M.E. 530.646 Lab 1: Homogeneous Transformation Matrices

Ryan Keating
Johns Hopkins University

As with Lab 0, put the ROS package for this assignment into your catkin workspace.
To complete this assignment, you will write code in both `lab1_main.cpp` and `lab1.cpp`. You will implement a number of different functions in `lab1.cpp` (which already includes their declarations), and you will use `lab1_main.cpp` to test them and complete the rest of the assignment. You will need to use the Eigen library, and so examples of how to use the necessary functions are included in `lab1_main.cpp`. You will also use rviz for debugging and visualization. To launch rviz for this lab, run the following command:

```
roslaunch me530646_lab1 rviz.launch
```

Similar to Lab 0, the following command will run the code in lab1_main.cpp:

```
rosrun me530646_lab1 lab1_main
```

Provided Functions

The following functions have been created for you and can be called in both `lab1_main.cpp` and `lab1.cpp`:

plotf This function is used to plot coordinate frames in rviz. It must be called on an instance of the `RvizPlotter` class. There are two versions of this function:

1. Takes a 4x4 homogenous transformation and the name for the coordinate frame. The frame is plotted relative to the world frame in rviz.
2. Takes a 4x4 homogeneous function and the names of both the parent frame and the child frame. The child frame is plotted relative to the parent frame in rviz.

plotv Takes two the name of a coordinate frame and two `Eigen::Vector3f`s. Plots a vector from the first point to the second point with respect to the given frame.

getTransformation Takes the names of a frame and its parent frame and returns the homogeneous transformation matrix from the parent to the child.

printEigen This function takes Eigen matrices or vectors and prints them out with nice formatting.

Assignment

Be sure to comment all of your code!

1. Write a function `rollr` that accepts a scalar roll value and returns the corresponding 3x3 rotation matrix.
2. Write a function `pitchr` that accepts a scalar pitch value and returns the corresponding 3x3 rotation matrix.
3. Write a function `pitchr` that accepts a scalar yaw value and returns the corresponding 3x3 rotation matrix.
4. Use the three functions above to write a function `rpyr` that accepts three scalar values (roll, pitch, yaw) and returns the 3x3 rotation matrix using the TaitBryan (roll,pitch,yaw) angle convention discussed in class. Experiment plotting various rotations in rviz.
5. Write a function `rpyrinv` that accepts a 3x3 rotation matrix and returns the a 3x1 vector $[roll, pitch, yaw]^T$. Experiment with `rpyr` and `rpyr` in prove the following:
   (a) `rpyr(a,b,c) = rpyr(a+`2π`,b,c)` $\forall$ a, b, c $\in \mathbb{R}$
   (b) `rpyr(R)` is numerically ill-defined for some rotation matrices. Why?
   (c) $[a, b, c]^T \neq$ `rpyrinv(rpy(a,b,c))` for some values of a, b, c—e.g. $(4\pi, 0, 0)$.
   (d) `R = rpyr(rpyrinv(R))` $\forall$ a,b,c where R is well defined.
6. Write a function `xf` that accepts six scalar values (x, y, z, roll, pitch, yaw) and returns the corresponding 4x4 homogenous transformation.
7. Write a function `finv` that accepts accepts a 4x4 homogeneous transformation and returns its matrix inverse. Use the formula derived in class, NOT Eigen's `inverse()` function.
8. Write a function `XYZIntrinsicPlot` that accepts an instance of `RvizPlotter` and three scalar values (roll, pitch, yaw) and animates an intrisic rotation about the x, y, and z axes in rviz.
9. Write a function `ZYXExtrinsicPlot` that accepts an instance of `RvizPlotter` and three scalar values (roll, pitch, yaw) and animates an extrinsic rotation about the z, y, and x axes in rviz.
10. Experiment with `plotv` to plot vectors. Experiment with rotating their coordinate frames.

Submission Guidelines

Hand in annotated screenshots of the rviz window and push your code to your lab repository.

This lab was adapted from a previous version of the course with permission of Dr. Louis Whitcomb.