POLITECNICO MILANO 1863

Introduzione all’Analisi di Missioni Spaziali

Laboratorio 1

Gianmario Merisio gianmario.merisio@polimi.it
Niccolò Faraco niccolo.faraco@polimi.it

Contenuti

- Descrizione **elaborato finale**

- Laboratorio 1:

  - Trasformazione **da cartesiane a parametriche** (*par2car*):

$$[\boldsymbol{r}, \boldsymbol{v}] \rightarrow [a, e, i, \Omega, \omega, \theta]$$

  - Trasformazione **da parametriche a cartesiane** (*car2par*):

$$[a, e, i, \Omega, \omega, \theta] \rightarrow [\boldsymbol{r}, \boldsymbol{v}]$$

  - **Rappresentazione** dell’orbita (*plotOrbit*)

- **MATLAB** tips

- **Playgroud**

FAC-SIMILE

Punto su **orbita iniziale**

Punto iniziale (km, km/s)					
r_x	r_y	r_z	v_x	v_y	v_z
-1,14414030E+04	-7,20985180E+03	-1,30298510E+03	1,2140E+00	-1,7110E+00	-4,7160E+00

Trasferimento orbitale

Punto su **orbita finale**

Punto finale (km, rad)					
a	e	i	Ω	ω	θ
2,9930E+04	1,5160E-01	3,0250E+00	6,5460E-01	2,7820E+00	2,6190E+00

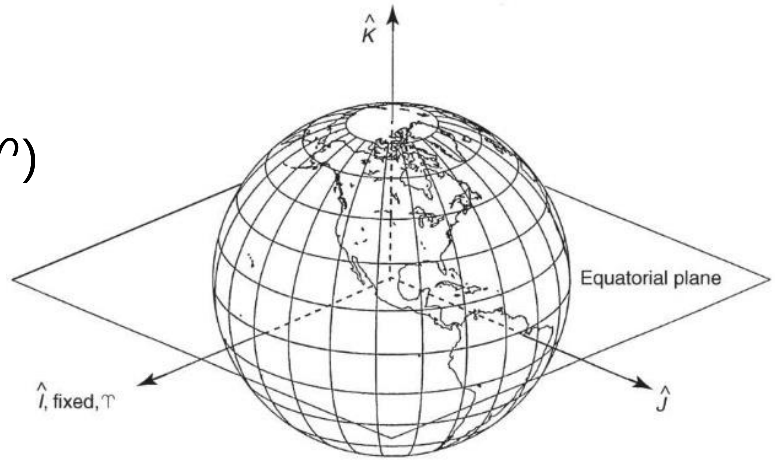
Obiettivo:
Progettare il trasferimento orbitale che permette al nostro satellite di spostarsi dal punto iniziale al punto finale.

Obiettivo:

Dato un vettore di stato $[\boldsymbol{r}, \boldsymbol{v}]$ in coordinate geocentriche inerziali (ECI), trovare i parametri kepleriani classici $[a, e, i, \Omega, \omega, \theta]$ associati

Sistema geocentrico inerziale *(ECI)*

- Sistema **inerziale**
- **Origine**: centro della **Terra**
- **Assi:**
  - $\hat{\imath}$ verso Equinozio di Primavera (punto ♈)
  - $\hat{k}$ verso Polo Nord
  - $\hat{\jmath}$ completa la terna



Algoritmo *car2par* (1)

INPUT: $[\boldsymbol{r}, \boldsymbol{v}], (\mu)$

1. Moduli di posizione e velocità

$$r = \|\boldsymbol{r}\|$$
$$v = \|\boldsymbol{v}\|$$

2. Energia meccanica specifica e **semiasse maggiore**

$$\varepsilon = \frac{1}{2}v^2 - \frac{\mu}{r} = -\frac{\mu}{2a} \quad \rightarrow \quad a = \left(\frac{2}{r} - \frac{v^2}{\mu}\right)^{-1}$$

3. Vettore momento angolare specifico

$$\boldsymbol{h} = \boldsymbol{r} \times \boldsymbol{v}$$
$$h = \|\boldsymbol{h}\|$$

NOTA: per i progetti $\mu = 398600\ \mathrm{km^3/s^2}$

4. Vettore eccentricità ed **eccentricità**

$$\boldsymbol{e} = \frac{\boldsymbol{v} \times \boldsymbol{h}}{\mu} - \frac{\boldsymbol{r}}{r}$$

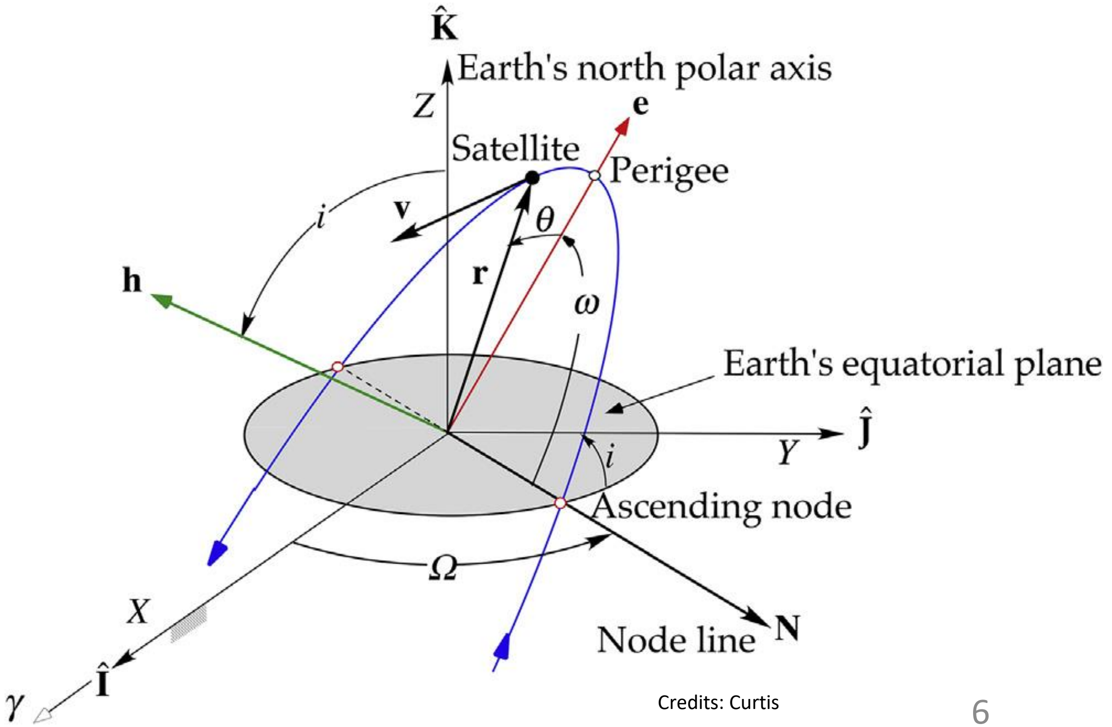
$$e = \|\boldsymbol{e}\|$$

5. **Inclinazione**

$$i = \text{acos}\left(\frac{\boldsymbol{h} \cdot \widehat{\boldsymbol{k}}}{h}\right) \xrightarrow{\text{In MATLAB}} \boldsymbol{h} \cdot \widehat{\boldsymbol{k}} = h(3)$$

6. Linea dei nodi

$$\boldsymbol{N} = \frac{\widehat{\boldsymbol{k}} \times \boldsymbol{h}}{\left\|\widehat{\boldsymbol{k}} \times \boldsymbol{h}\right\|}$$



Credits: Curtis

7. **Ascensione retta del nodo ascendente (RAAN)**

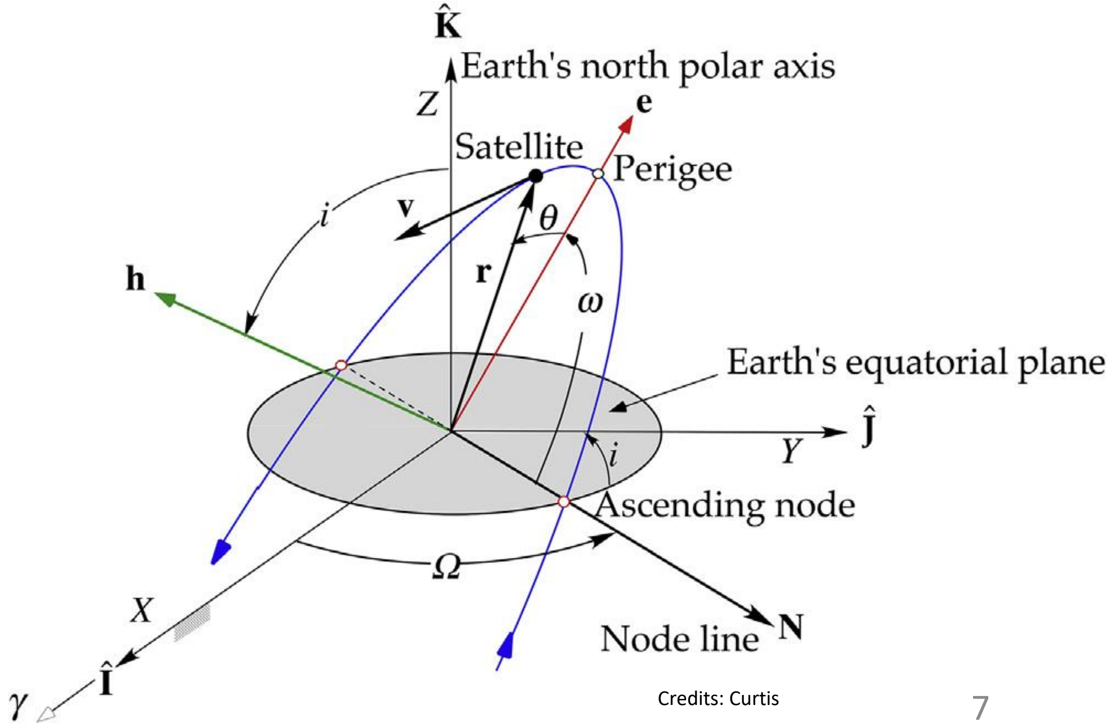
$$\Omega = \begin{cases} \mathrm{acos}(\boldsymbol{N} \cdot \hat{\boldsymbol{\imath}}) & \boldsymbol{N} \cdot \hat{\boldsymbol{\jmath}} \geq \boldsymbol{0} \\ 2\pi - \mathrm{acos}(\boldsymbol{N} \cdot \hat{\boldsymbol{\imath}}) & \boldsymbol{N} \cdot \hat{\boldsymbol{\jmath}} < \boldsymbol{0} \end{cases}$$

$N(1)$ $N(2)$

8. **Anomalia del pericentro**

$$\omega = \begin{cases} \mathrm{acos}\left(\frac{\boldsymbol{N} \cdot \boldsymbol{e}}{e}\right) & \boldsymbol{e} \cdot \widehat{\boldsymbol{k}} \geq \boldsymbol{0} \\ 2\pi - \mathrm{acos}\left(\frac{\boldsymbol{N} \cdot \boldsymbol{e}}{e}\right) & \boldsymbol{e} \cdot \widehat{\boldsymbol{k}} < \boldsymbol{0} \end{cases}$$

$e(3)$



Credits: Curtis

9. Anomalia vera

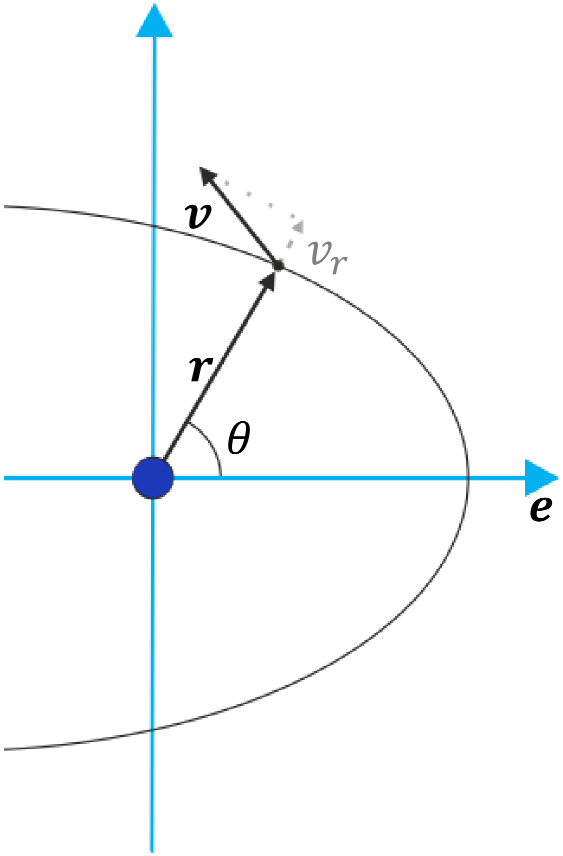
$$v_r = \frac{\boldsymbol{v} \cdot \boldsymbol{r}}{r}$$

Se $v_r > 0$, ci stiamo **allontanando** dal pericentro, quindi

$$\theta = \text{acos}\left(\frac{\boldsymbol{r} \cdot \boldsymbol{e}}{re}\right)$$

Se $v_r < 0$, ci stiamo **avvicinando** al pericentro, quindi

$$\theta = 2\pi - \text{acos}\left(\frac{\boldsymbol{r} \cdot \boldsymbol{e}}{re}\right)$$



OUTPUT: $[a, e, i, \Omega, \omega, \theta]$

car2par in MATLAB

```
function [a, e, i, OM, om, th] = car2par(rr, vv, mu)
```

```
% Trasformation from cartesian coordinates to Keplerian parameters
%
% [a, e, i, OM, om, th] = car2par(rr, vv, mu)
%
% -------------------------------------------------------------------------
% Input arguments:
% rr              [3x1]   position vector                         [km]
% vv              [3x1]   velocity vector                         [km/s]
% mu              [1x1]   gravitational parameter                 [km^3/s^2]
%
% -------------------------------------------------------------------------
% Output arguments:
% a               [1x1]   semi-major axis                         [km]
% e               [1x1]   eccentricity                            [-]
% i               [1x1]   inclination                             [rad]
% OM              [1x1]   RAAN                                    [rad]
% om              [1x1]   pericenter anomaly                      [rad]
% th              [1x1]   true anomaly                            [rad]
%
% -------------------------------------------------------------------------
```

Obiettivo:

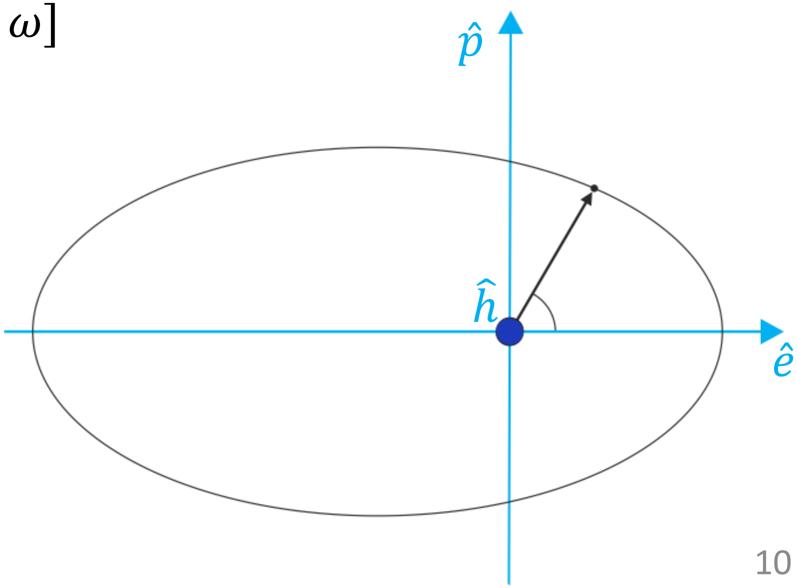
Dati i parametri kepleriani classici $[a, e, i, \Omega, \omega, \theta]$, trovare il vettore di stato $[\boldsymbol{r}, \boldsymbol{v}]$ in coordinate geocentriche inerziali (ECI) associato

Procedura

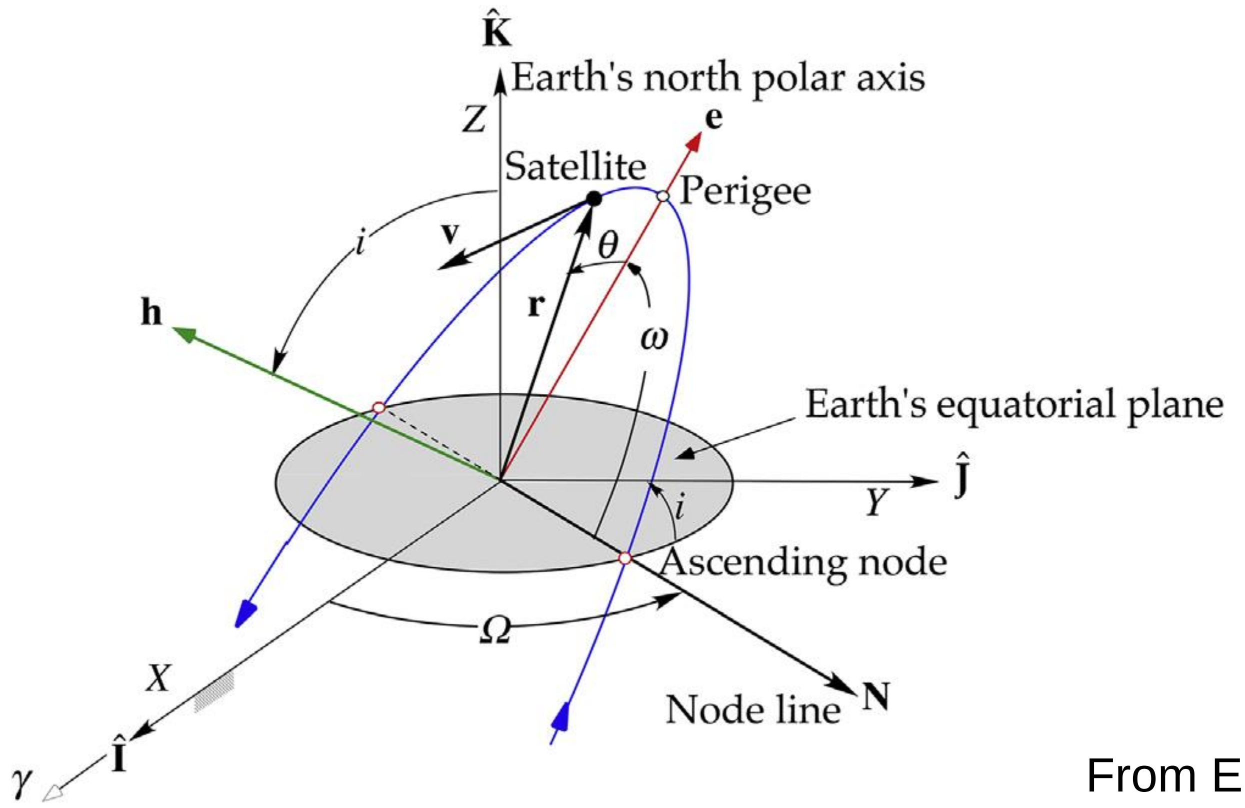
- Determinare il vettore di stato $[\tilde{\boldsymbol{r}}, \tilde{\boldsymbol{v}}]$ nel sistema di riferimento perifocale (PF) $[\hat{\boldsymbol{e}}, \hat{\boldsymbol{p}}, \hat{\boldsymbol{h}}]$ partendo dai parametri $[a, e, \theta]$
- Ruotare il vettore $[\tilde{\boldsymbol{r}}, \tilde{\boldsymbol{v}}]$ nel sistema di riferimento geocentrico inerziale (ECI) per ottenere $[\boldsymbol{r}, \boldsymbol{v}]$, utilizzando i parametri $[i, \Omega, \omega]$

Sistema perifocale (PF)

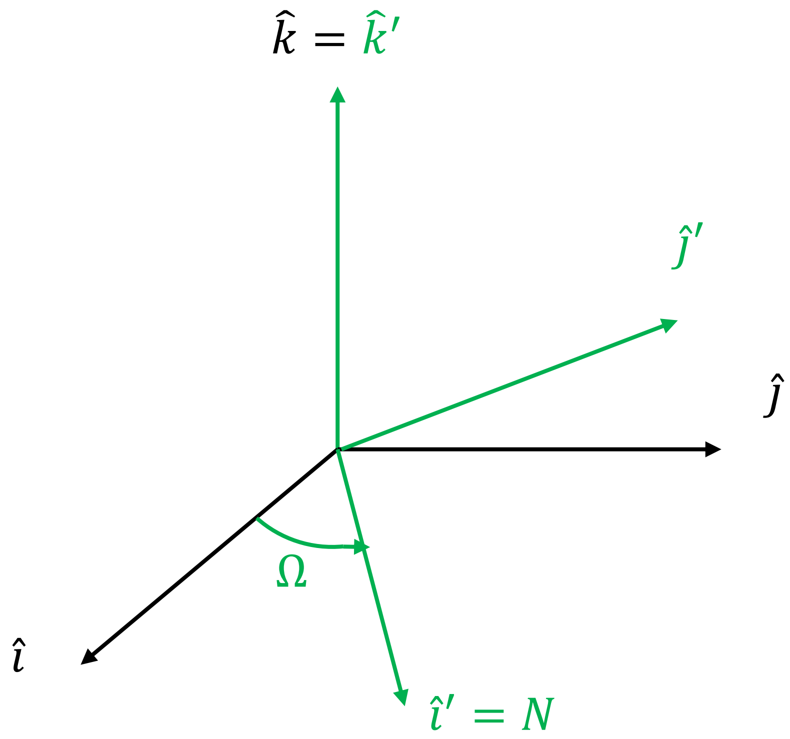
- **Origine**: **fuoco** *pieno* dell'orbita
- **Assi:**
  - $\hat{e}$ versore eccentricità
  - $\hat{p}$ direzione del semilato retto
  - $\hat{h}$ versore momento angolare



Panoramica dei sistemi di riferimento:



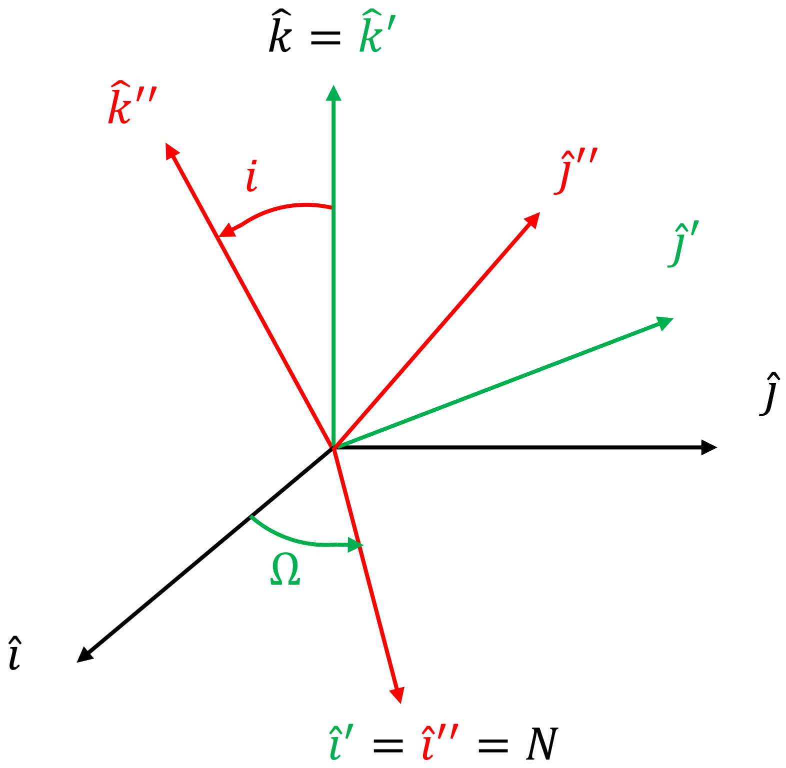
From ECI to PF:
3 rotations



Geocentrico inerziale (ECI): $[\hat{\imath}, \hat{\jmath}, \hat{k}]$

1) Rotazione di Ω intorno a $\hat{k}$

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



Geocentrico inerziale (ECI): $[\hat{\imath}, \hat{\jmath}, \hat{k}]$

⇓

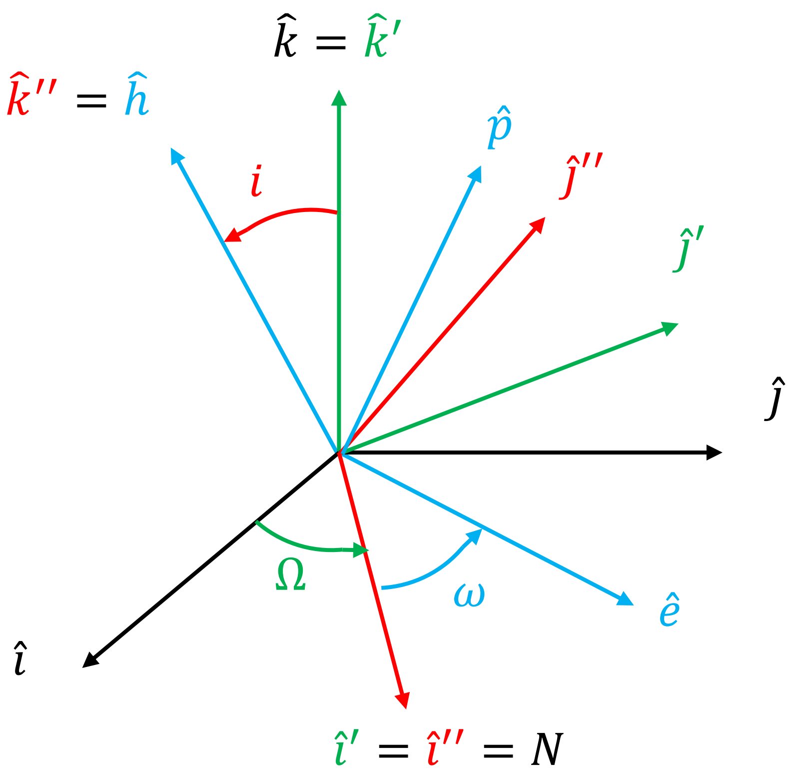
1) Rotazione di Ω intorno a $\hat{k}$

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

⇓

2) Rotazione di i intorno a $\hat{\imath}'$

$$R_i = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos i & \sin i \\ 0 & -\sin i & \cos i \end{bmatrix}$$



Geocentrico inerziale (ECI): $[\hat{\imath}, \hat{\jmath}, \hat{k}]$

1) Rotazione di Ω intorno a $\hat{k}$

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

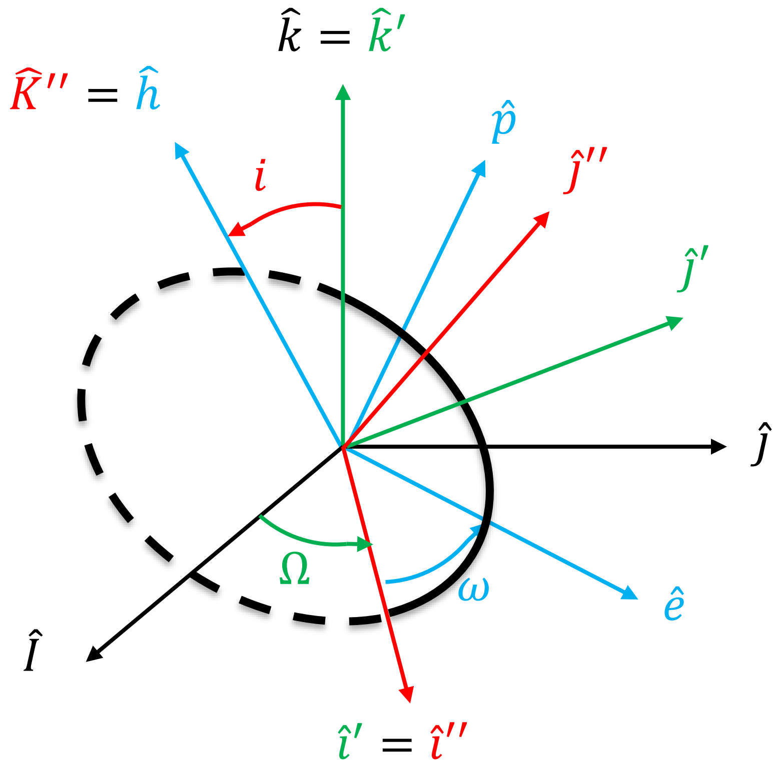
2) Rotazione di i intorno a $\hat{\imath}'$

$$R_i = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos i & \sin i \\ 0 & -\sin i & \cos i \end{bmatrix}$$

3) Rotazione di ω intorno a $\hat{k}''$

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

Sistema di riferimento perifocale (PF): $[\hat{e}, \hat{p}, \hat{h}]$



Matrice di rotazione totale

$$T_{ECI \to PF} = R_\omega R_i R_\Omega$$

$$T_{PF \to ECI} = T_{ECI \to PF}^T = R_\Omega^T R_i^T R_\omega^T$$

INPUT: $[a, e, i, \Omega, \omega, \theta], (\mu)$

1. Semilato retto

$$p = a(1 - e^2)$$

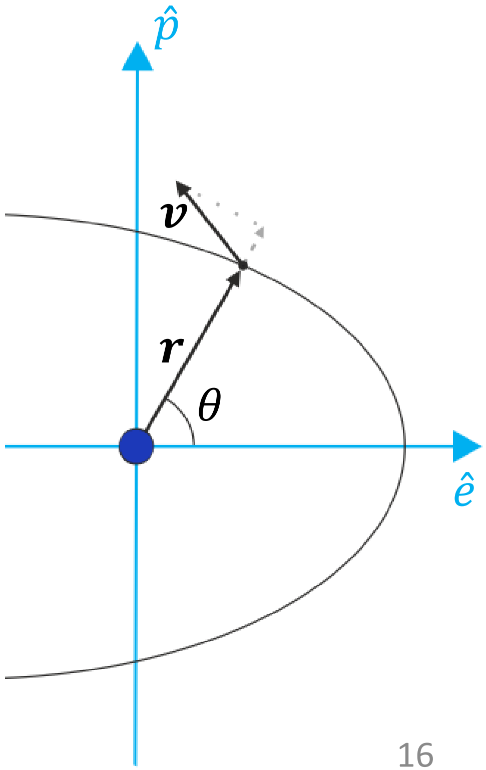
2. Modulo della posizione

$$r = \frac{p}{1 + e\cos\theta}$$

3. Vettore $[\tilde{\boldsymbol{r}}, \tilde{\boldsymbol{v}}]$ (*vettore di stato in PF*)

$$\tilde{\boldsymbol{r}} = r \begin{bmatrix} \cos\theta \\ \sin\theta \\ 0 \end{bmatrix}$$

$$\tilde{\boldsymbol{v}} = \sqrt{\frac{\mu}{p}} \begin{bmatrix} -\sin\theta \\ e + \cos\theta \\ 0 \end{bmatrix}$$



4. **Vettore $[\boldsymbol{r}, \boldsymbol{v}]$**

$$T_{ECI \rightarrow PF} = R_{\omega} R_{i} R_{\Omega}$$
$$\boldsymbol{r} = T^{T} \tilde{\boldsymbol{r}}$$
$$\boldsymbol{v} = T^{T} \tilde{\boldsymbol{v}}$$

OUTPUT: $[\boldsymbol{r}, \boldsymbol{v}]$

par2car in MATLAB

```matlab
function [rr, vv] = par2car(a, e, i, OM, om, th, mu)
```

```matlab
% Trasformation from Keplerian parameters to cartesian coordinates
%
% [rr, vv] = par2car(a, e, i, OM, om, th, mu)
%
% ---------------------------------------------------------------------------
% Input arguments:
% a              [1x1]   semi-major axis                     [km]
% e              [1x1]   eccentricity                        [-]
% i              [1x1]   inclination                         [rad]
% OM             [1x1]   RAAN                                [rad]
% om             [1x1]   pericenter anomaly                  [rad]
% th             [1x1]   true anomaly                        [rad]
% mu             [1x1]   gravitational parameter             [km^3/s^2]
%
% ---------------------------------------------------------------------------
% Output arguments:
% rr             [3x1]   position vector                     [km]
% vv             [3x1]   velocity vector                     [km/s]
```

Obiettivo:

Dati i parametri kepleriani classici $[a, e, i, \Omega, \omega]$, l'anomalia vera iniziale e finale (θ_0, θ_f) e la *step size* $\Delta\theta$, plottare l'orbita tridimensionale punto per punto.

Procedura

- Definire il vettore delle anomalie vere da θ_0 a θ_f con passo $\Delta\theta$
- Determinare il vettore di stato $[\boldsymbol{r}, \boldsymbol{v}]_i$ per ogni θ_i, dati i parametri kepleriani
- Plottare i vettori di stato in una figura

```
function plotOrbit(a, e, i, Om, om, th0, thf, dth, mu)

% 3D orbit plot
%
% plotOrbit(a, e, i, Om, om, th0, thf, dth, mu)
%
%--------------------------------------------------------------------------
% Input arguments:
% a              [1x1]   semi-major axis                     [km]
% e              [1x1]   eccentricity                        [-]
% i              [1x1]   inclination                         [rad]
% OM             [1x1]   RAAN                                [rad]
% om             [1x1]   pericenter anomaly                  [rad]
% th0            [1x1]   initial true anomaly                [rad]
% thf            [1x1]   final true anomaly                  [rad]
% dth            [1x1]   true anomaly step size              [rad]
% mu             [1x1]   gravitational parameter             [km^3/s^2]
```

Don’t reinvent the wheel!

MATLAB ha molte funzioni che ci semplificano la vita e che è inutile riscrivere. Es:

- *dot* – prodotto scalare
- *cross* – prodotto vettoriale
- *norm* – norma euclidea
- *rotx, roty, rotz* – matrici di rotazione (fate attenzione ai segni e gradi/rad!)
- *plot3* – 3D plot

Be careful

Prestate attenzione alla scrittura del codice e a non fare confusione con i vettori riga e colonna.

Es: Dato un vettore $\boldsymbol{a}[3 \times 1]$ e un vettore $\boldsymbol{b}[1 \times 3]$, $\boldsymbol{a} - \boldsymbol{b}$ restituirà una matrice 3x3

Test the code

Verificate sempre i codici che scrivete!

Es: Provate a usare in sequenza *par2car-car2par* e viceversa.

Playground

Adesso, **DIVERTITEVI**!