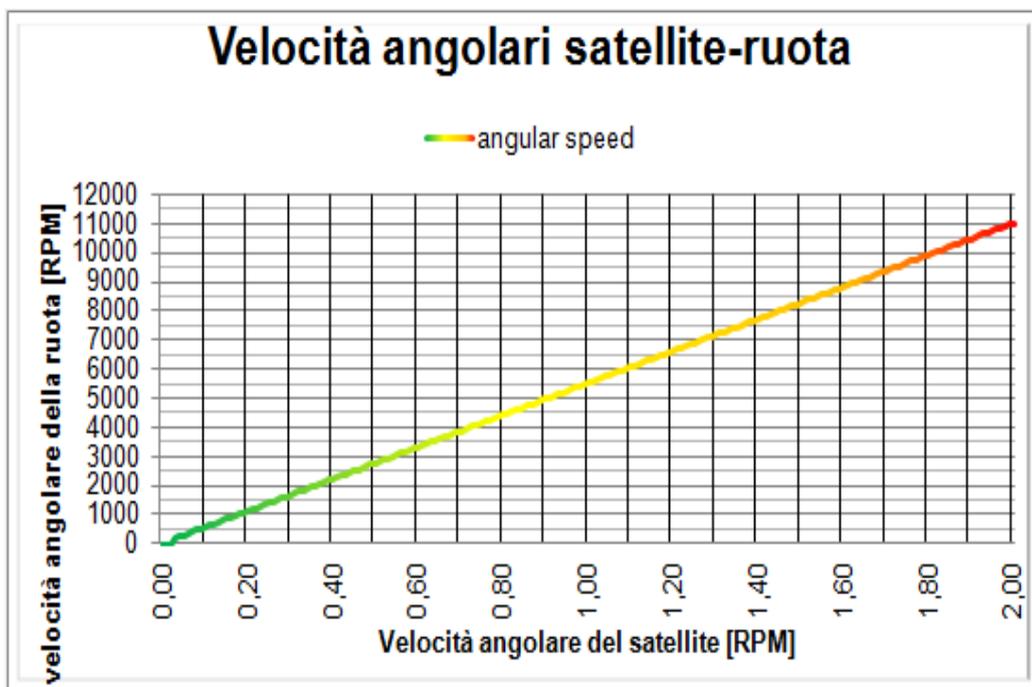
$$J_{RUOTA} = 40.4 \text{ g} \cdot \text{cm}^2 \cong 4 \cdot 10^{-6} \text{ Kg} \cdot \text{m}^2$$

Conosciamo ora tutti i dati necessari per legare la velocità della ruota a quella del satellite, tramite la formula trovata precedentemente:

$$\omega_{RUOTA} = - \frac{J_{SAT}}{J_{RUOTA}} \cdot \omega_{SAT}$$

Riportiamo i dati ottenuti nel grafico 5.10 in cui si nota come sia necessaria una velocità della ruota di circa 5500 RPM per ottenere una rotazione del satellite di 1 RPM.

Per fornire tale velocità di rotazione il motore scelto assorbe circa 4W di potenza. Rispetto al sistema magnetico è molto più oneroso in termini energetici e pertanto occorrerà tenere in conto un utilizzo delle ruote d'inerzia breve nel controllo d'assetto.



5.11 Velocità angolare della ruota riferita alla velocità angolare del satellite

Effettuando un ragionamento analogo è possibile calcolare quanto deve valere l'inerzia totale del sistema ruota+rotore anche nei due casi di satellite aramis considerati per mantenere inalterato, a parità di motore considerato (Datasheet in appendice D.1), il rapporto:

$$\frac{J_{SAT}}{J_{RUOTA}} = 5500$$

Il motore impiegato sarà quindi da rivedere con un altro in grado di lavorare in un punto di progetto più consono.

Vediamo quindi che per la configurazione cubica del satellite AraMiS, il cui momento d'inerzia è costante per tutti e tre gli assi ($J_{sat} = 0.045 [kg \cdot m^2]$) ci occorre un'inerzia per ogni singola ruota pari a

$$J_{ruota_{AC}} = 8.18 \cdot 10^{-6} [kg \cdot m^2]$$

Per la configurazione assialsimmetrica ci occorreranno sicuramente dei motori ridimensionati tali da poter mettere in rotazione masse dall'inerzia superiore, infatti:

$$J_{ruota_z} = \frac{J_{sat_z}}{5500} = 1.045 \cdot 10^{-4} [kg \cdot m^2]$$

$$J_{ruota_x} = J_{ruota_y} = \frac{J_{sat_{x,y}}}{5500} = 5.156 \cdot 10^{-4} [kg \cdot m^2]$$

Poiché il satellite in configurazione assialsimmetrica a sezione esagonale non presenta le ruote d'inerzia allineate con gli assi come nella configurazione cubica, si può calcolare tramite procedimenti matematici più complessi, il modo di pilotare efficacemente le ruote.

Sfruttando quindi queste tecniche di controllo, azionando più ruote contemporaneamente (sono 25 in totale quelle capaci di generare momenti sul piano XY) risulta possibile ottenere l'effetto di una ruota d'inerzia più grande, come quella calcolata.

Poiché si tratta di geometrie più complesse, in questa trattazione non verranno considerate simulazioni di assetto per questo genere di satellite anche se ne verrà illustrato l'algoritmo risolutivo.

6. PROGRAMMI DI CONTROLLO D'ASSETTO

6.1. CONSIDERAZIONI PRELIMINARI

Completato il processo preliminare di dimensionamento degli attuatori in funzione del satellite e delle condizioni operative occorre esaminare il problema del controllo d'assetto. La prima operazione da compiere è avere ben chiaro quali generi di manovre si vuole far compiere al satellite e con quale genere di attuatore, tenendo in gran conto sia le condizioni esterne che il posizionamento rispetto ai vari sistemi di riferimento.

Le manovre base del nostro satellite saranno in generale le seguenti:

- *Spin, De-spin*, si tratta di imprimere una velocità angolare rispetto a 1, 2 o 3 assi al satellite. Ciò può portare ad esempio il satellite da fermo in rotazione, da rotazione a fermo, o semplicemente variare la velocità angolare.
- *Rotazioni angolari*, significa, data la posizione e l'assetto del satellite, imprimergli una rotazione finita, ad esempio per orientare un pannello solare.
- *Mantenimento di un assetto fisso*, rispetto al sistema di riferimento di terra, come ad esempio asse z puntato sempre verso il centro della terra, asse X sempre parallelo al campo magnetico, etcc.

Questo comporta il mantenimento costante di alcuni determinati angoli con il sistema di riferimento.

- *Traking a punto fisso*, significa puntare col satellite sempre lo stesso punto sulla superficie terrestre, ad esempio la stazione di terra con l'asse delle antenne (asse z)

Come ipotesi semplificative già introdotte (vedi paragrafo 4.3.2 *più sopra*) considereremo l'orbita come circolare di centro uguale a quello della terra, inclinata di 86° e con periodo $T=90'$.

Il sistema di controllo che verrà presentato, simulato e analizzato non vuole avere le pretese di essere perfetto in termini di precisione e robustezza, ma piuttosto di essere un valido punto di partenza per il perfezionamento di un algoritmo di controllo dell'assetto, una volta ottenuti tutti i dati necessari sia della missione che strutturali del satellite.

6.2. ATTITUDE CONTROL AND DETERMINATION SYSTEM (ACDS).

L'Attitude control and determination system è il "cervello" del nostro satellite. Esso ha il compito di interpretare le istruzioni che provengono da terra ed elaborarle, in modo da fornire in uscita il pilotaggio necessario agli attuatori per compiere la missione richiesta.

Il primo problema da affrontare è la conversione tra i sistemi di riferimento in cui sono espressi gli assetti desiderati, e quello in cui sono ricevuti dall'ACDS da parte dei sensori. Una volta che gli assetti sono espressi rispetto al medesimo sistema di riferimento occorre determinare le quantità angolari che determinano la rotazione necessaria a svolgere la missione.

6.3. SPIN E DE-SPIN

Il controllo di assetto di spin e de-spin si occupa di portare il satellite ad una desiderata velocità di rotazione, ad esempio per garantirne la stabilità giroscopica, oppure, se effettuato tramite magnetotorquer, per desaturare le ruote d'inerzia con velocità angolare diversa da 0. Spesso lo scopo del de-spin è quello di fermare la rotazione del satellite generata dagli effetti di disturbo dell'ambiente atmosferico.

A livello software, tramite input si riceve il comando che contiene le informazioni relative al vettore velocità angolare che si desidera impostare al satellite. Tale vettore contiene le tre componenti della velocità angolare (ω_{xa} ω_{ya} ω_{za}) espresse in assi corpo principali del satellite. Contestualmente dai sensori installati a bordo si ricavano le misure istantanee del vettore velocità angolare posseduto dal satellite, allineate con gli assi principali.

Si procede quindi successivamente al confronto tra i due vettori ottenendo un vettore di variazione di velocità angolare, le cui tre componenti rappresentano la singola variazione di velocità angolare.

Tramite la seguente formula:

$$\Delta\omega_{RUOTA} = - \frac{J_{SAT}}{J_{RUOTA}} \cdot \Delta\omega_{SAT}$$

Si ricava quale deve essere la velocità da imprimere alla ruota per ottenere la variazione di velocità $\Delta\omega_{SAT}$ corrispondente all'assetto desiderato.

Ciò è vero, almeno in prima approssimazione, nel caso del satellite cubico in cui si considerino le inerzie lungo i tre assi come possedenti circa lo stesso valore. In questo caso infatti le equazioni del moto sono disaccoppiate e la loro soluzione è più semplice da trovare. Nella realtà la situazione può essere molto diversa.

Dall'equazione del momento della quantità di moto

$$\bar{L} = \bar{J}\bar{\omega}$$

Derivandola e sviluppandola rispetto ai tre assi principali porta alle seguenti 3 equazioni, che prendono il nome di equazioni di Eulero:

$$\begin{cases} M_x = J_x \dot{\omega}_x - (J_z - J_y) \omega_y \omega_z \\ M_y = J_y \dot{\omega}_y - (J_x - J_z) \omega_x \omega_z \\ M_z = J_z \dot{\omega}_z - (J_y - J_x) \omega_x \omega_y \end{cases}$$

Dalle quali si osserva la presenza degli effetti dovuti all'induzione di velocità angolari sugli assi diversi da quelli considerati. Si tratta di coppie giroscopiche, e sono effetti indesiderati che complicano la trattazione del

problema. Si vengono infatti a creare accoppiamenti inerziali tra i gradi di libertà del sistema, rendendo di fatto il moto rotatorio del satellite non svincolato lungo i tre assi.

Si nota che l'effetto dovuto a tali coppie è tanto più sensibile quanto più la differenza di inerzia tra gli assi è elevata. Per un satellite molto affusolato, come ad esempio nella configurazione prismatica esagonale, l'inerzia rispetto all'asse z è molto minore rispetto a quella nelle altre direzioni e il satellite nel complesso sarà più soggetto a tali effetti..

Per quanto riguarda i magnetotorquer non sono idonei a ottenere spin in brevi periodi di tempo, e il loro impiego è perlopiù ristretto alla desaturazione del satellite ogniqualvolta il campo magnetico terrestre si trovi in una situazione favorevole allo scopo.

Per la caratteristica generazione di coppia degli attuatori magnetici, che impedisce un controllo della rotazione attorno alla direzione del campo terrestre, occorre un attento esame delle condizioni in cui il satellite si trova a operare.

Nelle zone a bassa latitudine ad esempio si hanno campi magnetici quasi allineati alla tangente all'orbita, orientati verso nord. Ciò consente dunque di generare coppie attorno all'asse z o coppie attorno all'asse est-ovest. Alle alte latitudini invece, nelle zone polari, il campo magnetico è quasi esclusivamente verticale entrante o uscente dal centro della terra. Questo allineamento consente una generazione di coppie attorno agli assi x e y giacenti sul piano dell'orbita.

6.3.1. Algoritmo di spin

Consideriamo ora il modello matematico da applicare; Partendo dall'equazione del momento angolare

$$\bar{L} = \bar{J} * \bar{\omega}_i$$

Ne calcoliamo la derivata rispetto al tempo in forma vettoriale

$$\bar{M}_e = \frac{d}{dt} \bar{J} \cdot \bar{\omega}_i + \bar{J} \cdot \frac{d}{dt} \bar{\omega}_i + S(\bar{\omega}) \cdot (\bar{J} * \bar{\omega}_i)$$

E quindi

$$\bar{M}_e = \bar{J} \cdot \bar{\dot{\omega}}_b + S(\omega) \cdot (\bar{J} \cdot \bar{\omega}_l)$$

La formula precedente è l'espressione della conservazione del momento angolare, infatti in assenza di coppie esterne $M_e = 0$ il momento angolare resta costante (derivata nulla).

Sviluppando la scrittura e in funzione di un sistema di riferimento baricentrico allineato con gli assi principali di inerzia tale che la matrice \bar{J} sia diagonale, si ottengono le equazioni di Eulero, che considerando l'assenza di coppie esterne applicate al satellite, possono essere riscritte come

$$\begin{cases} J_x \dot{\omega}_x - (J_z - J_y) \omega_y \omega_z = 0 \\ J_y \dot{\omega}_y - (J_x - J_z) \omega_x \omega_z = 0 \\ J_z \dot{\omega}_z - (J_y - J_x) \omega_x \omega_y = 0 \end{cases}$$

Anche nel caso di riferimento baricentrico e principale di inerzia, le equazioni di Eulero sono difficili da integrare e quindi in genere si ricorre a metodi di integrazione numerica al calcolatore. Tuttavia la soluzione analitica si può ricavare in alcuni casi semplici e può essere usata per la valutazione dei risultati numerici.

Specializzando le equazioni nel caso in cui si consideri un satellite come PiCPoT o come AraMiS a configurazione cubica minima, si avrà che le inerzie lungo i tre assi saranno, se non identiche nel caso reale, comunque molto simili, rendendo minimo l'effetto delle coppie giroscopiche.

In questo caso la semplificazione introdotta porta alle seguenti equazioni

$$\begin{cases} J_x \dot{\omega}_x = 0 \\ J_y \dot{\omega}_y = 0 \\ J_z \dot{\omega}_z = 0 \end{cases}$$

Che una volta integrate, riportando solo la prima per brevità, si ottiene la relazione

$$(J_{RUOTA} + J_{SAT}) \cdot \omega_{INIZIALE} = J_{SAT} \cdot \omega_{SAT} + \omega_{RUOTA} \cdot J_{RUOTA}$$

La trattazione semplificata del pilotaggio in questo caso è già stata vista nel paragrafo 5.2.2, e si perviene alla velocità da imporre alla ruota:

$$\omega_{RUOTA} = \frac{(J_{RUOTA} + J_{SAT}) \cdot \omega_{Iniziale}}{J_{RUOTA}} - \frac{J_{SAT} \cdot \omega_{Finale}}{J_{RUOTA}}$$

Occorre quindi evitare di imporre velocità angolari del satellite troppo elevate per non superare il limite della ruota, che vede una velocità di rotazione nominale pari a 5500 RPM, che corrisponde a circa 1 RPM per il satellite.

I calcoli svolti sulle velocità dell'attuatore sono riferiti a PiCPoT, ma è un'approssimazione accettabile considerare rotazioni del satellite inferiori a 2RPM.

Se tale limite venisse superato si andrebbe incontro al fenomeno detto di saturazione della ruota d'inerzia. Si tratta di una condizione per cui il rotore si trova alla velocità di sincronismo e non è più possibile ottenere velocità superiori. Questo causa un'impossibilità di controllo del velivolo sull'asse saturato. Con metodi che studieremo più avanti vedremo come è possibile rimediare alla saturazione.

Come abbiamo esaminato precedentemente le architetture possibili sono molte. Nel caso in cui si consideri un satellite assialsimmetrico, come nel caso della configurazione esagonale prismatica, è possibile introdurre alcune considerazioni per cercare di semplificare il calcolo.

Innanzitutto l'inerzia rispetto all'asse z è molto minore rispetto alle inerzie rispetto agli altri due assi, che risultano invece uguali tra loro.

$$J_z \ll J_x \quad J_z \ll J_y \quad J_x = J_y$$

Per cui le equazioni di Eulero si riscrivono come

$$\begin{cases} J_x \dot{\omega}_x - (J_z - J_y) \omega_y \omega_z = 0 \\ J_y \dot{\omega}_y - (J_x - J_z) \omega_x \omega_z = 0 \\ J_z \dot{\omega}_z = 0 \end{cases}$$

Da cui si ricava che la velocità lungo l'asse z è svincolata, e pertanto è generabile senza tener conti di effetti parassiti non voluti, infatti:

$$\frac{d}{dt} J_z \omega_z = 0$$

$$\omega_z = \text{costante} = \omega_{z0} = \omega_{RW} \frac{J_{RW}}{J_{SAT}} + \omega_{spin}$$

In cui si considera ω_{RW} la velocità della ruota di reazione necessaria a raggiungere la velocità angolare ω_{spin} del satellite, richiesta dall'ACDS.

Per quanto riguarda le prime due equazioni, rappresentano un genere di spin poco utilizzato data la tipologia di satelliti. Il moto che si verrà a creare è di tipo conico attorno all'asse z. E' comunque interessante svilupparne l'analisi.

Tenendo in considerazione le prime due equazioni, possiamo raggruppare i termini in un'unica costante caratteristica λ , in cui si considera ω_z come già raggiunta, e quindi costante anch'essa.

$$\lambda = \frac{(J_z - J_x)}{J_x} \omega_z = - \frac{(J_y - J_z)}{J_y} \omega_z$$

Possiamo quindi riscrivere le prime due equazioni di eulero come:

$$\begin{cases} \dot{\omega}_x + \lambda \omega_y = 0 \\ \dot{\omega}_y - \lambda \omega_x = 0 \end{cases}$$

Effettuando l'operazione di derivata della prima equazione e procedendo con la sostituzione della seconda equazione otteniamo

$$\ddot{\omega}_x + \lambda^2 \omega_x = 0$$

Che ha la forma di un oscillatore armonico, e dalla seconda equazione, la relazione:

$$\omega_y = - \frac{\dot{\omega}_x}{\lambda}$$

Tenendo conto delle condizioni iniziali generiche del satellite

$$\begin{cases} \omega_x(0) = \omega_{x0} \\ \dot{\omega}_x(0) = \dot{\omega}_{x0} \\ \omega_y(0) = \omega_{y0} \\ \dot{\omega}_y(0) = \dot{\omega}_{y0} \end{cases}$$

Si ottengono dunque le due equazioni

$$\begin{cases} \omega_x = \omega_{x0} \cos(\lambda t) - \omega_{y0} \sin(\lambda t) \\ \omega_y = \omega_{x0} \sin(\lambda t) + \omega_{y0} \cos(\lambda t) \end{cases}$$

Si osserva che il vettore velocità angolare può essere scomposto in due vettori, uno fisso orientato lungo z e uno variabile, giacente sul piano xy, e rotante con velocità angolare λ attorno a z, e modulo pari a:

$$|\omega_{xy}| = \sqrt{\omega_x^2 + \omega_y^2}$$

Per ottenere il pilotaggio delle ruote d'inerzia in funzione della velocità di spin richiesta si considerino le componenti della velocità angolare x e y espressi rispetto agli assi corpo del satellite all'istante iniziale $t_0 = 0$.

$$\begin{cases} \omega_x = (\omega_{spin\ x} + \omega_{RWx} \frac{J_{RW}}{J_{SAT}}) \cos(\lambda t) - (\omega_{spin\ y} + \omega_{RWy} \frac{J_{RW}}{J_{SAT}}) \sin(\lambda t) \\ \omega_y = (\omega_{spin\ x} + \omega_{RWx} \frac{J_{RW}}{J_{SAT}}) \sin(\lambda t) + (\omega_{spin\ y} + \omega_{RWy} \frac{J_{RW}}{J_{SAT}}) \cos(\lambda t) \end{cases}$$

Uguagliando queste due equazioni con le precedenti si ottengono due equazioni in due incognite in funzione del tempo. Il pilotaggio delle ruote per tale puntamento deve essere continuo per mantenere il vettore velocità angolare aderente alle richieste dell'ACDS.

6.4. ROTAZIONE DEL SATELLITE

Una delle principali manovre che un satellite deve essere in grado di compiere è una rotazione spaziale attorno a uno o più assi, in modo che il satellite possa svolgere al meglio le sue funzioni.

Una rotazione spaziale può portare il satellite a orientare una certa faccia verso il sole, un sensore in una determinata porzione di cielo, le antenne verso la superficie terrestre e orientare il payload. Alcune di queste manovre sono più complesse di una semplice rotazione, che comunque costituisce il punto di partenza per un controllo di assetto più complicato.

6.4.1. Individuazione del problema

La condizione di partenza per una rotazione è avere un satellite privo di rotazioni, o al più dotato di singolo spin lungo un solo asse. Questo perché spin più complessi comprenderebbero tutti e tre gli assi, rendendo di fatto inutile il calcolo della rotazione.

Le condizioni di partenza sono gli angoli di Eulero (Ψ, Θ, Φ) della terna di assi corpo rispetto agli assi orbita, e gli angoli di Eulero del nuovo assetto, sempre rispetto agli assi orbita. Definiamo E_s e E_a le due terne di angoli di Eulero.

$$E_s = \begin{pmatrix} \Psi_s \\ \Theta_s \\ \Phi_s \end{pmatrix} \quad E_a = \begin{pmatrix} \Psi_a \\ \Theta_a \\ \Phi_a \end{pmatrix}$$

La prima difficoltà risulta nel trovare la terna di angoli di Eulero che portino direttamente a coincidere l'assetto attuale con quello desiderato.

Si costruisce dapprima la matrice teorica che permette il passaggio dal frame F_s al frame F_a in funzione dei 3 angoli reali di rotazione E_r .

$$R_{AS} = \begin{bmatrix} \cos \Theta_r \cos \Psi_r & \cos \Psi_r \sin \Phi_r \sin \Theta_r - \cos \Phi_r \sin \Psi_r & \cos \Phi_r \cos \Psi_r \sin \Theta_r + \sin \Psi_r \sin \Phi_r \\ \cos \Theta_r \sin \Psi_r & \sin \Psi_r \sin \Phi_r \sin \Theta_r + \cos \Phi_r \cos \Psi_r & \cos \Phi_r \sin \Psi_r \sin \Theta_r - \cos \Psi_r \sin \Phi_r \\ -\sin \Theta_r & \cos \Theta_r \sin \Phi_r & \cos \Phi_r \cos \Theta_r \end{bmatrix}$$

Tale matrice di rotazione si ricava come esplicitato al paragrafo 1.3.5. come prodotto delle tre rotazioni elementari dei tre angoli di Eulero.

Possiamo uguagliare tale matrice formale con il prodotto di due rotazioni successive, tali R_{os} e R_{ao} , che portano la terna iniziale a coincidere prima con gli assi orbita, e poi la rotazione che porta gli assi orbita a coincidere con quelli del nuovo assetto. Le grandezze necessarie per compiere questa rotazione sono tutte note.

$$R_{OS} = \begin{bmatrix} \cos \Psi_s & \sin \Psi_s & 0 \\ -\sin \Psi_s & \cos \Psi_s & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos \Theta_s & 0 & \sin \Theta_s \\ 0 & 1 & 0 \\ -\sin \Theta_s & 0 & \cos \Theta_s \end{bmatrix} \begin{bmatrix} \cos \Phi_s & \sin \Phi_s & 0 \\ -\sin \Phi_s & \cos \Phi_s & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$R_{OA} = \begin{bmatrix} \cos \Psi_a & \sin \Psi_a & 0 \\ -\sin \Psi_a & \cos \Psi_a & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos \Theta_a & 0 & \sin \Theta_a \\ 0 & 1 & 0 \\ -\sin \Theta_a & 0 & \cos \Theta_a \end{bmatrix} \begin{bmatrix} \cos \Phi_a & \sin \Phi_a & 0 \\ -\sin \Phi_a & \cos \Phi_a & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$R_{AS} = R_{OS} \cdot R_{OA}^T$$

Eguagliando l'equazione ricavata in funzione degli angoli di Eulero E_r con la matrice di rotazione R_{OA} si ricavano agevolmente i valori dei tre angoli di Eulero cercati.

$$E_r = \begin{cases} \Theta_r = -\sin^{-1} R_{AS}(3,1) \\ \Phi_r = \sin^{-1} \left(\frac{R_{AS}(3,2)}{\cos \Theta_r} \right) \\ \Psi_r = \cos^{-1} \left(\frac{R_{AS}(1,1)}{\cos \Theta_r} \right) \end{cases}$$

Tenendo conto però delle due soluzioni alternative delle funzioni arcoseno e arco coseno nella seconda e terza equazione. Nello specifico può diventare:

$$E_r = \begin{cases} \Theta_r = -\sin^{-1} R_{AS}(3,1) \\ \Phi_r = \pi - \sin^{-1} \left(\frac{R_{AS}(3,2)}{\cos \Theta_r} \right) \\ \Psi_r = \cos^{-1} \left(\frac{R_{AS}(1,1)}{\cos \Theta_r} \right) \end{cases}$$

$$E_r = \begin{cases} \Theta_r = -\sin^{-1} R_{AS}(3,1) \\ \Phi_r = \sin^{-1} \left(\frac{R_{AS}(3,2)}{\cos \Theta_r} \right) \\ \Psi_r = -\cos^{-1} \left(\frac{R_{AS}(1,1)}{\cos \Theta_r} \right) \end{cases}$$

$$E_r = \begin{cases} \Theta_r = -\sin^{-1} R_{AS}(3,1) \\ \Phi_r = \pi - \sin^{-1} \left(\frac{R_{AS}(3,2)}{\cos \Theta_r} \right) \\ \Psi_r = -\cos^{-1} \left(\frac{R_{AS}(1,1)}{\cos \Theta_r} \right) \end{cases}$$

La soluzione corretta si ottiene, una volta sostituiti i valori degli angoli trovati, dal confronto dei segni sugli altri termini della matrice.

6.4.2. Rotazione in tempo minimo

Una delle caratteristiche principali di un sistema di controllo di assetto sta nella rapidità con cui si può ottenere un determinato spostamento. In ambito spaziale questa peculiarità passa solitamente in secondo piano, in favore di manovre con consumo minimo.

Effettuando una breve analisi su un satellite in rotazione rispetto ad un singolo asse principale di inerzia possiamo valutare gli algoritmi necessari, svincolandoci dalla complessità matematica delle equazioni di Eulero.

Indicando con α l'angolo da percorrere, e considerando i piccoli spostamenti, la dinamica del satellite assume la forma

$$M = J\ddot{\alpha}$$

Poiché si vuole ottenere il tempo minimo occorre effettuare una minimizzazione vincolata del tipo

$$\min H = \int_{t_0}^{t_f} dt + \lambda \int_{t_0}^{t_f} (\dot{x} - Ax - Bu)dt$$

Che con le semplificazioni fatte e tenendo conto delle seguenti relazioni ci conduce

$$\dot{\alpha} = \frac{d\alpha}{dt} \Rightarrow dt = \frac{d\alpha}{\dot{\alpha}}$$

$$H = \int_{\alpha_0}^{\alpha_f} \frac{1}{\dot{\alpha}} d\alpha$$

Definendo poi l'accelerazione angolare col parametro u otteniamo un indicatore della coppia disponibile.

$$u = \ddot{\alpha} = \frac{M}{I}$$

Se l'indice u è sufficientemente elevato il problema del tempo minimo non è più necessario in quanto la coppia diventa impulsiva, cioè si considera come istantaneo il raggiungimento della velocità angolare scelta.

Integrando le precedenti si ottiene

$$\dot{\alpha} = \dot{\alpha}_0 + ut$$

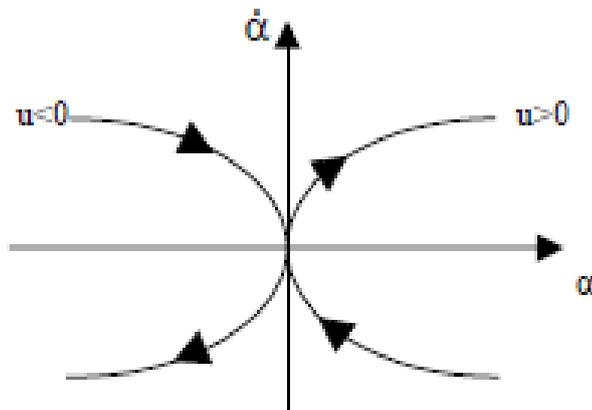
$$\alpha = \alpha_0 + \dot{\alpha}_0 t + \frac{1}{2} ut^2$$

Ricavando il tempo dalla prima e sostituendo nella seconda:

$$t = \frac{\alpha - \alpha_0}{u}$$

$$\alpha - \alpha_0 = \frac{1}{u} \dot{\alpha}_0 (\alpha - \alpha_0) + \frac{1}{2u} (\alpha - \alpha_0)^2$$

La formulazione così ottenuta rappresenta due parabole nel piano di fase, distinte tra loro per il segno di u , e quindi dal segno della coppia applicata.



6.1 Parabole nel piano di fase.

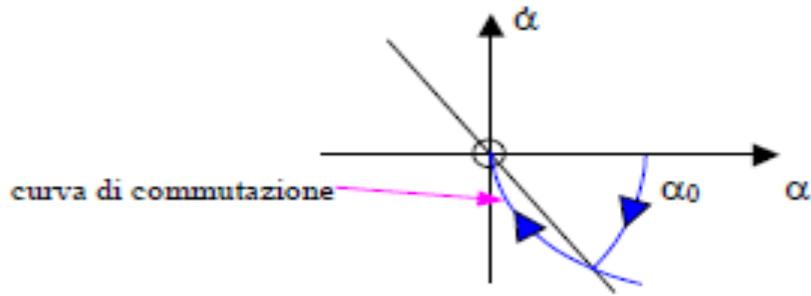
Se vogliamo portarci da una condizione da α_0 ad $\alpha_f = 0$ in soli due step consideriamo la parabola ad u_{max} vista precedentemente partendo dalla condizione α_0 , fino a intersecare la parabola di equazione

$$\alpha = \frac{1}{2u} \dot{\alpha}^2$$

Centrata in $\alpha_f = 0$, e prendiamo

$$u = -u_{max} \cdot \text{sign} \left(\alpha - \frac{1}{2u} \dot{\alpha} |\dot{\alpha}| \right)$$

Ottenendo una relazione grafica del seguente tipo

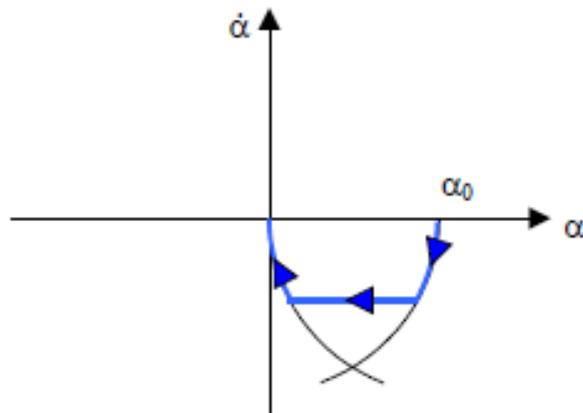


6.2 diagramma a tempo minimo e curva di commutazione

Si dimostra agevolmente essere la più rapida legge di controllo per partire e arrivare con velocità nulla. Il sistema accelera al massimo fino a metà e poi decelera.

Il tempo è implicito nella legge di controllo. A tal proposito è interessante variare la legge appena esaminata in funzione di un tempo di esecuzione maggiore di t_{min} che può essere vista come una manovra per minimizzare i consumi diminuendo l'accelerazione massima richiesta.

La composizione di questa manovra è vista come una manovra a tempo minimo con tratti costanti, come mostrato in figura 6.3



6.3 manovra a consumo ridotto

In questo caso il cambio di parabola avviene dopo un certo tempo finito. In questo lasso di tempo il satellite ruota a velocità costante. Fissare il tempo significa infatti determinare la massima velocità di rotazione del satellite.

6.4.3. Algoritmo di rotazione

Il modello matematico impiegato si basa sulle premesse appena fatte, considerando il fattore u come sufficientemente elevato. Inoltre la manovra non avviene a tempo minimo ma scegliendo tempi di azione tali da non saturare le ruote di reazione, ponendo come limite alla rotazione del satellite un valore di circa 1 RPM.

Si procede innanzitutto con una manovra di despin che porta il satellite a velocità angolare nulla (come visto precedentemente). A questo punto determinati gli angoli di Eulero mediante function (appendice B.5) imponiamo che varino linearmente col tempo, con derivata costante pari a

$$E_r = \begin{pmatrix} \Psi_r \\ \Theta_r \\ \Phi_r \end{pmatrix} \Rightarrow \dot{E}_r = \begin{pmatrix} \dot{\Psi}_r \\ \dot{\Theta}_r \\ \dot{\Phi}_r \end{pmatrix} = \begin{pmatrix} \Psi_r/t \\ \Theta_r/t \\ \Phi_r/t \end{pmatrix}$$

Dove t è il tempo di manovra. Dati angoli di Eulero e loro derivate è possibile calcolare il vettore velocità angolare da imprimere al satellite per il tempo t in modo da ottenere la rotazione.

$$\omega_r \begin{cases} \omega_{rx} = \dot{\Psi}_r \sin \Theta_r \sin \Phi_r + \dot{\Theta}_r \cos \Phi_r \\ \omega_{ry} = \dot{\Psi}_r \sin \Theta_r \cos \Phi_r - \dot{\Theta}_r \sin \Phi_r \\ \omega_{rz} = \dot{\Psi}_r \cos \Theta_r + \dot{\Phi}_r \end{cases}$$

Il tempo di manovra viene determinante mediante sostituzione ciclica di valori del tempo di manovra fino ad ottenere che tutte le componenti del vettore velocità siano di modulo inferiore alla velocità massima impostata (1 RPM).

Se la condizione è verificata si pone $T_{manovra} = t$ e si impone il vettore di spin ω_r al satellite per un tempo $T_{manovra}$. A questo punto risulta necessario riapplicare l'algoritmo di spin precedentemente formulato, sia per quanto riguarda satelliti cubici che assialsimmetrici, come nei due casi da noi considerati.

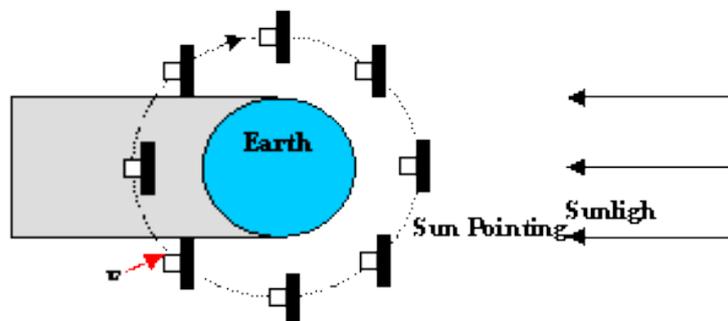
Per il file di matlab fare riferimento all'appendice A.3

6.5. PUNTAMENTO INERZIALE

Il puntamento inerziale è un tipo di assetto che prevede il mantenimento da parte del satellite di un assetto in riferimento ad assi di tipo inerziale. Come sistema inerziale viene assunto il sistema ECI.

Il riferimento ECI è in realtà un sistema pseudo inerziale, in quanto la terra non è fissa ma percorre un'orbita attorno al sole nel suo moto di rivoluzione. Tuttavia per i tempi caratteristici della nostra orbita tali effetti si possono considerare trascurabili.

Gli scopi per cui si ricorre al puntamento inerziale sono molti, tra cui citiamo, ad esempio, la necessità di avere un puntamento costante dei pannelli solari verso il sole, in qualunque punto dell'orbita ci si trovi, per permettere la ricarica delle batterie. Altre funzioni possono legarsi alle necessità di taluni telescopi spaziali per mantenere il puntamento sui vari corpi celesti per effettuare le osservazioni.



6.4 Esempio di satellite Sun-Pointing

6.5.1. Algoritmo di puntamento inerziale

Concettualmente la realizzazione del controllo di puntamento inerziale è molto semplice. Da considerazioni geometriche si ricava infatti che l'angolo formato tra la direzione x_{sat} del satellite a un generico tempo t e la direzione dell'asse inerziale di riferimento, preso parallelo a x_{sat} al tempo $t_0 = 0$ è uguale all'angolo percorso dal satellite lungo la sua orbita (che precedentemente è stata assunta come circolare).

A livello di software si definisce innanzitutto il tempo di manovra t_{man} . Tale tempo definisce, rispetto all'istante iniziale t_0 in cui viene ricevuto il comando, la posizione del satellite in cui inizierà la fase di sincronizzazione con gli assi inerziali. Il tempo di manovra deve essere un lasso di tempo sufficiente a garantire al satellite la possibilità di compiere una rotazione.

Successivamente, ricorrendo alle functions riportate in appendice B, si ricava qual è l'assetto inerziale ricercato all'istante $t_0 + t_{man}$ rispetto agli assi orbita. Una volta determinato questo resta invariato rispetto al frame orbitale. Occorre pertanto una rotazione, che deve avvenire in un tempo $t_{rot} < t_{man}$ tale da portare a coincidere gli assi del satellite con l'assetto desiderato, esattamente come già analizzato nel sottocapitolo precedente.

Terminata la rotazione occorre attendere la posizione $t_0 + t_{man}$ con un ciclo di *Idle* della durata di $t_{idle} = t_{man} - t_{rot}$.

Giunti alla posizione desiderata troviamo che il satellite risulta allineato col riferimento inerziale scelto. Occorre dunque imprimere al satellite una velocità angolare uguale in modulo e direzione, e contraria in verso, a quella di rotazione attorno alla terra.

Si procede dunque riportando il vettore velocità angolare dal riferimento orbitale al riferimento corpo del satellite. Il vettore di spin così viene impiegato come assetto desiderato nella subroutine di spin.

6.6. GROUND TRACKING

Un satellite in orbita attorno alla terra può svolgere numerose missioni, e la maggior parte di esse sono solitamente riconducibili alla terra stessa.

Il ground tracking è forse la più diffusa, e sicuramente una delle più complesse a causa della geometria del problema considerato. Permettere a un satellite di direzionare le antenne verso la stazione di terra o puntare il payload costantemente verso un determinato luogo sono le funzioni maggiormente ricorrenti nella filosofia di progetto del satellite PiCPoT, e si ritrovano parzialmente nel progetto AraMiS.

6.6.1. Geometria del problema pseudo-bidimensionale

Il primo problema da affrontare riguarda la geometria del problema. Ricavare gli angoli di assetto in funzione del tempo e della posizione sull'orbita non è stato un lavoro semplice, complice la scarsità di informazioni sull'argomento reperibile in letteratura.

Iniziamo considerando il caso pseudo-bidimensionale in cui il punto da tracciare, che chiameremo stazione di terra, sia esattamente sulla proiezione del satellite sulla superficie terrestre (detta ground track).

A questo punto possiamo definire con μ , detto angolo di puntamento, ossia l'angolo compreso tra la tangente all'orbita nella direzione del moto e la stazione di terra sulla ground track. Definiamo inoltre le grandezze:

R , raggio terrestre;

h , quota del satellite;

$R_T = h + R$, il raggio totale dal centro della terra al satellite.

Note tali grandezze si ricava γ , inteso come l'angolo al centro formato dalla verticale locale alla stazione di terra e la verticale del satellite (linea del nadir).

$$\gamma = \omega_{or} \cdot t - \gamma_0$$

$$\gamma_0 = \sin^{-1}(R/R_t)$$

Dove γ è stato definito a partire dal punto in cui il satellite interseca il piano tangente alla superficie terrestre nel suo moto ascendente. L'angolo γ corrispondente a questa condizione è per l'appunto γ_0 , definito come sopra. ω_{or} rappresenta invece la velocità di rotazione del satellite attorno al centro della terra, che per il nostro satellite vale

$$\omega_{or} = 1.16 \cdot 10^{-3} \text{ rad/s}$$

Si nota inoltre come γ valga 0 quando il satellite si trova sopra la stazione di terra.

Occorre però introdurre una correzione al valore di γ , in quanto il minimo valore dell'angolo che deve avere il satellite rispetto all'orizzonte della stazione di terra per poter essere visto e comunicare deve essere di 5° almeno. Introduciamo quindi il parametro correttivo $\varepsilon = 0.087 \text{ rad}$, ottenendo:

$$\gamma = \omega_{or} \cdot t - (\gamma_0 - \varepsilon)$$

Per il tipo di orbita considerata possiamo calcolare il parametro:

$$(\gamma_0 - \varepsilon) \cong 60^\circ = 1.047 \text{ rad}$$

E' possibile determinare per quanto tempo, al massimo, il satellite è in grado di mantenere il puntamento con la stazione di terra prima di "tramontare",

$$t_{\text{puntamento}} = \frac{2 \cdot (\gamma_0 - \varepsilon)}{\omega_{or}}$$

Che nel caso considerato è di circa 29 minuti.

Nei calcoli che seguono vengono inoltre indicate le distanze $R_{t1}(t)$ e $R_{t2}(t)$ che rappresentano rispettivamente la porzione di R_t interna ed esterna all'intersezione con il piano tangente alla stazione di terra.

$$R_{t1}(t) = R \cdot \cos(\gamma(t))$$

$$R_{t2}(t) = R_t - R_{t1}(t)$$

Mentre con d indichiamo la distanza tra la stazione di terra e l'intersezione del piano tangente con il raggio,

$$d(t) = R \cdot \sin(\gamma(t))$$

L'angolo μ risulta quindi calcolabile come,

$$\mu(t) = \frac{\pi}{2} - \tan^{-1} \frac{d(t)}{R_{T1}(t)}$$

Con

$$\mu(0) = \frac{\pi}{2} - \gamma_0 + \varepsilon$$

Poiché la rotazione che si rende necessaria è funzione del tempo, anche le velocità angolari che occorre imprimere alle ruote avranno andamento non lineare e le manovre di controllo che ne risultano sono piuttosto complesse.

Possiamo quindi pensare di calcolare la derivata con la definizione di rapporto incrementale per intervalli, ad esempio, di 1 secondo.

$$\dot{\mu} \cong \mu_d(t) = \frac{\mu(t+1) - \mu(t)}{(t+1) - (t)}$$

In modo da ottenere un controllo a tempo discreto sulla velocità angolare.

Secondo questo metodo infatti viene eseguito un assetto di spin della durata di 1 secondo, con vettore velocità aggiornato alla fine di ogni ciclo.

Un'ulteriore problema implicito in questa geometria consiste nel ruotare il satellite, come nel caso della manovra inerziale, prima che la rotazione di puntamento costata abbia inizio. La conseguenza principale porta alla necessità di individuare, data l'orbita percorsa, il punto in cui tale manovra ha inizio. Trovato tale punto, con una semplice manovra di rotazione rispetto al nuovo assetto (calcolato al punto iniziale come assetto zero) come quelle che abbiamo visto precedentemente.

Per la risoluzione di questo problema ci viene in aiuto la definizione di ground track e le equazioni che la regolano.

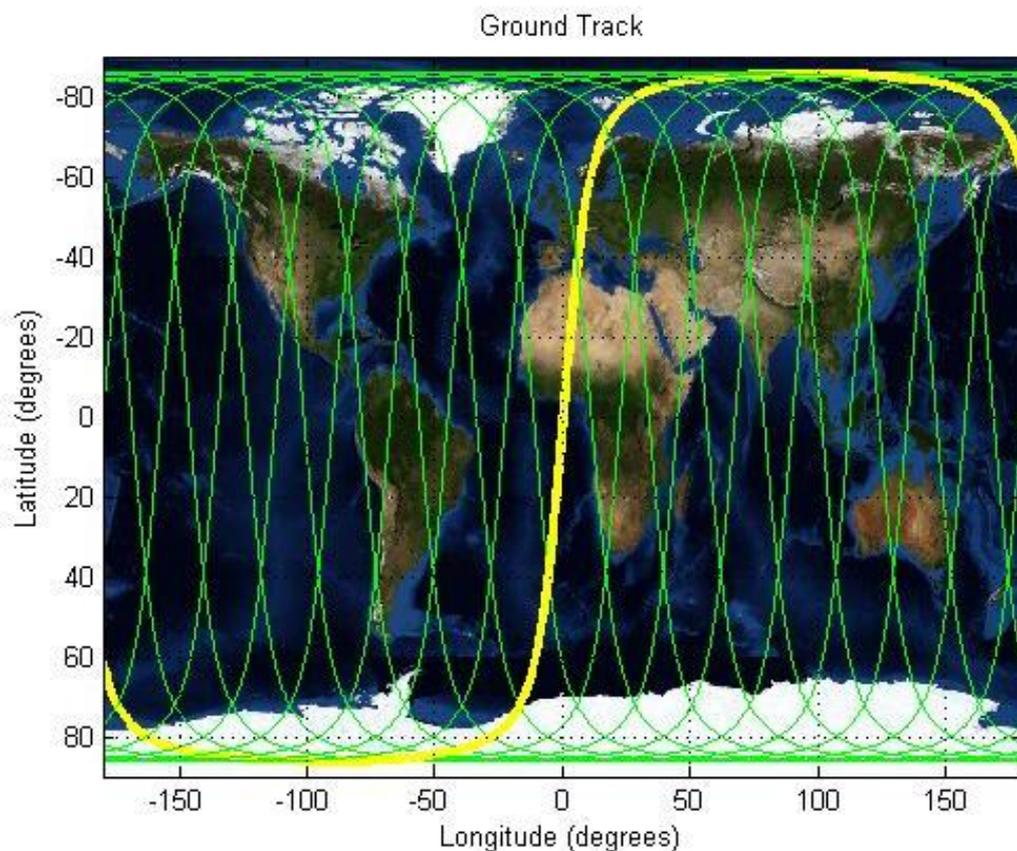
Dati *in* l'inclinazione dell'orbita e l_n e λ_n rispettivamente longitudine e latitudine del satellite della verticale nel sistema di riferimento terra, e rispetto al nodo ascendente.

$$\begin{cases} \sin(\lambda_n) = \sin(in) \cdot \sin(\omega_{or}t) \\ \tan(l_n + \omega_{er}t) = \cos(in) \cdot \tan(\omega_{or}t) \end{cases}$$

dove $\omega_{er} = 0.004178$ gradi/s è la velocità di rotazione della Terra attorno il suo asse e ω_{or} la velocità di rivoluzione del satellite.

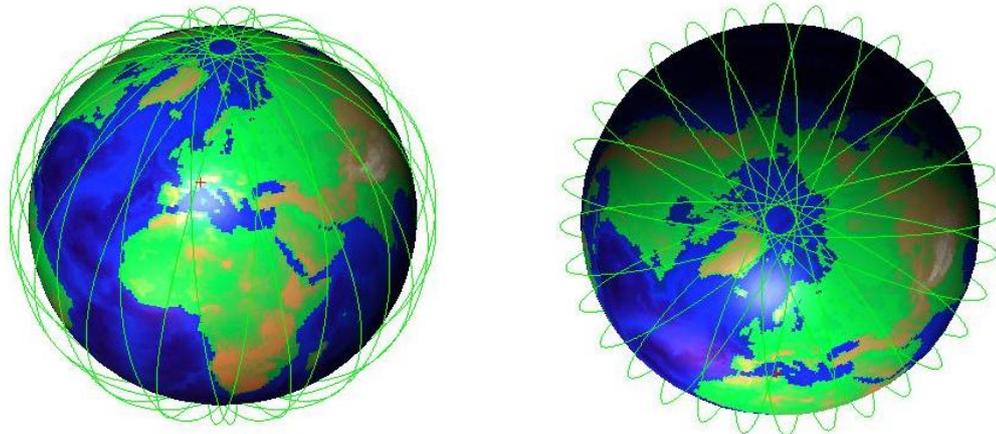
In figura 6.5 sono mostrate le orbite percorse da un satellite con orbita circolare e inclinazione $86^\circ 4'$

In generale le traiettorie dei satelliti sono molto complicate e per ottenerle si usano programmi di simulazione STK (Satellite Tool Kit, www.stk.com). Questi programmi includono anche i casi di orbite ellittiche.



6.5 Le 16 orbite che il satellite percorre attorno alla terra.

Nel nostro caso la gestione delle orbite è stata affidata a una modellizzazione matematica da me realizzata e che è inclusa nei software matlab in appendice, i quali generano le immagini qui riportate, sia bidimensionali che tridimensionali delle orbite sulla terra.



6.6 Immagini dell'involuppo orbitale del satellite 3D

Il satellite nel suo moto di rotazione attorno alla terra percorre 16 orbite nell'arco delle 24 ore, e queste risultano sfasate tra loro di un fattore $\pi/8$, tali orbite vengono distinte per i calcoli che seguono in base al coefficiente n_{orbit} , che identifica appunto a quale curva sulla ground track di figura 6.5 ci si sta riferendo.

6.6.2. Geometria del problema tridimensionale

La geometria del problema diventa più complessa se consideriamo una ground station che non sia sulla ground track del satellite, come molto spesso avviene nella realtà.

In questo caso il satellite non deve solo adattare il suo angolo di puntamento μ ma deve ruotare di un secondo angolo che chiameremo di deviazione δ .

Per poter ricavare analiticamente l'espressione dell'angolo δ occorre introdurre la grandezza D_{gt} , che rappresenta la minima distanza tra la ground station e la proiezione di questa sulla ground track.

Poiché i punti sulla terra sono espressi in coordinate sferiche e le distanze reciproche variano in base alla posizione sul globo, si preferisce passare a un sistema di coordinate cartesiane centrate come il sistema di coordinate sferiche. Il passaggio da uno all'altro è quello classico qui riportato:

$$\begin{cases} X_n = R_t \cos \lambda_n \sin l_n \\ Y_n = R_t \cos \lambda_n \cos l_n \\ Z_n = R_t \sin \lambda_n \end{cases}$$

A questo punto si procede prima alla vettorizzazione delle 16 orbite, distinte in base al coefficiente n_{orbit} . Si hanno quindi a disposizione tutte le coppie di latitudine e longitudine in cui si trova a essere il satellite lungo la sua orbita.

Date poi latitudine e longitudine della stazione di terra, si procede alla loro trasformazione in coordinate cartesiane

$$\begin{cases} X_{GS} = R \cos \lambda_{GS} \sin l_{GS} \\ Y_{GS} = R \cos \lambda_{GS} \cos l_{GS} \\ Z_{GS} = R \sin \lambda_{GS} \end{cases}$$

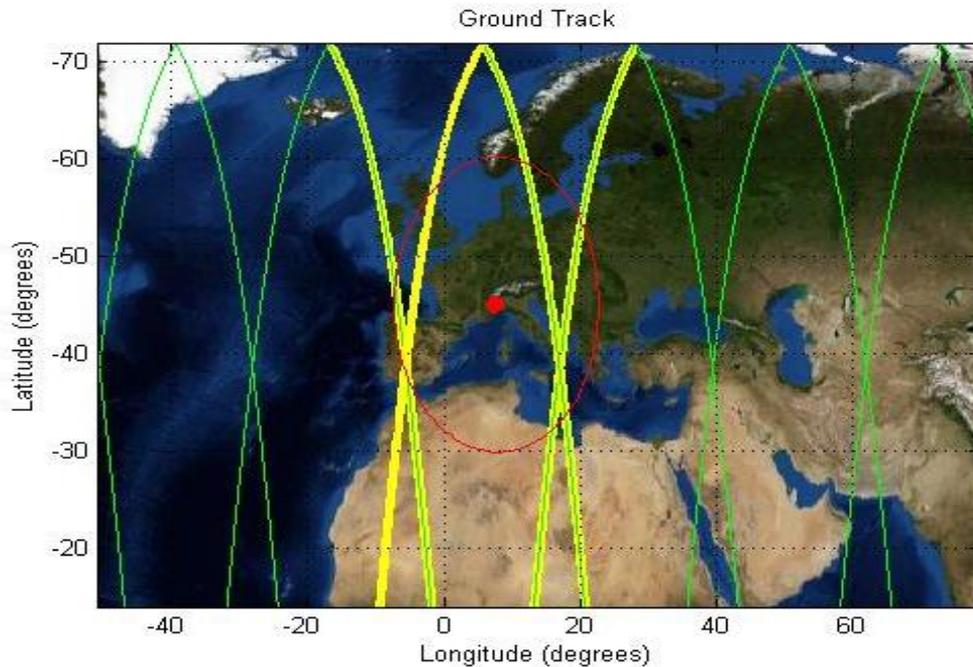
Si procede calcolando la distanza geometrica tra la posizione della ground station e quella del satellite. Il minimo di tali valori, è per l'appunto il parametro D_{gt} , variabile in funzione dell'orbita considerata.

$$D_{gt}(n_{orbit}) = \min(\sqrt{(X_n - X_{GS})^2 + (Y_n - Y_{GS})^2 + (Z_n - Z_{GS})^2})$$

Definendo ρ come un parametro comparativo, inteso come un indice della massima distanza angolare entro cui il satellite riesce a vedere la stazione di terra per un tempo accettabile. Tale valore può variare da un minimo di 0, che richiama il caso si ground station giacente sulla ground track, fino a un massimo valore pari a μ_0 , che corrisponde al satellite con elevazione ε rispetto al piano tangente.

Il tempo di puntamento è massimo per $\rho = 0$ e minimo per $\rho = \mu_0$, a patto di avere un orbita compresa in tale raggio d'azione.

Scelto ρ , vediamo che un'orbita è utile per il puntamento se
 $\rho - D_{gt} > 0$



6.7 Esempio di orbite coinvolte nel caso di GS situata in Torino Politecnico (coordinate N 45.062836, E 7.661419)

Un'esempio è mostrato in figura 6.7, per una ground station posta nel cortile del Politecnico di Torino, ipotizzando che vi sia una stazione per trasmissione dati al satellite. Si nota come le orbite coinvolte siano 4, due nella fase di salita del satellite e due nella fase di discesa. Il puntino rosso indica la GS, mentre in rosso è segnata la circonferenza di raggio $\rho = 15^\circ$ che determina i punti di vista del satellite.

Una volta ricavate tutte le orbite utili per effettuare un ground tracking, per ognuna di esse iterativamente si calcola l'angolo δ . Quest'angolo rappresenta la deviazione dell'asse x del satellite (più in generale l'asse che dovrebbe essere puntato verso la stazione di terra) rispetto alla congiungente alla proiezione della GS sulla ground track.

La sua espressione analitica in funzione di n_{orbit} è:

$$\delta(n_{orbit}) = \tan^{-1} \left(\frac{D_{gt} \cdot \sin(\mu(in))}{R_{t2}} \right)$$

Anche in questo caso poiché l'angolo varia in base al tempo, e quindi alla posizione, occorre discretizzare il calcolo della derivata come nel caso precedente. Per intervalli di tempo di 1 secondo calcoliamo quindi il rapporto incrementale,

$$\dot{\delta} \cong \delta_a(t) = \frac{\delta(t+1) - \delta(t)}{(t+1) - (t)}$$

Il valore ottenuto è l'intensità del vettore velocità angolare che occorre fornire al satellite per mantenere il puntamento sull'obiettivo.

Data la definizione dell'angolo δ si tratta di una rotazione attorno all'asse ortogonale sia all'asse di rotazione di μ che all'asse $x_{pointing}$ del satellite.

La combinazione dei due vettori velocità angolare $\omega_{ys} = \dot{\mu}$ e $\omega_{zs} = \dot{\delta}$ fornisce il vettore velocità ω_{gt} angolare necessario per effettuare il ground tracking. Si impiega tale vettore per impostare uno spin, che verrà aggiornato a ogni nuovo calcolo delle velocità per approssimare il più possibile la curva reale di velocità necessaria.

7. SIMULAZIONE DEI PROGRAMMI

In questa sezione effettueremo la simulazione numerica delle manovre precedentemente descritte, considerando i casi più significativi. Per simulazione di assetti più specifici i programmi così come sono stati impiegati sono riportati nell'appendice A, e nelle successive appendici (B e C) sono presenti file necessari al corretto funzionamento del software.

7.1. SIMULAZIONE DI SPIN

7.1.1. Imposizione spin da condizioni iniziali nulle

Consideriamo innanzitutto un satellite in condizioni iniziali nulle, ovvero con velocità angolare del satellite $\bar{\omega}_{iniziale} = [0]$, a cui verranno imposti i seguenti vettori di spin:

1. $\omega_{finale} = [1,0,0]$
2. $\omega_{finale} = [0,1,0]$
3. $\omega_{finale} = [0,0,1]$
4. $\omega_{finale} = [1,1,1]$

Si sono scelti tali valori sia perché sono prossimi al valore limite scelto a progetto (1 RPM) e perché permettono di valutare la risposta del satellite.

Ovviamente i risultati di queste simulazioni possono apparire forse banali, tuttavia come analizzato precedentemente sono basilari ai fini delle manovre successive.

CASO 1

Satellite PiCPoT					
	$\bar{\omega}_{iniziale}$ [RPM]	$\bar{\omega}_{finale}$ [RPM]	J_{ruote} [kg · m ²]	J_{sat} [kg · m ²]	$\bar{\omega}_{ruote}$ [RPM]
Asse X	0	±1	4 E-6	0.022	∓5500
Asse Y	0	0	4 E-6	0.022	0
Asse Z	0	0	4 E-6	0.022	0

Satellite AraMiS (Configurazione Cubica)					
	$\bar{\omega}_{iniziale}$ [RPM]	$\bar{\omega}_{finale}$ [RPM]	J_{ruote} [kg · m ²]	J_{sat} [kg · m ²]	$\bar{\omega}_{ruote}$ [RPM]
Asse X	0	±1	8.18 E-6	0.045	∓5500
Asse Y	0	0	8.18 E-6	0.045	0
Asse Z	0	0	8.18 E-6	0.045	0

CASO 2

Satellite PiCPoT					
	$\bar{\omega}_{iniziale}$ [RPM]	$\bar{\omega}_{finale}$ [RPM]	J_{ruote} [kg · m ²]	J_{sat} [kg · m ²]	$\bar{\omega}_{ruote}$ [RPM]
Asse X	0	0	4 E-6	0.022	0
Asse Y	0	±1	4 E-6	0.022	∓5500
Asse Z	0	0	4 E-6	0.022	0

Satellite AraMiS (Configurazione Cubica)					
	$\bar{\omega}_{iniziale}$ [RPM]	$\bar{\omega}_{finale}$ [RPM]	J_{ruote} [kg · m ²]	J_{sat} [kg · m ²]	$\bar{\omega}_{ruote}$ [RPM]
Asse X	0	0	8.18 E-6	0.045	0
Asse Y	0	±1	8.18 E-6	0.045	∓5500
Asse Z	0	0	8.18 E-6	0.045	0

CASO 3

Satellite PiCPoT					
	$\bar{\omega}_{iniziale}$ [RPM]	$\bar{\omega}_{finale}$ [RPM]	J_{ruote} [kg · m ²]	J_{sat} [kg · m ²]	$\bar{\omega}_{ruote}$ [RPM]
Asse X	0	0	4 E-6	0.022	0
Asse Y	0	0	4 E-6	0.022	0
Asse Z	0	±1	4 E-6	0.022	±5500

Satellite AraMiS (Configurazione Cubica)					
	$\bar{\omega}_{iniziale}$ [RPM]	$\bar{\omega}_{finale}$ [RPM]	J_{ruote} [kg · m ²]	J_{sat} [kg · m ²]	$\bar{\omega}_{ruote}$ [RPM]
Asse X	0	0	8.18 E-6	0.045	0
Asse Y	0	0	8.18 E-6	0.045	0
Asse Z	0	±1	8.18 E-6	0.045	±5500

CASO 4

Satellite PiCPoT					
	$\bar{\omega}_{iniziale}$ [RPM]	$\bar{\omega}_{finale}$ [RPM]	J_{ruote} [kg · m ²]	J_{sat} [kg · m ²]	$\bar{\omega}_{ruote}$ [RPM]
Asse X	0	±1	4 E-6	0.022	±5500
Asse Y	0	±1	4 E-6	0.022	±5500
Asse Z	0	±1	4 E-6	0.022	±5500

Satellite AraMiS (Configurazione Cubica)					
	$\bar{\omega}_{iniziale}$ [RPM]	$\bar{\omega}_{finale}$ [RPM]	J_{ruote} [kg · m ²]	J_{sat} [kg · m ²]	$\bar{\omega}_{ruote}$ [RPM]
Asse X	0	±1	8.18 E-6	0.045	±5500
Asse Y	0	±1	8.18 E-6	0.045	±5500
Asse Z	0	±1	8.18 E-6	0.045	±5500

7.1.2. Arresto di un satellite a partire da spin non nullo

Poiché tale simulazione è molto semplice ed è del tutto analoga alla precedente, si mostreranno solo casi in cui tutte e tre le componenti di partenza siano non nulle. Il vettore velocità angolare finale sarà invece $\omega_{finale} = [0,0,0]$

I casi che considereremo sono i seguenti

1. $\omega_{iniziale} = [1,1,1]$
2. $\omega_{iniziale} = [0.5,0.7,0.9]$
3. $\omega_{iniziale} = [1.4,1.6,1.8]$

Vedremo inoltre come nel caso 3 le velocità impostate siano volutamente superiori a quelle di progetto. Vedremo quindi i valori che le ruote dovrebbero assumere per ottenere lo spin voluto.

Quando una ruota satura occorre ricorrere al processo di desaturazione tramite attuatori magnetici.

CASO 1

Satellite PiCPoT					
	$\bar{\omega}_{iniziale}$ [RPM]	$\bar{\omega}_{finale}$ [RPM]	J_{ruote} [kg · m ²]	J_{sat} [kg · m ²]	$\bar{\omega}_{ruote}$ [RPM]
Asse X	0	±1	4 E-6	0.022	∓5500
Asse Y	0	±1	4 E-6	0.022	∓5500
Asse Z	0	±1	4 E-6	0.022	∓5500

Satellite AraMiS (Configurazione Cubica)					
	$\bar{\omega}_{iniziale}$ [RPM]	$\bar{\omega}_{finale}$ [RPM]	J_{ruote} [kg · m ²]	J_{sat} [kg · m ²]	$\bar{\omega}_{ruote}$ [RPM]
Asse X	0	±1	8.18 E-6	0.045	∓5500
Asse Y	0	±1	8.18 E-6	0.045	∓5500
Asse Z	0	±1	8.18 E-6	0.045	∓5500

CASO 2

Satellite PiCPoT					
	$\bar{\omega}_{iniziale}$ [RPM]	$\bar{\omega}_{finale}$ [RPM]	J_{ruote} [kg · m ²]	J_{sat} [kg · m ²]	$\bar{\omega}_{ruote}$ [RPM]
Asse X	0	±0.5	4 E-6	0.022	±2750
Asse Y	0	±0.7	4 E-6	0.022	±3850
Asse Z	0	±0.9	4 E-6	0.022	±4950

Satellite AraMiS (Configurazione Cubica)					
	$\bar{\omega}_{iniziale}$ [RPM]	$\bar{\omega}_{finale}$ [RPM]	J_{ruote} [kg · m ²]	J_{sat} [kg · m ²]	$\bar{\omega}_{ruote}$ [RPM]
Asse X	0	±0.5	8.18 E-6	0.045	±2750
Asse Y	0	±0.7	8.18 E-6	0.045	±3850
Asse Z	0	±0.9	8.18 E-6	0.045	±4950

CASO 3

Satellite PiCPoT					
	$\bar{\omega}_{iniziale}$ [RPM]	$\bar{\omega}_{finale}$ [RPM]	J_{ruote} [kg · m ²]	J_{sat} [kg · m ²]	$\bar{\omega}_{ruote}$ [RPM]
Asse X	0	±1.4	4 E-6	0.022	±7700
Asse Y	0	±1.6	4 E-6	0.022	±8800
Asse Z	0	±1.8	4 E-6	0.022	±9900

Satellite AraMiS (Configurazione Cubica)					
	$\bar{\omega}_{iniziale}$ [RPM]	$\bar{\omega}_{finale}$ [RPM]	J_{ruote} [kg · m ²]	J_{sat} [kg · m ²]	$\bar{\omega}_{ruote}$ [RPM]
Asse X	0	±1.4	8.18 E-6	0.045	±7700
Asse Y	0	±1.6	8.18 E-6	0.045	±8800
Asse Z	0	±1.8	8.18 E-6	0.045	±9900

7.2. SIMULAZIONE DI ROTAZIONE

Come analizzato nel capitolo 6.4 la rotazione è una manovra di spin ben calibrata che, se protratta per un certo intervallo di tempo, porta il satellite a compiere la rotazione desiderata.

Si è osservato che le manovre di spin sia per il satellite PiCPoT che per il satellite AraMiS in configurazione cubica minima avvengono col medesimo pilotaggio delle ruote di reazione. Ciò avviene grazie al fatto che le ruote di reazione di AraMiS sono state supposte come maggiorate rispetto a quelle di PiCPoT mantenendo costante il rapporto tra le inerzie del satellite e della ruota, portando così a coincidere le velocità caratteristiche di pilotaggio delle due manovre.

Per brevità di esposizione pertanto considereremo ogni singolo caso come se avessimo solo uno dei due satelliti senza distinguere i due casi.

Andremo quindi a studiare alcuni casi in cui, da un determinato assetto il nostro satellite dovrà riportarsi in una condizione in cui i suoi assi siano paralleli agli assi orbita. Successivamente il caso in cui, dall'assetto dell'orbita ci si vuole portare ad un generico assetto e, per ultimo, da un assetto generico ad un altro generico.

Definiamo con \bar{E}_{sns} il vettore degli angoli di Eulero di partenza del satellite, e con \bar{E}_{at} il vettore di angoli desiderato. Entrambi espressi con riferimento agli assi corpo.

Il vettore sotto riportato indica la condizione di allineamento agli assi orbita.

$$\bar{E}_0 = [0 \ 0 \ 0]^T$$

Consideriamo quindi il caso in cui, partendo da allineati agli assi orbita si voglia raggiungere un certo assetto. Successivamente il caso opposto, in cui da alcuni generici assetti ci si vuole ricondurre all'assetto iniziale. Infine il raggiungimento di un determinato assetto, non allineato all'orbita, partendo da una condizione casuale e non allineata.

7.2.1. Rotazione con $\bar{E}_{sns} = \bar{E}_0$

Per le simulazioni che seguono consideriamo i seguenti vettori di angoli di Eulero da raggiungere:

1. $\bar{E}_{at} = [45 \ 00 \ 00]^T$
2. $\bar{E}_{at} = [00 \ 30 \ 00]^T$
3. $\bar{E}_{at} = [00 \ 00 \ 130]^T$
4. $\bar{E}_{at} = [90 \ 90 \ 90]^T$
5. $\bar{E}_{at} = [20 \ 16 \ 45]^T$

Si ricorda che nei successivi casi i valori presi con segno positivo rappresentano rotazioni antiorarie, mentre valori negativi considerano rotazioni orarie

CASO 1

T_{man}	8.0 s
-----------	-------

	$\bar{E}_{sns} [^\circ]$	$\bar{E}_{at} [^\circ]$	$\bar{E}_{sns \rightarrow at} [^\circ]$
Ψ	0	45	45
Θ	0	0	0
Φ	0	0	0

assi	$\omega_{sat} [RPM]$	$\omega_r [RPM]$
X	0	0
Y	0	0
Z	-0.938	+5156

CASO 2

T_{man}	6.0 s
-----------	-------

	$\bar{E}_{sns} [^\circ]$	$\bar{E}_{at} [^\circ]$	$\bar{E}_{sns \rightarrow at} [^\circ]$
Ψ	0	0	0
Θ	0	30	30
Φ	0	0	0

assi	$\omega_{sat} [RPM]$	$\omega_r [RPM]$
X	0	0
Y	-0.833	+4583
Z	0	0

CASO 3

T_{man}	22 s
-----------	------

	\bar{E}_{sns} [°]	\bar{E}_{at} [°]	$\bar{E}_{sns \rightarrow at}$ [°]
Ψ	0	0	0
Θ	0	0	0
Φ	0	130	130

assi	ω_{sat} [RPM]	ω_r [RPM]
X	0	0
Y	0	0
Z	-0.985	+5417

CASO 4

T_{man}	16.0 s
-----------	--------

	\bar{E}_{sns} [°]	\bar{E}_{at} [°]	$\bar{E}_{sns \rightarrow at}$ [°]
Ψ	0	90	0
Θ	0	90	90
Φ	0	90	0

assi	ω_{sat} [RPM]	ω_r [RPM]
X	-0.937	+5156
Y	0	0
Z	0	0

CASO 5

T_{man}	11 s
-----------	------

	\bar{E}_{sns} [°]	\bar{E}_{at} [°]	$\bar{E}_{sns \rightarrow at}$ [°]
Ψ	0	20	20
Θ	0	16	16
Φ	0	45	45

assi	ω_{sat} [RPM]	ω_r [RPM]
X	-0.230	+1268
Y	+0.112	-618
Z	-0.937	+5352

Questi sono solo alcuni casi per mostrare il comportamento generale delle ruote d'inerzia sotto l'imposizione di determinati angoli.

7.2.2. Rotazione con $\bar{E}_{at} = \bar{E}_0$

Per le simulazioni che seguono consideriamo i seguenti vettori di angoli di Eulero come condizione iniziale. Si noti che sono volutamente gli stessi vettori del caso precedente in quanto si vogliono evidenziare le differenze di pilotaggio pur mantenendo gli stessi vettori.

1. $\bar{E}_{sns} = [45 \ 00 \ 00]^T$
2. $\bar{E}_{sns} = [00 \ 30 \ 00]^T$
3. $\bar{E}_{sns} = [00 \ 00 \ 130]^T$
4. $\bar{E}_{sns} = [90 \ 90 \ 90]^T$
5. $\bar{E}_{sns} = [20 \ 16 \ 45]^T$

Analogamente a prima, segni positivi rappresentano rotazioni antiorarie e segni negativi rotazioni orarie.

CASO 1

T_{man}	8.0 s
-----------	-------

	$\bar{E}_{sns} [^\circ]$	$\bar{E}_{at} [^\circ]$	$\bar{E}_{sns \rightarrow at} [^\circ]$
Ψ	45	0	-45
Θ	0	0	0
Φ	0	0	0

assi	$\omega_{sat} [RPM]$	$\omega_r [RPM]$
X	0	0
Y	0	0
Z	+0.938	-5156

CASO 2

T_{man}	6 s
-----------	-----

	$\bar{E}_{sns} [^\circ]$	$\bar{E}_{at} [^\circ]$	$\bar{E}_{sns \rightarrow at} [^\circ]$
Ψ	0	0	0
Θ	30	0	-30
Φ	0	0	0

assi	$\omega_{sat} [RPM]$	$\omega_r [RPM]$
X	+0.833	-4583
Y	0	0
Z	0	0

CASO 3

T_{man}	22.0 s
-----------	--------

	\bar{E}_{sns} [°]	\bar{E}_{at} [°]	$\bar{E}_{sns \rightarrow at}$ [°]
Ψ	0	0	0
Θ	0	0	0
Φ	130	0	-130

assi	ω_{sat} [RPM]	ω_r [RPM]
X	0	0
Y	0	0
Z	+0.985	-5417

CASO 4

T_{man}	16 s
-----------	------

	\bar{E}_{sns} [°]	\bar{E}_{at} [°]	$\bar{E}_{sns \rightarrow at}$ [°]
Ψ	90	0	0
Θ	90	0	-90
Φ	90	0	0

assi	ω_{sat} [RPM]	ω_r [RPM]
X	+0.937	-5156
Y	0	0
Z	0	0

CASO 5

T_{man}	8 s
-----------	-----

	\bar{E}_{sns} [°]	\bar{E}_{at} [°]	$\bar{E}_{sns \rightarrow at}$ [°]
Ψ	20	0	-3.718
Θ	16	0	-25.150
Φ	45	0	-41.331

assi	ω_{sat} [RPM]	ω_r [RPM]
X	+0.415	-2284
Y	+0.321	-1767
Z	+0.931	-5121

Questi sono solo alcuni casi per mostrare il comportamento generale delle ruote d'inerzia sotto l'imposizione di determinati angoli.

Si noti come, per rotazioni semplici, (casi 1-2-3-4) lo scambio tra posizione iniziale e finale comporta semplicemente l'inversione della rotazione e lo scambio dei segni. Per rotazioni più complesse come è il

caso 5 invece, lo scambio tra condizione iniziale e finale altera in modo sostanziale i valori degli angoli necessari per riportarsi alla condizione iniziale. Questo comportamento sarà ancora più evidente nel paragrafo successivo.

7.2.3. Rotazione generica

Per le seguenti simulazioni considereremo le seguenti coppie di vettori di angoli di Eulero. Per ogni coppia verrà effettuata una simulazione diretta, e una inversa, scambiando il ruolo dei due vettori. In questo modo si vuole simulare la rotazione opposta per tornare al punto di partenza.

1. $\bar{E}_{sns} = [00, 90, 00]^T$ $\bar{E}_{at} = [90, 00, 90]^T$
2. $\bar{E}_{sns} = [60, -45, 30]^T$ $\bar{E}_{at} = [-15, 25, 10]^T$
3. $\bar{E}_{sns} = [-25, 35, 80]^T$ $\bar{E}_{at} = [145, 00, 05]^T$
4. $\bar{E}_{sns} = [20, -20, 00]^T$ $\bar{E}_{at} = [00, 00, -135]^T$

CASO 1

Rotazione diretta

T_{man}	16 s
-----------	------

	\bar{E}_{sns} [°]	\bar{E}_{at} [°]	$\bar{E}_{sns \rightarrow at}$ [°]
Ψ	0	90	0
Θ	90	0	0
Φ	0	90	90

assi	ω_{sat} [RPM]	ω_r [RPM]
X	0	0
Y	0	0
Z	-0.937	+5156

Rotazione inversa

T_{man}	16.0 s
-----------	--------

	\bar{E}_{sns} [°]	\bar{E}_{at} [°]	$\bar{E}_{sns \rightarrow at}$ [°]
Ψ	90	0	0
Θ	0	90	0
Φ	90	0	-90

assi	ω_{sat} [RPM]	ω_r [RPM]
X	0	0
Y	0	0
Z	+0.937	-5156

CASO 2

Rotazione diretta

T_{man}	6 s
-----------	-----

	\bar{E}_{sns} [°]	\bar{E}_{at} [°]	$\bar{E}_{sns \rightarrow at}$ [°]
Ψ	60	-15	-86.917
Θ	-45	25	+10.491
Φ	30	10	-72.029

assi	ω_{sat} [RPM]	ω_r [RPM]
X	-0.113	+621
Y	-0.031	+173
Z	+0.972	-5347

Rotazione inversa

T_{man}	13 s
-----------	------

	\bar{E}_{sns} [°]	\bar{E}_{at} [°]	$\bar{E}_{sns \rightarrow at}$ [°]
Ψ	-15	60	+79.964
Θ	25	-45	-72.337
Φ	10	30	-0.933

assi	ω_{sat} [RPM]	ω_r [RPM]
X	+0.911	-5013
Y	+0.992	-5455
Z	-0.299	+1645

CASO 3

Rotazione diretta

T_{man}	42.0 s
-----------	--------

	\bar{E}_{sns} [°]	\bar{E}_{at} [°]	$\bar{E}_{sns \rightarrow at}$ [°]
Ψ	-25	145	-175.522
Θ	35	00	+15.883
Φ	80	05	-77.266

assi	ω_{sat} [RPM]	ω_r [RPM]
X	-0.200	+1099
Y	-0.019	+107
Z	+0.977	-5371

Rotazione inversa

T_{man}	15.0 s
-----------	--------

	\bar{E}_{sns} [°]	\bar{E}_{at} [°]	$\bar{E}_{sns \rightarrow at}$ [°]
Ψ	145	-25	+163.538
Θ	00	35	-00.917
Φ	05	80	-77.758

assi	ω_{sat} [RPM]	ω_r [RPM]
X	-0.026	+144
Y	+0.016	-89
Z	-0.953	+5241

CASO 4

Rotazione diretta

T_{man}	104 s
-----------	-------

	\bar{E}_{sns} [°]	\bar{E}_{at} [°]	$\bar{E}_{sns \rightarrow at}$ [°]
Ψ	20	0	+0.946
Θ	-20	0	-27.976
Φ	0	-135	-138.797

assi	ω_{sat} [RPM]	ω_r [RPM]
X	-0.148	+7662
Y	+0.126	-1135
Z	+0.958	-5098

Rotazione inversa

T_{man}	26.0 s
-----------	--------

	\bar{E}_{sns} [°]	\bar{E}_{at} [°]	$\bar{E}_{sns \rightarrow at}$ [°]
Ψ	0	20	+20.0
Θ	0	-20	-20.0
Φ	-135	0	+135.0

assi	ω_{sat} [RPM]	ω_r [RPM]
X	-0.060	+815
Y	-0.122	-691
Z	+0.986	-5269

Analogamente a prima, segni positivi rappresentano rotazioni antiorarie e segni negativi rotazioni orarie.

Si noti come, ad eccezione del caso particolare 1, in tutti gli altri casi la rotazione diretta e inversa non coincidono. Sia gli angoli di Eulero che le rotazioni da imporre sono differenti, anche se i tempi caratteristici di manovra sono piuttosto simili.

Questa peculiarità deriva direttamente dalla non commutatività del prodotto delle matrici di rotazione che conducono a questo risultato.

7.3. SIMULAZIONE DI PUNTAMENTO INERZIALE

Come nelle simulazioni precedenti, nel caso del puntamento inerziale si eseguono due distinte manovre. Una di allineamento, costituita da una rotazione, che porta gli assi nella posizione desiderata, come dettagliatamente spiegato nel cap. 6.5. Una seconda manovra, questa volta di spin, è richiesta per il mantenimento costante del puntamento, con un vettore velocità angolare che in assi orbita è uguale a quello che descrive la rotazione del satellite, e che occorre convertire in assi body per poter essere correttamente applicato.

Le simulazioni che verranno effettuate partiranno tutte da assetto nullo, ovvero sia con assi corpo allineati con gli assi orbita. Questo per poter apprezzare al meglio l'azione di puntamento compiuta, tralasciando parte della manovra iniziale di rotazione.

Per lo stesso motivo consideriamo anche nullo il tempo dall'ultimo passaggio per il nodo ascendente. Nell'ultima simulazione tuttavia esamineremo singolarmente entrambi gli effetti, rispetto ad un puntamento con condizioni iniziali nulle.

Il nuovo assetto può essere espresso sia come una terna di assi espressa nel riferimento ECI che come un vettore terna di angoli di Eulero. Le due espressioni sono infatti equivalenti tra di loro, e il passaggio da assi ad angoli avviene mediante la *function* presente in appendice B.6 "EuAt2Ang".

Simuliamo dunque di voler determinare i seguenti puntamenti:

1. $\bar{E}_{at} = [00, 00, 00]^T$
2. $\bar{E}_{at} = [30, 15, 00]^T$
3. $\bar{E}_{at} = [00, 20, -40]^T$
4. $\bar{E}_{at} = [135, -10, -15]^T$
5. $Ax_{at} = \begin{bmatrix} 0 & 0 & -1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix}$

CASO 1

T_{man}	32s
T_{idle}	48 s

Rotazione preliminare

	\bar{E}_{sns} [°]	\bar{E}_{at} [°]	$\bar{E}_{sns \rightarrow at}$ [°]
Ψ	0	0	-70.580
Θ	0	0	+11.7162
Φ	0	0	-119.942

Rotazione di puntamento

assi	ω_{sat} [RPM]	ω_r [RPM]
X	-0.034	+188
Y	-0.090	+496
Z	+0.985	-5416

assi	ω_{sat} [RPM]	ω_r [RPM]
X	$-1.92 \cdot 10^{-3}$	+11
Y	$-10.89 \cdot 10^{-3}$	+60
Z	$-1.13 \cdot 10^{-3}$	+6

CASO 2

T_{man}	31 s
T_{idle}	49s

Rotazione preliminare

	\bar{E}_{sns} [°]	\bar{E}_{at} [°]	$\bar{E}_{sns \rightarrow at}$ [°]
Ψ	0	30	-88.759
Θ	0	15	+36.877
Φ	0	00	-112.659

Rotazione di puntamento

assi	ω_{sat} [RPM]	ω_r [RPM]
X	-0.188	+1033
Y	-0.293	+1613
Z	+0.987	-5431

assi	ω_{sat} [RPM]	ω_r [RPM]
X	$-3.20 \cdot 10^{-3}$	+18
Y	$-10.33 \cdot 10^{-3}$	+57
Z	$-2.57 \cdot 10^{-3}$	+14

CASO 3

T_{man}	27 s
T_{idle}	53s

Rotazione preliminare

	\bar{E}_{sns} [°]	\bar{E}_{at} [°]	$\bar{E}_{sns \rightarrow at}$ [°]
Ψ	0	0	-63.382
Θ	0	20	-27.671
Φ	0	-40	-101.004

Rotazione di puntamento

assi	ω_{sat} [RPM]	ω_r [RPM]
X	+0.146	-802
Y	+0.202	-1113
Z	+0.970	-5335

assi	ω_{sat} [RPM]	ω_r [RPM]
X	$+3.21 \cdot 10^{-3}$	-18
Y	$-10.59 \cdot 10^{-3}$	+58
Z	$+0.98 \cdot 10^{-3}$	-5

CASO 4

T_{man}	38 s
T_{idle}	42 s

Rotazione preliminare

	\bar{E}_{sns} [°]	\bar{E}_{at} [°]	$\bar{E}_{sns \rightarrow at}$ [°]
Ψ	0	135	+140.556
Θ	0	-10	+40.716
Φ	0	-15	+117.688

Rotazione di puntamento

assi	ω_{sat} [RPM]	ω_r [RPM]
X	-0.273	+1502
Y	+0.345	-1897
Z	-987	+5409

assi	ω_{sat} [RPM]	ω_r [RPM]
X	$+10.08 \cdot 10^{-3}$	-55
Y	$-3.23 \cdot 10^{-3}$	+18
Z	$-3.37 \cdot 10^{-3}$	+19

CASO 5

Effettuiamo la conversione da Assi ad Angoli. Data la matrice:

$$Ax_{at} = \begin{bmatrix} 0 & 0 & -1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix}$$

che rappresenta gli assi della nuova terna di riferimento espressa rispetto agli assi ECI, ricaviamo la terna di angoli di Eulero, necessaria per il calcolo del puntamento inerziale:

$$\bar{E}_{at} = EuAt2Ang (Ax_{at}) = \begin{bmatrix} 0 \\ 90 \\ 0 \end{bmatrix}$$

Da cui otteniamo i seguenti risultati.

T_{man}	17
T_{idle}	63

Rotazione preliminare

	\bar{E}_{sns} [°]	\bar{E}_{at} [°]	$\bar{E}_{sns \rightarrow at}$ [°]
Ψ	0	0	-70.580
Θ	0	90	+11.716
Φ	0	0	-29.942

Rotazione di puntamento

assi	ω_{sat} [RPM]	ω_r [RPM]
X	-0.170	+933
Y	+0.064	-354
Z	+0.971	-5341

assi	ω_{sat} [RPM]	ω_r [RPM]
X	$+10.74 \cdot 10^{-3}$	-59
Y	$-2.10 \cdot 10^{-3}$	+12
Z	$+1.96 \cdot 10^{-3}$	-11

7.4. SIMULAZIONE DI GROUND TRACKING

La simulazione del ground tracking è la più complessa di tutte. In ingresso vengono forniti i parametri di latitudine e longitudine della stazione di terra. Il software si occupa di fornire, per ogni orbita coinvolta, i valori degli angoli calcolati ad ogni secondo, utilizzabili come feedback. Inoltre vengono visualizzati i dati di velocità ed accelerazione angolare del satellite e delle ruote di reazione sui tre assi per il controllo attivo della posizione.

Effettueremo alcune simulazioni di ground tracking per le seguenti località:

1. Torino Politecnico (45.063 *N*, 7.662 *E*)
2. Boston, MIT (42.36 *N*, 71.10 *O*)
3. Tokio (35.68 *N*, 139.77 *E*)
4. Reykjavik (64,10 *N*, 21.90 *O*)
5. Cape Town (33.91 *S*, 18.42 *E*)

I luoghi sono stati scelti sia per rilevanza scientifica che per testare il sistema di controllo per varie località sparse sul globo terrestre.

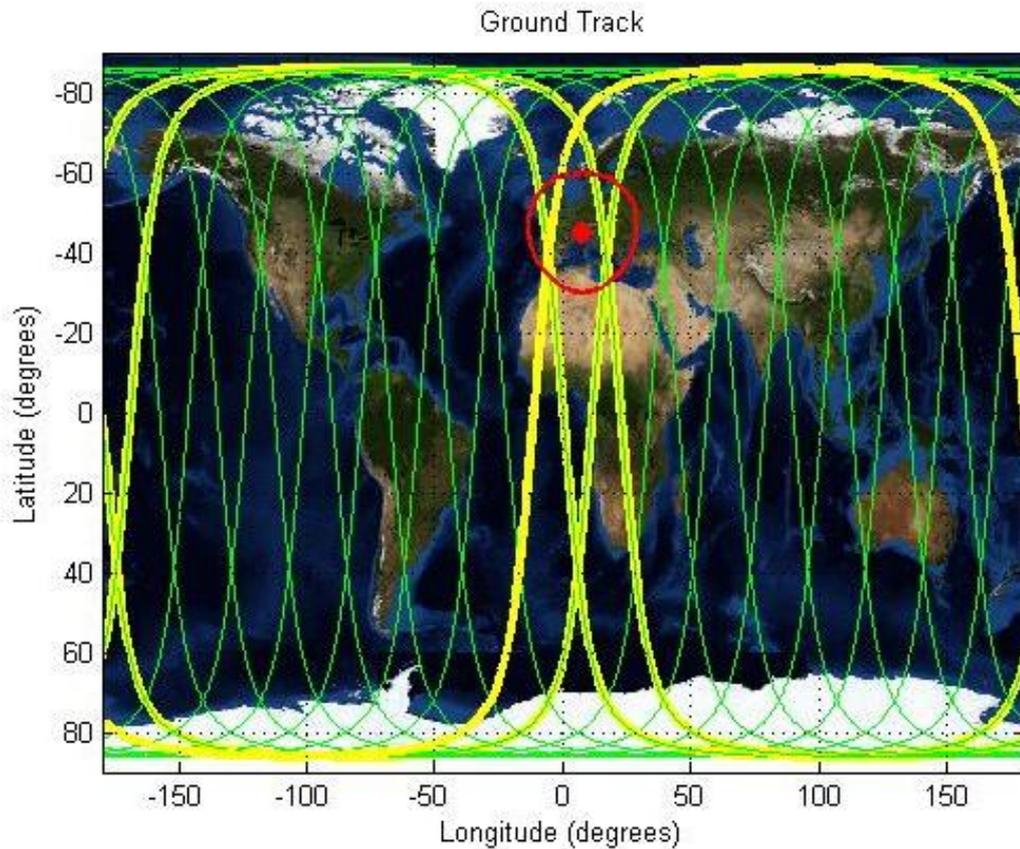
7.4.1. Politecnico di Torino

Il primo punto che supponiamo di dover tracciare è la zona del Politecnico di Torino, dove è situata la stazione per le telecomunicazioni col satellite. E' quindi ragionevole pensare che il satellite si trovi a dover essere puntato verso questo luogo più volte nella sua vita operativa.

Le coordinate, in gradi, e leggermente approssimate sono:

$$\begin{cases} \lambda_1 = 45.063^\circ N \\ l_1 = 7.662^\circ E \end{cases}$$

Di seguito riportiamo i dati provenienti dal software.



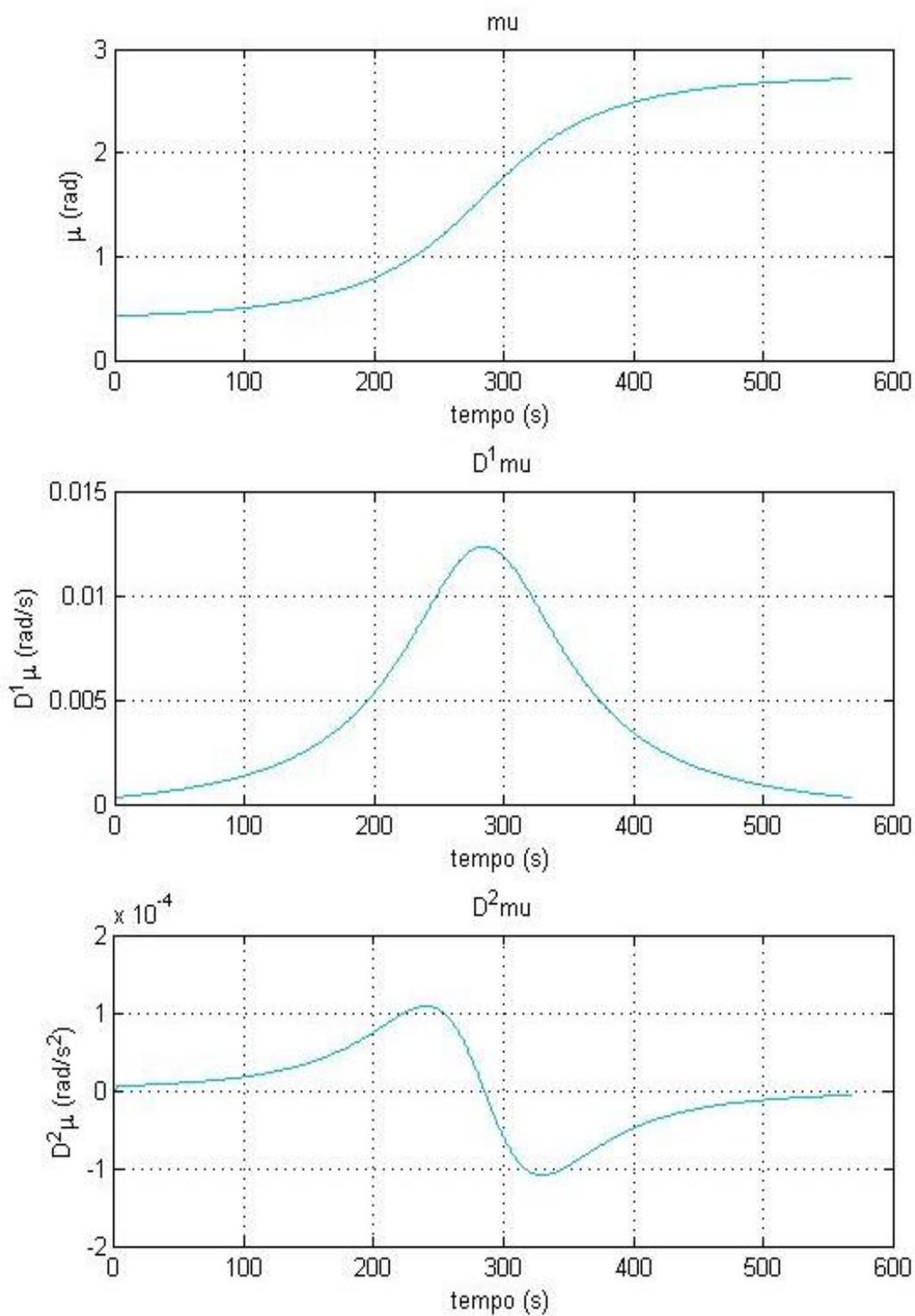
7.1 Ground track dei satelliti in grado di puntare la stazione di Torino Politecnico

Nell'immagine precedente si osservano in giallo le quattro orbite del satellite coinvolte per la manovra di tracking.

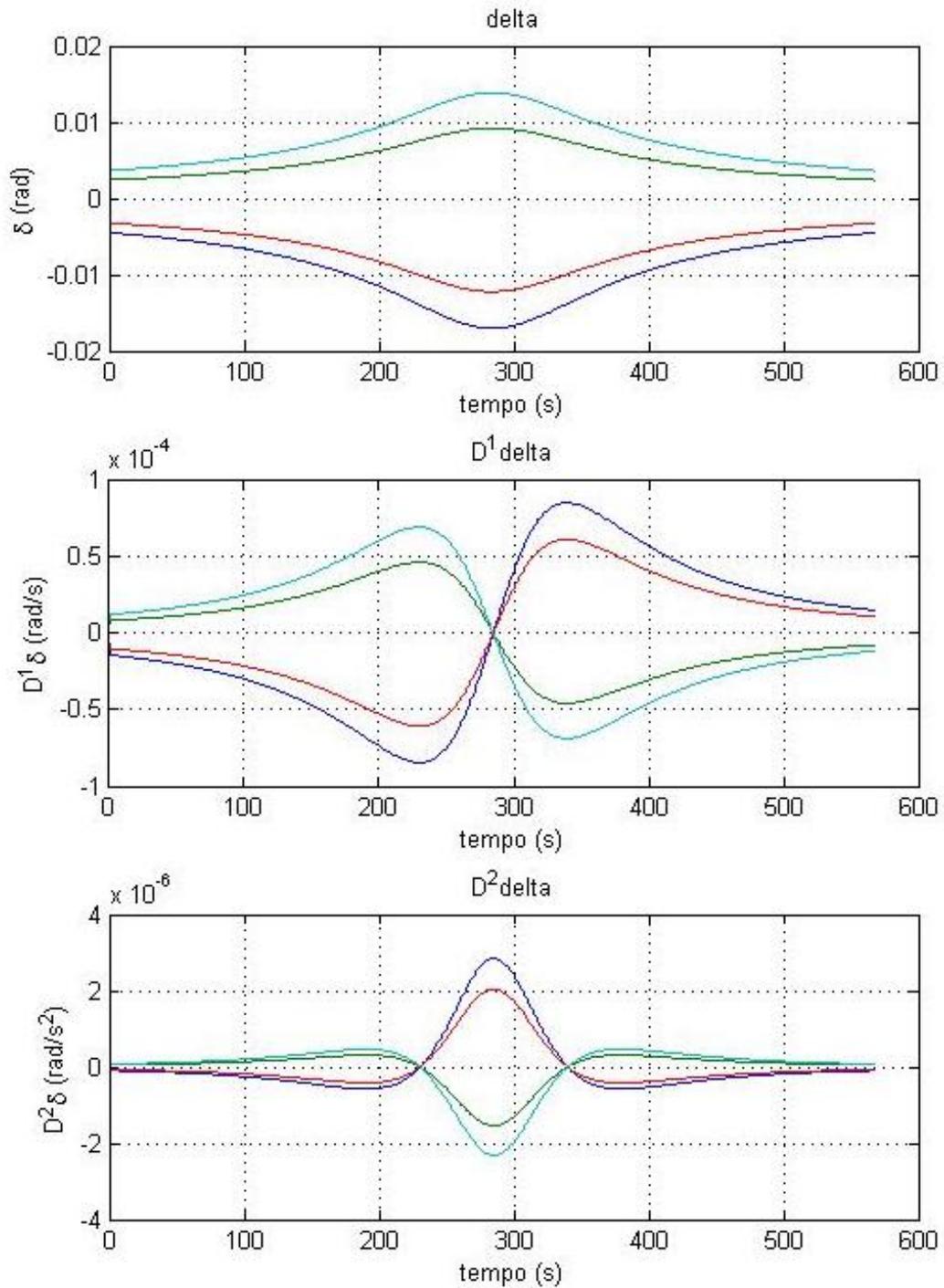
Successivamente vengono riportati i grafici relativi agli angoli μ e δ , con le rispettive derivate, e le velocità che devono assumere le ruote d'inerzia per poter permettere al satellite di eseguirle.

Ovviamente come già visto nel paragrafo 6.6 il movimento che deve compiere il satellite rispetto alla verticale, ossia la rotazione di angolo μ , è sempre la medesima per qualunque orbita e qualunque Ground Station si voglia tracciare, poiché i parametri che lo definiscono non dipendono dall'orbita.

Riportiamo qui di seguito quindi il grafico dell'andamento dell'angolo μ , evitando di riportarlo nelle simulazioni seguenti.



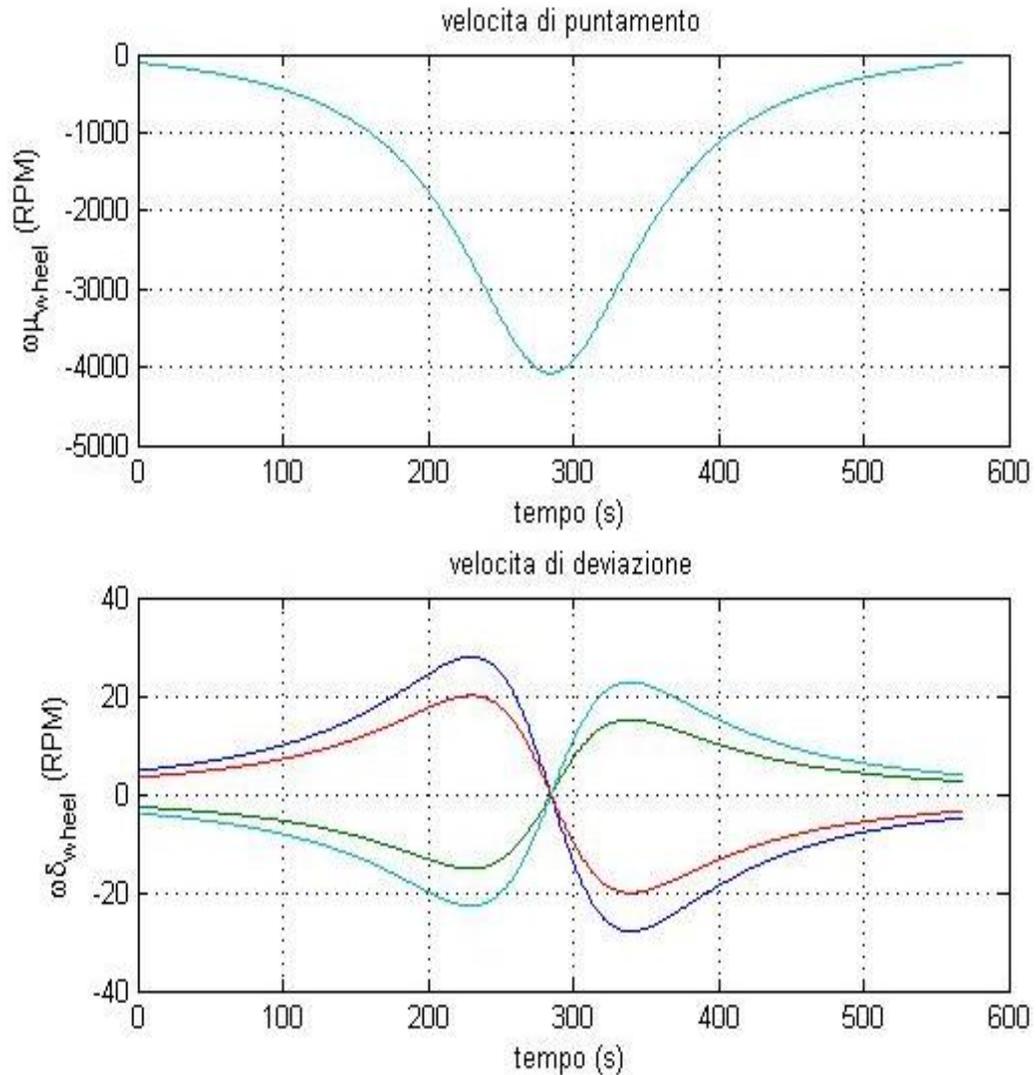
7.2 andamento dell'angolo di puntamento μ (in alto) e delle sue derivate, prima(in centro) e seconda (in basso)



7.3 andamento dell'angolo di deviazione δ (in alto) e delle sue derivate, prima(in centro) e seconda (in basso)

Nell'immagine precedente si vede invece come varia l'angolo δ col tempo e in funzione dell'orbita.

Le 4 curve possono apparire a due a due simmetriche, ma tale effetto è dovuto alla particolarità della Ground station di trovarsi pressoché equispaziata tra le 4 orbite che la comprendono.



7.4 andamento delle componenti di velocità da imprimere alle ruote

Nel grafico 7.4 vediamo quali sono le componenti μ e δ della velocità angolare che occorre imprimere al satellite. Come prima la velocità angolare μ ha il medesimo andamento in tutti e quattro i casi.

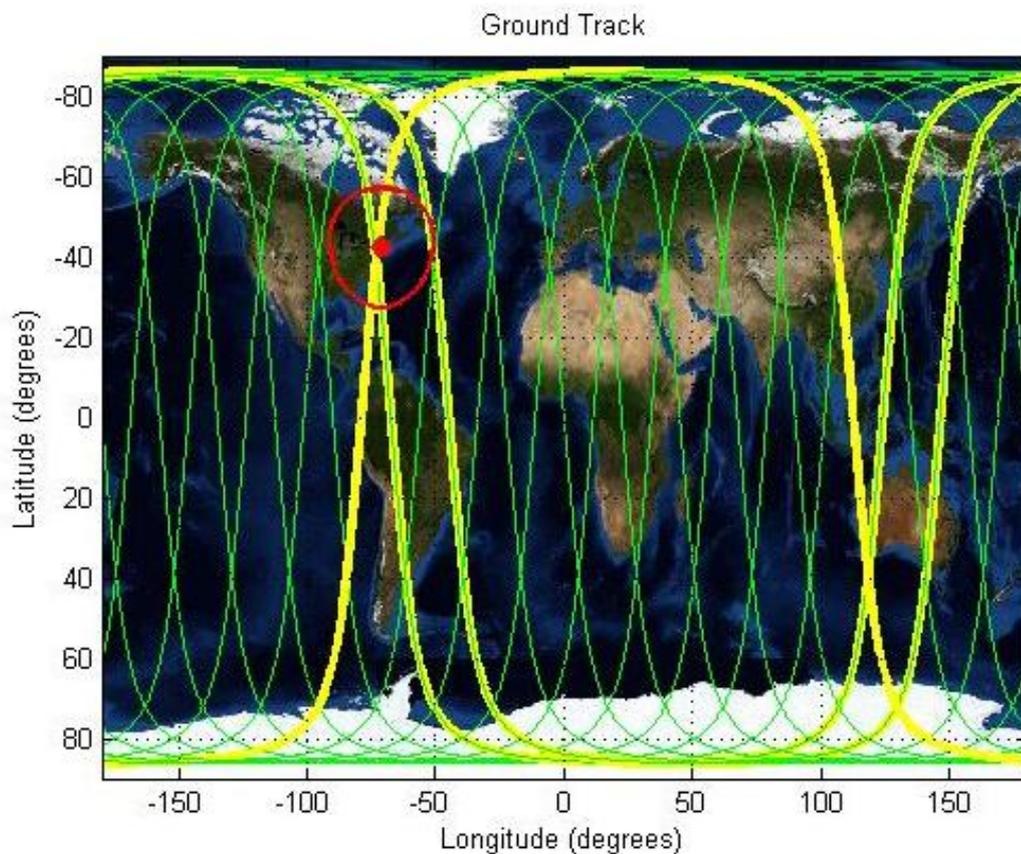
7.4.2. Boston, MIT

La seconda stazione che scegliamo di puntare è il MIT di Boston, supponendo che possa essere una seconda base, gemellata con il nostro Politecnico di Torino.

Le sue coordinate sono:

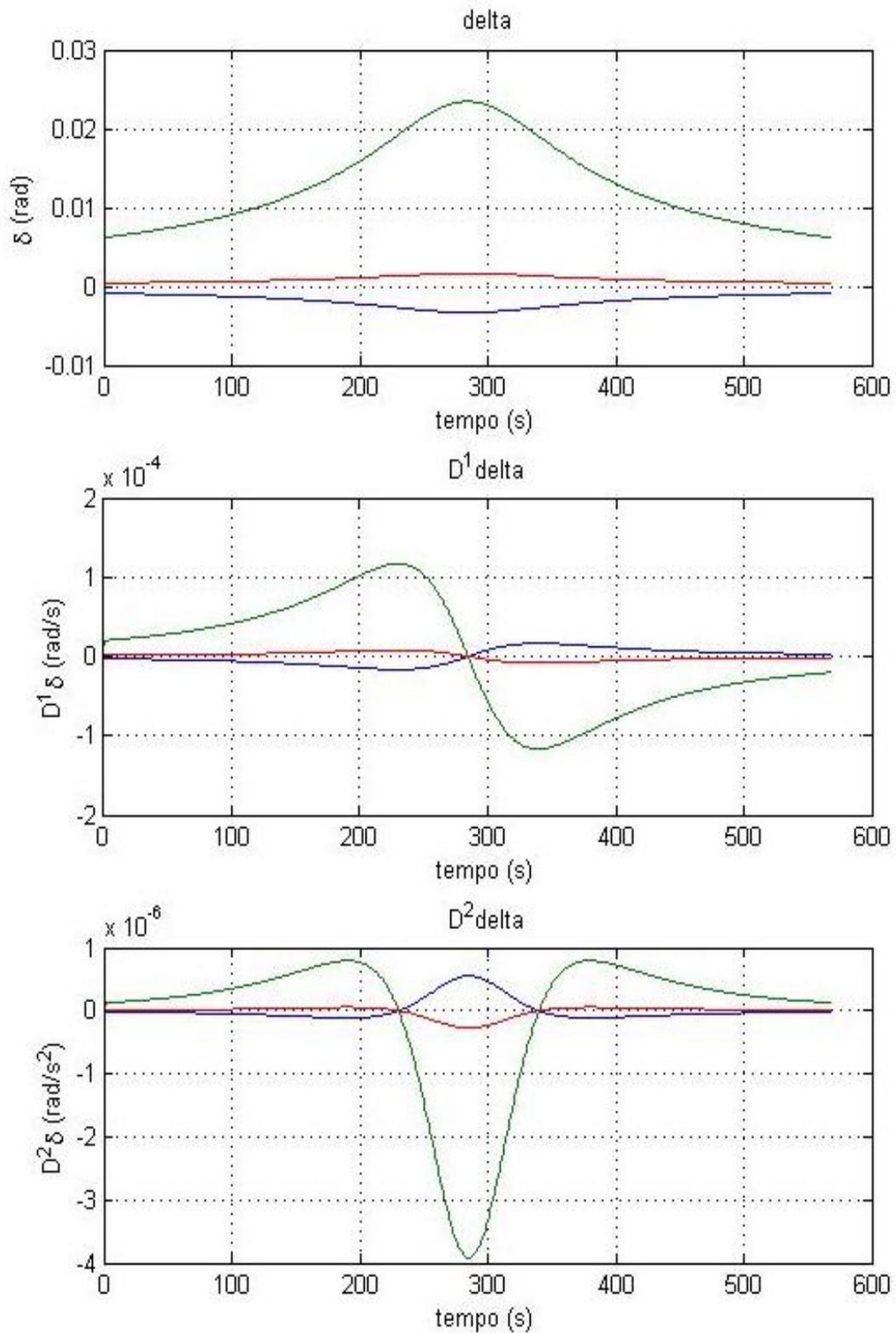
$$\begin{cases} \lambda_1 = 42.36^\circ N \\ l_1 = 71.10^\circ O \end{cases}$$

Dalla simulazione pervengono i seguenti dati:

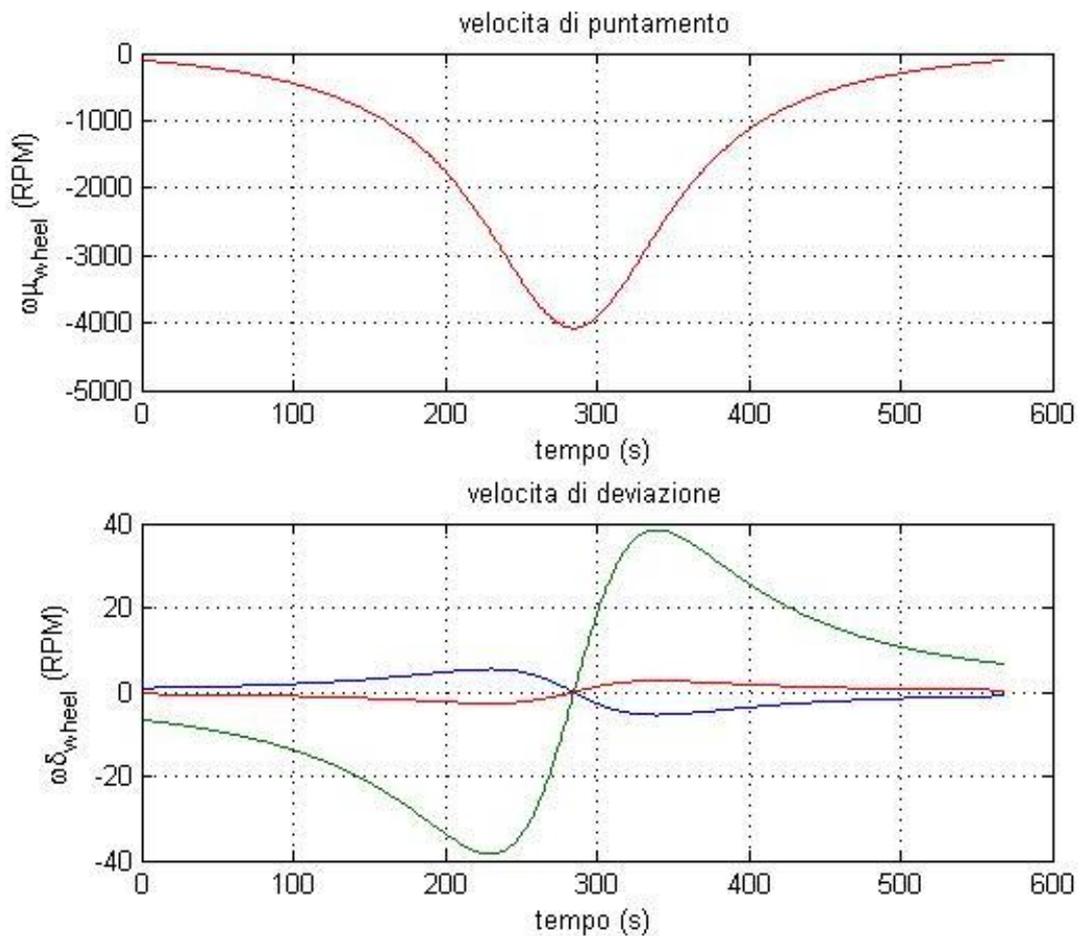


7.5 Ground track dei satelliti in grado di puntare il MIT di Boston

Vengono omessi per brevità i dati sull'andamento dell'angolo μ in quanto restano invariati rispetto al caso precedente. Per maggiori dettagli vedasi fig. 7.2.



7.6 andamento dell'angolo di deviazione δ (in alto) e delle sue derivate, prima(in centro) e seconda (in basso)



7.7 andamento delle componenti di velocità da imprimere alle ruote

Nel grafico 7.7 vediamo quali sono le componenti μ e δ della velocità angolare che occorre imprimere al satellite.

La differenza tra la curva rossa e quella blu rappresentano in un caso la presenza della ground station a destra del satellite e nell'altro la presenza a sinistra.

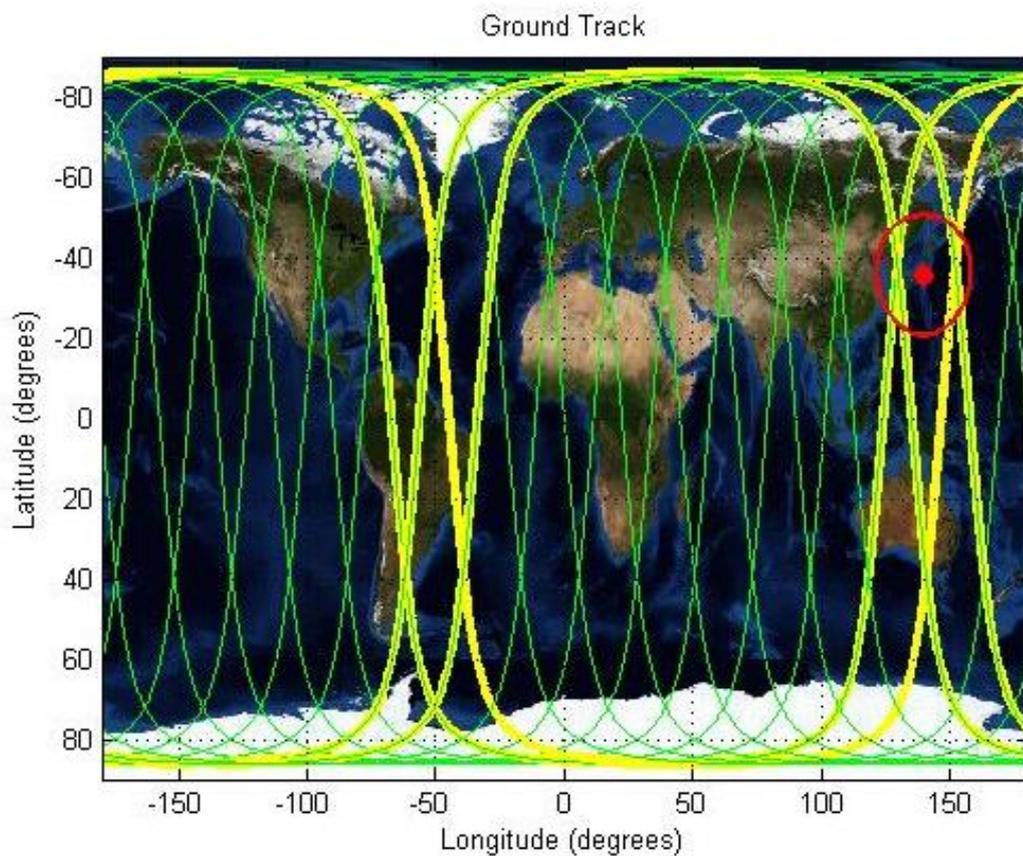
L'ampiezza maggiore della curva verde invece è dovuta alla maggiore distanza del punto da tracciare rispetto alla ground track.

7.4.3. Tokio

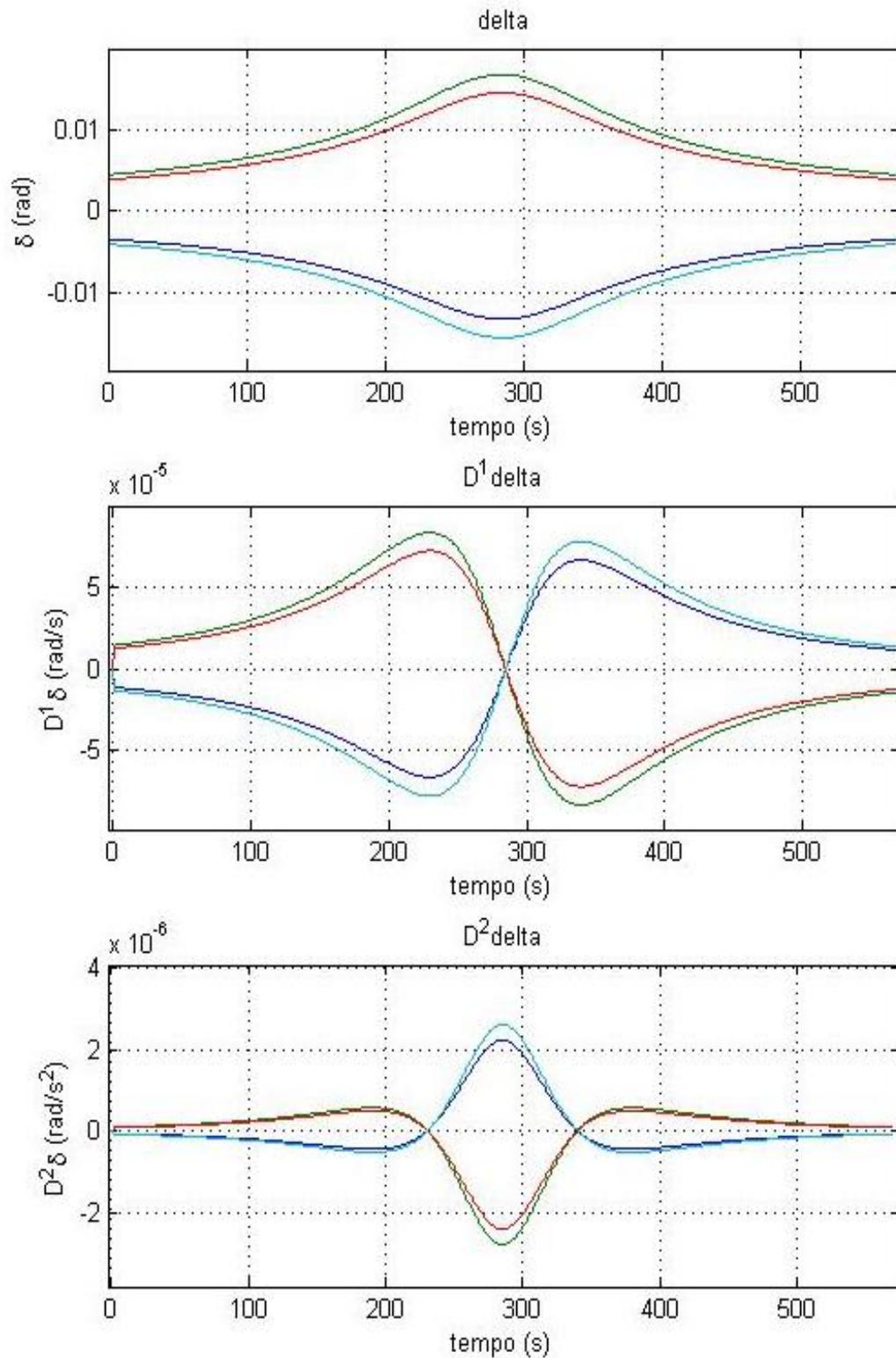
Per verificare e testare la bontà dell'algoritmo sono state scelte altre città per le loro peculiarità. Tokyo ad esempio, è scelta in quanto si tratta di una città posta ad una longitudine notevole. Le sue coordinate terrestri sono:

$$\begin{cases} \lambda_1 = 35.68^\circ N \\ l_1 = 139.77^\circ E \end{cases}$$

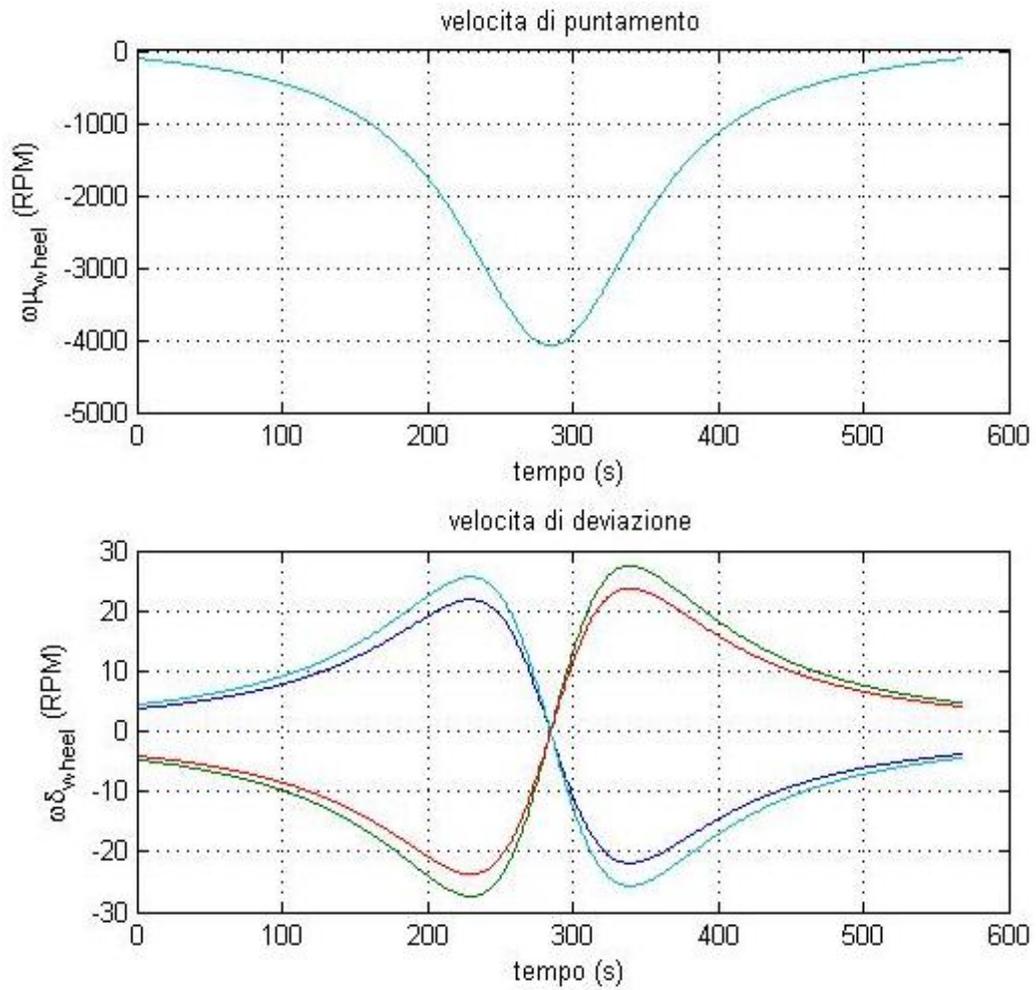
Dalla simulazione si ricavano le orbite coinvolte:



7.8 Ground track dei satelliti in grado di puntare la città di Tokyo



7.9 andamento dell'angolo di deviazione δ (in alto) e delle sue derivate, prima (in centro) e seconda (in basso)



7.10 andamento delle componenti di velocità da imprimere alle ruote

Nel grafico 7.10 vediamo quali sono le componenti μ e δ della velocità angolare che occorre imprimere al satellite.

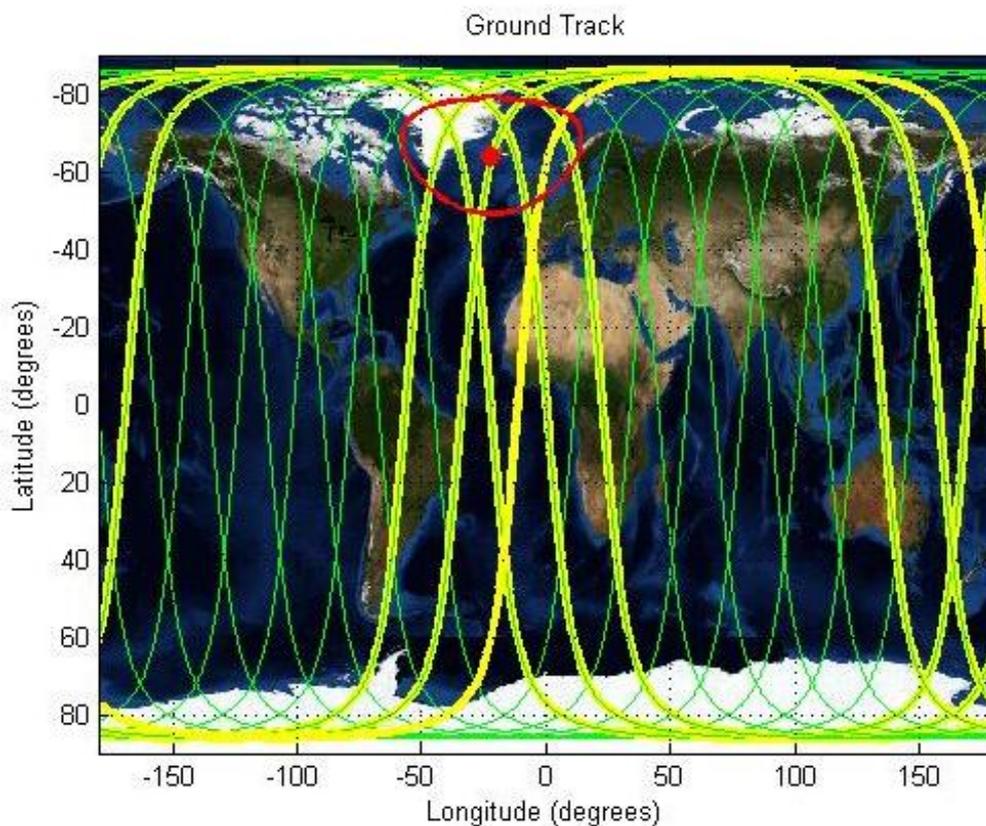
Notiamo inoltre che le curve sono molto simili a due a due e speculari rispetto all'asse x. Questo perché la città di Tokyo è posizionata centralmente rispetto alle 4 orbite che la tracciano.

7.4.4. Reykjavik

Un'altra città che vale la pena di considerare è Reykjavik, in quanto capitale più settentrionale al mondo. Le sue coordinate sono le seguenti

$$\begin{cases} \lambda_1 = 64.10^\circ N \\ l_1 = 21.90^\circ O \end{cases}$$

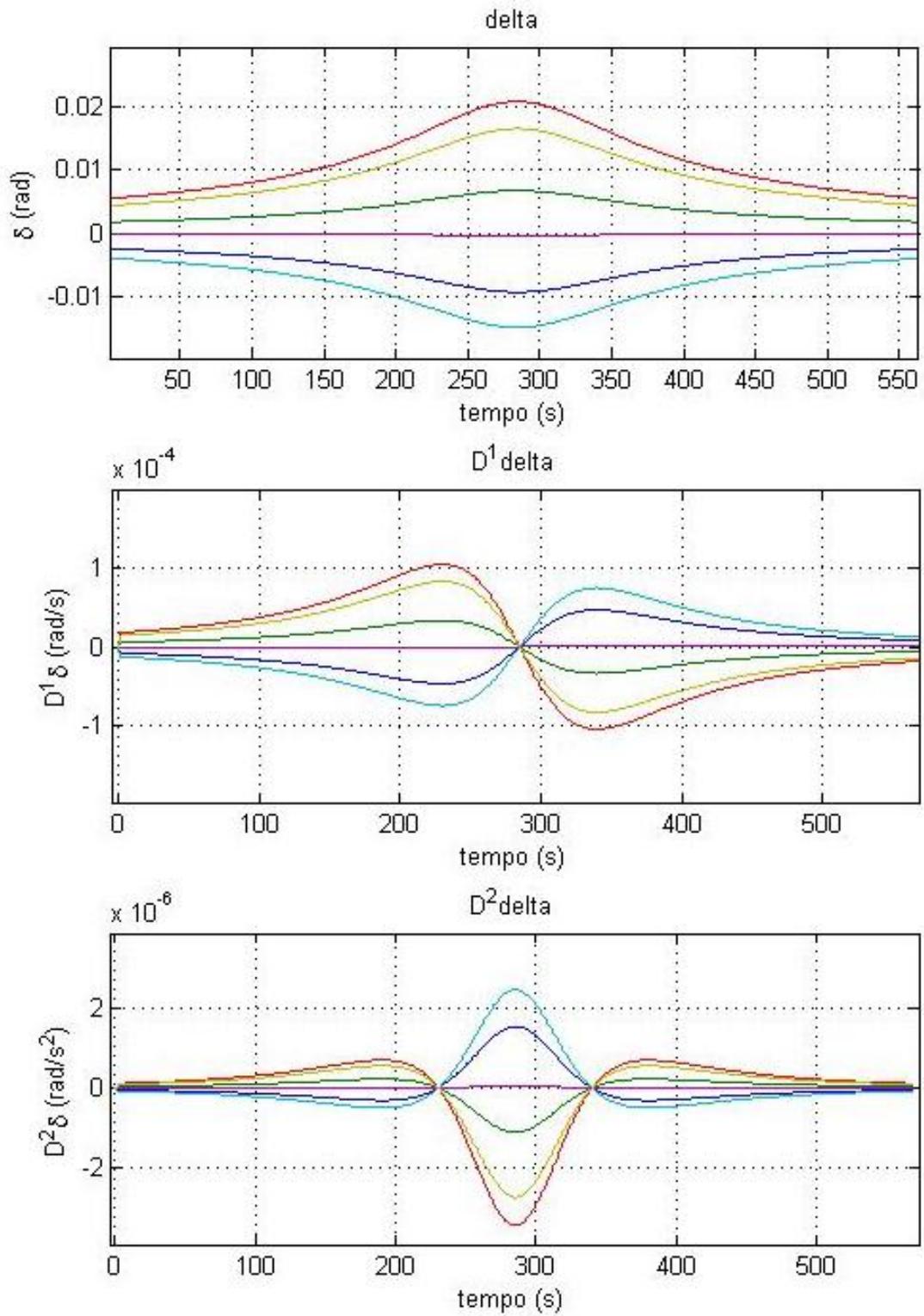
E il software ci fornisce i seguenti risultati:



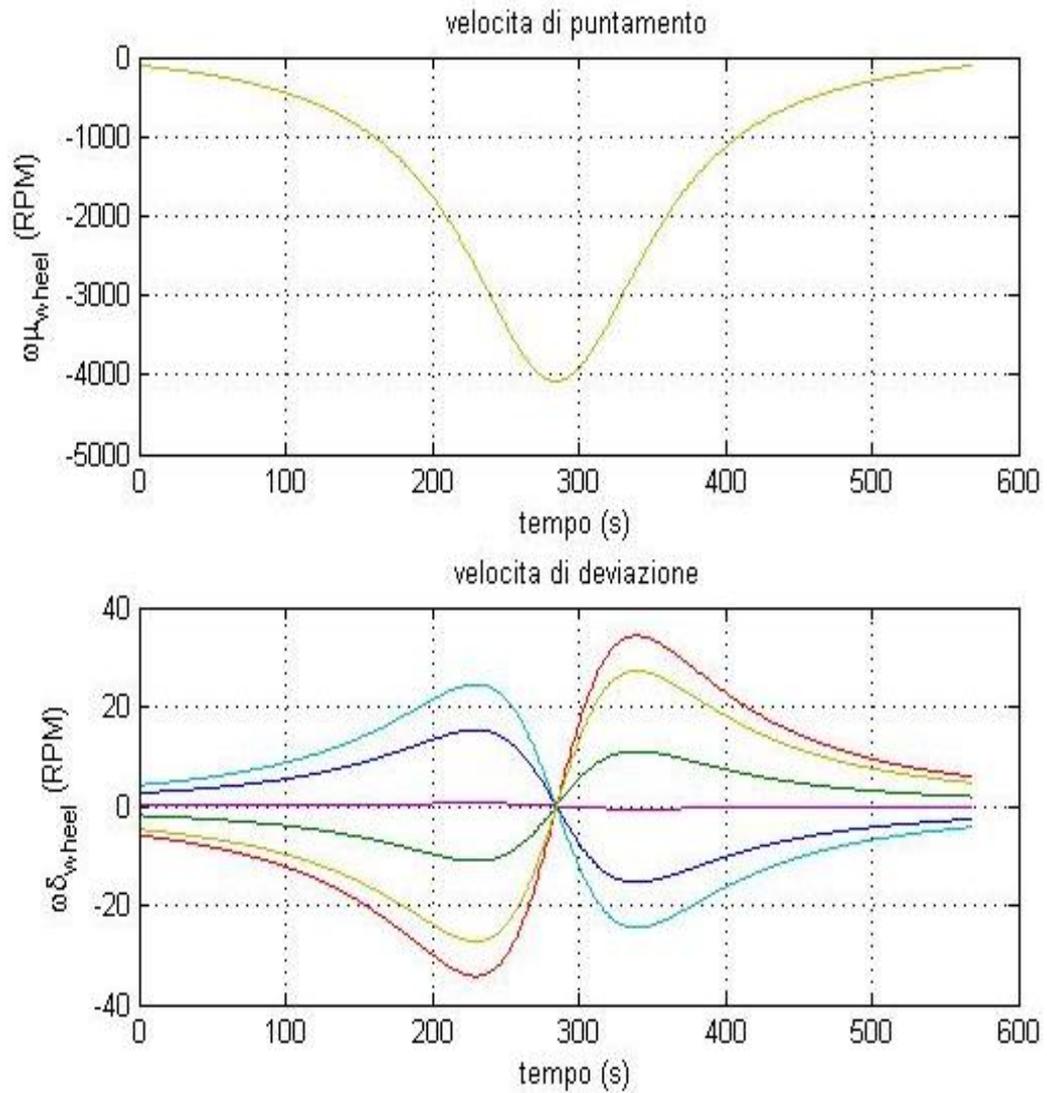
7.11 Ground track dei satelliti in grado di puntare la città di Reykjavik

E' immediato notare come, essendo tanto a nord, è tracciata da una moltitudine di orbite, ben sei. Ciò avviene perché le distanze tra gli apici delle orbite è inferiore alle alte latitudini a causa della geometria terrestre.

Si noti altresì che un'orbita passa pressoché perpendicolarmente alla verticale locale. Si vedranno in seguito le conseguenze.



7.12 andamento dell'angolo di deviazione δ (in alto) e delle sue derivate, prima(in centro) e seconda (in basso)



7.13 andamento delle componenti di velocità da imprimere alle ruote

La traccia in viola rappresenta esattamente l'orbita sopra la città,. La peculiarità di tale orbita è l'angolo di deviazione dalla ground track prossimo allo zero.

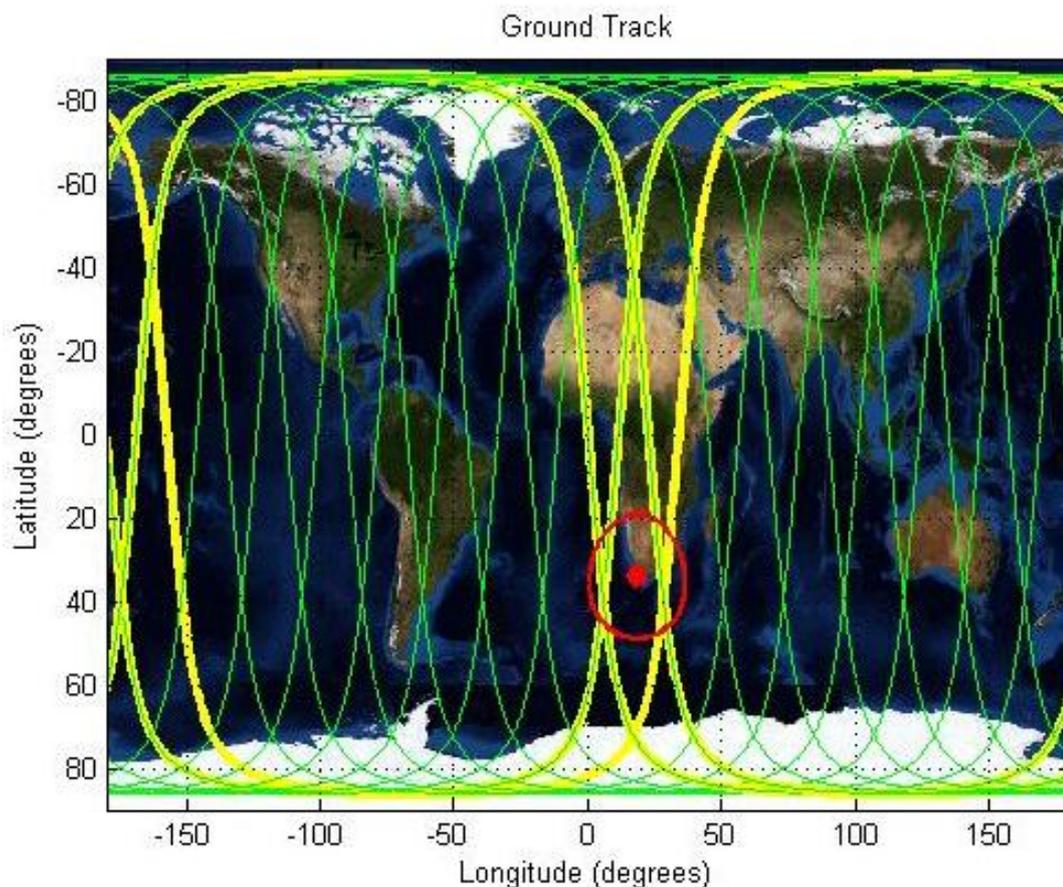
7.4.5. Cape Town

L'ultima città considerata nelle simulazioni è Città del Capo. E' stato difficile scegliere una città che fosse sufficientemente a sud dell'equatore per valutare i comportamenti di un tracciamento, infatti i maggiori poli scientifici e tecnologici sono situati a nord della terra.

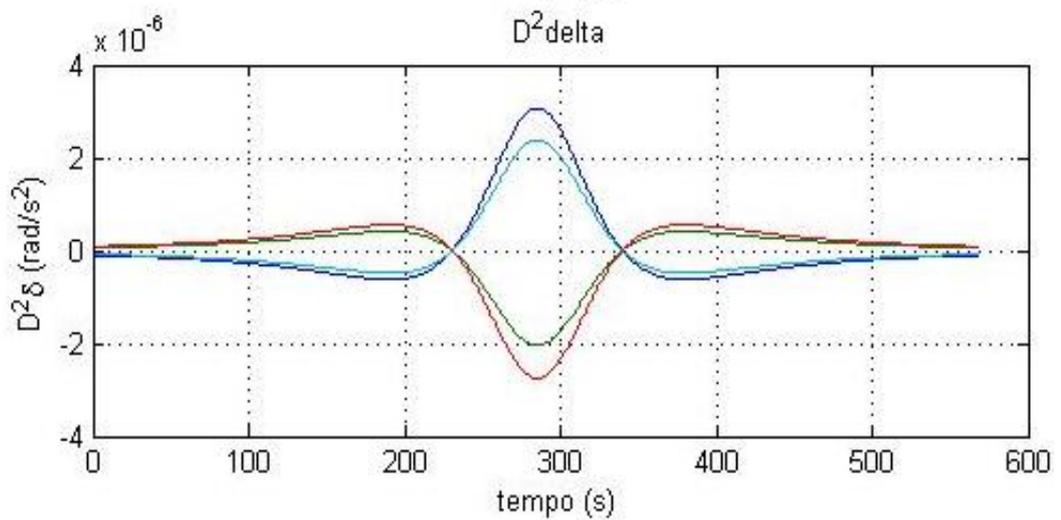
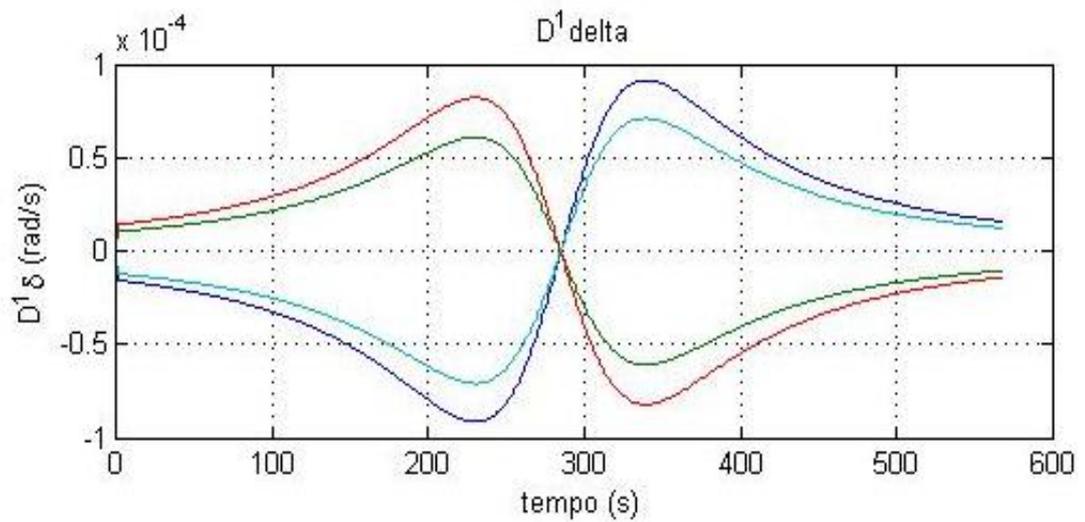
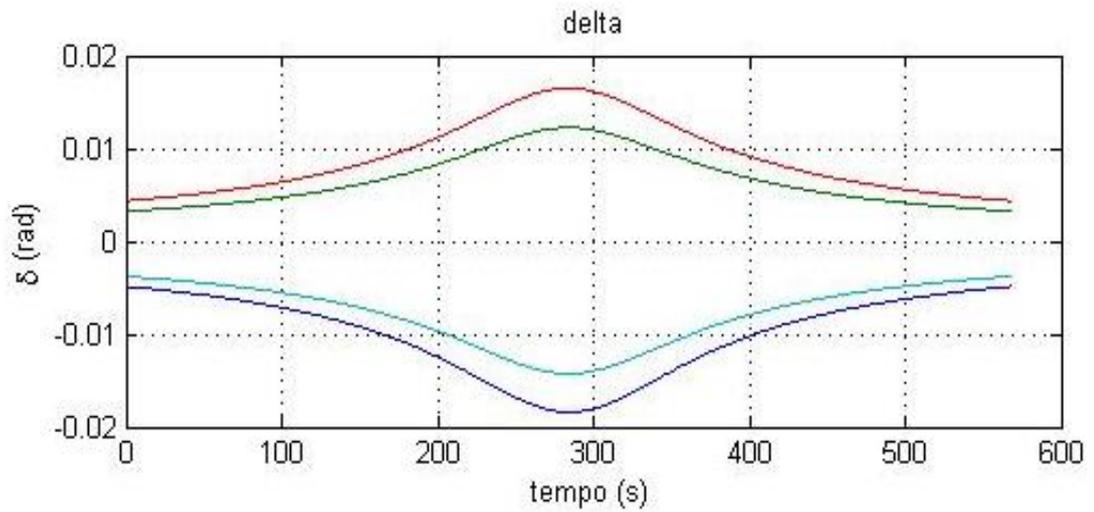
Si è quindi scelta Cape Town per la sua posizione geografica e nulla più. Le sue coordinate geografiche sono:

$$\begin{cases} \lambda_1 = 33.91^\circ S \\ l_1 = 18.42^\circ E \end{cases}$$

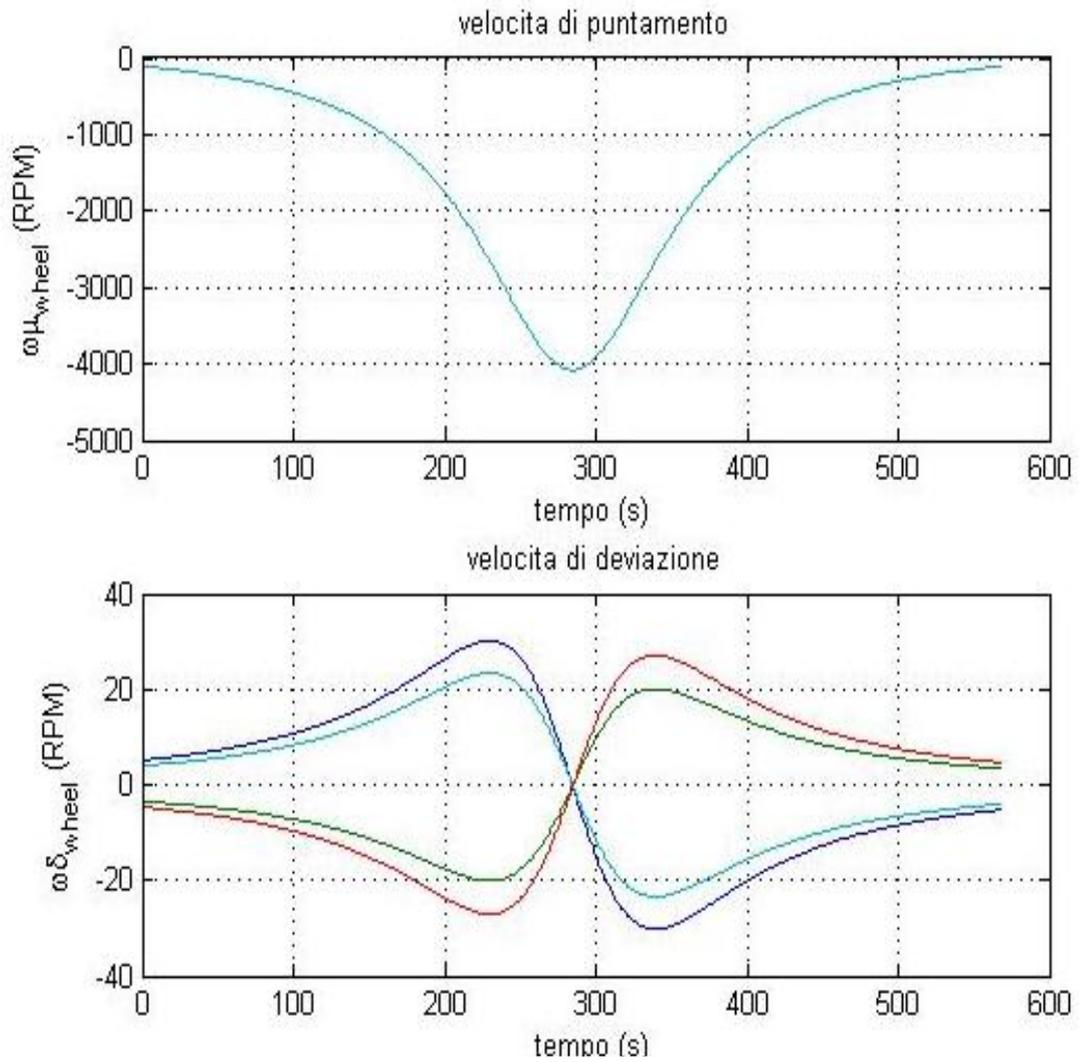
Da cui, tramite software, otteniamo i seguenti risultati:



7.14 Ground track dei satelliti in grado di puntare la città di Cape Town



7.15 andamento dell'angolo di deviazione δ (in alto) e delle sue derivate, prima(in centro) e seconda (in basso)



7.16 andamento delle componenti di velocità da imprimere alle ruote

Vediamo anche in questo caso un comportamento del tutto analogo ai precedenti, senza particolarità rilevanti da segnalare.

8. APPENDICI

APPENDICE A: FILE DI MATLAB

A.1. Attitude control system ([Att_CtrlSystem.m](#))

Routine principale di controllo d'assetto,

```
%%Att_CtrlSystem.m
%
% questo programma simula la variazione d'assetto del satellite in
% funzione dei comandi inviati da terra (attitude.dat), dei
% parametri orbitali rilevati (orbit.dat), e dalle informazioni
% ricevute dai sensori (loadSensor). Vengono caricati i dati ed
% elaborate le matrici di rotazione. successivamente viene
% attivata una e una sola routine in base alle scelte riportate
% sul file di controllo assetto. E' possibile un solo controllo
% alla volta per semplicità. nel caso due controlli fossere da
% eseguire si procede dando due comandi a breve distanza
% temporale.
%
%
%Author: Danilo Muru, 2011
%


---


clear all
clc

LoadData ;           %Carica I dati di navigazione da seguire
LoadSensor;         %Carica i dati

if Simu_en==1;
    fprintf('\n Selezionare il controllo di assetto voluto: \n
1=Assetto fisso inerziale; \n 2=Rotazione assi; \n 3=Spin-Despin;
\n 4=Tracking;');
    i = input ('\n Selezione: ');
    Attitude_en(i)=1;
end

if Attitude_en(1)==1; %Verifica se il comando IFix è
abilitato.
    Att_Sub_IFix;     %Se abilitato avvia la routine di
controllo scelta

    Attitude_en =zeros(1,4); %Azzera il vettore dei comandi
d'assetto

elseif Attitude_en(2)==1; %Verifica se il comando rotation è
abilitato.
    Att_Sub_Rot;     %Se abilitato avvia la routine di
controllo scelta
```

```

    Attitude_en =zeros(1,4); %Azzera il vettore dei comandi
d'assetto

elseif Attitude_en(3)==1; %Verifica se il comando spinning è
abilitato.
    bypass = 0;
    Att_Sub_Spi; %Se abilitato avvia la routine di
controllo scelta

    Attitude_en =zeros(1,4); %Azzera il vettore dei comandi
d'assetto

elseif Attitude_en(4)==1; %Verifica se il comando tracking è
abilitato.
    Att_Sub_Tra; %Se abilitato avvia la routine di
controllo scelta

    Attitude_en =zeros(1,4); %Azzera il vettore dei comandi
d'assetto

else %Nessun comando valido è stato fornito
    fprintf ('\n WARNING \n Nessun Comando fornito \n')
    fprintf (' Nessuna azione verrà compiuta \n');
    Attitude_en =zeros(1,4); %Azzera il vettore dei comandi
d'assetto
end

```

A.2. Spin sub system (**Att_Sub_Spi.m**)

Subroutine di controllo d'assetto: SPIN-DESPIN

```

%%Att_Sub_Spi.m

% Questa funzione una volta richiamata permette di calcolare i
% movimenti necessari a produrre una determinata variazione di
% velocità angolare dati come parametri di ingresso il nuovo
% vettore velocità angolare riferito agli assi body.
% l'orientamento del satellite può essere modificato in un secondo
% momento.
%
% Author: Danilo Muru, 2011
%

%Scelta dell'obiettivo da raggiungere.
if bypass ~= 1;
    set_spin = input('\n Impostare manualmente il vettore velocità
angolare da raggiungere?\n 1=Set, 2=Read from attitude, 3=No spin;
');
else
    set_spin=3;
end

```

```

if      set_spin == 1; %Imposta da prompt il vett rotazione scelto
Wa_Xb = input('\nWa_Xb : ');
Wa_Yb = input('\nWa_Yb : ');
Wa_Zb = input('\nWa_Zb : ');
Wa_body = [Wa_Xb, Wa_Yb, Wa_Zb];
elseif set_spin == 3; % Imposta il vett. nullo,ferma il satellite
Wa_Xb = 0;
Wa_Yb = 0;
Wa_Zb = 0;
Wa_body = [Wa_Xb, Wa_Yb, Wa_Zb];
end

Wd_body = Wa_body-Ws_body; % Calcolo la differenza di velocità
                          % angolare tra La condizione di
                          % partenza e l'assetto voluto.

%Comando d'assetto per spin usando il controllo delle ruote
inerziali:
fprintf('\n Soluzione mediante uso di ruote inerziali:')
if      Wd_body(1) ~= 0; %se lo spin non coincide occorre agire
sul controllo.
Wx_wheel= -(Ixx/I_wheel)*Wd_body(1);
if Wx_wheel<=MaxRPMW && Wx_wheel>=-MaxRPMW;
    fprintf('\n Asse X: Imprimere una velocità di %-4.2f RPM alla
ruota;', Wx_wheel);
else
    fprintf('\n Asse X: Velocità richiesta fuori dai parametri di
missione;');
end
end

if Wd_body(2) ~= 0; %se lo spin non coincide occorre agire sul
controllo.
Wy_wheel= -(Iyy/I_wheel)*Wd_body(2);
if Wy_wheel<=MaxRPMW && Wy_wheel>=-MaxRPMW;
    fprintf('\n Asse Y: Imprimere una velocità di %+4.2f RPM alla
ruota;', Wy_wheel);
else
    fprintf('\n Asse Y: Velocità richiesta fuori dai parametri di
missione;');
end
end

if Wd_body(3) ~= 0; %se lo spin non coincide occorre agire sul
controllo.
Wz_wheel= -(Izz/I_wheel)*Wd_body(3);
if Wz_wheel<=MaxRPMW && Wz_wheel>=-MaxRPMW;
    fprintf('\n Asse Z: Imprimere una velocità di %+4.2f RPM alla
ruota;\n', Wz_wheel);
else
    fprintf('\n Asse Z: Velocità richiesta fuori dai parametri di
missione;\n');
end
end
Ws_body = Wa_body;

```

A.3. Rotation sub system (Att_Sub_Rot.m)

SubRoutine di controllo d'assetto: ROTAZIONE

```
%%Att_Sub_Rot.m
%
% Questa funzione una volta richiamata permette di calcolare i
% movimenti necessari a produrre una determinata rotazione dati
% come parametri di ingresso i tre angoli di Eulero rispetto al
% riferimento orbita
%
% Author: Danilo Muru, 2011
%
if Ws_body ~= 0;    %Ferma la rotazione del satellite
    bypass=1;
    Att_Sub_Spi;
end

[E_BtoA] = EuAng_Rot(E_at,E_sns);
PSI_BtoA = E_BtoA(1); % Carica gli angoli di eulero necessari a
THE_BtoA = E_BtoA(2); % effettuare la rotazione tra la condizione
PHI_BtoA = E_BtoA(3); % iniziale (i) e la condizione finale(f).
    if PSI_BtoA>180
        PSI_BtoA = PSI_BtoA-2*pi;
    end
    if THE_BtoA>180
        THE_BtoA = THE_BtoA-2*pi;
    end
    if PHI_BtoA>180
        PHI_BtoA = PHI_BtoA-2*pi;
    end

T_m=1    ;
ok=0    ;
while ok==0
% definisce le derivate degli angoli di Eulero
PSI_d = PSI_BtoA/T_m;
THE_d = THE_BtoA/T_m;
PHI_d = PHI_BtoA/T_m;

% calcola le componenti e il vettore velocità angolare da
% applicare
Ws1 =  THE_d*cos(PHI_BtoA)+PSI_d*sin(THE_BtoA)*sin(PHI_BtoA);
Ws2 = -THE_d*sin(PHI_BtoA)+PSI_d*sin(THE_BtoA)*cos(PHI_BtoA);
Ws3 =  PHI_d + PSI_d*cos(THE_BtoA);

WRs = [Ws1, Ws2, Ws3];
WRr = WRs.*[Ixx Iyy Izz]/I_wheel;

% verifica della condizione di velocità massima
if abs(WRr(1))<=575.6 && abs(WRr(2))<=575.6 && abs(WRr(3))<=575.6
    ok=1;
else
    T_m=T_m+1;
end
```

```

End
% Se analizziamo la configurazione cubica otteniamo:
WRr = WRs.*[Ixx Iyy Izz]/I_wheel;

fprintf('\n Rotazione per %2.1f secondi,', T_m);
fprintf('\n Vel. del satellite:| Vel. ruote pilotate:')
fprintf('\n W_x %+1.3f RPM      | Wr_x  % 5.0f RPM',-Ws1,WRr(1));
fprintf('\n W_y %+1.3f RPM      | Wr_y  % 5.0f RPM',-Ws2,WRr(2));
fprintf('\n W_z %+1.3f RPM      | Wr_z  % 5.0f RPM' , -Ws3,WRr(3));
fprintf('\n [+] Rotazione Antioraria, [-] Rotazione Oraria\n')

```

A.4. Inertial pointing sub system (Att_Sub_IFix.m)

SubRoutine di controllo d'assetto: PUNTAMENTO INERZIALE

```

%%Att_Sub_IFix.m
%
% Questa funzione una volta richiamata permette di calcolare i
% movimenti necessari a produrre una determinata e continua
% rotazione tale da mantenere sempre costante il puntamento
% rispetto a un riferimento inerziale, ad esempio una stella, e/o
% a garantire il puntamento di uno strumento, dati come parametri
% di ingresso la posizione del satellite e dell'assetto rispetto
% al sistema fisso inerziale (ECI nel nostro caso).
%
% Author: Danilo Muru, 2011
%

E_atI = [1, 1.3, 0.5]; % Definiamo un assetto arbitrario da
                    % raggiungere in assi ECI (angoli di Eulero)
E_at_bk = E_at;% Salva il vettore assetto in una var. di backup.

t_man = 80; % Definiamo un tempo [s] i cui eseguire la manovra.
          % poichè il satellite si sposta con velocità
          % angolare costante rispetto a ECI e non è
          % trascurabile ai fini dell'allineamento.

[E_AI]= EuAng_Rot(E_at, [0,0,0]); % Dati gli angoli di eulero
[ROT] = ROT_FS(E_AI); % calcolo gli assi dell'assetto
At=eye(3)*ROT; % At in componenti ECI

At=At*ROT_IO(t_sns+t_man,i_sns); % Vettori assetto espressi in
                                % asse orbita istantanei, calcolato nella
                                % posizione in cui si troverà il satellite
                                % alla fine della manovra.

[Assetto] =EuAt2Ang (At); % Trova gli angoli per avere l'assetto
E_at=(180/pi)*Assetto;
Att_Sub_Rot; % Attua la Rotazione per allinearsi con l'assetto

```

```

fprintf ('\nIDLE per %4.1f secondi\n',t_man-T_m);
    % Attende di arrivare al punto in cui l'assetto è corretto per
    % iniziare la manovra di rotazione che porterà il satellite ad
    % essere sempre allineato con il riferimento ECI.

% Poichè nel passaggio tra ECI and ORBIT occorre tener conto della
% velocità angolare del satellite, deriviamo le matrici di
% rotazione, considerando che Rd=S(w)*R, dove R è la matrice di
% rotazione, Rd è la derivata della matrice di rotazione S(W) è
% la matrice di prodotto vettoriale della velocità angolare di R
% rispetto al frame fisso. S(W)=[0 -w1 +w2; +w1 0 -w3; -w2 +w3 0]
% nel nostro caso si ricava w2=w_or, w1=w3=0.
% per mantenere l'equilibrio imponiamo quindi una rotazione
% rispetto all'asse Yorbit.

Wo = [0, -w_or, 0];% Velocità angolare da applicare in assi orbita
Wb = Wo*ROT_OB(Assetto)*30/pi; % Vel angolare in assi body [RPM]
Wr = Wb*(Ixx/I_wheel);

fprintf('\n Vel. del satellite:| Vel. ruote pilotate:')
fprintf('\n W_x %+1.5f RPM      | Wr_x % 5.0f RPM',-Wb(1),Wr(1));
fprintf('\n W_y %+1.5f RPM      | Wr_y % 5.0f RPM',-Wb(2),Wr(2));
fprintf('\n W_z %+1.5f RPM      | Wr_z % 5.0f RPM',-Wb(3),Wr(3));
fprintf('\n [+] Rotazione Antioraria, [-] Rotazione Oraria\n')

```

A.5. Ground tracking sub system (Att_Sub_Tra.m)

SubRoutine di controllo d'assetto: TRACKING DELLA SUPERFICIE

```

%% Att_Sub_Tra.m
%
% Questa funzione una volta richiamata permette di calcolare i
% movimenti necessari a produrre una determinata e continua
% rotazione tale da mantenere sempre costante il puntamento
% rispetto a terra, ad esempio per garantire il puntamento di uno
% strumento, dati come parametri di ingresso la posizione del
% satellite e del punto fisso.
% Occorre inserire un valore negativo [-] per latitudine Nord e
% Longitudine Ovest, un valore positivo [+] per Sud ed Est
%
% Author: Danilo Muru, 2011
%


---


fprintf('\n Inserire latitudine e longitudine della stazione di
terra :\n');
LA_s = input(' LATITUDINE [°]: ');
LO_s = input(' LONGITUDINE [°]: ');

% Definizione costanti
R=6372; % Raggio terrestre
h=600; % Quota satellite
Rt=R+h; % Distanza totale dal centro della terra

```

```

w_er = 2*pi/24/60/60; % Velocità di rotazione della terra.
gamma0=acos(R/Rt); % Semiapertura con cui il satellite vede
                    % l'obiettivo
epsy=(5) /180*pi; % Altezza dall'orizzonte minima per vedere
                    % il satellite
ra = 15; %scostamento angolare massimo del punto dalla GT.
        % Definizione parametri

t_punt=ceil(2*(gamma0-epsy)/w_or); % Per quanto tempo nella sua
                                    % orbita il satellite riesce a
                                    % Vedere il punto.

% EARTH SPHERE (prepara il grafico con lo sfondo della terra)
for k=1:1
load('topo.mat','topo','topomap1');
contour(-179:180,-89:90,topo,[0 0],'b');
axis equal;
box on;
set(gca,'XLim',[-180 180],'YLim',[-90 90], ...
      'XTick',[-180 -120 -60 0 60 120 180], ...
      'Ytick',[-90 -60 -30 0 30 60 90]);
hold on;
image([-180 180],[-90 90],topo,'CDataMapping','scaled');
colormap(topomap1);
[x,y,z] = sphere(50);
cla reset;
axis square off;
props.AmbientStrength = 0.1;
props.DiffuseStrength = 1;
props.SpecularColorReflectance = .5;
props.SpecularExponent = 20;
props.SpecularStrength = 1;
props.FaceColor= 'texture';
props.EdgeColor = 'none';
props.FaceLighting = 'phong';
props.Cdata = topo;
surface(x,y,z,props);
light('position',[-1 0 1]);
light('position',[-1.5 0.5 -0.5], 'color', [.6 .2 .2]);
view(3), hold on;
end
% END EARTH SPHERE

% Print ground station position
Xs=sind(90+LA_s)*cosd(LO_s);
Ys=sind(90+LA_s)*sind(LO_s);
Zs=cosd(90+LA_s);

plot3(-1.1*Xs,-1.1*Ys,-1.1*Zs,'+r');

% Preallocazione vettori e costanti
Xa = zeros(17,1000); Ya = zeros(17,1000); Za = zeros(17,1000);
Xd1= zeros(17,1000); Yd1= zeros(17,1000); Zd1= zeros(17,1000);
Xd2= zeros(17,1000); Yd2= zeros(17,1000); Zd2= zeros(17,1000);
dista=zeros(17,1000);dista1=zeros(17,1000);dista2=zeros(17,1000);

```

```

% CALCOLO DELL'ORBITA PASSANTE SUL PUNTO DA TRACCIARE

%Preallocazione in memoria dei vettori e definizione dei flag.
K0=1; K1=1; K2=1;
ka=zeros(1,17); kb=zeros(1,17); kc=zeros(1,17);
sa=zeros(1,17); sb=zeros(1,17); sc=zeros(1,17);
Na_tr=[]; Nd1_tr=[]; Nd2_tr=[]; dist=0;
Na_tr=-5; Nd1_tr=-5; Nd2_tr=-5; dist=0;
LA=zeros(1000); LO=zeros(1000);
ta=zeros(1000); td1=zeros(1000); td2=zeros(1000);

for n=-4:12 %Confronta distanza orbita punto
    c=n+5;
% Definizione dei vettori tempo per velocità angolare
    wta = linspace(-1.57070,+1.57070,1000);
    wtd1 = linspace(+1.57090,+4.7123 ,1000);
    wtd2 = linspace(-4.7123 ,-1.57090 ,1000);

for n=-4:12 %Confronta distanza orbita punto
    c=n+5;
    wta = linspace(-1.57070,+1.57070,1000);
    wtd1 = linspace(+1.57090,+4.7123 ,1000);
    wtd2 = linspace(-4.7123 ,-1.57090 ,1000);
for i=1:1000
    %parte ascendente
    ta(i) = wta(i)/w_or;
    LA(i) = -asind(sind(i_or)*sin(wta(i)));
    LO(i) = atand(cosd(i_or)*tan(wta(i)))+
            +(w_er*ta(i)+n*w_er*t_or)*180/pi-N_or;
if sa(c)==0
    if LO(i)>180
        sa(c)=1;
        ka(c)=i;
    elseif LO(i)<-180
        ka(c)=-i;
    end
end
Xa(c,i) = sind(90-LA(i))*cosd(LO(i));
Ya(c,i) = sind(90-LA(i))*sind(LO(i));
Za(c,i) = cosd(90-LA(i));

    %parte discendente 1
    td1(i)= wtd1(i)/w_or;
    LA(i) = -asind(sind(i_or)*sin(wtd1(i)));
    LO(i) = +180+atand(cosd(i_or)*tan(wtd1(i)))+
            +(w_er*td1(i)+n*w_er*t_or)*180/pi-N_or;
if sb(c)==0
    if LO(i)>180
        sb(c)=1;
        kb(c)=i;
    elseif LO(i)<-180
        kb(c)=-i;
    end
end
Xd1(c,i)= sind(90-LA(i))*cosd(LO(i));
Yd1(c,i)= sind(90-LA(i))*sind(LO(i));
Zd1(c,i)= cosd(90-LA(i));

```

```

    %parte discendente 2
    td2(i)= wtd2(i)/w_or;
    LA(i) = -asind(sind(i_or)*sin(wtd2(i)));
    LO(i) = -180+atand(cosd(i_or)*tan(wtd2(i)))+
            +(w_er*td2(i)+n*w_er*t_or)*180/pi-N_or;

    if sc(c)==0
        if LO(i)>180
            sc(c)=1;
            kc(c)=i;
        elseif LO(i)<-180
            kc(c)=-i;
        end
    end

    Xd2(c,i)= sind(90-LA(i))*cosd(LO(i));
    Yd2(c,i)= sind(90-LA(i))*sind(LO(i));
    Zd2(c,i)= cosd(90-LA(i));
end %crea i vettori Xa e Xd per il confronto

dista(c,:)=sqrt(((Xa(c,:)-Xs).^2)+((Ya(c,:)-Ys).^2)+
                +((Za(c,:)-Zs).^2));

if ka(c)>0
    dista(c,ka(c):length(dista(c)))=
        =zeros(length(dista(c))-ka(c)+1);
elseif ka(c)<0 && ka(c)~-1000
    ka(c)=-ka(c);
    dista(c,1:ka(c))=zeros(1,ka(c));
end

    if min(dista(c,:))<=2*sind(ra/2)
        && min(dista(c,:))>=0.01
        Na_tr(K0)=n;
        K0=K0+1;
        k=1;
    end%controllo distanza ascendente

dstd1(c,:)=sqrt(((Xd1(c,:)-Xs).^2)+((Yd1(c,:)-Ys).^2)+
                +((Zd1(c,:)-Zs).^2));

if kb(c)>0
    dstd1(c,kb(c):length(dstd1(c)))=
        =zeros(length(dstd1(c))-kb(c)+1);
elseif kb(c)<0 && kb(c)~-1000
    kb(c)=-kb(c);
    dstd1(c,1:kb(c))=zeros(kb(c));
end

    if min(dstd1(c,:))<=2*sind(ra/2)
        && min(dstd1(c,:))~=0 && n<8
        Nd1_tr(K1)=n;
        K1=K1+1;
        k=1;
    end%controllo distanza discendente 1

dstd2(c,:)=sqrt(((Xd2(c,:)-Xs).^2)+((Yd2(c,:)-Ys).^2)+
                +((Zd2(c,:)-Zs).^2));

```

```

if kc(c)>0
dstd2(c, kc(c):length(dstd2(c)))=
    =zeros(length(dstd2(c))-kc(c))+1;
elseif kc(c)<0 && kc(c)~=-1000
kc(c)=-kc(c);
dstd2(c,1:kc(c))=zeros(1, kc(c));
end
    if min(dstd2(c,:))<=2*sind(ra/2)
        && min(dstd2(c,:))~=0 && n>0
        Nd2_tr(K2)=n;
        K2=K2+1;
        k=1;
    end %controllo distanza discendente 2

plot3(1.1*Xa(c,:),1.1*Ya(c,:),1.1*Za(c),'g');
hold on, grid on,
plot3(1.1*Xd1(c,:),1.1*Yd1(c,:),1.1*Zd1(c),'g');
hold on, grid on,
plot3(1.1*Xd2(c,:),1.1*Yd2(c,:),1.1*Zd2(c),'g');
hold on, grid on,
end

% Stampa l'orbita e il raggio di tracciamento del satellite
N_tr=[Na_tr,Nd1_tr,Nd2_tr];
figure
Print_Orbit(LA_s,LO_s,N_tr);
sca = scircle1(LA_s,LO_s, ra);

plot(LO_s,LA_s,'ro','MarkerFaceColor','r');
plot(sca(:,2), sca(:,1),'r','linewidth',2);

%Definizione e preallocazione vettori.
mu = zeros(1,t_punt); % Angolo tra la tangente all'orbita e
                    % la congiungente tra la proiezione del
                    % punto sulla ground track

D_mu = zeros(1,t_punt); % Derivata prima di mu (Vel angolare)
DD_mu = zeros(1,t_punt); % Derivata seconda (Acceleraz angolare)

Dw_mu = zeros(1,t_punt); % Velocità angolare alla ruota
DDw_mu= zeros(1,t_punt); % Accelerazione angolare alla ruota

delta = zeros(1,t_punt); % angolo tra la congiungente tra
                        % satellite e la proiezione del punto
                        % sull'orbita e tra satellite e punto
                        % da tracciare

D_delta = zeros(1,t_punt); % Derivata prima di delta(Vel angolare)
DD_delta= zeros(1,t_punt); % Derivata seconda (Acceleraz angolare)

Dw_delta =zeros(1,t_punt); % Velocità angolare alla ruota
DDw_delta=zeros(1,t_punt); % Accelerazione angolare alla ruota

gamma = zeros(1,t_punt); % angolo tra la verticale alla
                        % proiezione del punto dal centro della
                        % terra e tra la direzione satellite-
                        % centro della terra

```

```

t=linspace(0,t_punt-1,t_punt); % Tempo di simulazione.
LO_p=zeros(length(N_tr),1); dis_or=zeros(length(N_tr),1);

wt=asin(sind(LA_s)/sind(i_or));
t_is=wt/w_or;

for j=1:length(N_tr)

    c=N_tr(j)+5;

    if j>0 && j<=length(Na_tr)
        LO_p(j)=(atand(cosd(i_or)*tan(wt))+
            +N_tr(j)*w_er*t_or/pi*180-N_or);
        % Distanza minima GT-GS [°]
        dis_or(j)=min(dista(c,:))*sign(-LO_s+LO_p(j));

    elseif j>length(Na_tr) && j<=length([Na_tr Ndl_tr])
        LO_p(j)=+180+(atand(cosd(i_or)*tan(wt))
            +(N_tr(j)+0.5)*w_er*t_or/pi*180-N_or);
        % Distanza minima GT-GS [°]
        dis_or(j)=-min(distd1(c,:))*sign(-LO_s+LO_p(j));

    elseif j>length([Na_tr Ndl_tr]) && j<=length(N_tr)
        LO_p(j)=-180+(atand(cosd(i_or)*tan(wt))
            +N_tr(j)*w_er*t_or/pi*180-N_or);
        % Distanza minima GT-GS [°]
        dis_or(j)=min(distd2(c,:))*sign(-LO_s+LO_p(j));
    end

    for i=1:t_punt
        gamma(i)=(w_or*t(i)-(gamma0-epsy));
        Rt1=R*cos(gamma(i)); % Raggio interno
        Rt2=Rt-Rt1; % Raggio esterno
        d=R*sin(gamma(i)); % Distanza proiezione/raggio istant. orbita
        D=sqrt(d^2+Rt2^2); % Distanza sat-proiezione obiettivo su GT

        % DEFINIZIONE ANGOLI
        % * Definizione dell'angolo MU (rot Y)
        mu(i)=pi/2+atan(d/Rt2);
        % * Definizione dell'angolo DELTA (rot X)
        delta(i)=atand(dis_or(j)/Rt2*sin(mu(i)));

        if i~=1
            % VELOCITA' ANGOLARI
            % * Derivata prima di mu
            D_mu(i) = (mu(i)-mu(i-1))/(t(i)-t(i-1));
            % * Derivata prima di delta
            D_delta(i)= (delta(i)-delta(i-1))/(t(i)-t(i-1));

            % DATI RELATIVI ALLE RUOTE INERZIALI
            % * VELOCITA' DI ROTAZIONE ATTORNO A Y (Mu costante)
            Dw_mu(i) = -D_mu(i) *(Ixx/I_wheel)*60;
            % * VELOCITA' DI ROTAZIONE ATTORNO A X (Delta costante)
            Dw_delta(i)= -D_delta(i)*(Ixx/I_wheel)*60;
        end
    end
end

```

```

if i>2
% ACCELERAZIONI ANGOLARI
% * Derivata seconda di mu
DD_mu(i) = (D_mu(i)-D_mu(i-1))/(t(i)-t(i-1));
% * Derivata seconda di delta
DD_delta(i) = (D_delta(i)-D_delta(i-1))/(t(i)-t(i-1));

% ACCELERAZIONI ANGOLARI DELLE RUOTE
DDw_mu(i) = (Dw_mu(i) -Dw_mu(i-1) )/(t(i)-t(i-1));
DDw_delta(i) = (Dw_delta(i)-Dw_delta(i-1))/(t(i)-t(i-1));
end
end
% PLOTTAGGI

% * Plotta gli angoli in funzione del tempo
figure (5)
subplot (3,1,1), plot(t,alfa ), grid on ,hold on
subplot (3,1,1), title ('mu'),
xlabel('tempo (s)'),
ylabel ('\mu (rad)');
subplot (3,1,2), plot(t,D_alfa), grid on ,hold on
subplot (3,1,2), title ('D^1\mu'),
xlabel('tempo (s)'),
ylabel ('D^1\mu (rad/s)');
subplot (3,1,3), plot(t,DD_alfa ), grid on ,hold on
subplot (3,1,3), title ('D^2mu'),
xlabel('tempo (s)'),
ylabel ('D^2\mu (rad/s^2)');

figure (6)
subplot (3,1,1), plot(t,delta), grid on ,hold on
subplot (3,1,1), title ('delta'),
xlabel('tempo (s)'), ylabel ('\delta (rad)');
subplot (3,1,2), plot(t,D_delta), grid on ,hold on
subplot (3,1,2), title ('D^1delta'),
xlabel('tempo (s)'),
ylabel ('D^1\delta (rad/s)');
subplot (3,1,3), plot(t,DD_delta), grid on ,hold on
subplot (3,1,3), title ('D^2delta'),
xlabel('tempo (s)'),
ylabel ('D^2\delta (rad/s^2)');

% * Plotta le velocità angolari delle ruote in funzione del tempo
figure (7)
subplot (2,1,1), plot(t,Dw_alfa), grid on ,hold on
subplot (2,1,1), title ('velocita di puntamento'),
xlabel('tempo (s)'),
ylabel ('\omega\mu_w_h_e_e_l (RPM)')
subplot (2,1,2), plot(t,Dw_delta), grid on ,hold on
subplot (2,1,2), title ('velocita di deviazione'),
xlabel('tempo (s)'),
ylabel ('\omega\delta_w_h_e_e_l (RPM)')
end

```

APPENDICE B: SOUBROUTINE DI CARICAMENTO DATI:

B.1. Load Data (**loaddata.m**)

```
%% loaddata.m
%script per caricare i dati di orbita (orbit.dat) e assetto
%(attitude.dat) e assegnarli alle rispettive variabili.
%
% Author: Danilo Muru, 2011
%
load Dataorbit.dat % caricamento parametri orbita
                    % (vedi file per dettagli)
global h_or,h_or = Dataorbit(1); % quota
global t_or,t_or = Dataorbit(2); % periodo di rivoluzione
global w_or,w_or = 2*pi/t_or;    % velocità angolare del satellite
                                % lungo l'orbita
global i_or,i_or = Dataorbit(3); % inclinazione
e_or = Dataorbit(4); % eccentricità
P_or = Dataorbit(5); % argomento del perielio
global N_or, N_or = Dataorbit(6); % longitudine nodo ascendente
M_or = Dataorbit(7); % anomalia media
clear Dataorbit;    % elimina il vettore caricato poichè inutile

load Dataattitude.dat % caricamento assetto voluto
                       % (vedi file per dettagli)
Attitude_en (1) = Dataattitude(1); % verifica se il controllo IFix
è attivo
Attitude_en (2) = Dataattitude(2); % verifica se il controllo Rot
è attivo
Attitude_en (3) = Dataattitude(3); % verifica se il controllo Spi
è attivo
Attitude_en (4) = Dataattitude(4); % verifica se il controllo Tra
è attivo
Attitude_en; % Vettore attitude enable

PSI_at = Dataattitude(5)*pi/180; % Angolo di eulero psi
THE_at = Dataattitude(6)*pi/180; % Angolo di eulero theta
PHI_at = Dataattitude(7)*pi/180; % Angolo di eulero phi
global E at, E at = [PSI at, THE at, PHI at];
                    % Vettore angoli di eulero

Wa_Xb = Dataattitude(8); % Componente vel angolare X
Wa_Yb = Dataattitude(9); % Componente vel angolare Y
Wa_Zb = Dataattitude(10); % Componente vel angolare Z
global Wa_body, Wa_body = [Wa_Xb, Wa_Yb, Wa_Zb];
                          % Vettore velocità angolare

Simu_en = Dataattitude(11);
if Dataattitude(11)==1;
    fprintf ('\nSIMULATION MODE\n')
end
clear Dataattitude; % elimina il vettore caricato poichè inutile
```

```

load DataMechanics.dat

Ixx = DataMechanics(1); % Inerzia lungo l'asse x
Iyy = DataMechanics(2); % Inerzia lungo l'asse y
Izz = DataMechanics(3); % Inerzia lungo l'asse z
global I_mat, I_mat= [Ixx 0 0 ;
                    0 Iyy 0 ;
                    0 0 Izz] ;

% Matrice d'Inerzia

global I_wheel, I_wheel = DataMechanics(4);
% Momento d'inerzia della ruota
MaxRPMW = DataMechanics(5); % Velocità massima della ruota

clear DataMechanics; % elimina il vettore caricato in quanto
inutile

```

B.2. Load Sensor (loadsensor.m)

```

%% loadsensor.m
%
% Questo file fornisce all'algoritmo del controllo di assetto i
% dati relativi ai sensori installati e già elaborati, in modo da
% poter essere impiegati dal satellite per la regolazione autonoma
% dell'assetto.
%
%Author: Danilo Muru
%
global h_sns,
global i_sns,
global t_sns,
global i_or,

if Simu_en~=1 % Se non è una simulazione carica
    % direttamente i dati sensore
    set_sensor=3;
else
    set_sensor = input('\n Impostare manualmente le condizioni
iniziali del satellite? \n 0=Exit, 1=set, 2=random, 3=read from
sensor.dat; ');
end

if set_sensor == 0
elseif set_sensor == 1 % Dati del satellite impostati manualmente
    set_position = input(' Impostare la posizione? \n 1=set,
2=random; ');
    if set_position==1
        h_sns = input('Quota orbitale[km] : ');
        i_sns = input('Inclinazione orbitale[gradi]: ');
        lo_sns = 2*pi/360*input('Longitudine [-180°/+180°]: '); %rad
        la_sns = 2*pi/360*input('Latitudine [ -90°/+90° ]: '); %rad
        t_sns = t_or*lo_sns/360; % tempo dall'ultimo passaggio
        % per il nodo ascendente [s]

```

```

else
    %IMPOSTA DATI CASUALI
    %dati orbitali dei sensori
    h_sns = 600;           % Quota[km]
    i_sns = 86;           % Inclinazione orbitale[gradi]
    t_sns = t_or*rand;    % tempo dall'ultimo passaggio
                          % per il nodo ascendente [s]
    lo_sns = (w_or*t_sns); % Longitudine (approssimata)
    if lo_sns > pi;       % Esprime la longitudine da
        lo_sns = lo_sns - 2*pi; % [0 , 2*pi] a [-pi , pi]
    end
    la_sns = atan(cos(lo_sns)*tan(i_or)); % Latitudine
end

set_attitude = input(' Impostare l assetto iniziale? \n 1=set,
2=random; ');
if set_attitude == 1
    PSI_sns = input(' Psi Rot attorno a Z_b [°]: ')*pi/180;
    THE_sns = input(' Theta Rot attorno a X_b [°]: ')*pi/180;
    PHI_sns = input(' Phi Rot attorno a Z_b [°]: ')*pi/180;
    E_sns = [PSI_sns, THE_sns, PHI_sns]; % angoli di eulero
else
    %IMPOSTA DATI CASUALI
    %Sngoli di eulero del sistema di riferimento orbitale.
    PSI_sns = 2*rand*pi; %Psi Rot attorno a Z_b - PSI_at [°]
    THE_sns = rand*pi; %Theta Rot attorno a X_b - THE_at [°]
    PHI_sns = 2*rand*pi; %Phi Rot attorno a Z_b - PHI_at [°]
    E_sns = [PSI_sns, THE_sns, PHI_sns]; % angoli di eulero
end
set_spin = input(' Impostare il vettore velocità angolare
iniziale? \n 0=No Spin, 1=Set, 2=Random; ');
if set_spin == 1
    % vettore velocità angolare rispetto agli assi corpo
    Ws_Xb = input('Componente W_Xb [RPM]: ');
    Ws_Yb = input('Componente W_Yb [RPM]: ');
    Ws_Zb = input('Componente W_Zb [RPM]: ');
    Ws_body = [Ws_Xb, Ws_Yb, Ws_Zb];
elseif set_spin == 0
    %IMPOSTA SPIN NULLO
    % vettore velocità angolare rispetto agli assi corpo
    Ws_Xb = 0;
    Ws_Yb = 0;
    Ws_Zb = 0;
    Ws_body = [Ws_Xb, Ws_Yb, Ws_Zb];
end
elseif set_sensor == 2
    %IMPOSTA DATI CASUALI
    %dati orbitali dei sensori
    h_sns = 600;           % Quota[km]
    i_sns = 86;           % Inclinazione orbitale[gradi]
    t_sns = t_or*rand;    % tempo dall'ultimo passaggio dal nodo
                          % ascendente [s]
    lo_sns = (w_or*t_sns); % Longitudine (approssimata)
    if lo_sns > pi;
        lo_sns = lo_sns - 2*pi;
    end
    la_sns = atan(cos(lo_sns)*tan(i_or)); % Latitudine

```

```

%Angoli di eulero relativi al sistema di riferimento orbitale.
PSI_sns = 2*pi*rand; %Psi Rotazione attorno a Z_b - PSI_at [°]
THE_sns = pi*rand; %Theta Rotazione attorno a X_b - THE_at [°]
PHI_sns = 2*pi*rand; %Phi Rotazione attorno a Z_b - PHI_at [°]
E_sns = [PSI_sns, THE_sns, PHI_sns]; % angoli di eulero

% vettore velocità angolare rispetto agli assi corpo
Ws_Xb = -1.5+3*rand;
Ws_Yb = -1.5+3*rand;
Ws_Zb = -1.5+3*rand;
Ws_body = [Ws_Xb, Ws_Yb, Ws_Zb];

elseif set_sensor == 3
load Datasensor.dat %IMPORTA DATI SATELLITE
%dati orbitali dei sensori
h_sns = Datasensor(1); % Quota[km]
i_sns = Datasensor(2); % Inclinazione orbitale[gradi]
lo_sns = Datasensor(3); % Longitudine
la_sns = Datasensor(4); % Latitudine
t_sns = Datasensor(5); % tempo dall'ultimo passaggio per il
% nodo ascendente [s]

%Angoli di eulero relativi al sistema di riferimento orbitale.
PSI_sns = Datasensor(6)*pi/180; %Psi Rot su Z_b - PSI_at [°]
THE_sns = Datasensor(7)*pi/180; %Theta Rot su X_b - THE_at [°]
PHI_sns = Datasensor(8)*pi/180; %Phi Rot su Z_b - PHI_at [°]
E_sns = [PSI_sns, THE_sns, PHI_sns]; % angoli di eulero

% vettore velocità angolare rispetto agli assi corpo
Ws_Xb = Datasensor(9);
Ws_Yb = Datasensor(10);
Ws_Zb = Datasensor(11);
Ws_body = [Ws_Xb, Ws_Yb, Ws_Zb];

elseif
(set_sensor~=0)&&(set_sensor~=1)&&(set_sensor~=2)&&(set_sensor~=3)
LoadSensor;
end

lad_sns = la_sns/pi*180;
lod_sns = lo_sns/pi*180;

clear set_sensor %Pulisce la memoria dai vettori inutili
clear set_position %Pulisce la memoria dai vettori inutili
clear set_attitude %Pulisce la memoria dai vettori inutili
clear set_spin %Pulisce la memoria dai vettori inutili
clear set_magfield %Pulisce la memoria dai vettori inutili
clear Datasensor %Pulisce la memoria dai vettori inutili

```

APPENDICE C: FUNCTIONS DI MATLAB

C.1. Rotazione ECI-ORBIT (ROT_IO)

```
function [R_ItoO, R_OtoI] = ROT_IO(t_sns, i_or)
% ROTATION INERTIAL TO ORBIT
%
% [R_ItoO, R_OtoI] = ROT_IO(t_sns, i_or)
%
% Fornisce in uscita le matrici di rotazione:
% R_ItoO per convertire un vettore o una base di vettori dal
% riferimento inerziale ECI al riferimento orbitale.
% R_OtoI per convertire un vettore o una base di vettori dal
% riferimento orbitale al riferimento inerziale ECI.
% I parametri necessari in ingresso sono t_sns, ovvero il
% periodo trascorso dall'ultimo passaggio dal nodo ascendente.
% i_or, inclinazione dell'orbita dal piano di riferimento XYeci.
%
% Author: Danilo Muru, 2011
%


---


if t_sns==-1;
    t_sns = 90*60*rand;
end
global t_or;          % periodo ri rivoluzione del satellite
w_or = 2*pi/t_or;    % velocità angolare del satellite sull'orbita
b_0 = pi/2;         % angolo tra asse X_eci e asse X_orbit sull'asse X
global N_or; N_or=N_or;
beta = -(w_or*t_sns+N_or+b_0) ; % angolo tra asse X_eci e asse
                                % X_orbit istantaneo

%Rotazione rispetto a x, da ECI a Orbit
Rx_ItoO = [1      0      0      ;
           0 cos(-pi/2) -sin(-pi/2) ;
           0 sin(-pi/2)  cos(-pi/2)];

%Rotazione rispetto a z, da ECI a Orbit
Rz_ItoO = [ cos(i_or) sin(i_or) 0 ;
           -sin(i_or) cos(i_or) 0 ;
           0      0      1];

%Rotazione rispetto a y, da ECI a Orbit
Ry_ItoO = [ cos(beta) 0 sin(beta) ;
           0      1      0      ;
           -sin(beta) 0 cos(beta)];

%Matrice di rotazione da ECI a Orbit
R_ItoO = Rx_ItoO*Rz_ItoO*Ry_ItoO;
%Matrice di rotazione da Orbit a ECI
R_OtoI = R_ItoO';

return
```

C.2. Rotazione ECEF-ECI (ROT_EI)

```
function [R_EtoI, R_ItoE] = ROT_EI(T_e)
% ROTATION ECEF to ECI
%
% [R_EtoI, R_ItoE] = ROT_EI(T_e)
%
% Fornisce in uscita le matrici di rotazione:
% R_EtoI per convertire un vettore o una base di vettori dal
% riferimento rotante ECEF al riferimento inerziale ECI.
% R_ItoE per convertire un vettore o una base di vettori dal
% riferimento inerziale ECI al riferimento rotante ECEF.
% I parametri necessari in ingresso sono T_e, ovvero il periodo
% trascorso dall'ultimo allineamento tra ECI e ECEF.
%
% Author: Danilo Muru, 2011
%

if T_e==-1;
    T_e = 24*60*60*rand;
end

w_e = 2*pi/(24*60*60); % Velocità angolare della terra
Ae = w_e*T_e; % Angolo tra i due sistemi di riferimento

%Matrice di rotazione da ECEF a ECI
R_EtoI = [ cos(Ae) sin(Ae) 0 ;
          -sin(Ae) cos(Ae) 0 ;
           0 0 1];

%Matrice di rotazione da ECI a ECEF
R_ItoE = R_EtoI';
end
```

C.3. Rotazione ORBIT-BODY (ROT_OB)

```
function [ R_OtoB , R_BtoO ] = ROT_OB(E_in)
% ROTATION ORBIT TO BODY
%
% [R_OtoB , R_BtoO] = ROT_OB(E_in)
%
% Fornisce in uscita le matrici di rotazione:
% R_OtoB per convertire un vettore o una base di vettori dal
% riferimento solidale all'orbita al riferimento in assi corpo.
% R_BtoO per convertire un vettore o una base di vettori dal
% riferimento in assi corpo al riferimento solidale all'orbita.
% I parametri necessari in ingresso sono E_in, Ovvero gli angoli
% di Eulero Psi, Theta e Phi individuati tra il sistema solidale
% di assi body e il sistema fisso orbitale.
%
% Author: Danilo Muru, 2011
%
```

```

PSI = E_in(1);
THE = E_in(2);
PHI = E_in(3);

%Matrice di rotazione da Orbit Frame a Body Frame.
R_OtoB =
[ cos(PHI)*cos(PSI)-cos(THE)*sin(PHI)*sin(PSI) % R_OtoB (1,1)
  cos(PHI)*sin(PSI)+cos(THE)*sin(PHI)*cos(PSI) % R_OtoB (1,2)
  sin(THE)*sin(PHI) ; % R_OtoB (1,3)
  -sin(PHI)*cos(PSI)-cos(THE)*cos(PHI)*sin(PSI) % R_OtoB (2,1)
  -sin(PHI)*sin(PSI)+cos(THE)*cos(PHI)*cos(PSI) % R_OtoB (2,2)
  sin(THE)*cos(PHI) ; % R_OtoB (2,3)
  sin(THE)*sin(PSI) % R_OtoB (3,1)
  -sin(THE)*cos(PSI) % R_OtoB (3,2)
  cos(THE) ] ; % R_OtoB (3,3)

%Matrice di rotazione da Body Frame a Orbit Frame.
R_BtoO = R_OtoB';
End

```

C.4. Derivazione angoli di Eulero (EuAng_Der)

```

function [PSId,THEd,PHId] = EuAng_Der (E_in, W_in)
% DERIVAZIONE ANGOLI DI EULERO
%
% [PSId,THEd,PHId] = EuAng_Der (E_in, W_in)
%
% Fornisce in uscita le derivate degli angoli di eulero PHId,
% THEd, PSId, e il vettore che li contiene.
% Per il calcolo in ingresso sono necessari come parametri:
% E_in, rappresenta il vettore con gli angoli di eulero.
% W_in, rappresenta il vettore velocità angolare del satellite
% all'istante considerato.
%
% Author: Danilo Muru, 2011
%
% assegnazione angoli dal vettore in ingresso
PSI = E_in(1);
THE = E_in(2);
PHI = E_in(3);

% Calcoo delle derivate
PSId = ( W_in(1)*sin(PHI)+W_in(2)*cos(PHI))/sin(THE);
THEd = W_in(1)*cos(PHI)-W_in(2)*sin(PHI);
PHId = (-W_in(1)*sin(PHI)-W_in(2)*cos(PHI))/tan(THE)+W_in(3);

% E_dot = [PSId,THEd,PHId];
return

```

C.5. Rotazione angoli di Eulero (EuAng_Rot)

```
function [E_BtoA] = EuAng_Rot (E_at, E_sns)
% ROTAZIONE ANGOLI DI EULERO
%
% [PSI,THE,PHI] = EuAng_Rot (E_at, E_sns)
%
% Questo script serve per determinare gli angoli di Eulero tra
% gli assi corpo e gli assi corpo del nuovo assetto desiderato,
% avendo a disposizione solo l'assetto tra i due sistemi e un
% terzo di riferimento comune (tipicamente orbital frame). in
% questo modo è poi possibile sviluppare gli algoritmi per il
% controllo dell'assetto e per giungere a quello desiderato.
% Occorre aver definito i sei angoli di Eulero dai riferimenti
% all'orbital.
% Tali angoli sono inseriti nei vettori:
% E_sns: angoli tra assi corpo attuali e orbital frame.
% E_at : angoli tra assetto desiderato e orbital frame.
%
% Author: Danilo Muru, 2011
%

PSI_at = E_at(1) ;
THE_at = E_at(2) ;
PHI_at = E_at(3) ;
PSI_sns = E_sns(1);
THE_sns = E_sns(2);
PHI_sns = E_sns(3);

% 1 rotazione da B ad O, psi
L1= [ cos(PSI_at) -sin(PSI_at) 0 ;
      sin(PSI_at)  cos(PSI_at) 0 ;
      0            0          1] ;

% 2 rotazione da B ad O, teta
L2= [  cos(THE_at) 0 sin(THE_at) ;
      0            1 0 ;
      -sin(THE_at) 0 cos(THE_at)] ;

% 3 rotazione da B ad O, fi
L3= [ 1 0 0 ;
      0 cos(PHI_at) -sin(PHI_at) ;
      0 sin(PHI_at)  cos(PHI_at)] ;

% 1 rotazione da A ad O, psi
L4= [ cos(PSI_sns) -sin(PSI_sns) 0 ;
      sin(PSI_sns)  cos(PSI_sns) 0 ;
      0            0          1] ;

% 2 rotazione da A ad O, teta
L5= [  cos(THE_sns) 0 sin(THE_sns) ;
      0            1 0 ;
      -sin(THE_sns) 0 cos(THE_sns)] ;

% 3 rotazione da A ad O, phi
L6= [ 1 0 0 ;
      0 cos(PHI_sns) -sin(PHI_sns) ;
      0 sin(PHI_sns)  cos(PHI_sns)] ;
```

```

% Calcolo della matrice come prodotto delle due
% trasformazioni successive LBO e LAO, con L=LBO*LAO^T. espresso
% in termini delle singole matrici si ottiene la seguente
% espressione:
Ld_BtoA = L1 * L2 * L3 * L6' * L5' * L4';

% Calcolo i tre angoli di eulero confrontando la matrice a valori
% numerici con quella teorica.
THE_BtoA = -asin(Ld_BtoA(3,1));
PHI_BtoA = asin(Ld_BtoA(3,2)/cos(THE_BtoA));
PSI_BtoA = asin(Ld_BtoA(2,1)/cos(THE_BtoA));

Kc=0;
K =0;

% Il ciclo verifica che gli angoli trovati siano corretti. in caso
% contrario prova a correggerli fino a ottenere una perfetta
% sovrapposizione tra matrice reale ed ideale.
while Kc <= 4 && K<9 ;
    Kc=Kc+1;

% Selezione per sostituzione dell argomnto delle funzioni
% trigonometriche inverse
    if Kc==1
        PHI_BtoA = asin(Ld_BtoA(3,2)/cos(THE_BtoA));
        PSI_BtoA = asin(Ld_BtoA(2,1)/cos(THE_BtoA));
    elseif Kc==2;
        PHI_BtoA = pi-asin(Ld_BtoA(3,2)/cos(THE_BtoA));
        PSI_BtoA = asin(Ld_BtoA(2,1)/cos(THE_BtoA));
    elseif Kc==3;
        PHI_BtoA = asin(Ld_BtoA(3,2)/cos(THE_BtoA));
        PSI_BtoA = pi-asin(Ld_BtoA(2,1)/cos(THE_BtoA));
    elseif Kc==4;
        PHI_BtoA = pi-asin(Ld_BtoA(3,2)/cos(THE_BtoA));
        PSI_BtoA = pi-asin(Ld_BtoA(2,1)/cos(THE_BtoA));
    end;

% Matrice di rotazione da Attitude a Body Frame teorica.
% sosttuisco i valori trovati per ottenere un confronto.

L_BtoA =
[ ( cos(PSI_BtoA)*cos(THE_BtoA)) ,
  (sin(PHI_BtoA)*sin(THE_BtoA)*cos(PSI_BtoA) -
   +cos(PHI_BtoA)*sin(PSI_BtoA)),
  (cos(PSI_BtoA)*sin(THE_BtoA)*cos(PHI_BtoA) +
   +sin(PSI_BtoA)*sin(PHI_BtoA));
  ( cos(THE_BtoA)*sin(PSI_BtoA))
  (sin(PHI_BtoA)*sin(THE_BtoA)*sin(PSI_BtoA) +
   +cos(PHI_BtoA)*cos(PSI_BtoA)),
  (cos(PHI_BtoA)*sin(THE_BtoA)*sin(PSI_BtoA) -
   +sin(PHI_BtoA)*cos(PSI_BtoA));
  (-sin(THE_BtoA)) ,
  (sin(PHI_BtoA)*cos(THE_BtoA)) ,
  cos(PHI_BtoA)*cos(THE_BtoA) ];

```

```

%Controlla l'uguaglianza tra le due matrici
K=0;
for i = 1:3;
    for j = 1:3;
        a= abs(L_BtoA(i,j)-Ld_BtoA(i,j))<=0.0001;
        K= K+a;
    end
end
end

%In caso di successo stampa i valori degli angoli trovati.
if K==9 && Kc<20

    % In caso in cui la rotazione sia superiore ai 180° si
    % sostituisce con una rotazione opposta in modo da ridurre i
    % tempi necessari alla manovra.
    if PSI_BtoA>pi
        PSI_BtoA = PSI_BtoA-2*pi;
    end
    if THE_BtoA>pi
        THE_BtoA = THE_BtoA-2*pi;
    end
    if PHI_BtoA>pi
        PHI_BtoA = PHI_BtoA-2*pi;
    end

    fprintf('\nPSI_BtoA = %+3.4f°\nTHE_BtoA = %+3.4f°\nPHI_BtoA
= %+3.4f°\n',PSI_BtoA*180/pi,THE_BtoA*180/pi,PHI_BtoA*180/pi);

    E_BtoA=[PSI_BtoA,THE_BtoA,PHI_BtoA];
else
    fprintf('\nEULER ANGLES MISMATCH\n')
end

```

C.6. Angoli di Eulero da terna di assi orientata (EuAt2Ang)

```

function [E_BtoA] = EuAt2Ang (Axes)
% ANGOLI DI EULERO DA MATRICE DEGLI ASSI RISPETTO A UN RIFERIMENTO
% [0.0.0]
%
% [PSI,THE,PHI] = EuAng_Rot (E_at, E_sns)
%
% Questo script serve per determinare gli angoli di eulero di
% una terna di assi orientata rispetto ad un sistema di
% riferimento qualunque, con angoli di eulero pari a 0, 0, 0.
% Si ha in ingresso la matrice AXES 3x3 che contiene
% l'espressione degli assi desiderati in componenti del
% riferimento fisso.
%
% Author: Danilo Muru, 2011
%

```

```

PSI_at = 0 ;
THE_at = 0 ;
PHI_at = 0 ;

% 1 rotazione da B ad O, psi
L1= [ cos(PSI_at) -sin(PSI_at) 0 ;
      sin(PSI_at)  cos(PSI_at) 0 ;
      0            0          1] ;

% 2 rotazione da B ad O, teta
L2= [ cos(THE_at) 0 sin(THE_at) ;
      0          1 0 ;
      -sin(THE_at) 0 cos(THE_at)] ;

% 3 rotazione da B ad O, fi
L3= [ 1 0 0 ;
      0 cos(PHI_at) -sin(PHI_at) ;
      0 sin(PHI_at)  cos(PHI_at)] ;

% Calcolo della matrice come prodotto delle due trasformazioni
% successive LBO e LAO, con L=LBO*LAO^T. espresso in termini delle
% singole matrici si ottiene la seguente espressione:
Ld_BtoA = L1 * L2 * L3 * Axes';

% Calcolo i tre angoli di eulero confrontando la matrice a valori
% numerici con quella teorica.
THE_BtoA = -asin(Ld_BtoA(3,1));
PHI_BtoA = asin(Ld_BtoA(3,2)/cos(THE_BtoA));
PSI_BtoA = asin(Ld_BtoA(2,1)/cos(THE_BtoA));

Kc=0;
K =0;

% Il ciclo verifica che gli angoli trovati siano corretti. in caso
% contrario prova a correggerli fino a ottenere una perfetta
% sovrapposizione tra matrice reale ed ideale.
while Kc <= 4 && K<9 ;
    Kc=Kc+1;

% Selezione per sostituzione dell argomnto delle funzioni
% trigonometriche inverse
if Kc==1
    PHI_BtoA = asin(Ld_BtoA(3,2)/cos(THE_BtoA));
    PSI_BtoA = asin(Ld_BtoA(2,1)/cos(THE_BtoA));
elseif Kc==2;
    PHI_BtoA = pi-asin(Ld_BtoA(3,2)/cos(THE_BtoA));
    PSI_BtoA = asin(Ld_BtoA(2,1)/cos(THE_BtoA));
elseif Kc==3;
    PHI_BtoA = asin(Ld_BtoA(3,2)/cos(THE_BtoA));
    PSI_BtoA = pi-asin(Ld_BtoA(2,1)/cos(THE_BtoA));
elseif Kc==4;
    PHI_BtoA = pi-asin(Ld_BtoA(3,2)/cos(THE_BtoA));
    PSI_BtoA = pi-asin(Ld_BtoA(2,1)/cos(THE_BtoA));
end;

```

```

% Matrice di rotazione da Attitude a Body Frame teorica.
% sosttuisco i valori trovati per ottenere un confronto.

L_BtoA =
[ ( cos(PHI_BtoA)*cos(PSI_BtoA)
(sin(PHI_BtoA)*sin(PSI_BtoA)*cos(PSI_BtoA)-
+cos(PHI_BtoA)*sin(PSI_BtoA)),
(cos(PHI_BtoA)*sin(PSI_BtoA)*cos(PSI_BtoA)+
+sin(PHI_BtoA)*sin(PSI_BtoA));
( cos(PSI_BtoA)*sin(PSI_BtoA)
(sin(PHI_BtoA)*sin(PSI_BtoA)*sin(PSI_BtoA)+
+cos(PHI_BtoA)*cos(PSI_BtoA)),
(cos(PHI_BtoA)*sin(PSI_BtoA)*sin(PSI_BtoA)-
+sin(PHI_BtoA)*cos(PSI_BtoA));
(-sin(PSI_BtoA))
(sin(PHI_BtoA)*cos(PSI_BtoA))
cos(PHI_BtoA)*cos(PSI_BtoA) ];

%Controlla l'uguaglianza tra le due matrici
K=0;
for i = 1:3;
    for j = 1:3;
        a= abs(L_BtoA(i,j)-Ld_BtoA(i,j))<=0.0001;
        K= K+a;
    end
end
end

%In caso di successo stampa i valori degli angoli trovati.
if K==9 && Kc<20

    % In caso in cui la rotazione sia superiore ai 180° si
    % sostituisce con una rotazione opposta in modo da ridurre i
    % tempi necessari alla manovra.
    if PHI_BtoA>pi
        PHI_BtoA = PHI_BtoA-2*pi;
    end
    if THE_BtoA>pi
        THE_BtoA = THE_BtoA-2*pi;
    end
    if PSI_BtoA>pi
        PSI_BtoA = PHI_BtoA-2*pi;
    End

    fprintf('\nPSI_BtoA = %+3.4f°\nTHE_BtoA = %+3.4f°\nPHI_BtoA
= %+3.4f°\n',PSI_BtoA*180/pi,THE_BtoA*180/pi,PHI_BtoA*180/pi);

    E_BtoA=[PSI_BtoA,THE_BtoA,PHI_BtoA];
else
    fprintf('\nEULER ANGLES MISMATCH\n')
end

```

C.7. Grafici dell'orbita (Print_Orbit)

```
function Print_Orbit (LAp,LOp,N)

%% Print Orbit (LAp,LOp,N)
%
% Stampa il reticolo di Ground Track del satellite in funzione
% dell'orbita percorsa.
% Se forniti i dati LAp e LOp viene stampata anche la posizione
% istantanea del satellite sulla ground track. Fornendo viene
% colorata in giallo la o le determinate orbite (N=-4:16)
%
% Author: Danilo Muru
%

earth = imread('earth.jpg');
lv= size(earth,1);
lh= size(earth,2);

lats = (1:lv)*180/lv - 90;
lons = (1:lh)*360/lh - 180;
image(lons, -lats, earth(1024:-1:1,:::)), grid on, hold on;
title('Ground Track');
xlabel('Longitude (degrees)');
ylabel('Latitude (degrees)');

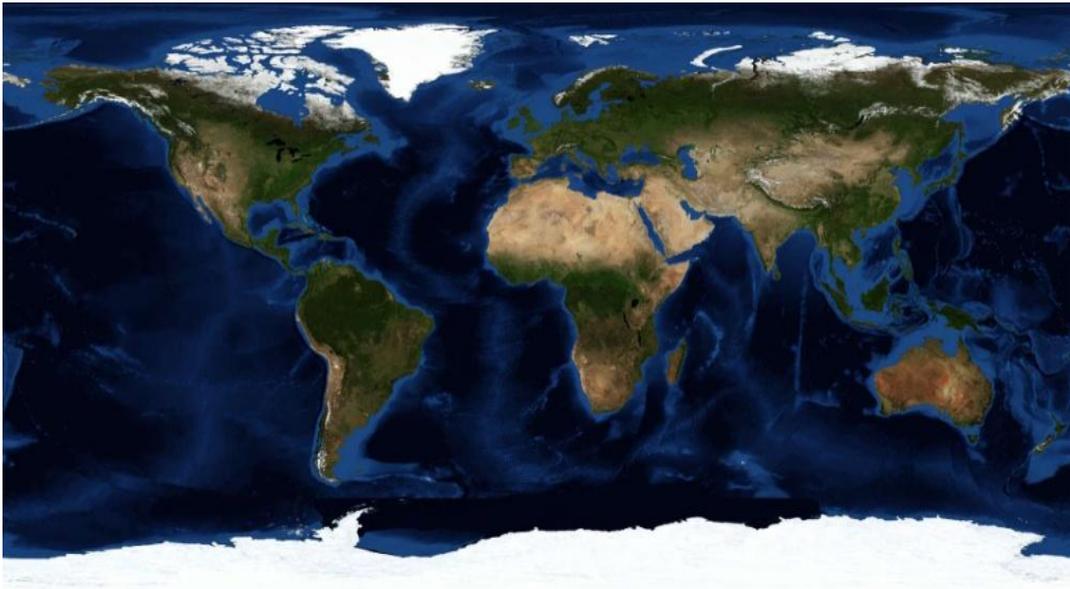
global N_or
global i_or
t_or =90*60;
w = 2*pi/t_or;
t_e = 24*60*60;
W = 2*pi/t_e;
in=i_or*pi/180;  %(84)*pi/180;

for n=-4:12 %Disegna le orbite
    %parte ascendente
    for j=-1:1
        wt = linspace(-1.57070+j*2*pi,+1.57070+j*2*pi,1000);
        t=[wt]/w;
        LA= asin(sin(in)*sin(wt))*180/pi;
        LO= j*360+(atan(cos(in)*tan(wt))+W*t+n*W*t_or)*180/pi -N_or;
        for k=1:length(N)
            if n==N(k)
                if n>8 && j>0
                    plot(LO,LA,'g'),hold on, grid on
                elseif n<0 && j<0
                    plot(LO,LA,'g'),hold on, grid on
                else
                    plot(LO,LA,'y','linewidth',3 ),hold on, grid on
                end
            else
                plot(LO,LA,'g'),hold on, grid on
            end
        end
    end %PLOT
end
```

```

%parte discendente
for j=-1:2:1
    wt = linspace(j*4.7123,j*1.57090,10000);
    t=[wt]/w;
    LA= asin(sin(in)*sin(wt))*180/pi;
    LO= j*180+(atan(cos(in)*tan(wt))+W*t+n*W*t_or)*180/pi -N_or;
    for k=1:length(N)
        if n==N (k)
            if n>8 && j>0
                plot(LO,LA,'g'),hold on, grid on
            elseif n<0 && j<0
                plot(LO,LA,'g'),hold on, grid on
            else
                plot(LO,LA,'y','linewidth',3),hold on, grid on
            end
        else
            plot(LO,LA,'g'),hold on, grid on
        end
    end %PLOT
end
end
end
plot (LOp,LAp,'dr','linewidth', 2),hold on,
end

```



8.1 *Earth.jpg su cui vengono stampate le orbite.*

APPENDICE D: FILE .DAT

D.1. DataAttitude.dat

```
%%Dati d'assetto del satellite
%
% Scelta del tipo di assetto da ottenere. Può essere attivo solo
% un comando per volta, Nel caso fossero necessari più comandi si
% reitera la procedura.
%
% Author: Danilo Muru
%
-----
0      %ECI    Fixed attitude, if 1 is enabled - IFix_en
0      %Rotation,          if 1 is enabled - Rot_en
0      %Spin/De-spin,      if 1 is enabled - Spi_en
0      %Tracking,          if 1 is enabled - Tra_en

%Euler attitude Angles for ZZX transformation.
0      %Psi,              Rotazione attorno a Z_b - PSI_at [°]
0      %Theta,            Rotazione attorno a X_b - THE_at [°]
0      %Phi,              Rotazione attorno a Z_b - PHI_at [°]

%componenti del vettore velocità angolare rispetto agli assi corpo
00     %Wa Xb
00     %Wa_Yb
00     %Wa_Zb

%Imposta una variabile per attivare la simulazione
1      %Simulation Mode, if 1 is enable
```

D.2. DataMechanics.dat

```
%%Dati meccanici del satellite
%
% Questo file fornisce all'algorithmo del controllo di assetto i
% dati strutturali del satellite, in modo da poter essere
% impiegati dal per la regolazione autonoma dell'assetto.
%
% Author: Danilo Muru
%
-----
%componenti della matrice d'inerzia:
0.022      %Ixx [kg*m^2]
0.022      %Iyy [kg*m^2]
0.022      %Izz [kg*m^2]

%Dati della ruota d'inerzia (uguali per tutti gli assi)
4E-6      %I_wheel [kg*m^2]
10000     %MaxRPMW [RPM]
```

D.3. DataOrbit.dat

```
%%Dati Orbitali
%
% Questo file fornisce all' algoritmo del controllo di assetto i
% dati orbitali ideali del satellite. viene considerata un orbita
% fissa e costante nel tempo.
%
%Author: Danilo Muru
%
-----
600      % Quota h_or[km]
5400     % Periodo T_or [secondi]
86       % Inclinazione orbitale i_or [gradi]
0        % Eccentricità e_or
90       % Argomento del perielio W_or [gradi]
90       % Longitudine nodo ascendente N_or [gradi]
0        % Anomalia media M_or
```

D.4. DataSensor.dat

```
%%Dati dei sensori
%
% Questo file fornisce all' algoritmo del controllo di assetto i
% dati ricevuti dai sensori (quando presenti).
% In ambiente simulativo questo file non viene caricato poichè non
% esistono feedback dei sensori. tutti i dati vengono impostati a
% mano o generati casualmente
%
%Author: Danilo Muru
%
-----
%dati orbitali dei sensori
0        % Quota[km]
0        % Inclinazione orbitale[gradi]
0        % Longitudine (random)
0        % Latitudine (PROB)
0        % tempo dall'ultimo passaggio per il nodo ascendente [s]

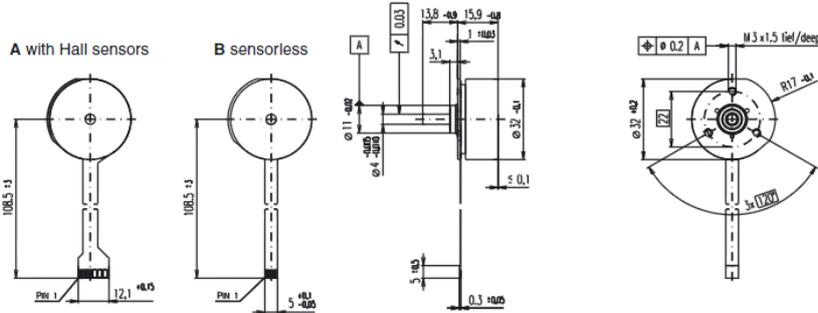
%Angoli di eulero relativi al sistema di riferimento orbitale.
0        %Psi,    Rotazione attorno a Z_b - PSI_at [°]
0        %Theta,  Rotazione attorno a X_b - THE_at [°]
0        %Phi,    Rotazione attorno a Z_b - PHI_at [°]

%componenti del vettore velocità angolare rispetto agli assi corpo
0        %Ws_Xb
0        %Ws_Yb
0        %Ws_Zb
```

APPENDICE E: ALTRO

E.1. Maxton EC 32 Flat motor Datasheet

EC 32 Flat motor $\varnothing 32$ mm, brushless, 15 Watt



M 1:2

- Stock program
- Standard program
- Special program (on request!)

Order Number

with Hall sensors
sensorless

267121

226006

Motor Data

		267121	226006
1 Assigned power rating	W	15	15
2 Nominal voltage	Volt	24.0	24.0
3 No load speed	rpm	4500	4500
4 Stall torque	mNm	85.8	85.8
5 Speed / torque gradient	rpm / mNm	54.0	54.0
6 No load current	mA	70	70
7 Terminal resistance phase to phase	Ohm	13.7	13.7
8 Max. permissible speed	rpm	10000	10000
9 Max. continuous current at 5000 rpm	A	0.56	0.56
10 Max. continuous torque at 5000 rpm	mNm	25.1	25.1
11 Max. efficiency	%	64.0	64.0
12 Torque constant	mNm / A	49.0	49.0
13 Speed constant	rpm / V	195	195
14 Mechanical time constant	ms	20.0	20.0
15 Rotor inertia	gcm ²	35.0	35.0
16 Terminal inductance phase to phase	mH	5.640	5.640
17 Thermal resistance housing-ambient	K / W	9.3	9.3
18 Thermal resistance winding-housing	K / W	5.2	5.2
19 Thermal time constant windings	s	7.7	7.7
20 Thermal time constant stator	s	78.7	78.7

Specifications

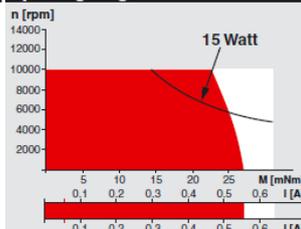
- Axial preload > 5 N
 - Max. ball bearing loads
 - axial (dynamic) 2.8 N
 - radial (7.5 mm from flange) 5.5 N
 - Force for press fits (static) 50 N (static, shaft supported) 1000 N
 - Ambient temperature range -40 ... +100°C
 - Max. permissible winding temperature +125°C
 - Weight of motor 58 g
 - Version with and without Hall sensors
 - 8 pole permanent magnet
 - 3 phased coil stator with 2 pole shoes each
 - Values listed in the table are nominal.
 - Connection with Hall sensors sensorless

Pin 1	3.5 ... 20 VDC	Motor winding 1
Pin 2	Hall sensor 3	Motor winding 2
Pin 3	Hall sensor 1	Motor winding 3
Pin 4	Hall sensor 2	neutral point
Pin 5	GND	
Pin 6	Motor winding 3	
Pin 7	Motor winding 2	
Pin 8	Motor winding 1	
 - Adapter Order number Order number

see p. 275	220300	220310
------------	--------	--------
 - Connector Article number Article number

AMP	1-487951-1	487951-4
MOLEX	52207-1190	52207-0490
MOLEX	52089-1110	52089-0410
- Pin for design with Hall sensors:
FPC, 11 pole, pitch 1.0 mm, top contact style
- For wiring diagram for Hall sensors, see p. 26

Operating Range



Comments

Details on page 149

- Curve of constant assigned power rating
- Continuous operation
 - In observation of above listed thermal resistances (lines 17 and 18) the maximum permissible winding temperature will be reached during continuous operation at 25°C ambient.
 - = Thermal limit.
- Short term operation
 - The motor may be briefly overloaded (recurring).

Recommended Electronics:

AEOS 35/3	page 262
DEC 24/1	263
DEC 50/5	264
DECV 50/5	265
EPOS 24/1	271
Notes	17

April 2005 edition / subject to change

maxon EC motor 185

9. BIBLIOGRAFIA

- [1] Ciani, M. (2003). *Studio del sistema di assetto del satellite atmocube Tramite attuatori magnetici.*
- [2] Gregorio, A. *Dispense: iv Geometria di una missione Spaziale.* Trieste.
- [3] Gregorio, A. *Dispense: vi Sistemi d'assetto.* Trieste.
- [4] L.Reyneri, M. (2011). Mechanical subsystem of the AraMiS architecture.
- [5] Lovera, M. *Introduzione al controllo d'assetto di piccoli satelliti.*
- [6] Marco Borri, A. F. (2006). *Appunti di meccanica aerospaziale.* Milano.
- [7] Masera, D. (2011). *Realizzazione del modello ingegneristico di un controllo di assetto magnetico per satelliti modulari.*
- [8] Molteni, M. (2007). *Controllo di assetto nello spazio.*
- [9] Øverby, E. J. (2004). *Attitude control for the Norwegian student satellite nCube.*
- [10] Scozzafava, C. (2008, agosto 7). *Euler Angles and Frame Angular Velocity.* Retrieved from www.southerlies.eu.
- [11] Svartveit, K. (2003). *Attitude determination of the NCUBE satellite.*
- [12] *Three Axis Reaction Wheel Attitude Control System for KitSat-3 microsatellite.*
- [13] Vico, F. (2008). *Sviluppo e collaudo di un sottosistema di controllo d'assetto per satellite modulare AraMiS.*
- [14] Zazzera, P. F. *Appunti del corso di meccanica del volo spaziale.* Milano.
- [15] Quagliotti, F. *Appunti del corso di meccanica del volo.* Torino.
- [16] Wikipedia: <http://it.wikipedia.org>

