Modeling and Control of Manipulators
Inverse equivalent angle-axis

University of Genoa, November 2018

1 Versor Lemma

Given two rotation matrices ${}^0_aR$ and ${}^0_bR$, projected on the same frame (e.g. the $\langle 0 \rangle$ world frame), we can express them as:

$$R_a = \begin{bmatrix} \boldsymbol{i_a} \ \boldsymbol{j_a} \ \boldsymbol{k_a} \end{bmatrix} \quad and \quad R_b = \begin{bmatrix} \boldsymbol{i_b} \ \boldsymbol{j_b} \ \boldsymbol{k_b} \end{bmatrix} \tag{1}$$

where the versors i, j, k are the matrices column vectors and identify the rotations around that axes. Between two rotation matrices the following relation holds:

$$(\boldsymbol{i_a} \cdot \boldsymbol{i_b}) + (\boldsymbol{j_a} \cdot \boldsymbol{j_b}) + (\boldsymbol{k_a} \cdot \boldsymbol{k_b}) = 1 + 2cos(\theta) \tag{2}$$

$$(\boldsymbol{i_a} \wedge \boldsymbol{i_b}) + \boldsymbol{j_a} \wedge \boldsymbol{j_b}) + (\boldsymbol{k_a} \wedge \boldsymbol{k_b}) = 2\boldsymbol{v}sin(\theta) \tag{3}$$

Where the axis $\boldsymbol{v}$ (which is a versor), and the angle θ, are the parameters representing the rotation vector $\boldsymbol{\rho}$, or misalignment vector, which identifies a rotation that goes from $\langle a \rangle$ to $\langle b \rangle$.

$$\boldsymbol{\rho} = \boldsymbol{v}\theta \tag{4}$$

In fact, remembering the exponential representation of a rotation matrix, we have that:

$${}^a_bR = R(\boldsymbol{\rho}) = R(\boldsymbol{v}, \theta) = e^{[\boldsymbol{v}\wedge]\theta} \tag{5}$$

where $[\cdot \wedge]$ is the skew-symmetric operator. Knowing this it's easy to understand that $\boldsymbol{v}$ is the axis along which $\langle a \rangle$ has to rotate, of an angle θ to reach $\langle b \rangle$. This will be useful for calculating the angular error (or misalignment) between our robot's end-effector and a desired goal frame, where $\langle a \rangle$ will be our end-effector frame and $\langle b \rangle$ the goal frame.

Remark (Norm of the rotation vector)**.** *Note that since $\boldsymbol{v}$ is a versor its norm equals one, so we have that the norm of $\boldsymbol{\rho}$ gives θ:*

$$\|\boldsymbol{\rho}\| = \|\boldsymbol{v}\| \cdot \|\theta\| = \theta \tag{6}$$

2 Inverse equivalent angle-axis algorithm

Inverting eq. 2 we obtain that:

$$cos(\theta) = ((\boldsymbol{i_a} \cdot \boldsymbol{i_b}) + (\boldsymbol{j_a} \cdot \boldsymbol{j_b}) + (\boldsymbol{k_a} \cdot \boldsymbol{k_b}) - 1)/2 \tag{7}$$

$$= (sum(R_a \circ R_b) - 1)/2 \tag{8}$$

where $\circ$ is the element by element product between the two matrices, and sum is the sum of the elements of the product result.

Inverting eq. 3 instead, remembering the remark in the previous section. we have:

$$sin(\theta) = \|\boldsymbol{v}sin(\theta)\| = \|(\boldsymbol{i_a} \wedge \boldsymbol{i_b}) + \boldsymbol{j_a} \wedge \boldsymbol{j_b}) + (\boldsymbol{k_a} \wedge \boldsymbol{k_b})/2\| \tag{9}$$

Now that we have the sin and cosine of the angle we have three cases to take care of, as shown in Fig. 1.

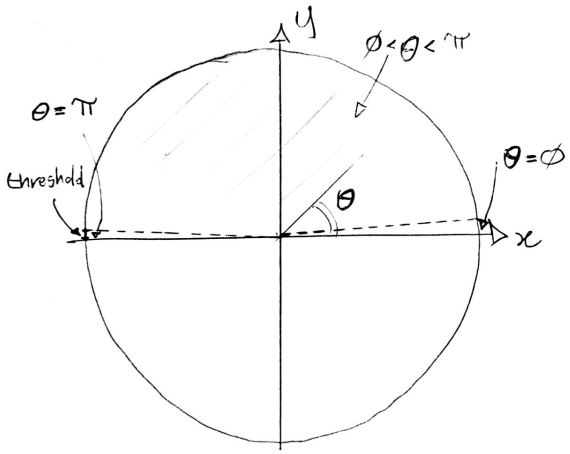

Figure 1: Different cases for θ, with the threshold.

Case 1. $\boxed{cos(\theta) = 0} \Rightarrow$ then $\theta = 0$
In this case the angle θ is zero so no rotation occurs ($\boldsymbol{v} = 0$, $\theta = 0$): $\boldsymbol{\rho} = \boldsymbol{v}\theta = [0\ 0\ 0]$.

Case 2. $\boxed{|cos(\theta)| < 1} \Rightarrow$ then $0 < \theta < \pi$.
In this case: $\theta = \text{atan2}(sin(\theta), cos(\theta))$
and: $\boldsymbol{v} = \frac{\{\boldsymbol{v}sin(\theta)\}}{sin(\theta)}$, where $\{\boldsymbol{v}sin(\theta)\}$ can be obtained from eq. 10, just by not taking the norm of the result.

Case 3. $\boxed{cos(\theta) = -1} \Rightarrow$ then $\theta = \pi$
In this case the rotation is of π (180°), so it's a flipping of the frame on a single axis. We just have to understand around which axis it is. To do this we simply sum element by element the two rotation matrices and check which column vector of the result is not zero. As it can be seen from Fig. 2, the column vector for which the sum is not zero will be the axis of rotation (and will have a norm almost equal to 2), while the other two axis, being flipped, will point in opposite directions and give a zero sum.

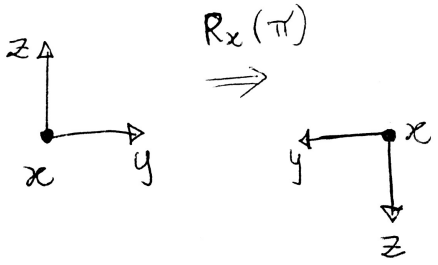

Figure 2: Example of Case 3 for a rotation around the x axis of π.

3 Implementing the algorithm

Due to the fact that there will always be numerical errors in a computer, values as "exactly 0" and "exactly 1" will almost never occur. For this reason we decide a small threshold to base our decisions. A reasonable error threshold could be $\delta = 1 \times 10^{-4}$ example of the different cases introducing the threshold is be the following:

```
function versor = VersorLemma(r1, r2)

% Do all necessary calculation here

if (costh >= (1 - errorThresh) )
    % Case when cos(th) is almost 1

elseif (abs(costh) < (1 - errorThresh))
    % Case when abs(cos(th)) smaller then 1
else
    % Case when cos(th) is almost -1 (by exclusion)
end
```