Homogenous Transformation Matrices Notes

Sam Parsons

March 24, 2025

1 Introduction

This document is my interpretation of my accumulation of research regarding homogenous transformation matrices and their application related to carriage motion error.

2 Rotation Matrices

A rotation matrix is a matrix that is used to perform a rotation in Euclidean space. For example, a 2D rotation matrix is given by:

$$R(\theta) = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}$$

where θ is the angle of rotation. To rotate a point with coordinates (x_1, y_1) to a new point (x_2, y_2), around the origin, we can multiply the point by the rotation matrix:

$$\begin{bmatrix} x_2 \\ y_2 \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} * \begin{bmatrix} x_1 \\ y_1 \end{bmatrix}$$

when rotating the point (2,3) by 45 degrees counterclockwise, this is the result:

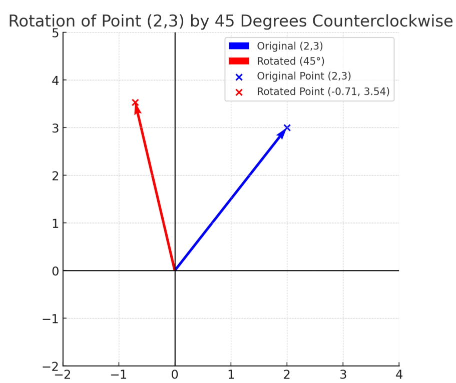

Figure 1: 2D Rotation Example

3 3D Rotation Matrices

In 3D, the individual rotation matrices change quite a bit. A rotation about the x-axis by an angle θ_x is given by:

$$R_x(\theta_x) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta_x & -\sin\theta_x \\ 0 & \sin\theta_x & \cos\theta_x \end{bmatrix}$$

A rotation about the y-axis by an angle θ_y is given by:

$$R_y(\theta_y) = \begin{bmatrix} \cos\theta_y & 0 & \sin\theta_y \\ 0 & 1 & 0 \\ -\sin\theta_y & 0 & \cos\theta_y \end{bmatrix}$$

And a rotation about the z-axis by an angle θ_z is given by:

$$R_z(\theta_z) = \begin{bmatrix} \cos\theta_z & -\sin\theta_z & 0 \\ \sin\theta_z & \cos\theta_z & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

A rotation around multiple axes (in this case $\theta_x, \theta_y, \theta_z$) can be achieved by multiplying the rotation matrices together:

$$R(\theta_x, \theta_y, \theta_z) = R_x(\theta_x)R_y(\theta_y)R_z(\theta_z)$$

This is the point (1,2,3) rotated by 30° around the x-axis, 60° around the y-axis, and 15° around the z-axis:

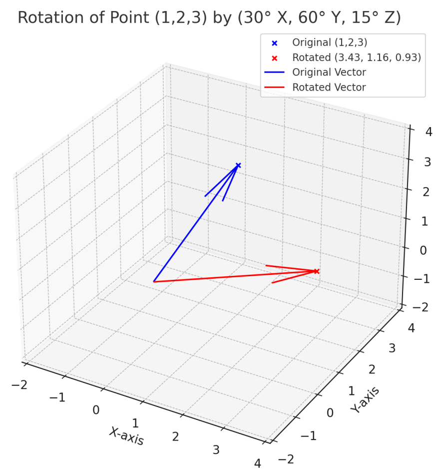

Figure 2: 3D Rotation Example

A full 3x3 rotation matrix can be created by multiplying together each individual axis' rotation matrix. multiplying the rotation matrices for the x, y, and z axes together yields the following 3x3 matrix:

$$R = \begin{bmatrix} \cos\theta_y \cos\theta_z & \cos\theta_z \sin\theta_x \sin\theta_y - \sin\theta_z \cos\theta_x & \cos\theta_z \cos\theta_x \sin\theta_y + \sin\theta_z \sin\theta_x \\ \sin\theta_z \cos\theta_y & \sin\theta_z \sin\theta_x \sin\theta_y + \cos\theta_z \cos\theta_x & \sin\theta_z \cos\theta_x \sin\theta_y - \cos\theta_z \sin\theta_x \\ -\sin\theta_y & \cos\theta_y \sin\theta_x & \cos\theta_y \cos\theta_x \end{bmatrix}$$

4 Adding Translation to 2D Rotation

Ading translation in 2d space is pretty simple. let's take our 2d rotation matrix from earlier:

$$\begin{bmatrix} x_2 \\ y_2 \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} * \begin{bmatrix} x_1 \\ y_1 \end{bmatrix}$$

The only modification to be made to the input matrix is the addition of a third row containing a 1:

$$\begin{bmatrix} x_2 \\ y_2 \\ 1 \end{bmatrix}$$

In order to add a translation into this matrix, we just add a third column to the rotation matrix, containing the values in which the point will be translated by. A third row is also added, containing two zeros and a 1. This is shown below:

$$\begin{bmatrix} x_2 \\ y_2 \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta & d_x \\ \sin\theta & \cos\theta & d_y \\ 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} x_1 \\ y_1 \\ 1 \end{bmatrix}$$

5 Building the final Transformation Matrix

The previous method can be extended to 3D space. recall the 3D rotation matrix mentioned earlier:

$$R = \begin{bmatrix} \cos\theta_y \cos\theta_z & \cos\theta_z \sin\theta_x \sin\theta_y - \sin\theta_z \cos\theta_x & \cos\theta_z \cos\theta_x \sin\theta_y + \sin\theta_z \sin\theta_x \\ \sin\theta_z \cos\theta_y & \sin\theta_z \sin\theta_x \sin\theta_y + \cos\theta_z \cos\theta_x & \sin\theta_z \cos\theta_x \sin\theta_y - \cos\theta_z \sin\theta_x \\ -\sin\theta_y & \cos\theta_y \sin\theta_x & \cos\theta_y \cos\theta_x \end{bmatrix}$$

All that we have to do now is add a 3x1 translation vector to the 3x3 rotation matrix, as well as a bottom row of three zeroes and a 1. This gives the final 4x4 transformation matrix:

$$R = \begin{bmatrix} \cos\theta_y \cos\theta_z & \cos\theta_z \sin\theta_x \sin\theta_y - \sin\theta_z \cos\theta_x & \cos\theta_z \cos\theta_x \sin\theta_y + \sin\theta_z \sin\theta_x & d_x \\ \sin\theta_z \cos\theta_y & \sin\theta_z \sin\theta_x \sin\theta_y + \cos\theta_z \cos\theta_x & \sin\theta_z \cos\theta_x \sin\theta_y - \cos\theta_z \sin\theta_x & d_y \\ -\sin\theta_y & \cos\theta_y \sin\theta_x & \cos\theta_y \cos\theta_x & d_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

The input vector is also extended to a 4x1 vector, with the last element being 1:

$$\begin{bmatrix} x_1 \\ y_1 \\ z_1 \\ 1 \end{bmatrix}$$

Combined, this yields the following:

$$\begin{bmatrix} x_2 \\ y_2 \\ z_2 \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\theta_y \cos\theta_z & \cos\theta_z \sin\theta_x \sin\theta_y - \sin\theta_z \cos\theta_x & \cos\theta_z \cos\theta_x \sin\theta_y + \sin\theta_z \sin\theta_x & d_x \\ \sin\theta_z \cos\theta_y & \sin\theta_z \sin\theta_x \sin\theta_y + \cos\theta_z \cos\theta_x & \sin\theta_z \cos\theta_x \sin\theta_y - \cos\theta_z \sin\theta_x & d_y \\ -\sin\theta_y & \cos\theta_y \sin\theta_x & \cos\theta_y \cos\theta_x & d_z \\ 0 & 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} x_1 \\ y_1 \\ z_1 \\ 1 \end{bmatrix}$$

6 Applying multiple transformations

We can also take our overall transformation matrix, and convert it to the following form for simplicity:

$$T_{A\to B} = \begin{bmatrix} R_{A\to B} & d_{A\to B} \\ 0 & 1 \end{bmatrix}$$

since sometimes multiple transformations are needed, we can multiply transformation matrices together. For example:

$$T_{A\to D} = T_{A\to B} T_{B\to C} T_{C\to D}$$

To do this using the full matrices, we can multiply the first two matrices together, then multiply the result by the third matrix.

Compute $T_{A\to B} \cdot T_{B\to C}$

$$\begin{bmatrix} R_{A\to B} & d_{A\to B} \\ 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} R_{B\to C} & d_{B\to C} \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} R_{A\to B} R_{B\to C} & R_{A\to B} d_{B\to C} + d_{A\to B} \\ 0 & 1 \end{bmatrix}$$

Then multiply that by $T_{C\to D}$ to get your answer

$$T_{A\to D} = \begin{bmatrix} R_{A\to B} R_{B\to C} & R_{A\to B} d_{B\to C} + d_{A\to B} \\ 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} R_{C\to D} & d_{C\to D} \\ 0 & 1 \end{bmatrix}$$

$$T_{A\to D} = \begin{bmatrix} R_{A\to B} R_{B\to C} R_{C\to D} & R_{A\to B} R_{B\to C} d_{C\to D} + R_{A\to B} d_{B\to C} + d_{A\to B} \\ 0 & 1 \end{bmatrix}$$

7 Application to linear carriage error

These principles are to be used to process raw straightness and angular error data into a usable and quantifiable format. For the sake of simplicity, I will be using an artificial data set comprising of 11 points across an "x-axis" spanning 100 mm, with straightness errors ranging between ±3 mm. Angular errors are all within ±5 arcseconds across each axis

d_x	d_y	d_z	$\boldsymbol{\theta_x}$	$\boldsymbol{\theta_y}$	$\boldsymbol{\theta_z}$
0	0	0	0	0	0
1	1	-0.25	-0.5	0.1	0.25
2	1.5	-0.5	-1	0.15	0.5
3	1	-0.75	-1.5	0.175	0.7
4	0.25	-1	-2	0.15	0.9
5	-0.75	-0.5	-2.5	0.075	1
6	-1.5	0.25	-3	-0.075	1.05
7	-2.25	0.75	-3.5	-0.1	0.95
8	-2	1.5	-4	-0.125	0.75
9	-1.75	2.25	-4.5	-0.25	0.45
10	-1.5	2.75	-5	-0.3	0.1

Table 1: Straightness and Angular Errors Moving Along X-axis

The artificial carriage is a rectangular prism, with the following dimensions:

- Length: 4 mm (arbitrary)
- Width: 6 mm
- Height: 2 mm

The true straightness error of this carriage in the Y and Z axes can be shown by plotting the error values vs X axis travel, then applying a linear regression to find a reference straight line. This plot is shown below:

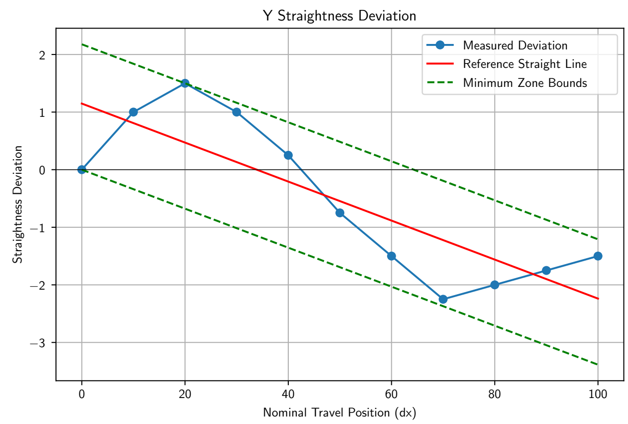

Figure 3: Straightness Error vs. Displacement

By taking the error bounds around the reference straight line, our true straightness error can be measured:

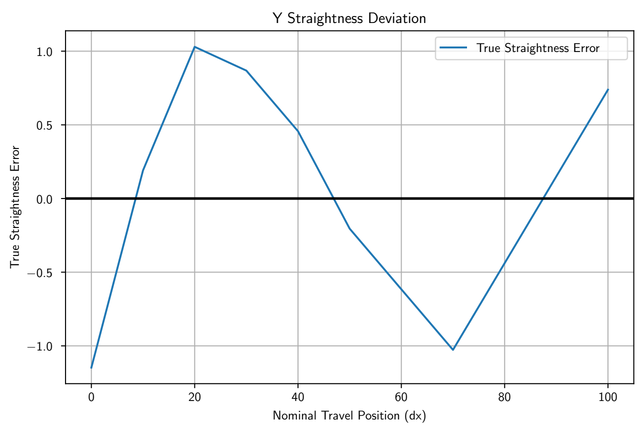

Figure 4: True Straightness

In this case, the true straightness error in the Y axis is around 2 mm. angular error can be shown just as a plot of the error values vs X axis travel. This plot, involving error around the Z axis (or yaw) is shown below:

When combined with the straightness error , the full homogenous transformation matrix can be used to map out motion. Take our carriage from earlier. Imagine a cross section of the carriage perpendicular to the X axis. By applying the homogenous transformation matrix to the points surrounding this cross section as the carriage moves across the X axis, a movement map can be generated.

This map is shown below:

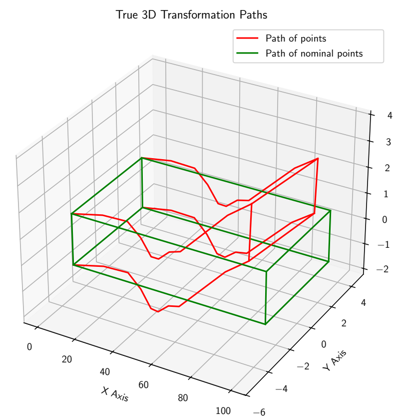

Figure 5: Carriage Motion Map

You can see the error in the carriage motion in the Y and Z directions. To emphasize the angular error, the angular values can be scaled. The plot below shows the same motion map, but with each angular error scaled by their own individual significant amounts:

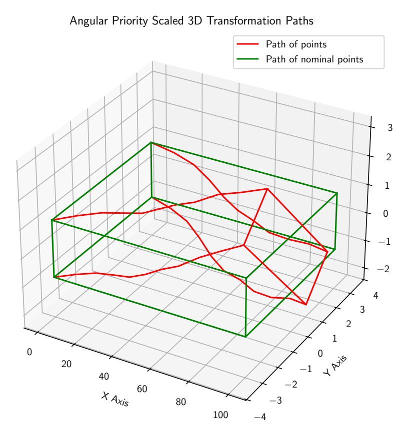

Figure 6: Carriage Motion Map with Angular Errors Scaled

This transformation and visualization can be applied to the piezoelectric actuated- motion stage.

8 Appendix