In[1]:=
```
Needs@"QuaternionsHM`"
```

Quaternion from Roll-Pitch-Yaw angles

When a frame orientation is assembled as a chain of quaternion multiplications it is done as intrinsic rotations. That means that the rotation axis in each factor is relative to the frame being rotated.

An Euler angles example: quatFromθV[α,{0,0,1}]**quatFromθV[β,{0,1,0}]**quatFromθV[γ,{1,0,0}].
1st rotation is around the Z-axis of the reference frame. 2nd rotation is around the *new* Y-axis.
3rd rotation is around the *newest* X-axis.

There are application areas where it is common to express an orientation as the result of extrinsic rotations. That means that the rotation axes in a chain of rotations stay fixed in the static reference frame.
Roll-Pitch-Yaw orientation of an aircraft is one example.

Two alternative methods to convert RPY angles to a quaternion:
1. Invert the component angle-axis rotations and multiply in RPY order. Then invert the product.
2. Multiply with the angle-axis factors in reverse RPY order.

These methods also work when the rotation axes do not coincide with the base axes of the reference frame.

Function

In[6]:=
```
? quatFromRPYangles
```

Out[6]=

Symbol

quatFromRPYangles[{α,β,γ}]. Returns the quaternion formed by rotating by α around the initial *x* axis, then by β around the initial *y* axis, and finally by γ around the initial *z* axis.

quatFromRPYangles[{α,β,γ}, {*a*,*b*,*c*}]. Returns the quaternion formed by rotating by α around the fixed base axis *a*, then by β around the fixed base axis *b*, and finally by γ around the fixed base axis *c*.

In[2]:=
```
quatFromRPYangles[{α_,β_,γ_},axes:{_,_,_}:{1,2,3}]:=With[
  {v1=UnitVector[3,axes[[1]]],v2=UnitVector[3,axes[[2]]],v3=UnitVector[3,axes[[3]]]},
  (quatToFromθV[α,-v1]**quatToFromθV[β,-v2]**quatToFromθV[γ,-v3])^-1
]
```

Examples

In[7]:=
```
(* Roll axis: X, Pitch axis: Y, Yaw axis: Z *)
quatFromRPYangles[{20. °, 30. °, 40. °}]
quatToFromEulerZYX[40, 30, 20]
```

Out[7]= quat[0.909255, 0.0704393, 0.296883, 0.283114]

Out[8]= quat[0.909255, 0.0704393, 0.296883, 0.283114]

In[9]:=
```
(* Roll axis: X, Pitch axis: Z, Yaw axis: Y *)
quatFromRPYangles[{20. °, 30. °, 40. °}, {1, 3, 2}]
quatToFromθV[40. °, {0, 1, 0}] ** quatToFromθV[30. °, {0, 0, 1}] ** quatToFromθV[20. °, {1, 0, 0}]
```

Out[9]= quat[0.878512, 0.244792, 0.36758, 0.182148]

Out[10]= quat[0.878512, 0.244792, 0.36758, 0.182148]

Sequence of translated and rotated frames

A sequence of two translated and rotated frames:
The 1st frame is translated and rotated relative to the reference frame.
The 2nd frame is translated and rotated relative to the 1st frame.

What is the translation and rotation of the combined frames, relative to the reference frame?

Function

In[3]:=
```
linkTransRotFrames[{transA:{_,_,_},rotA_quat},{transB:{_,_,_},rotB_quat}]:=Module[
  {transC,rotC},
  transC=transA+quatRotateVector[rotA,transB];
  rotC=rotA**rotB;
  {transC,rotC}
]
```

Examples

In[11]:=
```
frameA1 = {{1, 1, 0}, quatToFromθV[45. °, {0, 0, 1}]};
frameB1 = {{0, √2, 0}, quatToFromθV[45. °, {0, 0, 1}]};
frameC1 = linkTransRotFrames[frameA1, frameB1]
```

Out[13]=
```
{{0, 2, 0}, quat[0.707107, 0, 0, 0.707107]}
```

In[14]:=
```
frameA2 = {{3, 0, 0}, quatToFromEulerZYX[-90, 0, 90]};
frameB2 = {{0, 7, -5}, quatToFromθV[π, {1, 0, 0}]};
frameC2 = linkTransRotFrames[frameA2, frameB2]
```

Out[16]=
```
{{8, 0, 7}, quat[0.5, -0.5, 0.5, -0.5]}
```

Reverse of a translated and rotated frame

A frame is translated and rotated relative to the reference frame.

What is the reverse translation and rotation, relative to this frame, that will take it back to the reference frame?

Function

In[4]:=
```
reverseOfTransRotFrame[{transFwd:{_,_,_},rotFwd_quat}]:=Module[
  {transRev,rotRev},
  transRev=quatRotateVector[rotFwd^-1,-transFwd];
  rotRev=rotFwd^-1;
  {transRev,rotRev}
]
```

Examples

In[17]:=
```
reverseOfTransRotFrame[frameC1]
```

Out[17]=
```
{{-2, 0, 0}, quat[0.707107, 0., 0., -0.707107]}
```

In[18]:=
```
reverseOfTransRotFrame[frameC2]
```

Out[18]=
```
{{0, 7, -8}, quat[0.5, 0.5, -0.5, 0.5]}
```