Calcolare asse ed angolo tramite la formula di Rodriguez

Il problema è quello di determinare l'asse e l'angolo di rotazione data una matrice di rotazione R attraverso la formula di Rodriguez.

N.B.:La notazione c,s equivale rispettivamente a cos(ϑ), sin(ϑ).

Questo problema è detto problema orientamento inverso che consiste nel determinare, data una matrice di rotazione, asse e angolo:

$$v, \theta \longleftarrow R$$

Inizialmente occorre verificare se la matrice data è una matrice di rotazione. Le condizioni che devono essere soddisfatte affinché R sia di rotazione sono:

$$R\,R^T = I$$
$$\det(R) = 1$$

O in maniera equivalente:

$$R^T R = I$$
$$\det(R) = 1$$

(1)Se la matrice R è di rotazione, l'asse di rotazione viene determinato dall'autovettore relativo all'autovalore 1. In particolare, occorre imporre:

$$(\lambda I - R)\,v = R\begin{pmatrix} a \\ b \\ c \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} \text{con } \lambda = 1$$

e trovare i coefficienti a, b, c di v affinché il prodotto $(I - R)\,v = \mathbf{0}$ e $v \neq \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$.

Il vettore v normalizzato risultante è l'asse di rotazione.

(2)In aggiunta, per calcolare l'angolo di rotazione, occorre risolvere il seguente sistema:

$$\begin{cases} R = I + S^2(v)\,(1 - c) + S(v)\,s \\ R^T = I + S^2(v)\,(1 - c) - S(v)\,s \end{cases}$$

$$(a)\frac{R - R^T}{2} = S(v)\,s$$

$$(b)\frac{R + R^T}{2} - I = S^2(v)\,(1 - c)$$

L'angolo ϑ viene determinata dalla seguente formula:

$$(c)\theta = \text{atan2}(s, c)$$

In cui s viene calcolato prendendo un qualsiasi elemento $a_{i,j} \neq 0$ con $a_{i,j} \in S(v)$ con il corrispettivo elemento $b_{i,j}$ nella matrice $(a) \frac{R - R^T}{2}$ e risolvendo l'equazione:

$$b_{i,j} = a_{i,j} \sin(\theta)$$

Invece c, viene calcolato prendendo un qualsiasi elemento $c_{i,j} \neq 0$ con $c_{i,j} \in S^2(v)$ con il corrispettivo elemento $d_{i,j}$ nella matrice $(b) \frac{R + R^T}{2} - I$ e risolvendo l'equazione:

$$c_{i,j}(1 - \cos(\theta)) = d_{i,j}$$

La funzione isRotation(M) prende in input una matrice M e verifica se la matrice data è di rotazione. In particolare, verifica se $M M^T = I$(i) e $\det(M) = 1$(ii).
Inoltre, se la matrice è simbolica cioè dipendente dalla variabile theta, restituisce le condizioni per cui la matrice M è di rotazione; altrimenti, la matrice non è di rotazione.

```
(%i1) isRotation(M):=block([res],
                          I:ident(3),
                          MMT:trigsimp(expand(M.transpose(M))),
                          detM:trigsimp(expand(determinant(M))),

                          if MMT=I  and detM=1
                             then(

                                  return(res:1)
                                  )

                          else(

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

(%o1) isRotation(M) := **block** ([res], I: ident(3), MMT: trigsimp(expand($M \cdot$ transpose(M))), detM: trigsimp(expand(determinant(M))), **if** MMT $= I \wedge$ detM $= 1$ **then** return(res: 1) **else** res: R is not rotation matrix)

La funzione axes(M) prende in input una matrice $M = \text{adj}(I - R)$ di rotazione e ne calcola il corrispondente asse di rotazione e,utilizzando la procedura del punto (1), selezione come asse di rotazione una colonna non nulla della matrice M.

```
(%i2) axes(M):=block([res],
                     columns:transpose(M),
                    res:zeromatrix(3,1),
                     for i:1 thru length(columns) do(
                       if(columns[i][1]# 0 or columns[i][2]#0 or
      columns[i][3]#0)
                          then ( return(m: transpose(columns[i])))

                     ),res:m

                     )
```

(%o2) axes(M) := **block** ([res], columns: transpose(M), res: zeromatrix(3, 1),

for i **thru** length(columns) **do if** $(\text{columns}_i)_1 \neq 0 \vee (\text{columns}_i)_2 \neq 0 \vee (\text{columns}_i)_3 \neq 0$ **then** return(m: transpose(columns_i)), res: m)

La funzione skewMatrix(x) prende in input un asse x di rotazione normalizzato e ne calcola la corrispettiva matrice antisimmetrica $S_v(\theta)$.

```
(%i3) skewMatrix(x):=block([res],
                             S: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)*x[3-remainder(i+j,3)][1],
                                          S[i][j]:temp,
                                          S[j][i]:-temp

                                           )
                                  )
                              ),
                              res:S
                              )
```

(%o3) skewMatrix(x) := **block** ([res], S: 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: $(-1)^{j-i}\,(x_{3-\text{remainder}(i+j,3)})_1$, $(S_i)_j$: temp, $(S_j)_i$: $-$temp), res: S)

La funzione sinRotation(skewMat, RRT2) prende in input una matrice antisimmetrica $S_v(\theta)$ e la matrice $\frac{R-R^T}{2}$ ricavata dalla risoluzione del sistema della formula di Rodriguez e risolvere l'equazione (a).

```
(%i4) sinRotation(skewMat,RRT2):=block([res],

                                for i:1 thru 3 do(
                                   for j:1 thru 3 do(
                                      a:skewMat[i][j],

                                      if a# 0
                                         then (b:RRT2[i][j],

                                               return(

                                          value:b/a
                                          ))
                                      )
                                  ),res:value


                                )
```

(%o4) sinRotation(skewMat, RRT2) := **block** ([res], **for** i **thru** 3 **do for** j **thru** 3 **do** (a:

$(\mathrm{skewMat}_i)_j$, **if** $a \neq 0$ **then** $\left(b\colon (\mathrm{RRT2}_i)_j, \mathrm{return}\left(\mathrm{value}\colon \frac{b}{a}\right)\right)\Big)$, res: value$\Big)$

La funzione corRotation(x,y) prende in input una matrice antisimmetrica $S_v(\theta)$ e la matrice $\frac{R+R^T}{2} - I$ ricavata dalla soluzione del sistema della formula di Rodriguez e risolve l'equazione (b).

```
(%i5) cosRotation(x,y):=block([res],

                              for i:1 thru 3 do(
                                 for j:1 thru 3 do(
                                    c:x[i][j],
                                    if(c#0) then(
                                               d:y[i][j],

                                               return(t:(c-d)/c))
                                 )
                              ),
                               res:t

                              )
```

(%o5) $\mathrm{cosRotation}(x, y) := \mathbf{block}\Big([\mathrm{res}], \mathbf{for}\ i\ \mathbf{thru}\ 3\ \mathbf{do}\ \mathbf{for}\ j\ \mathbf{thru}\ 3\ \mathbf{do}\ \Big(c\colon (x_i)_j, \mathbf{if}\ c \neq 0\ \mathbf{then}\ \left(d\colon (y_i)_j, \mathrm{return}\left(t\colon \frac{c-d}{c}\right)\right)\Big), \mathrm{res}\colon t\Big)$

La funzione degree(v,M) prende in input un asse di rotazione v e una matrice M. Invoca le funzioni sinRotation e cosRotation per salvare nelle variabili sinR e cosR i valori rispettivamente di $\sin(\theta)$ e $\cos(\theta)$. Infine applica (c) per restituire il valore di ϑ.

```
(%i6) degree(v,M):=block([sinR,cosR,res],
                          S:skewMatrix(v),
                          I:ident(3),
                          RRsin:trigsimp((M-transpose(M))*1/2),
                          RRcos:trigsimp(((M+transpose(M))*1/2)-I),
                          sinR:sinRotation(S,RRsin),

                          SS:S.S,
                          cosR:cosRotation(SS,RRcos),
                          res:atan2(expand(sinR),expand(cosR))

                            )
```

(%o6) $\mathrm{degree}(v, M) := \mathbf{block}\Big([\mathrm{sinR}, \mathrm{cosR}, \mathrm{res}], S\colon \mathrm{skewMatrix}(v), I\colon \mathrm{ident}(3), \mathrm{RRsin}\colon \mathrm{trigsimp}\left(\frac{(M - \mathrm{transpose}(M))\,1}{2}\right), \mathrm{RRcos}\colon \mathrm{trigsimp}\left(\frac{(M + \mathrm{transpose}(M))\,1}{2} - I\right), \mathrm{sinR}\colon \mathrm{sinRotation}(S, \mathrm{RRsin}), \mathrm{SS}\colon S \cdot S, \mathrm{cosR}\colon \mathrm{cosRotation}(\mathrm{SS}, \mathrm{RRcos}), \mathrm{res}\colon \mathrm{atan2}(\mathrm{expand}(\mathrm{sinR}), \mathrm{expand}(\mathrm{cosR}))\Big)$

La funzione axesDegree(R) prende in input una matrice R, simbolica e non, per restituire il corrispondente asse e angolo di rotazione, verificando se l'effettiva matrice in input sia di rotazione. In caso contrario, restituisce errore.

```
(%i7)  axesDegree(R):=block([v,theta,res],
                          isRot:isRotation(R),
                          if isRot=1 then (
                             I:ident(3),
                             adjR:adjoint(I-R),
                             v:axes(adjR),
                             vNorm:v/sqrt(v.v),
                             theta:degree(vNorm,R),
                             print("Axe, degree"),
                             res:[vNorm,theta]



                          )
                          else res:"R is not rotation matrix"



                         )
```

(%o7) $\mathrm{axesDegree}(R) := \mathbf{block}\left([v, \vartheta, \mathrm{res}], \mathrm{isRot}\colon \mathrm{isRotation}(R), \mathbf{if}\ \mathrm{isRot} = 1\ \mathbf{then}\left(I\colon \mathrm{ident}(3), \mathrm{adjR}\colon \mathrm{adjoint}(I - R), v\colon \mathrm{axes}(\mathrm{adjR}), \mathrm{vNorm}\colon \frac{v}{\sqrt{v \cdot v}}, \vartheta\colon \mathrm{degree}(\mathrm{vNorm}, R), \mathrm{print}(\text{Axe, degree}), \mathrm{res}\colon [\mathrm{vNorm}, \vartheta]\right)\ \mathbf{else}\ \mathrm{res}\colon \text{R is not rotation matrix}\right)$

Asse e angolo corrispettivi ad una rotazione lungo l'asse x:

```
(%i8) R:matrix([1,0,0],[0,-1,0],[0,0,-1])
```

(%o8) $\begin{pmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{pmatrix}$

```
(%i9) axesDegree(R)
```

Axe, degree

(%o9) $\left[\begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}, \pi\right]$

Asse e angolo corrispettivi ad una rotazione attorno asse y:

```
(%i10) R:matrix([0,0,1],[0,1,0],[-1,0,0]);
```

(%o10) $\begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ -1 & 0 & 0 \end{pmatrix}$

```
(%i11) axesDegree(R);
```

Axe, degree

(%o11) $\left[\begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}, \frac{\pi}{2}\right]$

Asse e angolo corrispettivi ad una rotazione attorno l'asse z:

```
(%i12) R:matrix([0,-1,0],[1,0,0],[0,0,1]);
```

$$(\%o12)\quad \begin{pmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

```
(%i13) axesDegree(R);
```

Axe, degree

$$(\%o13)\quad \left[\begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}, \frac{\pi}{2}\right]$$

Matrice simbolica lungo x:

```
(%i14) R[x](theta) := matrix([1,0,0],
                        [0,cos(theta),-sin(theta)],
                        [0,sin(theta), cos(theta)]);
```

$$(\%o14)\quad R_x(\vartheta) := \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos(\vartheta) & -\sin(\vartheta) \\ 0 & \sin(\vartheta) & \cos(\vartheta) \end{pmatrix}$$

```
(%i15) axesDegree(R[x](theta));
```

Axe, degree

$$(\%o15)\quad \left[\begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}, \mathrm{atan2}(\sin(\vartheta), \cos(\vartheta))\right]$$

Matrice simbolica lungo y:

```
(%i16) R[y](theta) := matrix([cos(theta),0,sin(theta)],
                        [0,1,0],
                        [-sin(theta),0, cos(theta)]);
```

$$(\%o16)\quad R_y(\vartheta) := \begin{pmatrix} \cos(\vartheta) & 0 & \sin(\vartheta) \\ 0 & 1 & 0 \\ -\sin(\vartheta) & 0 & \cos(\vartheta) \end{pmatrix}$$

```
(%i17) axesDegree(R[y](theta));
```

Axe, degree

$$(\%o17)\quad \left[\begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}, \mathrm{atan2}(\sin(\vartheta), \cos(\vartheta))\right]$$

Matrice simbolica lungo z:

```
(%i18) R[z](theta) := matrix([cos(theta),-sin(theta),0],
                        [sin(theta),cos(theta),0],
                        [0,0, 1]);
```

$$(\%o18)\quad R_z(\vartheta) := \begin{pmatrix} \cos(\vartheta) & -\sin(\vartheta) & 0 \\ \sin(\vartheta) & \cos(\vartheta) & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

```
(%i19) axesDegree(R[z](theta));
```

Axe, degree

$$(\%o19)\quad \left[\begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}, \mathrm{atan2}(\sin(\vartheta), \cos(\vartheta))\right]$$

Matrice non di rotazione:

```
(%i20) R:matrix([1,1,0],[0,1,1],[1,1,1]);
```

(%o20) $\begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}$

```
(%i21) axesDegree(R);
```

(%o21) R is not rotation matrix

Matrice razionale di rotazione:

```
(%i22) R:matrix([1,0,0],[0,sqrt(3)/2,-1/2],[0,1/2,sqrt(3)/2]);
```

(%o22) $\begin{pmatrix} 1 & 0 & 0 \\ 0 & \frac{\sqrt{3}}{2} & -\frac{1}{2} \\ 0 & \frac{1}{2} & \frac{\sqrt{3}}{2} \end{pmatrix}$

```
(%i23) axesDegree(R);
```

Axe, degree

(%o23) $\left[\begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}, \frac{\pi}{6}\right]$

```
(%i24) R:matrix([0,-1,0],[1,0,0],[0,0,1]);
```

(%o24) $\begin{pmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}$

```
(%i25) axesDegree(R)
```

Axe, degree

(%o25) $\left[\begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}, \frac{\pi}{2}\right]$

Matrice di rotazione risultante dalla rotazione Ra sull'asse y e Rb sull'asse z;

$$R_a = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ -1 & 0 & 0 \end{pmatrix}; R_b = \begin{pmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

$$R_a\, R_b = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ -1 & 0 & 0 \end{pmatrix} \begin{pmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 0 & 0 & 1 \\ -1 & 0 & 0 \\ 0 & -1 & 0 \end{pmatrix}$$

```
(%i26) R[ab]:matrix([0,1,0],[0,0,-1],[-1,0,0]);
```

(%o26) $\begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & -1 \\ -1 & 0 & 0 \end{pmatrix}$

```
(%i27) axesDegree(R[ab])
```

Axe, degree

(%o27) $$\left[\begin{pmatrix}\frac{1}{\sqrt{3}}\\ \frac{1}{\sqrt{3}}\\ -\frac{1}{\sqrt{3}}\end{pmatrix}, \frac{2\,\pi}{3}\right]$$

(%i28)