ROB 541 Assignment 1: Groups

Ross L. Hatton

Last updated: September 28, 2024

1. Create a software structure for performing group operations. You should have object-oriented classes (Python as a default, but I am interested in seeing implementations in other languages such as Matlab or R) for groups and group elements. See Chapter 1 programming exercises for suggested code structure.

   Your group class should be able to take functions defining the group operation and its inverse, and group elements should store their parameter values and the group they are associated with. The group element class should support left and right actions and left and right inverse actions, using the group operation and the parameter values of the acting and acted-on elements to construct a new group element. Implementing infix notation can be helpful, but is not necessary.

   Check your code by verifying that it provides correct answers for the two scale-shift group compositions in Example 1.3.16.

   Deliverables:

   - Compute the gh and hg compositions of group elements $g = (0, 1, -\frac{\pi}{4})$ and $h = (1, 2, -\frac{\pi}{2})$.
   - Use your code to illustrate g, h, gh, and hg. Your $SE(2)$ illustrations can be elaborate (like the triangles in the book) or simpler (e.g., two points connected by a line, with a clear indication of which point is at the location and which is showing orientation).
   - Use your code to illustrate the group elements representing the position of g with respect to h and of h with respect to g.

2. Create a software structure for performing representation group operations. This code will be similar to your original group class, except that instead of specifying the group operation and its inverse, you construct a group by specifying its representation and derepresentation functions. Your group element class should store its configuration information in matrix representation form (calling the representation function when constructing a group element from a list of parameter values and the derepresentation function to export its list of parameter values), and use matrix multiplication and inverse multiplication for operations.

   Deliverables: Demonstrate that you can substitute your representation-group code for your group code above, and that the calculations and plots come out the same.

3. Augment your software structure to include an adjoint action operation. For future purposes (and in defiance of some language naming conventions), this operation should be called `AD` (because `Ad` and `ad` are extensions of the adjoint operation onto different mathematical objects, and we want to keep them separate.)

Deliverables:

- For the group elements $g_1 = (0, 1, -\frac{\pi}{4})$ and $g_2 = (1, 2, -\frac{\pi}{2})$, compute the relative position $h_{\frac{2}{1}}$ of g_2 with respect to g_1 and the adjoint at g_1 of this relative position, and demonstrate that the left action of the adjointed relative position on g_1 brings it to g_2.
- For a group element $h_1 = (-1, 0, \frac{\pi}{2})$, illustrate that moving g_1 by h_1 and g_2 by $h_2 = AD^{-1}_{h_{\frac{2}{1}}} h_1$ preserves the relative displacement between the two elements.