MSS II — LEZIONE 10 — 06-11-18

ANGOLI DI EULERO

$$T = \begin{bmatrix} R & P \\ 0\ 0\ 0 & 1 \end{bmatrix}$$

$P \rightarrow$ Traslazione 3DOF $(3\times 1)$

$R \rightarrow$ Rotazione $(3\times 3)$

$R\ [\hat{x}\ \hat{y}\ \hat{z}]$ : questi tre versori, le tre colonne della matrice R sono legati dalle condizioni di ortogonalità. Si hanno nove elementi legati da 6 eq.:

9 elementi:

- 3 eq. ortogonalità
- 3 eq. di normalità

} 3 Parametri indipendenti 3DOF

Allora siccome non è possibile specificare la R direttamente come le P, si usano gli angoli di Eulero, una terna di rotazioni elementari, rispetto a tre assi indipendenti

$$R = R_x(\alpha)\, R_y(\beta)\, R_z(\gamma) \qquad EA\ \ x-y-z\ 1$$

Scanned by CamScanner

La R è ottenuta da rotazioni concatenate di assi mobili, l'asse corrente è diverso da quello che si aveva all'inizio.

Si può rappresentare la stessa concatenazione attraverso un'altra terna di angoli:

EA $z - y - x$

$= R_z(\alpha) R_y(\beta) R_x(\gamma)$

È un modo diverso per rappresentare la stessa orientazione (cambiano gli angoli)

Esistono dodici terne di angoli di Eulero.

Si può avere, ad esempio:

EA $x - y - x$

Il primo asse x è differente dall'ultimo, che è stato modificato da y

EA $x - x - y$ non è una terna di Eulero (due rotazioni successive



Scanned by CamScanner

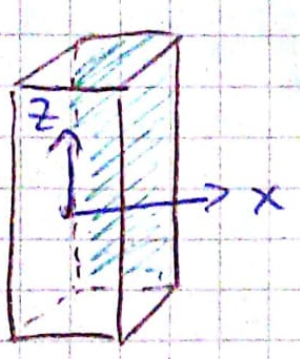
sullo stesso asse

quindi uso la terna di Eulero

R ha tre DOF, in modo che venga soddisfatto che $R\,[\hat{x}\ \hat{y}\ \hat{z}] \rightarrow 3\ DOF \rightarrow$

$$R = R_x(\alpha)\,R_y(\beta)\,R_z(\gamma)$$

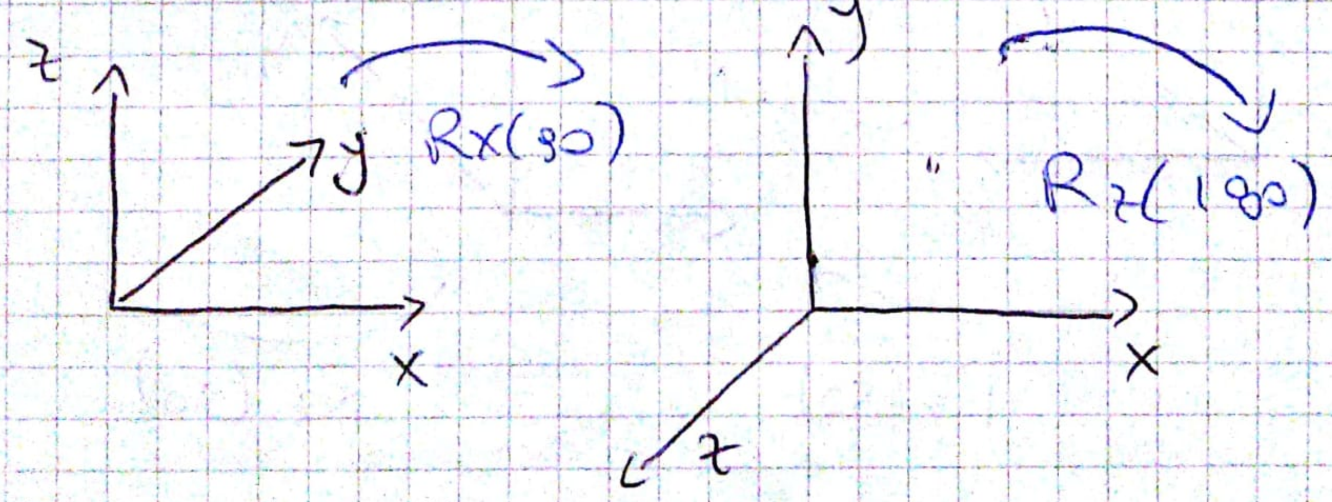
Esempio: si consideri una barretta [A]




si vuole vedere l'orientazione della barretta quando la sua orientazione sarà descritta dai seguenti angoli EA $x - z - x$

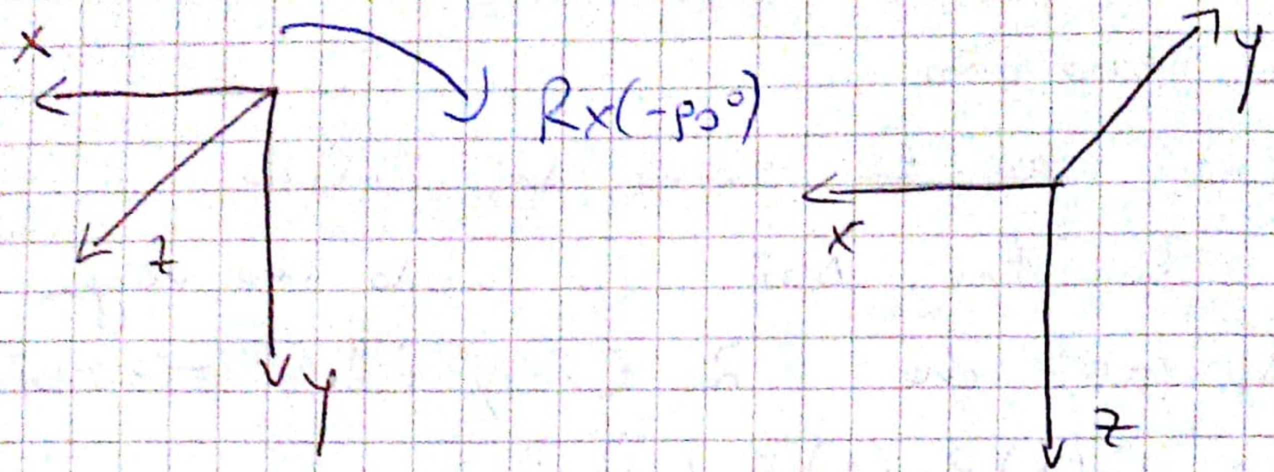
$90° \quad 180° \quad -90°$

$$R_x(90°)\,R_z(180°)\,R_x(-90°)$$

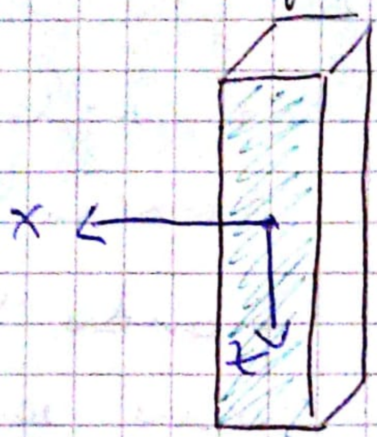





Scanned by CamScanner

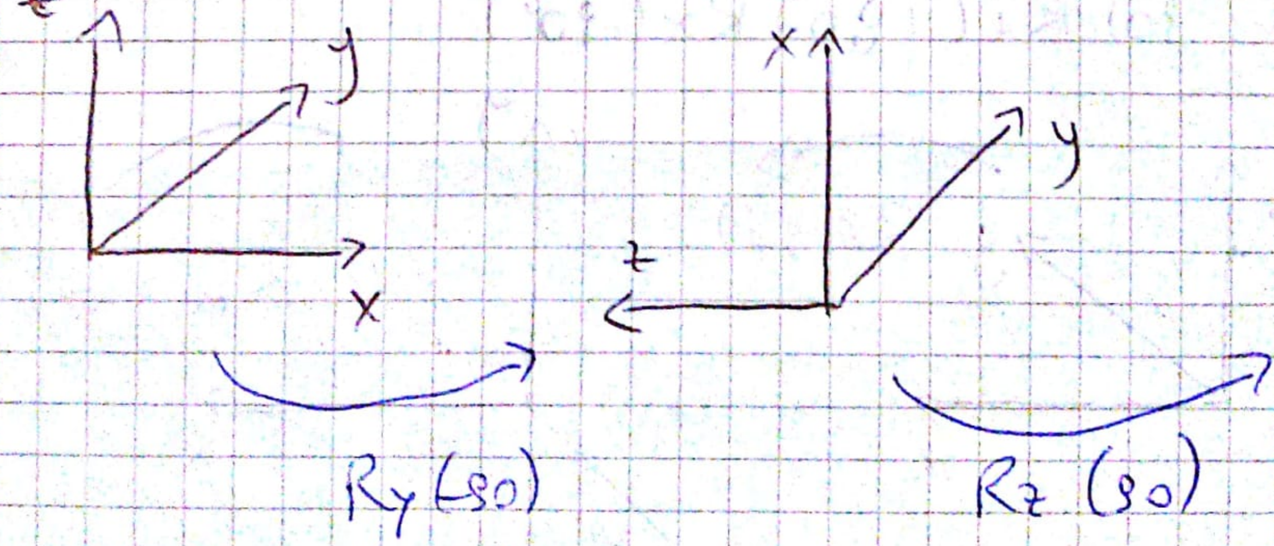


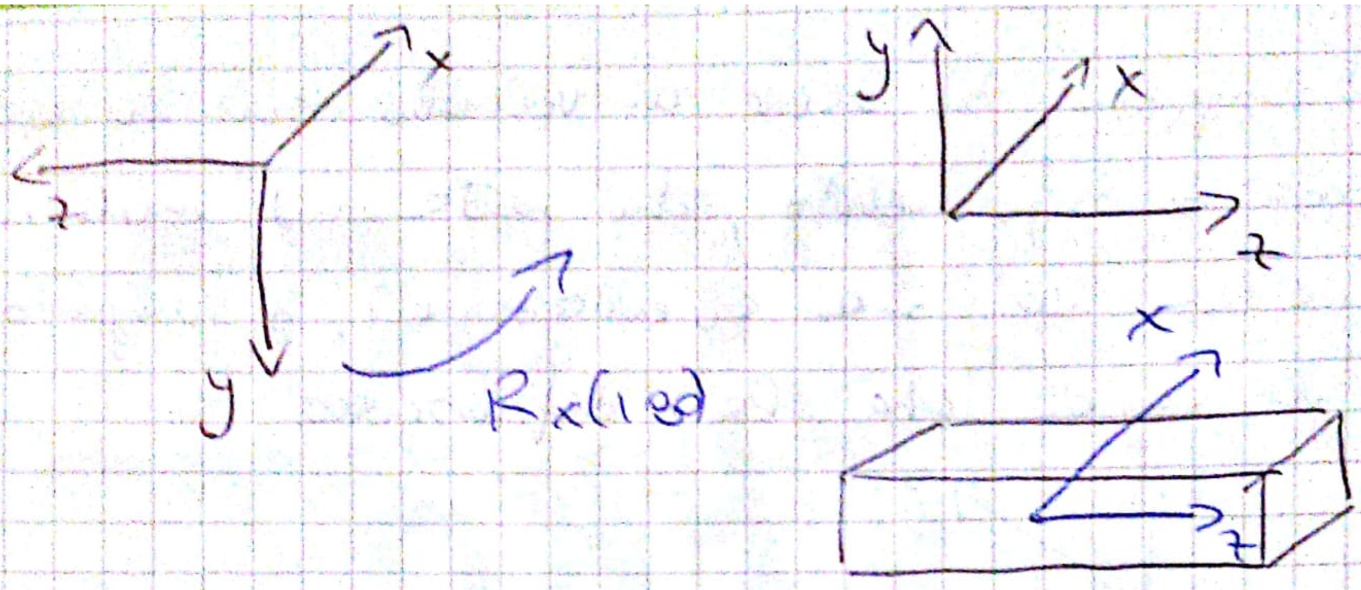

Si rispetto cio il corpo al sistema di riferimento.




Esempio: EA Y-Z-X

-90 90 180






Si può ottenere una sorta di cinematica diretta o inversa, in base al dato usato.

$$R = R_x(\,)\ R_y(\,)\ R_z(\,)$$

orientazione
ROSA

angoli

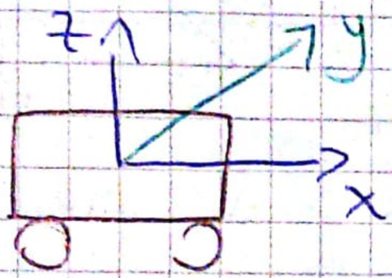
DK ←

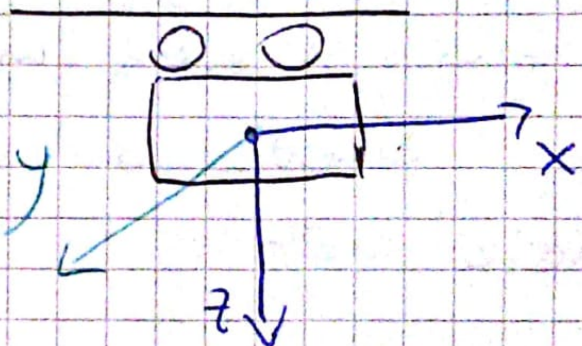
cinematica diretta

cinematica inversa

→ IK

Si suppone di avere un veicolo con un sistema di riferimento, che ruota. Si vuole ricavare la sua orientazione, o meglio gli angoli che la definiscono.




$$\begin{cases} 180 \quad 0 \quad 0 \\ 0 \quad 0 \quad 180 \end{cases}$$

EA X-Z-X

Nella cinematica inversa, ad ogni matrice corrispondono più terne (non è univoca come nella cinematica diretta → soluzioni univoche) → soluzioni multiple

$$\left.\begin{matrix} 0 & 180 & 0 \\ 180 & 0 & 180 \end{matrix}\right\} \text{EA} \quad Y-X-Z$$

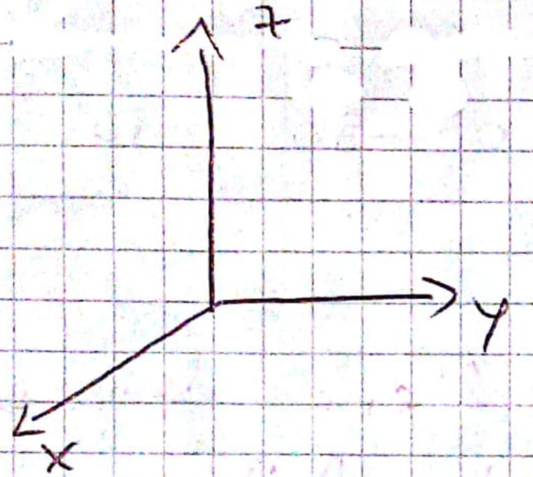


Scanned by CamScanner

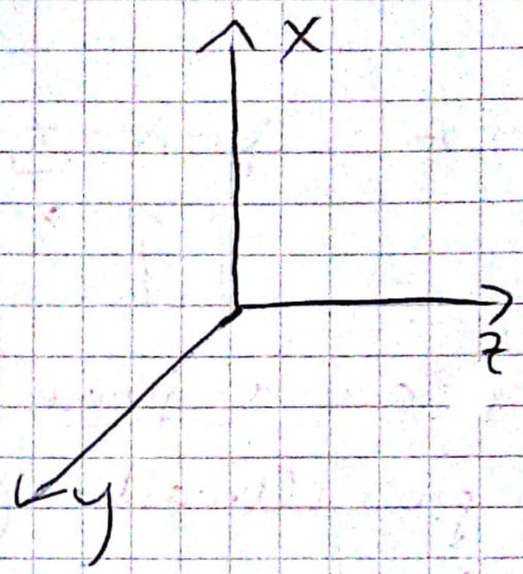
Si calcolano queste sequenze di angoli:

EA: X - Z - X    EA: Y - X - Z

Da




a




Bisogna considerare l'ultimo asse della trasformazione, perché è quello che verifica se si è trasformato il primo sistema in quello finale

In questo caso nella terna da identificare, bisogna trovare due rotazioni che allineano l'asse x all'asse x del sistema finale

Ruotando o attorno ad x o attorno a z non si riesce ad allineare l'ultimo x -> rotazioni entrambe non nulle

EA $x - z - \boxed{x}$ $\begin{cases} 90° \quad 90° \quad 180° \\ -90° \end{cases}$

Primo terna

EA $Y - X - \boxed{Z}$ * $\begin{cases} 0° \quad -90° \quad -90° \end{cases}$

* ca le prime due l'asse deve essere già allineato prima di effettuare la trasformazione

$$R = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ \vdots & \vdots & \vdots \\ \vdots & \vdots & r_{33} \end{bmatrix} \quad (3 \times 3)$$

oppure

$$R = \begin{bmatrix} r_{11} & r_{12} & r_{13} & 0 \\ \vdots & \vdots & \vdots & 0 \\ \vdots & \vdots & r_{33} & 0 \end{bmatrix}$$

$(4 \times 4)$ a seconda della necessità

se $z - y - z$ allora $R$ si può scrivere come



Scanned by CamScanner

$$R = R_z(\alpha) R_y(\beta) R_z(\gamma) = \begin{bmatrix} \times & \times & C_\alpha S_\beta \\ \times & \times & S_\alpha S_\beta \\ -S_\beta C_\gamma & S_\beta S_\gamma & C_\beta \end{bmatrix}$$

× termini che non servono per quello che si sta facendo ora

→ 3 equazioni dipendenti in 3 incognite

Bisogna ricavare le 3 eq. indipendenti
→ 3 incognite

Come individuare queste 3 eq. dalle nove eq. scalari della forma matriciale?
Bisogna manipolare gli elementi per ricavarle

Per la terna Z-Y-Z si considera l'elemento $C_\beta$ per ricavare $\beta$

$\beta = \pm \arccos(r_{33})$ già si vede l'ambiguità, segno sia positivo che negativo, questo determinerà la soluzione multipla

Noto β, allora:

$$\alpha = \operatorname{atan2}\left(\frac{r_{23}}{s_\beta}, \frac{r_{13}}{s_\beta}\right)$$

$$\gamma = \operatorname{atan2}\left(\frac{r_{32}}{s_\beta}, -\frac{r_{31}}{s_\beta}\right)$$

È la procedura, della cinematica inversa, per ricavare gli angoli di Eulero a partire dalla matrice di rotazione

Casi particolari: $\beta = 0$, $\beta = 180$

il seno di $\beta$ diventa $\varnothing$ e non esistono i rapporti per calcolare $\alpha$ e $\gamma$

Allora possono essere calcolati in un altro modo, sfruttando gli elementi $r$ nella matrice $R$, che a questo punto si semplificano e quelli [illegible] fuori sono tutti zero (annullabili) o uno



Scanned by CamScanner

Nel caso di $\beta = 0$, infatti la R si riduce alla seguente forma:

$$R_{\beta=0} = \begin{bmatrix} c\alpha c\gamma - s\alpha s\gamma & -c\alpha s\gamma - s\alpha s\gamma & \dots \\ s\alpha c\gamma + c\alpha s\gamma & -s\alpha s\gamma + c\alpha c\gamma & \dots \\ \dots & \dots & \dots \end{bmatrix}$$

Usando le formule di addizione e sottrazione trigonometriche si ottiene $\alpha = 0$, $\gamma = \operatorname{atan2}(-r_{12}, r_{11})$ che non sono funzione degli angoli ma della loro somma algebrica perciò si sceglie il primo angolo a zero e si ricava l'altro, per alla somma, per convenzione.

Questo approccio si utilizza per ogni altra terna degli angoli di Eulero e per ogni altro meccanismo.

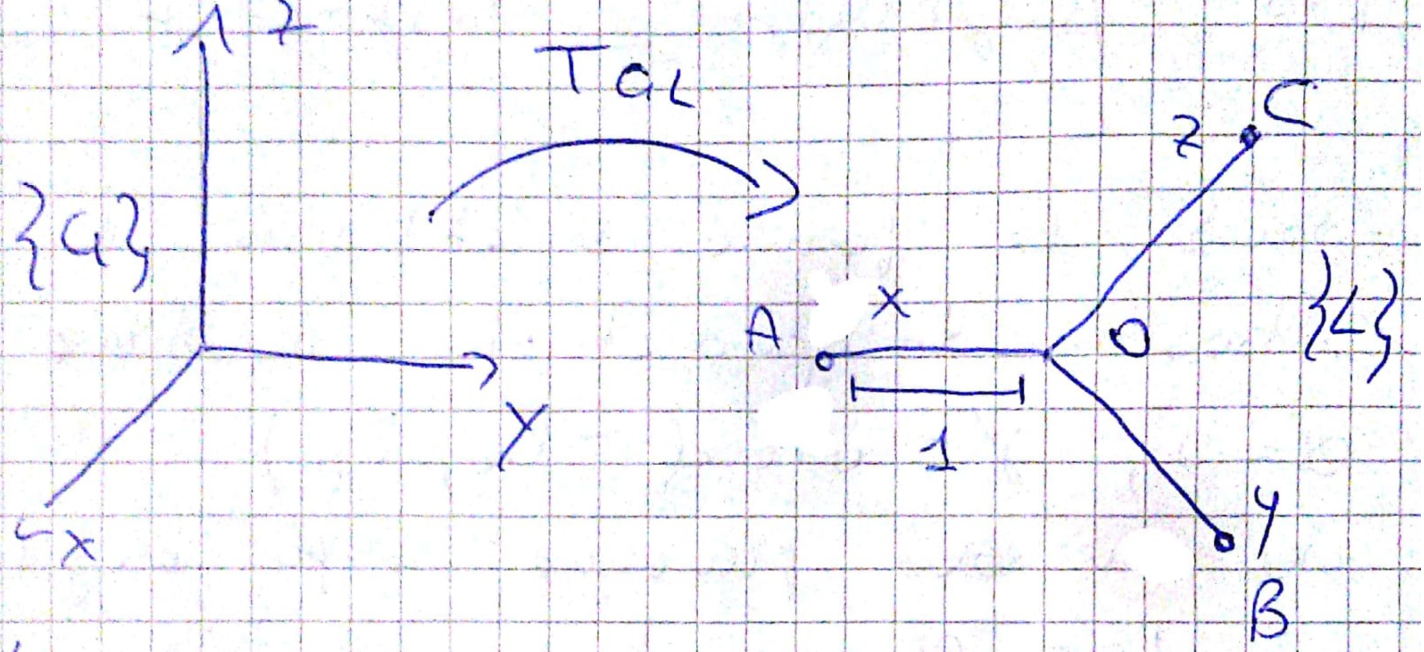
Approccio generico

In Matlab : Matrici elementari

$D_x$, $D_y$, $D_z$ , $R_x$, $R_y$, $R_z$

$D_x(10) * D_y(4) * D_z(1) * R_x(0,3) * R_y(-0,5) * R_z(0,7)$

FUNCTION PLOT




${}^{L}O = [0\ 0\ 0\ 1]^T \qquad O^{G} = T_{GL}\,{}^{L}O$

${}^{L}A = [1\ 0\ 0\ 1] \qquad {}^{G}A = ?$

${}^{L}B = [0\ 1\ 0\ 0]$

box effinge un riquadro tridimensionale



Scanned by CamScanner

plot (eye(4)) e poi plot (T)

sistema globale → sistema locale

$R = R_x(0,1)\, R_y(-0,6)\, R_z(0,8)$

$[a, b, c] = eu\_zyx(R)$

pocket_IMU che usa il file di dati acquisito dalle app del telefono Orientazione

Verificare che in un movimento accoppiato la rappresentazione sia quella corretta

Import Data → Numeric Matrix



Scanned by CamScanner