Homework 0

Out: Friday, Jan 20

Due: Friday, Feb 3rd @ 5:00pm EST

Welcome to Computational Aspects of Robotics. This is homework 0, an introductory homework for the course to gauge your background and preparedness. You should have taken linear algebra, and must be comfortable with Python to take this course. This homework covers the basics of rigid $SE(3)$ transforms. The assignment should help you practice the material from the first lecture!

This homework is one of five assignments. It is worth 50 points, and all other homework assignments are worth 100 points (possibly more with extra credit.)

For problems 1-3, compile your answers in `hw0.pdf` using the LaTeX template provided on the course website. We recommend Overleaf for this purpose. Add `hw0.pdf` to the directory `hw0/supplemental`.

For problem 4, you will directly be editing python files provided by the TAs. If there are any know issues please document them in a problem 4 section of `hw0.pdf`. Otherwise you can keep this section blank.

When you are done, zip the `hw0` directory and upload to coursework. We will review your answers, provide constructive feedback, and run your code against our grading scripts!

Problem 1 *(8 points)*

Given the following matrices:

$$A = \begin{bmatrix} 1 & 0 & 0 & 20 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 30 \\ 0 & 0 & 0 & 0 \end{bmatrix}, B = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 0 & -1 & 3 \\ 0 & 1 & 0 & 5 \\ 0 & 0 & 0 & 1 \end{bmatrix}, C = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 0 & -1 & 3 \\ 0 & -1 & 0 & 5 \\ 0 & 0 & 0 & 1 \end{bmatrix}, D = \begin{bmatrix} -1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 3 \\ 0 & 1 & 0 & 5 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

1. (4 *point*) Determine if each matrix belongs to the $SE(3)$ group of valid homogeneous transforms. Justify your answers and show any relevant calculations.

2. (4 *point*) For any transform $T \in SE(3) \cap \{A, B, C, D\}$ compute the inverse transform T^{-1}. Verify $T^{-1}T = I$, $TT^{-1} = I$. Show any relevant calculations.

Problem 2 *(8 points)*

For this problem, no need to show your work, but give a list of the steps or expressions you evaluated to get the answer.
Given the following $SE(3)$ poses:

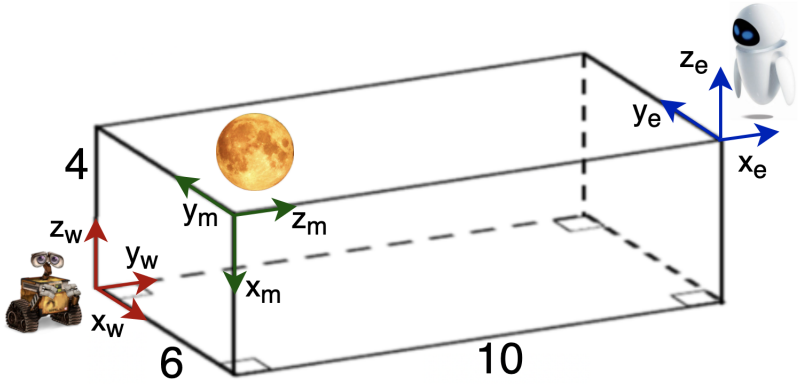
$$ {}^0T_1 = \begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 10 \\ 0 & 1 & 0 & 3 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \; {}^1T_2 = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & -3 \\ 0 & 0 & 1 & -10 \\ 0 & 0 & 0 & 1 \end{bmatrix}. $$

1. (4 *point*) The coordinate of a 3D point p, is given as ${}^2p = [2, 4, 6]^T$ in the coordinate space of frame 2. Compute ${}^1p$, the coordinate of p in frame 1.

2. (4 *point*) Compute the transform ${}^0T_2$, the transform that takes a point from coordinate space of frame 2 to the coordinate space of frame 0. Write a few sentences describing this transformation (in terms of rotations about certain axes and translation along certain axes.)

Problem 3 *(6 points)*

Here we have three coordinate frames. Eve's reference frame e, Wall-e's reference frame w, and the moon's reference frame m.



1. (3 *point*) Write the following poses ${}^mT_w$, ${}^eT_m$, ${}^eT_w$ as $SE(3)$s. Pick one of these poses and write a sentence about what this pose represents.

2. (3 *point*) Verify that ${}^eT_m \; {}^mT_w = {}^e\, T_w$, and ${}^eT_w = ({}^wT_e)^{-1}$. Show your work.

Problem 4 *(28 points)*

In this problem you will implement light-weight transforms functions in python.

Make sure your default python interpreter is python3. If not use the command `python3` and `pip3` instead of `python` and `pip`. Dependencies for hw0 can be installed using the following commands, where we first create a python virtual environment:

```
python -m venv hw0
```

To activate your environment run:

```
source hw0/bin/activate // Linux or OSX
hw0\Scripts\activate.bat // Windows
```

To install dependencies, once you have activated your environment run:

```
pip install -r requirements.txt
```

To exit the environment when you are not working on this project, run:

```
deactivate // Linux of OSX
hw0\Scripts\deactivate.bat // Windows
```

You can read more about python virtual environments here.

1. (28 *point*) In the first part of this problem, you will implement parts of `transforms.py`. This file contains some generic transforms definitions related to problems 1-4. In `transforms.py` you should implement the following:

   ```
   transform_is_valid(...)
   transform_concat(...)
   transform_point3s(...)
   transform_inverse(...)
   ```

   Detailed descriptions of function inputs and outputs are given in `transforms.py`. We have also provided `transforms_test.py` to provide basic unit tests for your transform functions. Before proceeding, it is a good idea to make sure all of these tests pass. Subsequent parts of the project can depend on some or all of these functions depending on your implementation. To run unit tests, execute the command:

```
python transforms_test.py
```

Note: these tests are meant to help you, but are not necessarily exhaustive. You are encouraged to do further testing if you feel certain cases are not properly tested.