Cinematica diretta 2 DOF planare

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

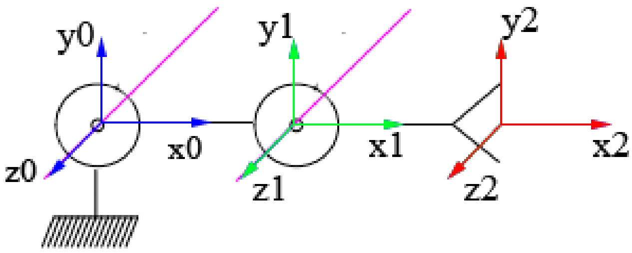


Tabella Denavit-Hartenberg 2 DOF Planare

	ϑ	d	α	a
1	q_1	0	0	L_1
2	q_2	0	0	L_2

Tabella 1.

Funzioni ausiliarie:

```
(%i1) isRotation(M):=block([MC,res,I,MMT,detM],
                        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"
                              )
                        )
```

(%o1) isRotation(M) := **block** ([MC, res, I, MMT, detM], 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)

```
(%i2) inverseLaplace(SI,theta):=block([res,M,MC,aC,b],
                             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

                          )
```

(%o2) inverseLaplace(SI, ϑ) := **block** ([res, M, MC, aC, b], 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)

```
(%i3) rotLaplace(k,theta):=block([res,S,I,temp],
                           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)


                         )
```

(%o3) rotLaplace(k, ϑ) := **block** ([res, S, I, temp], S: ident(3), I: ident(3), **for** i **thru** 3 **do for** j **thru** 3 **do if** $i=j$ **then** $(S_i)_j$: 0 **elseif** $j>i$ **then** (temp: $(-1)^{j-i}\, k_{3-\mathrm{remainder}(i+j,3)}$, $(S_i)_j$: temp, $(S_j)_i$: $-$temp), res: inverseLaplace(invert($s\, I - S$), ϑ))

```
(%i4) Av(v,theta,d):=block([res,Tot,row,Atemp,A],
                           Trot:rotLaplace(v,theta),
                           row:matrix([0,0,0,1]),
                           Atemp:addcol(Trot,d*transpose(v)),
                           A:addrow(Atemp,row),
                           res:A
                           )
```

(%o4) Av(v, ϑ, d) := **block** ([res, Tot, row, Atemp, A], Trot: rotLaplace(v, ϑ), row: $\begin{pmatrix} 0 & 0 & 0 & 1 \end{pmatrix}$, Atemp: addcol(Trot, d transpose(v)), A: addrow(Atemp, row), res: A)

```
(%i5) Q(theta,d,alpha,a):=block([res,tempMat,Qtrasf],
                               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
                               )
```

(%o5) $Q(\vartheta, d, \alpha, a) := \mathbf{block}\ ([\mathrm{res}, \mathrm{tempMat}, \mathrm{Qtrasf}], \mathrm{tempMat}\colon \mathrm{Av}([0,0,1], \vartheta, d) \cdot \mathrm{Av}([1,0,0], \alpha, a), \mathrm{Qtrasf}\colon \mathrm{zeromatrix}(4,4), \mathbf{for}\ i\ \mathbf{thru}\ 4\ \mathbf{do}\ \mathbf{for}\ j\ \mathbf{thru}\ 4\ \mathbf{do}\ (\mathrm{Qtrasf}_i)_j\colon \mathrm{trigreduce}((\mathrm{tempMat}_i)_j), \mathrm{res}\colon \mathrm{Qtrasf})$

```
(%i6) let(sin(q[1]), s[1]);
```

(%o6) $\sin(q_1) \longrightarrow s_1$

```
(%i7) let(sin(q[2]), s[2]);
```

(%o7) $\sin(q_2) \longrightarrow s_2$

```
(%i8) let(cos(q[1]),c[1]);
```

(%o8) $\cos(q_1) \longrightarrow c_1$

```
(%i9) let(cos(q[2]),c[2]);
```

(%o9) $\cos(q_2) \longrightarrow c_2$

```
(%i10) let(sin(q[1]+q[2]), s[12]);
```

(%o10) $\sin(q_2 + q_1) \longrightarrow s_{12}$

```
(%i11) let(cos(q[1]+q[2]),c[12]);
```

(%o11) $\cos(q_2 + q_1) \longrightarrow c_{12}$

```
(%i12)
```

Cinematica diretta:

```
(%i12) Q[3](q1,q2,L1,L2):=trigsimp(trigrat(trigreduce(trigexpand(Q(q1,0,0,
       L1).Q(q2,0,0,L2)))));
```

(%o12) $Q_3(q1, q2, L1, L2) := \mathrm{trigsimp}(\mathrm{trigrat}(\mathrm{trigreduce}(\mathrm{trigexpand}(Q(q1,0,0,L1) \cdot Q(q2,0,0,L2)))))$

```
(%i13) Qplanare:Q[3](q[1],q[2],L[1],L[2]);
```

(%o13)
$$\begin{pmatrix} \cos(q_2+q_1) & -\sin(q_2+q_1) & 0 & L_2\cos(q_2+q_1) + L_1\cos(q_1) \\ \sin(q_2+q_1) & \cos(q_2+q_1) & 0 & L_2\sin(q_2+q_1) + L_1\sin(q_1) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

```
(%i14) letsimp(Qplanare);
```

(%o14)
$$\begin{pmatrix} c_{12} & -s_{12} & 0 & L_2 c_{12} + L_1 c_1 \\ s_{12} & c_{12} & 0 & L_2 s_{12} + L_1 s_1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

(%i15)

Cinematica inversa

Per effettuare la cinemetica inversa di orientamento sono necessari almeno 3 DOF (gradi di libertà), quindi si effettua solamente la cinematica inversa di posizione per questo tipo di robot. Occorre inizialmente individuare lo spazio di lavoro, le soluzioni generi e singolari ed infine le variabili di giunto q_i.

Dalla cinematica diretta del robot 2DOF sappiamo che il punto x,y viene identificato dal vettore:

$$\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} L_2 \cos(q_2+q_1) + L_1 \cos(q_1) \\ L_2 \sin(q_2+q_1) + L_1 \sin(q_1) \\ 0 \end{pmatrix}$$

$$\begin{pmatrix} x \\ y \end{pmatrix} = R(q_1+q_2) \begin{pmatrix} L_2 \\ 0 \end{pmatrix} + R(q_1) \begin{pmatrix} L_1 \\ 0 \end{pmatrix}$$

Poiché $R(q_1+q_2) = R(q_1)\, R(q_2)$, è possibile mettere in evidenza $R(q_1)$:

$$\begin{pmatrix} x \\ y \end{pmatrix} = R(q_1) \left\{ R(q_2) \begin{pmatrix} L_2 \\ 0 \end{pmatrix} + \begin{pmatrix} L_1 \\ 0 \end{pmatrix} \right\}$$

La matrice di rotazione $R(q_1)$ ha non varia la norma del vettore $\begin{pmatrix} \mathrm{x} \\ \mathrm{y} \end{pmatrix}$ e $R(q_2) \begin{pmatrix} L_2 \\ 0 \end{pmatrix} + \begin{pmatrix} L_1 \\ 0 \end{pmatrix}$. Quindi possiamo imporre che abbiano la stessa norma. In particolare:

$$\left\| \begin{pmatrix} x \\ y \end{pmatrix} \right\|_2 = \left\| R(q_2) \begin{pmatrix} L_2 \\ 0 \end{pmatrix} + \begin{pmatrix} L_1 \\ 0 \end{pmatrix} \right\|_2$$

$$\begin{pmatrix} L_2 & 0 \end{pmatrix} R(q_2)^T R(q_2) \begin{pmatrix} L_2 \\ 0 \end{pmatrix} + \begin{pmatrix} L_1 & 0 \end{pmatrix} \begin{pmatrix} L_1 \\ 0 \end{pmatrix} + 2 \begin{pmatrix} L_1 & 0 \end{pmatrix} R(q_2) \begin{pmatrix} L_2 \\ 0 \end{pmatrix}$$

$$x^2 + y^2 = L_2^2 + L_1^2 + 2\, L_1\, L_2 \cos(q_2)$$

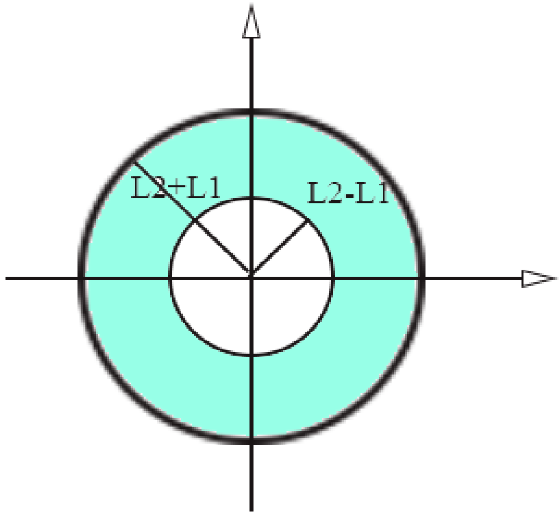
$$\cos(q_2) = \frac{x^2 + y^2 - L_2^2 - L_1^2}{2\, L_1\, L_2}$$

Poiché $-1 \leqslant \cos(q_2) \leqslant 1$, otteniamo infine lo spazio operativo del 2DOF planare:

$$-1 \leqslant \frac{x^2 + y^2 - L_2^2 - L_1^2}{2\, L_1\, L_2} \leqslant 1$$

$$L_2^2 + L_1^2 - 2\, L_1\, L_2 \leqslant x^2 + y^2 \leqslant 2\, L_1\, L_2 + L_2^2 + L_1^2$$

Le disequazioni delimitano due circonferenze di raggio $L_2 + L_1$ e $L_2 - L_1$, in cui la regione valida presa inconsiderazione è quella regione di spazio compresa tra le curve.



A questo punto è possibile determinare l'espressione della variabile di giunto q_2:

$$\cos(q_2) = \frac{x^2 + y^2 - L_2^2 - L_1^2}{2\,L_1\,L_2}$$

$$\sin(q_2) = \pm\sqrt{1 - \cos(q_2)^2} = \pm\sqrt{1 - \frac{x^2 + y^2 - L_2^2 - L_1^2}{2\,L_1\,L_2}}$$

In particolare, si hanno due soluzioni generiche se $\left|\frac{x^2 + y^2 - L_2^2 - L_1^2}{2\,L_1\,L_2}\right| \neq 1$. Nel caso in cui $\left|\frac{x^2 + y^2 - L_2^2 - L_1^2}{2\,L_1\,L_2}\right| = 1$, si ha una soluzione singolare.

A questo punto la quantità $R(q_2)\begin{pmatrix} L_2 \\ 0 \end{pmatrix} + \begin{pmatrix} L_1 \\ 0 \end{pmatrix}$ è nota:

$$\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} \cos(q_1) & -\sin(q_1) \\ \sin(q_1) & \cos(q_1) \end{pmatrix} \begin{pmatrix} L_2\cos(q_2) + L_1 \\ \sin(q_2)\,L_2 \end{pmatrix}$$

$$\begin{pmatrix} \cos(q_1) \\ \sin(q_1) \end{pmatrix} = \frac{1}{(L_2\cos(q_2) + L_1)^2 + L_2^2\sin(q_2)^2} \begin{pmatrix} L_2\cos(q_2) + L_1 & \sin(q_2)\,L_2 \\ -\sin(q_2)\,L_2 & L_2\cos(q_2) + L_1 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}$$

$$= \frac{1}{L_2^2\cos(q_2)^2 + L_1^2 + 2\,L_2 L_1 \cos(q_2) + L_1 + L_2^2\sin(q_2)^2} \begin{pmatrix} (L_2\cos(q_2) + L_1)\,x + \sin(q_2)\,L_2\,y \\ -\sin(q_2)\,L_2\,x + (L_2\cos(q_2) + L_1)\,y \end{pmatrix}$$

Da cui si ottiene la variabile di giunto q_1;

$$q_1 = \mathrm{atan2}\left(\frac{-\sin(q_2)\,L_2\,x + (L_2\cos(q_2) + L_1)\,y}{L_2^2\cos(q_2)^2 + L_1^2 + L_2^2\sin(q_2)^2}, \frac{(L_2\cos(q_2) + L_1)\,x + \sin(q_2)\,L_2\,y}{L_2^2\cos(q_2)^2 + L_1^2 + L_2^2\sin(q_2)^2} \right)$$

Poiché la quantità $L_2^2\cos(q_2)^2 + L_1^2 + L_2^2\sin(q_2)^2 > 0$ è possibile semplificara all'interno della funzione atan2:

$$q_1 = \mathrm{atan2}(\sin(q_2)\,L_2\,x - (L_2\cos(q_2) + L_1)\,y, (L_2\cos(q_2) + L_1)\,x - \sin(q_2)\,L_2\,y)$$

Il problema della cinematica inversa ed, in particolare il problema di orientamento inverso, per il robot 2DOF planare è risolto.

```
Data la cinematica diretta del robot 2DOF planare, si calcola la cinematica
inversa:
```

```
(%i12) Q2DOF:Q[3](%pi/3,%pi/3,10,15);
```

(%o12) $Q_3(\frac{\pi}{3},\frac{\pi}{3},10,15)$

```
(%i13) calculate(x,y,L1,L2):=block([q2plus,q2minus,q1,res],
                                  c2:(x^(2)+y^(2)-L1^(2)-L2^(2))/(2*L1*L2),
                                  s2:sqrt(1-c2^2),

                                  c1Num:combine(expand((L2*c2+L1)*x+s2*L2*y)),
                                  s1Num:combine(expand(-s2*L2*x+(L2*c2+L1)*y)),

       q1Den:L2^(2)*c2^(2)+L1^(2)+L2^(2)*s2^(2)+2*L1*L2*c2,
                             if abs(c2)=1 then print("La soluzione è singolare")
                                  elseif q1Den>0 then(
                                     print("La soluzione non è singolare"),
                                     q1:atan2(s1Num,c1Num),
                                     q2alto:atan2(s2,c2),
                                     q2basso:atan2(-s2,c2),
                                     res:[[q2alto,q1],[q2basso,q1]])
                                  else (
                                     q1:atan2(s1Num/q1Den,c1Num/q1Den),
                                     q2alto:atan2(s2,c2),
                                     q2basso:atan2(-s2,c2),
                                     res:[[q2alto,q1],[q2basso,q1]])



                                  )
```

(%o13) calculate(x, y, $L1$, $L2$) := **block** $\Bigg($[q2plus, q2minus, q1, res], $c2$: $\frac{x^2+y^2-L1^2-L2^2}{2\,L1\,L2}$, $s2$: $\sqrt{1-c2^2}$, c1Num: combine(expand($(L2\,c2+L1)\,x+s2\,L2\,y$)), s1Num: combine(expand($(-s2)\;L2\;x+(L2\;c2+L1)\;y$)), q1Den: $L2^2\;c2^2+L1^2+L2^2\;s2^2+2\;L1\;L2\;c2$, **if** $|c2|=1$ **then** print(La soluzione è singolare) **elseif** q1Den >0 **then** (print(La soluzione non è singolare), q1: atan2(s1Num, c1Num), q2alto: atan2($s2$, $c2$), q2basso: atan2($-s2$, $c2$), res: [[q2alto, q1], [q2basso, q1]]) **else** $\Big($$q$1: atan2$\left(\frac{s1\text{Num}}{q1\text{Den}},\frac{c1\text{Num}}{q1\text{Den}}\right)$, q2alto: atan2($s2$, $c2$), q2basso: atan2($-s2$, $c2$), res: [[q2alto, q1], [q2basso, q1]]$\Big)\Bigg)$

```
(%i14) inv2DOF(x,y,link1,link2):=block([res],


                                circleInt:link1^2+link2^2-2*link1*link2,
                                circleEst:link1^2+link2^2+2*link1*link2,
                              if x^2+y^2>= circleInt and x^2+y^2<= circleEst then
                                (
                                 print("Il punto x,y è nello spazio di lavoro"),
                                 res:calculate(x,y,link1,link2)

                                )
                                else res:"Punto x,y non è ammissibile"

                              )
```

(%o14) inv2DOF(x, y, link1, link2) := **block** ([res], circleInt: link1^2 + link2^2 + (-2) link1 link2, circleEst: link1^2 + link2^2 + 2 link1 link2, **if** $x^2+y^2\geq$ circleInt $\wedge$ $x^2+y^2\leq$ circleEst **then** (print(Il punto x,y è nello spazio di lavoro), res: calculate(x, y, link1, link2)) **else** res: Punto x,y non è ammissibile)

```
(%i15) inv2DOF(-5/2,((25*sqrt(3))/2),10,15);
```

Il punto x,y è nello spazio di lavoro

La soluzione non è singolare

(%o15) $\left[\left[\frac{\pi}{3},\frac{\pi}{3}\right],\left[-\frac{\pi}{3},\frac{\pi}{3}\right]\right]$

(%i16)

Singolarità di velocità

$$\left\{\begin{array}{l} x = L_1c_1 + L_2c_{12} \\ y = L_1s_1 + L_2s_{12} \end{array}\right.$$

$$J = \frac{\delta h}{\delta q} = \left(\begin{array}{cc} -L_1s_1 - L_2s_{12} & -L_2s_{12} \\ L_1c_1 + L_2c_{12} & L_2c_{12} \end{array}\right)$$

$$\det(J) = L_1L_2s_1 \Longrightarrow \det(J) = 0 \rightarrow q_2 = \left\{\begin{array}{l} 0 \\ \pi \end{array}\right.$$

```
(%i7) x:L[2]*cos (q[2]+q[1])+L[1]*cos (q[1]);
```

(%o7) $L_2\cos(q_2+q_1)+L_1\cos(q_1)$

```
(%i8) y:L[2]*sin (q[2]+q[1])+L[1]*sin (q[1]);
```

(%o8) $L_2\sin(q_2+q_1)+L_1\sin(q_1)$

```
(%i9) J:matrix([diff(x,q[1]),diff(x,q[2])],
                [diff(y,q[1]),diff(y,q[2])]
              );
```

(%o9) $\begin{pmatrix} -L_2\sin(q_2+q_1)-L_1\sin(q_1) & -L_2\sin(q_2+q_1) \\ L_2\cos(q_2+q_1)+L_1\cos(q_1) & L_2\cos(q_2+q_1) \end{pmatrix}$

```
(%i12) dJ:trigsimp(trigexpand(determinant(J)));
```

(%o12) $L_1\,L_2\sin(q_2)$

Se $q_2=0$:

```
(%i13) Jq2:subst(q[2]=0,J);
```

(%o13) $\begin{pmatrix} -L_2\sin(q_1)-L_1\sin(q_1) & -L_2\sin(q_1) \\ L_2\cos(q_1)+L_1\cos(q_1) & L_2\cos(q_1) \end{pmatrix}$

```
(%i14) nullspace(Jq2);
```

Proviso: notequal($-L_2\sin(q_1)$, 0)

(%o14) $\mathrm{span}\left(\begin{pmatrix} -L_2\sin(q_1) \\ (L_2+L_1)\sin(q_1) \end{pmatrix}\right)$

Si hanno singolarità per $v \in \mathrm{Im}\left\{\begin{pmatrix} -L_2\sin(q_1) \\ (L_2+L_1)\sin(q_1) \end{pmatrix}\right\}$.

Se $q_2=\pi$

```
(%i15) Jq2:subst(q[2]=%pi,J);
```

(%o15) $\begin{pmatrix} L_2\sin(q_1)-L_1\sin(q_1) & L_2\sin(q_1) \\ L_1\cos(q_1)-L_2\cos(q_1) & -L_2\cos(q_1) \end{pmatrix}$

```
(%i16) nullspace(Jq2);
```

Proviso: notequal$(L_2 \sin(q_1), 0)$

(%o16) $\mathrm{span}\left(\begin{pmatrix} L_2 \sin(q_1) \\ (L_1 - L_2)\sin(q_1) \end{pmatrix}\right)$

Si hanno singolarità per $v \in \mathrm{Im}\left\{\begin{pmatrix} L_2 \sin(q_1) \\ (L_1 - L_2)\sin(q_1) \end{pmatrix}\right\}$

(%i17)