ME 384R ASBR: THA1 - Programming Assignment Report

Daniyal Maroufi and Anas Yousaf

February 20, 2025

1 Problem

1.1 PA 1

For problem 1 in the programming assignment, the task was to write separate functions that could convert a rotation matrix in *SO(3)* into its equivalent axis-angle representation, quaternion representation, ZYZ representation, and roll-pitch-yaw representation.

1.2 PA 2

For problem 2 in the programming assignment, the task was to create separate functions that could return the equivalent rotation matrix if given either an axis-angle representation or a quaternion representation.

1.3 PA 3

For problem 3 in the programming assignment, the task was to create MATLAB functions that could take an initial transformation matrix, a screw axis representation, and a screw axis distance and calculate the transformation matrix at the given intermediate and final configurations. Furthermore, the program would plot the rigid body in the desired configurations as well as the screw axis from the initial to the final configuration.

2 Method

A helper function *is equal_tol* was created for all the test functions written to ensure that two values were equal within a tolerance.

2.1 PA 1

For problem 1, four functions were created along with four test functions and one helper function. The four functions written are: *rotmat2axisangle*, *rotmat2quaternion*, *rotmat2rollpitchyaw*, and *rotmat2zyz*. These four functions take an input rotation matrix and convert it to its corresponding representation. The four corresponding test functions created are: *test_rotmat2axisangle*, *test_rotmat2quaternion*, *test_rotmat2rollpitchyaw*, and *test_rotmat2zyz*. These four test functions were used to verify the viability of the functions written. Finally, the helper function *is_valid_rotation_matrix* was used to ensure that the input rotation matrices were in *SO(3)*.

2.2 PA 2

For problem 2, two functions were written along with two corresponding test functions. The two functions, *axisangle2rotmat* and *quaternion2rotmat*, return the rotation matrix equivalent to the corresponding representation given. The two test functions used to verify these functions are *test_axisangle2rotmat* and *test_quaternion2rotmat*.

2.3 PA 3

For problem 3, three primary functions were written, along with two functions to help with plotting, one helper function, and one test function. The primary functions include *screw2transfmat* and *transfmat2screw*, which are used to convert between transformation matrices and the corresponding screw axis representation, as well as *inputTransfmatScrew*, which takes in a transformation matrix and screw axis and returns its final configuration as described by the problem statement. The plotting function *plotTransformAxes* plots the x,y,z axes for a given transformation matrix while the function *plotScrewAxis* plots the axis of a given screw representation. The helper function *is_valid_transformation_matrix* checks to ensure that any input matrix is a valid homogeneous transformation matrix. Finally, the test function *test_inputTransfmatScrew* is used to test the *inputTransfmatScrew* function and plot the required plots.

3 Code Details

3.1 PA 1

The *rotmat2axisangle* function computes the axis-angle representation, (ω, θ), corresponding to a given rotation matrix, R, based on the following equations from class:

$$\theta = cos^{-1}(\frac{1}{2}(tr(R) - 1)) \qquad \hat{\omega} = \frac{1}{2sin\theta}(R - R^T) \tag{1}$$

The function also checks that the input matrix is a valid rotation matrix and handles the special case when there is 180° of rotation.

The *rotmat2quaternion* function computes the quaternion, $Q = (q_0, \bar{q})$, corresponding to a given rotation matrix, R, based on the following equations from class:

$$q_0 = \frac{\sqrt{1 + tr(R)}}{2} \tag{2}$$

$$q_1 = \frac{1}{2}(sgn(r_{32} - r_{23})\sqrt{r_{11} - r_{22} - r_{33} + 1}) \tag{3}$$

$$q_2 = \frac{1}{2}(sgn(r_{13} - r_{31})\sqrt{r_{22} - r_{33} - r_{11} + 1}) \tag{4}$$

$$q_3 = \frac{1}{2}(sgn(r_{21} - r_{12})\sqrt{r_{33} - r_{11} - r_{22} + 1}) \tag{5}$$

where $\bar{q} = [q_1, q_2, q_3]$ and the elements of the rotation matrix are denoted as r_{ij}. The function also checks that the input matrix is a valid rotation matrix and handles the special case where the trace of the rotation matrix is close to zero.

The *rotmat2rollpitchyaw* computes the ZYX Euler angle representation corresponding to a given rotation matrix, R, using the following equations from class:

$$\phi = Atan2(r_{21}, r_{11}) \qquad \theta = Atan2(-r_{31}, \sqrt{r_{32}^2 + r_{33}^2}) \qquad \psi = Atan2(r_{32}, r_{33}) \tag{6}$$

where elements of the rotation matrix are denoted as r_{ij}. The function also checks that the input matrix is a valid rotation matrix and determines which quadrant the angles are in.

The *rotmat2zyz* computes the ZYZ Euler angle representation corresponding to a given rotation matrix, R, using the following equations from class:

$$\phi = Atan2(r_{23}, r_{13}) \qquad \theta = Atan2(\sqrt{r_{13}^2 + r_{23}^2}, r_{33}) \qquad \psi = Atan2(r_{32}, -r_{31}) \tag{7}$$

where elements of the rotation matrix are denoted as r_{ij}. The function also checks that the input matrix is a valid rotation matrix, determines which quadrant the angles are in, and handles the singularity case if the rotation is about the z-axis.

The helper function *is_valid_rotation_matrix* checks that the input rotation matrix is valid by calculating that $||R^T R - I|| < threshold$ and that $|det(R) - 1| < threshold$.

3.2 PA 2

The *axisangle2rotmat* function calculates the rotation matrix, R, given an input axis-angle representation, $(\omega,\ \theta)$, based on the Rodrigues' formula from class:

$$R = e^{\hat{w}\theta} = I + [w]sin\theta + [w]^2(1 - cos\theta) \tag{8}$$

If the norm of the axis is found to be less than a tolerance, then the function returns the identity matrix and if the norm is not 1, then the function determines that the input is not a valid axis-angle representation.

The *quaternion2rotmat* function calculates the rotation matrix, R, given an input quaternion representation, $Q = (q_0, \bar{q})$, using the following equations from class:

$$R = \begin{bmatrix} q_o^2 + q_1^2 - q_2^2 - q_3^2 & 2(q_1q_2 - q_0q_3) & 2(q_0q_2 + q_1q_3) \\ 2(q_0q_3 + q_1q_2) & [q_o^2 - q_1^2 + q_2^2 - q_3^2] & 2(q_2q_3 - q_0q_1) \\ 2(q_1q_3 - q_0q_2) & 2(q_0q_1 + q_2q_3) & [q_o^2 - q_1^2 - q_2^2 + q_3^2] \end{bmatrix} \tag{9}$$

The function also checks to make sure that the input is a unit quaternion.

3.3 PA 3

The *screw2transfmat* function computes the homogeneous transformation matrix, T, corresponding to the input screw transformation, $\{q, \hat{s}, h\}, \theta$, by first converting it to the twist vector interpretation using the formula:

$$S = \begin{bmatrix} w \\ v \end{bmatrix} = \begin{bmatrix} \hat{s} \\ -\hat{s} \times q + \hat{s}h \end{bmatrix} \tag{10}$$

which is then used to calculate the transformation matrix using the following formula:

$$T = e^{[w]\theta} = \begin{bmatrix} e^{[w]\theta} & (I\theta + (1 - cos\theta)[w] + (\theta - sin\theta)[w]^2)v \\ 0 & 1 \end{bmatrix} \tag{11}$$

The rotation matrix, $e^{[w]\theta}$, is calculated using the *axisangle2rotmat* function. Additionally, if the $||\hat{w}|| < threshold$, then the rotation matrix is the identity matrix and the transformation is considered as a pure translation.

The *transfmat2screw* function computes the screw axis and angle corresponding to the input transformation matrix. The function first calculates the axis-angle representation (θ, ω) of the rotation matrix to find the distance along the screw axis, θ, as well as the six-element twist representation of the screw axis using the following equation:

$$S = \begin{bmatrix} w \\ v \end{bmatrix} \tag{12}$$

$$v = G^{-1}(\theta)p \qquad G^{-1}(\theta) = \frac{1}{\theta}I - \frac{1}{2}[w] + (\frac{1}{\theta} - \frac{1}{2}cot\frac{\theta}{2})[w]^2 \tag{13}$$

The angular velocity, ω, and linear velocity, v, are used to compute the screw axis representation using the following equations:

$$-\hat{s} \times q = v - \hat{s}h \qquad \hat{s} = \frac{\omega}{||\omega||} \qquad h = \frac{\omega \cdot v}{||\omega||} \tag{14}$$

The function also handles the case if the input rotation matrix is the identity matrix, meaning that the transformation is purely translation.

The *inputTransfmatScrew* function is the primary function used to complete the tasks for problem 3. It takes in an intial transformation configuration as well as a screw configuration represented by the 3-component screw axis and the distance along the screw, θ. It plots the initial transformation using the *plotTransformAxes* function. The function then calculates the intermediate and final transformation configurations using the *screw2transfmat* function and plots them on the same plot as the inital transformation. Finally, it calculates the screw axis and rotation angle to return to the origin using the inverse of the final transformation matrix as an input into the *transfmat2screw* function. The return to origin transformation is plotted on another plot along with the representation of the screw axis using the *plotScrewAxis* function.

The plotting function *plotTransformAxes* takes in a transformation matrix as an input, finds the origin of the frame using the position vector, calculates the endpoints of the axes using the rotation matrix, and then plots each axis in a different color. The *plotScrewAxis* function takes in the 3-component screw axis representation, extracts the coordinates from the point and the direction from the axis, and then plots the screw axis.

The helper function *is_valid_transformation_matrix* checks the validity of the input matrix by ensuring that it is real, has size 4x4, contains a valid rotation matrix, and a valid translation vector.

3.4 Test Functions

To validate the functions created in PA 1 and PA 2, test functions were created to debug any issues that could potentially arise. For the *rotmat2axisangle* and *rotmat2quaternion* functions, the test cases were 90° rotation around the z-axis, 60° rotation around the x-axis, the identity matrix, 180° rotation around the x-axis, the zero matrix, an invalid matrix, a large rotation angle, a small rotation angle, a negative rotation angle, a non-orthogonal matrix, a matrix with NaN values, a matrix with Inf values, and a matrix with values less than -1. These same test cases were used in reverse to test the *axisangle2rotmat* and *quaternion2rotmat* functions.

For the *rotmat2rollpitchyaw* function, the test cases were the identity matrix, the zero matrix, an invalid rotation matrix, a small rotation angle around the y-axis, a 90° rotation around the x-axis, a singularity case with a 90° rotation around the y-axis, a 60° rotation around the x-axis, and a 180° rotation around the z-axis. Finally, for the *rotmat2zyz* test function, the test cases used were a 90° rotation around the y-axis, the identity matrix, the zero matrix, an invalid rotation matrix, a small rotation angle around the y-axis, a rotation matrix with a set of angles, a singularity case, a rotation around the z-axis with a θ of 0, and a rotation around the z-axis with a θ of π.

To validate the functions for PA 3, a test function was created with the test case described in the problem statement. The initial transformation matrix was given as:

$$T = \begin{bmatrix} 1 & 0 & 0 & 2 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{15}$$

and the input screw axis representation and distance traveled along the screw being:

$$q = (0, 2, 0) \qquad \hat{s} = (0, 0, 1) \qquad h = 2 \qquad \theta = \pi \tag{16}$$

4 Results

The graph visualizations required for PA 3 are shown below. Figure 1 depicts the screw axis transformation using the provided inputs from the homework assignment. Figure 2 depicts the transformation of the final screw axis to the origin as well as the screw axis itself.

The functions for PA 1 and PA 2 were able to pass all of the test functions created. The explicit inputs and outputs for each test function can be found in the *Test_Reports* pdf.

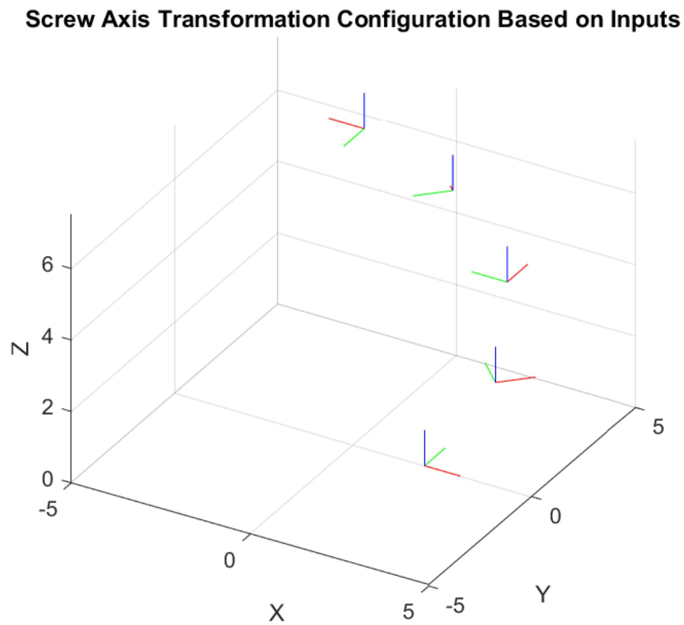

Figure 1: Visualization depicting the screw axis transformation performed in PA 3. The first axis shown is the initial transformation matrix. The other axes show the configuration of the transformation matrix as it moves along a given screw axis and rotation, with the intermediate and final configurations being shown.

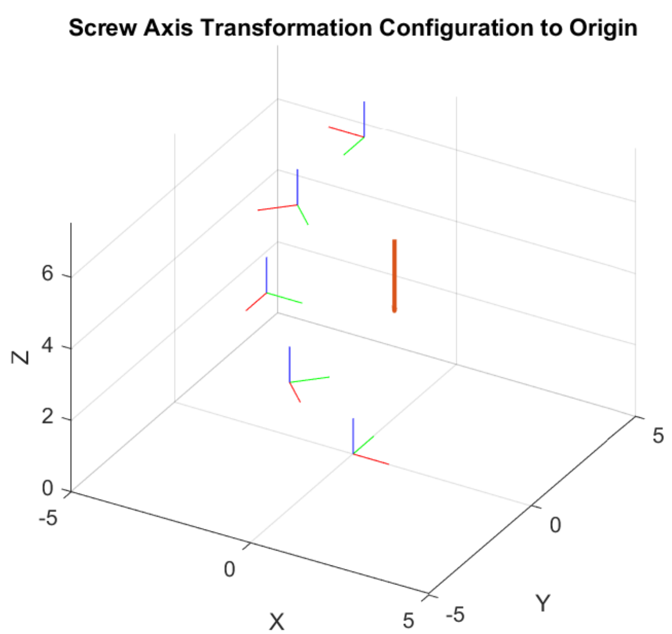

Figure 2: Visualization depicting the screw axis transformation performed in PA 3. The first axis shown is the final transformation matrix from the input calculation. The other axes shown are the final transformation returning to the origin. The red arrow represents the screw axis that the transformation travels along.

5 Discussion

The debugging process for this assignment involved creating test functions to identify and resolve potential issues. For PA 1 and PA 2, it was important to test a variety of cases to ensure that the created functions were viable for all types of inputs. Additionally, adding the helper function `is_valid_rotation_matrix` was useful in detecting invalid inputs early without expending the computing power in calculating incorrect representation. For PA 3, the functions were validated using a specific test case, and any issues were addressed during this process. The helper functions `is_valid_transformation_matrix` was useful in catching errors in the calculations early on. Additionally, the plotting functions *plotTransformAxes* and *plotScrewAxis* were useful in verifying the calculations by enabling visualization of the shape of the transformation. Finally, the `isequal_tol` helper function was also important to ensure that test cases could be verified smoothly.

6 Conclusion

This assignment involved creating functions that could convert between different representations of rotations such as rotation matrices, axis-angle representation, quaternions, and Euler angles. Additionally, functions were created that could convert between transformation matrices and screw axis representations. Finally, functions were created to calculate transformation configurations and plot them as the configuration travels along a screw axis representation. It was important to develop robust error handling in the created functions and implement various test cases to ensure that the functions were applicable for a variety of uses. Furthermore, the graphing functions were key in verifying proper calculations to ensure that transformation configurations were represented correctly.

7 Contributions

Daniyal Maroufi worked on HA 1, HA 3, HA 5, HA 7, HA 9, PA 1, and PA 3.
Anas Yousaf worked on HA 2, HA 4, HA 6, HA 8, PA 2, and PA 3.

References

[1] K. M. Lynch and F. C. Park, *Modern Robotics: Mechanics, Planning, and Control*, 1st ed. Cambridge University Press, May 2017. DOI: `10.1017/9781316661239`.

[2] *3D Rotation Converter*. [Online]. Available: `https://www.andre-gaschler.com/rotationconverter/` (visited on 02/18/2025).

[3] K. R. Driggs-Campbell, "Lecture 08 Rigid Body Motions,"

[4] K. R. Driggs-Campbell, "Lecture 09 transformations, twists, screws, and wrenches,"