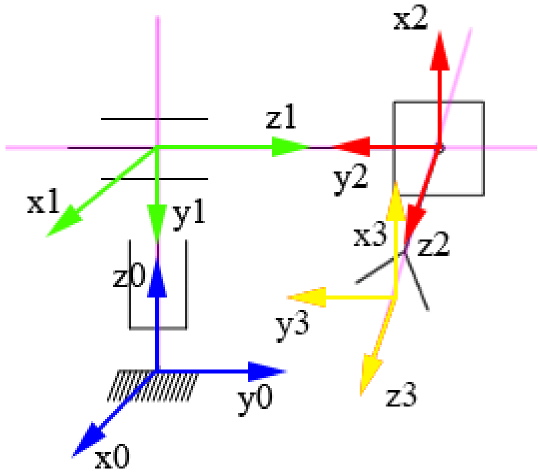
SCinematica diretta Robot Cartesiano

***N.B.:* le grandezze diverse da quelle di giunto q_i sono L_i, D_i. Esse sono rispettivamente la distanza tra i sistemi di riferimento R_i e R_{i+1} nelle operazioni della matrice avvitamento $A_z(\theta, d)$ e $A_x(\alpha, a)$.**



	ϑ	d	α	a
1	0	q_1	$-\frac{\pi}{2}$	0
2	$-\frac{\pi}{2}$	q_2	$-\frac{\pi}{2}$	0
3	0	q_3	0	0

Tabella 1.

Funzioni ausiliarie:

```
(%i1) inverseLaplace(SI,theta):=block([res],
                              M:SI,
                              MC:SI,
                              for i:1 thru 3 do(
                                for j:1 thru 3 do
                                   (
                                     aC:M[i,j],
                                     b:ilt(aC,s,theta),
                                     MC[i,j]:b
                                    )
                                 ),
                              res:MC

                           )
```

(%o1) inverseLaplace(SI, ϑ) := **block** ([res], M: SI, MC: SI, **for** i **thru** 3 **do for** j **thru** 3 **do** (aC: $M_{i,j}$, b: ilt(aC, s, ϑ), MC$_{i,j}$: b), res: MC)

```
(%i2) rotLaplace(k,theta):=block([res],
                      S:ident(3),
                      I:ident(3),
                     for i:1 thru 3 do
                      (
                      for j:1 thru 3 do
                         (
                          if i=j
                             then S[i][j]:0
                          elseif j>i
                             then (
                            temp:(-1)^(j-i)*k[3-remainder(i+j,3)],
                                   S[i][j]:temp,
                                   S[j][i]:-temp

                                    )
                          )
                       ),
                      res:inverseLaplace(invert(s*I-S),theta)


                     )
```

(%o2) $\mathrm{rotLaplace}(k,\vartheta):=\mathbf{block}\ ([\mathrm{res}], S\colon \mathrm{ident}(3), I\colon \mathrm{ident}(3),$ $\mathbf{for}\ i\ \mathbf{thru}\ 3\ \mathbf{do}\ \mathbf{for}\ j\ \mathbf{thru}\ 3\ \mathbf{do}\ \mathbf{if}\ i=j\ \mathbf{then}\ (S_i)_j\colon 0\ \mathbf{elseif}\ j>i\ \mathbf{then}\ (\mathrm{temp}\colon$ $(-1)^{j-i}\,k_{3-\mathrm{remainder}(i+j,3)}, (S_i)_j\colon \mathrm{temp}, (S_j)_i\colon -\mathrm{temp}), \mathrm{res}\colon \mathrm{inverseLaplace}(\mathrm{invert}(s\,I-S),\vartheta))$

```
(%i3) Av(v,theta,d):=block([res],
                      Trot:rotLaplace(v,theta),
                      row:matrix([0,0,0,1]),
                      Atemp:addcol(Trot,d*transpose(v)),
                      A:addrow(Atemp,row),
                      res:A
                      )
```

(%o3) $\mathrm{Av}(v,\vartheta,d):=\mathbf{block}\ ([\mathrm{res}], \mathrm{Trot}\colon \mathrm{rotLaplace}(v,\vartheta), \mathrm{row}\colon \begin{pmatrix} 0 & 0 & 0 & 1 \end{pmatrix}, \mathrm{Atemp}\colon \mathrm{addcol}(\mathrm{Trot},$ $d\,\mathrm{transpose}(v)), A\colon \mathrm{addrow}(\mathrm{Atemp},\mathrm{row}), \mathrm{res}\colon A)$

```
(%i4) Q(theta,d,alpha,a):=block([res],
                           tempMat:Av([0,0,1],theta,d).Av([1,0,0],alpha,a),
                           Qtrasf:zeromatrix(4,4),
                           for i:1 thru 4 do
                      (
                      for j:1 thru 4 do
                         (
                           Qtrasf[i][j]:trigreduce(tempMat[i][j])
                                    )
                         ),

                           res:Qtrasf
                           )
```

(%o4) $Q(\vartheta,d,\alpha,a):=\mathbf{block}\ ([\mathrm{res}], \mathrm{tempMat}\colon \mathrm{Av}([0,0,1],\vartheta,d)\cdot \mathrm{Av}([1,0,0],\alpha,a), \mathrm{Qtrasf}\colon$ $\mathrm{zeromatrix}(4,4), \mathbf{for}\ i\ \mathbf{thru}\ 4\ \mathbf{do}\ \mathbf{for}\ j\ \mathbf{thru}\ 4\ \mathbf{do}\ (\mathrm{Qtrasf}_i)_j\colon \mathrm{trigreduce}((\mathrm{tempMat}_i)_j), \mathrm{res}\colon \mathrm{Qtrasf})$

```
(%i5) let(sin(q[1]), s[1]);
```

(%o5) $\sin(q_1) \longrightarrow s_1$

```
(%i6) let(sin(q[2]), s[2]);
```

(%o6) $\sin(q_2) \longrightarrow s_2$

```
(%i7) let(cos(q[1]),c[1]);
```

(%o7) $\cos(q_1) \longrightarrow c_1$

```
(%i8) let(cos(q[2]),c[2]);
```

(%o8) $\cos(q_2) \longrightarrow c_2$

```
(%i9) let(sin(q[1]+q[2]), s[12]);
```

(%o9) $\sin(q_2 + q_1) \longrightarrow s_{12}$

```
(%i10) let(cos(q[1]+q[2]),c[12]);
```

(%o10) $\cos(q_2 + q_1) \longrightarrow c_{12}$

```
(%i11)
```

Cinematica diretta:

```
(%i11) Q[rc](q1,q2,q3):=trigsimp(trigrat(trigreduce(trigexpand(
                        Q(0,q1,-(%pi/2),0).
                         Q(-(%pi/2),q2,-(%pi/2),0).
                         Q(0,q3,0,0)
                         ))));
```

(%o11) $Q_{\mathrm{rc}}(q1, q2, q3) := \mathrm{trigsimp}\Big(\mathrm{trigrat}\Big(\mathrm{trigreduce}\Big(\mathrm{trigexpand}\Big(Q\Big(0, q1, -\frac{\pi}{2}, 0\Big) \cdot Q\Big(-\frac{\pi}{2}, q2, -\frac{\pi}{2}, 0\Big) \cdot Q(0, q3, 0, 0)\Big)\Big)\Big)\Big)$

```
(%i12) Q[rc](q[1],q[2],q[3]);
```

(%o12) $\begin{pmatrix} 0 & 0 & 1 & q_3 \\ 0 & -1 & 0 & q_2 \\ 1 & 0 & 0 & q_1 \\ 0 & 0 & 0 & 1 \end{pmatrix}$

```
(%i13)
```

Cinematica inversa

Dato che il robot cartesiano è un robot con 3 gradi di libertà (3DOF) è possibile effettuare l'analisi della cinematica inversa di posizione e di orientamento inverso.

Occorre inizialmente individuare lo spazio di lavoro, le soluzioni generiche, singolari ed infine le variabili di giunto q_i.

Dalla cinematica diretta sappiamo che:

$$Q_{\text{cartesiano}} = \begin{pmatrix} 0 & 0 & 1 & q_3 \\ 0 & -1 & 0 & q_2 \\ 1 & 0 & 0 & q_1 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Cinematica inversa di posizione

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} q_3 \\ q_2 \\ q_1 \end{pmatrix}$$

Da cui lo spazio di lavoro, idealmente, è $\mathbb{R}^3$, non vi sono singolarità ma solamente una soluzione. Inoltre, possiamo semplicemente risolvere il problema di cinematica inversa di posizione ponendo:

$$q_1 = z$$
$$q_2 = y$$
$$q_3 = x$$

Orientamento inverso

La risoluzione del problema di orientamento inverso si basa sulla scelta di una terna di Eulero o nautica in condizione non singolari. In particolare, l'elemento più semplice deve risultare diverso da ± 1. In particolare:

$$R_{\text{cartesiano}} = \begin{pmatrix} 0 & 0 & 1 \\ 0 & -1 & 0 \\ 1 & 0 & 0 \end{pmatrix}$$

$$\text{Terna nautica: } R_{\text{yzx}} = \begin{pmatrix} c_y c_z & \cdots\cdots\cdots \\ s_z \quad c_z c_x & -c_z s_x \\ s_y c_z & \cdots\cdots\cdots \end{pmatrix}$$

Poiché l'elemento $s_z = 0 \neq \pm 1$ è possibile risolvere il problema di orientamento inverso con la terna nautica yzx. Infatti:

$$s_z = 0 \neq \pm 1 \longrightarrow c_z = \pm\sqrt{1 - s_z} = \pm 1$$

$$\phi_z = \text{atan2}(s_z, c_z)$$

$$\phi_z = \{0, \pi\}$$

$$\begin{matrix} c_y c_z = 0 & & c_y = 0 \\ & \xrightarrow{c_z = \pm 1} & \\ s_y c_z = 1 & & s_y = \pm 1 \end{matrix}$$

$$\phi_y = \text{atan2}(\pm s_y, c_y)$$

$$\phi_y = \left\{\frac{\pi}{2}, -\frac{\pi}{2}\right\}$$

$$\begin{matrix} c_z c_x = -1 & & c_x = \mp 1 \\ & \xrightarrow{c_z = \pm 1} & \\ -c_z s_x = 0 & & s_x = 0 \end{matrix}$$

$$\phi_x = \text{atan2}(s_x, \mp c_x)$$

$$\phi_x = \{\pi, 0\}$$

Riassumendo, le soluzioni sono:

$$\begin{pmatrix} \pi \\ \frac{\pi}{2} \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ -\frac{\pi}{2} \\ \pi \end{pmatrix}$$

In alternativa utilizzando una terna di Eulero:

$$R_{\text{zyz}} = \begin{pmatrix} \cdots\cdots\cdots\cdots & -c_\alpha s_\beta \\ \cdots\cdots\cdots\cdots & -s_\alpha s_\beta \\ s_\beta c_\gamma \quad - s_\beta s_\gamma & c_\beta \end{pmatrix}$$

$$c_\beta = 0 \longrightarrow s_\beta = \pm 1$$

$$\beta = \text{atan2}(\pm s_\beta, c_\beta)$$

$$\beta = \left\{\frac{\pi}{2}, -\frac{\pi}{2}\right\}$$

$$\begin{matrix} c_\alpha s_\beta = 1 & & c_\alpha = \mp 1 \\ & -s_\beta = \pm 1 \rightarrow & \\ s_\alpha s_\beta = 0 & & s_\alpha = 0 \end{matrix}$$

$$\alpha = \text{atan2}(s_\alpha, \mp c_\alpha)$$

$$\alpha = \{\pi, 0\}$$

$$\begin{matrix} c_\gamma s_\beta = 1 & & c_\gamma = \pm 1 \\ & -s_\beta = \pm 1 \rightarrow & \\ s_\gamma s_\beta = 0 & & s_\gamma = 0 \end{matrix}$$

$$\gamma = \text{atan2}(s_\gamma, \mp c_\gamma)$$

$$\gamma = \{0, \pi\}$$

Riassumendo, le soluzioni sono:

$$\begin{pmatrix} \pi \\ \frac{\pi}{2} \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ -\frac{\pi}{2} \\ \pi \end{pmatrix}$$

```
(%i13) R(k,theta):= block([res],
                       if k = x
                           then res:matrix([1,0,0],
                          [0,cos(theta),-sin(theta)],
                          [0,sin(theta), cos(theta)])
                       elseif k = y
                           then res:matrix([cos(theta),0,sin(theta)],
                          [0,1,0],
                          [-sin(theta),0, cos(theta)])
                       elseif k = z
                           then res:matrix([cos(theta),-sin(theta),0],
                          [sin(theta),cos(theta),0],
                          [0,0, 1])
                        else
                          res:"Incorrect axis of rotation"
                        )
```

(%o13) $R(k, \vartheta) := \textbf{block}\left([\text{res}], \textbf{if } k = x \textbf{ then } \text{res:} \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos(\vartheta) & -\sin(\vartheta) \\ 0 & \sin(\vartheta) & \cos(\vartheta) \end{pmatrix} \textbf{elseif } k = y \textbf{ then } \text{res:} \right.$

$\begin{pmatrix} \cos(\vartheta) & 0 & \sin(\vartheta) \\ 0 & 1 & 0 \\ -\sin(\vartheta) & 0 & \cos(\vartheta) \end{pmatrix}$ **elseif** $k = z$ **then** res: $\begin{pmatrix} \cos(\vartheta) & -\sin(\vartheta) & 0 \\ \sin(\vartheta) & \cos(\vartheta) & 0 \\ 0 & 0 & 1 \end{pmatrix}$ **else** res: Incorrect axis of rotation)

(%i15) `Reulero:R(z,gamma).R(y,beta).R(z,alpha);`

(%o15) $(\cos(\alpha)\cos(\beta)\cos(\gamma) - \sin(\alpha)\sin(\gamma), -\cos(\alpha)\sin(\gamma) - \sin(\alpha)\cos(\beta)\cos(\gamma), \sin(\beta)\cos(\gamma); \cos(\alpha)\cos(\beta)\sin(\gamma) + \sin(\alpha)\cos(\gamma), \cos(\alpha)\cos(\gamma) - \sin(\alpha)\cos(\beta)\sin(\gamma), \sin(\beta)\sin(\gamma); -\cos(\alpha)\sin(\beta), \sin(\alpha)\sin(\beta), \cos(\beta))$

(%i17) `Rnautico:R(y,phi[y]).R(z,phi[z]).R(x,phi[x]);`

(%o17) $(\cos(\varphi_y)\cos(\varphi_z), \sin(\varphi_x)\sin(\varphi_y) - \cos(\varphi_x)\cos(\varphi_y)\sin(\varphi_z), \sin(\varphi_x)\cos(\varphi_y)\sin(\varphi_z) + \cos(\varphi_x)\sin(\varphi_y); \sin(\varphi_z), \cos(\varphi_x)\cos(\varphi_z), -\sin(\varphi_x)\cos(\varphi_z); -\sin(\varphi_y)\cos(\varphi_z), \cos(\varphi_x)\sin(\varphi_y)\sin(\varphi_z) + \sin(\varphi_x)\cos(\varphi_y), \cos(\varphi_x)\cos(\varphi_y) - \sin(\varphi_x)\sin(\varphi_y)\sin(\varphi_z))$

(%i18)
```
isRotation(M):=block([MC,res],
                         I:ident(3),
                         MC:ident(3),
                         for i:1 thru 3 do
                         (
                         for j:1 thru 3 do
                            (
                               MC[i][j]:M[i][j]
                               )
                            ),
                         MMT:trigsimp(expand(MC.transpose(MC))),
                         detM:trigsimp(expand(determinant(MC))),

                         if MMT=I  and detM=1
                            then(

                                 return(res:1)
                                 )

                         else(

                               res: "R is not rotation matrix"
                               )
                         )
```

(%o18) isRotation(M) := **block** ([MC, res], I: ident(3), MC: ident(3), **for** i **thru** 3 **do for** j **thru** 3 **do** $(\mathrm{MC}_i)_j$: $(M_i)_j$, MMT: trigsimp(expand(MC $\cdot$ transpose(MC))), detM: trigsimp(expand(determinant(MC))), **if** MMT $= I \wedge$ detM $= 1$ **then** return(res: 1) **else** res: R is not rotation matrix)

```
(%i21) invCartesiano(Qdiretta):=block([pos,orien1,orien2,res],
                                        rotation:isRotation(Qdiretta),
                                        if rotation=1 then(
                                            pos:transpose(
                                                  [
                                                  Qdiretta[1][4],
                                                  Qdiretta[2][4],
                                                  Qdiretta[3][4]
                                                  ]),
                                            sz:Qdiretta[2][1],
                                            cz:sqrt(1-sz^(2)),
                                            phiz1:atan2(sz,cz),
                                            phiz2:atan2(sz,-cz),
                                            cy:Qdiretta[1][1]/cz,
                                            sy:Qdiretta[3][1]/cz,
                                            phiy1:atan2(sy,cy),
                                            phiy2:atan2(-sy,cy),
                                            cx:Qdiretta[2][2]/cz,
                                            sx:Qdiretta[2][3]/cz,
                                            phix1:atan2(sx,cx),
                                            phix2:atan2(sx,-cx),
                                            orien1:transpose([phix1,phiy1,phiz1]),
                                            orien2:transpose([phix2,phiy2,phiz2]),
                                            res:[pos,orien1,orien2]

                                        )
                                        else res:rotation
                                        );
```

(%o21) $\mathrm{invCartesiano}(\mathrm{Qdiretta}) := \mathbf{block}\Big([\mathrm{pos}, \mathrm{orien1}, \mathrm{orien2}, \mathrm{res}], \mathrm{rotation}\colon \mathrm{isRotation}(\mathrm{Qdiretta}), \mathbf{if}\ \mathrm{rotation} = 1\ \mathbf{then}\ \Big(\mathrm{pos}\colon \mathrm{transpose}([(\mathrm{Qdiretta}_1)_4, (\mathrm{Qdiretta}_2)_4, (\mathrm{Qdiretta}_3)_4]), \mathrm{sz}\colon (\mathrm{Qdiretta}_2)_1, \mathrm{cz}\colon \sqrt{1-\mathrm{sz}^2}, \mathrm{phiz1}\colon \mathrm{atan2}(\mathrm{sz}, \mathrm{cz}), \mathrm{phiz2}\colon \mathrm{atan2}(\mathrm{sz}, -\mathrm{cz}), \mathrm{cy}\colon \frac{(\mathrm{Qdiretta}_1)_1}{\mathrm{cz}}, \mathrm{sy}\colon \frac{(\mathrm{Qdiretta}_3)_1}{\mathrm{cz}}, \mathrm{phiy1}\colon \mathrm{atan2}(\mathrm{sy}, \mathrm{cy}), \mathrm{phiy2}\colon \mathrm{atan2}(-\mathrm{sy}, \mathrm{cy}), \mathrm{cx}\colon \frac{(\mathrm{Qdiretta}_2)_2}{\mathrm{cz}}, \mathrm{sx}\colon \frac{(\mathrm{Qdiretta}_2)_3}{\mathrm{cz}}, \mathrm{phix1}\colon \mathrm{atan2}(\mathrm{sx}, \mathrm{cx}), \mathrm{phix2}\colon \mathrm{atan2}(\mathrm{sx}, -\mathrm{cx}), \mathrm{orien1}\colon \mathrm{transpose}([\mathrm{phix1}, \mathrm{phiy1}, \mathrm{phiz1}]), \mathrm{orien2}\colon \mathrm{transpose}([\mathrm{phix2}, \mathrm{phiy2}, \mathrm{phiz2}]), \mathrm{res}\colon [\mathrm{pos}, \mathrm{orien1}, \mathrm{orien2}]\Big)\ \mathbf{else}\ \mathrm{res}\colon \mathrm{rotation}\Big)$

```
(%i22) invCartesiano(Q[rc](q[1],q[2],q[3]));
```

(%o22) $$\left[\begin{pmatrix} q_3 \\ q_2 \\ q_1 \end{pmatrix}, \begin{pmatrix} \pi \\ \frac{\pi}{2} \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ -\frac{\pi}{2} \\ \pi \end{pmatrix}\right]$$

```
(%i24) invCartesiano(Q[rc](10,15,20));
```

(%o24) $$\left[\begin{pmatrix} 20 \\ 15 \\ 10 \end{pmatrix}, \begin{pmatrix} \pi \\ \frac{\pi}{2} \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ -\frac{\pi}{2} \\ \pi \end{pmatrix}\right]$$

```
(%i25)
```

Singolarità di velocità

```
(%i11) x:q3;
```
(%o11) $q3$

```
(%i12) y:q2
```
(%o12) $q2$

```
(%i13) z:q1
```
(%o13) $q1$

```
(%i31) J:matrix([diff(x,q1),diff(x,q2),diff(x,q3)],
               [diff(y,q1),diff(y,q2),diff(y,q3)],
               [diff(z,q1),diff(z,q2),diff(z,q3)]);
```
(%o31) $\begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix}$

```
(%i32) dJ:determinant(J);
```
(%o32) -1

```
(%i33)
```

Poiché $\det(J) \neq 0 \;\; \forall$q, non vi sono singolarità cinematiche di velocità. Infatti è possibile ottenere tutte le velocita che $\in \mathrm{Im}\{J\} = \left[\begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} \right]$.
In aggiunta le velocità che risultano singolare sono date da $\ker\{J\} = \left\{ \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} \right\}$.

```
(%i33) Jtr:-transpose(J);
```
(%o33) $\begin{pmatrix} 0 & 0 & -1 \\ 0 & -1 & 0 \\ -1 & 0 & 0 \end{pmatrix}$

```
(%i34) dJtr:determinant(Jtr);
```
(%o34) 1

```
(%i35)
```

Poiché $\det(J) \neq 0 \;\; \forall$q, non vi sono singolarità cinematiche di forza. Infatti è possibile ottenere tutte le forze che $\in \mathrm{Im}\{J\} = \left[\begin{pmatrix} -1 \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 0 \\ -1 \end{pmatrix}, \begin{pmatrix} 0 \\ -1 \\ 0 \end{pmatrix} \right]$.
In aggiunta le forze che risultano singolare sono date da $\ker\{J\} = \left\{ \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} \right\}$.