Esempio:

```
RotationMatrix[θ, {0, 0, 1}] // MatrixForm
```

$$\begin{pmatrix} \text{Cos}[\theta] & -\text{Sin}[\theta] & 0 \\ \text{Sin}[\theta] & \text{Cos}[\theta] & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Rotazione rigida delle posizioni delle ancore, ricostruite tramite Multidimensional Scaling

Avendo come set di punti da ricostruire:
A0 = {0, 0, 0}, A1 = {0, 2.9810, 0}, A2 = {4.7220, 1.7210, 0}, A3 = {0.1430, 1.6420, 2.2240}
è stato applicato l'algoritmo di Multidimensional Scaling in Matlab, che ricevendo in ingresso le distanze euclidee relative tra i punti, ordinate, ha restituito in uscita come coordinate per i punti ricostruiti:
A0 = {-1.1749   1.6474   -0.4577}, A1 = {-1.0275 -1.2752 -1.0259}, A2 = {3.5511 0.0252 0.0835}, A3 = {-1.3487 -0.3973 1.4001},
che traslati di {-1.1749 1.6474 -0.4577} sono:
A0 = {0   0   0} A1 = {0.1474 -2.9226 -0.5682}, A2 = {4.7259 -1.6222 0.5412}, A3 = {-0.1738 -2.0447   1.8578}
Si vuole individuare una matrice di rotazione che riporti le coordinate dei punti ai valori originali ovvero espresse nel sistema di riferimento di interesse.

```
A0 = {0, 0, 0};
```

Posizione reale di A1:

```
A1 = {0, 2.9810, 0.0};
```

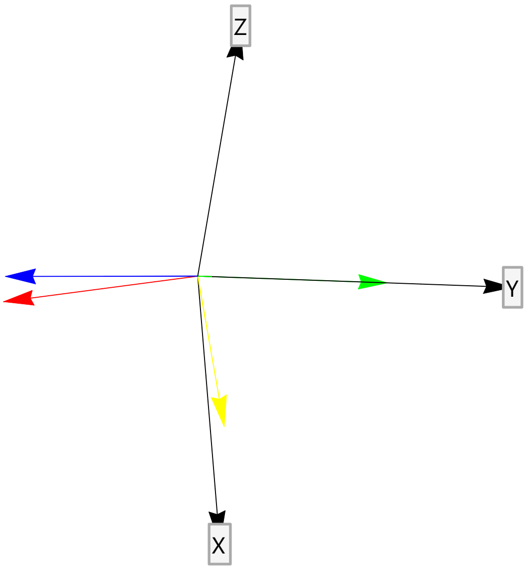
Posizione ricostruita di A1, traslando tutti i punti per avere A0 nell'origine.

```
A1R = {0.1474, -2.9226, -0.5682};
```

Considero il punto di stesse coordinate x, y e con z = 0 per individuare l'angolo della prima rotazione.

```
A = {0.1474, -2.9226, 0};
```

```
Graphics3D[{Black, Arrow[{{0, 0, 0}, {5, 0, 0}}], Black,
  Arrow[{{0, 0, 0}, {0, 5, 0}}], Black, Arrow[{{0, 0, 0}, {0, 0, 5}}],
  Green, Arrow[{A0, A1}], Red, Arrow[{A0, A1R}], Blue,
  Arrow[{A0, A}] , Yellow, Arrow[{{0, 0, 0}, {2.9445, 0.2055, 0}}] ,
  Text[Panel["X", FrameMargins → 0], {5, 0, 0}],
  Text[Panel["Y", FrameMargins → 0], {0, 5, 0}],
  Text[Panel["Z", FrameMargins → 0], {0, 0, 5}]},
 AspectRatio -> Automatic, Boxed → False]
```




```
OA1R = Norm[A1R];
```

```
OA = Norm[A]
```

2.92631

```
(*α= ArcCos[OA/OA1R]*)
```

```
axis = Cross[A, A1R]
```

{1.66062, 0.0837527, 0.}

```
axisN = Normalize[axis]
```

{0.998731, 0.0503705, 0.}

Costruzione della matrice di rotazione asse angolo:

```
nSkew = {{0, -Part[axisN, 3], Part[axisN, 2]},
   {Part[axisN, 3], 0, -Part[axisN, 1]}, {-Part[axisN, 2], Part[axisN, 1], 0}};
```

```
nKro = {{Part[axisN, 1] * Part[axisN, 1],
    Part[axisN, 1] * Part[axisN, 2], Part[axisN, 1] * Part[axisN, 3]} ,
   {Part[axisN, 2] * Part[axisN, 1], Part[axisN, 2] * Part[axisN, 2],
    Part[axisN, 2] * Part[axisN, 3]}, { Part[axisN, 3] * Part[axisN, 1],
    Part[axisN, 3] * Part[axisN, 2], Part[axisN, 3] * Part[axisN, 3]}};
```

```
α = ArcCos[(OA^2 + OA1R^2 - (0.5682)^2) / (2 * OA * OA1R)]
```

0.191783

```
Raα = nKro + (IdentityMatrix[3] - nKro) * Cos[-α] + nSkew * Sin[-α]
```

```
{{0.999953, 0.000922321, -0.00960109},
 {0.000922321, 0.981713, 0.190367}, {0.00960109, -0.190367, 0.981666}}
```

```
R1 = RotationMatrix[-α, axis]
```

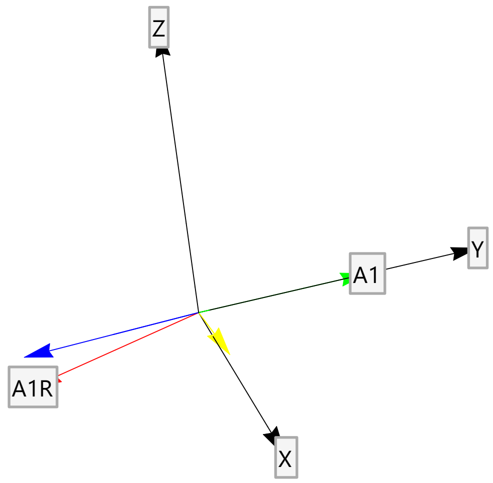
```
{{0.999953, 0.000922321, -0.00960109},
 {0.000922321, 0.981713, 0.190367}, {0.00960109, -0.190367, 0.981666}}
```

Rotazione oraria di α attorno a axis

```
A1Rproj = Dot[R1, A1R]
```

```
{0.150153, -2.97718, 0.}
```

```
Graphics3D[{Black, Arrow[{{0, 0, 0}, {5, 0, 0}}], Black,
  Arrow[{{0, 0, 0}, {0, 5, 0}}], Black, Arrow[{{0, 0, 0}, {0, 0, 5}}], Green,
  Arrow[{A0, A1}], Red, Arrow[{A0, A1R}], Blue, Arrow[{A0, A1Rproj}] , Yellow,
  Arrow[{{0, 0, 0}, axis}] , Text[Panel["X", FrameMargins → 0], {5, 0, 0}],
  Text[Panel["Y", FrameMargins → 0], {0, 5, 0}],
  Text[Panel["Z", FrameMargins → 0], {0, 0, 5}],
  Text[Panel["A1R", FrameMargins → 0], A1R],
  Text[Panel["A1", FrameMargins → 0], A1]},
 AspectRatio -> Automatic, Boxed → False]
```



Rimane da individuare l'angolo di cui ruotare attorno a Z, allineandosi al vettore verde. Proietto A su X.

```
(*B = {0,-2.9226,0};*)
```

```
B = {0, -2.97718, 0};
```

```
OB = Norm[B]
```

```
2.97718
```

```
θ = ArcCos[OB / Norm[A1Rproj]]
```

```
0.0504166
```

```
θ = ArcCos[(OB^2 + (Norm[A1Rproj])^2 - (0.150153)^2) / (2 * OB * Norm[A1Rproj])]
```

```
0.0503919
```

```
Rz = RotationMatrix[Pi - θ, {0, 0, 1}];
```

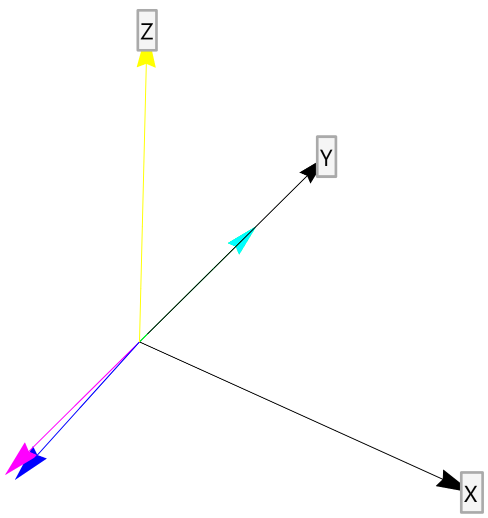
```
A1fin = Dot[Rz, A1Rproj]
```

$\{9.53324\times10^{-8}, 2.98097, 0.\}$

```
A1
```

```
{0, 2.981, 0.}
```

```
Graphics3D[{Black, Arrow[{{0, 0, 0}, {5, 0, 0}}], Black,
  Arrow[{{0, 0, 0}, {0, 5, 0}}], Yellow, Arrow[{{0, 0, 0}, {0, 0, 5}}], Green,
  Arrow[{A0, A1}], (*Red,Arrow[{A0,A1R}],*)Blue, Arrow[{A0, A1Rproj}] ,
  (*Yellow,Arrow[{{0,0,0},axis}] ,*) Magenta, Arrow[{{0, 0, 0}, B}], Cyan,
  Arrow[{{0, 0, 0}, A1fin}], Text[Panel["X", FrameMargins → 0], {5, 0, 0}],
  Text[Panel["Y", FrameMargins → 0], {0, 5, 0}],
  Text[Panel["Z", FrameMargins → 0], {0, 0, 5}]},
 AspectRatio -> Automatic, Boxed → False]
```




```
EuclideanDistance[A1, A1fin]
```

```
0.0000322445
```

Verifichiamo se ruotando rigidamente gli altri punti per la matrice di rotazione complessiva si ottengano dai punti ricostruiti i punti reali.

```
A2 = {4.7220, 1.7210, 0};
```

```
A2R = {4.7260, -1.6222, 0.5412};
```

```
A2RR = Rz.R1.A2R
```

```
{-4.63829, 1.72097, 0.885466}
```

Manca un’ultima rotazione attorno l’asse individuato dal prodotto vettoriale tra il vettore passante dall’origine al punto A2RR e il versore dell’asse y. Questo corrisponde all’arcos della componente z dell’asse individuato col prodotto vettore.

```
axis12 = Normalize[Cross[A2RR, {0, 1, 0}]]
```

```
{-0.187517, 0., -0.982261}
```

Questo angolo è descritto dal vettore congiungente A2R e il piano xy. Voglio portare a 0 la componente z

```
μ = ArcCos[Part[axis12, 3]]
```

```
2.95296
```

```
Ry = RotationMatrix[μ, {0, 1, 0}]
```

```
{{-0.982261, 0., 0.187517}, {0., 1., 0.}, {-0.187517, 0., -0.982261}}
```

```
A2fin = Ry.Rz.R1.A2R
```

$\{4.72205, 1.72097, 1.22125 \times 10^{-15}\}$

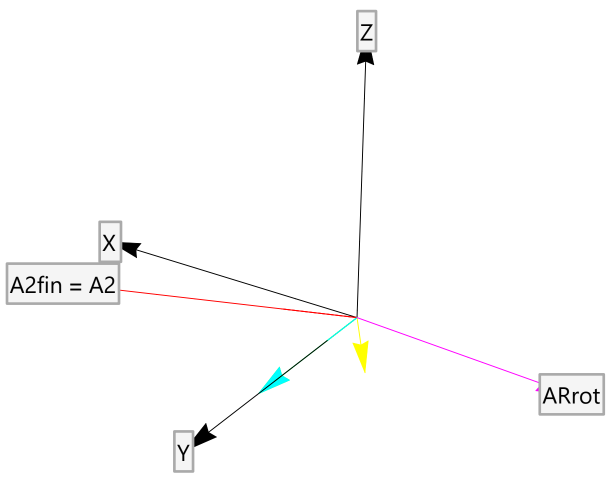
```
A3 = {0.1430, 1.6420, 2.2240};
```

```
A3R = {-0.1738, -2.0447, 1.8578};
```

```
A3fin = Ry.Rz.R1.A3R
```

```
{0.142994, 1.64196, -2.22395}
```

```
Graphics3D[{Black, Arrow[{{0, 0, 0}, {5, 0, 0}}], Black,
  Arrow[{{0, 0, 0}, {0, 5, 0}}], Black, Arrow[{{0, 0, 0}, {0, 0, 5}}],
  Green, Arrow[{A0, A1}], Yellow, Arrow[{{0, 0, 0}, axis12}] , Magenta,
  Arrow[{{0, 0, 0}, A2RR}], Cyan, Arrow[{{0, 0, 0}, A1fin}], Red,
  Arrow[{A0, A2fin}], Text[Panel["X", FrameMargins → 0], {5, 0, 0}],
  Text[Panel["Y", FrameMargins → 0], {0, 5, 0}],
  Text[Panel["Z", FrameMargins → 0], {0, 0, 5}],
  Text[Panel["ARrot", FrameMargins → 0], A2RR],
  Text[Panel["A2fin = A2", FrameMargins → 0], A2fin] },
 AspectRatio -> Automatic, Boxed → False]
```



Rotazione rigida del SET4 di punti, ricostruiti tramite Multidimensional Scaling

```
C0 = {0, 0, 0};
```

Posizione reale di A1.

```
C1 = {0, 5.1230, 0};
```

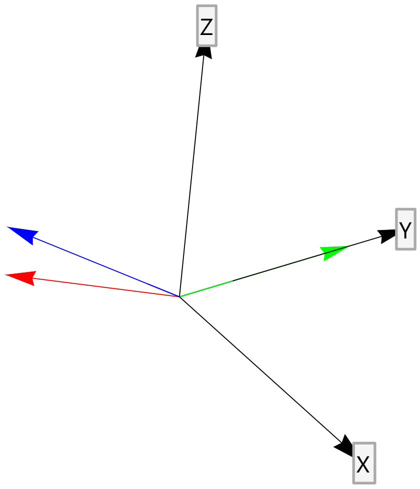
Posizione ricostruita di A1, avendo traslato tutti i punti per avere A0 nell’origine.

```
C1R = {-4.4758, -2.0700, -1.3884};
```

Considero il punto di stesse coordinate x, y e con z = 0.

```
CC =  {-4.4758, -2.0700 , 0};
```

```
Graphics3D[{Black, Arrow[{{0, 0, 0}, {7, 0, 0}}], Black,
  Arrow[{{0, 0, 0}, {0, 7, 0}}], Black, Arrow[{{0, 0, 0}, {0, 0, 7}}],
  Green, Arrow[{C0, C1}], Red, Arrow[{C0, C1R}], Blue,
  Arrow[{C0, CC}], Text[Panel["X", FrameMargins → 0], {7, 0, 0}],
  Text[Panel["Y", FrameMargins → 0], {0, 7, 0}],
  Text[Panel["Z", FrameMargins → 0], {0, 0, 7}]},
 AspectRatio -> Automatic, Boxed → False]
```




```
OC1R = Norm[C1R];
```

```
OC  = Norm[CC]
```

4.9313

```
(*γ= ArcCos[OC/OC1R]*)
```

```
γ = ArcCos[(OC^2 + OC1R^2 - (-1.3884)^2) / (2 * OC * OC1R)]
```

0.274444

```
axis2  = Cross[CC, C1R]
```

{2.87399, -6.2142, 0.}

```
R2  = RotationMatrix[-γ, axis2]
```

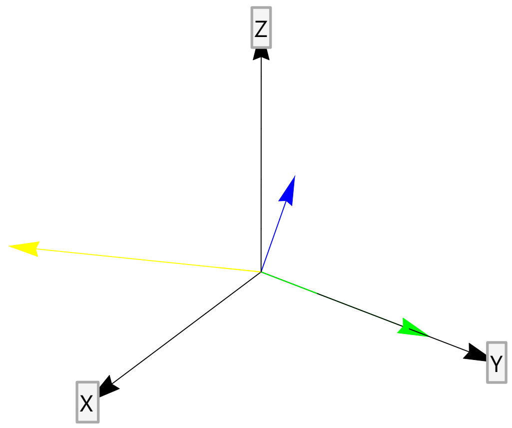
{{0.96917, -0.0142584, 0.245979},
 {-0.0142584, 0.993406, 0.113762}, {-0.245979, -0.113762, 0.962576}}

Rotazione oraria di γ attorno a axis2

```
C1Rproj  = Dot[R2, C1R]
```

$\{-4.64981, -2.15048, 8.88178 \times 10^{-16}\}$

```
Graphics3D[{Black, Arrow[{{0, 0, 0}, {7, 0, 0}}], Black,
  Arrow[{{0, 0, 0}, {0, 7, 0}}], Black, Arrow[{{0, 0, 0}, {0, 0, 7}}], Green,
  Arrow[{C0, C1}], (*Red,Arrow[{C0,C1R}],*)Blue, Arrow[{C0, C1Rproj}] , Yellow,
  Arrow[{{0, 0, 0}, axis2}], , Text[Panel["X", FrameMargins → 0], {7, 0, 0}],
  Text[Panel["Y", FrameMargins → 0], {0, 7, 0}],
  Text[Panel["Z", FrameMargins → 0], {0, 0, 7}] },
 AspectRatio -> Automatic, Boxed → False]
```



Rimane da individuare l'angolo di cui ruotare attorno a Z, allineandosi al vettore verde. Proietto C1Rproj su X.

```
DD = {0, -2.15048, 0};

OD = Norm[DD]
```

2.15048

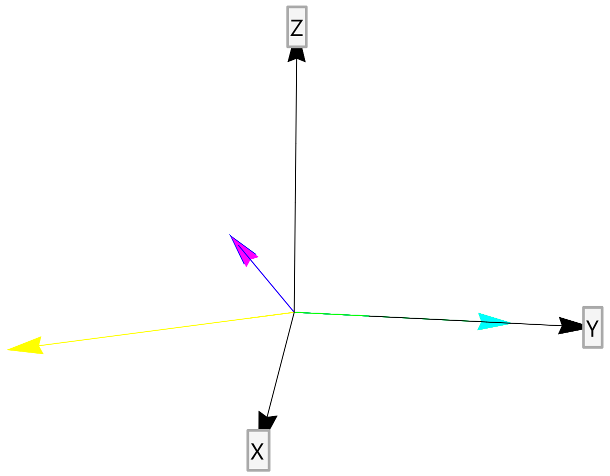
```
(*ϕ= ArcCos[OD/Norm[C1Rproj]]*)

(*ϕ=ArcCos[(OD^2+(Sqrt[4.64981^2+2.15048^2])^2-(-4.46032)^2)/
   (2*OD*Sqrt[4.64981^2+2.15048^2])]*)

ϕ = ArcCos[(OD^2 + (Norm[C1Rproj])^2 - (-4.64981)^2) / (2 * OD * Norm[C1Rproj])]
```

1.1376

```
Rz2 = RotationMatrix[Pi + ϕ, {0, 0, 1}];

C1fin = Dot[Rz2, C1Rproj]
```

$\{0.0000114296, 5.12302, 8.88178\times 10^{-16}\}$

```
Graphics3D[{Black, Arrow[{{0, 0, 0}, {7, 0, 0}}], Black,
  Arrow[{{0, 0, 0}, {0, 7, 0}}], Black, Arrow[{{0, 0, 0}, {0, 0, 7}}], Green,
  Arrow[{C0, C1}], (*Red,Arrow[{C0,C1R}],*)Blue, Arrow[{C0, C1Rproj}] ,
  Yellow, Arrow[{{0, 0, 0}, axis2}] , Magenta, Arrow[{{0, 0, 0}, CC}], Cyan,
  Arrow[{{0, 0, 0}, C1fin}], Text[Panel["X", FrameMargins → 0], {7, 0, 0}],
  Text[Panel["Y", FrameMargins → 0], {0, 7, 0}],
  Text[Panel["Z", FrameMargins → 0], {0, 0, 7}]},
 AspectRatio -> Automatic, Boxed → False]
```




```
EuclideanDistance[C1, C1fin]
```

0.0000235696

Verifichiamo se ruotando rigidamente gli stessi punti per la matrice di rotazione complessiva si ottengano dai punti ricostruiti i punti reali.

```
C2 = {-4.7220, -3.9870, 0};

C2R = {5.7796, -1.7342, -1.3349};

C2RR = Rz2.R2.C2R
```

{-4.00011, -3.98693, -2.50932}

```
axis22 = Normalize[Cross[C2RR, {0, 1, 0}]]
```

{0.531406, 0., -0.847117}

```
ν = ArcCos[Part[axis22, 3]]
```

2.58133

```
Ry2 = RotationMatrix[-ν, {0, 1, 0}]
```

{{-0.847117, 0., -0.531406}, {0., 1., 0.}, {0.531406, 0., -0.847117}}

```
C2fin = Ry2.Rz2.R2.C2R
```

$\{4.72203, -3.98693, -2.22045\times10^{-16}\}$

```
C3 = {0, 0, 2.2240};
```

```
C3R = {-0.0327 , 1.2870, -1.8135};
```

```
C3fin = Ry2.Rz2.R2.C3R
```

$\{5.31373\times10^{-6}, 0.000023141, 2.22401\}$

```
DD - C1Rproj
```

$\{4.64981, -3.91568\times10^{-7}, -8.88178\times10^{-16}\}$

Rotazione rigida del SET5 di punti, ricostruiti tramite Multidimensional Scaling

```
J0 = {0, 0, 0};
```

Posizione reale di A1.

```
J1 = {0, 1.15, 0};
```

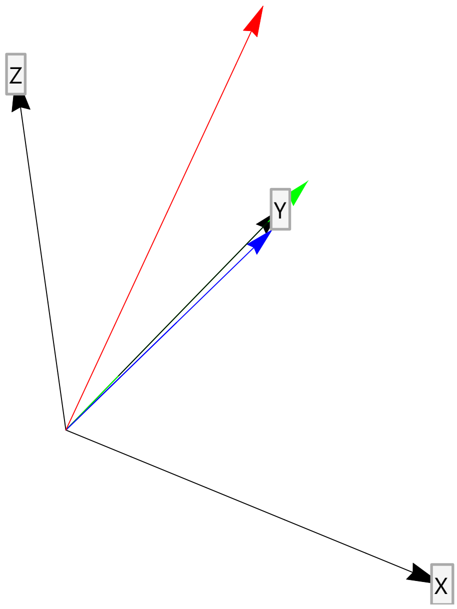
Posizione ricostruita di A1, avendo traslato tutti i punti per avere A0 nell'origine.

```
J1R = {0.0281, 0.9074, 0.7060};
```

Considero il punto di stesse coordinate x, y e con z = 0.

```
J =  {0.0281, 0.9074, 0.0};
```

```
Graphics3D[{Black, Arrow[{{0, 0, 0}, {1, 0, 0}}], Black,
  Arrow[{{0, 0, 0}, {0, 1, 0}}], Black, Arrow[{{0, 0, 0}, {0, 0, 1}}],
  Green, Arrow[{J0, J1}], Red, Arrow[{J0, J1R}], Blue,
  Arrow[{J0, J}], Text[Panel["X", FrameMargins → 0], {1, 0, 0}],
  Text[Panel["Y", FrameMargins → 0], {0, 1, 0}],
  Text[Panel["Z", FrameMargins → 0], {0, 0, 1}]},
 AspectRatio -> Automatic, Boxed → False]
```




```
OJ1R = Norm[J1R];
```

```
OJ = Norm[J]
```

0.907835

```
(*γ= ArcCos[OC/OC1R]*)
```

```
β = ArcCos[(OJ^2 + OJ1R^2 - (0.7060)^2) / (2 * OJ * OJ1R)]
```

0.660979

```
axis3 = Cross[J, J1R]
```

{0.640624, -0.0198386, 0.}

```
R3 = RotationMatrix[-β, axis3]
```

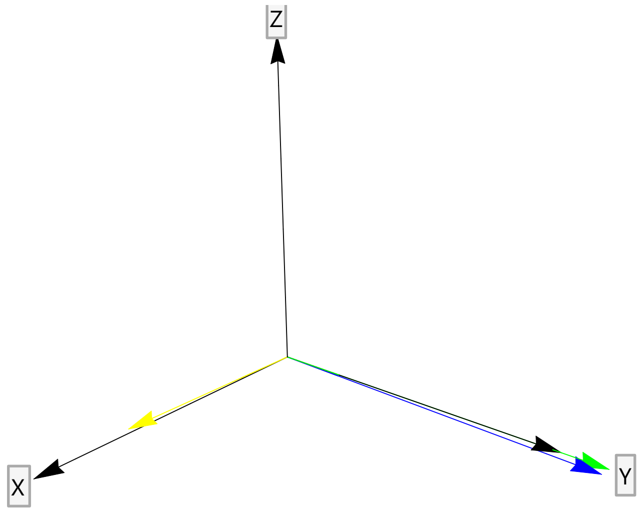
{{0.999798, -0.00651578, 0.0190016},
 {-0.00651578, 0.789594, 0.613596}, {-0.0190016, -0.613596, 0.789392}}

Rotazione oraria di β attorno a axis3

```
J1Rproj = Dot[R3, J1R]
```

{0.035597, 1.14949, 0.}

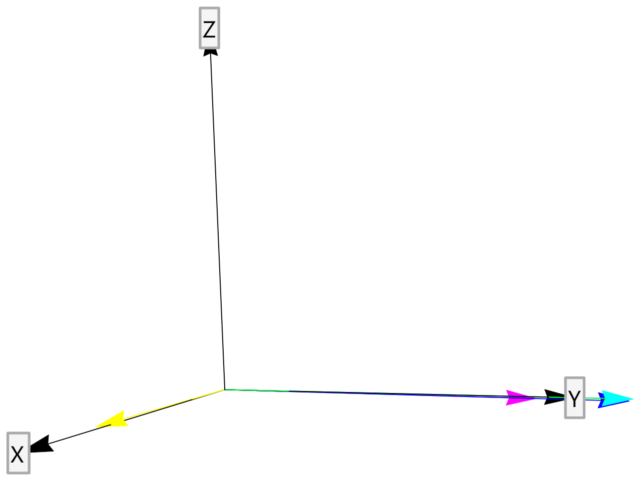
```
Graphics3D[{Black, Arrow[{{0, 0, 0}, {1, 0, 0}}], Black,
  Arrow[{{0, 0, 0}, {0, 1, 0}}], Black, Arrow[{{0, 0, 0}, {0, 0, 1}}], Green,
  Arrow[{J0, J1}], (*Red,Arrow[{C0,C1R}],*)Blue, Arrow[{J0, J1Rproj}] , Yellow,
  Arrow[{{0, 0, 0}, axis3}] , Text[Panel["X", FrameMargins → 0], {1.05, 0, 0}],
  Text[Panel["Y", FrameMargins → 0], {0, 1.2, 0}],
  Text[Panel["Z", FrameMargins → 0], {0, 0, 1.05}]},
 AspectRatio -> Automatic, Boxed → False]
```



Rimane da individuare l'angolo di cui ruotare attorno a Z, allineandosi al vettore verde. Proietto J1Rproj su X.

```
K = {0, 1.14949, 0};

OK = Norm[K]
1.14949

(*ϕ= ArcCos[OD/Norm[C1Rproj]]*)

(*ϕ=ArcCos[(OD^2+(Sqrt[4.64981^2+2.15048^2])^2-(-4.46032)^2)/
   (2*OD*Sqrt[4.64981^2+2.15048^2])]*)

ψ = ArcCos[(OK^2 + (Norm[J1Rproj])^2 - (0.035597)^2) / (2 * OK * Norm[J1Rproj])]
0.0309577

Rz3 = RotationMatrix[ψ, {0, 0, 1}];

J1fin = Dot[Rz3, J1Rproj]
{2.66308 × 10^-8, 1.15004, 0.}
```

```
Graphics3D[{Black, Arrow[{{0, 0, 0}, {1, 0, 0}}], Black,
  Arrow[{{0, 0, 0}, {0, 1, 0}}], Black, Arrow[{{0, 0, 0}, {0, 0, 1}}],
  Green, Arrow[{J0, J1}], Blue, Arrow[{J0, J1Rproj}] , Yellow,
  Arrow[{{0, 0, 0}, axis3}] , Magenta, Arrow[{{0, 0, 0}, J}], Cyan,
  Arrow[{{0, 0, 0}, J1fin}], Text[Panel["X", FrameMargins → 0], {1, 0, 0}],
  Text[Panel["Y", FrameMargins → 0], {0, 1, 0}],
  Text[Panel["Z", FrameMargins → 0], {0, 0, 1}]},
 AspectRatio -> Automatic, Boxed → False]
```




```
EuclideanDistance[J1, J1fin]
```

```
0.0000436383
```

Verifichiamo se ruotando rigidamente gli stessi punti per la matrice di rotazione complessiva si ottengano dai punti ricostruiti i punti reali.

```
J2 = {4.1230, 2, 0};
```

```
J2R = {3.9862, 2.2510, 0.2059};
```

```
J2fin = Rz3.R3.J2R
```

```
{3.91462, 1.99987, -1.29441}
```

```
J2RR = Rz3.R3.J2R
```

```
{3.91462, 1.99987, -1.29441}
```

```
axis32 = Normalize[Cross[J2RR, {0, 1, 0}]]
```

```
{0.313944, 0., 0.949442}
```

```
Part[axis32, 3]
```

```
0.949442
```

```
υ = ArcCos[Part[axis32, 3]]
```

```
0.319344
```

```
Ry3 = RotationMatrix[-υ, {0, 1, 0}]
```

```
{{0.949442, 0., -0.313944}, {0., 1., 0.}, {0.313944, 0., 0.949442}}
```

```
J2fin = Ry3.Rz3.R3.J2R
```

$\{4.12307, 1.99987, 1.11022 \times 10^{-15}\}$

J3 è la coordinata vera, J3fin quella ricostruita

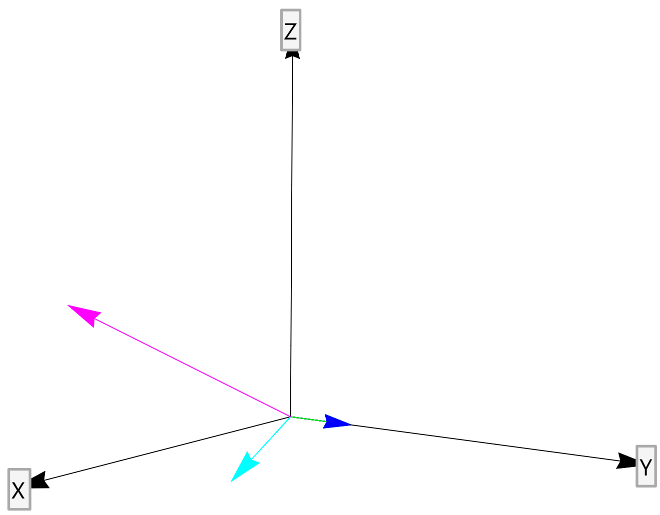
```
J3 = {5, 0, 2.2240};
```

```
J3R = {5.4325, -0.5012 , 0.4276};
```

```
J3fin = Ry3.Rz3.R3.J3R
```

```
{4.99999, -0.000217535, 2.22401}
```

```
Graphics3D[
 {Black, Arrow[{{0, 0, 0}, {6, 0, 0}}], Black, Arrow[{{0, 0, 0}, {0, 6, 0}}],
  Black, Arrow[{{0, 0, 0}, {0, 0, 6}}], Green, Arrow[{J0, J1}], Blue,
  Arrow[{J0, J1fin}] , Magenta, Arrow[{{0, 0, 0}, J3fin}], Cyan,
  Arrow[{{0, 0, 0}, J2fin}], Text[Panel["X", FrameMargins → 0], {6, 0.1, 0}],
  Text[Panel["Y", FrameMargins → 0], {0.1, 6, 0}],
  Text[Panel["Z", FrameMargins → 0], {0.05, 0, 6}]},
 AspectRatio -> Automatic, Boxed → False]
```




```
K - J1Rproj
```

$\{-0.035597, -2.59315 \times 10^{-6}, 0.\}$