

SpinConv

Conversion from a rotation representation type to another

Paolo de Leva

University "Foro Italico", Rome, Italy

SpinConv converts from a rotation representation type to another. It supports the following rotation representation types:

- Rotation matrix
- Euler angles (including Tait-Bryan angles, a.k.a Cardan angles)
- Axis-angle (rotation about a single axis represented by an Euler vector)
- Quaternion

SpinConv does the same job as SpinCalc by John Fuller (MATLAB Central file #20696). The results are identical, or equivalent, as you can check yourself by calling

```
testSpinConv('output')
```

Negligible output differences may be produced by rounding errors, because multiplications are sometimes executed in a different order.

SpinConv is a different implementation of the same algorithms, basically exploiting vectorized indices instead of loops, to perform iterative computations. In some cases, the new implementation is much faster. For instance, when the number of rotations to be converted (N) is very large, the conversion from DCM to EV is performed about 30 times faster. In some other cases, however, **SpinConv** may be slightly slower, especially when N=1. For details about execution time for all kinds of conversion, just run

```
testSpinConv('speed').
```

SpinConv also fixes a few bugs or inconsistencies that are present in version 1.3 of **SpinCalc**, including a non-standard output range for Euler Angles (in some cases larger than 180 degrees), not consistent with the imposed constraint for the corresponding input range (see input and output ranges below). Some of these bug fixes require additional computations (for instance, the instruction ANY used in the Boolean condition of some IF statements) and partly explain the slight increase in execution time that is observed in some cases.

CONVENTIONS AND CONSTRAINTS

All representation types accepted as input and returned as output by **SpinConv** are meant to represent the rotation of a 3D coordinate system (CS) relative to a rigid body or vector space ("*alias*" transformation), rather than vice-versa ("*alibi*" transformation).

DCMs (also known as rotation matrices) are special orthogonal matrices with determinant 1. Each matrix **R** contains in its rows the versors of the rotated CS represented in the original CS, and in its columns the versors of the original CS represented in the rotated CS. This format is typically used when the column-vector convention is adopted: point coordinates are arranged in column vectors \mathbf{v}_i , and the desired rotation is applied by pre-multiplying \mathbf{v}_i by **R**:

$$\text{rotated } \mathbf{v}_i = \mathbf{R} \mathbf{v}_i.$$

With row vectors, the same rotation can be obtained by post-multiplying by the transpose of **R**:

$$\text{rotated } \mathbf{v}_i = \mathbf{v}_i \mathbf{R}^T.$$

“EA” type (Euler angles) includes both *proper* Euler angles (1st and 3rd rotation about the same axis) and Tait-Bryan angles (rotations about three distinct axes).

Both Euler vectors and quaternions are unit vectors.

INPUT RANGES

- Second Euler angle must be $-90 < \theta < 90$ degrees ($\theta = -90$ and $\theta = 90$ are not allowed).
- Second Tait-Bryan angle must be $0 < \theta < 180$ degrees ($\theta = 0$ and $\theta = 180$ are not allowed).
- All other angles may have any value, in degrees.

OUTPUT RANGES

- Euler angles (ψ , θ , ϕ) range from -90 to 90 degrees
- Tait-Bryan angles (ψ , θ , ϕ) range from 0 to 180 degrees
- Angles about Euler vectors range from 0 to 180 degrees

NOTE

When INPUT contains DCMs and $N > 1$, SpinConv calls MULTIPROD and MULTITRANSF (Both contained in MATLAB Central file #8773).