Middle East Technical University   Department of Computer Engineering

CENG 460

Introduction to Robotics

Fall 2019-2020

Homework 1 - Spatial Transformations

Due date: November 3, 2019, Sunday, 23:55

1 Theory (60 pts)

Provide a solution for the following questions. All rotation matrices are to be assumed right handed.

a. **Hands-On Practice (20 pts)** Compute the transformation matrices for the following sets of operations by hand. All vectors and points are in global frame coordinates unless stated otherwise.

   i. Translate an object by length 2 along a vector $\mathbf{k}$. This vector is obtained by a rotation of $[1, 0, 0]$ along fixed x axis by $\pi/6$ CCW, then another rotation along fixed y axis by $\pi/4$ CCW. Then rotate the same object around vector $\mathbf{k}$ by $\pi/2$ CCW.

   ii. Inverse of i.

   iii. Write i and ii in unit quaternion q and a translation vector t form, where an object at t_0 with orientation p maps to $(qpq^{-1}, t_0 + t)$.

   iv. An depiction of a robotic arm is shown in Figure 1 with some frames. The arm consists of two rods rod_1, rod_2, of length $l_1, l_2 > 0$, where rod_2 makes angle θ with the positive y axis if rod_1 was assumed to be aligned with positive x direction, and rod_1 is perpendicular to rod_2.

   - Find ${}^{M_1}_{M_{12}}T = T_1$, relative transform of the frame at the connection to the frame at the free end of rod_1.
   - Find ${}^{M_{12}}_{M_2}T = T_2$, relative transform of the frame at the free end of rod_2 to the frame at the connection.
   - Find ${}^{M_1}_{M_2}T = T_3$.

b. **Similarity Transform and Commuting (25 pts)**

   i. Let T be the 2D homogenous transform representing a translation by vector $\mathbf{p} = [x_0, y_0, z_0] \neq 0$ in the global frame G. Let D be the homogeneous transform representing a rotation by θ in G. Calculate $A = TDT^{-1}$ and verbally explain what it does. This is a useful trick called *similarity transform*, it is used to do operations that are easier to represent in algebraic bases that are different from current one.

   ii. Show that $\mathbf{p}$ is an eigenvector of A with eigenvalue 1. Verbally explain the geometric sense of this phenomenon.

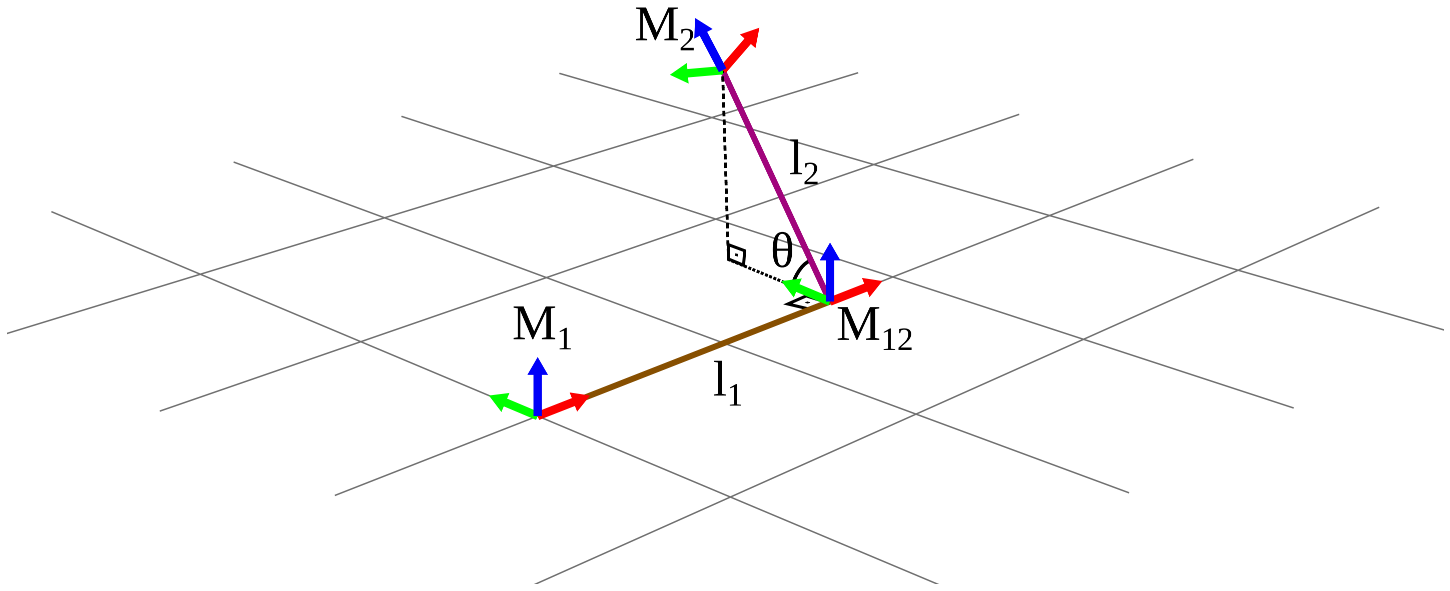

Figure 1: A robotic arm with the attached frames M_1, M_{12}, M_2. The unit $\hat{x}$,$\hat{y}$,$\hat{z}$ vectors of the frames are colored with red, green, blue respectively. M_1 is located at the free end of rod_1 and $\hat{x}_{M_1}$ is aligned with it. M_{12} is situated on the connection with the same orientation as M_1. $\hat{z}_{M_2}$ is in the same direction with rod_2, $\hat{x}_{M_2} = \hat{x}_{M_1}$, and the frame is located at the free end.

iii. Show that if 3D rotation matrices $R_1, R_2 \neq I$ share their axis of rotation, then $R_1R_2 = R_2R_1$. (Hint: Proving this is very *similar* to proving "If 2 2D homogenous transformation matrices $T_1, T_2 \neq I$ share their center of rotation, then $T_1T_2 = T_2T_1$." You do not need to answer this, but can you also see the generalization for 3D on this one?)

iv. Give a counter example that **disproves** the converse claim: "For 3D rotation matrices $R_1, R_2 \neq I$, if $R_1R_2 = R_2R_1$ then they share their axis of rotation."

v. *Post-multiplying rule:* Let ${}^G_BT$ be the transformation of a body with the body frame B with respect to the global frame G in 2D. Let T_1 be a transformation representing a translation by an arbitrary vector $\mathbf{t}$ and a rotation θ from positive x axis. We already know that if $\mathbf{t}$ and θ were defined from the x,y axes of G, the final transform of the body would be $T_1 {}^G_BT$ in G. Show that if $\mathbf{t}$ and θ were defined from B, the final transform of the body would be ${}^G_BTT_1$ in G. (Hint: Use similarity transform. You can use this proof without changing anything to show post-multiplying applies to 3D as well).

c. **Pons Asinorum (15 pts)** Some rotations are applied to a body in the global frame G. Another frame B is also defined on the body, and the axes of this frame also change according to the orientation. The rotations are applied in axes that pass through origin in the following order, as listed:

1. Rotate the object CCW around ${}^{\mathbf{G}}\mathbf{k_1} = [3, 4, 12]$ (G indicates ${}^{\mathbf{G}}\mathbf{k_1}$ is written in G coordinates) by $14°$ degrees.
2. Rotate the object CCW around ${}^{\mathbf{G}}\mathbf{k_2} = [-6, -1.5, 2]$ by $113°$ degrees.
3. Rotate the object CCW around ${}^{\mathbf{G}}\mathbf{k_3} = [24, 6, -8]$ by $23°$ degrees.
4. Rotate the object CCW around ${}^{\mathbf{B}}\mathbf{k_4} = [3, 4, 12]$ (B indicates ${}^{\mathbf{B}}\mathbf{k_4}$ is written in B coordinates) by $256°$ degrees.

This sequence of rotations can be summarized with the operation RDR^{-1}, where $D = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$ and R is a rotation matrix. Find R without explicitly multiplying the 4 matrices that correspond to these events.

(Hint: Use all the facts you have learned from the previous question. Notice that $\mathbf{k_1} \cdot \mathbf{k_2} = 0$, in a very *similar* manner to the columns of a rotation matrix, and try to figure out what D does to an arbitrary frame.)

2 Programming (40 pts)

In this part, you are going to simulate n_r identical robotic arms that collectively carry object o. The arm shape is the same as Figure 1.

Any arm can instantaneously change its parameters $l_1[t], l_2[t], \theta[t]$ at any time step, however the free end of rod_1 stays fixed. In addition to that, each arm has a *grab* utility that binds o to the free end of rod_2. While a particular arm is grabbing o, o moves along with the arm as if an imaginary rod between the free end of rod_2 and o existed. For grabbing o, there are two things to look out for. First one is the *hold* signal $h[t] \in \{0,1\}$ for each arm. The second is the condition $c_{sph} \in \{0,1\}$ which indicates if the object is within the sphere of radius d centered at the free end of rod_2. Lastly, let $g \in \{0,1\}$ be the variable that indicates that the object is currently being grabbed. The conditions for grabbing according to these properties is as follows at a particular time step $t = i$ for an arm a_j:

- If $h_{a_j}[i] = 0$:
  - If $g_{a_j} = 0$ (o is not being grabbed by the arm) , do nothing.
  - Else, set $g_{a_j} = 0$. **do not move o along with the arm in this time step.**
- If $h_{a_j}[i] = 1$:
  - If $c_{sph_{a_j}}$ is not satisfied, do nothing.
  - Else If $g_{a_k} = 0$ for all other arms ($j \neq k$), set $g_{a_j} = 1$, and move o along with the arm until g_{a_j} is set to 0 (again, in the time step where g_{a_j} is set to 0, do not move o along with with the arm.)

Edge Cases and Race Conditions

This is not an Operating Systems course, so here are some relaxations to this problem:

- For an ungrabbed object, there will be at most one arm with c_{sph} satisfied and $h = 1$.
- An arm may drop o at the same time step where another arm may grab it. You can avoid the race condition by dropping o if it needs to be dropped first, and then checking if another arm grabs it.

2.0.1 Implementation Specifications

Implement the functions in the given stub file with Python3. You may use libraries other than NumPy, SciPy, and Matplotlib, but you need to get permission from the course assistant first (preferably by opening a topic at the ODTUCLASS forum). The code will be reviewed with white-box technique, therefore please **comment** your intents within the hard-to-understand parts of your code.

3 Submission

You can submit handwritten solutions for the Theory part, however they need to be scanned (to pdf format), **and the hardcopies should be delivered to B202 within the next day after submission**. Submit all your files, including the scans and the filled in stubs within a zipfile named `eXXXXXXX_hw1_460_2019f.zip` to ODTUCLASS.