Name: Tahsincan
Surname: Köse ID: 2188423

Homework 1 - Pose Representations and Motion

1

We can write the circle equation in matrix form as $(p-pc)^T M(p-pc) = 0$, which matrix M $= \begin{bmatrix} \frac{1}{r} & 0 & 0 \\ 0 & \frac{1}{r} & 0 \\ 0 & 0 & 1 \end{bmatrix}$, in more compact form $\begin{bmatrix} A_{2x2} & 0_{2x1} \\ 0_{1x2} & 1 \end{bmatrix}$, where $A = \begin{bmatrix} \frac{1}{r} & 0 \\ 0 & \frac{1}{r} \end{bmatrix}$. Applying an arbitrary transformation T produces a new equation such that $(Tp-Tp_{c'})^T M'(Tp-Tp_{c'}) = 0$. Taking T out of the parentheses and applying Matrix Transpose conversions, we get $(p - p_{c'})T^T M'T(p - p_{c'}) = 0$. This is a very sutiable respresentation, because problem now reduces to show that $T^T M'T$ in fact equals to M.

Recall that $T = \begin{bmatrix} R_{2x2} & t_{2x1} \\ 0_{1x2} & 1 \end{bmatrix}$, further let's write M' in more compact form such as $M' = \begin{bmatrix} A'_{2x2} & 0_{2x1} \\ 0_{1x2} & 1 \end{bmatrix}$ and $A' = kA$. By using these two matrices we can finally write:

$$T^T M'T = \begin{bmatrix} R_{2x2} & t_{2x1} \\ 0_{1x2} & 1 \end{bmatrix} \begin{bmatrix} A'_{2x2} & 0_{2x1} \\ 0_{1x2} & 1 \end{bmatrix} \begin{bmatrix} R^T_{2x2} & 0_{2x1} \\ t^T_{1x2} & 1 \end{bmatrix} \rightarrow \begin{bmatrix} (RA'R^T + tt^T)_{2x2} & t_{2x1} \\ t^T_{1x2} & 1 \end{bmatrix}$$

Applying Gauss-Jordan elimination gives the simpler matrix $\begin{bmatrix} (RA'R^T)_{2x2} & 0_{2x1} \\ 0_{1x2} & 1 \end{bmatrix}$. Important thing here is the first rotation of A' with R^T, then the rotation with R. We know that this is equal to A', since $RR^T = 1$. Ultimate form is the exact same of M by regulation of constant term k. We can conclude our proof by saying that, applying transformation didn't change the radius of circle as we intuitively know. The new center $p_{c'} = p_c + t$.

2

$[\hat{w}]_\times$ represents the skewed matrix of vector $\hat{w} = [w_x \; w_y \; w_z]$:

$$\begin{bmatrix} 0 & -w_z & w_y \\ w_z & 0 & -w_x \\ -w_y & w_x & 0 \end{bmatrix}$$

a)

We calculate eigenvalues by the equation $\lambda.\hat{v} = [\hat{w}]_\times.\hat{v}$. After replacements, we get $\det([\hat{w}]_\times$ - $\lambda.\mathbb{I}_{3x3}) = 0$.

$$\begin{vmatrix} -\lambda & -w_z & w_y \\ w_z & -\lambda & -w_x \\ -w_y & w_x & -\lambda \end{vmatrix} \qquad = \text{-}\lambda(\lambda + w_x^2\ ) + w_z(\text{-}\lambda.w_z \text{ - } w_x.w_y) + w_y(w_z.w_x \text{ - } \lambda.w_y)\ \mathbf{(1)}$$

Simplifying **(1)** gives the equation $-\lambda^3 = \lambda(w_x^2 + w_y^2 + w_z^2)$. In the problem statement it is denoted that $\hat{w}$ is a unit vector. Hence, the part of the right hand side term which calculates the square of $|\hat{w}|$ come outs as 1. Problem reduces to finding the roots of $\lambda^3 + \lambda = 0$. If we do the math, it becomes $\lambda(\lambda^2 + 1) = 0$. It is now obvious that the roots are 0, i and $-i$.

For $\lambda = 0$, the calculation of eigenvector is straightforward. Applying Gauss-Jordan elimination rules after some intermediate steps, the resulting matrix is:

$$\begin{bmatrix} 0 & -w_z & w_y \\ w_z & 0 & -w_x \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix}$$

Obviously, v_3 is an independent variable. Hence, $v_3 = k$. Also, $-w_z.v_2 + w_y.v_3 = 0$ and $w_z.v_1$ - $w_x.v_3 = 0$. Replacing v_3 with k and applying chain conversions, we get $v_2 = \frac{k.w_y}{w_z}$ and $v_1 = \frac{k.w_x}{w_z}$. They can be written in vector form as k $\times \begin{bmatrix} \frac{w_x}{w_z} \\ \frac{w_y}{w_z} \\ 1 \end{bmatrix}$

Other eigenvectors have complex parts because of their complex eigenvalues.

b

Recall that the general form of Rotation matrix along Z axis is:

$$\text{R} = \begin{bmatrix} cos\theta & -sin\theta & 0 \\ sin\theta & cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Let's consider a coordinate frame whose Z axis is aligned with unit vector $\hat{w}$. Then R above becomes valid to go further the calculation.

$$\text{R}.\hat{v} = \lambda.\hat{v} \rightarrow (\text{R-}\lambda.\ \mathbb{I}_{3x3}).\hat{v} = 0 \rightarrow \begin{bmatrix} cos\theta - \lambda & -sin\theta & 0 \\ sin\theta & cos\theta - \lambda & 0 \\ 0 & 0 & 1-\lambda \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix} = 0.$$

Calculating the determinant gives the equation:

$\rightarrow$(1 -λ)(($cos\theta - \lambda)^2$ + $sin\theta^2$) = 0

$\rightarrow$(1 -λ)(λ^2 - 2λ.$cos\theta$ + 1) = 0 **(1)**

Obviously, $\lambda = 1$ is a root of this equation. For other roots, let's write $e^{i\theta}$ and $e^{-i\theta}$ in polar forms, such that $cos\theta + i.sin\theta$ and $cos\theta - i.sin\theta$. Since we are sure that complex conjugate of a complex number is natively another root to the equation, we will only solve for the former complex root, since the latter one is the complex conjugate of it.

Replacing λ with $cos\theta + i.sin\theta$ in the second part of the equation unrolls as:

$cos\theta^2$ - $sin\theta^2$ + $2isin\theta cos\theta$ - $2cos\theta^2$ - $2isin\theta cos\theta$ + $cos\theta^2$ + $sin\theta^2$
All these terms cancel their contrary parts. Hence this proves, $cos\theta + i.sin\theta$ is indeed a root to the equation. This concludes the proof.

The eigenvector of 1 can be computed by replacing the λ in matrix and solving for a $\hat{v}$ that satisfies R.$\hat{v}$ = $\hat{v}$.

$\begin{bmatrix} cos\theta - 1 & sin\theta & 0 \\ sin\theta & cos\theta - 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix} = \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix}$ $\rightarrow$ Solving these by applying eliminations:

$\begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix} = \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix}$ It is obvious that $v_3 = k$, $v_2 = 0$, $v_1 = 0$.

The eigenvector is k $\times$ $\begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$, which is the axis of rotation.

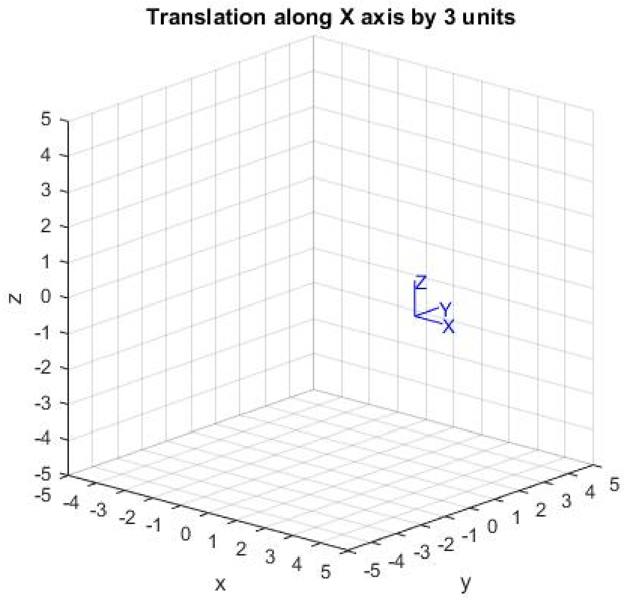
c

The rotation matrix R = $[r_1 r_2 r_3]$ constitutes of unit orthogonal vectors, since $R \in SO(3)$. We are then sure that $r_1 = r_2 \times r_3$. Problem reduces to $r_1^T r_1$. r_1 is unit from the definition and the multiplication of $r_1^T r_1$ equals 1. Also we know that having det(R) = 1 is another characteristic of $SO(3)$ rotations. This finishes the proof, that is det(R) = $r_1^T r_1$. Moreover, this is also true for r_2 and r_3.

3

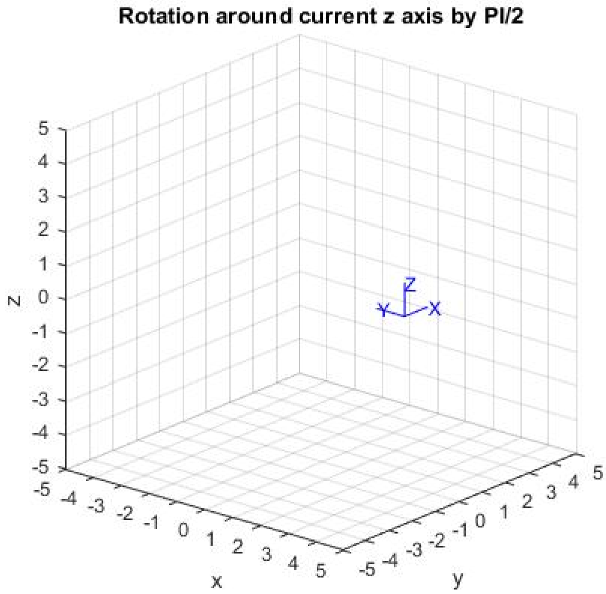
The transformation corresponds to translation along x axis by 3 units is below:

$$T_1 = \begin{bmatrix} 1 & 0 & 0 & 3 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$



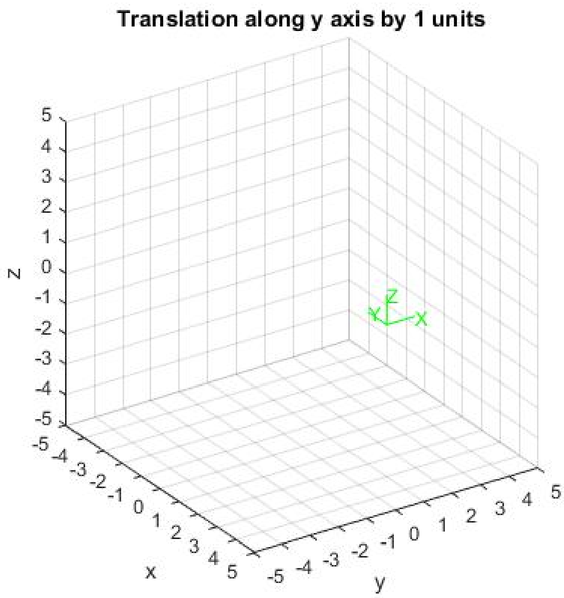
The following rotation about the current z axis $T_2 = \begin{bmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$

Multiplying T_1 and T_2, we get $T_1T_2 = \begin{bmatrix} 0 & -1 & 0 & 3 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$



And lastly, the translation along **fixed** y axis corresponds to the **current** x axis, which is $T_3 = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$

Multiplying T_1T_2 with T_3, we get $T_1T_2T_3 = \begin{bmatrix} 0 & -1 & 0 & 3 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$



Assigning O = [0 0 0] as the original frame's origin, we can easily compute O_1 after each transformation by computing $O_1 = T_i O$. In conclusion, respectively [3 0 0], [3 0 0], [3 1 0] are the coordinates of O_1 with respect to original frame.

4

T_1^0 is the translation along z axis by 1 unit followed by the rotation of $\pi/2$ about new y axis followed by the rotation of $\pi/2$ about new x axis. It can be written as $T_1^0 = transl([0\,0\,1]) \times troty(\pi/2) \times trotx(\pi/2)$. In order to reduce overhead of computation I will make use of Toolbox and easily produce the transformation matrix:

$$T_1^0 = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ -1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

T_2^0 is the translation along y axis by 1 unit followed by the rotation of $\pi/2$ about x axis followed by the rotation of $-\pi/2$ about y axis . It can be written as $T_1^0 = transl([0\,1\,0]) \times trotx(\pi/2) \times troty(-\pi/2)$. In order to reduce overhead of computation I will make use of Toolbox and easily produce the transformation matrix:

$$T_2^0 = \begin{bmatrix} 0 & 0 & -1 & 0 \\ -1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

T_2^1 is the translation along x axis by 1 unit and along z axis by -1 unit followed by the rotation of $\pi/2$ about z axis followed by the rotation of $-\pi/2$ about y axis. It can be written as $T_2^1 = transl([1\,0 - 1]) \times trotz(\pi/2) \times troty(-\pi/2)$. In order to reduce overhead of computation I will make use of Toolbox again. The transformation matrix:

$$T_2^1 = \begin{bmatrix} 0 & -1 & 0 & 1 \\ 0 & 0 & -1 & 0 \\ 1 & 0 & 0 & -1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

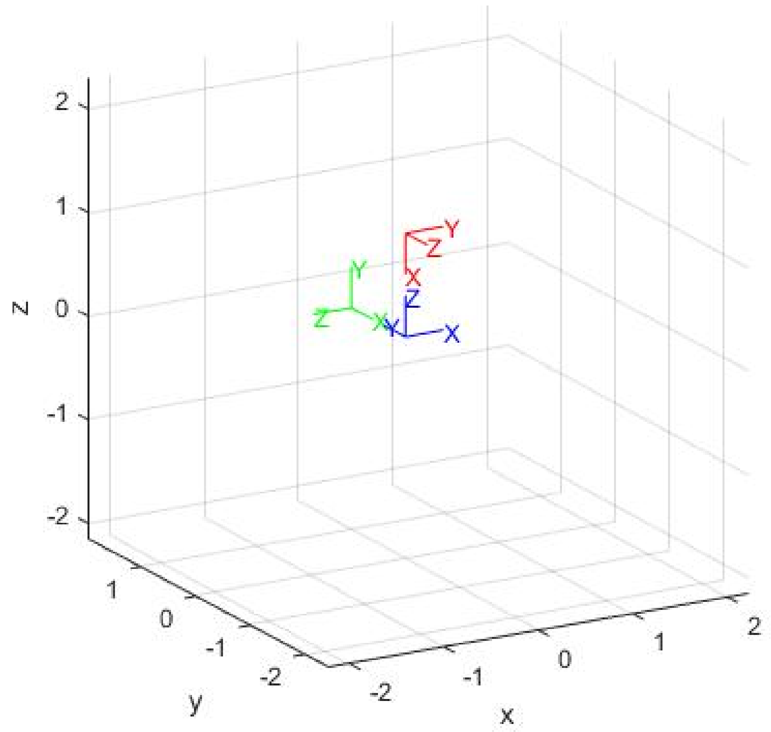


Figure 1: Original frame and the other 2 frames produced by Transformations

The equality of $T_2^0 = T_1^0 T_2^1$ can be easily verified through Matrix Multiplication. In order to illustrate:

$$\begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ -1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & -1 & 0 & 1 \\ 0 & 0 & -1 & 0 \\ 1 & 0 & 0 & -1 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 0 & -1 & 0 \\ -1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \text{ which is indeed } T_2^0.$$

5

I started to solve this question through searching how to represent a cube in the MATLAB consistently. After brief research, I encountered with **patch** function which takes the faces and vertices as input. It resembles to a Scene definition file in Computer Graphics.

After succesfully representing the cube, I applied some rotations to the cube with dead-simple functions of the MATLAB. After, I have read what is a Twist and the Toolbox practice of it in the book, I started to experiment with it in the MATLAB command tool. I understood that, this is basically a rotational movement (optionally translational, with *"pitch"* argument) about a vector which particularly goes through a point. Then, I created a **while** loop with its condition set to 1 (always true) and applied transformations with angles starting from 0 to ∞. I didn't use **tranimate,** but, instead **trplot** with pauses for 0.25 seconds for each 0.2 radian.

6

Rotation around x axis occurs about the vector $\hat{v}_x = [1 \quad 0 \quad 0]$. Applying it to the definition of quaternion in the page 45 of *Robotics, Vision and Control* produces:
$\dot{q}_x = cos\frac{\theta}{2} < [sin\frac{\theta}{2} \quad 0 \quad 0] >$. We also know that a vector $\hat{v}$ can be rotated by $\dot{q}$ through $\dot{q} \circ \hat{v} \circ \dot{q}^-1$. Computing it first with $\dot{q}_x \circ \dot{v}_x$ produces $-sin\frac{\theta}{2} < [cos\frac{\theta}{2} \quad 0 \quad 0] >$, then multiplying the outcome with $\dot{q}_x^-1$:
$-sin\frac{\theta}{2} < [cos\frac{\theta}{2} \quad 0 \quad 0] > \circ\, cos\frac{\theta}{2} < [-sin\frac{\theta}{2} \quad 0 \quad 0] = 0 < [sin^2\frac{\theta}{2} + cos^2\frac{\theta}{2} \quad 0 \quad 0] >$, which comes out as $0 < [1 \quad 0 \quad 0]$. We know that this is a pure quaternion of vector $\hat{v}_x$, hence this proves that x axis remain fixed for those transformations. We can apply the same process for $\hat{v}_y$ and $\hat{v}_z$ to further strengthen the proof but it is cumbersome to go each step again and again. One can be sure on that y and z axis remain fixed after transformations by only observing the unit vector representations of y and z axis and their corresponding quaternions.