# ECE447: Robotics Engineering

## Lecture 3: Rigid Motions and Homogeneous Transformations (Part 1)

**Dr. Haitham El-Hussieny**

Electronics and Communications Engineering
**Faculty of Engineering (Shoubra)**
**Benha University**

Spring 2019

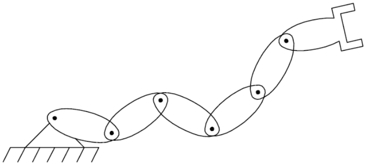
# Lecture Outline:

1. Introduction.
2. Representation of Translation.
3. Representation of Rotations.
4. Representation of Rotations in 3D.

# Table of Contents

# Introduction:

- A robot manipulator is schematically represented as a kinematic chain.
- It is composed by a series of rigid bodies, the **links**, connected by **joints**.
- The resulting end-effector motion is obtained by composition of the **elementary motions** of each link with respect to the previous one.
- To manipulate an object in space, it is necessary to describe the end-effector **position and orientation**.



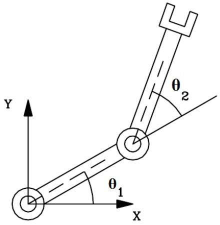
# Introduction:

- **Kinematics** is the study of **how the robot moves** not why it moves (Dynamics).
- In robotic manipulation we are concerned with two common **kinematic** problems:

# Introduction:

- **Kinematics** is the study of **how the robot moves** not why it moves (Dynamics).
- In robotic manipulation we are concerned with two common **kinematic** problems:

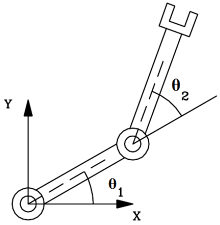
**Forward Kinematics**



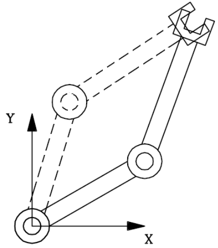
# Introduction:

- **Kinematics** is the study of **how the robot moves** not why it moves (Dynamics).
- In robotic manipulation we are concerned with two common **kinematic** problems:

**Forward Kinematics**

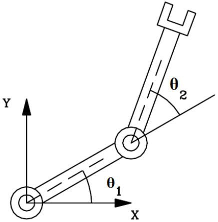



**Inverse Kinematics**



# Introduction:

## Forward Kinematics



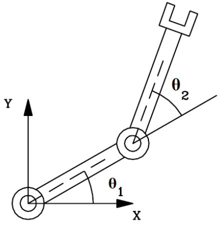

**Given**: Joint Variables **q** ($\theta$ or $d$)
**Required**: Position and orientation of end-effector, **p**.

$$\mathbf{p} = f(q_1, q_2, \ldots, q_n) = f(\mathbf{q})$$

EASY!

# Introduction:

## Forward Kinematics

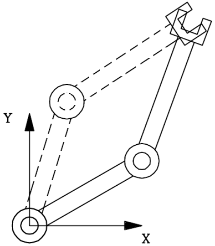



**Given**: Joint Variables **q** ($\theta$ or $d$)
**Required**: Position and orientation of end-effector, **p**.

$$\mathbf{p} = f(q_1, q_2, \ldots, q_n) = f(\mathbf{q})$$

EASY!

## Inverse Kinematics



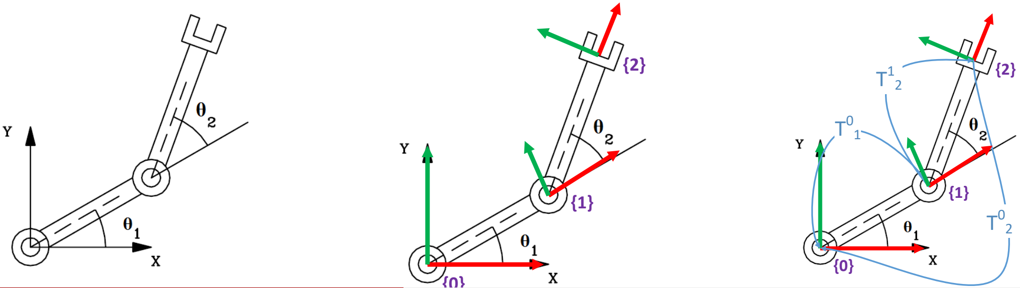

**Given**: Position and orientation of end-effector, **p**.
**Required**: Joint Variables **q** ($\theta$ or $d$) to get **p**

$$\mathbf{q} = f(\mathbf{p})$$

DIFFICULT (May be infinite solutions exist)!

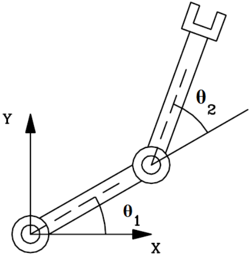
# Introduction:

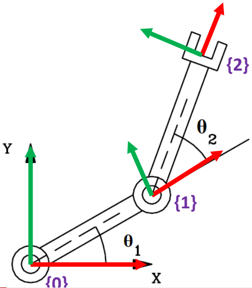
- Solving the manipulator kinematics requires the assignment of different coordinate frames on each joint and the end-effector.

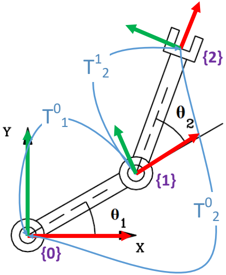


# Introduction:

- Solving the manipulator kinematics requires the assignment of different coordinate frames on each joint and the end-effector.
- The goal is to find the transformation between the end-effector and the base frames.



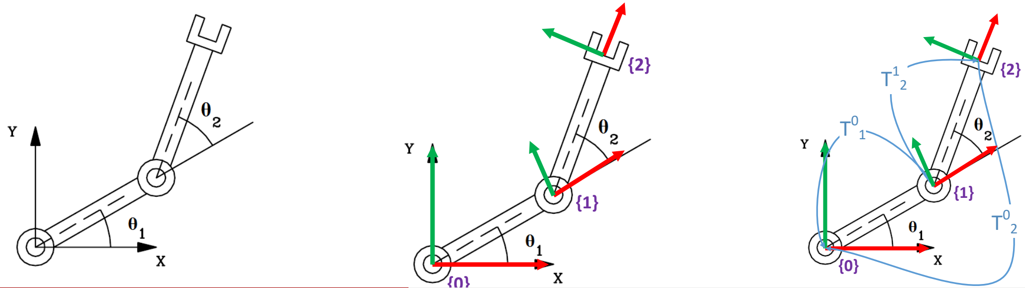



# Introduction:

- Solving the manipulator kinematics requires the assignment of different coordinate frames on each joint and the end-effector.
- The goal is to find the transformation between the end-effector and the base frames.
- In robotic manipulators, two basic transformations are used: **Translation** and **Rotation**.

$$T_2^0 = T_1^0 * T_2^1$$

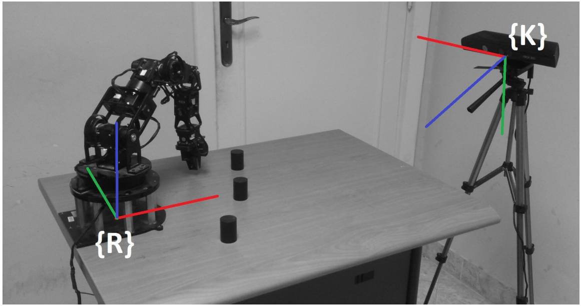


# Table of Contents

# Representation of Translation:

**Coordinate Frame:**

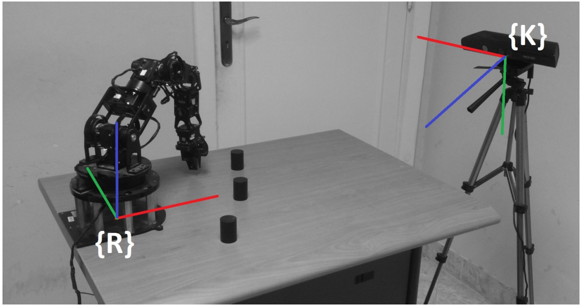
- The **position** and **orientation** of an object in space is referred to as its **pose**.



# Representation of Translation:

**Coordinate Frame:**

- The **position** and **orientation** of an object in space is referred to as its **pose**.
- Any description of an object's pose must always be made in relation to some **coordinate frames**.



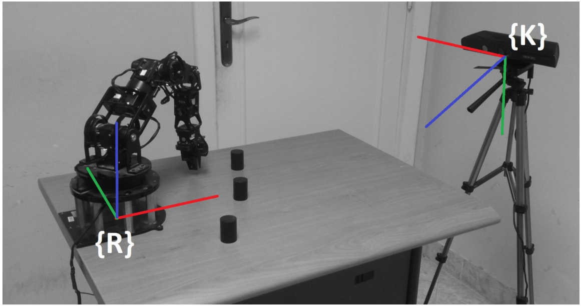

Coordinate Frame:

A set $n$ of orthonormal basis vectors spanning $\mathbb{R}^n$.

# Representation of Translation:

**Coordinate Frame:**

- The **position** and **orientation** of an object in space is referred to as its **pose**.
- Any description of an object's pose must always be made in relation to some **coordinate frames**.
- In robotics, it is often convenient to keep track of multiple coordinate frames. (Camera, robot, user, world, . . . etc.)



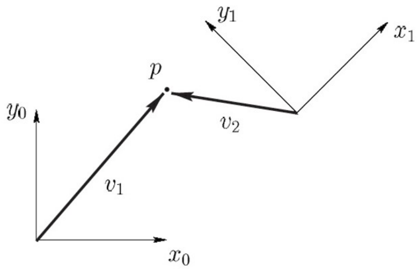

Coordinate Frame:

A set $n$ of orthonormal basis vectors spanning $\mathbb{R}^n$.

# Representation of Translation:

**Representation of a Point:**

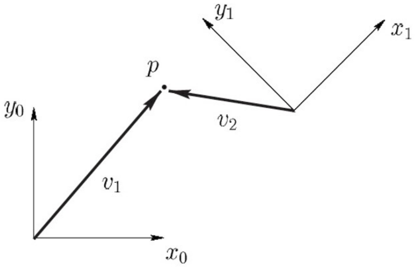
- A point corresponds a particular location in the space. For example the point $p$.



# Representation of Translation:

**Representation of a Point:**

- A point corresponds a particular location in the space. For example the point $p$.
- A point has different representation (coordinates) in different frames.



# Representation of Translation:

## Representation of a Point:

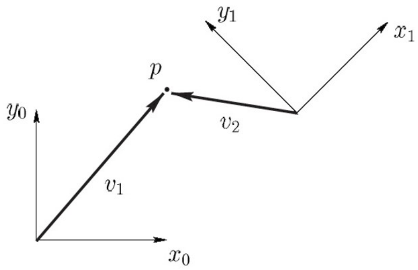
- A point corresponds a particular location in the space. For example the point $p$.
- A point has different representation (coordinates) in different frames.

Example (Point $p$):

$$\textbf{w.r.t. frame } \{0\}\textbf{:}\ p^0 = v_1 = \begin{bmatrix} 6 \\ 5 \end{bmatrix}$$

$$\textbf{w.r.t. frame } \{1\}\textbf{:}\ p^1 = v_2 = \begin{bmatrix} -2.8 \\ 4.2 \end{bmatrix}$$

Note: the frame of reference is written in right superscript style. $p^k$, $o_1^0$



# Representation of Translation:

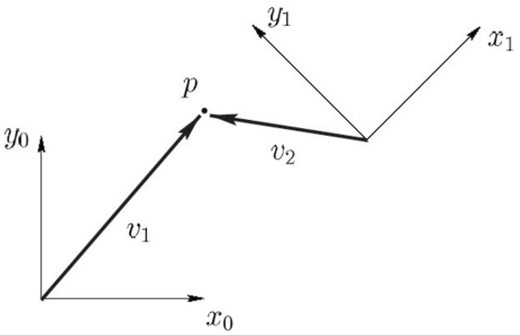
**Representation of a Point:**

- Since the origin of a frame is just a point, we can express the origin of one frame with respect to another.

Example:

$$o_1^0 = \begin{bmatrix} 10 \\ 5 \end{bmatrix} \quad o_1 \text{ represented in frame } \{0\}$$

$$o_0^1 = \begin{bmatrix} -10.6 \\ 3.5 \end{bmatrix} \quad o_0 \text{ represented in frame } \{1\}$$



# Representation of Translation:

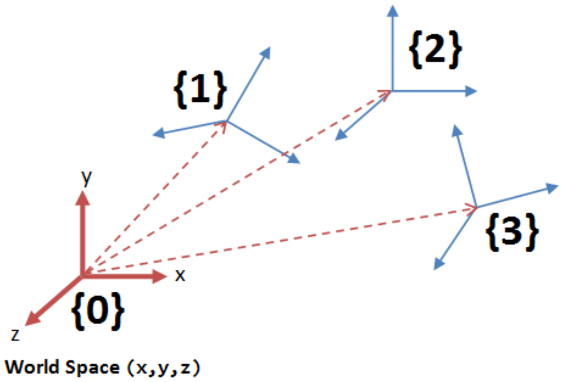
## Representation of a Point:

- The point translation could be also represented in 3-dimensional space.

Example:

$$o_1^0 = \begin{bmatrix} x_1^0 \\ y_1^0 \\ z_1^0 \end{bmatrix} \quad o_1 \text{ represented in frame } \{0\}$$

$$o_0^1 = \begin{bmatrix} x_0^1 \\ y_0^1 \\ z_0^1 \end{bmatrix} \quad o_0 \text{ represented in frame } \{1\}$$

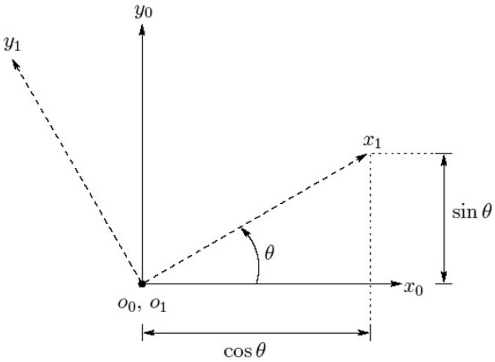


# Table of Contents

# Representation of Rotations:

## Rotation in 2D:

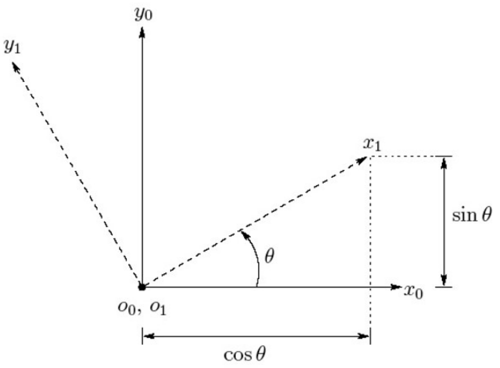
- Frame $\{1\}$ is obtained by rotating frame $\{0\}$ by an angle $\theta$.



# Representation of Rotations:

## Rotation in 2D:

- Frame $\{1\}$ is obtained by rotating frame $\{0\}$ by an angle $\theta$.
- We need to find the relative orientation between these two frames.



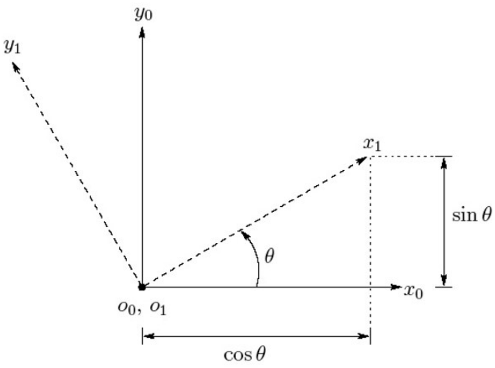
# Representation of Rotations:

## Rotation in 2D:

- Frame $\{1\}$ is obtained by rotating frame $\{0\}$ by an angle $\theta$.
- We need to find the relative orientation between these two frames.
- We can specify the orientation by finding the representation of coordinate vectors of frame $\{1\}$ w.r.t frame $\{0\}$.

$$R_1^0 = \left[ \; x_1^0 | \; y_1^0 \; \right]$$

$x_1^0$ and $y_1^0$ are the unit vector $x_1$ and $y_1$ represented in frame $\{0\}$.

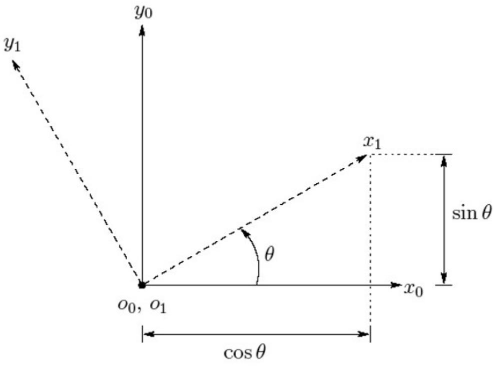


# Representation of Rotations:

## Rotation in 2D:

$$R_1^0 = \left[\; x_1^0 |\; y_1^0 \;\right] = \begin{bmatrix} \hat{x}_1 . \hat{x}_0 & \hat{y}_1 . \hat{x}_0 \\ \hat{x}_1 . \hat{y}_0 & \hat{y}_1 . \hat{y}_0 \end{bmatrix}$$

Projecting the axes of frame $\{1\}$ onto the axes of frame $\{0\}$:



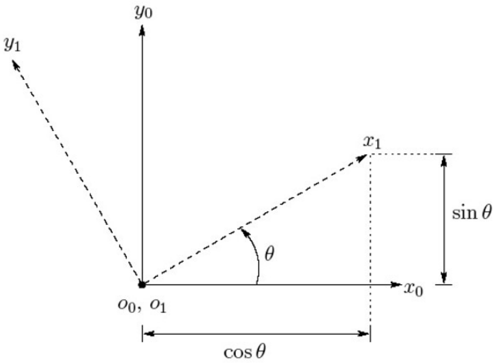
# Representation of Rotations:

## Rotation in 2D:

$$R_1^0 = \left[\ x_1^0 |\ y_1^0\ \right] = \begin{bmatrix} \hat{x}_1.\hat{x}_0 & \hat{y}_1.\hat{x}_0 \\ \hat{x}_1.\hat{y}_0 & \hat{y}_1.\hat{y}_0 \end{bmatrix}$$

Projecting the axes of frame $\{1\}$ onto the axes of frame $\{0\}$:

$$x_1^0 = \begin{bmatrix} \cos(\theta) \\ \sin(\theta) \end{bmatrix} \qquad y_1^0 = \begin{bmatrix} -\sin(\theta) \\ \cos(\theta) \end{bmatrix}$$



# Representation of Rotations:

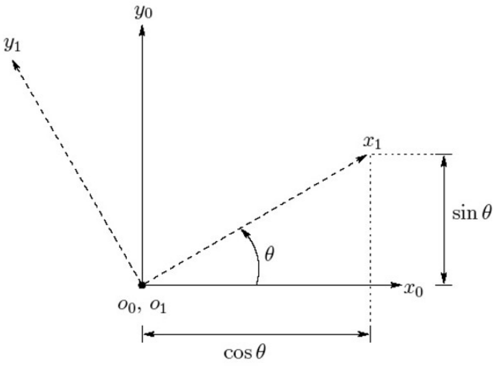
## Rotation in 2D:

$$R_1^0 = \left[\ x_1^0 |\ y_1^0\ \right] = \begin{bmatrix} \hat{x}_1.\hat{x}_0 & \hat{y}_1.\hat{x}_0 \\ \hat{x}_1.\hat{y}_0 & \hat{y}_1.\hat{y}_0 \end{bmatrix}$$

Projecting the axes of frame $\{1\}$ onto the axes of frame $\{0\}$:

$$x_1^0 = \left[ \begin{array}{c} \cos(\theta) \\ \sin(\theta) \end{array} \right] \qquad y_1^0 = \left[ \begin{array}{c} -\sin(\theta) \\ \cos(\theta) \end{array} \right]$$

$$R_1^0 = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix}$$ **Rotation Matrix**

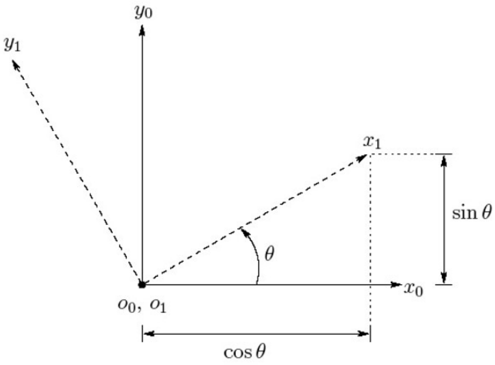


# Representation of Rotations:

## Rotation in 2D:

**Properties of Rotation Matrices:**

1. $\det(R_1^0) = \cos^2(\theta) + \sin^2(\theta) = 1.$




$$R_1^0 = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix}$$

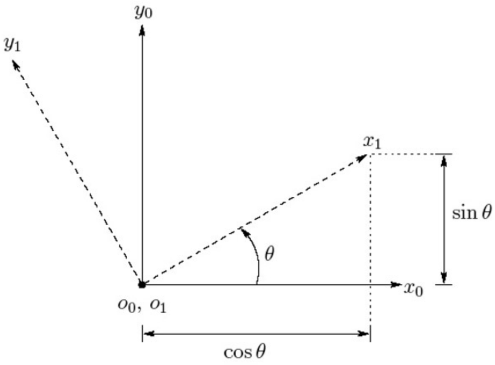
# Representation of Rotations:

## Rotation in 2D:

**Properties of Rotation Matrices:**

1. $\det(R_1^0) = \cos^2(\theta) + \sin^2(\theta) = 1.$
2. Inverse Rotation:

$$R_0^1 = \left[\begin{array}{c|c} x_0^1 & y_0^1 \end{array}\right] = \begin{bmatrix} \hat{x}_0 . \hat{x}_1 & \hat{y}_0 . \hat{x}_1 \\ \hat{x}_0 . \hat{y}_1 & \hat{y}_0 . \hat{y}_1 \end{bmatrix}$$




$$R_1^0 = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix}$$

# Representation of Rotations:

## Rotation in 2D:

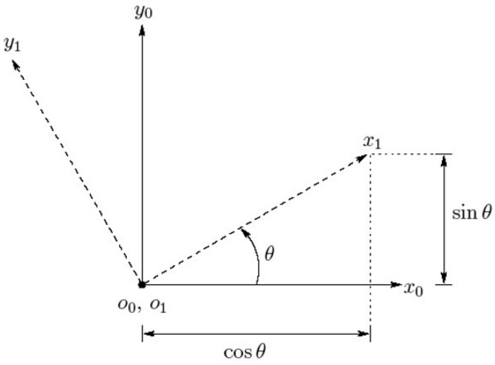
**Properties of Rotation Matrices:**

1. $\det(R_1^0) = \cos^2(\theta) + \sin^2(\theta) = 1.$
2. Inverse Rotation:

$$R_0^1 = [\ x_0^1 |\ y_0^1\ ] = \begin{bmatrix} \hat{x}_0.\hat{x}_1 & \hat{y}_0.\hat{x}_1 \\ \hat{x}_0.\hat{y}_1 & \hat{y}_0.\hat{y}_1 \end{bmatrix}$$

$$R_0^1 = \begin{bmatrix} \cos(\theta) & \sin(\theta) \\ -\sin(\theta) & \cos(\theta) \end{bmatrix} = (R_1^0)^T$$

Projecting the axes of frame $\{0\}$ onto the axes of frame $\{1\}$.




$$R_1^0 = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix}$$

# Representation of Rotations:

## Rotation in 2D:

**Properties of Rotation Matrices:**

3. Inverse of Rotation Matrix:

$$(R_1^0)^{-1} = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix}^{-1} = (R_1^0)^T = (R_0^1)$$

# Representation of Rotations:

## Rotation in 2D:

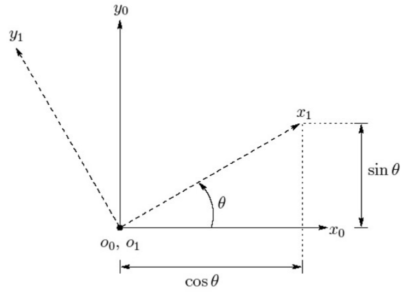
**Properties of Rotation Matrices:**

3. Inverse of Rotation Matrix:

$$(R_1^0)^{-1} = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix}^{-1} = (R_1^0)^T = (R_0^1)$$

4. $R$ is Special Orthogonal matrix $SO(n)$:

$$R\ R^T = I_n$$



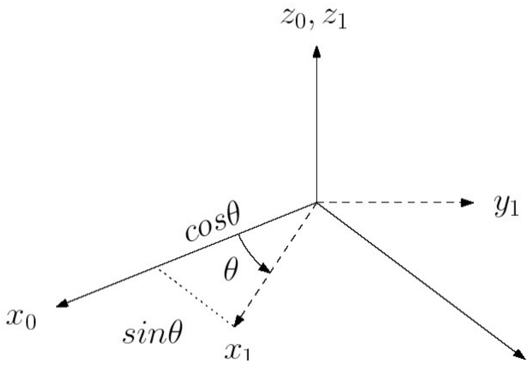

$$R_1^0 = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix}$$

# Table of Contents

# Representation of Rotations in 3D:

We need to project frame $\{1\}$ into frame $\{0\}$:

$$R_1^0 = \left[\ x_1^0 |\ y_1^0 |\ z_1^0\ \right] = \begin{bmatrix} \hat{x}_1.\hat{x}_0 & \hat{y}_1.\hat{x}_0 & \hat{z}_1.\hat{x}_0 \\ \hat{x}_1.\hat{y}_0 & \hat{y}_1.\hat{y}_0 & \hat{z}_1.\hat{y}_0 \\ \hat{x}_1.\hat{z}_0 & \hat{y}_1.\hat{z}_0 & \hat{z}_1.\hat{z}_0 \end{bmatrix}$$



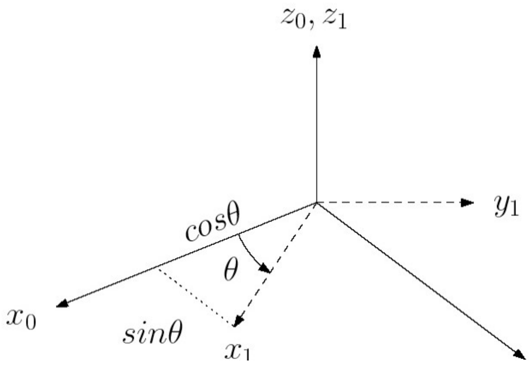
# Representation of Rotations in 3D:

We need to project frame $\{1\}$ into frame $\{0\}$:

$$R_1^0 = \left[\ x_1^0 |\ y_1^0 |\ z_1^0\ \right] = \begin{bmatrix} \hat{x}_1.\hat{x}_0 & \hat{y}_1.\hat{x}_0 & \hat{z}_1.\hat{x}_0 \\ \hat{x}_1.\hat{y}_0 & \hat{y}_1.\hat{y}_0 & \hat{z}_1.\hat{y}_0 \\ \hat{x}_1.\hat{z}_0 & \hat{y}_1.\hat{z}_0 & \hat{z}_1.\hat{z}_0 \end{bmatrix}$$

$$(R_1^0) = \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} = R_{z,\theta}$$

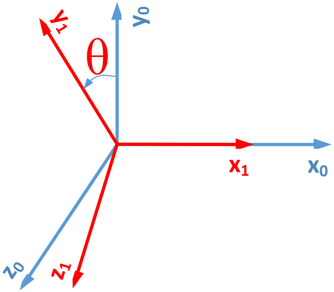
$R_{z,\theta}$ is the basic rotation matrix around z-axis.



# Representation of Rotations in 3D:

## Basic Rotation Matrices:

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



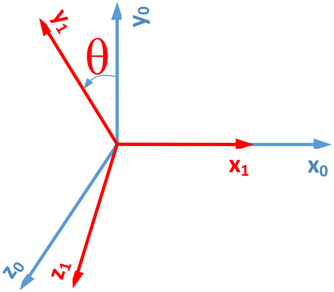

$R_{x,\theta}$

# Representation of Rotations in 3D:

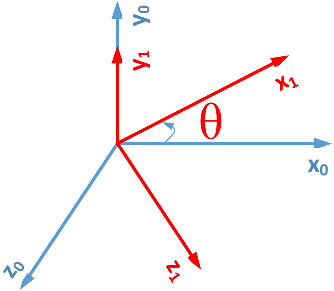
## Basic Rotation Matrices:

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

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




$R_{x,\theta}$




$R_{y,\theta}$

# Robot Operating System (ROS)

- **Problem:** Lack of standard for robots.
- **ROS:** is an open-source robot operating system:
  - A set of **software libraries** and **tools** that help you build robot applications that **work across a wide variety of robotic platforms**.
  - Originally developed in 2007 at the Stanford Artificial Intelligence Laboratory and development continued at Willow Garage.
  - Since 2013 managed by OSRF (Open Source Robotics Foundation).
  - ROS is working under Linux (Recommended Ubuntu).

ROS official page

# Questions?

haitham.elhussieny@feng.bu.edu.eg