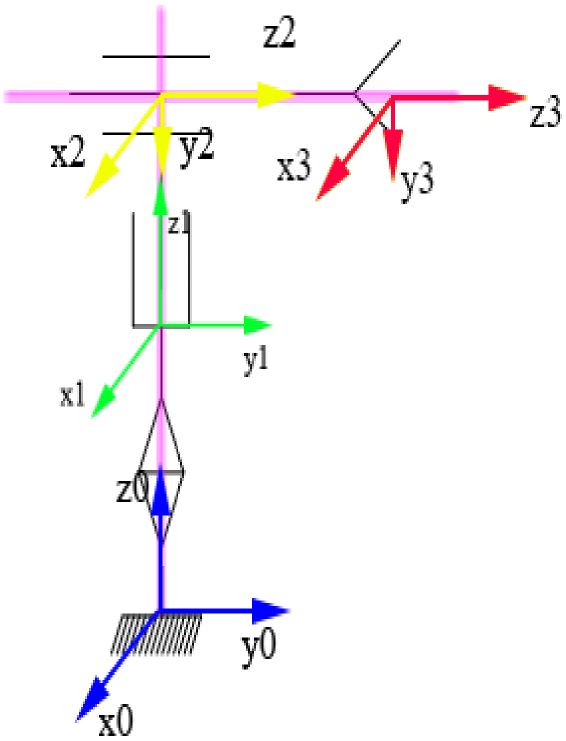
**Cinematica diretta Robot Cilindrico**

***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 $A_x(\alpha, a)$.**




| | $\vartheta$ | $d$ | $\alpha$ | $a$ |
|---|---|---|---|---|
| 1 | $q_1$ | $L_1$ | 0 | 0 |
| 2 | 0 | $q_2$ | $-\frac{\pi}{2}$ | 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, $\vartheta$) := **block** ([res], $M$: SI, MC: SI, **for** $i$ **thru** 3 **do for** $j$ **thru** 3 **do** (aC: $M_{i,j}$, $b$: ilt(aC, $s$, $\vartheta$), 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)

(%i11) Q[cilindrico](q1,q2,q3,L1):=trigsimp(trigrat(trigreduce(trigexpand(
                              Q(q1,L1,0,0).
                               Q(0,q2,-(%pi/2),0).
                               Q(0,q3,0,0)
                               ))));
```

(%o11) $Q_{\text{cilindrico}}(q1, q2, q3, L1) := \text{trigsimp}\Big(\text{trigrat}\Big(\text{trigreduce}\Big(\text{trigexpand}\Big(Q(q1, L1, 0, 0) \cdot Q\Big(0, q2, -\frac{\pi}{2}, 0\Big) \cdot Q(0, q3, 0, 0)\Big)\Big)\Big)\Big)$

```
(%i12) Qcilindrico:Q[cilindrico](q[1],q[2],q[3],L1);
```

(%o12) $\begin{pmatrix} \cos(q_1) & 0 & -\sin(q_1) & -q_3 \sin(q_1) \\ \sin(q_1) & 0 & \cos(q_1) & q_3 \cos(q_1) \\ 0 & -1 & 0 & L1 + q_2 \\ 0 & 0 & 0 & 1 \end{pmatrix}$

```
(%i13) letsimp(Qcilindrico);
```

(%o13) $\begin{pmatrix} c_1 & 0 & -s_1 & -s_1\, q_3 \\ s_1 & 0 & c_1 & c_1\, q_3 \\ 0 & -1 & 0 & L1 + q_2 \\ 0 & 0 & 0 & 1 \end{pmatrix}$

```
(%i14)
```

### Cinematica Inversa Robot Cilindrico

Al fine di risolvere il problema di cinematica inversa del robot cilindrico occorre risolevere il problema di posizione ed orientamente inverso.Inizialmente individuare lo spazio di lavoro, le soluzioni generiche, singolari ed infine le variabili di giunto $q_i$ ed in seguito determinare l'orientamento del robot.

Dalla cinematica diretta del robot cilindrico sappiamo che:

$$Q_{\text{cilindrico}} = \begin{pmatrix} c_1 & 0 & -s_1 & -s_1\, q_3 \\ s_1 & 0 & c_1 & c_1\, q_3 \\ 0 & -1 & 0 & q_2 + L_1 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

## Cinematica inversa di posizione

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} -\sin(q_1)\, q_3 \\ \cos(q_1)\, q_3 \\ q_2 + L_1 \end{pmatrix}$$

La variabile di igunto $q_2$, poiché $L_1$ e $z$ sono noti:

$$q_2 = z - L_1$$

Quindi occorre risolvere:

$$\begin{cases} x = -\sin(q_1)\, q_3 \\ y = \cos(q_1)\, q_3 \end{cases} \longrightarrow \begin{cases} x^2 = \sin(q_1)^2\, q_3^2 \\ y^2 = \cos(q_1)^2\, q_3^2 \end{cases}$$

$$x^2 + y^2 = q_3^2\, (\sin(q_1)^2 + \cos(q_1)^2)$$

Determinando di conseguenza lo spazio operativo $:= x^2 + y^2 = q_3^2$

Rappresenta un cilindro il cui asse di rotazione è una soluzione singolare ottenuta da:

$$q_3 = \pm\sqrt{x^2 + y^2} \quad 2\,\text{soluzioni generiche}$$

$$q_3 = 0 \longrightarrow \sqrt{x^2 + y^2} = 0 \longrightarrow \text{soluzione singolare}$$

Per determinare $q_1$ occorre supporre che $x^2 + y^2 \neq 0 \longrightarrow q_3 \neq 0$:

$$\begin{cases} x = -\sin(q_1)\, q_3 \\ y = \cos(q_1)\, q_3 \end{cases} \longrightarrow \begin{cases} \sin(q_1) = -\frac{x}{q_3} \\ \cos(q_1) = \frac{y}{q_3} \end{cases}$$

Infine:

$$q_1 = \text{atan2}(\sin(q_1), \cos(q_1))$$

## Orientamento inverso

La risoluzione del problema di orientamento inverso si basa sulla scelta di una terna di Eulero o nautica in condizione non singolari.

$$R_{\text{cilindrico}} = \begin{pmatrix} c_1 & 0 & -s_1 \\ s_1 & 0 & c_1 \\ 0 & -1 & 0 \end{pmatrix}$$

$$R_{\text{zyx}} = \begin{pmatrix} c_y c_z & \cdots\cdots\cdots \\ c_y c_z & \cdots\cdots\cdots \\ -s_y & s_x c_y & c_x c_y \end{pmatrix}$$

Poiché l'elemento $-s_y = 0 \neq \pm 1$ è possibile risolvere il problema di orientamento inverso con la terna nautica zyx. In particolare:

$$s_y = 0 \longrightarrow c_y = \pm 1 \longrightarrow \phi_y = \text{atan2}(s_y, c_y) \longrightarrow \phi_y = \begin{cases} 0 \\ \pi \end{cases}$$

$$\begin{cases} c_y s_x = -1 \\ c_y c_x = 0 \end{cases} \longrightarrow \phi_x = \text{atan2}(\mp s_x, c_x) \longrightarrow \phi_x = \begin{cases} -\frac{\pi}{2} \\ \frac{\pi}{2} \end{cases}$$

$$\begin{cases} c_y c_z = c_1 \\ c_y s_z = s_1 \end{cases} \longrightarrow \phi_z = \text{atan2}(\pm s_1, \pm c_1) \longrightarrow \phi_z = \begin{cases} q_1 \\ q_1 + \pi \end{cases}$$

Riassumendo, le soluzioni sono:

$$\begin{pmatrix} -\frac{\pi}{2} \\ 0 \\ q_1 \end{pmatrix}, \begin{pmatrix} \frac{\pi}{2} \\ \pi \\ q_1 + \pi \end{pmatrix}$$

In alternativa, utilizzando una terna di Eulero:

$$R_{\text{yxy}} = \begin{pmatrix} \cdots & \sin(\alpha)\sin(\beta) & \cdots \\ \sin(\beta)\sin(\gamma) & \cos(\beta) & -\sin(\beta)\cos(\gamma) \\ \cdots & \cos(\alpha)\sin(\beta) & \cdots \end{pmatrix}$$

$$\cos(\beta) = 0 \longrightarrow \sin(\beta) = \pm\sqrt{1 - \cos(\beta)^2} = \pm 1$$

$$\beta = \text{atan2}(\pm\sin(\beta), \cos(\beta))$$

$$\beta = \begin{cases} \frac{\pi}{2} \\ -\frac{\pi}{2} \end{cases}$$

$$\begin{cases} \sin(\alpha)\sin(\beta) = 0 \\ \cos(\alpha)\sin(\beta) = -1 \end{cases} \longrightarrow \begin{cases} \sin(\alpha) = 0 \\ \cos(\alpha) = \mp 1 \end{cases}$$

$$\alpha = \text{atan2}(\sin(\alpha), \mp\cos(\alpha))$$

$$\alpha = \begin{cases} \pi \\ 0 \end{cases}$$

$$\begin{cases} \sin(\beta)\sin(\gamma) = s_1 \\ -\sin(\beta)\cos(\gamma) = c_1 \end{cases} \longrightarrow \begin{cases} \sin(\gamma) = \pm s_1 \\ \cos(\gamma) = \mp c_1 \end{cases}$$

$$\gamma = \text{atan2}(\pm s_1, \mp c_1)$$

$$\gamma = \begin{cases} q_1 \\ q_1 + \pi \end{cases}$$

Riassumendo, si hanno 2 soluzioni:

$$\begin{pmatrix} \frac{\pi}{2} \\ \pi \\ q_1 + \pi \end{pmatrix}, \begin{pmatrix} -\frac{\pi}{2} \\ 0 \\ q_1 \end{pmatrix}$$

```
(%i16) 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"
                           )
```

(%o16) $R(k,\vartheta):=\mathbf{block}\left([\mathrm{res}],\mathbf{if}\ k=x\ \mathbf{then}\ \mathrm{res}:\begin{pmatrix}1 & 0 & 0\\ 0 & \cos(\vartheta) & -\sin(\vartheta)\\ 0 & \sin(\vartheta) & \cos(\vartheta)\end{pmatrix}\mathbf{elseif}\ k=y\ \mathbf{then}\ \mathrm{res}:\begin{pmatrix}\cos(\vartheta) & 0 & \sin(\vartheta)\\ 0 & 1 & 0\\ -\sin(\vartheta) & 0 & \cos(\vartheta)\end{pmatrix}\mathbf{elseif}\ k=z\ \mathbf{then}\ \mathrm{res}:\begin{pmatrix}\cos(\vartheta) & -\sin(\vartheta) & 0\\ \sin(\vartheta) & \cos(\vartheta) & 0\\ 0 & 0 & 1\end{pmatrix}\mathbf{else}\ \mathrm{res}:\mathrm{Incorrect\ \ axis\ \ of\ \ rotation}\right)$

```
(%i17) 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"
                                  )
                            )
```

(%o17) $\mathrm{isRotation}(M):=\mathbf{block}\ ([\mathrm{MC},\mathrm{res}], I:\mathrm{ident}(3), \mathrm{MC}:\mathrm{ident}(3),$
$\mathbf{for}\ i\ \mathbf{thru}\ 3\ \mathbf{do}\ \mathbf{for}\ j\ \mathbf{thru}\ 3\ \mathbf{do}\ (\mathrm{MC}_i)_j:(M_i)_j, \mathrm{MMT}:\mathrm{trigsimp}(\mathrm{expand}(\mathrm{MC}\cdot\mathrm{transpose}(\mathrm{MC}))),$
$\mathrm{detM}:\mathrm{trigsimp}(\mathrm{expand}(\mathrm{determinant}(\mathrm{MC}))), \mathbf{if}\ \mathrm{MMT}=I\wedge\mathrm{detM}=1\ \mathbf{then}\ \mathrm{return}(\mathrm{res}:1)\ \mathbf{else}\ \mathrm{res}:\mathrm{R}$
$\mathrm{is\ not\ rotation\ matrix}\ )$

```
(%i28) invCilindrico(x,y,z,phi,L1):=block([R,pos1,pos2,orien1,orien2,res],

                                    if x^2+y^2#0 then(
                                      R:matrix([cos(phi),0,sin(phi)],
                                               [sin(phi),0,cos(phi)],
                                               [0,1,0]),
                                      q3:cabs(trigsimp(sqrt(x^2+y^2))),
                                      q1alto:atan2(-x/q3,y/q3),
                                      q1basso:atan2(x/q3,-y/q3),
                                      q2:z-L1,
                                      pos1:[q1alto,q2,q3],
                                      pos2:[q1basso,q2,-q3],
                                      sy:R[3][1],
                                      cy:sqrt(1-sy^2),
                                      phiy2:atan2(sy,cy),
                                      phiy1:atan2(sy,-cy),
                                      sx:R[3][2],
                                      cx:R[3][3],
                                      phix1:atan2(-sx,cx),
                                      phix2:atan2(sx,cx),
                                      cz:R[1][1],
                                      sz:R[2][1],
                                      phiz1:atan2(sz,cz),
                                      phiz2:atan2(-sz,-cz),
                                      orien1:[phix1,phiy1,phiz1],
                                      orien2:[phix2,phiy2,phiz2],
                                      res:[pos1,pos2,orien1,orien2]
                                    )
                                    else res:"La configurazione è singolare"

                                  );
```

(%o28) $\mathrm{invCilindrico}(x,\ y,\ z,\ \varphi,\ L1) := \mathbf{block}\left([R,\ \mathrm{pos1},\ \mathrm{pos2},\ \mathrm{orien1},\ \mathrm{orien2},\ \mathrm{res}],\ \mathbf{if}\ x^2 + y^2 \neq 0\ \mathbf{then}\left(R\colon \begin{pmatrix} \cos(\varphi) & 0 & \sin(\varphi) \\ \sin(\varphi) & 0 & \cos(\varphi) \\ 0 & 1 & 0 \end{pmatrix},\ q3\colon \mathrm{cabs}\left(\mathrm{trigsimp}\left(\sqrt{x^2+y^2}\right)\right),\ q1\mathrm{alto}\colon \mathrm{atan2}\left(\frac{-x}{q3}, \frac{y}{q3}\right),\ q1\mathrm{basso}\colon \mathrm{atan2}\left(\frac{x}{q3}, \frac{-y}{q3}\right),\ q2\colon z - L1,\ \mathrm{pos1}\colon [q1\mathrm{alto},\ q2,\ q3],\ \mathrm{pos2}\colon [q1\mathrm{basso},\ q2,\ -q3],\ \mathrm{sy}\colon (R_3)_1,\ \mathrm{cy}\colon \sqrt{1-\mathrm{sy}^2},\ \mathrm{phiy2}\colon \mathrm{atan2}(\mathrm{sy},\ \mathrm{cy}),\ \mathrm{phiy1}\colon \mathrm{atan2}(\mathrm{sy},\ -\mathrm{cy}),\ \mathrm{sx}\colon (R_3)_2,\ \mathrm{cx}\colon (R_3)_3,\ \mathrm{phix1}\colon \mathrm{atan2}(-\mathrm{sx},\ \mathrm{cx}),\ \mathrm{phix2}\colon \mathrm{atan2}(\mathrm{sx},\ \mathrm{cx}),\ \mathrm{cz}\colon (R_1)_1,\ \mathrm{sz}\colon (R_2)_1,\ \mathrm{phiz1}\colon \mathrm{atan2}(\mathrm{sz},\ \mathrm{cz}),\ \mathrm{phiz2}\colon \mathrm{atan2}(-\mathrm{sz},\ -\mathrm{cz}),\ \mathrm{orien1}\colon [\mathrm{phix1},\ \mathrm{phiy1},\ \mathrm{phiz1}],\ \mathrm{orien2}\colon [\mathrm{phix2},\ \mathrm{phiy2},\ \mathrm{phiz2}],\ \mathrm{res}\colon [\mathrm{pos1},\ \mathrm{pos2},\ \mathrm{orien1},\ \mathrm{orien2}]\right)\ \mathbf{else}\ \mathrm{res}\colon \text{La configurazione è singolare}\right)$

```
(%i29) QdirettaC: Q[cilindrico](q[1],q[2],q[3],15);
```

(%o29)
$$\begin{pmatrix} \cos(q_1) & 0 & -\sin(q_1) & -q_3 \sin(q_1) \\ \sin(q_1) & 0 & \cos(q_1) & q_3 \cos(q_1) \\ 0 & -1 & 0 & q_2 + 15 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

```
(%i42) invCilindrico(10,10,10,%pi/2,15);
```

(%o42) $\left[\left[-\frac{\pi}{4},-5,5\,2^{\frac{3}{2}}\right],\left[\frac{3\,\pi}{4},-5,-5\,2^{\frac{3}{2}}\right],\left[-\frac{\pi}{2},\pi,\frac{\pi}{2}\right],\left[\frac{\pi}{2},0,-\frac{\pi}{2}\right]\right]$

```
(%i43) QdirettaC: Q[cilindrico](-(%pi/4),-5,5*2^((3/2)),15);
```

(%o43) $\begin{pmatrix} \frac{1}{\sqrt{2}} & 0 & \frac{1}{\sqrt{2}} & 10 \\ -\frac{1}{\sqrt{2}} & 0 & \frac{1}{\sqrt{2}} & 10 \\ 0 & -1 & 0 & 10 \\ 0 & 0 & 0 & 1 \end{pmatrix}$

```
(%i44)
```

**Singolarità di velocità**

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} -\sin(q_1)\,q_3 \\ \cos(q_1)\,q_3 \\ q_2 + L_1 \end{pmatrix}$$

$$J = \frac{\delta h}{\delta q} = \begin{pmatrix} -q_3 c_1 & 0 & -s_1 \\ -q_3 s_1 & 0 & c_1 \\ 0 & 1 & 0 \end{pmatrix}$$

$$\det(J) = 0 \Leftrightarrow q_3 = 0$$

$$J(q_3 = 0) = \begin{pmatrix} 0 & 0 & -s_1 \\ 0 & 0 & c_1 \\ 0 & 1 & 0 \end{pmatrix}$$

$$a\begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} + b\begin{pmatrix} -s_1 \\ c_1 \\ 0 \end{pmatrix} = \Im m \rightarrow \mathrm{Ker}\{J\} = \Im m\left\{\begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}\right\}$$

In singolarità con $v = \begin{pmatrix} V \\ 0 \\ 0 \end{pmatrix} \forall v \Rightarrow w = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$

```
(%i1) x:-sin(q[1])*q[3]
```

(%o1) $-q_3 \sin(q_1)$

```
(%i2) y:cos(q[1])*q[3]
```

(%o2) $q_3 \cos(q_1)$

```
(%i3) z:q[2]+L[1]
```

(%o3) $q_2 + L_1$

```
(%i4) J(q1,q2,q3):=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)]);
```

(%o4) $J(q1, q2, q3) := \begin{pmatrix} \mathrm{diff}(x, q1) & \mathrm{diff}(x, q2) & \mathrm{diff}(x, q3) \\ \mathrm{diff}(y, q1) & \mathrm{diff}(y, q2) & \mathrm{diff}(y, q3) \\ \mathrm{diff}(z, q1) & \mathrm{diff}(z, q2) & \mathrm{diff}(z, q3) \end{pmatrix}$

```
(%i5) J:J(q[1],q[2],q[3]);
```

(%o5) $\begin{pmatrix} -q_3\cos(q_1) & 0 & -\sin(q_1) \\ -q_3\sin(q_1) & 0 & \cos(q_1) \\ 0 & 1 & 0 \end{pmatrix}$

```
(%i6) dJ:trigsimp(determinant(J));
```

(%o6) $q_3$

```
(%i9) Jq3:subst(q[3]=0,J);
```

(%o9) $\begin{pmatrix} 0 & 0 & -\sin(q_1) \\ 0 & 0 & \cos(q_1) \\ 0 & 1 & 0 \end{pmatrix}$

```
(%i10) nullspace(Jq3);
```

Proviso: notequal$(-\sin(q_1), 0) \wedge$ notequal$(-\sin(q_1), 0)$

(%o10) $\mathrm{span}\left(\begin{pmatrix} -\sin(q_1) \\ 0 \\ 0 \end{pmatrix}\right)$

Se $q_1 \neq 0$, le singolarità di velocità si hanno per $v \in \mathrm{Im}\left\{\begin{pmatrix} -\sin(q_1) \\ 0 \\ 0 \end{pmatrix}\right\}$.

Se $q_1 = 0$:

```
(%i11) Jq31:subst(q[1]=0,Jq3);
```

(%o11) $\begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix}$

```
(%i12) nullspace(Jq31);
```

(%o12) $\mathrm{span}\left(\begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}\right)$

$q_1 \neq 0$, le singolarità di velocità si hanno per $v \in \mathrm{Im}\left\{\begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}\right\}$.

**Singolarità di forza**

```
(%i13) Jtr(q1,q2,q3):=-transpose(J(q1,q2,q3));
```

(%o13) $\mathrm{Jtr}(q1, q2, q3) := -\mathrm{transpose}(J(q1, q2, q3))$

```
(%i14) Jtrn:Jtr(q[1],q[2],q[3]);
```

(%o14) $\begin{pmatrix} q_3\cos(q_1) & q_3\sin(q_1) & 0 \\ 0 & 0 & -1 \\ \sin(q_1) & -\cos(q_1) & 0 \end{pmatrix}$

```
(%i15) dJtr:trigsimp(determinant(Jtrn));
```

(%o15) $-q_3$

```
(%i16) Jq3:subst(q[3]=0,Jtrn);
```

(%o16) $\begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ \sin(q_1) & -\cos(q_1) & 0 \end{pmatrix}$

```
(%i18) nullspace(Jq3);
```

Proviso: notequal$(\cos(q_1), 0)$

(%o18) $\mathrm{span}\left(\begin{pmatrix} \cos(q_1) \\ \sin(q_1) \\ 0 \end{pmatrix}\right)$

(%i19)

$\mathrm{Ker}\{\mathrm{J}\}=\begin{pmatrix} \text{-cos } (q_1) \\ \text{sin } (q_1) \\ 0 \end{pmatrix} \rightarrow \forall q_1 \neq 0$

Non si possono applicare forze t.c. $\tau = \mathrm{Im}\{\begin{pmatrix} \text{-cos } (q_1) \\ \text{sin } (q_1) \\ 0 \end{pmatrix}\}$

Se $q_1$=0:

```
(%i19) Jq1:subst(q[1]=0,Jq3);
```

(%o19) $\begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & -1 & 0 \end{pmatrix}$

```
(%i20) nullspace(Jq1);
```

(%o20) $\mathrm{span}\left(\begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}\right)$

(%i21)

Non si possono applicare forze t.c. $\tau = \mathrm{Im}\{\begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}\}$