RBE 3005
วิศวกรรมหุ่นยนต์ (Robotics Engineering)

สาขาวิศวกรรมหุ่นยนต์

คณะวิศวกรรมศาสตร์และเทคโนโลยีอุตสาหกรรม

มหาวิทยาลัยราชภัฏสวนสุนันทา

บทที่ 2 เฟรม พิกัด และการแปลงเอกพันธุ์

- ทำไมถึงต้องเรียนรู้จลนศาสตร์ (kinematics)
- นิยาม ศัพท์พื้นฐาน ที่ต้องรู้
  - องศาอิสระ degree of freedom
  - Joint space (joint configuration)
  - Operational space (Operational configuration)
  - พื้นที่ทำงาน workspace
  - Dexterous space
- การแปลงเอกพันธุ์ (Homogeneous Transformation)
  - เฟรม
  - ตำแหน่ง position
  - โอเรียนเทชั่น orientation
  - ทรานส์ฟอร์มเมชั่น transformation

ทำไมถึงต้องเรียนรู้จลนศาสตร์ (kinematics)

Kinematics คือการคำนวณหาค่าตำแหน่ง ความเร็ง และความเร่ง ที่จุดต่างๆ ของวัตถุแข็งเกร็งในระบบพลศาสตร์ โดยใช้วิธีทางคณิตศาสตร์ โดยไม่ได้นำแรงที่กระทำแก่วัตถุมาพิจารณา แยกเป็น

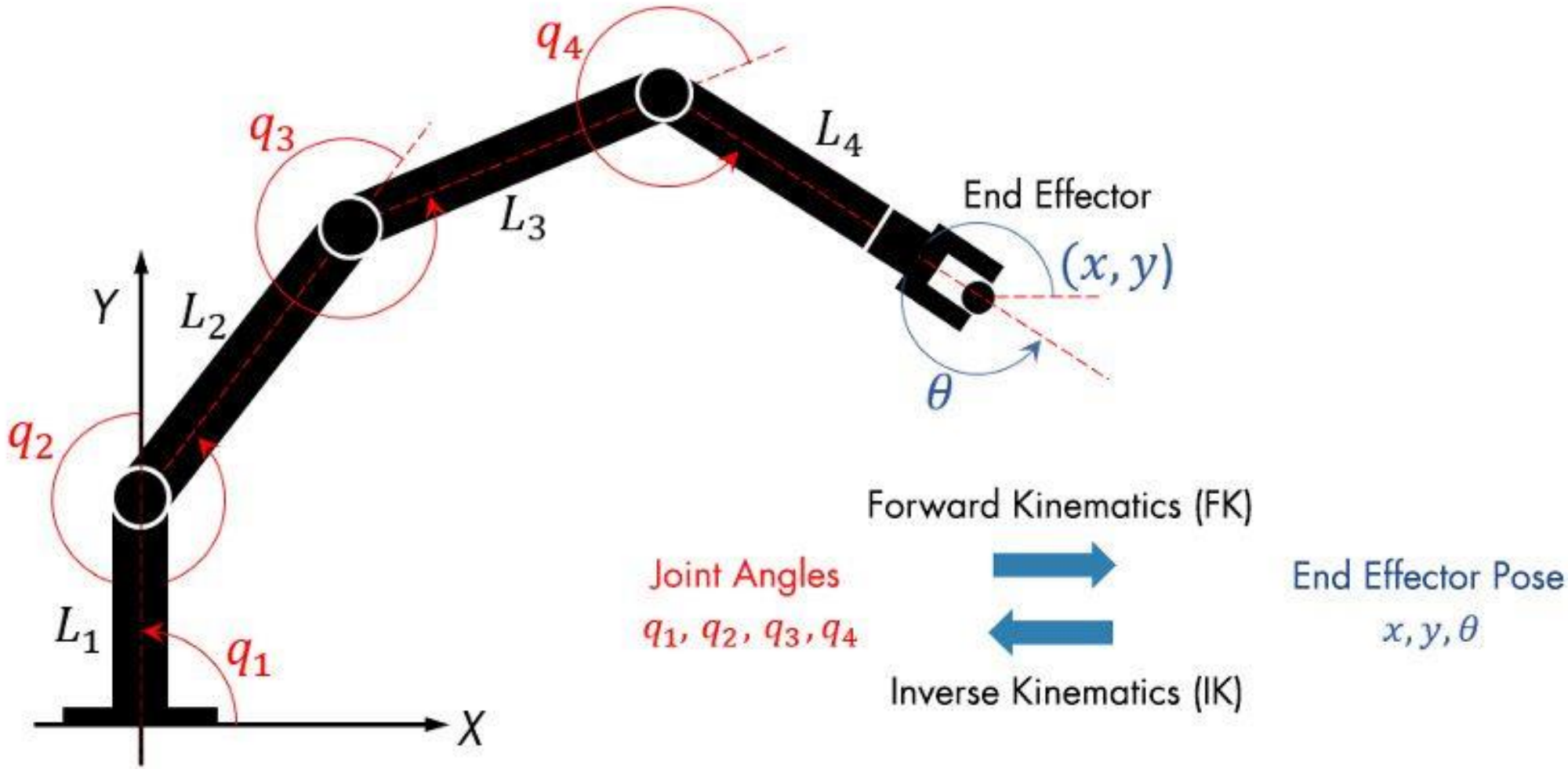
- **จลนศาสตร์ข้างหน้า** (Forward kinematics)

การระบุตำแหน่ง คำนวณหาความเร็ว ความเร่ง ของเฟรมเครื่องมืออ้างอิงกับเฟรมฐาน เมื่อรู้ค่าตำแหน่ง ความเร็ว ความเร่ง ของข้อต่อต่างๆ

- **จลนศาสตร์ผกผันกลับ** (Inverse kinematics)

การคำนวณย้อนกลับมาหาตำแหน่ง ความเร็ว ความเร่ง ของตัวแปรที่ข้อต่อ เมื่อรู้ค่าตำแหน่ง ความเร็ว ความเร่ง ของเฟรมเครื่องมือ

เนื่องจากหุ่นยนต์มีโครงสร้างและองค์กอบที่ซับซ้อน จึงจำเป็นต้องมีกรรมวิธี ขั้นตอนในการพิจารณา วิเคราะห์ เพื่อเป็นหลักในการวิเคราะห์ คำนวณ ดังนั้นการติดตั้ง**เฟรม**ให้กับข้อต่อและส่วนต่างๆ ของหุ่นยนต์อย่างเป็นระบบจะช่วยให้การคำนวณจลนศาสตร์ของเฟรมเป็นไปอย่างมีระบบและได้ผลลัพธ์ที่แม่นยำ


q_4
q_3
L_4
L_3
End Effector
(x, y)
Y
L_2
q_2
θ
L_1
q_1
X
Forward Kinematics (FK)
Joint Angles
q_1, q_2, q_3, q_4
End Effector Pose
x, y, θ
Inverse Kinematics (IK)

พลศาสตร์ (dynamics)

- สมการความสัมพันธ์ของตำแหน่ง ความเร็ว ความเร่ง เมื่อมีแรงมากระทำ แรงมีทั้งแรงบิดและแรงเชิงเส้น

$$\sum F = ma$$

$$\sum T = J\omega$$

วิศวกรหุ่นยนต์มีหน้าที่ออกแบบโครงสร้างทางกลของหุ่นยนต์จะต้องทราบแรงและแรงบิดที่เกิดที่ข้อต่อต่างๆ เพราะต้องนำไปคำนวณหาแรง แรงบิดที่ต้องใช้สำหรับขับเคลื่อนข้อต่อ และเพื่อเลือกขนาดของตัวขับเคลื่อนและมอเตอร์ หรือขนาดของโหลดสูงสุดที่รับได้สำหรับขาดของข้อต่อ ก้านต่อที่ออกแบบ
ดังนั้นสมการคณิตศาสตร์ ซึ่งเป็นโมเดลของหุ่นยนต์จะถูกพัฒนาเพื่อใช้ในการคำนวณหาค่าต่างๆ และถูกนำไปอ
แบบตัวควบคุมเพื่อให้หุ่นยนต์สามารถทำงานได้สำริดผลตามที่โปรแกรมเอาไว้

นิยาม ศัพท์พื้นฐาน ที่ต้องรู้

- **องศาอิสระ** degree of freedom

การเคลื่อนที่ในระบบสามมิติจะต้องการพารามิเตอร์อันหนึ่งที่ซึ่งสามารถบรรยายค่าตำแหน่งและมุมการหมุน จำนวนพารามิเตอร์อันนี้ที่เป็นอิสระต่อกันที่ใช้ในการบรรยายการเคลื่อนที่ของวัตถุเรียกว่า องศาอิสระ degree of freedom องศาอิสระจะถูกจำกัดลงจนเหลือค่าที่น้อยที่สุด เกิดจาการนำโครงสร้างทางกลและข้อต่อมาสร้างเป็นเครื่องจักรกล เช่นหุ่นยนต์ รถยนต์ เป็นต้น ค่า dof ที่น้อยที่สุด บ่งว่าต้องการอินพุทกี่ตัว

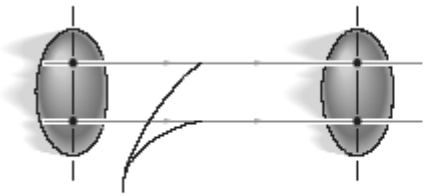
- Joint space (joint configuration)
- Operational space (Operational configuration)
  - พื้นที่ทำงาน (workspace) คือ เซตทั้งหมดของตำแหน่งที่ตัวทำงานส่วนปลายเข้าถึงได้
  - พื้นที่ทำงานได้คล่องแคล่ว (Dexterous space) คือ พื้นที่ที่ตัวทำงานส่วนปลายเข้าถึงได้ทุกทิศทาง

การเคลื่อนที่ของวัตถุแข็งเกร็ง

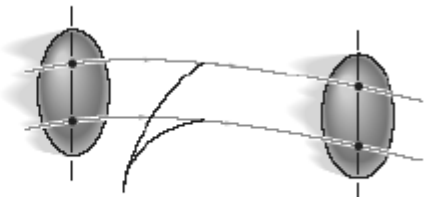
- วัตถุแข็งเกร็ง **(rigid body)** หมายถึง วัตถุที่รูปร่างไม่เปลี่ยนแปลงในขณะที่เคลื่อนที่ และตำแหน่งสัมพัทธ์ระหว่างจุดในวัตถุคงที่ ก

การเคลื่อนที่ของวัตถุมีได้สามแบบ

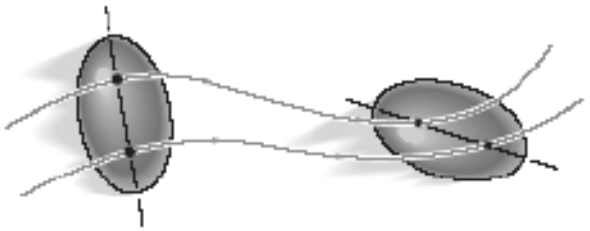
- การเลื่อน
- การหมุน
- การเลื่อนผสมการหมุน



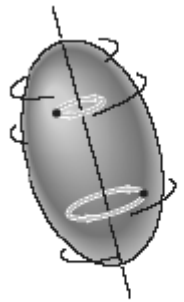
Path of rectilinear translation
(a)



Path of curvilinear translation
(b)



General plane motion
(d)



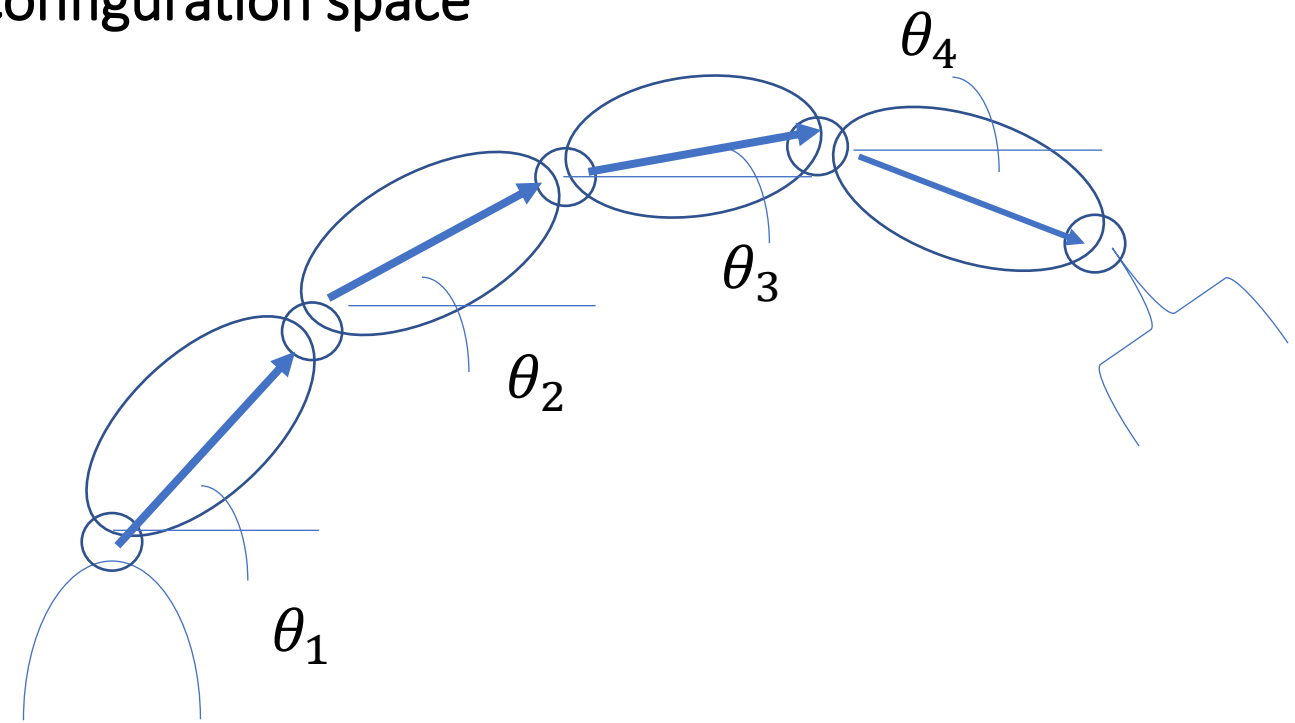
Rotation about a fixed axis
(c)

เพื่อการวิเคราะห์การเคลื่อนที่ เราติดพิกัดไปยังวัตถุ เรียนกว่า เฟรม

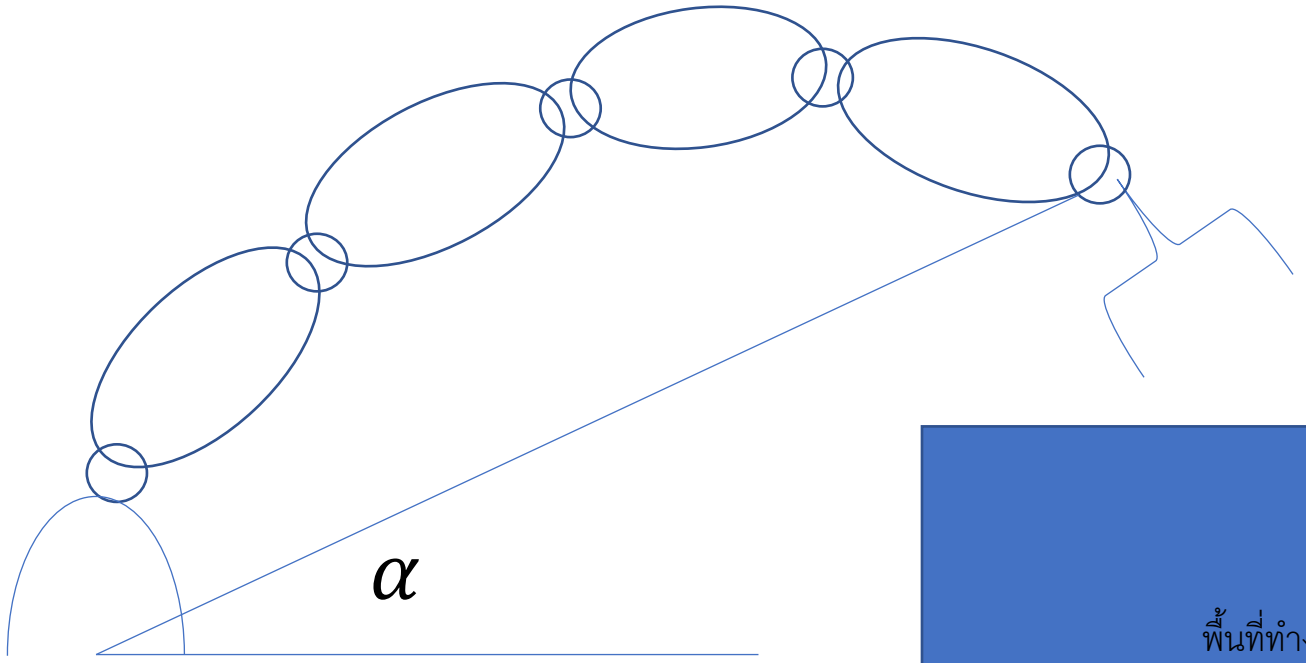
https://youtu.be/SOESSCXGhFo?feature=shared

Joint space (joint configuration)

The space in which the location of all the links of a **robot** are represented is called **joint space**, or **configuration space**



Operational space (Operational configuration)



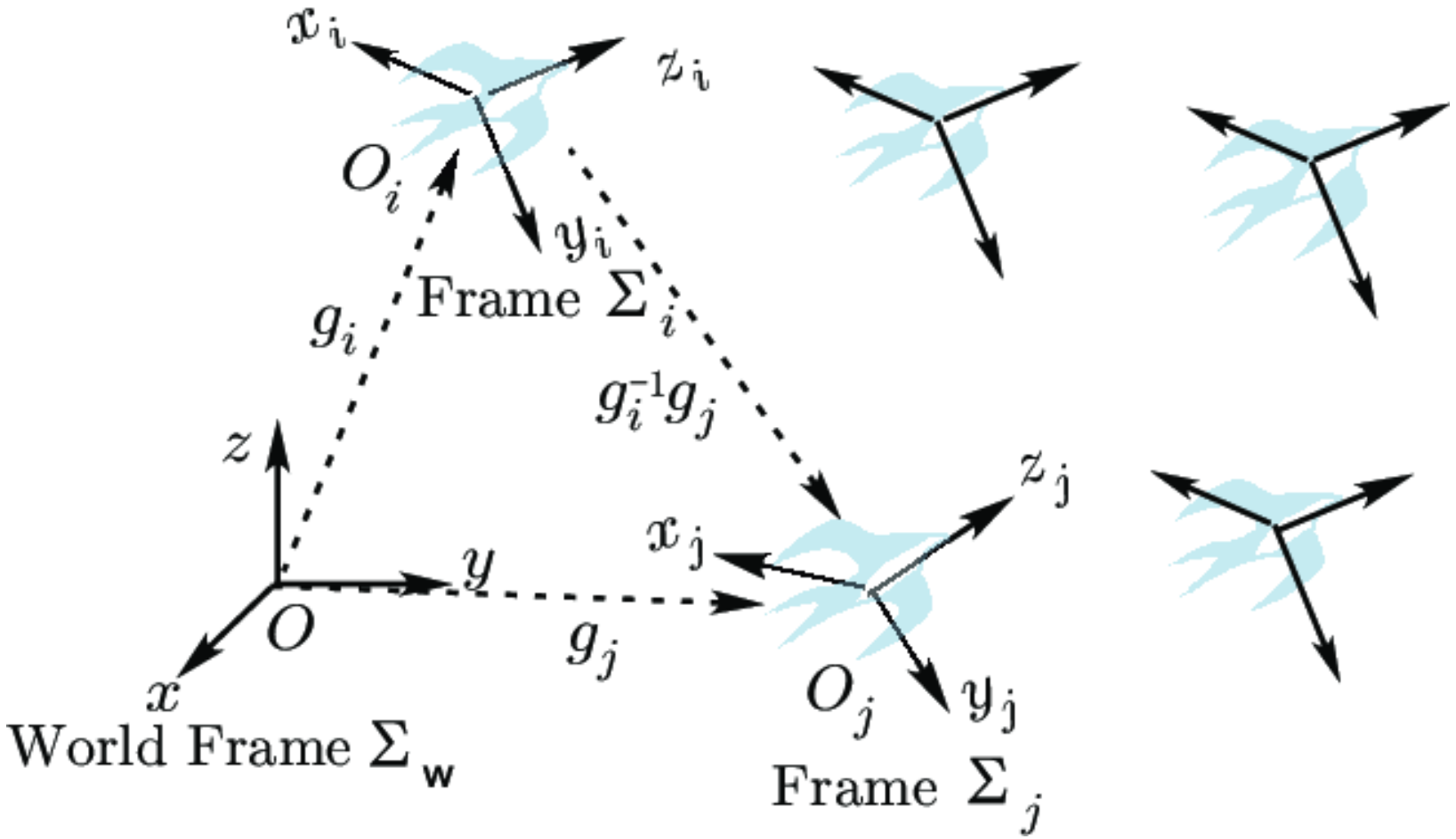
พื้นที่ทำงาน workspace

Dexterous space

Review Math

- Dot Product
- Cross Product
- Matrix
  - Det
  - Inverse matrix
  - Other properties

Description: POSITION,ORIENTATION,FRAMES



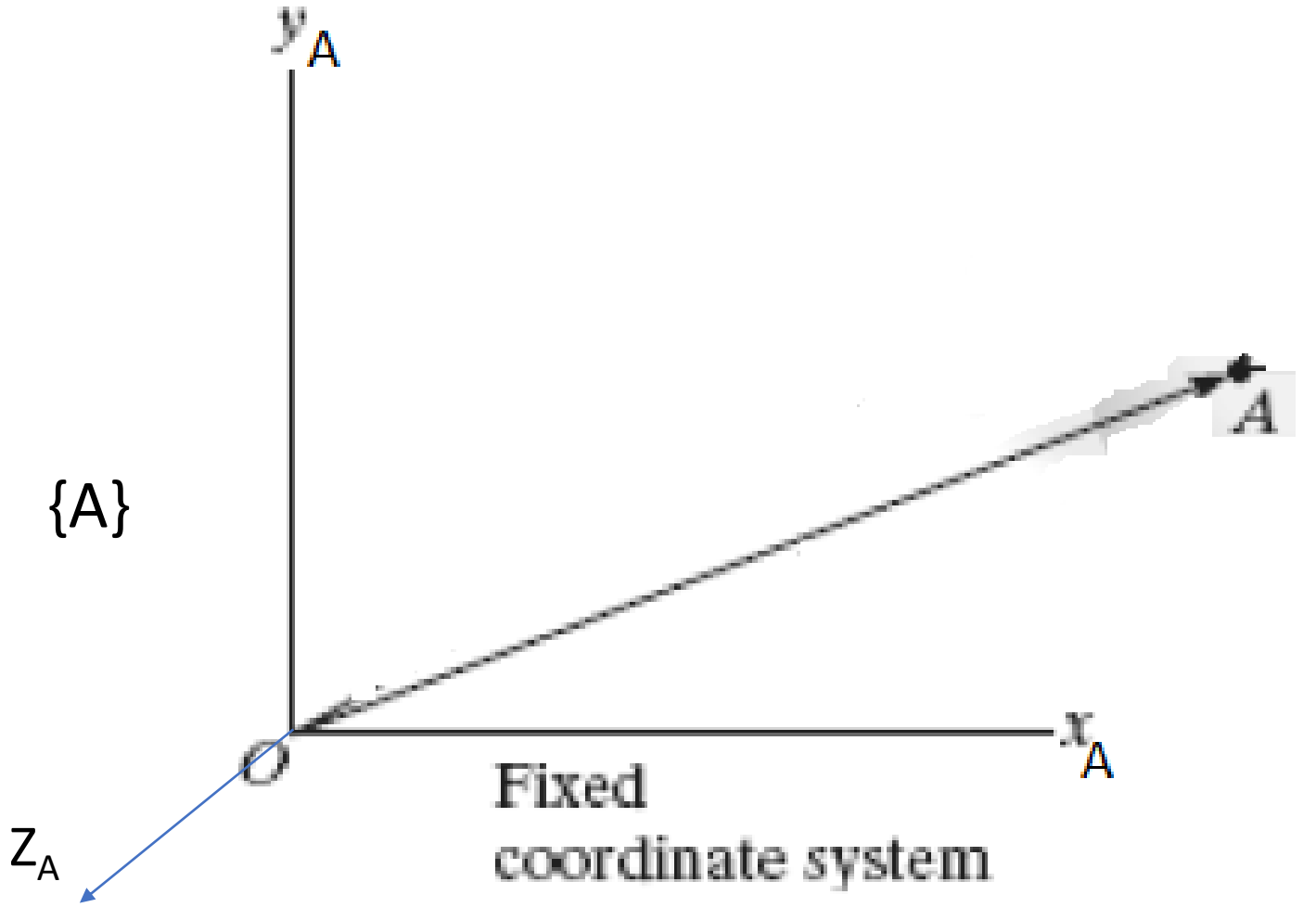
Description of POSITION

Coordinate system { A }

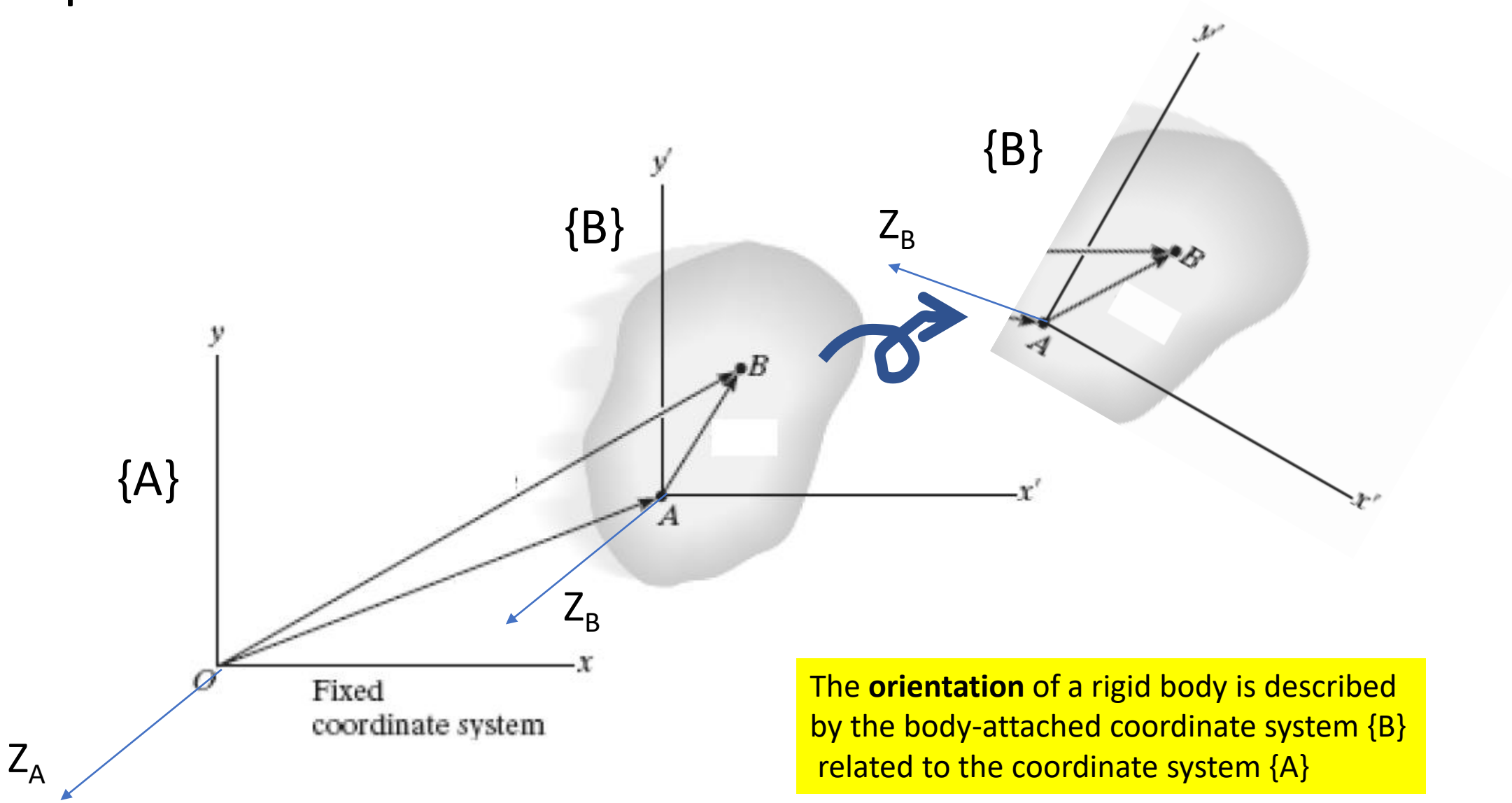
$$^{A}P = \begin{bmatrix} p_x \\ p_y \\ p_z \end{bmatrix}$$

{A}

Position of point A is described by a vector.



Description of ORIENTATION



The **orientation** of a rigid body is described by the body-attached coordinate system {B} related to the coordinate system {A}

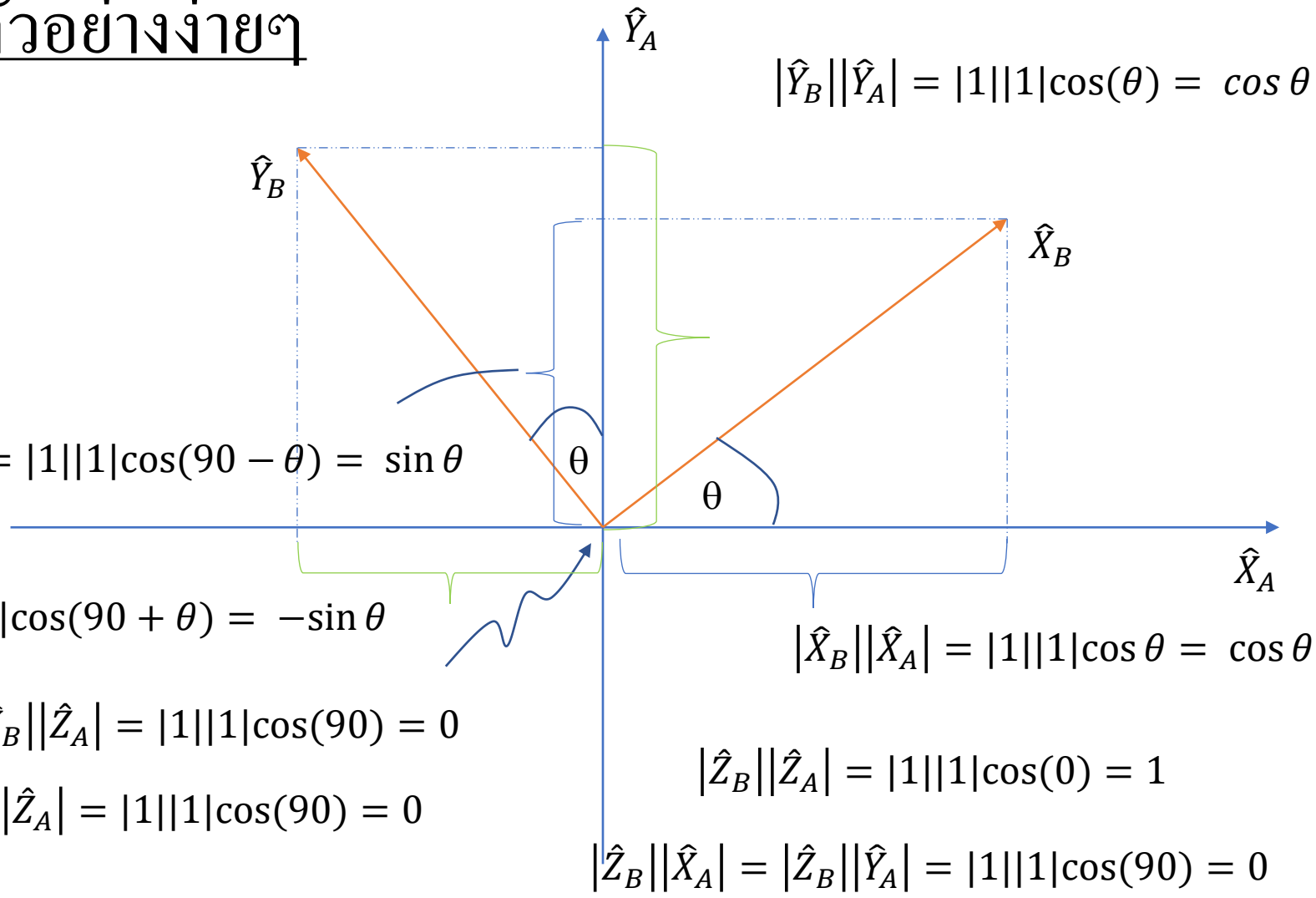
Rotational Matrix

จากหลักการที่ว่า องค์ประกอบของเวกเตอร์ ใดๆ สามารถเขียนได้ด้วย
ผลของการฉายของมันลงบนแกน x – y – z ของกรอบอ้างอิง (frame) อันนั้นๆ

$$
{}^{A}_{B}R = \begin{bmatrix} {}^{A}\hat{X}_B & {}^{A}\hat{Y}_B & {}^{A}\hat{Z}_B \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}
$$

$$
= \begin{bmatrix} \hat{X}_B \cdot \hat{X}_A & \hat{Y}_B \cdot \hat{X}_A & \hat{Z}_B \cdot \hat{X}_A \\ \hat{X}_B \cdot \hat{Y}_A & \hat{Y}_B \cdot \hat{Y}_A & \hat{Z}_B \cdot \hat{Y}_A \\ \hat{X}_B \cdot \hat{Z}_A & \hat{Y}_B \cdot \hat{Z}_A & \hat{Z}_B \cdot \hat{Z}_A \end{bmatrix} = \begin{bmatrix} {}^{B}X_A^T \\ {}^{B}Y_A^T \\ {}^{B}Z_A^T \end{bmatrix}
$$

พิสูจน์จากตัวอย่างง่ายๆ



$$|\hat{Y}_B||\hat{Y}_A| = |1||1|\cos(\theta) = \cos\theta$$

$$|\hat{X}_B||\hat{Y}_A| = |1||1|\cos(90-\theta) = \sin\theta$$

$$|\hat{Y}_B||\hat{X}_A| = |1||1|\cos(90+\theta) = -\sin\theta$$

$$|\hat{X}_B||\hat{X}_A| = |1||1|\cos\theta = \cos\theta$$

$$|\hat{X}_B||\hat{Z}_A| = |1||1|\cos(90) = 0$$

$$|\hat{Y}_B||\hat{Z}_A| = |1||1|\cos(90) = 0$$

$$|\hat{Z}_B||\hat{Z}_A| = |1||1|\cos(0) = 1$$

$$|\hat{Z}_B||\hat{X}_A| = |\hat{Z}_B||\hat{Y}_A| = |1||1|\cos(90) = 0$$

Rewrite axis of {B} representation expressed in {A}

$$\hat{X}_B = \cos\theta \hat{X}_A + \sin\theta \hat{X}_B$$

$$\hat{Y}_B = \sin(-\theta)\, \hat{X}_A + \cos\theta\, \hat{X}_B$$

$$\hat{Z}_B = \hat{Z}_A$$

เรามักจะเขียนเวคเตอร์ในรูปแบบเมตริกซ์ดังนั้น

$$\hat{X}_B = \begin{bmatrix} \cos\theta \\ \sin\theta \\ 0 \end{bmatrix} \qquad \hat{Y}_B = \begin{bmatrix} -\sin\theta \\ \cos\theta \\ 0 \end{bmatrix} \qquad \hat{Z}_B = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

ดังนั้น

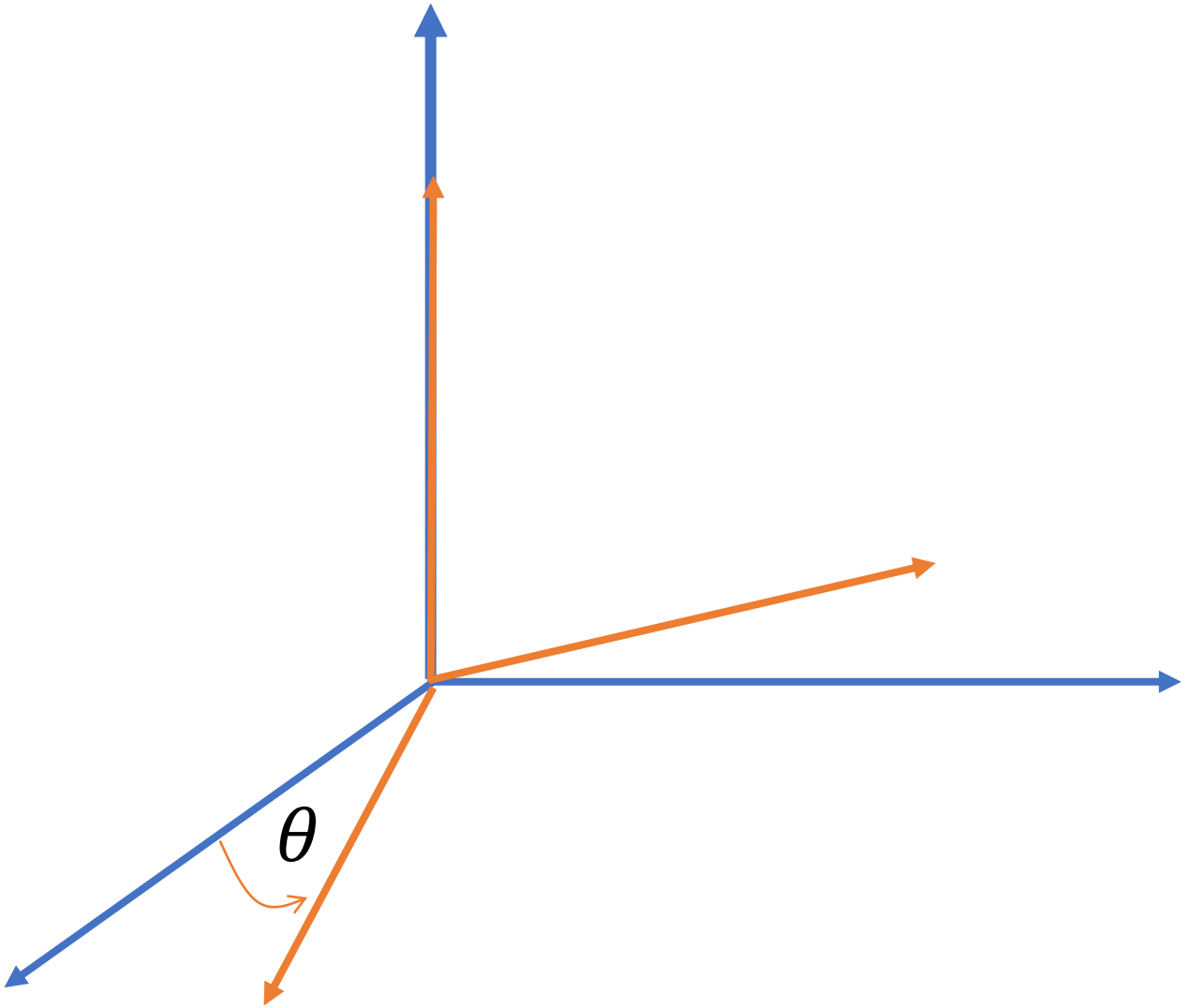
$${}^A_B R = \begin{bmatrix} {}^A\hat{X}_B & {}^A\hat{Y}_B & {}^A\hat{Z}_B \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Properties

$$ {}^{B}_{A}R \quad = {}^{A}_{B}R^{T} = {}^{B}_{A}R^{-1} $$

As can be shown!!

Example 2.1 (จากหนังสือ)



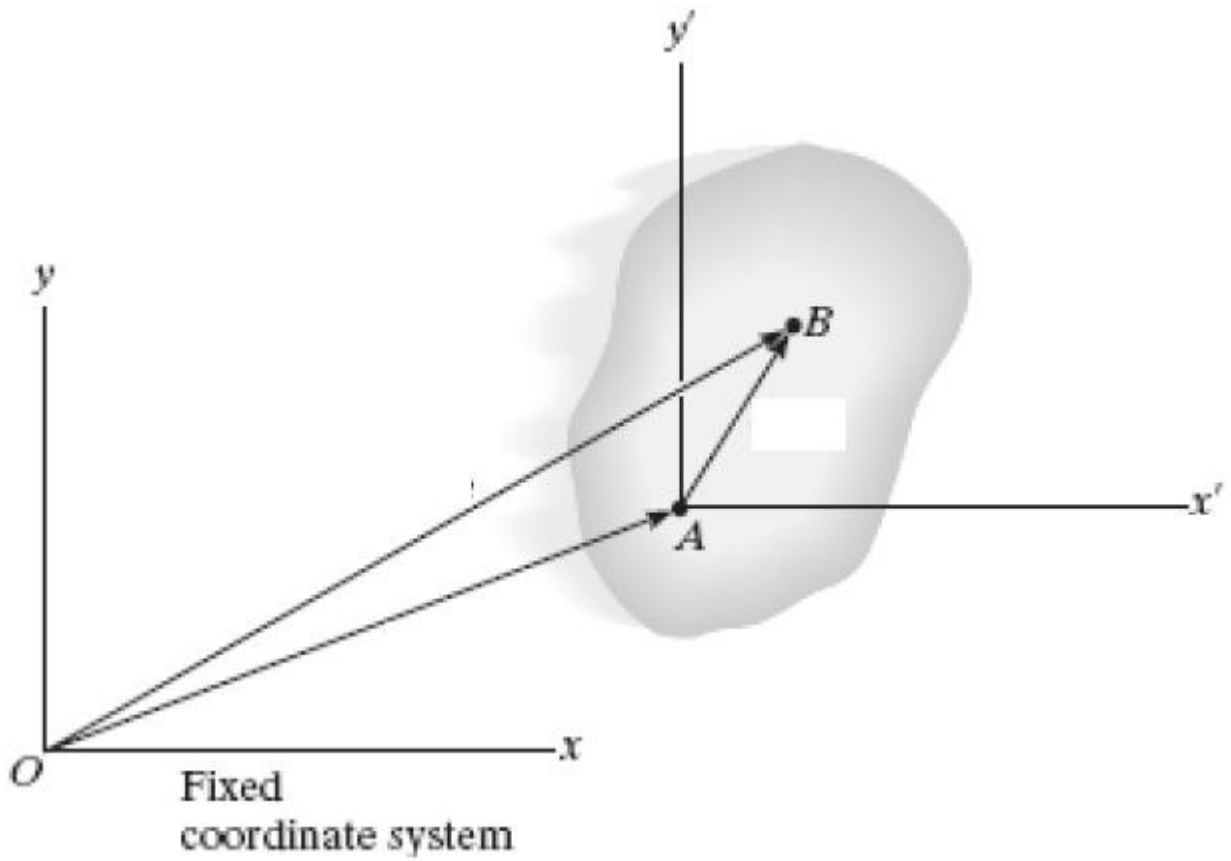
Example 2.1 (จากหนังสือ)

Rotational Matrix ,$R_z(\theta) = 30^\circ$

Basic Rotational Matrix

Description of FRAME

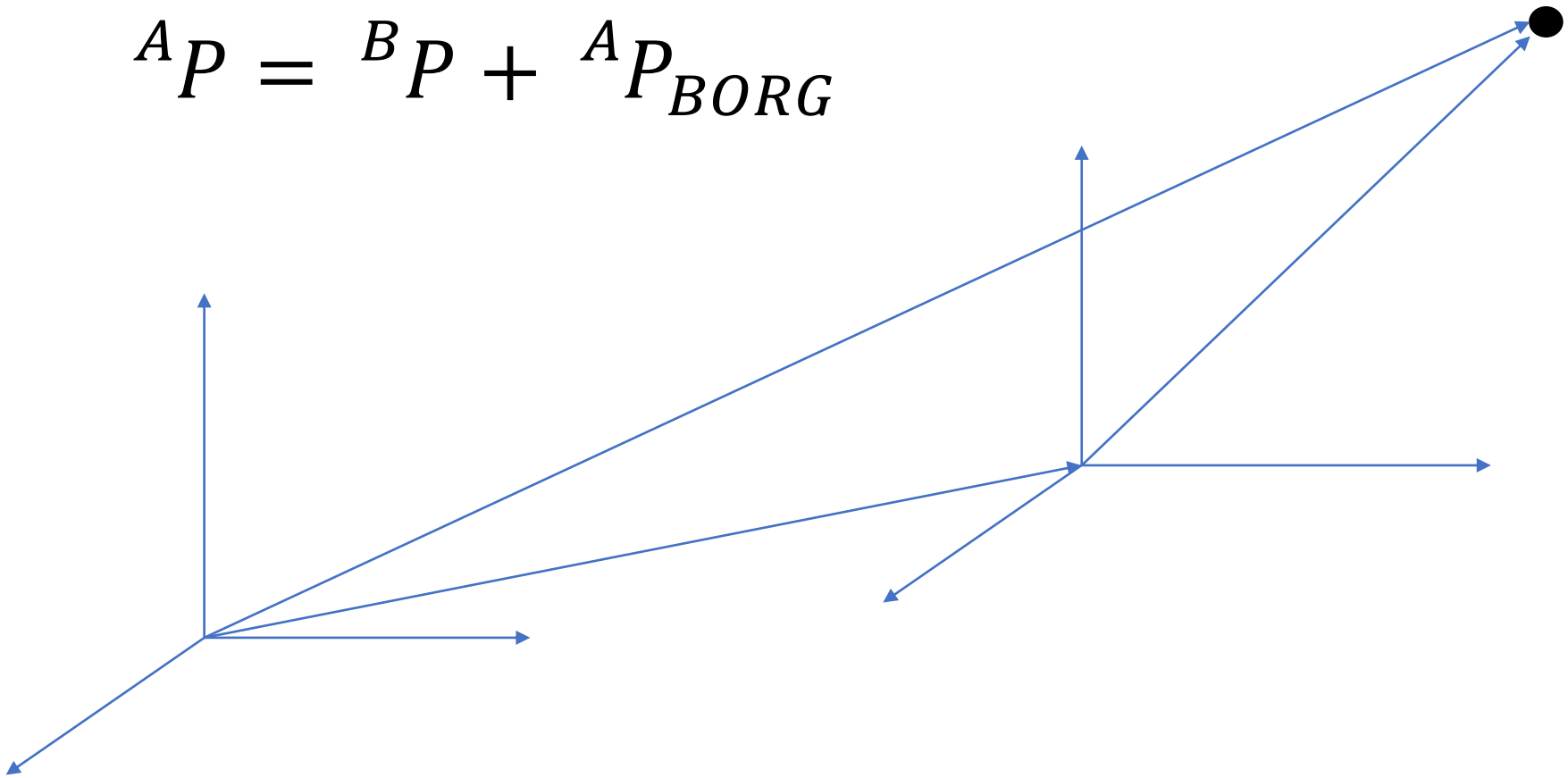
$$\{B\}=\left\{{}^{A}_{B}R + P_{BORG}\right\}$$



Mapping:
Changing description from frame to frame

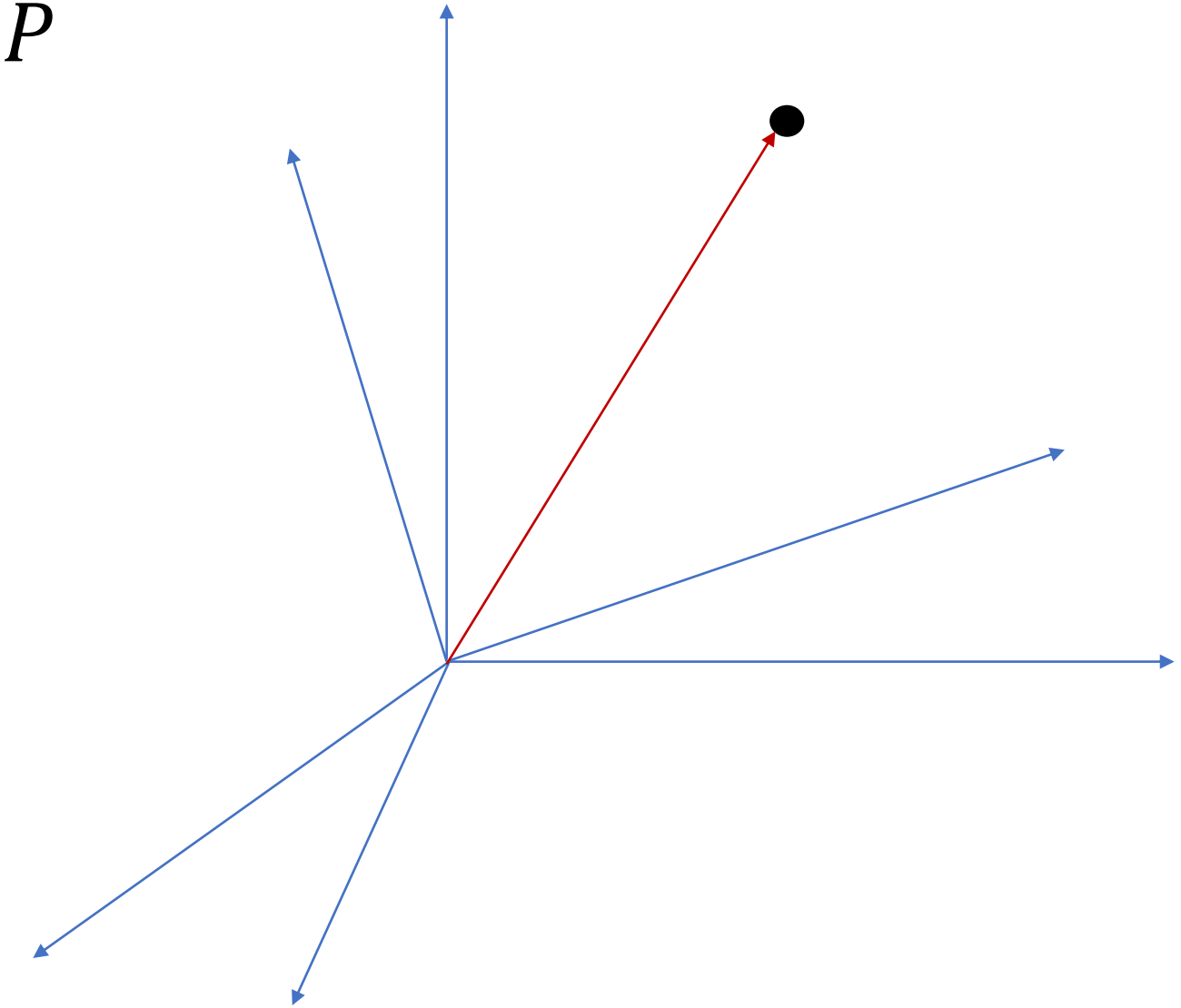
- Mapping involving translated frames

$$^{A}P = {}^{B}P + {}^{A}P_{BORG}$$



- Mapping involving rotated frames

$$ {}^{A}P = {}^{A}_{B}R \, {}^{B}P $$



Example 2.2 (from textbook)

Example 2.2 (from textbook

- Mapping involving general frames

$$ {}^{A}P = {}^{A}_{B}R\,{}^{B}P + {}^{A}P_{BORG} $$

$$ \begin{bmatrix} {}^{A}P \\ 1 \end{bmatrix} = \left[\begin{array}{ccc|c} & {}^{B}_{A}R & & {}^{A}P_{BORG} \\ \hline 0 & 0 & 0 & 1 \end{array}\right] \begin{bmatrix} {}^{B}P \\ 1 \end{bmatrix} $$

4x4

Homogeneous Transformation Matrix

Example 2.3 (textbook)

Figure 2.10 shows a vector ${}^{A}P_1$. We wish to compute the vector obtained by rotating this vector about $\hat{Z}$ by 30 degrees. Call the new vector ${}^{A}P_2$.

The rotation matrix that rotates vectors by 30 degrees about $\hat{Z}$ is the same as the rotation matrix that describes a frame rotated 30 degrees about $\hat{Z}$ relative to the reference frame. Thus, the correct rotational operator is

$$R_z(30.0) = \begin{bmatrix} 0.866 & -0.500 & 0.000 \\ 0.500 & 0.866 & 0.000 \\ 0.000 & 0.000 & 1.000 \end{bmatrix}. \tag{2.30}$$

Given

$${}^{A}P_1 = \begin{bmatrix} 0.0 \\ 2.0 \\ 0.0 \end{bmatrix}, \tag{2.31}$$

we calculate ${}^{A}P_2$ as

$${}^{A}P_2 = R_z(30.0)\,{}^{A}P_1 = \begin{bmatrix} -1.000 \\ 1.732 \\ 0.000 \end{bmatrix}. \tag{2.32}$$

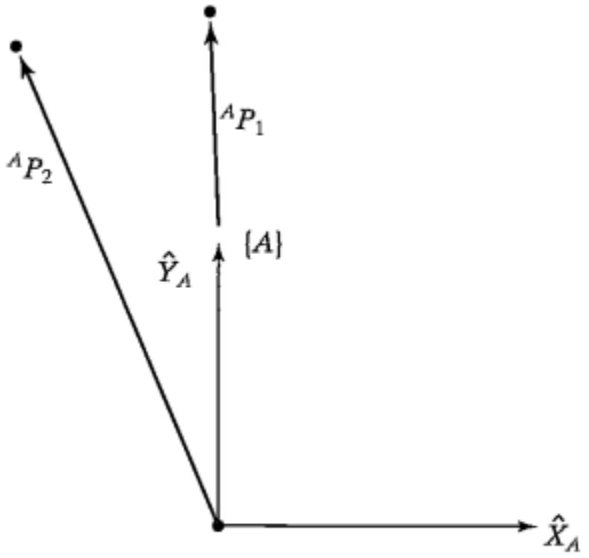


FIGURE 2.10: The vector ${}^{A}P_1$ rotated 30 degrees about $\hat{Z}$.

Example 2.4 (textbook)

Figure 2.11 shows a vector ${}^{A}P_1$. We wish to rotate it about $\hat{Z}$ by 30 degrees and translate it 10 units in $\hat{X}_A$ and 5 units in $\hat{Y}_A$. Find ${}^{A}P_2$, where ${}^{A}P_1 = [3.0\ 7.0\ 0.0]^T$.

The operator T, which performs the translation and rotation, is

$$T = \begin{bmatrix} 0.866 & -0.500 & 0.000 & 10.0 \\ 0.500 & 0.866 & 0.000 & 5.0 \\ 0.000 & 0.000 & 1.000 & 0.0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

Given

$${}^{A}P_1 = \begin{bmatrix} 3.0 \\ 7.0 \\ 0.0 \end{bmatrix}, \tag{2.35}$$

we use T as an operator:

$${}^{A}P_2 = T\ {}^{A}P_1 = \begin{bmatrix} 9.098 \\ 12.562 \\ 0.000 \end{bmatrix}. \tag{2.36}$$

Note that this example is numerically exactly the same as Example 2.2, but the interpretation is quite different.

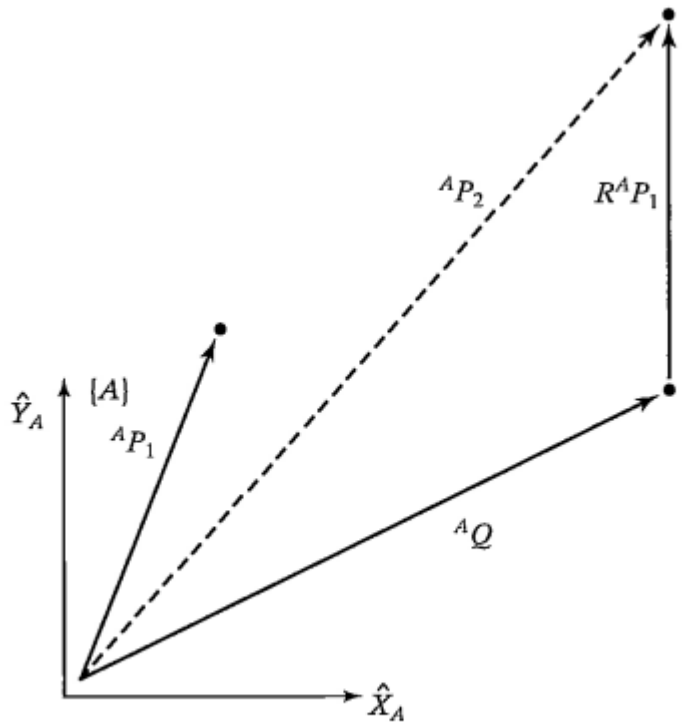


FIGURE 2.11: The vector ${}^{A}P_1$ rotated and translated to form ${}^{A}P_2$.

Summary of Interpretations

As a general tool to represent frames, we have introduced the *homogeneous transform*, a 4×4 matrix containing orientation and position information.

We have introduced three interpretations of this homogeneous transform:

1. It is a *description of a frame*. ${}^{A}_{B}T$ describes the frame $\{B\}$ relative to the frame $\{A\}$. Specifically, the columns of ${}^{A}_{B}R$ are unit vectors defining the directions of the principal axes of $\{B\}$, and ${}^{A}P_{BORG}$ locates the position of the origin of $\{B\}$.
2. It is a *transform mapping*. ${}^{A}_{B}T$ maps ${}^{B}P \rightarrow {}^{A}P$.
3. It is a *transform operator*. T operates on ${}^{A}P_1$ to create ${}^{A}P_2$.

Transformaiton Arithmatic

In this section, we look at the multiplication of transforms and the inversion of transforms. These two elementary operations form a functionally complete set of transform operators.

Compound transformations

In Fig. 2.12, we have ${}^{C}P$ and wish to find ${}^{A}P$.

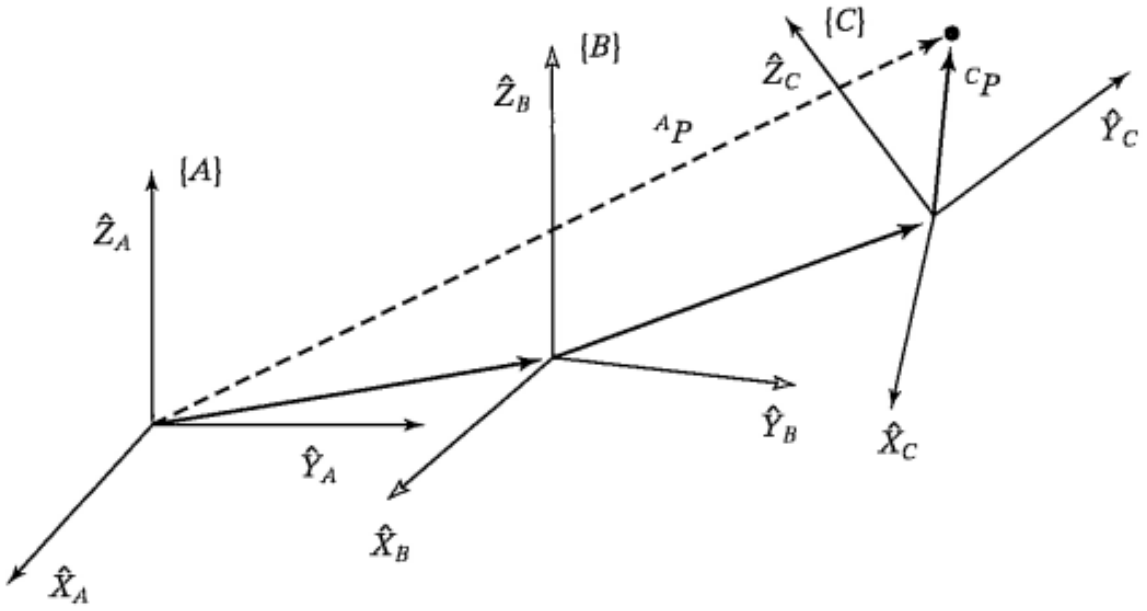


FIGURE 2.12: Compound frames: Each is known relative to the previous one.

Frame $\{C\}$ is known relative to frame $\{B\}$, and frame $\{B\}$ is known relative to frame $\{A\}$. We can transform ${}^{C}P$ into ${}^{B}P$ as

$$ {}^{B}P = {}^{B}_{C}T \, {}^{C}P; \tag{2.37} $$

then we can transform ${}^{B}P$ into ${}^{A}P$ as

$$ {}^{A}P = {}^{A}_{B}T \, {}^{B}P. \tag{2.38} $$

Combining (2.37) and (2.38), we get the (not unexpected) result

$$ {}^{A}P = {}^{A}_{B}T {}^{B}_{C}T {}^{C}P, \tag{2.39} $$

from which we could define

$$ {}^{A}_{C}T = {}^{A}_{B}T {}^{B}_{C}T. \tag{2.40} $$

Again, note that familiarity with the sub- and superscript notation makes these manipulations simple. In terms of the known descriptions of $\{B\}$ and $\{C\}$, we can give the expression for ${}^{A}_{C}T$ as

$$ {}^{A}_{C}T = \left[\begin{array}{ccc|c} & {}^{A}_{B}R \, {}^{B}_{C}R & & {}^{A}_{B}R \, {}^{B}P_{CORG} + {}^{A}P_{BORG} \\ \hline 0 & 0 & 0 & 1 \end{array}\right] \tag{2.41} $$

Inverting a transform

Consider a frame $\{B\}$ that is known with respect to a frame $\{A\}$—that is, we know the value of ${}^{A}_{B}T$. Sometimes we will wish to invert this transform, in order to get a description of $\{A\}$ relative to $\{B\}$—that is, ${}^{B}_{A}T$. A straightforward way of calculating the inverse is to compute the inverse of the 4×4 homogeneous transform. However, if we do so, we are not taking full advantage of the structure inherent in the transform. It is easy to find a computationally simpler method of computing the inverse, one that does take advantage of this structure.

To find ${}^B_A T$, we must compute ${}^B_A R$ and ${}^B P_{AORG}$ from ${}^A_B R$ and ${}^A P_{BORG}$. First, recall from our discussion of rotation matrices that

$$ {}^B_A R = {}^A_B R^T. \tag{2.42} $$

Next, we change the description of ${}^A P_{BORG}$ into $\{B\}$ by using (2.13):

$$ {}^B({}^A P_{BORG}) = {}^B_A R \, {}^A P_{BORG} + {}^B P_{AORG}. \tag{2.43} $$

The left-hand side of (2.43) must be zero, so we have

$$ {}^B P_{AORG} = -{}^B_A R \, {}^A P_{BORG} = -{}^A_B R^T \, {}^A P_{BORG}. \tag{2.44} $$

Using (2.42) and (2.44), we can write the form of ${}^B_A T$ as

$$ {}^B_A T = \left[\begin{array}{ccc|c} & {}^A_B R^T & & -{}^A_B R^T \, {}^A P_{BORG} \\ \hline 0 & 0 & 0 & 1 \end{array}\right]. \tag{2.45} $$

Note that, with our notation,

$$ {}^{B}_{A}T = {}^{A}_{B}T^{-1}. $$

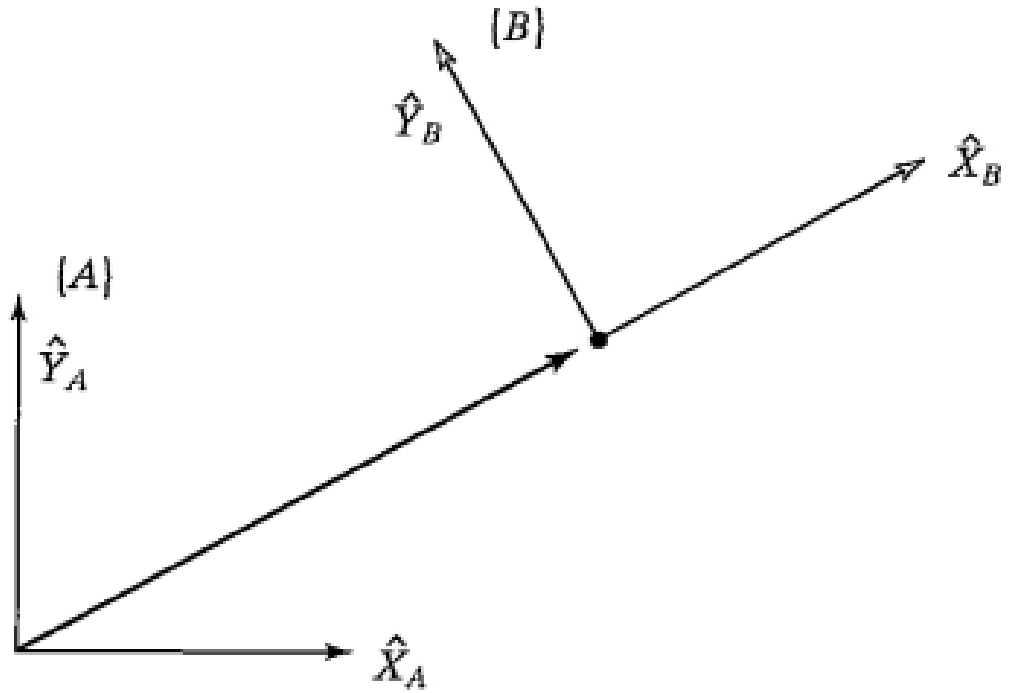
Equation (2.45) is a general and extremely useful way of computing the inverse of a homogeneous transform.

EXAMPLE 2.5

Figure 2.13 shows a frame $\{B\}$ that is rotated relative to frame $\{A\}$ about $\hat{Z}$ by 30 degrees and translated four units in $\hat{X}_A$ and three units in $\hat{Y}_A$. Thus, we have a description of ${}^{A}_{B}T$. Find ${}^{B}_{A}T$.

The frame defining $\{B\}$ is

$$ {}^{A}_{B}T = \begin{bmatrix} 0.866 & -0.500 & 0.000 & 4.0 \\ 0.500 & 0.866 & 0.000 & 3.0 \\ 0.000 & 0.000 & 1.000 & 0.0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{2.46} $$



Using (2.45), we compute

$$ {}^{B}_{A}T = \begin{bmatrix} 0.866 & 0.500 & 0.000 & -4.964 \\ -0.500 & 0.866 & 0.000 & -0.598 \\ 0.000 & 0.000 & 1.000 & 0.0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{2.47} $$

TRANSFORM EQUATIONS

Figure 2.14 indicates a situation in which a frame $\{D\}$ can be expressed as products of transformations in two different ways. First,

$$ {}^{U}_{D}T = {}^{U}_{A}T\ {}^{A}_{D}T; \tag{2.48} $$

second;

$$ {}^{U}_{D}T = {}^{U}_{B}T\ {}^{B}_{C}T\ {}^{C}_{D}T. \tag{2.49} $$

We can set these two descriptions of ${}^{U}_{D}T$ equal to construct a **transform equation**:

$$ {}^{U}_{A}T\ {}^{A}_{D}T = {}^{U}_{B}T\ {}^{B}_{C}T\ {}^{C}_{D}T. \tag{2.50} $$

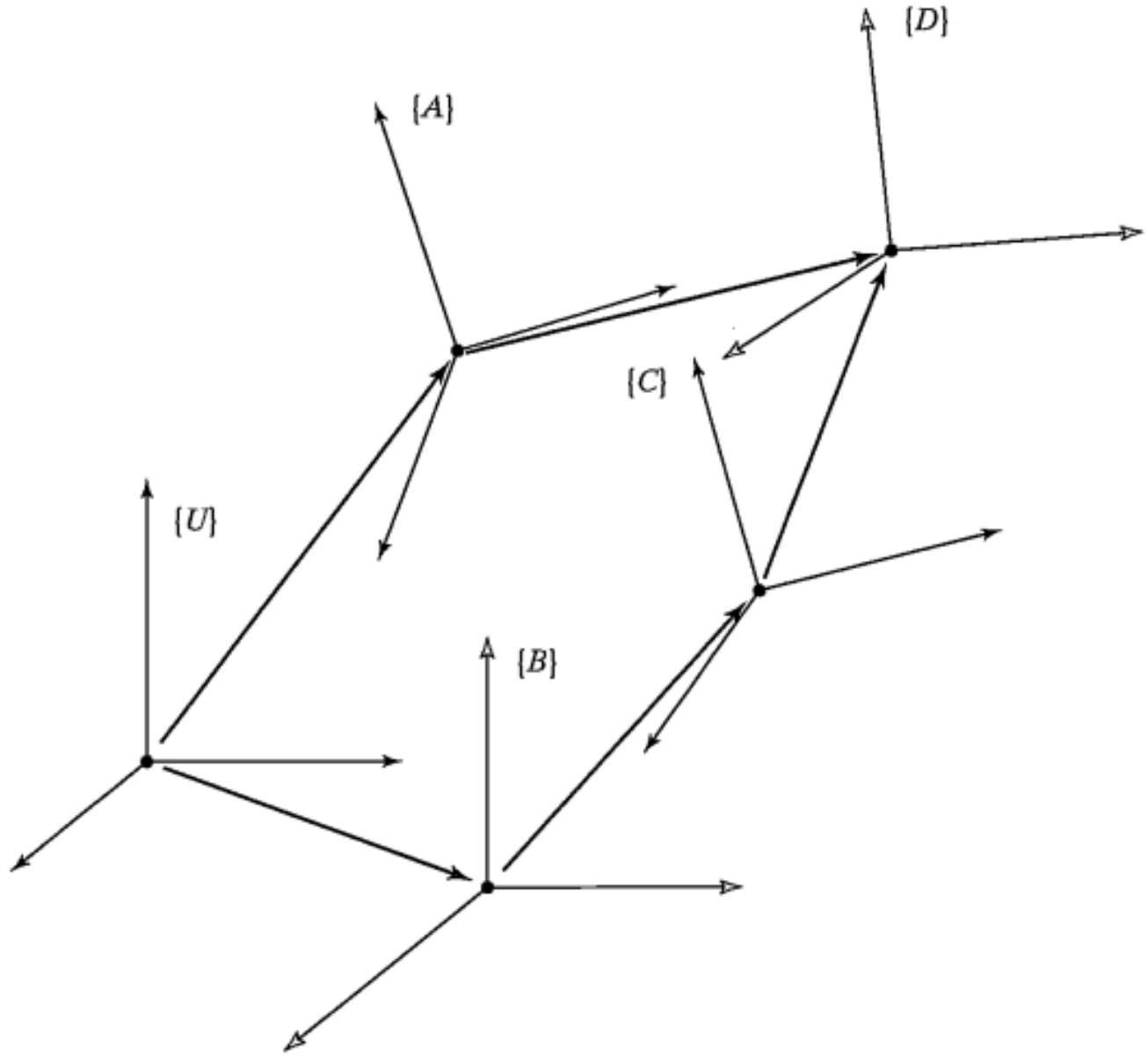


FIGURE 2.14: Set of transforms forming a loop.

EXAMPLE 2.6

Assume that we know the transform ${}^{B}_{T}T$ in Fig. 2.16, which describes the frame at the manipulator's fingertips $\{T\}$ relative to the base of the manipulator, $\{B\}$, that we know where the tabletop is located in space relative to the manipulator's base (because we have a description of the frame $\{S\}$ that is attached to the table as shown, ${}^{B}_{S}T$), and that we know the location of the frame attached to the bolt lying on the table relative to the table frame—that is, ${}^{S}_{G}T$. Calculate the position and orientation of the bolt relative to the manipulator's hand, ${}^{T}_{G}T$.

Guided by our notation (and, it is hoped, our understanding), we compute bolt frame relative to the hand frame as

$$ {}^{T}_{G}T = {}^{B}_{T}T^{-1}\, {}^{B}_{S}T\, {}^{S}_{G}T. \qquad (2. $$

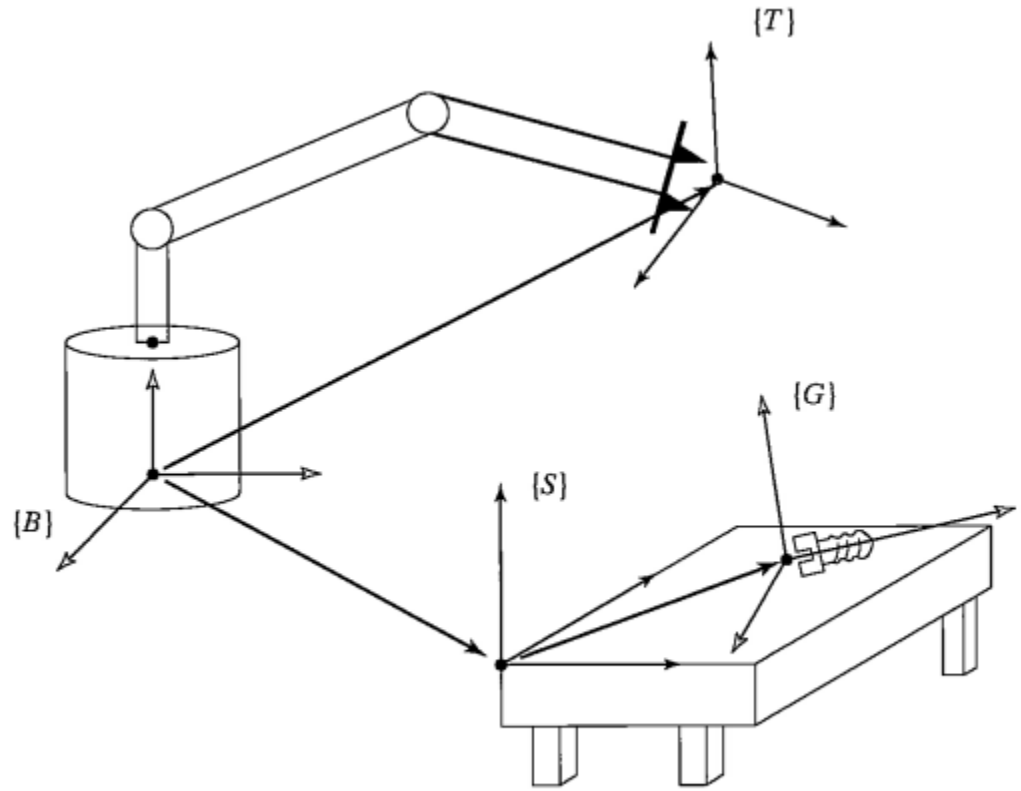
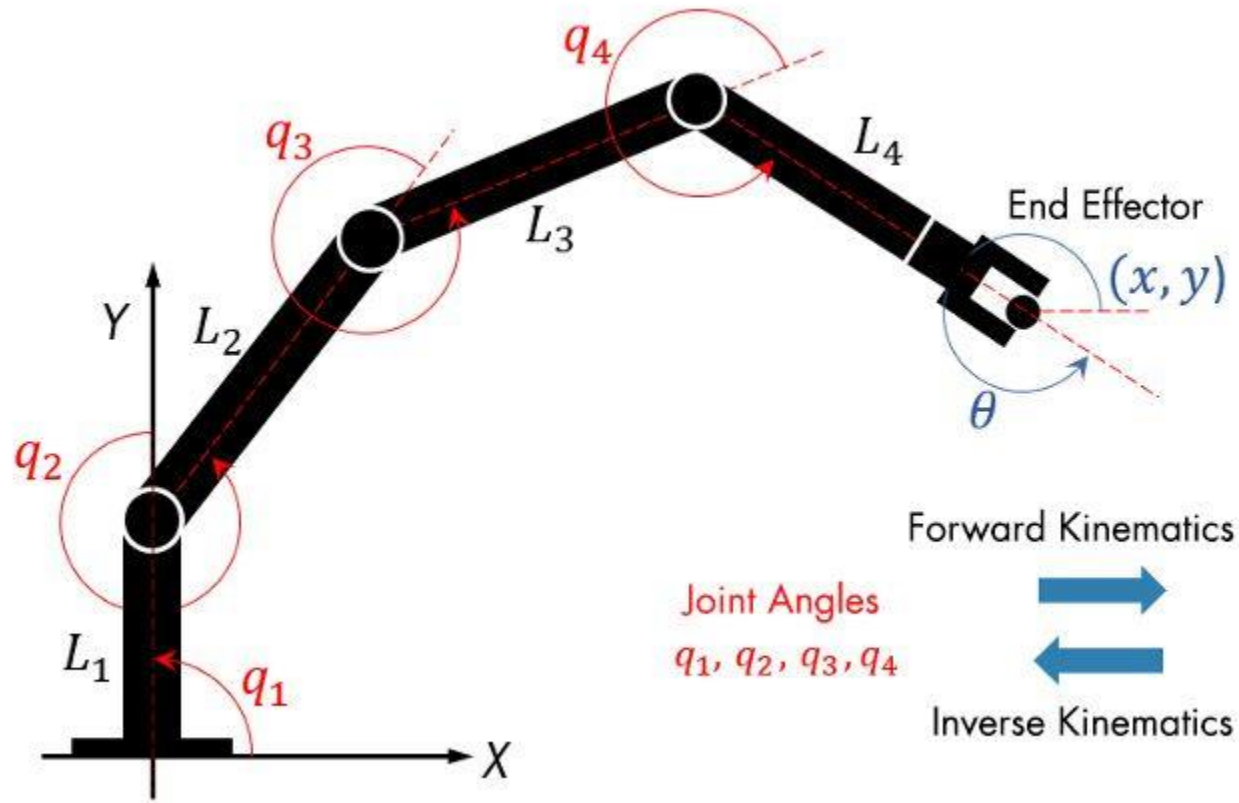


FIGURE 2.16: Manipulator reaching for a bolt.

Conclusion
Spatial Description

- Task Description
- Transformations
- Representations



Forward Kinematics (FK)

Joint Angles
q_1, q_2, q_3, q_4

End Effector Pose
x, y, θ

Inverse Kinematics (IK)