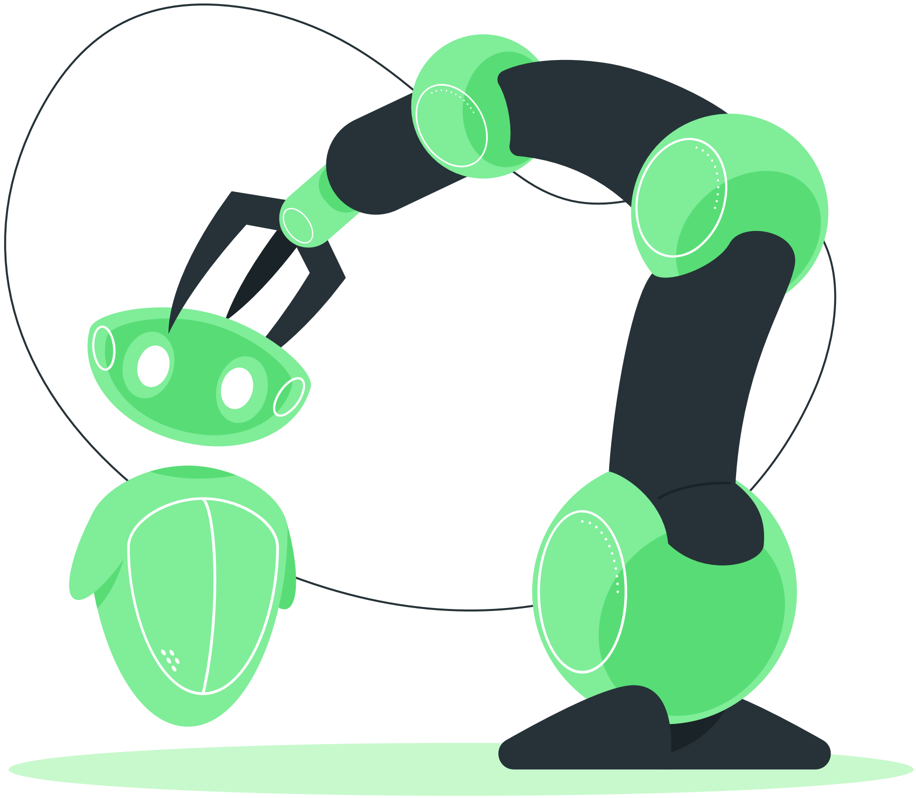
Application of Matrices in Robotics



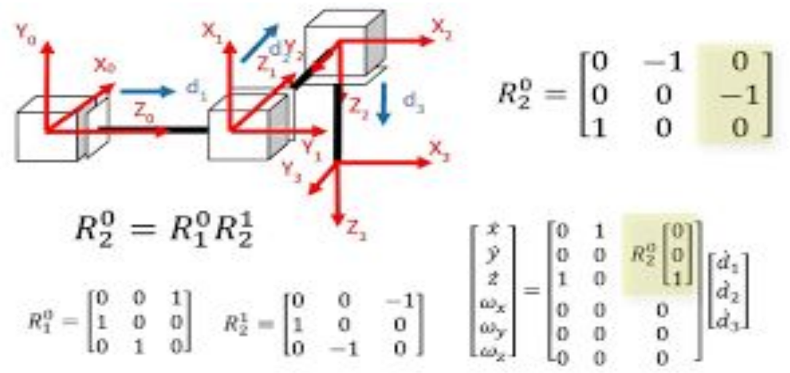
Presentation by
Sahil Maniyar
Amalu Manu
Roshan Mehta
Vishvavardhan Singh
Jatin Modi
Mucharla Harsha Vardhan
Vedika Mundra
Tanishka Singh

Table of Contents

Table of Contents

Homogeneous Transformation Matrices

In robotics, Homogeneous Transformation Matrices (HTM) have been used as a tool for describing both the position and orientation of an object and, in particular, of a robot or a robot component.

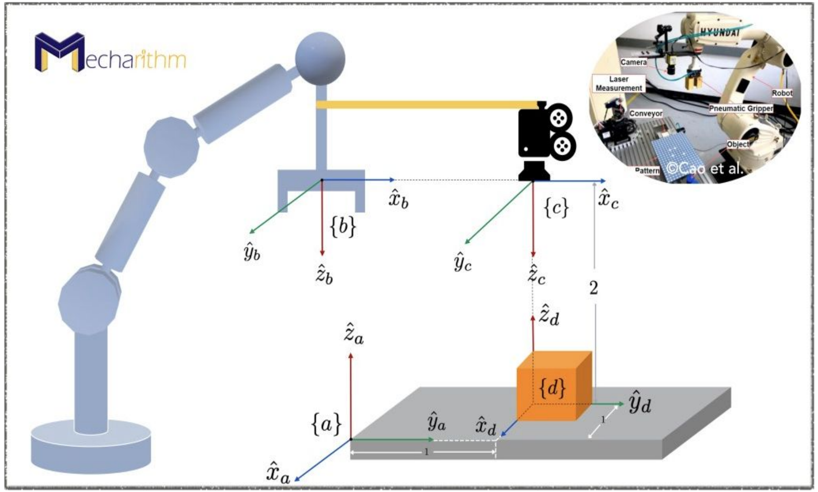


How transformation works in Robotics

- To describe the **angle** that the servos need to reach the **desired position** in space

- For example, how an autonomous underwater vehicle needs to reach or align itself with several different obstacles inside water

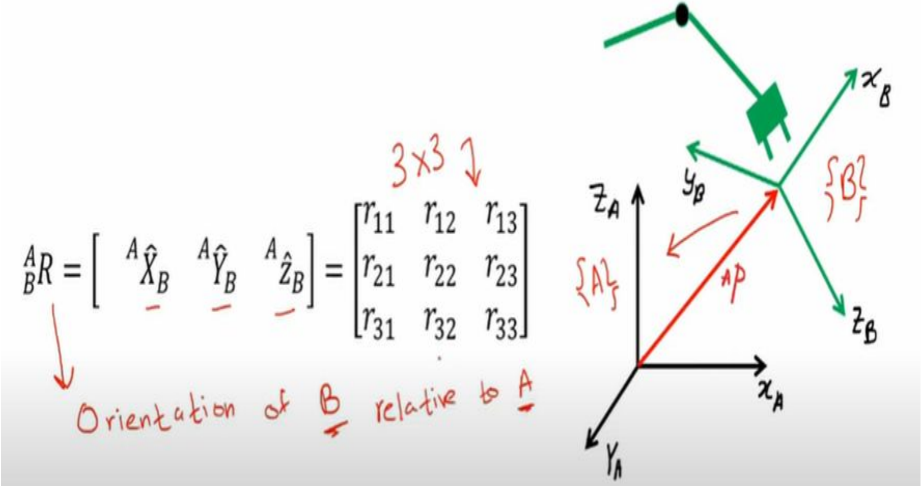
- It helps us to determine the **movement of the parts** of the robots with respect to one another



Orientation and Positional Representation

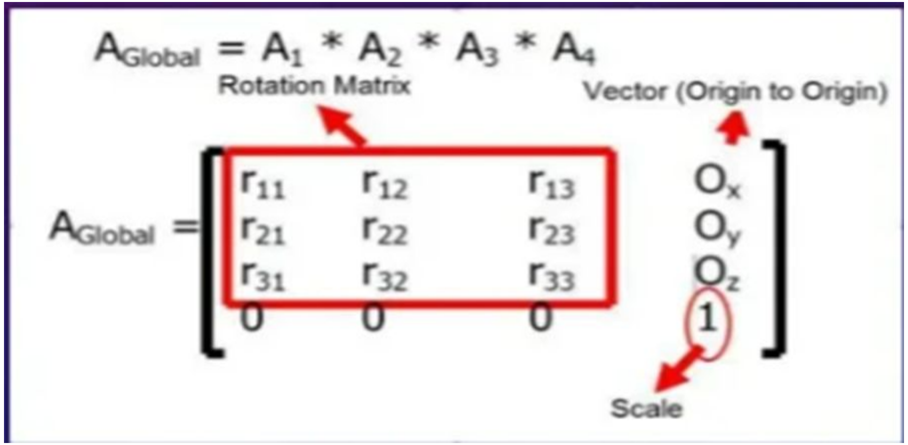
•A fundamental requirement in robotics is to represent the position and orientation of objects in an environment. such objects include robots, cameras, workpieces, obstacles, and paths. position and orientation together are referred to as pose.

• The columns of the rotational matrix form an sub-orientation matrix while the vector is the frame's origin offset.



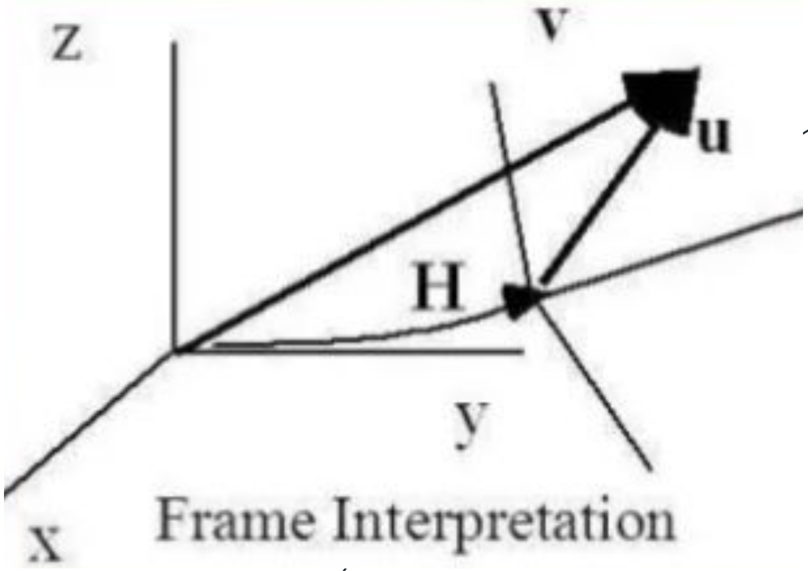
•Now here, column 1 is a vector that orients the frames's axis respectively. similar interpretations are made for the frame's y and z axes represented by columns 2 and 3.

•Also, the origin vector with three components represents the frame's origin relative to the reference axis.



Frame Interpretation of Transformation.

- Here, we have been given a vector “u”, and its transformation.
- Transformation is represented by v = Hu
- Now, this vector has components as Ux, Uy and Uz in a column, and it as to be extended to 4*1.

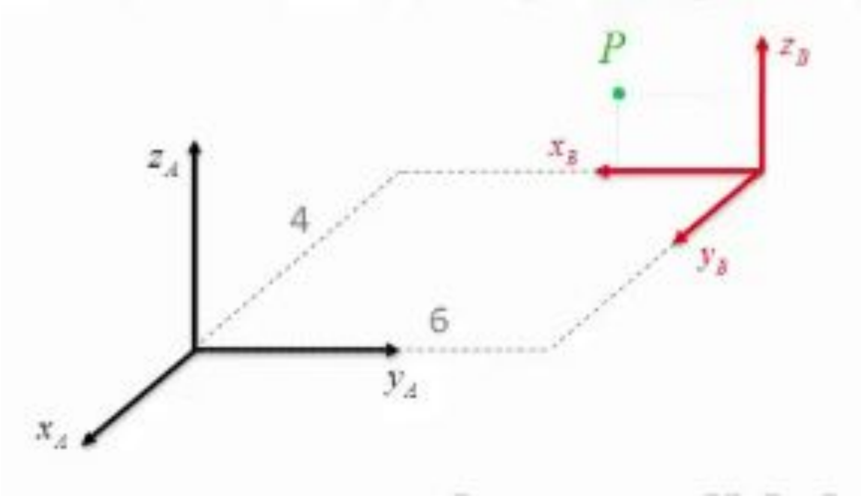


Graphical Representation

Considering frame {A} and {B}. Point P is given by (2,0,1), its coordinates with respect to frame {A] using homogeneous transformation matrix is :

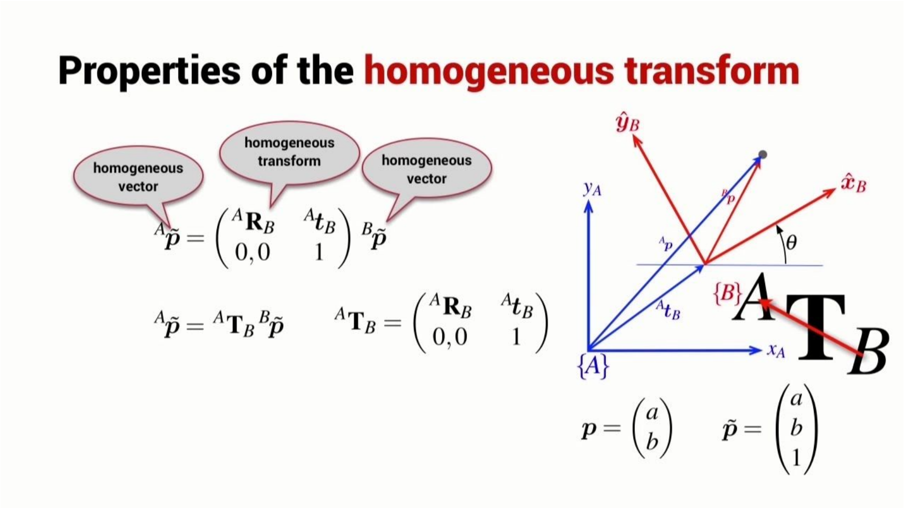
Solution

$$\begin{bmatrix} {}^{A}p \\ 1 \end{bmatrix} = \begin{bmatrix} {}^{A}R_{B} & {}^{A}t_{B} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} {}^{B}p \\ 1 \end{bmatrix} \Rightarrow \begin{bmatrix} {}^{A}p \\ 1 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & -4 \\ -1 & 0 & 0 & 6 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 2 \\ 0 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} -4 \\ 4 \\ 1 \\ 1 \end{bmatrix}$$

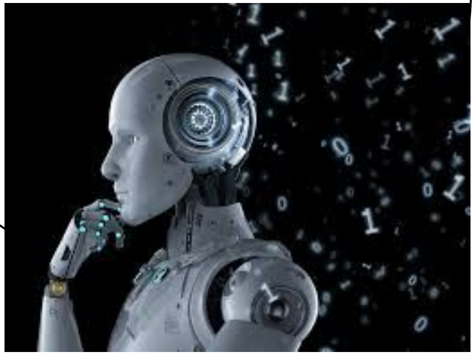


Theoretical Explanation

- The homogeneous Transformation effectively merges a frame orientation matrix, and frame translation vector into one matrix.
- The order of the operation should be viewed as rotation first, then translation.



- It can be viewed as position or orientation relationship of one frame relative to another frame called the reference frame.
- It can be interpreted as frame A described relative to the first or base frame while frame B is described relative to frame A. We can also interpret B in the base frame transformed by A in the base frame. Both the interpretations give the same result.



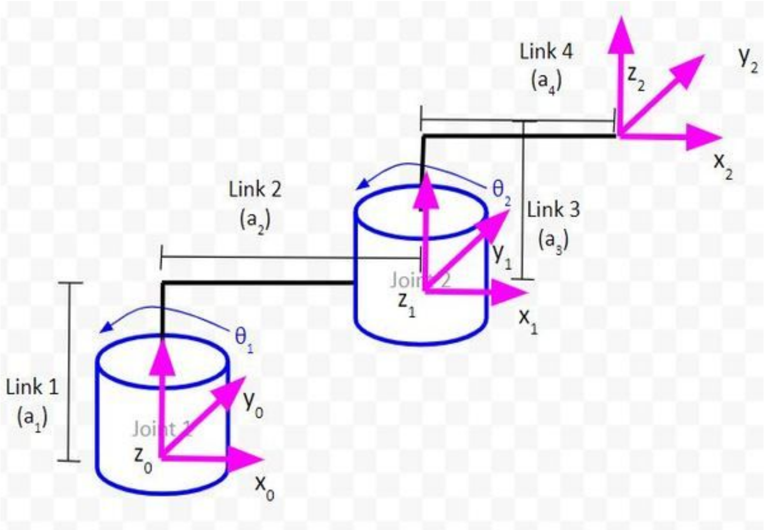
How to find Rotation Matrices for Robotic Arms

In the case of a two degree of freedom robotic arm, we have three coordinate frames (i.e. reference frames)

. the base coordinate frame (x_0, y_0, and z_0 axes in the image below) could be our global reference frame

. We then have a local reference frame (x_1, y_1, z_1) that is rotated at an angle from the global reference frame.

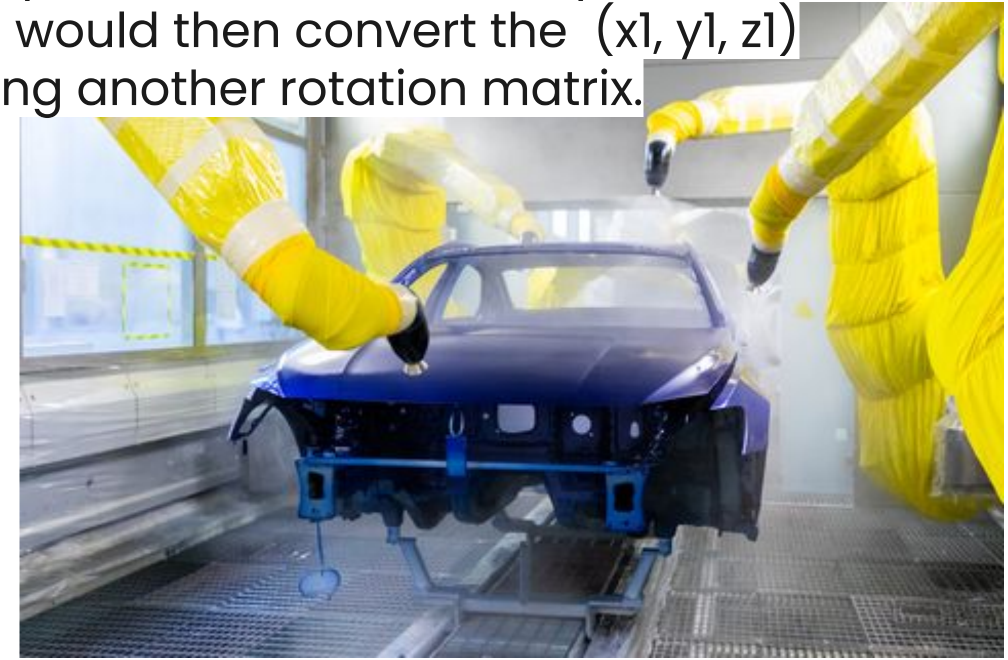
After that, we have another local reference frame (x_2, y_2, z_2) that is rotated relative to the local reference frame before it (x_1, y_1, z_1)



Imagine the base reference frame (x0, y0, z0) is a person. Our base frame might ask, ”Hey, I see the (x2, y2, z2) reference frame is rotated.

we would need to convert the (x2, y2, z2) frame to the (x1, y1, z1) frame using a rotation matrix. We would then convert the (x1, y1, z1) frame to the (x0, y0, z0) frame using another rotation matrix.

This is particularly useful for applications where you need the gripper to point a particular direction (e.g. a robotic arm painting an automobile).



How to find Displacement Vectors for Robotic Arms

rot_mat_0_1 =

$$\begin{bmatrix} \cos\Theta_1 & -\sin\Theta_1 & 0 \\ \sin\Theta_1 & \cos\Theta_1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Displacement in the x direction

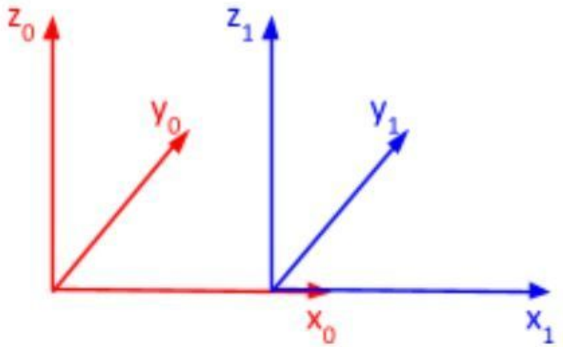
Rotation Matrix - You can see that frame 1 has no rotation relative to frame 0. Therefore, the first term we'll use to calculate the rotation matrix from frame 0 to 1 will be the identity matrix.

This rotation matrix shows how the axes of frame 1 project onto the axes of frame 0 when there is no rotation.1 is a rotation around the z_0 axis.

We therefore need to multiply the matrix we found above by the standard form of z matrix.

rot_mat_0_1 =

$$\begin{bmatrix} \cos\Theta_1 & -\sin\Theta_1 & 0 \\ \sin\Theta_1 & \cos\Theta_1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$



Displacement Vector - Now that we know how frame 1 is rotated relative to frame 0, we now need to know how frame 1 is displaced relative to frame 0.

Our displacement vector needs to have three elements:

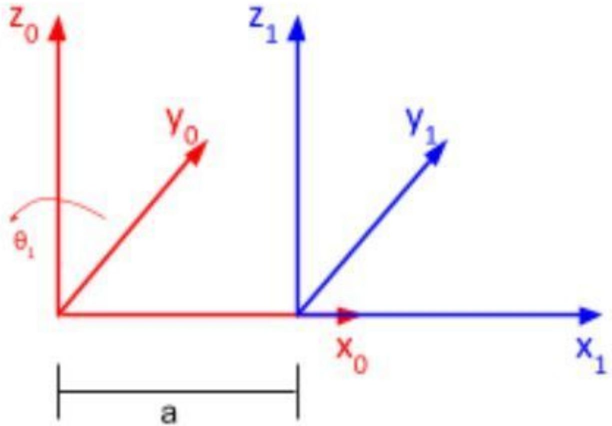
- Change in the position along the x-direction
- Change in the position along the y-direction
- Change in the position along the z-direction

For example, let's say that the distance from the origin of frame 1 to the origin of frame 0 is a. There is no displacement in the y and z directions.

Here is the displacement vector:

disp_vec_0_1 =

$$\begin{bmatrix} a \\ 0 \\ 0 \end{bmatrix}$$



Reference

https://www.researchgate.net/publication/322337716_Application_of_Homogeneous_Transformation_Matrices_to_the_Simulation_of_VLP_Systems

https://www.slideshare.net/SIMRANPARDESHI/use-of-matrix-in-robotics#:~:text=Transformation%20matrices%20can%20be%20used,different%20obstacles%20inside%20the%20water

https://automaticaddison.com/how-to-find-the-rotation-matrices-for-robotic-arms/#:~:text=In%20robotics%2C%20the%20orientation%20of,coordinate%20axes%20of%20another%20frame.

Thank You