

Coordinate Frames and Transforms

1 Specifying Position and Orientation

- We need to describe in a compact way the position of the robot. In 2 dimensions (planar mobile robot), there are 3 degrees of freedom (DOF): X, Y position and 1 angular orientation parameter θ
- In 3 dimensions there are 6 DOF: X, Y, Z position and 3 angular orientation parameters specifying orientation of the gripper in space. There are a number of ways to specify these angles which we will discuss later.
- In 2-D, we can specify both position and orientation using a translation vector (2x1 vector) and a rotation matrix (2x2) which encodes the orientation information.

2 2D Rotation Matrix

- Orthonormal matrix: columns are orthogonal basis vectors of unit length.
- Row vectors are also orthogonal unit vectors
- Determinant = 1 (Right handed system) -1(Left handed)
- Columns establish axes of new coordinate system with respect to previous frame

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

EXAMPLE:

$$ROT(90) = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \quad ROT(45) = \begin{bmatrix} .7 & -.7 \\ .7 & .7 \end{bmatrix}$$

Suppose we have 2 coordinate systems, A and B that differ by a rotation. If we have the coordinates of a point in coordinate system B , ${}^B P$, we can find the equivalent set of coordinates in coordinate system A by using the rotation matrix to transform the point from one system to the other:

$${}^A P = {}^A R_B {}^B P$$

The inverse rotation matrix $({}^A R_B)^{-1}$ is just the transpose of the original rotation matrix:

$$({}^A R_B)^{-1} = ({}^A R_B)^T = {}^B R_A$$

You can check this out by multiplying a 2D rotation matrix by its transpose which yields the identity matrix.

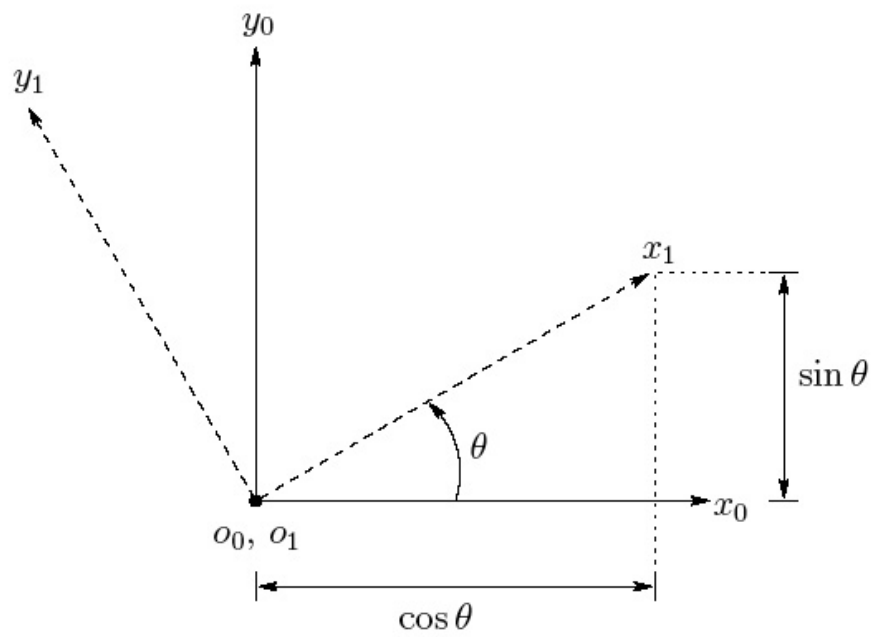
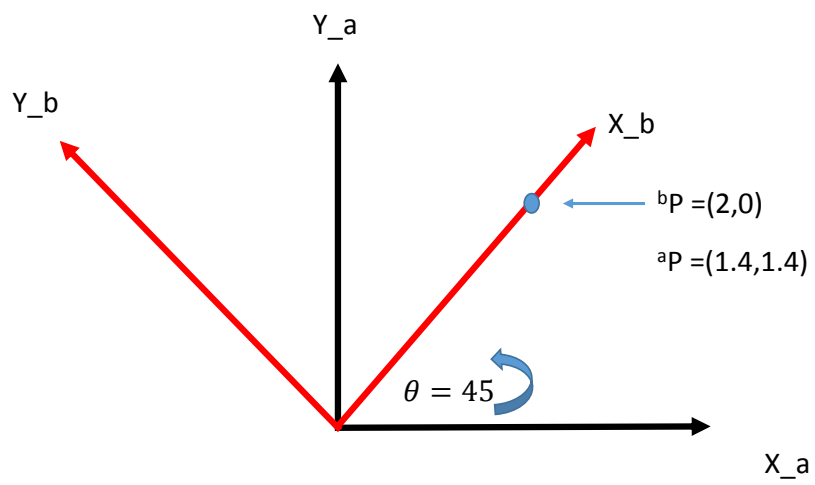


Figure 2.2: Coordinate frame $o_1x_1y_1$ is oriented at an angle θ with respect to $o_0x_0y_0$.

2D Rotation Matrix



$${}^aP = {}^aR_b {}^bP$$

$${}^aR_b = \begin{pmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{pmatrix} = \begin{pmatrix} .7 & -.7 \\ .7 & .7 \end{pmatrix}$$

Going left to right, rotations are done in the new or local frame established by the previous rotations. As we go right to left, the transformations are done in global coordinates.

3 Including Translations: Homogeneous Coordinates

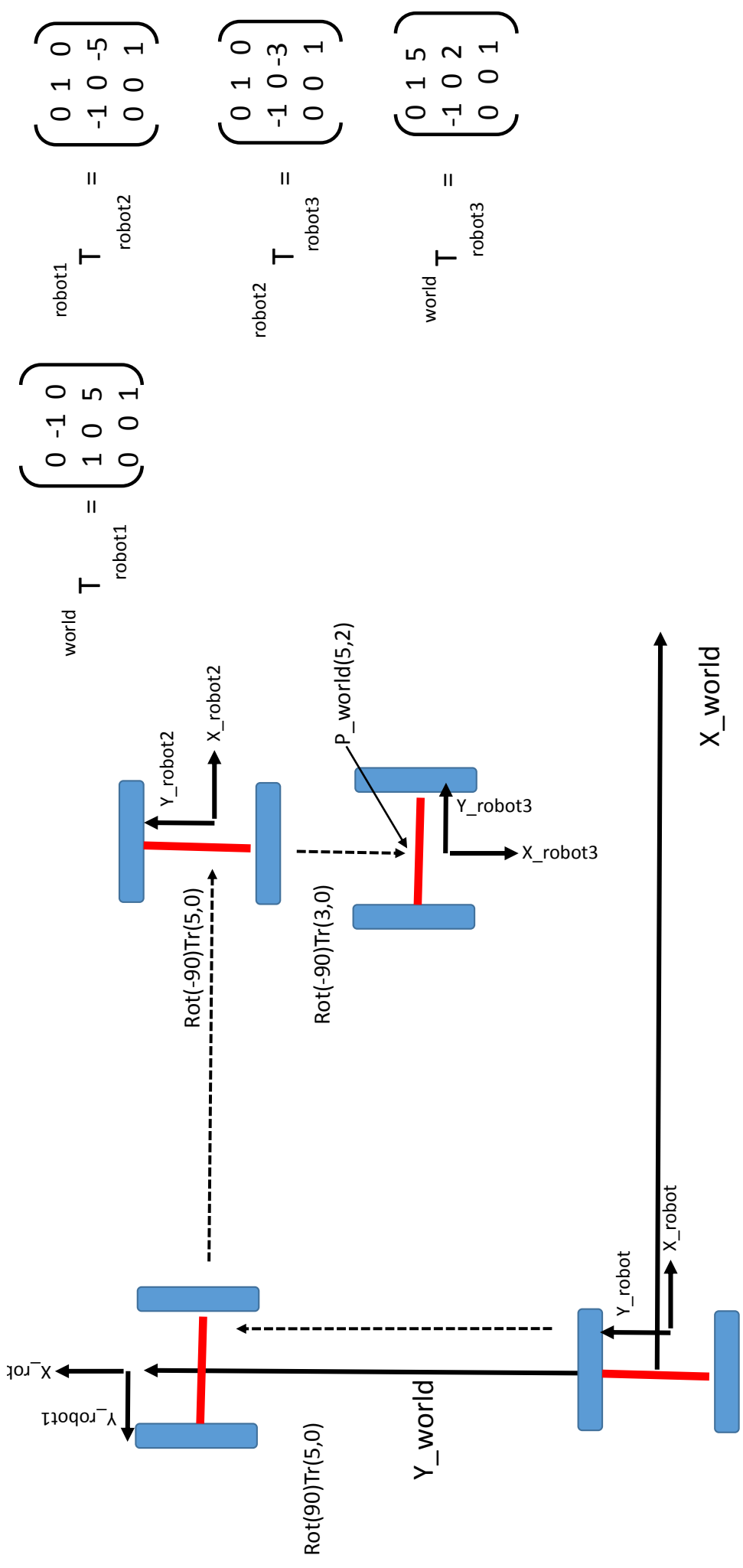
- When we want to establish a relationship between two 2D coordinate systems (we refer to these as coordinate **frames**), we need to represent this as a translation from one frame's origin to the new frame's origin, followed by a rotation of the axes from the old frame to the new frame.
- Transforming a 2-D point with a 2x2 matrix allows for scaling, shearing and rotation, but not translation.
- However we can use a method known as **homogeneous coordinates** to embed both a translation and rotation into one 3x3 matrix. You can think of this as embedding our 2D space in a 3D space.
- By using a 3x3 matrix, we can add translation to the transformation. Since we need to apply 3x3 matrices to 3-D vectors, we add an arbitrary scaling factor (typically with value 1) to the 2-D coordinates of a point to make it a 3x1 vector. You can think of the 2-D point as the projection into 2-D of an arbitrarily scaled 3-D point.
- By using a 4x4 matrix, we can add translation to the transformation. Since we need to apply 4x4 matrices to 4-D vectors, we add an arbitrary scaling factor (typically with value 1) to the 3-D coordinates of a point. You can think of the 3-D point as the projection into 3-D of a 4-D point.
- Homogeneous coordinates allow us to embed a lower dimensional space in a higher dimensional space. So a point in 2D space $[P_x, P_y]^T$ can be represented by a 3D point $[P_x, P_y, 1]^T$ where the third coordinate is an arbitrary scaling factor which we can also choose to be 1.

We can define a 3x3 transform from coordinate frame A to coordinate frame B as:

$${}^A T_B = \begin{bmatrix} \cos\theta & -\sin\theta & P_x \\ \sin\theta & \cos\theta & P_y \\ 0 & 0 & 1 \end{bmatrix}$$

Note that the first and second columns of the transform matrix specify the coordinates of the X and Y axes of the new coordinate frame. The third column is the origin of the new coordinate frame with respect to the previous frame. So in the transform above, the new X axis is pointing in direction $(\cos\theta, \sin\theta)$, and the new origin is at location (P_x, P_y) .

Homogeneous transforms contain BOTH rotation and translation information. The upperleft 2x2 matrix is the rotation matrix and the 2x1 third column vector is the translation. **It is important to remember that translation is done first, then rotation** when using a transform like this that embeds both rotation and translation.



$$\text{world } T_{\text{robot1}} = \begin{pmatrix} 0 & -1 & 0 \\ 1 & 0 & 5 \\ 0 & 0 & 1 \end{pmatrix}$$

$$\text{robot1 } T_{\text{robot2}} = \begin{pmatrix} 0 & 1 & 0 \\ -1 & 0 & -5 \\ 0 & 0 & 1 \end{pmatrix}$$

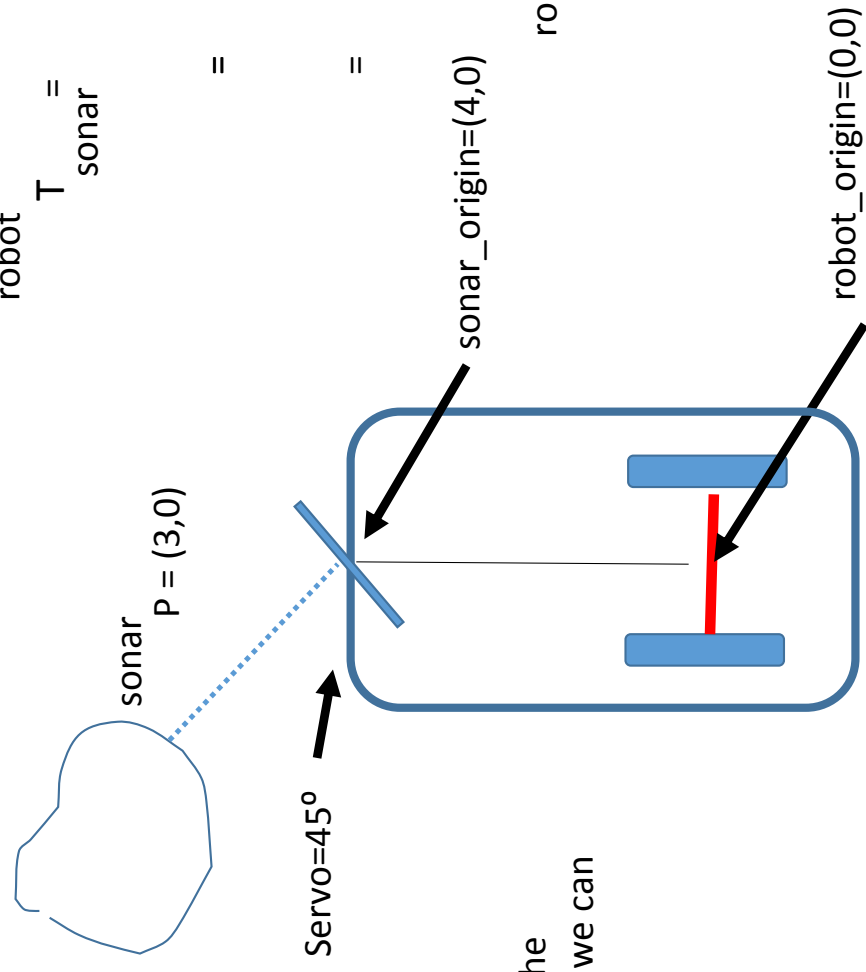
$$\text{robot2 } T_{\text{robot3}} = \begin{pmatrix} 0 & 1 & 0 \\ -1 & 0 & -3 \\ 0 & 0 & 1 \end{pmatrix}$$

$$\text{world } T_{\text{robot3}} = \begin{pmatrix} 0 & 1 & 5 \\ -1 & 0 & 2 \\ 0 & 0 & 1 \end{pmatrix}$$

Robot Coordinate Frames and Transforms:

Robot rotates 90, moves forward 5, rotates -90, moves forward 5, rotates -90, moves forward 3

$$\begin{aligned}
 \text{robot } T_{\text{sonar}} &= \begin{bmatrix} \cos(90-\text{servo}) & -\sin(90-\text{servo}) & 4 \\ \sin(90-\text{servo}) & \cos(90-\text{servo}) & 0 \\ 0 & 0 & 1 \end{bmatrix} \\
 &= \begin{bmatrix} \cos(90-45) & -\sin(90-45) & 4 \\ \sin(90-45) & \cos(90-45) & 0 \\ 0 & 0 & 1 \end{bmatrix} \\
 &= \begin{bmatrix} .7 & -.7 & 4 \\ .7 & .7 & 0 \\ 0 & 0 & 1 \end{bmatrix}
 \end{aligned}$$



If we find an obstacle in the sonar coordinate system, we can find its coordinates in the robot system.

$$\begin{aligned}
 \text{robot } P_{\text{sonar}} &= \begin{bmatrix} .7 & -.7 & 4 \\ .7 & .7 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 3 \\ 0 \\ 1 \end{bmatrix} \\
 &= \begin{bmatrix} 6.1 \\ 2.1 \\ 1 \end{bmatrix}
 \end{aligned}$$

Mapping sensor values to locations in space

4 Extensions to 3D

Similar to what we did in 2D, we can also specify rotations and translations in 3D using homogeneous coordinates. We can represent a point $[x, y, z]^T$ in 3D as a 4D homogeneous vector $[x, y, z, 1]^T$.

To specify rotation, we use a 3D rotation matrix. Since we can rotate about any of the three axes (X, Y, or Z) we can specify each canonical rotation matrix:

$$ROT(Z, \theta) = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$ROT(X, \theta) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta \\ 0 & \sin\theta & \cos\theta \end{bmatrix}$$

$$ROT(Y, \theta) = \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix}$$

Finally we can add translation in the 4th column of the transform matrix to define a transform from coordinate system i to $i + 1$:

$${}^i T_{i+1} = \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

The upper left 3x3 matrix is the rotation and the last column is the translation, and n, o, a are the unit vectors of the $i + 1$ frame's X, Y, Z axes relative to frame i , and frame $i + 1$'s origin is at $[p_x, p_y, p_z]^T$ relative to frame i .

We can also define an **inverse** transform. To calculate the inverse of 4x4 homogeneous transform, we simply take the transpose of the 3x3 rotation matrix, and use the negated dot products of the original translation against each column of the original transform:

$$T = \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T^{-1} = \begin{bmatrix} n_x & n_y & n_z & -p \cdot n \\ o_x & o_y & o_z & -p \cdot o \\ a_x & a_y & a_z & -p \cdot a \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

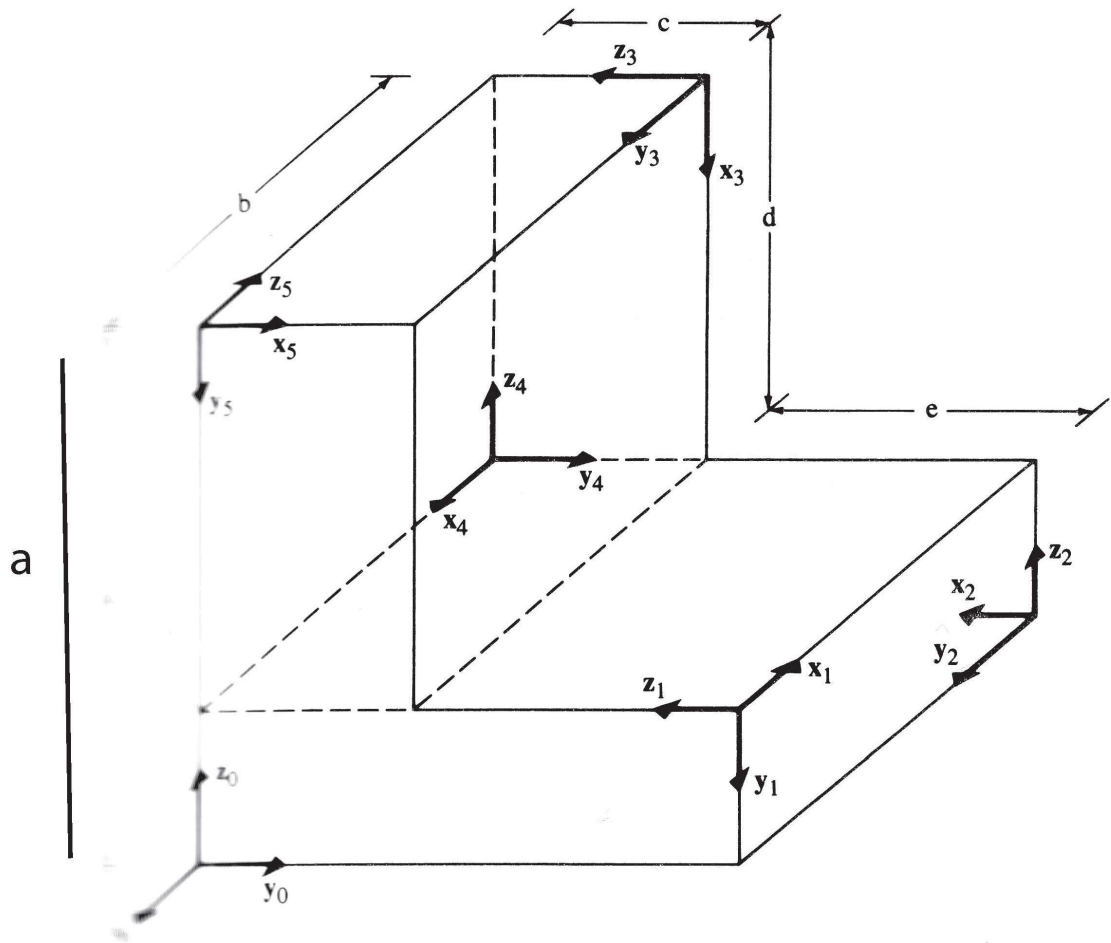


Figure 1: Exercise: try to write transforms for frames 0T_1 , 0T_2 , 3T_5 , and 0T_5 . Also prove that ${}^0T_3 = {}^0T_1 {}^1T_2 {}^2T_3$